pedocs
Open Access Erziehungswissenschaften

DIPF

Schneijderberg, Christian; Wieczorek, Oliver; Steinhardt, Isabel

Qualitative und quantitative Inhaltsanalyse. Digital und automatisiert. Eine anwendungsorientierte Einführung mit empirischen Beispielen und Softwareanwendungen

Weinheim : Beltz Juventa 2022, 483 S. - (Standards standardisierter und nichtstandardisierter Sozialforschung)

Quellenangabe/ Reference:
Schneijderberg, Christian; Wieczorek, Oliver; Steinhardt, Isabel: Qualitative und quantitative Inhaltsanalyse. Digital und automatisiert. Eine anwendungsorientierte Einführung mit empirischen Beispielen und Softwareanwendungen. Weinheim : Beltz Juventa 2022, 483 S. - (Standards standardisierter und nichtstandardisierter Sozialforschung) - URN: urn:nbn:de:0111-pedocs-292092 - DOI: 10.25656/01:29209

https://nbn-resolving.org/urn:nbn:de:0111-pedocs-292092
https://doi.org/10.25656/01:29209

in Kooperation mit / in cooperation with:

BELTZ JUVENTA

http://www.juventa.de

Nutzungsbedingungen

Dieses Dokument steht unter folgender Creative Commons-Lizenz: http://creativecommons.org/licenses/by-nc-nd/4.0/deed.de - Sie dürfen das Werk bzw. den Inhalt unter folgenden Bedingungen vervielfältigen, verbreiten und öffentlich zugänglich machen: Sie müssen den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen. Dieses Werk bzw. dieser Inhalt darf nicht für kommerzielle Zwecke verwendet werden und es darf nicht bearbeitet, abgewandelt oder in anderer Weise verändert werden.
Mit der Verwendung dieses Dokuments erkennen Sie die Nutzungsbedingungen an.

Terms of use

This document is published under following Creative Commons-License: http://creativecommons.org/licenses/by-nc-nd/4.0/deed.en - You may copy, distribute and transmit, adapt or exhibit the work in the public as long as you attribute the work in the manner specified by the author or licensor. You are not allowed to make commercial use of the work or its contents. You are not allowed to alter, transform, or change this work in any other way.

By using this particular document, you accept the above-stated conditions of use.

Kontakt / Contact:

pedocs
DIPF | Leibniz-Institut für Bildungsforschung und Bildungsinformation
Informationszentrum (IZ) Bildung
E-Mail: pedocs@dipf.de
Internet: www.pedocs.de

Mitglied der Leibniz-Gemeinschaft

Standards standardisierter und nichtstandardisierter Sozialforschung

Christian Schneijderberg | Oliver Wieczorek | Isabel Steinhardt

Qualitative und quantitative Inhaltsanalyse: digital und automatisiert

Eine anwendungsorientierte Einführung mit empirischen Beispielen und Softwareanwendungen

BELTZ JUVENTA

Christian Schneijderberg | Oliver Wieczorek | Isabel Steinhardt
Qualitative und quantitative Inhaltsanalyse: digital und automatisiert

Standards standardisierter und
nichtstandardisierter Sozialforschung

Herausgegeben von Nicole Burzan |
Paul Eisewicht | Ronald Hitzler

Christian Schneijderberg | Oliver Wieczorek |
Isabel Steinhardt

Qualitative und quantitative Inhaltsanalyse: digital und automatisiert

Eine anwendungsorientierte Einführung mit empirischen Beispielen und Softwareanwendungen

BELTZ JUVENTA

Die Autor_innen

Christian Schneijderberg, Dr., ist promovierter Soziologe. Seit 2009 forscht er am International Center for Higher Education Research (INCHER) und lehrt im Fach Soziologie an der Universität Kassel. Im akademischen Jahr 2020/21 hat Christian Schneijderberg die Professur „Soziologie, Methoden und Techniken der empirischen Sozialforschung“ in der Soziologie der RWTH Aachen vertreten.

Oliver Wieczorek, Dr. rer. pol., ist promovierter Soziologe und seit 2022 am International Center for Higher Education Research (INCHER) in Kassel angestellt. Zuvor war er an der Zeppelin Universität Friedrichshafen und an der Universität Bamberg beschäftigt.

Isabel Steinhardt, Prof. Dr., ist Professorin für Bildungssoziologie an die Universität Paderborn. Zuvor war Isabel Steinhardt wissenschaftliche Mitarbeiterin in der Soziologie und dem International Center for Higher Education Research (INCHER) der Universität Kassel.

Danke an den Open Access Publikationsfond der Universitätsbibliothek Kassel!

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Der Text dieser Publikation wird unter der Lizenz **Creative Commons Namensnennung – Nicht kommerziell – Keine Bearbeitungen 4.0 International (CC BY-NC-ND 4.0)** veröffentlicht. Den vollständigen Lizenztext finden Sie unter: https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode.de. Verwertung, die den Rahmen der **CC BY-NC-ND 4.0 Lizenz** überschreitet, ist ohne Zustimmung des Verlags unzulässig. Das gilt insbesondere für die Bearbeitung und Übersetzungen des Werkes. Die in diesem Werk enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Quellenangabe/Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.

Dieses Buch ist erhältlich als:
ISBN 978-3-7799-7036-1 Print
ISBN 978-3-7799-7037-8 E-Book (PDF)

1. Auflage 2022

© 2022 Beltz Juventa
in der Verlagsgruppe Beltz · Weinheim Basel
Werderstraße 10, 69469 Weinheim
Einige Rechte vorbehalten

Herstellung: Ulrike Poppel
Satz: text plus form, Dresden
Druck und Bindung: Beltz Grafische Betriebe, Bad Langensalza
Beltz Grafische Betriebe ist ein klimaneutrales Unternehmen (ID 15985-2104-100)
Printed in Germany

Weitere Informationen zu unseren Autor_innen und Titeln finden Sie unter: www.beltz.de

Inhalt

Danksagung		13
1.	**Einleitung**	15
1.1	Kurzübersicht zu den Inhalten der einzelnen Kapitel	16
1.2	Welche inhaltsanalytische Auswertungstechnik ist die geeignete Methode für Sie?	20
1.3	Copylefts und Copyrights von Software	26
2.	**Inhaltsanalyse von Kommunikation**	27
2.1	Zusammenhang von Kommunikation und Inhaltsanalyse	27
2.1.1	Textbasierte Inhaltsanalyse von Kommunikation	30
2.1.2	Analysedreischritt: Kontext-Verstehen, Inhalte-Verstehen und dem Publikum verständlich machen	31
2.2	Definition von qualitativer und quantitativer Inhaltsanalyse	33
2.2.1	Formen qualitativer Inhaltsanalyse	37
2.2.2	Quantitative Inhaltsanalyse: ein Überblick	41
2.2.3	Inhaltsanalytische Kombination von qualitativen und quantitativen Auswertungstechniken	44
2.3	Gütekriterien der Inhaltsanalyse	48
3.	**Spezifika von Daten: Möglichkeiten und Grenzen sozialwissenschaftlicher Inhaltsanalysen**	54
3.1	Einleitung	54
3.2	Merkmale von forschungs- und prozessproduzierten Daten	56
3.2.1	Forschungsproduzierte Daten: Datenerhebung	61
3.2.2	Prozessproduzierte Daten: Datentypen und Operationalisierung	64
3.2.3	Datenerhebung für wissenschaftliche Analysen mit prozessproduzierten Daten	68
3.3	Forschungsablauf und Gliederung wissenschaftlicher Ausarbeitungen	71
3.3.1	Schematische Darstellung: Forschung mit empirischen Daten	71
3.3.2	Gliederungsschema wissenschaftlicher Ausarbeitungen (z. B. Haus-, Bachelor-, Master- und Doktorarbeit)	71
3.4	Nachnutzung und Bereitstellung von Daten	74

4. Induktiv-qualitative Inhaltsanalyse 83
4.1 Einleitung 83
4.1.1 Methodische Anforderungen der induktiv-qualitativen Inhaltsanalyse 86
4.1.2 Forschungsprozess als Schritt für Schritt-Ablaufschema 88
4.2 Die Textdaten der induktiv-qualitativen Inhaltsanalyse 90
4.2.1 Entstehungskontext 90
4.2.2 Anonymisierung 91
4.2.3 Beispieltext: autoethnographischer Beitrag zum Studienbeginn Wintersemester 2020/21 93
4.3 Vorgehen der induktiv-qualitativen Inhaltsanalyse 101
4.3.1 Übersicht gewinnen: Inhalte von Textdokument(en) zusammenfassen 101
4.3.2 Strukturierende Analyse der Textdaten 102
4.3.3 Induktive Kategorienentwicklung 103
4.3.4 Und das Ganze noch n-Mal von vorn 106
4.3.5 Forschungspragmatische Entscheidungen zur Auswertung 107
4.3.6 Kodieren des Datenmaterials 108
4.4 Qualitative Inhaltsanalyse: manifeste und latente Inhalte verstehen und für Dritte verständlich machen 111
4.4.1 Erklären und Interpretieren als Teil der induktiv-qualitativen Inhaltsanalyse 111
4.4.2 Systematisches Verstehen durch Interpretation manifester und latenter Inhalte 112
4.4.3 Beispiele für Erklären, Interpretieren und theoriegeleitete Reflexion 113

5. Deduktiv-qualitative Inhaltsanalyse 120
5.1 Ablaufschema der deduktiv-qualitativen Inhaltsanalyse 120
5.2 Forschungsstand und Forschungsfrage: Schritt 1 124
5.3 Erstellung des deduktiven Kategoriensystems: Schritt 2 126
5.4 Vertraut machen mit dem Material: Schritt 3 129
5.5 Deduktive Kodierung: Schritt 4 131
5.6 Erweiterung des Kategoriensystems: Schritt 5 135
5.7 Zusammenführung der Ergebnisse: Schritt 6 140
5.8 Vergleich der Fälle 141

6. Induktiv-quantitative Inhaltsanalyse und Auswertungstechniken am Beispiel der Kombination von AntConc und MAXQDA 143
6.1 Einleitung 143
6.1.1 Begründung der Softwareverwendung 143
6.1.2 Erkenntnisziele der quantitativen Inhaltsanalyse 144
6.1.3 Forschungsprozess als Schritt für Schritt-Ablaufschema 146
6.2 Die Textdaten 148
6.2.1 Lehr-Lern-Forschungsprojekt Autoethnographie „Zwei Wochen Studium im Wintersemester 2020/21“ 148
6.2.2 Datenschutz und Einwilligungserklärung 149
6.2.3 Rohdaten und Datenbereitung 150
6.2.4 Informationen in umfangreichen Textkorpora finden 151
6.3 Schlag- bzw. Suchworte im Korpus mit AntConc identifizieren 154
6.3.1 Grundeinstellungen AntConc 154
6.3.2 Funktionen für die Identifikation von Schlagworten als Suchworte 155
6.3.3 Identifikation von Schlagworten als erster Schritt der Analyse 157
6.4 Quantitative Analyse mit MAXQDA 159
6.4.1 Dateien in MAXQDA importieren 159
6.4.2 Suchworte in Kategorien und Kodes überführen 160
6.4.3 Datenbereinigung 166
6.4.4 Kodes ordnen 169
6.4.5 Quantitative Ergebnisse als Präsentation manifester Inhalte und zur kodegeleiteten Auswahl für vertiefende Analysen 170

7. Deduktiv-quantitative Inhaltsanalyse: das Bibliometric Literature Review 177
7.1 Einleitung 177
7.2 Schwächen von Datenbanken und Suchmaschinen 181
7.2.1 Google Scholar 183
7.2.2 CrossRef 186
7.2.3 ResearchGate 186
7.2.4 Web of Science und Scopus 187
7.3 Schritt 1: Erkenntnisinteresse als Fragestellung formulieren 188
7.3.1 Literaturüberblick 188
7.3.2 Mapping 189
7.3.3 Themenanalyse 190
7.4 Schritt 2: Auswahl der Datenbank und Suchfokus 190
7.4.1 Literaturüberblick 190
7.4.2 Mapping 191
7.4.3 Themenanalyse 192

7.5 Schritt 3: Datenauswahl und Datenbereinigung 193
7.5.1 Literaturüberblick 193
7.5.2 Mapping und Themenanalyse 196
7.6 Schritt 4: bibliometrische Analysen und Inhaltsanalyse 196
7.6.1 Literaturüberblick 196
7.6.2 Mapping 200
7.6.3 Themenanalyse 206
7.7 Schritt 5: Interpretation der Ergebnisse 208
7.7.1 Literaturüberblick 208
7.7.2 Mapping 208
7.7.3 Themenanalyse 209

8. Automatisierte induktiv-quantitative Inhaltsanalyse: Datenerhebung und -vorbereitung 210
8.1 Einleitung 210
8.2 Typen automatisierter Verfahren der quantitativen Textanalyse 214
8.2.1 Maschinelles Lernen: Definition und Anwendungsgebiete 214
8.2.2 Ziele von Verfahren maschinellen Lernens im Bereich der automatisierten, quantitativen Textanalyse 215
8.2.3 Beispiele für die Funktionsweise der Verfahren der automatisierten quantitativen Textanalyse 216
8.2.4 Textkorpora und geeignete Datengrundlagen für die Erstellung eines eigenen Textkorpus 217
8.2.5 Datenzugänge 219
8.2.6 Technische Umsetzung der Online-Datenerhebung durch Webscraping 220
8.3 Aufbereitung der Daten 223
8.3.1 Vereinheitlichung der Datenstruktur 223
8.3.2 Typen von und Umgang mit fehlerhaften Daten 224
8.3.3 Textaufbereitung 225
8.4 Wo finde ich online Hilfe? 226

9. Quantitative Inhaltsanalyse mittels Korrespondenzanalyse 228
9.1 Einleitung 228
9.1.1 Kommunikation als Textdaten in einer Matrix 229
9.1.2 Schritt für Schritt-Ablauf einer Korrespondenzanalyse 231
9.2 Einführung in RStudio 232
9.2.1 Installation von R und von RStudio 232
9.2.2 Aufbau von RStudio 234
9.3 Vorbereitende Schritte für die Korrespondenzanalyse in RStudio 238

9.3.1 Software-„Pakete“ in RStudio importieren und aktivieren 238
9.3.2 Pakete installieren 239
9.3.3 Pakete in RStudio laden 240
9.3.4 Dateien einlesen 242
9.3.5 Auswahl der Variablen für die Analyse 243
9.3.6 Verwendung regulärer Ausdrücke und Exklusion fehlender Werte 246
9.4 Durchführung einer Korrespondenzanalyse 247
9.4.1 Test/Voraussetzungen für die Durchführung einer Korrespondenzanalyse 247
9.4.2 Durchführung der Korrespondenzanalyse 249
9.4.3 Auswahl der Dimensionszahl für die spätere Interpretation 252
9.4.4 Speichern von Grafiken 256
9.5 Auswertung der Korrespondenzanalyse 258
9.5.1 Erzeugung der Grafiken für die Interpretation der Ergebnisse 258
9.5.2 Interpretation der ersten beiden Dimensionen 260
9.5.3 Interpretation der Dimension 3 265
9.5.4 Interpretation der Dimension 4 268
9.5.5 Erkunden und Exportieren der durch die Korrespondenzanalyse erzeugten Informationen 271
9.6 Schlussworte 276

10. Sentiment-Analyse als induktiv-quantitative Inhaltsanalyse 278
10.1 Einleitung 278
10.1.1 Schritt für Schritt-Ablauf einer Sentiment-Analyse 280
10.1.2 Freude? Angst? Welches Gefühl möchten Sie erforschen? 281
10.1.3 Datengrundlage 282
10.2 Sentiment-Analyse in R 282
10.2.1 Verwendete Pakete in RStudio 283
10.2.2 Datenbereinigung 284
10.2.3 Durchführung der Sentiment-Analyse 295
10.3 Sentiment-Analyse in Python 314
10.3.1 Python, Spyder und Packages 314
10.3.2 Spyder-Benutzeroberfläche 316
10.3.3 Ausführen von Befehlen und Überblick über verschiedene Datentypen 326
10.3.4 Einlesen von Daten in Spyder 329
10.3.5 Daten aufbereiten 331
10.3.6 Ausführung der Sentiment-Analyse 341
10.4 Zusammenfassung und abschließende Worte 356

11. Topic Modeling mittels Latent Dirichlet Allocation 358
11.1 Einleitung 358
11.1.1 Schritt für Schritt-Ablauf des Topic Modelings 359
11.1.2 Daten und Forschungsfrage 360
11.2 Topic Modeling mit Python 361
11.2.1 Struktur und Tücken von Filmskripten als Datenmaterial 361
11.2.2 Fehlerbereinigung 364
11.2.3 Benötigte Pakete 370
11.2.4 Pakete und Daten einlesen 371
11.3 Aufbereitung der Daten für die Analyse 383
11.3.1 Text tokenisieren 384
11.3.2 Festlegen der Stopwords 384
11.3.3 Entfernen von Stopwords und Wortfragmenten 387
11.3.4 Beschränkung der Wörter auf Nomen, Verben und Adjektive 389
11.3.5 Lemmatisierung und Stemming der Filmskripte 391
11.3.6 Schritte 1 bis 6 der Datenaufbereitung in einem Python-Programmskript zusammenfassen 392
11.3.7 Wie die Maschine lernt, Wissenschaft auszusprechen 394
11.3.8 Lexikon erzeugen und zu seltene bzw. zu häufige Wörter entfernen 400
11.3.9 Korpus im „bag of words"-Format erzeugen 403
11.3.10 Wörter im Korpus gewichten 404
11.4 Durchführung einer Latent Dirichlet Allocation 405
11.4.1 Benötigte Pakete laden 406
11.4.2 Die Wort-Themen-Assoziationen: ein „Gefühl" für die Daten bekommen 410
11.4.3 Berechnung der Modellkohärenz 416
11.4.4 Berechnung einer Vielzahl von Topic Models mit for-Schleife 418
11.5 Auswertung der Topic Models und Interpretation der Ergebnisse 422
11.5.1 Perplexity- und Coherence-Scores: die softwaregesteuerte Maschine hilft beim Lesen, ein interpretierbares Modell wählen wir aus 423
11.5.2 Sichtung der Themen und Visualisierung über Themen, Texte und Zeitpunkte hinweg 426
11.5.3 Nächste Schritte zum Verständnis der Daten: Themen verstehen 445
11.5.4 Daten noch besser verstehen 449
11.5.5 Datenbasierte Entscheidungen für die Analyse treffen 453

11.5.6 Latente Inhalte an den Beispielen Them! und X-Men erklären, deuten und interpretieren 456
11.6 Zusammenfassung und abschließende Worte 459

12. Die Schlussworte: keine Angst vor Daten, Software und Interpretation 462
12.1 Keine Datenanalyse ohne Interpretation 464
12.2 Analysedreischritt: Kontext-Verstehen, Inhalte-Verstehen und dem Publikum verständlich machen 467

Literatur 469

Autor*innenvorstellung 483

Danksagung

Wir möchten uns herzlich bei den Reihenherausgeber*innen, Nicole Burzan, Paul Eisewicht und Ronald Hitzler, für die Einladung bedanken, das vorliegende Methodenbuch „Qualitative und quantitative Inhaltsanalyse: digital und automatisiert“ zu verfassen und zu veröffentlichen. Von der Einladung im Oktober 2019 bis zur Veröffentlichung vergingen mehr als zwei Jahre. Obwohl wir drei eng miteinander arbeitende Autor*innen sind, kann ein so vielseitiges Werk und dessen Inhalte nicht ohne weitere Unterstützung gelingen.

Für inhaltliches Feedback zu verschiedenen Kapiteln gilt unser Dank Matthias Philipper und Ronny Röwert. Die Vermittlung von Inhalten in einem Lehrbuch war/ist dabei stets ein Spagat zwischen begrifflicher Klarheit, systematischen Methoden- und Analysebeschreibungen und softwaretechnischen Erklärungen sowie deren Darstellung in möglichst gut verständlicher Sprache (inklusive aller Kommata!). Für die Unterstützung bei der sprachlichen Überarbeitung möchten wir uns bei Christiane Rittgerott, Sarina Rohr und Svenja Dilger bedanken. Frank Engelhardt von Beltz Juventa gilt unser Dank für seine stets raschen und sachlichen Antworten sowie Geduld, insbesondere im Prozess des umfangreichen Publikationsvertragverstehens. Zuletzt möchten wir uns bei der Universitätsbibliothek Kassel bedanken, deren Publikationsfond das Erscheinen dieses Buches in Open Access ermöglicht.

Christian Schneijderberg, Oliver Wieczorek und Isabel Steinhardt
Kassel und Paderborn, Mai 2022

1. Einleitung

Zur Inhaltsanalyse gibt es bereits diverse Methodenbücher. Warum sollten Sie also genau das vorliegende Methodenbuch „Qualitative und quantitative Inhaltsanalyse: digital und automatisiert" lesen? Erstens schließt dieses Buch eine Lücke, der wir in der Lehre und bei Methodenworkshops immer wieder begegnet sind. Diese Lücke besteht darin, dass selten eine Übersicht über und Gegenüberstellung verschiedener Methoden der qualitativen und quantitativen Inhaltsanalyse gegeben wird. Zudem haben bisherige Lehr- und Methodenbücher in den Sozialwissenschaften teil- und vollautomatisierte Verfahren der Textanalyse noch nicht aufgegriffen und diese mit bereits etablierten Verfahren qualitativer und quantitativer Inhaltsanalyse verknüpft.

Zweitens bietet dieses Buch im ersten Teil eine systematische und anwendungsorientierte Einführung zu den Grundlagen inhaltsanalytischer empirischer Sozialforschung. Im zweiten Teil werden detaillierte Anleitungen von digital unterstützten qualitativen inhaltsanalytischen Auswertungstechniken gegeben. Den teil- und vollautomatisierten quantitativen inhaltsanalytischen Auswertungstechniken widmet sich der dritte Teil des Buches. Bei digital unterstützten Verfahren werden Sie durch Software bei Ihrer Analyse unterstützt, beispielsweise bei der digitalen Organisation Ihrer Daten und bei der digitalen Durchführung der Auswertung. Bei teilautomatisierten Verfahren der Inhaltsanalyse nehmen Ihnen Software und Tools Teile der Datenanalyse ab, wohingegen bei vollautomatisierten Verfahren der gesamte Auswertungsprozess und oftmals auch die Datenbeschaffung durch Programme und Tools erfolgt. Das bedeutet keinesfalls, dass die vollautomatisierte Inhaltsanalyse schneller oder leichter von der Hand geht als die digital unterstützte Inhaltsanalyse. Sie benötigen für die unterschiedlichen Verfahren unterschiedliche Kenntnisse und für unterschiedliche Anwendungsschritte der Verfahren unterschiedlich viel Zeit. Am Ende sind alle Verfahren gleich zeitintensiv und aufwendig. Das zeigen wir Ihnen noch anhand einer von uns erstellten Heuristik.

Drittens sind die Kapitel so aufgebaut, dass sie jeweils einzeln gelesen werden können. Jedes Kapitel erklärt ein inhaltsanalytisches Verfahren Schritt für Schritt an anwendungsorientierten, empirischen Beispielen. Die Kapitel haben einen Umfang von 17 bis 66 Seiten reiner Anleitung. Durch Empirie und Code werden die Seitenzahlen umfangreicher, was aber jeweils kein Indiz dafür ist, ob Verfahren kompliziert sind oder nicht. Sie können die einzelnen Kapitel für das individuelle Methodenlernen ebenso wie für die Methodenlehre verwenden. Beim Lesen wird Ihnen auffallen, dass sich die einzelnen Kapitel in der Sprache und Konzeption unterscheiden. Dies liegt daran, dass die Kapitel unterschiedliche Hauptautor*in-

nen hatten. Und doch ist das Buch eine Gemeinschaftsarbeit der Autor*innen, da alle Kapitel ausführlich miteinander diskutiert und redigiert wurden.

Information über Methoden wird, viertens, durch die Anleitung zur Nutzung von Auswertungssoftware unterstützt. Dazu werden Einführungen in die Software AntConc, MAXQDA, Python, RStudio und VosViewer gegeben und anhand von empirischen Beispielen erläutert. Damit Sie die Möglichkeiten und Grenzen sowohl von Auswertungssoftware als auch unterschiedlichen Auswertungstechniken der Inhaltsanalyse als Methode kennenlernen können, wenden wir unterschiedliche inhaltsanalytische Auswertungstechniken auf dieselben empirischen Daten an. Zusatzmaterialen finden Sie zudem auf dem Blog „sozmethode" (siehe Box 1.1). Der Blog soll gleichzeitig auch als Austauschmöglichkeit und Weiterentwicklungsplattform für die vorgestellten Verfahren dienen. Insofern möchten wir alle ermutigen, uns methodische Reflexionen zu Ihrer Anwendung der unterschiedlichen Verfahren der Inhaltsanalyse zukommen zu lassen, die wir gerne, nach Prüfung, auf dem Blog veröffentlichen. Denn methodische Reflexionen aus unterschiedlichen Forschungskontexten können für andere sehr hilfreich sein. Also bitte: Seien Sie mutig und melden sich bei uns!

Box 1.1: Weitere Materialien, Informationen und Anwendungsbeispiele online
Zudem stellen wir Ihnen auf dem Blog *https://sozmethode.hypotheses.org/methodenbuch* Anleitungen und Weiterentwicklungen sowie Probecodes zur Verfügung. Schauen Sie gerne auch dort einmal vorbei, um sich Hilfestellungen zu holen.

Wir möchten Sie gerne ermutigen, Ihre eigenen Anwendungsbeispiele, die durch Nutzung dieses Buches entstanden sind, auf dem Blog zu veröffentlichen. Was veröffentlicht werden kann und wie das Prozedere ist, erfahren Sie auf dem Blog. Wir freuen uns auch über Weiterentwicklungen der Verfahren oder Code und sind sehr an einem Austausch mit Ihnen interessiert.

1.1 Kurzübersicht zu den Inhalten der einzelnen Kapitel

Nach der Erklärung der Systematik des Buchaufbaus werden die Inhalte der einzelnen Kapitel knapp vorgestellt.

Kapitel 2: Überblick der qualitativen und quantitativen Inhaltsanalyse

In diesem Kapitel definieren wir, was wir unter Inhaltsanalyse verstehen, was Inhaltsanalyse mit Kommunikation zu tun hat, was manifeste und latente Inhalte sind und welche Varianten der Inhaltsanalyse in diesem Buch behandelt werden. Zudem geben wir Ihnen einen kurzen Überblick über Varianten der qualitativen Inhaltsanalyse, die unserer Auffassung nach in induktive und deduktive qualitative Inhaltsanalyse eingeteilt werden kann. Darüber hinaus geben wir einen Überblick über Verfahren der quantitativen Inhaltsanalyse. Sie lernen zudem zwei Regeln kennen: die Anschlussregel und die Ausschlussregel qualitativer

und quantitativer Sozialforschung, die Ihnen dabei helfen, Ihre Forschung einzuordnen. Am Ende des Kapitels geben wir Ihnen noch einen Überblick über Gütekriterien in Bezug auf die Inhaltsanalyse.

Kapitel 3: Spezifika von Daten – Möglichkeiten und Grenzen sozialwissenschaftlicher Inhaltsanalysen

In Kapitel 3 werden die Spezifika von Daten und damit verbundene Möglichkeiten und Grenzen für sozialwissenschaftliche Inhaltsanalysen forschungspragmatisch und erkenntnisorientiert dargestellt. Unterschieden werden hierbei die Qualitäten von forschungs- und prozessproduzierten Daten von der Konzeption (unter Berücksichtigung von Datentypen, Forschungszielen und der Operationalisierung) über die Datengenese (inklusive Stichprobenziehung, Datenerhebung und Datenart), Analysemöglichkeiten (aufgrund von Datenverwahrung und wissenschaftlicher Analysearten) und Ergebnisverwertung bis zur Archivierung von Daten. Die Spezifika von empirischen Daten prägen nicht nur den Forschungsprozess, sondern auch die (schriftliche) Präsentation der Erkenntnis. Diese wird abschließend in einem groben Gliederungsschema wissenschaftlicher Ausarbeitungen zusammengefasst, welches von der Haus- über Bachelor- und Master- bis zur Doktorarbeit angewandt werden kann.

Kapitel 4 und 5: Induktiv-qualitative Inhaltsanalyse und deduktiv-qualitative Inhaltsanalyse

Die zwei Kapitel zur digital unterstützten qualitativen Inhaltsanalyse präsentieren die grundlegenden Unterschiede zwischen induktiv-qualitativer Inhaltanalyse (Kapitel 4) und deduktiv-qualitativer Inhaltsanalyse (Kapitel 5) anhand desselben Beispiels. Das Beispiel-Datenmaterial ist eine autoethnographische Aufzeichnung, gemeinhin auch bekannt als Tagebucheinträge, einer*eines Studierenden zum Corona-Pandemie bedingten Online-Wintersemester 2020/21. So können Sie die Ziele der jeweiligen Auswertungstechnik anwendungsorientiert nachvollziehen. Zudem erklären wir Ihnen die Gemeinsamkeiten und Unterschiede von induktiven und deduktiven Datenanalysen und den jeweiligen Erkenntnisgewinn. Bei der induktiv-qualitativen Inhaltsanalyse werden die Kodes (siehe Box 1.2) zur Datenanalyse aus dem jeweiligen Datenmaterial gewonnen. Im Gegensatz dazu werden bei der deduktiven Auswertungstechnik die Kodes für die Datenanalyse vor der Datenauswertung festgelegt. Systematisch erfolgt die deduktive Erstellung der Kodes auf Basis von Theorie, konzeptionellen Überlegungen und/oder existierenden Erkenntnissen aus vorhergegangenen empirischen Untersuchungen. Selbstverständlich muss die Kode-Genese dem zu untersuchenden Gegenstand angemessen sein, und, wie auch

Box 1.2: Unterschied von Kode und Code
Zur sprachlichen Klarheit verwenden wir die Schreibweise von Kode mit K für sämtliche Aspekte des Kodierens der qualitativen und teilautomatisierten Inhaltsanalyse. Die Schreibweise Code mit C wird für das Programmieren von Befehlen in Python und R verwendet.

bei der induktiven Datenauswertung, erforschen Sie ja nicht zufällig ein Thema, sondern Ihr Forschungsinteresse ist durch vorherige Studien, Beobachtung, Reflexion eines sozialen Phänomens usw. beeinflusst und durch eine oder mehrere Forschungsfragen definiert.

Kapitel 6: Induktiv-quantitative Inhaltsanalyse und Auswertungstechniken am Beispiel der Kombination von AntConc und MAXQDA

In Kapitel 6 wird eine Auswertungstechnik vorgestellt, mit der Sie 500 Seiten und mehr Text induktiv-quantitativ auswerten können. Diese Technik ist geeignet für größere Interviewstudien und Sekundärauswertungen von prozessproduzierten Daten (z. B. Wahlprogramme oder Twitter). Am oben bereits beschrieben Beispiel der Autoethnographien wird die sequentielle Auswertung mithilfe der Software AntConc (Freeware; Sozio-/Politolinguistik) und MAXQDA (Firmware; lexikalische Suche und Autocodierung) erklärt. Geleitet durch Erkenntnisinteresse und Forschungsfrage(n) wird Schritt für Schritt erklärt, wie der Textkorpus über Suchworte systematisch erschlossen wird, und wie darin gefundene Informationen ausgewertet werden. Die teilautomatisierte Erschließung des Textkorpus und induktiv-quantitative Inhaltsanalyse produziert einen über die Suchworte bzw. Schlagworte kategorisierten Überblick, welcher auch als *distant reading* (Moretti 2000; 2013) bezeichnet wird. Im Gegensatz zur qualitativen Inhaltsanalyse ist das Ziel der induktiv-quantitativen Inhaltsanalyse, die manifesten Inhalte zu erfassen, sie zu deuten, daraus Erkenntnisse zu gewinnen und Schlüsse zu ziehen.

Kapitel 7: Deduktiv-quantitative Inhaltsanalyse – das Bibliometric Literature Review

In diesem Kapitel stellen wir Ihnen das Bibliometric Literature Review vor. Dabei handelt es sich um ein Verfahren der deduktiv-quantitativen Inhaltsanalyse, mit dessen Hilfe Sie einen systematischen Überblick über Publikationen erhalten. Als Grundlage wird zunächst erläutert, was Publikationen eigentlich sind und worauf die systematische Analyse von Literatur beruht: den Zitationen. Um die Analysen durchführen zu können, müssen Sie auf Datenbanken zurückgreifen, weshalb eine Auswahl an Datenbanken und deren Vor- und Nachteile dargestellt werden. Mit diesen Grundlagen werden dann drei Forschungsfragen vorgestellt, die mit dem Bibliometric Literature Review bearbeitet werden: Erstens der Literaturüberblick und die Frage: Was sind die zentralen Publikationen zu einem Thema bzw. in einem Forschungsfeld? Zweitens das Mapping und die Frage, wie ein Forschungsfeld (zu einem bestimmten Thema) aufgebaut ist, d. h. welche Zusammenhänge sich finden lassen. Und drittens, die Themenanalyse, also welche Inhalte in einem Forschungsfeld bzw. zu einem Thema diskutiert werden?

Kapitel 8: Automatisierte induktiv-quantitative Inhaltsanalyse – Datenerhebung und -vorbereitung

Dieses Kapitel stellt die verschiedenen Typen von online verfügbaren Daten und die Schritte zur Erhebung dieser Daten vor (Webscraping usw.). Darüber hinaus werden verschiedene Typen automatisierter quantitativer Inhaltsanalyse, Probleme und Lösungen für diese Probleme besprochen. Zu den Problemen zählen der Umgang mit fehlenden, fehlerhaften, irrelevanten oder duplizierten Daten, der Umgang mit dem Datenzugang, automatisierte Datenerhebung und eventuelle rechtliche Fallstricke. Dabei liegt der Fokus auf Webscraping, es werden aber auch andere Datenzugänge wie Application Programming Interfaces (APIs) und Online-Repositorien vorgestellt.

Kapitel 9: Quantitative Inhaltsanalyse mittels Korrespondenzanalyse

Das Kapitel soll Ihnen einen Überblick über die Korrespondenzanalyse und über die Schritte geben, die nötig sind, damit Sie dieses Verfahren in RStudio anwenden können. Dazu wird zunächst ein Kurzabriss über die Grundidee der Korrespondenzanalyse geboten. Danach wird erklärt, wie Sie R und RStudio installieren können und wie die Benutzeroberfläche aufgebaut ist. Es folgt eine Erläuterung, was Pakete sind, wie man diese installiert und in die Arbeitsoberfläche lädt. Im nächsten Schritt werden Befehle zur Aufbereitung der Daten vorgestellt, ehe Befehle beschrieben werden, mit deren Hilfe Sie die Korrespondenzanalyse durchführen. Das Kapitel schließt mit der Interpretation von Analysen und der Erläuterung, wie Sie diese Ergebnisse exportieren. In diesem Kapitel erwarten Sie circa sechs Seiten Code, die Auswertung empirischer Ergebnisse sowie zehn Abbildungen.

Kapitel 10: Sentiment-Analyse als induktiv-quantitative Inhaltsanalyse

In diesem Kapitel bieten wir Ihnen eine Einführung in die Sentiment-Analyse in R und Python. Hierfür gehen wir zunächst auf die Grundlagen der Sentiment-Analyse ein und bieten Ihnen einen historischen Abriss über deren Entwicklung. Danach zeigen wir Ihnen Aufbereitungsschritte und die Durchführung in R und stellen dabei die benötigten Pakete vor. Danach fahren wir mit der Aufbereitung für und der Durchführung der Sentiment-Analyse in Python fort. Zuletzt zeigen wir Ihnen, wie Sie die Sentiment-Werte für unterschiedliche Gruppen vergleichen können. In dem vorliegenden Kapitel werden circa zwölf Seiten für die Vorstellung von Programmier-Codes in R und Python verwendet.

Kapitel 11: Topic Modeling mittels Latent Dirichlet Allocation

Dieses Kapitel hat das Ziel, Ihnen eine Einführung in das Topic Modeling mit dem Python-Paket *gensim* zu geben. Wir fokussieren uns dabei auf die Latent Dirichlet Allocation (LDA) als eine Methode des Topic Modelings und zeigen Ihnen am Beispiel eines Korpus aus Filmskripten, wie Sie die Daten aufbereiten

müssen, wie eine LDA technisch umgesetzt wird, welche Kriterien Sie für die Modellauswahl anlegen, wie Sie die Themen interpretieren und wie Sie das Modell visualisieren können. In diesem Kapitel werden circa 16 Seiten für die Syntax aufgewandt und Vorgehen sowie Output in neun Abbildungen und 18 Tabellen visualisiert.

Kapitel 12: Die Schlussworte – keine Angst vor Daten, Software und Interpretation

Am Ende des Buches stellen wir vier Punkte heraus, die Ihnen als Orientierung für Ihre Forschung und die Anwendung textanalytischer Verfahren dienen können. Zudem geben wir Ihnen drei Tipps für die Interpretation Ihrer Daten und schließen das Buch mit dem Analysedreischritt, den wir in Kapitel 2 ausführlich erläutern.

1.2 Welche inhaltsanalytische Auswertungstechnik ist die geeignete Methode für Sie?

Sie haben nun einen ersten Überblick über das vorliegende Lehrbuch erhalten und stehen eventuell vor der Frage, mit welchem Kapitel Sie beginnen sollen. Deshalb laden wir Sie ein, sich mit einer von uns entwickelten Heuristik ein erstes Bild der Inhaltsanalyse zu machen. Sie können dabei anhand einfacher Parameter entscheiden, welches Verfahren für Sie in Betracht kommen könnte (siehe Abbildung 1.1). Die Heuristik ist allerdings nur eine erste grobe Orientierung, damit Sie entscheiden können, welches Kapitel in diesem Buch für Sie das wichtigste ist. Bevor Sie sich jedoch einer bestimmten inhaltsanalytischen Auswertungstechnik zuwenden, empfehlen wir Ihnen die Lektüre der beiden einführenden Kapitel zur Methode der Inhaltsanalyse (Kapitel 2) und Möglichkeiten und Grenzen von Daten (Kapitel 3). Diese Empfehlung gilt vor allem dann, wenn Sie zum ersten Mal empirisch arbeiten und/oder die Inhaltsanalyse anwenden.

In der empirischen Sozialforschung existiert eine große Zahl an Methoden. Entsprechend ist es notwendig, dass Sie forschungspragmatisch eine geeignete Methode für Ihre Untersuchung auswählen. Die Methode sollten Sie stets mit Blick auf Ihr Forschungsinteresse und dafür verfügbare oder zu erhebende Daten auswählen.

1. Erkenntnisinteresse der Forschung

a) Tiefe der Analyse: Die Kategorie „Tiefe der Analyse" zielt auf das vertiefte, qualitative und interpretierende Verstehen von empirischen Daten ab. Damit ist zugleich das Herausarbeiten von Sinnebenen und textimmanenten und über die Texte hinausreichende Bedeutungen verbunden. Das ist eine zeitaufwändige Angelegenheit, die genaues Lesen, eingehende Reflexion und aufeinander auf-

bauende Analysen und Interpretationen von Abschnitten, Sinnebenen und Verweisen erfordert. Dies lässt sich keinesfalls durch einfaches und einmaliges Lesen Ihres Textmaterials verwirklichen, welches hier sinnbildlich die „Oberfläche" der Sinnstruktur Ihrer Texte darstellt. Daher können Sie auch nur einen kleinen Teil sozialer Realität und einen Ausschnitt eines Phänomens untersuchen. In der empirischen Sozialforschung werden solche tiefgehenden Analysen entweder mit Einzelfallanalysen, beispielsweise einem Interview oder einer ethnographischen Beobachtung, oder mit der Analyse weniger Texte und Untersuchungseinheiten wie Organisationen in Verbindung gebracht. Daher wird in der englischsprachigen Methodenliteratur auch von *small-n-studies,* auf Deutsch „Klein-n-Untersuchungen" gesprochen (Ebbinghaus 2009; Schneijderberg und Götze 2021; Smelser 2003). In der Methodenliteratur gilt es auch als „ein Fall", wenn eine Organisation mithilfe von Interviews untersucht oder eine Nation anhand von statistischen Daten oder Dokumenten analysiert wird.

b) Breite der Analyse: Im Gegensatz zum vertieften Verstehen und Interpretieren von empirischem Datenmaterial bei Klein-n-Studien, zielen Groß-n-Studien (Englisch: *large-n-studies*) auf die Analyse von vielen empirischen Fällen ab. Es gibt keine festgelegte Abgrenzung zwischen Klein-n- und Groß-n-Studien. Grob gesagt, können Sie von einer qualitativen Groß-n-Studie bei mehr als 25 Interviews bzw. bei kürzeren Interviews ab einem n von 35 ausgehen. In der Statistik können Sie je nach Methode ab 80 bis 120 und mehr Fällen von einer Groß-n-Studie ausgehen.[1] Ziel dieser Studien ist es in der Regel, abstrakte, allgemeingültige Gesetzmäßigkeiten oder Zusammenhänge zwischen Konzepten und Phänomenen zu finden. In unserem Falle ist das Ziel, Muster in gesprochener Sprache oder geschriebenen Text(en) zu finden, die genutzt werden können, um Sinnzuschreibungen, Interpretationsleistungen oder die Stimmungslage vieler Personen zu rekonstruieren.

In gewisser Weise können Sie sich die Balance zwischen Breite und Tiefe der Erkenntnis wie einen Lautstärkeregler vorstellen. Je größer die Fallzahl und Breite des Erkenntnisinteresses, desto kleiner ist das Potenzial jedes einzelnen Falles, neue Erkenntnis zu ermöglichen und desto kleiner ist auch dessen empirische Bedeutung. Parallel zur Abnahme der Bedeutung einzelner empirischer Fälle nimmt auch die Durchdringungstiefe des Datenmaterials ab – ein Wassertropfen fällt in einem See genauso wenig auf wie die Antwort eines Untersuchungssubjektes als Anteil am Durchschnittswert bei einer Anzahl von 7 283 Antworten.

---

1 Bei statistischen Berechnungen mit Mehrebenenmodellen kann die Varianz bereits ab circa 20 Fällen auf der ersten Ebene und mehr als 10/12 Fällen auf der zweiten Ebene modelliert werden (Rabe-Hesketh und Skrondal 2012).

2. Datenumfang

Der Begriff Datenumfang wurde hier gewählt, da es keinen direkten Zusammenhang der Anzahl von Dokumenten, der darin befindlichen Textmenge sowie dem Aufwand gibt, der zur Analyse der Texte nötig ist. Zur Einschätzung des Datenumfangs müssen Sie daher die drei Komponenten Datenmenge, Samplegröße und Sampling-Prozess und damit auch die Zeit berücksichtigen, die Sie für die Sichtung und Analyse der einzelnen Texte überhaupt in Ihrem Forschungsprozess veranschlagen können. Die Kombination aus allen drei Faktoren kann den Zeitaufwand sehr schnell anwachsen lassen, den Sie für Ihren gesamten Forschungsprozess einplanen müssen und damit auch, wann Sie eventuell davon absehen sollten, alle Interviews oder Textdaten erheben oder auswerten zu wollen.

a) Datenmenge: Je nachdem, welchen Fall Sie untersuchen wollen, kann die Datenmenge sehr unterschiedlich sein. Nehmen Sie z. B. ein Interview, das circa eine Stunde gedauert hat, als einen Fall. Das einstündige Interview ergibt ein Transkript (also die Verschriftlichung des Interviews) von etwa 50 Seiten, bei 1 500 Zeichen (exklusive Leerzeichen) pro Seite. Im Gegensatz dazu würde ein Tweet als ein Fall nur maximal 240 Zeichen umfassen. Sie sehen also, dass der Umfang pro Fall sehr unterschiedlich groß sein kann, was dazu führt, dass Sie sich in Bezug auf Breite und Tiefe der Analyse überlegen müssen, was von Ihnen leistbar ist. So entspräche beispielsweise der Umfang von etwa 312,5 Tweets dem Umfang eines einstündigen Interviews. Möglich wäre in diesem Fall die Analyse des einstündigen Interviews mit der qualitativen Inhaltsanalyse. Für ein automatisiertes Verfahren der Inhaltsanalyse wie zum Beispiel dem Topic Modeling würden Sie allerdings mehr Daten benötigen. Hier könnten Sie beispielsweise auf Drehbücher zugreifen, wie wir das anhand von 626 Drehbüchern zu Filmen mit Repräsentation von Wissenschaft gemacht haben. Mit durchschnittlich 70 Seiten pro Drehbuch wird ein Datenumfang von fast 39 000 Seiten analysiert. Die Daumenregel lautet hier: Je mehr Datenmaterial Sie haben, umso mehr Sinn macht es, eine Analyseform zu wählen, die auf die Breite der Analyse und nicht auf die Tiefe fokussiert.

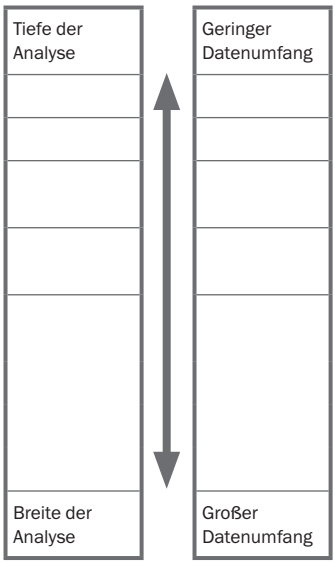
b) Samplegröße: Mithilfe von Auswahlverfahren, sogenannten Sampling-Strategien, können Sie die ursprünglich vorhandene Datenmenge reduzieren und dadurch eine tiefgehende Analyse durchführen. Beispielsweise könnten Sie aus 50 Interviews nur die Antworten zu einer spezifischen Frage auswerten oder Sie wählen nach bestimmten Kriterien eine Anzahl Filmdrehbücher aus der Grundgesamtheit von 626 Filmdrehbüchern aus. Durch die Reduktion der Datenmenge können Sie dann eine qualitative Inhaltsanalyse durchführen, die zu einem bestimmten Teilaspekt der Interviews eine Antwort liefert. Dabei sollten Sie aber auch bedenken, Ihre Fallauswahl zu dokumentieren und nach nachvollziehbaren Regeln zu gestalten.

c) Sampling-Prozess: Der Sampling-Prozess selbst kann nämlich sehr zeitaufwändig sein und sehr viele Überlegungen und Entscheidungen Ihrerseits erfordern! So kann es zum Beispiel für Sie oder im Rahmen ihres angelegten theoretischen Rahmens interessant sein, Studierende unterschiedlicher Fachsemester und Geschlechter hinsichtlich ihrer Einbettung ins Studienleben zu untersuchen. Ist dies der Fall, dann sollten Sie sich überlegen, ob Sie die einzelnen Gruppen (Semester x Geschlecht) in Ihren Interviews anteilig repräsentieren möchten, oder ob die Gruppen, die insgesamt geringer an der Studierendenschaft vertreten sind, in Ihren Interviews stärker repräsentiert werden sollten. Folglich würden Sie hier eine proportionale oder antiproportionale geschichtete Stichprobe ziehen (siehe Häder und Häder 2019 für Stichprobenziehung in der quantitativen Sozialforschung; und Akremi 2019 für den Bereich der qualitativen Sozialforschung). Überlegen Sie sich gut, welche (potenziellen) Unterschiede zwischen den Gruppen vorliegen könnten, die Ihnen eine systematische und differenzierte Analyse Ihres Phänomens ermöglichen. Sie könnten auch so vorgehen, dass Sie erst einmal einige wenige Personen für Interviews auswählen, dann die Interviews auswerten und auf Basis dieser Auswertungen weiter Interviewpartner auswählen, die einen möglichst großen Kontrast zu Ersteren bieten. Bei diesem Verfahren handelt es sich um das Theoretical Sampling, das in der Grounded Theory entwickelt wurde (Glaser und Strauss 2008). Behalten Sie aber auch den Zeitaufwand für die Erhebung der Daten im Auge! Nicht alle Personen werden Ihnen sofort und bereitwillig zustimmen, interviewt zu werden. Manchmal werden Sie auch weite Reisen auf sich nehmen müssen, oder erst das Vertrauen der Interviewpartner*innen gewinnen müssen, indem Sie sich für eine (mehr oder minder) lange Zeit in deren Umfeld aufhalten. Manchmal kann dies Monate oder Jahre in Anspruch nehmen, wie an ethnographischen Feldstudien festgestellt werden kann. Somit sollten Sie sich auch hier fragen, ob das zusätzliche Interview, das Sie führen könnten, den Aufwand wert ist.

Diese Gedanken sollen Ihnen verdeutlichen, dass nicht bloß die Datenmenge und das Sampling einer Stichprobe mit einer spezifischen Größe, sondern auch der Auswahl- und Erhebungsprozess sehr viele Entscheidungen abverlangen wird, die nicht leichtfertig getroffen werden sollten. Sie sollten auch nicht vergessen, dass Zeit und Ressourcen investiert werden müssen, um diese Daten zu erheben! Auch hier gilt, dass kein direkter Zusammenhang zwischen der Stichprobengröße, der Datenmenge und der Analysetiefe vorherrschen muss, sondern dass der Zeitaufwand im Zweifel sehr viel größer ist, als im Vorfeld gedacht.

Erkenntnisinteresse der Forschung und Datenumfang stellen in Abbildung 1.1 die beiden Regler dar, die Ihnen eine erste Orientierung geben können, wie der Zusammenhang zwischen Tiefe und Breite sowie Datenmenge und Datenauswahl sind. Der Sampling-Prozess selbst und die Mühen, die damit verbunden sind, variieren hingegen zu stark, um in der Grafik abgebildet zu werden.

Abbildung 1.1 Heuristik zur Auswahl der geeigneten inhaltsanalytischen Auswertungstechnik



	Kapitel
Induktiv-qualitative Inhaltsanalyse	4
Deduktiv-qualitative Inhaltsanalyse	5
Induktiv-quantitative Inhaltsanalyse (teilautomatisiert)	6
Deduktiv-quantitative Inhaltsanalyse (teilautomatisiert)	7
Induktiv-quantitative Inhaltsanalyse (vollautomatisiert)	8
• Korrespondenzanalyse	9
• Sentiment-Analyse	10
• Topic Modeling	11

Steht der Regler oben wie bei vertiefter Analyse und geringem Datenumfang, dann ist eine induktiv-qualitative Inhaltsanalyse geeignet. Das heißt, Sie haben entweder eine geringe Datenmenge oder haben aus einer großen Datenmenge zu einem spezifischen Thema eine Datenauswahl getroffen. Die induktiv-qualitative Inhaltsanalyse steht in Bezug auf die Tiefe der Analyse ganz oben, da es sich um ein interpretatives Verfahren handelt, das für wenig Textmenge viel Zeit benötigt.

Etwas weniger Zeit für die Interpretation wird für die deduktiv-qualitative Inhaltsanalyse benötigt, da das Kategoriensystem aus der Theorie und/oder empirischen Studien aufgebaut wird und nicht wie bei der induktiv-qualitativen Inhaltsanalyse aus den Daten heraus entwickelt werden muss. Da Sie weniger Zeit für die Interpretation benötigen, ist die Breite der Analyse, also der Datenumfang höher.

Wie Sie der Abbildung 1.1 entnehmen können, stehen die beiden qualitativen Verfahren in Bezug auf Tiefe der Analyse und geringen Datenumfang ganz oben. Die weiteren Verfahren, die ab Kapitel 6 folgen, sind quantitative Verfahren, die einen immer größeren Datenumfang bearbeiten helfen, da sie mit teil- und vollautomatisierten Verfahren arbeiten.

Die induktiv-quantitative Inhaltsanalyse in Kapitel 6, für die die Programme AntConc und MAXQDA genutzt werden, ermöglicht trotz induktivem Vorgehen einen Einbezug eines größeren Datenumfangs, da anhand von Schlagworten das

Datenmaterial erschlossen wird und darauf aufbauend Erkenntnisse, Deutungen, Schlüsse und Interpretationen gezogen werden.

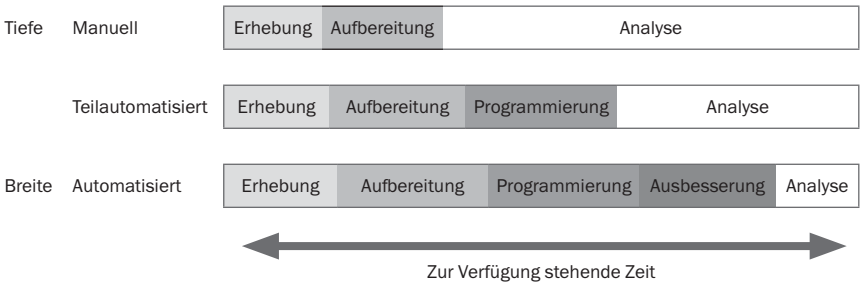
Das Bibliometric Literatur Review als deduktiv-quantitative Inhaltsanalyse ermöglicht die Auswertung eines relativ breiten Datenumfangs, der „mittel" vertieft ausgewertet wird. Durch die deduktive Auswahl des Datenmaterials werden anhand von Suchkriterien Metadaten von Publikationen automatisiert ausgewertet. Diese Ergebnisse müssen dann vertieft interpretiert werden.

Die Korrespondenzanalyse (Kapitel 9) geht noch einen Schritt weiter und entnimmt Worte aus den Texten, um auf Basis deren gemeinsamen Auftretens in allen analysierten Texten einen Themenraum zu erstellen. Dieser Themenraum ist durch Gegensätzlichkeiten gekennzeichnet (z. B. Studium versus Freizeit) und bietet Ihnen die Möglichkeit, sowohl Texte als auch Merkmale von Interviewpartner*innen zusätzlich in diesen Themenraum zu projizieren. Dies bietet Hinweise darauf, welche Interviews Sie tiefergehend analysieren sollten. Die Auswertungstiefe jedes einzelnen Textes ist als eher gering einzustufen.

Noch geringer ist die Analysetiefe einzelner Texte bei der Sentiment-Analyse (Kapitel 10) und Topic Modeling-Verfahren (Kapitel 11). Hier treten die Texte fast gänzlich in den Hintergrund, während der Fokus auf Maßzahlen oder auf die Visualisierung komplexer Themenzusammenhänge und Zeitverläufe gelegt wird. Maßzahlen können zum Beispiel die durchschnittliche (positive oder negative) Stimmung oder die Themenzusammensetzung innerhalb der Texte oder des Textkorpus sein. Texte werden dabei eher zur Illustration herangezogen oder um Fehler und Anomalien in der Zuordnung der Sentiment-Werte oder Themen zu Texten zu entdecken.

Das bedeutet nicht zwangsläufig, dass Sie weniger Aufwand mit automatisierten Verfahren der Textanalyse haben werden als bei der qualitativen Analyse von Texten. Das Zeitbudget, das Sie zur Verfügung haben, muss anders genutzt werden. Sie mögen vielleicht kaum Zeit damit verbringen, bei einem Topic Model in einzelne Texte zu lesen. Womöglich werden Sie viele der Texte (vor allem, wenn es sich um hunderttausende Texte handelt) niemals in Gänze oder auch nur auszugsweise lesen können. Viel mehr Zeit sollten Sie aber für die Datenaufbereitung, Datenbereinigung und Programmierung einkalkulieren. Vor allem sollten Sie sich die Zeit nehmen, Fehler in den Daten zu finden, die Ihnen ansonsten verzerrte Ergebnisse liefern, die entweder nicht zu deuten sind oder schlimmstenfalls ein falsches Bild von Ihrem Phänomen vermitteln. Sie sollten viel Zeit einplanen, um den Arbeitsprozess (den sogenannten Workflow) zu planen und zu dokumentieren, da Sie ansonsten ungleich viel mehr Zeit damit verbringen werden, Fehler zu beheben. Dahingegen verlangt die Erhebung von Interviewdaten und deren Transkription zwar auch viel Zeit, nimmt aber vor dem Hintergrund der Zeit, die für die eigentliche Analyse verwendet wird, einen geringeren Anteil Ihres Zeitbudgets in Anspruch. Um Ihnen diesen Zusammenhang zu verdeutlichen, haben wir eine zweite Heuristik erstellt (Ab-

Abbildung 1.2 Heuristik zum Arbeitsaufwand inhaltsanalytischer Auswertungstechniken



bildung 1.2), welche die Verwendung der Zeit innerhalb Ihres Forschungsprozesses darstellt.

Nachdem Sie nun einen Überblick haben, was Sie in diesem Buch erwartet, möchten wir Ihnen noch einmal empfehlen, Kapitel 2 und Kapitel 3 zu Beginn zu lesen und sich dann mit den von uns gegebenen Entscheidungshilfen durch das Buch zu arbeiten. Wir wünschen Ihnen dabei viel Erfolg und methodischen Spaß.

1.3 Copylefts und Copyrights von Software

AntConc, MAXQDA, Python, R und RStudio sowie VosViewer sind urheberrechtlich geschützte Software. Die Copylefts (z. B. von Freeware (z. B. AntConc und VosViewer) und von Freinutzungslizenzen (z. B. GNU 1-3 und Apache 2-0)) und/oder Copyrights (z. B. MAXQDA, MIT-Lizenz und BSD-2-Clause) der genannten Software ermöglichen die kostenfreie Verwendung für bestimmte wissenschaftliche Zwecke. Die in diesem Kapitel/Buch verwendete Auswertungssoftware, gegebenenfalls inklusive Programmiercode-Beispielen, erfolgt ausschließlich zum wissenschaftlichen Zweck der Lehre. Die Autor*innen und Beltz Juventa ermöglichen damit, dass jede*r den zu Beispielzwecken präsentierten Programmiercode verwenden und modifizieren darf, solange Verwendung und Modifikationen nicht kommerziellen und anderweitig ausschließlichen Zwecken dienen. Wie bei allen urheberrechtlich geschützten Inhalten, ist, analog zum wissenschaftlich redlichen Arbeiten, der Verweis (z. B. durch Zitation) auf den Ursprung von Code entweder physisch (z. B. gedruckte Publikation) oder digital (z. B. auf Austauschplattformen wie GitHub) kenntlich zu machen.

2. Inhaltsanalyse von Kommunikation

In diesem Kapitel wird definiert, was wir unter Inhaltsanalyse verstehen, was Inhaltsanalyse mit Kommunikation zu tun hat und welche Varianten der Inhaltsanalyse in diesem Buch behandelt werden. Zudem geben wir Ihnen einen kurzen Überblick über Varianten der qualitativen Inhaltsanalyse, die unserer Auffassung nach in induktive und deduktive qualitative Inhaltsanalyse eingeteilt werden kann, sowie einen Überblick über die quantitative Inhaltsanalyse. Sie lernen zudem zwei Regeln kennen, die Anschlussregel und die Ausschlussregel qualitativer und quantitativer Sozialforschung, die Ihnen dabei helfen, Ihre Forschung einzuordnen. Am Ende des Kapitels geben wir Ihnen noch einen Überblick über Gütekriterien in Bezug auf die Inhaltsanalyse.

2.1 Zusammenhang von Kommunikation und Inhaltsanalyse

Die Methoden und Techniken der sozialwissenschaftlichen Inhaltsanalyse sind unmittelbar mit der Frage nach dem Sinn von Kommunikation verknüpft. „Wie geht es dir?" – diese Kommunikationseröffnung haben Sie sicherlich schon unendlich häufig gehört. Wenn Sie sich solch eine Kommunikationseröffnung vor Augen führen, dann kann der Satz „Wie geht es dir?" sehr unterschiedliche Bedeutungen haben. Dahinter kann ein echtes Interesse an dem Wohlbefinden der gefragten Person stecken. Es kann aber auch nur eine Floskel sein, weil ein betretenes Schweigen aufgetreten ist, das überbrückt werden soll. Aufgrund der Art und Weise wie die Frage gestellt wurde, am Tonfall, an der Beziehung der Beteiligten, an der Körperhaltung und der Situation, in der die Frage gestellt wurde, kann nun abgelesen werden, wie eine gesellschaftskonforme Antwort ausfallen sollte. Also entweder mit einem „Mir geht es gut, danke", wenn es nur eine Floskel war, oder mit einer ausführlicheren Beschreibung des tatsächlichen Befindens, wenn es sich nicht um eine Floskel handelte. Im Verlauf des Gesprächs tragen Informationen über etwaiges psychisches, emotionales, körperliches, finanzielles usw. Wohlergehen dazu bei, dass die*der Fragende das Gutgehen verstehen kann. Es müssen also Erklärungen geliefert werden, die das Verstehen ermöglichen. Gleichzeitig kommt es in einer Kommunikationssituation zu vielen Deutungen und Interpretationen, die automatisch stattfinden. Beispielsweise erfolgt auf die Antwort „Das Baby hat zum ersten Mal durchgeschlafen" auf die Frage „Wie geht es dir?" die Deutung, weil das Baby durchgeschlafen hat, haben auch die Eltern wieder durchgeschlafen. Dadurch sind die Eltern weniger übermüdet und es geht

ihnen gut. Deutungen sind dabei hochgradig von kulturellen Kontexten abhängig und nicht selten falsch, was allerdings erst bei Nachfragen ans Licht kommt.

In dem Beispiel wird deutlich, dass es sich um codierte Kommunikation handelt. Die Eltern werden etwas gefragt und reden nicht über sich selbst, sondern über das Kind. Codierte Kommunikation meint, dass allein über die Bedeutung der Worte der inhaltliche Zusammenhang nicht verstanden werden kann. Vielmehr muss die Kommunikation gedeutet und interpretiert, also kodiert, werden, um sie zu verstehen. Deshalb kann es bei Kommunikationseröffnungen auch zu Irritationen kommen, wenn eine falsche Kodierung stattfindet. Wenn beispielsweise Frage und Antwort nicht zusammenpassen, weil die Floskel „Wie geht es dir?" falsch eingeordnet wurde. Vielleicht haben Sie auch schon die Situation erlebt, dass beispielsweise beim Bäcker der Verkäufer fragte „Wie geht es Ihnen?" und die einkaufende 70-Jährige die aktuelle Krankheitsgeschichte erzählt, da hier die Floskel eben als ehrliche Frage gewertet wird; und der Bericht der Krankheitsgeschichte bei den übrigen Anwesenden dazu führt, nicht zu wissen, wo hinzuschauen ist, oder sie miteinander non-verbal, durch den Austausch von Blicken, das Unbehagen kommunizieren.

Anhand der Beispiele möchten wir deutlich machen, dass Kommunikation einfach funktionieren, aber auch irritierend sein kann. Um Kommunikation, egal ob verbal, non-verbal oder schriftlich, zu verstehen, müssen Grundbedingungen erfüllt sein. Erstens müssen zunächst die Sprache, Gesten oder Symbole (z. B. in Form von Schriftzeichen) verstanden werden. Zweitens braucht es die Kenntnis des Kontextes. In Kommunikation wird immer auch der kulturelle Kontext einer Gesellschaft vermittelt, der sehr unterschiedlich sein kann. Und drittens kommt es auf die Motivation der Kommunikation an. Um das Beispiel von oben aufzugreifen, muss bekannt sein, dass es sich bei „Wie geht es dir?" je nach Kontext und Motivation, in der die Frage gestellt wurde, um eine ehrliche Frage oder um eine Floskel handelt.

Was aber ist, wenn wir diese intimen, reich mit Kontextinformationen gesättigten Situationen verlassen und medial vermittelte Kommunikationsräume wie beispielsweise Social Media-Plattformen betreten? Zunächst einmal werden wir feststellen, dass Kommunikationen anders gerahmt und mit weniger Kontextinformationen aufgeladen werden. Um bei dem vorigen Beispiel zu bleiben: Wenn jemand „Wie geht es dir?" fragt, dann verfügt der*die Adressat*in in einer Online-Kommunikation über keine Informationen jenseits des kommunizierten Textes. Wenn nicht gerade eine (durchaus mit Verzögerungen oder Ausfällen behaftete) Videokommunikation stattfindet, dann fehlen Stimmfarbe, Tonlage, Gestik und Mimik des*der Sprechenden und damit wichtige Hinweise, wie die Kommunikation zu deuten ist.

Doch auch wenn die reine, digitale Textkommunikation mit weniger Kontextinformationen angereichert ist, so können wir dennoch davon ausgehen, dass der Kommunikation ein geteilter Sinn unterliegt und dass die an dieser Kommuni-

kation beteiligten Akteure darüber reflektieren können, in welchem Kontext sie kommunizieren. Geht man ferner davon aus, dass Sprache strukturiert ist und mit Sinn belegte Ausdrucksformen in weitere, aus dem Kommunikationsfluss rekonstruierbaren Regelmäßigkeiten folgen, dann lässt sich deren Sinngehalt zumindest annähernd aus großen Textmengen auslesen. An dieser Stelle kommen wir in den Bereich der automatisierten bzw. computergestützten, quantitativen Textanalyse (Bail 2014; Heiberger und Riebling 2016).

Mit der Hilfe solcher Verfahren sind beispielsweise die Identifikation von wiederkehrenden Themen sowie der in der digitalen Kommunikation transportierten Stimmungen in einem ersten Schritt möglich. Im zweiten Schritt müssen Themen und Stimmungslagen aber noch immer von den Forscher*innen gedeutet und qualitativ eingeordnet werden, ehe sie in einem dritten Schritt in Bezug zueinander gesetzt und in den größeren Zusammenhang eingeordnet werden können. Dieser letzte Schritt setzt aber voraus, dass Informationen über den Kommunikationsfluss vorliegen (z. B. wer spricht zu wem/über wen).[1]

Ähnlich wie beim genannten Beispiel mit der 70-jährigen Dame beim Bäcker kann es auch im Falle digitaler Kommunikation zu Irritationen kommen, die aus der fehlerhaften Einordnung der Kommunikation durch die beteiligten Akteure entsteht. So zeigen beispielsweise Diskussionen in Foren und Kommentarspalten von Online-Ausgaben von Zeitungen, dass ironische Aussagen oft als ernstgemeinte Aussagen gewertet werden und gegebenenfalls in persönlichen Anfeindungen resultieren. Um dies zu erkennen und interpretieren, sind die Urteile der Forscher*innen gefragt, da selbst neuere, computergestützte Analysewerkzeuge Ironie und Sarkasmus eher schlecht erfassen können (Ghanem et al. 2020; Thelwall et al. 2012).

Warum nun aber die Ausführungen über Kommunikation? Weil Kommunikation das ist, was bei qualitativer und quantitativer Inhaltsanalyse analysiert wird (Bauernschmidt 2020; Schreier et al. 2019; Stamann et al. 2016). Um Kommunikation mittels Inhaltsanalyse untersuchen zu können, braucht es allerdings meist einen Zwischenschritt: die Verschriftlichung. Außer natürlich es handelt sich um schriftliche Kommunikation wie beispielsweise Zeitungsartikel, wissenschaftliche Artikel oder offene Fragen einer Fragebogenuntersuchung. Wenn die Kommunikation nicht schriftlich vorliegt, müssen beispielsweise Interviews transkribiert, Bilder schriftlich beschrieben oder Beobachtungen verschriftlicht werden. Das heißt für Mischformen, wie beispielsweise Tweets, die aus Text und Bildern bestehen, dass die Bilder zunächst beschrieben, um dann zusammen mit dem Text analysiert zu werden. Das gilt im Übrigen auch für Emojis, die sich in

1 Eine in der Politikwissenschaft angewandte Technik, die diesen Schritt ermöglicht, ist die Diskurs-Netzwerkanalyse (Leifeld 2020). Mit deren Hilfe können beispielsweise Struktur und Dynamik politischer Debatten, Polarisierung innerhalb des politischen Diskurses oder Konsensformulierung untersucht werden.

digitaler Kommunikation oftmals finden, beispielsweise, wenn Chatprotokolle analysiert werden. Erst wenn die gesamte Kommunikation als Text vorliegt, kann eine Inhaltsanalyse durchgeführt werden.

2.1.1 Textbasierte Inhaltsanalyse von Kommunikation

Text ist zunächst ganz banal eine Abfolge von Buchstaben, die Sinn ergeben. Buchstaben als Symbole werden in einem Prozess der Kommunikation oder Verschriftlichung zu einem Bedeutungszusammenhang. Dieser Bedeutungszusammenhang wird verstanden, wenn folgende Grundbedingungen erfüllt sind.

1. *Kenntnis der Symbole:* Wenn ich die Schriftzeichen nicht verstehe, dann kann ich auch den Inhalt nicht verstehen. Das gilt nicht nur für Schriftzeichen, sondern beispielsweise auch für Emojis.
2. *Kenntnis des Kontextes:* Nur wenn ich den Kontext verstehe, weil ich die kulturelle Bedeutung verstehe, kann ich die Symbole deuten und die Kommunikation dahinter verstehen. Deshalb ist es beispielsweise bei Interviews so wichtig nachzufragen, was genau gemeint ist und nicht die eigene Deutung der gesprochenen Symbole in Form von Sprache als gegeben hinzunehmen. In Kommunikation wird das Wissen der eigenen Sozialisation in eine (oder mehrere) Gesellschaft(en) transportiert. Das gilt nicht nur für eher traditionelle Formen des Interviews und der Verschriftlichung des Gesagten, sondern auch für medial generierte Kommunikation wie beispielsweise Tweets, Kommentare in Foren oder Facebook oder Blogeinträge. Das heißt, wir können Texte oftmals nur verstehen, wenn wir sie einordnen können. Wir erkennen beispielsweise ein Kochrezept sofort an den enthaltenen Textsegmenten mit Gramm-Angaben oder Gedichte an der Reimstruktur (Schwarz-Friesel und Consten 2014). Hinzu kommt bei digitalen Formen der Kommunikation, dass wir die Entstehungsbedingungen meist nicht kennen und die Einordnung des Gesagten daher erschwert wird.
3. *Kenntnis der Motivation:* Texte sind meist keine neutralen Zeugnisse, die die Realität abbilden, sondern haben ein Ziel und einen Zweck (Schwarz-Friesel und Consten 2014, S. 8). „Wissenschaftlicher ausgedrückt handelt es sich hierbei um die persuasive [Überzeugungs-]Funktion von Sprache, Menschen zum Handeln zu bewegen, sie glücklich oder unglücklich zu machen, sie zu überzeugen oder zu überreden. Dieses persuasive Potenzial von Texten ergibt sich aus der Instrument- und Handlungsfunktion von Sprache, Bewusstseinsinhalte zu aktivieren oder zu verändern, Gefühle zu wecken oder zu intensivieren und Handlungsimpulse auszulösen“ (Schwarz-Friesel und Consten 2014, S. 9). Das gilt auch für die Analyse von Pressekonferenzen von Unternehmen, Politikern, oder Parteien, deren Inhalt nur dann tiefgehend zugäng-

lich wird, wenn zugleich die Motivation und der Kontext bekannt sind und daher eine Deutung erlauben.[2]

Nur wenn die drei Grundbedingungen bekannt sind, kann eine Inhaltsanalyse gelingen. Dabei gibt es Textsorten, die einfacher zu verstehen sind als andere (Kracauer 1952). Schwer sind beispielsweise oftmals moderne Gedichte, weshalb wir in unserer Schulzeit gelernt haben, Gedichtinterpretationen zu schreiben und dazu Begleithefte brauchten. Wohingegen, wenn man einmal gelernt hat zu backen, die Textform des Backrezeptes ein einfacher Text ist. Hier braucht es höchstens in alten Rezepten Interpretationen über Mengenabgaben (oder wissen Sie auf Anhieb, was eine Unze ist?) (Schwarz-Friesel und Consten 2014).

2.1.2 Analysedreischritt: Kontext-Verstehen, Inhalte-Verstehen und dem Publikum verständlich machen

Die textbasierte Inhaltsanalyse von Kommunikation kann dabei selbst als ein kontinuierlicher Kommunikationsvorgang verstanden werden. Wie im vorigen Abschnitt erklärt, ist das textuelle und symbolische Verständnis Voraussetzung für das Kontext-Verstehen des inhaltsanalytisch untersuchten sozialen Phänomens. Das Kontext-Verstehen durch die*den Forscher*in ermöglicht das Inhalte-Verstehen der empirischen Daten durch Erklären, Deuten, Interpretieren und Schließen. Erst wenn die Verstehens-Voraussetzungen eins und zwei durch die*den Forscher*in erfüllt sind, kann der dritte Kommunikationsschritt „Publikum-Verstehen" bzw. einem Publikum die empirische Analyse verständlich machen erfolgreich anschließen. Eine sozialwissenschaftliche Untersuchung kann grob in drei Analyseschritte unterteilt werden (Tabelle 2.1). Spezifiziert durch die Forschungsfrage gilt es, in Analyseschritt 1 das soziale Phänomen anhand der sozialen, objektbezogenen, räumlichen und zeitlichen Aspekte zu erfassen. Für die Inhaltsanalyse von Dokumenten, Datenarrangements usw. ermöglicht das Kontext-Verstehen die Einordnung von dem, was kommuniziert wird und danach, warum es kommuniziert wird. Im zweiten Analyseschritt zielt die Was-Frage der kommunizierten Inhalte auf die inhaltsanalytische Erfassung der offensichtlichen, d. h. manifesten Inhalte.

Im Vergleich zu den manifesten Inhalten sind die in der Kommunikation verborgenen Bedeutungen, d. h. die latenten Inhalte von Kommunikation, in

2 Ähnliches findet sich in der Geschichtswissenschaft, wenn zwischen Überrest und Tradition unterschieden wird (Lersch und Stöber 2008). Erstere sind unbeabsichtigt der Nachwelt erhaltene Schriftstücke (z. B. Notizen), letztere bewusst verfasste Statements, die einen bestimmten Eindruck vermitteln (Memoiren) oder bestimmte Handlungsabläufe festhalten sollen (z. B. geschäftliche Verträge, Aufzeichnungen bürokratischer Vorgänge).

Tabelle 2.1 Analyse als Dreischritt von Kontext-, Inhalte- und Publikum-Verstehen

Analyseschritt 1	Analyseschritt 2	Analyseschritt 3
Kontext-Verstehen soziales Phänomen	Inhalte-Verstehen durch Erklären, Deuten, Interpretieren und Schließen	Publikum-Verstehen
Subjekt(e) Objekt(e) Raum/Räume Zeit/Perioden	*Was wird kommuniziert?* Manifester Inhalt (1. Sinnebene) *Warum wird kommuniziert?* Latenter Inhalt (2. Sinnebene)	Wissenschaft Expert*innen Allgemeine Gesellschaft …

der Regel schwieriger zu erfassen. Daher zielt der Analyseschritt 2b auf die Erfassung der zweiten Sinnebene von Kommunikation. Für die Analyse der Warum-Frage (Tilly 1984) von Kommunikation werden in der Methodenliteratur unterschiedliche, jedoch im Aktivitätskern Inhaltsanalyse synonyme Begriffe verwendet. Beispielsweise betitelt Rainer Diaz-Bone (2019) seinen Text mit „Formen des Schließens und Erklärens" und Hubert Knoblauch und Kolleg*innen (2018) überschreiben die Einleitung ihres Methodenhandbuchs mit „Interpretativ Forschen". Als synonyme Begriffe zielt keine der Analyseaktivitäten „erklären", „deuten", „interpretieren" und „schließen" auf die Konstruktion von mehr oder weniger logischen Zusammenhängen. Sowohl Erklärungen, Deutungen, Interpretationen als auch Schlüsse müssen alle plausibel und für Dritte gut nachvollziehbar mit dem untersuchten sozialen Phänomen im spezifischen Kontext in Zusammenhang stehen.

Im Verstehen-Dreischritt ist der Analyseschritt 2b stark abhängig von der Art der Forschung. Beispielsweise werden bei einer Untersuchung im Stil der Grounded Theory die Ad-hoc-Hypothesen überprüft bzw. es werden neue Ad-hoc-Hypothesen gebildet. In deduktiven Untersuchungen werden latente Inhalte von Kommunikation unter Zuhilfenahme der untersuchungsleitenden Theorie erklärt, gedeutet, interpretiert und erschlossen. Die Theorie ermöglicht dabei eine bestimmte Sichtweise der Inhaltsanalyse, wobei die Weiterentwicklung von Theorie zwecks besserer Erfassung des untersuchten sozialen Phänomens anschließen kann. Hier wird auch gelegentlich vom Blick durch eine bestimmte Theorie-Brille gesprochen. Theorieentwicklung oder Theoretisierung der Empirie ist das explizite Ziel von induktiven Untersuchungen. Bei einer induktiven Untersuchung werden dabei die latenten Inhalte erfasst und abstrakter beschrieben, sodass nachfolgende Forschung zu ähnlichen sozialen Phänomenen von der Theorieentwicklung profitieren kann.

Das im vorigen Absatz dargestellte Vorgehen im Analyseschritt 2b knüpft

nahtlos an den Analyseschritt 3 für andere Wissenschaftler*innen als Adressat*innen bzw. Publikum der Analyse an. Zur Vereinfachung des Publikum-Verstehens für die Wissenschaft ist es hilfreich, die gängigen fachwissenschaftlichen Begriffe zu verwenden – allerdings ohne beispielsweise soziologischer klingen zu wollen als Soziologieprofessor*innen. Ein gewisses Maß an einschlägiger Sprache ist auch in Publikationen empfehlenswert, welche ein Fach- oder Expert*innenpublikum ansprechen sollen. Fremdwort- und Fachjargon-Abstinenz ist empfehlenswert, wenn das Publikum unspezifischer ist, beispielsweise Leser*innen einer Zeitung. Um den Zeitungsleser*innen das soziale Phänomen verständlich zu machen, d. h. ihr Interesse zu wecken, müssen jedoch sämtliche Aspekte der Analyseschritte 1 und 2 berücksichtigt werden.

2.2 Definition von qualitativer und quantitativer Inhaltsanalyse

Wir haben nun geklärt, dass das Verstehen und Verständlich-Machen von Kommunikation in Textform der Analysegegenstand der sozialwissenschaftlichen Inhaltsanalyse ist. Aber was genau ist nun die Inhaltsanalyse? So einfach wie sie scheint, ist diese Frage allerdings nicht zu beantworten. Es gibt sehr viele unterschiedliche Definitionen dessen, was Inhaltsanalyse eigentlich ist (Prasad 2019). Kristallisiert man den Kern der Inhaltsanalyse heraus, dann geht es, sowohl bei quantitativer als auch bei qualitativer Inhaltsanalyse, um die systematische und regelgeleitete Erhebung und Analyse von Texten (Kommunikationsinhalt) durch die Interpretation von manifesten und latenten Bedeutungen über die Zerlegung der Texte in beispielsweise Kategorien, Themen, Topics oder Muster.

Was bedeutet diese Definition konkret? Um sie zu verstehen, zergliedern wir sie in ihre Einzelteile und betrachten jeden dieser Teile. Wie schon mehrfach angedeutet, gibt es die quantitative und die qualitative Inhaltsanalyse.[3] Die quantitative Inhaltsanalyse hat zum Ziel: „Die unüberschaubare soziale Wirklichkeit, die uns umgibt […], wird auf ihre zentralen Strukturen reduziert, um die Muster sichtbar zu machen, die ‚hinter den Dingen' stehen. Diesen Mustern wird eine

3 Wir geben im Folgenden nicht die gängige Unterscheidung zwischen quantitativer und qualitativer Inhaltsanalyse wieder, wie sie zum Beispiel von Rössler (2010, S. 19) geäußert wird: „Damit sollte auch der grundsätzliche Unterschied in der Vorgehensweise beider Zugänge deutlich geworden sein: Die standardisierte Medieninhaltsanalyse definiert vor der Untersuchung ihres Materials eine Reihe von bedeutsamen Kriterien, anhand derer sie ihr Material untersucht, während interpretative Verfahren ihre Aussage erst aus dem Material heraus entwickeln". Vielmehr gehen wir davon aus, dass die qualitative Inhaltsanalyse durchaus theorie- und hypothesengeleitet an ihr Material herangehen kann. Ebenso können quantitative Verfahren genutzt werden, um Muster zu erkennen, die dann erst gedeutet werden.

größere Bedeutung zugeschrieben als dem einzelnen Fall" (Rössler 2010, S. 19 f.). Erkennbar werden die Muster durch Messung und Quantifizierung (Früh 2011). Gemessen wird dabei beispielsweise die Häufigkeit von bestimmten Wörtern, Gesten in Videos oder dem Auftreten von Phrasen bzw. Adressierungen. Für die quantitative Inhaltsanalyse ist zentral, dass vor der Untersuchung festgelegt wird, was untersucht werden soll, also dass sowohl Analyseeinheiten als auch die Definitionen der Analyseeinheiten festgelegt sind. Ziel ist es, das Textmaterial in Variablen mit definierten Ausprägungen zu überführen, um quantitative Analysen wie beispielsweise Häufigkeitsberechnungen durchführen zu können. Dies können einerseits Codes[4] sein, die eine gewisse Abstraktion voraussetzen, aber auch Wörter, Wortstämme, oder Lemmata, die als Variablen erfasst bzw. innerhalb und zwischen Texten ausgezählt werden.

Die qualitative Inhaltsanalyse entstand in den 50er Jahren. Als Begründer der qualitativen Inhaltsanalyse gilt Kracauer (1952), der in seinem Artikel „The Challenge of Qualitative Content Analysis" aufzeigt, warum es eine qualitative Analyse von Inhalten braucht und worin die Unterschiede zwischen qualitativer und quantitativer Forschung liegen. Als Unterschiede benennt Kracauer, dass zum einen die Interpretation und Deutung von Kommunikation nicht objektiv ist, da sie von Subjekten mit eigenen Wertvorstellungen vorgenommen werden. Das bedeutet, dass sich qualitative Deutungen nicht in einfachen Skalen, also „gut, mittel, schlecht", ausdrücken können, da es sich dabei um subjektive Deutungen handeln würde. Bei Deutungen und Interpretationen von Kommunikation sollten deshalb keine Skalierungen unternommen werden (Kracauer 1952, S. 632). Als zweiten Unterschied benennt Kracauer den Umgang mit latenten, also nicht offensichtlichen, Inhalten.[5] Im Gegensatz zur quantitativen Inhaltsanalyse können latente Inhalte bei der qualitativen Inhaltsanalyse ermittelt werden, wodurch die Gefahr, die Daten zu vereinfachen, abgewendet wird (Kracauer 1952, S. 633).

Die Unterscheidung, die Kracauer hier in Bezug auf die Inhalte aufzeigt, spricht einen weiteren Aspekt der Definition an: die Frage nach manifesten und latenten Inhalten und damit nach dem Erkenntnisinteresse. Ganz einfach formuliert, gilt das Erkenntnisinteresse der Tiefe, die eine Untersuchung haben soll, also der manifesten oder latenten Sinnstruktur (Papilloud und Hinneburg 2018).

---

4 Der Begriff des „Codes" hat dabei mehrere Bedeutungen. Zum einen ist ein Code eine durch eine*n Forscher*in zugewiesene Bedeutung zu einer bestimmten Textstelle. Zum anderen bedeutet „Code" (bzw. Programm- oder Sourcecode) eine Abfolge von Befehlen in einer Programmiersprache, die vom Menschen les- und interpretierbar ist sowie zugleich von einem Computer eingelesen und vollständig ausgeführt werden kann.

5 „While it may be able to avoid obscure poems, it is much concerned with texts in which latent meanings not only pervade the manifest content, but also are intricately related to the objectives for which the analysis is under-taken. Such latent elements may strongly resist quantification, and occasionally the quantification is actually foregone" (Kracauer 1952, S. 634).

Greifen wir das Beispiel der Begrüßung wieder auf. Wenn untersucht werden soll, welche unterschiedlichen Begrüßungsformeln es gibt, dann bleibt die Untersuchung auf der ersten Sinnebene, auf der Ebene, die als Common-Sense oder manifeste Inhalte zu verstehen ist. Przyborski und Wohlrab-Sahr (2014, S. 20) definieren den Common-Sense als das, „was kompetente Gesellschaftsmitglieder unmittelbar erschließen könnten, wenn sie sich Zeit für eine systematische Rekonstruktion nehmen würden". Sollen also Begrüßungsformeln untersucht werden, dann kann z. B. eine teilnehmende Beobachtung am Tresen einer Bäckerei durchgeführt werden, bei der notiert wird, welche unterschiedlichen Begrüßungsformen über den Tag verteilt verwendet wurden. Es können aber auch Strichlisten geführt werden, die anschließend quantitativ ausgewertet werden. Möglich wäre auch, eine Analyse von Büchern, Zeitungsinterviews, Chatprotokollen oder transkribierten Videomitschnitten durchzuführen, bei der mit einem Algorithmus die Begrüßungsfloskeln extrahiert werden. Hier könnte dann z. B. analysiert werden, wie sich im Zeitverlauf die Begrüßungsformen verändern, ob Altersgruppen unterschiedlich angesprochen werden usw.

Wenn Untersuchungen auf der zweiten Sinnebene durchgeführt werden sollen, dann sind die latenten Sinnstrukturen von Interesse – also das implizite Wissen, das sich nicht sofort erschließt. Das Beispiel der Brottheke aufgreifend, würde hier interessieren, warum die Antwort der alten Dame auf die Begrüßungsformel „Wie geht es Ihnen?" zu Irritationen führte. Warum wissen wir eigentlich, dass es sich nicht um eine ehrliche Frage handelt? Oder warum hat sich die alte Dame nicht konform verhalten? Auch hier könnte wieder die teilnehmende Beobachtung oder Interviews mit den beteiligten Personen geführt werden, um an das „Warum hat die Situation zu Irritationen geführt?" zu kommen. Dabei wäre von besonderem Interesse, was die geteilten Annahmen der Anwesenden sind, das, was Mannheim (1980) als konjunktiven Erfahrungsraum beschreibt – also Verhaltensweisen oder Aussagen, die nicht erklärungsbedürftig sind, weil sie auf gemeinsamen Erfahrungs- und Wissensstrukturen beruhen. Es ist dies, was oben als Kontext und Motivation von Kommunikation beschrieben wurde. Um an die manifesten und latenten Sinnstrukturen der Inhalte zu kommen, wurden in der quantitativen und qualitativen Inhaltsanalyse unterschiedliche Techniken entwickelt bzw. wird auf unterschiedliche Methoden zurückgegriffen. Ein weiterer Teil der Definition ist hier angesprochen: die systematische und regelgeleitete Erhebung und Analyse von Texten.

Für die Erhebung von Daten, die mittels inhaltsanalytischer Verfahren ausgewertet werden, kann auf eine Reihe an Methoden der empirischen Sozialforschung zurückgegriffen werden. Wie oben angeführt, können das beispielsweise die teilnehmende Beobachtung, Interviews in allen möglichen Variationen,[6] die

6 Wie zum Beispiel Expert*inneninterview, problemzentriertes Interview, narratives Interview.

Erhebung offener Fragen in Surveys oder die Stichprobenziehung von Tweets sein. Unterschieden werden kann hier zwischen forschungsproduzierten Daten, also solchen, die ich extra für meine Untersuchung erhebe (z. B. Interviews), und prozessproduzierten Daten, die bereits existieren und für meine Untersuchung genutzt werden können (z. B. Tweets).

Techniken der Inhaltsanalyse sind die unterschiedlichen Wege, wie Texte zerlegt werden, und damit Inhalte und Bedeutungen zum Vorschein kommen. Je nach Technik der Inhaltsanalyse werden dabei Kategorien gebildet, Themen extrahiert, lassen sich *topics* ausgeben oder Muster erkennen. Genau diese Techniken wollen wir in Bezug auf digitale und automatisierte Möglichkeiten in diesem Buch näher beschreiben und Anwendungsanleitungen geben, weshalb wir sie hier nicht weiter ausführen.

Dabei gibt es nicht die *eine* oder sogar die *beste* methodische Herangehensweise an einen Untersuchungsgegenstand, auch nicht für die qualitative und quantitative Inhaltsanalyse. Vielmehr kommt es darauf an, was, wie oben aufgezeigt, das Erkenntnisinteresse ist. Deshalb geben wir in diesem Methodenbuch sowohl Entscheidungshilfen als auch Anleitungen, welche sozialwissenschaftliche Methode der Datenerhebung und dazugehöriger Auswertungstechniken für Ihre Untersuchung geeignet sein können und welche digitalen und automatisierten Möglichkeiten es gibt.

Grundlegend stellen sich bei der Auswahl der Erhebungsmethode samt dazugehörigen Auswertungstechniken für empirische Sozialforschung zwei miteinander zusammenhängende Fragen:[7] Erstens, welches Erkenntnisinteresse habe ich? Ist das Erkenntnisinteresse eher eine vertiefte Analyse des „Warums", um an die latenten Sinnstrukturen zu gelangen, oder soll ein Überblick des „Wie" gegeben und damit die manifesten Inhalte analysiert werden? Auch die Kombination von beiden Vorgehensweisen ist möglich, allerdings unter strikter Trennung der qualifizierenden und quantifizierenden *Logiken* des Vorgehens (siehe Ausschlussregel, die wir gleich erläutern werden).

Die zweite zentrale Frage umfasst den Gegenstand der Untersuchung. Das heißt, welche Subjekte und/oder Objekte eignen sich zur empirischen Untersuchung des sozialen Phänomens, das ich untersuchen möchte? Unter Untersuchungssubjekten sind Personen zu verstehen, mit denen beispielsweise ein Expert*inneninterview durchgeführt wird (Meuser und Nagel 2002). Untersuchungsobjekte für eine qualitative und quantitative Inhaltsanalyse reichen dabei von Dokumenten mit Text- und anderen audiovisuellen Inhalten bis zu

7 Für empiriebasierte Prüfungsleistungen im Studium – auch als Studien- und Hausarbeiten bezeichnet –, Bachelor- und Masterarbeiten ist es grundsätzlich empfehlenswert, dass Sie sich für eine Methode und die dazugehörige Auswertungstechnik entscheiden. Die Erfahrung zeigt, dass mehr Methoden – in der Regel – nicht bessere Ergebnisse empirischer Sozialforschung schaffen.

materiellen Einheiten (z. B. ein gedrucktes Buch) und deren sozio-technischen Einflüsse auf individuelles und soziales Handeln. Objekte können dabei separat oder in Kombination Gegenstand der Untersuchung eines sozialen Phänomens werden. Die Kombination von Objekten wird als Datenarrangement bezeichnet, welches beispielsweise bei einer Webseite aus Text, Bild und anderen Symbolen (z. B. Daumen hoch und Smiley) besteht.

Nachdem wir Ihnen nun eine grundlegende Definition der Inhaltsanalyse gegeben haben, möchten wir die Formen der Inhaltsanalyse, die wir in diesem Buch behandeln werden, in den methodischen Kanon einordnen. Das erscheint uns wichtig, da Sie im Laufe Ihres Studiums und in eigenen Recherchen mit den unterschiedlichen Formen in Berührung gekommen sind. Beginnen werden wir mit den Formen der qualitativen Inhaltsanalyse und dann werden wir auf neue Formen der quantitativen Inhaltsanalyse eingehen.

2.2.1 Formen qualitativer Inhaltsanalyse

Wenn Sie auf dieses Buch gestoßen sind, dann haben Sie wahrscheinlich bereits nach dem Begriff der qualitativen Inhaltsanalyse gesucht und sind dabei auf unterschiedliche Bezeichnungen der qualitativen Inhaltsanalyse gestoßen. Wie Schreier (2012) ausführt, sind die Unterschiede zwischen den Formen nicht so groß, wie es zunächst den Anschein hat. Unserer Auffassung nach gibt es einen zentralen Unterschied, der für die qualitative Inhaltsanalyse wichtig ist: induktiv versus deduktiv. Auf diese Unterscheidung lassen sich unseres Erachtens alle Formen der qualitativen Inhaltsanalyse, wie sie derzeit in der Literatur vorhanden sind, reduzieren (siehe dazu auch das Kapitel 7, in dem wir eine systematische Analyse der Literatur zur qualitativen Inhaltsanalyse durchgeführt haben). Bevor wir die einzelnen Varianten der qualitativen Inhaltsanalyse kurz erläutern und induktiv und deduktiv zuordnen, definieren wir, was unter induktiv und deduktiv zu verstehen ist.

Bei der qualitativen Inhaltsanalyse handelt es sich um die systematische und regelgeleitete Erhebung und Analyse von Texten (Kommunikationsinhalt) durch die Interpretation von manifesten und latenten Bedeutungen über die Zerlegung der Texte in Kategorien. Die Kategorienentwicklung ist nun das Unterscheidungsmerkmal zwischen induktiver und deduktiver qualitativer Inhaltsanalyse.

Bei der induktiven qualitativen Inhaltsanalyse werden die Kategorien aus dem Material heraus entwickelt (was auch als Emergieren bezeichnet wird). Die induktiv-qualitative Inhaltsanalyse kommt deshalb vor allem dann zum Einsatz, wenn zu einem Phänomen noch wenig bekannt ist und nicht auf empirische Studien oder auf Theorien zurückgegriffen werden kann. Sie können sich das Verfahren ganz kurz zusammengefasst so vorstellen, dass Sie Daten erheben (z. B. offene Interviews) oder Sekundärdaten verwenden, an die Sie mit einem offenen

Blick ohne Hypothesen oder Vorannahmen herangehen und die Kategorien aufgrund des Materials entwickeln (eine genaue Anleitung finden Sie in Kapitel 4). Da das Kategoriensystem aus dem Material entsteht, kommen beim Kodierprozess oftmals neue Kategorien hinzu. Das heißt, das Kodieren, also das Zuordnen von Textstellen zu Kategorien, ist hier eher als zyklischer Prozess zu verstehen.

Bei der deduktiven qualitativen Inhaltsanalyse wiederum werden die Kategorien aus einer oder mehreren Theorien sowie empirischen Studien abgeleitet. Das heißt, die Kategorien stehen bereits fest, bevor Sie an das Material gehen. Meist haben Sie das Kategoriensystem bei der deduktiven qualitativen Inhaltsanalyse auch bereits im Kopf, wenn Sie beispielsweise Interviews erheben, denn die Kategorien spiegeln sich in Ihren Leitfragen der Interviews wider. Beim Kodierprozess durchsuchen sie dann, sehr kurz gefasst, Ihr empirisches Material in Bezug auf die Kategorien und ordnen die Textstelle den Kategorien zu, was als Kodierung bezeichnet wird.

Im „Kodieralltag" ist diese scheinbar klare Unterscheidung aber oftmals weniger klar. Denn zum einen geht die qualitative Forschung davon aus, dass die Annahmen und Vorkenntnisse der Forschenden automatisch in die Interpretation von Datenmaterial einfließen (Armat et al. 2018). Niemand kann sich davon freimachen, wie er*sie sozialisiert wurde oder was er*sie bereits im Leben gelesen, gesehen, gehört hat. Deshalb ist es gerade bei qualitativer Forschung wichtig, sich seiner Vorannahmen bewusst zu werden (Strauss 2007) und für andere transparent zu machen, wie die Interpretation bzw. Kategorisierung erfolgt ist. Das nennt man intersubjektive Nachvollziehbarkeit. Sie ist das zentrale Gütekriterium qualitativer Forschung. Zudem passiert es beim deduktiven Forschungsprozess sehr häufig, dass Material beim Kodieren „übrig" bleibt. Das heißt, Material lässt sich nicht in die aus der Theorie abgeleiteten Kategorien kodieren (Graneheim et al. 2017), da es entweder für die Forschungsfrage nicht relevant ist oder, weil es Aspekte umfasst, die bisher in der verwendeten Theorie noch nicht beleuchtet wurden. Das bedeutet, dass das Kategoriensystem erweitert werden muss, also Kategorien umfasst, die aus dem Material selbst entwickelt werden (induktiv).

Zusammenfassend kann hier also festgehalten werden, dass beide Verfahren, also die induktive und die deduktive qualitative Inhaltsanalyse, immer auch Merkmale des jeweils anderen Verfahrens haben. Der zentrale Unterschied liegt aber im Beginn des Verfahrens. Bei der induktiven qualitativen Inhaltsanalyse werden die Kategorien aus dem Material heraus abgeleitet, wohingegen bei der deduktiven qualitativen Inhaltanalyse die Kategorien vor der Analyse des Materials aus Theorien oder empirischen Studien abgeleitet wurden und dann an das Material herangetragen werden. In dieser Unterscheidung lassen sich dann auch die unterschiedlichen Varianten im deutschen Sprachraum der qualitativen Inhaltsanalyse einordnen.

Induktiv-qualitative Inhaltsanalyse

- *Zusammenfassende Inhaltsanalyse* (Mayring 2010): Hier werden durch Paraphrasierung und Reduktion Kategorien aus dem Material heraus entwickelt. Bei Mayring besteht entsprechend die Unterscheidung zwischen induktiver und deduktiver Herangehensweise in dem regelgeleiteten Vorgehen. Wenn das Verfahren der zusammenfassenden Inhaltsanalyse durchgeführt wird, handelt es sich um ein induktives Vorgehen, auch wenn Kategorien, die aus dem Material entstehen, im Verlauf der Analyse einer Theorie zugeordnet werden können bzw. eine theoretische Rahmung entsteht.
- *Qualitative Inhaltsanalyse (data-driven)* (Schreier 2012): Für Schreier ist die qualitative Inhaltsanalyse klar in der qualitativen Forschung verortet, weshalb sie die Abgrenzung zu hermeneutischen Verfahren (sie spricht von Code Analysis) darin sieht, dass die qualitative Inhaltsanalyse vor allem für deskriptive Auswertungen geeignet ist, nicht für analytisch theoriebildende Verfahren wie die Grounded Theory (Glaser und Strauss 2008; Strauss 2007). Wir ordnen Schreier eher der induktiven Vorgehensweise zu, da sie als ersten Schritt das Aufbrechen des Materials ansieht.
- *Conventional Content Analysis* (Hsieh und Shannon 2005): Die vorgeschlagene Variante orientiert sich stark an dem Vorgehen der Grounded Theory, sieht den Unterschied allerdings darin, dass durch die qualitative Inhaltsanalyse eine Beschreibung des Phänomens vorgenommen und nicht eine (gegenstandsbezogene) Theorie entwickelt werden soll.
- *Inductive Approach* (Graneheim et al. 2017): Das Vorgehen ist durch eine Suche nach Mustern gekennzeichnet. Bei der Analyse suchen Forschende nach Gemeinsamkeiten und Unterschieden in den Daten, die in Kategorien überführt werden. Dabei wird von den Daten zu einem theoretischen Verständnis übergegangen, also vom Konkreten und Spezifischen zum Abstrakten und Allgemeinen.

Deduktiv-qualitative Inhaltsanalyse

- *Strukturierende Inhaltsanalyse* (Mayring 2010) mit den Untervarianten „formale Strukturierung", „inhaltliche Strukturierung", und „typisierende Strukturierung": Das Kategoriensystem ist bereits vor dem Kodieren aus einer oder mehreren Theorien bzw. bereits durchgeführten Studien oder Literatur abgeleitet und die Zuordnung der Textstellen zu den Kategorien erfolgt anhand festgelegter Kodierregeln. Wenn das Verfahren der strukturierten Inhaltsanalyse durchgeführt wird, handelt es sich um ein deduktives Vorgehen, auch dann, wenn z. B. Fragen des Interviewleitfadens aus Alltagsbeobachtungen generiert wurden. Wenn diese dann als Kategorien durch Kodierregeln in die Auswertung aufgenommen werden, handelt es sich dabei um deduktives Kodieren, auch wenn die Kategorien hier nicht theoretisch abgeleitet wurden. Die Aufnahme weiterer Kategorien in die strukturierende Inhaltsana-

lyse bleibt bei Mayring weitgehend unklar, weshalb Steigleder (2008) in ihrem Buch dazu eine Anleitung vorschlägt.

- *Inhaltlich strukturierende qualitative Inhaltsanalyse* (Kuckartz 2018): Im Methodenbuch von Kuckartz wird die qualitative Inhaltsanalyse exemplarisch an Interviewmaterial veranschaulicht, weshalb die Hauptkategorien anhand des Interviewleitfadens entwickelt werden. Dieser wurde durch intensive Literaturstudien erstellt, was unserer Auffassung nach Theoriearbeit und die Arbeit mit empirischen Studien umfasst. Deshalb stufen wir die inhaltlich strukturierende qualitative Inhaltsanalyse nach Kuckartz als deduktive Variante ein, wobei zu den Hauptkategorien Subkategorien hinzukommen können, die induktiv gebildet werden. Die induktive Kategorienbildung ist bei Kuckartz an das offene Kodieren der Grounded Theory angelehnt (Glaser und Strauss 2008; Strauss 2007).
- *Extrahierende Inhaltsanalyse* (Gläser und Laudel 2010): Bei der Extraktion werden zunächst deduktive Kategorien aus der Theorie abgeleitet, indem eine theoretische Analyse des Problems/Gegenstands stattfindet und daraus die Kategorien abgeleitet werden (Gläser und Laudel 2010). Diese Kategorien erhalten Indikatoren, also Regeln, die eine klare Zuordnung zu der Kategorie ermöglichen (was Mayring als Kodierregeln bezeichnet). Wird nun das Material kodiert, können die Kategorien angepasst und ebenso induktiv neue Kategorien generiert werden.
- *Directed Content Analysis* (Hsieh und Shannon 2005) oder *Deductive Approach* (Graneheim et al. 2017) oder *concept-driven* (Schreier 2012): Ausgehend von Theorien und vor allem Leerstellen in Theorien oder bisher nicht geprüften Theorien wird ein Kategoriensystem entwickelt, das an Textmaterial (meist Interviews) überprüft werden soll, mit dem Ziel einer empirischen oder theoretischen Weiterentwicklung. Da Theorien weiterentwickelt werden sollen, kommt Textstellen, die keiner deduktiven Kategorie zugeordnet werden können, eine besondere Bedeutung zu und sie werden in neu entwickelte Kategorien, die aus dem Material entstehen (induktiv), kodiert.

Mit dieser Auflistung wollten wir aufzeigen, dass sich die qualitative Inhaltsanalyse in die zwei Varianten der induktiven und deduktiven Inhaltsanalyse einteilen lässt. Die Unterschiede der Formen, wie sie in der Liste aufgeführt sind, bestehen in den Beschreibungen der einzelnen Auswertungsschritte und in der epistemologischen Verortung der Autor*innen – also welchen theoretischen und empirischen Grundannahmen sie sich verpflichtet fühlen. In unseren Ausführungen versuchen wir, ein forschungspragmatisches Vorgehen zu beschreiben, das es Ihnen ermöglicht, die Methode der qualitativen Inhaltsanalyse möglichst einfach, aber regelgeleitet und damit transparent und nachvollziehbar durchzuführen.

2.2.2 Quantitative Inhaltsanalyse: ein Überblick

Im Gegensatz zur vertiefenden Auswertung von Textdaten mit der qualitativen Inhaltsanalyse zielt die quantitative Inhaltsanalyse auf das Erkennen von inhaltlich manifesten und latenten Mustern in Kommunikation basierend auf Häufigkeiten und statistischen Korrelationen. Wie in Kapitel 1.2 dargestellt, unterscheiden sich die Datenmengen, welche bei der quantitativen Inhaltsanalyse mehrere hundert bis hunderttausend und mehr Seiten umfassen können. Anders als die qualitativen Formen der Inhaltsanalyse mit ihrem jeweiligen theoretischen Unterbau ist die quantitative Inhaltsanalyse empirie- und problemgetrieben, atheoretisch und nicht zuletzt vom technologischen Fortschritt abhängig. Teil- und vor allem vollautomatisierte computergestützte Verfahren der Textanalyse sind den Bereichen der Computerlinguistik (*natural language processing,* NLP) und der Künstlichen Intelligenz (*artificial intelligence,* AI) zuzuordnen. Die Hauptanwendung von vollautomatisierter Inhaltsanalyse zielt darauf, Bedeutungen und Muster innerhalb der menschlichen Sprache (= natürliche Sprachen wie Deutsch oder Englisch) zu erkennen (z. B. Korrespondenzmuster in Kapitel 9 und latente Themen in Kapitel 11), Gesprochenes zu transkribieren, automatische Übersetzungen anzubieten oder Stimmungen zu erkennen, die im Text transportiert werden (Sentiment-Analyse in Kapitel 10; Otter et al. 2020). Im Gegensatz dazu wurden teilautomatisierte Auswertungsverfahren für die computergestützte Korpuslinguistik inklusive Polito- und Soziolinguistik (z. B. Niehr 2014; Veith 2002) entwickelt, deren Ziel das Entdecken manifester Muster von Sprache ist (z. B. Inhaltshäufigkeiten in Kapitel 6 und Netzwerkmuster in Kapitel 7). Teil- und vollautomatisierter Inhaltsanalyse ist gemein, dass die Daten- und das Textformat einheitlich aufbereitet sein müssen, damit die Software die Texte erkennen und auswerten kann. Grundlegende Unterschiede zwischen den quantitativen Formen der Textanalyse betreffen:

1. Teilautomatisierte Inhaltsanalyse basieren zu etwa gleichen Teilen auf dem Ineinandergreifen von menschlichen und von maschinellen Auswertungsschritten sowie der Genese manifester Ergebnisse (Tabelle 2.1). Der Aspekt teilautomatisiert hebt weiter hervor, dass die Auswertung und Ergebnisgenese nicht in einem finalen Durchlauf (nach viel Training; siehe unten), sondern nach und nach, also sequenziell, erfolgt. Dabei liegt die Auswertung jeden Schrittes in Ihrer Hand – und wird Ihnen in der Regel nicht durch Algorithmen und statistische Maßzahlen abgenommen (z. B. Werte zur Bestimmung der Trennschärfe von Themen). Die verwendete Software wie beispielsweise AntConc verfügt meist über ein (rudimentäres) Anwendungsinterface und erfordert keine Programmierkenntnisse. Die Datenmengen sind in der Regel noch so überschaubar, dass sie mit sehr viel Zeit und Aufwand induktiv ausgewertet werden könn(t)en (z. B. durch Zählen von Worthäufigkeiten).

Induktiv bedeutet hier, dass von einzelnen ganz spezifischen Aussagen ausgehend immer allgemeinere, abstraktere Kategorien hergeleitet werden, die am Ende entweder theoretische Aussagen oder, durch In-Bezug-Setzen von Kategorien, die Detailbeschreibung von untersuchten Phänomenen ermöglichen.

2. Vollautomatisierte Inhaltsanalyse trennt die Auswertung der Daten durch die Software (z. B. Python und R) vom Erkennen latenter Ergebnisse durch die*den Sozialwissenschaftler*in. Für die Auswertung muss die Software trainiert werden (= *machine learning*), um nicht-manifeste Merkmale, d. h. Muster, Bedeutungen und Themen, in großen Textmengen erkennen zu können, welche wir mit den Kapazitäten des menschlichen Gehirns nicht in der Lage sind zu verarbeiten. Das bedeutet aber nicht, dass Sie einen Knopf drücken und Ihnen die Maschine ein fertig interpretiertes Modell liefert. Im Gegenteil: Sie müssen die statistischen Maßzahlen und zugeordneten Textpassagen, Wörter, Themen und Stimmungslagen in Bezug zueinander setzen und interpretieren. Sie werden dabei sehen, dass in Zahlen auch sehr viel (konstruierter) Sinngehalt stecken kann, der ergänzend zu Texten (qualitativ) interpretiert werden muss. Insofern unterscheidet sich die automatisierte Textanalyse dadurch, dass Sie sowohl Texte als auch die Zahlen interpretieren und aufeinander beziehen müssen. Ein weiterer Unterschied besteht in der Flexibilität der Datentypen, welche sich für die Analyse eignen – was auch Probleme mit sich bringen kann. In Gegensatz zu einem einheitlichen Textkorpus (z. B. sehr viele Zeitungsartikel zu einem bestimmten Thema oder ein themenbezogener Austausch auf Twitter), kann es bei einem sehr heterogenen Textkorpus (z. B. Filmskripte, in denen auch Wissenschaft thematisiert wird; Kapitel 11) notwendig sein, dass Schlüsselwörter von den Sozialforscher*innen vorgegeben werden müssen und Sie zudem die Texte und die dazugehörigen Informationen vereinheitlichen. Anhand der Schlüsselwörter kann die Software darauf trainiert werden, abzugleichen, welche Wörter dem Schlüsselwort ähnlich (z. B. Biologie, biologisch, bio usw.) und in nächster Umgebung zum Schlüsselwort platziert sind (z. B. Doktor und Labor, chemische Reaktion usw.).

Die Versuche, Texte teil- und vollautomatisiert zu analysieren, können bis in die 1950er Jahre zurückverfolgt werden (Deng und Liu 2018, S. 2–18). Die problemzentrierte, technikabhängige Herangehensweise verdeutlicht die folgende Aussage von Jones (1994, S. 3) sehr gut, die die Aussagen von Fred Thompson, einem Pionier im Bereich der Computerlinguistik, auf der Konferenz der Association of Computing Machinery (ACM) aus dem Jahr 1987 zusammengefasst hat.

> „Work in the field has concentrated first on one problem, then on another, sometimes because solving problem X depends on solving problem Y but sometimes just because

problem Y seems more tractable than problem X. We may indeed be seduced by the march of computing technology into thinking we have made intellectual advances in understanding how to do NLP, though better technology has also simply eliminated some difficulties we sweated over in earlier years. But more importantly, better technology means that when we return to long-standing problems they are not always so daunting as before".

In den 1950er Jahren bestand der Fokus der Computerlinguistik darin, Texte automatisch aus dem Russischen ins Englische zu übersetzen (Wissenschaft im Dienst des Kalten Krieges), was jedoch häufig aufgrund langer Berechnungszeiten (mehrere Minuten pro Satz) und fehlender systematischer Programmiersprachen (wie Python) technisch äußert herausfordernd war (Plath 1967). In dieser Zeit wurden lexikon- und regelbasierte Ansätze entwickelt, die sich in unserem Buch in der Sentiment-Analyse (Übersetzung von Worten und Semantik in positive oder negative Stimmungslagen; Kapitel 10) wiederfinden.

Konnte mit den limitierten Methoden der 1950er und 1960er Jahre jegliche Leistung des Computers durch geschulte Übersetzer*innen übertroffen werden, wurden in den folgenden gut 50 Jahren die Methoden ausgebaut, sodass wir heutzutage beispielsweise über eine automatisierte Sprachergänzung (z. B. wenn Sie Suchbegriffe in Google eingeben), Spam-Filter oder Topic Modeling-Ansätze verfügen. Für die Entwicklung kamen in den 1960er und 1970er Jahren zwei relevante Entwicklungen zusammen: Einerseits die Entwicklung von Künstlicher Intelligenz, die beispielsweise Sinngehalte durch sogenannte semantische Netzwerke (Findler 1979; Schank und Tesler 1969) in Relation zueinander brachte und Inferenzen (= statistisches Schließen) auf Sinngehalte ermöglichte; andererseits die von Chomsky (2009) entwickelte generative Grammatik mitsamt der Annahme, dass Schlüsselelemente der Sprache im menschlichen Gehirn repräsentiert sind und somit mit der angeborenen Fertigkeit des Menschen verknüpft sind, Sprache und Sinn zu verstehen. Letzteres erlaubte eine vertiefende Analyse von Sprache, die deren logischen, syntaktischen und semantischen Aufbau mit Wissensbeständen und Aussagen in Verbindung bringen konnte, die Glaubenssätze und Intentionen von Sprecher*innen aus den Texten in Verbindung bringen konnten.

Die soziolinguistischen bzw. auf den Textkorpus bezogenen Merkmale von Sinnverstehen in und Sinnausdruck durch Sprache bildet die Grundlage von teil- und vollautomatisierter Inhaltsanalyse. Jedoch mussten für die vollautomatisierte Inhaltsanalyse von den Forscher*innen noch Vor- und Zusatzarbeiten geleistet werden, welche noch heute in der teilautomatisierten Inhaltsanalyse präsent sind. Aufgrund noch immer geringer technischer Ressourcen (ein „Supercomputer" aus den 1970ern kann nicht einmal mit der Rechenkapazität Ihres Handys mithalten), wurden extensive Regelwerke erstellt, mit deren Hilfe Bedeutungen aus den Texten extrahiert wurden.

Die Phase der regelgeleiteten Erforschung von Sprache wurde in den 1980ern durch ein empiristisches Paradigma ersetzt (Murphy 2012). Mit diesem neuen Paradigma etablierte sich das *machine learning,* also maschinelles Lernen, als eigenes Forschungsfeld, das sich auf Mustererkennung *(pattern recognition)* und die Extraktion von Informationen aus maschinenlesbarem Text spezialisierte. Hierbei werden zwei Typen maschinellen Lernens voneinander unterschieden: das nicht-überwachte und das überwachte maschinelle Lernen. Nicht-überwachtes maschinelles Lernen sollte Strukturen innerhalb von Daten (nicht bloß Texten!) ohne die Zuarbeit von Forscher*innen identifizieren (Solan et al. 2005). Überwachtes maschinelles Lernen greift auf von menschlichen Expert*innen erstellte Regeln und Zuordnungen zurück (z. B. Kodes, die im Rahmen qualitativer Analysen vergeben wurden und diesen Kodes zugehörige Textstellen), um Muster zu erkennen. Doch auch hier waren anfangs die Möglichkeiten des *machine learning* sehr begrenzt. Das lag schlicht darin begründet, dass wenige maschinenlesbare Texte vorlagen (stellen Sie sich eine Welt ohne Internet vor, in der Text fast ausschließlich in gedruckter Form vorkommt) und die Rechnerkapazitäten von Computern bei weitem nicht ausreichen, um das Lernverhalten eines menschlichen Gehirns zu simulieren (Benvenuto und Piazza 1992).

Beim *machine learning* erlernt Software durch „Training" statistische Modelle anhand von Daten zu formulieren. Die Trainingsdaten bilden dann die Grundlage dafür, dass die Software ähnliche Muster in anderen Datenkorpora durch Abgleich (z. B. von Listen) erkennen kann. Seit den 1990er Jahren wird zunehmend auf sogenannte vielschichtige neuronale Netzwerke und Deep-Learning-Methoden zurückgegriffen, um Muster in großen Textdatenmengen zu erkennen und die Textinhalte zu gruppieren bzw. zu klassifizieren. Parallel hierzu verbreitete sich das Internet (und damit rasant maschinenlesbare Inhalte), die wiederum zu Verfeinerungen der verwendeten Algorithmen und Methoden zur Behebung von Klassifikations- und Zuordnungsproblemen in der Computerlinguistik führten.

2.2.3 Inhaltsanalytische Kombination von qualitativen und quantitativen Auswertungstechniken

Wie weiter oben bereits angedeutet, lassen sich für die qualitative und quantitative Textanalyse geeignete Daten auf unterschiedliche Weise erheben. Sie lassen sich aber ebenso miteinander kombinieren, um den untersuchten Gegenstandsbereich aus verschiedenen Perspektiven zu beleuchten und damit zu einem tiefgreifenden Verständnis des Phänomens zu gelangen. Zentral bei der Anwendung und Kombination von Erhebungsmethoden und den Techniken der Inhaltsanalyse sind zwei Regeln: Die Anschlussregel und die Ausschlussregel qualitativer und quantitativer Sozialforschung.

Die Anschlussregel betont die grundsätzliche Kombinierbarkeit von qualita-

tiven und quantitativen Methoden empirischer Sozialforschung (Burzan 2016; Creswell und Plano Clark 2011; Kelle 2019; Tashakkori und Teddlie 2003), also die Nutzung der qualitativen zusammen mit der quantitativen Inhaltsanalyse oder auch mit anderen Methoden.

Die Ausschlussregel empirischer Sozialforschung lautet:

a) qualitative Methoden und dazugehörige Auswertungstechniken ermöglichen keine quantifizierbaren Erkenntnisse bzw.
b) quantitative Methoden und dazugehörige Auswertungstechniken ermöglichen keine über das Quantifizierte hinausgehenden qualitativen Erkenntnisse.

Die Ausschlussregel empirischer Sozialforschung kann gut an zwei Beispielen der Inhaltsanalyse erläutert werden. Stellen Sie sich vor, Sie haben eine bestimmte Menge an Texten als Daten. Aufgrund Ihres Erkenntnisinteresses und theoretischer Vorannahmen haben Sie bestimmte Begriffe oder Schlagworte identifiziert. Nun gehen Sie durch die Texte und kodieren (= markieren) mithilfe einer Auswertungssoftware die Begriffe. Nach Abschluss des Kodiervorgangs haben Sie nun eine bestimmte Anzahl n der Begriffsnennungen. Dieses n der Begriffsnennungen zeigt Ihnen die Häufigkeit, jedoch nicht die Bedeutung der jeweiligen Begriffe im Kontext der Texte an. Folglich können Sie einen quantitativen Überblick des Kodes als Ergebnis präsentieren. Die Häufigkeit der Begriffe ermöglicht Ihnen jedoch keinen vertieften, qualitativen Einblick in die Inhalte der Texte. Selbst wenn Sie statt nur des Begriffs systematisch den Satz, in dem der Begriff vorkommt, und den vorherigen und nachfolgenden Satz codiert haben, so wäre eine qualitative Inhaltsanalyse dieser Ausschnitte ein weiterer bzw. neuer Auswertungsschritt, welcher jedoch in Abhängigkeit zur vorhergegangenen Quantifizierung steht und folglich keine qualitative Inhaltsanalyse der Texte ist.

Methodenplurale Forschung bzw. der auch in der deutschsprachigen Literatur häufig verwendete englischsprachige Begriff Mixed Methods-Forschung betont, dass für empirische Sozialforschung qualitative und quantitative Methoden sowohl

a) parallel als auch
b) sequentiell

genutzt werden können. Ein Beispiel für die parallele Verwendung ist die Nutzung der qualitativen Inhaltsanalyse von Texten, Videos und Symbolen auf einer Webseite, um ein tieferes Verständnis der Inhalte dieser Website zu bekommen. Das Verständnis würde dann mit einer parallel stattfindenden Fragebogenerhebung der Nutzer*innen der untersuchten Webseite verglichen. Sollen hingegen die Besonderheiten der internetbasierten Kommunikation auf einer Webseite

untersucht werden, so wäre ein sequenzielles Forschungsdesign zu wählen. Zuerst würden dann qualitativ die Inhalte der Webseite ausgewertet und als Ergebnisse festgehalten werden. In einem zweiten Schritt könnten die Erkenntnisse anschließend zur Formulierung von empirisch fundierten Annahmen in Form von Hypothesen für die Fragebogenerhebung genutzt werden. Existiert jedoch wenig Wissen über die Nutzer*innen der Webseite, beispielsweise, weil alle Nutzer*innen nur Pseudonyme verwenden oder die Neuheit einer Webseite bzw. deren Kommunikationsmöglichkeiten eine unbekannte Community an Nutzer*innen (z. B. Alter, Geschlecht und soziale Herkunft) anzieht, so könnten zuerst durch eine Fragebogenerhebung Informationen gesammelt werden, um anschließend die Inhalte der Webseite informiert mit einer qualitativen Inhaltanalyse auswerten zu können.

Im Bereich computergestützter, quantitativer Inhaltsanalysen beispielsweise ist eine parallele Erhebung von Themen und Stimmungslagen möglich, die in den Onlinekommunikationen ablaufen. In dem Fall werden Textinhalte von Tweets, Foreneinträgen etc. mit Haltungen zu Themen in Verbindung gebracht und im Anschluss auf wenige, zentrale Dimensionen heruntergebrochen. Eine Dimension ist dabei als eine latente Bedeutungseinheit, z. B. als Thema, zu interpretieren. Da in einem analysierten Gespräch bzw. einer Vielzahl von Kommunikationen auch viele Themen behandelt werden können und Menschen auf eine je eigene Art über diese Themen sprechen, ergeben sich dementsprechend viele, komplex zueinanderstehende Dimensionen. So können beispielsweise die Nutzer eines virtuellen Campus in einem angelegten Thread zunächst über die Studiensituation an der eigenen Universität, dann über den Übertritt zum Arbeitsmarkt, dann über die Finanzierung des eigenen Studiums und zuletzt über die Dozierenden sprechen. Je nachdem, wie „fein" das Programm eingestellt wurde, mit dem diese Gespräche und Kommunikationen analysiert werden, werden entsprechend wenige/viele Themen und Dimensionen gefunden. Diese Dimensionen tragen dann wiederum eigene Bedeutungen, die die Forscher*innen herausdestillieren müssen.

In den vergangenen drei Jahrzehnen sind im Bereich der Computerlinguistik, des maschinellen Lernens und der Informationsgewinnung *(information retrieval)* Methoden entstanden, die eine automatisierte Analyse von Texten ermöglichen. Diese in der Soziologie weitestgehend nicht wahrgenommenen Methoden nutzen statistische Verfahren, um Strukturen in Texten zu erkennen.[8] Die

8 Dabei gibt es einige Ausnahmen, in denen Methoden der automatisierten, quantitativen Textanalyse für kultursoziologische Fragestellungen (Breiger et al. 2018; Edelmann und Mohr 2018; Kozlowski et al. 2019; Mohr und Bogdanov 2013; Mohr et al. 2015), Fragestellungen der politischen Soziologie (Fuhse et al. 2020) oder Wissenschaftsforschung (Grothe-Hammer und Kohl 2020; Munoz-Najar Galvez et al. 2019; Schwemmer und Wieczorek 2020; Wieczorek et al. 2021) genutzt wurden. Darüber hinaus gibt es einige wenige anwendungsorientierte Reflexionen über die Verwendung von automatisierten Verfahren und

zugrundeliegende Annahme ist dabei, dass die statistisch entdeckten Strukturen auf Sinngehalte, angesprochene Themen oder Stimmungslagen hindeuten, die einer Vielzahl analysierter Texte zugrunde liegen (DiMaggio et al. 2013). Mehr noch: Vertreter*innen aus den Bereichen des *information retrieval,* maschinellen Lernens und der Computerlinguistik gehen davon aus, dass mithilfe der statistischen Modelle auch Voraussagen – sogar über psychische Eigenschaften der Schreibenden oder gesundheitliche Befunde! – oder Empfehlungen möglich sind (Nguyen und Shirai 2015; Obermeyer und Emanuel 2016; Resnik et al. 2013; Wang und Blei 2011). Dieser Aspekt der Voraussage – oder auch Prädiktion – unterscheidet automatisierte Methoden der Textanalyse von anderen Formen der Ihnen aus dem Studium bekannten quantitativen Methoden empirischer Sozialforschung wie zum Beispiel Regressionsanalysen.

Auch wenn hier der Duktus vorherrscht, dass mithilfe automatisierter Verfahren latente Sinnstrukturen in Texten objektiv erkannt, ohne menschliches Zutun extrahiert, interpretiert und sogar prognostiziert werden können, so kommen auch Verfahren der automatisierten quantitativen Textanalyse nicht ohne Interpretationsleistungen von Forscher*innen aus. Einige Forscher*innen greifen mittlerweile sogar bewusst auf Methoden der qualitativen Sozialforschung zurück, um die Grobkörnigkeit der Analyseergebnisse automatisierter quantitativer Textanalysen auszugleichen und reflektieren dabei, wie eine Integration automatisierter Methoden und qualitativer Methoden möglich ist (Andreotta et al. 2019; Dillon et al. 2020; Fawcett et al. 2019; Lauer et al. 2018; Schneiker et al. 2019). Das ist vor allem der Erkenntnis geschuldet, dass die Ergebnisse automatisierter Methoden häufig schlecht für den Menschen interpretierbar sind, Inhalte und Themen unplausibel sind, oder Muster offenbaren, die der Intuition der Forschenden widersprechen und damit auf Fehler in den Daten oder theoretischen Ansätzen hindeuten.

Wir dürfen aber umgekehrt auch nicht davon ausgehen, dass automatisierte Verfahren der quantitativen Textanalyse und genereller Verfahren des maschinellen Lernens eine Gefahr für die qualitative Sozialforschung darstellen (Mills 2018; Strong 2014), selbst wenn sie die Möglichkeit bewerben, Phänomene vorherzusagen. Wir vertreten in diesem Buch vielmehr die Ansicht, dass automatisierte Verfahren den Methodenkanon der Soziologie bereichern, sofern sie gegenstandsangemessen verwendet werden, bestehende Methoden ergänzen und sie stets hinsichtlich ihrer Grenzen kritisch reflektiert werden. Denn wir dürfen nicht vergessen, dass wir als Forscher*innen immer mit Erwartungen, Erfahrungen und mitunter sogar Vorurteilen in den Forschungsprozess eintreten und diesen strukturieren. Das gilt für alle Schritte: für die Wahl der Theorie, der Forschungsfragen, der Auswahl der Texte, Methoden und deren Interpretation gleichermaßen.

---

daran geknüpfte Datenstrukturen im deutschsprachigen Raum (Heiberger und Riebling 2016; Riebling 2018).

Hier können uns automatisierte Methoden der quantitativen Textanalyse helfen, indem sie als eine Art Korrektiv wirken, das unseren Blick auf Texte, Inhalte oder Strukturen lenkt, denen wir keine Beachtung geschenkt oder die wir vollkommen anders interpretiert hätten.

Das gilt im Besonderen für die Aspekte der Datenauswahl und Datenerhebung, wenn wir bedenken, welche ungeheuren Datenmassen in den letzten Jahren durch Nutzer*innen im Internet oder durch Geräte (wie z. B. Ihre Smart-Watch oder Überwachungskameras im öffentlichen Raum) produziert wurden (Cao et al. 2020). Kein Mensch kann diese Datenmengen allein sichten, geschweige denn deren Sinngehalt interpretieren. Zudem können Sie ebenso schwerlich abschätzen, welche Datenvorselektion Algorithmen wie zum Beispiel die Google-Suche oder Recommender-Systeme auf Plattformen wie Spotify für Sie auf Basis Ihres Surfverhaltens (und das ähnlicher Nutzer*innen) treffen und wie hierdurch die Aufmerksamkeit auf bestimmte Inhalte gelenkt wird, die für Sie attraktiv erscheinen sollen (Seaver 2019). Wenn wir davon ausgehen, dass das, was uns an Informationen angezeigt wird, durch Algorithmen und Computerprogramme vorselektiert wird und wir selbst eine Vorselektion basierend auf unseren Erfahrungen und Entscheidungen treffen müssen, wäre es dann nicht besser, zunächst Methoden anzuwenden, mit deren Hilfe wir keine Auswahl treffen müssen, sondern alle uns verfügbaren (und nicht vorselektierten!) Daten zu Nutze machen? Die uns dadurch sogar potenziell helfen können, über unseren Tellerrand hinauszublicken und gegebenenfalls auf Phänomene aufmerksam zu machen, die sonst aus Zeitgründen oder aufgrund der von uns angelegten Perspektive unbemerkt geblieben wären? Entsprechend gehen wir davon aus, dass Sie die durch die Verwendung automatisierter Verfahren gewonnenen Erkenntnisse dazu nutzen können, um weitere qualitative Untersuchungen zu informieren und umgekehrt die Methoden der qualitativen Textanalyse nutzen können, um an den latenten Sinngehalt der in Kommunikationen transportierten Muster zu gelangen.

2.3 Gütekriterien der Inhaltsanalyse

Die Inhaltsanalyse fokussiert darauf, manifeste und latente Muster in Textdaten zu erkennen, indem deduktive und/oder induktive Kategorien mittels digital unterstützter, teilautomatisierter oder vollautomatisierter Verfahren entwickelt werden. Wie Muster in den Daten erkannt werden, steht im Zentrum jedes inhaltsanalytischen Verfahrens. Das Erkennen erfolgt dabei regelgeleitet und nach spezifischen Systematiken, die es gilt einzuhalten. Je nachdem, ob Sie dabei einer qualitativen oder quantitativen Logik folgen, gibt es Gütekriterien, die es zu beachten gilt. Für vollautomatisierte Verfahren gibt es zudem Gütekriterien, die sich auf Algorithmen beziehen.

Für die quantitative Inhaltsanalyse gelten die drei Gütekriterien quantitativer

Forschung, also Objektivität, Validität und Repräsentativität. Das Kriterium der Objektivität wird dabei unterteilt in Durchführungs-, Auswertungs- und Interpretationsobjektivität (Krebs und Menold 2019). Unter Durchführungsobjektivität ist die standardisierte Erhebung mittels z. B. validiertem (also erprobtem) Fragebogen gemeint. Die Standardisierung wird dadurch gewährleistet, dass alle Befragten identisch formulierte Fragen (Stimuli) erhalten (ebd.). Übertragen auf die quantitative Inhaltsanalyse fällt darunter das replizierbare Vorgehen bei der Datengewinnung, also z. B. die Festlegung von Suchstrings beim Abrufen von Literaturdaten oder die Auswahl von Hashtags bei Twitter-Daten. Die Auswertungsobjektivität beruht auf der nachvollziehbaren Dokumentation der Aufbereitung der Daten, also sind alle Daten vollständig und fehlerfrei bzw. welche Daten wurden nicht in die Auswertung aufgenommen? Für die quantitative Inhaltsanalyse bedeutet das z. B., dass dokumentiert wird, wie Textdaten in einzelne Sequenzen zerlegt wurden, welche Wortarten in die Analyse aufgenommen werden oder wie mit Rechtschreibfehlern umgegangen wird. Interpretationsobjektivität, so führen Krebs und Menold (2019, S. 490) aus, kann es nicht allumfassend geben, da „Interpretationen subjektiven Bewertungen (Werturteilen) unterliegen (können)". Umso wichtiger ist es für Sie, Ihre Interpretationen nachvollziehbar zu dokumentieren und in Ihrer Arbeit zu verschriftlichen (darauf kommen wir gleich noch ausführlicher).

Das zweite Gütekriterium quantitativer Forschung ist Reliabilität, also die Zuverlässigkeit einer Messung. Eine Messung ist reliabel, wenn sie zu zwei Zeitpunkten zu dem gleichen Ergebnis kommt, also auch von anderen Forschenden replizierbar (also wiederholbar) ist. Auch hier ist es entscheidend, dass genau dokumentiert wird, was und wie gemessen bzw. erhoben wurde. Um zu bestimmen, wie reliabel eine Messung bzw. Erhebung ist, wurden diverse Tests entwickelt. Auf diese Tests werden wir in den Kapiteln der quantitativen Inhaltsanalyse noch näher eingehen und anhand von Beispielen zeigen, wie dieses Gütekriterium angewandt werden kann.

Das dritte Gütekriterium ist die Validität. Kurz zusammengefasst meint Validität, dass Ergebnisse gemessen werden, die auch gemessen werden sollen. Das heißt, dass Sie im Vorfeld wissen müssen, was Sie eigentlich messen wollen, dass dies auch in Hypothesen artikuliert werden kann und in den richtigen Verfahren für Ihre Forschungsfrage zum Ausdruck kommt. Validität benötigt entsprechend immer Vorannahmen, die meist aus Theorien abgeleitet sind und bezieht sich „weniger auf ein Messinstrument, als vielmehr auf die Qualität der Schlussfolgerungen, die mit einem Messergebnis möglich sind" (Krebs und Menold 2019, S. 496). Die Validität einer Forschung bezieht sich entsprechend immer auf den gesamten Forschungsprozess und zielt auf die Passung zwischen Forschungsfrage, Theorie und Empirie sowie auf Forschungsmethoden.

Im Gegensatz zur quantitativen Forschung werden Sie in Methodenbüchern oftmals lesen, dass es für die qualitative Sozialforschung nicht *die* Gütekriterien

gibt, auf die sich alle Forschenden einigen können (Flick 2019). Warum und welche Berechtigung diese Diskussionen haben, wollen wir an dieser Stelle nicht weiter ausführen. Uns ist es in Bezug auf die qualitative Inhaltsanalyse wichtig, zwei Gütekriterien herauszustellen. Das sind die Gütekriterien Transparenz und Intersubjektivität.

Das Gütekriterium Transparenz bezieht sich auf zwei Ebenen: Zum einen auf den Erhebungs- und Auswertungsprozess und zum anderen auf die Verschriftlichung (Flick 2019). Für den Erhebungs- und Auswertungsprozess ist es zentral (wie wir dies auch detailliert noch in Kapitel 3 beschreiben), den Forschungsprozess zu dokumentieren (exemplarisch Steinhardt 2015). Also warum wurde sich für welche Methode entschieden? Warum wurden welche Daten erhoben (Sampling-Strategie)? Wie wurden die Daten aufbereitet? Wie wurden die Daten ausgewertet, welche Schritte wurden dabei gegangen? Welche Ergebnisse wurden produziert und wie wurden diese interpretiert? All das sind Fragen, die für die Transparenzschaffung Ihres Forschungsprozesses im Methodenteil Ihrer Arbeit dokumentiert werden sollten. Dazu ist es hilfreich, sich ausführliche Notizen zu Ihren Überlegungen zu machen, z. B. in einem Feldtagebuch, das durchaus auch die Notiz-App Ihres Smartphones sein kann. Wichtig ist, dass eine dauerhafte Dokumentation stattfindet, die Ihnen dabei hilft, Ihr Vorgehen transparent und nachvollziehbar aufzuschreiben, sodass es die Leser*innen Ihrer Arbeit verstehen können.

Daran schließt sich das zweite Gütekriterium an: die Intersubjektivität. In der qualitativen Forschung steht im Zentrum, wie aus Daten manifeste, aber vor allem auch latente Inhalte extrahiert werden können. Das erfolgt durch Interpretation des Materials. Dabei muss sich die bzw. der Forscher*in darüber im Klaren sein, dass die Interpretationen, die sie bzw. er macht, immer mit den eigenen Annahmen, dem eigenen theoretischen Vorwissen als auch Alltagswissen und den eigenen Lebenserfahrungen verbunden sind (Strauss 2007). Als Forscher*in kann ich nicht „Tabula Rasa" in meinem Kopf machen und „objektiv" an die Daten herangehen (Steinhardt 2015). Vielmehr ist es zentral, die eigenen Annahmen und vor allem Interpretationsansätze deutlich zu machen (Charmaz 2011). Also: Warum habe ich was wie interpretiert, sodass jemand, der meine Interpretationen liest, diese nachvollziehen kann und als sinnvoll einschätzt. Für das Erreichen von Intersubjektivität ist es immer hilfreich, die Interpretationsergebnisse, also z. B. warum welche induktiven Kategorien entwickelt wurden, mit anderen Personen zu diskutieren und deren Interpretation zu erfahren. Bei hermeneutischen Verfahren ist die Interpretation in Gruppen z. B. ein zentraler Kern des Verfahrens. Ebenso ist die Interpretation in einer Gruppe bzw. mit einer anderen Person für die qualitative Inhaltsanalyse, aber auch für die Interpretation der Sentiments bzw. Topics bei vollautomatisierten Verfahren der quantitativen Inhaltsanalyse, sehr sinnvoll.

In unseren Augen sind das die zentralen Gütekriterien, die es bei der qua-

litativen Inhaltsanalyse zu beachten gilt. Wie Sie merken, ist das von Mayring (2010) eingeführte Gütekriterium der Interkoderreliabilität nicht unter den von uns herangezogenen Gütekriterien. Das hat den Grund, dass gerade für die Logik einer qualitativen Forschung ein quantitatives Maß nicht sinnvoll ist. Interkoderreliabilität bedeutet, dass zwei oder mehr Forscher*innen dasselbe Datenmaterial auswerten und dass dann verglichen wird, welche Kategorien und dazugehörige Kodes sie gleich (= Standardisierung) und anders (= Standardfehler) vergeben haben. Reliabilität bezeichnet die Zuverlässigkeit bzw. die Verlässlichkeit von (quantitativen) Messungen basierend auf der Annahme, dass eine Messung potenziell wiederholbar ist „bei unterstellter Stabilität des zu messenden Sachverhalts" (Diaz-Bone 2015, S. 169). Interkoderreliabilität adressiert also die Messgenauigkeit und Messfehler der kodierenden Subjekte, also der Forscher*innen. Folglich ist der Bezugspunkt von Interkoderreliabilität das kognitive Analysepotenzial der kodierenden Subjekte zwecks intersubjektiver Nachvollziehbarkeit deren manueller Kodiertätigkeit.

Laut Mayring (2010, S. 51) soll Interkoderreliabilität einen Beitrag zur Gültigkeit (im Soziolekt – Umgangssprache von Soziolog*innen: Validität) der Ergebnisse leisten, welche nach Flick (2019, S. 476 ff.) dem qualitativen Gütekriterium „Validierung durch Kommunikation" im Sinne der „Transparenz der Vorgehensweisen" (Flick 2019, S. 483) der Erzielung von Ergebnissen zuzuordnen ist. Folglich ist kommunikative Validierung, beispielsweise mit der*dem Betreuer*in der Qualifikationsarbeit, mit Kommiliton*innen oder mit Kolleg*innen, fester Bestandteil des gesamten qualitativen Forschungsprozesses vom Forschungsdesign über Materialauswahl und Erhebungsinstrumente bis zur Darstellung von Ergebnissen (siehe auch Strübing et al. 2018).

Zu diesen allgemeinen Gütekriterien kommen für den Bereich der automatisierten Textanalysen Qualitätskriterien hinzu, die die Genauigkeit prüfen, mit der die in der Analyse verwendeten Algorithmen ein bestimmtes Ergebnis korrekt erkennen oder vorhersagen können. Viele Studien aus dem Bereich des *machine learnings* sind mit Verzerrungen behaftet (Myrtveit et al. 2005; van Atteveldt et al. 2021), adressieren diese aber selten, da die Vorhersagekraft verschiedener Algorithmen bei bekannten Datensätzen in der Regel im Vordergrund steht. Entsprechend beziehen sich die Gütekriterien auf Maßzahlen und die Qualitätssicherung auf einige Vorgehensweisen, die hier in Kürze dargestellt werden. Keine Sorge, wir verschonen Sie mit statistischen Formeln.

Lassen Sie uns nun diese Maßzahlen anhand des Beispiels mit der alten Dame aus Kapitel 2.2 herleiten. Stellen Sie sich vor, Sie wollen die Stimmung messen, die der Kommunikation zwischen den beteiligten Personen zugrunde liegt. Sie lesen den transkribierten Text der alten Dame, beispielsweise „Danke, mir geht es gut" oder „Danke, mir geht es schlecht", und ordnen dann erstere Antwort einem neutralen Sentiment (Stimmungslage) zu, letzteres einem negativen Sentiment. Auf erstere Deutung der Stimmungslage sind Sie gekommen, weil es sich um eine

sozial erwünschte Antwort handelt. Ihr Algorithmus, der die Stimmung erkennen soll (siehe Sentiment-Analyse, Kapitel 10), würde die erste Aussage in ein positives, die zweite Aussage in ein negatives Sentiment zuordnen. Nun stimmen Sie mit dem Algorithmus bei der zweiten Aussage (negatives Sentiment) überein, bei ersterer allerdings nicht. Wir verlassen uns hier auf Ihren Verstand und unterstellen dem Algorithmus eine Fehleinschätzung des ersten Sentiments.[9]

Bei den Gütekriterien geht es nun darum, diese Fehleinschätzungen zu quantifizieren, d. h. abzählbar zu machen. Konkret ließe sich am vorliegenden Beispiel die Frage beantworten, wie häufig der Algorithmus Ihrer Sentiment-Analyse fälschlich eine positive Stimmungslage identifiziert hat, obwohl das Sentiment neutral oder negativ ausfallen würde. Hierfür wird die sogenannte Precision bzw. Präzision berechnet, welche der Anteil der korrekt gemessenen Ausgaben Ihres Algorithmus an der Gesamtzahl aller korrekten Ausgaben und falsch positiven Ausgaben Ihres Algorithmus misst (hier: korrekt zugeordnetes Sentiment + falsch zugeordnetes positives Sentiment). Daneben gibt es das Recall-Maß, das den Anteil korrekt zugeordneter Ausgaben im Verhältnis zu korrekt zugeordneten Ausgaben und falsch zugewiesenen negativen Ausgaben misst. Das würde in unserem Falle beispielsweise bedeuten, dass wir korrekt zugeordnete Aussagen durch die Summe korrekt zugeordneter Sentiments und fälschlicherweise als negatives Sentiment zugeordneten Aussagen teilen und dann diese Maßzahl erhalten würden. Daneben gibt es den sogenannten F1-Score, der aus Precision und Recall zusammengesetzt ist und damit widerspiegelt, wie klein beide Fehlertypen in der Messung sind. Zusätzlich gibt es sogenannte Konfusionsmatrizen, die bei mehreren Abstufungen (z. B. stark negatives, negatives, neutrales, positives und sehr positives Sentiment) die korrekten Zuordnungen des Algorithmus zu zuvor erhobenen menschlichen Kategorisierungen aufzeigen. Darüber hinaus

9 Bei vorangegangenen Bewertungen durch menschliche Koder würde man hier von einer „Ground Truth" sprechen, das heißt einer offenen und offensichtlichen Wahrheit. Dabei ist auch hier Vorsicht geboten! So ist es beispielsweise einfach, eine Aussage wie „Der Stuhl ist rot" prüfen zu lassen, sofern sich ein Stuhl oder mehrere Stühle mit unterschiedlichen Farben in einem Raum oder in einer Bilddatei befinden und Sie die Möglichkeit haben, auf diesen Stuhl zu verweisen. Je abstrakter aber die zu prüfenden Konzepte werden (z. B. Emotionen, makrotheoretische Konzepte), desto mehr Interpretationen und Subjektivität fließen in die Sichtweise ein, die Ihr Computer bzw. Ihr Algorithmus als objektiv „wahr" für die eigene Bewertung heranziehen wird. Anders ausgedrückt ist mit zunehmenden Abstraktionsgrad die Ground Truth auch im stärkeren Maße sozial konstruiert und von den Bewertenden abhängig (siehe Grosman und Reigeluth 2019; Jaton 2021). Entsprechend können Sie nicht davon ausgehen, dass die Ergebnisse Ihrer automatisierten Analyse verallgemeinerbar sind. Im Gegenteil, viele Algorithmen kaschieren und verschleiern, auf welcher Basis die Grundlage der Kategorisierungen berechnet worden ist, was die Reliabilität und Validität der Anwendungen weiter senkt.

zeigt der Matthews Korrelationskoeffizient (Yao und Shepperd 2020) die Übereinstimmung zwischen Algorithmus und menschlicher Kategorisierung an.[10]

Ergänzend müssen Sie sich stets vor Augen führen, dass die meisten Analysen aus dem Bereich der automatisierten Textanalyse auf Korpora zurückgreifen, die aus einer Datenbank gewonnen werden. Obwohl diese Korpora in der Regel über eine hohe Anzahl von Texten verfügen, die bis in die Millionen hineingehen können, so handelt es sich streng genommen um eine Einzelfallanalyse. Entsprechend gibt es Probleme mit der Übertragbarkeit auf andere Fälle bzw. Datenkorpora. Um diesen Problemen zu begegnen, bedienen sich Forscher*innen aus dem Bereich der Computational Social Sciences eines Tricks. Sie teilen den Datensatz in ein Trainings- und ein Testset auf. Die Modelle werden, wie der Name schon andeutet, nur mit den Daten des Trainingssets berechnet. Danach wird geprüft, inwiefern das Modell die gleichen Ergebnisse beim Testset liefert. Darüber hinaus gibt es die Möglichkeit zur Kreuzvalidierung. Dabei handelt es sich um ein Verfahren aus der Statistik, bei dem ein Datensatz in verschiedene (z. B. zehn) Stichproben unterteilt wird und die Ergebnisse bei einer Gruppe (z. B. die Zuordnung von Aussagen zu Sentiments) in einer Stichprobe auch in allen anderen Stichproben zu finden sind. Dies wird für alle Stichproben durchgeführt (Browne 2000). Es werden somit künstlich Fälle geschaffen, mit deren Hilfe die Verallgemeinerbarkeit der Analyseergebnisse geprüft werden soll. Inwiefern das gelingen kann, sei dahingestellt, da wir im Falle dieser Verfahren von der gleichen Chance aller ausgehen müssen, diese Daten zu generieren, was de facto aufgrund des Digital Divides[11] nicht gegeben ist (Baur et al. 2020).

10 Vielleicht haben Sie vom Korrelationskoeffizienten im Rahmen Ihrer Statistikvorlesung gehört. Dieser nimmt Werte im Bereich von +1 bis −1 ein. In unserem Falle würde +1 eine perfekte Übereinstimmung zwischen Algorithmus und menschlicher Kategorisierung anzeigen, 0 eine vollkommen zufällige Übereinstimmung und −1 eine völlige Diskrepanz zwischen der Vorhersage des Algorithmus und den menschlichen Kategorisierungsversuchen.

11 Darunter subsummiert man Ungleichheiten beim Zugang von Nutzer*innen zu Datenbanken, Unterschieden in der technischen Ausstattung bzw. im Internetzugang und Kenntnissen im Umgang mit dem Internet und darin befindlichen Anwendungen.

3. Spezifika von Daten: Möglichkeiten und Grenzen sozialwissenschaftlicher Inhaltsanalysen

In Kapitel 3 werden die Spezifika von Daten und damit verbundene Möglichkeiten und Grenzen für sozialwissenschaftliche Inhaltsanalysen forschungspragmatisch und erkenntnisorientiert dargestellt. Unterschieden werden hierbei die Qualitäten von forschungs- und prozessproduzierten Daten von der Konzeption (unter Berücksichtigung von Datentypen, Forschungszielen und der Operationalisierung) über die Datengenese (inklusive Stichprobenziehung, Datenerhebung und Datenart), Analysemöglichkeiten (aufgrund von Datenverwahrung und wissenschaftlichen Analysearten) und Ergebnisverwertung bis zur Archivierung von Daten. Die Spezifika von empirischen Daten prägen nicht nur den Forschungsprozess, sondern auch die (schriftliche) Präsentation der Erkenntnis. Diese wird abschließend in einem groben Gliederungsschema wissenschaftlicher Ausarbeitungen zusammengefasst, welches von der Haus- über Bachelor- und Master- bis zur Doktorarbeit angewandt werden kann.

3.1 Einleitung

Sozialwissenschaftliche Forschung ist immer von Erkenntnisinteresse geleitet. Das Erkenntnisinteresse muss durch eine konkrete Fragestellung und vor allem bei quantitativen Untersuchungen aus einer Theorie, existierender Forschung oder Beschreibung eines sozialen Phänomens abgeleitet und durch Hypothesen präzisiert werden. Durch Theorie, existierende Forschung und Beschreibung eines sozialen Phänomens wird auch der empirische Untersuchungsgegenstand näher bestimmt. Je nach Erkenntnisinteresse und Untersuchungsgegenstand gibt es spezifische Analysemöglichkeiten und -grenzen, die sowohl durch die Qualität der Daten als auch durch den Datenzugang gegeben sind.

Die Qualität der Daten bezieht sich auf die Art und Eigenschaften von Daten. Wir können beispielsweise zwischen Daten unterscheiden, die als Text, als aufgezeichnetes Video einer Diskussion einer Expert*innenrunde im Fernsehen, als Bildmaterial, als Datentabelle oder als (strukturierte oder unstrukturierte) Datentypen vorliegen. Mit jeder Datenform sind eigene Methoden und eigene Herangehensweisen zur Datenerhebung und Datenaufbereitung verknüpft. So werden Sie bei Text-, Video- oder Bilddaten schwerlich statistische Berechnungen durchführen können, es sei denn, Sie investieren viel Zeit, um eine eigene Datentabelle

mit Merkmalen dieser Dateien anzulegen. Datentabellen hingegen erlauben es Ihnen recht schnell, statistische Berechnungen durchzuführen, eignen sich in der Regel aber nicht dafür, textanalytisch ausgewertet zu werden. Das liegt an der numerischen Qualität der Informationen in den Datentabellen, die Ihnen dadurch Rechenoperationen wie die Auszählung von Häufigkeiten, prozentualen Verteilungen und Berechnung von Zusammenhängen erlauben.

Die Grenzen von zahlenbasierten Untersuchungen ergeben sich jedoch ebenso aus der numerischen Qualität von Zahlen. Zahlen sprechen niemals für sich. Selbst wenn Zahlenkorrelationen signifikant sind, was bedeutet, dass Zahlenwerte von gemessenen Variablen mit hoher Wahrscheinlichkeit nicht zufällig gemeinsam auftreten. Im Gegenteil müssen die Zahlen und sich aus ihnen ergebende Zusammenhänge von den Forschenden im sozialen Kontext (siehe Tabelle 2.1, Analyseschritt 1) verstanden (siehe Tabelle 2.1, manifester Inhalt, Analyseschritt 2a) und dann erklärt, gedeutet und interpretiert (latenter Inhalt) werden, sodass die zahlenbasierten Ergebnisse für die Adressat*innen der Untersuchungsergebnisse verständlich werden. Erklärungen, Deutungen, Interpretationen und Schlüsse im Hinblick auf Zahlen basieren bei deduktiven Untersuchungen auf derselben Theorie, welche die Untersuchungsgestaltung leitet.

Bei induktiv-qualitativen Untersuchungen sollte Theorie zur Reflexion der Ergebnisse und deren latenten Inhalten herangezogen werden (siehe Kapitel 2, 4 und 6). Adressat*innen der Untersuchung können andere Sozialwissenschaftler*innen oder Publika ohne Statistikkenntnisse sein. In beiden Fällen gilt für die Ergebnispräsentation eine den Adressat*innen angemessene Sprache zu wählen, wobei auch Sozialwissenschaftler*innen gut verständliche und nachvollziehbare Erklärungen in Publikationen und Präsentationen erwarten, welche die Zahlen (z. B. in Tabellen) und Testabkürzungen ergänzen (siehe Tabelle 2.1, Analyseschritt 3).

Im Vergleich zur Absolutheit oder Endgültigkeit von Zahlen können und müssen alle weiteren in sozialwissenschaftlichen Untersuchungen verwendeten Textdaten zunächst von einer Rohform in eine andere Datenart überführt werden, die für qualitative, semiautomatisierte quantitative und vollautomatisierte quantitative Textanalysen geeignet sind. Für die Auswertung werden Sprachaufnahmen (z. B. Interviews oder Gesang) und Videos transkribiert oder zusammenfassend als Text verschriftlicht. Folglich ist Text neben Zahlen die für die Sozialwissenschaften zweite wesentliche Datenart. Selbstverständlich kann Text auch in Zahlen übersetzt und damit noch weiter von der Rohform entrückt werden, als dies bei Transkripten oder Zusammenfassungen der Fall ist (siehe Kapitel 9, 10 und 11).

Durch die Übersetzung bzw. Umwandlung von Text in Zahlen verändert sich jedoch die Qualität der qualitativen Daten. Damit ist nicht gemeint, dass die Qualität besser oder schlechter wird, denn auch wenn ein Wort eine Eigenbedeutung hat, muss auch die Bedeutung im Untersuchungszusammenhang erst verstanden

werden, bevor die*der Forschende erklärt, deutet, interpretiert und Schlüsse zieht. Die Qualitätsveränderung bei der zahlenförmigen Abbildung eines Wortes bzw. der Häufigkeit des Wortes in einem Dokument oder einem umfänglicheren Dokumentenkorpus a) entreißt das Wort aus der Textförmigkeit und b) bildet einen Wortschein als Zahl ab. Der Wortschein ist analog zum Geldschein zu begreifen. Mit einem 5-Euro-Schein können Waren im Gegenwert von fünf Euro erstanden werden. Jedoch ist der 5-Euro-Schein an sich nicht fünf Euro wert. Die Bank händigt den 5-Euro-Schein aufgrund des Gegenwertes auf einem Konto aus. Wie die Differenz von Qualität und Quantität bei einem 5-Euro-Schein, so bilden die in Zahlen umgewandelten Worte nicht objektiv die Qualität eines Wortes oder von Text insgesamt ab, sondern stellen einen Wert dar, der für Berechnungen bzw. in Statistiken genutzt werden kann. Folglich ist die Qualität nicht eine Objektivierung von Text, sondern eine zahlenbasierte Metaanalyse von Text. Analog zu *close* und *distant reading* (Moretti 2000; 2013), auf Deutsch etwa vertieftes und mit Abstand Lesen, kann Text bzw. Worten die Qualität von vertieftem Lesen und Zahlen die Qualität von Abstandslesen zugesprochen werden. Für das wissenschaftliche Verstehen und das daraus resultierende Erklären, Deuten, Interpretieren und Schließen müssen die Worte als Zahlen kontextualisiert und, je nach Erkenntnisinteresse, mit der entsprechenden Methode analysiert werden. Die qualitativen Grenzen der zahl-abstrahierten Analyse von Text weist jedoch gleichzeitig auf die Möglichkeiten der sozialwissenschaftlichen Analyse insbesondere von großen Textmengen hin, welche händisch nicht im selben Umfang oder nur selektiv analysiert werden können.

3.2 Merkmale von forschungs- und prozessproduzierten Daten

Wie bei Zahlendaten für die quantitative bzw. Text für die quantifizierende Sozialforschung hängen die Analysemöglichkeiten und -grenzen der qualitativen Datenarten vom Entstehungszusammenhang ab. Beim Entstehungszusammenhang wird grob zwischen forschungs- und prozessproduzierten Daten unterschieden. Forschungsproduziert bedeutet, dass die Planung und Datenerhebung nach wissenschaftlichen Maßstäben und unter Einhaltung von Gütekriterien erfolgt ist (siehe Kapitel 2.3). In der Forschung produzierte Daten haben den Primärzweck, wissenschaftlich verwertbare Ergebnisse zu produzieren. Praktische Anwendbarkeit, beispielsweise für die sogenannte evidenzbasierte Politikberatung, sind gegebenenfalls ein sekundärer Zweck, da die empirischen Belege (Evidenz) der Beratungsleistung durch wissenschaftliche Kriterien legitimiert werden. In der Regel werden forschungsproduzierte Daten basierend auf einem bestimmten Erkenntnisinteresse produziert. Diese Erhebungen zwecks Primäranalyse, also erstmaliger Analyse, sind auf das zu untersuchende soziale Phäno-

men zugeschnitten. Entsprechend sollte gut nachvollziehbar dokumentiert sein, welche Entscheidungen im Forschungsprozess getroffen wurden. Typische Entscheidungen sind beispielsweise die Auswahl von Merkmalen eines untersuchten Phänomens, damit der Fragebogen oder der Leitfaden sowie die durchschnittliche Erhebungszeit kürzer werden. Darüber hinaus ermöglicht Ihnen der Fokus auf wenige Merkmale ein tiefergehendes Verständnis für den fokussierten Teil des Phänomens, was es Ihnen zugleich ermöglicht, neue Erkenntnis zu erzeugen. Das ist meist schwieriger zu bewerkstelligen, wenn Sie sich nicht fokussieren, sondern alle Aspekte inklusive kleinteiliger Detailfragen im Auge behalten. Im schlimmsten Falle stellen Sie am Ende Ihres Forschungsprozesses fest, dass Sie keine Zeit mehr haben, um alle Aspekte des Phänomens für Sie oder Ihre Gutachter zufriedenstellend zu untersuchen. Bedenken Sie bitte, dass großangelegte Untersuchungen meist in Monographien mit mehreren hundert Seiten münden und Jahre, manchmal sogar mehr als ein Jahrzehnt andauern. Fragen Sie sich also, ob Sie wirklich die Zeit haben, ein Phänomen in allen für Sie interessanten Facetten zu beleuchten oder nicht.

Dennoch kann bei Primäranalysen (siehe Box 3.1) ermöglichenden Erhebungen davon ausgegangen werden, dass in Übereinstimmung mit der Fragestellung und/oder den Hypothesen eine relativ uneingeschränkte Auswertung möglich ist. Im Gegensatz zu Primäranalysen werden bei Sekundäranalysen stärkere Einschränkungen in Kauf genommen. In der Regel können mit Sekundäranalysen keine umfassenden, jedoch hinreichende Erkenntnisse zur manifesten (Was-Frage) und latenten Inhaltsanalyse (Warum-Frage) eines sozialen Phänomens gewonnen werden. Der Nachteil der teilweisen Einschränkung des Erkenntnispotenzials – die Daten wurden für eine andere Untersuchung erhoben oder sind nicht mehr aktuell – wird für Wissenschaftler*innen dadurch aufgewogen, dass sie keine aufwendige Erhebung planen und durchführen müssen. Wie für Primär- muss auch für Sekundäranalysen eine Datenkontrolle und gegebenenfalls Datenbereinigung durchgeführt werden. Auch hier gilt, dass Sie sich gut überlegen sollten, auf welche Merkmale Sie Ihre Untersuchung von Sekundärdaten hin fokussieren. Sie müssen sich nämlich stets fragen, inwiefern das Forschungsdesign, Theorien, oder auch subjektive (Vor-)Urteile in die Generierung der von Ihnen ausgewählten Sekundärdaten fließen. Zu der Komplexität des für Sie interessierenden, aber schon vorausgewerteten Phänomens, tritt also die Komplexität hinzu, dass Sie den Forschungsprozess und die Perspektive der Forscher*innen nachvollziehen müssen, die den Blick schon auf bestimmte Aspekte Ihres Phänomens geworfen haben.

Box 3.1: Kurzdefinitionen Primär- und Sekundäranalyse

Primäranalyse: Eine Primäranalyse ist die Erstauswertung eigens innerhalb eines Forschungsprozesses erhobener Daten.

Sekundäranalyse: Eine Sekundäranalyse ist die erneute Auswertung von Daten, die durch andere Forscher*innen oder prozessorientiert von Unternehmen erhoben und bereits ausgewertet wurden.

Die Berücksichtigung der Primär- und Sekundäranalysemöglichkeiten und

Tabelle 3.1 Merkmale von Daten nach ausgewählten Erhebungsarten (Fortsetzung nächste Seite)

	Forschungsproduzierte Daten		Prozessproduzierte Daten		
	Groß-n Fallzahlen Untersuchungen	*Klein-n* Fallzahlen Untersuchungen	Dokumente als (Massen-)Daten	Webbasierte Massendaten (Big Data)	Administrative Massendaten (Big Data)
Konzeption					
Datentypen	Fragebogenerhebung	Beobachtung/Befragung/Diskurs/Vergleich usw.	(Dreh-)Bücher/Wahlprogramme/Musik/Videos und ähnliches	Geschäftsbedingungen von Organisationen/Unternehmen	Gesetze/Verordnungen/Rechtstexte
Ziel	Beantwortung Forschungsfrage(n)	Beantwortung Forschungsfrage(n)	Kommunikation (z. B. Unterhaltung)	Information (Datensammlung als sekundäres bzw. Beiprodukt)	Datenprozessierung gemäß gesetzlicher Vorschriften
Operationalisierung	Deduktive Reflexion theoretischer Basis	Deduktive Reflexion/induktive Schaffung theoretischer Basis	Uneinheitlich/abhängig von Spezifika der Medien	Uneinheitlich/abhängig von Spezifika der (sozialen) Medien	Rechtsvorschriften
Datengenese					
Stichprobenziehung	Vollerhebung/Teilerhebung (bewusste und Zufallsauswahl)	Fallauswahl	Dokumentenauswahl	Datenbearbeitung als Datensammlung	Benachrichtigung/Fall (z. B. Gericht)
Datenerhebung	Befragungspopulation	Subjekte (z. B. Interviewte*r)/Objekte (z. B. Länder)/soziale Phänomene	Dokumente	Datenarrangements	Datenbank/-archiv
Datenart	Zahlen	Audios/Bilder/Texte/Videos/Zahlen	Audios/Bilder/Texte/Videos/Zahlen	Audios/Bilder/Texte/Videos/Zahlen	Audios/Bilder/Texte/Videos/Zahlen

	Forschungsproduzierte Daten		Prozessproduzierte Daten		
	Groß-n Fallzahlen Untersuchungen	*Klein-n* Fallzahlen Untersuchungen	Dokumente als (Massen-)Daten	Webbasierte Massendaten (Big Data)	Administrative Massendaten (Big Data)
Archivierung					
Vollständig	Ja	Ja	Ja	Teilweise	Ja
Analysen					
Datenverwahrung	Wissenschaftler*innen	Wissenschaftler*innen	Individuen/Kollektive/Organisationen als Autor*innen	Datensammel-/Informationsdienstleistungsunternehmen	Verwaltungsangestellte/Institutionsangehörige
Datenanalyse	Quantitative Analysemethoden (z. B. multivariate Statistik)	Qualitative Analysemethoden (z. B. Dichte Beschreibung)	Qualitative/quantitative Analysemethoden	Datenaufarbeitung vor quantitativer Analyse notwendig	Zweckgebundene Analyse
Wissenschaftliche Analysearten	Primär- und Sekundärauswertungen	Primär- und Sekundärauswertungen	Sekundärauswertungen	Sekundärauswertungen	Sekundärauswertungen
Anwendung					
Ergebnis	Wissenschaft (und teilweise praktische) Implikationen (z. B. Handlungsanleitung)/Schlussfolgerungen	Wissenschaft (und teilweise praktische) Implikationen (z. B. Handlungsanleitung)/Schlussfolgerungen	Begrenzter Nutzen, wenn Grundgesamtheit unklar und/oder fehlende Repräsentativität	Begrenzter Nutzen, wenn Population unklar und fehlende Repräsentativität	Praktische Bedeutung: Verwaltung, Politik, Regierungs- bzw. Steuerungszwecke

Quelle: Graeff und Baur (2020, S. 262); übersetzt und ergänzt durch Autor*innen

-einschränkungen verdeutlichen, dass ein wissenschaftlicher Forschungsprozess je nach Datentyp und Erhebungsart teilweise bereits zu Beginn die zu erzielenden Ergebnisse in Abgleich mit dem Erkenntnisinteresse berücksichtigen muss. Wie in Tabelle 3.1 zusammengefasst, ist in Abhängigkeit von der Erhebungsart das Forschungsdesign (siehe Box 3.2) spezifisch zu gestalten. Im Gegensatz zu prozessproduzierten Daten, kann die Erhebungsart bei der Primärerhebung forschungsproduzierter Daten nach dem Erkenntnisinteresse ausgewählt werden. Der jeweiligen qualitativen und/oder quantitativen Methodenwahl(en) entsprechend, kann der Konzeptions-, Datengenese-, Analyse- und Ergebnisverwendungsprozess im Forschungsdesign festgehalten werden. Im Gegensatz zu Primärdatenauswertungen sind Sekundärauswertungen *messy business,* auf Deutsch etwa ein unordentliches, verfahrenes oder schwieriges Geschäft. Der Geschäftscharakter ist besonders deutlich bei webbasierten Massendaten, welche nicht uneingeschränkt, teilweise nur gegen Entgelt und in der Regel nur nach den Bedingungen des die Daten besitzenden Unternehmens für wissenschaftliche Untersuchungen erhoben werden können. Die Kategorien und die Bestimmungen für ausgewählte Erhebungsarten in Tabelle 3.1 werden im Folgenden ausführlich dargestellt.

Box 3.2: Kurzdefinition Forschungsdesign
Ein Forschungsdesign enthält grob die fünf Elemente Konzeption (Datentypen, Ziel und Operationalisierung), Datengenese (Stichprobenziehung, Datenerhebung und Datenart), Archivierung (vollständig oder nicht), Analysen (Datenverwahrung, Datenanalyse und wissenschaftliche Analysearten) und (Ergebnis-)Anwendung. Ein Forschungsdesign definiert einen Rahmen, welcher die theoretischen und methodischen Herangehensweisen sowie den Forschungsprozess transparent und das Erzielen der fragemotivierten Ergebnisse wahrscheinlich macht.

In Tabelle 3.1 ist als Beispiel forschungsproduzierter Daten der Datentyp Fragebogenerhebung für Untersuchungen mit großen Fallzahlen angegeben. Das n symbolisiert hierbei die Anzahl der untersuchten Fälle. Groß-n-Untersuchungen sind jedoch nicht auf die quantitative Sozialforschung begrenzt. Qualitative Groß-n-Untersuchungen wären beispielsweise Interviewstudien mit vielen Interviewten, wobei nicht nur die Anzahl der Interviews, sondern auch der Umfang, d. h. die Länge der Interviews, zu berücksichtigen ist. Folglich ist eine qualitative Studie mit 135 Interviews sicher als Groß-n-Untersuchung zu bezeichnen, ebenso wie 35 einstündige Interviews, welche transkribiert einhundert Seiten plus n Seiten an Text produzieren. Damit sind weiterhin selbstverständlich qualitative Primäranalysen durchzuführen, jedoch könnte eine ergänzende quantifizierende Auswertung ebenfalls sinnvoll sein. Grundsätzlich sind Klein-n-Untersuchungen Analysen, welche mit den Methoden der Objektiven Hermeneutik, Dokumentarischen Methode und den Protokollen der teilnehmenden Beobachtungen durchgeführt werden (siehe Kapitel 2). Als Daumenregel können Studien mit zehn Fällen und weniger als Klein-n-Untersuchungen bezeichnet werden. Wie in der Sozialforschung üblich, gibt es jedoch auch die Kombination von „Groß-n-in Klein-n-Untersuchungen“ (Schneijderberg und Götze 2021), beispielsweise bei der Durchführung einer Fragebogenerhebung in vier Ländern basierend auf

einem theoretisch begründeten Vergleich der bewusst als Stichprobe ausgewählten vier Länder. Ein weiteres Beispiel für eine Groß-n- in Klein-n-Untersuchung ist die Befragung aller Mitarbeiter*innen in vier Großunternehmen (< 500 Mitarbeiter*innen).

3.2.1 Forschungsproduzierte Daten: Datenerhebung

Unabhängig davon, ob Sie Unternehmen, Länder oder Befragungspopulation untersuchen, muss die für forschungsproduzierte Daten gewählte Stichprobe begründet werden. Der Wortteil *Ziehung* bei Stichprobe verweist auf eine Zufallsauswahl (z. B. durch Losverfahren). Obwohl der Begriff heutzutage allgemeingültig scheint, gibt es ganz spezielle Formen der Stichprobenziehungsverfahren für standardisierte und nicht-standardisierte Verfahren, worauf zwei Kapitel aus einem Methoden-Sammelband hinweisen: „Stichprobenziehung in der qualitativen Sozialforschung" (Akremi 2019) und „Stichprobenziehung in der quantitativen Sozialforschung" (Häder und Häder 2019). Unangemessen erscheint der Begriff *Ziehung* jedoch weder für Groß-n- noch für Klein-n-Untersuchungen. Zum Beispiel ist der Zufall bei der empirischen Sozialforschung stets mit am Werk, denn Mitarbeiter*innen von begründet ausgewählten Großunternehmen können die Teilnahme an der Fragebogenerhebung ebenso verweigern wie für ein Interview angefragte Personen. Die selbstgewählte Nichtteilnahme von zu der Befragung eingeladenen Personen wird als Selbstselektion bezeichnet.

Außer bei komplett zufälliger Auswahl der Befragten unterliegt jegliche Stichprobenziehung der sorgfältigen Auswahl und Begründung durch die*den Forschenden. Das ist umso eher der Fall, je stärker das Potenzial zur Selbstselektion oder zur selektiven Abdeckung einzelner Gruppen gegeben ist, obwohl ein Phänomen weitaus größere Personenkreise betreffen kann. Ein Beispiel für eine nicht-zufällige Stichprobe stellt das Schneeballverfahren (z. B. eine interviewte Person empfiehlt eine oder mehrere weitere zu Befragende) dar, ein weiteres ist ein sogenanntes *convenience sampling* (auf Deutsch: Bequemlichkeitsstichprobe), bei dem die Fälle erhoben werden, auf die leicht zurückgegriffen werden kann.

Die begründete Auswahl für die Stichprobenziehung beginnt bei der Beschreibung des zu untersuchenden sozialen Phänomens und Formulierung der Fragestellung sowie gegebenenfalls der Hypothesenbildung. Teil der begründeten Auswahl für die Stichprobenziehung ist die Identifikation der zu Befragenden, unabhängig davon, ob für eine Groß-n-Fragebogenerhebung oder für eine Klein-n-Interview- oder Beobachtungsstudie. Handelt es sich bei der Untersuchung um eine Groß-n-Untersuchung (z. B. Fragebogenerhebung bei Professor*innen und wissenschaftlichen Mitarbeiter*innen zu Arbeitsbedingungen in der Wissenschaft) in einer Klein-n-Untersuchung (z. B. Vergleich deutschsprachiger Länder

Deutschland, Österreich und Landesteile der Schweiz), so ist die quasi-natürliche Sprachraumbegrenzung nicht ausreichend als Begründung für die Fallauswahl.

Zuerst muss auch bei ähnlichen, jedoch in zumindest einem oder mehreren Merkmalen abweichenden Fällen die Vergleichbarkeit der Fälle überprüft und begründet werden (z. B. Seawright und Gerring 2008). Nehmen wir als Beispiel die jeweiligen Ausgestaltungen des wissenschaftlichen Feldes im deutschsprachigen Raum. Die Hochschulen in Deutschland, Österreich und der deutschsprachigen Schweiz haben sich über fast zwei Jahrhunderte nach dem sogenannten Humboldt'schen Universitätsideal entwickelt (Östling 2020), und in allen drei Fällen existieren im Hochschulsystem die Hochschultypen Fachhochschule und Universität. Die drei miteinander verglichenen Fälle sollten neben einigen Gemeinsamkeiten auch Unterschiede aufweisen, welche relevanten Einfluss auf das Erkenntnisinteresse „Arbeitsbedingungen in der Wissenschaft" haben könnten. Zwar sind alle drei Fälle föderale Bundesstaaten, jedoch unterstehen in Österreich die Hochschulen der Bundesregierung in Wien und nicht wie in Deutschland und der Deutschschweiz den Landesregierungen. Weiter kann an deutschschweizerischen Hochschulen ein Einfluss des in den französischsprachigen Landesteilen sogenannten Napoleonischen Hochschulideals angenommen werden. Beide Hochschulideale sind Idealtypen für forschungsbasierte (Humboldt) und Funktionseliten für den Staat (Napoleon) ausbildende Hochschulsysteme, welche in der sozialen Realität und unter den Bedingungen des globalen Hochschulmarktes (nach Anglo-US-Amerikanischen Vorbildern) kontinuierlichen kleinen und größeren Entwicklungen unterliegen.

Ergänzend zum das öffentliche Hochschulsystem umwandelnden Wissenschaftskapitalismus (Münch 2014) und damit einhergehender Einführung von Elementen der Unternehmenssteuerung (Vormbusch 2012), zum Beispiel Zielvereinbarungen und Qualitätsmanagement, wirken sich der kontinuierliche Anstieg der Studierendenzahlen auf die Arbeitsbedingungen bzw. deren Ausdifferenzierung aus. Das bedeutet, dass z. B. mehr Lehraufgaben für einen Teil von Wissenschaftler*innen und Hochschulen und mehr Forschungsaufgaben für andere Wissenschaftler*innen, beispielsweise an als exzellent ausgezeichneten Hochschulen, zugewiesen werden. Die sogenannte „Exzellenzstrategie" (Hartmann 2010; Münch 2007) zur Förderung der globalen Wettbewerbsfähigkeit ausgewählter Universitäten in Deutschland bildet eine weitere anzunehmende Einflussgröße auf die Arbeitsbedingungen in der Wissenschaft in Deutschland (Schneijderberg und Götze 2020), welche weder in Österreich noch der Deutschschweiz existiert. Die knappen Beschreibungen zu den drei Hochschulsystemen weisen darauf hin, dass bei der Untersuchung von sich gegebenenfalls angleichenden Rahmenbedingungen von Arbeitsbedingungen in der Wissenschaft für den Fallvergleich die historischen und politischen Zusammenhänge berücksichtigt werden müssen, hier wird in der quantitativen wie auch in der qualitativen Sozialforschung von Kontingenz gesprochen (Ebbinghaus 2005; 2009).

Die gezielte, zweckorientierte Klein-n-Fallauswahl ist einer Zufallsauswahl bei der Stichprobenziehung insbesondere dann vorzuziehen, wenn wenig Wissen über ein soziales Phänomen existiert (Akremi 2019; Suri 2011). Einzelfall- oder Klein-n-Beobachtungs- oder Interviewuntersuchungen eignen sich zur explorativen Ergründung eines neuen, neu entdeckten und existierenden, jedoch bisher nicht wissenschaftlich erforschten sozialen Phänomens. Zur vertieften qualitativen Wissensproduktion eignet sich beispielsweise die „Dichte Beschreibung" (Geertz 2002; siehe Box 3.3), welche die beiden Charakteristika der (ethnographischen) Beobachtung eines sozialen Phänomens mit der analytischen Untersuchung sozialer Organisationen und diese stützenden Strukturen kombiniert. Die analytische Untersuchung fußt auf einer systematischen Beschreibung von Mustern menschlichen Zusammenlebens in Gruppen. Ziel der „Dichten Beschreibung" ist die Erstellung eines Bedeutungsnetzes (*web of significance,* Geertz 2002, S. 12), welches die Ableitung einer wissens- und kulturkonstruktivistischen Theorie zur Untersuchung des sozialen Phänomens in vergleichbaren Fällen ermöglicht.

Box 3.3: Kurzdefinition dichte Beschreibung
Von Clifford Geertz (2002) als *thick* description* zur Kulturbeobachtung eingeführt. „Dichte Beschreibung" zielt als Musteranalyse auf das Verstehen von menschlichen Handlungen und alltäglichen Routinen. Ziel einer Dichten Beschreibung ist die Erklärung/Deutung/Interpretation von Strukturen als sozial-konstruktives Zusammenspiel bzw. Netz wissenskultureller Bedeutung *(web of significance).*

* thick = dicht [Natur, z. B. Wald, Nebel und Haar]; dick [Bau, z. B. Wand]; sämig [Gastronomie, z. B. Bratensoße]

Wie Groß-n-Untersuchungen so können Klein-n-Untersuchungen sowohl als eigenständige Studien als auch mit anderen Groß-n- und Klein-n-Untersuchungsmethoden kombiniert werden. Burzan (2016) bezeichnet dies als methodenplurale Forschung, wobei in der deutschsprachigen Literatur häufig der englischsprachige Begriff Mixed Methods verwendet wird (Kelle 2008; 2019). Die vertiefte und systematische Wissensproduktion von Klein-n-Untersuchungen eignet sich beispielsweise zur Identifikation von Zusammenhängen und für die Entwicklung von Indikatoren (Smelser 2003, S. 646). Darunter werden Messinstrumente (z. B. Fragen oder Bilder als Stimuli zur Hervorrufung von Emotionen) verstanden, die einen kleinen Teil des untersuchten Phänomens adressieren. Diese Indikatoren können dann für Fragebogenerhebungen operationalisiert werden. Operationalisierung bedeutet Messbarmachung, und beschreibt die Unterteilung eines empirisch beobachteten sozialen Phänomens oder einer Theorie in analytische Teile, die in Form von einfach verständlichen Fragen übertragen werden. Das beschriebene sequentielle Mixed Methods-Design (Creswell und Plano Clark 2011; Tashakkori und Teddlie 2003), also der Reihe nach erfolgendem Vorgehen von qualitativer und darauf aufbauender quantitativer Untersuchung, kann auch andersherum erfolgen. Das liegt beispielsweise vor, wenn explorative Groß-n-Fragebogenerhebungen vorbereitend zur vertieften Untersuchung (z. B. mit Interviews) einer nun identifizierten Population (z. B. neue Berufsgruppe) genutzt werden.

Die Möglichkeiten und Grenzen von Groß-n- und Klein-n-Untersuchungen sowie deren Kombination zwecks Vergleich (z. B. von Ländern) sind in Tabelle 3.2 zusammengefasst.

3.2.2 Prozessproduzierte Daten: Datentypen und Operationalisierung

In Tabelle 3.2 werden die Begriffe quantitativ und standardisiert sowie unstandardisiert und qualitativ jeweils als Synonyme für Datentypen verwendet. Im Gegensatz zu prozessproduzierten Daten erfolgt beispielsweise bei Fragebogenerhebungen die Standardisierung der Datenart (z. B. in Zahlen) durch Forschende. Auch nicht-standardisierte Beobachtungsdaten werden unter aktivem Zutun von Forschenden in Form von Feldprotokollen oder durch einen Leitfaden nur teilstandardisierte Interviews produziert. Damit ist eines der herausragenden Unterschiedsmerkmale von forschungs- und prozessproduzierten Daten benannt: In der Regel haben Forschende auf prozessproduzierte Daten keinen Einfluss. Parteien produzieren vor Wahlen Wahlprogramme, teilweise unbekannte Individuen legen Videos ab oder hinterlassen Texte von unterschiedlicher (Sprach-)Qualität und (Text-)Quantität auf einschlägigen Internetplattformen usw. An das Unterschiedsmerkmal von forschungs- und prozessproduzierten Daten gekoppelt ist, dass über Beobachtung und Befragung von Individuen und Gruppen forschungsproduzierte Daten, vorwiegend in Text- und Zahlenform, entstehen. Hingegen existiert bei prozessproduzierten Daten zuerst das Dokument bzw. die Daten in anderer Form, welche dann den Blick auf Individuen und Gruppen von Individuen ermöglichen.

Der Prozess der Datenproduktion weist dabei erhebliche Unterschiede auf. Beispielsweise wird bei administrativen Vorgängen in Rechtstexten definiert, welche Schriftdokumente (z. B. in öffentlicher Verwaltung und bei Gericht) zwecks nachvollziehbarer bürokratischer Ordnung gesammelt und mit oder ohne zeitliche Begrenzung archiviert werden müssen (Bick und Müller 1984). Im Gegensatz zu administrativ-bürokratischen Dokumenten sowie von Dokumenten, die Individuen und Organisationen (z. B. Parteien) produzieren, sind nicht bei allen im alltäglichen Leben produzierten Daten die Urheber*innen bekannt. Bei Twitter, YouTube, Facebook usw. verfügbare Daten können von sogenannten *fake user accounts* stammen. Zudem tragen nur bestimmte Alters- und Bevölkerungsgruppen zu den spezifischen Datenaufkommen in sozialen Medien im Internet bei, wodurch im Internet verfügbare Massendaten nie für die Gesamtbevölkerung repräsentativ sind (Graeff und Baur 2020; siehe Box 3.4). Wenn allerdings alle Nutzer*innen einer Internetplattform Ihre Grundgesamtheit darstellen, aus der eine Stichprobe gezogen wird, dann kann diese Stichprobe zumindest annähernde Repräsentativität erlangen, sofern Sie die Verteilung von Parametern

Tabelle 3.2 Charakteristika von Klein-n- und Groß-n-Untersuchungen

	Groß-n	Klein-n	Groß-n in Klein-n Vergleich
Ideale Stichprobenziehung	Zufallsauswahl	Zweckorientierte Fallauswahl	Kombination zweckorientierte Klein-n-Fallauswahl und Zufallsauswahl Groß-n
Daten/Methoden	Standardisierte/quantitative Daten	Unstandardisierte/qualitative Daten	Standardisierte/quantitative Daten (z. B. Fragebogen) Unstandardisierte/qualitative Daten des untersuchten sozialen Phänomens für Fälle (z. B. Länder)
Stärken/Möglichkeiten	Repräsentative Abbildung	Vertieftes Verstehen des Falls bzw. der Fälle durch dichte Beschreibung	Repräsentativität für Fall (z. B. Groß-n in Ländern) Systematisierung und Berücksichtigung Kontexte für Vergleich der Fälle (z. B. durch Theorie und/oder analytischen Rahmen)
Schwächen/Grenzen	Erkenntnisse sind auf Untersuchungspopulation begrenzt Selektivität innerhalb Untersuchungspopulation wird häufig nicht reflektiert	Erkenntnisse sind auf Fall bzw. Fälle begrenzt Hinterfragungsnotwendige Abbildung ähnlicher bzw. gleicher Fälle	Hinterfragungsnotwendige Abbildung des sozialen Phänomens in ähnlichen bzw. gleichen Fällen (z. B. global)
Anwendungsvoraussetzungen	Vertieftes Vorwissen von zu untersuchendem sozialen Phänomen Messäquivalenz ist gegeben Verfügbarkeit standardisierter Daten	Erhalt vertiefter Erkenntnisse zu sozialem Phänomen in Kontext des Falls bzw. der Fälle Explorative Herangehensweise (z. B. zwecks Hypothesenentwicklung) an zu untersuchendem sozialen Phänomen	Erhalt vertiefter Erkenntnisse zu sozialem Phänomen in Kontexten Vergleichbarkeit der Fälle muss gegeben sein Verfügbarkeit standardisierter Daten

Quelle: Schneijderberg und Götze (2021: Tabelle 2); übersetzt und ergänzt durch Autor*innen

Box 3.4: Kurzdefinition Repräsentativität
Begriff der quantitativen Forschung. Untersuchte Gruppe, Dokumente usw. ist/sind entweder eine Vollerhebung oder eine (zufällig oder quotiert ausgewählte) Stichprobe als Teilmenge der Grundgesamtheit. Die Untersuchung der Stichprobe (kleine Anzahl) ermöglicht datenbasierte Aussagen über die Grundgesamtheit (große Anzahl).

(wie Alter, Geschlecht, soziale Herkunft) der Nutzer*innen kennen, auf deren Basis Sie Unterschiede zwischen Gruppen treffen können. In diesem Falle würden Sie eine Groß-n- (Nutzer*innen) mit einer Klein-n- (eine Plattform) Analyse kombinieren.

In Tabelle 3.1 werden prozessproduzierte Daten nach den Kategorien Dokumente als (Massen-)Daten, webbasierte Massendaten (Big Data) und administrative/bürokratische Massendaten (Big Data) differenziert. Das Ziel, das der Generierung von Daten der verschiedenen Datentypen unterliegt, verdeutlicht ein zweites wesentliches Unterscheidungsmerkmal prozess- und forschungsproduzierter Daten. Während forschungsproduzierte Daten mit dem Ziel der Beantwortung einer oder mehrerer Forschungsfrage(n) erhoben bzw. produziert werden, ermöglichen bereits existierende prozessproduzierte Daten die Beantwortung von Forschungsfragen, ohne dass eine Prägung durch Theorie, Methode und Sichtweisen von Forscher*innen erfolgte. Entsprechend sind Forschungsziele ein sekundäres oder Beiprodukt prozessproduzierter Datenaufkommen.

Dokumente als (Massen-)Daten wie Bücher und Wahlprogramme von Parteien werden mit dem abstrakten Ziel der Kommunikation produziert. Selbstverständlich unterscheidet sich trotz gemeinsamem Ziel der Zweck der Produktion von Dokumenten. Beispielsweise Wahlprogramme werden unter anderem mit dem Zweck verfasst, die Politikziele von Parteien zu kommunizieren, um Stimmen von Wähler*innen bei Wahlen zu erhalten. Dahingegen werden Bücher in Romanform insbesondere zur Unterhaltung produziert, während Ratgeberliteratur der Information und Entscheidungsunterstützung der Leser*innen dienlich ist. Die operationalisierte Materialität von Büchern und Wahlprogrammen ist sowohl die Druck- als auch die elektronische, digitale Form (z. B. als PDF), wobei beide Dokumenttypen auch nur gedruckt (z. B. alte Bücher) oder in elektronischer Form vorliegen können. Die (teilweise) vorhandene Überschneidung der operationalisierten Materialität von Dokumenten als (Massen-)Daten bedeutet jedoch lediglich, dass nicht-digital vorliegende Dokumente für die Auswertung digitalisiert werden müssen. Analog müssen beispielsweise digital als Video oder Audio existierende Dokumente ebenfalls in eine digitale Textform übertragen werden.

Die Digitalisierung von Druckerzeugnissen kann bei der Untersuchung der Bürokratie administrativer Vorgänge als Arbeitsschritt anfallen, ist jedoch kein Merkmal webbasierter Massendaten. Im Gegensatz zu Dokumenten als (Massen-)Daten ist bei webbasierten Massendaten die operationalisierte Materialität keine abgeschlossene Analyseeinheit. Ziel webbasierter Massendaten ist Information, welche stetig und in Echtzeit fortgeschrieben wird. Folglich ist die Flüchtigkeit und damit Unvollständigkeit von Information konstitutiv für webbasierte Daten.

Teilweise ist die zeitlich befristete Verfügbarkeit der Informationen programmatisch (z. B. Snapchat). Der Informationsfluss wird dabei jeweils webseitenspezifisch in den allgemeinen Geschäftsbedingungen der eine Webseite anbietenden Unternehmen – oder allgemeiner Organisationen – festgelegt. Die webbasierten Informationen werden geordnet von den

a) Unternehmen (z. B. Google Algorithmus und Twitter Hashtag),
b) Inhalte beitragenden Individuen und
c) Nutzer*innen der Inhalte.

Gemeinsam sind webbasierten Massendaten und Dokumenten als (Massen-) Daten die Merkmale Information, beispielsweise als „Verminderung des Kenntnis- oder Aktualitätsgefälles zwischen Kommunikator[*in] und Rezipient[*in]" (Schulz 2009, S. 161), und Kommunikation als „(symbolische Interaktion) zwischen Menschen auf einer technischen Grundlage" (Beck 2010, S. 16). Jedoch ist die dokumentenvermittelte Informationsvermittlung eine einseitige und dokumentarisch abgeschlossene Kommunikation von Kommunikator*in zu Rezipient*in, auf die Letztere*r nicht im Dokument eingehen kann. Bei webbasierten Massendaten kann reziproke bzw. interaktive Kommunikation entstehen (z. B. Antwort auf einen Tweet). Bei prozessproduzierten Daten haben webbasierte Massendaten das Alleinstellungsmerkmal, dass deren Informationen nach bestimmten Kriterien als Daten erfasst werden (Willke 2004, S. 31).

Die Webbasiertheit ermöglicht die kontinuierliche Sammlung von Daten über Nutzer*innen, die Inhalte und Kommunikationen beitragen. Folglich ist der Tausch persönlicher Daten gegen Information ein Alleinstellungsmerkmal der prozessproduzierten webbasierten Daten. Beispielsweise passt der Google Algorithmus das Informationsangebot den Nutzungsgewohnheiten der Informationssuchenden an – ebenso wie die personalisierten Werbeangebote (Courtois et al. 2018; Powers 2017). Hingegen werden explizite Metriken als Aktivitätsdaten von Informationsbeitragenden bei Twitter angezeigt (z. B. Tweets und Followers). YouTube ist so programmiert, dass Nutzer*innen anzeigt wird, wie oft beispielsweise das Musikvideo eines Liedes aufgerufen wurde; jedoch sammelt das Video als Dokument und damit abgeschlossene Analyseeinheit keine Daten über Nutzer*innen. Zudem ist das Lied als Musikvideo nicht nur bei YouTube, sondern auch auf der Bandwebseite, der Webseite der Musikfirma, anderen Musikwebseiten usw. in identischer Form verfügbar. Variiert die Darbietungsform, so wird dies durch die Angabe kenntlich gemacht, ob es sich um ein offizielles Video, einen Konzertmitschnitt oder andere Dokumente handelt. Entsprechend multipliziert sich die Dokumentenanzahl des nun in verschiedenen Videos dargebotenen Liedes, wodurch ein systematischer Vergleich des Liedtextes, der gesanglichen Variationen usw. zwischen den Dokumenten möglich ist.

3.2.3 Datenerhebung für wissenschaftliche Analysen mit prozessproduzierten Daten

Das Beispiel Musikvideo verdeutlicht, dass ein Dokument zwar Teil einer Webseite sein kann, was jedoch umgekehrt nicht der Fall ist. Daher wird in Tabelle 3.1 für webbasierte Massendaten auch in der Kategorie bei zu erhebenden Daten Datenarrangement angegeben. Für die wissenschaftliche Analyse des Datenarrangements ist es daher notwendig, die webbasierten Massendaten in ihren unterschiedlichen Versionen als Entität zu erfassen und in eine Datenbank zu übertragen und zu bereinigen. Eine Entität ist nichts anderes als ein individuelles Objekt (z. B. Musikvideo) innerhalb einer Datenbank, das Beziehungen zu anderen Entitäten gleicher oder unterschiedlicher Art (z. B. Künstler, Musiklabels) aufweist und über identifizierbare Eigenschaften (wie z. B. Länge des Musikstücks, Audioeigenschaften, Anzahl Viewer) verfügt (Chen 1976; Elmasri et al. 1985). Für diese Entitäten ist in unserem Falle zu berücksichtigen, dass die Dokumentation webbasierter Massendaten nur für den im Forschungsdesign angegebenen Zeitabschnitt erfolgt. Aufgrund der Echtzeitcharakteristik webbasierter Massendaten kann das Datenarrangement des Webseiteninhalts, und damit der für Ihre Untersuchung interessanten Entitäten, schon weniger als eine Sekunde nach Forschungsarchivierung verändert sein (z. B. neuer Tweet). Das betrifft somit auch die Relationen und Eigenschaften Ihrer Entitäten. Auch ist eine Veränderung der für die Forschung dokumentierten Webseiteninhalte nicht immer nachzuvollziehen (z. B. durch Löschen oder Modifikation von Webseiteninhalten wegen Verstoß gegen die AGBs oder aufgrund eines *Shitstorms*).

Im Unterschied zu webbasierten Massendaten sind Veränderungen in Dokumenten als (Massen-)Daten in der Regel nachvollziehbar, auch wenn der Nachvollzug mit Aufwand verbunden sein mag. Beispielsweise kann ein Bild mit einem anderen Bild verglichen werden, auch wenn in einer Bilderserie (z. B. im Falle von Aufnahmen von Webcams an berühmten touristischen Orten) von einmal täglich über einen spezifizierten Zeitraum aufgenommenen Bildern des immer gleichen Ausschnitts ein Bild fehlt oder nicht verwertbar ist. Bei Bildern als Dokumentendaten ist klar dokumentierbar, dass ein Bild fehlt oder nicht verwendbar ist (z. B. Aufnahmen historischer Bauten oder Persönlichkeiten in Archiven) und die Bilderserie damit unvollständig ist.

Bei Dokumenten als (Massen-)Daten ist die Vollständigkeit der Daten nicht in jedem Fall gegeben. Ein Schlüssel zur Vollständigkeitsprüfung und damit der Definition des Nutzens für wissenschaftliche Zwecke ist die Existenz von Urheber*innen. Beispielsweise können Wahlprogramme Parteien, Bücher Autor*innen, Liedgut Sänger*innen oder Bands und Bilder Künstler*innen zugeordnet werden. Jedoch ist es (fast) unmöglich, alle Filme bzw. Drehbücher mit Beteiligung von Wissenschaftler*innen in Haupt- und Nebenrollen als auswertbare Dokumente aufzufinden. Selbiges Definitions- und Auswahlproblem besteht

beispielsweise bei der Untersuchung von Filmen mit Wissenschaftsbezug mittels Topic Modeling (siehe Kapitel 11), wobei die Einschränkung auf englischsprachige Filme die Vollständigkeit der Daten eher ermöglicht. Im Gegensatz zu Dokumenten als (Massen-)Daten ist *per se* eine Unvollständigkeit webbasierter Massendaten anzunehmen. Das ist der Selbstselektion aller potenziellen Nutzer*innen (Begrenzung durch fehlenden Internetzugang, keine ausreichenden Kenntnisse) und der damit einhergehenden unklaren Abgrenzung der Nutzerpopulation geschuldet.

Vollständig sollten jedoch prozessproduzierte Daten als durch Rechtsvorschriften definierte Sammlung von administrativen Massendaten sein. Die Archivierung von Verwaltungsabläufen und Beweisen von Gerichtsfällen kann dabei sämtliche Datenarten, wie Audios, Bilder, Texte, Videos und Zahlen, erfordern. Zwar ist die Textform durch Protokolle und jedwelche andere Form von Akten die überwiegende Dokumentenform, jedoch können bei einem Strafverfahren Videoaufnahmen einer Überwachungskamera ebenso wie Bilder vom Tatort Bestandteil der zu archivierenden Dokumente sein.

Die rechtsvorschriftlich geregelte Dokumentenarchivierung wird zu Datenanalysezwecken für praktische Anwendungen genutzt. Derart praktische Anwendungen können Evaluationen von Verwaltungsabläufen mit dem Ziel der Optimierung der Bürokratie sein oder die Durchführung einer Musteranalyse in bestimmten Straffällen mit dem Ziel der Aufklärung von Straftaten mitsamt Profilen von Täter*innen. Die zweckgebundenen Analysen werden von Institutionsangehörigen wie Polizist*innen und gegebenenfalls spezialisierten Verwaltungsangestellten durchgeführt. Insbesondere bei personenbezogenen und datenschutzrechtlich sehr heiklen Dokumenten wie Akten von Patient*innen, Gerichtsverfahren usw. ist die Datenanalyse nur einem bestimmten Personenkreis zugänglich. Entsprechend ist bei administrativen Massendaten nicht die Archivierung und Vollständigkeit gemäß Anzahl der Fälle problematisch, sondern der Zugang und die Nutzung für wissenschaftliche Untersuchungen. Die Art und Weise der Datenauswertung sowie Ergebnispräsentation werden durch Datenschutz- und Anonymisierungsvereinbarungen geregelt.

Prozessproduzierte Daten werden in der Regel auf Social Media-Plattformen oder kommerziellen Plattformen generiert. Die Daten können Sie mittels sogenannter Application Programming Interfaces (APIs) herunterladen. APIs stellen Schnittstellen zwischen Computerprogrammen und Datenbanken dar, mit deren Hilfe es durch gezielte Anfragen möglich ist, große Datenmengen herunterzuladen. Beispiele hierfür sind die Amazon API, Twitter API oder Reddit API. Um die jeweiligen Anfragen stellen zu können, muss man sich zunächst bei den Diensten, mit denen die API verknüpft ist, als Nutzer*in registrieren. Nach der Registrierung bekommt man einen Nutzungsschlüssel per Mail oder im jeweiligen Dienst zugeschickt bzw. zugewiesen. Mit dessen Hilfe kann man auf der Seite selbst, oder in Programmierumgebungen (R oder Python) Datenanfragen

stellen. Diese werden, sofern die abgerufenen Daten in der Datenbank vorliegen, zwischengespeichert und müssen dann in eigene Dateien oder ein geteiltes Datenformat abgespeichert werden. Eine weitere Möglichkeit zur Erhebung prozessproduzierter Daten ist das Webscraping. Beim Webscraping wird öffentlich zugänglicher Content von zuvor angegebenen Websites automatisch heruntergeladen.

In beiden Fällen müssen Sie sich mit forschungsethischen und rechtlichen Fragestellungen auseinandersetzen, wobei dies beim Webscraping stärker der Fall ist als beim Gebrauch der APIs. Im Falle von APIs sollten Sie ganz genau die Nutzungsbedingungen lesen, die Informationen darüber enthalten, wie viele Datenanfragen in welchem Zeitraum Sie stellen dürfen, welche Informationen Sie herunterladen dürfen und welche nicht. Beachten Sie dabei stets, dass Sie die Daten dahingehend prüfen müssen, ob sich aus diesen Daten Nutzer*innen eindeutig identifizieren lassen. Sie sollten auf jeden Fall die Daten hinreichend anonymisieren und so aggregieren (d. h. auf Gruppenebene hochrechnen), dass die einzelnen Nutzer*innen hinter der Datenmenge zurücktreten.

Ethische Bedenken ergeben sich aus dem Webcrawling im Forschungsprozess mit Bezug auf Betreiber*innen der gecrawlten Websites und deren Nutzer*innen (Thelwall und Stuart 2006). Dazu zählt, dass der Zugriff auf die Website bei der Datenerhebung die Geschwindigkeit des Seitenabrufs für Dritte verringert, bei zu hoher Anfragezahl des Crawlers sogar verunmöglicht. Daneben kann es Probleme mit der Privatsphäre und der Anonymität der Nutzer*innen geben, deren Daten abgerufen und von der*dem Forschenden gespeichert werden.

In diesem Zusammenhang betonen Gold und Latonero (2017, S. 300–302) zurecht, dass Webcrawler schnell auf sensible, persönliche Daten zurückgreifen können, wenn (Text-)Daten und andere Inhalte einer oder mehrerer Websites erhoben werden. Ferner sollten wir uns vor Augen führen, dass Nutzer*innen von Webdiensten in den seltensten Fällen die allgemeinen Geschäftsbedingungen (AGBs) gelesen haben und somit oftmals keine Vorstellung von dem Ausmaß haben, in dem die Daten erhoben und ausgewertet werden (Obar und Oeldorf-Hirsch 2020). Darüber hinaus können Webcrawler geistiges Eigentum entwenden, Geschäftsmodelle einer Seite untergraben oder Informationen erheben, die Geschäftsgeheimnisse verletzen (Krotov et al. 2020).

3.3 Forschungsablauf und Gliederung wissenschaftlicher Ausarbeitungen

3.3.1 Schematische Darstellung: Forschung mit empirischen Daten

Die vielfältigen Merkmale von Daten und damit verbundenen Herausforderungen für den empirischen Forschungsprozess werden abschließend in ein Schritt für Schritt-Ablaufschema des Forschungsprozesses eingeordnet (Abbildung 3.1, nächste Seite), welches danach in ein grobes Kapitelordnungsschema für eine wissenschaftliche Ausarbeitung übertragen wird (Abbildung 3.2). Um Redundanzen zu vermeiden, wird in Abbildung 3.1 eine stichwortartige, schematisch geordnete Zusammenfassung der obigen Inhalte präsentiert. Als Zusammenfassung der Erläuterungen des vorliegenden Kapitels können Sie das Ablaufschema Forschung mit empirischen Daten in Abbildung 3.1 auch zur Orientierung bei der Planung Ihrer empirischen Sozialforschung verwenden.

3.3.2 Gliederungsschema wissenschaftlicher Ausarbeitungen (z. B. Haus-, Bachelor-, Master- und Doktorarbeit)

Die grobe Ordnung des inhaltsanalytischen Forschungsprozesses kann relativ klar strukturiert in eine wissenschaftliche Ausarbeitung übertragen werden (Schritt 12 „Publikation" in Abbildung 3.1). Das grobe Gliederungsschema bildet den Kern wissenschaftlicher Ausarbeitungen – seien es Studienarbeit, Zeitschriftenbeitrag oder Buch – und es ist unabhängig von deren Länge, wobei eine Bachelorarbeit und sicher eine Doktorarbeit mehr Unterkapitel als eine Hausarbeit aufweisen. Das grobe Gliederungsschema wissenschaftlicher Ausarbeitungen passt auch zu den Lesegewohnheiten von Dozent*innen und Prüfer*innen, welche als Vielleser*innen quasi analog zum Spannungsbogen eines Kriminalromans einen solchen Kapitelaufbau erwarten. Selbstverständlich ist etwas Kreativität immer möglich, auch bei der Kapitelbenennung, doch sollten Sie es sich und anderen leichtmachen, insbesondere, wenn diese anderen Ihre wissenschaftliche Ausarbeitung bewerten müssen.

In Abbildung 3.2 wird im Schema von grundlegend in wissenschaftlichen Ausarbeitungen erwarteten Kapiteln zwischen deduktiver, d. h. theoriegeleiteter bzw. theorieüberprüfender, empirischer Sozialforschung und induktiver, d. h. explorativer, empiriegetriebener, Sozialforschung unterschieden. Dieser Unterschied zeigt sich bereits bei der Übertragung von Forschungsschritt 2 in die Schriftform. Bei deduktiver Sozialforschung gilt es nicht nur, den bestehenden Stand der Forschung aufzuarbeiten, um deutlich erklären zu können, worauf wir aufbauen und wo die Forschungslücke identifiziert wurde. Deduktive Sozialforschung zeichnet sich eben durch die Leitung einer ausformulierten theoretischen Herangehens-

Abbildung 3.1 Ablaufschema der Forschung mit empirischen Daten

Schritt 1	Thema einer Forschungsidee skizzieren und zumindest stichpunktartig das Erkenntnisinteresse in der Forschungsdokumentation schriftlich festhalten (Stichwort: Nachvollziehbarkeit der Forschung).
Schritt 2	Recherche bisheriger Forschung zum Thema allgemein (= Kontextualisierung) und spezifisch zwecks Identifikation der Forschungslücke. Stand der Forschung bzw. fehlender Forschung schriftlich zusammenfassen. Damit erfolgt die empirie- und/oder theoriebegründete Präzisierung der Forschungsidee (siehe Schritt 1) und Begründung von Forschungsfrage(n) sowie gegebenenfalls Hypothesenbildung.
Schritt 3	Hard- und softwaretechnische Voraussetzungen für Datensammlung und Auswertung abhängig von operationalisierter Materialität der Datenart schaffen.
Schritt 4	Datenzugang klären und gegebenenfalls Stichprobenziehung.
Schritt 5	Abhängig von Forschungsfrage(n) und identifiziertem (Nicht-)Stand der Forschung entweder a) induktive, z. B. explorative, empiriegetriebene oder vom Forschungsstil der Grounded Theory geleitete Herangehensweise, oder b) deduktive, z. B. überprüfende und theoriebasierte Herangehensweise, oder c) Mixed Methods-Forschungsdesign zur Untersuchung des sozialen Phänomens wählen und begründen.
Schritt 6	Datensammlung (auch als *data mining* bezeichnet) gemäß begründeten Regeln und/oder Suchwörtern (auch als *search string*s bezeichnet).
Schritt 7	Datenbereinigung und Erstellung einer Datenbank. Gegebenenfalls besondere Anforderungen (z. B. Anonymisierung) für die Erstellung von *student-* und/oder *scientific-use files* zur Datennachnutzung durch Dritte berücksichtigen.
Schritt 8	Datenauswertung: • Methodengeleitet (z. B. deduktiv oder induktiv; siehe Schritt 4). • Gegebenenfalls weiterführende iterative Datenexploration als Kunst der Datenanalyse bzw. Versuchs- und Irrtumsvorgehen (auch als *trial-and-error* bezeichnet – *kein* p-hacking!).
Schritt 9	Deskriptive Analyse (Antworten auf Was-Frage; Ziel: empirische Ergebnisse selbst verstehen).
Schritt 10	„Diskussion" der Ergebnisse als theoriebasierte Analyse und/oder Reflexion der empirischen Ergebnisse zwecks Theorieentwicklung (Antworten auf Warum-Fragen; Ziel: Erklären, Deuten, Interpretieren und Schließen).
Schritt 11	Zusammenfassung der wichtigsten Ergebnisse und gegebenenfalls Hinweise auf weitere Forschungsschritte basierend auf *lessons learned*.
Schritt 12	Publikationen: • Wissenschaft (z. B. Studienarbeit und Zeitschriftenbeitrag). • Allgemeine Öffentlichkeit (z. B. Bericht, Zeitungsbeitrag, Blog und Podcast).
Schritt 13	Archivierung der Daten gemäß Vorgaben zu „guter wissenschaftlicher Praxis" der Deutschen Forschungsgemeinschaft (DFG).

weise aus (siehe auch Kapitel 5 und 7). Hingegen ist induktive empirische Sozialforschung maximal von theoretischen Annahmen von Forscher*innen getrieben, welche ein beobachtetes soziales Phänomen zu erfassen suchen. Solche Annahmen gilt es dann auch mit Bezug zum Stand der Forschung bzw. zum Fehlen vorheriger Forschung und damit geeigneter theoretischer Herangehensweisen in einer wissenschaftlichen Ausarbeitung zu formulieren. Weder qualitative noch quantitative induktive Sozialforschung ist theoriefreie Forschung. Im Gegensatz zur deduktiven Sozialforschung (z. B. Inhaltsanalyse) ist bei induktiver Sozialforschung (siehe Kapitel 4, 6, 9, 10 und 11) die Theorie aber nicht von Beginn an untersuchungsleitend, sondern das empirische Material und daraus generierte analytische Kategorien sind der Theorie vorangestellt. Entsprechend kann bzw. sollte zur Unterstützung der Analyse vor allem von latenten Inhalte auf sozialwissenschaftliche Theorie(n) zurückgegriffen werden.

Das Kapitel „Methodische Herangehensweise" vereinigt die Forschungsschritte 3 bis 9, ist jedoch nicht natürlicherweise das längste Kapitel in Ihrer wissenschaftlichen Ausarbeitung. Grundsätzlich sollte dieses Kapitel stets die gewählte (inhaltsanalytische) Methode darstellen. Darstellen bedeutet, dass Sie ohne umfassende Nacherzählung aus dem Methodenbuch die wesentlichen systematischen Kriterien einer Methode mit Fokus auf Ihre Untersuchung methodisch darstellen. Bei qualitativer und quantitativer Sozialforschung müssen Sie die Fallauswahl (d. h. das Sampling bzw. die Stichprobenziehung), die Art der Datengenese (siehe Tabelle 3.1) als Entstehungskontext der empirischen Daten ebenso erklären, wie Ihr methodisch-systematisches Vorgehen bei der Datenanalyse. An dieser Stelle sei nochmals daran erinnert, dass das aufwendige Datenbereinigen prozessproduzierter Daten vor allem bei Klein-n qualitativen Inhaltsanalysen weniger umfangreich ist (siehe Kapitel 4).

Bei induktiven Inhaltsanalysen wie auch induktiver empirischer Sozialforschung allgemein nimmt die theoretische Reflexion und gegebenenfalls angestrebte Theorieentwicklung einen besonderen Stellenwert bei der Analyse der Ergebnisse ein. Auch bei der Theorieentwicklung sollten Sie zur Diskussion der Ergebnisse existierende Sozialtheorien heranziehen, um nachvollziehbar zu argumentieren, warum diese teilweise oder gar nicht für die abstrahierende, d. h. verallgemeinerbare Erklärung verwendet werden können. Folglich heißt Theorieentwicklung in der Regel die Weiterentwicklung von bestehenden Theorien basierend auf der Diskussion empirischer Ergebnisse. Bei der Diskussion der Ergebnisse und noch vielmehr in den Schlussfolgerungen sollten Sie die Ergebnisse anderer Forschung explizit darstellen, um anzuzeigen, was Sie darüberhinausgehend, im Unterschied dazu oder gar neues an Erkenntnissen mit Ihrer empirischen Sozialforschung erzielt haben.

In der empirischen Sozialforschung werden in der Regel mehr Informationen durch Erhebungen gesammelt, als in einer wissenschaftlichen Ausarbeitung verarbeitet und dargestellt werden können. Ebenso wie die Auswahl der geeigneten

Abbildung 3.2 Übertrag des Ablaufschemas der Forschung mit empirischen Daten in Kapitelstrukturschema für Studienarbeit, Zeitschriftenartikel oder Buch

Deduktive Forschung	Schritt(e)	Induktive Forschung
Einleitung	1	Einleitung
Stand der Forschung	2	Stand der Forschung
Theoretische Herangehensweise	2	
Methodische Herangehensweise	3, 4, 5, 6, 7 & 8	Methodische Herangehensweise
Deskriptive Darstellung und Diskussion der Ergebnisse	9 & 10	Deskriptive Darstellung und Diskussion der Ergebnisse
	10	Theoretische Reflexion bzw. Theorieentwicklung
Schlussfolgerungen	11	Schlussfolgerungen
Datenanhang	optional	Datenanhang

Daten für eine wissenschaftliche Ausarbeitung insgesamt müssen Sie für die Präsentation von Daten in einer wissenschaftlichen Ausarbeitung eine Auswahl treffen. Aufgrund von Zeichen-, Wort- oder Seitenbegrenzungen (siehe z. B. die Studien- und Prüfungsordnung Ihres Studiengangs) können Sie nur eine bestimmte Anzahl an Abbildungen, Tabellen, Ankerbeispielen usw. im Text präsentieren. Nicht unmittelbar für das Verständnis von Leser*innen Ihrer wissenschaftlichen Ausarbeitung benötigte oder als Belege notwendige Daten können Sie daher im Datenanhang Ihrer wissenschaftlichen Ausarbeitung mit einreichen. Empirische Datenanhänge können sehr umfangreich sein. Daher sollten Sie überlegen, ob es notwendig (z. B. von Prüfer*innen gefordert) ist, den Datenanhang auszudrucken oder diesen elektronisch zur Verfügung zu stellen (z. B. über einen Dateiordner in einer Cloud).

3.4 Nachnutzung und Bereitstellung von Daten

An dieser Stelle wollen wir Ihnen kurz erläutern, wie Sie offene Daten nutzen oder wie Sie für eine Sekundärauswertung an Daten kommen können. Zudem möchten wir Ihnen ein paar Tipps an die Hand geben, wie Sie Daten, die Sie vielleicht selbst erhoben haben, für andere Forschende öffnen und dabei nachnutzbar machen können.

Die Nachnutzung von personenbezogenen Daten, also solchen, die zum Beispiel durch eine Fragebogenerhebung oder durch Interviews erhoben wurden, können Sie nicht so einfach aus dem Netz herunterladen. Denn personenbezogene Daten unterliegen in Europa der sogenannten Datenschutzgrundverordnung (DSGVO), die solche Daten besonders schützt. Deshalb finden Sie solche Forschungsdaten nur bei Forschungsdatenzentren (FDZ). Diese Zentren sorgen dafür, dass die Daten in anonymisierter Form archiviert und einer Nachnutzung unter bestimmten Voraussetzungen zugänglich gemacht werden. Als Studierende oder Promovierende ist es beispielsweise meist notwendig einen Antrag für die Nutzung zu stellen.

Für sozialwissenschaftliche Daten ist das Forschungsdatenzentrum GESIS das wohl größte und bekannteste in Deutschland. GESIS bietet derzeit 6 500 Datensätze zur Sekundärnutzung an.[1] Es handelt sich dabei um anonymisierte Daten, die von GESIS geprüft wurden. Um diese Datensätze zu nutzen, ist eine Anmeldung bei GESIS notwendig. Eine Bereitstellung der Daten erfolgt nur im Rahmen wissenschaftlicher Auswertungen und nur für ein befristetes Vorhaben. Das heißt nach Ablauf Ihres Forschungsprojektes müssen Sie die Daten bei sich wieder löschen! Die meisten Datensätze sind quantitative Daten, die aus großen Umfragen (meist Fragebogenerhebungen) gewonnen wurden. Immer mehr werden aber auch qualitative Daten archiviert und einer Nachnutzung zugänglich gemacht. Da hier die Anonymisierung aber wesentlich schwieriger und aufwendiger ist, schrecken noch viele Forscher*innen davor zurück, auch wenn es prinzipiell möglich und auch wünschenswert ist (Steinhardt et al. 2020).

Das Feld der Forschungsdatenzentren entwickelt sich gerade sehr dynamisch weiter, sowohl was den Umfang an Daten angeht als auch die Spezialisierungen. Um einen Überblick zu bekommen, welche Forschungsdatenzentren akkreditiert sind (also einer Qualitätsprüfung unterzogen wurden), können Sie eine Liste[2] beim Rat für Sozial- und Wirtschaftsdaten (RatSWD) einsehen, der auch die Akkreditierungen vornimmt. Zudem gibt es eine Metasuche, durch die Sie alle Datenbanken der beim RatSWD beteiligten Forschungsdatenzentren durchsuchen können.[3]

Neben der Nachnutzung von Forschungsdaten können Sie sich ebenfalls überlegen, ob Sie, wenn Sie selbst Daten erhoben haben, diese zur Nachnutzung durch andere Forscher*innen bereitstellen wollen. Wir möchten Ihnen das sehr ans Herz legen, wohl wissend, dass das gerade bei Studienabschlussarbeiten einen Mehraufwand bedeutet, der kaum geleistet werden kann. Bei Promotionen

---

1 Die Suchfunktion bei GESIS finden Sie hier: *www.gesis.org/angebot/daten-finden-und-abrufen.*

2 Liste der akkreditierten Forschungsdatenzentren: *www.konsortswd.de/datenzentren/alle-datenzentren.*

3 Suchseite des RatSWD: *www.konsortswd.de/datenzentren/datensuche-in-den-fdz.*

sollte die Archivierung und Fragen der Nachnutzung aber auf jeden Fall bedacht werden.

Für offene Daten, also solche, die frei im Internet abrufbar sind, gibt es beispielsweise die Möglichkeit, die Zusammenstellung der Daten zu dokumentieren und zu veröffentlichen. Die gesammelten Daten selbst können meist nicht für eine Nachnutzung bereitgestellt werden. Was genau ist damit gemeint? Sie können beispielsweise für Ihre Forschung Daten wie Tweets oder auch die Metadaten von Publikationen von Social Media-Plattformen oder Datenbanken erhalten und diese auswerten. Welche Daten Sie zu welchem Zeitpunkt, nach welchen Suchkriterien etc. für sich heruntergeladen haben, das können Sie als Dokumentation veröffentlichen. Die Daten selbst, also die Tweets oder die Metadaten der Publikationen dürfen Sie aber nicht als Datensatz veröffentlichen, da es sich dabei nicht um Ihre eigenen Daten handelt, Sie also nicht „Eigentümer*in" der Daten sind. Eigentümer*innen sind in diesen Fällen die Social Media-Plattformen oder die Eigentümer*innen der Datenbanken.

Anders verhält es sich bei Datensätzen, die Sie selbst durch eine Fragebogenbefragung oder Interviews erhoben haben. Diese Daten können Sie in anonymisierter Form einer Nachnutzung zur Verfügung stellen, allerdings nur, wenn Sie bei der Erhebung die informierte Einwilligung der Nachnutzung eingeholt haben. Möglich ist dies auch im Nachhinein, allerdings viel komplizierter, weshalb die Nachnutzung unbedingt bei der Erhebung mitgedacht werden sollte. Wenn Sie beispielsweise eine Fragebogenerhebung machen und nicht bereits beim Ausfüllen des Fragebogens die Erlaubnis zur Nachnutzung der Daten einholen, dann ist es im Nachhinein kaum möglich, die Erlaubnis aller Befragten einzuholen. Und da Sie ja meist nicht wissen, wer welcher Fall ist, können Sie diesen einen Fall nicht eliminieren und deshalb den gesamten Datensatz nicht für eine Nachnutzug zur Verfügung stellen.

Wenn Sie die informierte Einwilligung der Befragten haben, dann stellen Sie die Daten aber bitte nicht einfach ins Netz! Nutzen Sie dafür bitte ein Forschungsdatenzentrum. Denn dieses hilft Ihnen dabei, die notwendige Anonymisierung durchzuführen, sodass Sie nicht mit der Datenschutzgrundverordnung in Konflikt geraten. Zudem bedarf es für die sinnvolle Nachnutzung Kontextinformationen, die ebenfalls zur Verfügung gestellt werden müssen. Deshalb haben Forschungsdaten, die in einem Forschungsdatenzentrum abgelegt werden, immer einen Methodenbericht. In diesem wird beschrieben, wie die Daten erhoben wurden, durch wen, wann, in welcher Form sie vorliegen, wie sie in der Primäruntersuchung ausgewertet wurden, welche Ergebnisse publiziert sind und vor allem, welche Maßnahmen durchgeführt wurden, um eine Anonymisierung zu gewährleisten.

Wenn Sie Daten haben, die keinem Datenschutz oder Urheberrecht unterliegen (z. B. Literaturkorpora alter Literatur), dann können Sie offene Repositorien verwenden, um Ihre Daten online verfügbar zu machen. Repositorien sind

große Datenbanken, in denen bereits aufbereitet und kommentierte Datensätze angelegt sind. Eine der größten ist die Harvard Database, in der mehrere zehntausende Datensätze zu Replikationszwecken gespeichert werden und durch die Zuweisung einer DOI zitierfähig sind. Dabei können Nutzer*innen in den Suchfeldern gezielt nach Textkorpora suchen. Bei einem Repositorium (auf Englisch: *Content Repository*) handelt es sich um ein Datenmanagementsystem.

Anhang 3.1: Kurzbeschreibung des Lehr-Forschungsprojekts „Zwei Wochen Studium im Wintersemester 2020/21“

Im Wintersemester 2020/21 findet das zweite sogenannte Corona-Semester statt. Mit der Autoethnographie soll erhoben werden, wie Studierende mit den spezifischen Lehr- und Lernbedingungen, studienbezogenen Kontakten usw. umgehen. Zentral für die Autoethnographie sind die Erzählungen des Studienalltags – insbesondere (neu) etablierte und wiederkehrende Routinen – und persönliche Reflexionen zur Gestaltung des Studienbeginns im Wintersemester 2020/21. Nicht von Interesse sind alltägliche Handlungen, welche nicht zum Studium gehören (Aufstehen, mit Familie und Freunden sprechen ohne Studienbezug, Mahlzeiten usw.). Ergänzend zum Erkenntniskern Studienalltag ist für die vergleichende Untersuchung der Autoethnographien der Studierenden von Interesse, dass Studierende teilweise im 1. Semester sind, d. h. die Studienanfangsphase beschreiben, und Studierende teilweise im 3. oder höheren Semester eingeschrieben sind, d. h. ihren Studienalltag im Corona-Semester im Vergleich zu Semestern mit Präsenzlehre und Universitätsleben schildern und reflektieren können.

Informationen über den Umgang mit der Autoethnographie

Der Datenschutz verlangt Ihre ausdrückliche und informierte Einwilligung, was wir mit Ihrer Autoethnographie machen dürfen. Die verantwortliche Leitung des Lehr-Forschungsprojektes „Zwei Wochen Studium im Wintersemester 2020/21“ liegt bei [Lehrendenname], Institut für [Name], [Universitätsname]. Die Durchführung des Lehr-Forschungsprojektes geschieht auf der Grundlage der Datenschutzgrundverordnung (DSGVO) der Europäischen Union und hält den Ethik-Kodex der Deutschen Gesellschaft für Soziologie *(www.soziologie.de/de/die-dgs/ethik/ethik-kodex.html)* ein. Die an dem Lehr-Forschungsprojekt beteiligten Personen unterliegen vertraglich der Schweigepflicht und sind auf das Datengeheimnis verpflichtet.

Das Lehr-Forschungsprojekt dient allein wissenschaftlichen Zwecken. Wir sichern Ihnen folgendes Verfahren zu, damit Ihre Angaben nicht mit Ihrer Person in Verbindung gebracht werden können.

- Wir gehen sorgfältig mit dem Erzählten um, sowohl mit Blick auf Inhalte Ihrer Autoethnographie als auch der sicheren Aufbewahrung Ihrer Autoethnographie-Dateien.
- Falls nicht vorher durch Sie geschehen, wird Ihre Autoethnographie anonymisiert, d. h. alle Personen-, Orts-, Straßennamen sowie alle persönlichen Angaben wie z. B. Alter, Beruf werden gestrichen oder pseudonymisiert.
- Ihr Name und Ihre E-Mail-Adresse werden am Ende des Projektes in unseren Unterlagen gelöscht, sodass lediglich die anonymisierte Autoethnographie existiert. Die von Ihnen unterschriebene Erklärung zur Einwilligung in die

Auswertung wird in einem gesonderten Ordner an einer gesicherten und nur der Projektleitung zugänglichen Stelle (bzw. Datentreuhänder*innen) aufbewahrt. Sie dient lediglich dazu, bei einer Überprüfung durch die*den Datenschutzbeauftragte*n nachweisen zu können, dass Sie mit der Auswertung einverstanden sind. Sie kann mit Ihrer Autoethnographie nicht mehr in Verbindung gebracht werden.

- Wenn Ihre Zustimmung erfolgt ist, wird die Autoethnographie für das Lehr-Forschungsprojekt verwendet und innerhalb des Seminars interpretiert. Dabei wird Ihre Anonymität gewahrt.
- Wenn Ihre Zustimmung erfolgt ist, wird die anonymisierte Autoethnographie für Forschungszwecke verwendet und in Sequenzen, die nicht auf Sie als Person schließen lassen, auch zum Zwecke des kollaborativen analogen und online Interpretierens genutzt.
- Wenn Ihre Zustimmung erfolgt ist, werden gegebenenfalls einzelne Sequenzen nach den Bestimmungen der Creative-Common-Lizenz CC-BY-SA für Lehrzwecke genutzt. Das bedeutet, dass die Sequenzen auch von Dritten bearbeitet und unter den gleichen oder vergleichbaren Lizenzbestimmungen veröffentlicht werden dürfen. Das heißt die Daten dürfen zu Forschungszwecken verwendet werden, aber nicht zu kommerziellen Zwecken (siehe *https://irights.info/wp-content/uploads/userfiles/CC-NC_Leitfaden_web.pdf;* letzter Aufruf: 01. 11. 2020).

Hinweis auf die Rechte der Befragten

Die Teilnahme an dem wissenschaftlichen Forschungsvorhaben ist freiwillig; sollten Sie nicht teilnehmen, entstehen Ihnen keine Nachteile. Sie haben jederzeit die Möglichkeit, die folgenden Rechte geltend zu machen.

- *Recht auf Auskunft* über die von Ihnen verarbeiteten personenbezogenen Daten (Art. 15 DSGVO),
- *Recht auf Berichtigung* Sie betreffender unrichtiger personenbezogener Daten (Art. 16 DSGVO),
- *Recht auf Löschung* Sie betreffender personenbezogener Daten (Art. 17 DSGVO),
- *Recht auf Einschränkung der Verarbeitung* Sie betreffender personenbezogener Daten (Art. 18 DSGVO),
- *Recht auf Widerspruch* gegen die Verarbeitung Sie betreffender personenbezogener Daten (Art. 21 DSGVO).
- Sie haben zudem das Recht, sich bei einer *Datenschutz-Aufsichtsbehörde* über die Verarbeitung Ihrer personenbezogenen Daten durch uns zu *beschweren* (Art. 77 DSGVO).
- Sofern Sie in die Verarbeitung Ihrer Daten eingewilligt haben, besteht die Möglichkeit, diese jederzeit für die Zukunft zu *widerrufen* (Art. 7 Abs. 3

DSGVO). In diesem Fall müssen alle personenbezogenen Daten entweder gelöscht oder anonymisiert werden.

Ihre Rechte sind grundsätzlich schriftlich bei dem zur Datenverarbeitung Verantwortlichen geltend zu machen. Kontakt: [E-Mail-Adresse]

Wenn Sie an dem wissenschaftlichen Forschungsvorhaben teilnehmen wollen, lesen und unterschreiben Sie bitte die beiliegende Einwilligungserklärung.

Das Original bzw. eine Kopie der Einwilligungserklärung bleibt bei Ihnen. Das Informationsschreiben zum wissenschaftlichen Forschungsvorhaben verbleibt ebenfalls bei Ihnen.

Anhang 3.2: Einwilligungserklärung zur Verarbeitung und Weitergabe der Daten „Autoethnographie“

Lehr-Forschungsprojekt „Zwei Wochen Studium im Wintersemester 2020/21“ im Rahmen der [Veranstaltungsname], Dozent*in: [Lehrendenname], Institut für [Name], [Universitätsname].

Ich bin über das Vorgehen bei der Datenspeicherung und Auswertung der von mir gegebenen Autoethnographie persönlich und mittels eines schriftlichen Handzettels „Kurzbeschreibung Lehr-Forschungsprojekt Zwei Wochen Studium im Wintersemester 2020/21“ informiert worden, der mir auch ausgehändigt wurde. Mir ist bewusst, dass die Teilnahme an dieser Autoethnographie freiwillig ist und ich zu jeder Zeit die Möglichkeit habe, mein Einverständnis zurückziehen, ohne dass mir dadurch irgendwelche Nachteile entstehen.

Ich bin mit damit einverstanden,

- dass Texte und Sequenzen der Autoethnographie in anonymisierter Form im Rahmen des oben angegebenen Lehr-Forschungsprojektes in der Lehrveranstaltung vor Ort interpretiert werden: ☐ ja ☐ nein
- dass Texte und Sequenzen der Autoethnographie in anonymisierter Form im Rahmen des oben angegebenen Lehr-Forschungsprojektes kollaborativ analog wie online interpretiert werden: ☐ ja ☐ nein
- dass die anonymisierte Autoethnographie durch [Lehrendenname] und unter Aufsicht durch [Lehrendenname] zu Forschungszwecken verwendet werden darf: ☐ ja ☐ nein
- dass Texte und Sequenzen der Autoethnographie in anonymisierter Form im Rahmen einer Lerneinheit für Studierende verwendet werden (Open Education Ressource) und damit unter einer CC-BY-SA Lizenz stehen: ☐ ja ☐ nein

Unter den oben angegebenen Bedingungen erkläre ich mich bereit, das Interview zu geben.

Vor- und Nachname ausschreiben (z. B. Druckbuchstaben): [Bitte eintragen]

Ort und Datum: [Bitte eintragen]

Unterschrift: [Bitte eintragen]

Eintrag und Unterschrift Option 1 (rechtlich sicherer!): Dokument ausdrucken, mit *Kugelschreiber* ankreuzen sowie unterschreiben und dann, z. B. mit dem Handy, ein Foto machen → Datei speichern unter „Einwilligungserklärung Vor- und Nachname“.

Option 2 (nur wenn kein Drucker verfügbar!): Entsprechendes Kästchen bei ja/nein durch „X“ ersetzen; Unterschrift fotografieren, speichern und über „Einfügen“ → „Bilder“ in Word importieren; die Größe des Unterschriftbildes kann durch ziehen an den Ecken (nicht Seiten) verändert werden; ebenso besteht die Möglichkeit unter „Format“ → „Zuschneiden“ den weißen Rand zu verkleinern. Bei Option 2 bitte Dokument als PDF speichern: „Datei“ → „speichern unter“ → Speicherort auswählen → „Dateityp auswählen“ (Drop-down-Menü) → PDF.

Bitte alle in eckigen Klammern und grau hinterlegten Einträge vor dem Speichern löschen! Vielen Dank.

4. Induktiv-qualitative Inhaltsanalyse

Bei der Inhaltsanalyse handelt es sich um die systematische und regelgeleitete Erhebung und Analyse von Texten (Kommunikationsinhalt), durch die Interpretation von manifesten und latenten Inhalten und deren Bedeutungen. Diese werden durch Zerlegen der Texte in Kategorien mittels Kodieren ermittelt. Die induktiv-qualitative Inhaltsanalyse kommt zum Einsatz, wenn Sie Inhalte von Dokumenten, Gruppendiskussionen, Interviews usw. vertieft verstehen wollen und/oder zu einem Phänomen noch wenig bekannt ist, und nicht auf empirische Studien oder auf Theorien zurückgegriffen werden kann. Sie können sich das Verfahren kurz zusammengefasst so vorstellen, dass Sie Daten erheben (z. B. Interviews) oder Sekundärdaten verwenden (im Kapitel verwendetes Beispiel: Autoethnographie), an die Sie mit einem offenen Blick ohne Hypothesen oder Vorannahmen herangehen und die Kategorien aufgrund des Materials entwickeln. Das Kategoriensystem ist für die induktiv-qualitative Inhaltsanalyse nicht vorgegeben, sondern entsteht mit Bezug zum empirischen Material für die Erschließung des empirischen Materials. Die Methodeneinführung in diesem Kapitel enthält empirisches Originalmaterial (mehr als sechs Seiten), sechs Tabellen (z. B. Interpretationsanleitung am Beispiel) und eine Abbildung.

4.1 Einleitung

Die induktiv-qualitative Inhaltsanalyse bezeichnet in der empirischen Sozialforschung eine besonders stark auf das Text- und gegebenenfalls dazugehörige Datenmaterial (z. B. Icons und Bilder von Social Media-Plattformen) ausgerichtete methodische Auswertung. In der Methodenliteratur bezeichnet der Begriff „induktiv" die Entwicklung von Kategorien aus dem empirischen Material heraus. Reichertz (2016, S. 138) definiert Induktion als „sichere Ableitung" von Kategorien, die dann als strukturierendes Element die induktiv-qualitative Inhaltsanalyse leiten. Im Gegensatz zur Induktion ist die deduktiv-qualitative Inhaltsanalyse eine Methode, die Datenmaterial (z. B. Gruppendiskussionen, Interviews, Parteiprogramme, Tweets und Zeitungsartikel) anhand eines aus Theorie(n) und vorherigen empirischen Untersuchungen erstellten (= Deduktion) Kategoriensystems analysiert (siehe Kapitel 5).

Die systematische und regelgeleitete Erhebung und Analyse von Texten (Kommunikationsinhalt) mit dem Ziel der Erschließung und Interpretation von manifesten und latenten Inhalten (siehe Erinnerung in Box 4.1) und deren Bedeutungen erfolgt in mehreren Schritten: Bei der induktiv-qualitativen Inhalts-

Box 4.1: Der Verstehensdreischritt zur Erinnerung

Analyseschritt 1: Kontext-Verstehen des untersuchten sozialen Phänomens [a) Subjekt(e), b) Objekt(e), c) Raum/Räume und d) Zeit/Perioden].

Analyseschritt 2: Inhalte-Verstehen durch Erklären, Deuten, Interpretieren und Schließen.

a) Manifester Inhalt [Was-Frage der Kommunikation (1. Sinnebene)?]
b) Latenter Inhalt [Warum-Frage der Kommunikation (2. Sinnebene)?]

Analyseschritt 3: Publikum-Verstehen [adressat*innenspezifisch Ergebnisse zusammenfassen, für Präsentation arrangieren und/oder für Publikation verschriftlichen].

analyse werden die Kategorien aus dem Material heraus entwickelt (im Soziolekt: Emergieren). Die induktiv-qualitative Inhaltsanalyse kommt zum Einsatz, wenn Sie Inhalte von Dokumenten (z. B. Akten, Briefe und Wahlprogramme von Parteien), Gruppendiskussionen, Interviews usw. vertieft verstehen wollen und/oder zu einem Phänomen noch wenig bekannt ist und nicht auf empirische Studien oder auf Theorien zurückgegriffen werden kann. Sie können sich das Verfahren kurz zusammengefasst so vorstellen, dass Sie Daten erheben (z. B. offene Interviews) oder Sekundärdaten verwenden, an die Sie mit einem offenen Blick ohne Hypothesen oder Vorannahmen herangehen und die Kategorien aufgrund des Materials entwickeln (siehe Kapitel 2.2.1).

Das Kategoriensystem ist für die induktiv-qualitative Inhaltsanalyse nicht vorgegeben, sondern entsteht aus dem Material heraus in einem iterativen Prozess. Die Wiederholung (im Soziolekt: Iteration) derselben Handlung bedeutet die systematische Erfassung des Materials und darin enthaltener Muster, welche als Kategorien zum thematischen Ordnen und Strukturieren der Analyse dienen. In der induktiv-qualitativen Sozialforschung werden die Kategorien stets mit Bezug zum empirischen Material für die Erschließung des empirischen Materials in Kodes übersetzt, welche das Kodieren der Empirie leiten. Bei einer Analyse, in der Fälle (Dokumente, Interviews etc.) verglichen werden, suchen Forschende nach Gemeinsamkeiten und Unterschieden in den Daten, die in Kategorien überführt werden. Dabei wird von den Daten zu einem theoretischen Verständnis übergegangen, also vom Spezifischen zum Allgemeinen.

Bevor Sie mit der Planung beginnen, sollten Sie sich zwei Charakteristika der induktiv-qualitativen Inhaltsanalyse vergegenwärtigen.

1. Umfang Datenmaterial und Zeitaufwand für Analyse: Die Forschung am empirischen Material ist eher anspruchsvoll und zeitintensiv. Sie ist anspruchsvoll und zeitintensiv, da Sie die auszuwertenden Daten (z. B. den Text) gut kennenlernen, ordnen und verstehen müssen, um manifeste und latente Inhalte identifizieren, erklären und interpretieren zu können. Dabei muss das empirische Material nicht sehr umfangreich sein. Die als Beispiel in diesem Kapitel als Datenmaterial verwendete Autoethnographie mit zehn Tagebucheinträgen aus zwei Wochen zum Thema „Wie gestalten Sie Ihren Studienalltag zu Beginn des Online-Semesters 2020/21?“ war im Original etwa fünfeinhalb Seiten lang (bei Schriftgröße Arial 11). Ihr qualitatives Erkenntnisinteresse vorerst hintenanstellend, sollten

Sie sich forschungspragmatisch von folgenden erfahrungsbasierten Daumenregeln für den Abgleich der Parameter Zeit für die Forschung und Seitenumfang der dazugehörigen Qualifikationsarbeit leiten lassen.

a) Studien- bzw. Hausarbeit im Bachelor oder Master (ca. 10–20 Seiten): maximal ein Text zu Übungszwecken.
b) Bachelorarbeit (ca. 50 Seiten): maximal drei Texte.
c) Masterarbeit (ca. 80 Seiten): maximal fünf Texte.
d) Doktorarbeit (ca. 250 Seiten): etwa zehn plus nicht allzu viel mehr Texte.

2. Methodengeleitete Entscheidungsfreudigkeit: Die induktiv-qualitative Inhaltsanalyse bietet zwar methodische Anleitungen für die Beantwortung einer Forschungsfrage und Ableitung von Erkenntnissen aus empirischen Daten, die z. B. durch Kategorien und dazugehörigen Kodes erfasst werden. Die sichere Ableitung betont jedoch nur den konkreten Bezug von Induktion zur Empirie, jedoch nicht, wie Sie die Erkenntnisse benennen oder als Kategorie identifizieren. Entsprechend müssen Sie viele Entscheidungen treffen, und die getroffenen Entscheidungen gegenüber Mitstudierenden, Lehrenden, Prüfer*innen usw. erklären und rechtfertigen.[1]

Diese gleich zu Beginn betonten beiden Charakteristika qualitativ-induktiver Inhaltsanalyse sollen Sie nicht von der methodengeleiteten Anwendung dieser Auswertungstechnik abhalten, in welche dieses Kapitel ja systematisch und an empirischen Beispielen einführt. Dennoch sollten Sie sich der methodischen Konsequenzen Ihrer Entscheidung für die induktiv-qualitative Inhaltsanalyse und ebenfalls möglicher alternativer Auswertungstechniken bewusst sein. Alternativen zur induktiv-qualitativen Inhaltsanalyse sind die deduktiv-qualitative Inhaltsanalyse (Kapitel 5), die teilautomatisierte quantitative Inhaltsanalyse mit zwei sich ergänzenden Auswertungstechniken (Kapitel 6) und die vollautomatisierte quantitative Korrespondenzanalyse (Kapitel 9). Vermutlich haben Sie in Kapitel 3 in diesem Buch bereits mehr über die Möglichkeiten und Grenzen von Daten und Methoden gelesen. Daran anknüpfend sollten Sie Ihre Entscheidung für oder wider die induktiv-qualitative Inhaltsanalyse von methodisch-systematischen, jedoch forschungspragmatischen Gesichtspunkten leiten lassen, welche im Folgenden grob skizziert und diskutiert werden.

1 Beispielsweise gilt dies auch für spezifische, darauf aufbauende Methoden wie die objektive Hermeneutik (z. B. Oevermann 2013).

4.1.1 Methodische Anforderungen der induktiv-qualitativen Inhaltsanalyse

In den Kapiteln 4.3.1 (Übersicht gewinnen), 4.3.2 (Strukturierende Analyse) und 4.3.3 (Induktive Kategorien- und Kodeentwicklung) wird das methodische Vorgehen am Beispiel einer Autoethnographie als empirisches Dokument vorgestellt. Werten Sie jedoch zwei und mehr Dokumente aus, ganz unabhängig davon, ob es sich dabei um Autoethnographien, Gruppendiskussionen, Interviews oder Teile von Parteiprogrammen handelt, müssen Sie das Vorgehen „Übersicht gewinnen", „strukturierende Analyse" und „induktive Kategorien- und Kodeentwicklung" ein-, zwei- bis n-Mal gemäß der Anzahl der auszuwertenden Dokumente wiederholen.[2]

Das n ist begrenzt, wenn Sie beispielsweise für eine Bachelorarbeit drei oder für eine Masterarbeit fünf Interviews führen. Das n müssen Sie begründen, wenn Sie beispielsweise 50 Autoethnographien für die Sekundäranalyse vorliegen haben und nur eine Auswahl davon induktiv-qualitativ auswerten möchten. Haben Sie die Daten selbst erhoben, führen also eine Primäranalyse durch, so müssen Sie das n von Gruppendiskussionen, der zu führenden Interviews usw. bereits im Forschungsdesign begründen (für qualitative Stichprobenziehung siehe Akremi 2019 und Kapitel 5). Die als Beispiel verwendete Autoethnographie der*des Studierenden wäre ein Fall für den (relativ) normalen Studienstart bei Beeinträchtigung durch die Bedingungen der Corona-Pandemie. Wollen Sie die Vielfalt oder gar extreme Fälle (z. B. Borchardt und Göthlich 2009; Muno 2009; Seawright und Gerring 2008) von sozialen Bedingungen des Studienbeginns im Wintersemester 2020/21 abbilden, so könnten Sie, falls vorhanden, einen Fall einer*eines Studierenden mit Care-Aufgaben auswählen, die*der zusätzlich zum Studienstart die Kinderbetreuung gewährleisten muss, da die Kinder wegen positiver Covid-19-Testbefunde temporär nicht in die Kita oder Schule gehen können. Weitere gut nachvollziehbare Fallentscheidungen wären für die Autoethnographie einer*eines Studierenden, die*der das Studium durch einen Nebenerwerb teilfinanzieren muss, oder einer*eines Seniorstudierenden, welche*r sich nach Renteneintritt den interessengeleiteten Studienwunsch erfüllt.

Ebenso wie die Begründung der Fallauswahl systematisch durch Erkenntnisinteresse und in Verbindung mit der Methode vor Beginn der Forschung erfolgen muss, so sollten Sie auch nicht während der Auswertung Ihre inhaltsanalytische Herangehensweise ändern. Seien Sie sich gewahr, dass die induktiv-qualitative Inhaltsanalyse sehr arbeits- und zeitaufwendig ist. Daher ist es forschungspragmatisch notwendig, dass Sie unter Berücksichtigung Ihrer verfügbaren Zeit für die induktiv-qualitative Forschung den Umfang des auszuwertenden Datenmate-

2 n repräsentiert konventionell eine numerisch unklare Anzahl (z. B. von auszuwertenden Fällen) und dürfte Ihnen aus dem Mathematikunterricht in der Schule bekannt sein.

rials begrenzen, damit Sie nicht mitten im Forschungsprozess nach Abkürzungen suchen müssen.

So gehen wir davon aus, dass aus dem gesamten empirischen Material iterativ immer neue induktive Kategorien hinzukommen. Entsprechend sehen wir das Kodieren des empirischen Materials ohne Berücksichtigung des gesamten empirischen Datenmaterials sehr kritisch, wie es Mayring (2010) vorschlägt, da der qualitative Textbezug verloren geht. Beispielsweise empfiehlt Mayring (2010, S. 84), dass nach 10 % bis 50 % der Materialdurchsicht die induktiv entwickelten Kategorien für die Kodierung festgelegt werden und nicht mehr verändert werden sollen. Folglich werden Inhalte, die nicht im festgelegten Kategoriensystem kodiert werden können, nicht mehr berücksichtigt. Das kann zu Verzerrungen führen und entspricht deshalb nicht dem Ziel der induktiven Forschung. Dieses gravierende qualitative Manko kann auch nicht durch quasi-Quantifizierung mit „Interkoderreliabilität" (Mayring 2010, S. 51, S. 61 f.) als Vorschlag zur Überwindung des Qualitativ-quantitativ-Gegensatzes behoben werden (siehe Gütekriterien der Inhaltsanalyse in Kapitel 2.3). Interkoderreliabilität bedeutet, dass zwei oder mehr Forscher*innen dasselbe Datenmaterial auswerten und dann verglichen wird, welche Kategorien und dazugehörige Kodes sie gleich (= Standardisierung) und anders (= Standardfehler) vergeben haben. Reliabilität bezeichnet die Zuverlässigkeit bzw. die Verlässlichkeit von (quantitativen) Messungen basierend auf der Annahme, dass eine Messung potenziell wiederholbar ist „bei unterstellter Stabilität des zu messenden Sachverhalts" (Diaz-Bone 2015, S. 169). Qualitativ gewendet, adressiert Interkoderreliabilität also die Messgenauigkeit und Messfehler der kodierenden Subjekte. Folglich ist der Bezugspunkt von Interkoderreliabilität das kognitive Analysepotenzial der kodierenden Subjekte zwecks intersubjektiver Nachvollziehbarkeit derer manueller Kodiertätigkeit.

Laut Mayring (2010, S. 51) soll Interkoderreliabilität einen Beitrag zur Gültigkeit (im Soziolekt: Validität) der Ergebnisse leisten, welche nach Flick (2019, S. 476–478) dem qualitativen Gütekriterium „Validierung durch Kommunikation" im Sinne der „Transparenz der Vorgehensweisen" (Flick 2019, S. 483) dem Erzielen von Ergebnissen zuzuordnen ist. Folglich ist kommunikative Validierung, beispielsweise mit der*dem Betreuer*in der Qualifikationsarbeit, mit Kommiliton*innen oder mit Kolleg*innen, fester Bestandteil des gesamten qualitativen Forschungsprozesses vom Forschungsdesign über die Materialauswahl und Erhebungsinstrumente bis zur Darstellung von Ergebnissen (siehe auch Strübing et al. 2018). Bei Mayring (2010) sind jedoch nur 10 % bis 50 % der Messung reliabel, was zur logischen Folge hat, dass die Validität der Ergebnisse unbekannte Maße annimmt. Anders ausgedrückt: Wenn Sie dem vom Mayring (2010) empfohlenen Vorgehen folgen, ist es sehr gravierend für die Validität der Ergebnisse Ihrer qualitativen Inhaltsanalyse, da Sie mitten im induktiven Auswertungsprozess und bei vollem Wissen, dass 50 % bis 90 % mehr qualitative Informationen im empirischen Material stecken, die induktive Inhaltsanalyse abbrechen.

Methodisch ist das von Mayring (2010) empfohlene Vorgehen auch deshalb als höchst kritisch einzustufen, weil die qualitative und quantitative Logik durcheinandergeworfen wird, was im Ergebnis eine oft intransparente Vermischung von manifesten Häufigkeiten und als qualitativ dargestellte Erklärungen bedeutet (siehe Ausschlussregel Kapitel 2.2.4). Zum Beispiel hätte die quantitative Auszählung von Veranstaltungsbesuchen und dazugehörige Erzählungen keine qualitative Bedeutung, wenn Tagebucheinträge von Studierenden zum Semesterbeginn inhaltlich analysiert werden, bei denen Veranstaltungsbeuche Teil der Aufgabenstellung sind. Denn wenn das Thema Veranstaltungen nicht häufig(er) erwähnt würde in der Autoethnographie, so wäre das Thema der Autoethnographie verfehlt.

Methodisches Durcheinander von qualitativer und quantitativer Inhaltsanalyse sollten Sie bereits bei der Erstellung des Forschungsdesigns durch den Abgleich Ihres Forschungsinteresses und bei Berücksichtigung des Datenumfangs vermeiden (siehe Kapitel 3 zu Spezifika von Daten und damit verbundenen Möglichkeiten und Grenzen sozialwissenschaftlicher Analysen). Wenn Sie größere Textmengen – d. h. mehr empirische Daten als in diesem Kapitel zu induktiv-qualitativer Inhaltsanalyse vorgestellt wird – auswerten möchten, dann sollten Sie die Inhaltsanalyse systematisch methodengeleitet entweder

- deduktiv (siehe deduktiv-qualitative Inhaltsanalyse, Kapitel 5),
- quantitativ teilautomatisiert (siehe quantitative Inhaltsanalyse und Auswertungstechniken, Kapitel 6) oder
- quantitativ vollautomatisiert (siehe Korrespondenzanalyse, Kapitel 9)

durchführen.

Dennoch sollten Sie das methodenpuristische Plädoyer nicht falsch verstehen: Sie können selbstverständlich qualitative und quantitative Auswertungstechniken für Inhaltsanalysen kombinieren, sofern Sie diese unter Berücksichtigung von jeweils qualitativer und quantitativer Logik in einem Mixed Methods-Forschungsdesign systematisch mit Blick auf den angestrebten Erkenntnisgewinn im Forschungsdesign verbinden (z. B. Burzan 2016; Kelle 2019).

4.1.2 Forschungsprozess als Schritt für Schritt-Ablaufschema

Die Diskussion um die Transparenz von empirischer Sozialforschung, intersubjektiver Nachvollziehbarkeit und Validität der Ergebnisse zeigt, dass auch qualitative Sozialforschung regelhaften Mustern folgt. Die Gütekriterien und die darin ausgedrückten regelhaften Muster helfen, die methodische Systematik der wissenschaftlichen Erkenntnis vom Alltagserkennen zu unterscheiden (z. B. Feyerabend 1980; Fleck 2019; Reichertz 2016; siehe auch Kapitel 2). In der qualitativen

Sozialforschung definieren die regelhaften Muster bestimmte methodische Bedingungen als Kriterien und Verfahren für den qualitativen Forschungsprozess. So betonen auch Flick (2019), Strübing et al. (2018) und andere (siehe Beiträge Akremi et al. 2018; Lamnek und Krell 2016; siehe auch Kapitel 2), dass induktiv-qualitative Sozialforschung nicht theoriefrei ist und die vertiefte qualitative Inhaltsanalyse erst nach der Interpretation und theoretischen Reflexion von manifesten und gegebenenfalls latenten Inhalten abgeschlossen ist (z. B. Knoblauch et al. 2018; Strübing et al. 2018).

Das Ablaufschema für die induktiv-quantitative Inhaltsanalyse in Abbildung 4.1 wird hier nicht weiter ausgeführt, um Redundanzen zu vermeiden. Die einzelnen Schritte werden im folgenden Text sowohl theoretisch-methodisch als auch mit Beispielen erklärt. Eine Ausnahme ist Schritt sechs, welcher genereller Natur ist. Wenn Sie mit personenbezogenen Daten arbeiten, so müssen Sie nicht nur die Anonymität der Befragten (siehe Kapitel 4.2.2), sondern auch die Sicherheit der Daten bei der Archivierung gewährleisten, d. h. Audiodateien, Listen mit Kontaktangaben, nicht anonymisierte Transkripte usw. an einem sicheren, gegebenenfalls passwortgeschützten Ort (z. B. Server und USB-Stäbchen) aufbewahren. Der Aspekt der Datennachnutzung ist für Sie nur relevant, wenn

Abbildung 4.1 Ablaufschema der induktiv-qualitativen Inhaltsanalyse

Schritt 1	Erkenntnisinteresse als Fragestellung formulieren, basierend auf 1. rezipierter Literatur, 2. eigenen vorherigen empirische Untersuchungen, und/oder 3. sonstigem Vorwissen (z. B. eigene Erfahrungen) und Erstellung des Forschungsdesigns (inklusive Methoden und Begründung der Fallauswahl).
Schritt 2	Datenaufbereitung: 1. Datenqualität und Entstehungskontext 2. Anonymisierung von Daten
Schritt 3	Vorbereitung induktiv-qualitative Inhaltsanalyse: 1. Übersicht zu Textmaterial und Inhalte zusammenfassen 2. Strukturierende Analyse des Datenmaterials 3. Induktive Kodeentwicklung und Kategorien 4. Kodierung des Datenmaterials
Schritt 4	Durchführung qualitative Inhaltsanalyse: 1. Manifeste und latente Inhalte identifizieren 2. Manifeste und latente Inhalte erklären und interpretieren 3. Theoretische Reflexion
Schritt 5	Ergebnisse zusammenfassen und Erstellung einer Präsentation, Studienarbeit und/oder Publikation.
Schritt 6	Sichere Archivierung der Daten und, wenn möglich, Aufbereitung zur Nachnutzung.

Sie im Rahmen eines Seminars oder anderen größeren Forschungsprojekten empirische Daten erheben (siehe Kapitel 3.4).

4.2 Die Textdaten der induktiv-qualitativen Inhaltsanalyse

4.2.1 Entstehungskontext

Für qualitative Sozialforschung insgesamt und insbesondere für die induktiv-qualitative Inhaltsanalyse ist der Entstehungskontext der empirischen Daten von großer Bedeutung (siehe Kapitel 2.2). Die große Bedeutung des Entstehungskontexts hat vor allem zwei Gründe: Erstens sind die empirischen Daten als exklusiver Bezugspunkt von induktiven Auswertungstechniken der qualitativen Inhaltsanalyse in einem bestimmten räumlichen und zeitlichen Sozialkontext entstanden. Der Entstehungskontext der vorliegenden Empirie ist die Autoethnographie einer*eines Studierenden, die*der im Wintersemester 2020/21 ihr*sein Studium an einer Universität in Deutschland online begann. Zweitens, quasi als wechselseitige Unterstützung von Erstens, begrenzt der Entstehungskontext erkenntnistheoretisch und analytisch den empirischen Rahmen einer induktiv-qualitativen Inhaltsanalyse.

Wie einleitend betont, bedeutet die induktive Auswertungstechnik und qualitative Inhaltsanalyse keine voraussetzungslose Herangehensweise an die empirischen Daten, denn Sie als Sozialforscher*in verfügen in der Regel über allerlei Erfahrungs-, Forschungs-, Hörensagen-, theoretisches usw. Vorwissen über den Untersuchungsgegenstand (z. B. Armat et al. 2018; Strauss 2007). Auch bedeutet eine induktive Auswertungstechnik nicht, dass Sie theoriefrei forschen, jedoch ist Theorie erst zu einem späteren Zeitpunkt ein wichtiger Schritt im Ablauf induktiv-qualitativer Inhaltsanalyse (Schritt 4 in Abbildung 4.1).

Der Entstehungskontext erfasst auch, dass textbasierte Kommunikation nichts Natürliches ist, wie Tag und Nacht, Wind und Regen, sondern Textkommunikation stets der Konstruktion und Rekonstruktion bedarf. Die sozialwissenschaftliche Konstruktionsleistung von Text wird in der Regel von menschlichen Subjekten erbracht. Menschen kommunizieren von sich aus und/oder als Reaktion auf vorangegangene Kommunikation (z. B. Wolff 2000). Teilweise ist der Entstehungskontext von Text vage (z. B. gezeichnete Symbole in Höhle) und nicht dokumentiert (z. B. dokumentiert in Archiv). Dies gilt es bei der Beschreibung des Entstehungskontexts kenntlich zu machen und bei der Analyse zu berücksichtigen (siehe Kapitel 3).

Im Zusammenhang von Möglichkeiten und Grenzen aufgrund von Datenbeschaffenheiten für empirische Sozialforschung sollte an dieser Stelle darauf hingewiesen werden, dass es sich um eine Sekundäranalyse handelt. Analysiert

werden Tagebucheinträge einer*eines Studierenden, die nach der Methode der Autoethnographie erstellt wurden. Die Sekundäranalyse wird ermöglicht durch die von der*dem Studierenden unterschriebene Einwilligungserklärung (siehe Anhang 3.2). Das Wissen über den Entstehungskontext und die Zusicherung von Anonymität im Austausch für die Daten (siehe Anhang 3.1) sind von hoher Relevanz für einen wichtigen Schritt der Datenpräsentation im folgenden Abschnitt.

4.2.2 Anonymisierung

Die Anonymisierung von Autoethnographien, Gruppendiskussionen, Interviews und anderen personenidentifizierenden Daten erfüllt vor allem drei Zwecke. Anonymisierung …

a) … bietet einen individuellen Schutz für die Informationen gebenden Subjekte,
b) … ermöglicht die*dem Interviewten frei über das gewählte Thema zu sprechen und
c) … verschleiert die Wiedererkennbarkeit von Personen, Institutionen, Orten usw.

Außer zu Lehrzwecken werden in wissenschaftlichen Veröffentlichungen keine ganzen Texte angegeben. Ausnahmen im Studium können sozialwissenschaftliche Methoden- und Empirieseminare sein, in denen die*der Dozent*in Sie auffordert, dass Sie beispielsweise Transkripte von Interviews im Anhang der Studienarbeit mit abgeben sollen. Ein erster Schritt der Anonymisierung ist die Vergabe von Identifikationsnummern (idno als „ID“ und „no“ von Englisch *number* ist als Abkürzung flüssiger zu lesen als das Deutsche „nr“), wie im unteren Beispiel „idno05“ und die Verwendung der genderneutralen Schreibweise „die*der Studierende“ im Text. Die Autoethnographie der*des Studierenden idno05 ist eine von 50 von Studierenden freiwillig für die Forschung zur Verfügung gestellten autoethnographischen Aufzeichnungen aus einer Veranstaltung mit etwa vier- bis fünfmal so vielen Studierenden. Die Mehrheit der Veranstaltungsteilnehmer*innen hat sich entweder nicht an der freiwilligen Übung beteiligt oder möchte ihre Tagebucheinträge für sich behalten. Die Aufschlüsselung, welche*r Studierende sich hinter welcher „idno“ verbirgt, wurde in einer Excel-Datei dokumentiert. Sollte die Datei auf einem Laufwerk oder in einer Cloud lagern, auf welche(s) mehrere Personen Zugriff haben, so ist es unerlässlich, dass Sie die Datei mit einem Passwortschutz versehen.

Die Identifikationsnummer sollten Sie sowohl im Fließtext als auch bei Zitaten immer zum Verweis auf die Stellen im Originaltext verwenden (z. B. idno05: Absatz 7/6; siehe Tabelle 4.1). Idno ist hier im Buch aufgrund der Kürze gewählt, Sie könnten jedoch auch Studierende_05 zwecks Identifikation verwenden. Ha-

ben Sie beispielsweise drei Interviews geführt, so könnten Sie auch Interview_1, Interview_2 und Interview_3 zwecks Anonymisierung wählen. Möglich wäre auch die Vergabe von anderen Realnamen, wobei Sie bei heterogenen Gruppen von Interviewten stets das biologische Geschlecht mittransportieren. Sozio-biologisch neutral würde in Ihrem Fließtext eine Gegenüberstellung von Aussagen etwa wie folgt lauten: Im Gegensatz zur Aussage in Interview_1, wurde in Interview_3 für … und gegen … argumentiert.

Pragmatisch sollten Eingriffe in den empirischen Originaltext einheitlich und gut sichtbar vorgenommen werden. Im Beispiel in Tabelle 4.1 werden eckige Klammern verwendet, welche selten in Fließtexten oder Social Media-Textschnipseln verwendet werden. Generell sollten allgemein bekannte (schriftbezogene) Konventionen die praktische Entscheidung der*des Forscher*in zur Kenntlichmachung von Eingriffen der*des Forschenden in den Text leiten, wie drei Punkte für Auslassungen. Entsprechend wurde „[…]" für gelöschten Text gewählt. Idealerweise sollten knappe Ersatzbegriffe gewählt werden, wobei es inhaltlich unerheblich ist, ob „[Stadtname]" oder [Name der Stadt] verwendet wird.

Die Anonymisierung sollte auch von inhaltlichen Gesichtspunkten geleitet sein. Beispielsweise wird die Nennung des Studiengangs ausgelassen und durch die Verwendung von „[Hauptfach]" und „[Nebenfach]" Platzhaltern anonymisiert. Damit wird kenntlich, dass Einträge zu unterschiedlichen Pflicht- und Wahlpflichtveranstaltungen des Studiengangs vorliegen, und auch, dass die*der Studierende eine weder dem Haupt- noch Nebenfach zugehörige Wahlveranstaltung besucht – gekennzeichnet durch [fachfremder Zusatzkurs]. Bei einem Tagebucheintrag ist auch der Tagesablauf relevant. Daher wurden zwar Zeiten von Veranstaltungen gelöscht, um deren Recherchierbarkeit zu verhindern, denn das Internet vergisst nicht und die Suchmaschinen sind gut, jedoch durch relativ eindeutige Tageszeiten wie „[Vormittag]", „[Nachmittag]" und „[früher Abend]" ersetzt. Durch die Anonymisierung der Namen von Stadt und Universität wird die Wahrscheinlichkeit der Auffindbarkeit der*des Studierenden weiter verringert, da es in Deutschland über 100 Universitäten unter den etwa 400 Hochschulen und mehr als 8 000 Studiengänge gibt.

Selbstverständlich kann es sein, dass Sie eine Information übersehen, welche durch Verwendung eines bestimmten Wortes oder einer Beschreibung vermeintlich den Ort oder die Universität für *Kenner*innen* sichtbar erscheinen lässt. Das kann passieren, und ist bei nachweisbarem Aufwand durch die empirische Sozialforscher*in zwar ärgerlich, jedoch entschuldbar. Beispielsweise könnten Sie als Leser*in dieses Buches Nachforschungen anstellen, von welcher Universität die Textdaten der Autoethnographie stammen könnten. Dazu könnte Sie bei den Lebensläufen der drei Autor*innen beginnen. Das würde die Auswahl auf wenige Universitäten in Deutschland einschränken. Aber selbst wenn Sie die Suche weiterführen und eine Publikation von Isabel Steinhardt zu Autoethnographie finden (Autor:innengruppe AEDiL 2021), können Sie nicht sicher sein,

ob nicht ein*e Kolleg*in von einer anderen Universität uns die Datenanalyse unter ihrer*seiner Aufsicht ermöglicht hat (siehe Einwilligungserklärung der Studierenden in Anhang 3.2). Sie sehen, in der Regel lohnt schon der Aufwand, der Neugier nachgehen zu wollen, nicht, und ist auch nicht von Interesse. Selbst unter den Studierenden der Veranstaltung ist nicht bekannt, wer eingewilligt hat, dass ihre*seine Autoethnographie für die Forschung verwendet werden darf. Natürlich wissen es die Studierenden selbst. Doch ohne explizite (nachträgliche) Selbstanzeige oder Bestätigung von Nachfragen bleiben die Kommiliton*innen und andere Dritte darüber im Unklaren.

Teilweise ist es für die Anonymisierung notwendig, stärker in den Text einzugreifen. Durch den Eingriff sollte jedoch der Kern der empirischen Information erhalten bleiben bzw. nicht verfälscht werden. Beispielsweise wurde in folgendem Textausschnitt der Satz umgebaut, um den Lesefluss zu gewähren: „[…] [der Studierendengruppe] [aus Land] in [Stadtname]“ (Tabelle 4.1, Abschnitt 2/7). Durch Einfügen von „[…]“ wird eine Auslassung im Text kenntlich gemacht, und durch „[aus Land] in [Stadtname]“ Ergänzungen im Text durch die*den Forscher*in. Die Information „[aus Land]“ wurde statt einer Auslassung eingefügt, da es an allen Universitäten Studierendenzusammenschlüsse nach Land, geographischen Regionen (z. B. Lateinamerika) ergänzend zu Studierendenorganisationen wie Fachschaften und den Allgemeinen Studierendenausschuss (AStA) gibt. Dies wissend, können Sie davon ausgehen, dass der Anonymisierung genüge getan wurde. Sind Sie unsicher und können Informationen nicht einschätzen, so sollten sie im Zweifel immer vorsichtig sein und Informationen unkenntlich machen oder gar teilweise löschen, welche zur Identifikation von Informant*innen führen könnten.

4.2.3 Beispieltext: autoethnographischer Beitrag zum Studienbeginn Wintersemester 2020/21

Nach der erfolgten Anonymisierung und um den oben angesprochenen Unsicherheiten beim induktiven Kodieren zu begegnen, wird im Folgenden der gesamte Text dargestellt, welcher von einer*einem Studierenden erstellt wurde (Tabelle 4.1). So können Sie

a) nachvollziehen, wie beispielhaft Kodes induktiv entwickelt wurden, und
b) versuchen, eigene Kodes induktiv aus den als Text vorliegenden Informationen (= Datenmaterial) zu entwickeln.

Die*der Studierende hatte beim Autoethnographieren pro Tag einen zusammenhängenden Absatz als Text geschrieben. In Tabelle 4.1 wurde der Text der*des Studierenden für jeden Tag in Sinneinheiten unterteilt, welche bei der Analyse im

Text als Verweis oder für ein Ankerbeispiel verwendet werden können – analog zu Absatzmarken in MAXQDA (siehe Kapitel 4.2.3).

Die Untergliederung der Daten in Sinneinheiten greift der Analyse im folgenden Abschnitt etwas voraus, da damit eine erste Ordnung des Datenmaterials vorgenommen wird. Eine Sinneinheit definiert sich als zusammenhängende Beschreibung einer Veranstaltung (z. B. Vorlesung und Seminar), einer Handlung bzw. eines Handlungszusammenhangs. Eine neue Sinneinheit wurde in Tabelle 4.1 erstellt, wenn beispielsweise ein neuer Tagesabschnitt samt neuer Handlung begann und die Handlung sich gut erkennbar änderte, wie zum Beispiel: frühes Aufstehen wegen Nervosität am ersten Uni-Tag (Absatz 1/1), Kaffeekochen (Absatz 1/2) und Kaffee trinken am Computer beim E-Mail-Checken (Absatz 1/3). Im Gegensatz zum ersten Uni-Tag sind die drei Elemente am zweiten Uni-Tag nicht mehr als Einzelhandlungen relevant, sondern nur im Verbund, was sich auch an der zusammenhängenden Darstellung in Absatz 2/1 zeigt. Es ist auch nachvollziehbar, dass beispielsweise an einem Tag viel passiert und daher mehr Sinneinheiten im autoethnographischen Text identifiziert werden können (insbesondere Tag 1), und dass an anderen Tagen weniger Berichtenswertes im Leben von Studierenden passiert, wodurch nur vier Sinneinheiten identifiziert werden können (z. B. Tage 5, 6, 8 und 9). Sie sehen, dass die Erstellung von Sinneinheiten themenorientiert (z. B. Handlungen) ist und für Dritte gut nachvollziehbar begründet werden muss, jedoch keine komplizierte Wissenschaft ist.

Tabelle 4.1 Autoethnographie Studienbeginn Wintersemester 2020/21 (Fortsetzung auf den nächsten Seiten)

Absatz	Tagebucheintrag
1	Montag, den [Tag] [Monat] 2020
1/1	Um 7:30 Uhr hat mein Wecker geklingelt, und ich stand nicht ganz so ausgeschlafen auf, durch meinen ersten Tag an der Uni war ich etwas aufgeregt und konnte nicht so gut schlafen wie gewohnt.
1/2	Ich ging runter in die Küche und habe mir einen Kaffee gemacht, so wie jeden Morgen. Meine Mutter war auch noch in der Küche und hat mich gefragt, was denn meine erste Vorlesung sein wird. Kurz habe ich erklärt, dass es sich hierbei um die Vorlesung [Hauptfach] handelt, und diese um [Vormittag] Uhr anfängt.
1/3	Mit meinem Kaffee in der Hand bin ich hoch in mein Zimmer gegangen, habe meinen Laptop hochgefahren und meine E-Mails gecheckt. In den E-Mails habe ich noch verschiedene Bescheide zu den Fixplätzen erhalten für Module, in die ich mich für dieses Semester eingeschrieben habe.
1/4	Ich habe mich daraufhin versucht, bei Moodle einzuloggen, um mir nochmal anzuschauen, was mich heute in der Vorlesung erwarten wird, ich hatte mir bereits am Wochenende den Text durchgelesen, aber zur Sicherheit habe ich noch mal nachgesehen.

Absatz	Tagebucheintrag
1/5	Daraufhin habe ich mir etwas Anständiges angezogen und mich fertig gemacht, um am Zoom-Meeting teilzunehmen.
1/6	Als es dann [Vormittag] war, wollte ich mich bei Moodle einloggen, um zum Link des Zoom-Meetings zu gelangen, aber das hat sich als nicht möglich herausgestellt, da Moodle lahmgelegt war, durch all die neuen Studenten, die sich versucht haben anzumelden. Einer aus der WhatsApp Gruppe der Erstis unseres Studiengangs hatte den Zoom-Link gespeichert, und ihn in die Gruppe gesendet, sodass wir doch noch zeitlich an der Zoom-Vorlesung teilnehmen konnten.
1/7	In meiner ersten Zoom-Vorlesung ging es dann um die Autoethnographie und wir haben die Aufgabe erhalten, selbst eine zu schreiben. Wir sollen zehn Tage autoethnographisch festhalten, und uns auf dieses Experiment einlassen.
1/8	Nach dieser Vorlesung war mein Uni-Tag auch schon zu Ende, da ich an diesem Tag nur diese eine Vorlesung hatte.
1/9	Am Mittagstisch habe ich über die erste Vorlesung berichtet, und habe erzählt, wie spannend es war.
1/10	Am Nachmittag dann, habe ich mich dazu entschieden, erst einmal alles zu sortieren und einzuordnen was ich bis jetzt erhalten habe von Texten, Aufgaben und Informationen der jeweiligen Module, die ich an der Uni belege. In einem Ordner habe ich für jedes Modul eine Unterteilung eingerichtet und die schon erhaltene Literatur herausgedruckt und eingeordnet. Danach habe ich mich dazu entschieden schon den nächsten Text für ein anderes Modul zu lesen, um mich etwas vorzuarbeiten.
1/11	Nach dem Abendessen habe ich mich noch dazu entschieden, die Prüfungsordnung durchzulesen, weil man diese ja gut verinnerlichen soll, damit man in Kenntnis ist von all seinen Rechten während der Klausurphase.
1/12	Um 23:00 Uhr habe ich mich dazu entschieden, ins Bett zu gehen, bevor ich aber schlussendlich das Licht ausgemacht habe, habe ich noch den Stundenplan für Morgen geguckt in der [Universität]-App auf meinem Handy.
2	Dienstag, den [Tag] [Monat] 2020
2/1	Um 7:30 Uhr hat mein Wecker geklingelt, ich habe mein Handy genommen und geschaut, ob ich eine E-Mail bekommen habe und meinen Stundenplan auf der [Universität]-App aufgerufen, um zu schauen, welche Vorlesung ich heute als erstes habe.
2/2	Dann bin ich runter in die Küche gegangen, habe mir meinen Kaffee gemacht und mich wieder nach oben begeben, um meinen Laptop anzumachen, und bei Moodle reinzuschauen, ob ich eine Benachrichtigung erhalten habe und es irgendetwas Neues gibt.
2/3	Dann um [Vormittag] Uhr hat auch schon meine erste Vorlesung begonnen, die Einführung in [Hauptfachfach]. Ich hatte mir den Text zur Vorlesung schon durchgelesen am Vorabend und konnte so dem Gesprochenen der Vorlesung gut folgen.
2/4	Nach diesen 90 Minuten hatte ich bis [Nachmittag] keine Vorlesung mehr, so habe ich mich also anders beschäftigt und noch weiter im Internet nach Wohnungsanzeigen gesucht, und welche angeschrieben. Ich bin momentan nämlich noch auf Wohnungssuche, und durch die aktuelle Situation der Corona-Pandemie gestaltet sich diese schwieriger als gedacht.

Absatz	Tagebucheintrag
2/5	Gegen [Nachmittag] ist uns, Erstis, aufgefallen, dass wir keinen Zoom-Link zu unserer Vorlesung für [Nachmittag] hatten. Kurz vor Beginn der Vorlesung hat sich dann herausgestellt, dass diese Vorlesung in [Nebenfach] asynchron stattfinden wird. Das heißt, der Dozent lädt die Vorlesung im Moodle hoch und wir können uns diese dann anschauen, wann wir Zeit haben. Ich habe mich dann gegen [früher Abend] dazu entschieden, mir schon mal die Hälfte anzuschauen, also den Einführungsteil und etwas vor dem ersten Thema, was wir behandeln werden.
2/6	Nach dieser Einführung habe ich etwas gegessen und mich für das Seminar vorbereitet was um [Abend] stattfindet. Das Seminar ging dann von [Abend].
2/7	Daraufhin habe ich meine Notizen weggeräumt und bin zu einer Zoom-Veranstaltung […] [der Studierendengruppe] [aus Land] in [Stadtname] hinzugestoßen, die die Erstis herzlich willkommen geheißen haben, [ihre Studierendengruppe] etwas vorgestellt haben und sich über unsere Mitgliedschaft freuen würden.
2/8	Gegen 23:00 Uhr bin ich dann schlafen gegangen, nachdem ich mir meinen Stundenplan angeschaut hatte und mir den Wecker für 7:30 Uhr gestellt hatte.
3	Mittwoch, den [Tag] [Monat] 2020
3/1	Um 7:30 Uhr hat mein Wecker geklingelt, ich bin noch nicht ganz ausgeschlafen aufgestanden und bin in die Küche gegangen, um mir einen Kaffee zu machen. Ich habe mich an den Küchentisch gesetzt und mit [Geschwister] über meine ersten zwei Tage an der Uni gesprochen, er hat momentan Herbstferien und findet es noch immer komisch, dass wir nicht mehr in dieselbe Schule gehen.
3/2	Nachdem ich meinen Kaffee ausgetrunken hatte, habe ich mir eine Wasserflasche mit hoch auf mein Zimmer genommen, mich an meinen Schreibtisch gesetzt und den Laptop angemacht, um an der Zoom-Veranstaltung von [Vormittag] teilzunehmen. In dem Propädeutikum [Hauptfach] haben wir darüber gesprochen, was auf uns zukommen wird das kommende Semester.
3/3	Anschließend hatten wir das Seminar für [Hauptfach] von [Vormittag]. Am Mittagstisch habe ich über meine erste Uni-Woche berichtet, und darüber, dass ich am Donnerstag noch die Übungsgruppe des Modules [fachfremder Zusatzkurs] haben werde und mir dazu unbedingt noch die Vorlesung anschauen will für Aufgabe [B], auch wenn es über Aufgabe [A] gehen wird, aber falls ich Fragen hätte, ich sie stellen könnte.
3/4	Nach dem Abendessen habe ich mich zusammengerauft und mich dazu entschieden, die Vorlesung doch noch heute anzufangen, um morgen besser voran zu kommen. Daraufhin habe ich mir den Stundenplan herausgesucht und geschaut, was noch so ansteht und was noch zeitnah gemacht werden muss.
3/5	Ich finde es eine große Umstellung von meine[r Schule], den langen Ferien, jetzt wieder anzufangen mit Lernen, jedoch bin ich richtig gespannt, was auf mich zukommt und lasse mich gerne positiv von diesem neuen Bachelorstudium [Hauptfach] überraschen.
3/6	Um 22:30 Uhr habe ich mich ins Bett gelegt und mir keinen Wecker gestellt, da ich morgen ausschlafen kann, und die Übungsgruppe erst nachmittags stattfindet.
4	Donnerstag, den [Tag] [Monat] 2020
4/1	Bis [Vormittag] habe ich ausgeschlafen, weil in unserem Stundenplan nichts mehr steht, auf Moodle jedoch wurde uns mitgeteilt, dass wir in den Übungsgruppen für das Modul [fachfremder Zusatzkurs] verschiedene Stunden haben. Ich bin in der Übungsgruppe [A] und habe somit Donnerstag [Nachmittag] diese Stunde.

Absatz	Tagebucheintrag
4/2	Nach dem Aufstehen bin ich in die Küche gegangen und habe mir wie jeden Morgen einen Kaffee gemacht, mein [Geschwister] war auch schon wach und wir haben darüber gesprochen, was wir heute machen würden. Daraufhin habe ich ihm kurz erläutert, dass ich mich sofort an die Übung für [fachfremden Zusatzkurs] setzen will, da ich erstens gut vorbereitet sein will und zweitens noch die Vorlesung und die Übung [B] für Montag zu erledigen habe.
4/3	Als ich wieder an meinem Schreibtisch saß, habe ich mein Laptop angemacht und meine E-Mails gecheckt, meine private sowie die [Universität]-Mail-Adresse. Es gab Neuigkeiten bezüglich einer Besichtigung für eine Wohnung in [Stadtname]. Nach kurzer Absprache hatte ich also einen Besichtigungstermin für Samstag.
4/4	Dann habe ich bei Moodle reingeschaut, ob es dort Neues gibt und mich dann an die Vorlesung von [fachfremder Zusatzkurs] gesetzt, die wirklich viel Zeit in Anspruch nimmt, da ich keine Vorkenntnisse in diesem Fach besitze. Über den ganzen Tag verteilt, ist mir aufgefallen, dass ich mir immer wieder Gedanken darüber mache, da ich wirklich Schwierigkeiten bei der Umsetzung habe der Aufgaben, die wir bekommen. Ich habe mir zum besseren Verständnis das Buch: [fachfremder Zusatzkurs] bestellt, und hoffe, dass es die ersehnte Hilfe zum Verständnis bringt.
4/5	Am Nachmittag bin ich noch einmal über meine erste Übung gegangen, die wir bereits Montag abgegeben hatten, um vorbereitet an der Übungsgruppe teilzunehmen. Um [Nachmittag] dann hat diese Stunde begonnen; am Anfang jeder Übungsstunde werden [Anzahl] Matrikelnummern gezogen und diese Studenten werden dann [getestet werden] am Ende der Stunde, ich war dieses Mal nicht dabei und auch etwas froh darüber, da ich mich noch nicht so sicher in der Materie fühle.
4/6	Dann war mein Studienalltag auch schon zu Ende, ich habe mein Laptop zugeklappt und den Rest des Tages mit anderen Sachen beschäftigt, die in der Woche liegengeblieben sind wegen dem neuen Mittelpunkt meines Alltags, dem Studium.
5	Freitag, den [Tag] [Monat] 2020
5/1	Freitags habe ich keine Vorlesungen und kann Nacharbeit der verschiedenen Module machen, aber mich auch schon vorarbeiten, und alle erhaltenen Texte bereits ausdrucken und einordnen, damit ich eine klarere Sicht auf die nächste Woche habe und auf das, was ansteht. Ich bin also etwas später als sonst aufgestanden, weil ich mich schwer tue damit, wieder früher aufzustehen nach den langen Ferien, auch wenn ich gearbeitet habe in dieser Zeit.
5/2	Um [Vormittag] habe ich mir einen Kaffee gemacht, mich an den Küchentisch gesetzt und mir auf meinem Handy Moodle aufgerufen, um zu schauen, ob es dort Neuigkeiten gibt. Nachdem ich auch meine E-Mails durchhatte, war es auch schon Mittag.
5/3	Am Mittagstisch haben wir als Familie ein weiteres Mal über die Besichtigungstermine unterhalten, die ich bekommen hatte. Morgen wollen wir nach [Stadtname] fahren und uns die drei Wohnungen anschauen, bis jetzt hat sich die Wohnungssuche nämlich schwierig gestaltet wegen der Pandemie, die mich als Ausländerin [Name des Landes] vor die Tatsache stellt, dass ich nur eine gewisse Zeit in Deutschland auf Durchreise sein darf. Ich habe mir daraufhin alles rausgesucht an Dokumenten, die ich mit zur Besichtigung nehmen wollte, habe noch alle Adressen notiert, und mir den Weg bis dorthin angeschaut, um den Durchblick zu erhalten. Am Nachmittag habe ich weiterhin auf den Immobilienseiten im Internet herumgestöbert, jedoch vergeblich, also habe ich auf Erfolg gehofft bezüglich der drei anstehenden Besichtigungen am Samstag.

Absatz	Tagebucheintrag
5/4	Das Buch, was ich für das Modul [fachfremder Zusatzkurs] bestellt hatte, wurde auch am heutigen Tag geliefert, sodass ich sofort die beiden ersten Kapitel darin gelesen habe, um besser mit der Übung 02 klarzukommen. Am Abend habe ich mir dann noch in Moodle angeschaut, ob sich während des Tages dort etwas getan hat, habe mir einen Wecker gestellt, um zeitig aufzustehen, und nach [Stadtname] zu fahren für die Besichtigungstermine, damit ich nach [Stadtname] ziehen kann für mein Studium.
6	Montag, den [Tag] [Monat] 2020
6/1	Mit diesem Montag, bricht für mich die zweite Woche meines Studiums in [Hauptfach] an. Um 9:00 Uhr hat mein Wecker geklingelt, und ich bin aufgestanden, habe mir einen Kaffee in der Küche gemacht und habe mich sofort an mein Laptop gesetzt, um mir das Video im Moodle zu meiner ersten Vorlesung anzuschauen, da ich am Wochenende keine Zeit gefunden hatte dafür. Es hat sich herausgestellt, dass dies nicht sonderlich die beste Idee war, weil ich mir so unnötig Stress bereitet habe.
6/2	In der Vorlesung haben wir kurz in Breakout-Session über die vergangene Woche und unsere Erfahrungen im Hinblick unserer Aufgabe der Autoethnographie gesprochen und darüber diskutiert, wie wir unsere Informationen festhalten, um sie nachher niederzuschreiben. Interessant war, dass wirklich jeder seine eigene Vorgehensweise hat, wir haben uns dann noch darüber unterhalten, dass es teilweise schwierig ist auf die Angegebene Wortzahl zu kommen da unser Alltag in Bezug auf das Studium sich einerseits einschränkt wegen der Pandemie, also dass es sich um ein Online-Semester handelt, aber auch dass es sich monoton anfühlt, da wir halt außerhalb nichts machen können wegen der Pandemie, damit ist das Treffen von Mitstudierenden gemeint, was ja im Normalfall zu dem Studentenalltag dazugehört.
6/3	Am Nachmittag habe ich dann den finalen Mietvertrag für meine erste Eigene Wohnung erhalten, was mir ermöglicht am kommenden Jahresanfang nach [Stadtname] zu ziehen und mein Studium dort weiterzuführen in Hoffnung auf ein normales und nicht digitales Semester an der [Universität] [Stadtname].
6/4	Dann habe ich mir ein Teil der Vorlesung des [fachfremden Zusatz-]Moduls angeschaut und Notizen dazu gemacht. Am Ende des Tages habe ich mir den Podcast zu einer anderen Vorlesung geschaut und war dann fertig mit allem bezüglich der Uni an diesem Tag. Ich habe noch meinen Wecker gestellt und mir den Stundenplan angeschaut und bin schlafen gegangen.
7	Dienstag, den [Tag] [Monat] 2020
7/1	Um 7:45 Uhr hat mein Wecker geklingelt, bin aufgestanden und habe mir einen Kaffee gemacht. Dann habe ich mich an den großen Tisch im Wohnzimmer gesetzt und mein Laptop aufgeklappt, um meine Mails zu checken und bei Moodle reinzuschauen. Unser Professor hatte entschieden die Vorlesung etwas später zu starten, erst um [Vormittag], sodass ich mir noch alles zurechtgelegt habe, den vorherigen Kurs nochmal flüchtig überschauen konnte, als wir dann pünktlich angefangen haben.
7/2	Nach der Vorlesung habe ich mich an den Mietvertrag meiner zukünftigen Wohnung gesetzt und mir diesen gründlich durchgelesen, bis ich dann unterzeichnet habe und ihn zur Post gebracht habe, damit dieser nach [Stadtname] zum Vermieter geschickt wird.
7/3	Am Mittagstisch ging es dann darum, was heute noch ansteht bezüglich der Uni und was ich schon vorbereiten kann für meine Kurse aber auch für meine zukünftige Wohnung.
7/4	Wie jeden Dienstag habe ich dann bis [früher Abend] keine Kurse, wir bekommen die Vorlesung für [Nebenfach] nämlich hochgeladen und können uns die individuell anschauen und so unseren Studienalltag ein bisschen selbst gestalten.

Absatz	Tagebucheintrag
7/5	Leider hatte das an diesem Tag nicht so gut funktioniert, da sie etwas zu spät online auf Moodle war, und man sie sich nicht mehr anschauen konnte vor dem Seminar, da die Vorlesung zu lange war. wir hatten auch schon rausgefunden, dass wir den Professor fragen können, ob er früher die Folien auf Moodle hochladen könnte, aber irgendwie hatte das nicht funktioniert. Unser Seminar um [früher Abend] wurde dann noch etwas früher beendet, weil es sich um eine Art Einführung handelte in die [Hauptfach].
7/6	So hatte ich noch einige Minuten, bis ich an einem anderen Zoom-Meeting teilgenommen habe, was von [der Studierendengruppe] aus [Stadtname] organisiert wurde. Es handelte sich hierbei um die Versteigerung der [Studienanfänger*innen], die so [in die Studierendengruppe] aufgenommen wurden, es war sehr spaßig und man hat viele neue Leute kennengelernt. Wegen den pandemischen Umständen lernt man in unserem ersten Studiensemester, was komplett online stattfindet, leider nicht so viele Leute kennen, deswegen war ich positiv davon überrascht, dass so viele Online-Treffen ermöglicht wurden, um doch einen bestmöglichen Anschluss zu finden. Diese Versteigerung ging auch etwas länger, sodass ich erst nach Mitternacht ins Bett kam und mir nur noch schnell den Wecker gestellt habe.
8	Mittwoch, den [Tag] [Monat] 2020
8/1	Um 7:45 Uhr hat mein Wecker geklingelt und ich habe auf schlummern gedrückt, was dazu geführt hat, dass ich 5 Minuten vor der Vorlesung wach wurde, also habe ich mich beeilt, mir schnell einen Kaffee gemacht und mich vor meinen Laptop gesetzt. Ich war noch rechtzeitig für die Vorlesung und habe nichts verpasst.
8/2	Ich hatte dann bis [Mittag] Vorlesung, am Mittagstisch wurde ich über den gestrigen Abend ausgefragt. Ich erzählte kurz, was ich machen musste […]. Des Weiteren haben wir uns noch um die Studienbeihilfe unterhalten, die jeder […] Student auf [Antrag] erhalten kann. Um diese jedoch anzufragen, benötige ich recht viele Dokumente, unter anderem auch den Mietvertrag, den ich erst diese Woche zur Post gebracht hatte und ihn noch nicht zurückerhalten habe.
8/3	Nach dem Mittagsessen hatte ich keine Vorlesungen oder Seminare mehr, somit hatte ich am Nachmittag frei und konnte Vor- und Nacharbeiten für die Uni leisten. Ich hatte mich dazu entschieden Texte durchzulesen und die [fachfremde Zusatzkurs-]Vorlesung, die auf Moodle hochgeladen war, zu Ende zu schauen. Das [fachfremde Zusatzkurs]-Modul hatte sich als größeres Hindernis und Herausforderung entpuppt als ich angenommen habe, immer wieder ertappe ich mich im Laufe des Tages, wie ich mir den Kopf über dieses Modul zerbreche, deswegen hatte ich mich zum Abend hin dazu entschieden meine Abgabepartnerin der Übungsgruppe darauf anzusprechen, sie teilte meine Meinung und hat erläutert, dass sie das rasante Zunehmen der Erwartungen in den Übungen etwas zu extrem finden würde.
8/4	Am Ende des Tages hatte ich genug von der Uni und hatte mich dazu entschieden, keinen Text mehr zu lesen, sodass ich mir nur noch eine To-Do-Liste geschrieben habe für Morgen und meinen Wecker gestellt habe, um Morgen zeitgemäß aufzustehen, um noch etwas geschafft zu bekommen.
9	Donnerstag, den [Tag] [Monat] 2020
9/1	Um 9:30 Uhr hat mein Wecker geklingelt und ich bin aufgestanden, um mir einen Kaffee zu machen. Ich habe mich an den Küchentisch gesetzt und mir die Nachrichten angeschaut und meine E-Mails gecheckt auf meinem Handy. Da ich meine To-Do-Liste gestern bereits geschrieben hatte, wusste ich genau, was heute ansteht. Also bin ich nach oben gegangen und habe mich an meinen Schreibtisch gesetzt, diesen etwas aufgeräumt, damit ich genug Platz hatte, um produktiv zu arbeiten und habe mich an mein Laptop gesetzt.

Absatz	Tagebucheintrag
9/2	Als erstes habe ich mir Texte durchgelesen und mich anhand des Buches zur [fachfremder Zusatzkurs] auf die Übungsstunde am Nachmittag vorbereitet. Irgendwie raubt dieses Modul die meiste Zeit und ich würde mich gerne mehr mit den anderen Modulen auseinandersetzen, auch bei der Nacharbeit der verschiedenen Module die zur [Hauptfach] ist mir aufgefallen, dass ich bereits viele von den angesprochenen Methoden etwas kenne, da ich [Name der Schule] [Hauptfach] […] hatte.
9/3	Am Nachmittag habe ich mir noch die fehlenden Texte ausgedruckt, da ich mir gerne alles unterstreiche was ich wichtig finde in einem Text, und ich kein Tablet besitze und die Lektüre auf einem Laptop in meinen Augen doch recht umständlich ist. Um [Nachmittag] hatten wir die Übungsstunde, und dieses Mal stand meine Matrikelnummer in der Liste der [zu Prüfenden], was mich etwas nervös gemacht hat, da ich dieses Mal etwas Schwierigkeiten bei der abzuleistenden Übung hatte. [Die Prüfung] verlief aber gut und ich habe es bestanden, ich wurde ein paar Verständnis-Fragen gefragt, die ich ohne Problem beantworten konnte, da er diese in der Stunde angesprochen hatte und ich mir zu den Erklärungen Notizen gemacht hatte.
9/4	Nach dieser Übung haben wir als Familie zusammen zu Abend gegessen, ich habe von dem bestandenen [Prüfung] berichtet und darüber, was diese Pluspunkte bedeuten im Hinblick auf die Klausur. Bevor ich schlafen ging, stellte ich mir keinen Wecker, da ich freitags keine Vorlesungen oder ähnliches an „Präsenz Veranstaltungen“ an der Uni habe.
10	Freitag, den [Tag] [Monat] 2020
10/1	Um 10:00 Uhr bin ich aufgestanden und habe mir einen Kaffee gemacht und etwas Kleines gegessen, bin dann nach oben gegangen und habe mein Zimmer einmal komplett aufgeräumt und mich dann an meinen Schreibtisch gesetzt, um Sachen für die Uni zu machen. Ich finde, dass man sich besser konzentrieren kann, wenn um einen Ordnung herrscht und nicht eine Unordnung, irgendwie kommt es mir dann strukturierter vor und ich komme mit dem, was ich erreichen will an dem Tag, besser voran.
10/2	Also habe ich mein Laptop aufgeklappt und bei Moodle reingeschaut, um zu sehen, was es dort Neues gibt, und mich bei der [Universität]-Mail eingeloggt, um zu sehen, welche Mails ich bekommen habe. Nachdem ich eine grobe Übersicht hatte, habe ich mich an die Vorbereitung für Montag gesetzt und mir angeschaut, was so ansteht die kommende Woche, schnell ist mir aufgefallen, dass ich noch Nacharbeit leisten muss, obwohl auch schon Vorarbeit zu den neuen Themen der kommenden Woche anstanden, somit hatte ich am Wochenende genug zu tun, wie es sich herausstellte.
10/3	Nach dem Mittagsessen habe ich mich an das auszufüllende Formular für die Studienbeihilfe gesetzt, um damit voran zu kommen, es stellte sich heraus, dass ich das Ganze auch online ausfüllen könnte und es mit meinem Token unterzeichnen kann. Also habe ich mich an das Formular gesetzt und alles ausgefüllt und hochgeladen bei den Teilen, wo ich keine Hilfe von meinen Eltern benötigt habe.
10/4	Im Verlauf des Nachmittags, haben meine Mutter und ich uns noch entschieden in nächst gelegene Stadt, [Stadtname], zu fahren und kleine Besorgungen zu tätigen, unter anderem waren auch verschiedene Sachen dabei für meine zukünftige Wohnung, die ich im Januar beziehen werde, wegen meines Studiums in [Stadtname].
10/5	Als wir wieder Zuhause waren habe ich nochmal bei Moodle reingeschaut und die neu dazu gekommenen Texte ausgedruckt und in meinem Ordner eingeordnet, damit ich alles komplett habe. Am Abend habe ich mir dann eine To-Do-Liste für die kommende Woche geschrieben und mir aufgelistet, welche Dokumente ich noch brauche, um die Studienbeihilfe anzufragen.

Quelle: Autoethnographie Online-Studienalltag im Wintersemester 2020/21

4.3 Vorgehen der induktiv-qualitativen Inhaltsanalyse

4.3.1 Übersicht gewinnen: Inhalte von Textdokument(en) zusammenfassen

Die Erfassung der manifesten Inhalte des empirischen Materials (Autoethnographie in Tabelle 4.1) sollte in der hier beschriebenen Auswertung stets von der Forschungsfrage „Wie gestalten Sie Ihren Studienalltag zu Beginn des Online-Semesters 2020/21?“ geleitet sein. Zur Nachvollziehbarkeit sollten Sie die Inhalts- und Themenübersicht in einer knappen schriftlichen Zusammenfassung festhalten. Damit wird die (bereits in der Schule erlernte Praktik) des Zusammenfassens zum ersten Auswertungsschritt (siehe Schritt 3 in Abbildung 4.1). Je nach eigenem Anspruch, persönlicher Arbeitsweise und von Dozent*innen (in Rolle Lehrende*r und/oder Prüfer*in) formulierten Anforderungen können Sie den untersuchten Text als Fließtext oder übersichtliche Liste zusammenfassen, wobei ein knapper Fließtext die Listenpunkte ergänzen könnte. Knapp zusammengefasst sind die zentralen manifesten Inhalte der Autoethnographie in Tabelle 4.1.

1. Ins Studium einfinden und Studienanforderungen einschätzen lernen (z. B. Ablauf und Vorbereitung von Veranstaltungen in unterschiedlichen Studienfächern, Online-Tools, wie Moodle, Zoom usw.).
2. Organisation des neuen Studienalltags (z. B. Aufstehen, Zeiten zwischen Studienveranstaltungen nutzen und To-Do-Listen).
3. Im digitalen Raum neue Leute kennenlernen (z. B. in Veranstaltungen, über Messenger und bei Studierendengruppe).
4. Leben zu Hause (z. B. Familienmitglieder und gemeinsames Essen).
5. Umzug an den Studienort (z. B. Wohnungssuche und Mietvertrag).
6. Studienfinanzierung (z. B. Studienbeihilfe und Bürokratie).
7. Autoethnographie (z. B. Austausch mit Mitstudierenden und Wörterzahl erreichen).

Die sieben Themen sind hoffentlich auch für Sie gut zu erkennen, sie sind jedoch nicht komplett überschneidungsfrei. Beispielsweise ist zu Thema 7 der Austausch mit den anderen Studierenden über die Autoethnographie in Absatz 6/2 sowohl aufgrund der Methode Autoethnographie methodisch interessant (Stichwort: Reflexivität der Forschung) als auch aufgrund des Kennenlernens unterschiedlicher Anforderungen im Studium (Thema 1). Für die folgenden Erklärungen, beginnend mit der induktiven Kodeentwicklung, wurde als Beispiel das Thema 3, im digitalen Raum neue Leute kennenlernen, gewählt. Aus den anderen Themen können Sie zu Übungszwecken selbst eines wählen und dies analog zur Beispielerklärung durcharbeiten.

4.3.2 Strukturierende Analyse der Textdaten

Das Textmaterial für die induktiv-qualitative Inhaltsanalyse kann sehr unterschiedlich strukturiert sein. Wie bei der Autoethnographie in Tabelle 4.1 kann es ein zusammenhängender Fließtext sein. Die Textdaten einer Micromessenger-Kommunikation sind hingegen in viele kurze Textschnipsel unterteilt, welche im Datenarrangement möglicherweise um Bilder, Icons usw. ergänzt werden. Transkripte von Gruppendiskussionen und Interviews weisen ganz klare Unterteilungen von Absätzen nach Sprecher*innenwechseln auf. In der Auswertungssoftware (z. B. MAXQDA) werden diese Absätze jeweils automatisch mit Absatzmarken versehen oder, falls von Ihnen so voreingestellt, mit Zeilennummern. Zeilennummern erlauben präzisere Verweise auf Textstellen, führen jedoch bei längeren Texten zu umfangreicheren Verweisen. Unabhängig, ob Sie Zeilen- oder Absatznummern verwenden, erfüllen beide die unerlässliche Funktion als Referenz für Verweise auf das empirische Material.

Die Strukturierung des Datenmaterials kann parallel zur Identifikation zentraler Themen im empirischen Textmaterial erfolgen. Sollten Sie jedoch eine sehr strukturierte, d. h. einen Arbeitsvorgang nach dem anderen abarbeitende Person sein, so können Sie selbstverständlich die Auswertungsschritte Themenidentifikation und strukturierende Datenordnung nacheinander durchführen. Für die Strukturierung des Datenmaterials sind grundsätzlich zwei Strukturierungsvorgänge zu unterscheiden.

1. Strukturierung als Gliederung des Textes und
2. Strukturierung des Textes durch Sinneinheiten.

Bei einem Interviewtranskript entfällt beispielsweise der erste Strukturierungsvorgang, da die Auswertungssoftware diese Arbeit bei der Vergabe von Absatz- oder Zeilennummern übernimmt. Für die Strukturierung des Interviewtextes nach Sinneinheiten können Sie sich an den Absätzen orientieren. Sie müssen jedoch davon ausgehen, dass eine Sinneinheit auch mitten in einem Absatz beginnen kann – erkennbar durch einen manifesten Wechsel des Themas –, einen ganzen weiteren Absatz dauern und dann nach dem ersten Satz im folgenden Absatz enden kann. Wie beispielsweise Miles und Huberman (1994, S. 56) betonen, sind Sinneinheiten zusammenhängende Inhalte, welche in einer durchgängigen Sequenz präsentiert werden. Das schließt nicht aus, dass an anderer Stelle im Text eine ähnliche oder ergänzende thematische Sinneinheit existiert.

Wie in Kapitel 4.2.3 für die Autoethnographie in Tabelle 4.1 dargelegt, können Sie bei einem Fließtext die Zusammenfassung der Inhalte durch die Identifikation von Sinneinheiten im Textmaterial vornehmen. Die Strukturierung des Textes können Sie idealerweise direkt in der von Ihnen verwendeten Auswertungs-

software durchführen oder vor dem Import in die Auswertungssoftware in einem Textverarbeitungsprogramm (z. B. OpenOffice Writer).

4.3.3 Induktive Kategorienentwicklung

Durch die Identifikation von strukturierenden Sinneinheiten und die Erfassung zentraler Inhalte sollten Sie Ihr Datenmaterial gut genug kennengelernt haben. In manchen Lehrbüchern wird auch davon gesprochen, dass Sie die Daten kennen oder nun *ein Gefühl für die Daten* haben (sollten). Geleitet von Ihren Kenntnissen der Daten können Sie nun die Analyse fortführen. Dies wird am Beispiel Thema 3, *im digitalen Raum neue Leute kennenlernen,* mitsamt den eingeklammerten Beispielen aus der Zusammenfassung erläutert, welche eine erste Suchanleitung für die Durchsicht des empirischen Materials darstellen.

Bitte machen Sie sich in Ihrem Forschungstagebuch (siehe Kurzdefinition in Box 4.2) Notizen zur Dokumentation von potenziellen Informationen, welche aus dem empirischen Material heraus als induktiver Kode geeignet sein könnten, und auch zu ersten Ideen zu Zusammenhängen, Bedeutungen und Interpretationsansätzen. Wir können Ihnen aus eigener Erfahrung versichern, dass Sie ohne Dokumentation im (digitalen und/oder analogen) Forschungstagebuch viele gute Ideen wieder vergessen, sofern Sie kein eidetisches Gedächtnis haben, da der Auswertungsprozess kognitiv sehr fordernd ist.

Box 4.2: Kurzdefinition Forschungstagebuch
Ein Forschungstagebuch erfüllt den Zweck, dass Sie gute Ideen für Ihre Forschung nicht vergessen (z. B. Kategorienentwicklung, Interpretation und theoretische Reflexion) und Transparenz im Sinne der Dokumentation Ihres Forschungsprozesses und dazugehöriger Entscheidungen schaffen (vgl. Gütekriterien Kapitel 2.3). Sie werden rasch merken, dass Sie mit der induktiv-qualitativen Inhaltsanalyse sehr viele kognitive Arbeiten gleichzeitig durchführen. Beispielsweise sind die Anonymisierung, Zusammenfassung und Strukturierung des empirischen Materials wichtige Analyseschritte und dienen nicht nur der Ordnung der Textdaten. Aus Erfahrung können wir Ihnen sagen, dass Sie sich sehr ärgern werden, wenn Ihnen eine offensichtliche Interpretationsidee nicht mehr einfällt und Sie sehr viel Zeit darauf verwenden, ein wichtiges Zitat in der gelesenen Literatur wieder zu finden – oder auch nicht wieder zu finden. Die Literaturorganisation und Lektüre sollten Sie selbstverständlich in einem Literaturverwaltungsprogramm (z. B. Zotero) organisieren. Abhängig von Ihrer bevorzugten Arbeitsweise können Sie ein Forschungstagebuch elektronisch (z. B. in Word) oder analog auf Papier (z. B. Büchlein) führen.

Bei der sorgfältigen Durchsicht des Textmaterials in Tabelle 4.1 könnten Sie folgende Textstellen für eine potenzielle Kodierung „soziale Kennenlernenaktivitäten" identifizieren.

- „E-Mails gecheckt" (Absatz 1/3), „ob ich eine E-Mail bekommen habe" (Absatz 2/1) „private sowie die [Universität]-Mail-Adresse" (Absatz 4/3) und „meine E-Mails durchhatte" (Absatz 5/2).
- „WhatsApp Gruppe" (Absatz 1/6).
- „[Studierendengruppe]" (Absätze 2/7 und 7/6).

- „Breakout-Session“ statt analogem „Treffen von Mitstudierenden gemeint, was ja im Normalfall [ohne Corona-Pandemie] zu dem Studentenalltag dazugehört“ (Absatz 6/2).
- „Meine Abgabepartnerin der Übungsgruppe“ (Absatz 8/3).

Die Möglichkeiten, Kontakte zu knüpfen und neue Leute kennenzulernen, sind aufgrund von Covid-19-Abstandsregelungen auf Textnachrichten, Anrufe und virtuelle Treffen beschränkt. Dennoch sehen wir, dass sich die*der Studierende an der elektronischen und online stattfindenden Kommunikation aktiv beteiligt. Der Austausch mit Mitstudierenden ist vor allem durch die Kontakte in und außerhalb der Veranstaltungen im Tagebucheintrag gekennzeichnet. Daher ist es möglich, diese Aktivitäten als Sozialkontakte zu kategorisieren. Sozialkontakte bilden somit die Überkategorie, der alle anderen Kategorien mitsamt Kodes zugeordnet werden (Tabelle 4.2).

Tabelle 4.2 Kategorien und dazugehörige Kodes am Beispiel Sozialkontakte (einfaches Schema)

Kategorie	Unterkategorien
Sozialkontakte	E-Mail-Kommunikation
	Messenger-Kommunikation (Handy)
	In der Veranstaltung
	Mit Veranstaltungsbezug (z. B. Lern-Tandem (bzw. -Gruppe))
	Studierendengruppe (= ohne Veranstaltungsbezug)

Wir sehen, dass sich die induktiv entwickelten Kodes sehr nah am Text orientieren. Die Unterkategorie „Mit Veranstaltungsbezug/Lern-Tandem (bzw. -Gruppe)“ ist eher durch Zufall so differenziert, dass klar wird, dass in der Umgangssprache von Lerngruppe gesprochen wird, auch wenn diese aus zwei Personen besteht. Das Beispiel zeigt, dass wir uns mit Blick auf unser Datenmaterial entscheiden können, welche Kategorien wir vergeben, wie feingliedrig diese ausfallen und mit welchen Methoden wir grundsätzlich vorgehen. Würden wir nur diese eine Autoethnographie einer induktiv-qualitativen Inhaltsanalyse unterziehen, so wäre Lern-Tandem zutreffender, da es nur die*der Studierende plus „Abgabepartnerin“ ist. Würden wir für eine Bachelorarbeit drei Textdokumente oder für eine Masterarbeit fünf Textdokumente auswerten, so könnten wir aufgrund des umgangssprachlichen Gebrauchs die Kategorie die Bezeichnung „Lern-Gruppe“ oder „Lerngruppe“ wählen.

Die Kategorisierung mit den Unterkategorien in Tabelle 4.2 reicht für die Identifikation von relevanten Textstellen und die Analyse von manifesten Inhalten und Mustern des Studienalltags aus. Ist das Textmaterial jedoch umfangreicher als in Tabelle 4.1, so ist es empfehlenswert, die induktiv entwickelten Kategorien stärker analytisch differenziert der Kategorie „Sozialkontakte" zuzuordnen. In Tabelle 4.3 werden dabei drei Unterkategorien verwendet, welche sich immer weiter verästeln. Ein Kategoriensystem, auch als Kategorienbaum bezeichnet, sollte gut nachvollziehbar sein. Gut nachvollziehbar bedeutet jedoch nicht, dass ein Kategoriensystem selbsterklärend ist. Mit Englischkenntnissen ist *face-to-face* als Kommunikation von Angesicht zu Angesicht zu entziffern. Jedoch ist *face-to-face* ein Fachbegriff im Soziolekt für persönliche Kommunikation vor Ort – an analogen wie digitalen Orten. Der Studienalltag ist durch *face-to-face*-Kommunikation in Veranstaltungen (z. B. Vorlesungen und Seminaren), und zwar ohne Veranstaltungsbezug, jedoch mit unmittelbarem Bezug zum Studium gekennzeichnet (z. B. in Studierendengruppe). *Face-to-face*-Kommunikation „mit Veranstaltungsbezug" kann Kommunikation unter Studierenden und mit Dozierenden beinhalten. Beispielsweise in einer Vorlesung, welche im Wintersemsemester 2020/21 per Zoom angeboten wurde, machte die*der Dozierende Gebrauch von Breakout-Sessions – im analogen Raum wären dies Gruppenarbeiten –, in denen die Studierenden sich austauschen oder etwas erarbeiten sollten.

Tabelle 4.3 Kategorien und dazugehörige Kodes am Beispiel Sozialkontakte (differenziertes Schema)

Kategorie	Unterkategorie 1	Unterkategorie 2	Unterkategorie 3
Sozialkontakte	Face-to-face	Mit Veranstaltungsbezug	Zoom
			Breakout-Sessions
			Lerngruppe
		Ohne Veranstaltungsbezug	Studierendengruppe
	Nachrichten	E-Mail	–
		Messenger (z. B. Signal)	–

Es ist nicht notwendig, dass für jede „Unterkategorie 2" mehr als eine „Unterkategorie 3" existiert. Die weitere Ausdifferenzierung wäre übertrieben detailliert, wenn jeweils für Unterkategorie 2 jeweils nur eine Unterkategorie 3 existiert. In diesem Fall gibt es nur die Ausprägung von Unterkategorie 1 und dazugehörige Unterkategorien 2, wie bei „Nachrichten" mit den Unterkategorien „E-Mail" und „Messenger" in Tabelle 4.3 zu sehen ist. Die jeweiligen Kategorienamen sollten

gut voneinander unterscheidbar und nicht schwer auseinanderzuhalten oder gar widersprüchlich sein. Die Wahl für die Unterkategorie 1 „Nachrichten" verlangte bereits eine große Reflexionsleistung, was durch die Eindeutigkeits- und Ausschlussprinzipien der Kategorien bedingt war. Ausgeschlossen wurde Textkommunikation, da sowohl per E-Mail als auch über Messenger ergänzend zu Text auch Bilder und/oder Audionachrichten versandt werden können. Weiter auszuschließen war die Verwendung von digital und online in den Kategorienamen, denn sowohl *face-to-face*-Kommunikation über Zoom als auch der Nachrichtenaustausch über Messengerdienste sind ebenfalls als Kommunikation über Social Media zu werten.[3]

4.3.4 Und das Ganze noch n-Mal von vorn

In den Kapiteln 4.3.1 (Übersicht gewinnen), 4.3.2 (Strukturierende Analyse) und 4.3.3 (Induktive Kategorien- und Kodeentwicklung) wurde das methodische Vorgehen am Beispiel einer Autoethnographie als empirisches Dokument vorgestellt. Werten Sie jedoch zwei und mehr Autoethnographien aus, müssen Sie die im Vorgehen angelegten Arbeitsschritte Übersicht gewinnen, strukturierende Analyse und induktive Kategorienentwicklung der induktiv-qualitativen Inhaltsanalyse ein-, zwei- bis n-Mal, gemäß der Anzahl auszuwertender Dokumente, wiederholen.

Das n der zu vergleichenden Fälle der induktiv-qualitativen Inhaltsanalyse müssen Sie selbstverständlich bei der Erstellung des Forschungsdesigns mit Blick auf die Forschungsfrage und unter Berücksichtigung des Datenumfangs festgelegt haben. Um die Vielfalt (z. B. Borchardt und Göthlich 2009) von sozialen Bedingungen des Studienbeginns im Wintersemester 2020/21 abzubilden, könnten wir aus dem Sample der 50 Autoethnographien ergänzend zum hier kodierten Beispiel der Autoethnographie der*des Studierenden_05 als Fall für den (relativ) normalen Studienstart bei Beeinträchtigung durch die Bedingungen der Corona-Pandemie wie folgt ergänzen: Autoethnographien von Studierenden, die Care-Aufgaben haben wie Kinderbetreuung, und eine Autoethnographie von Studierenden mit Nebenbeschäftigung (siehe Tabelle 6.6 in Kapitel 6 und Kapitel 9.5.2). Eine dritte für die qualitative Vielfalts-Stichprobe (Akremi 2019) gut nachvollziehbare Fallentscheidung ist die Auswahl einer Autoethnographie einer*eines Studierenden, die*der das Studium durch einen Nebenerwerb teilfinanzieren muss, und deshalb Überschneidungen von Arbeits- und Studienzeiten meistern muss.

---

3 Im Übrigen ist WhatsApp eine ganz üble Datenkrake mit wenig Datenschutz – eine sichere Alternative wäre der Anbieter Signal.

Sie sehen, dass durch die Fallauswahl neue induktive Kategorien und dazugehörige Kodes entwickelt werden (z. B. zu Nebenerwerb und Kindern), welche nicht in der Autoethnographie der*des Studierenden_05 vorkommen. Ebenso ist offensichtlich, dass aufgrund der Vielfalt der drei Fälle beispielsweise das Thema 2 bzw. die Kategorie „Studienorganisation" um weitere Unterkodes ergänzt werden (Kapitel 4.3.1). Wie in der Einleitung (Kapitel 4.1.1) dargelegt, möchten wir daher erneut dringend davon abraten, empirisches Material zu kodieren, bevor alle Kategorien induktiv auf Basis Ihres gesamten empirischen Materials erstellt sind.

4.3.5 Forschungspragmatische Entscheidungen zur Auswertung

Eine forschungspragmatische Entscheidung inhaltlicher Art müssen Sie teilweise jedoch auch während des Auswertungsprozesses treffen. Je nachdem, wie eng oder weit wir die Fragestellung „Wie gestalten Sie Ihren Studienalltag zu Beginn des Online-Semesters 2020/21?" beantworten möchten, wären auch die familiären Sozialkontakte mit in die Auswertung aufzunehmen. Die*der Studierende scheint intensive Sozialkontakte mit der Familie zu pflegen. Der offensichtlich enge Zusammenhalt ist in Tabelle 4.1 abgebildet durch die Sozialkontakte zu Mutter (Absätze 1/2 und 10/4), dem Geschwister (Absätze 3/1 und 4/2) und indirekt beim Mittagessen (Absätze 1/9, 3/3, 5/3 7/3 und 8/2). Bei einer weiten Beantwortung der Fragestellung müssten wir bedenken, dass auch ohne Einschränkungen durch die Corona-Pandemie Studierende teils zeitweise und teils dauerhaft bei der Familie wohnen. Folglich wäre der Entschluss aufgrund der Möglichkeit, online an den Veranstaltungen teilzunehmen und (vorerst) nicht an den Studienort zu ziehen als analog zur Pendeldistanz, zwischen Wohn- und Studienort zu analysieren.

Wir sehen, dass die Entscheidung der engen und weiten Auswertung auf inhaltlichen Gründen basieren kann, denn ein Studienstart muss als neuer Lebensabschnitt beispielsweise durch Sozialkontakte oder die Unterstützung der Familie auch verarbeitet werden. Die Entscheidung der engen oder der weiten Auswertung des Datenmaterials über induktiv entwickelte Kodes können wir nicht nur vom Erkenntnisinteresse abhängig machen, sondern auch durch die Relation von Analyseleistung und Umfang der schriftlichen Ausarbeitung. Wenn Sie eine Studienarbeit für ein Seminar schreiben, so gibt es in der Regel Vorgaben zum Umfang der schriftlichen Ausarbeitung (z. B. in Studien- und Prüfungsordnung und Handreichung des Studiengangs zu schriftlichen Ausarbeitungen). Bei Studienarbeiten im Bachelor können das 10 bis 12 Seiten, im Master 15 bis 20 Seiten und für eine Studienabschlussarbeit entsprechend mehr Seiten sein. Wenn Sie im Bachelor studieren und nur eingeschränkten Platz zur Präsentation Ihrer Analyse haben, so wäre eine enge Beantwortung der Fragestellung zu wählen, da Sie sonst die aufwendig durchgeführten manifesten und latenten Inhaltsanalysen

nicht präsentieren können, was sehr frustrierend ist. Bitte bedenken Sie bei der forschungspragmatischen Entscheidung für eine enge oder weite Beantwortung der Fragestellung auch, dass Sie nicht nur einen Text, sondern das Datenmaterial von drei oder sogar mehr Dokumenten verarbeiten wollen.

Den Rahmen der Einführung in die induktiv-qualitative Inhaltsanalyse würden ebenfalls die Erklärungen der induktiven Kodeentwicklung zu den anderen im empirischen Material gefundenen Themen sprengen. Zu Übungszwecken der induktiven Kodeentwicklung können Sie eines der oben in Kapitel 4.2.1 genannten anderen sechs Themen wählen. Bei der Übung ist es nicht ausgeschlossen, dass Sie ein weiteres, achtes, Thema im empirischen Material entdecken, welches bisher übersehen wurde. Ein Thema oder weitere Kategorien eines Themas im Textmaterial zu übersehen, kann bei einer induktiv-qualitativen Inhaltsanalyse nicht ausgeschlossen werden. Die*der Sozialforscher*in ist kognitiv stark gefordert im Auswertungsprozess und muss sehr viele empirische Informationen verarbeiten, zueinander in Beziehung setzen, ordnen und klassifizieren. Im Gegensatz zur deduktiv-qualitativen Inhaltsanalyse, welche nach einer theoretisch vordefinierten *Brille* das Datenmaterial kodiert (siehe Kapitel 5), sind Sie bei der induktiv-qualitativen Inhaltsanalyse näher an der Empirie dran. Besteht Ihre Empirie aus zwei und mehr Dokumenten, so gewinnen Sie Sicherheit durch den *cross-check,* also den Vergleich des empirischen Datenmaterials und daraus entwickelten induktiven Kategorien und dazugehörigen Kodes.

4.3.6 Kodieren des Datenmaterials

Bitte bedenken Sie, dass Sie sich zwar schon mitten in der empirischen Analyse befinden, dennoch die induktive Kategorienentwicklung und die darauf aufbauende Aktivität des Kodierens des Textmaterials der Vorbereitung der vertieften qualitativen Inhaltsanalyse dient. Für die qualitative Inhaltsanalyse von Sozialkontakten (= Kategorie) haben wir uns für eine enge Beantwortung der Forschungsfrage mit dem differenzierten Kodierschema (Tabelle 4.3) entschieden. Unabhängig davon, ob Sie Atlas.ti, MAXQDA, NVivo oder eine andere Analysesoftware verwenden, müssen Sie vor dem Kodieren erst das empirische Datenmaterial importieren und Kodes als Abbildung der entwickelten Kategorien anlegen. Dazu finden Sie im Internet über die Webseite des Anbieters oder eine Suchmaschine (z. B. datengeschützt über *https://duckduckgo.com*) für das entsprechende Analyseprogramm knappe Einführungen und weiterführende Tutorate.

Falls Sie dachten, dass Kodieren ein quasi-automatischer Vorgang ist und dass alle Entscheidungen bereits getroffen sind, so liegen Sie falsch. Nach der großen Entscheidung zu Forschungsdesign und der forschungspragmatischen Entscheidung für die enge Beantwortung der Forschungsfrage, müssen Sie sich

darauf festlegen, wie Sie kodieren und dann bei der Aktivität des Kodierens (fast) im Minutentakt Entscheidungen treffen. Die Aktivität des Kodierens für die induktiv-qualitative Inhaltsanalyse wird in der Regel manuell (= in Handarbeit) durchgeführt. Zwar könnten Sie Einzelworte wie „E-Mail" und „Zoom" von der Analysesoftware autokodieren lassen, das klappt aber bei den Unterkodes „ohne Veranstaltungsbezug" oder „Nachrichten" schon nicht mehr, da diese aus dem Kontext der Aussage erschlossen und deshalb nicht automatisch von der Analysesoftware erkannt werden. Bevor Sie mit der manuellen Kodierung beginnen können, müssen Sie entscheiden, wie (viel Text) Sie kodieren. Die Entscheidung, wie viel Text Sie kodieren, beschränkt sich primär auf folgende Möglichkeiten.

1. Kodierung von Einzelworten: Aus dem Beispiel des Kodierschemas in Tabelle 4.3 können Sie hierfür nur die Unterkodes 3 verwenden. Inhaltlich bedeutet diese Kodierung keinen Erkenntnismehrwert, denn wir haben bei der Zusammenfassung (Kapitel 4.3.1) der strukturierenden Analyse (Kapitel 4.3.2) und der induktiven Kategorien- und Kodeentwicklung (Kapitel 4.3.3) die qualitativen und manifesten inhaltlichen Bedeutungen von „Breakout-Sessions" oder „E-Mails" bereits erfasst. Dennoch kann die kleinteilige Kodierung mit Kodes identischen Einzelworten für eine vertiefte Auswertung sinnvoll sein, beispielsweise, wenn Sie vergleichend für drei oder fünf Autoethnographien von Studierenden die Regelmäßigkeit des Checkens und Schreibens von „E-Mails" als ein Mittel zur Pflege von Sozialkontakten analysieren wollen. Durch das Kodieren des Einzelwortes „E-Mail" (mitsamt möglicher alternativer Schreibweisen Email und eMail) können Sie in der Analysesoftware den Kode aktivieren (= MAXQDA-Befehl; siehe auch Kapitel 6.4.2) und die entsprechenden Stellen werden Ihnen in der Autoethnographie angezeigt. Je nach vorheriger Einstellung in der Analysesoftware werden dann zitierfähig die Zeilen- oder Absatznummern mit ausgegeben (siehe Erklärung in Kapitel 4.3.2 und Tabelle 4.1).

2. Kodierung von Sätzen: Ergänzend oder alternativ zur Einzelwortkodierung können Sie ganze Sätze kodieren. Im Gegensatz zur Einzelwortkodierung (Unterkategorie 3), können Sie mit der Kodierung von Sätzen auch die Unterkategorie 2 und/oder Unterkategorie 1 vergeben (siehe Tabelle 4.3). Das bedeutet auch, dass Sie beispielsweise einen Satz „In der Vorlesung haben wir kurz in Breakout-Session über die vergangene Woche und unsere Erfahrungen im Hinblick unserer Aufgabe der Autoethnographie gesprochen und darüber diskutiert, wie wir unsere Informationen festhalten, um sie nachher niederzuschreiben" (Absatz 6/2) sowohl als „*face-to-face*" (Unterkategorie 1), „mit Veranstaltungsbezug" (Unterkategorie 2) und „Breakout-Sessions" (Unterkategorie 3) kodieren können. Grundsätzlich können Sie einen Satz mehreren Kodes zuordnen, d. h. das Kodierschema definiert, wie häufig ein Satz als empirisches Material kodiert wird. Die durchschnittliche Satzlänge macht es jedoch relativ unwahrscheinlich, dass

für einen Satz alle drei Unterkategorien kodiert werden können – außer bei den durch die deutsche Grammatik ermöglichten halb- bis ganzseitigen Schachtelsätzen, welche jedoch in den Tagebucheinträgen wie auch in Transkripten von Interviews und Social Media-Kommunikation selten vorkommen.

3. Kodierung von Absätzen: Woran die Kodierung von ganzen Sätzen scheitert, wird durch die Kodierung von Absätzen ermöglicht. Im Gegensatz zu Einzelwort- und Satzkodierungen zielt das Kodieren von Absätzen auf die inhaltliche Erfassung von abstrakteren Bedeutungszusammenhängen im empirischen Material. Die Unterkodes definieren eine durch die Nummerierung klar erkennbare hierarchische Beziehung zueinander. Entsprechend stehen die Kodes auch für spezifische „Sozialkontakte" (Kategorie) im Studium, beispielsweise in „Breakout-Sessions" (Unterkategorie 3) oder die sehr allgemeine *„face-to-face-"* (Unterkategorie 1) Kommunikation. Die Kode-Hierarchie von konkret zu abstrakt verläuft also von „Breakout-Sessions" (Unterkategorie 3) über „mit Veranstaltungsbezug" (Unterkategorie 2) zu *„face-to-face"* (Unterkategorie 1).

4. Kodierung von Sinneinheiten: Unabhängig von grammatikalischen Regeln der Textgestaltung (z. B. Satz und Absatz) werden Sinneinheiten kodiert. Sinneinheiten können die Ebenen von der Kategorie (= höchste Aggregatsebene) bis zur letzten Ebene der Unterkategorien adressieren, wie unten für Unterkategorie 3 „Studierendenorganisation" (Tabelle 4.4) beispielhaft erklärt wird. Das Kodieren von Sinneinheiten zielt auf die Unterscheidung größerer Zusammenhänge in der Empirie, welche in Kapitel 4.3.1 grob durch sechs Themen benannt wurden (siehe auch Kapitel 4.3.2).

Die vier Möglichkeiten des Kodierens dürfen Sie nicht als Entweder-oder-Entscheidung verstehen. Jede Kodierungsart liefert Informationen in einem vordefinierten Umfang. Dieser reicht von einem Wort über einen Satz und ganzen Absatz bis hin zu einer Sinneinheit. Letztere können sich über Absätze hinweg erstrecken, sich jedoch auch in einem Satz erschöpfen. Durch die vorangegangene zusammenfassende und strukturierende Auswertung kennen Sie das Datenmaterial inzwischen sehr gut, sodass sie durch das Kodieren ein Ineinandergreifen von Breite und Tiefe des Erkenntnisgewinns ohne Nachkodieren einschätzen können – ja, Sie müssen weitere Entscheidungen treffen.

Weitere Ergebnisse für das manuelle Kodieren werden hier nicht präsentiert. Beispiele für Kodierungen finden Sie in diesem Buch im Kapitel 5 und im Kapitel 6.4.5 sowie den sehr gut gemachten und online verfügbaren Video-Tutorials der Auswertungssoftwareanbieterinnen.

4.4 Qualitative Inhaltsanalyse: manifeste und latente Inhalte verstehen und für Dritte verständlich machen

4.4.1 Erklären und Interpretieren als Teil der induktiv-qualitativen Inhaltsanalyse

Aufbauend auf der induktiven Entwicklung von Kategorien und dazugehörigen Kodes ermöglicht die induktiv-qualitative Inhaltsanalyse das vertiefte Verstehen bestimmter manifester und latenter Inhalte. Ein vertieftes Verstehen von in Sinneinheiten eingebetteten Inhalten ist insbesondere dann anspruchsvoll, wenn die Bedeutung von Informationen im empirischen Material nicht eindeutig ist. Nicht eindeutig heißt, dass die manifesten Inhalte einer oder mehrerer Sinneinheiten (Textausschnitt = empirischer Entdeckungszusammenhang) auf implizite, latente Strukturen hinweisen, welche als biographische, emotionale, normative und auf die soziale Situation bezogen Aussagen explizit gemacht werden sollen. Wie in Kapitel 2 dargelegt, zielt das vertiefte Verstehen darauf, tiefergehende Erkenntnisse zu den latenten Inhalten aus empirischen Daten zu heben (Schritt 4 in Abbildung 4.1).

Erklärungen und Interpretationen aus dem empirischen Material herauszuarbeiten, können Sie sich als Dreiteiler[4] des Verstehens vorstellen.

1. *Verstehende Analyse (Reformulierung der manifesten Inhalte in eigenen Worten):* Der erste Teil mag Ihnen als nicht unbedingt notwendig und redundant zum Originaltext erscheinen. Sie sichern sich damit jedoch dagegen ab, dass Ihr Verstehen des Originaltextes nur im Kopf plausibel klingt, und werden feststellen, dass Sie beim Schreiben schon viel stärker die manifesten Inhalte auswerten. Die verstehende Analyse können Sie je nach Präferenz und Ihrem Arbeitsstil entsprechend als Fließtext oder nach identifizierten manifesten Inhalten nummeriert erfassen (siehe Tabelle 4.6). Je nach *Gehalt* des Originaltextes kann die Reformulierung auch länger als der Originaltext sein.
2. *Deskriptive Analyse (Identifikation des Sinns/der Bedeutung in eigenen Worten):* Für die deskriptive Analyse sollten Sie den Text (wieder) stärker zusammenfassen und Ihre ersten Erkenntnisse bündeln. Im Vergleich zum Nacherzählcharakter der verstehenden Analyse sollten Sie für die Deskription einen Schreibstil wählen, wie Sie eine Beschreibung für Dritte in Ihrer Studienarbeit, Studienabschluss- oder Doktorarbeit anfertigen würden.
3. *Diskussion der manifesten und latenten Inhalte (Muster benennen, Kernaussage(n) erklären und interpretieren sowie theoriegeleitete Reflexion):* Der letzte Teil des Verstehensdreiteilers ist der umfänglichste und anspruchsvollste

4 Ähnliche Auswertungsmuster werden auch für andere qualitative Auswertungstechniken angeboten (z. B. Bohnsack et al. 2013).

Formulierungsschritt. Für die Diskussion müssen Sie wörtlich die manifesten und latenten Inhalte besprechen und alternative Erklärungen argumentieren, Deutungsvorschläge machen und den Leser*innen Interpretationsangebote für ein nachvollziehbares Verstehen des empirischen Materials und seiner Bedeutung liefern. Die für die Erklärung, Interpretation und Reflexion der Empirie hinzugezogene theoretische Herangehensweise sollte für das empirische Datenmaterial insgesamt und nicht nur für eine Sinneinheit geeignet sein.

Der Verstehensdreiteiler kann als Vorgehensanleitung schematisch wie in Tabelle 4.4. illustriert werden.

Tabelle 4.4 Schematische Darstellung des Verstehens von Sinneinheiten

Originaltext	Teil 1: Verstehende Analyse (Reformulierung der manifesten Inhalte in eigenen Worten)	Teil 2: Deskriptive Analyse (Identifikation des Sinns/der Bedeutung in eigenen Worten)	Teil 3: Diskussion der manifesten und latenten Inhalte (Muster benennen, Kernaussage(n) Erklären und Interpretieren sowie theoriegeleitete Reflexion)
Abschnitt 1			
Abschnitt 2			
Abschnitt n			

4.4.2 Systematisches Verstehen durch Interpretation manifester und latenter Inhalte

Als Beispiel für eine qualitative Inhaltsanalyse wird die der induktiv erstellten Kategorie *Studierendenorganisationen* (Kategorie *Sozialkontakte*) zugeordnete Sinneinheit 7/6 aus Tabelle 4.1 verwendet. Bereits in Abschnitt 2/7 hat die*der Studierende berichtet, dass sie*er eine Willkommensveranstaltung von Studierenden aus ihrem*seinem Heimatland online besucht hat. Das Beispiel wurde auch gewählt, um zu zeigen, dass Sie als Forscher*in trotz Anonymisierung die Informationen nutzen können, die von Autoethnographierenden, Interviewten und anderen Informant*innen bereitgestellt wurden. Verfügen Sie über diese Informationen, sollten Sie diese daher bei der Auswertung (Zusammenfassung, Interpretation usw.) nutzen, jedoch stets darauf achten, dass Sie die vorgenommene Anonymisierung im Text nicht aufheben. Wichtig ist dabei, dass Sie nach Abschluss des Schreibprozesses der Ergebnisdarstellung nochmals kritisch mit Fokus auf die Anonymisierung über den Text gehen. Das kann zur Folge haben, dass Sie interessante und sorgsam ausformulierte Erkenntnisse nachträglich lö-

schen müssen, denn der Schutz der Informant*innen geht gemäß den ethischen Leitlinien empirischer Sozialforschung vor (siehe Kapitel 3).

Um latente Muster zu heben und damit die Interpretation der manifesten und latenten Inhalte des Datenmaterials zu ermöglichen, ist es für die qualitative Inhaltsanalyse wichtig, dass Sie

a) sich an das Textmaterial herantrauen und die empirischen Informationen verarbeiten,
b) den Bezug zum Originaltext (= Empirie) stets gut erkennbar machen (um zu vermeiden, dass Dritte den Eindruck haben, Sie würden sich Ergebnisse *ausdenken,* damit die Analyse interessanter wird),
c) einer zwar statisch anmutenden, jedoch schrittweisen und damit systematischen Vorgehensweise für die vertiefende Inhaltsanalyse folgen,
d) sowohl für sich als auch für Dritte (z. B. Lehrende und Betreuer*innen von Abschlussarbeiten) gut dokumentiert und damit nachvollziehbar machen, wie Sie zu Erklärungen, Interpretationen usw. gelangt sind.

4.4.3 Beispiele für Erklären, Interpretieren und theoriegeleitete Reflexion

Eine weitere Unterteilung des Textes in Sinneinheiten kann ihnen einerseits erleichtern, sich an das Datenmaterial heranzutrauen und andererseits die Textdaten aufzubrechen, d. h. in verschiedene Sinneinheiten zu untergliedern. Wie diese Untergliederung dabei vonstattengeht, hängt von der Komplexität der Inhalte ab. Ziel einer weiteren Unterteilung der Sinneinheiten ist dabei stets, die verstehende Analyse zu erleichtern. Exemplarisch haben wir für das Beispiel Sinneinheit 7/6 eine Unterteilung nach Sätzen vorgenommen (Tabelle 4.5, nächste Seite). Wäre die Sinneinheit eine längere Textsequenz (z. B. mehrere Sätze) gewesen, so hätte eine weniger kleinteilige Unterteilung der Sinneinheit sinnvoll sein können. Für die Unterteilung sollte stets das eigene Verstehen-Können der Empirie leitend sein.

Im Beispiel 7/6 ist der erste Satz (= Abschnitt 7/6/1) ein Überleitungssatz, der den Namen der Studierendengruppe und die Information, dass die Veranstaltung online stattfand, als manifeste Informationen enthält. Inhaltlich könnte Ihre Neugier als empirische Sozialforscher*in durch Satz 2 geweckt sein, welcher die Information der nicht weiter spezifizierten – und für Sie womöglich ungewöhnlichen – Kennenlernaktivität „Versteigerung der [Studienanfänger*innen]“ enthält. Diese Kennenlernaktivität hat der*dem Studierenden Spaß gemacht, da sie*er bis „nach Mitternacht“ (Abschnitt 7/6/4) dabei mitgemacht hat. Nach Mitternacht ist eine unbestimmte, jedoch manifeste Information, da wir wissen, dass die*der Studierende sonst gegen 23:00 Uhr (Abschnitte 1/12 und 2/8, Tabelle 4.1) oder

Tabelle 4.5 Unterteilung Sinneinheit zum Oberkode „Sozialkontakte“ und zum Unterkode 3 „Studierendengruppe“ (Absatz 7/6; Tabelle 4.1)

Absatz	Tagebucheintrag
7/6/1	So hatte ich noch einige Minuten, bis ich an einem anderen Zoom-Meeting teilgenommen habe was von den [Name der Studierendengruppe] aus [Stadtname] organisiert wurde.
7/6/2	Es handelte sich hierbei um die Versteigerung der [Studienanfänger*innen], die so im [Name der Studierendengruppe] aufgenommen wurden, es war sehr spaßig und man hat viele neue Leute kennengelernt.
7/6/3	Wegen den pandemischen Umständen lernt man in unserem ersten Studiensemester, was komplett online stattfindet, leider nicht so viele Leute kennen, deswegen war ich positiv davon überrascht, dass so viele Online-Treffen ermöglicht wurden, um doch einen bestmöglichen Anschluss zu finden.
7/6/4	Diese Versteigerung ging auch etwas länger, sodass ich erst nach Mitternacht ins Bett kam und mir nur noch schnell den Wecker gestellt habe.

22:30 Uhr (3/6) zu Bett geht. Satz 3 ist eine in die Kennenlernaktivität eingeschobene Information bzw. Reflexion der*des Studierenden, dass die Covid-19-Schutzvorkehrungen das übliche bzw. erwartete Kennenlernen von Personen im neuen Lebensabschnitt Studium stark erschwert, jedoch nicht unmöglich macht. Zudem können wir Abschnitt 7/6/3 entnehmen, dass die*der Studierende erfreut war, dass mehr Personen an der Onlineveranstaltung teilgenommen haben, als von ihr*ihm erwartet.

Weitere manifeste Inhalte der Sinneinheit 7/6 sind, dass die*der Studierende aktiv Kontakt zu anderen Studierenden sucht – sowohl zu Individuen als auch über Kollektive. Die Studierendengruppe ist dabei das Kollektiv, zu dem die*der Studierende qua Staatsangehörigkeit Zugang hat. Folglich ist die Studierendengruppe eine Option der Kontaktermöglichung. Wie wir oben in der Zusammenfassung (Kapitel 4.3.1) und induktiven Kodeentwicklung (Kapitel 4.3.2) schon festgehalten haben, ist die*der Studierende dabei, aktiv in das Leben als Studierende*r am Studienort einzutauchen. Daher wäre es beispielsweise zu weit hergeholt, die Staatsangehörigkeit als Exklusionskriterium für andere Studierende hervorzuheben oder gar weiterführend zu interpretieren. Dies widerspricht dem oben induktiv identifizierten Muster (Kategorie „Sozialkontakte“) und würde über den Text hinausgehen. Der Text stellt jedoch bei der hier angelegten induktiv-qualitativen Inhaltsanalyse den Bezugsrahmen dar. Manifest ist auch, dass sie*er zum zweiten Mal die Studierendengruppe besucht (Abschnitte 2/7 und 7/6). Daraus können wir schließen, dass sie*er das Kennenlernen positiv wahrgenommen hat und deshalb wiedergekommen ist. Beim Wiederkommen wurde die*der Studierende aufgenommen, was auf ein längerfristiges Engagement hindeutet, und hat sich durch Teilnahme am Ritual der Versteigerung potenziell längerfristig an das Kollektiv gebunden. Die*der Studierende schreibt hier nur, dass

sie*er in der Studierendengruppe „viele neue Leute kennengelernt“ (Satz 7/6/2) hat, wobei unklar bleibt, a) wie die Versteigerung erfolgt ist, und b) ob dadurch ein Mitglied der Studierendengruppe zum Bezugspunkt in der und als Bindeglied in die Studierendengruppe bestimmt wurde.

Während Ihr Verstehen der manifesten Inhalte der Sinneinheit methodisch auf der Textebene anzusiedeln ist, zielt Ihr Verstehen der latenten Inhalte hinter die manifeste Textebene und explizite Bedeutung der einzelnen Worte. Zum Überwinden der Verstehensgrenze der „Versteigerung von Studienanfänger*innen“ müssen wir den latenten Inhalt durch plausible Erklärungen und durch das empirische Material, dessen Entstehungskontext und die von induktiven Kodierungen gestützten Interpretationen heben und dadurch verständlich machen. Erklärungen sollten plausibel sein und Interpretationen sollten Angebote sein für das Verstehen durch Dritte (z. B. Lehrende und Leser*innen). Kern Ihrer Aufgabe des Erklärens und Interpretierens ist folglich, die latenten Inhalte für Dritte verständlich zu machen. In der Regel bietet empirisches Material mehr als nur eine mögliche Erklärung oder Interpretation an. Hier sollten sie weder kognitiv faul und mit einer Erklärung und/oder Interpretation zufrieden sein noch sollten Sie den Anspruch an sich haben, alles Erdenkliche erklären und interpretieren zu müssen. Sie sichern sich jedoch ab und weisen Ihre Anstrengung, das empirische Material zu durchdringen, nach, wenn Sie zwei bis vier Erklärungen und Interpretationen diskutieren können und darauf aufbauend eine eigenständige Argumentation entwickeln können.

Um sich dem Verstehen der latenten Inhalte von „Versteigerung von Studienanfänger*innen“ über Erklären und Interpretieren zu nähern, bietet der Text keine weiterführenden Informationen. Anhaltspunkte sind die manifesten Informationen zur Versteigerung als Ritual, um in die Studierendengruppe aufgenommen zu werden, das Ziel „Leute kennengelernt“ und „sehr spaßig“ als Bewertung des Aufnahmerituals. Ihr Alltagsverstehen bietet erste Anhaltspunkte zur Erschließung der sozialen Bedeutung von Aufnahmeritualen, beispielsweise, dass Aufnahmerituale symbolische Handlungen darstellen, und als solche nach regelgeleiteten und festlichen Abläufen verfahren. Vorgegebene Regeln von Versteigerungen unterscheiden zumindest die Rollen Auktionator*in, Anbieter*in eines Gutes und mehr als zwei das angebotene Gut Nachfragende. Beispielsweise bei der Versteigerung eines Bildes eröffnet die*der Auktionator*in mit dem von der*dem Anbieter*in festgelegten Mindestpreis. Der darauffolgende formelle Ablauf beinhaltet, dass die Nachfrager*innen des Bildes ihre Gebote vorbringen, solange bis kein höheres Gebot erfolgt. Formal ist die Versteigerung mit dem symbolischen Hammerschlag und den Worten zum Ersten, zum Zweiten und zum Dritten beendet. Die Handlung der Auktion ist abgeschlossen, wenn die*der Höchstbietende das Geld gegen das Bild tauscht.

Beim Aufnahmeritual der Versteigerung von Studienanfänger*innen ist die*der Studierende klar in der Rolle des zu ersteigernden Subjektes. Es ist plau-

sibel, anzunehmen, dass sie*er sich vorstellt, um ein Bild von sich zu vermitteln und sich gegebenenfalls durch Nennung von Können, Interessen usw. anpreist. Da die Versteigerung der*dem Studierenden Spaß gemacht hat, könnte es auch sein, dass sie*er wie auch die anderen Studienanfänger *innen etwas Lustiges, erträglich Peinliches usw. in der Vorstellung beisteuern mussten oder eine Aufgabe (z. B. Darstellung bzw. Vorführung von …) lösen mussten. Die Rolle der Nachfrager*innen wird vermutlich von Mitgliedern der Studierendengruppe eingenommen, welche die Studienanfänger*innen (im Plural) ersteigern. In diesem Zusammenhang könnte die Interpretation von „viele neue Leute kennengelernt" lauten, dass auch die Mitglieder sich angepriesen haben. Der Sozialfigur studentische*r Tutor*in folgend, könnten individuelle Betreuungsangebote aufgrund von gemeinsamem Studienfach sowie die Universität und den Ort kennenlernen die geldwerten analog gehandhabten qualitativen Angebote bestimmt haben. Den „Preis weiter hochgetrieben" für die Ersteigerung von Studienanfänger*innen würde dieser Interpretation weiterfolgend bedeuten, dass ein Abgleich von Interessen, Können, Aufgabenlösungskompetenzen usw. von Bietenden und Angebotssubjekt den Bieter*innenwettstreit der Nachfrager*innen entschieden hat. Aus der Sinnhaftigkeit des Aufnahmerituals ist mit relativ hoher Wahrscheinlichkeit der Schluss zu ziehen, dass die*der Auktionator*in im feierlichen Versteigerungsritual und die symbolisch gehaltvolle Aufnahme in die Studierendengruppe durch ein Mitglied mit formaler Rolle (z. B. Vorsitzende*r) vollzogen wurde.

Wie bereits mehrfach in diesem Kapitel betont, ist induktiv-qualitative Sozialforschung keine theoriefreie Forschung. Im Gegensatz zur deduktiv-qualitativen Inhaltsanalyse ist die Theorie aber nicht von Beginn an untersuchungsleitend (siehe Kapitel 5), sondern das empirische Material und daraus generierte analytische Kategorien sind der Theorie vorangestellt. Entsprechend kann bzw. sollte zur Unterstützung der Analyse latenter Inhalte auf sozialwissenschaftliche Theorie(n) zurückgegriffen werden. Zu Symbolen und Ritualen bietet sich beispielsweise der Theorieklassiker von Geertz (2002) zum Verstehen kultureller Systeme und dazugehöriger Praktiken sowie darauf aufbauende Studien an. Das Kulturverständnis von Geertz (2002) betrachtet das Konstruierende, d. h. die Rekonstruktion, Modifikation und Neukonstruktion kultureller Praktiken und dazugehöriger Sozialstrukturen in Interaktion.

Selbst wenn wir nicht wissen, wie die Versteigerung von Studienanfänger*innen in der sozialen Realität abgelaufen ist, so hilft die theoretische Herangehensweise von Geertz (2002) die im vorigen Absatz angebotenen Analogien (mit der Versteigerung eines Bildes), Erklärungen, Interpretationen und gezogene Schlüsse, den sozialen Kode und die symbolische Bedeutung des „spaßigen" Aufnahmerituals soziologisch zu erfassen. Die*der Studierende ermöglicht uns keine weiteren Deutungen des Aufnahmerituals, jedoch konstruiert der Tagebucheintrag eine von verschiedenen Möglichkeiten, im digitalen Raum Sozialkontakte zu knüpfen. Die implizit mitgelieferte Deutung ist, dass Studierende sich auf den

digitalen Raum einlassen wollen müssen, was selbstverständlich eine hier vorgenommene Interpretation ist, da sie*er dies nicht explizit sagt, sondern „Leute kennenlernen" trotz den Einschränkungen von Onlinetreffen reflektiert.

Der Fokus der*des Studierenden auf „Sozialkontakte knüpfen" kann zwar nicht überdecken, dass eine Versteigerung durch das (soziale) Einpreisen auch eine Praxis des Bewertens ist, wie der Vergleich zur Versteigerung eines Bildes deutlich machte. Gemäß der Theorie der Soziologie des Bewertens (z. B. Beiträge in Nicolae et al. 2019) wird ein Gegenstand bzw. im Falle der Versteigerung von Studienanfänger*innen, ein Subjekt bewertet. Das Bewerten ist dabei als Evaluation und In-Wert-Setzen zu verstehen (z. B. Vatin 2013). Da die Versteigerung von Studienanfänger*innen jedoch eine Versteigerung von mehreren Subjekten bedeutet, könnte ein Wertvergleich entstehen, falls ein*e Studierende*r einen höheren Versteigerungswert erzielt als ein*e andere Studierende*r. Da die*der Studierende den Spaß der Versteigerung und das Länger-Aufbleiben, um die Leute besser kennenlernen zu können, betont, scheint das Moment des Bewertens einer Versteigerung zumindest für sie*ihn nicht negativ oder wahrnehmbar und damit in der Autoethnographie erwähnenswert gewesen zu sein. Folglich würden Sie mit weiterführenden Analysen zur sozialen Bedeutung von Bewerten den für induktiv-qualitative Inhaltanalysen wichtigen Bezugsrahmen der Daten verlassen, und keine Antwort zu der hier im Fokus stehenden Fragestellung „Wie gestalten Sie Ihren Studienalltag zu Beginn des Online-Semesters 2020/21?" leisten. Für die sozialwissenschaftliche Interpretation war die Annäherung an die Versteigerung der Studienanfänger*innen unter Zuhilfenahme der Bildversteigerung von Wert, da die soziale Situation, die verschiedenen Rollen und der Ablauf mitsamt symbolischer Bedeutung von Versteigerungen definiert werden konnten. Einerseits war das Gedankenexperiment eine Hilfestellung für unsere Interpretation und bot mögliche Erklärungen der nicht alltäglichen und im Datenmaterial undefinierten Situation „Versteigerung von Studienanfänger*innen". Andererseits kann eine bekannte Analogie auch Leser*innen verständlich machen, wieso Sie bestimmte Begriffe verwenden und welche Beziehungen Sie aufarbeiten – die Welt ist nicht voller Soziolog*innen, die sofort bei Interaktion das Analyseset Rollenkonstellationen, Handlungen, symbolische Praktiken usw. aktivieren können.

Sie sehen, dass Erklärungs- und Interpretationsangebote zur Näherung an die Bedeutung von latenten Inhalten ein methodisches Vorgehen erfordern, welches zugleich anspruchsvoll ist und intuitiv entstehen kann. Intuitiv ist die Näherung an „Versteigerung der Studienanfänger*innen" beispielsweise über das Alltagsverstehen und Aufschlüsselung der möglichen Bedeutungen über die Versteigerungsanalogie eines Bildes. Anspruchsvoll ist das Erschließen latenter Inhalte, da Sie sich in die Bedeutung von Sinnzusammenhängen *hineindenken* müssen und Deutungen *wagen* müssen, sowie die Hinzunahme von sozialwissenschaftlicher Theorie.

Eine Vorgehensweise, Ihr Hineindenken sowie Erklärungs- und Interpreta-

Tabelle 4.6 Beispiel Abschnitt 7/6 für das vertiefte Verstehen von Sinneinheiten

Originaltext	Teil 1: Verstehende Analyse (Reformulierung der manifesten Inhalte in eigenen Worten)	Teil 2: Deskriptive Analyse (Identifikation des Sinns/der Bedeutung in eigenen Worten)	Teil 3: Diskussion der manifesten und latenten Inhalte (Muster benennen, Kernaussage(n) Erklären und Interpretieren sowie theoriegeleitete Reflexion)
So hatte ich noch einige Minuten, bis ich an einem anderen Zoom-Meeting teilgenommen habe, was von den [Name der Studierendengruppe] aus [Stadtname] organisiert wurde. Es handelte sich hierbei um die Versteigerung der [Studienanfänger*innen], die so im [Name der Studierendengruppe] aufgenommen wurden, es war sehr spaßig und man hat viele neue Leute kennengelernt. Wegen den pandemischen Umständen lernt man in unserem ersten Studiensemester, was komplett online stattfindet, leider nicht so viele Leute kennen, deswegen war ich positiv davon überrascht, dass so viele Online-Treffen ermöglicht wurden, um doch einen bestmöglichen Anschluss zu finden. Diese Versteigerung ging auch etwas länger, sodass ich erst nach Mitternacht ins Bett kam und mir nur noch schnell den Wecker gestellt habe.	Die*der Studierende kann einige Minuten verschnaufen, bevor sie*er am Zoom-Treffen der Studierendengruppe teilnimmt. Teil des Aufnahmerituals in die Studierendengruppe ist die sogenannte Versteigerung von Studienanfänger*innen. Die Versteigerung verfolgt den Zweck des Kennenlernens anderer Studierende*r und hat der*dem Studierenden viel Spaß gemacht. Der Spaßcharakter wird unterstrichen durch die abschließende Bemerkung, dass die*der Studierende erst nach Mitternacht ins Bett ging. Die*der Studierende ist erfreut darüber, dass auch (viele) andere Studierende die – im Gegensatz zu analogen Treffen – als mühsam und hinderlich empfundene Onlinekommunikation auf sich nehmen, um neue Leute kennenzulernen.	Für die*den Studierenden hat neue Leute kennenlernen eine hohe Priorität bzw. die*der Studierende erscheint sehr kontaktfreudig und neugierig auf „neue Leute“. Das Kennenlernen scheint für sie*ihn Teil des Ankommens am Studienort zu sein. Ergänzend zur individuellen Kontaktaufnahme nimmt sie*er auch von Studierenden organisierte Angebote wahr, um „neue Leute“ kennenzulernen. Sie*er relativiert die Beschränkungen der Corona-Pandemie bedingten Einschränkungen und betont, dass dennoch Kennenlernen samt Kennenlernspielen online möglich ist, und sogar Spaß machen kann. Beispiel ist das Kennenlernspiel „Versteigerung der Studienanfänger*innen“, welches gleichzeitig das Aufnahmeritual der Studierendengruppe bedeutet.	[Die vorigen Erklärungen, Interpretation und theoriegeleiteten Reflexionen zum Aufnahmeritual mit dem Ziel Leute kennenlernen können hier aus Platzgründen nicht wiederholt werden]

tionsmöglichkeiten voranzutreiben und gleichzeitig transparent zu dokumentieren, wurde als Schema in Tabelle 4.4 vorgeschlagen. Dieses Schema zum Vorgehen bei der qualitativen Inhaltsanalyse ist ein Vorschlag, den Sie je nach Bedarf und gewonnener Sicherheit im Analyseprozess auch anpassen können. Aufgrund der Orientierung am Papierformat könnte der OpenOffice Writer und Microsoft Word nicht genügend Platz bieten für die Dokumentation der vertiefenden Inhaltsanalyse (siehe vierte Spalte in Tabelle 4.6). Daher könnten Sie den OpenOffice Calculator oder Microsoft Excel verwenden, welche die flexible Einstellung der Anzahl an Tabellenspalten und -zeilen sowie der jeweiligen Spaltenbreite ermöglichen.

In das Schema aus Tabelle 4.4 wird in Tabelle 4.6 zur Erklärung wieder der Absatz 7/6 aus Tabelle 4.1 mit den zwei Unterkategorien 3 („Studierendengruppe" und „Zoom") und Unterkategorie 2 („ohne Veranstaltungsbezug") eingefügt. Wie Sie sehen, sind die Erklärungen und Interpretationen nicht auf die Unterkategorien 2 und 3 begrenzt, sondern berücksichtigen auch die Perspektive der *face-to-face-* (Unterkategorie 1) Sozialkontakte (Kategorie) im Online-Semester 2020/21.

5. Deduktiv-qualitative Inhaltsanalyse

Wie in Kapitel 2 beschrieben, unterscheiden wir bei der qualitativen Inhaltsanalyse zwei Varianten: die induktive und die deduktive Inhaltsanalyse. In diesem Kapitel wird nun die deduktiv-qualitative Inhaltsanalyse beschrieben. Dazu stellen wir Ihnen zunächst wieder ein Ablaufschema vor und zeigen dann an einem konkreten Beispiel, wie die deduktiv-qualitative Inhaltsanalyse durchgeführt werden kann.

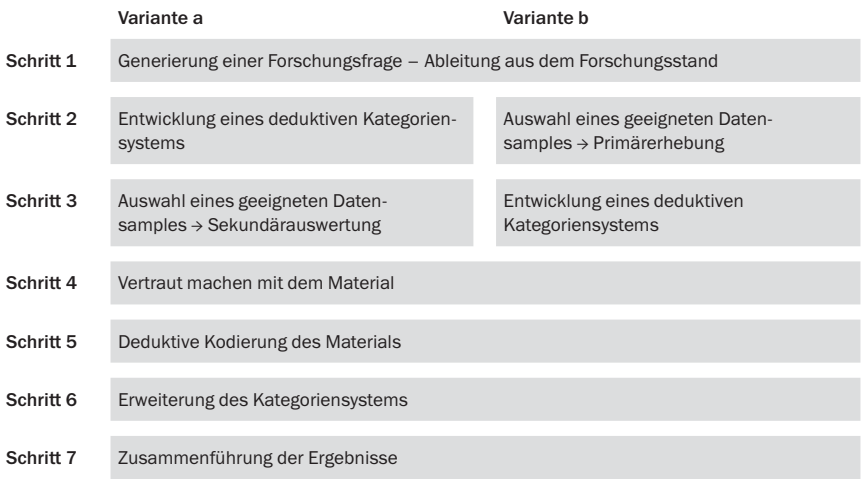
5.1 Ablaufschema der deduktiv-qualitativen Inhaltsanalyse

Die deduktiv-qualitative Inhaltsanalyse ist eine Methode, die Datenmaterial (z. B. Interviews, Gruppendiskussionen, Zeitungsartikel oder Tweets) anhand eines deduktiv erstellten Kategoriensystems analysiert. Deduktion bezeichnet die Herleitung von empirisch überprüfbaren Aussagen (Hypothesen) aus einer Theorie oder empirischen Studien heraus. Die deduktiven Kategorien werden dabei aus einer oder mehreren Theorien sowie empirischen Studien abgeleitet. Das heißt, dass die Kategorien an das Datenmaterial herangetragen werden und nicht zunächst aus dem Datenmaterial entstehen. Angewandt wird die deduktiv-qualitative Inhaltsanalyse entsprechend dann, wenn ein klarer Fokus der Analyse vorliegt und Kategorien aus Theorien und/oder empirischen Studien operationalisiert werden können, die auf Vorarbeiten zu den zu untersuchenden Phänomenen fußen (Elo und Kyngäs 2008, S. 109). Entsprechend ist es bei der deduktiv-qualitativen Inhaltsanalyse möglich, Hypothesen zu generieren und diese anhand des Materials auch zu testen.

Der Beginn jeder deduktiv-qualitativen Inhaltsanalyse ist entsprechend eine umfassende Beschäftigung mit dem Forschungsstand zu dem Thema, das von Interesse ist (für eine Anleitung, wie der Forschungsstand systematisch aufbereitet werden kann, siehe Kapitel 7). Denn für die Entwicklung der Forschungsfrage und vor allem für die Entwicklung des deduktiven Kategoriensystems ist es von entscheidender Bedeutung, den Forschungsstand zu kennen. Der Forschungsstand bezieht sich dabei auf bereits durchgeführte Studien, auf Theorien, die zur Erklärung eines bestimmten Phänomens verwendet wurden, das mit Ihrem Thema in Verbindung steht. Vielleicht interessieren Sie sich aber auch für eine bestimmte Theorie und wollen diese testen. Auch dann ist es entscheidend, dass Sie sich mit den empirischen Studien vertraut machen, die diese Theorien angewandt haben. Denn bei der deduktiv-qualitativen Inhaltsanalyse handelt es sich um eine empirische Methode, bei der qualitatives Datenmaterial ausgewertet wird.

Nachdem Sie sich mit dem Forschungsstand zu dem Thema, das Sie unter-

Abbildung 5.1 Ablaufschema der deduktiv-qualitativen Inhaltsanalyse



suchen wollen, vertraut gemacht haben, geht es im zweiten Schritt (Variante a) darum, das deduktive Kategoriensystem zu entwickeln. Bei diesem Schritt geht es ganz konkret darum, dass Sie so eindeutige Kategorien wie möglich entwickeln, die überschneidungsfrei sind und nach klaren und nachvollziehbaren Regeln zugewiesen werden (siehe Kapitel 5.3). Oder es findet, wie in Variante b dargestellt, die Auswahl des Datensamples und die Erhebung der Daten statt. Entsprechend eignet sich Variante a für die Auswertung von Sekundärdaten, wohingegen Variante b auf Primärdaten angewiesen ist. Das heißt, Sie würden zum Beispiel Interviews vor der Erstellung des Kategoriensystems führen. Hierbei würden die Kenntnisse des Forschungsstandes genutzt werden, um einen Leitfaden (siehe Kurzdefinition in Box 5.1) für die Interviews zu erstellen und in Bezug auf die Forschungsfrage ein sinnvolles Sampling, also eine sinnvolle Auswahl an zu befragenden Personen zu wählen.

Unter Sampling wird allgemein die Auswahl von Fällen verstanden, die untersucht werden sollen. Für die qualitative Inhaltsanalyse können das z. B. Interviews, teilnehmende Beobachtungen, Zeitungsartikel, Publikationen, politische Dokumente oder Tweets sein. Die Fälle werden dabei nicht zufällig ausgewählt, sondern nach

Box 5.1: Kurzdefinition Interviewleitfaden
Es gibt unterschiedliche Varianten von Interviews, die sich vor allem in Bezug auf ihre Strukturierung unterscheiden. Entsprechend unterscheidet sich auch die Strukturierung der Interviewleitfäden.
a) *Stark strukturierter Leitfaden:* Die Fragen sind ausformuliert und werden immer gleichgestellt.
b) *Semi strukturierter Leitfaden:* Es gibt Kernfragen, die in allen Interviews gestellt werden, aber nicht zwangsläufig in der gleichen Reihenfolge gestellt werden. Zudem sind Fragen möglich, die sich aus dem Interview ergeben.
c) *Wenig strukturierter Leitfaden:* Nur die Einstiegsfrage ist formuliert und es gibt Themenbereiche, die angesprochen werden können. Die Fragen ergeben sich aus dem Gespräch.

Box 5.2: Weitere Materialien und Informationen online
Auf dem Blog „sozmethode" finden Sie Beispiele zur deduktiven Inhaltsanalyse unter *https://sozmethode.hypotheses.org/category/deduktive-qualitative-inhaltsanalyse.*

inhaltlicher Repräsentativität (Lamnek und Krell 2016). Das heißt, es werden die Fälle ausgewählt, die im Hinblick auf die eigene Forschungsfrage die reichhaltigsten Informationen liefern können (Misoch 2015, S. 186). Im Folgenden werden drei gängige Sampling-Strategien vorgestellt: das Sampling anhand festgelegter Kriterien, das theoretische Sampling und das Schneeballsystem.

- *Sampling nach festgelegten Kriterien:* Bei diesem Sampling-Verfahren werden ausgehend vom Forschungsstand Kriterien entwickelt, die dazu dienen sollen, ein möglichst breites oder enges Sample zu wählen. Wenn z. B. untersucht werden soll, wie Studierende durch die Corona-Krise in ihrer sozialen Einbettung in Hochschulen betroffen sind, dann würden bei einem breiten Sample Studierende aus möglichst vielen unterschiedliche Studiengängen, möglichst vielen Semestern, mit möglichst vielen unterschiedlichen Ausprägungen (Alter, Geschlecht, körperliche Einschränkungen, sozialer Status etc.) ausgewählt werden. Denn das könnten alles Faktoren sein, die auf die soziale Einbettung wirken. Wenn aber z. B. bei einer Abschlussarbeit nicht so viele Studierende befragt werden können, dann muss ein Fokus gewählt werden, z. B. auf ein Studienfach und nur auf Erstsemester. Dann würden Sie aber versuchen, andere Faktoren wie Geschlecht, Alter etc. möglichst divers zu erfassen.
- *Theoretisches Sampling:* Beim Theoretischen Sampling, das aus der Grounded Theory (Glaser und Strauss 2008; Strauss 2007) stammt, wird zunächst ein Fall ausgewählt, der sofort analysiert wird, um aufgrund der Ergebnisse den nächsten Fall auszuwählen. Diese Sampling-Strategie ist allerdings nicht mit der deduktiv-qualitativen Inhaltsanalyse vereinbar, sondern wird hier nur der Vollständigkeit halber aufgeführt.
- *Schneeballsystem:* Auch beim Schneeballsystem wird ein erster Fall identifiziert, der als reichhaltiger Informationsträger zur Beantwortung der Forschungsfrage angenommen wird. Am Ende z. B. des Interviews wird dann die Person gebeten, andere Personen zu benennen, die ebenfalls für die Beantwortung der Forschungsfrage von Interesse sein könnten.

Nachdem das Sample und die Datenerhebung erfolgt ist sowie das Kategoriensystem entwickelt wurde, machen Sie sich im vierten Schritt mit dem Material vertraut. In Variante b mag es Ihnen seltsam erscheinen, dass Sie sich mit dem Material, das Sie selbst erhoben haben, vertraut machen sollen. Sie werden aber feststellen, dass auch wenn Sie z. B. Interviews geführt haben, diese in transkribierter Form komplett anders zu lesen sind, als Sie die Interviews in Erinnerung hatten. Deshalb wäre unser Ratschlag auch immer: Transkribieren Sie das Interview komplett. Indem Sie Fragen stellen wie „who is telling? where is this

happening? when did it happen? what is happening? why?" (Elo und Kyngäs 2008, S. 109), machen Sie sich mit dem Material vertraut. Es ist dabei sinnvoll, die ersten Ideen dazu sofort aufzuschreiben, um die ersten analytischen Blicke zu dokumentieren.[1] Sich mit dem Material vertraut zu machen, geht in der Methodenliteratur oftmals damit einher, Verlaufsprotokolle (Bremer und Teiwes-Kügler 2013) oder Zusammenfassungen des Falles (Kuckartz 2018) zu schreiben. Das erscheint uns allerdings sehr aufwendig und bei einer deduktiv-qualitativen Inhaltsanalyse nicht notwendig.

Nachdem wir einen Überblick über das Material bekommen haben, starten wir mit der deduktiven Kodierung des Materials (Schritt 5). Wie das genau abläuft, beschreiben wir noch in aller Ausführlichkeit anhand eines Beispiels in Kapitel 5.5.

Im Grunde besteht der Kodierprozess daraus, das vorliegende Textmaterial den Kategorien zuzuordnen. Hierfür kann das Computerprogramm MAXQDA genutzt werden (das wir im Folgenden nutzen werden), es gibt aber noch weitere Programme wie Atlas.ti oder NVivo. Bei der deduktiv-qualitativen Inhaltsanalyse werden aus der Theorie und empirischen Studien Hauptkategorien entwickelt und meist auch eine Ebene der Subkategorien. Darüber hinaus ist es aber häufig der Fall, dass weitere Subkategorien für die Ausdifferenzierung der Kategorien notwendig sind. Diese werden dann an dem Material laufend entwickelt und weiterentwickelt. Das Kategoriensystem in seiner endgültigen Form steht also erst am Ende des Kodierprozesses fest, also wenn das gesamte Material kodiert wurde. Da es ein dauerhafter Prozess ist, ist es am Ende dieses Prozesses sinnvoll, zu überprüfen, ob die kodierten Textstellen vor allem des ersten kodierten Materials noch zu den Kategorien und Subkategorien passen. In anderen Worten: Sie prüfen die Passung zwischen Textstellen und Kategorien nochmals, wenn das gesamte Material einmal anhand des deduktiven Kategoriensystems kodiert wurde. Hier ist wichtig zu verstehen, dass sich die Subkategorien immer auf Kategorien beziehen, die deduktiv gebildet wurden. Deshalb ist es sehr wahrscheinlich, dass nach dem deduktiven Kodierprozess Textstellen nicht kodiert werden konnten. Wenn Sie feststellen, dass diese Textstellen aber zu einer Beantwortung ihrer Forschungsfrage beitragen, dann erfolgt Schritt sechs und Sie erweitern das Kategoriensystem, indem Sie weitere Kategorien aus dem Material entwickeln. Das ist aber nur notwendig, wenn das Textmaterial und die daraus entwickelten Kategorien zur Beantwortung der Forschungsfrage dienen. Im letzten Schritt fassen Sie die Ergebnisse zusammen.

---

1 Für empirisches Arbeiten empfiehlt es sich immer, ein sogenanntes Forschungstagebuch zu führen, in dem Sie alles notieren, was mit Ihrer Forschung zu tun hat. Also alle Ideen, Gedanken, Interpretationsansätze, spannende Literatur, die Sie noch lesen wollen, usw. Ob das Forschungstagebuch dabei analog oder digital (z. B. die Notiz-App auf Ihrem Smartphone) ist, ist egal.

5.2 Forschungsstand und Forschungsfrage: Schritt 1

Im Folgenden möchten wir Ihnen eine Anleitung für die deduktiv-qualitative Inhaltsanalyse geben, die anhand eines konkreten Beispiels aufzeigt, wie die Entwicklung eines Kategoriensystems und die Kodierung erfolgt. Dazu ist es im ersten Schritt notwendig, eine Forschungsfrage zu generieren, die aus einem Forschungsstand abgeleitet ist. Konkret untersucht unser Beispiel, wie Studierende das Ankommen an die Hochschule im zweiten Corona-Semester (im Wintersemester 2020/21) erlebt haben. Das Ankommen gestaltete sich dabei schwierig, da der physische Sozialraum Hochschule nicht betreten werden konnte, denn die Hochschulen waren zu diesem Zeitpunkt alle geschlossen. Der gesamte Lehrbetrieb war auf Online-Lehre umgestellt, sodass viele Tätigkeiten des Studierens nicht möglich waren wie das Betreten eines physischen Seminarraums oder Hörsaals, das Vor-Ort-Treffen von Kommiliton*innen, das gegenseitige Kennenlernen bei der Orientierungswoche oder das zufällige Gespräch auf dem Flur, in der Bibliothek oder mit der*dem Sitznachbar*in. Weggefallen sind aber auch zufällige Gespräche oder der informelle Austausch mit Dozierenden. All das kann unter Interaktion in der Hochschule gefasst werden. Wie hat sich diese Interaktion nun aber bei Studierenden im ersten Semester gestaltet, die nicht die Möglichkeit hatten, den physischen Sozialraum Hochschule zu nutzen? Mit dieser Forschungsfrage wollen wir uns im Weiteren beschäftigen.

Für die deduktiv-qualitative Inhaltsanalyse führen wir nun eine systematische Literatursuche durch (siehe dazu Kapitel 7), um uns Studien anzusehen, die sich mit der Frage beschäftigen, wie Studierende mit den Bedingungen der Corona-Semester umgegangen sind. Wir haben diverse Studien gefunden, die Befragungen von Studierenden durchgeführt haben und die Studienbedingungen fokussieren (Arndt et al. 2020; Händel et al. 2020a; Händel et al. 2020b; Kreulich et al. 2020; Lörz et al. 2020; Traus et al. 2020). In diesen Studien wird deutlich, dass den Studierenden vor allem die Interaktion mit Kommiliton*innen und Dozierenden fehlt. Ebenso aber auch der Sozialraum Hochschule selbst, der als fehlender Lernraum adressiert wird. Wir finden zudem Studien, in denen Lehrende zu Wort kommen und über die Bedingungen der Lehre in der Corona-Pandemie berichten bzw. reflektieren (Arndt et al. 2020; Autor:innengruppe AEDiL 2021; Kreulich et al. 2020). Die gefundenen Studien sind empirische Zusammenfassungen, die vor allem Einblicke in die Situationen und die Herausforderungen mit den veränderten Bedingungen in der Corona-Pandemie aufzeigen. Theoretische Reflexionen in Bezug auf die Interaktion und den Sozialraum Hochschule finden sich bei Steinhardt (2021). In dieser Studie wird auf die „Student Engagement Theory" von Tinto (1975; 1997) eingegangen und anhand der Theorie herausgearbeitet, wie ein Sozialraum für Studierende in der Online-Lehre geschaffen werden kann (Steinhardt 2021).

Der Fokus auf den Sozialraum, der mit Tinto (1975; 1997) erfasst werden

kann, bewegt uns dazu, diese Theorie heranzuziehen. Im Kontext der Hochschulforschung ist Tintos Theorie eine weit verbreitete Herangehensweise (Braxton et al. 2000), um zu erklären, warum Studierende ihr Studium abbrechen und Erklärungsansätze zu liefern, wie Hochschulen dem entgegensteuern können. Zentraler Ansatzpunkt ist, dass die Studienabbruchsneigung von Studierenden variiert, je nachdem, wie stark sie in Hochschulen eingebunden sind und wie gut die Passung zwischen akademischem Selbstbild und Institution ist. Im Folgenden wird keine Rezeption von Tintos Theorie gegeben, sondern der Befund herausgegriffen, dass je stärker Studierende in das Hochschulleben involviert sind, desto größer sind ihr Wissenserwerb und die Entwicklung ihrer Fähigkeiten und desto weniger wahrscheinlich ist ein Abbruch des Studiums (Tinto 1997, S. 600).

Bekannt ist aus unterschiedlichen Studien in Deutschland, dass sich die Einbindung von Studierenden sich in Bezug auf die soziale, institutionelle und akademische Einbindung unterscheidet (Dahm und Lauterbach 2016; Müller und Braun 2018). Entsprechend wird angenommen, dass durch eine stärkere Einbindung von Studierenden in die Hochschule und das Hochschulleben, z. B. durch Peer-Groups oder durch das Involvement von Lehrenden, die Abbruchneigung verringert werden kann. Dabei spielt vor allem der Sozialraum Hochschule eine große Rolle, als physischer Raum, in dem sozialer Austausch (z. B. Flurgespräche, gemeinsames Kaffeetrinken), Interaktion mit anderen Studierenden (z. B. gemeinsames Lernen) und wissenschaftlichen Mitarbeiter*innen sowie Professor*innen möglich sind. Durch die Corona-Pandemie und das Schließen der Hochschulen ist nun gerade dieser Sozialraum als physischer Raum weggebrochen. Uns interessiert deshalb, welche Aspekte des (fehlenden) Sozialraums Studierende beschreiben, was ihnen fehlt, und wie sie sich selbst (nicht) eingebunden fühlen.

Das Heranziehen der Theorie von Tinto sowie die Ergebnisse der empirischen Studien führt uns zu folgenden Annahmen: Wir gehen davon aus, dass sich das Hochschulleben durch den Wegfall des physischen Sozialraums in der Wahrnehmung der Studierenden verändert hat. Unter Hochschulleben verstehen wir erstens die Interaktion mit Kommiliton*innen und Lehrenden, ebenso die Interaktion im Seminarraum, aber auch die Mitarbeit z. B. an einem Lehrstuhl oder in Hochschulgruppen wie dem Allgemeinen Studierendenausschuss AStA. Zweitens ist für das Hochschulleben der Sozialraum entscheidend, weshalb von Interesse ist, inwieweit das Fehlen des Sozialraums, also z. B. Lernräume oder die Bibliothek, eine Rolle für das Ankommen an der Hochschule spielt. All das waren Faktoren, die in der Corona-Pandemie weggefallen sind, weshalb es für die Forschungsfrage von entscheidender Bedeutung ist, zu analysieren, ob der Wegfall von Studierenden benannt wird und welche Bedeutung dem Wegfall gegeben wird.

5.3 Erstellung des deduktiven Kategoriensystems: Schritt 2

Nachdem wir nun die theoretische Einordnung geleistet haben, können wir Kategorien ableiten. Zentral für das Erstellen von Kategorien ist, dass diese klar voneinander abgrenzbar sind und klare Definitionen aufweisen (Graneheim et al. 2017; Mayring 2015). Aus dem vorherigen Abschnitt ist bereits deutlich geworden, dass es zwei zentrale Bereiche gibt: Interaktion und Sozialraum. Das werden auch unsere zentralen Hauptkategorien, die nun in einem zweiten Schritt Subkategorien erhalten werden, die voneinander klar abgrenzbar sein müssen.

Für die deduktiv-qualitative Inhaltsanalyse nutzen wir als unterstützendes Programm MAXQDA (siehe Hinweis in Box 5.3, siehe zu Copylefts und Copyrights Kapitel 1.3). Leider existiert bisher noch kein Open Source-Programm für die qualitative Inhaltsanalyse.[2] In Deutschland sind zwei Programme Marktführer: MAXQDA und Atlas.ti. Ersteres wurde vor allem für die qualitative Inhaltsanalyse entwickelt, wohingegen Atlas.ti aus dem Zusammenhang der Grounded Theory entstand. Beide Programme funktionieren ähnlich und ermöglichen die digitale Organisation und digital unterstützte Analyse von Textmaterial wie beispielsweise Interviews, Gruppendiskussionen oder Dokumente, aber auch Videos oder Bilder. MAXQDA hat zudem eine Schnittstelle, um Tweets direkt in das Programm zu laden. MAXQDA ist eine lizensierte Software, d. h., wenn diese nicht an Ihrer Universität als Campuslizenz zur Verfügung steht, müssen Sie diese als Studierendenlizenz kaufen. Von MAXQDA gibt es auch eine Lehrlizenz, sodass in einem Seminar für ein Semester kostenlos mit dem Programm gearbeitet werden kann. Das Programm läuft identisch auf Windows- und Mac-Rechnern.

Box 5.3: Hinweis MAXQDA 1
Von MAXQDA gibt es ein sehr gutes Manual (also die Nutzungsanleitung als Handbuch) mit Online-Tutorials für Ihr Programm, in dem Schritt für Schritt die Nutzung erläutert wird. Deshalb haben wir hier keine eigene Anleitung verfasst. Das Manual finden Sie auf der Startseite von MAXQDA unter „Online-Manual“.
www.maxqda.de

Für unser weiteres Vorgehen ist nun der erste Schritt die Erarbeitung des Kategoriensystems (siehe Hinweis in Box 5.4). Da es in MAXQDA möglich ist, eine Excel-Tabelle als Kategoriensystem direkt zu importieren, haben wir Tabelle 5.1 in Excel erstellt. Um das Kategoriensystem in unserer Excel-Tabelle in MAXQDA importieren zu können, muss die Excel-Tabelle als erste Überschrift „Code“ enthalten, damit MAXQDA weiß, dass darunter die Kategoriennamen zu finden sind. Subkategorien können in der Spalte „Code“ mithilfe eines sogenannten Backslashs „\“ eingefügt werden. Die zweite Spalte muss die Überschrift „Memo“

2 Es gibt das Annotationsprogramm hypothes.is, das eine Kodierung von digitalem Text (also PDFs oder Webseiten) ermöglicht, sowohl in einer geschützten Gruppe (oder allein) als auch im Internet. Allerdings ist es noch nicht möglich, ein Kategoriensystem zu erstellen, weshalb es hier nicht genutzt wurde.

enthalten, hierunter sind die Definitionen der Kategorien aufgelistet. Um den Import zu starten, wählen Sie in MAXQDA auf dem Tab „Codes“ die Funktion „Codes und Memos aus Excel-Tabelle importieren“ (MAXQDA 2020, S. 162). MAXQDA erstellt daraufhin automatisch die Kategorien und weist unter „Memo“ die entsprechende Definition zu. Bitte beachten Sie, dass der Kategorienname insgesamt nicht länger als 63 Zeichen (inklusive Leerzeichen) sein darf, sonst wird er abgeschnitten. Wenn Sie das Kategoriensystem importiert haben, dann können Sie durch einen Rechtsklick mit der Maus auf den Codenamen das Eigenschaftsfenster aufrufen. Dort können Sie dann Farben für unterschiedliche Kategorien vergeben. Die Farbzuweisung erleichtert Ihnen den schnellen Überblick über Ihre Kategorien und kodierten Textstellen. Wenn Sie Ihr Material kodieren, können Sie unter Memo auch noch Ankerbeispiele einfügen, um die Definition der Kategorien noch eindeutiger zu machen. Als Ankerbeispiel werden Textstellen verstanden, die in besonders eindeutiger Weise die Kategorie oder Subkategorie abbilden (Mayring 2010).

Box 5.4: Hinweis MAXQDA 2
Wichtig ist zu wissen, dass in MAXQDA Kategorien als „Codes“ bezeichnet werden. Dies ist irreführend, da in der qualitativen Inhaltsanalyse ein Code eigentlich die kodierte Textstelle ist.

Tabelle 5.1 Deduktives Kategoriensystem (Fortsetzung nächste Seite)

Code	Memo
Interaktion Kommiliton*innen	Erste Hauptkategorie
Interaktion Kommiliton*innen\Peer-Group	Studierende berichten von Kontakt zu ihrer Peer-Group, von fehlendem Kontakt zu ihrer Peer-Group oder davon, dass sie noch keine Möglichkeit hatten, sich eine Peer-Group aufzubauen, da sie gerade erst angefangen haben zu studieren. Eine Peer-Group ist eine soziale Gruppe, der sich zugehörig gefühlt wird.
Interaktion Kommiliton*innen\Lerngruppe	Studierende berichten, dass sie Lerngruppen gegründet haben, Lerngruppen vermissen oder sich im Digitalen keine Lerngruppen gründen lassen. Im Gegensatz zur Peer-Group bezieht sich eine Lerngruppe auf einen spezifischen Zusammenhang, wie z. B. die Vorbereitung auf ein Referat oder auf eine Klausur.
Interaktion Kommiliton*innen\Sozialleben	Studierende berichten von z. B. dem Wegfall von Studierendenpartys oder Aktivitäten im AStA.
Interaktion Kommiliton*innen\Flurgespräche	Studierende berichten, dass ihnen Flurgespräche und zufällige Treffen fehlen z. B. vor Seminarbeginn oder in der Bibliothek, in denen sich z. B. über Seminarinhalte ausgetauscht wurde. Möglich ist auch, dass Studierende von Flurgesprächen z. B. in Breakout-Sessions berichten.
Interaktion Lehrende	Zweite Hauptkategorie
Interaktion Lehrende\Erreichbarkeit	Studierende berichten von ihren Kontakten und der Erreichbarkeit zu Lehrenden, z. B. via E-Mail oder Zoom-Sprechstunde.

Code	Memo
Interaktion Lehrende\ Kommunikation	Studierende berichten über die (fehlende) Kommunikation mit Lehrenden, darunter fallen auch gegebene oder fehlende Informationen zu den Lehrveranstaltungen.
Interaktion Lehrende\ Integration Seminarkontext	Studierende berichten darüber, wie sie im Seminar eingebunden wurden, z. B. in Zoom-Sitzungen. Oder sie berichten darüber, dass die Lehrenden ein reines Abarbeiten von gestellten Aufgaben erwarteten.
Interaktion Lehrende\ Rücksichtnahme im Seminar	Studierende berichten darüber, dass Lehrende aktiv in Erfahrung gebracht haben, vor welchen Herausforderungen Studierende stehen. Zum Beispiel in Bezug auf Care-Arbeit, schlechte Internetverbindung oder andere Herausforderungen.
Sozialraum Hochschule	Dritte Hauptkategorie
Sozialraum Hochschule\ Lernräume	Studierende beschreiben die aktuelle Situation in Bezug auf Lernräume. Dabei kann sowohl das Fehlen adäquater Möglichkeiten zum Lernen zu Hause vorkommen als auch das Fehlen der Möglichkeit, an der Hochschule Lernräume zu nutzen.
Sozialraum Hochschule\ Bibliothek	Studierende berichten von den Auswirkungen, die der Wegfall der Bibliothek als sozialer Raum für sie hat (→ Problematik, nicht an Literatur zu kommen, ist hierbei nicht relevant).
Sozialraum Hochschule\ Soziales Leben in Hochschulräumen	Studierende beschreiben das Fehlen des sozialen Lebens an der Hochschule, z. B. das Wegfallen von Partys an der Hochschule. Wichtig: Es geht hier um den Wegfall der Räumlichkeiten, nicht um die Interaktion mit Kommiliton*innen → siehe dazu die Kategorie „Interaktion Kommiliton*innen\ Sozialleben“.
Sozialraum Hochschule\ Berufstätigkeit	Studierende berichten von ihrer Berufstätigkeit in der Hochschule, z. B. als Mensamitarbeiter*in oder studentische Hilfskraft.

Wie an dem hier deduktiv entwickelten Kategoriensystem deutlich wird, gibt es zwei Oberkategorien und weitere Subkategorien. Es ließe sich noch eine dritte Ebene hinzufügen, die zwischen positiv, neutral und negativ untergliedert. Da wir zu diesem Zeitpunkt aber noch nicht wissen, wie die Beurteilungen der Studierenden zu den Aspekten ausfallen, lassen wir dies zunächst offen – auch um die unterschiedlichen Beurteilungsebenen besser miteinander vergleichen zu können. Diese vierte Ebene würde dann aufgrund des Materials kontinuierlich eingefügt werden.

Wie Sie an den Definitionen bereits ablesen konnten, geht es uns bei diesem Beispiel nicht nur um die manifesten Inhalte, sondern auch um latente Inhalte (Elo und Kyngäs 2008, S. 109). Wenn Sie sich für ein deduktives Kategoriensystem entscheiden, müssen Sie auch entscheiden, ob nur manifeste Inhalte kodiert werden sollen oder auch latente Inhalte gehoben werden. Manifester Inhalt wird beschrieben, ist also direkt aus dem Text zu verstehen, wohingegen latente Inhalte interpretiert werden müssen (Graneheim et al. 2017).

Ein fiktiver manifester Inhalt wäre z. B.: „Ich habe keine Lerngruppe während

des Semesters gründen können". Diese würden der deduktiven Kategorie „Interaktion Kommiliton*innen\Lerngruppe" zugeordnet. Wohingegen latente Inhalte einer Interpretation von Textpassagen bedürfen. Wichtig ist, dass auch latente Inhalte am Text orientiert und keine „wilden" Spekulationen sind. Ein Beispiel für eine latente Textstelle wäre fiktiv z. B. „Ich wusste nicht, was ich in der Vorlesung machen sollte". Aus dem Kontext der Textstelle könnte interpretiert werden, dass die Unsicherheit aufgrund fehlender Informationen durch den*die Dozierende*n entstanden ist. Dann würde diese Textstelle der Kategorie „Interaktion Lehrende\Kommunikation" zugeordnet werden.

5.4 Vertraut machen mit dem Material: Schritt 3

Wie im Ablaufschema (Abbildung 5.1) gezeigt, gibt es für die deduktive Kodierung zwei Varianten. Für unser Beispiel haben wir uns für eine Sekundärauswertung von Textmaterial entschieden. Das heißt, wir greifen auf Material zurück, das nicht für diese deduktiv-qualitative Inhaltsanalyse erhoben wurde. Denn uns war wichtig, Ihnen zu zeigen, vor welche Herausforderungen Sie gestellt sind, wenn Sie eine Sekundärauswertung machen. Bei einer Primärerhebung haben Sie selbst Einfluss darauf, welche Fragen Sie z. B. bei Interviews stellen. Dazu würden Sie sich mit dem Thema vertraut machen und Leitfragen aus der Theorie oder bereits durchgeführten Studien ableiten. Diese Leitfragen können dann auch die Hauptkategorien ihrer deduktiven Kategorisierung werden.

Im vorliegenden Beispiel haben wir uns dazu entschieden, das gleiche Material auszuwerten wie bei der induktiv-qualitativen Inhaltsanalyse (siehe Kapitel 4). Dadurch können wir aufzeigen, wie unterschiedlich induktive und deduktive Kodierungen ausfallen können und zu welchen unterschiedlichen Ergebnissen die Kategoriensysteme und Kodierungen führen.

Das gewählte Datenmaterial ist eine autoethnographische Aufzeichnung, gemeinhin auch bekannt als Tagebucheinträge, einer*eines Studierenden zum Corona-Pandemie bedingten Online-Wintersemester 2020/21. Das Lehr-Lern-Forschungsprojekt Autoethnographie zu „Zwei Wochen Studium im Wintersemester 2020/21" war Teil einer Methoden-Vorlesung in einem Bachelorstudiengang an einer Universität in Deutschland. Das inhaltliche Interesse der Autoethnographie war, wie die Anfang November 2020 aufkommende zweite Corona-Welle das Studium und, für die neuen Studierenden, den Studieneinstieg prägt. Die Untersuchung leitende Fragestellung für die Studierenden lautete: „Wie gestalten Sie Ihren Studienalltag zu Beginn des Online-Semesters 2020/21?" Die Aufgabenstellung war, dass die Studierenden in einem Zeitraum von 14 Tagen etwa zehn autoethnographische Aufzeichnungen verfassen sollten. Zur Orientierung erhielten die Studierenden einen Beispieltext einer autoethnographischen Aufzeichnung aus einem anderen Forschungsprojekt, ergänzend zu Methodenlitera-

tur, Erklärungen zu Autoethnographie in einem Video und der Anleitung durch die*den Lehrenden in der Vorlesung.

Bei einer Sekundärauswertung wissen Sie zunächst nicht, was in dem Material alles enthalten ist. Das heißt, es kann durchaus sein, dass Sie nicht alle Fragen, die Sie aufgrund der Aufarbeitung des Forschungsstandes stellen, werden beantworten können. Zentral für eine Sekundärauswertung ist, den Kontext des Materials zu kennen. Deshalb gibt es zum Beispiel bei Repositorien, die qualitative Daten zur Sekundärauswertung zur Verfügung stellen,[3] immer sehr ausführliche Datenbeschreibungen. Diese umfassen in der Regel folgende Punkte (als Beispiel für einen Methodenbericht Steinhardt und İkiz-Akıncı 2020).

Organisationaler Rahmen

- Beschreibung des Erhebungskontextes, wie zum Beispiel Forschungsprojekt oder wie in unserem Beispiel Lehr-Forschungsprojekt

Methodisches Vorgehen

- Erhebungsmethode, z. B. narratives Interview, Leitfadeninterview oder wie in unserem Beispiel Autoethnographie
- Sampling, z. B. anhand von Kriterien, Theoretical Sampling oder Schneeballsystem (siehe dazu Kapitel 3.2.1 und Kapitel 5.1)
- Auswertungsmethode der Primärerhebung, also inwiefern und mit welchen Ergebnissen die Daten bereits ausgewertet wurden
- Datenschutzfragen, also ob das Einverständnis einer Sekundärnutzung vorliegt

Datenaufbereitung

- Aufbereitung des Textmaterial, bei Interviews zum Beispiel die Transkriptionsregeln
- Anonymisierung (siehe dazu Kapitel 4.2.2)

Nachdem nun geklärt ist, welches Material wir nutzen, machen wir uns mit dem Material vertraut. Wie beschrieben, handelt es sich um eine Autoethnographie von einer*einem Studierenden. Das heißt, in den insgesamt zehn Einträgen, die wir uns hier als Beispiel anschauen (siehe das gesamte Material in Kapitel 4, Tabelle 4.1), wurden der Studienalltag und Belange, die mit dem Studienalltag im weitesten Sinne zu tun haben, beschrieben.

---

3 Derzeit gibt es 40 Repositorien (auch Forschungsdatenzentren genannt), die durch eine Akkreditierung gewährleisten, nach Standards zu arbeiten. Die Liste der Repositorien finden Sie hier: *www.konsortswd.de/datenzentren/alle-datenzentren.*

Als ersten Schritt lesen wir uns das gesamte Material durch, um einen Eindruck zu erhalten. Wie von Tagebuchaufzeichnungen zu erwarten, strukturieren sich die Texte anhand des Tagesablaufs und fokussieren dabei Gegebenheiten, die mit dem Studium in Zusammenhang stehen. Beschrieben wird, wann aufgestanden wurde, wann was für die einzelnen Vorlesungen und Veranstaltungen gelernt wurde oder welche Tools für das Online-Studium genutzt wurden. Berichtet wird aber auch von Interaktionen mit Studierenden, Interaktion mit Lehrenden kommt wiederum kaum vor. In den Daten fallen zudem zwei Aspekte auf, die besonders häufig in den Autoethnographien von Studierenden angesprochen werden.

1. Aufgrund der Corona-Pandemie und damit einhergehend des Online-Studiums zu Hause werden *Interaktionen mit den Eltern und Geschwistern* beschrieben: „Ich ging runter in die Küche und habe mir einen Kaffee gemacht, so wie jeden Morgen. Meine Mutter war auch noch in der Küche und hat mich gefragt, was denn meine erste Vorlesung sein wird. Kurz habe ich erklärt, dass es sich hierbei um die Vorlesung [Hauptfach] handelt, und diese um [Vormittag] Uhr anfängt" (Absatz 1/2).
2. Es finden sich *Ablaufbeschreibungen des Lernens und Studierendens,* wie zum Beispiel „Ich habe mich daraufhin versucht bei Moodle einzuloggen, um mir nochmal anzuschauen, was mich heute in der Vorlesung erwarten wird, ich hatte mir bereits am Wochenende den Text durchgelesen, aber zur Sicherheit habe ich noch mal nachgesehen" (Absatz 1/4).

Da diese Aspekte wenig mit unserer Forschungsfrage zu tun haben, werden wir diese im Weiteren bei der deduktiven Kodierung zunächst nicht berücksichtigen.

5.5 Deduktive Kodierung: Schritt 4

Wie bereits ausgeführt, gibt es auf der Webseite von MAXQDA *(www.maxqda.de)* eine sehr gute Handreichung zum Arbeiten mit dem Programm. Deshalb beschreiben wir an dieser Stelle nicht, wie Sie das Programm installieren können oder wie es aufgebaut ist. Für das Verständnis des Ablaufs der deduktiven Kodierung ist wichtig, dass das Kategoriensystem via Excel-Liste in MAXQDA importiert wurde. Zudem haben wir die Autoethnographie, die uns als Beispiel dient, als Dokument in MAXQDA hochgeladen. Bevor Sie weiterarbeiten, müssen Sie alle zu analysierenden Dokumente in MAXQDA importieren. Zum Öffnen der Dokumente gehen Sie bitte

1. im MAXQDA Menü zum Tab „Importieren",
2. klicken das erste Icon „Texte, PDFs, Tabellen" an,

3. wählen auf Ihrem Computer die zu analysierenden Dokumente durch Markieren im jeweiligen Dateiordner aus, und
4. schließen den Importvorgang durch Anklicken des Buttons „Öffnen" ab.

Zwei Tipps für den Import der Dokumente: Erstens sollte am besten erst eine Dokumentgruppe angelegt werden (z. B. inhaltlich „Tagebucheinträge" oder nach Erhebungszeitpunkten „t1", „t2" oder nach Standorten „Uni München" und „Uni Kassel"). Zweiten kann man Dokumente auch direkt aus dem Windows Explorer oder dem macOS Finder per Klicken-und-Ziehen in die „Liste der Dokumente" in ein Projekt einfügen.

Die zu analysierenden Dokumente erscheinen in der MAXQDA-Standardeinstellung im Fenster oben links („Liste der Dokumente"). Per Doppelklick auf einen Dokumentnamen wird das Dokument im Fenster „Dokument-Browser" geöffnet und kann dort kodiert werden. Wenn man (später) alle Dokumente aktivieren möchte, geht das per Rechtsklick auf die Wurzel in der „Liste der Dokumente" und Auswahl der Funktion „Alle Dokumente aktivieren" oder noch schneller: Durch Klick auf das Symbol links neben „Dokumente".

Wenn Sie das Dokument aktiviert haben wird es im „Dokument-Browser" angezeigt, sodass wir mit diesem Dokument arbeiten können. Das heißt, es können Textstellen markiert werden und durch das Ziehen der markierten Textstelle auf einen „Code" im Fenster „Liste der Codes" einer Kategorie zugeordnet werden. Nach der Kodierung werden die kodierten Textstellen in der „Liste der codierten Segmente" angezeigt, quasi in einem „Resultatsfenster" (MAXQDA 2020; siehe dazu auch das Kapitel „die MAXQDA-Oberfläche"). Diese Liste kann nun exportiert werden, indem Sie dazu das Symbol „Exportieren" bei „Liste der codierten Segmente" anklicken. Ausgewählt werden kann dann zwischen einer Word- oder Excel-Liste. Wir haben die Excel-Liste gewählt, da diese übersichtlicher ist. In der Liste werden die Farbmarkierung, der Kategorienname, die kodierten Textstellen inklusive der Absatznummer angezeigt. Für uns ist hier nur die Kategorie und die dazugehörige Textstelle von Bedeutung. Sich die Excel-Tabelle herunterzuladen, ist vor allem dann sinnvoll, wenn Sie mit einer MAXQDA-Lehrlizenz oder in einem PC-Pool an der Hochschule arbeiten und Ihre Arbeit nicht dauerhaft in MAXQDA auf Ihrem Computer gespeichert wird. Ansonsten können Sie auch weiterhin in MAXQDA arbeiten und zum Beispiel weiter im „Smart-Coding-Tool" oder auch direkt im „Dokument-Browser" weiterarbeiten.

Für die deduktive Kodierung unseres Beispiels ergibt sich folgendes Resultat.

Tabelle 5.2 Deduktive Kodierung mittels deduktivem Kategoriensystem
(Fortsetzung nächste Seite)

Kategorie (= MAXQDA Code)	Kodierte Textpassage (= MAXQDA Segment)
Interaktion Kommiliton*innen\Peer-Group	Als es dann [Vormittag] war, wollte ich mich bei Moodle einloggen, um zum Link des Zoom-Meetings zu gelangen, aber das hat sich als nicht möglich herausgestellt, da Moodle lahmgelegt war, durch all die neuen Studenten, die sich versucht haben anzumelden. Einer aus der WhatsApp Gruppe der Erstis unseres Studiengangs hatte den Zoom-Link gespeichert, und ihn in die Gruppe gesendet, sodass wir doch noch zeitlich an der Zoom-Vorlesung teilnehmen konnten. (Absatz 1/6) Gegen [Nachmittag] ist uns, Erstis, aufgefallen, dass wir keinen Zoom-Link zu unserer Vorlesung für [Nachmittag] hatten. Kurz vor Beginn der Vorlesung hat sich dann herausgestellt, dass diese Vorlesung in [Nebenfach] asynchron stattfinden wird. Das heißt der Dozent lädt die Vorlesung im Moodle hoch und wir können uns diese dann anschauen, wann wir Zeit haben. (Absatz 2/5)
Interaktion Kommiliton*innen\Lerngruppe	Am Mittagstisch habe ich über meine erste Uni Woche berichtet, und darüber, dass ich am Donnerstag noch die Übungsgruppe des Modules [fachfremder Zusatzkurs] haben werde und mir dazu unbedingt noch die Vorlesung anschauen will für Aufgabe 2, auch wenn es über Aufgabe 1 gehen wird, aber falls ich Fragen hätte ich sie stellen könnte. (Absatz 3/3) Bis [Vormittag] habe ich ausgeschlafen, weil in unserem Stundenplan nichts mehr steht, auf Moodle jedoch wurde uns mitgeteilt, dass wir in den Übungsgruppen für das Modul [fachfremder Zusatzkurs] verschiedene Stunden haben. Ich bin in der Übungsgruppe 01 und habe somit Donnerstag [Nachmittag] diese Stunde. (Absatz 4/1) In der Vorlesung haben wir kurz in Breakout-Session über die vergangene Woche und unsere Erfahrungen im Hinblick unserer Aufgabe der Autoethnographie gesprochen und darüber diskutiert, wie wir unsere Informationen festhalten, um sie nachher niederzuschreiben. (Absatz 6/2) Das [fachfremde Zusatzkurs]-Modul hatte sich als größeres Hindernis und Herausforderung entpuppt als ich angenommen habe, immer wieder ertappe ich mich im Laufe des Tages wie ich mir den Kopf über dieses Modul zerbreche, deswegen hatte ich mich zum Abend hin dazu entschieden meine Abgabepartnerin der Übungsgruppe darauf anzusprechen, sie teilte meine Meinung und hat erläutert, dass sie das rasante zunehmen der Erwartungen in den Übungen etwas zu extrem finden würde. (Absatz 8/3)
Interaktion Kommiliton*innen\Sozialleben	Daraufhin habe ich meine Notizen weggeräumt und bin zu einer Zoom-Veranstaltung […] [der Studierendengruppe] [aus Land] in [Stadtname] hinzugestoßenen, die die Erstis herzlich willkommen geheißen haben, [ihre Studierendengruppe] etwas vorgestellt haben und sich über unsere Mitgliedschaft freuen würden. (Absatz 2/7) Interessant war, dass wirklich jeder seine eigene Vorgehensweise hat, wir haben uns dann noch darüber unterhalten, dass es teilweise schwierig ist auf die Angegebene Wortzahl zu kommen da unser Alltag in Bezug auf das Studium sich einerseits einschränkt wegen der Pandemie, also dass es sich um ein Online-Semester handelt, aber auch dass es sich monoton anfühlt, da wir halt außerhalb nichts machen können wegen der Pandemie, damit ist das Treffen von Mitstudierenden gemeint, was ja im Normalfall zu dem Studentenalltag dazugehört. (Absatz 6/2)

Kategorie (= MAXQDA Code)	Kodierte Textpassage (= MAXQDA Segment)
	So hatte ich noch einige Minuten, bis ich an einem anderen Zoom-Meeting teilgenommen habe was von [der Studierendengruppe] aus [Stadtname] organisiert wurde. Es handelte sich hierbei um die Versteigerung der [Studienanfänger*innen], die so [in die Studierendengruppe] aufgenommen wurden, es war sehr spaßig und man hat viele neue Leute kennengelernt. (Absatz 7/6) Wegen den pandemischen Umständen lernt man in unserem ersten Studiensemester, was komplett online stattfindet, leider nicht so viele Leute kennen, deswegen war ich positiv davon überrascht, dass so viele online Treffen ermöglicht wurden um doch einen bestmöglichen Anschluss zu finden. Diese Versteigerung ging auch etwas länger, sodass ich erst nach Mitternacht ins Bett kam und mir nur noch schnell den Wecker gestellt habe. (Absatz 7/6)
Interaktion Lehrende\ Kommunikation	Kurz vor Beginn der Vorlesung hat sich dann herausgestellt, dass diese Vorlesung in [Nebenfach] asynchron stattfinden wird. Das heißt der Dozent lädt die Vorlesung im Moodle hoch und wir können uns diese dann anschauen, wann wir Zeit haben. (Absatz 2/5) In dem Propädeutikum [Hauptfach] haben wir darüber gesprochen, was auf uns zukommen wird das kommende Semester. (Absatz 3/2) Bis [Vormittag] habe ich ausgeschlafen, weil in unserem Stundenplan nichts mehr steht, auf Moodle jedoch wurde uns mitgeteilt, dass wir in den Übungsgruppen für das Modul [fachfremder Zusatzkurs] verschiedene Stunden haben. Ich bin in der Übungsgruppe 01 und habe somit Donnerstag [Nachmittag] diese Stunde. (Absatz 4/1) Unser Professor hatte entschieden die Vorlesung etwas später zu starten, erst um [Vormittag], sodass ich mir noch alles zurechtgelegt habe, den vorherigen Kurs nochmal flüchtig überschauen konnte, als wir dann pünktlich angefangen haben. (Absatz 7/1) Wie jeden Dienstag habe ich dann bis [früher Abend] keine Kurse, wir bekommen die Vorlesung für [Nebenfach] nämlich hochgeladen und können uns die individuell anschauen und so unseren Studienalltag ein bisschen selbst gestalten. Leider hatte das an diesem Tag nicht so gut funktioniert, da sie etwas zu spät online auf Moodle war, und man sie sich nicht mehr anschauen konnte vor dem Seminar, da die Vorlesung zu lange war. (Absatz 7/4 und Absatz 7/5)
Interaktion Lehrende\ Integration Seminarkontext	Am Nachmittag bin ich noch einmal über meine erste Übung gegangen, die wir bereits Montag abgegeben hatten, um vorbereitet an der Übungsgruppe teilzunehmen. Um [Nachmittag] dann hat diese Stunde begonnen am Anfang jeder Übungsstunde werden 13 Matrikelnummern gezogen und diese Studenten werden dann testiert am Ende der Stunde, ich war dieses Mal nicht dabei und auch etwas froh darüber, da ich mich noch nicht so sicher in der Materie fühle. (Absatz 4/5)
Interaktion Lehrende\ Erreichbarkeit	[…] wir hatten auch schon rausgefunden, dass wir den Professor Fragen können, ob er früher die Folien auf Moodle hochladen könnte, aber irgendwie hatte das nicht funktioniert. (Absatz 7/5)
Interaktion Lehrende\ Rücksichtnahme im Seminar	[…] wir hatten auch schon rausgefunden, dass wir den Professor Fragen können, ob er früher die Folien auf Moodle hochladen könnte, aber irgendwie hatte das nicht funktioniert. (Absatz 7/5)

Aufgrund der Farbmarkierung fällt sofort auf, dass es keine blauen Markierungen gibt, also keine Kodierungen in Bezug auf den Sozialraum Hochschule. Die Erklärung hierfür ist einfach: Es handelt sich bei der kodierten Autoethnographie um Aufzeichnungen von einem*einer „Ersti", also Erstsemesterstudierenden. Insofern wurde der Campus der Hochschule noch nicht betreten und es gibt keine Erfahrungen mit dem Sozialraum Hochschule.

Die Kodierungen zeigen insgesamt, dass es sich bei diesem Fall um jemanden handelt, der sich anscheinend schnell in die neue Situation des Studierens eingefunden hat. Berichtet wird von einer „Ersti-Gruppe", die sich bei Problemen via WhatsApp hilft und der sich die Person als Gruppenmitglied zugehörig fühlt, da sie dauerhaft von „Wir" spricht. Es wird vom Austausch in Lerngruppen berichtet, auch findet ein Austausch mit einer „Lernpartnerin" statt, d. h. es hat bereits eine Einbindung in die Hochschule stattgefunden, das Ankommen verläuft problemlos. Beschrieben wird zudem der Besuch und die Aufnahme in eine [Studierendengruppe], wodurch ein digitales Sozialleben implementiert wird. Dieses wird besonders hervorgehoben, da im Austausch mit anderen Kommiliton*innen deutlich wird, dass ein „normales" Studierendenleben vor Ort gerade nicht möglich ist.

Auf einzelne Besonderheiten bei der deduktiven Kodierung möchten wir noch hinweisen. So sind die Textstellen der Kategorie „Interaktion Lehrende\ Kommunikation" alles implizit formulierte Aussagen, die nicht direkt aus dem Wortlaut darauf hinweisen, dass es sich um die Kommunikation mit Lehrenden handelt. Als Beispiel: „Kurz vor Beginn der Vorlesung hat sich dann herausgestellt, dass diese Vorlesung in [Nebenfach] asynchron stattfinden wird. Das heißt der Dozent lädt die Vorlesung im Moodle hoch und wir können uns diese dann anschauen, wann wir Zeit haben". Die Textpassage interpretierend wird deutlich, dass den Studierenden (es wird die „Wir"-Form verwendet) erst kurz vor Beginn bewusst wurde, dass die Lehrveranstaltung asynchron stattfindet. Diese fehlende Information wurde als Kommunikation zwischen Dozierender*Dozierendem und Studierenden interpretiert. Denn auch eine fehlende Kommunikation ist eine Form der Kommunikation. Dabei ist hier nicht ausschlaggebend, wodurch das Kommunikationsproblem verursacht wurde.

5.6 Erweiterung des Kategoriensystems: Schritt 5

Wie Sie an den wenigen Textstellen schon erahnen konnten, die bei der deduktiven Kodierung kodiert wurden, sind noch viele Textstellen vorhanden, die bisher nicht kodiert wurden. Was und wie viel übrigbleibt, hängt dabei entscheidend von der Auswahl der theoretischen Perspektive ab. In unserem Beispiel ist das eine starke Fokussierung auf den Sozialraum und die Eingebundenheit in Hochschule, aber eben nicht auf das Lehr-Lernhandeln oder die Interaktion mit der

Familie. Da wir keine umfassende Analyse dessen machen wollen, was in dem Material steckt (das wäre sonst eine induktiv-qualitative Inhaltsanalyse), sondern bewusst fokussieren, stellt das kein Problem dar. Trotzdem ist es wichtig im Hinterkopf zu behalten, dass es für die Testung von Theorien zentral ist, sich das verbleibende Material anzusehen, um Theorien gegebenenfalls zu erweitern. Wenn der Zweck der deduktiv-qualitativen Inhaltsanalyse zum Beispiel darin besteht, einem Modell (wie z. B. das Modell von Tinto in unserem Beispiel) neue Dimensionen hinzuzufügen, dann sind die verbleibenden Daten wichtige Beiträge (Graneheim et al. 2017).

Deshalb betrachten wir im nächsten Schritt die nicht kodierten Textstellen und prüfen, ob diese Textstellen genutzt werden sollten, um das Kategoriensystem zu erweitern, was allerdings nur sinnvoll ist, wenn dadurch ein Mehrwehrt für die Beantwortung der Forschungsfrage entsteht. Also muss zunächst geprüft werden, was in unserem Beispiel Textstellen sein könnten, die sich auf den Sozialraum Hochschule und die Integration ins Studium beziehen. Um das zu ermitteln, wird das Material noch einmal gelesen, vor allem die Textstellen, die bisher noch nicht kodiert wurden. Dabei schließen wir sofort, wie oben schon beschrieben, Textstellen aus, die sich auf die Interaktion mit der Familie beziehen. Dazu zählt beispielsweise „Meine Mutter war auch noch in der Küche und hat mich gefragt was denn meine erste Vorlesung sein wird". Des Weiteren schließen wir Alltagsbeschreibungen aus wie zum Beispiel „Um 7:30 Uhr hat mein Wecker geklingelt, und ich stand nicht ganz so ausgeschlafen auf, durch meinen ersten Tag an der Uni war ich etwas aufgeregt und konnte nicht so gut schlafen wie gewohnt. Ich ging runter in die Küche und habe mir einen Kaffee gemacht, so wie jeden Morgen". Wenn wir diese Textstellen ausschließen, bleiben Beschreibungen des Studienalltags und Passagen zur Wohnungssuche übrig.

Mit diesen Aspekten arbeiten wir nun weiter und analysieren, inwiefern in diesen Textstellen vom Sozialraum Hochschule und Einbindung in die Hochschule die Rede ist. Nehmen wir die Passagen der Wohnungssuche, so stellen wir fest, dass es um eine Wohnungssuche in der Stadt geht, in der die Hochschule liegt. Es besteht das Bedürfnis, vor Ort das Studium aufnehmen zu können und es sollen, trotz digitalem Semester, alle Möglichkeiten ausgeschöpft werden, möglichst bald in die Stadt der Hochschule zu ziehen. Die Verbindung zum Sozialraum Hochschule ist damit gegeben, denn ohne in der Stadt zu sein, kann auch die Hochschule und der Campus nicht besucht werden. Insofern generieren wir die Kategorie „Sozialraum Hochschulort" mit folgender Definition: „Studierende berichten von ihrem Ankommen in dem Ort der Hochschule, z. B. in Bezug auf die Wohnungssuche". Entsprechend der Definition gehen wir den Text erneut durch und kodieren die entsprechenden Textpassagen, was zum Ergebnis in Tabelle 5.3 führt.

Tabelle 5.3 Deduktive Kodierung in die Kategorie „Sozialraum Hochschulort“

Kategorie (= MAXQDA Code)	Kodierte Textpassage (= MAXQDA Segment)
Sozialraum Hochschulort	Nach diesen 90 Minuten hatte ich bis [Nachmittag] keine Vorlesung mehr, so habe ich mich also anders beschäftigt und noch weiter im Internet nach Wohnungsanzeigen gesucht, und welche angeschrieben. Ich bin momentan nämlich noch auf Wohnungssuche, und durch die aktuelle Situation der Corona-Pandemie gestaltet sich diese schwieriger als gedacht. (Absatz 2/4) Es gab Neuigkeiten bezüglich einer Besichtigung für eine Wohnung in [Stadtname]. Nach kurzer Absprache hatte ich also einen Besichtigungstermin für Samstag. (Absatz 4/3) Am Mittagstisch haben wir als Familie ein weiteres Mal über die Besichtigungstermine unterhalten, die ich bekommen hatte. Morgen wollen wir nach [Stadtname] fahren und uns die drei Wohnungen anschauen, bis jetzt hat sich die Wohnungssuche nämlich schwierig gestaltet wegen der Pandemie, die mich als [Ausländer*in; Name des Landes] vor die Tatsache stellt, dass ich nur eine gewisse Zeit in Deutschland auf Durchreise sein darf. Ich habe mir daraufhin alles rausgesucht an Dokumenten, die ich mit zur Besichtigung nehmen wollte, habe noch alle Adressen notiert, und mir den Weg bis dorthin angeschaut, um den Durchblick zu erhalten. Am Nachmittag habe ich weiterhin auf den Immobilienseiten im Internet herum gestöbert jedoch vergeblich also habe ich auf Erfolg gehofft bezüglich der drei anstehenden Besichtigungen am Samstag. (Absatz 5/3) Am Abend habe ich mir dann noch in Moodle angeschaut, ob sich während des Tages dort etwas getan hat, habe mir einen Wecker gestellt, um zeitig aufzustehen, und nach [Stadtname] zu fahren für die Besichtigungstermine, damit ich nach [Stadtname] ziehen kann für mein Studium. (Absatz 5/4) Am Nachmittag habe ich dann den finalen Mietvertrag für meine erste Eigene Wohnung erhalten, was mir ermöglicht am kommenden Jahresanfang nach [Stadtname] zu ziehen und mein Studium dort weiterzuführen in Hoffnung auf ein normales und nicht digitales Semester an der [Universität] [Stadtname]. (Absatz 6/3) Nach der Vorlesung habe ich mich an den Mietvertrag meiner zukünftigen Wohnung gesetzt und mir diesen gründlich durchgelesen, bis ich dann unterzeichnet habe und ihn zur Post gebracht habe, damit dieser nach [Stadtname] zum Vermieter geschickt wird. (Absatz 7/2) Im Verlauf des Nachmittags, haben meine Mutter und ich und noch entschieden in nächst gelegene Stadt, [Stadtname], zu fahren und kleine Besorgungen zu tätigen, unteranderem waren auch verschiedene Sachen dabei für meine zukünftige Wohnung die ich im Januar beziehen werde, wegen meines Studiums in [Stadtname]. (Absatz 10/4)

In diesem Beispiel wird deutlich, dass die*der Studierende zwar aufgrund der Pandemie bisher Schwierigkeiten hatte, eine Wohnung zu finden, nicht aber aus finanziellen oder anderen Problemen. Vielmehr wird deutlich, dass die Unterstützung durch die Familie gegeben ist, was sich aus dem Kontext der Passage

„Am Mittagstisch haben wir als Familie ein weiteres Mal über die Besichtigungstermine unterhalten, die ich bekommen hatte. Morgen wollen wir nach [Stadtname] fahren und uns die drei Wohnungen anschauen" erschließt. Ziel der Wohnungssuche ist es, an dem Standort zu sein, an dem studiert wird, „damit ich nach [Stadtname] ziehen kann für mein Studium".

Neben den Textpassagen zur Wohnungssuche finden sich des Weiteren Textpassagen, die sich auf die Organisation des Studiums beziehen, wie zum Beispiel „habe ich noch den Stundenplan für Morgen geguckt in der [Universität]-App auf meinem Handy". Gerade zu Beginn des Studiums fällt es vielen Studierenden schwer, sich zurecht zu finden. Alles ist neu und meist wenig strukturiert. Das heißt Textpassagen, die auf die Organisation des Studiums hinweisen, können viel darüber verraten, wie gut die jeweiligen Studierenden im Studium angekommen sind. Dafür bilden wir die Kategorie „Studienorganisation" und definieren diese als: „Studierende berichten über ihre Organisation des Studiums, zum Beispiel wie sie den Überblick behalten oder wie und wo sie Studienmaterialien erhalten". Entsprechend der Definition gehen wir den Text erneut durch und kodieren die entsprechenden Textpassagen, was zum Ergebnis in Tabelle 5.4 führt.

Tabelle 5.4 Deduktive Kodierung in die Kategorie „Studienorganisation"
(Fortsetzung nächste Seite)

Kategorie (= MAXQDA Code)	Kodierte Textpassage (= MAXQDA Segment)
Studienorganisation	In den Emails habe ich noch verschiedene Bescheide zu den Fixplätzen erhalten für Module, in die ich mich für dieses Semester eingeschrieben habe. (Absatz 1/3)
	Ich habe mich daraufhin versucht bei Moodle einzuloggen, um mir nochmal anzuschauen, was mich heute in der Vorlesung erwarten wird. (Absatz 1/4)
	Als es dann [Vormittag] war, wollte ich mich bei Moodle einloggen, um zum Link des Zoom-Meetings zu gelangen, aber das hat sich als nicht möglich herausgestellt, da Moodle lahmgelegt war, durch all die neuen Studenten, die sich versucht haben anzumelden. (Absatz 1/6)
	[…] bevor ich aber schlussendlich das Licht ausgemacht habe, habe ich noch den Stundenplan für Morgen geguckt in der [Universität]-App auf meinem Handy. (Absatz 1/12)
	Um 7:30 Uhr hat mein Wecker geklingelt, ich habe mein Handy genommen und geschaut, ob ich eine E-Mail bekommen habe und meinen Stundenplan auf der [Universität]-App aufgerufen, um zu schauen welche Vorlesung ich heute als erstes habe. (Absatz 2/1)
	[…] und bei Moodle reinzuschauen, ob ich eine Benachrichtigung erhalten habe und es irgendetwas Neues gibt. (Absatz 2/2)

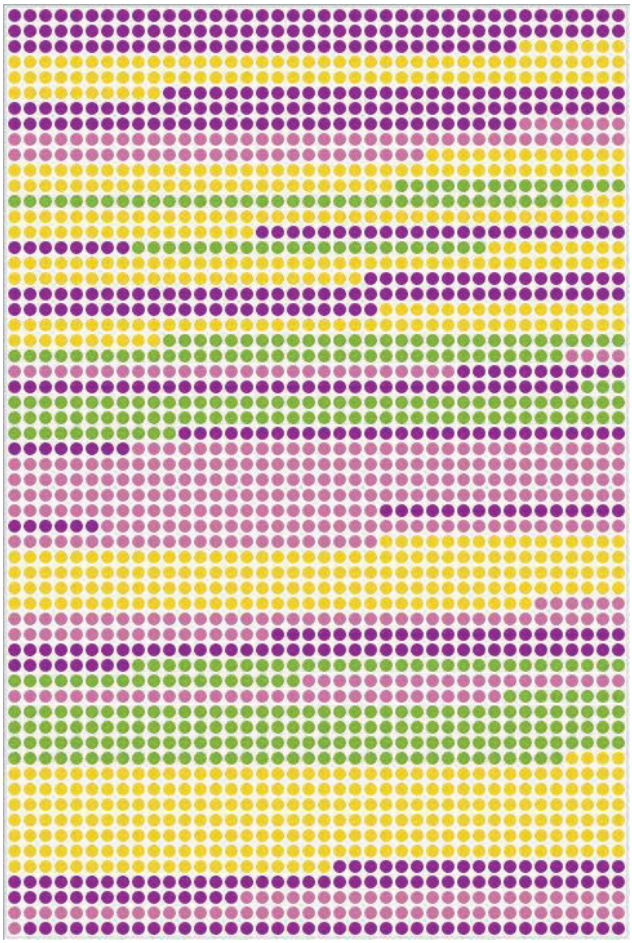
Kategorie (= MAXQDA Code)	Kodierte Textpassage (= MAXQDA Segment)
	Gegen 23:00 Uhr bin ich dann schlafen gegangen, nachdem ich mir meinen Stundenplan angeschaut hatte und mir den Wecker für 7:30 Uhr gestellt hatte. (Absatz 2/8)
	Daraufhin habe ich mir den Stundenplan herausgesucht und geschaut was noch so ansteht und was noch zeitnah gemacht werden muss. Ich finde es eine große Umstellung von meine[r Schule], den langen Ferien, jetzt wieder anzufangen mit lernen, jedoch bin ich richtig gespannt was auf mich zukommt und lasse mich gerne positiv von diesem neuen Bachelorstudium, [Hauptfach] überraschen. (Absatz 3/4 und 3/5)
	Dann habe ich bei Moodle reingeschaut, ob es dort neues gibt und mich dann an die Vorlesung von [fachfremder Zusatzkurs] gesetzt, die wirklich viel Zeit in Anspruch nimmt, da ich keine Vorkenntnisse in diesem Fach besitze. (Absatz 4/4)
	Um [Vormittag] habe ich mir einen Kaffee gemacht, mich an den Küchentisch gesetzt und mir auf meinem Handy Moodle aufgerufen, um zu schauen, ob es dort Neuigkeiten gibt. (Absatz 5/2)
	Am Abend habe ich mir dann noch in Moodle angeschaut, ob sich während des Tages dort etwas getan hat. (Absatz 5/4)
	Ich habe noch meinen Wecker gestellt und mir den Stundenplan angeschaut und bin schlafen gegangen. (Absatz 6/4)
	Um 7:45 Uhr hat mein Wecker geklingelt, bin aufgestanden und habe mir einen Kaffee gemacht. Dann habe ich mich an den großen Tisch im Wohnzimmer gesetzt und mein Laptop aufgeklappt, um meine Mails zu checken und bei Moodle reinzuschauen. (Absatz 7/1)
	Also habe ich mein Laptop aufgeklappt und bei Moodle reingeschaut, um zu sehen, was es dort Neues gibt, und mich bei der [Universität]-Mail eingeloggt, um zu sehen welche Mails ich bekommen habe. Nachdem ich eine grobe Übersicht hatte, habe ich mich an die Vorbereitung für Montag gesetzt und mir angeschaut was so ansteht die kommende Woche. (Absatz 10/2)
	Als wir wieder Zuhause waren habe ich nochmal bei Moodle reingeschaut und die neu dazu gekommenen Texte ausgedruckt und in meinem Ordner eingeordnet, damit ich alles komplett habe. (Absatz 10/5)

An den Textpassagen wird deutlich, dass die*der Studierende keine Probleme hat, sich in ihrem*seinem Studium zurecht zu finden. Aus den Textpassagen geht eine starke Strukturierung hervor, zudem werden unterschiedliche Tools, wie Moodle, Apps und Mailprogramme, genutzt, um den Überblick über das Studium zu behalten. Die einzelnen Arbeitsschritte sind organisiert, d. h. es herrscht eine hohe Selbstorganisationskompetenz vor. Das bringt uns bereits zum nächsten und letzten Schritt unseres Ablaufschemas.

5.7 Zusammenführung der Ergebnisse: Schritt 6

Im letzten Schritt der deduktiv-qualitativen Inhaltsanalyse bringen wir nun die Ergebnisse zusammen. Dazu gibt es die Möglichkeit in MAXQDA, zunächst einen Farbverlauf der kodierten Textpassagen zu erstellen, indem das Visualisierungstool „Dokument-Portrait" genutzt wird. Das „Dokument-Portrait" kann allerdings immer nur für ein Dokument erstellt werden. Abrufen können sie das „Dokument-Portrait" über das Menüband: „Visual Tools → Dokument-Portrait" oder durch die Auswahl „Dokument-Portrait" im Kontextmenü eines Dokuments (MAXQDA 2020). Das Ergebnis des „Dokument-Portraits" unseres Beispiels sehen Sie in Abbildung 5.2.

Abbildung 5.2 Dokument-Portrait der deduktiven Kodierung



Aufgrund des Farbverlaufs der Kategorien (hier werden nur die Hauptkategorien angezeigt) wird ersichtlich, dass Studienorganisation und die Interaktion mit Kommiliton*innen in der kodierten Autoethnographie am häufigsten vorkommen. Mehr Aussagekraft hat das Portrait allerdings nicht. Denn eine qualitative Inhaltsanalyse dient nicht dazu, Häufigkeitsaussagen zu machen (siehe dazu die Ausschlussregel in Kapitel 2). Für den hier analysierten Fall lassen sich folgende Ergebnisse festhalten.

- Die*der Studierende scheint gut in das Studium eingebettet zu sein, was sich daran zeigt, dass sie*er keine Probleme mit der Studierendenorganisation beschreibt. Vielmehr gibt es viele Hinweise darauf, wie sie*er durch digitale Tools den Studienalltag im Griff hat.
- Auch die soziale Integration in das Studium scheint für die*den Studierenden gut zu funktionieren, da sowohl eine Peer-Group („wir Erstis") als Referenz benannt wird als auch Lerngruppen beschrieben werden. Durch den Beitritt in eine [Studierendengruppe] findet zudem ein digitales Sozialleben statt.
- Der Sozialraum Hochschule als physischer Ort wird von der*dem Studierenden nicht benannt, was nicht verwundert, da sie*er bisher noch nicht am Campus studiert hat, sondern direkt in das Online-Semester gestartet ist. Allerdings besteht der Wunsch nach einer Einbindung an den Hochschulstandort, was sich an der (erfolgreichen) Wohnungssuche am Hochschulort zeigt.
- Mit Rückgriff auf Tinto und die existierenden Studien könnte die Ad-hoc-Hypothese generiert werden, dass die*der Studierende sehr wahrscheinlich wenig Schwierigkeiten in seinem Studium aufgrund von Passungsverhältnissen haben wird.

5.8 Vergleich der Fälle

Bei einer deduktiv-qualitativen Inhaltsanalyse wäre an dieser Stelle nicht das Ende erreicht, sondern erst der Anfang gemacht, denn diese Variante der Inhaltsanalyse lebt vom Vergleich unterschiedlicher Fälle. Erst durch den Vergleich von mehreren Fällen (die Anzahl der Fälle ist dabei von der Forschungsfrage abhängig) kämen Sie zu einem umfassenden Ergebnis ihrer Forschungsfrage. In unserem Fall würden wir das gesamte Material der autoethnographischen Aufzeichnungen (siehe Kapitel 6), also alle 50 Autoethnographien auswerten. Dabei handelt es sich um eine Textmenge von circa 250 Seiten, was mit der deduktiv-qualitativen Inhaltsanalyse zwar anspruchsvoll, aber für eine MA-Abschlussarbeit durchaus machbar ist.

Für eine BA-Arbeit könnten Sie sich überlegen, ein Sampling aus dem Datenmaterial zu ziehen, das aber gut begründet sein muss. Sie könnten zum Beispiel überlegen, dass Sie nur Fälle von Studierenden auswerten wollen, die einer

Nebentätigkeit nachgehen oder Care-Aufgaben haben und deshalb neben dem Studium eine weitere Verpflichtung haben. Die Annahme dahinter könnte sein, dass dadurch die Eingebundenheit in das Hochschulleben schwieriger ist.

Durch den Vergleich zwischen den Autoethnographien wäre es sehr wahrscheinlich notwendig, das Kategoriensystem immer weiter zu entwickeln und auf der Ebene der Subkategorien anzupassen. So müssten Sie wahrscheinlich positive und negative Subkategorien entwickeln. Also zum Beispiel unter der Kategorie „Interaktion Kommiliton*innen\Peer-Group" zwei Subkategorien bilden, die dann lauten „Interaktion Kommiliton*innen\Peer-Group\vorhanden" und „Interaktion Kommiliton*innen\Peer-Group\nicht vorhanden". Entsprechend dieser Kategorie würden auch die anderen Kategorien weitere Subkategorien erhalten. Zudem könnten durch die Kodierung der Autoethnographien weitere Kategorien hinzukommen, was dann wiederum zu einer Überprüfung der bereits kodierten Autoethnographien führen würde. Denn Sie müssen sicherstellen, dass die neu entwickelten Kategorien nicht vielleicht doch in den bereits kodierten Autoethnographien vorkommen. Es kann immer passieren, dass Sie zu Beginn einem Aspekt zu wenig Aufmerksamkeit geschenkt haben. Deshalb kann es gerade zu Beginn des Kodierprozesses sinnvoll sein, Ihr Material zusammen mit anderen auszuwerten. Damit entsprechen Sie auch dem qualitativen Gütekriterium „Validierung durch Kommunikation" im Sinne der „Transparenz der Vorgehensweisen" (Flick 2019, S. 483). Kommunikative Validierung erfolgt beispielsweise mit der*dem Betreuer*in der Qualifikationsarbeit, mit Kommiliton*innen oder mit Kolleg*innen und sollte fester Bestandteil des gesamten qualitativen Forschungsprozesses von Forschungsdesign über Materialauswahl und Erhebungsinstrumente bis zur Darstellung von Ergebnissen sein (siehe auch Strübing et al. 2018).

Hätten wir alle Autoethnographien codiert, dann könnte es uns das Dokument-Portrait ermöglichen, einen schnellen visuellen Vergleich zwischen den Dokumenten vorzunehmen, um z. B. in einen vertieften Vergleich zu einer Kategorie zu gehen. Dazu würden wir zum Beispiel ähnliche Farbschema oder sehr unterschiedliche Farbschema miteinander vergleichen, was als minimaler bzw. maximaler Vergleich bezeichnet wird (Glaser und Strauss 2008).

Durch die Vergleiche wird zudem ermöglicht, bisherige Leerstellen in theoretischen Annahmen und dem von Ihnen dokumentierten Forschungsstand aufzudecken. So könnte in unserem Beispiel der Aspekt des digitalen Sozialraums, also einem Sozialraum ohne physische Hochschule existieren, der ebenso zu einer gelungenen Einbindung von Studierenden beitragen könnte (Steinhardt 2021). Damit würde dann die Theorie erweitert werden.

6. Induktiv-quantitative Inhaltsanalyse und Auswertungstechniken am Beispiel der Kombination von AntConc und MAXQDA

In diesem Kapitel wird eine Auswertungstechnik vorgestellt, wie sie 500 Seiten und mehr Text quantitativ inhaltsanalytisch auswerten können. Die Auswertungstechnik ist geeignet für größere Interviewstudien und Sekundärauswertungen von prozessproduzierten Daten (z. B. Wahlprogramme oder Twitter). Am Beispiel einer Sekundärauswertung (Autoethnographie zu: „Wie gestalten Sie Ihren Studienalltag zu Beginn des Online-Semesters 2020/21?“) wird die sequentielle Auswertung mithilfe der Software AntConc (Freeware; Sozio-/Politolinguistik) und MAXQDA (Kaufsoftware; lexikalische Suche und Autocodierung) erklärt. Geleitet durch Erkenntnisinteresse und Forschungsfrage(n) wird Schritt für Schritt erklärt, wie der Textkorpus über Suchworte systematisch erschlossen wird, und wie darin gefundene Informationen ausgewertet werden. Die teilautomatisierte Erschließung des Textkorpus und quantitative Inhaltsanalyse produzieren einen über die Suchworte bzw. Schlagworte kategorisierten Überblick, welcher auch als „distant reading“ bezeichnet wird. Im Gegensatz zur vertieften qualitativen Inhaltsanalyse ist das Ziel der quantitativen Inhaltsanalyse, die manifesten Inhalte zu erfassen sowie daraus Erkenntnisse, Deutungen, Schlüsse und Interpretationen zu ziehen.

6.1 Einleitung

6.1.1 Begründung der Softwareverwendung

Die Auswahl von AntConc und MAXQDA für die quantitative Inhaltsanalyse hat mehrere Gründe (siehe zu Copylefts und Copyrights Kapitel 1.3). Erstens sind beide Auswertungsprogramme für verschiedene Computermodelle und Betriebssysteme verfügbar (was natürlich auch für Alternativprogramme zutrifft). Zum Beispiel können Sie AntConc kostenlos unter *www.laurenceanthony.net/software/antconc* für Macintosh OS X, Linux und Windows herunterladen und installieren. Zudem existieren für beide Programme Online-Tutorials und diverse Lern-Videos. Dabei müssen Sie bedenken, dass die Auswertungsprogramme für unterschiedliche Anwendungsbereiche entwickelt wurden. MAXQDA wurde spezifisch für die sozialwissenschaftliche Auswertung von Text aus Gruppendiskussionen, Interviews und in anderen Kontexten erstellten Texten entwickelt.

Entsprechend verfügt MAXQDA über viele inhaltsanalytische Funktionen für die Organisation des systematisch erstellten quantitativen Datenarrangements und die Analyse der manifesten Inhalte der Autoethnographien der Studierenden. AntConc ist ein Programm für Sprach- und Literaturwissenschaften, welches für die Korpus-Linguistik entwickelt wurde, jedoch ebenso für interdisziplinäre soziolinguistische Untersuchungen genutzt werden kann. Obwohl MAXQDA über ähnliche Funktionen verfügt (z. B. lexikalische Suche), sind die Analysemöglichkeiten von AntConc fokussierter auf Sprache in einem umfangreichen Textkorpus an sich. Zudem bieten die automatisiert erstellten (Zwischen-)Ergebnispräsentationen von AntConc hilfreiche Einblicke in die Daten, welche mit MAXQDA deutlich zeit- und arbeitsaufwendiger erschaffen werden müssen.

Zweitens haben beide Programme sowohl eine graphische Anwendungsoberfläche als auch einfache Menüführung, sodass Sie keine Programmier- bzw. Skriptkenntnisse benötigen wie bei einer Auswertung mit RStudio (siehe Kapitel 9) oder Topic Modeling (siehe Kapitel 11). Durch die hier vorgestellten Auswertungstechniken lernen Sie anhand der Kombination von AntConc und MAXQDA systematisch den Ablauf eines Forschungsprozesses der quantitativen Inhaltsanalyse kennen. Weiter erhalten Sie durch diese Herangehensweise wichtige Inhaltseinblicke in einen Textkorpus, denn Sie beginnen unmittelbar mit der Datenanalyse und bekommen somit ein „Gefühl" für den Inhalt (siehe auch Box 6.1).

Box 6.1: Weitere Materialien und Informationen online
Auf dem Blog „sozmethode" finden Sie Beispiele zur quantitativen Inhaltsanalyse unter *https://sozmethode.hypotheses.org/category/teilautomatisierte-inhaltsanalyse.*

Drittens können Vor- und Nachteile von Freeware (AntConc) und Kaufsoftware (MAXQDA) dargelegt werden, wobei MAXQDA für die Lehre auf Antrag durch die*den Lehrende*n Freilizenzen zur zeitlich begrenzten Nutzung für Studierende anbietet. Trotz einer kostengünstigen Lizenz für Studierende verursacht MAXQDA Kosten, welche bei frei zur Verfügung gestellter Software nicht anfallen (selbstverständlich können Sie stets mit einer Spende zur Softwareentwicklung von Freeware beitragen).

6.1.2 Erkenntnisziele der quantitativen Inhaltsanalyse

Die Verfügbarkeit von Text in digitaler Form und teilweise frei verfügbarer Analysesoftware ermöglicht und/oder reizt zu teilweise schnellen Inhaltsanalysen von Textdaten und Dokumenten jeglicher Art. Sicherlich, eine *quick-and-dirty*-Analyse unter Zuhilfenahme von Software bietet zumindest mehr Erkenntnisse als nur oberflächliches und selektives Herumstöbern in umfangreichen Textkorpora und Dokumentensammlungen. Jedoch bergen beide unsystematischen Vorgehensweisen die Gefahr von Schnellschlüssen, beispielsweise aufgrund der unvollständigen Erfassung der Inhalte. Mit (etwas) mehr Aufwand kann jedoch

durch die Kombination von softwareunterstützten Analysetechniken mit überschaubarem Aufwand eine systematische Auswertung erfolgen. Dabei ist jedoch zu bedenken, dass die schrittweise Kombination von quantitativen Auswertungstechniken die Daten neu arrangieren. Folglich muss für die Analyse gemäß der Ausschlussregel empirischer Sozialforschung berücksichtigt werden, dass die quantitative Herangehensweise und dazugehörige Auswertungstechniken keine Erkenntnisse nach dem qualitativen Paradigma ermöglichen (siehe zu Copylefts und Copyrights Kapitel 1.3), sondern Sie in einem Datenarrangement manifeste Inhalte qualifizieren durch erklären, deuten, interpretieren und Schlüsse ziehen.

Die Erklärungen, Deutungen, Interpretationen und Schlüsse, d. h. die Informationsgewinne des automatisiert erstellten Datenarrangements verbleiben auf der Ebene der manifesten Inhaltsanalyse (siehe auch Früh 2017; Hutter 2020; Rössler 2017). Folglich können quantitative Inhaltsanalysen einen Beitrag zur Beantwortung der Was-Frage und nicht der Warum-Frage von Kommunikation leisten (siehe Erinnerung in Box 6.2, Kapitel 2.1 und Tabelle 2.1). Durch die systematische Herausnahme der Textstellen anhand von Worttokens erfolgt eine Dekontextualisierung der Informationen. Worttoken ist ein Begriff aus den Sprachwissenschaften. Das englische Wort *token* bedeutet ins Deutsche übersetzt Merkmal, Zeichen und Vorkommnis. Ein Wort als empirisches Vorkommnis, also das einem Worttoken zugrundeliegende Schlagwort für die soziolinguistische Analyse der Wortebene (z. B. Niehr 2014, S. 69–75) erfüllt methodisch eine Doppelfunktion: gezielte Suche nach Information in einem umfangreichen Wort-Datensatz und für die soziolinguistische Bedeutungsbeimessung. Für die Informationssuche in den Wort-Daten werden Schlagworte bzw. deren Worttoken als Suchworte für die quantitative Inhaltsanalyse genutzt. Die schlagwortgetriebene Bedeutungsbeimessung unterscheidet zwischen der Grundbedeutung von Worten als Begriffe (linguistisch: denotative Bedeutung) und Nebenbedeutungen von Begriffen (linguistisch: konnotative Bedeutung). In der Linguistik wird weiter differenziert zwischen

Box 6.2: Erinnerung Verstehendreischritt
Analyseschritt 1: Kontext-Verstehen des untersuchten sozialen Phänomens [a) Subjekt(e), b) Objekt(e), c) Raum/Räume und d) Zeit/Perioden].

Analyseschritt 2: Inhalte-Verstehen durch Erklären, Deuten, Interpretieren und Schließen.
a) Manifester Inhalt [Was-Frage der Kommunikation (1. Sinnebene)?]
b) Latenter Inhalt [Warum-Frage der Kommunikation (2. Sinnebene)?]

Analyseschritt 3: Publikum-Verstehen [adressat*innenspezifisch Ergebnisse zusammenfassen, für Präsentation arrangieren und/oder für Publikation verschriftlichen].

> „denotativen und emotiven Bedeutungsbestandteilen. Damit soll ausgedrückt werden, dass sich die Bedeutung von Ausdrücken [bzw. Begriffen] nicht darin erschöpft, Dinge zu bezeichnen. Häufig nehmen wir gleichzeitig mit der Bezeichnung von Dingen oder Sachverhalten eine Bewertung vor. So evozieren [lateinisch für hervorrufen] bestimmte Ausdrücke spezielle Konnotationen bei den Rezipienten" (Niehr 2014, S. 67).

Die über Schlagworte bzw. Worttokens geleitete Suche und damit erschlossenen Sätze oder Absätze ermöglichen es den Forschenden, Strukturen und Muster in den Daten zu erkennen. Die quantitativ angeleitete Inhaltsanalyse ermöglicht jedoch keinen vertieften Einblick in die latenten Inhalte der einzelnen Texte des Korpus an sich – das leistet die Inhaltsanalyse nach dem qualitativen Paradigma. An dieser Stelle sei nochmals betont, dass es kein methodisches besser oder schlechter von qualitativer und quantitativer Inhaltsanalyse gibt, sondern das Erkenntnisinteresse die Methodenwahl leiten soll.

Die Beschreibung des Erkenntnisinteresses ist folglich auch der erste Schritt bei der Gestaltung des Forschungsdesigns, welches schematisch in Abbildung 6.1 dargestellt ist. Das Erkenntnisinteresse wird dafür in einer (oder mehreren) Forschungsfragen formuliert. Die explizite Niederschrift der Forschungsfrage(n) ist unerlässlich, um den Forschungsprozess für Dritte transparent bzw. nachvollziehbar zu gestalten. Eine Forschungsfrage kann durch eine in der Literatur entdeckte Forschungslücke motiviert werden, durch eine theoretische Fragestellung begründet werden und/oder ein bisher nicht untersuchtes, von der*dem Forschenden beobachtetes soziales Phänomen erfassen helfen.

6.1.3 Forschungsprozess als Schritt für Schritt-Ablaufschema

Bei einer Untersuchung mit Erhebung von Primärdaten, d. h. die*der Forscher*in erhebt selbst die empirischen Daten, leitet die Forschungsfrage den gesamten Forschungsprozess. Bei der Sekundärauswertung von bereits existierenden Daten bzw. Datenarrangements berücksichtigt die Forschungsfrage zwar den sozialen (z. B. kulturellen, ökonomischen und politischen) und (digital-)räumlichen Kontext,[1] sie ist jedoch durch die vorhandenen Daten eingeschränkt. Bei forschungsproduzierten Daten kann die Forschungsfrage der Primäranalyse auch für die Sekundäranalyse verwendet werden, wie im folgenden Beispiel des Lehr-Forschungsprojekts Autoethnographie zu „Zwei Wochen Studium im Wintersemester 2020/21“ beschrieben wird. Das Beispiel und die generierten Ergebnisse werden genutzt, um das in diesem Kapitel vorgestellte methodische Vorgehen zu illustrieren. Nach einer kurzen Darstellung des Lehr-Forschungsprojekts und der datenschutzrechtlichen Voraussetzungen orientiert sich die Kapitelstruktur am Forschungsdesign, welches schematisch in Abbildung 6.1 dargestellt ist.

1 Unter (digital-)räumlichen Kontext verstehen wir die physische bzw. die online konstruierte Sprechsituation, in der eine Kommunikation stattfindet.

Abbildung 6.1 Ablaufschema einer quantitativen inhaltsanalytischen Kombination von Auswertungstechniken

Schritt 1	Erkenntnisinteresse als Fragestellung formulieren. Für deduktive, d. h. nicht rein datengetriebene Forschung, theoriegeleitet Hypothesen formulieren.
Schritt 2	Datenannäherung: Lesen eines Teils der Dokumente im Textkorpus und Notizen in Forschungstagebuch anfertigen, um ein „Gefühl" für Daten und Inhalte zu erlangen.
Schritt 3	Daten für entsprechende Software vorbereiten (Datenbereinigung): • Dateiformat anpassen • Dokumente bereinigen (z. B. nicht lesbare Symbole usw. entfernen)
Schritt 4a	Quantitative Inhaltsanalyse mit AntConc: • Alle Dokumente des Korpus importieren (idealerweise als separate Analyseeinheiten) • Wortliste erstellen • Sinn-Wörter identifizieren und in Worttokens umwandeln (z. B. durch Lemmatisierung und Ergänzung benachbarter Wörter) • Worttokens zur Analyse des Textkorpus anwenden • Zwischenergebnisse sichern • Worttokens als Wortlisten • *Keywords in context* (KWIC) • Konkordanz-Diagramme
Schritt 4b	Quantitative Inhaltsanalyse mit MAXQDA: • Alle Dokumente des Korpus importieren (idealerweise als separate Analyseeinheiten) • Mit AntConc identifizierte Worttokens für lexikalische Suche und anschließende Autokodierung verwenden • Überprüfung der Ergebnisse der Autocodierung und gegebenenfalls Überarbeitung (z. B. der Codes oder des automatisch erfassten Textausschnitts) und Wiederholung Ergebnisse sichern: • Codeschema (inkl. Codehäufigkeiten) und Codewolke als Wordle (für Ergebnispräsentation) • Ausgabe der codierten Textstellen
Schritt 5	Auswertung des Datenarrangements der automatisch codierten Textstellen: • Manifeste Inhalte (Was-Frage beantworten als Analyseschritt 2a; für jeden Code getrennt wiederholen) • Theoriebasierte Reflexion der Ergebnisse (bei deduktiver Herangehensweise), oder • Theorieentwicklung (bei induktiver Herangehensweise)
Schritt 6	Ergebnisse zusammenfassen und Erstellung einer Präsentation, Studienarbeit und/oder Publikation
Schritt 7	Sichere Archivierung der Daten und, wenn möglich, Aufbereitung zur Nachnutzung

6.2 Die Textdaten

6.2.1 Lehr-Lern-Forschungsprojekt Autoethnographie „Zwei Wochen Studium im Wintersemester 2020/21“

Das Lehr-Lern-Forschungsprojekt Autoethnographie zu „Zwei Wochen Studium im Wintersemester 2020/21“ war Teil einer Lehrveranstaltung an einer Universität in Deutschland. Angesichts der Covid-19-Beschränkungen konnte Lernen durch Anwenden von sozialwissenschaftlichen Methoden nicht draußen im Feld erfolgen. Wie bei Tagebucheinträgen dient die Autoethnographie der methodisch angeleiteten Auto-, also Selbstbeobachtung (Ellis et al. 2010). Die Selbstbeobachtung dient der Erforschung und Reflexion eigener Tagesabläufe, (neuen)

Tabelle 6.1 Merkmale von autoethnographischen Daten (nach Kategorien in Tabelle 3.1)

	Forschungsproduzierte Daten *Groß-n* Fallzahlen Untersuchung
Konzeption	
Datentypen	Autoethnographie bzw. Selbstbeobachtung
Ziel	Beantwortung der Forschungsfrage
Operationalisierung	Deduktive Reflexion oder induktive Schaffung theoretischer Basis
Datengenese	
Stichprobenziehung	Vollerhebung bei Studierenden in Methodenvorlesung (Übung zum Anwenden-Lernen), welche durch Datenbereitstellung der Studierenden (Selbstselektion) zur Stichprobe wurde
Datenerhebung	Subjekte (= Studierende) und soziales Phänomen (= Beginn Onlinesemester)
Datenart	Text
Archivierung	
Vollständig	Ja
Analysen	
Datenverwahrung	Wissenschaftler*innen
Datenanalyse	Quantitative Analysemethode
Wissenschaftliche Analyseart	Sekundärauswertung
Anwendung	
Ergebnis	Wissenschaft (und teilweise praktische) Implikationen (z. B. Handlungsanleitung)/Schlussfolgerungen

Mustern und von (nicht) bewusst wahrgenommenen Ritualen. Das inhaltliche Interesse der autoethnographischen Selbstbeobachtungen war, wie die Anfang November 2020 aufkommende zweite Corona-Welle das Studium und, für die neuen Studierenden, den Studieneinstieg prägt. Die untersuchungsleitende Fragestellung für die Studierenden lautete: „Wie gestalten Sie Ihren Studienalltag zu Beginn des Online-Semesters 2020/21?" Die Aufgabenstellung war, dass die Studierenden in einem Zeitraum von 14 Tagen etwa zehn autoethnographische Aufzeichnungen verfassen sollten. Zur Orientierung erhielten die Studierenden einen Beispieltext einer autoethnographischen Aufzeichnung aus einem anderen Forschungsprojekt, ergänzend zu Methodenliteratur, Erklärungen zu Autoethnographie in einem Video (verfügbar bei Moodle) und der Anleitung durch die*den Lehrenden in der Vorlesung.

Methodisch ist bei der Verwendung der Autoethnographie als Beispiel für die quantitative Inhaltsanalyse folgendes zu beachten: Erstens wird aus der qualitativen Methode der Autoethnographie eine quantitative Textanalyse. Zweitens erfolgt durch die Sekundäranalyse eine Entkoppelung der Erhebungsmethode (= Tagebuchschreiben der Studierenden) und der Auswertungstechnik (Tabelle 6.1).

6.2.2 Datenschutz und Einwilligungserklärung

Für die Sekundäranalyse können die Textdaten der Studierenden nicht ohne deren Einwilligung und nur nach Anonymisierung beispielsweise von Namen von Personen und Orten usw. verwendet werden. Informationen über die geplante Verwendung und Anonymisierung der Daten wurden den Studierenden in einem Handzettel ausgehändigt. Das Informationsblatt wird Handzettel genannt, da es den Teilnehmer*innen der Untersuchung bündig, auf einer, maximal zwei Seiten oder weniger, wichtige Informationen vermitteln soll (siehe Kurzdefinition in Box 6.3 und Beispiel in Anhang 1 in Kapitel 3).

Box 6.3: Kurzdefinition Handzettel
Handzettel:
a) Kurzvorstellung Forschende und Institution, und Kontaktangaben (z. B. E-Mailadresse und Telefonnummer).
b) Untersuchungsgegenstand klar benennen, ohne jedoch Antworterwartungen kenntlich zu machen.
c) Untersuchungspopulation benennen und erklären, warum angesprochene Personen untersucht werden und für ein Interview zur Verfügung stehen sollten.
d) Erwarteten Aufwand ehrlich angeben, sonst ist es wahrscheinlich, dass Datenerhebung unvollständig ist (z. B. Interviewte reservieren zu wenig Zeit).
e) Verwendung der Interviewdaten spezifizieren, zum Beispiel für Forschung, Veröffentlichung, Bericht usw., sowie unbedingt auf Anonymisierung und Datenschutz hinweisen.

Auf dem Handzettel sind besonders Informationen zum Datenschutz hervorzuheben, d. h. dem Umgang mit den personenbezogenen Daten durch Forschende. Auch wenn beim Datenschutz auf den rechtlichen Rahmen verwiesen wird, so ist es essenziell, eine gut verständliche Sprache zu verwenden. Die Verständlichkeit der Erklärungen ist umso wichtiger, als Teilneh-

mer*innen an der Untersuchung explizit schriftlich ihre Zustimmung durch Unterschrift geben müssen. Dies wird als Einwilligungserklärung zur Verarbeitung und gegebenenfalls Weitergabe der Daten bezeichnet (siehe Anhang 3.2). Von den [Anzahl der angemeldeten Studierenden zwecks Anonymisierung gelöscht] Studierenden der „[Name der Veranstaltung]" haben 50 die Einwilligungserklärung unterschrieben, dass ihre Autoethnographie für Forschung und Lehre verwendet werden darf. Die Zustimmung beinhaltet die Anonymisierung von Personen-, Studienfach-, Orts- usw. Namen (zu Anonymisierung siehe Kapitel 4.2.2).

6.2.3 Rohdaten und Datenbereitung

Die 50 Autoethnographien der Studierenden bilden die Datengrundlage für das Beispiel, anhand dessen die quantitative Inhaltsanalyse erklärt wird. Die Studierenden haben überwiegend ihre Autoethnographie als Worddatei zur Verfügung gestellt. Trotz der Bitte um Übergabe der autoethnographischen Aufzeichnungen in einer Word-Datei, haben wenige Studierende diese als PDF-Datei übergeben. Erfahrungsgemäß trat ein Problem mit Zeilenumbrüchen im Text auf, welches durch veraltete PDF-Versionen durch die „Als Text speichern …"-Konvertierung hervorgerufen wird. Nicht elegant, jedoch relativ zeiteffizient ist die manuelle Bereinigung der Daten bei der überschaubaren Textmenge. Hier können Sie auf dem Bildschirm das Original und die elektronische Datei nebeneinander anordnen und in einem konvertierten Dokument (z. B. txt- oder Word-Datei) von unten mit der Bereinigung anfangen, da die Zeilen immer gleich lang sind. Beispielsweise den Zeigefinger der linken Hand auf die Nach-oben-Pfeil-Taste legen und abwechselnd mit Zwei-Finger-Technik der rechten Hand mit Daumen auf der Taste „Leerzeichen" und dem kleinen Finger auf der „Entf"-Taste gleichbleibendem Rhythmus arbeiten. Das ist eine Fleißarbeit, welche gegebenenfalls leichter bei Musik erfolgt; erfahrungsgemäß ist es empfehlenswert bei längeren Dokumenten regelmäßig eine Pause einzulegen, um einem Krampf in der Hand vorzubeugen 😊. Bitte erzeugen Sie keinen Fließtext, d. h. stellen Sie die Absätze wieder her, da sonst in MAXQDA bei der Autokodierung die Funktion den „gesamten Absatz erfassen" nicht verwendet werden kann.

Wesentlich eleganter können die verlorengegangenen Zeilenumbrüche sowohl im OpenOffice Writer als auch in Word wiederhergestellt werden. Ein Zeilenumbruch kann im OpenOffice Writer automatisiert mit der „Suchen und Ersetzen" Funktion einfach ersetzt werden. Das Dollarzeichen „$" ist im Suchen-Feld der Befehl für „Zeilensprung ersetzen durch", wobei im Ersetzen-Feld dann ein „[Leerzeichen]" eingegeben werden muss.[2] Für die Bereinigung größerer Do-

2 Andersherum kann durch Eingabe des Befehls „\n" im Ersetzen-Feld ein Zeilenumbruch erzeugt werden.

kumentenkorpusse ist die Anwendung eines Python Skripts zu empfehlen, wobei fertige und einfach anzuwendende Skripte über eine Suchmaschine gefunden werden können, welche auch fehlerhafte Zeichen korrigieren (z. B. Softwareanwendung icon-v-loop).

Eine Fehlerquelle bei der Erstellung von Suchwortlisten für die quantitative Erschließung eines Textkorpus bietet auch die alte und neue Rechtschreibung. Beispielsweise eher ältere Semester verwenden gewohnheitsmäßig statt dem Doppel-„S" der neuen Rechtschreibregeln weiterhin das Scharf-„ß". Auch ist zu bedenken, dass es keine einheitliche Umsetzung der neuen Rechtschreibregeln im deutschen Sprachraum gibt. Während in Deutschland Doppel-„S" und „ß" parallel existieren, wurde in der Deutschschweiz und Österreich konsequent auf Doppel-„S" umgestellt, sodass beispielsweise der deutsche „Spaß" (mit gut nachvollziehbar festgelegten neuen Rechtschreibregeln), ein „Spass" in der Deutschschweiz und Österreich ist – aufgrund des quantitativen Paradigmas der standardisierten Sozialforschung ist der konsequenten Vereinheitlichung zu applaudieren.

6.2.4 Informationen in umfangreichen Textkorpora finden

Warum die Datenbereinigung so einen zentralen Stellenwert hat, zeigen wir Ihnen anhand von zwei Wordles (auf Deutsch: Wortwolken). In Abbildung 6.2 wurden alle Worte des Textkorpus visualisiert. Deshalb sehen Sie vor allem (Possessiv-) Pronomen, Binde-, Füll-, Modal- und viele mehr Worte, welche wichtig in der Kommunikation sind, jedoch nicht den Sinn oder den Bedeutungsrahmen des Textkorpus erkennen lassen. Zwar findet die*der aufmerksame Beobachter*in im Wordle zweimal das Wort „vorlesung" – einmal im Singular neben dem zentralen „ich", welches typisch für autoethnographische Daten ist, und einmal im Plural am rechten Rand im rechten Winkel zu „machen" –, welche lediglich durch die Worte „moodle", „text" und „uhr" auf den manifesten Textinhalt der Kommunikation zu „Studium" im Textkorpus schließen lassen.

Das Substantiv „vorlesung" ist hier klein geschrieben, da bei der Datenbereinigung für gewöhnlich alle Worte – auch Nomen und Namen – auf Kleinschreibung umgestellt werden (siehe Kapitel 6.3.1). Auf diese Weise können zwei Probleme adressiert werden. Einerseits eine gegebenenfalls falsche Schreibweise (z. B. können Nomen klein, Verben großgeschrieben sein), andererseits dient dies der Vorbereitung zur Lemmatisierung oder zum *Stemming,* bei dem die verschiedenen Wortarten auf den gemeinsamen Stamm reduziert werden (siehe Kapitel 6.4.2). Andernfalls bekommen Sie mehrere Tokens heraus, die das Gleiche bedeuten, aber mal groß, mal klein geschrieben wurden.

Im Gegensatz zu den unvollständigen Eindrücken zum Inhalt des Textkorpus (Abbildung 6.2) bietet die auf Basis einer systematischen, quantitativen Inhalts-

Abbildung 6.2 Wordle gesamter Text Autoethnographien

Quelle: Autoethnographie WS 2020/21

Abbildung 6.3 Kode-Wortwolke

Quelle: Autoethnographie WS 2020/21

Tabelle 6.2 Die jeweils 25 häufigsten Wörter und Worttokens im Textkorpus

Wörter			Worttokens bzw. Schlagworte		
Rang	Häufigkeit	Token	Rang	Häufigkeit	Token
1	8 551	ich	1	1 279	vorlesung
2	4 875	und	2	380	seminar
3	3 966	die	3	321	zoom
4	2 754	der	4	284	moodle
5	2 673	zu	5	265	uni
6	2 062	um	6	261	text
7	1 990	habe	7	228	studium
8	1 950	mir	8	164	thema
9	1 921	in	9	155	arbeit
10	1 898	mich	10	154	übung
11	1 625	das	11	142	semester
12	1 559	den	12	129	veranstaltung
13	1 474	uhr	13	126	sitzung
14	1 379	mit	14	120	video
15	1 329	es	15	119	gruppe
16	1 279	vorlesung	16	109	dozent
17	1 224	nicht	17	106	laptop
18	1 212	auf	18	104	themen
19	1 200	noch	19	102	lese
20	1 174	da	20	96	professor
21	1 172	für	21	64	corona
22	1 158	an	22	63	modul
23	1 125	wir	23	56	mathe
24	1 042	meine	24	48	fach
25	1 018	auch	25	47	buch

Quelle: Autoethnographie WS 2020/21

analyse in MAXQDA erstellte Codewolke die wesentlichen manifesten Inhalte des Textkorpus ab (Abbildung 6.3). Zentral ist das Thema „Vorlesung" in verschiedenen wissenschaftlichen Disziplinen (z. B. „[Studienfach 1]" und „[Studienfach 2]") der Geistes- und Sozialwissenschaften (z. B. „[Studienfach 8]" und „[Studienfach 6]") an einer „Universität". Für die Interpretation des Inhalts des Textkorpus bieten Worttokens wie „Corona/Covid-19", „(Erklär-)Video" und „Zoom" jedoch nur in Verbindung mit der Kontextinformation „Wintersemester 2020/21" einen Hinweis darauf, dass es sich um Texte zu einem Online-Semester handelt.

Die Wordles in Abbildungen 6.2 und 6.3 basieren im Kern auf quantifizierten, d. h. ausgezählten, Wortlisten, wobei Abbildung 6.2 Worthäufigkeiten und Abbildung 6.3 Häufigkeiten von Kategorien wiedergibt. Den Ursprung der Wordles bilden einfache Auszählungen von Worten im Textkorpus in AntConc. Wie in Tabelle 6.2 selektiv dargestellt – der Textkorpus besteht aus etwa 10 000 unterschiedlichen und insgesamt etwa 150 000 Worten (etwa 500 A4 Seiten zu je 300 Wörtern) –, sind je nach Textkorpus unterschiedliche Worte häufig. Das Wort „ich" als mit Abstand häufigstes Wort ist erklärbar durch die für Autoethnographien typische Ich-Erzähler*innenperspektive. Jedoch ist auffällig, dass beispielsweise Präpositionen wie „die" und „der" weniger als halb so häufig im Textkorpus vorkommen und Sinnworte wie „vorlesung" nur 1 279-mal.

6.3 Schlag- bzw. Suchworte im Korpus mit AntConc identifizieren

6.3.1 Grundeinstellungen AntConc

Wie andere Freeware kann AntConc *as it is* und ohne Haftungsübernahme durch die*den Programmierer*in heruntergeladen werden. Bei AntConc bedeutet *as it is* beispielsweise, dass nur txt-Dateien für die Analyse eingelesen werden können. Falls Sie Word-Dateien haben, können diese in Word über „Datei speichern unter …" durch die Auswahl des Dateityps „Nur Text" in txt-Dateien konvertiert werden.

As it is, bietet AntConc eine überschaubare Anwendungsoberfläche und speichert keine Analysen. Folglich müssen sie die Einstellungen, beispielsweise für die Erkennung von westeuropäischen, auf Latein basierenden Sprachen bei jedem Neustart durch Anklicken im Menü einstellen: unter „Global Settings → Character Encoding" müssen Sie bei „Current Encoding" die „Edit-Taste" drücken und im Drop-down-Fenster bei „Standard Encodings" → „Western Latin1 (iso-8859-1)" auswählen. Danach im Fenster unten die „Apply-Taste" anklicken. Wenn Sie vergessen, die Spracheinstellung zu konfigurieren, dann werden deutsche Sonderzeichen wie „ä", „ö", „ß" usw. im Text nicht erkannt. Bei wiederholter

Benutzung von AntConc machen Sie die Spracheinstellungen (fast) automatisch – dieser Schritt dauert dann weniger als fünf Sekunden.

Box 6.4: GO ONLINE for AntConc
Eine detaillierte Beschreibung der Benutzungsoberfläche, des Aufbaus und der Anwendungsmöglichkeiten von AntConc finden Sie unter *https://sozmethode.hypotheses.org/1240*.

Die überschaubare Anwendungsoberfläche von AntConc ist jedoch sehr übersichtlich (Box 6.4). Die zu analysierenden Dateien des Textkorpus werden in der linken Spalte *Corpus Files* sichtbar, wenn Sie über „File" (Datei) → „Open File(s)"… (Datei(en) Öffnen …) → aus dem entsprechenden Ordner auf Computer die zu importierende txt-Dateien (AINSI formatiert; z. B. im Editor) markieren (z. B. durch Halten der Strg-Taste) → „Öffnen klicken". Alle Analyseinstrumente (Tools) sind als Reiter (Tabs) im rechten Fenster von AntConc erreichbar. Zu jedem Tool gibt es spezifische Einstellungen, welche Sie über Tool „Preferences" manipulieren können. Die für die Auswertung der Autoethnographien der Studierenden genutzten Tools werden weiter unten mit Anwendungsbezug erklärt.

6.3.2 Funktionen für die Identifikation von Schlagworten als Suchworte

Über Worttokens können die manifesten Inhalte im Kontext für die Metaanalyse erschlossen werden. Wie oben in Kapitel 6.1.2 erklärt, erfüllen Worttokens methodisch eine Doppelfunktion: Erstens sind Worttokens zentral für die soziolinguistische Bedeutungsbeimessung, d. h. die Analyse der manifesten Inhalte nach Abschluss der empirischen Erschließung des Textkorpus (Schritt 5 in Abbildung 6.1). Zweitens werden Worttokens für die gezielte Suche nach Informationen in einem umfangreichen Wort-Datensatz eingesetzt. Folglich werden für die quantitative Suche nach Informationen die Schlagworte als Suchworte definiert. Um die zentralen Suchworte für die Analyse zu identifizieren, gibt es mehrere Analysefunktionen von AntConc.

- *Wortlisten (word list):* Nach empirischem Vorkommen geordnete Wortlisten bieten eine erste quantitative Orientierungsmöglichkeit in großen Textkorpora (siehe Tabelle 6.3). Durch die Ordnung der Wortlisten können die Suchworte für die gezielte quantitative Inhaltsanalyse identifiziert werden. Wie in Abbildung 6.2, ist es dafür empfehlenswert, die Wortliste ohne Beachtung der Groß- und Kleinschreibung *(case-insensitive)* zu erstellen, d. h., dass Wörter in Groß- und Kleinschreibung gleichbehandelt werden (ist voreingestellt bei „Tool Preferences" → „Word List Preferences" → bei „Other options" das Häkchen bei „Treat all data as lowercase"). Je nach Wörter- bzw. Komposita-Suche – Komposita sind zusammengesetzte Wörter (z. B. Prüfungsordnung) –, können Wortlisten nach abfallenden oder steigenden Häufigkeiten sowie

nach Anfang oder Ende des Wortes sortiert werden (AntConc-Taste „Ordered by …"). Die Erstellung einer Suchwortliste kann in AntConc durch folgende Funktionen unterstützt werden.[3]

- *Wortgruppen (word clusters):* In den ausgewählten Textdateien wird eine (z. B. nach Häufigkeiten) geordnete Liste von Wortgruppen erstellt, die im Text neben dem Suchbegriff vorkommt. Aus den Daten konnten so Suchbegriffe *(search strings)* überprüft und gegebenenfalls modifiziert und durch Synonyme ergänzt werden.
- *Benachbarte Wörter (collocates):* Eine geordnete Liste von Nachbarwörtern des Suchbegriffs zu erstellen, bietet eine Alternative oder Ergänzung zur Wortsuche mit Wortgruppen.
- *Konkordanz (concordance):* Das Werkzeug erstellt Konkordanz-Zeilen zu Suchworten im Kontext (KWIC: **k**ey **w**ord **i**n **c**ontext) in ausgewählten Texten. Für die Inhaltsanalyse bedeutet Konkordanz die Erstellung einer alphabetisch geordneten Liste der Suchwörter in einem Dokument nach dem Kriterium der Übereinstimmung.

Nach dem Löschen von Füll-, Binde- usw. Wörtern kann die Wortliste in den genannten vier AntConc-Funktionen zur weiteren Datenbereinigung genutzt werden. Beispielsweise zeigt die geordnete Wortliste 77 Treffer zu „schwer", was basierend auf Ihrem Kontextwissen zum untersuchten sozialen Phänomen (Online-)Studiensemester ein zu erwartendes und sinnvolles Wort ist. Die automatisierte Wörtersuche im Tool „Concordance" zum „Search Term" „schwer*" erfasst durch die Ergänzung um das Sternchen * jedoch unterschiedslos alle Worte, welche mit „schwer …" beginnen und selektiert nicht nach inhaltlichen Kriterien. Folglich präsentiert die Konkordanz diverse Treffer zu was schwer fällt und auch womit sich Studierende im Universitätsalltag nicht „schwergetan" haben (*KWIC Hit* 78 in Tabelle 6.3). Durch die Manipulation der Einstellung „Search Window Size" von 50 auf 100 Zeichen (siehe Abbildung 6.4) zeigt AntConc genügend Kontextinformationen, um ein Gefühl für die verschiedenen Kontexte zu bekommen

3 Eine weitere hilfreiche Funktion von AntConc für die quantitative Inhaltsanalyse ist die Erstellung einer Schlüsselwortliste (Keyword List) durch den Vergleich mit einem Referenz-Korpus (Reference Corpus). Dabei wird die Liste der Schlüsselwörter, die ungewöhnlich häufig oder selten sind, in den Textkorpora verglichen. Als statistisches Maß für die Identifikation, ob ein Wort im analysierten Textkorpus als Schlüsselwort zu betrachten ist, sind Chi-Quadrat-Test (χ^2; Chi-Squared) und Plausibilitäts- oder Mutmaßlichkeitsfunktion (Log Likelihood) voreingestellt. Damit bietet es eine grundlegende Funktion zum Test, inwiefern bestimmte Wörter unabhängig voneinander auftreten, oder ob das Auftreten der Schlüsselwörter einer unterliegenden Struktur folgt. Diese Logik findet bei der Korrespondenzanalyse in Kapitel 9 Anwendung. Für Autoethnographien von Studierenden zum Semesterstart 2020/21 konnte kein Referenzkorpus gefunden werden. Jedoch können als Kontrollgruppen beispielsweise ähnliche Daten von Twitter, von Wahlprogrammen im Vergleich zu Zeitungsartikeln zur aktuellen Politik usw. verwendet werden.

und klassifizieren zu können, was Studierenden zu Beginn des Online-Semesters 2020/21 „schwer“ fiel.

6.3.3 Identifikation von Schlagworten als erster Schritt der Analyse

Das Beispiel zeigt, dass Sie bereits nach kurzer Zeit mit der Datenanalyse begonnen haben (im Ablaufschema Schritt 4a; Abbildung 6.1), wobei Datenanalyse noch bedeutet, ein Gefühl für die Daten bekommen. Sie befinden sich schließlich inmitten eines quantitativ angeleiteten Datenordnungsprozess. Wäre der Textkorpus umfangreicher, so müssten Sie in einem nächsten Datenbereinigungsschritt alle Informationen zu „nicht schwer“ usw. entfernen, um nur Textstellen mit der Bewertung „schwer“ in der empirischen Analyse zu berücksichtigen. Hier zeigt sich eine Begrenzung von AntConc, welche Sie bei Freeware in unterschiedlichen Formen antreffen werden. Wie oben erklärt, kann AntConc die Analyse nicht speichern. Jedoch bietet die hilfreiche Einfachheit von AntConc eine flexible Lösung an.

Sie die Konkordanz-Auswertung wie in Tabelle 6.3 (nächste Seite) in AntConc kopieren und dann in ein anderes Programm einfügen (z. B. Microsoft Excel oder OpenOffice Calculator). Dazu klicken Sie im Reiter Konkordanz einfach irgendwo in das Feld „Hits“ (= Treffer, welche als Verweisstelle genutzt werden können), halten die Taste „Strg“ oder „Apfel“ gedrückt, während Sie zuerst durch Drücken der Taste „A“ alles markieren, wodurch die „Hit-Spalte“ blau hinterlegt wird, dann durch das Drücken der Taste „C“ den gesamten Inhalt kopieren und durch Drücken der Taste „V“ in das von Ihnen gewählte andere Programm einfügen – erst jetzt lassen Sie die Taste „Strg“ oder „Apfel“ los!

In Tabelle 6.3 wird eine Beschränkung von AntConc sichtbar. Beispielsweise bei den „Hits“ 448 und 522 beginnt der Textausschnitt zur Kontextualisierung der Suchworte mit einem Buchstaben bzw. mitten im Wort. Entsprechend der Manipulation der Einstellung „Search Window Size“ wird entsprechend der eingegebenen Zeichenzahl eine Ausgabe erstellt. Wie unten dargestellt wird, ist ein Vorteil der lexikalischen Suche und anschließenden Autokodierung in MAXQDA, dass bei den Einstellungen angegeben werden kann, dass der gesamte Absatz um das Suchwort mitkodiert wird.

Bitte passen Sie auf, dass Sie sich nicht in den interessanten Ergebnissen verlieren. Zwar ist eine Auswertung kein Wettrennen, doch es ist davon auszugehen, dass Sie einen begrenzten Zeitrahmen haben – entweder von Ihnen selbst oder durch eine Abgabefrist gesetzt. Folglich sollten Sie die ersten Einblicke in Ihr digitales oder analoges Forschungstagebuch eintragen, und an der Systematisierung der Suchworte für die Auswertung mit MAXQDA weiterarbeiten.

Tabelle 6.3 Ausgewählte Beispiele der Konkordanz zum Suchwort „schwer*“ im Kontext (KWIC: **k**ey **w**ord **i**n **c**ontext)

Hit	Sentence 1	KWIC	Sentence 2	File (Dateinamen anonymisiert)
598	-Plattformen oder über WhatsApp recht viel Kommunikation/Austausch statt, es ist aber für mich sehr	schwer	, wirkliche Beziehungen aufzubauen. Währenddessen schreibe ich noch mit einem guten Freund über mein	Dateiname_170603.txt
498	ich denn studiere und ich antworte [Studienfach 1]. Viele wirken erstaunt und behaupten, es wäre ein sehr	schwerer	Studiengang bei dem man am Ende kaum verdienen würde. Was aber meiner Ansicht nach nicht stimmt.	Dateiname_170243.txt
522	mer, darum musste ich meine Diät ändern. Wäre dieses Semester mit Präsenz gewesen, wäre es deutlich	schwerer	und zeitaufwendiger gewesen Mahlzeiten vorzubereiten. Ich checke meine Mails. Ich gehe auf Moodle u	Dateiname_141827.txt
130	mich auf die zwei Veranstaltungen. Ich bin unfassbar erschöpft. So sehr, dass es mir gerade sogar	schwerfällt	, diesen Eintrag zu schreiben. Ich bin heute Morgen um fünf vor sieben aufgewacht und bin direkt	Dateiname_153929.txt
448	s der Austausch leichter fällt. Ich bin auch erleichtert zu hören, dass es den anderen auch manchmal	schwerfällt	, richtige Formulierungen und genügend Inhalt zu finden. Ich bin schon sehr gespannt, wie wir unsere	Dateiname_118438.txt
375	auch Minuten zurückspulen, da ich vieles auf Anhieb nicht verstanden habe und es mir auch sehr	schwerfiel	, nebenbei mitzuschreiben, da alles viel zu schnell ging. Zudem war meine Konzentration auch nicht m	Dateiname_172871.txt
78	ich einen Text und fasse hierbei die wichtigsten Punkte zusammen. Ich habe mich dabei nicht sehr	schwergetan	, da ich den Text sehr einfach fand. Daher hat es mir sehr Spaß gemacht den Text	Dateiname_172871.txt

Quelle: Autoethnographie WS 2020/21

6.4 Quantitative Analyse mit MAXQDA

6.4.1 Dateien in MAXQDA importieren

Bevor Sie weiterarbeiten, müssen Sie alle zu analysierenden Dokumente in MAXQDA importieren. Im Beispiel Autoethnographien der Studierenden handelt es sich um 50 Dokumente. Durch den Import einzelner Dokumente können Sie Vergleiche auf Dokumentebene durchführen. Zum Öffnen der Dokumente gehen Sie bitte

1. im MAXQDA Menü zum Tab „Importieren",
2. klicken das erste Icon „Texte, PDFs, Tabellen" an,
3. wählen auf Ihrem Computer die zu analysierenden Dokumente durch Markieren im jeweiligen Dateiordner aus, und
4. schließen den Importvorgang durch Anklicken des Buttons „Öffnen" ab.

Die zu analysierenden Dokumente erscheinen in der MAXQDA-Standardeinstellung im Fenster oben rechts („Liste der Dokumente"). Grundsätzlich müssen Sie für die Datenanalyse im Fenster oben links „Liste der Dokumente" alle Dokumente mit der „Aktivieren"-Funktion auswählen, indem Sie Sie bitte entweder

1. im Feld links oben („Liste der Dokumente") entweder ein Dokument anklicken oder vor die Dokumente auf die sogenannte Wurzel klicken,
2. über die Tastenkombination „Strg + a" alle zu analysierenden Dokumente markieren,
3. mit der rechten Maustaste auf die markierten Dokumente klicken, und im sich öffnenden Befehlsfenster „aktivieren" anklicken oder
4. am schnellsten durch Klick auf das Symbol links neben „Dokumente".

Nun werden alle ausgewählten Dokumentennamen rot angezeigt (siehe auch Erklär-Video auf MAXQDA Webseite).

Doch bevor Sie mit der Analyse der Textdaten beginnen, müssen die Suchworte für die Verwendung als Kategorien und für das Kodieren vorbereitet werden. Dafür bietet MAXQDA zwei Funktionen: erstens, das MAXDictio (siehe Menüleiste), welches es Ihnen erlaubt, sogenannte Diktionäre, also Wörterbücher, zu erstellen und zu verwalten. In einem Diktionär können Sie Kategorien und dazugehörige Suchbegriffe definieren. Etwas umständlicher ist die Verwendung der zweiten, der lexikalischen Suche und darin integrierten Autokodierfunktion. Auf Letztere wird im Folgenden fokussiert, um die einzelnen Schritte und Herausforderungen für den Forschungsprozess zu erklären.

6.4.2 Suchworte in Kategorien und Kodes überführen

Kodes werden durch die*den Forscher*in gebildet. Dazu werden die Such- bzw. Schlüsselworte (Worttokens) für einen Textkorpus nach einer systematischen Durchsicht neu geordnet. Hierfür kann die Wortliste aus AntConc beispielsweise in Microsoft Excel oder den OpenOffice Calculator kopiert werden. Es ist empfehlenswert, die Rohdaten, d. h. die Gesamtwortliste, in einem Datenblatt oder gar einer separaten Datei zu speichern, sollte etwas schiefgehen. Im Calculator oder Excel-Arbeitsdatenblatt können in einem ersten Ordnungsschritt alle Binde-, Füll-, Modal- usw. -Worte entfernt werden. Das selektive Ergebnis ist in Spalte B der Tabelle 6.2 abgebildet. Die Häufigkeit der Sinnworte weist teilweise eine Ähnlichkeit mit den Worttokens in der Codewolke in Abbildung 6.3 auf. Die Qualität der Information ist jedoch sehr unterschiedlich. Die Worte in Spalte B der Tabelle 6.3 sind nur eine Ausprägung des Wortes, worauf beispielsweise die Worte „vorlesung" im Singular und „vorlesungen" im Plural im Wordle zum gesamten Text der Autoethnographien hinweisen (Abbildung 6.2). Selbstverständlich können Sie weiter händisch die Liste durchgehen und sämtliche Worte in all ihren Variationen zusammensuchen und die Häufigkeiten addieren. Das ist sehr aufwendig.

Ergänzend zu Plural und Singular bestehen für die lexikalische Suche und Autokodierung in MAXQDA weitere Herausforderungen. Die deutsche Sprache erzeugt durch die Fälle (z. B. Akkusativ, Dativ und Genitiv) und die fast unbegrenzten Möglichkeiten der Zusammenführung von unterschiedlichen Worten in einem Wort (z. B. Soziologie und Vorlesung zu Soziologievorlesung) viele Wortvariationen. Hinzukommen die uneinheitlichen Schreibweisen von Gender wie Dozentin und Dozent, Dozent*in, Dozent:in und Dozierende.

Box 6.5: Kurzdefinition Lemma

Lemma		Wortstamm
Semester	Semestern	*semester*
Semester	Semesters	
Dozentin	Dozentinnen	dozent*
Dozentin	Dozent	
Dozentin	Dozentinnen	
Dozentin	Dozenten	
Dozentin	Dozierende	doz*
Dozentin	Dozierenden	
dozieren	doziere	dozier*
dozieren	dozierend	
Dozieren	Dozierens	
dozieren	dozierest	
dozieren	dozieret	
dozieren	dozierst	
dozieren	doziert	
dozieren	dozierte	
dozieren	dozierten	
dozieren	doziertest	
dozieren	doziertet	
dozierend	dozierende	
dozierend	dozierendem	
dozierend	dozierenden	
dozierend	dozierender	
dozierend	dozierendes	
Dozierende	Dozierenden	
usw.		

Für die Erfassung sämtlicher Schreibweisen gibt es zumindest zwei Möglichkeiten der Verwendung von Lemma oder von Wortstämmen. Wie im Box 6.5 abgebildet, wird über Lemmatisierung die Grundform eines Wortes bestimmt, der alle möglichen Schreibweisen zugeschrieben werden. Das Lemma als Worttoken kann für die lexikalische Suche als Schlagwort genutzt werden, wodurch die Häufigkeiten abgebildet und Textabschnitte identifiziert werden können.

Statt einer Lemmatisierung kann für die lexikalische Suche und Autokodierung auch der Wortstamm verwendet werden. Das Worttoken erfasst über den Wortstamm ebenfalls die unterschiedlichen Ausprägungen des Such- und Schlagwortes. Der Wortstamm muss empirisch für den Untersuchungsfall identifiziert werden. Anwendungsfreundlich ist die Schlagwortsuche über den Wortstamm durch die Verwendung des Asteriskus-Zeichens „*“ (auf Deutsch: Sternchen). Das * ergänzt und verbindet den Wortstamm mit den heterogenen Schriftformen und erfasst auch zusammengesetzte Wörter. Beispielsweise kann über das Suchwort „*semester*“ sowohl „Semesters“ (Genetiv) und „Semestern“ (Plural) als auch „Wintersemester“ gesucht werden.

Die Verwendung der Schlagwortsuche mit dem Wortstamm ist jedoch nicht per se unproblematisch. Wie in der Erklärung in Box 6.5 abgebildet, exkludiert im Beispiel der Wortstamm „dozent*“ die Aktivitäten von Dozent*innen mit dem Wortstamm „dozier*“ und andersherum. Angesichts der eindeutigen Forschungsfrage und daraus resultierenden Inhalten der Autoethnographien könnte auch über den minimalen Wortstamm „doz*“ die lexikalische Suche und Autokodierung vorgenommen werden, da eine automatisierte Suche im Korpus keine weiteren mit „doz“ beginnenden Worte hervorbringt. Zentral für die Bestimmung des Suchwortstamms ist folglich die Eindeutigkeit. In einem inhaltlich heterogenen Textkorpus aus Zeitungsartikeln und Wahlprogrammen von Parteien würde beispielsweise das kurze Suchwort „bio*“ eine große Bandbreite an empirischen Vorkommen erzeugt, wie „Bioprodukte“, „biologische Waffen“ und „Biotop“, welche dann manuell geordnet werden müssen.

Die Bestimmung von Worttokens für die lexikalische Suche und anschließende Autokodierung muss noch einen weiteren Aspekt der Sprachverwendung bedenken: Synonyme. Wie für die Lemmatisierung findet die Suchmaschine auch Synonymlisten. Die Beachtung von Synonymen ist ebenfalls empfehlenswert, um nachgelagerte Ordnungsnotwendigkeiten durch eine systematische Vorbereitung der Kodierung zu vermeiden. Für den Textkorpus der studentischen Autoethnographien können den mit AntConc generierten Wortlisten ebensolche Synonyme entnommen werden, welche dann durch einen von Ihnen festgelegten Kode erfasst werden können. Wie in Tabelle 6.4 dargestellt, sollten Sie für den verwendeten Code ein allgemeinverständliches Wort wählen wie „Computer,“ wenn

Tabelle 6.4 Beispiel Suchworte in Kodes zusammenfassen

Suchworte						Kode
laptop	pc	computer	tablet	rechner	ipad	Computer
corona	pandemie	covid-19				Corona/Covid-19

Quelle: Autoethnographie WS 2020/21

es einzig darum geht, das Arbeitsgerät zu erfassen, mit dem Studierende Vorlesungen folgen, Texte lesen usw. Werden im Korpus Synonyme ähnlich häufig verwendet, so können Sie auch beide Worte „Corona/Covid-19" für den Kode verwenden.

Die in MAXQDA verwendete Funktion „lexikalische Suche" (Lupe-Icon) ist im Menü beim Tab „Analyse" zu finden. Im Workflow der lexikalischen Analyse ist eine Autokodierung von Schlagworten in einem Textkorpus enthalten. Für die Autokodierung müssen die Schlagworte dann in Kodes übersetzt werden. Für die lexikalische Suche in MAXQDA (im Menü im Tab „Analyse" zu finden) klicken Sie bitte auf den Text „Lexikalische Suche" – und nicht auf das Lupe-Icon! – und wählen Sie bitte die Funktion „Erweiterte lexikalische Suche" aus. Im Fenster „Erweiterte lexikalische Suche" müssen Sie folgende Angaben machen.

1. *Feld „Einer dieser Suchbegriffe":* Eingabe der Schlagworte ohne Komma oder Semikolon, wobei Groß- und Kleinschreibung nicht relevant ist. Für das Beispiel in Tabelle 6.4 wären die Suchworte (alle kleingeschrieben): laptop* pc* ipad* tablet* computer* rechner*, welche mit dem Kode „Computer" erfasst werden könnten. Für einen zu vergebenden Kode „Dozent*in" (siehe Beispiel in Box 6.5) wäre bei „Einer dieser Suchbegriffe" dozent* dozier* einzugeben. Bitte notieren Sie sich stets in Ihr (elektronisches) Forschungstagebuch, welche Suchworte Sie gruppiert haben für die Kodierung.
2. *Feld „In Dokumenten":* Anklicken bzw. Häkchen setzen.
3. *Feld „Text-Dokumente":* Anklicken bzw. Häkchen setzen bei „innerhalb von" und dann beispielsweise Suchworte erfassen in „Absätze".
4. *Button „Suche" anklicken.*

Das Ergebnis Ihrer lexikalischen Suche wird in einem neuen Fenster angezeigt. Ähnlich wie in AntConc „KWIC" (**k**ey **w**ord **i**n **c**ontext) können Sie im neuen Fenster das Suchwort bzw. die Suchworte des Kodes im Textausschnitt ansehen. Anders als AntConc KWIC zeigt MAXQDA nun jeweils den Absatz mit bis zu 255 Zeichen an, in dem sich das bzw. die Suchworte für die Kodierung befinden. Bitte kontrollieren Sie sorgfältig die angezeigten Textausschnitte, um sicherzustellen, dass Sie wenige Fehltreffer über die lexikalische Suche generieren. Grundsätzlich sind Fehltreffer bei einer Schlagwortsuche anzunehmen bzw. unvermeidbar. Es sollten jedoch nicht zu viele Fehltreffer sein – dazu gleich mehr in Kapitel 6.4.3 Datenbereinigung.

Für die lexikalische Suche können Sie auch thematische Zusammenstellungen vornehmen, und anschließend in einem Kode zusammenfassen. Bei der Durchsicht der mit AntConc erstellten Wortlisten könnte Ihnen beispielsweise auffallen, dass Studierende in den Autoethnographien verschiedene Bewertungen vorgenommen haben. Die Bewertungen können Sie nach positiven, negativen und neutralen Schlagworten für die lexikalische Suche zusammenfassen (Tabelle 6.5).

Tabelle 6.5 Bewertungen

Negativ	Neutral	Positiv
leider	*motiv*	neu*
stress*	*allt*	interess*
schwer*	*routin*	freu*
nerv	*anspruch*	spaß*
schlecht*		

Quelle: Autoethnographie WS 2020/21

Bevor Sie sich entscheiden, die Autokodierung durchzuführen, sollten Sie unbedingt einen Test durchführen, um die Beobachtung zu überprüfen. Als Test können Sie eine nach den Bewertungsarten positiv, negativ und neutral gruppierte lexikalische Suche oder für jedes Schlagwort spezifische lexikalische Suche durchführen. Der Test wird Ihre Datenanalyse vereinfachen, wie folgendes Ergebnis für das Beispiel in Tabelle 6.5 zeigt. Für die an die lexikalische Suche anschließende Autokodierung der Dokumente ist es nur sinnvoll, einen, die negativen Suchworte gruppierenden Kode „negative Bewertungen" (leider, stress*, nerv*, schlecht*, schwach*) vorzunehmen. Durch Ergänzungen wie „nicht", „un", „los" usw. können die positiven und neutralen Bewertungsworte mehrheitlich nicht eindeutig erfasst und zugeordnet werden, beispielsweise „das Gespräch in den Breakout-Sessions war uninteressant …", „… hat mich nicht interessiert …", „interesselos verfolgte ich auf dem Computer dem Seminar …" usw. Folglich werden einzeln die Autokodes „Motivation", „Freuen/Freude", „Interesse/interessant/interessiert" und „Spaß" verwendet, um über die Kontextualisierung im Absatz die Bewertungen als manifeste Inhalte auswerten zu können.

Die Ergebnisse können Sie im Suchfenster jeweils als Excel-Dokument (rechts im Suchfester als Icon mit x auf grüner Fläche angezeigt) exportieren und dann als Forschungsdokumentation und/oder für die spätere Analyse speichern. Dies ist notwendig, vor allem, wenn Sie MAXQDA als Lehrlizenz oder in einem PC-Pool an der Hochschule nutzen, d. h. Ihre Arbeit nicht dauerhaft in MAXQDA auf Ihrem Computer gespeichert wird. In der Forschungsdokumentation sollten Sie beispielsweise die Suche nach „*kind*" speichern (Tabelle 6.6). MAXQDA gibt in der Vorschau den Abschnitt aus, welcher aus Platzgründen verkürzt dargestellt wird, sowie die Fundstelle als Segment mit der Absatznummerierung im entsprechenden Dokument an. Sie sehen, dass von 23 Treffern mehr als die Hälfte einer Person zugeordnet wird, welche in der Ergebnisauswertung selbstverständlich anonymisiert wird als idno30. Dazu wurde in einer Liste den Studierenden eine Identifikationsnummer zugewiesen, statt sich 50 Namen auszudenken.

Tabelle 6.6 Suchwort „*kind*“

Vorschau	Dokument	Suchbegriff	Seg.
D[as [Ä]ltere ist ein Kleinkind und [die*der] Jüngste ein Baby	idno30	kind	1
froh den Laptop für den heutigen Tag beiseite zu legen und mit meiner Familie bis zum Schlafen gehen der Kinder zu	idno30	Kind	1
D[as [Ä]ltere ist ein Kleinkind und [die*der] Jüngste ein Baby	idno30	kind	2
D[as [Ä]ltere ist ein Kleinkind und [die*der] Jüngste ein Baby	idno30	kind	3
Nach diesem Seminar bin ich endgültig erschöpft und sehr froh, dass nur noch drei Stunden bleiben, bevor die Kinder ins	idno30	Kind	3
D[as [Ä]ltere ist ein Kleinkind und [die*der] Jüngste ein Baby	idno30	kind	4
D[as [Ä]ltere ist ein Kleinkind und [die*der] Jüngste ein Baby	idno30	kind	6
D[as [Ä]ltere ist ein Kleinkind und [die*der] Jüngste ein Baby	idno30	kind	7
Meine Kinder mache ich für 21:00 Uhr bettfertig	idno30	Kind	7
D[as [Ä]ltere ist ein Kleinkind und [die*der] Jüngste ein Baby	idno30	kind	8
Nach diesem Seminar bin ich endgültig erschöpft und sehr froh, dass nur noch drei Stunden bleiben, bevor die Kinder ins	idno30	Kind	8
D[as [Ä]ltere ist ein Kleinkind und [die*der] Jüngste ein Baby	idno30	kind	9
Am Morgen bin ich zufällig über eine Serie gestolpert, die ich mir als Kind gerne angeschaut habe	idno31	Kind	14
Ich habe mich gefragt, ob sich diese Analyse auch auf Kinderserien übertragen lässt und mir die Folge unter diesem Gesichtspunkt	idno31	Kind	14
Ich habe betrachtet, was die Serie mir als Kind sagen und beibringen sollte und hatte Erfolg	idno31	Kind	14
Heute beginnt mein Tag schon um 7:00 Uhr morgens, da ich meinen kleinen Bruder zum Kindergarten bringen muss, bevor meine	idno16	Kind	7
Außerdem wurde uns auch gezeigt, dass wir auf Probleme stoßen, welche wir vorher nicht kannten, wie zum Beispiel störende Kinder im	idno08	Kind	34
Sie ist Pädagogin und beschäftigt sich aktuell intensiv mit dem Thema Kindererziehung. Das	idno21	Kind	152
Jedoch gibt es höchstwahrscheinlich auch andere Eltern, welche sehr viel Druck auf ihre Kinder ausüben, was sich in einem kleinen	idno26	Kind	12
Es waren wahrscheinlich die Kinder oder Jugendlichen in Dorf, die Klingelstreiche spielen	idno34	Kind	10
Ich prüfe am Fenster, ob es wieder die Kinder sind, die Streiche spielen	idno34	Kind	12
Das führt soweit, dass sie mit ihrem Kind nach Hamburg umzieht	idno46	Kind	33
Bestimmt, weil ich in meiner Kind- und Jugendzeit starker Ungerechtigkeit ausgesetzt war	Idno50	Kind	18

Quelle: Autoethnographie WS 2020/21

Nur für zwei Studierende (idno30 und idno16) bedeutet die Präsenz eines oder mehrerer Kinder im Haushalt eine Organisationsaufgabe, wenn auch eine sehr unterschiedliche. Kind(er) sind Teil des Studiums (idno08, idno21 und idno26), spielen vom Studium ablenkende Streiche (idno34) und sind Gegenstand der Reflexion (idno46 und Idno50). Insgesamt spricht die Heterogenität der Treffer eindeutig dafür, das Schlagwort „*kind*“ von der Analyse auszuschließen, da weder ein eindeutiges noch quantitativ verwertbares Ergebnis erzielt wird.

Sind Ihre Tests der Inhalte der von den Suchwerten erzielten Treffer abgeschlossen, so können Sie nun die Autokodierung beginnen. Im lexikalische Suche-Fenster „Suchergebnisse“ sehen Sie unter dem Suchbefehl „ANY: …“ eine Icon-Leiste. Hier wählen Sie das Kodier-Symbol (rotes Icon mit Plus-Kreuz auf hellgrünem Hintergrund, welches durch Pop-up-Fensterchen „Suchergebnisse mit neuem Kode autokodieren“ bestätigt) und machen im sich öffnenden Fenster „Autokode“ folgende Angaben:

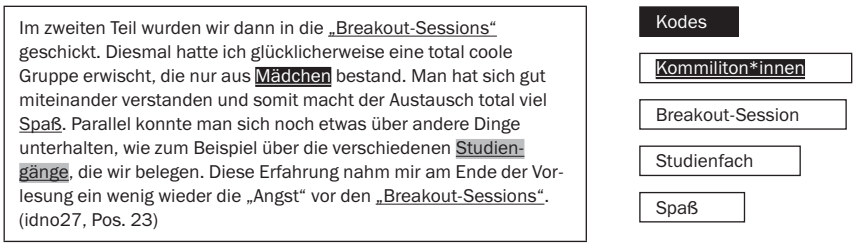
1. *Feld „Code“:* Bitte ersetzen Sie den vorgeschlagenen Kode-Namen „Autocode – ANY: …“ durch einen eindeutigen Namen, welcher dann auch in der Kodewolke (Abbildung 6.3) allgemeinverständlich angezeigt werden kann.
2. *Feld „Farbe“:* Es ist nicht unbedingt notwendig, jedem Kode eine Farbe zuzuweisen. Jedoch ist die visuelle Unterstützung durch farbliche Kennzeichnung der Kodes hilfreich, um in umfangreichen Dokumenten schnell erkennen zu können, welche Kodes in einem Absatz vorkommen. Erste Zusammenfassungen, beispielsweise zu Bewertungen (Tabelle 6.5), können Sie als Anhaltspunkt verwenden, um verschiedene Kodes mit derselben Farbe zu kennzeichnen. Im Beispiel könnten Sie dann „Bewertungen“ als Oberkode mit den Autokodierungen zu „negativ“, „Motivation“, „Freuen/Freude“, „Interesse/interessant/interessiert“ und „Spaß“ als Unterkodes mit derselben Farbe versehen.
3. *Feld „Code-Memo“:* Im Textfeld können Sie Anmerkungen machen, welche als Erinnerungshilfe dienen, beispielsweise für erste Auswertungsideen. Gespeichert werden hier auch die für einen Kode verwendeten Suchworte. So können Sie sich jederzeit die Suchworte aufrufen, ohne in einem anderen Dokument (z. B. Ihrem Forschungstagebuch) danach suchen oder ohne das Programm wechseln zu müssen. Zudem haben Sie so an einer zweiten Stelle diese wichtige Information abgelegt und gespeichert.
4. *Button „OK“ anklicken.*

Die erweiterte lexikalische Suche mit anschließender Autokodierung müssen Sie für sämtliche Worttokens (Schlagwörter) wiederholen.

6.4.3 Datenbereinigung

Im Gegensatz zur qualitativen Problematik des Übersehens oder Verpassens von relevanten Inhalten (siehe Kapitel 5), wird bei der quantitativen Inhaltsanalyse der Verlust von Informationen und Ungenauigkeiten der Datenerfassung anders bewertet. Analog zur vollautomatisierten quantitativen Inhaltsanalyse (siehe Kapitel 8, 9 und 10), bestehen zumindest folgende Vorgehensoptionen, um die Informationsgüte der über die Autokodierung erfassten Textsegmente einschätzen zu können. Ein erster Schritt zur Einschätzung der durch Suchwörter generierten Textsegmente bietet Plausibilisieren basierend auf Alltagswissen und Vergleichen. Durch Vergleichen können beispielsweise die Verwendung der Wörter „Wecker" und „Handy klingeln" als analog die Weckfunktion abbildend eingeordnet werden und folglich in einem Kode „Wecker/Handy klingeln" erfasst werden (Überkode „(Aus-/Ein-)Schlafen"). Das gleiche Vorgehen empfiehlt sich für die Übersetzung von in den Texten verwendeter Sprache in Kodes, beispielsweise die Kodierung des Wortes „Spaß" als positive Bewertung und Übersetzung des Wortes „Mädchen" als Kommiliton*innen (Abbildung 6.4).

Abbildung 6.4 Ankerbeispiel Kommiliton*innen kennenlernen



Quelle: Autoethnographie WS 2020/21

Das Alltagswissen hilft auch beim Aufspüren von potenziellen Fehlern mit Fokus auf das untersuchte Phänomen. Beispielsweise ist es plausibel, anzunehmen, dass das Such- und Schlagwort „Arbeit(en)" multiple Anwendungen im Sprachgebrauch hat. Von Studierenden wird „Arbeit(en)" verwendet als „für die Uni arbeiten" oder „sich an die Arbeit machen" ebenso wie für „Arbeiten gehen" als Synonym für eine (Neben-)Erwerbstätigkeit. In den Autoethnographien der Studierenden wurde arbeiten vor allem für Nebenerwerb verwendet (Abbildung 6.5). Dennoch generierte das Suchwort „*arbeit*" erwartbar auch Fehltreffer wie „Arbeitsplatz" (Abbildung 6.6).

Fehltreffer, wie „Arbeitsplatz" (Abbildung 6.6) und „Motivationsschub", die Küche aufzuräumen statt für das Studium etwas tun (Abbildung 6.5), müssen

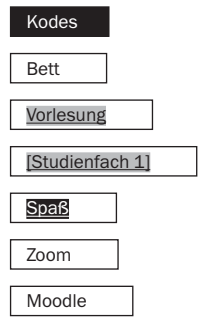
Abbildung 6.5 Ankerbeispiel Arbeit und Fehltreffer „Motivation“

„Heute ist keine Arbeit gewesen und trotzdem wachte ich pünktlich um 8:00 Uhr auf, da ich sowieso schon wach war, habe ich auch die Morgenroutine mit Sport und Dehnen beibehalten, hab aber auf das Duschen verzichtet, da ich sowieso mit trockener Haut zu kämpfen habe. Die übliche KKK (Kaffee, Kippe, Klo) war wieder drin (was ehrlich gesagt sehr gut tat, da ich mein Darm sehr auf die Zigarette und dem Kaffee angewiesen ist). Danach bekam ich einen Motivationsschub, spülte die Küche und räumte die Küchenschränke auf. Mein Mitbewohner ist dadurch sehr grantig aufgewacht und beschwerte sich ein wenig, aber mir war das egal, dass der Faulpelz gern mal früh aufstehen könnte. Ich verstehe nicht wie man 10–13 h am Stück schlafen kann. Wir haben dann zusammen einen Kaffee getrunken (ich meinen zweiten, was sehr ungewöhnlich ist) und eine Zigarette geraucht (was nicht so ungewöhnlich ist). Ich dachte im Zuge dessen viel darüber nach mit dem Rauchen aufzuhören. Aber noch fällt es mir etwas schwer, da ich es schlichtweg noch zu sehr genieße“ (idno50, Pos. 23).

Quelle: Autoethnographie WS 2020/21

Abbildung 6.6 Ankerbeispiel Corona-Studienbedingungen und Fehltreffer „Arbeit“

Eigentlich wäre ich in [Stadtname] gewesen und hätte mich noch an die neue Umgebung gewöhnen müssen. Stattdessen sitze ich nun an meinem Arbeitsplatz zu Hause. Da ich mich im Keller meiner Meinung nach besser konzentrieren kann, da das Bett nicht so nahe ist, habe ich mir hier einen Arbeitsplatz eingerichtet. Hier unten habe ich meine Ruhe. Nachdem ich meinen Arbeitsplatz eingerichtet und mir Tee gekochte habe kann ich mich ab 9:00 Uhr problemlos auf die Vorbereitung für die [Studienfach 1] Vorlesung konzentrieren. Die Moodle-Seite der [Universitätsname] ist an diesem Morgen stark überlastet, weshalb die Seite mir häufig „Error“ anzeigt. Trotz dessen kann ich jedoch an allen Vorlesungen problemlos teilnehmen und bin zudem positiv überrascht von den Zoom-Vorlesungen. (idno15, Pos. 2)



Quelle: Autoethnographie WS 2020/21

für die quantitative Inhaltsanalyse größerer und großer Datenarrangements systematisch erfasst werden – sollen nicht in zeitraubender Fleißarbeit alle 5 918 kodierten Segmente überprüft werden (Tabelle 6.7). Analog zur statistischen Wahrscheinlichkeitsberechnungen können im Datenarrangement die Irrtumswahrscheinlichkeit oder Standardfehler bestimmt werden, wobei je nach Datengüte bei Big Data eine Fehlerwahrscheinlichkeit von 10 % jedoch auch 20 % als akzeptabel gilt (vgl. Steinhardt et al. 2017). Für die Ermittlung der Fehlerwahrscheinlichkeit muss aus dem Datenarrangement eine Stichprobe gezogen werden.

Stichproben können grob nach zwei, idealerweise kombinierten, Vorgehensweisen gezogen werden.

1. Zufallsstichprobe: In den aus MAXQDA exportierten „Projektbestandteilen als Excel-Datei" wird das Datenblatt kodierte Segmente kopiert und in einer neuen Excel-Datei gespeichert. Per Zufall wählen Sie 100 der 5 918 kodierten Segmente aus. Für die 100 kodierten Segmente überprüfen Sie, wie häufig die Schlagworte bzw. 49 vergebenen Kodes falsche Treffer erzielt haben. Hierbei gleichen Sie nur ab, ob der jeweils in der Liste angegebene Kode korrekt ist. Falls weitere Kodes im Textausschnitt enthalten sind, müssen Sie diese ignorieren. Ausgangspunkt der Fehlersuche könnten die oben beschriebenen Treffer zu „Arbeit(en)" und „Motivation" sein. Die gezielte Fehlersuche sollte jedoch nicht zu einer fehlerhaften Bewertung der Güte führen, sondern einen Durchschnittswert über alle Daten produzieren. Folglich müssen Sie auch weniger fehleranfällige bzw. eindeutige Kodes berücksichtigen, beispielsweise Studienfächer, Professor*in und Propädeutikum. Durch die Auswahl von 100 Treffern können Sie dann den prozentualen Anteil als Irrtumswahrscheinlichkeit angeben. Insgesamt ist dieser Wert jedoch als sehr zufällig entstanden zu bewerten, weshalb eine Systematisierung der Ermittlung der Fehlerwahrscheinlichkeit dringend empfohlen wird.

2. Systematische Stichprobe: Systematik zeichnet sich durch ein standardisiert festgelegtes Verfahren aus. Beispielsweise kann unter Berücksichtigung der 49 Kodes festgelegt werden, dass für 100 Tests von jedem Kode zumindest zwei Segmente des Datenarrangements überprüft werden. Unter Berücksichtigung der Kode-Häufigkeiten in Tabelle 6.7 könnte festgelegt werden, dass für jeden Kode der erste und für alle Kodes mögliche x-te Treffer in der von MAXQDA erstellten Liste überprüft wird. Für die Übersicht sollten Sie die entsprechenden Textsegmente zwecks Nachvollziehbarkeit und Transparenz in ein neues Excel-Blatt kopieren. In der Spalte nach den Textsegmenten können Sie das Ergebnis der Überprüfung festhalten, beispielsweise durch das Eintragen von 1 für Fehler. So können Sie die Anzahl der Fehler von Excel addieren lassen. Wenn Sie unsicher bezüglich der Aussagekraft des Ergebnisses (z. B. unerwartet hoher Wert) sind, dann lohnt sich die Wiederholung mit einer zweiten systematisch erstellten Stichprobe. Der Mittelwert beider Stichproben kann dann als Prozentangabe für die Fehlerwahrscheinlichkeit angegeben werden.

Der durch eine systematisch erstellte Zufallsstichprobe ermittelte Wert sollte als Information der Ergebnistabelle, im Beispiel Tabelle 6.7, hinzugefügt werden. Es ist dann nicht notwendig alle Häufigkeitsangaben in Tabelle 6.7 zu korrigieren, d. h. den aufgrund der Fehlerwahrscheinlichkeit ermittelten Wert zu normalisieren.

Hingegen müssen Sie die ermittelte Fehlerwahrscheinlichkeit bei der Er-

klärung und Interpretation der manifesten Inhalte berücksichtigen. Eine Fehlerwahrscheinlichkeit von 20 % würde bedeuten, dass jeder fünfte Treffer als falsch zu werten ist. Weniger Vorsicht bei Ergebniserklärungen und -interpretation ist bei Fehlerwahrscheinlichkeiten von 10 % und weniger geboten, insbesondere von unter 5 % – analog zum 95 % Vertrauens- bzw. Konfidenzintervall in der Statistik (z. B. Diaz-Bone 2019, S. 157–159).

6.4.4 Kodes ordnen

Die Anzahl der Schlagwörter ist in der Regel nicht identisch mit der Anzahl der Kodes. Im Beispiel der Sekundärauswertung der Autoethnographie ist es beispielsweise empfehlenswert, die ermittelten Schlagworte „Schlafen", „Einschlafen" und „Ausschlafen" zusammenzufassen. Abhängig von Ihrer Fragestellung können Sie eine plausible Zusammenfassung von Synonymen und sinnverwandten bzw. dieselbe Handlung oder denselben Gegenstand beschreibenden Schlagworten bereits für die Autokodierung vornehmen. Wichtig ist, dass Sie Ihr Vorgehen dokumentieren und nachvollziehbar beschreiben.

Plausible Zusammenfassung bedeutet in MAXQDA-Sprache, dass Sie zunächst einen Oberkode (im Programm „Obercode") erstellen, welcher bestimmte Merkmale in den Textdaten erfassen hilft. Für die Beantwortung der Forschungsfrage „Wie gestalten Sie Ihren Studienalltag zu Beginn des Online-Semesters 2020/21?" ist es plausibel, anzunehmen, dass Studierende die Gestaltung des Studienalltages bewerten. Entsprechend wurden in Tabelle 6.5 abgebildete Suchwörter den Bewertungsarten positiv, neutral und negativ zugeordnet. Die Zuteilung der Suchworte zu Bewertungsarten erfolgte aufgrund vorhandener, jedoch nicht umfassender manifester Inhaltskenntnisse – bitte vergegenwärtigen Sie sich stets, dass Sie eine *distant reading* Metaanalyse eines größeren, großen bzw. sehr großen Textkorpus vornehmen.

Grundsätzlich sollten sie bei Zusammenfassungen in Kodes von Schlagworten und plausiblen Zuordnungen vorsichtig vorgehen. Beispielsweise hat die kursorische Durchsicht der Konkordanz des Suchwortes „*motiv*" dazu geführt, es als neutrale Bewertung zu klassifizieren, da über die Schlagwortsuche keine eindeutige Zuordnung zu positiver oder negativer Bewertung möglich war (z. B. Treffer sind: sehr motiviert, unmotiviert und Dozent*in motivierte uns …). Die Streuung der erfassten Textstellen im Korpus mit dem Suchwort „*motiv*" können Sie sich in Tabelle 6.3 (ausgewählte Beispiele der Konkordanz zum Suchwort „schwer*") vergegenwärtigen.

Sollte Ihr Textkorpus jedoch umfangreicher sein und/oder Unsicherheit über die zu erzielenden Treffer mit der Schlagwortsuche bestehen, so ist es empfehlenswert, die Erschließung des Textkorpus in MAXQDA über jedes Schlagwort = ein Kode vorzunehmen. In diesem Fall erfolgt die Ordnung durch Sie nach Ab-

schluss von lexikalischer Suche und Autokodierung. Um Ordnung schaffen zu können, müssen Sie selbstverständlich zumindest oberflächliche Einblicke in bzw. ein Gefühl für Ihre Daten haben. Sie verfügen über verschiedene Möglichkeiten, um Ordnung zu schaffen, die sich an den manifesten Inhalten des Datenarrangements orientieren:

1. Nutzen Sie die „Export"-Funktion von MAXQDA (in der Menüleiste den Tab „Reports" als Blatt-mit-rotem-Pfeil-Icon abgebildet). Bei der „Export"-Funktion wählen Sie dann „Projektbestandteile als Excel-Datei" aus, und speichern die exportierte Datei, idealerweise unter Verwendung des/der Kodenamen(s). Die von MAXQDA ausgegebene Excel-Datei zeigt in der unteren Leiste mehrere Blätter *(sheets)* an. Im Blatt „Codierte Segmente" bitte alle Zeilen markieren (z. B. durch Klicken auf das Dreieck oberhalb Zeile 1), und dann in der Menüleiste bei „Sortieren und Filtern" (Trichtersymbol von A–Z) „Benutzerdefiniertes sortieren" auswählen. Bitte beachten Sie, dass das Häkchen bei „Daten haben Überschriften" gesetzt ist, wenn Sie über Spalte „Sortieren nach" im Drop-down-Menü beispielsweise „Code" auswählen. Die Auswahl des/der zuerst zu analysierenden Kodes sollte sich an Ihrer Fragestellung orientieren, d. h. zentral für Ihre Untersuchung sein. Alternativ könnten Sie nach der Spalte „Segmente" filtern, um doppelte Segmente auszusortieren.
2. Analog zum Beispiel des über das Suchwort „*motiv*" erstellten Kodes „Motivation" können Sie in MAXQDA im Fenster „Liste der Codes" (standardmäßig links unten im Interface) einen neuen Code (Symbol mit maigrün gerahmtem Pluszeichen) als Oberkode hinzufügen. Analog zu Oberkodes in Tabelle 6.7 ist es wichtig, eine plausible Begründung für den Oberkode niederzuschreiben, beispielsweise im Forschungstagebuch und/oder in der Memo-Funktion in MAXQDA. Die Memofunktion können Sie jederzeit durch Anklicken des entsprechenden Kodes mit der rechten Maustaste wieder aufrufen. Dem von Ihnen generierten Oberkode können Sie die Suchworte abbildenden Kodes zuordnen, indem Sie diese auf den Oberkode verschieben (Drag-and-drop-Funktion).

6.4.5 Quantitative Ergebnisse als Präsentation manifester Inhalte und zur kodegeleiteten Auswahl für vertiefende Analysen

Für die Metanalyse mithilfe einer quantitativen Inhaltsanalyse bedeutet die Sortierung der Kodes und teilweise manuelle Vergabe von Oberkodes ein zentrales Ergebnis. Angesichts der sehr unterschiedlichen Lebensumstände der Studierenden sollten Sie jedoch vorsichtig sein, die quantitativen Abbildungen der Kode-Häufigkeiten im Datenarrangement als äquivalent zur Bedeutung zu werten. Sicher, eine größere Häufigkeit deutet auf eine größere Bedeutung im

Textkorpus hin, was jedoch bei inhaltlich homogeneren Daten (z. B. einer Stunde Twitter-Diskussion zu einem bestimmten Thema) einfacher analytisch gezeigt werden kann. Die Ergebnisse in Tabelle 6.7 (nächste Seite) zeigen in weiten Teilen erwartbare Kodes im Kontext Studium.

Vergegenwärtigen Sie sich aber bitte, dass häufige Codes nicht zwangsläufig dazu geeignet sind, um Unterschiede zwischen den Sprecher*innen – im vorliegenden Falle der Studierenden – und den Sinngehalt der Aussagen aufzudecken. Häufig ist es so, dass, wie am Beispiel des Kodes „Kinder" gesehen, das Auftreten dieses Kodes für sich genommen eine Unterscheidung zwischen den Studierenden erlaubt. In diesem Fall zwischen denjenigen, die Kinder adressieren und denjenigen, die dies nicht tun. Auch die Abwesenheit des Kodes kann eine Information darüber enthalten, was den Studierenden in ihren Autoethnographien wichtig erscheint bzw. was nicht. Im nächsten Schritt kann das gemeinsame Auftreten dieser Kodes verwendet werden, um zwischen Studierenden mit eigenen Kindern bzw. Kindern als Gegenstand ihres Studiums zu differenzieren. Diese Kodes – vor allem in dieser Kombination – sind nicht unter den häufigsten Kodierungen, wie Tabelle 6.2 zeigt, aber dennoch zur Analyse und Differenzierung geeignet.

Trotz der Tatsache, dass bei der quantitativen Inhaltsanalyse der Korpus aller 50 Autoethnographien der Studierenden analysiert wird, soll hier auch kurz auf die rechte Spalte in Tabelle 6.7 „In n Dok. beobachtet" eingegangen werden. Die Identifikation von zur Beantwortung der Forschungsfrage relevanten Absätzen in den Textdaten über Schlagworte bezieht sich dabei ausschließlich auf das erstellte Datenarrangement. Als Datenarrangement sind die entsprechend kodierten Absätze aus ihrem ursprünglichen Kontext im Dokument, d. h. der jeweiligen Autoethnographie der Studierenden, entrissen. Dennoch können Sie über den Hinweis, in wie vielen Dokumenten der Kode vorkommt („In n Dok. beobachtet") Schlüsse über die Diffundierung des durch den Kode symbolisierten Themas im Dokumentenkorpus ziehen. Dies ist hilfreich, um beispielsweise eine vertiefende Analyse vornehmen zu können, um die Beantwortung der Forschungsfrage „Wie gestalten Sie Ihren Studienalltag zu Beginn des Online-Semesters 2020/21?" an gewisse Punkte anzuschließen und die thematische Inhaltsanalyse weiter auszuführen.

Für die Auswahl relevanter Kodes für die vertiefende Analyse der Daten ist jedoch Verschiedenes zu bedenken. In Tabelle 6.7 sind in der letzten Spalte nie alle 50 Texte angegeben. Dies lässt sich durch eine diverse Sprach- und Grammatikanwendung erklären. Beispielsweise kommt (wider Erwarten) der häufigste Kode „Vorlesung" nur in 41 von 50 Texten vor. Anstatt dezidiert „Vorlesung" in den Autoethnographien zu verwenden, haben neun Studierende stets „Veranstaltungen" geschrieben und den Begriff sowohl für „Seminare" als auch „Vorlesungen" verwendet. In Ermangelung weiterer Anhaltspunkte fehlen diese Angaben in den Unterkodes „Seminare" als auch „Vorlesungen". Von der hier gewählten

Tabelle 6.7 MAXQDA Übersicht der Kodes und absolute (n) und relative (%) Häufigkeiten der kodierten Segmente (Seg.) in den Dokumenten (Dok.) (Fortsetzung nächste Seite)

Oberkode	Kode	n kodierte Seg. (in Dok.)	% kodierte Seg. (in Dok.)	In n Dok. beobachtet
Bewertungen	Motivation	76	1,28	25
	Freuen/Freude	43	0,73	23
	Interesse/Interessant/ Interessiert	111	1,88	35
	Spaß	35	0,59	19
	Negativ (leider, Stress, nerv, schlecht, schwach)	188	3,18	39
IT/Kommunikation	Breakout-Sessions	47	0,79	19
	Computer	143	2,42	33
	Moodle	185	3,13	38
	WhatsApp	48	0,81	21
	Zoom	205	3,46	37
Lesen	Buch/Bücher	55	0,93	21
	Lesen (Verb)	193	3,26	38
	Text (unspezifisch)	208	3,51	39
(Aus-/Ein-) Schlafen	Aufstehen	97	1,64	31
	Bett	106	1,79	29
	Schlafen/Einschlafen/ Ausschlafen (Verben)	138	2,33	34
	Wecker/Handy klingeln	90	1,52	24
Studienfächer	Studienfach (unspezifisch)	149	2,52	37
	[Studienfach 5]	40	0,68	10
	[Studienfach 4]	26	0,44	11
	[Studienfach 2]	20	0,34	11
	[Studienfach 6]	4	0,07	4
	[Studienfach 9]	49	0,83	11
	[Studienfach 7]	21	0,35	9
	[Studienfach 3]	74	1,25	26

Oberkode	Kode	n kodierte Seg. (in Dok.)	% kodierte Seg. (in Dok.)	In n Dok. beobachtet
	[Studienfach 1]	122	2,06	34
	[Studienfach 8]	14	0,24	4
	[Studienfach 10]	25	0,42	10
Studium/ Studieren	Aufgabe(n)	227	3,84	38
	(Erklär-)Video	203	3,43	39
	Dozent*in/Lehrende	120	2,03	28
	Professor*in	91	1,54	21
	Propädeutikum	62	1,05	22
	Seminar	177	2,99	30
	Studium/Studieren(de)	251	4,24	39
	Thema	183	3,09	37
	Üben/Übung	128	2,16	34
	Vorlesung	547	9,24	41
	Wissen(schaft)	136	2,30	35
	Ablenkung/Prokrastination	15	0,25	11
	Arbeit	385	6,51	40
	Autoethnographie	143	2,42	34
	Corona/Covid-19	62	1,05	32
	Kaffee	84	1,42	18
	Kommiliton*innen	118	1,99	30
	Konzentration	83	1,40	29
	Routine(n)	43	0,73	17
	Schule	55	0,93	23
	Universität/[Universitätsname]	293	4,95	40
Insgesamt		**5 918**	**100**	–

Quelle: Autoethnographie WS 2020/21

differenzierten Kodierung von „Seminaren" und „Vorlesungen" sollte bei größeren Textkorpora, d. h. mehr Daten, eher abgesehen werden. Analysieren Sie einen Textkorpus mit deutlich über 500 A4 Seiten, so steht der Erkenntnisgewinn in der Regel in keinem Verhältnis zum Aufwand.

Die Zusammenfassung von „Seminaren" und „Vorlesungen" als „Veranstaltungen" wäre auch für Vergleiche zu wählen. Beispielsweise würde in Autoethnographien von Studierenden naturwissenschaftlicher Studiengänge eher die Veranstaltungsform Labor anstelle von und teilweise ergänzend zu Seminar erwartbar sein. Folglich wird über das Zusammenfassen im Kode „Veranstaltungen" eine Äquivalenz hergestellt, welche selbstverständlich zulasten der detaillierten Auswertung geht. Wie Moretti (2000, S. 57–58) für die Verstehensheuristik des *distant reading* betont, wird durch die zusammenfassende Kodierung eine Selektion vorgenommen, welche die fokussierte Analyse von bestimmten Inhalten und Themen im Sinne des weniger ist mehr ermöglicht. Der Wissensgewinn des Überblicks bzw. der Metaanalyse bedeutet folglich eine Distanzierung von den Texten im Korpus, und damit der qualitativen Dimension des *close reading*.

Im vorgestellten Beispiel mit etwa 500 A4-Seiten handelt es sich weder komplett um *distant reading* noch um *close reading*. Durch die systematische Datenerschließung und unter Verwendung von zwei Softwareprogrammen verknüpften Auswertungstechniken könnte eine methodische Einordnung als *closer-distant reading* erfolgen. Beispielsweise lautet eine Antwort auf die Fragestellung „Wie gestalten Sie Ihren Studienalltag zu Beginn des Online-Semesters 2020/21?", dass das Online-Semester nicht nur andere Formen der Gruppenarbeit mit sich bringt (Breakout-Sessions in Online-Vorlesungen; Ankerbeispiel in Abbildung 6.4) – im Gegensatz zum Analogstudium vor der Corona-Pandemie. Es ist auch plausibel, davon auszugehen, dass das Kaffee-Kippe-Klo-Ritual (Abbildung 6.5) durch den Beginn der Vorlesungszeit durcheinandergebracht wurde, und damit die temporäre Koffein-Nikotin-Metabolismus-Konfusion unabhängig von Analog- oder Online-Semester einen ähnlichen Gang verzeichnet hätte.

Jedoch ist im speziellen Kontext des Wintersemesters 2020/21 das Einrichten des Arbeitsplatzes im Keller des Elternhauses nur eine relevante Information mit dem Voraussatz „eigentlich wäre ich in [Stadtname] gewesen …" (Abbildung 6.6) – eigentlich am Studienort und damit weg vom Bett („da das Bett nicht so nahe ist", Abbildung 6.6). Dies lässt den Schluss zu, dass die*der Studierende dieses Arrangement in einem Semester ohne Corona-Einschränkungen nicht gewählt hätte. Diesbezüglich sehr fraglich ist die Aussage in Abbildung 6.5: „Mein Mitbewohner ist dadurch sehr grantig aufgewacht und beschwerte sich ein wenig, aber mir war das egal, dass der Faulpelz gern mal früh aufstehen könnte. Ich verstehe nicht wie man 10–13 h am Stück schlafen kann" (idno50, Pos. 23). Vermutlich hätte die*der Studierende ohne das Online-Studium nicht so genau die Schlafgewohnheiten des Mitbewohners beeinträchtigt, da sie*er mehr Zeit an der Universität verbracht hätte. Bei solchen Zweifelsfällen ist es ratsam, einen

Textabschnitt bei einer vertiefenden Analyse des Kodes „(Ein-/Aus-)Schlafen“ im Datenarrangement vorsichtig zu bewerten oder gar auszuschließen und zwecks Nachvollziehbarkeit im (elektronischen) Forschungstagebuch oder einem Memo in MAXQDA zu vermerken.

Wie Sie vielleicht inzwischen bemerkt haben, wird die quantitative Inhaltsanalyse durch die Verwendung von Ankerbeispielen besser nachvollziehbar. Ein gutes Ankerbeispiel zeichnet sich dadurch aus, dass es – im Vergleich mit anderen möglichen Beispielen – besonders anschaulich das individuelle Verhalten, soziales Handeln, einen Inhalt usw. veranschaulicht. Ankerbeispiele können sowohl etwas *Typisches* oder *häufig Vorkommendes* abbilden als auch zur Illustration von *Einzigartigem* oder *Ausnahmen* verwendet werden.

Ein Ankerbeispiel für die veränderte Bedeutung des Bettes im Online-Semester ist in Abbildung 6.7 vorgestellt. Es ist eindeutig, dass ohne Online-Semester die*der Studierende nicht sämtliche Morgenveranstaltungen in der autoethnographierten Woche vom Bett aus verfolgen hätte können. Wie Sie sehen, wurde für das Ankerbeispiel nicht nur der über die lexikalische Suche und anschließende Autokodierung erfasste einzelne Absatz, sondern drei Absätze ausgewählt. Der Kode „Zoom“ erzielte in zwei aufeinander folgenden Absätzen Treffer, wie in der als „Projektbestandteile als Excel-Datei“ ausgegebenen Liste erkennbar war. Der dritte Absatz wurde nach Blick in das entsprechende Dokument in MAXQDA hinzugefügt, da dieser die Folgen im Sinne einer Synthese wiedergibt, welche

Abbildung 6.7 Ankerbeispiel Sinneinheit ergänzen für Auswertung

„Nach einem morgen mit viel Freizeit findet dann meine richtige erste Vorlesung statt. Vorlesung [Studienfach 1], der Dozent wirkt sehr sympathisch und bodenständig. Dort wird mir dann erklärt, dass ich hier solch eine Art Tagebuch schreiben soll. Auch neue mit Studierende lerne ich dort kennen, die meisten Anwesenden aus meinem Studiengang dort kenne ich jedoch schon. Der Pandemie zuschulden natürlich nur aus im Internet stattfindenden Zoom-Meetings.

Die vielen schwarzen Kacheln und die nur schwer zuzuordnenden Namen machen das ganze deutlich Anonymer. Generell kann mein Gehirn, welches Menschen sehr stark nach oberflächlichen Merkmalen in bestimmte Kisten einordnet nicht auf seine gewohnte Art und Weise funktionieren. Von ungefähr 160 Teilnehmern im Zoom-Meeting haben gerade so 35 Leute ihre Kamera an oder ein Profilbild eingestellt. Also kann ich als „Ersti“ nicht genau definieren wie Studierende der Fachschaft [Studienfach 7] so aussehen.

Ein paar Vorteile hat das Ganze trotzdem, ich kann mir morgens direkt nach dem Aufstehen die Vorlesung im Bett angucken und meine Freundin die neben mir liegt bekommt ohne überhaupt eingeschrieben zu sein 80 % der Vorlesung mit und lernt auch was (auch wenn es sie nicht interessiert)“ (idno25: Pos. 4–6).

Kodes

- Vorlesung
- Dozent*in
- [Studienfach 1]
- Autoethnographie
- Zoom
- Studium/Studieren
- [Studienfach 7]
- Bett
- Aufstehen
- Interesse

Quelle: Autoethnographie WS 2020/21

Routinen sich abzeichnen und welche Nebenwirkungen, in diesem Fall auf die Freundin, das Online-Semester haben kann (Stichwort: Wissensverschmutzung). Entsprechend ergeben die drei Absätze eine Sinneinheit („unit of meaning“ nach Miles und Huberman 1994, S. 56).

Durch die sinnvolle manuelle Ergänzung von Absätzen zur Sinneinheit, können die Erkenntnisse für Dritte besser dargestellt werden. Diese Illustration zur vereinfachten Nachvollziehbarkeit von Ergebnissen müssen Sie durch ihre Erklärungen, Interpretationen und Schlussfolgerungen ergänzen. Ankerbeispiele dienen der Unterstützung, sie ersetzen ausdrücklich nicht eine explizite Ergebnisformulierung sowie Erklärungen, Schlüsse und Interpretationen der Empirie durch Sie.

7. Deduktiv-quantitative Inhaltsanalyse: das Bibliometric Literature Review

In diesem Kapitel stellen wir Ihnen das Bibliometric Literature Review vor. Dabei handelt es sich um ein Verfahren der deduktiv-quantitativen Inhaltsanalyse, mit dessen Hilfe Sie einen systematischen Überblick zu Publikationen erhalten. Als Grundlage wird zunächst erläutert, was Publikationen eigentlich sind und worauf die systematische Analyse von Literatur beruht: den Zitationen. Um die Analysen durchführen zu können, müssen Sie auf Datenbanken zurückgreifen, weshalb eine Auswahl an Datenbanken und ihre Vor- und Nachteile dargestellt werden. Mit diesen Grundlagen werden dann drei Forschungsfragen, die mit dem Bibliometric Literature Review bearbeitet werden, vorgestellt: Erstens der Literaturüberblick, und die Frage: Was sind die zentralen Publikationen zu einem Thema bzw. in einem Forschungsfeld? Zweitens das Mapping und die Frage: Wie ist ein Forschungsfeld (zu einem bestimmten Thema) aufgebaut, d. h. welche Zusammenhänge lassen sich finden? Und drittens die Themenanalyse, also: Welche Inhalte werden in einem Forschungsfeld bzw. zu einem Thema diskutiert?

7.1 Einleitung

Wenn Sie für eine Hausarbeit oder Abschlussarbeit ein Literaturverzeichnis erstellen, dann werden von Ihnen unterschiedliche Angaben gefordert. Normalerweise sind das mindestens Autor*innenname(n), Erscheinungsdatum, Titel und Ort der Veröffentlichung. All diese Angaben sind bibliometrische Daten. Da es eine enorm große Menge an Publikationen gibt, gibt es auch eine unglaublich große Anzahl an bibliometrischen Daten. Hier stoßen wir bereits auf die erste Frage: Was ist aber eigentlich eine Publikation? Nach Stock (2000) verstehen wir unter Publikationen folgende Formen der Kommunikation:

- Buchhandelsmedien, also Bücher (Monografien und Sammelwerke), Zeitschriften oder Zeitungen,
- graue Literatur, also zum Beispiel Working Papers, Technical Reports, Datenberichte oder Hochschulschriften, die nicht in einem Verlag veröffentlicht wurden, und
- Internet-Dokumente.

Grundsätzlich ist unter einer Publikation ein Kulturgut zu verstehen, das in die Öffentlichkeit gebracht wird. Auch ein Bild kann somit eine Publikation sein,

wenn es in einer Ausstellung publiziert wird. Da wir uns hier aber auf die textbasierte Kommunikation fokussieren, betrachten wir die Daten, die einen textlichen Inhalt transportieren und damit nur verschriftlichte Publikationen. Zudem beschränken wir uns im Folgenden auf wissenschaftliche Publikationen, also auf solche, die entweder von Wissenschaftler*innen produziert oder in wissenschaftlichen Organen, wie wissenschaftlichen Zeitschriften oder Wissenschaftsblogs, veröffentlich wurden. Dies ist eine wichtige Ergänzung, da bisher die Zuschreibung Wissenschaftler*in an die Zugehörigkeit zu einer wissenschaftlichen Institution festgemacht wird, was Wissenschaftler*innen, die ohne Anstellung sind, oder Personen in Firmen sowie Citizen Science[1]-Forscher*innen ausschließen würde.

Nachdem nun geklärt ist, was Publikationen alles sein können, muss als nächster Schritt geklärt werden, welche bibliometrischen Daten eine Publikation alles haben kann.

1. Die Daten, die eine Publikation beschreiben, die so gut wie immer vorliegen. Das sind die Namen der Autor*innen, das Veröffentlichungsdatum, der Ort der Veröffentlichung (Zeitschriftenname, Herausgeberband, Ort und Name des Verlags, Internetadresse) und eindeutige Identifikatoren der Publikation,[2] zum Beispiel die ISBN bei Büchern, die ISSN bei Zeitschriften oder Blogs und DOI bei Zeitschriftenartikeln, Kapiteln in Büchern und Internetdokumenten auf zentralen Plattformen wie zum Beispiel Zenodo.[3]
2. Zusätzliche Daten zu den Autor*innen, die nicht immer oder nur zum Teil vorliegen. Das sind zum Beispiel die Affiliation, also die Institutionen, an denen die Autor*innen angestellt sind, das Land bzw. die Länder, in denen die Institutionen liegen, Angaben zu den Fächern bzw. Disziplinen, denen sich die Autor*innen zugehörig fühlen.
3. Enthalten Publikationen, zumindest Artikel und Buchkapitel, oftmals ein Abstract, also eine kurze Zusammenfassung des Inhalts sowie Schlagwörter, die ebenfalls auf die zentralen Aspekte des Inhalts verweisen.

---

1 „Citizen Science beschreibt die Beteiligung von Personen an wissenschaftlichen Prozessen, die nicht in diesem Wissenschaftsbereich institutionell gebunden sind. Dabei kann die Beteiligung in der kurzzeitigen Erhebung von Daten bis hin zu einem intensiven Einsatz von Freizeit bestehen, um sich gemeinsam mit Wissenschaftlerinnen bzw. Wissenschaftlern und/oder anderen Ehrenamtlichen in ein Forschungsthema zu vertiefen“ (Bonn et al. 2016, S. 13).

2 ISBN = Internationale Standardbuchnummer; ISSN = International Standard Serial Number; DOI = Digital Object Identifier.

3 Neben Verlagen vergeben auch Internetplattformen DOIs. Dazu zählen nicht kommerzielle Preprint-Server (z. B. Zenodo, SocArXiv), bei denen noch nicht veröffentlichte Publikationen öffentlich (Open Access) zugänglich gemacht werden können. Unter Preprints versteht man Manuskripte, die entweder noch keinen Begutachtungsprozess durchlaufen haben oder bereits begutachtet und zur Veröffentlichung freigegeben wurden, aber nicht die Veröffentlichungsversion darstellen. DOIs werden auch durch kommerzielle Plattformen wie ResearchGate oder Academia vergeben.

4. Neben diesen Daten, die die jeweilige Publikation beschreiben, gibt es viertens das Literaturverzeichnis, also eine Liste der Publikationen, die zitiert wurden. Das Literaturverzeichnis wird auch als Bibliografie bezeichnet.
5. Kann über eine Publikation in Erfahrung gebracht werden, wie oft sie in anderen Publikationen zitiert wurde. Dies wird hier, wie in der Forschung zu Bibliometrie und üblich, als Zitation bezeichnet.

Diese Daten, vor allem in aggregierter Form, also wenn viele davon zusammengenommen werden, ermöglichen es, Muster und damit manifeste Strukturen sowohl auf der Ebene der Publikationen als auch auf der Ebene der Inhalte zu untersuchen, was wir als *Bibliometric Literature Review (BLR)* bezeichnen. Das BLR ist eine Methode zur systematischen Analyse bibliometrischer Daten, um manifeste Kommunikationsinhalte zu ermitteln. Im Zentrum des BLR stehen also die Daten, die in und durch eine Publikation erzeugt werden und die einen Rückschluss auf die Kommunikation einer Disziplin, eines Forschungsfeldes oder eines spezifischen Themas zulassen.

Das klingt noch sehr abstrakt, deshalb möchten wir Ihnen an dieser Stelle ein paar Beispiele geben, wozu das BLR genutzt werden kann, also für welche Fragestellungen es sich eignet. Stellen Sie sich beispielsweise vor, Sie arbeiten an Ihrer Abschlussarbeit und stehen nun vor der Frage, was sind eigentlich die zentralen Publikationen eines bestimmten Themas? Hier kann Ihnen das BLR helfen, einen Überblick über ein bestimmtes Thema zu erhalten. Eine andere Fragestellung, die mit dem BLR beantwortet werden kann, ist, wie unterschiedliche Themen eines Forschungsfeldes zusammenhängen, wie sie sich zeitlich entwickelt haben oder wo Überschneidungen oder Abgrenzungen bestehen. Drittens könnte Sie interessieren, welche Inhalte zu einem bestimmten Thema besonders häufig diskutiert werden. Für solche und noch weitere Fragestellungen dient das BLR. Diese Methode kann also einerseits von Ihnen genutzt werden, um sich einen systematischen Überblick zu einem Thema zu verschaffen oder als zentrale Methode Ihrer Arbeit, wenn es um die Betrachtung eines Themas bzw. eines Forschungsfeldes an sich geht.

Aber was ist das BLR nun konkret? Das BLR adaptiert die Forschungsschritte des *Systematic Literature Review (SLR)* mit der Ergänzung um bibliometrische Analyseverfahren. Dazu zählen die Ko-Zitationsanalyse, die bibliografische Kopplung und die Ko-Occurance (siehe Kapitel 7.5). Nicht erschrecken, die Analyseverfahren werden noch im Einzelnen beschrieben. Zunächst muss aber die Grundlage, auf der das BLR aufbaut, erläutert werden, das SLR.

Das SLR ist eine Methode zur systematischen und reproduzierbaren[4] Er-

4 Reproduzierbarkeit, auch Replizierbarkeit genannt, bedeutet, dass andere Forscher*innen zu den gleichen Ergebnissen gelangen, wenn Sie sich an Ihr Forschungsvorgehen (Datenerhebung, Aufbereitung, Analyse) halten.

stellung von Literaturüberblicken zu einem bestimmten Thema (Fink 2019), um Forschungslücken in dem Feld aufzuzeigen (Petticrew und Roberts 2008). Das SLR wurde entwickelt, um die Subjektivität der Literaturrecherche, der Literaturauswahl und Interpretation durch Forscher*innen zu überwinden. Allerdings kann es nie eine ganz objektive Auswahl von Artikeln geben, da bereits durch die Wahl eines Forschungsthemas und die Benennung von Suchbegriffen Subjektivität vorliegt. Die große Stärke des SLR liegt aber in der Transparentmachung des Forschungsprozesses durch die Regelgeleitetheit. Diese Regelgeleitetheit liegt auch dem BLR zugrunde, inkludiert allerdings bibliometrische Auswertungsverfahren. Folgende Schritte sind für das BLR grundlegend, die in Abbildung 7.1 zusammengefasst sind und im Weiteren ausführlich beschrieben werden. Dabei betrachten wir gemeinsam unterschiedliche Beispiele, um zu verdeutlichen, wie der Forschungsprozess aufgebaut werden kann.

Bevor jedoch die einzelnen Schritte des BLR ausführlich anhand von drei Fragestellungen dargestellt werden können, müssen die Möglichkeiten und Grenzen der Datenbanken und Daten, die genutzt werden können, aufgezeigt werden. Denn das Wissen, vor allem um die Grenzen der vorhandenen Daten, ist entscheidend für das Verständnis des methodischen Vorgehens.

Abbildung 7.1 Ablaufschema des Bibliometric Literature Reviews

Schritt 1	Erkenntnisinteresse als Fragestellung formulieren → Implikation für die Auswertungsverfahren
Schritt 2	• Auswahl der Datenbank und Reflexion der damit verbundenen Implikationen • Auswahl des Suchfokus (z. B. Zeitschrift, Suchstring, Forschungsfeld) und Reflexion der Implikationen
Schritt 3	• Festlegung praktischer und methodologischer Auswahlkriterien, welche Publikationen in das Datensample aufgenommen werden • Datenbereinigung
Schritt 4	• Durchführung der bibliometrischen Analysen und Inhaltsanalyse • Deskriptive Beschreibung • Bibliometrische Kopplung • Co-Zitationsanalyse • Co-Occuranceanalyse
Schritt 5	Interpretation der Ergebnisse – Kontext herstellen
Schritt 6	Ergebnisse zusammenfassen und Erstellung einer Präsentation, Studienarbeit und/oder Publikation
Schritt 7	Sichere Archivierung der Daten – siehe dazu Kapitel 3

7.2 Schwächen von Datenbanken und Suchmaschinen

Damit Sie Ihre bibliometrischen Daten interpretieren können und keine falschen Aussagen treffen, ist es zentral zu wissen, welche Schwächen die einzelnen Daten und Datenbanken haben. Zwar kann durch das BLR sehr gut dargestellt werden, wie Forschungsfelder aufgebaut sind und was zentrale Publikationen oder Themen sind. Aber die Erkenntnisse beruhen auf Zitationsmetriken und beziehen Datenbanken ein, die nie alle Daten umfassen. Das heißt die Reichweite der Aussagen ist eingeschränkt. Für ein Thema, Forschungsfeld oder Disziplin haben die Ergebnisse hohe Aussagekraft, können aber nicht auf ein anderes Thema, Forschungsfeld oder Disziplin übertragen werden. Grundsätzlich gilt, dass eine Metrik wie zum Beispiel die Häufigkeit der Zitationen immer im Kontext des jeweiligen Forschungsfeldes, mit den je eigenen wissenschaftskulturellen Gegebenheiten betrachtet werden muss, da es sonst zu starken Verzerrungen kommen kann.

Nehmen wir zum Beispiel die Metrik Zitationszahlen. Diese geben keinen automatischen Hinweis auf die Qualität oder Wichtigkeit einer wissenschaftlichen Publikation. Denn:

- Eine Publikation kann so innovativ sein, dass sonst niemand zu dem Thema forscht und entsprechend die Publikation nicht zitiert wird. Wenn eine Publikation im Mainstream des Forschungsfeldes ist, dann wird sie meist häufiger zitiert, als wenn sie am Rande eines Forschungsfeldes angesiedelt ist.
- Je kleiner Forschungsfelder sind, umso weniger Autor*innen gibt es, die die Publikation zitieren könnten. Deshalb sind die Zitationsanzahlen in kleinen Forschungsfeldern oftmals geringer.
- Forschungsfelder, die interdisziplinär sind, haben oftmals unterschiedliche wissenschaftliche Kontexte und entsprechend unterschiedliches Zitationsverhalten.
- Es gibt Forschungsfelder, in denen vor allem Bücher und Beiträge in Sammelwerken veröffentlicht werden. Diese Publikationsformen werden aber nicht von allen Datenbanken erfasst (oder nur unzureichend).
- Werden Publikationen in anderen Sprachen als Englisch verfasst, ist die Zitationswahrscheinlichkeit geringer, da die internationale Wissenschaftssprache Englisch ist.

Hinzu kommt, dass bei unterschiedlichen Datenbanken unterschiedliche Zitationen gezählt werden – ein Beispiel folgt gleich. Zunächst muss aber erläutert werden, was Datenbanken hier eigentlich meint. Es gibt für bibliometrische Untersuchungen zwei große Datenbanken, über die all die oben aufgeführten Daten abgerufen werden können. Das ist zum einen Web of Science (WoS) und zum anderen Scopus. Neben diesen kommerziellen Datenbanken gibt es diverse

Suchmaschinen, über die nach Publikationen gesucht werden kann, die aber auch anzeigen, wie viele Zitationen eine Publikation hat. Die bekannteste ist sicherlich Google Scholar, weniger bekannt sind CrossRef und Semantic Scholar. Zudem gibt es noch soziale Netzwerke wie zum Beispiel ResearchGate, in denen Wissenschaftler*innen Publikationen ablegen, suchen oder kommentieren können. Auch bei ResearchGate werden Zitationen von Publikationen angezeigt. Wie unterschiedlich die Zitationszahlen in den Datenbanken, Suchmaschinen und sozialen Netzwerken dann tatsächlich sind, möchten wir anhand von zwei Beispielen darstellen, an denen die Autor*innen dieses Buches beteiligt sind (Stand 12. 07. 2021).

Jungblut, J., Vukasovic, M. & Steinhardt, I. (2018 first published). Higher education policy dynamics in turbulent times – access to higher education for refugees in Europe. In: Studies in Higher Education 45, S. 1–12. *https://doi.org/10.1080/03075079.2018.1525697*.

- 23 Zitationen bei Google Scholar
- 17 Zitationen bei ResearchGate
- 10 Zitationen bei Semantic Scholar
- 8 Zitationen bei CrossRef
- 3 Zitationen bei WoS
- 6 Zitationen bei Scopus

Steinhardt, I., Schneijderberg, C., Götze, N., Baumann, J. & Krücken, G. (2016 first published). Mapping the quality assurance of teaching and learning in higher education: The emergence of a specialty? In: Higher Education, 74(2), S. 221–237. *https://doi.org/10.1007/s10734-016-0045-5*

- 74 Zitationen bei Google Scholar
- 51 Zitationen bei ResearchGate
- 44 Zitationen bei Semantic Scholar
- 31 Zitationen bei CrossRef
- 24 Zitationen bei WoS
- Zitationen bei Scopus werden von der Zeitschrift nicht angegeben

Die Unterschiede zwischen den einzelnen Zitationszahlen ergeben sich aufgrund der Beschaffenheit der Datenbanken und aufgrund der verwendeten Daten, was im Folgenden ausführlicher beschrieben wird. Bevor wir auf die einzelnen Anbieter eingehen, noch eine grundsätzliche Anmerkung: Datenbanken und Suchmaschinen sind keine neutralen Artefakte. Vielmehr sind sie Produkte, denen spezifische Interessen eingeschrieben sind. Die meisten Datenbanken sind kommerzielle Produkte bzw. dienen Konzernen, die kommerzielle Interessen haben,

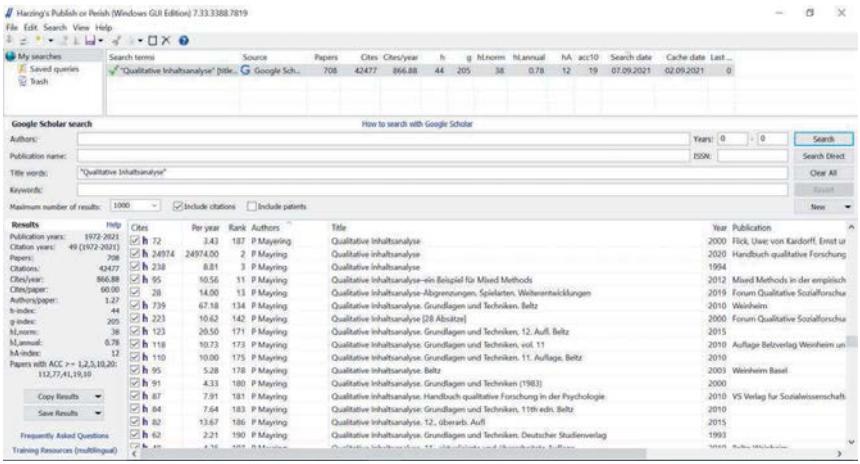
zum Beispiel über Werbung Gewinne zu erwirtschaften oder ihre Daten zu verkaufen. Aber auch nicht-kommerzielle Datenbanken haben Interessen, die sich an der Auswahl der zur Verfügung gestellten Daten zeigen.

7.2.1 Google Scholar

Zunächst muss festgehalten werden, dass Google Scholar keine Datenbank ist. Vielmehr ist es eine Suchmaschine, die durch *crawling,* also dem Durchsuchen von Webseiten, Daten zusammenstellt. In diesem Fall sammelt Googles „Crawler" Informationen über Publikationen und Zitationen auf den Webseiten, auf denen die Publikationen bereitgestellt wurden. Hierzu wird das Verfahren des *parsing* verwendet. Darunter wird die Analyse von Symbolen (in diesem Fall Buchstaben und Zahlen) unter Einbezug grammatikalischer Regeln oder Dokumentenlayouts, wie größere Schrift bei Überschriften, verstanden. Eine Zahlenfolge von vier Zahlen ist sehr wahrscheinlich eine Jahresangabe und wird von Google Scholar entsprechend identifiziert. Hierdurch können bibliografische Angaben (Autor*in, Titel usw.) erkannt werden. Allerdings gibt es keine Standards für die Daten, sodass diese oftmals fehlerhaft sind (dazu gleich mehr) (sehr gut aufbereitet sind die Schwierigkeiten mit Google Scholar bei Orduna-Malea et al. 2017). Bekannt ist, dass Google Scholar die Webseiten von Hochschulen durchsucht und Informationen aus Datenbanken von Verlagen abruft, mit denen zuvor Verträge über den Abruf der publikationsrelevanten Informationen geschlossen wurden. Allerdings ist nicht bekannt, welche Verlage dies sind. Zudem bedarf es eines Tools wie Publish or Perish (Harzing 2019), um die Daten von Google Scholar nutzbar und vor allem bearbeitbar zu machen. Es handelt sich um ein Open Source-Programm, das kostenlos von der Seite *harzing.com* heruntergeladen werden kann. Das Programm läuft sowohl auf Windows- als auch auf Mac-Rechnern. Sie können sich das Programm Publish or Perish als Tool vorstellen, das Ihre Suche bei Google Scholar in eine Excel-Tabelle übersetzt, mit der Sie dann weiterarbeiten können. Allerdings sind die Daten von Google Scholar von sehr geringer Qualität (Martín-Martín et al. 2021). Wir möchten Ihnen hier ein Beispiel geben, damit Sie nachvollziehen können, warum wir derzeit davon abraten, Google Scholar zu verwenden.

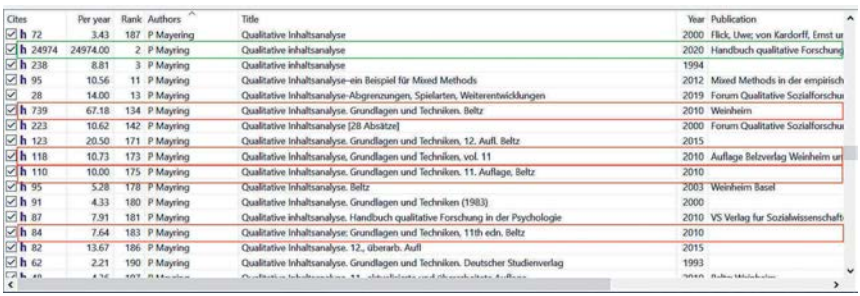
Auf dem folgenden Bild sehen Sie einen Screenshot einer Suche bei Publish or Perish. Wichtig ist dabei, dass diese Suche eins zu eins widerspiegelt, was Ihnen bei Google Scholar bei einer Suche angezeigt würde. Gesucht wurde mit der Phrase „Qualitative Inhaltsanalyse" im Titel der Publikationen („Title words"). Es wurde keine zeitliche Einschränkung gemacht. Insgesamt wurden 708 Treffer gefunden. Abbildung 7.2 umfasst einen kleinen Ausschnitt mit Treffern des Autors Philipp Mayring. An diesem Beispiel können wir einige Probleme der Datenqualität von Google Scholar erkennen, die wir gleich gemeinsam durchgehen.

Abbildung 7.2 Screenshot Suche „Publish or Perish“



Dazu haben wir den Ausschnitt vergrößert, sodass es besser nachvollziehbar wird.

Abbildung 7.3 Datenfehler von Google Scholar zu „Philipp Mayring“ – doppelte Publikationen



Cites	Per year	Rank	Authors	Title	Year	Publication
h 72	3.43	187	P Mayering	Qualitative Inhaltsanalyse	2000	Flick, Uwe; von Kardorff, Ernst u
h 24974	24974.00	2	P Mayring	Qualitative Inhaltsanalyse	2020	Handbuch qualitative Forschung
h 238	8.81	3	P Mayring	Qualitative Inhaltsanalyse	1994	
h 95	10.56	11	P Mayring	Qualitative Inhaltsanalyse-ein Beispiel für Mixed Methods	2012	Mixed Methods in der empirisch
28	14.00	13	P Mayring	Qualitative Inhaltsanalyse-Abgrenzungen, Spielarten, Weiterentwicklungen	2019	Forum Qualitative Sozialforschu
h 739	67.18	134	P Mayring	Qualitative Inhaltsanalyse. Grundlagen und Techniken. Beltz	2010	Weinheim
h 223	10.62	142	P Mayring	Qualitative Inhaltsanalyse [28 Absätze]	2000	Forum Qualitative Sozialforschu
h 123	20.50	171	P Mayring	Qualitative Inhaltsanalyse. Grundlagen und Techniken. 12. Aufl. Beltz	2015	
h 118	10.73	173	P Mayring	Qualitative Inhaltsanalyse. Grundlagen und Techniken, vol. 11	2010	Auflage Belzverlag Weinheim ur
h 110	10.00	175	P Mayring	Qualitative Inhaltsanalyse. Grundlagen und Techniken. 11. Auflage, Beltz	2010	
h 95	5.28	178	P Mayring	Qualitative Inhaltsanalyse. Beltz	2003	Weinheim Basel
h 91	4.33	180	P Mayring	Qualitative Inhaltsanalyse. Grundlagen und Techniken (1983)	2000	
h 87	7.91	181	P Mayring	Qualitative Inhaltsanalyse. Handbuch qualitative Forschung in der Psychologie	2010	VS Verlag fur Sozialwissenschaft
h 84	7.64	183	P Mayring	Qualitative Inhaltsanalyse: Grundlagen und Techniken, 11th edn. Beltz	2010	
h 82	13.67	186	P Mayring	Qualitative Inhaltsanalyse. 12., überarb. Aufl	2015	
h 62	2.21	190	P Mayring	Qualitative Inhaltsanalyse. Grundlagen und Techniken. Deutscher Studienverlag	1993	

Wenn Sie Abbildung 7.3 betrachten, dann sehen Sie beim zweiten angezeigten Treffer (grün umrandet), dass diese Publikation 24 974 Zitationen hat (1. Spalte = „Cites“), und zwar innerhalb eines Jahres (2. Spalte „Per year“). Solche Angaben sollten Sie immer stutzig machen! Hintergrund dieser Angabe ist, dass Google Scholar insgesamt 22 Publikationen von Mayring, die den Titel „Qualitative Inhaltsanalyse“ tragen, in einen Treffer zusammengefasst hat. Das entspricht aber natürlich nicht einer Publikation und verzerrt das Ergebnis absolut. Bei der Überprüfung zeigt sich zudem, dass die Anzahl der Zitationen, die hier angegeben

ist, nicht der Anzahl an Zitationen der 22 zusammengezogenen Publikationen entspricht. Es bleibt also unklar, woher diese hohe Anzahl an Zitationen stammt.

Als zweites Beispiel haben wir Ihnen Treffer rot umrandet, die alle die gleiche Publikation sind, durch das Verfahren des *crawlings* von Google Scholar aber als unterschiedliche Publikationen angegeben werden (Abbildung 7.3). Denn, wie Sie sehen, haben die Treffer unterschiedliche Untertitel (Beltz, vol. 11, 11. Auflage Beltz, 11th edn. Beltz), je nachdem wie sie auf Homepages oder in Literaturverzeichnissen angegeben wurden. Nun könnten Sie natürlich diese Treffer zusammenzählen. Allerdings müssen Sie dann auch den gesamten Datensatz durchgehen, da nicht nur die Literaturangaben, sondern auch die Autor*innennamen oftmals falsch sind. Wie Abbildung 7.4 zeigt, gibt es nicht nur Treffer für Philipp Mayring als „P Mayring", sondern auch als „M Philipp" oder „M Philippe".

Abbildung 7.4 Datenfehler von Google Scholar zu „Philipp Mayring" – falscher Autor*innenname

Cites	Per year	Rank	Authors	Title	Year	Publication
0	0.00	119	M Mucundorfeanu	Qualitative Inhaltsanalyse der Zeitung Neuer Weg	2012	Journal of Media Research-Revis
0	0.00	689	M Nake	Motivationsstrategien in der Gamification Handbuchliteratur: eine qualitative Inhaltsanalyse der Handb...	2014	
3	3.00	39	M Niederberger, S ...	Die qualitative Inhaltsanalyse in den Gesundheitswissenschaften: Ergebnisse eines systematischen Revie...	2020	Forum Qualitative Sozialforschu
0	0.00	114	M Pawicki	Kuckartz, Udo (2012). Qualitative Inhaltsanalyse. Methoden, Praxis, Computerunterstützung. Weinheim ...	2014	Journal for educational research
h 102	7.29	176	M Philipp	Qualitative Inhaltsanalyse. Grundlagen und Techniken	2007	Beltz Deutscher Studien Verlag
h 80	3.81	182	M Philipp	Qualitative inhaltsanalyse	2000	Forum Qualitative Sozialforschu
4	0.22	298	M Philipp	Qualitative Inhaltsanalyse. Grundlagen und Techniken, Beltz	2003	
2	0.15	450	M Philipp	Qualitative Inhaltsanalyse. Grundlagen und Techniken. 10. Auflage, Beltz	2008	
2	0.10	472	M Philippe	Qualitative Inhaltsanalyse: Grundlagen und Techniken	2000	
3	0.33	296	M Plattner	Aufgabenkultur im Sportunterricht: eine qualitative Inhaltsanalyse der Bewegungsaufgaben von Mobile	2012	
2	0.22	448	M Plattner	Aufgabenkultur im Sportunterricht. Eine qualitative Inhaltsanalyse der Bewegungsaufgaben von Mobile...	2012	
0	0.00	139	M Reimann	Alternative Konfliktlösungsoptionen im Golfkrieg: eine qualitative Inhaltsanalyse deutscher Medien (199...	1997	
3	0.38	76	M Rogge	Die qualitative Inhaltsanalyse als Mittel zur empirischen Konkretisierung fremdsprachendidaktischer Par...	2013	Kolloquium Fremdsprachenunte
0	0.00	614	M Salih	Pressekommentare in der Medienberichterstattung über den Irakkrieg von 1991 und 2003: eine qualita...	2007	
0	0.00	696	M Schiffers	Forschungsdesign und qualitative Datenauswertung: vergleichendes Analyseraster der congruence anal...	2019	Lobbyisten am runden Tisch
23	2.88	21	M Schnell, C Schulz...	Der Patient am Lebensende: Eine Qualitative Inhaltsanalyse	2013	

Was wir mit diesen Beispielen zeigen wollen, ist, dass Google Scholar keine Datenbank, sondern eine Suchmaschine ist, was Auswirkungen auf die Datenqualität hat. Findet bei Datenbanken eine Systematisierung der Dateneingabe statt und vor allem eine Qualitätskontrolle, ist dies bei einer Suchmaschine nicht gegeben. Bibliometrische Datenbanken legen für Publikationen Metadaten an, diese werden von Google Scholar aber nicht verwendet. Das bedeutet, dass einige Fehler auftreten. Erstens werden Autor*innennamen oftmals falsch ausgelesen (wie auch in unserem Beispiel aufgezeigt) (Orduna-Malea et al. 2017). Zweitens, werden Überschriften zum Beispiel nur erkannt, wenn sich ihr Layout von dem restlichen Text unterscheidet (Orduna-Malea et al. 2017, S. 12). Drittens treten oftmals Fehler bei der Bestimmung der Jahreszahlen etc. auf: „errors because the parsers identified any chain of 4 digits as a potential publication date, including page numbers or area codes or street addresses in author affiliation" (Orduna-Malea et al. 2017, S. 12).

Einen weiteren Aspekt möchten wir hier noch anfügen, warum eine echte Datenbank, die kontrolliert wird bzw. Metadaten nach Standards auswählt, Google Scholar vorzuziehen ist. In der Wissenschaft gibt es im Zuge der Digitalisierung und der einfachen Veröffentlichungsmöglichkeiten im Internet sogenannte *Predatory Journals,* die keine wissenschaftlichen Standards einhalten und Pseudowissenschaft fördern (Grudniewicz et al. 2019). Deshalb ist es wichtig nochmals zu betonen, dass es einer fundierten Auswahl der Publikationen bedarf, und auf Datenbanken zurückgegriffen werden sollte, die keine *Predadory Journals* enthalten und auch keine Zitationen in diesen.

7.2.2 CrossRef

CrossRef ist eine nicht kommerzielle Datenbank, die Metadaten (unter anderem auch Zitationen) zur Verfügung stellt. Dafür arbeitet CrossRef mit Verlagen und mit Open Access Repositorien zusammen, indem diese CrossRef die Literaturverzeichnisse der Artikel zur Verfügung stellen und CrossRef dafür die Zitationen als Verlinkung liefert. Dadurch stellt CrossRef eine Datenbank von über 120 Millionen Publikationen zur Verfügung, die eine DOI haben. Denn nur durch die DOI kann eine klare Zuordnung erfolgen. Das heißt, dass zum Beispiel ältere Bücher nicht in der Datenbank auftauchen, da diese meist keine DOI haben. Über eine API kann zudem der gesamte Datensatz heruntergeladen und zum Beispiel mit Python ausgewertet werden. Dafür sind allerdings fortgeschrittene Kenntnisse von Python möglich, da CrossRef keine Operatoren ermöglicht. Das heißt es kann bei CrossRef nicht nach Suchphrasen gesucht werden, sondern nur nach einzelnen Worten.

7.2.3 ResearchGate

Neben den Suchmaschinen und Datenbanken gibt es mit ResearchGate eine kostenfreie, aber kommerzielle Social Media-Anwendung, die neben der Bereitstellung von Publikationen auch die soziale Interaktion zwischen Autor*innen ermöglicht. Notwendig ist dafür ein Account bei ResearchGate, der kostenlos angelegt werden kann. Dann ist es möglich, in seinem Account Publikationen oder Präsentationen abzulegen und Projekte zu erstellen, ebenso kann nach diesen gesucht werden. Zudem ist es möglich, anderen Mitgliedern von ResearchGate Fragen zu stellen, Publikationen zu kommentieren oder Direktnachrichten zu schicken.

ResearchGate erfasst für die Publikationen, die mit einem Account verknüpft sind, Zitationen. Diese werden entweder durch einen *crawler* im Web ermittelt, oder in Dokumenten, die von ResearchGate Mitgliedern hochgeladen werden.

Allerdings wird von ResearchGate keine Möglichkeit zur Verfügung gestellt, diese Daten für Forschungszwecke zu erhalten. Auch die Möglichkeit, die Daten durch Webscraping zu erhalten, ist kaum möglich (Martín-Martín et al. 2021). Deshalb ist ResearchGate keine Alternative, um ein Bibliometric Literature Review durchzuführen.

7.2.4 Web of Science und Scopus

Web of Science (WoS) und Scopus sind zwei Datenbanken, die von den Firmen Clarivate Analytics bzw. Elsevier betrieben werden. Beide Datenbanken greifen auf die Daten von Verlagen zurück und beinhalten peer-reviewed Artikel,[5] Bücher und Konferenzbeiträge. Mittlerweile sind darunter auch Open Access-Zeitschriften. Beide Datenbanken sind kostenpflichtig, wobei WoS an vielen Hochschulen über einen VPN-Client genutzt werden kann, wohingegen Scopus in Deutschland über Bibliotheken kaum zugänglich ist. Neuerdings, aufgrund aufkommender Konkurrenz, bietet Scopus einen kostenlosen Zugang für Forscher*innen an. Inwiefern davon auch Studierende profitieren können, ist bisher aber noch unklar.

In Bezug auf die Daten haben die zwei Datenbanken fünf zentrale Schwächen. Erstens sind die Datenbanken vor allem auf englischsprachige Veröffentlichungen ausgerichtet, was zu einer Unterrepräsentanz von anderen Sprachen führt (und auch zu Ungleichheits- und Kolonialisierungstendenzen, vgl. Tennant 2020). Zweitens beinhalten die Datenbanken mittlerweile Bücher, aber diese sind noch immer unterrepräsentiert. So gibt WoS an, 123 000 Bücher[6] und Scopus 194 000 Bücher[7] in ihrer Datenbank zu haben, der Springer-Verlag stellt auf seiner Homepage aber allein 300 000 Bücher zur Auswahl. Insofern haben beide Datenbanken ein großes Defizit, wenn es um die Bereitstellung von Büchern geht, die vor allem in den Geisteswissenschaften und in Teilen der Sozialwissenschaften nach wie vor die zentralen Publikationsorgane sind. Drittens sind die Datenbanken Gatekeeper und nehmen nicht alle Zeitschriften in ihren Index auf, sondern sortieren nach eigenen Qualitätskriterien die „flagships" aus, so werden nur 10 bis 12 % der Anträge auf Aufnahme bei WoS akzeptiert.[8] Kleine Nischenzeitschriften haben entsprechend kaum Chancen, in den Index aufgenommen zu werden, da es vor allem um die Zitationszahlen geht. Viertens muss bei der Wahl der Datenbank bedacht werden, welche Publikationen abgebildet werden.

5 Peer-Review ist die Bewertung (Review) eines wissenschaftlichen Artikels durch eine*n unabhängige*n Gutachter*in, einer*einem sogenannten Peer.

6 *https://clarivate.libguides.com/webofscienceplatform/coverage* (26. 07. 2021).

7 *www.elsevier.com/data/assets/pdf_file/0017/114533/Scopus_GlobalResearch_Factsheet2019_FINALWEB.pdf* (26. 07. 2021).

8 *https://clarivate.libguides.com/webofscienceplatform/coverage* (26. 07. 2021).

So ist WoS die größere Datenbank in Bezug auf das Alter (1900 bis heute), allerdings mit dem Schwerpunkt auf die Natur- und Technikwissenschaften. Scopus hat einen stärkeren Schwerpunkt auf die Sozialwissenschaften, deckt allerdings nur den Zeitraum 1970 bis heute ab. Fünftens zählen WoS und Scopus nur Zitationen, die in Publikationen erfolgen, die in ihrer Datenbank abgebildet werden. Wird also zum Beispiel ein Artikel in einem Buch zitiert, das nicht in WoS/Scopus gelistet ist, dann wird diese Zitation nicht gezählt.

7.3 Schritt 1: Erkenntnisinteresse als Fragestellung formulieren

Am Beginn jedes Forschungsprozesses steht die Fragestellung. Für das BLR ist es wichtig, dass Sie sich über die Forschungsfrage sicher sind, bevor Sie die Forschung beginnen, da Sie unterschiedliche Daten sowie Datenbanken heranziehen und je nach Forschungsfrage die Analysemethoden auswählen werden.[9] Um das zu verdeutlichen, wird im Folgenden anhand von drei unterschiedlichen Fragestellungen der Forschungsprozess veranschaulicht. Sie können also im Folgenden die einzelnen Schritte für alle drei Fragestellungen lesen oder pro Fragestellung die einzelnen Schritte, da die Fragestellungen in den Kapiteln extra markiert sind.

Die drei Fragestellungen, die hier behandelt werden, sind:

1. *Literaturüberblick:* Was sind die zentralen Publikationen zu einem Thema bzw. in einem Forschungsfeld?
2. *Mapping:* Wie ist ein Forschungsfeld (zu einem bestimmten Thema) aufgebaut, d. h. welche Zusammenhänge lassen sich finden?
3. *Themenanalyse:* Welche Inhalte werden in einem Forschungsfeld bzw. zu einem Thema diskutiert?

7.3.1 Literaturüberblick

Wahrscheinlich haben Sie in Ihrem Studium bisher in Einführungskursen gelernt, die Suchmaschine Ihrer Bibliothek zu nutzen und sich zu einem Thema Literatur ausgeben zu lassen. Diese Literatur ist dann nach Relevanz oder Erscheinungsdatum sortiert. Haben Sie dabei schon einmal darüber nachgedacht, was Relevanz eigentlich bedeutet? Oder Sie nutzen Google Scholar, um Ihre Literatur-

9 Hier unterscheidet sich die BLR von den Forschungsstilen der Ethnographie oder der Grounded Theory, bei der das Phänomen im Vordergrund steht und die Forschungsfrage im Laufe der Forschung verändert und angepasst werden kann.

suche durchzuführen. Schaffen Sie es dabei über die zweite Seite der Trefferliste hinaus? Wussten Sie, dass Google nicht preisgibt, nach welchem Algorithmus die Treffer angezeigt werden?

Sie merken, beides sind keine systematischen und nachvollziehbaren Wege, um einen Literaturüberblick zu erstellen. Deshalb soll unter der ersten Fragestellung solch ein exemplarischer Literaturüberblick dargestellt werden. Dazu ist es zunächst entscheidend, die Forschungsfrage klar zu benennen. Als Beispiel dient hier die Forschungsfrage: Was sind die zentralen Publikationen zur Methode „Qualitative Content Analysis"? Dazu muss zunächst geklärt werden, was eigentlich unter zentral zu verstehen ist. In der Bibliometrie wird dabei die Zentralität einer Publikation anhand der Häufigkeit ihrer Zitationen gemessen. Eine Publikation ist dann besonders zentral, wenn sie besonders häufig von anderen Publikationen zitiert wurde. Damit darf auf keinen Fall verwechselt werden, dass diese Publikationen auch immer besonders qualitativ hochwertig sind. Zitation ist nicht zwangsläufig ein Ausdruck von Qualität! Vielmehr verweisen Zitationen allein auf die Aufmerksamkeit (sowohl im positiven wie negativen Sinne), die eine Publikation erhält.

7.3.2 Mapping

Im Gegensatz zum Literaturüberblick geht es beim Mapping nicht nur darum, die zentralen Publikationen zu identifizieren, sondern darüber hinaus in welchem Verhältnis die Publikationen zueinanderstehen. Deshalb werden für Mappings auch größere Themenkomplexe herangezogen, die sich nicht über eine qualitative Inhaltsanalyse, bei der das Material tatsächlich alles gelesen wird, erschließen lassen. Konkret wollen wir das Forschungsfeld der Inhaltsanalyse betrachten und dadurch herausfinden, welche unterschiedlichen Themen innerhalb des Feldes diskutiert werden, welche Charakteristika sie haben und wie sie zusammenhängen.

Das Mapping greift auf zwei Methoden zurück, um diese Zusammenhänge darzustellen. Erstens die Ko-Zitationsanalyse und zweitens die Bibliografische Kopplung. Die Ko-Zitationsanalyse stellt dar, welche Publikationen zusammen in einer anderen Publikation zitiert wurden. Die Bibliografische Kopplung wiederum stellt dar, welche Publikationen aufgrund der zitierten Publikationen Ähnlichkeiten aufweisen. Mit der Ko-Zitationsanalyse ist ein Blick in die Vergangenheit eines Forschungsfeldes möglich, da dargestellt wird, welche Publikationen besonders zentral (= häufig zitiert) wurden und wie diese zitierten Publikationen zueinanderstehen. Die Bibliografische Kopplung hingegen zeigt einen Blick auf die Gegenwart, da die Publikationen des Forschungsfeldes miteinander in Verbindung gebracht werden.

7.3.3 Themenanalyse

Die dritte Möglichkeit, mit bibliometrischen Verfahren zu arbeiten, ist die Themenanalyse, also die Ko-Occurance Analyse. Bei der Themenanalyse werden Worte miteinander in Verbindung gebracht, die in Publikationen verschriftlicht sind. Die Beziehung zwischen den Worten wird dann in einer Map dargestellt. Dazu werden die Worte analysiert, die entweder im Titel oder im Titel und Abstract einer Publikation vorhanden sind. Eine Analyse der gesamten Publikationen ist nicht möglich, da die Verlage die gesamten Inhalte nicht zur Verfügung stellen (das ist gemeint, wenn von einer Paywall die Rede ist).

Für die Entwicklung unserer Forschungsfrage steht nun im Mittelpunkt, was wir über die Themenanalyse herausfinden möchten. Um Ihnen ein umfassendes Bild geben zu können, was zu einem Forschungsfeld bzw. Thema mit bibliometrischen Verfahren möglich ist, bleiben wir bei unserem Beispiel und nehmen das Forschungsfeld Inhaltsanalyse weiter in den Fokus. Das heißt, wir stellen die Frage: Welche Themen tauchen in Bezug zur Inhaltsanalyse in Publikationen auf und in welcher Verbindung stehen sie zueinander?

7.4 Schritt 2: Auswahl der Datenbank und Suchfokus

Als zweiter Schritt muss nach der Klärung der Forschungsfrage die Entscheidung getroffen werden, welche Datenbank und welcher Suchfokus gewählt wird.

7.4.1 Literaturüberblick

Beim Literaturüberblick wollen wir herausfinden, welche Publikationen im Zusammenhang mit der Methode der qualitativen Inhaltsanalyse besonders zentral sind. Wie oben ausgeführt, gibt es derzeit keine Datenbank, die für deutschsprachige Publikationen, inklusive Bücher, Suchergebnisse zur Verfügung stellt, die qualitativ hochwertig sind. Deshalb haben wir für den Literaturüberblick WoS gewählt und neben deutschen Suchwörtern auch die englischen Entsprechungen zu den Suchwörtern verwendet.

Für den Literaturüberblick ist es wichtig, den Suchfokus zu entwickeln bzw. zu schärfen. Hierzu geht eine systematische Literatursuche mit einer vertieften Auseinandersetzung mit dem Forschungsfeld einher. So könnte in unserem Beispiel gefragt werden, was die zentralen Publikationen in einem bestimmten Forschungszusammenhang sind (z. B. qualitative Inhaltsanalyse in der Soziologie oder noch spezifischer in den Gender Studies). Dann müssten die Suchwörter oder Suchphrasen *(search strings)* an das Forschungsinteresse angepasst werden und nicht nur nach der Phrase „qualitative Inhaltsanalyse" bzw. „qualitative con-

tent analysis", sondern nach einer gekoppelten Phrase „qualitative Inhaltsanalyse" *und* „Soziologie" gesucht werden. Da wir einen Überblick zu Publikationen über die Methode der qualitativen Inhaltsanalyse haben wollen, suchen wir nach allen Publikationen.

Neben den Suchwörtern bzw. Suchphrasen ist als zweite Entscheidung zu überlegen, ob diese nur im Titel oder im gesamten Inhalt gesucht werden sollen. Was unter „gesamter Inhalt" zur Verfügung steht, variiert dabei je nachdem, wie frei zugänglich die Publikationen sind. Ist beispielsweise ein Zeitschriftenartikel hinter einer Bezahlschranke, dann wird von diesem nur Titel, Keywords und Abstract durchsucht. Steht ein Artikel Open Access zur Verfügung, ist also frei zugänglich, dann kann der gesamte Inhalt durchsucht werden.

Für unser Beispiel wählen wir die Suchphrasen „qualitative content analys*", „qualitative content analyz*" und „qualitative Inhaltsanalyse" und suchen diese nur im Titel. Es wurde dabei sowohl nach „analyz" als auch nach „analys" gesucht, um sowohl die britische als auch die amerikanische Schreibweise zu inkludieren. Das Sternchen in der ersten Suchphrase bedeutet, dass sowohl nach „analyse" als auch nach „analysis" und „analyses" gesucht wird. Hintergrund für die Titelsuche ist, dass vor allem Publikationen interessieren, die die qualitative Inhaltsanalyse als zentralen Gegenstand haben und die Methode deshalb im Titel nennen werden.

7.4.2 Mapping

Um ein Mapping durchführen zu können, braucht es eine Datenbank, die Zitationen in Beziehung zu Publikationen setzt. Das wird derzeit über den DOI (Document Object Identifier) durchgeführt. Solch eine Datenbank ist WoS, die wir für unser Mapping auswählen. Damit beschränken wir uns gleichzeitig (siehe Ausführungen Kapitel 7.1.5), da die Datenbank nur eine spezifische Auswahl an Publikationen anbietet. Gleichwohl ist es eine Datenbank, die von den meisten Hochschulen angeboten wird, über die Sie also Daten abrufen können.

Für unsere Forschungsfrage des Mappings ist es wichtig, sich klar zu machen, welches Forschungsfeld genau betrachtet werden soll und wie eine Eingrenzung der Treffer erreicht werden kann. Wichtig ist es, die Schlagworte so zu wählen, dass möglichst alle relevanten Treffer enthalten sind und dabei möglichst wenig Treffer aufgeführt werden, die nicht zum Thema passen *(silence and noise)* (Zitt und Bassecoulard 2006). Für unsere Untersuchung ist zentral, dass wir das Forschungsfeld betrachten wollen, das Inhalte von (verschriftlichter) Kommunikation untersucht. Das wird im englischen Sprachraum als „content analysis", „text analysis" und „document analysis" bezeichnet. Sie merken, es braucht gewisse Grundkenntnisse des Forschungsfeldes, um dieses einem Mapping zu unterziehen. Diese drei Phrasen wurden für die weitere Suche genutzt.

Bei WoS haben wir zunächst mit der „Topic Search“ gearbeitet und dort nach der Phrase (TS = „Content Analys*“) gesucht. Bei der „Topic Search“ wird sowohl im Titel, im Abstrakt und in den Keywords nach der Phrase gesucht. Das Sternchen in der ersten Suchphrase bedeutet, dass sowohl nach „analyse“ als auch nach „analysis“ gesucht wird, also nach allen Varianten, die mit dem Wortstamm „Analys“ möglich sind. Diese Suche ergab aber über 74 000 Treffer, was für ein Mapping viel zu viel ist. Deshalb war bereits bei der ersten Phrase klar, dass eine „Topic Search“ nicht das Mittel der Wahl sein kann. Daher wurde eine differenziertere Suche durchgeführt, die nur im Titel („TI“) nach den Phrasen sucht. Getrennt wurden die unterschiedlichen Phrasen mit einem „OR“ (Oder), sodass jeweils nur eine der Phrasen im Titel auftauchen muss. Es wurde dabei sowohl nach „analyz“ als auch nach „analys“ gesucht, um sowohl die britische als auch die amerikanische Schreibweise zu inkludieren. Wichtig bei der Suche mit WoS ist, dass die Suche in Klammern ist, damit die Phrasen getrennt und gleichzeitig gemeinsam gesucht werden. Insgesamt sieht die Suchabfrage dann wie folgt aus:

```
(((((((TI=("Qualitative Content Analys*")) OR TI=("Qualitative Content Analyz*"))
OR TI=("Quantitative Content Analys*")) OR TI=("Quantitative Content Analyz*"))
OR TI=("text analys*")) OR TI=("text analyz*")) OR TI=("document analys*")) OR
TI=("document analyz*")
```

Mit dieser Suchanfrage haben wir 1 621 Treffer erhalten, was eine sehr gute Anzahl für ein Mapping ist (vgl. Steinhardt et al. 2017).

7.4.3 Themenanalyse

Um unsere Forschungsfrage zu beantworten, welche Themen mit Bezug zur Inhaltsanalyse auftauchen und wie sie zueinander in Verbindung stehen, greifen wir auf die gleiche Datenbasis zurück wie beim Mapping (siehe Kapitel 7.4.2). Dadurch können wir die Themenanalyse zurückkoppeln an die Auswahl der Publikationen und auch in Verbindung zum Forschungsfeld setzen. Denn Sie müssen bedenken, dass es hier um die Analyse von Publikationen geht. Geht es Ihnen um das Erschließen von Themenfeldern als solche, dann sollten Sie eher zum Verfahren des Topic Modeling greifen (siehe Kapitel 11) oder zur Analyse mit MAXQDA und AntConc (siehe Kapitel 6).

7.5 Schritt 3: Datenauswahl und Datenbereinigung

7.5.1 Literaturüberblick

Nachdem entschieden ist, dass die Suchphrasen „qualitative content analys*“ und „Qualitative Inhaltsanalyse“ sind, die wir bei WoS suchen lassen, gilt es nun, die Datenauswahl und die Datenbereinigung durchzuführen.

Was ist nun wichtig für Sie zu wissen, um mit WoS arbeiten zu können? Sie müssen prüfen, ob Sie einen Zugang zu WoS über Ihre Hochschule haben. Das finden sie im Katalogportal Ihrer Hochschulbibliothek unter Datenbanken heraus. Dort müssen Sie nach „Web of Science Core Collection“ suchen. Wenn Sie nicht vor Ort in Ihrer Bibliothek sind, um die Suche durchzuführen, benötigen Sie einen VPN-Client, sodass Sie über die Hochschule eingeloggt sind. Denn WoS kann nur kostenfrei genutzt werden, wenn der Computer im jeweiligen Hochschulnetz angemeldet ist.

Box 7.1: Online Anleitung BLR
Wir haben eine sehr ausführliche Anleitung zur Arbeit mit WoS geschrieben, die Sie auf dem Blog „sozmethode“ unter folgender Adresse finden: *https://sozmethode.hypotheses.org/1049*.

Zurück zu unserem Beispiel: Wir haben am 14. 09. 2021 die Suche bei WoS nach den zwei Phrasen durchgeführt. Erhalten haben wir 371 Treffer, eine Auswahl der acht Treffer mit den höchsten Zitationswerten sehen Sie in Abbildung 7.5 (nächste Seite). Hierzu haben wir die Treffer nach „Citations: highest first“ sortiert, nicht nach Relevanz, wie es die Grundeinstellung von WoS vorsieht.

Wenn Sie in WoS suchen, dann bietet WoS eine Reihe an deskriptiven Statistiken an, die Sie nutzen können, um einen ersten Überblick für ihren Literaturüberblick zu erhalten. Das sind zum Beispiel Überblicke über die Forschungsfelder, in denen die Publikationen veröffentlicht wurden. In unserem Fall zeigt sich, dass die meisten Publikationen mit der Qualitativen Inhaltsanalyse in englischer Sprache im Feld der Gesundheit geschrieben wurden. Als Hintergrundinformation dazu: In den USA gibt es eine sehr lange Tradition der qualitativen Forschung im Bereich des Gesundheitswesens. Deshalb ist dieses Ergebnis nicht verwunderlich, wenn die Hintergründe bekannt sind. Anzunehmen ist deshalb, dass es sich bei diesen Artikeln vor allem um die Nutzung der Methode handelt, nicht aber um methodische Auseinandersetzungen.

Um das genauer zu erkennen, ist es wichtig, sich die Trefferliste genauer anzusehen. In unserem Beispiel fällt erstens auf, dass die gefundenen Publikationen im Titel alle die Phrase haben, die wir suchen (gelb hinterlegt). Zweitens fällt auf, dass es zwei unterschiedliche Arten von Publikationen gibt, wie wir schon vermutet hatten. Zum einen diejenigen, die sich mit der Methode der Qualitativen Inhaltsanalyse an sich beschäftigen und solche, die die Methode anwenden (siehe dazu Abbildung 7.5, Treffer Nr. 8). Da wir einen Überblick über die Methode der qualitativen Inhaltsanalyse haben möchten, also Treffer, die sich mit der Me-

Abbildung 7.5 Trefferliste WoS

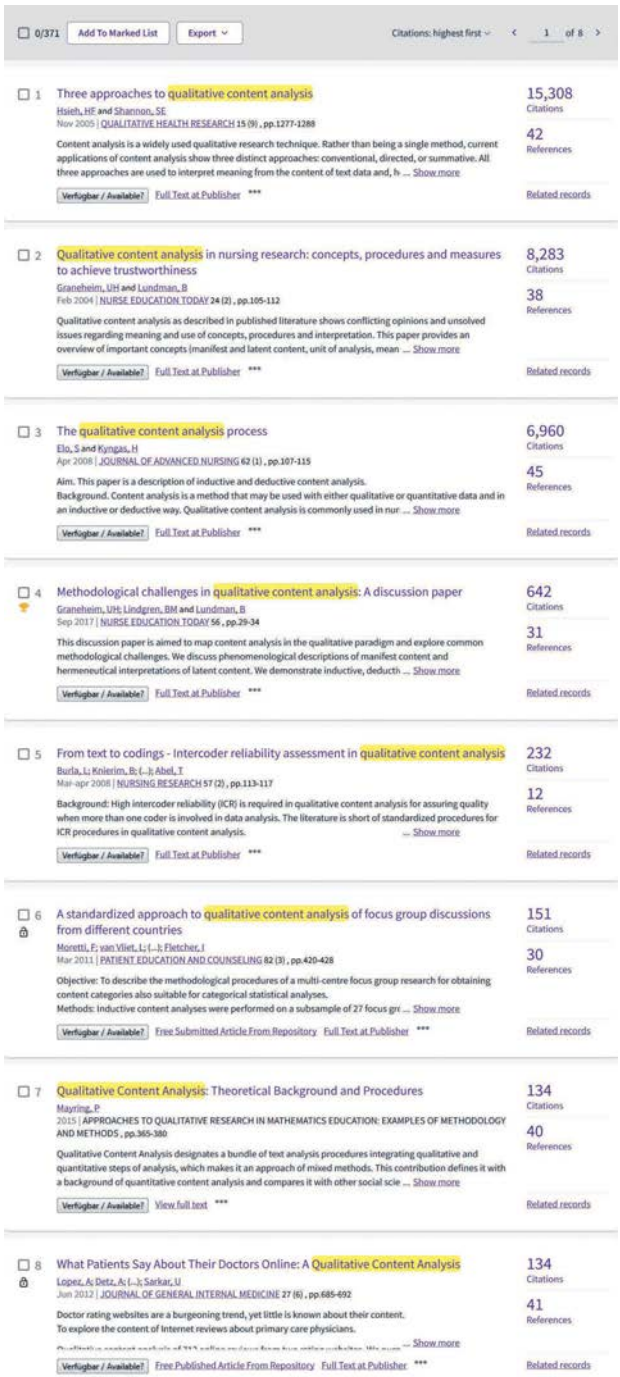
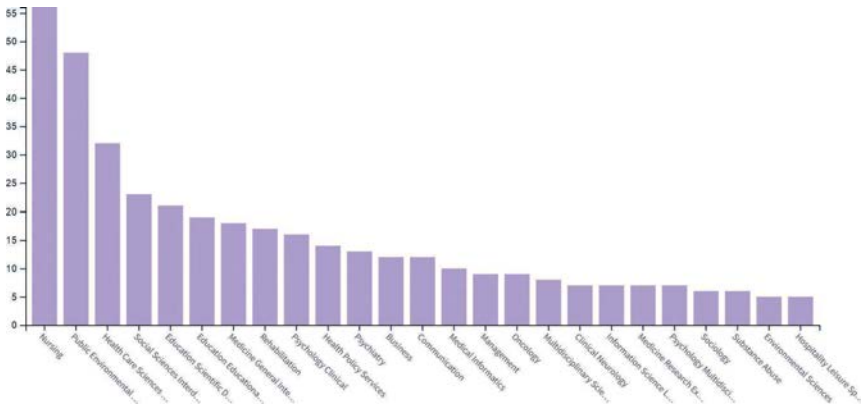


Abbildung 7.6 Research Fields in WoS



thodenentwicklung beschäftigen bzw. einen Überblick über die Methode geben, sortieren wir die Treffer aus, die eine Anwendung beschreiben. Damit haben wir gleichsam unsere Kriterien für die Datenauswahl festgesetzt. Eine andere Auswahl hätte zum Beispiel nur Treffer eines bestimmten „Research Fields" umfassen können, also zum Beispiel nur der Soziologie. Dies können Sie bei WoS tun, indem Sie Ihre Ergebnisse danach filtern. In unserem Beispiel würden dann aber viele Publikationen zur Methodenbeschreibung bzw. -weiterentwicklung verloren gehen.

Deshalb gehen wir einen anderen Weg und wählen die Treffer anhand unseres Kriteriums aus. Konkret heißt das, um ein Beispiel zu zeigen, dass die Publikation mit dem Titel „Performing qualitative content analysis" als relevant aufgenommen wurde. Die Publikation mit dem Titel „Cancer Patients' Informational Needs: Qualitative Content Analysis" nicht, da hier nicht die Methode als solche im Fokus steht, sondern „Cancer Patients' Informational Needs". Für die Datenauswahl können sie entweder direkt in WoS die Trefferliste durchgehen und anhand Ihrer Kriterien auswählen und dann die ausgewählten Ergebnisse als „Tab delimited file" herunterladen. Oder sie laden die gesamte Ergebnisliste herunter und sortieren in Excel Ihre Trefferliste. Das Zweite würden wir Ihnen empfehlen, da in Excel Ihre Arbeit nicht so leicht verloren gehen kann (z. B. bei Internetproblemen) und Sie selbst Ihre Auswahl zu einem späteren Zeitpunkt leichter nachvollziehen können. Wenn Sie aufgrund des Titels nicht direkt entscheiden können, ob der Treffer Ihren Kriterien entspricht, dann schauen Sie am besten in den Abstrakt der Publikation, also in die kurze Zusammenfassung des Inhalts.

Wenn Sie die Trefferliste heruntergeladen haben, dann prüfen Sie diese auf mögliche Dopplungen sowohl bei den Namen als auch bei den Titeln der Publikationen. In unserem Beispiel gab es keine Dopplungen bei den Namen, aber eine Dopplung bei den Titeln. Dabei zeigte sich, dass es sich bei beiden um ein

„Book Review“ handelt, also eine Buchbesprechung, die für unsere Auswertung nicht von Bedeutung ist, deshalb werden diese beiden Publikationen nicht weiter berücksichtigt und aussortiert. Aussortiert wurde zudem ein Artikel, der eine Workshopankündigung ist und damit nicht mit einem Forschungsartikel vergleichbar. Nach der Auswahl und Überprüfung bleiben in unserem Beispiel 26 Publikationen (von 371) übrig, die unserem Kriterium nach Methodenüberblick bzw. weiterentwicklung entsprechen.

7.5.2 Mapping und Themenanalyse

Wenn Sie bereits das Kapitel zuvor gelesen haben, dann überspringen Sie bitte diesen Abschnitt.

Was ist nun wichtig für Sie zu wissen, um mit WoS arbeiten zu können? Sie müssen prüfen, ob Sie einen Zugang zu WoS über Ihre Hochschule haben. Das finden sie im Katalogportal Ihrer Hochschulbibliothek unter Datenbanken heraus. Dort müssen Sie nach „Web of Science Core Collection“ suchen. Wenn Sie nicht vor Ort in Ihrer Bibliothek sind, um die Suche durchzuführen, benötigen Sie einen VPN-Client, sodass Sie über die Hochschule eingeloggt sind. Denn WoS kann nur kostenfrei genutzt werden, wenn der Computer im jeweiligen Hochschulnetz angemeldet ist.

Box 7.2: Online Anleitung BLR
Wir haben eine sehr ausführliche Anleitung zur Arbeit mit WoS geschrieben, die Sie auf dem Blog „sozmethode“ unter folgender Adresse finden: *https://sozmethode.hypotheses.org/1049.*

In unserem Beispiel des Mappings haben wir am 02. 09. 2021 die oben beschriebene Titelsuche durchgeführt und 1 621 Treffer erhalten. Diese Treffer haben wir in 500er Schritten bei WoS heruntergeladen und die einzelnen Pakete dann in einer Liste zusammengeführt. Im Anschluss haben wir geprüft, ob in der Liste Dopplungen vorhanden sind, was nicht der Fall war. Exkludiert haben wir *Reviews* und *Book Reviews,* da beide Formate Zusammenfassungen von anderen Publikationen liefern, was für unsere Analyse verzerrend wirken könnte. Diese Exklusion können Sie einfach vornehmen, da in der Liste ausgewiesen ist, um welche Publikationsform es sich handelt. Nach der Bereinigung blieben 1 350 Publikationen übrig, mit denen weitergearbeitet wurde.

7.6 Schritt 4: bibliometrische Analysen und Inhaltsanalyse

7.6.1 Literaturüberblick

Wir haben nun eine Liste mit 26 Publikationen, die wir weiter auswerten werden. Als ersten Schritt sehen wir, dass innerhalb der Treffer 13 Veröffentlichungen

aus dem Bereich der Gesundheitswissenschaften stammen (grau hinterlegt). Interessant ist, dass zwei Publikationen aus den 80er Jahren stammen und dann erst wieder Publikationen ab 2004 in der Trefferliste auftauchen. 20 der 26 Publikationen sind nach 2011 erschienen. Es handelt sich bei der qualitativen Inhaltsanalyse, zumindest in Artikeln in der Datenbank WoS, also um eine Methode, die vor allem in jüngeren Publikationen diskutiert wurde. Soweit zur deskriptiven Auswertung der Liste.

Tabelle 7.1 Publikationen WoS „Qualitative Content Analys*" (Fortsetzung nächste Seiten)

Nr.	Autor*innen	Titel der Publikation	Publikationsort	Erscheinungsjahr	Zitationen/Jahr
3	Lorenzer, A.	The possibility of a qualitative content analysis – a deep hermeneutic interpretation between ideology criticism and psychoanalysis	Argument	1981	0,1
3	Glassner, B.; Corzine, J.	Library research as fieldwork – a strategy for qualitative content analysis	Sociology and Social Research	1982	0,2
1	Graneheim, U. H.; Lundman, B.	Qualitative content analysis in nursing research: concepts, procedures and measures to achieve trustworthiness	Nurse Education Today	2004	487,2
1	Hsieh, H. F.; Shannon, S. E.	Three approaches to qualitative content analysis	Qualitative Health Research	2005	956,8
2	Burla, L.; Knierim, B.; Barth, J.; Liewald, K.; Duetz, M.; Abel, T.	From text to codings – Intercoder reliability assessment in qualitative content analysis	Nursing Research	2008	17,8
1	Elo, S.; Kyngas, H.	The qualitative content analysis process	Journal of Advance Nursing	2008	535,4
1	Moretti, F.; van Vliet, L.; Bensing, J.; Deledda, G.; Mazzi, M.; Rimondini, M.; Zimmermann, C.; Fletcher, I.	A standardized approach to qualitative content analysis of focus group discussions from different countries	Patient Education and Counseling	2011	15,1
3	Oleinik, A.	Mixing quantitative and qualitative content analysis: triangulation at work	Quality & Quantity	2011	2,4

Nr.	Autor*innen	Titel der Publikation	Publikationsort	Erscheinungsjahr	Zitationen/Jahr
1/3	Mayring, P.	Qualitative Content Analysis: Theoretical Background and Procedures	Approach to Qualitative Research in Mathematic Education	2015	22,3
3/4	Bakharia, A.; Bruza, P.; Watters, J.; Narayan, B.; Sitbon, L.	Interactive Topic Modeling for aiding Qualitative Content Analysis	Proceedings of the 2016 ACM Conference on Human Information Interaction and Retrieval (CHIIR'16)	2016	1,0
1	Plumins, M.; Sceulovs, D.; Gaile-Sarkane, E.	Competitiveness definitions' and concepts qualitative content analysis	Smart and Efficient Economy: Preparation for the Future innovative Economy	2016	0,4
1	Graneheim, U. H.; Lindgren, B. M.; Lundman, B.	Methodological challenges in qualitative content analysis: A discussion paper	Nurse Education Today	2017	160,5
1	Armat, M. R.; Assarroudi, A.; Rad, M.; Sharifi, H.; Heydari, A.	Inductive and Deductive: Ambiguous Labels in Qualitative Content Analysis	Qualitative Report	2018	9,6
1	Assarroudi, A.; Nabavi, F. H.; Armat, M. R.; Ebadi, A.; Vaismoradi, M.	Directed qualitative content analysis: the description and elaboration of its underpinning methods and data analysis process	Journal of Research in Nursing	2018	38,0
1/3	Hall, C. M.	Quantitative and qualitative content analysis	Handbook of Research Methods for Tourism and Hospitality Management	2018	1,0
1	Ramakrishna, R. A. R.	Using intuitive judgment in qualitative content analysis: implications for research in varieties of English	International Journal of Business and Society	2018	0,0
3/4	Renz, S. M.; Carrington, J. M.; Badger, T. A.	Two Strategies for Qualitative Content Analysis: An Intramethod Approach to Triangulation	Qualitative Health Research	2018	11,6

Nr.	Autor*innen	Titel der Publikation	Publikationsort	Erscheinungsjahr	Zitationen/Jahr
1	Faria-Schutzer, D. B.; Bastos, R.; Alves, V.; Campos, C. J. G.; Surita, F. G.; Turato, E. R.	The seven steps of the clinical-qualitative content analysis: a data-processing technique for research into clinical care settings	European Psychiatry	2019	0,0
1	Hoft, N.; Heckmann, M.; Jankowicz, D.	Systematic Integration of Quantitative Measures into the Qualitative Content Analysis of Constructs	Journal of Constructivist Psychology	2019	1,0
1	Kibiswa, N. K.	Directed Qualitative Content Analysis (DQICA): A Tool for Conflict Analysis	Qualitative Report	2019	1,5
1	Pohontsch, N. J.	Qualitative Content Analysis	Rehabilitation	2019	2,0
1	Hall, M.; Wucherpfennig, F.; Rubel, J. A.	Qualitative Content Analysis	Psychotherapie Psychosomatik Medizinische Psychologie	2020	1,0
1	Lindgren, B. M.; Lundman, B.; Graneheim, U. H.	Abstraction and interpretation during the qualitative content analysis process	International Journal of Nursing Studies	2020	40,0
1	Manic, Z.	Performing qualitative content analysis	Sociologija	2020	1,0
1	Selvi, A. F.	Qualitative content analysis	Routledge Handbook of Research Methods in Applied Linguistics	2020	11,0
1	de Faria-Schutzer, D. B.; Surita, F. G.; Alves, V. L. P.; Bastos, R. A.; Campos, C. J. G.; Turato, E. R.	Seven steps for qualitative treatment in health research: the Clinical-Qualitative Content Analysis	Ciencia & Saude Coletiva	2021	0,0

Für den Literaturüberblick gehen wir nun in die Texte und betrachten deren Inhalt. Um erste Anhaltspunkte zur Beantwortung unserer Fragestellung zu finden, können wir mit den Texten beginnen, die die meisten Zitationen pro Jahr auf sich vereinen. Dabei muss Ihnen aber klar sein, dass damit noch kein Qualitätsanspruch einhergeht, nur, dass andere Personen die Publikation (aus welchem Grund auch immer) für zitierwürdig gehalten haben. Die Anzahl der Publikationen insgesamt entnehmen wir der WoS-Liste in der Spalte mit der Überschrift „TC", das für „Times Cited Counts" steht. Diese Anzahl teilen wir durch das Alter der Publikation und erhalten dadurch den Durchschnitt der Zitationen pro Jahr.

Deutlich wird in der Tabelle 7.1, dass es vier Publikationen gibt, die im Durchschnitt über 150 Zitationen pro Jahr haben. Mit diesen starten wir den Literaturüberblick und lesen die Publikationen.

Nach dem Lesen der meistzitierten Artikel gilt es nun, eine Kategorisierung der Artikel durchzuführen, also eine Ordnung nach Kriterien. Bei einem Literaturüberblick erfolgt die Kategorisierung dabei meist induktiv, wenn die wichtigsten Informationen aus den Publikationen zusammengefasst und damit geordnet werden (siehe dazu Kapitel 4). In unserem Beispiel zeigen sich drei Kategorien, wobei es Publikationen gibt, die in mehrere Kategorien eingeordnet werden können.

- Tabelle 7.1, Spalte 1 die Nummer 1: Texte, die eine Zusammenfassung und Strukturierung der qualitativen Inhaltsanalyse durchführen. Besonderer Fokus liegt dabei auf dem Coding Prozess und dezidierten Anleitungen, wie die Qualitative Inhaltsanalyse in ihren unterschiedlichen Varianten durchgeführt werden kann.
- Tabelle 7.1 Spalte 1 die Nummer 2: Texte, die sich mit Qualitätskriterien der Qualitativen Inhaltsanalyse befassen.
- Tabelle 7.1 Spalte 1 die Nummer 3: Texte, die Vergleiche zwischen anderen Methoden und der Qualitativen Inhaltsanalyse durchführen oder diese mit anderen Methoden zusammen verwenden (Mixed Methods-Design bzw. Triangulation).
- Tabelle 7.1 Spalte 1 die Nummer 4: Hier werden Möglichkeiten der computergestützten qualitativen Inhaltsanalyse vorgestellt.

7.6.2 Mapping

Nachdem wir die Datenbasis haben, geht es nun an die Datenauswertung. Für die Erstellung der Ko-Zitationsanalyse und der Bibliografischen Kopplung verwenden wir das Open Source-Tool VosViewer (siehe zu Copylefts und Copyrights Kapitel 1.3). Dieses können Sie unter *www.vosviewer.com* einfach herunterladen. VosViewer ermöglicht die heruntergeladene txt-Datei von WoS einfach in das Programm einzuspeisen und führt dann die Analysen durch. Wie oben beschrieben, ist für das Mapping sowohl die Ko-Zitationsanalyse als auch die Bibliografische Kopplung interessant. Für beide Analysen erstellt VosViewer Maps, die sich darin unterscheiden, welche Publikationen in der Map angezeigt werden. Bei der Ko-Zitationsanalyse werfen wir einen Blick in die „Vergangenheit". Entsprechend werden in der Map die zitierten Publikationen angezeigt, die besonders häufig in den Publikationen zitiert wurden, die unsere Trefferliste (also die 1 350 Publikationen) umfasst. Damit ist ein Spezifikum der Ko-Zitationsanalyse angesprochen: Beiträge, die schon länger publiziert sind, erscheinen sehr häufig.

Hingegen sind neuere Beiträge in einer Ko-Zitationsanalyse unterrepräsentiert, schlicht, weil die Beiträge noch nicht lange publiziert und damit zitiert werden konnten.

Die Map der Bibliografischen Kopplung bildet hingegen die Publikationen der Trefferliste ab und verbindet die Publikationen, die die gleichen Publikationen zitieren (also die gemeinsame Bibliografie). Damit werden aktuelle Entwicklungen in einem Forschungsfeld aufgezeigt, was dazu führen kann, dass bei einem nicht kohärent zusammenhängenden Forschungsfeld keine klare Map als Ergebnis zu sehen ist (dazu gleich mehr).

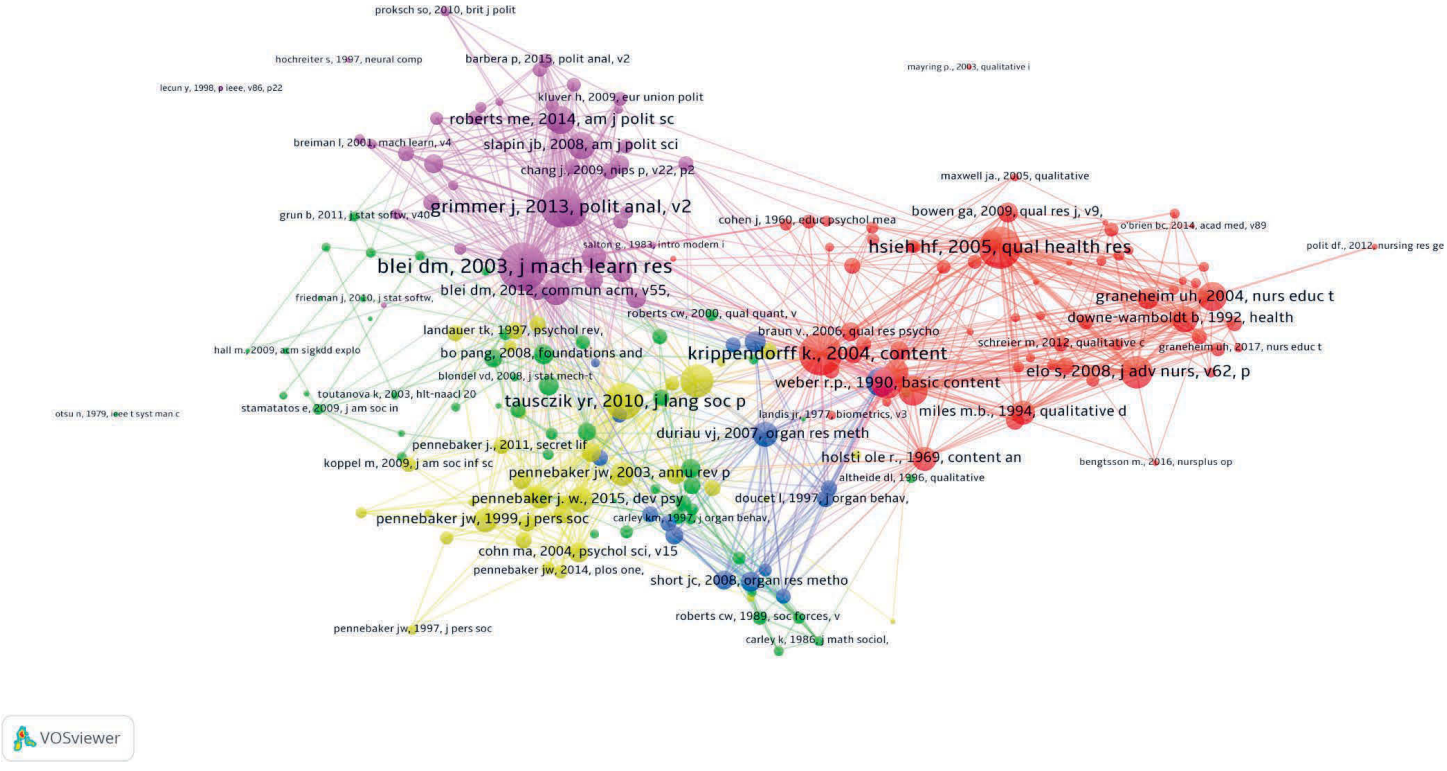
Zunächst zeigen wir Ihnen die Map der Ko-Zitationsanalyse. Dazu wurde die txt-Datei von WoS in VosViewer geladen. Als Auswahl, wie viele Zitationen in der Map angezeigt werden sollen, hat sich in unseren Studien eine Anzahl von um die 200 bewährt. Die Zahl können Sie bei VosViewer erreichen, indem Sie den *Threshold* entsprechend manipulieren. In unserem Beispiel liegt der *Threshold* bei einem Minimum von fünf Zitationen, sodass 218 Publikationen in der Map angezeigt werden. Wurde die Map erstellt, können Sie zum einen die Normalisationsmethode auswählen, wir bevorzugen hier „Association Strength". Ebenso können Sie das Layout verändern, sodass sich die Map zusammenzieht oder etwas weiter auseinandergezogen wird. Bei VosViewer ist die Einstellung „Attraction 2" und „Repulsion 1" automatisch eingestellt, wir haben für unser Beispiel „Attraction 6" und „Repulsion 2" gewählt, um die Map auseinander zu ziehen. Es gibt noch eine Reihe an Möglichkeiten, das Layout zu verändern: z. B. Farbe, Hintergrund, Größe der Symbole usw. Dazu können Sie auch gut in das Manual von VosViewer schauen *www.vosviewer.com/documentation/Manual_VOSviewer_1.6.17.pdf.*

Box 7.3: Erstellen von Maps mit VosViewer
Wir haben eine sehr ausführliche Anleitung geschrieben, wie Sie mit VosViewer arbeiten können. Diese finden Sie auf dem Blog „sozmethode" unter *https://sozmethode.hypotheses.org/1049.*

Wichtig zum Verständnis der Map der Ko-Zitationsanalyse (Abbildung 7.7, nächste Seite) ist nun, dass es nicht um eine Analyse der Inhalte der einzelnen Texte geht, sondern um die Analyse der Grafik und was die Anordnung der farblich unterschiedenen Cluster bedeutet. In unserem Beispiel gibt es fünf Cluster.

Als erstes fällt auf, dass die Punkte des violetten, gelben und roten Clusters größer sind als die der anderen Cluster. Größere Punkte bedeuten in der Map, dass die Artikel häufiger zitiert wurden und mit mehr anderen Publikationen in Verbindung stehen. Die zweite Erkenntnis ist, dass das rote (62 Publikationen) und violette Cluster dichter sind, d. h. die Punkte und Verbindungen klarer beieinanderstehen. Das führt dazu, dass diese Cluster größer erscheinen, obwohl sie von der sie umfassenden Anzahl an Publikationen nicht so unterschiedlich sind. So umfasst das rote Cluster 67, das violette Cluster 44, das grüne Cluster 54 und das gelbe Cluster 37, wohingegen sofort erkennbar ist, dass das blaue Cluster (16 Publikationen) kleiner ist. Drittens fällt auf, dass das grüne Cluster über die

Abbildung 7.7 Ko-Zitationsanalyse mit Daten aus WoS



ganze Map verteilt erscheint, also nicht richtig eingrenzbar ist. Gleichzeitig wird anhand der Map deutlich, dass das grüne Cluster kaum Berührungspunkte mit dem roten Cluster hat, das wiederum kaum Berührungspunkte mit dem violetten Cluster hat. Nun ist es spannend herauszufinden, warum das der Fall ist, also, warum die Cluster wie zueinander in Verbindung stehen.

Dazu laden wir uns bei VosViewer die Map-Daten herunter und können dann die erhaltene txt-Datei in Excel öffnen (siehe genaue Anleitung auf dem Blog). Wir erhalten eine Datei, in der die Publikationsabkürzungen, inklusive DOI oder Buchnamen, angegeben sind und in der die Clusternummer und die Zentralität anhand von drei Maßzahlen angegeben ist. Die Angaben der Zentralität sind „weight<Links>", also die Anzahl der Verbindungen zwischen den Publikationen, „weight<Total link strength>", also wie stark die Verbindungen sind und „weight<Citations>", also wie häufig die Publikationen zitiert wurden. Wir können die Excel-Liste nun nach Cluster und Zentralität der Publikation sortieren und über die DOI bzw. den Buchtitel die zentralen Publikationen der Cluster recherchieren, erfahrungsgemäß reichen dazu fünf bis 10 Publikationen pro Cluster (Steinhardt et al. 2017). Über die Recherche erhalten wir dann einen Überblick über die zentralsten Publikationen und können beurteilen, warum welche Cluster thematisch zusammenstehen. In unserem Beispiel heißt das:

- *Das rote Cluster:* Im roten Cluster sind die Gesundheitswissenschaften die zentrale disziplinäre Verortung. Es lassen sich zwei Schwerpunkte ausmachen. Einerseits die qualitative Inhaltsanalyse, die sich um die zentralen Publikationen clustern, die wir bereits aus der Literaturanalyse zur qualitativen Inhaltsanalyse kennen (Elo und Kyngäs 2008; Graneheim und Lundman 2004; Hsieh und Shannon 2005). Andererseits Grundlagenwerke zur Inhaltsanalyse mit einem verbindenden Element zur computerunterstützten Inhaltsanalyse (Krippendorff 2004) und zur quantitativen Inhaltsanalyse (Weber 1990). Das ist auch der Grund, warum diese beiden Texte den Übergang zum blauen und gelben Cluster bilden.
- *Das blaue Cluster:* Im blauen Cluster finden sich ausschließlich Publikationen aus den Wirtschaftswissenschaften (Duriau et al. 2007; Morris 1994; Short et al. 2010; Short und Palmer 2008), weshalb hier ein eigenes Cluster gebildet wurde. In den Publikationen wird die quantitative Inhaltsanalyse als Methode inklusive Nutzung von computerunterstützter automatischer Auswertung besprochen. Das zentrale Buch des Clusters (Neuendorf 2002) gibt eine Einführung in die quantitative Inhaltsanalyse und steht deshalb nah an den Büchern des roten Clusters.
- *Das gelbe Cluster:* Wie das rote und blaue Cluster ist auch das gelbe Cluster ein disziplinär geprägtes Cluster, nämlich der Psychologie. Die zentralen Publikationen behandeln Fragen, wie automatisierte Textanalyse Wörter in psychologisch sinnvollen Kategorien codieren kann (Newman et al. 2003; Penne-

baker et al. 2015; Tausczik und Pennebaker 2010). Auffällig ist in diesem Cluster zudem, dass es von dem Autor James W. Pennebaker dominiert wird, der bis auf das zentrale Buch bei allen zentralen Publikationen beteiligt ist. Auch im gelben Cluster gibt es ein Buch (Stone et al. 1966), das wie die Bücher der anderen Cluster zentral in der Gesamtmap liegt und die computerunterstützte Inhaltsanalyse behandelt.

- *Das violette Cluster:* Auch das violette Cluster hat mit der Politikwissenschaft einen disziplinären Fokus. Schwerpunkt der zentralsten Publikationen ist, wie politische Texte als Daten durch die computergestützte Inhaltsanalyse ausgewertet werden können (Grimmer und Stewart 2013; Laver et al. 2003). Genutzt wird hierfür das Verfahren des Topic Modeling, weshalb die Beschreibung des Verfahrens ebenso Raum einnimmt (Blei 2012; Blei et al. 2003; Roberts et al. 2014).
- *Das grüne Cluster:* Es bleibt noch das grüne Cluster, das sich in die anderen Cluster hinein erstreckt und nicht klar abgrenzbar ist. Dies liegt daran, dass es sich hier nicht um ein disziplinäres Cluster handelt, sondern dass es die Frage nach Bedeutung *(meaning)* von Text/Worten behandelt (Miller 1995; Mohr 1998), die mittels Verfahren und Methodiken der computergestützten Textanalyse (Popping 2000) ermittelt werden können, wie „Statistical Natural Language Processing" (Manning und Schütze 1999) oder Sentiment-Analyse (Pang und Lee 2008).

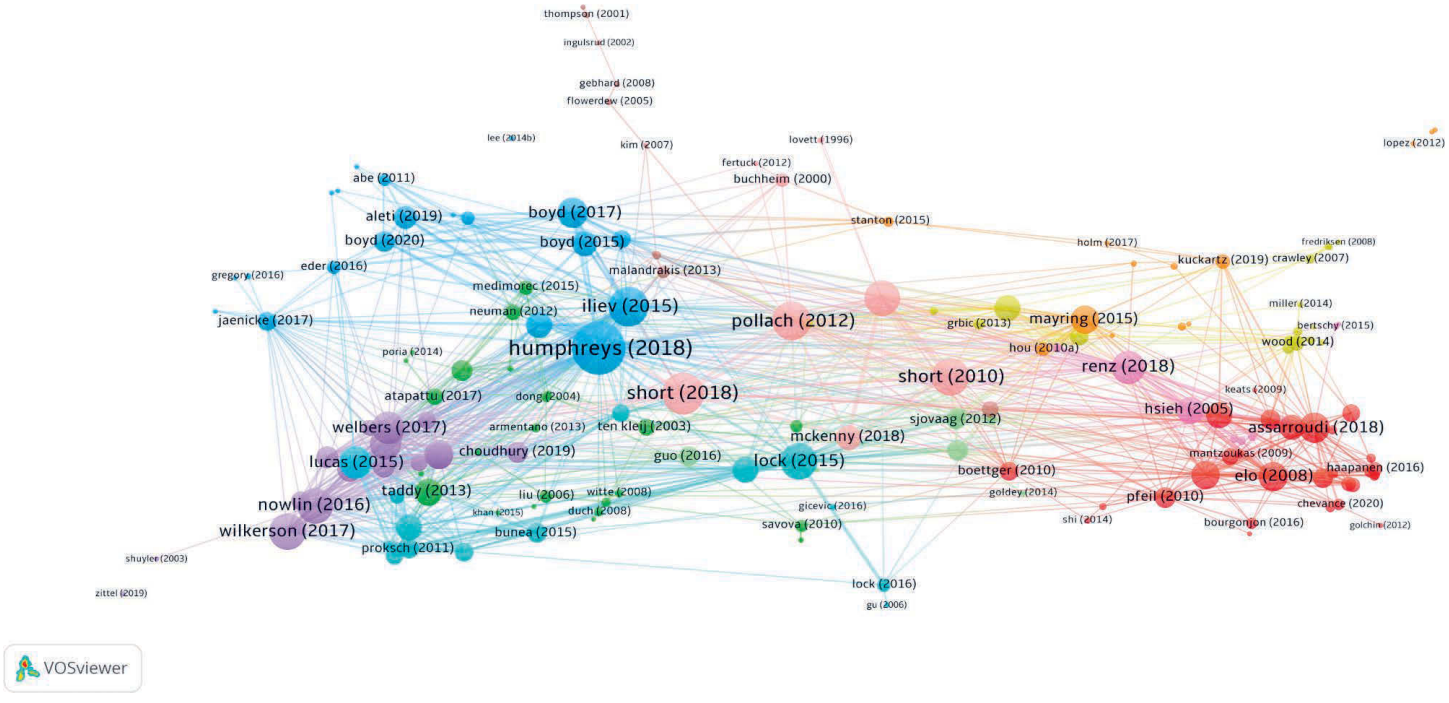
Zusammenfassend lässt sich also einordnen, dass das Forschungsfeld der Inhaltsanalyse durch eine Unterteilung in fachdisziplinäre Cluster gekennzeichnet ist und durch die Methoden vor allem der computergestützten quantitativen Inhaltsanalyse geprägt wird.

Bibliografische Kopplung

Für die Bibliografische Kopplung nehmen wir den gleichen Datensatz wie für die Ko-Zitationsanalyse und lassen uns bei VosViewer die Map erstellen. Dazu wählen wir auch bei der Bibliografischen Kopplung einen *Threshold* aus, der an die 200 Publikationen umfasst. Dieser ist bei elf Zitationen erreicht, wobei schlussendlich nur 177 Publikationen in die Map aufgenommen werden, da die anderen 23 Publikationen keine Verbindungen haben. Wie Abbildung 7.8 zeigt, liegt hier eine Map vor, die nur sehr schwer zu interpretieren ist. Trotz der geringen Anzahl an ausgewählten Publikationen ergeben sich keine klar voneinander abgrenzbaren Cluster, mit elf Clustern gibt es zudem sehr viele Cluster. Wir haben, um es zu testen, alle Varianten mit sehr niedrigem *Threshold* und sehr hohem *Threshold* ausprobiert und können Ihnen sagen, dass sich an der Map in der Übersichtlichkeit nichts verändert.

Was bedeutet das Ergebnis nun? Es bedeutet, dass die Inhaltsanalyse derzeit kein homogenes Forschungsfeld ist, sondern, wie die Ko-Zitationsanalyse auf-

Abbildung 7.8 Bibliografische Kopplung mit Daten aus WoS



zeigt, ein Feld ist, das sich aus sehr unterschiedlichen Disziplinen speist und auf unterschiedliche Wissensbasen zugreift.

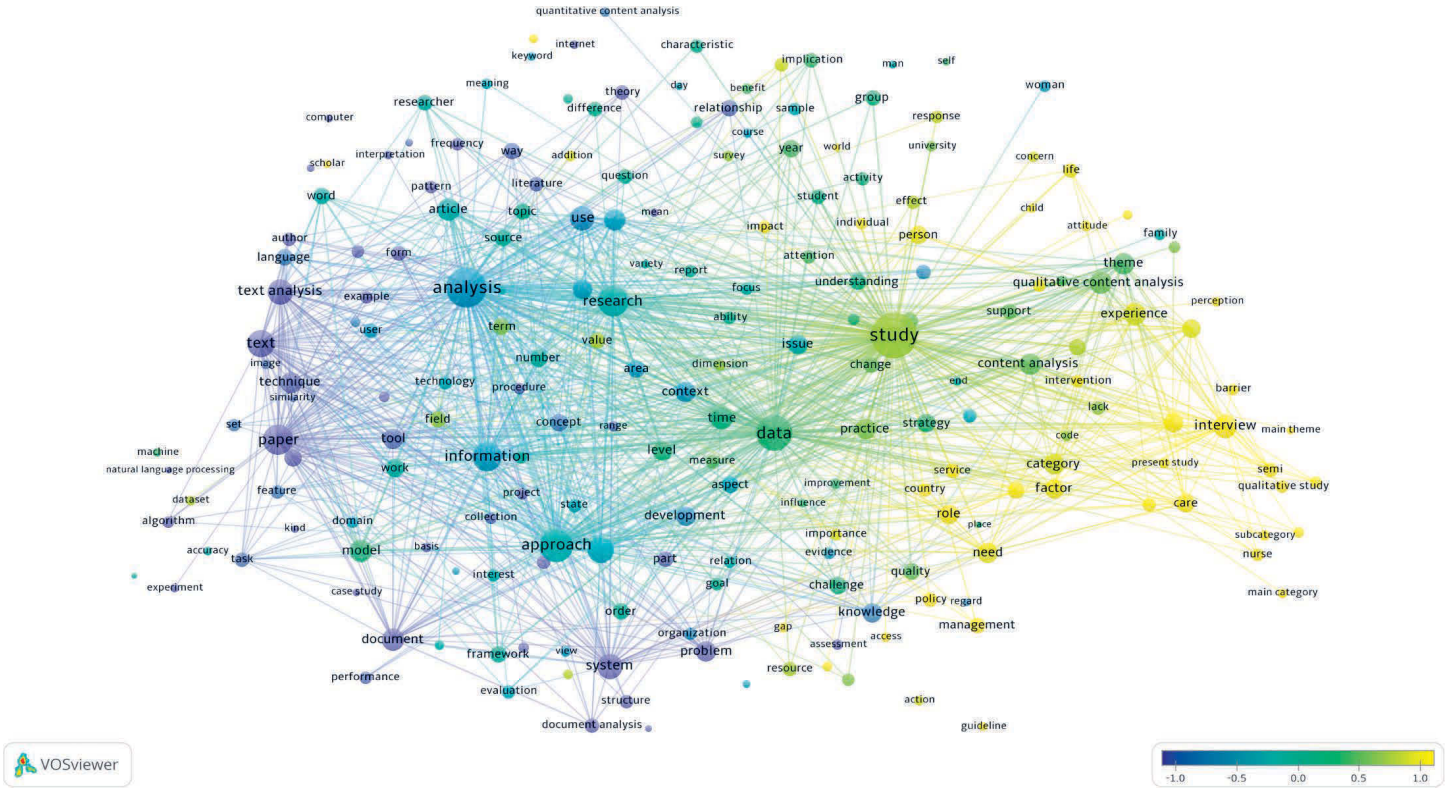
7.6.3 Themenanalyse

Nachdem wir die Datenbasis haben, geht es nun an die Datenauswertung. Für die Erstellung der Themenanalyse (Ko-Occurance Analyse) verwenden wir das Open Source-Tool VosViewer. Dieses können Sie unter *www.vosviewer.com* einfach herunterladen. VosViewer ermöglicht es Ihnen, die heruntergeladene txt-Datei von WoS einfach in das Programm einzuspeisen und dann die Analysen durchzuführen. Die Ko-Occurance Analyse bietet einen Überblick über die zentralen Worte, die im Literaturkorpus im Titel und/oder Abstract verwendet werden. Mit den Worten erstellt VosViewer eine Map, die Zusammenhänge zwischen den Worten darstellt. Dazu kann bei VosViewer zwischen „Binary Counting" und „Full Counting" gewählt werden. Bei „Binary Counting" wird die Anzahl der Dokumente angegeben, in denen ein Begriff mindestens einmal vorkommt. Bei „Full Counting" wird die Gesamtzahl der Vorkommen eines Begriffes in allen Dokumenten angegeben. Wir haben uns für Binary Counting" entschieden, da es uns um die Verbreitung bestimmter Begriffe geht. Bei „Full Counting" kann es leicht zu Verzerrungen kommen, wenn in wenigen Publikationen ein Begriff besonders häufig verwendet wird.

Als nächstes gilt es festzulegen, ab welcher Anzahl an Nennungen ein Begriff in die Map aufgenommen wird. Die Anzahl, die die Aufnahme eines Begriffs in die Map begrenzt, wird Threshold genannt. In unseren bisherigen Untersuchungen haben wir festgestellt, dass ein Threshold sinnvoll ist, der um die 200 Wörter in die Map aufnimmt. In unserem Beispiel war der Threshold bei 26 Nennungen erreicht, sodass 209 Wörter in die Map aufgenommen wurden. Für die Visualisierung haben wir uns für „Overlay Visualiation" entschieden, da durch den Farbverlauf (siehe Abbildung 7.9) die Verbindungen zwischen den Begriffen gut zu erkennen sind und sich gleichzeitig Cluster zeigen.

In der Map fällt uns als erstes auf, dass es mittig mehrere zentrale Punkte gibt: „analysis, approach, data, research, study". Das verwundert nicht, da dies Begriffe sind, die in jeder Publikation vorkommen, wenn empirisch gearbeitet wird. Aufschluss geben uns entsprechend eher die kleineren Punkte und der Farbverlauf. So zeigt uns die Map, dass am linken Rand ein dunkelblaues Cluster existiert, das die Analyse von Textprodukten, wie zum Beispiel „Paper" repräsentiert, die Verfahren der Textanalyse behandeln, die zum Beispiel durch Algorithmen unterstützt werden und in Verbindung zu den Sprachwissenschaften stehen. In Verbindung mit der Textanalyse steht die Dokumentenanalyse, die sich unten leicht links im Cluster findet. Die Dokumentenanalyse steht mit Begriffen in Verbindung wie „evaluation, organization, problem, system". Das zeigt an, dass hier die

Abbildung 7.9 Ko-Occurance Analyse mit Daten aus WoS



Analyse von Dokumenten von Organisationen untersucht werden, die sich auf bestimmte Probleme fokussieren. Rechts in der Map befindet sich ein gelbes und hellgrünes Cluster. In diesen steht die quantitative und qualitative Inhaltsanalyse im Mittelpunkt. Das gelbe Cluster wiederum zeigt mit „nurse, care, interview" an, dass es sich dabei um das Cluster in den Gesundheitswissenschaften handelt, welches wir auch bereits bei der Ko-Zitationsanalyse entdeckt haben.

7.7 Schritt 5: Interpretation der Ergebnisse

7.7.1 Literaturüberblick

In unserem Beispiel des Literaturüberblicks können wir nun zusammenfassen, dass erstens die englischsprachige Auseinandersetzung mit der Methodenbeschreibung und weiterentwicklung, die in der Datenbank WoS abgebildet ist, vor allem im Bereich der Gesundheitswissenschaften stattfindet. Zweitens lassen sich vier Kategorien von Literatur finden, die auf unterschiedliche und manchmal auch überlappende Inhalte fokussieren.

- Erstens die Zusammenfassung und Strukturierung der qualitativen Inhaltsanalyse. Besonderer Fokus liegt dabei auf dem Coding Prozess und dezidierten Anleitungen, wie die qualitative Inhaltsanalyse in ihren unterschiedlichen Varianten durchgeführt werden kann.
- Zweitens Texte, die sich mit Qualitätskriterien der Qualitativen Inhaltsanalyse befassen.
- Drittens Vergleiche bzw. Zusammenführung zwischen anderen Methoden und der Qualitativen Inhaltsanalyse.
- Viertens Texte, die auf Möglichkeiten der computergestützten qualitativen Inhaltsanalyse fokussieren.

Mit diesen Ergebnissen könnten Sie nun einen systematischen Überblick über die Literatur, also den Stand der Forschung, schreiben. Dieser kann Ihnen dabei helfen zu identifizieren, wo es noch Forschungslücken gibt bzw. wo sich die Forschung nicht einig ist und wo es spannend wäre, noch genauer hinzuschauen.

7.7.2 Mapping

Für das Mapping zeigt sich, dass es sich bei „Content Analysis" nicht um ein Forschungsfeld handelt, sondern um ein Thema, das in sehr unterschiedlichen Disziplinen eine Rolle spielt (siehe Ko-Zitationsanalyse). In den Disziplinen, die in der Map abgebildet sind, spielen dabei unterschiedliche Verfahren und Anwen-

dungsarten der Inhaltsanalyse eine Rolle. Für Ihre Arbeit könnten Sie nun zum Beispiel vertieft in die Analyse einzelner Cluster gehen. Oder Sie untersuchen, warum die Sozialwissenschaften kaum einen Beitrag in der Map haben.

7.7.3 Themenanalyse

Durch die Themenanalyse erhalten Sie einen guten Überblick über die Begrifflichkeiten, die in dem von Ihnen untersuchten Forschungsfeld zentral sind. Durch die Map mit farblichem Verlauf lassen sich die Begriffe zudem in Clustern darstellen, sodass Abgrenzungen sichtbar werden. Dadurch haben Sie die Möglichkeit, Interpretationen Ihrer Ergebnisse vorzunehmen.

8. Automatisierte induktiv-quantitative Inhaltsanalyse: Datenerhebung und -vorbereitung

In diesem Kapitel werden die verschiedenen Typen von online verfügbaren Daten und die Schritte zur Erhebung dieser Daten wie Webscraping usw. vorgestellt. Darüber hinaus werden verschiedene Typen automatisierter quantitativer Inhaltsanalyse und Probleme und Lösungen für diese Probleme besprochen. Zu diesen Problemen zählen der Umgang mit fehlenden, fehlerhaften, irrelevanten oder duplizierten Daten, dem Datenzugang, automatisierte Datenerhebung und eventuelle rechtliche Fallstricke. Dabei liegt der Fokus auf Webscraping, es werden aber auch andere Datenzugänge wie APIs und Online-Repositorien vorgestellt.

8.1 Einleitung

Durch die seit den 1990er Jahren stetig wachsende Nutzung des Internets und den damit verbundenen Kommunikationswegen bietet sich Forscher*innen die Möglichkeit zur Analyse online vermittelter Kommunikation, die aus unterschiedlichen Quellen stammen. Zu diesen Quellen zählen Social Media-Plattformen wie zum Beispiel Facebook, Blogs, Microbloggingdienste (z. B. Twitter oder Instagram), kommerzielle Plattformen (siehe Dolata 2018 für einen Überblick) und Rezensionsportale wie die Internet Movie Database (IMDB) oder Wikis (Wikipedia). In Box 8.1 finden Sie einige der gängigsten Datenquellen der Onlinekommunikation, die in bisherigen Studien für automatisierte quantitative Textanalyse verwendet werden. Sie stellen in einigen Bereichen der empirischen Sozialforschung mittlerweile die gängigen Datenquellen mit prozessproduzierten Daten (siehe Kapitel 3.2) zur Untersuchung gesellschaftlicher Phänomene dar, beispielsweise im Falle der Herausbildung politischer Haltungen und politischer Kommunikation (Silva und Proksch 2021; Stier et al. 2018).

Box 8.1: Für Forschung verwendete Datenquellen Social Media-Plattformen

Blogs: Online-Tagebücher, die von einzelnen bzw. wenigen Personen betrieben werden. In ihnen werden entweder persönliche Erfahrungen, oder themenbezogene Beiträge (Posts) veröffentlicht.

Microblogs: Online-Plattformen, auf denen viele Nutzer*innen Kurznachrichten mit wenigen Zeichen veröffentlichen.

Kommerzielle Plattformen: Hierunter werden Suchmaschinen, Videoplattformen wie YouTube, Handelsplattformen wie Amazon, Vermittlungsplattformen wie Airbnb und Crowdfundingplattformen wie Kickstarter subsumiert.

Rezensionsportale: Hierbei handelt es sich um Datenbanken und Websites, auf denen Nutzer*innen Bewertungen und Rezensionen zu Filmen, Musik, Büchern und anderen Medien verfassen und speichern können.

Im Rahmen von Online-Kommunikationen werden so ungeheure Mengen an Texten und damit verknüpften Daten wie Bilder und Videos erzeugt, dass sie von einzelnen Forscher*innen und Forschungsteams niemals in Gänze oder in einem hinreichend großen Ausmaß qualitativ ausgewertet werden können. Doch nicht bloß die bislang generierte Datenmenge[1] sprengt unsere Vorstellungskraft. Auch die Tatsache, dass jederzeit neue Kommunikationen, Bilder, Videos und Metadaten (z. B. über die genutzten Geräte, über lokale Ortung durch GPS-Standortverlauf) hinzukommen und an vorangegangene Kommunikationen anschließen, macht uns die umfassende, qualitative Auswertung der Datenmengen unmöglich. Ergänzend zum enormen Umfang zeichnen sich diese Datenmengen durch ihre Komplexität aus, die durch die Kombination von Texten mit anderen Datentypen wie Audio, Bild, Video und Zahlen erzeugt wird.[2]

Box 8.1 – Fortsetzung

Wikis: Website, auf denen Informationen bereitgestellt werden und zugleich durch deren Nutzer*innen kollaborativ ergänzt oder verändert werden können.

Foren: Website, auf der Diskussionen zu speziellen Themen (z. B. Textanalyseverfahren) bzw. Unterthemen geführt werden.

Internetarchive: Websites oder Datenbanken, auf denen entweder frühere Versionen von Websites durch sogenannte Wayback-Machines und archivierte Texte und Dokumente gespeichert sind.

---

1 Dabei beträgt allein die 2020 generierte Datenmenge mehr als 40 Zettabyte (Zetta = 10^{21}) und insgesamt befinden sich circa 850 Zettabyte an Daten auf Datenträgern (Cao et al. 2020). Bei der Annahme, dass ein Buch mit 100 Seiten (ohne Bilder) circa fünf Megabyte groß ist, dann würde allein die 2020 generierte Datenmenge 1 000 000 000 000 000 000 Seiten umfassen.

2 Eine Auswertung dieser Texte ist mit den gängigen Mitteln qualitativer und quantitativer Sozialforschung herausfordernd. Das liegt daran, dass Analysemethoden verwendet werden müssen, die in den Bereich „Big Data" fallen. Big Data weist nach Kitchin und McArdle (2016) vier Dimensionen auf: 1) *volume* = Datenmenge ist zu groß, um sie mit herkömmlichen Computern auszuwerten; 2) *variety* = große Vielfalt der Datentypen; 3) *velocity* = hohe Steigungsrate der Datenmenge, gepaart mit steigender Komplexität der inneren Verweisstruktur; 4) *veracity* = hohe Variabilität der Datenqualität.

Damit sind fünf Probleme verknüpft. Erstens ist die Textmenge zu groß, um sie in gebotener Tiefe und im Detail zu sichten. Zweitens weisen online generierte Textkorpora eine Verweisstruktur auf, die beispielsweise in Form von Hyperlinks oder Hashtags vorliegt. Somit ist der Text immer in Bezüge eingebettet, die über diesen hinausweisen. Drittens tritt ein Text häufig in Verknüpfung mit anderen Datentypen auf (Bilder, Videos, Audiodateien). Somit muss man Entscheidungen darüber treffen, welche Datentypen und Informationen zusätzlich in die Analyse einbezogen werden und welche Methoden für diese zusätzliche Auswertung herangezogen werden. Viertens treten diese Korpora in teilstrukturierter Form auf, deren Aufbereitung und Auswertung (noch) nicht Teil der methodischen Ausbildung von Soziologiestudent*innen ist. Darunter versteht man Informationen, die nicht in die gewohnten Tabellen- bzw. Datenmatrix-Strukturen gespeichert sind, sondern als Teil einer Datenbank Informationen mit sich tragen, die zum Beispiel auf andere Texte, Websites, Personen oder Verweise (Hashtags) hinweisen (Buneman 1997). Fünftens sorgt die Generierung von Textdaten (z. B. auf Twitter) in Echtzeit dafür, dass die Erfassung von Diskurs- und Sinnstrukturen mit traditionellen Methoden qualitativer und quantitativer Sozialforschung nicht möglich ist.

Damit wir in diese Datenwelten eintauchen können und uns darin nicht verlieren (Diaz-Bone et al. 2020), können wir automatisierte Methoden der induktiven quantitativen Textanalyse anwenden. Induktive Verfahren ermöglichen es uns, Muster in großen Textmengen zu erkennen und deren manifeste und latente Inhalte und damit kommunizierte Sinngehalte zu erarbeiten. Zur Erinnerung: Der manifeste Inhalt zielt auf die Beantwortung der Frage ab, was direkt im Text kommuniziert wird, während der latente Inhalt auf das „Warum" der Kommunikation gerichtet ist. Die Analyse latenter Sinngehalte hat damit das Ziel, die in Kommunikationen impliziten, d. h. verborgenen Bedeutungen zu erfassen (siehe Kapitel 2.1). Dieser Sinn wird dabei von den Forschenden rekonstruiert bzw. zugeschrieben.

Manifeste Inhalte können wir bei automatisierten induktiv-quantitativen Textanalysen beispielsweise durch Auszählungen von Wörtern, Wortkombinationen, Wortumfeldanalysen und Scores (auf Deutsch: Punkte) ermitteln, beispielsweise, wie häufig positiv oder negativ konnotierte Wörter im Zeitverlauf vorkommen (siehe Sentiment-Analyse, Kapitel 10). Der latente Sinngehalt kann in der Logik der automatisierten quantitativen Textanalyse durch die Rückführung manifester Inhalte auf (unterstellte) Ursachen erschlossen werden. Das bedeutet, dass wir zum Beispiel versuchen, systematisch gemeinsam auftretende Wörter über Texte hinweg zugrundeliegenden Themen zuzuordnen. Hier greift die Annahme, dass Personen bestimmte Wörter wählen, um über ein Thema zu sprechen – und dadurch die Wahrscheinlichkeit vorstrukturieren, auf bestimmte Wortkombinationen in den einzelnen Texten zu treffen (Blei 2012; Chang et al. 2009). Themen liegen in dieser Sichtweise nicht offen in den manifesten einzelnen Aussagen vor, sondern ermitteln sich erst über den Text und in Summe vieler Texte hinweg und sind daher in der Kommunikation latent vorhanden. Themen werden durch mathematische Modelle berechnet, die versuchen, diese Muster (d. h., wiederkehrende, gemeinsam auftretende Wörter und Wortketten) zu identifizieren und in neuen Texten zu finden. Normalerweise wird dazu der Textdatensatz (in Soziolinguistik: Korpus) aufgeteilt, und zunächst nur an einem Teil die Themen ermittelt und dann geprüft, inwiefern die ermittelten Themen auch beim anderen Teildatensatz vorliegen. Die statistischen Verfahren, die Themen durch Wortzusammenhänge extrahieren und prognostizieren wollen, stellen den größten Unterschied zu qualitativen Verfahren dar, bei denen der latente Sinngehalt durch Erklären, Interpretieren und Kontextualisierung ganzer Texte oder ausgewählter Textstellen durch die*den Forscher*in erfolgt. Qualitative und quantitative inhaltsanalytische Auswertungstechniken haben somit spezifische Stärken, welche in einem Mixed Methods-Forschungsdesign (z. B. Burzan 2016; Kelle 2008; Kelle 2019) kombiniert werden können.

Automatisierte, d. h., computergestützte Verfahren der Textanalyse sind den Bereichen der Computerlinguistik und der Künstlichen Intelligenz zuzuordnen. Ihre Hauptaufgabe besteht darin, Bedeutungen und Muster innerhalb der

menschlichen Sprache (= natürliche Sprachen wie Deutsch oder Englisch)[3] zu erkennen, Gesprochenes zu transkribieren, automatische Übersetzungen anzubieten oder Stimmungen zu erkennen, die im Text transportiert werden (Otter et al. 2020). Dabei lassen sich die Versuche, Texte automatisiert zu analysieren, bis in die 1950er Jahre zurückverfolgen (Deng und Liu 2018, S. 2–18). Grundlage hierfür waren die generative Grammatik von Noam Chomsky und die Annahme, dass Schlüsselelemente der Sprache im menschlichen Gehirn repräsentiert sind und somit mit der angeborenen Fertigkeit des Menschen verknüpft sind, Sprache und Sinn zu verstehen (Chomsky 2009). Diese „regelgeleitete Phase" hielt bis in die 1980er Jahre an, ehe sie durch ein empiristisches Paradigma ersetzt wurde, bei dem Texte bzw. menschliche Sprache mittels statistischer Verfahren und darin angelegter Modelle analysiert wurde (Murphy 2012).[4] Dabei wurden die Modelle zunächst anhand von Daten „trainiert" und danach genutzt, um ähnliche Muster in anderen Datenkorpora zu erkennen. Seit den 1990er Jahren wird zunehmend auf vielschichtige neuronale Netzwerke und sogenannte Deep-Learning-Methoden zurückgegriffen, um Muster in großen Textdatenmengen zu erkennen und die Inhalte zu gruppieren bzw. zu klassifizieren.[5]

Da neuronale Netzwerke allerdings eine Black Box mit undurchsichtigen Berechnungswegen sind und somit unklar ist, wie sie ihre Ergebnisse erzeugen (Dinov 2018), wollen wir Ihnen in der Folge Verfahren der automatisierten computergestützten Textanalyse zeigen, deren Ergebnisse und einzelne Arbeitsschritte von Ihnen interpretiert werden können. Diese Interpretationsmöglichkeit erlaubt es uns erst, Verfahren maschinellen Lernens mit Methoden der qualitativen Inhaltsanalyse zu verknüpfen. Vor diesem Hintergrund ist das Ziel dieses Kapitels 8, eine Einführung in die automatisierten Methoden quantitativer Textanalyse zu geben. Dabei werden die Verfahren der Korrespondenzanalyse (siehe Kapitel 9), der Sentiment-Analyse (siehe Kapitel 10) und der Latent Dirichlet Allocation (LDA, eine Form des Topic Modeling) (siehe Kapitel 11) vorgestellt. Da wir Ihnen auch Befehle und Programmcodes zeigen, werden wir die Schriftart `Lucida Console` für R-Programmcode und `Consolas` für Python-Programm-

3 In Abgrenzung dazu gibt es formale Sprachen, zu denen Programmiersprachen wie Python oder R gehören. Sie sind dadurch gekennzeichnet, dass sie Befehle und Algorithmen in für Menschen lesbarer Form anzeigen und diese zugleich für den PC ausführbar machen.

4 Zu diesen Verfahren zählen die „support vector machine", der Expectation-maximization-Algorithmus, Entscheidungsbäume und bayesianische Netzwerke. In etwa zeitgleich traten Formen neuronaler Netzwerke auf, die extrem flexibel auf die eingegebenen Daten reagieren konnten.

5 Beim Deep-Learning wird eine dem Aufbau des menschlichen Gehirns nachempfundene Programmierung verwendet, bei der ein Input durch eine vielschichtige, „neuronale" Struktur, deren Verbindungen sich in jedem Berechnungsschritt verändern können, verarbeitet wird und in einen Output (z. B. die Einordnung eines Textes in eine Kategorie) übersetzt, der den Nutzer*innen dann angezeigt wird. Diese Neuverschaltung der künstlichen Neuronen ist Lernprozessen im menschlichen Gehirn nachempfunden (Schmidhuber 2015).

code verwenden. Dabei handelt es sich um die Schriftarten, die in den von uns verwendeten grafischen Oberflächen RStudio und Spyder standardmäßig eingestellt sind. Codezeilen, Ein- und Ausgaben werden farblich hervorgehoben. Codezeilen in R werden in grau, in Python in rosa in den Textfluss eingebettet, Ausgaben in R und Python werden mit schwarzem Hintergrund angezeigt (siehe zu Copylefts und Copyrights Kapitel 1.3).

8.2 Typen automatisierter Verfahren der quantitativen Textanalyse

In den weiteren Unterkapiteln orientieren wir uns an den folgenden Leitfragen, die den Ablauf einer Untersuchung mit Verfahren der automatisierten quantitativen Textanalyse widerspiegeln.

- Welche Typen von inhaltsanalytischen Verfahren gibt es?
- Woher bekomme ich die Daten für die automatisierte quantitative Textanalyse? Was muss ich dabei beachten?
- Wie führe ich computergestützt eine automatisierte quantitative Textanalyse durch?
- Wo finde ich online Hilfe, wenn ich bei bestimmten Punkten nicht weiterkomme?
- Wie verknüpfe ich computergestützte quantitative und qualitative Textanalyse miteinander?
- Was sind die Limitationen automatisierter, quantitativer Textanalyse?

8.2.1 Maschinelles Lernen: Definition und Anwendungsgebiete

Zuerst können wir festhalten, dass es unterschiedliche Typen von automatisierten Verfahren der quantitativen Textanalyse gibt. *Automatisiert* soll heißen, dass die einzelnen Erhebungs-, Analyse- und Qualitätssicherungsaspekte ohne menschliches Zutun nach zuvor festgelegten, für den Computer anwendbaren Regeln durchgeführt werden. Diese Verfahren unterscheiden sich sowohl hinsichtlich des Grades, in dem menschliches Expert*innenwissen in die Analysen einfließt, als auch hinsichtlich des Zeitpunktes innerhalb des Forschungsprozesses, an dem diese Expertise eingebracht wird. Dabei wird zwischen nicht-überwachten und überwachten Methoden maschinellen Lernens

Box 8.2: Definitionen
Automatisiert = Durchführung von mindestens eines Aspektes des Forschungsvorhabens durch den Computer.

Maschinelles Lernen = Vorgang, bei dem der Computer anhand von Eingaben und statistischer Modelle letztere durch die Eingabe neuer Daten verändert und zuvor unbekannte Fälle berechnen kann.

unterschieden, die den jeweiligen Verfahren zugrunde liegen. *Maschinelles Lernen* bezeichnet einen Vorgang, bei dem ein Computer durch die Verwendung von Algorithmen zunächst Modelle erzeugt und schrittweise durch den automatisierten Einbezug neuer Informationen anpasst (Molina und Garip 2019). Sie werden bei der Mustererkennung (z. B. Detektion von Themen in Texten, Gesichtserkennung, Erkennung von Tumoren in der Medizin, Unterscheidung zwischen E-Mails und Spam) angewandt und nutzen statistische Berechnungen, um diese Muster in noch unbekannten Daten wiederzuerkennen.

8.2.2 Ziele von Verfahren maschinellen Lernens im Bereich der automatisierten, quantitativen Textanalyse

Das erste Ziel von Verfahren des maschinellen Lernens im Bereich der automatisierten quantitativen Textanalyse ist, latent vorliegende Strukturen zu erkennen, die einen Textkorpus charakterisieren. Das kann das zahlenmäßige Auftreten von Wörtern oder Wortkombinationen oder deren Einbettung in Textzusammenhänge meinen. Zweitens werden Verfahren maschinellen Lernens im Rahmen sozialwissenschaftlicher Analysen von Texten verwendet, um (potenziell) im Text als manifeste Inhalte vorliegende Themen zu erkennen, die in einer Vielzahl von Texten auftreten. Drittes Ziel solcher Verfahren ist es, die in den Texten enthaltenen Stimmungen (Englisch: *sentiments*) zu identifizieren. Das können beispielsweise subjektive Aussagen sein, die positive oder negative Wertungen gegenüber einer Sachlage, einer Person oder Gruppe enthalten. Es kann bei einer entsprechend feinen Analyse sogar sein, dass spezifischere Stimmungslagen in Texten untersucht werden können, die beispielsweise mit Rassismus (Zainuddin et al. 2018), Hate-Speech (Badjatiya et al. 2017) oder Diskriminierung von Bevölkerungsgruppen wie beispielsweise Migrant*innen verknüpft sind (Bosco et al. 2017; Ozduzen et al. 2020). Zuletzt werden diese Verfahren genutzt, um zuvor erkannte Muster in neuen Texten wiederzuerkennen. Dieser Verwendungszweck der Verfahren automatisierter quantitativer Textanalyse wird hauptsächlich in der Informatik verwendet. Dabei steht nicht im Vordergrund, ein Phänomen umfassend zu verstehen und zu erklären, sondern Muster oder Verhaltensweisen möglichst akkurat zu messen und durch Schätzen zu prognostizieren.

Im Bereich der automatisierten, quantitativen Textanalyse lassen sich Methoden verorten, die den Bereichen des *nicht-überwachten* und überwachten maschinellen Lernens zuzuordnen sind. Das Ziel von Methoden des nicht-überwachten maschinellen Lernens ist es, Strukturen innerhalb von Daten ohne die Zuarbeit von Forscher*innen zu identifizieren (Solan et al. 2005). Sie beruhen auf statistischen Verfahren, die geeignet sind, um Zusammenhänge zwischen Datenpunkten (z. B. Begriffen innerhalb von Texten) zu erkennen. Im Fall von Texten werden zwei Eigenschaften der Sprache genutzt, um diese Zusammenhänge zu

erkennen. Erstens wird ausgenutzt, dass menschliche Sprachen Regeln folgen, die den Sprecher*innen vorgeben, wie Sätze strukturiert sind und wie Aussagen innerhalb dieser geregelten Struktur formuliert werden sollen. So ist es beispielsweise im Englischen üblich, erst Subjekt, dann Prädikat und zuletzt ein Objekt zu verwenden und um an diese Kernstruktur weitere Informationen anzulagern. Zweitens wird die Tatsache ausgenutzt, dass bestimmte Wörter und Redewendungen genutzt werden, um Situationen, Handlungen, oder Tatsachen zu beschreiben. Zu diesen Verfahren zählen verschiedene Arten des *Topic Modelings*, die (multiple) *Korrespondenzanalyse, Hauptkomponentenanalyse* oder *multiple Faktorenanalyse.* Von den aufgezählten Verfahren werden die ersten beiden im Verlauf der folgenden Kapitel erläutert und deren Anwendung anhand von Beispielen in RStudio und Python demonstriert.

Dagegen benötigt überwachtes maschinelles Lernen menschlichen Input, um bestimmte Muster zu erkennen und gegebenenfalls bewerten zu können. Im Kontext computergestützter, induktiv-quantitativer Textanalyse sind dabei *Sentiment-Analysen* (Liu 2012) von besonderer Bedeutung. Sie nutzen zuvor erhobene Bewertungen von Wörtern oder Phrasen, um deren Stimmung als positiv, neutral oder negativ zu kategorisieren oder auf einer Skala (z. B. von −1 = sehr negativ bis +1 = sehr positiv) anzuordnen. Dadurch lassen sich beispielsweise Stimmungslagen in Bezug auf politische Entscheidungen oder Reformen, parlamentarischer Debatten (Abercrombie und Batista-Navarro 2020) oder gegenüber Migranten (Heidenreich et al. 2020), oder der Stimmungslagen in Musik (Napier und Shamir 2018), Bewertungen von Kritiker*innen gegenüber Filmen oder Romanen, oder positive/negative Bewertungen von Kund*innen im Falle von Produkten nachzeichnen.

8.2.3 Beispiele für die Funktionsweise der Verfahren der automatisierten quantitativen Textanalyse

Gehen wir an dieser Stelle zum Beispiel der alten Dame aus Kapitel 2 in diesem Buch zurück, die beim Bäcker nach ihrem Befinden gefragt wird, dann ließe sich durch nicht-überwachte Verfahren maschinellen Lernens durch die Verwendung ihrer Wörter und der Struktur ihrer Aussagen feststellen, worüber sie gesprochen hat. Vorausgesetzt, man hat im Vorfeld die Antworten vieler Personen auf die Frage „Wie geht es dir?“ erfasst, dann würde der Computer nun versuchen, die spezifische Antwort der alten Dame in eine oder mehrere zuvor berechnete Kategorien einzuordnen. Dabei könnte es sein, dass ihre Aussage in eine Kategorie eingeordnet wird, die wir als situationsadäquate Antwort interpretieren – beispielsweise, wenn sie „Danke, mir geht es gut“ antwortet. Vielleicht wird die Aussage der alten Dame aber auch einer Kategorie zugeordnet, die durch Begriffe charakterisiert ist, die mit rheumatischen Beschwerden assoziiert sind. Es ist aber

auch möglich, dass ihre Antwort in mehrere Kategorien eingeordnet wird, beispielsweise eine, die auf einen früheren Arbeitskontext hinweist, indem wir uns vorstellen, dass sie Ärztin gewesen ist („Patienten", „Praxis" und „medizinische Fortbildung"), oder die auf die Beschreibung familiärer Verhältnisse hindeuten (z. B. Schwester, Schwiegersohn, Enkel und Cousin). Es kann aber ebenso sein, dass die Aussage der alten Dame in Kategorien eingeordnet wird, die zwar für den Computer Sinn ergeben, aber für den Menschen nicht interpretierbar sind. Daher ist es zwangsläufig nötig, dass angewandte Verfahren des nicht-überwachten maschinellen Lernens wiederholt kontrolliert werden.

Neben den reinen Eigenschaften der Sprache nutzen diese Verfahren auch Eigenschaften von Sprecher*innen aus, um innerhalb von Texten Strukturen zu erkennen. Sie erkennen beispielsweise ein charakteristisches Vokabular, das nach Schichtzugehörigkeit, Generation, politischer Gesinnung oder (sub-)kulturspezifisch variieren kann (sogenannte Soziolekte, siehe Guy 2013). Zwei Beispiele lassen sich an dieser Stelle zur Illustration heranziehen. Das erste Beispiel stammt aus der 2015 und 2016 stattgefundenen Flüchtlingskrise, die durch den Syrienkonflikt hervorgerufen wurde. Flüchtlinge werden bei dieser Debatte beispielsweise mit Begriffen wie „Abschiebung," „Asyl" und „Schleuser," mit „Integration," „Ausbildung," und „Bleiberecht" oder aber der Bekämpfung von Fluchtursachen in Verbindung gebracht. Vertreter*innen politischer Parteien haben dabei je nach ihrer Position zur Migrations- und Außenpolitik eine höhere oder geringere Wahrscheinlichkeit, bestimmte Begriffskombinationen in ihren Reden oder der öffentlichen Kommunikation zu verwenden oder sogar polemische und populistische Aussagen zu tätigen (z. B. Stier et al. 2017). Ein zweites Beispiel findet sich innerhalb der Soziologie, wenn sich Vertreter*innen verschiedener Denktraditionen über ein- und dasselbe Thema äußern. So werden sich beispielsweise Forscher*innen, die einen diskursanalytischen Ansatz (z. B. Diaz-Bone 2006; Fairclough 2013; Maeße und Sparsam 2017) verfolgen und Macht- und Herrschaft untersuchen, anderer Termini bedienen als Vertreter des Strukturfunktionalismus (Luhmann 1984; Münch 1986; Parsons 1968) oder der Konflikttheorie (Collins 1990; Coser 1957).

8.2.4 Textkorpora und geeignete Datengrundlagen für die Erstellung eines eigenen Textkorpus

Um eine automatisierte, quantitative Inhaltsanalyse durchzuführen, benötigen wir zunächst einmal Textdaten, die nach der Erhebung für die Auswertung aufbereitet werden müssen. Doch wie kommen wir an diese Daten – und welche Form müssen sie haben?

Grundsätzlich gibt es zwei Möglichkeiten, um Textdaten für eine automatisierte quantitative Inhaltsanalyse zu gewinnen. Entweder wir greifen auf

bereits bestehende Textkorpora zurück, oder wir erstellen selbst einen Datenkorpus durch manuelles oder automatisiertes Herunterladen von Texten und dazugehörigen Kontextinformationen (sogenannte Metadaten wie z. B. Veröffentlichungsort und Adressat*innen). Als Textkorpus wird eine Sammlung von Texten bezeichnet, die einer bestimmten Struktur (z. B. eine Tabelle mit klar bezeichneten Spaltennamen) folgt. Um automatisierte Verfahren als Auswertungstechniken der quantitativen Inhaltsanalyse verwenden zu können, werden entsprechend große Textkorpora benötigt. Diese können wenige hundert Texte umfassen, aber auch Millionen Texte beinhalten. Dabei gilt, dass die Genauigkeit der angewendeten Verfahren und die Anzahl der gefundenen Themen von der Anzahl der Texte und der Textlänge abhängen, die zur Verfügung stehen (Sbalchiero und Eder 2020).

Box 8.3: Zentrale Konzepte für die Datenerhebung und Datenaufbereitung
Korpus: Strukturierte Sammlung von Texten oder Wörtern.

Repositorium: Datenmanagementsystem mit einer großen Auswahl von Datensätzen zur Sekundäranalyse.

Webscraping: Automatischer Download von Informationen von Websites.

Advanced Programming Interface (API): Programmierschnittstelle, die gezielte Anfragen an Datenbanken ermöglicht.

Es gibt dabei eine Vielzahl von Korpora, darunter das *Deutsche Referenzkorpus* (DeReKo), den *Google Ngram Corpus,* oder das *Corpus of Contemporary American English.* Das DeReKo existiert seit 1965 und umfasst zeitgenössische Romane, Gedichte, wissenschaftliche Texte, Zeitungstexte und wissenschaftliche Texte (Kupietz et al. 2010).[6] Es umfasst mehr als zwei Millionen Wörter und ist der größte, deutschsprachige Korpus, der derzeit verfügbar ist. Der *Google Ngram Corpus* enthält Wörter und Wortketten (sogenannte N-Grams, wobei n für die Anzahl der verketteten Wörter steht). Basis hierfür sind digitalisierte Bücher, die auch auf *Google Books* einsehbar sind. Der Datenkorpus enthält dabei die Wörter und Wortketten aus 6 % aller Bücher, die jemals publiziert worden sind (Lin et al. 2012). Das *Corpus of Contemporary American English* umfasste zuletzt mehr als 400 Millionen Wörter und mehrere hunderttausend Texte (Davies 2010). Die Grundlage sind Transkripte von TV- und Radiosendungen, Belletristik und Poesie, Zeitungen und wissenschaftliche Publikationen.

Daneben gibt es eine Reihe spezialisierter, offline generierter (und zumeist online verfügbar gemachter) Textkorpora. Diese wurden in der bisherigen soziologischen Forschung fruchtbar gemacht und umfassen folgende Texttypen.

- Parlamentsdebatten (Curran et al. 2018; Fuhse et al. 2020),
- Zeitungsartikel (DiMaggio et al. 2013),

6 Dieser Korpus kann nach Registrierung unter folgendem Link genutzt werden: *https://cosmas2.ids-mannheim.de/cosmas2-web.*

- wissenschaftliche Publikationen (Daenekindt und Huisman 2020; Grothe-Hammer und Kohl 2020; Schwemmer und Wieczorek 2020; Wieczorek et al. 2021), aber auch
- Belletristik, Lyrik oder Liedtexte (Devi und Saharia 2020; Kozlowski et al. 2019).

Bei diesen Daten handelt es sich um „medium data" (Heiberger und Riebling 2016; Riebling 2018) (d. h. am einzelnen PC lesbare, mittelgroße und mittelkomplexe Daten). Damit ist gemeint, dass es sich um prozessproduzierte Daten mit einer komplexen Struktur handelt, die durch menschliche Einwirkung (z. B. Transkription von Parlamentsdebatten) generiert wurden. Diesen als Dokumenten vorliegenden Daten fehlt allerdings der dynamische Aspekt von Big Data, d. h. deren Produktion und Erfassung in Echtzeit sowie die Vielfalt der erfassten Datentypen, die ausgewertet werden können.

8.2.5 Datenzugänge

Um auf solche Textkorpora zurückgreifen zu können oder sie selbst zu generieren, stehen dabei generell vier Möglichkeiten zur Verfügung.

- *Repositorien:* Bei einem Repositorium (auf Englisch: *content repository*) handelt es sich um ein Datenmanagementsystem. Repositorien sind große Datenbanken, in denen bereits aufbereitete und kommentierte Datensätze angelegt sind. Eine der größten ist die Harvard Database, in der mehrere zehntausend Datensätze zu Replikationszwecken gespeichert werden und durch die Zuweisung einer DOI zitierfähig sind. Dabei können Nutzer*innen in den Suchfeldern gezielt nach Textkorpora suchen. Ein in Deutschland kuratiertes Repositorium ist beispielsweise das Datenarchiv der Sozialwissenschaften der GESIS.
- *Application Programming Interfaces (APIs):* APIs sind Schnittstellen zwischen Computerprogrammen und Datenbanken, die es mittels gezielter Anfragen ermöglichen, große Datenmengen herunterzuladen. Beispiele hierfür sind die Amazon API, Twitter API oder Reddit API. Um die jeweiligen Anfragen stellen zu können, müssen Sie sich zunächst bei den Diensten, mit denen die API verknüpft ist, als Nutzer*in registrieren. Nach der Registrierung bekommen Sie einen Nutzungsschlüssel per Mail oder im jeweiligen Dienst zugeschickt bzw. zugewiesen. Mit dessen Hilfe können Sie auf der Seite selbst oder in Programmierumgebungen (R oder Python) Datenanfragen stellen. Diese werden, sofern die abgerufenen Daten in der Datenbank vorliegen, zwischengespeichert und müssen dann in eigene Dateien oder ein geteiltes Datenformat abgespeichert werden. APIs sind nicht zuletzt Programmierschnittstellen, mit deren Hilfe Forscher*innen gezielt Anfragen an Datenban-

ken stellen können, die ihnen (mehr oder minder detaillierte) Informationen zurückgeben.
- *Manuelle Eingabe (z. B. Transkription):* Dies ist nur mit erheblichem zeitlichem Aufwand möglich und wird daher als Datenzugang in diesem Kapitel nicht weiter berücksichtigt.
- *Webscraping:* Beim Webscraping wird hingegen öffentlich zugänglicher Content von zuvor angegebenen Websites automatisch heruntergeladen.

8.2.6 Technische Umsetzung der Online-Datenerhebung durch Webscraping

Beim Webscraping gibt es aus technisch-praktischer, rechtlicher und ethischer Sicht einige Dinge zu beachten, wenn Sie mit diesen Daten arbeiten möchten. Hier ist festzuhalten, dass Probleme und Anwendung dieser Datenerhebungsverfahren in der Literatur zwar umfassend von der technischen Seite beleuchtet wurden, die ethischen und juristischen Konsequenzen aber in der Fachliteratur kaum diskutiert werden (Krotov und Tennyson 2018). Doch auch die ethische und rechtliche Dimension muss zumindest in gebotener Kürze angeschnitten werden, damit Sie sich der Fallstricke bewusst sind, die mit der Online-Datenerhebung einhergehen. Ethische Bedenken ergeben sich aus dem Webcrawling im Forschungsprozess, dass dies keinen direkten Nutzen für die Betreiber*innen der gecrawlten Websites und deren Nutzer nach sich zieht (Thelwall und Stuart 2006). Dazu zählt, dass der Zugriff auf die Website bei der Datenerhebung die Geschwindigkeit des Seitenabrufs für Dritte verringert, bei zu hoher Anfragezahl des Crawlers sogar verunmöglicht wird. Daneben gibt es Probleme mit der Privatsphäre und der Anonymität der Nutzer*innen, deren Daten abgerufen und von der*dem Forschenden gespeichert werden.

In diesem Zusammenhang betonen Gold und Latonero (2017, S. 300–302) zurecht, dass Webcrawler schnell auf sensible, persönliche Daten zurückgreifen können, wenn (Text-)Daten und andere Inhalte einer oder mehrerer Websites erhoben werden. Dazu zählen neben den textuellen Inhalten (z. B. Blogtexten) Kommunikationen inklusive Informationen über deren Adressat*innen, Selbstbeschreibungen sowie den Zeitpunkt, an dem die Kommunikation stattgefunden hat und den Ort, von dem aus die Kommunikation abgesendet wurde. Mit den im weiteren Verlauf des Kapitels angesprochenen Analysemethoden können mit diesen Daten politische oder sexuelle Präferenzen, religiöse Haltungen oder Weltsichten rekonstruiert und Nutzer (selbst mit Pseudonym) identifiziert und geschädigt werden. Ferner sollten wir uns vor Augen führen, dass Nutzer*innen von Webdiensten in den seltensten Fällen die allgemeinen Geschäftsbedingungen (AGBs) gelesen haben und somit oftmals keine Vorstellung von dem Ausmaß haben, in dem die Daten erhoben und ausgewertet werden (Obar und Oeldorf-

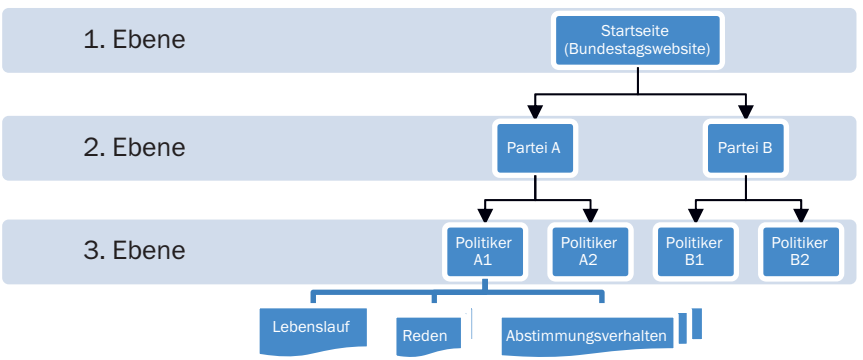
Hirsch 2020). Anders ausgedrückt können Webcrawler – auch in den Händen von Forschenden – zu Überwachungszwecken missbraucht werden, was weder wünschenswert ist noch Ziel eines Forschungsunterfangens sein sollte.

Darüber hinaus können Webcrawler geistiges Eigentum entwenden, Geschäftsmodelle einer Seite untergraben oder Informationen erheben, die Geschäftsgeheimnisse verletzen (Krotov et al. 2020). Unter bestimmten Bedingungen können dennoch kommerzielle Websites gecrawlt werden, sofern die erhobene Text- und Datenmenge stark begrenzt ist und die Daten selbst derart verändert sind, dass sie weder kommerziell noch im ursprünglichen Sinne genutzt werden können. Vor diesem Hintergrund können wir nur dazu raten, die Ethikkommission (z. B. Ihrer Hochschule) zu Rate zu ziehen, wenn Webcrawler für ein eigenes Projekt Anwendung finden sollen. Die Tabelle 8.1 bietet Ihnen eine Checkliste an, um zu prüfen, ob und wenn, dann wie Sie eine Website crawlen können.

Sofern es keine ethischen Bedenken gibt und eventuelle datenschutzrechtliche Probleme ausgeschlossen werden können, können Daten mittels Webcrawling erhoben werden. Doch wie genau funktioniert das technisch? Welche Schritte müssen wir durchführen und was müssen wir beachten?

Zuerst müssen wir festlegen, welche Seiten gecrawlt werden sollen, was der Startpunkt ist und wie tief Sie in der Struktur der Seite nach Informationen suchen. Sie müssen sich aber auch darüber im Klaren sein, welche Informationen heruntergeladen werden sollen (und dürfen!). So kann es sein, dass Sie sich für die Lebensläufe, Parteizugehörigkeiten, Reden und das Abstimmungsverhalten aller im Bundestag vertretenen Politiker*innen bei Gesetzesentwürfen interessieren. In diesem Falle würden Sie auf der Website des Bundestages starten, dann schrittweise die Seiten der Parteien und dann der Politiker*innen ansteuern. Entsprechend existiert eine „Tiefe“ von drei Ebenen, d. h. die Informationen sind auf der Startseite des Bundestages (1. Ebene), der Parteiseite (2. Ebene) und der

Abbildung 8.1 Datenerhebung 1: Schematische Darstellung des Crawlingprozesses am Beispiel von Politikern auf der Bundestagswebsite



Politiker*innen (3. Ebene) erhalten. Sie können sich dies als hierarchischen Baum vorstellen, wie die Abbildung 8.1 verdeutlicht.

Wenn Sie die relevanten Seiten identifiziert haben, dann erkunden Sie, welche Elemente der Seite (z. B. Textkästen, Verlinkungen, PDF-Dateien) heruntergeladen oder zur Identifizierung weiterer potenzieller Informationsträger genützt werden können. Im Firefox-Browser können Sie die Elemente der Website aufrufen, indem Sie entweder F12 auf der Tastatur drücken oder auf ein Element (z. B. Textkasten) mit der rechten Maustaste klicken und dann im Aufklappmenü auf „Inspect" oder in der deutschsprachigen Version auf „Untersuchen" drücken. Dann öffnet sich eine Übersicht über den HTML- bzw. JavaScript oder PHP-Code der Seite am unteren Rand des Bildschirms. Hier können Sie nach den Elementen suchen, die heruntergeladen werden sollen. Nun müssen Sie eine sich wiederholende Abfrage (Schleife) programmieren, die nacheinander die gewünschten Informationen abfragt, herunterlädt und als HTML, Textdatei oder Excel-Sheet abspeichert (siehe auch Box 8.4).

Box 8.4: Weitere Materialien und Informationen online
Um eine Anleitung zum (sicheren) Webcrawling zu erhalten, können Sie auf der Seite des Blogs *https://sozmethode.hypotheses.org/category/webcrawling* vorbeischauen.

Auf jeden Fall sollten Crawler nicht in zu rascher Abfolge Anfragen an die Seite stellen (z. B. eine Abfrage alle 5–6 Sekunden). Sie sollten, falls Seiten mehrfach verlinkt werden, diese nur einmal ansteuern und Sie sollten das Datenerhebungsverhalten Ihres Webcrawlers wiederholt überprüfen, um nicht in sogenannten Spider-Traps (= Mechanismus der Seite, um fehlerhafte Informationen, sogenannter „junk", zurückzugeben) zu geraten und unnötigen Datenaufkommen (sogenannter „traffic") zu erzeugen. Ein praktischer Hinweis ist, die *robots.txt*

Tabelle 8.1 Checkliste für das Datencrawling

	Checkliste
1	Erlauben es die AGBs und die Nutzungsbedingungen, Inhalte der Seite abzurufen und abzuspeichern?
2	Wem gehören die Daten/sind sie proprietär oder sind sie öffentlich?
3	Dürfen die Daten an Dritte weitergegeben werden? Wenn ja, in welcher Form (z. B. aggregiert)?
4	Ist geklärt, welche Informationen/Inhalte konkret Sie herunterladen dürfen?
5	Gibt es eine Obergrenze der Datenmenge, die heruntergeladen werden darf? Wenn ja, wie hoch ist diese Grenze?
6	Kann sichergestellt werden, dass sensible, personenbezogene Daten unkenntlich gemacht werden?
7	Wurde sichergestellt, dass die entsprechende Website nicht mit Anfragen überfrachtet wird? Wurde ein Verzögerungstimer für Anfragen eingebaut?

der Webseiten zu lesen, die gecrawlt werden sollen. Diese enthält Informationen, ob und wenn, dann in welchem Umfang eine Website gecrawlt werden darf. Die Checkliste in Tabelle 8.1 gibt einen Fragenkatalog zur Hand, der zunächst beantwortet werden sollte, ehe der Crawling-Prozess gestartet wird.

8.3 Aufbereitung der Daten

8.3.1 Vereinheitlichung der Datenstruktur

Nachdem die Daten heruntergeladen und gespeichert wurden, müssen die einzelnen Texte zunächst bereinigt und in eine Form übersetzt werden, die von den Topic Modeling-Ansätzen (siehe Kapitel 11) bzw. der Sentiment-Analyse (siehe Kapitel 10) eingelesen und bearbeitet werden können. Tabelle 8.2 bietet einen Überblick über die Aufbereitungsschritte, welche zur Vorbereitung der induktiv-quantitativen Inhaltsanalyse durchgeführt werden müssen.

Der erste Schritt der Datenaufbereitung ist es, die Datenstruktur zu vereinheitlichen. Das heißt, dass Sie beispielsweise einen Ordner erstellen, in dem alle heruntergeladenen Texte als Textdaten gespeichert werden. Die Textdaten selbst haben dann ein identifizierbares Muster, mit denen Sie die einzelnen Abschnitte erkennen können (z. B. „Titel“, oder „Interviewer*in“, „Interviewpartner*in“). Sie

Tabelle 8.2 Aufbereitung 1: Idealtypische Aufbereitungsschritte

Bereinigungsschritt	Beschreibung
1	Vereinheitlichung der Datenstruktur
2	Entfernung von Duplikaten
3	Ergänzung/Entfernung unvollständiger Daten
4	Irrelevante Daten entfernen
5	Fehlerhaft zusammengefasste Daten auftrennen
6	Berichtigen weiterer Fehler (z. B. Rechtschreibung)
7	Speichern des Datensatzes
8	Reduktion des Datensatzes auf die wesentlichen Analyseaspekte
9	Entfernen von Stoppwörtern und Satzzeichen
10	Generierung von Bi- bzw. N-Grammen
11	Übersetzen der Texte in ein „Bag of Words“-Format
12	Speichern des veränderten Datensatzes

können aber auch eine Excel-Datei erstellen, in der beispielsweise eine Spalte die Titel der Texte, eine weitere das Publikationsdatum der jeweiligen Texte, eine weitere den Text enthält. Diese Zeilen können beliebig viele Zusatzinformationen enthalten. Dabei sollte auf jeden Fall festgehalten werden, welche Informationen Sie für die Auswertung (und auch: für welchen Schritt der Auswertung) verwenden.

8.3.2 Typen von und Umgang mit fehlerhaften Daten

Sie müssen stets davon ausgehen, dass sich Fehler in Datenerhebungsprozesse einschleichen können, die sich im Datenaufbereitungsprozess bemerkbar machen. Tabelle 8.3 stellt verschiedene Probleme in Tabellenform dar, die hierbei auftreten können.

Es kann sein, dass Daten mehrfach heruntergeladen werden. In dem Falle handelt es sich um ein Duplikat, welches zu Verfälschungen im Datenmaterial führen kann. Sie sollten Duplikate stets entfernen, da durch Doppelung der Daten, Themen oder generelle Stimmungslagen nur ungenau erkannt werden. Bei einer großen Zahl von Duplikaten kann es sogar passieren, dass Sie Themen erzeugen, die inhaltlich nichts anderes als das Duplikat an sich sind. Im Falle der Sentiment-Analyse würde ein Duplikat ein verzerrtes Bild der positiven oder negativen Gesamtstimmung wiedergeben.

Es kann aber ebenfalls sein, dass Daten unvollständig heruntergeladen werden. In diesem Fall können Sie versuchen, die fehlenden Daten durch andere Quellen (z. B. Websites) entweder automatisiert oder per Hand zu ergänzen. Alternativ

Tabelle 8.3 Beispiele für Datenprobleme, die beim Herunterladen und Zusammenführen der Daten vorkommen können

Nr.	Titel	Text	Veröffentlichungsdatum	Quelle	Problem
1	Titel 1	„Beispieltext“	22.06.2021	Quelle 1	Duplikat
2	Titel 1	„Beispieltext“	22.06.2021	Quelle 1	Duplikat
3	Titel 1	„Beispieltext“	22.06.2021	Quelle 1	Duplikat
4	Titel 2	„Ohne Wörter“	23.06.2021		Unvollständige Daten
5		„Mit Wörtern“		Quelle 2	Unvollständige Daten
6		„Ohne Wörter“	23.06.2021	Quelle 3	Unvollständige Daten/ Potenzielles Duplikat
7	Titel 6	„Etwas mit Katzen“	23.06.2021	Quelle 1	Potenziell falsch heruntergeladen

verzichten Sie auf diese Daten und streichen diese aus dem Datensatz. Zudem kann es sein, dass Informationen zwar vollständig heruntergeladen wurden, aber inhaltlich nicht mit dem Thema verknüpft sind, das untersucht werden soll. Dies ist beispielsweise der Fall, wenn Sie Selbstbeschreibungen von Personen (z. B. in bestimmten Berufsfeldern) untersuchen möchte, durch die Datenerhebung aber nur einen tabellarischen Lebenslauf erhalten.

Hiernach sollten Sie auf Schreibfehler achten und die Texte gegebenenfalls korrigieren. Das ist dann relevant, wenn Sie eine Sentiment-Analyse durchführen wollen und deren Algorithmus darauf angewiesen ist, korrekt geschriebene Wörter vorliegen zu haben. Dieser Schritt ist ebenfalls relevant, wenn Sie einen Topic Modeling Ansatz wählen und keine Themen erzeugen wollen, die aus Transkriptions-, Schreib- oder Tippfehlern bestehen. Es empfiehlt sich, spätestens nach diesem Schritt den von Ihnen bearbeiteten Datensatz unter einem anderen Namen (am besten stets mit Datum und Versionsangabe) zwischenzuspeichern und sich im Anschluss zu überlegen, welche Daten, Textausschnitte und zusätzlichen Informationen Sie für Ihre weitere Analyse verwenden wollen.

8.3.3 Textaufbereitung

Im nächsten Schritt werden Stoppwörter (z. B. „ist", „war", „aber" usw.) aus den Texten entfernt. Stoppwörter sind Wörter, die am häufigsten in einer Sprache vorkommen und damit nicht zum Verständnis von Texten beitragen und darüber hinaus keine grammatikalische oder syntaktische Funktion innehaben. Diese Binde-, Füll- usw. -Wörter machen den Großteil von Text aus (ca. 85 %; siehe Kapitel 6.2.4). Für die Analyse relevant sind jedoch vor allem Nomen, Verben, Adjektive und Adverbien. Wörter, die zu häufig vorkommen, eignen sich zudem nicht, Unterschiede zwischen Texten herauszuarbeiten und erschweren üblicherweise die Analysen, die Sie durchführen möchten. Danach werden die im Text vorkommenden Wörter entweder gestemmt oder lemmatisiert. Stemming bezeichnet dabei den Vorgang, Wörter auf ihren Wortstamm zu reduzieren (z. B. fliegen, flog, Flug → flieg), während Lemmatisierung die Reduktion von Wörtern auf ihre Grundform (z. B. ist, war, gewesen → sein) darstellt (Jivani 2011).

Im Anschluss können Sie Bi- oder N-Gramme, also Wortketten, generieren, die aus zwei oder mehr Wörtern bestehen. Dies könnten Namen, Fachausdrücke (z. B. qualitative Analyse) oder andere Begriffe wie Social Media sein, die häufig in den von Ihnen analysierten Kommunikationen vorkommen. Diese fügen Sie zu einer Zeichenkette (sogenannte *strings*) zusammen, die vom Computer dann als einzelnes Wort erkannt werden. So würde beispielsweise Social Media zu Social_Media umgeformt werden. Sie sollten aber eine klare Regel aufstellen und dokumentieren, unter welchen Voraussetzungen diese Wortketten ausgewertet werden. Eine Möglichkeit wäre, deren Auftreten in einem zuvor definierten Pro-

zentsatz der Texte, beispielsweise mehr als 5 % oder weniger als 50 % der Texte zu testen. Ein anderer Ansatz ist, dass diese Bi- und N-Gramme mit einer Mindesthäufigkeit in Ihren Texten vorkommen. Wie Sie diese Mindestzahl festlegen, hängt von der Anzahl der Texte und dem alleinigen Auftreten der Wörter ab, aus dem die Bi- oder N-Gramme erzeugt werden. Bei kleinen Textmengen (z. B. bei 100 Texten) könnte der Wert bei 5 oder 10 liegen, bei größeren bei 25 usw.

Zuletzt werden die Wörter aus der im Text enthaltenen Reihenfolge gerissen und ausgezählt. Dies erzeugt ein sogenanntes „bag of words"-Format (auf Deutsch ungefähr Wortklumpen), dem die Annahme zugrunde liegt, dass unterschiedliche Themen mit unterscheidbaren Wörtern adressiert werden – und diese systematisch über die Texte hinweg verteilt sind, wenn Personen über verschiedene Themen reden (z. B. die alte Dame, die sozial konform antwortet, über das Rückenleiden spricht oder über ihre Zeit als berufstätige Ärztin).

Wenn Sie diese Schritte durchgeführt haben, dann empfiehlt es sich, den Datensatz zuletzt nochmals unter einem anderen Namen abzuspeichern und im Dateinamen zu vermerken, dass es sich um die bereinigten Daten handelt. Dieser Schritt kann Ihnen bei größeren Projekten helfen, die Übersicht über die unterschiedlichen Versionen der Daten bis zum fertigen Datensatz für die Inhaltsanalyse zu bewahren.

8.4 Wo finde ich online Hilfe?

Da die Topic Modeling-Ansätze und die dazugehörige Datenerhebung, Datenaufbereitung und Kontrolle in R und in Python durchgeführt wird, müssen Sie im Zweifelsfalle auch wissen, wo Sie online Hilfe finden und Fragen stellen können. Das ist umso wichtiger, da der vorliegende Text nur eine Einführung in einen kleinen Teilbereich der Datenanalyse und Programmierung in R und Python darstellt, den Programmier- und Auswertungsmöglichkeiten aber kaum Grenzen gesetzt sind. Direkte Hilfe können Sie in beiden Programmierumgebungen bekommen – wie in den folgenden Teilkapiteln zu den Analysemethoden in R und Python angesprochen wird. Daneben gibt es auch Onlineplattformen, bei denen man Hilfe bekommen bzw. nachschlagen kann. Dabei sind zwei Quellen besonders relevant: GitHub und StackOverflow.

GitHub ist eine kollaborative Online-Plattform, in der verschiedene Nutzer*innen Programmcodes hochladen, dokumentieren und verändern können. Unter einem Paket ist eine Sammlung von Befehlen und Funktionen zu verstehen, mit deren Hilfe Sie beispielsweise statistische Berechnungen durchführen, Ihre Daten aufbereiten oder Grafiken erstellen können. Kernbestandteil von GitHub ist eine geteilte Versionskontrolle des Programmcodes. Das heißt, dass jegliche Veränderung in einem Code dokumentiert wird und verschiedene Versionen (an denen zudem verschiedene Programmierer*innen arbeiten) parallel

existieren können (Cosentino et al. 2017). Somit können Sie nicht bloß nachvollziehen, wie die Pakete aufgebaut sind und welcher Programmiercode die Grundlage für diese Pakete darstellt, sondern es werden Beispiele und Hilfestellungen geboten, die Ihre eigenständige Arbeit mit R und Python erleichtern. Das betrifft auch die Pakete, die in den folgenden Kapiteln 9, 10 und 11 Verwendung finden.

Bei StackOverflow handelt es sich um eine Art Forum, in dessen Teilbereichen Fragen zu Programmiersprachen (Python, R, c#, c++, JavaScript), speziellen Paketen (z. B. *pandas* für die Datenbearbeitung in Python) und statistischen Verfahren angeboten werden. Dabei werden Probleme, die auftreten können und in der Ausgabe der jeweiligen Programmierumgebungen aufgelistet werden, mit Beispielcode und der besagten Ausgabe[7] hochgeladen. Die Antworten auf die gestellten Fragen und Probleme beinhalten ebenfalls Codebeispiele, mit deren Hilfe das ursprüngliche Problem (z. B. der Datensatz kann nicht richtig eingeladen werden, die Bereinigung der Daten funktioniert nicht) behoben werden kann.

7 Bei einer Ausgabe handelt es sich um Berechnungsergebnisse, Rückmeldungen des Programmes oder angezeigte Fehlermeldungen, die nach der Ausführung des Codes in R oder Python auftreten. Im Verlauf der weiteren Kapitel wird genauer hierauf eingegangen.

9. Quantitative Inhaltsanalyse mittels Korrespondenzanalyse

Das Kapitel soll Ihnen einen Überblick über die Korrespondenzanalyse und die Schritte geben, die nötig sind, damit Sie dieses Verfahren in RStudio anwenden können. Dazu wird zunächst ein Kurzabriss über die Grundidee der Korrespondenzanalyse geboten. Danach wird erklärt, wie Sie R und RStudio installieren können und wie die Benutzeroberfläche aufgebaut ist. Es folgt eine Erläuterung, was Pakete sind, wie man diese installiert und in die Arbeitsoberfläche lädt. Im nächsten Schritt werden Befehle zur Aufbereitung der Daten vorgestellt, ehe Befehle beschrieben werden, mit deren Hilfe Sie die Korrespondenzanalyse durchführen. Das Kapitel schließt mit der Interpretation von Analysen und der Erläuterung, wie Sie diese Ergebnisse exportieren. In diesem Kapitel erwarten Sie circa sechs Seiten Code, Auswertung empirischer Ergebnisse und zehn Abbildungen.

9.1 Einleitung

In Kapitel 8 haben wir Ihnen einen Überblick über die gängigen Datenquellen und Typen der automatisierten Textanalyse gegeben und mit Beispielen unterfüttert. Wir haben dabei zwischen überwachten und nicht-überwachten Formen des maschinellen Lernens unterschieden. Wir haben Ihnen mehrere Möglichkeiten vorgestellt, wie Sie auf bestehende Daten zurückgreifen und auf deren Basis Datensätze erstellen können. Bei diesen Möglichkeiten handelt es sich um Repositorien, APIs und Webcrawling. Wir haben dabei einen Fokus auf die rechtlichen und forschungsethischen Probleme des Webcrawlings gelegt. Danach haben wir Ihnen den Ablauf der Datenakquise idealtypisch skizziert und zuletzt aufgeführt, bei welchen Webpräsenzen Sie Hilfe finden können. Nun widmen wir uns gemeinsam mit Ihnen der Korrespondenzanalyse, die als ein soziologisches Verfahren Einzug in die Welt der automatisierten Textanalyse gefunden hat und zu den Verfahren des unüberwachten maschinellen Lernens gezählt werden kann (Giraudel und Lek 2001; Niyogi et al. 2011).

Ziel der Korrespondenzanalyse ist es, Strukturen aus großen Datentabellen mit vielen Variablen herauszuarbeiten und Ihnen dadurch eine Interpretation dieser Strukturen zu ermöglichen. Im vorliegenden Fall hilft die Korrespondenzanalyse dabei, Themen – und genauer: Themengegensätze – aus den Texten heraus zu präparieren. Das geschieht auf Basis der Wörter, die die Sprecher*innen verwenden. Kommen beispielsweise systematisch Begriffe gemeinsam vor, während andere in den gleichen Texten immer systematisch ausgelassen werden, dann erkennt

diese Methode eine „Dimension" in den Daten, die von Ihnen gedeutet werden muss (Hjellbrekke 2019, S. 36 f.). Vielleicht sind Sie bereits mit der Korrespondenzanalyse durch die Lektüre von Bourdieu in Berührung gekommen, der diese Methode bzw. deren Erweiterung (multiple Korrespondenzanalyse) dazu nutzte, um verschiedene gesellschaftliche Bereiche, sogenannte Felder, zu analysieren. Sie diente dazu, Gegensätzlichkeiten zwischen gesellschaftlichen Gruppen, beispielsweise bei der Aneignung kultureller Güter und Bildungsaspirationen aufzuzeigen (Bourdieu 2016), oder dazu, die Organisationsstruktur der Wissenschaft und daran geknüpfte Machtstrukturen zu rekonstruieren (Bourdieu 2010).

Betrachtet werden bei der Korrespondenzanalyse Variablen, im vorliegenden Falle Wörter als Textdaten, die in mindestens 5 % und maximal 95 % der Texte auftreten. Der zugrundeliegende Gedanke ist, dass wir nur Variablen oder Wörter analysieren sollten, die zur Unterscheidung von Themen und damit – statistisch gesprochen – zur Konstruktion von Dimensionen tauglich sind. Zudem prägen Variablen, die zu selten oder häufig vorkommen (im Soziolekt: Ausreißerwerte), diese Dimensionen sehr stark und können daher sehr verzerrend wirken und die Interpretation der Analysen erschweren. Wenn beispielsweise ein Wort selten vorkommt, dann kann eine Dimension den Gegensatz zwischen Texten ausdrücken, in denen ein Wort vorkommt und Texten, in denen das Wort nicht vorkommt. Eine weitere inhaltliche Interpretation wäre hingegen nicht möglich. Das gleiche gilt unter umgekehrten Vorzeichen für Variablen und Wörter, die in mehr als 95 % der Texte vorkommen. Hier würde eine durch die Korrespondenzanalyse entdeckte Dimension durch die wenigen Texte verzerrt werden, in denen diese Wörter nicht vorkommen.

Grundsätzlich können Sie diesem Problem auf zwei Arten begegnen. Erstens durch den Ausschluss aller Wörter, die in weniger als 5 % bzw. mehr als 95 % der Texte enthalten sind. Zweitens können Sie diese Wörter als sogenannte passive Variablen in die Korrespondenzanalyse aufnehmen (Hjellbrekke 2019, S. 35 f.). Passiv heißt, dass sie nicht für die Ermittlung der Themendimensionen herangezogen werden, aber ermittelt wird, wo (in Relation zu anderen Begriffen) sich diese passiven Variablen auf den Dimensionen befinden, die durch die Korrespondenzanalyse ermittelt werden. Im Folgenden beschränken wir uns auf die erste Variante der Korrespondenzanalyse, um die Aufbereitung der Variablen nicht in die Länge zu ziehen und den Auswertungsteil nicht zu überfrachten.

9.1.1 Kommunikation als Textdaten in einer Matrix

Bevor wir beginnen, vergegenwärtigen wir uns an dieser Stelle noch einmal die Bedeutung von Kommunikationen, die in Textform transkribiert werden können und erinnern uns damit an das Beispiel der alten Dame. Ihre Antwort auf die Frage „Wie geht es Ihnen?" könnte entweder „Danke, mir geht es gut", oder

„Danke, aber mir tut der Rücken weh“ lauten. Wenn genügend Personen die Frage entweder auf die eine oder die andere Weise beantwortet haben und sich beide Antworten somit systematisch ausschließen, dann wird die Korrespondenzanalyse eine Dimension erkennen, an deren einem Ende entweder „sozial konforme Antworten“ bzw. der „Bekundung von Gesundheit“ (erste Antwort), an deren anderem Ende die „Nennung von Krankheiten“ oder „Rückenproblemen“ (zweite Antwort) steht. Wie dieser Umstand visualisiert, d. h. in eine Punktwolke (= Scatterplot) umgesetzt wird, wie diese zu interpretieren ist und welche Maßzahlen damit einhergehen, wird im weiteren Verlauf des Kapitels verdeutlicht. Damit diese Einordnung gelingt, benötigen Sie eine bestimmte Anordnung der Daten in einen Datensatz. Diese Anordnung wird in Tabelle 9.1 dargestellt.

Bei diesem Datensatz handelt es sich um eine Text*-Begriffs-Matrix. In der ersten Spalte Text würde sich beispielsweise eine fortlaufende Nummer, der Titel des Textes, oder dessen Inhalt befinden. Auf jeden Fall sollten Sie eine Nummerierung, eine Buchstabenfolge oder ähnliches wählen, um die Texte identifizieren zu können, mit denen Sie arbeiten. Danach folgen hier Spalten, in denen die Begriffe pro Text ausgezählt vorkommen.[1]

Tabelle 9.1 Datenstruktur von Texten zur Durchführung einer Korrespondenzanalyse

Text	Begriff 1	Begriff 2	…	Begriff M
1	3	0	…	1
2	2	2	…	1
3	0	5	…	3
…	…	…	…	…
N	1	0	…	0

Die Spalten sind nicht auf Begriffe begrenzt, sondern können auch andere Informationen beinhalten, die Sie in die Analyse aufnehmen können. Das kann beispielsweise das Alter, die Herkunft oder das Geschlecht des*der Sprecher*in sein, welche*r den jeweiligen Text hervorgebracht hat. Es können auch Informationen über die Adressat*innen enthalten sein, an die ein bestimmter Text gerichtet ist.

1 Sie können auch die Anzahl der Begriffe durch die Gesamtlänge des jeweiligen Textes teilen und auf diese Weise ermitteln, welchen Stellenwert oder welches Gewicht die jeweiligen Begriffe innerhalb eines Textes einnehmen. Dann könnten Sie zum Beispiel eine multiple Faktorenanalyse berechnen und die Dimensionen ebenfalls als Themen interpretieren.

Diese Informationen können, wenn sie im Datensatz vorkommen, ebenfalls in die Analyse aufgenommen werden. Bei Geschlecht und Herkunft handelt es sich dabei um nominalskalierte Merkmale, beim Alter um ein metrisches Merkmal, das neben den Wörtern (ebenfalls nominalskaliert) gleichzeitig in die Analyse einfließen kann (siehe Box 9.1 für Kurzerklärung der Skalenniveaus). Dies ist ein besonderer Vorteil der Korrespondenzanalyse, da andere quantitativ-statistische Verfahren auf ein bestimmtes Skalenniveau beschränkt sind. Im Gegensatz dazu können Sie unterschiedliche Datentypen, vor allem kategoriale Daten (= Daten, die in Kategorien wie gleich/ungleich, mehr/weniger als eingeteilt sind, und somit auch nominal- und ordinalskalierte Merkmale) verwenden (Blasius und Schmitz 2013). Sie sollten dabei lediglich beachten, dass die Anzahl der Ausprägungen ratioskalierter Merkmale nicht zu groß sein sollte, da sonst eine oder mehrere Dimensionen durch eine Variable mit vielen Ausprägungen dominiert werden kann.

Box 9.1: Skalenniveaus

Nominalskalierte Merkmale: Hierbei handelt es sich um Merkmale, die nur die Unterscheidung gleich/ungleich kennen. Beispiele sind Wörter, Geschlecht von Personen, Parteizugehörigkeit.

Ordinalskalierte Merkmale: Diese Merkmale haben eine Rangfolge wie zum Beispiel Schulnoten. Die Abstände zwischen den Ausprägungen (z. B. gut/sehr gut) sind nicht interpretierbar.

Kardinalskalierung 1 – Intervallskalierte Merkmale: Diese Merkmale haben einen interpretierbaren, gleichmäßigen Abstand zwischen den Ausprägungen (z. B. Grad Celsius), aber keinen natürlichen Nullpunkt.

Kardinalskalierung 2 – Ratioskalierte Merkmale: Hierbei handelt es sich um Merkmale, bei denen nicht nur die Abstände zwischen den Ausprägungen interpretierbar sind, sondern zugleich einen natürlichen Nullpunkt aufweisen. Hierzu zählt zum Beispiel das Alter oder Einkommen.

Sie können, müssen die Begriffe aber nicht manuell auszählen, die in den Texten vorkommen. Hier können Sie entweder in R oder Python eigene Auszählungen programmieren bzw. durchführen. Sie können aber auch die Wörter oder Wortstämme, die Sie jeweils interessieren, in AntConc (siehe Kapitel 6.3.2) oder MAXQDA (siehe Kapitel 6.4.2) auszählen lassen und als Excel-, csv-Datei oder vergleichbare Tabellenformate abspeichern und dann in R einlesen. Bei größeren Textmengen kann die Aufbereitung mehrere Stunden oder sogar Tage in Anspruch nehmen. Es ist daher ratsam, sich im Vorfeld genau zu überlegen, welche Texte und wie viele Texte sie analysieren wollen.

9.1.2 Schritt für Schritt-Ablauf einer Korrespondenzanalyse

Um eine Korrespondenzanalyse durchzuführen, müssen Sie die folgenden Schritte beachten, die im Ablaufplan in Abbildung 9.1 zusammengefasst werden.

1. Erstens müssen Sie die Pakete installieren, in denen die Befehle abgespeichert sind, um die Korrespondenzanalyse durchzuführen.
2. Zweitens müssen die Pakete, in denen sich die Befehle befinden, in die R-Programmierumgebung geladen werden.

3. Drittens müssen die Daten geladen werden, ehe viertens genau der Teil ausgewählt wird, der die Grundlage der Korrespondenzanalyse sein soll.
4. Zuletzt muss der Befehl ausgeführt werden, der die Analyse durchführt sowie Grafiken erstellt und alle Ergebnisse gedeutet werden.

Abbildung 9.1 Schritt für Schritt-Durchführung der Korrespondenzanalyse

Schritt 1	**Vorbereitung und Installation** 1. R und RStudio installieren 2. Pakete installieren 3. Pakete und Dateien in RStudio laden
Schritt 2	**Datenaufbereitung** 1. Entfernen von ungeeigneten Worttypen 2. Entfernen von Wörtern, die zu selten oder zu häufig vorkommen 3. Entfernen fehlender Werte
Schritt 3	**Datenanalyse** 1. Test, ob die Durchführung einer Korrespondenzanalyse angebracht ist 2. Auswahl der Anzahl zu interpretierender Dimensionen/Themen 3. Erkundung von Texten und Wörtern im aufgespannten Themenraum 4. Sichtung von Texten oder Textabschnitten, die charakteristisch für die einzelnen Themen sind
Schritt 4	**Daten verfügbar machen** 1. Abbildungen speichern 2. Informationen aus der Korrespondenzanalyse in einen eigenen Datensatz überführen 3. Datensatz abspeichern
Schritt 5	Ergebnisse zusammenfassen und Erstellung einer Präsentation, Studienarbeit und/oder Publikation
Schritt 6	Sichere Archivierung der Daten und, wenn möglich, Aufbereitung zur Nachnutzung

9.2 Einführung in RStudio

9.2.1 Installation von R und von RStudio

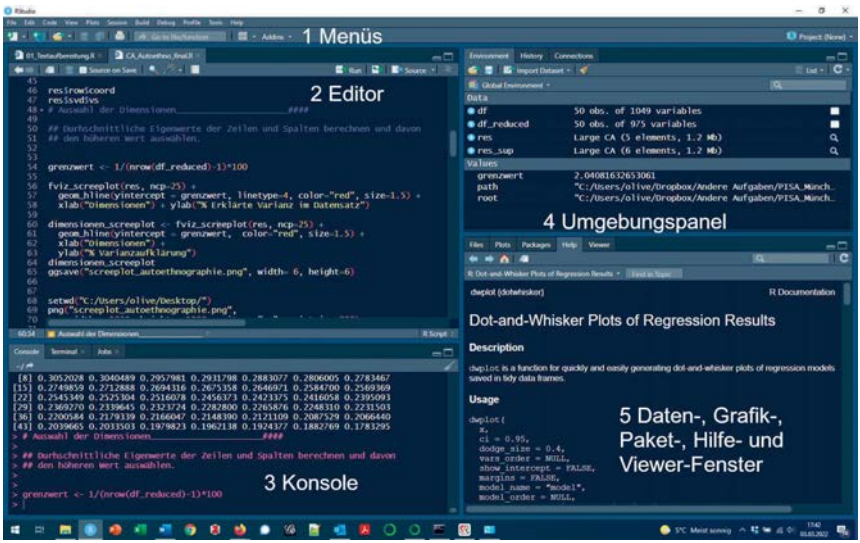
Um die Korrespondenzanalyse durchzuführen, müssen Sie zunächst R installieren. R ist eine Programmiersprache, die speziell für statistische Berechnungen entwickelt wurde und ist darauf ausgerichtet, Berechnungen mit möglichst wenig Codezeilen durchzuführen. Vor allem: R ist Freeware, d. h. Sie können R kostenfrei herunterladen und auf eine große Community zählen, die die Pakete weiterentwickelt und Sie zu potenziellen Problemen schnell online Hilfe finden werden (siehe zu Copylefts und Copyrights Kapitel 1.3).

Sie können R unter *https://cran.r-project.org* herunterladen. Wenn Sie die

Seite aufrufen, sehen Sie ganz oben ein Fenster mit dem Titel „Download and Install R“. Hier können Sie die für Ihr Betriebssystem passende Version auswählen (Windows, Linux und macOS). Wenn Sie auf einen der Links klicken, dann öffnet sich in Ihrem Browser ein neues Fenster, in dem Sie verschiedene Versionen von R herunterladen können. Hier wählen Sie die erste Option *„base“* aus bzw. klicken auf den Link „install R for the first time“. Von dort aus kommen Sie auf eine weitere Seite, auf der beispielsweise „Download R for Windows“ (mit der Versionsnummer) steht, sofern Sie zuvor die Windows-Version ausgewählt haben. Hier klicken Sie auf den Download-Link und speichern das Installationsprogramm von R auf Ihrem Rechner. Nach dem Download führen Sie die Installationsdatei aus und folgen den Anweisungen (z. B. geben Sie an, wo Sie R auf Ihrem Rechner installieren wollen).

Darüber hinaus empfiehlt es sich, nach der R-Installation auch RStudio zu installieren. RStudio bietet eine grafische Oberfläche und ein Drop-down-Menü, das den Datenimport, die Installation von Paketen mit Befehlen und sowohl Erstellung als auch das Abspeichern von Grafiken erleichtert. Die Oberfläche sieht beim erstmaligen Start in etwa wie auf der folgenden Abbildung 9.2 aus. Beachten Sie dabei, dass sich diese Ansicht im Zeitverlauf, beispielsweise durch Versionsupdates, etwas verändern kann. Im Großen und Ganzen ist die Oberfläche aber gleichgeblieben, sodass Sie sich zwischen den Versionen recht schnell zurechtfinden müssten.

Abbildung 9.2 Bedienoberfläche von RStudio mitsamt Aufbau von Menüs und einzelnen Fenstern



9.2.2 Aufbau von RStudio

Die Oberfläche von RStudio ist aus insgesamt fünf Bereichen aufgebaut: Menüs, der *Editor,* die Ausgabe bzw. *Console,* das Umgebungspanel (Environment, History, Connections) sowie das Daten-, Grafik-, Paket-, Hilfe- und Viewer-Panel. Wir verwenden in der Folge die englische Terminologie, da RStudio standardmäßig in englischer Sprache installiert wird und der Großteil der Community Englisch spricht und entsprechend die englischen Fachwörter verwendet.

9.2.2.1 Das Menü

Die Menüs beinhalten Reiter, in denen Sie Daten öffnen, speichern oder exportieren *(File),* Arbeitsschritte rückgängig machen oder wiederherstellen können (R-Befehl in Englisch: *Edit*) oder Ihren Code organisieren können, den sie im Editor (siehe Kapitel 9.2.2.2) schreiben. Darüber hinaus können Sie durch die von Ihnen erstellten Grafiken *(Plots)* navigieren und diese speichern, neue Arbeitssitzungen starten *(Build)* oder die von Ihnen geschriebene Programmstruktur prüfen *(Debug).* Ferner können Sie verschiedene Nutzerprofile anlegen und verwalten *(Profile),* generelle Optionen festlegen, Pakete installieren oder updaten (*Tools,* siehe Kapitel 9.2.2.5) sowie Hilfe und eine Anleitung aufrufen und auf Updates prüfen *(Help).*

Unter diesen Reitern finden Sie einige *Buttons* (auf Deutsch: Tasten). Diese werden nun von links nach rechts beschrieben. Der erste Button (das leere Blatt mit einem Plus in der oberen linken Ecke) erlaubt es Ihnen, ein neues Dokument anzulegen. Von der Auswahl, die sich hier öffnet, ist „R Script" am relevantesten, da Sie mit diesem Button ein neues *Script* anlegen können, um Ihre Berechnungen durchzuführen. Rechts daneben ist der Button „create a project". Hiermit können Sie ein neues Programmierprojekt anlegen, wozu zugleich ein neuer Ordner auf Ihrer Festplatte angelegt wird. Der dritte Button, der wie ein Aktenordner mit einem Pfeil aussieht, bietet Ihnen eine Auswahl aus den Skripten, die Sie erstellt und gespeichert haben. Die Diskette ermöglicht Ihnen das Speichern Ihres Skriptes (alternativ: auf „File" → „Save" im Menü oben klicken), während die Disketten alle geöffneten Skripte speichern. Der Drucker ermöglicht Ihnen, Ihr Skript zu drucken. Das danebengelegene Feld ist ein Suchfeld, bei dem Sie innerhalb des Editors (siehe Kapitel 9.2.2.2) nach Elementen Ihres Codes suchen können. Das ist besonders hilfreich, wenn Sie länger an einem Projekt arbeiten und viele Codezeilen geschrieben haben.[2] Das Fenster (zweiter Button von rechts) erlaubt Ihnen

2 Der Übersichtlichkeit halber empfiehlt es sich aber, die Codedateien bzw. Skripte möglichst kurz zu halten und detailliert zu kommentieren. Das erleichtert Ihnen, sich auch nach langer Zeit in Ihrem Projekt zurechtzufinden. Speichern und benennen Sie den Code dabei

das Einstellen der Fenster, mit denen Sie den Code bearbeiten. Zuletzt sind Add-Ins durch Dritte programmierte Zusatzfunktionen, mit deren Hilfe Sie weitere Bereiche von R einstellen und nach Ihren Bedürfnissen anpassen können (z. B. für die automatische Formatierung Ihres Codes).

9.2.2.2 Der Editor

Im Editor (2) schreiben Sie Ihren Programmcode. Anders als im Bereich qualitativer Forschung bezeichnet Code hier die manuell einzugebenden Befehlszeilen, die Ihr Computer ausführen soll. Zur Unterscheidung mit dem qualitativen Kodieren (siehe Kapitel 4.3.6) wird hier die englische Schreibweise Code verwendet. Die Befehle, auch *Funktionen* genannt, die in diese Zeilen geschrieben werden, dienen dazu, R genau zu sagen, was zu tun ist. Einzelne Befehle werden dabei von Klammern begleitet, in denen einerseits die angesteuerten Daten und andererseits Optionen eingegeben werden. Ein Befehl hat beispielsweise den Aufbau:

Code 9.1 Grundstruktur eines Befehls in R. Beispiel: CA()-Befehl zur Ausführung der Korrespondenzanalyse

```
CA(Datensatz, ncp = 10)
```

Code 9.1 ermöglicht es Ihnen, eine Korrespondenzanalyse auszuführen (nachdem Sie das Paket *FactoMineR* installiert haben, siehe Kapitel 9.3.1). Die Funktion wird mit `CA()` eingeleitet. An der ersten Stelle in der Klammer steuern Sie Ihren Datensatz an, in dem die Texte, Wörter und Zusatzinformationen gespeichert sind. Dann sehen Sie eine Option, die durch ein Komma abgetrennt ist. Diese Option gibt an, dass die Ergebnisse der ersten zehn gefundenen Dimensionen detailliert ausgegeben werden. Es gibt darüber hinaus noch weitere Optionen, auf die im Verlauf des Kapitels eingegangen wird.

Ergänzend zu Funktionen sind *Objekte* zentrale Elemente von RStudio. Ein Beispiel für ein Objekt ist ein Datensatz. Dabei handelt es sich um eine Datenmatrix, die aus Zeilen, Spalten und Zellen besteht. Objekte können auch aus Buchstabenfolgen (sogenannten *strings*), ganzen Zahlen *(integers),* Zahlen mit Nachkommastellen *(floating point numbers),* aber auch Listen, Vektoren oder logischen Abfragen (z. B. wahr/falsch, größer, kleiner usw.) bestehen. Sie müssen darauf achten, dass R Groß- und Kleinschreibung unterscheidet. Das heißt, wenn Sie eine Variable haben, die als `text` gespeichert wird, dann kann Sie nicht mit

---

auch in diesem Falle wieder so, dass Sie eindeutig aufzeigen, wann und mit welchen Verfahren sowie Inhalten das Skript befüllt wurde.

`Text` aufgerufen werden. Sollten Sie zugleich eine Variable namens `Text` haben, dann handelt es sich für R um zwei unterschiedliche Variablen bzw. Objekte. Ganz wichtig ist die in Code 9.2 aufgeführte Zuweisung, die mittels eines Pfeiles, meist <-, erfolgt:

Code 9.2 Zuweisung einer Variablen zu einem Objekt

```
Ergebnis <- CA(Datensatz, ncp = 10)
```

Wenn Sie diese Befehlszeile ausführen, bewirkt dies, dass Sie das Ergebnis einer Korrespondenzanalyse, die Sie mit CA durchgeführt haben, in das Objekt „Ergebnis" speichern. Dieses Objekt taucht dann im Umgebungspanel (siehe Kapitel 9.2.2.4) auf.

Zuletzt können sie mit der Raute, d. h. mit #, einen Kommentar in eine Zeile eintragen. Das kann nach dem Befehl bzw. der dadurch aufgerufenen Funktion erfolgen. Sie können aber auch zu Beginn der Zeile eine Raute eintragen und dann eine Notiz für Sie, andere Studierende oder Dozierende schreiben, um zu dokumentieren, was Sie gemacht haben. Dies ist besonders wichtig, wenn Sie über längere Zeit an einem Skript schreiben und andere Personen nachvollziehen müssen, was Sie gemacht haben.

Ausführen können Sie die Befehle, indem Sie entweder die Code-Zeilen, die Sie ausführen wollen, markieren. Das machen Sie, indem Sie die linke Maustaste gedrückt halten und über den gewünschten Code fahren. Dieser wird dann grau markiert. Dann können Sie oben rechts im Editor auf den Button „Run" oder die Tastenkombination „Strg + Enter" drücken. Die Befehlszeilen werden dann entsprechend ausgeführt.

9.2.2.3 Die Konsole

Im Gegensatz zum Editor hat die Konsole (3) zwei Funktionen. Einerseits kann auch hier Code eingegeben und durch Drücken der Enter-Taste ausgeführt werden. Dabei wird dieser Code aber nicht wie im Editor gespeichert, sondern einfach nur ausgeführt. Sie können hier mittels der Pfeiltasten (z. B. Pfeil rauf und Pfeil runter) zwischen den ausgeführten Eingaben hin- und herwechseln. Dennoch werden diese nicht über die Sitzung hinaus gespeichert – d. h., wenn Sie R beenden, dann ist auch das, was Sie in die Konsole übergeben haben, verloren. Daher empfehlen wir dringend alle Befehle im Editor zu verfassen, damit Sie diese a) wiederholt anwenden können und b) transparent Ihren Auswertungsvorgang darlegen können.

Zweitens hat die Konsole die Aufgabe, Ihnen Rückmeldung über den aus-

geführten Code zu geben. Dabei wird angezeigt, welche Befehle ausgeführt werden und was das Ergebnis der Ausführung ist. Darüber hinaus zeigt die Konsole Fehlermeldungen an, falls Sie eine Funktion falsch aufgerufen haben. Diese Fehlermeldung können Sie beispielsweis aus der Konsole herauskopieren und in eine Google-Suchanfrage eingeben. In der Regel werden Sie dann auf StackOverflow oder andere Seiten weitergeleitet, in denen der Fehler genauer beschrieben wird.

9.2.2.4 Das Umgebungspanel

Im Umgebungspanel (4) finden Sie drei übergeordnete Reiter: „Environment", „History" und „Connections". Hiervon ist für das weitere Vorgehen das Environment-Panel am relevantesten. Zur kurzen Übersicht: Das History-Panel bietet eine Übersicht über die Codezeilen, die Sie im Verlauf der Sitzung ausgeführt haben. Das Connections-Panel erlaubt es Ihnen, eine Verbindung zu Datenbanken aufzubauen, in denen Datensätze gespeichert sind. Das Environment-Panel hat eine eigene Liste von Buttons, eine Suche sowie eine Übersicht über Variablen und Objekte, die hineingeladen worden sind.

9.2.2.5 Das Daten-, Grafik-, Paket-, Hilfe- und Viewer-Panel

Das Daten-, Grafik-, Paket-, Hilfe- und Viewer-Panel (5) bietet Ihnen erstens einen Dateibrowser (wie den Windows Explorer) im Reiter „Files" an, dort können Sie festlegen, wo Ihre Dateien gespeichert werden sollen bzw. auf welches Verzeichnis R zurückgreifen soll, wenn Dateien eingelesen werden. Hier können Sie mit den Buttons neue Ordner auf Ihrer Festplatte anlegen, mit „Delete" Ordner und Dateien löschen, mit „Rename" umbenennen, oder unter „More" kopieren und bestimmte Ordner als Arbeitspfad angeben. Wenn Sie verschiedene Projekte verfolgen und nicht den Überblick verlieren wollen, empfiehlt es sich, hier neue Arbeitspfade (also Ordner in Windows) anzulegen und die Skripte und andere Dateien (wie Datensätze oder erzeugte Grafiken) separat zu speichern.

Zweitens findet sich hier eine Ansicht über Ihre Grafiken, die im Reiter „Plots" einsehbar ist. Hier können Sie die Grafiken mit dem „Export"-Button[3] abspeichern, mit der „Zoom"-Funktion vergrößern, oder mit den Pfeil-Buttons durch die in der Sitzung erzeugten Grafiken schalten.

Drittens können Sie im „Packages"-Reiter Pakete mit Befehlen installieren,

3 Wenn Sie allerdings Grafiken mit hoher Auflösung erzeugen möchten, dann empfiehlt es sich, diese mittels Code zu erzeugen, da Sie hier mehr Optionen zum Beispiel zu Größe, Dateiformat (z. B. PNG-Dateien, PDF-Dateien) haben.

auf die neueste Version updaten, oder durch das Setzen eines Häkchens in der Packages Library in RStudio laden.

Viertens gibt es das „Hilfe"-Fenster. Wenn Sie Hilfe zu einem Befehl benötigen, oder auf Online-Handbücher zugreifen möchten, dann werden Sie hier fündig. Standardmäßig wird eine Auswahl an Links angezeigt (z. B. „Learning R Online"), die Sie bei Ihrem Lernprozess begleiten. Hilfe zu spezifischen Funktionen bzw. Befehlen wird hier angezeigt, wenn Sie im Editor `help()` bzw. `??` eingeben. Dabei steuert der `help`-Befehl ganz explizit die Hilfe zu einem Befehl an, der in den Klammern angegeben wird, beispielsweise `help(read_csv)`.[4] Dagegen sucht das doppelte Fragezeichen, d. h. `??`, vor einem Befehl im gesamten R-Handbuch nach entsprechenden Stellen, in denen die Suchanfrage vorkommt.

Zuletzt bietet der Viewer einen Miniatur-Browser an, mit dessen Hilfe Sie auf dem PC gespeicherte Dateien anzeigen lassen können. Dies ist hilfreich, wenn Sie beispielsweise Websites bei einem Crawling-Prozess gespeichert haben, oder aber, wenn Sie web-basierte interaktive Zusatzprogramme (sogenannte Shiny-Apps) in R importiert haben.[5]

9.3 Vorbereitende Schritte für die Korrespondenzanalyse in RStudio

9.3.1 Software-„Pakete" in RStudio importieren und aktivieren

Doch welche Pakete benötigen wir, um eine Korrespondenzanalyse durchzuführen? Im Verlauf dieses Kapitels werden wir *FactoMineR* und *factoextra* verwenden.[6] FactoMineR beinhaltet Befehle, mit deren Hilfe Sie unterschiedliche Verfahren der explorativen Datenanalyse anwenden können (Lê et al. 2008). Explorativ heißt hier, dass diese Verfahren geeignet sind, um Sinngehalte aus komplexen, nahezu unübersichtlichen und vieldimensionalen Datentypen (z. B. Texte mit tausenden Wörtern und verschiedensten angesprochenen Themen) herauszuarbeiten. FactoMineR erlaubt es Ihnen, Korrespondenzanalysen, multiple Korrespondenzanalysen, Hauptkomponentenanalysen bzw. Faktorenanalysen sowie hierarchische Clusteranalysen durchzuführen. Daneben gibt es die multiple Faktorenanalyse, hierarchische multiple Faktorenanalyse oder duale multiple Faktorenanalyse. Für Interessierte, die mehr Informationen zu den Analyseformen erhalten wollen, die nicht im vorliegenden Kapitel behandelt werden, sei auf das

4 `read_csv()` ist der Befehl, mit dessen Hilfe Sie später einfache Tabellen im .csv-Format in R hineinladen können. Hierauf wird im weiteren Verlauf des Kapitels ebenfalls eingegangen.

5 Sie benötigen hierfür das Paket *httpuv* und können dieses mittels `viewer()` aus dem *rstudioapi*-Paket anzeigen lassen.

6 Daneben gibt es noch die Pakete *ca, ade4, MASS* und *ExPosition,* die mit eigenen Befehlen für die Korrespondenzanalyse aufwarten.

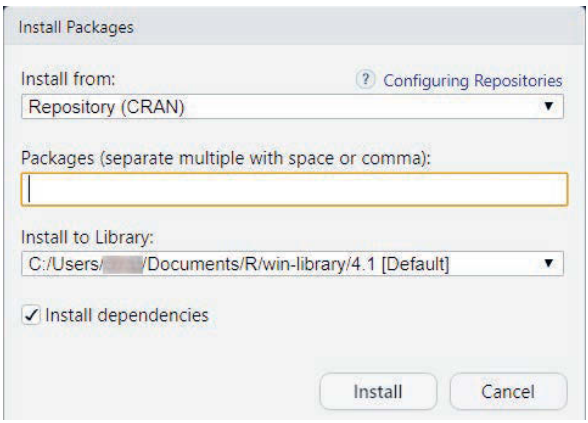
Einführungsbuch von Baur und Blasius (2018) verwiesen. Das Paket factoextra beinhaltet Befehle, mit deren Hilfe Sie die Ergebnisse ihrer Korrespondenzanalyse besser visualisieren können.

Optional können Sie die Pakete *xlsx, ggplot2* und *tidyverse* installieren und nutzen. Das Paket *xlsx* bietet Ihnen Befehle an, mit deren Hilfe Sie Excel-Dateien bequem einladen können. Das Paket *ggplot2* wiederum bietet Ihnen Befehle an, mit deren Hilfe Sie übersichtliche, formschöne Grafiken erzeugen und speichern können. Ferner können Sie mit diesem Paket Grafiken, die sie in factoextra erstellt haben, verändern. *tidyverse* beinhaltet Befehle, die Ihnen den Umgang mit Datensätzen erleichtern, beispielsweise, wenn Sie neue Variablen erstellen und alte verändern wollen.

9.3.2 Pakete installieren

Da Sie die Pakete FactoMineR und factoextra für die Durchführung der Korrespondenzanalyse und deren Visualisierung brauchen, sollten Sie wenigstens diese beiden installieren, ehe Sie mit der Analyse fortfahren. Pakete können Sie auf zwei Arten installieren: einerseits über die grafische Oberfläche. Beim Start von RStudio sehen Sie unten rechts einen Kasten, in dem „Files“, „Plots“, „Packages“, „Help“ und „Viewer“ steht. Hier klicken Sie auf den Reiter „Packages“, um eine Übersicht über die bereits installierten und geladenen Pakete zu bekommen. Geladene Pakete werden dabei mit einem Häkchen angezeigt, was bedeutet, dass Sie die Befehle, die in ihnen enthalten sind, auch nutzen können. Hier klicken Sie auf die Schaltfläche „Install“ und gelangen in ein Menü, in dem Sie nach den jeweils benötigten Paketen suchen können. Wie dies aussieht, sehen Sie in Abbildung 9.3.

Abbildung 9.3 Installationsfenster von Packages in RStudio



Dabei bedeutet „Install from“, dass Sie den Server aussuchen können, von dem aus RStudio versuchen wird, das Paket herunterzuladen und anschließend zu installieren. Das „Packages“-Feld ist das Feld, in dem die Suchanfragen für die Pakete und die Pakete ausgewählt werden, die Sie im Nachhinein installieren möchten. „Directory“ gibt an, in welchem Ordner auf Ihrem PC die Pakete gespeichert werden. Zuletzt bedeutet „Install dependencies“, dass Pakete direkt mitheruntergeladen und installiert werden, auf denen die von Ihnen benötigten Pakete aufbauen.

Sie können Pakete andererseits durch Eingabe des Befehls `install.packages("[NAME DES PAKETES]")` installieren. Dabei können Sie den Platzhalter `["NAME DES PAKETES"]` durch ein beliebiges Paket, beispielsweise FactoMineR ersetzen. Entsprechend müssen Sie, um FactoMineR zu installieren und später die Befehle anwenden zu können, `install.packages("FactoMineR")` angeben. Um mehrere Pakete zu installieren, können Sie in den `install.packages`-Befehl mit `c([Paket1],[Paket2],[Paket3])` mehrere Pakete nacheinander installieren. Der `c()`-Befehl reiht mehrere Objekte (z. B. Variablen, Begriffe, Zahlen etc.) aneinander. Dabei sollten Sie beachten, die Pakete, die Sie installieren wollen, in Anführungszeichen zu setzen. Die beiden Varianten des Installationsbefehls sind in Code 9.3 aufgelistet.

Code 9.3 Befehle zur Installation eines oder mehrerer Packages

```
#Installieren der Pakete####
install.packages("FactoMineR")
install.packages(c("xlsx","tidyverse"))
```

9.3.3 Pakete in RStudio laden

Nun müssen Sie die Pakete in Ihre Sitzung laden und sie dadurch aktivieren, um die Befehle ausführen zu können, die in diesen enthalten sind. Damit Sie die Pakete nicht bei jedem Ausführen Ihres Skriptes neu installieren, sollten Sie die Installationsbefehle zudem mit dem Rautenzeichen # kommentieren.

Wie bei der Installation gibt es zwei Möglichkeiten, die Pakete in die R-Programmierumgebung zu laden: das Laden über den Editor (oben links) oder durch Anwählen unten rechts im Packages-Panel. Um Pakete über den Editor zu laden, müssen Sie den `library()`-Befehl eingeben (siehe Code 9.4). Dieser funktioniert ganz einfach durch die Eingabe von `library([PAKET])`, wobei `[PAKET]` beispielsweise `FactoMineR` sein kann, welches Sie für die Durchführung der Korrespondenzanalyse benötigen. Darüber hinaus benötigen Sie `factoextra, tidyselect, tidyverse` und `ggplot2`, um beispielsweise übersichtliche Grafi-

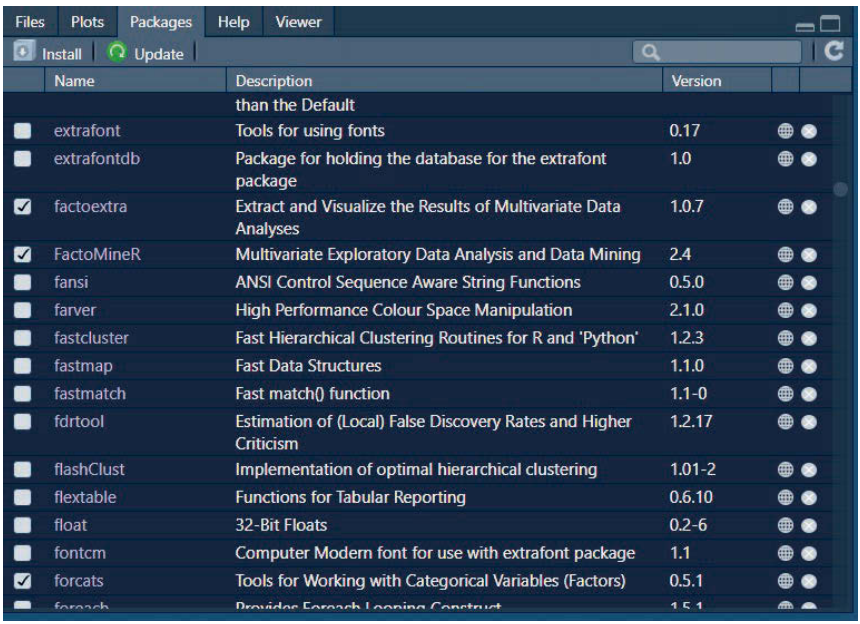
ken zu erstellen oder Daten einfach vorselektieren zu können. Ausgeführt werden können die Befehle auch wieder, indem Sie eine Zeile bzw. mehrere Zeilen anwählen und mittels „Strg + Enter“ ausführen. Alternativ können Sie die Zeile(n) anwählen und klicken auf die „Run“-Schaltfläche oberhalb Ihres Editors.

Code 9.4 Befehle zum Aktivieren von Packages in RStudio

```
#Aktivieren der Pakete####
library(FactoMineR)
library(factoextra)
library(tidyselect)
library(tidyverse)
library(ggplot2)
```

Die zweite Möglichkeit, Pakete zu laden, ist das Packages-Paneel unten rechts in der Ansicht. Dazu müssen Sie den Reiter „Packages“ anwählen und können danach mit der Suchleiste (Lupe mit Eingabefeld) nach dem gewünschten Paket suchen. Alternativ können Sie auf der rechten Seite durch die Liste scrollen und laden die relevanten Pakete, indem Sie einen Haken in der entsprechenden Box setzen.

Abbildung 9.4 Menü zum Einlesen und Updaten der Packages



9.3.4 Dateien einlesen

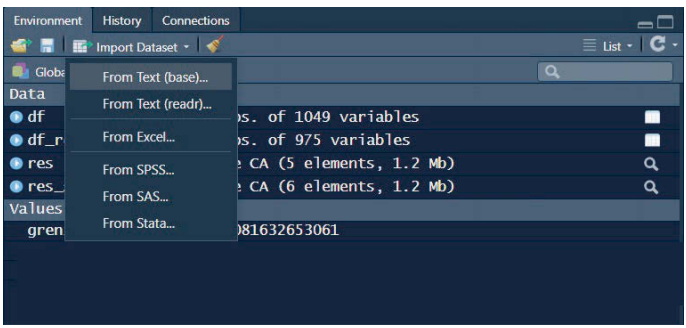
Nachdem Sie die Pakete für die Datenverarbeitung eingelesen haben, müssen Sie die Daten in Ihre RStudio Sitzung laden. Wie bei den beiden vorangegangen Schritten, können Sie auch hier die Daten entweder per Befehl oder mittels des Menüs im Environments- und History-Panel öffnen. Um einen Datensatz zu laden, der als csv-Datei gespeichert ist, können Sie den Befehl `read.csv()` verwenden und müssen dann den Dateipfad angeben. Ferner müssen Sie diese Datei einem Objekt zuweisen und ihr einen Namen geben. Folgende Codezeile 9.5 zeigt, wie Sie einen Datensatz laden können.

Code 9.5 Einlesen von Datentabellen mittels read.csv()

```
#Einladen der Daten_____________________________####
df <- read.csv("C:/Users/Buch_Textanalyse/Daten/aufbereitete_texte.csv")
```

Dabei ist `df` (Kurzform für Englisch: **d**ata**f**rame) der Name des Objektes, dem der Datensatz zugewiesen wurde. Sie können dieses Objekt im weiteren Verlauf Ihrer Analyse ansteuern, indem Sie `df` in den Editor eingeben und die Befehlszeilen ausführen oder im zweiten Falle Enter drücken. Wenn Sie dies tun, dann können Sie den Datensatz aufrufen oder Analysen mit den Variablen durchführen, die im Datensatz enthalten sind. Zur Erinnerung: Der Pfeil `<-` weist dem Objekt `df` den Datensatz zu. In unserem Fall sind dies die autoethnographischen Aufzeichnungen der Studierenden, welche beispielhaft bereits mit drei anderen Auswertungstechniken induktiv-qualitativ (siehe Kapitel 4), deduktiv-qualitativ (siehe Kapitel 5) und teilautomatisiert quantitativ inhaltsanalytisch ausgewertet wurden (siehe Kapitel 6).

Abbildung 9.5 Menüführung für den Import tabellarischer Datenformate und DataFrames aus SPSS, SAS und Stata



Um die Daten mittels des Drop-down-Menüs zu öffnen, klicken Sie zunächst oben rechts auf die Schaltfläche „Import Dataset“. Hiernach öffnet sich ein Menü, bei dem Sie auswählen können, welchen Datentypus sie laden wollen. „From Text“ bedeutet, dass Sie eine Textdatei einlesen wollen. Diese können beispielsweise die Endungen txt, tsv oder csv haben.

Wenn dieser Schritt erfolgreich ist, dann erscheint Ihr Datensatz im „Data“-Fenster mitsamt einer Beschreibung der Beobachtungen (hier: Texte) sowie der Variablen (hier: Wörter). Sie können auch den Datensatz gesondert betrachten, indem Sie den Befehl `View()` eingeben und in den Klammern den Namen Ihres Datensatzes eingeben, der die Grundlage für Ihre Analyse in RStudio darstellt. Mit `summary(df)` zeigen Sie die deskriptiven Statistiken aller Variablen an, aus denen der Datensatz besteht und für die die statistischen Maßzahlen berechnet werden können. Standardmäßig werden Ihnen Minimum, 1. Quantil, Median, Mittelwert, 3. Quartil und Maximum angezeigt.[7]

Die Zusammenfassung einzelner Variablen können Sie entweder durch eine kombinierte Verwendung des `summary()`-Befehls und des Aufrufs einer Variablen im Datensatz durch das $-Zeichen erreichen, zum Beispiel `summary(df$vorlesung)`, falls Sie wissen wollen, wie die deskriptiven Maßzahlen im Falle des Wortes „Vorlesung“ sind. Alternativ steuern Sie die Variable durch die Verwendung des Variablennamens in Anführungszeichen und eckigen Klammern an, zum Beispiel `summary(df["vorlesung"])`. Sie können auch mehrere Variablen ansteuern und die Maßzahlen ausgeben lassen, indem Sie in der eckigen Klammer Wörter mit dem `c()`-Befehl ansteuern oder die Spaltennummern der Variablen im Datensatz ansteuern (z. B. `summary(df[c("seminar","uebung","professor")])`, oder `summary(df[c(235,293,664)])`, da diese Worte an den entsprechenden Stellen im Datensatz vorliegen), der als Grundlage für die Korrespondenzanalyse dient).

9.3.5 Auswahl der Variablen für die Analyse

Im nächsten Schritt müssen Sie den Datensatz auf diejenigen Variablen begrenzen, auf deren Basis die Korrespondenzanalyse durchgeführt wird. Die Begrenzung erfolgt über eine gezielte Auswahl der zu analysierenden Variablen. Um die Variablen auszuwählen, gibt es zwei Möglichkeiten.

---

7 Wenn Sie weitere, deskriptive Statistiken zu Ihren Variablen berechnen wollen, dann können Sie das Paket *psych* installieren. Der enthaltene `describe()`-Befehl liefert neben den genannten Maßzahlen die Anzahl der gültigen Beobachtungen, Spannweite, Standardabweichung, Schiefe und Krümmung der Variablen. Gegenwärtig problematisch am Paket psych ist, dass es zu Konflikten mit anderen Paketen kommen kann. Daher sollten Sie stets nach der Verwendung von psych das Paket wieder deaktivieren mit dem Befehl `detach(psych)`.

1. Sie wählen gezielt Variablen über den Variablennamen wie in Code 9.6 aus. Die Variablennamen erhalten Sie, indem Sie `variable.names()` eingeben und in die Klammer den Namen Ihres Datensatzes eingeben – im vorliegenden Fall:

Code 9.6 Befehl zur Ausgabe der Variablennamen eines Datensatzes

```
variable.names(df)
```

Durch Ausführen des Codes 9.6 erhalten Sie dann unten links im Ausgabefenster (auch „Konsole“ genannt) eine Liste mit den Variablennamen. Wenn Sie einen Datensatz haben und sich beispielsweise nur für bestimmte Begriffe interessieren, dann können Sie diese in der Folge aus der Liste abtippen oder direkt herauskopieren. Um die Daten selbst auszuwählen und in einen neuen Datensatz zu überführen, bieten sich nun zwei Möglichkeiten an. Erstens können Sie die Begriffe bzw. Variablen, mit denen Sie arbeiten möchten, mit `c()` aneinanderreihen und in eckige Klammern hinter den Datensatz `df` setzen (z. B. `df[c("Frau","Mann")]`). Alternativ können Sie alle Variablen auswählen, indem Sie die Spaltenposition angeben, in der diese im Datensatz gespeichert sind. Sie werden bestimmt bemerkt haben, dass bei Ausführung des Befehls `variable.names(df)` Zahlen neben den einzelnen Variablennamen aufgelistet wurden. Dabei handelt es sich um die Spaltenzahlen. Wenn Sie die Namen durch die entsprechenden Zahlen ersetzen und durch den `c()`-Befehl innerhalb der eckigen Klammern ansteuern, dann erhalten Sie identische Teildatensätze, wie Ihnen Code 9.7 zeigt.

Code 9.7 Erstellung von Teildatensätzen über Variablennamen und Spaltennummern

```
#Auswahl einiger weniger Begriffe als Variablen
teildatensatz1 <- df[c("monat","stoff","begriff","kosten","literatur")]
teildatensatz2 <- df[c(943,944,945,951,955)]

# Abfrage, ob beide Datensätze identisch sind
identical(teildatensatz1, teildatensatz2)
```

Der `identical()`-Befehl in Code 9.7 testet dabei, ob beide Objekte – hier: Teildatensätze – identisch sind. Sie sind identisch, wenn nach Ausführung dieser Kodezeile `TRUE` in der Konsole erscheint. Sollten Sie hingegen `FALSE` sehen, so sind beide Objekte, die Sie geprüft haben, nicht identisch.

Alternativ können Sie eine durchgängige Reihe von Variablen mit dem Sequenz-Befehl ansteuern. Dieser wird mit `seq()` durchgeführt. Wenn Sie nun `df[seq(250,300)]` eingeben, dann werden alle Wörter bzw. Variablen angesteuert, die sich in den Spalten 250 bis 300 befinden. Diese können dann für weitere Analysen verwendet werden.

2. Sie können Variablen gezielt durch den `select()`-Befehl auswählen. Sie können sich hier das Leben erleichtern, indem Sie die Pakete `tidyverse` und `tidyselect` installieren und in die Sitzung laden und durch eine Kombination des Befehls und des Syntax (= Anordnung und Aufruf des Codes im Editor) Spalten, Zeilen oder bestimmte Anteile der Daten auswählen. Sie sehen hier die Zeichenfolge %>%. Diese zeigt R an, dass Sie auf einen Datensatz zurückgreifen, der sich links von diesen Zeichen befindet und dass rechts von diesen Zeichen dann die Befehle aufgeführt werden, die sich auf bestimmte Variablen oder Bereiche des Datensatzes beziehen. Sie haben dabei mehrere Möglichkeiten, Variablen auszuwählen: Diese umfassen die Suche nach Variablen, die mit einem (oder mehreren) Buchstaben starten, diese irgendwo enthalten bzw. mit diesen Buchstaben enden oder Zahlen enthalten. Die dazugehörenden, in Code 9.8 aufgeführten Befehle lauten wie folgt.

Code 9.8 Auswahl von Variablen mittels regulärer Ausdrücke in tidyverse und tidyselect (Fortsetzung nächste Seite)

```
#Laden der Pakete
library(tidyselect)
library(tidyverse)

#Auswahl von Variablen
df %>% select(starts_with("a")) # wählt alle Variablen aus, die mit "a" starten
df %>% select(-starts_with("a")) # wählt alle Variablen aus, die nicht mit "a"
starten

df %>% select(ends_with("a")) # wählt alle Variablen aus, die auf "a" enden
df %>% select(-ends_with("a")) # wählt alle Variablen aus, die nicht auf "a"
enden

df %>% select(contains("soziologie")) # wählt alle Variablen aus, die das
spezifische Wort Soziologie beinhalten

df %>% select(matches("[a-f]e")) # wählt alle Variablen aus, die den regulären
Ausdruck [a-f]e beinhalten
```

```
df %>% select(num_range("X", 1:5)) # wählt alle Variablen aus, die ein X
beinhalten und danach eine Zahl zwischen 1 und 5

## Reduktion des Datensatzes auf alle Variablen, die mit f, g, … bis s beginnen:

Reduzierter_datensatz <- df %>% select(matches("^[f-s]"))
```

9.3.6 Verwendung regulärer Ausdrücke und Exklusion fehlender Werte

Wir haben an dieser Stelle einen regulären Ausdruck (kurz: regex) eingeführt. Unter regulären Ausdrücken versteht man in RStudio und Python eine Kombination von Buchstaben und Zeichen (sogenannte Zeichenketten bzw. Strings), mit deren Hilfe man eine Zeichenabfolge oder eine Menge von Buchstaben an einer bestimmten Stelle in einem Wort, einem Variablennamen etc. auffindet. Am Beispiel des letzten Befehls bedeutet `[f-s]`, dass alle Variablen, die Buchstaben von f bis s beinhalten, ausgewählt werden. Das ^-Zeichen sagt, dass diese Buchstaben an der ersten Stelle der Variablen auftreten.[8]

Der letzte Schritt, der noch zu bestreiten ist für die Vorbereitung der Korrespondenzanalyse, ist der Ausschluss fehlender Fälle (siehe Code 9.9). Dies ist dahingehend relevant, da die statistischen Berechnungen, die Sie in der Regel in RStudio und spezieller im Falle der Korrespondenzanalyse durchführen, auf vollständige Daten angewiesen sind. Vollständige Daten liegen vor, wenn jede Variable (hier: Wort) einen eigenen Wert hat (z. B. das Wort „Soziologie“ kommt fünfmal vor, „Fernsehen“ keinmal usw.). Eine solche Auswahl können Sie vornehmen, indem Sie `na.omit()` eingeben und das Ergebnis einem neuen Objekt mit <- zuweisen.

Code 9.9 Befehl zum Ausschluss fehlender Werte

```
## Exklusion aller fehlenden Werte
Datensatz_ohne_fehlende_werte <- na.omit(df)
```

8 Wenn Sie genauer auf die Möglichkeiten eingehen möchten, wie Sie mit solchen Zeichenketten bzw. Strings umgehen können, können Sie Hilfe und einen Überblick auf dem Cheatsheet *https://raw.githubusercontent.com/rstudio/cheatsheets/master/strings.pdf* finden.

Kombinieren wir alle Befehle, die bis hierhin angesprochen wurden, dann sieht ein Beispielskript, mit dem Sie sowohl alle Pakete als auch Ihre Daten laden und aufbereiten können, wie in Code 9.10 aus.

Code 9.10 Gesamtsyntax zum Einlesen der Pakete und Daten, Erkundung des Datensatzes und Erstellen von Teildatensätzen

```
#Einladen der Pakete_____________________________####
library(FactoMineR)
library(factoextra)
library(tidyselect)
library(tidyverse)
library(ggplot2)

#Einladen der Daten_____________________________####
df <- read.csv("C:/Dateipfad/Datensatz.csv") # hier Ihren Dateipfad und den
Namen Ihres Datensatzes ergänzen

## Erkundung des Datensatzes_____________________####
summary(df)
variable.names(df)

## Selektion von Variablen_____________________####
df_reduced <- df %>% select(-starts_with("X")) # Entfernt potenziell leere
Variablen
df_reduced <- na.omit(df_reduced)

## Betrachten des Datensatzes_____________________####
View(df_reduced)
```

9.4 Durchführung einer Korrespondenzanalyse

9.4.1 Test/Voraussetzungen für die Durchführung einer Korrespondenzanalyse

Wenn Sie die Daten ausgewählt haben, die Sie in der Korrespondenzanalyse analysieren wollen, dann sollten Sie nun testen, ob sich eine Analyse überhaupt lohnt. Das bedeutet in unserem Falle, dass wir prüfen müssen, ob überhaupt eine Struktur in den Daten vorliegt oder ob die Wörter nur zufällig gemeinsam auftreten (z. B. Soziologie, Fernsehen, Seminar usw.). Wenn sie nur zufällig gemein-

sam auftreten (was bei der Sprache und der Aufgabenstellung einer Autoethnographie bei Studierenden während der Corona-Zeit sehr unwahrscheinlich ist), dann können wir auch nicht davon ausgehen, dass Themen in den Texten zu finden sind, die wir interpretieren können. Doch wie genau können wir testen, ob es eine Strukturierung gibt?

Hier können Sie den Chi-Quadrat-*Unabhängigkeitstest* verwenden. Dieser beantwortet generell die Frage, ob Zeilen und Spalten einer Matrix bzw. eines Datensatzes unabhängig voneinander sind. In unserem Falle würde dies bedeuten, dass wir testen, ob die verwendeten Wörter unabhängig von den Texten der 50 Studierenden auftreten. Dabei werden zwei konkurrierende Hypothesen aufgestellt.

- **H0:** Zeilen und Spalten sind unabhängig voneinander (bzw. Wörter werden unabhängig von Texten verwendet).
- **H1:** Zeilen und Spalten sind abhängig voneinander (bzw. Wörter werden systematisch in Texten verwendet).

Der in Code 9.11 aufgeführte Befehl, mit dessen Hilfe der Chi-Quadrat-Unabhängigkeitstest aufgerufen werden kann, lautet in R `chisq.test()` und produziert den Output 9.1 in Ihrer Konsole.

Code 9.11 Durchführung eines Chi-Quadrat-Tests in R

```
# Chi²-Test -----------------------------------------------------------------
chisq.test(df_reduced) # Chi-Quadrat-Test auf dem reduzierten Datensatz
```

Output 9.1 Ergebnis des Chi-Quadrat-Tests

```
Pearson's Chi-squared test

data: df_reduced
X-squared = 105599, df = 47726, p-value < 2.2e-16
```

Doch was bedeuten all diese Werte? Wie sind sie zu interpretieren und woraus kann man schließen, ob nun eine Struktur der Wörter über die Texte hinweg vorliegt oder nicht?

Der wichtigste Wert, der Ihnen angezeigt wird, ist der p-Wert bzw. `p-value`. Dieser zeigt die Irrtumswahrscheinlichkeit an, dass ihre Daten keiner Struktur folgen bzw. die Wörter zufällig über die Texte hinweg verteilt sind und Sie fälschlicherweise annehmen, dass diese Wörter doch einer sinnvoll zu deutenden

Struktur folgen (sogenannte α-Fehler). Dieser Wert sollte möglichst gering sein – mindestens aber nach Konvention kleiner als 0.05. Ein Wert von 0.05 bedeutet, dass Sie in fünf von 100 Fällen fälschlicherweise von einem Zusammenhang zwischen Wörtern und Texten und damit einer systematischen Auswahl dieser Wörter von Studierenden ausgehen, obwohl diese zufällig gewählt wurden.

9.4.2 Durchführung der Korrespondenzanalyse

Kommen wir nun zur Korrespondenzanalyse selbst. Im Grunde benötigen Sie hierzu nur einen einzelnen Aufruf des `CA()`-Befehls. Dieser ist so strukturiert, dass Sie zunächst die Daten angeben müssen, auf den sich dieser Befehl bezieht, ehe Sie die Optionen angeben, beispielsweise ob Sie direkt eine Grafik angezeigt haben wollen oder nicht (`graph = TRUE` oder `graph = FALSE`), oder die Anzahl der Dimensionen (= potenzieller Themen, `ncp =`), für die Informationen ausgegeben werden.

Für eine Korrespondenzanalyse ohne Grafik, die Informationen zu zehn potenziellen Themen umfasst und noch keinen eigenen Graphen erzeugt, können Sie entsprechend folgende Zeilen in Ihren Editor eingeben und ausführen. Beachten Sie dabei, dass sie „Strg + Enter“ oder den „Run“-Knopf drücken, nachdem Sie die entsprechenden Zeilen markiert haben.

Code 9.12 Durchführung einer Korrespondenzanalyse und Ausgabe der Ergebnisse mit CA() und summary()

```
res <- CA(datensatz, graph=FALSE, ncp = 10)
summary(res)
```

Der `summary()`-Befehl zeigt Ihnen an, welche Ergebnisse die Korrespondenzanalyse geliefert hat. Die Ergebnisse haben wir zuvor mittels des Pfeiles in das Objekt namens `res` abgespeichert. Dabei werden Ihnen in der Konsole mehrere Ausgaben angezeigt. Einerseits die sogenannten Eigenwerte (auf Englisch: *eigenvalues*). Diese zeigen an, wie viel Varianz der Gesamtdaten durch die einzelnen Dimensionen, d. h. potenzieller Themen abgedeckt werden. Varianzaufklärung heißt hier nur, dass R versucht, zu erkennen, wie stark die Themen strukturiert sind und wie viel dieser Struktur durch die jeweilige Dimension erfasst wird. Je höher die Werte, desto mehr Varianz wird aufgeklärt. Erwarten Sie bei den Themen selbst keine hohen Werte pro Thema, was darauf zurückzuführen ist, dass Sprache komplex ist und Sie entsprechend viele Variablen, d. h. Wörter haben, die in die Analyse einfließen. Um diese Varianzaufklärung zu erkunden, bietet Ihnen die Ausgabe in der Konsole die Eigenwerte und Varianzaufklärung einzel-

ner Dimensionen (Ausgabe: `% of var.`) und kumulierte Varianzaufklärung an (Ausgabe: `cumulative % of var.`), wie in Output 9.2 deutlich wird.

Output 9.2 Exemplarische Ausgabe der Varianzaufklärung der ersten fünf Dimensionen der Korrespondenzanalyse

```
Eigenvalues
                        Dim.1  Dim.2  Dim.3  Dim.4  Dim.5
Variance                0.142  0.136  0.124  0.113  0.111
% of var.               4.290  4.118  3.757  3.421  3.368
Cumulative % of var.    4.290  8.408 12.165 15.587 18.955
```

Die Ausgabe bietet Ihnen weiter Informationen zu den Texten und Wörtern, die sich bei der Verortung auf den einzelnen Themendimensionen sowie durch das Varianzaufklärungspotenzial für die einzelnen Themen ergeben. Dabei werden insgesamt vier Werte für die jeweils zehn ersten Zeilen (= Texte) und Spalten (= Wörtern) ausgegeben. Dabei handelt es sich erstens um die „Trägheit", d. h. *inertia,* die mit 1000 multipliziert wird (`Iner*1000`). Dieser Wert deutet an, wie viel der Varianz in der gesamten Korrespondenzanalyse durch den einzelnen Text (`Rows`) bzw. das einzelne Wort (`Columns`) erklärt wird. Erwarten Sie auch hier keine zu hohen Werte, da Texte in der Regel hochdimensional sind und damit für den PC viele Themen enthalten, von denen nicht alle zwangsläufig sinnvoll durch die Forscher*innen interpretiert werden können.

Dem folgen zweitens die Koordinaten der ersten zehn Texte und Wörter auf den jeweiligen Dimensionen (z. B. Dim.1 für die Koordinaten auf der ersten Dimension). Dabei gilt, dass Werte, die nahe 0 liegen, sehr gewöhnlich sind und in vielen Texten vorliegen. Je weiter entfernt der Wert von 0 ist (z. B. ± 2), desto charakteristischer ist der Text bzw. der Begriff für die jeweilige Dimension (Blasius und Schmitz 2013). Da die Korrespondenzanalyse immer von Gegensätzen ausgeht, ist es in der Folge sinnvoll, die Texte und Wörter zu betrachten, die auf den jeweiligen Dimensionen am weitesten von 0 entfernt sind.

Die dritte Information der Ausgabe leistet einen Beitrag zur Aufklärung der Varianz der jeweiligen Dimensionen, die *contribution* (oder `ctr`). Auch hier gilt, dass größere Werte einen höheren Prozentsatz der Varianz auf den einzelnen Dimensionen erklären und damit die jeweilige Dimension genauer vorstrukturieren. Wenn ein Text beispielsweise einen Beitrag von 3,507 (idno 9) bei der 1. Dimension aufweist, dann vereinigt dieser 3,5 % der Varianz der ersten Dimension auf sich. Bedenkt man, dass wir 50 Texte aufgenommen haben, dann wäre bei einer völligen Zufälligkeit der Ergebnisse zu erwarten, dass dieser Text 1/50 = 2 % der Varianz der ersten Dimension aufklärt. Wenn sich im Verlauf der späteren, visuellen Interpretation der Ergebnisse zeigen sollte, dass dieser Text typisch

dafür ist, welches Thema Dimension 1 beinhaltet, dann liefert dieser Wert einen Hinweis darauf, dass Sie sich diesen Text genauer ansehen und mit qualitativen Methoden auswerten sollten. Gleiches gilt für die Wörter. Alle Wörter, die überdurchschnittlich viel Varianz in den Daten für die jeweilige Dimension erklären, kommen potenziell dafür in Frage, tiefergehend analysiert zu werden. Da wir mit einem Korpus bestehend aus 975 Wörtern (aufrufbar mittels des `ncol()`-Befehls) arbeiten, kommen potenziell alle Wörter in Frage, die mehr als 1/975 = 0.1 % Varianz der jeweiligen Dimension aufklären.

Zuletzt gibt es den Kosinus²-Wert (`cos2`). Dieser zeigt an, wie gut ein Text bzw. eine Variable auf einer Dimension „liegt". Je kleiner der Wert ist, desto eher liegt ein Punkt auf einer Linie und je näher am Koordinatenursprung ist dieser (mit einer Richtung), während größere Werte eine Abweichung von dieser Linie und eine große Distanz anzeigen. Hier können Sie auf große Werte achten, die weit vom Zentrum Ihrer Verteilung entfernt sind und daher die Themen am weitesten aufspannen und die Gegensätze am besten verkörpern (z. B. Universitätsbesuch versus Freizeit), die Sie in Ihren Daten vermuten können. Output 9.3 liefert Ihnen eine Beispielausgabe dieser Werte.

Output 9.3 Informationen über Position und Varianzaufklärung der ersten zehn Texte und Wörter auf den ersten beiden Dimensionen (Fortsetzung nächste Seite)

```
Rows (the 10 first)
          Iner*1000   Dim.1  ctr  cos2     Dim.2  ctr  cos2
1         |  45.476 | 0.205 0.435 0.014  | 0.274 0.804 0.024
2         |  70.117 | 0.197 0.192 0.004  | 0.335 0.581 0.011
3         |  47.552 | 0.046 0.025 0.001  | -0.171 0.366 0.010
4         |  60.491 | -0.043 0.014 0.000 | 0.046 0.017 0.000
5         |  62.586 | -0.119 0.195 0.004 | 0.072 0.074 0.002
6         |  84.368 | 0.047 0.040 0.001  | -0.088 0.147 0.002
7         |  52.979 | 0.441 2.680 0.072  | 0.008 0.001 0.000
8         |  61.423 | -0.277 1.693 0.039 | 0.136 0.425 0.009
9         |  64.198 | -0.551 3.507 0.077 | -0.182 0.397 0.008
10        |  50.950 | 0.301 1.101 0.031  | 0.002 0.000 0.000

Columns (the 10 first)
                   Iner*1000   Dim.1  ctr  cos2     Dim.2  ctr  cos2
frau               |   3.529 | 0.230 0.012 0.005  | -0.106 0.003 0.001
genug              |   1.933 | -0.150 0.011 0.008 | 0.098 0.005 0.003
vervollstaendigen  |   2.206 | -0.528 0.043 0.028 | 0.261 0.011 0.007
social             |   8.232 | 0.375 0.078 0.013  | 0.317 0.058 0.010
kennen             |   5.946 | -0.091 0.008 0.002 | 0.205 0.044 0.010
einrichten         |   2.772 | -0.740 0.085 0.043 | -0.297 0.014 0.007
```

```
vortag              |   2.074 | -0.031 0.001 0.000 | 0.030 0.001 0.000
gesicht             |   2.998 | 1.467 0.333 0.157  | 0.140 0.003 0.001
sozialforschung     |   1.618 | 0.024 0.000 0.000  | 0.145 0.003 0.002
wecken              |   4.710 | 0.034 0.000 0.000  | -0.023 0.000 0.000
```

Sollten Sie passive Variablen definiert haben (siehe Kapitel 9.3.4), dann können Sie diese mit der Option `col.sup =` in die Analyse einbeziehen. Beachten Sie, dass Sie hierfür die Spaltennummern benötigen, die den Variablen zugewiesen sind. Diese Spaltennummern können Sie ermitteln, indem Sie `View(colnames(datensatz))` eingeben. Vergessen Sie bitte nicht, das Objekt datensatz zu ersetzen, wenn Sie Ihren Datensatz anders benannt haben. Dann öffnet sich ein Fenster, das den Variablennamen und die Zeilennummer beinhaltet. Wenn Sie mehrere Variablen auswählen wollen, dann können Sie die Spaltenzahlen entweder mit dem `c()`-Befehl oder mit `seq()` auswählen. Texte wählen Sie analog als passive Beobachtungen aus, indem Sie `row.sup =` als Option im `CA()`-Befehl angeben (siehe Code 9.13). Wenn Sie passive Variablen und/oder Beobachtungen angeben wollen, dann verändert sich der Befehl leicht und hat dann folgende Form.

Code 9.13 Durchführung einer Korrespondenzanalyse mit passiven Variablen und passiven Beobachtungen

```
# Exklusion des Textes Nr. 30 und der Begriffe an Stellen 664 und 761
res <- CA(datensatz, graph=FALSE, ncp = 10,
          row.sup=30, col.sup = c(664,761))
summary(res)
```

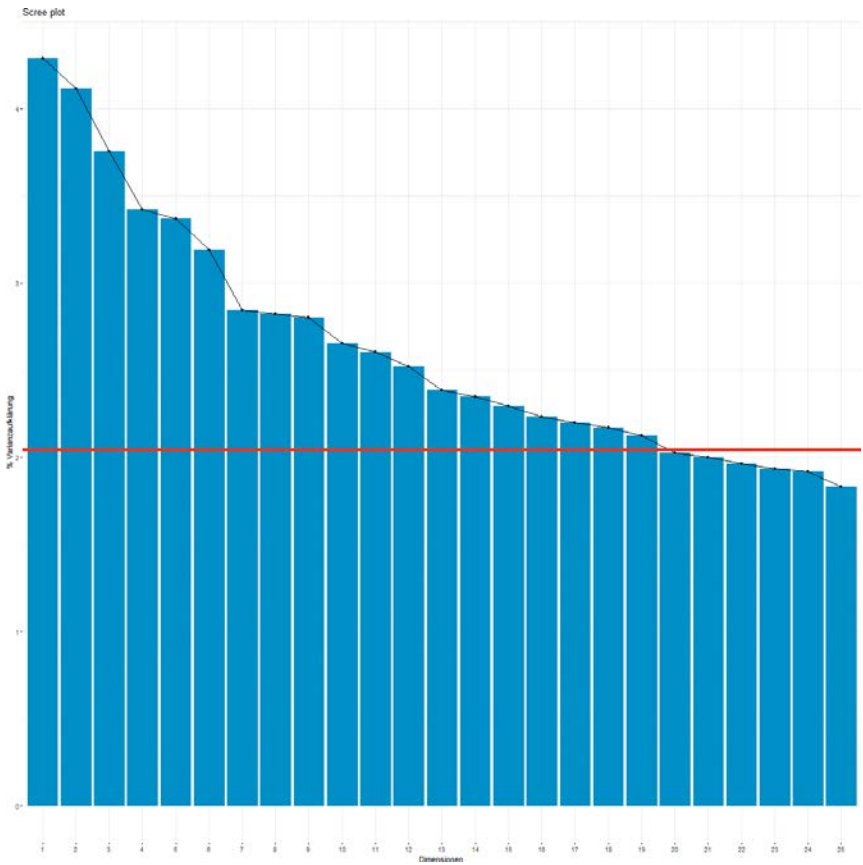
9.4.3 Auswahl der Dimensionszahl für die spätere Interpretation

Je größer der Datensatz, umso mehr müssen Sie entscheiden, wie viele Dimensionen bzw. Themen Sie sich grafisch ansehen und inhaltlich interpretieren sollten. Hjellbrekke (2019) empfiehlt hier, dass Sie potenziell alle Dimensionen interpretieren sollten, die überdurchschnittlich viel Varianz erklären. Dieser Durchschnitt errechnet sich aus 1 geteilt durch die Anzahl der Dimensionen.[9] Im vorliegenden

9 Dabei ermittelt R zunächst, wie viele Spalten und wie viele Zeilen Ihr Datensatz hat. Danach wird die Spalten- und Zeilenzahl verglichen und die geringere Menge zur Berechnung der maximalen Dimensionszahl herangezogen und von dieser Menge eins abgezogen. Im vorliegenden Fall haben wir 50 Texte und 975 Wörter. Entsprechend wählt R nun 50 aus und subtrahiert 1, woraus sich die Gesamtzahl von 49 Dimensionen ergibt.

Fall sagt uns RStudio, dass es insgesamt 49 Dimensionen gefunden hat, d. h. alle Dimensionen potenziell interpretiert werden können, die mehr als 2,04 % der Varianz aufklären. Dies wären im vorliegenden Fall die ersten 19 Dimensionen. Damit Sie aber nicht immer in der Konsole und dem Output nachschauen müssen, der durch den `summary()`-Befehl ausgegeben wird, können Sie auch eine Grafik ausgeben lassen, in der Sie die Varianzaufklärung und diesen kritischen Wert anzeigen lassen können. Die hierfür benötigten Pakete sind `factoextra` und `ggplot2`. Wenn Sie diese Pakete hineingeladen haben, dann können Sie Abbildung 9.6 erzeugen und interpretieren. Sie zeigt einen sogenannten Screeplot, in dem Informationen über die Varianzaufklärung der einzelnen Dimensionen, absteigend nach dem Prozentsatz der erklärten Varianz, enthalten sind.

Abbildung 9.6 Screeplot des Varianzaufklärungspotenzials der ersten 25 in der Korrespondenzanalyse gefundenen Dimensionen



In der ersten Zeile erzeugen wir den ermittelten Wert von 49 Dimensionen, der später die rote „Grenze“ zieht, die signalisiert, wie viele Dimensionen wir potenziell analysieren sollten. Diese Zeile weist dem Objekt `grenzwert` den Wert 1/49*100 = 2.04 % zu. Dieser Wert wird dann als Ausgangspunkt für die rote Linie herangezogen. In der darauf folgenden Befehlszeile geben wir mittels `fviz_screeplot()` an, dass wir eine Grafik erstellen wollen, bei der die aufgeklärte Varianz der einzelnen Dimensionen als Balken abgebildet werden und zwischen diesen Werten eine gepunktete Linie verlaufen soll. Dabei rufen wir die Ergebnisse der Korrespondenzanalyse, die wir vorher in `res` gespeichert haben, an der ersten Stelle im Code-Befehl auf. Wenn Sie die Ergebnisse als Objekt mit Namen `Ergebnisse` abgespeichert haben, dann müssten Sie entsprechend `fviz_screeplot(Ergebnisse)` eingeben. Standardmäßig gibt Ihnen R eine Übersicht über die ersten zehn Dimensionen aus. Mit dem Befehl `ncp =`, der nach dem Komma folgt, können Sie die Anzahl der angezeigten Dimensionen verstellen. In unserem Falle haben wir angegeben, dass wir 25 Dimensionen abgebildet haben möchten. Die Wahl der abgebildeten Dimensionen sollte an dieser Stelle zur Illustration dienen. Sie können natürlich auch 10, 20, oder 50 Dimensionen abbilden – sofern die einzelnen Werte hinreichend kenntlich gemacht werden. Wenn Sie einen einfachen Screeplot mit 5 Dimensionen haben möchten und Ihre Ergebnisse in dem Objekt `Ergebnisse` abgespeichert haben, dann würde der Befehl `fviz_screeplot(Ergebnisse, ncp=5)` lauten. Sie hätten auch alle Dimensionen abbilden können oder auch nur diejenigen, die nach einer weiteren Vorauswahl eine höhere Varianzaufklärung als der Grenzwert aufweisen, den wir weiter oben ermittelt haben. Sie haben auch die Möglichkeit, die X- und Y-Achsenbeschriftungen zu verändern, indem Sie hinter dem `fviz_screeplot()+ xlab("[Titel der X-Achse"]) + ylab(["Titel der Y-Achse"])` angeben. Sie müssen nur [„Beschriftung“] durch einen Text Ihrer Wahl ersetzen. Beachten Sie dabei, die Beschriftung in Anführungszeichen zu setzen. Damit sagen Sie RStudio, dass es sich um eine Zeichenfolge handelt. Insgesamt zeigt Ihnen Code 9.14, wie Sie einen Screeplot mit Balken und in „Dimensionen“ umbenannte X-Achse sowie eine umbenannte Y-Achse erzeugen können.

Code 9.14 Visualisierung eines Screeplots für die Auswahl der potenziell zu interpretierenden Dimensionen

```
grenzwert <- 1/49*100

dimensionen_screeplot <- fviz_screeplot(res, ncp=25) +
 geom_hline(yintercept = grenzwert, linetype=4, color="red", size=1.5) +
 xlab("Dimensionen") +
 ylab("% Varianzaufklärung")
```

Den Code müssen Sie sich wie eine Aneinanderreihung von in sich abgeschlossenen, jedoch sich ergänzenden Befehlselementen vorstellen – analog zu kleinen Schachteln (jede Code-„Schachtel" durch Klammern ist abgeschlossen), die in eine größere Schachtel passen. Folglich sehen Sie in Code 9.14 danach ein + Zeichen und einen Zeilenumbruch. Dieses Pluszeichen zeigt R an, dass noch ein weiterer für die Berechnung notwendiger Befehl folgt, mit dem etwas an der bereits erstellten Grafik verändert werden wird. Dieser benötigt einen Zeilenumbruch, damit R zwischen dem ersten Befehl, der hier den Screeplot erstellt, und dem zweiten Befehl, der die horizontale Linie erstellt, unterscheiden kann. Der `geom_hline()`-Befehl stammt aus dem `ggplot2`-Paket und benötigt als Eingabe einen Wert, auf dem die horizontale Linie (entlang der Y-Achse des Screeplots) gezogen wird. Würden wir an dieser Stelle `yintercept = 3` eingeben, dann würde eine horizontale Linie gezeichnet werden, die beim Wert 3 auf der Y-Achse beginnt.

In der Schachtel bzw. dem Befehlselement folgen die Optionen für Linienart, -dicke und -farbe, die jeweils durch Kommata abgetrennt werden. Die Option `linetype` kennt sechs verschiedene Ausprägungen, welche durch Zahlen codiert sind. Wenn Sie eine 1 eingeben, so erzeugen Sie eine lange, durchgezogene Linie. Bei einer 2 wird hingegen eine gestrichelte Linie generiert, während eine 3 eine gepunktete Linie erzeugt. Eine 4 an dieser Stelle erzeugt eine Strichpunktlinie, eine 5 eine langgezogene, gestrichelte Linie und eine 6 letzten Endes eine Punkt-Strichlinie. Farbe und Liniendicke geben Sie durch die `color`- sowie die `size`-Option an. Mittels der `color`-Option können Sie einstellen, welche Farbe die Linie hat, während die `size`-Option die Größe Ihrer Buchstaben (wie z. B. in Word oder Open Office) einstellt. Dabei haben wir im vorliegenden Fall die Farbe Rot gewählt. Es gibt aber viele weitere Farben und Farbpaletten. Um die einzelnen Farben einzustellen, können Sie einen Standard-Farbnamen (z. B. „rot" und „blau", jedoch nicht grünviolett) oder eine Zeichenkombination (z. B. aus RGB-Farbpalette) angeben, die von Ihrem PC als Farbe dekodiert wird. Eine Übersicht über die Farbnamen erhalten Sie, wenn Sie auf *www.duckduckgo.com* oder einer anderen Suchmaschine nach „r colors cheat sheet" suchen.[10] Hier sind die Farben übersichtlich aufgelistet. Die Farbpaletten können Sie in der Anleitung von `ggplot2` nachlesen und Ihre Grafik nach Belieben anpassen.

Diese Visualisierung erlaubt es Ihnen zudem, nach einem weiteren Kriterium abzuschätzen, wie viele Dimensionen Sie wohl zu interpretieren haben. Dabei handelt es sich um das sogenannte Ellenbogen-Kriterium (von Englisch *ellbow-criterion*). Ein sogenanntes Ellenbogen-Kriterium liegt vor, wenn die Varianzaufklärung nach einer Dimension weitaus schwächer abnimmt, als dies bei den Dimensionen davor der Fall gewesen ist. Varianzaufklärung meint, dass ein Anteil der in den Daten vorliegenden Streuung auf die Anwesenheit einiger weniger

10 Beispielsweise auf *www.stat.columbia.edu/~tzheng/files/Rcolor.pdf.*

Variablen (hier: Wörter) oder auf eine Dimension (= Thema) zurückzuführen ist. Wenn die erste Dimension bzw. das erste Thema eine Varianzaufklärung von 5 % hat, dann heißt dies lediglich, dass 5 % der Streuung in unserem Datensatz auf die Wörter zurückgehen, die dieses Thema konstituieren. Je mehr Varianz einige wenige Dimensionen erklären, als desto stärker strukturiert wird der Datensatz bezeichnet und desto einfacher und klarer wird die Interpretation Ihres statistischen Modells ausfallen. Das hat zudem den Vorteil, dass Sie zugleich weniger Themen zu interpretieren haben und damit die Zeit beispielsweise für weitere qualitative Analysen nutzen können.

Das Ellenbogen-Kriterium tritt in Abbildung 9.2 bei den Dimensionen 4 und 7 am deutlichsten zutage, was darauf hindeutet, dass Sie eventuell 4 bzw. 7 Dimensionen hinreichend gut deuten können. Für wie viele Dimensionen Sie sich letzten Endes entscheiden, kann Ihnen aber kein statistisches Verfahren abnehmen. Die statistischen Verfahren und visuelle Unterstützungen (z. B. in Screeplots) liefern lediglich Hinweise darauf, wie viele Dimensionen Sie sinnvoll interpretieren können und im vorliegenden Falle, welche Themen womöglich in den 50 autoethnographischen Texten vorliegen.

9.4.4 Speichern von Grafiken

Wenn Sie die Grafik abspeichern wollen, dann können Sie dies auf zwei Arten machen. Sie werden bemerkt haben, dass sich, nachdem Sie den Befehl `fviz_screeplot()` ausgeführt haben, eine Grafik geöffnet hat. Hier können Sie entweder mit einem Rechtsklick auf diese Grafik einen Dialog öffnen, in dem sie die Bilddatei als Bitmap kopieren oder als Metadatei/Postscript-Datei speichern können. Oben finden Sie aber auch ein Menü. Wenn Sie hier auf Datei klicken, dann öffnet sich ein Reiter, in dem Sie auswählen können, ob Sie die Grafik als Metafile, Postscript, PDF, PNG, TIFF, BMP oder JPEG speichern wollen. Sie können darüber hinaus noch auf „kopieren“ klicken, um die Grafik direkt in ein anderes Dokument einzufügen.

Manchmal benötigen Sie aber auch eine angepasste Version einer Grafik. So kann es sein, dass Sie die Grafik in einer bestimmten Größe oder Auflösung benötigen (z. B. wenn Sie die Grafik als Abbildung in eine Abschlussarbeit einfügen möchten, welche in den Druck geben wird). Hierzu können Sie die Grafik, die Sie erzeugt haben, zunächst einem Objekt zuweisen und dieses dann an einem zuvor ausgewählten Ort auf Ihrer Festplatte speichern. Um eine PNG-Grafikdatei zu speichern, können Sie wie in Code 9.15 vorgehen.

Code 9.15 Erstellen und Speichern einer Screeplot-Grafik auf Ihrer Festplatte

```
dimensionen_screeplot <- fviz_screeplot(res, ncp=25) +
 geom_hline(yintercept = grenzwert, linetype=4, color="red", size=1.5)

setwd("C:/Users/[Ihr PC-Name]/Desktop/") # Grafik auf Desktop speichern
png("screeplot_autoethnographie.png",
  width = 600, height = 600, units = "px", pointsize=300)
print(dimensionen_screeplot)
dev.off()
```

Der `setwd()`-Befehl gibt einen Ordner an, an dem Sie die Grafik speichern möchten. Im vorliegenden Falle wurde der Einfachheit halber der Desktop angegeben. Sie müssen natürlich den Pfad verändern, wenn Sie die Datei auf Ihrem Desktop speichern wollen. Beachten Sie hierbei, dass der Pfad, der angegeben ist, mit / (Slash, über Zahl 7 auf Tastatur) und nicht standardmäßig mit \ (Backslash, z. B. wenn Pfad aus Explorer kopiert wird) abgetrennt wird. Sie können hier aber auch einen eigenen Pfad angeben, beispielsweise einen Pfad, in dem Sie die Grafiken oder den generellen „output" speichern. Auch hier ist es empfehlenswert, die Grafiken mit einem für Sie praktischen Namen zu versehen (z. B. mit Erstellungsdatum oder beispielsweise Abbildung1.png, wenn Sie schon eine feste Reihenfolge für Grafiken beispielsweise für Haus- und Abschlussarbeiten vor Augen haben).

Der folgende `png()`-Befehl gibt an, dass eine Grafik erstellt werden soll. Dabei geben Sie zunächst an, wie die Grafik heißen soll, danach stellen Sie Breite (mit Code: `width`) und Höhe (mit Code: `height`) ein. Mit der `units`-Option geben Sie an, auf welcher Basis die Größe des Bildes berechnet werden soll. An dieser Stelle wurde in Code 9.15 angegeben, dass die Größe des Bildes in Pixel (`px`) berechnet werden soll. Sie können aber auch `cm`, `mm`, oder `in` (für Inches) angeben. Die Option `pointsize` gibt die generelle Auflösung an. Standardmäßig werden Grafiken mit `pointsize = 12` erstellt, was eine sehr niedrige Auflösung darstellt. Normalerweise benötigen Druckereien, wenn Sie beispielsweise Ihre Abschlussarbeit erstellen, aber Grafiken mit einer weitaus höheren Auflösung (300, 600 oder 1200 dpi).

Der `print()`-Befehl „schreibt" bzw. zeichnet dann die Grafik. Hier müssen Sie das Objekt angeben, in dem Sie den Screeplot (oder später andere Plots) gespeichert haben. Zuletzt sagt `dev.off()`, dass der Speichervorgang der Grafik beendet werden soll. Danach haben Sie an dem Ort auf Ihrer Festplatte Zugriff auf die Grafik, die sie gespeichert haben.

9.5 Auswertung der Korrespondenzanalyse

9.5.1 Erzeugung der Grafiken für die Interpretation der Ergebnisse

Kommen wir nun zum nächsten Analyseschritt: der grafischen Erkundung der Dimensionen, die durch die Korrespondenzanalyse erzeugt wurden. Sie können die Ergebnisse Ihrer Korrespondenzanalyse mittels des `fviz_ca()`-Befehls visualisieren, indem Sie das Objekt ansteuern, in dem die Ergebnisse der Korrespondenzanalyse gespeichert sind. Da wir uns hier aber mit dem Problem konfrontiert sehen, dass wir viele Variablen haben, empfiehlt es sich, die angezeigten Begriffe zu reduzieren und die Beschriftung so in der Grafik zu platzieren, dass die Beschriftungen der einzelnen Wörter und der Textnummern überlappungsfrei sind. Der Befehl gibt Ihnen stets eine Draufsicht auf zwei Dimensionen, was Ihnen die Möglichkeit eröffnet, unsere Abbildung wie eine Karte zu lesen. Wie genau Sie dies bewerkstelligen, erklären wir Ihnen in Kapitel 9.5.2.

Auf jeden Fall lässt sich eine Fläche, auf der verschiedenfarbige Punkte angezeigt werden, intuitiver als eine Ausgabe mit vielen Zahlenwerten interpretieren, oder als eine Linie (= eine Dimension), auf der Ihre Datenpunkte wie auf einer Perlenkette angeordnet sind und sich stark überlagern. Ein dreidimensionaler Raum wäre ebenfalls nicht so gut interpretierbar, da durch die Tiefe (= 3. Dimension), die für die Interpretation wichtigen Abstände nicht genau erfasst werden können.

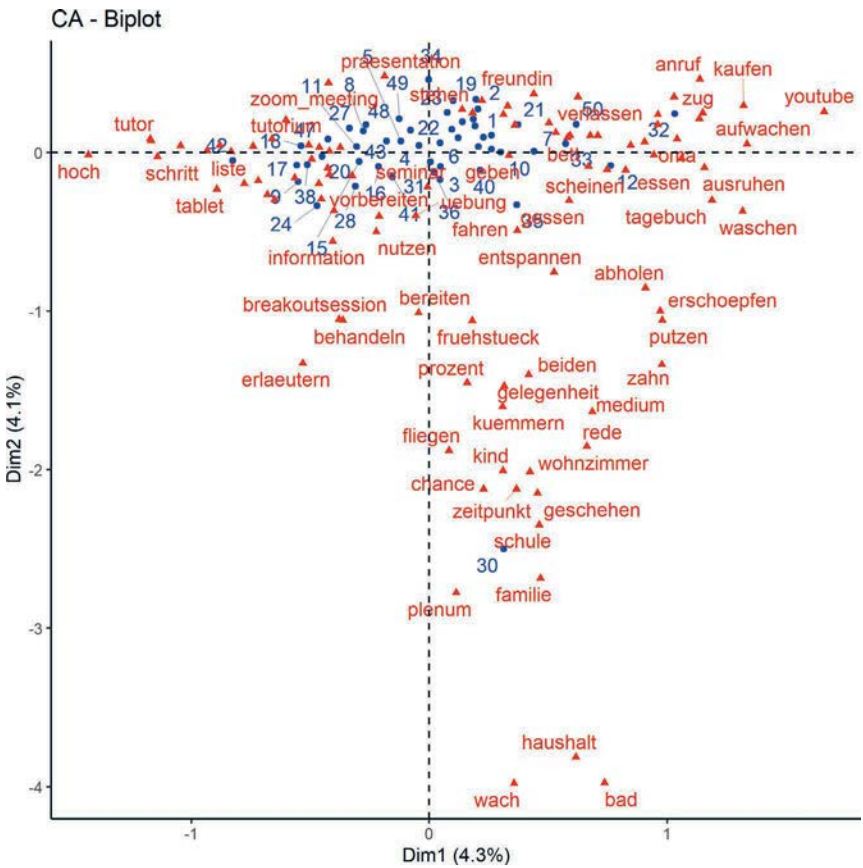
Weiterhin haben wir pragmatisch beschlossen, die Anzahl der angezeigten Wörter auf die 100 zu beschränken, die am stärksten zur Aufklärungskraft der zwei Dimensionen beitragen, die jeweils betrachtet werden. Das hängt damit zusammen, dass wir nicht beliebig viele Dimensionen abbilden und für die Dimensionen, für die wir die Abbildung erstellen, möglichst aussagekräftige Begriffe finden wollen. Das verhindert zugleich, dass die Grafik überfrachtet wird. Bei zu vielen abgebildeten Begriffen können wir irgendwann nicht mehr zwischen den Punkten unterscheiden oder die Begriffe nicht lesen. Damit blockieren Sie Ihren Interpretationsprozess.

Code 9.16 Visualisierung der ersten beiden Dimensionen sowie Dimension 1 und 3 der Korrespondenzanalyse

```
# Visualisierungen der ersten beiden Dimensionen________####
fviz_ca(res, select.col = list(contrib = 100), repel = TRUE)

# Dimensionen 1 und 3
fviz_ca(res, select.col = list(contrib = 100), axes=c(1,2), repel = TRUE)
```

Code 9.16 ermöglicht es Ihnen, die ersten beiden Dimensionen bzw. Dimension 1 und Dimension 3 abzubilden. Die Option `select.col =` sagt, dass hier eine Auswahl der Spalten, d. h. der Wörter getroffen wird, die in die Analyse eingebettet wurden. Das nun folgende `list(contrib = 100)` zeigt an, dass auf eine Liste der 100 Begriffe mit dem höchsten Varianzaufklärungsbeitrag zurückgegriffen wird. Die Option `repel = TRUE` sagt R, dass die Beschriftung so platziert werden soll, dass die Beschreibung der Variablen überlappungsfrei ist. Die `axes`-Option signalisiert, welche Dimensionen für die Analyse herangezogen werden sollen. Hier sagt `c(1,2)`, dass eine Auswahl von Dimension 1 und 2 getroffen wird, die angesteuert werden sollen, um eine Grafik zu erstellen (Abbildung 9.7).

Abbildung 9.7 Abbildung der ersten beiden Dimensionen der Korrespondenzanalyse



9.5.2 Interpretation der ersten beiden Dimensionen

Abbildung 9.7 zeigt dabei die Verortung der Texte (in blau) und der Wörter (in rot) auf den ersten beiden Dimensionen an. Darüber hinaus sehen wir unten und links im Bild den Beitrag der beiden Dimensionen zur Varianzaufklärung im gesamten Datensatz. Betrachten wir nun die erste Grafik, die ausgegeben wurde. Bitte beachten Sie, dass die Bedeutung für die Interpretation aus der statistisch ermittelten Distanz (Nähe und Abstände z. B. von Wörtern zueinander) abzulesen ist. Je näher die Wörter beieinander verortet sind, umso eher treten sie gemeinsam auf. Für Texte gilt, dass sie ein umso ähnlicheres Vokabular verwenden, je näher sie in der Grafik zueinander positioniert sind. Umgekehrt ist das genutzte Vokabular umso verschiedener, je größer die Distanz zwischen den Texten ist. Für Wörter gilt analog, dass diese systematisch nicht gemeinsam in den Texten auftreten.

Als weiteres Distanzmaß gilt, dass je näher die Wörter und Texte zum Koordinatenursprung sind (d. h. dort, wo sich die gestrichelten 0-Linien kreuzen), desto gleichmäßiger sind die Wörter über die Texte hinweg verteilt. Ferner gilt, dass je größer die Distanz der Wörter untereinander ist, desto besser sind diese dazu geeignet, um Unterschiede zwischen den Texten herauszuarbeiten. Texte wiederum, die nahe des Koordinatenursprungs verortet werden, nutzen (zumindest auf diesen Dimensionen) ein allgemeines, wenig ausdifferenzierendes Vokabular, das ungeeignet ist, um die Dimensionen aufzuspannen. Die überwiegend geringen Distanzen in Abbildung 9.7 sind beim vorliegenden Datenmaterial der Autoethnographien durch die Fokussierung der Fragestellung auf „Wie gestalten Sie Ihren Studienalltag zu Beginn des Online-Semesters 2020/21?“ zu erklären.

Für die Interpretationen beginnen wir nun bei der X-Achse, welche die Dimension 1 darstellt. Wir beginnen mit der ersten Dimension, weil sie die meiste, in den Daten vorhandene Varianz aufklärt und damit die größte Unterscheidung zwischen Textinhalten zulässt. Zudem ist die Dimension 1 in den meisten Fällen am einfachsten zu interpretieren. Betrachten wir die Wörter, die am weitesten rechts und links in Abbildung 9.7 vertreten sind, so sehen wir Wörter, die den Gegensatz der Dimension 1 verdeutlichen. Die Distanz der Wörter trägt damit zur Interpretation des Themas bzw. der beiden Themen bei, die auf dieser Dimension abgebildet sind. Links sehen wir Wörter wie „hoch“, „tutor“, „schritt“, „tablet“, „notieren“ oder auch „tutorium“, „oeffnen“ oder „sitzung“, während auf der rechten Seite „youtube“, „aufwachen“ „ausruhen“, „waschen“, „kaufen“ „anruf“ oder „abendessen“ stehen. Hierauf basierend, können wir von der Opposition zwischen sozialen Handlungen, die mit dem Studium zu tun haben, und Freizeitaktivitäten ausgehen. Um den Unterschied zu verdeutlichen, sollten wir zumindest einen Blick in die Texte Nummer 32 (ganz rechts) und 42 (ganz links) werfen (enge Interpretation).

Besser noch für die Erklärung der Unterschiede – im Sinne einer weiten Interpretation – wäre die Lektüre der Texte 12, 33 und 50 auf der rechten sowie 9,

17, 18, 38 auf der linken Seite in Abbildung 9.7. Bei der Auswahl der vertieft zu analysierenden Texte sollten Sie stets auf den Rahmen und die Fragestellung Ihrer Analyse achten. Die Lektüre und vertiefte Inhaltsanalyse würde folglich nicht nur den Rahmen dieses Kapitels sprengen, sondern vermutlich auch den einer Hausarbeit. Bei einer Bachelorarbeit, jedoch sicherlich bei einer Master- und Doktorarbeit, ist allerdings genügend Platz (z. B. sind oft in Studien- und Prüfungsordnungen Wörter- oder Seitenanzahl angegeben) diese Texte einer Inhaltsanalyse zu unterziehen, wie beispielsweise in Kapitel 5 (deduktiv-qualitative Inhaltsanalyse) in diesem Buch erklärt, oder sie für eine Untersuchung im Stil der Grounded Theory auszuwählen.[11]

Für die enge Interpretation finden wir in Idno. 32 Passagen, die auf Freizeittätigkeiten hindeuten, die neben den Studienaktivitäten durchgeführt werden.

> „Wenn ich vom Rathaus nach Hause komme, **gehe** ich zu dm, um **Waschmittel** für die **Waschmaschine** zu **kaufen**. Nach dem Einkaufen **gehe** ich nach Hause und **esse** gegen 12:30 Uhr zu **Mittag**. Ich **esse** Tiefkühlpizza und Chicken Nuggets, weil ich mir nicht die Mühe machen will, sie zu machen. Danach gehe ich nach unten, um meine Lebensmittel und Papierabfälle in den gemeinsamen Mülleimer zu werfen. Ich habe vor, um 16:00 Uhr mit einer **Freundin** von mir in [ORT] einen **Spaziergang** zu machen, also werde ich mich **entspannen** und bis dahin **YouTube** schauen" (Idno. 32, Hervorhebungen nicht im Original).

Dahingegen scheint die*der Verfasser*in der Autoethnographie Idno. 42 gedanklich ganz im Studium angekommen zu sein, wie folgender Absatz verdeutlicht.

> „An diesem Donnerstag erwarten mich neue **Vorlesungen**, neue **Professoren** und neue Themen. Ich stehe um 8:00 Uhr auf, um mich in Ruhe für den Tag fertig zu machen. Auf meinem Smartphone öffne ich die [**Universitätsname**] **App** und schaue nach meinem heutigen **Stundenplan**. Daraufhin hole ich mir ein Glas und eine Wasserflasche und setzte mich an meinen Schreibtisch. Dort starte ich mein **Tablet** und meinen **Laptop**. Auf dem **Laptop** öffne ich das Mailprogramm und schaue nach neuen E-Mails. Schließlich sind es 10:30 Uhr und mein **Tutorium** in [**Studienfach 4**] beginnt. Die **Tutorin** stellt sich vor und anschließend jeder von uns. Schnell merke ich, dass die **Tutorin** einen guten persönlichen Bezug zu uns aufbaut und uns das Gefühl gibt, dass wir alle in einem Boot sitzen. Ein Boot mit Höhen und Tiefen, aber ein Boot mit einem klaren Ziel. Mithilfe der **Tutorin** besprechen wir den ersten pädagogischen Text und machen uns eine **Zusammenfassung**. Meine **Zusammenfassung** erstelle ich

11 Bei Verwendung dieser Texte als Grundlage für ein theoretisches Sampling nach der Methode der Grounded Theory (Holton 2007) könnten wir annehmen, dass es sich hier um (proto-)typische Texte handelt, die gegenteilige Positionen in Hinblick auf ein untersuchtes Thema einnehmen und damit zur weiteren Theoriebildung beitragen können.

digital auf meinem Tablet. Hinterher stellt die Tutorin uns das Glossar vor. In diesem werden wir im Laufe des Semesters die erlernten Fachbegriffe notieren und erläutern. Nach diesem Tutorium startet um 12:30 Uhr meine erste [**Studienfach 9**] **Vorlesung**. Der **Professor** stellt uns den Verlaufsplan vor und beginnt mit der Vorlesung über die ersten Inhalte. Ich habe mir für das Studium einen Stift für mein **Tablet** gekauft. Mit diesem Stift notiere ich mir digital die Beispielrechnungen aus der Vorlesung. Nun ist es 14:00 Uhr und die Mathe Vorlesung ist zu Ende. Um 16:30 Uhr startet dann meine letzte Vorlesung für heute. Zwischenzeitlich bereite ich das **Tutorium** und die [**Studienfach 9**] **Vorlesung** von eben nach. Um 16:30 Uhr melde ich mich dann wieder bei **Zoom** an, um an der letzten Vorlesung für heute teilzunehmen. Die Vorlesung handelt von der Einführung in die Didaktik. Um 18:00 Uhr beende ich dann mit dem Abschluss der Vorlesung meinen **Studientag**. Abends bin ich noch mit den anderen aus meinem **Studiengang** verabredet. Durch Corona natürlich **virtuell per Zoom-Meeting**. Den Abend nutzen wir, um uns über die Woche auszutauschen" (Idno. 42, links verortet, Hervorhebungen nicht im Original).

Wie Sie sehen, benötigen Sie auch für automatisierte Verfahren der quantitativen Textanalyse die Fähigkeit, Texte qualitativ auszuwerten. Die beiden vorliegenden Textausschnitte stellen hier deutlich die aus Abbildung 9.7 hervorgehende *Opposition dar zwischen Handlungen, die zum Studium gehören, und Freizeitaktivitäten.* Auf diese Weise können wir die Themen der ersten Dimension auch mit Studien- (Kategorie 1) und Freizeitaktivitäten (Kategorie 2) verknüpfen.

Bei Dimension 2 wird die Kategorisierung etwas schwieriger. Hier ist keine so eindeutige Opposition von Wörtern zu erkennen wie in Dimension 1. Viele Begriffe sind nahe des Koordinatenursprunges, während andere wie „haushalt", „bad", „familie", „wach", „schule" „kind", „chance", „zeitpunkt" oder „kuemmern" recht typisch für diese Dimension scheinen. Auch „plenum" ist stark auf dieser Achse verortet, was eventuell ein Ausreißer sein kann, der durch die Verwendung in Text Nr. 30 hervorgerufen sein kann, welcher weit unten auf der Koordinatenachse (leicht verdeckt durch „schule") angesiedelt ist. Hier kann vermutet werden, dass Dimension 2 nur ein Thema abdeckt, das sich auf Familie und Kinder bezieht. Um zu kontrollieren, inwiefern Text Nr. 30 ein Ausreißer ist, der die Dimension 2 erst aufspannt, könnten Sie den Text zu einer passiven Beobachtung machen. Wenn die Dimension 2 dann trotzdem erhalten bleibt, dann ist das Thema „Kinder und Familie" stabil. Wenn aber Dimension 2 nach Ausschluss von Text Nr. 30 gänzlich anders strukturiert ist, dann handelt es sich bei Dimension 2 um ein statistisches Artefakt, das durch diesen Text erstellt wird. Unter einem statistisch erzeugten Artefakt ist die Verfälschung von statistischen Berechnungen durch fehlerhafte Datenerhebung oder Messung von Konzepten, wie vielleicht im vorliegenden Fall, zu verstehen. Konkret können Sie die Option `row.sup = 30` in Ihren `CA()`-Befehl eingeben. Der modifizierte Befehl würde im vorliegenden Falle `CA(df_reduced, ncp = 10, graph = FALSE,`

`row.sup = 30`) lauten. Wir behalten den Text aber aus Interpretations- und Illustrationsgründen in der Folge in der Analyse.

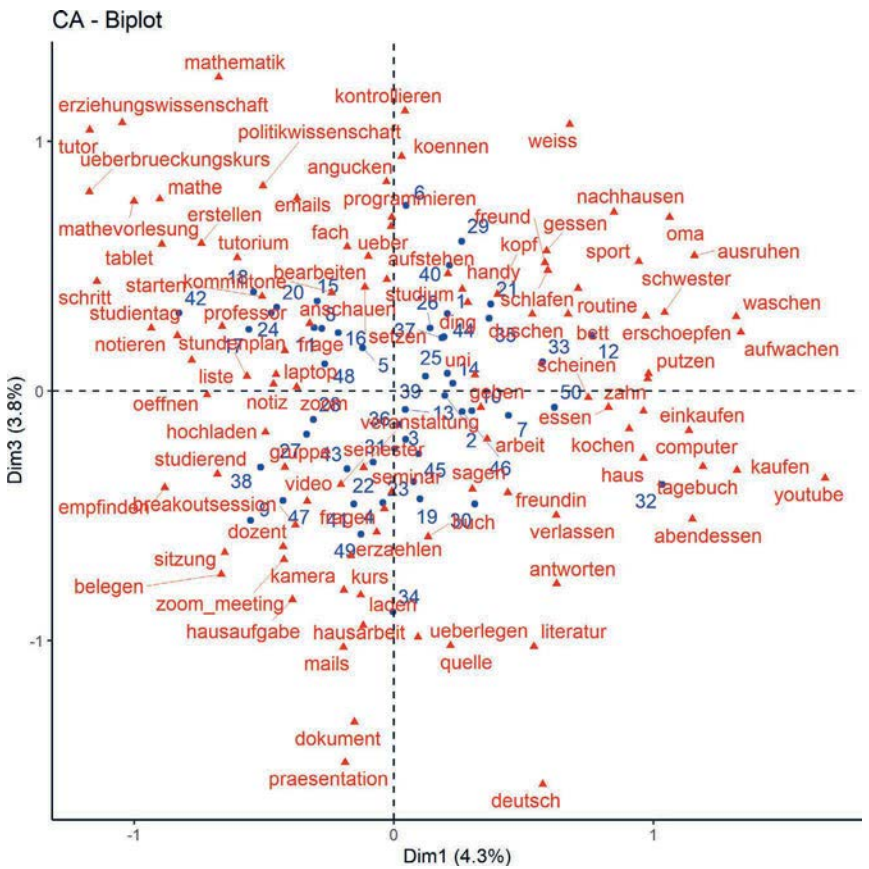
Wie in Kapitel 6.4.2 (teilautomatisierte induktiv-quantitative Inhaltsanalyse) erklärt wird, wird das Wort Kinder im Kontext des Pädagogikstudiums, Nervfaktoren (Klingelstreich) und auch als Teil der eigenen Familie in den Autoethnographien genannt. Um ganz genau zu klären, welche Themen die Dimension 2 konstruieren, sollten wir nun den Text mit Idno. 30 betrachten und interpretieren. Auch hier gilt, dass eine tiefgehende Untersuchung außerhalb der Möglichkeit des vorliegenden Kapitels steht. Deswegen gehen wir hier nur kursorisch auf einige Textpassagen ein und heben die Begriffe hervor, die charakteristisch für Dimension 2 sind.

> „Mein Tag beginnt stets um 6:30 Uhr, da ich [**Kind 1**] für die **Schule** vorbereite. Während er*sie ihre Zähne putzt, bereite ich ihm*ihr Frühstück vor. Nachdem er*sie im Bad fertig ist, mache ich ihm*ihr im Wohnzimmer die Haare, unserer Kultur gemäß. Bevor mein*e [Ehepartner*Ehepartnerin] ihn*sie dann letztendlich zur Schule fährt, beten wir für einen erfolgreichen **Schultag**. Zu diesem **Zeitpunkt** ist es dann schon ungefähr 7:20 Uhr und [**Kind 2 und Kind 3**] bereits wach. [**Kind 2**] ist ein **Kleinkind** und [**Kind 3**] ein **Baby**. Nun bereite ich auch für sie jeweils ihr **Frühstück** vor. In der Zeit, wo sie mit ihrem Frühstück beschäftigt sind, hole ich meinen Laptop, melde mich auf der Moodle-Plattform der [Universität] an, um zu schauen, was heute für mich ansteht. Der Stundenplan ist für mich noch sehr neu, weshalb ich ihn noch nicht richtig einprägen konnte. Laut dem Plan soll zunächst von 10:30 Uhr bis 12:00 Uhr die Vorlesung [Titel der Veranstaltung] stattfinden. Ich lese mich schonmal in die im Moodle-Raum zur Verfügung gestellten Texte ein. Nachdem [**Kind 2 und Kind 3**] fertiggegessen haben, spiele ich mit ihnen bis 9:30 Uhr, da ich verbleibende Stunde vor Vorlesungsbeginn dafür nutzen möchte, mich weiterhin in die Texte einzulesen. Um 10:30 Uhr fängt die Vorlesung an. Der Professor stellt sich uns erstmal vor und heißt uns herzlich Willkommen ein weiteres Online-Semester zu bestreiten. Er erläutert seine Expertise und macht eine kleine Einführung in das Themengebiet der Sozialforschung. Anschließend schickt er uns in Breakout-Sessions, damit wir Studierenden zum einen die Chance bekommen andere Studierende kennenzulernen und ebenfalls herauszufinden, ob uns in Anbetracht der Aufgabe dieses Tagebuch zu verfassen, noch Fragen offengeblieben sind. Die werden nach der Breakout-Session im Plenum geklärt. Nachdem die Vorlesung endet, schaue ich nach [**Kindern 2 und 3**] und [**Lebenspartner*Lebenspartnerin**], der*die während meiner Vorlesung auf sie aufpasste. Er*sie ist ziemlich erschöpft, da er*sie eine neue Arbeitsstelle hat, in der er*sie in Teilzeit Nachtschichten absolviert. Ich **kümmere** mich wieder um [**Kinder 2 und 3**], während mein*e [**Lebenspartner*Lebenspartnerin**] sich darauf vorbereitet [**Kind1**] von der **Schule abzuholen**. Als sie wiederkommen, erzählt [**Kind1**] mir von seinem*ihrem **Schulalltag**. Ich erzähle [**Kind1**] ebenfalls von meiner [Veranstaltung] und meinen Eindrücken, nämlich, dass ich es interessant fand und mich bereits auf

die Aufgabe freue das Uni-Tagebuch zu schreiben. Nach unserer Unterhaltung koche ich für meine **Familie** und mache ebenfalls den **Haushalt**. Es ist nun bereits 16:00 Uhr und somit Zeit für meine zweite und gleichzeitig letzte Vorlesung des Tages“ (Idno. 30, weit unten verortet, Hervorhebungen nicht im Original)

Wir scheinen hier eine*n engagierte*n Mutter*Vater und Studierende*n vor uns zu haben, die*der sowohl von ihren*seinen Eindrücken in der Universität als auch vom Alltag im Familienkontext schreibt. Dies erhärtet die weiter oben genannte Interpretation, sodass wir die Dimension 2 „Eigene Familie und Kinder“ nennen können. Bei der Verortung der genannten Begriffe fällt dabei auf, dass sie auf der X-Achse leicht nach rechts verteilt sind, was darauf hindeutet, dass Familientätigkeiten nahe an „Freizeittätigkeiten“ angesiedelt sind, die jenseits des Studienkon-

Abbildung 9.8 Visualisierung von Dimensionen 1 und 3 der Korrespondenzanalyse



textes angesprochen werden. Auch wenn diese häuslichen Arbeiten noch immer Arbeiten sind, so meint dies nichts anderes, als dass es hier eine Überschneidung mit dem Vokabular der ersten Dimension gibt und hier ganz speziell mit den Wörtern, die nun einmal von anderen im Freizeitkontext genannt werden.

9.5.3 Interpretation der Dimension 3

Wenden wir uns nun der dritten Dimension zu, die in Abbildung 9.8 gemeinsam mit der ersten Dimension zu sehen ist.

In Abbildung 9.8 wird ein Gegensatz zwischen Wörtern wie „praesentation" „deutsch", „dokument", „literatur", „quelle", „hausarbeit", „dokument", „hausaufgabe", „antworten", „kamera" und „zoom_meeting" (unten) und „mathematik", „kontrollieren", „überbruckungskurs," „programmieren", „weiss" und „erstellen" (oben) aufgespannt. Dies scheint eine Art Auffächerung der ersten Dimension – genauer: der Handlungen – zu sein, die im Studium durchgeführt werden. Erstere deuten darauf hin, dass die Personen Teilnehmer*innen von Seminaren sind und hierfür Dokumente, Literatur und Quellen lesen bzw. vorbereiten, an regen Diskussionen bei Zoom-Meetings teilnehmen, auf Fragen antworten, hierfür ihre Kamera aktivieren müssen und mit weiteren Studierenden Arbeitstreffen abhalten. Weiter oben scheinen Übungen abgehalten zu werden, was durch die Wörter „kontrollieren", „können", „weiss" (bzw. Wissen), aber auch durch die Nennung von Fächern wie [Studienfach 9] und [Studienfach 3] und [Studienfach 4] angedeutet wird. Zudem deutet „programmierung" auf ein übungsintensives Studium hin. Auch hier gilt, dass wir zumindest einen Blick in die Texte werfen sollten, die auf der Dimension 3 am weitesten voneinander entfernt sind, um die Vermutung zu prüfen, dass es sich hier um den Gegensatz zwischen Seminar und Übung handelt. Hierzu ziehen wir die Texte 29 (oben) und 34 (unten) exemplarisch heran.[12]

> „Der Wecker klingelt um 7:00 Uhr, aber aufgrund meines schlechten Schlafs und der Tatsache, dass ich es morgens doch etwas entspannter angehen lassen kann, stehe ich erst um halb acht auf und stärke mich durch ein Frühstück, das vielleicht kleiner als normal ausfällt, da ich durch die Aufregung zunächst nicht so viel runter bekomme wie sonst. Ich berichte meinem Vater am Frühstückstisch von den Tagesplänen der **Vorlesungen,** sodass dieser ungefähr im Bilde ist, wann ich für kleinere Erledigungen und Hilfe im Haushalt verfügbar bin. Die Zeit vor der ersten **Vorlesung** verbringe ich,

12 Doch auch hier gilt, dass Sie eigentlich mehrere Texte sichten sollten, die besonders hoch bzw. besonders niedrig auf den Achsen positioniert sind. Wie man diese Texte am ehesten herausfiltert bzw. anzeigen lässt oder in eine neue Datei schreibt, wird weiter unten nochmal explizit ausgeführt.

indem ich mich nochmals in das Thema einlese und mir die **Dateien** im **Moodle-Raum** angucke.

Als die **Vorlesung via Zoom** endlich losgeht bin ich doch erst ziemlich nervös, obwohl alles nur digital stattfindet und ich gar keinen echten Menschen sehe. Als der Start des Moduls technisch jedoch ein bisschen holprig startet, legt sich meine Aufregung etwas und der **Rest der Vorlesung läuft entspannter ab.** In einer größeren Mittagspause berichte ich meiner Familie verbal und meiner Freundin digital, wie es ablief. Als nächstes findet für mich ein **Propädeutikum** statt, welches für mich ebenfalls das allererste ist. Die **Gruppe** ist etwas kleiner und ich beteilige mich mehr daran als in der größeren **Vorlesungsrunde.** […]

Nach einer Mittagspause, in der ich mich auch wieder mit meinen Eltern über das **Studium** unterhalte, arbeite ich noch einige Zeit weiter an verschiedenen Dingen für das **Studium** und höre erst damit auf, als noch eine **weitere Vorlesung** beginnt. Diese ist wieder sehr kurzweilig und es macht Spaß zuzuhören. Danach bin ich mit meinen Freunden verabredet. Ich halte mich allerdings über die **Messenger** darüber auf dem Laufenden, wie weit meine **Kommilitonen** mit ihrer **Lerngruppe** gekommen sind, mit der sie sich getroffen haben. Es ärgert mich ein bisschen, dass ich nicht dabei bin und ich nehme mir vor, nächstes Mal teilzunehmen" (Idno. 29, oben verortet, Hervorhebungen nicht im Original).

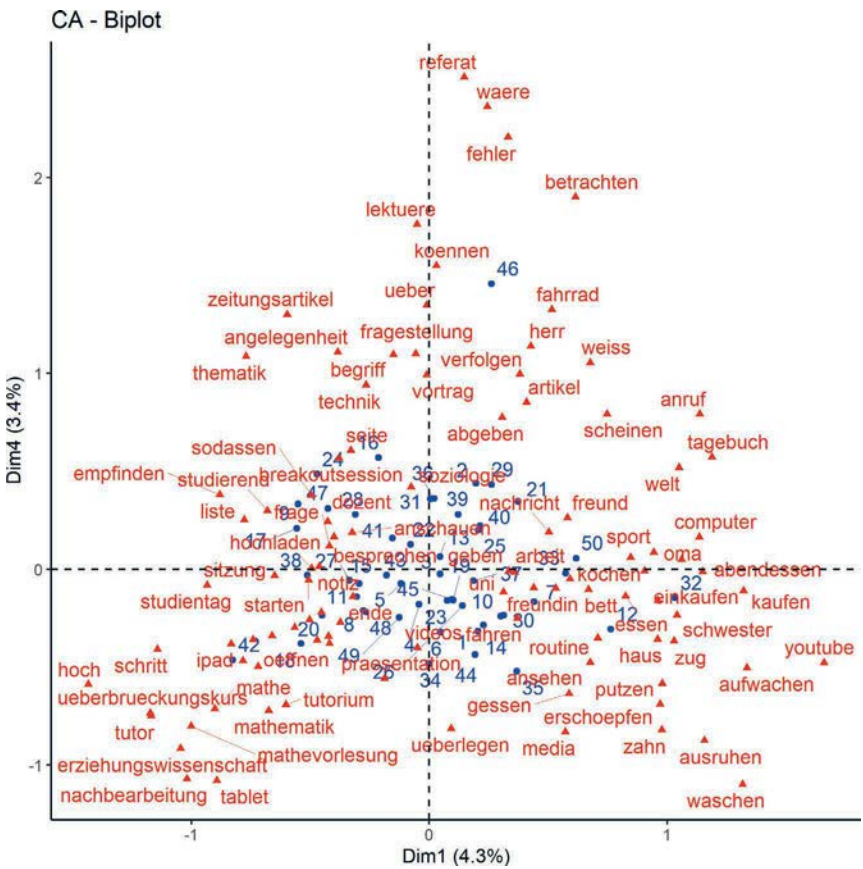
Im Textausschnitt wird bereits deutlich, dass der erste Deutungsvorschlag die Handlungen nicht in Gänze erfasst, die oben auf der Dimension 3 verortet sind. Anders als gedacht, scheint die*der Verfasser*in der Autoethnographie Nr. 29 nicht bloß von Gruppenarbeiten, sondern auch von Vorlesungen zu berichten. Blicken wir nun in Transkript Nr. 34, um zu prüfen, ob wenigstens die erste Interpretation der unteren Bereiche von Dimension 3 der qualitativen Überprüfung standhält.

„Ich stehe um 10:00 Uhr auf. Mir ist aufgefallen, dass wir für die **Präsentation** nicht wirklich ein **Feedback** bekommen haben. Eine Freundin schlägt mir vor, dass wir den Dozenten danach fragen können. Die anderen haben es nicht mehr erwähnt, daher nehme ich an, dass es nicht so wichtig ist. Ich muss für das [Studienfach 8] **Proseminar** und das […] **Proseminar** jeweils eine **Hausarbeit** schreiben. Ich überlege mir **Themen** für die **Hausarbeit**. Ich mache mir Sorgen über die **Literatur-Beschaffung**. Ich weiß nicht, wie es in der [**Fach**] **Bibliothek** ist, aber in der [**Fach**] **Bibliothek** darf man zehn Minuten für **Bücher** suchen. Man soll die **Bücher** so schnell wie möglich zurückgeben, sobald man fertig ist. Ich stelle mir das schwer und anstrengend vor. Von meinen bisherigen Erfahrungen weiß ich, dass ich immer eine ganze Liste mit Titeln raussuchen musste und sie dann alle mindestens mir kurz anschauen musste, um zu sehen, ob es brauchbar war für mich. Die dauert länger als zehn Minuten und was ist, wenn ich mehr als fünf **Bücher** finde. Ich fahre mit dem Zug. Ich möchte auch nicht zehn **Bücher** mit mir rumschleppen oder dreimal hin und zurück nach [Stadt] fahren. Letztes Semester habe ich nur **Artikel** benutzt, die ich online finden konnte. In der **Be-**

wertung hat mir die **Dozentin** gesagt, dass es nicht genug **Quellen** waren. Am liebsten möchte ich auch dieses Semester **Artikel online** finden. Ich brauche mehr Zeit, um das Fach zu verstehen“ (Idno. 34, unten verortet, Hervorhebungen nicht im Original).

In diesem Text deutet sich an, dass es sich tatsächlich um Seminare, allerdings aus Studienfach 8 handelt, die für den unteren Bereich (d. h. negative Werte bei der Dimension 3) charakteristisch sind. Es scheint zudem eine implizite Überlagerung zwischen [Studienfach 9] und [Studienfach 6] versus [Studienfach 8] auf dieser Dimension vorzuliegen. Entsprechend muss auch in diesem Falle die Interpretation weiter angepasst werden. Bei der Dimension 3 sehen wir somit eine Opposition zwischen *Vorlesungen und Gruppenarbeiten in den Studienfächern 6 und 9 versus Seminare in Studienfach 8.*

Abbildung 9.9 Visualisierung der Dimensionen 1 und 4 der Korrespondenzanalyse



9.5.4 Interpretation der Dimension 4

Kommen wir nun zur Dimension 4 (Abbildung 9.9) und damit zur Dimension, bei der der erste Ellbow im Screeplot verortet ist. Hier sehen wir keine eindeutige Opposition, sondern die Begriffe sind unten (negativer Bereich) insgesamt sehr weit aufgefächert. Oben rechts sehen wir Begriffe wie „referat", „fehler", „lektüre", „können", „fragestellung" „thematik", „zeitungsartikel" „artikel", aber auch „fahrrad", „ueber", oder „herr". Auf der einen Seite scheint dies auf die Vorbereitung von Referaten hinzudeuten, aber auch auf etwas, was Studierende ohne Digitaltechnik, wahrscheinlich analog, machen. Folgen wir der Auffächerung unten in Abbildung 9.9, dann sehen wir einerseits Wörter wie „nachbereitung", „tablet", „tutor", [Studienfach 4], [Studienfach 9], „tutor", „tutorium" oder „überbrückungskurs", die unten links verortet sind. Unten rechts sind Wörter wie „waschen", „ausruhen", „media", „erschöpfen", „ueberlegen", „putzen" „zug", „youtube".

Hier könnte somit eine Auffächerung der ersten Dimension stattfinden. Sie müssen sich an der Stelle fragen, was die Gemeinsamkeit dieser aufgefächerten Begriffe ist. Ist es die Nutzung von Medien? Ist es ein bestimmter Zweck oder ist es ein Ziel? Aufschluss geben können auch hier wieder bestimmte Texte, die als typisch für die aufgefächerten Bereiche der statistischen Verteilung angesehen werden können. Für den unteren Bereich schauen wir daher mindestens in die Texte 42 und 35 sowie weiter oben in Text 46. Wenn Sie sich unsicher sind, ob die Dimension 4 in der gegebenen Form ebenfalls Bestand hat oder ein Datenartefakt darstellt, dann empfiehlt es sich auch hier, Text Nr. 46 passiv zu setzen. Danach schauen Sie nochmals, ob sich die Dimension 4 signifikant verändert hat und die Gegensätze in gleicher oder ähnlicher Form auftreten.

Auch hier gilt, dass Textausschnitte zur Illustration herangezogen werden, aber an der Stelle kein Anspruch erhoben wird, eine vollständige und tiefgreifende, qualitative Analyse vorzulegen. Beginnen wir mit Text 46. Hier lesen wir:

> „Danach wieder an die **Referate** und zuerst das deutsche **Referat** einmal vortragen und die **Auffälligkeiten**, die einen **flüssigen Vortrag** behindern, wegarbeiten. Danach das [Studienfach 8] **Referat** ebenso verbessern. Um 9:30 Uhr dann zur Krankengymnastik, 11:00 Uhr zwei wichtige Anrufe tätigen. Dann habe ich mir ein Mittagessen verdient.
>
> 6:45 Uhr Start mit dem ersten **Referat**: So ein Durchlauf bringt immer etwas. Hier fallen mir vier zu **ergänzende** bzw. zu **korrigierende Folien** auf. Ich glaube, der **Vortrag** ist jetzt rund. Ich muss ihn nur noch etwas in **Form** bringen, aber **inhaltlich** ist er **komplett** und das Zeitlimit wird auch eingehalten" (Idno. 46, oben verortet, Hervorhebungen nicht im Original).

Allein an dieser Passage erkennen wir, dass die*der Verfasser*in dieses autoethnographischen Textes den Fokus sehr auf die Vorbereitung des Referates gelegt hat. Zudem scheint diese*r zugleich viele Fahrradausflüge zu unternehmen („Gegen

14:30 Uhr setze ich mich für eine Stunde auf das Fahrrad"), was in der Intensität und Häufigkeit der Nennung bei anderen Studierenden nicht vorkommt. Gleiches gilt für die Wahrnehmung von Zeitungsartikeln. Insgesamt scheint dieses Ende der Dimension 4 durch eine Überlagerung von Referatsvorbereitungen und analogen Tätigkeiten gekennzeichnet zu sein.

„Heute ist Freitag und somit der letzte Tag meiner ersten Uni-Woche. Ich beginne mit einem Blick in den **Stundenplan** auf meiner [**Universitätsname**] **App**. Da die [Übung] [**Studienfach 9**] erst nächste Woche startet, habe ich heute ausnahmsweise einen weiteren Vorlesungsfreien Tag. Um mich zu organisieren, schreibe ich mir wieder eine Liste mit allen Sachen, die ich noch erledigen muss. Hierzu setzte ich mich mit einem Glas Wasser an den Schreibtisch. Ich fange damit um 8:30 Uhr an. Mein erster Punkt auf der Liste ist die **Anmeldung für den** [**Studienfach 9**] **Überbrückungskurs**. Von dem **Kurs** erhoffe ich mir die Aufarbeitung einiger **Themen** und damit eine Erleichterung in den [**Studienfach 9**] **Vorlesungen**. Nach der erfolgten Anmeldung, arbeite ich dann das [**Studienfach 9**] **Übungsblatt** auf dem **Tablet** durch. Mir fällt es nun sehr leicht, die digitalen Notizen zu nutzen und nur mit meinem **Tablet** zu arbeiten. Ich merke, dass es für mich einfacher ist, weil ich dadurch immer alles geordnet habe und die **Lehrmaterialien** immer bei mir führe. Dann öffne ich meine **E-Mails** und schaue nach, ob es neue wichtige Informationen gibt. Anschließend lese ich mir die Literatur für die nächste [Studienfach 1] Vorlesung durch und schaue mir das **Video** auf Moodle an. Die **Vorlesung** findet immer [Wochentag]vormittag statt. In diesem Modul schreibe ich eine Autoethnographie. Dazu habe ich mir zwischendurch immer wieder digitale Notizen gemacht, die ich nun zu einem Fließtext zusammenführe" (Transkript Nr. 42, unten links verortet, Hervorhebungen nicht im Original).

„Bei der **Vorlesung**, handelte es sich um das Fach [**Studienfach 10**], mit dem **Schwerpunkt, Texte zu verstehen**. Es wurde das erste Thema des Faches vorgestellt, wo es darum ging Alte Texte zu verstehen, anhand von drei Leitfragen. Diese Vorlesung war dann auch pünktlich um 14:00 Uhr zu Ende. Danach habe ich mich noch per E-Mail für die Übung des Faches anmelden können.

Infolgedessen musste ich meinem **Vater dabei helfen**, ein **Zimmer zu laminieren**, da wir dabei sind, dieses Zimmer zu renovieren. Dies hat aber sehr gut funktioniert und ich war um circa 16:30 Uhr fertig. Nachdem dies geschafft war, habe ich mich durch **Social Media** auf den neusten Stand gebracht, mit besonderem Fokus auf die Wahlen in den USA.

Hinterher habe ich mich an meinen Schreibtisch gesetzt, um den **Text für den dritten Tag aufzuschreiben**. Bevor ich damit angefangen, habe ich wie immer den letzten Tag Revue passieren lassen und mir das in Form von Stichpunkten notiert, damit ich einen Leitfaden habe und in richtiger Reihenfolge aufschreibe. Damit war ich dann auch ungefähr um 19:00 Uhr fertig.

Ich habe mir dann noch die **Fußballspiele angeschaut** und habe nebenbei wieder

in **Social Media gesurft** und mir so die **Zeit vertrieben**, da ich nicht mehr wirklich was wichtiges zu tun hatte. Als die Spiele vorbei waren, bin ich dann runter gegangen und es **wurde besprochen, was zum Abend gegessen** wird. Als wir uns alle einig waren, haben wir das **Essen bestellt und abgeholt**. Beim Essen haben wir uns dann gemeinsam über den Tag geredet, was wir jeweils so gemacht haben. Mit dem Essen waren wir dann um 22:00 Uhr fertig.

Da ich noch vor hatte, an diesem Tag mein **Homeworkout** zu machen, habe ich mich nachdem Essen noch etwas **ausgeruht** und alles **verdaut**. Dann habe ich um 23:00 Uhr noch mit dem Sport angefangen und habe dann die Übungen gemacht, die ich mir vorher überlegt hatte. Nachdem ich mit dem Sport fertig, bin ich noch Duschen gegangen und mir die Zähne geputzt, um im Anschluss mich ins Bett zu legen. Ich habe mir dann wie immer noch einige **Videos auf YouTube angesehen** und bin währenddessen circa um 1:30 Uhr eingeschlafen" (Idno. 35, unten rechts verortet, Hervorhebungen nicht im Original).

Bei beiden Texten werden Vorlesungen und Übungen, d. h. Formen der Vermittlung von Wissen, angegeben, die durch Dozierende vorbereitet wurden. Dabei überwiegt der Anteil der Wörter, die mit Handlungen mit Studienbezug assoziiert sind, im Falle des Transkripts Nr. 42, während bei Nr. 35 auch ein Fokus auf Freizeittätigkeiten gelegt wird. Zwar sind auch hier aktive Anteile vorhanden, aber die Wörter, die auf passive Tätigkeiten hindeuten, sind in beiden Texten stärker präsent. Insgesamt scheint es auch hier eine Überlagerung aus jeweils zwei Themen zu geben, bei dem die Achse vom aktiven Auseinandersetzen mit Themen auf der einen Seite unter Verbindung mit analogen Aktivitäten (oben) hin zu der Anwendung digitaler Praktiken und einem Vokabular verläuft, das auf passive Tätigkeiten hindeutet. Tabelle 9.2 bietet eine Übersicht über die Interpretationen und durch die Dimensionen aufgeklärte Varianz.

Tabelle 9.2 Interpretationen der ersten vier Dimensionen der Korrespondenzanalyse und dazugehörige, aufgeklärte Varianz Ihres Datensatzes

Dimension	Interpretation	% aufgeklärte Varianz
1	Universitäre Tätigkeiten versus Freizeittätigkeiten	4,3 %
2	Eigene Familie und Kinder	4,1 %
3	Mathematik/Sozialwissenschaftliche Vorlesungen und Gruppenarbeiten versus Geistes- und Kulturwissenschaftliche Seminare	3,8 %
4	Aktives Auseinandersetzen mit Themen und analoge Aktivität versus digitale Praktiken/passive Tätigkeiten.	3,4 %
		15,6 %

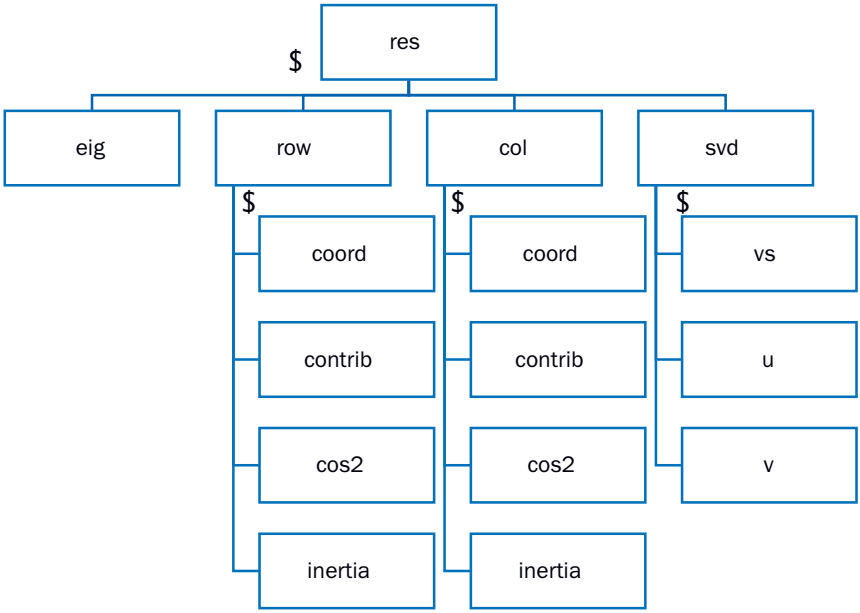
Denken Sie aber bei der Interpretation empirischer Materialien bitte daran, dass Sie idealerweise mehr als eine Interpretation entwickeln, und Ihre Interpretation nicht als einzig mögliche Interpretation(en) ansehen. Bilden Sie Teams mit Ihren Kommiliton*innen und interpretieren Sie die Ergebnisse gemeinsam und reflektieren Sie bei Ihrer Interpretation zugleich Ihre eigene Haltung und Erfahrungen, aus denen heraus Sie die Ergebnisse interpretiert haben. Um es mit Bourdieu (2012, S. 191) zu sagen: Ohne selbstreflexiv zu sein, laufen Sie Gefahr, durch Ihre Rolle als Forscher*in Ihre Position in dem von Ihnen untersuchten Raum und Ihr Verhältnis zu diesem auszublenden. Aus dieser Position heraus konstruieren Sie die Ergebnisse und lassen dabei – unbewusst und nicht böswillig – Ihre Sichtweise zu stark einfließen. Selbstverständlich können Sie nicht nicht Sie selbst sein, denn auch davor, nämlich bei der Berechnung dieses Raumes und der vorangestellten Datenakquise fließt Ihre Sichtweise, Erkenntnisinteresse, ja Position im sozialen Raum ein. Daher sind die Ergebnisse, wie Sie auch hier präsentiert wurden, niemals so objektiv, wie die zahlenmäßig-statistische Methode suggeriert. Wenn Sie diesen Umstand bei Ihrer Untersuchung mitdenken und immer wieder Dritte miteinbinden, können Sie der Gefahr entgegenwirken, Ihre Ergebnisse als objektiv richtig und gar „natürlich“ trotz Ihrer Konstruktionsleistung anzusehen.

9.5.5 Erkunden und Exportieren der durch die Korrespondenzanalyse erzeugten Informationen

Die Informationen der Wörter und Texte, auf deren Basis die Grafiken erstellt wurden, können von Ihnen nochmals separat aufgerufen und in andere Programme wie zum Beispiel Excel exportiert werden. Da wir diese Informationen im Objekt `res` gespeichert haben, das durch den `CA()`-Befehl erzeugt wurde, erkunden wir nun gemeinsam dessen Aufbau und Inhalt. Dieses Objekt hat eine Baumstruktur, in der alle Informationen zu den Ergebnissen Ihrer Analyse gespeichert sind. Da dieser Baum hierarchisch aufgebaut ist, sprechen wir in der Folge von den Ebenen dieses Baumes, auf denen die Informationen gespeichert sind. Die nächstniedrigere Ebene können Sie mit dem Dollar-Zeichen $ aufrufen. Das gilt nicht nur für die Ergebnisse der Korrespondenzanalyse, sondern Sie können beispielsweise einzelne Variablen in Ihrem Datensatz aufrufen (z. B. `df$vortag` wenn Sie wissen wollen, wer wie oft das Wort „Vortrag“ im Text verwendet hat). Das von uns betitelte Objekt res hat folgende Struktur (Abbildung 9.10, nächste Seite), auf deren wichtigste Bestandteile in der Folge eingegangen wird.

Mit `res$call` können Sie einsehen, auf welcher Basis der `CA()`-Befehl die Ergebnisse berechnet hat (z. B. zeigt Ihnen `res$call$ncp` die Anzahl der Dimensionen an, für die Ihnen R die Koordinaten für die Texte und Wörter berechnet). Die Informationen auf der Ebene von svd zeigen Ihnen die Zwischenschritte bei der Berechnung der Korrespondenzanalyse an, auf deren Basis Koordinaten,

Abbildung 9.10 Aufbau der Ausgabe einer Korrespondenzanalyse in R



Beträge zur Varianzaufklärung und Abdeckung der Wörter und Texte auf den jeweiligen Dimensionen berechnet wurden.[13]

Weitaus interessanter sind die `row`- und `col`-Objekte, die in `res` gespeichert sind. Über `row` können die Zeilenobjekte aufgerufen werden – das sind in unserem Falle die Texte selbst. Jeder Text hat einen speziellen Ort innerhalb des Raumes, der durch die Korrespondenzanalyse aufgespannt wird, trägt zur Erklärung der Varianz in den jeweiligen Dimensionen bei und kann mehr oder minder gut auf dieser Dimension verortet werden. Den Ort selbst können Sie mittels `coord`, den Beitrag zur Varianzaufklärung mit `contrib`, und die Abdeckung der jeweiligen Dimension mit `cos2` ansteuern. Gleiches gilt für Texte, die Sie mit `$row` ansteuern. Diese Informationen können Sie auch in einen separaten Datensatz überführen und als Excel-Datei oder in Tabellenform exportieren. Das kann Ihnen dabei behilflich sein, Ihre Datendokumentation und technischen Anhang für eine Haus- oder Abschlussarbeit zu erstellen und Ihr Vorgehen transparent zu machen.

13 Das verwendete mathematische Verfahren ist die Singulärwertzerlegung, deren englische Übersetzung *singular value decomposition* lautet. Daher verwendet R eine Abkürzung des englischen Begriffs für diese Verfahren.

Nun erstellen wir zu Illustrationszwecken einen Datensatz, der die Koordinaten und den Beitrag zur Erklärung der Varianzen der Wörter über die Dimensionen der Korrespondenzanalyse hinweg enthält. Hierfür beginnen wir mit der Erstellung eines Datentabellen-Objektes (Englisch: *data frame*) in R, in dem die Koordinaten der Wörter auf den einzelnen Dimensionen gespeichert sind (Code 9.17).

Code 9.17 Erstellung von Datensätzen, die Positionen und Varianzaufklärung der Wörter pro Dimension enthalten

```
# Erstellung eines Datensatzes, der die Position Wörtern auf Dimensionen
beinhaltet_____#####

koordinaten <- as.data.frame(res$col$coord)
beitrag <- as.data.frame(res$col$contrib)
```

Sie sehen, dass wir ein Objekt erstellt haben, das wir `koordinaten` genannt haben. Diesem haben wir eine Datentabelle mit den Koordinaten überwiesen, indem wir zunächst R erklärt haben, dass wir eine Datenmatrix generieren möchten. Das gleiche haben wir mit den Beiträgen der Wörter zur Aufklärung der Varianz auf den Dimensionen in der folgenden Zeile gemacht. Hierfür haben wir den `as.data.frame()`-Befehl genutzt. In den Klammern dieses Befehls haben wir die Informationen angegeben, die wir zu einer Datenmatrix umwandeln möchten. Diese sind `res$col$coord`, also die Koordinaten der Wörter, die in dem Objekt enthalten sind, die durch die Korrespondenzanalyse erzeugt wurden. Analog wurde der Beitrag zur Varianzaufklärung durch `res$col$contrib` aufgerufen und durch den Befehl transformiert.

Ein Problem müssen wir lösen, bevor wir die beiden Datenmatrizen miteinander kombinieren können. Wenn Sie mit `colnames(koordinaten)` und `colnames(beitrag)` die Variablennamen aufrufen, dann bemerken Sie, dass beide Datenmatrizen die gleichen Spaltennamen haben! Sie heißen in beiden Fällen `Dim 1, Dim 2` usw. Daher müssen wir sie in beiden Fällen umbenennen, um beide Datensätze zu kombinieren. Damit Sie in RStudio später die Variablennamen ohne Probleme ansteuern können, empfiehlt es sich daher, zunächst die Leerzeichen auf einmal zu entfernen und dann Zeichenfolgen zu ergänzen, mit denen Sie sich und den Lesern Ihres Kodes und Ihrer Arbeiten klar zu erkennen geben, um welche Variablen es sich handelt (Code 9.18). Daher wird in der Folge `koordinaten_` zu Beginn der Variablen im Falle der Datenmatrix ergänzt, in der die Koordinaten gespeichert sind. Analog hierzu wollen wir `beitrag_` am Beginn der Variablen in der Datenmatrix für die Beiträge zur Varianzerklärung ergänzen. Dies können Sie wie folgt erreichen.

Code 9.18 Befehle zur Veränderung von Spaltennamen im Datensatz am Beispiel der Koordinaten von Wörtern auf Dimensionen

```
colnames(koordinaten) <- gsub(" ", "_", colnames(koordinaten))
koordinaten <- koordinaten %>% rename_all(~paste0("koordinaten_", .x))
```

Wie weiter oben angegeben, ruft der Befehl `colnames()` die Variablennamen Ihres Datensatzes auf. In unserem Falle sind dies die Dimensionsnamen. Der `gsub()`-Befehl auf der rechten Seite wird verwendet, um eine Zeichenfolge zu ersetzen. Das `" ", "_"` zeigt an, dass alle Leerzeichen durch einen Unterstrich ersetzt werden sollen. Das darauf folgende `colnames()` sagt, dass diese Ersetzung von Zeichen für die Variablennamen einer Datenmatrix oder eines Datensatzes durchgeführt werden sollen. In der darauffolgenden Zeile geben wir durch den `rename_all()`-Befehl an, dass wir alle Variablennamen ersetzen wollen. Die Tilde ~ und das `.x` zeigen R an, dass hier alle Variablennamen angesteuert werden sollen. Der `paste0()`-Befehl sagt R wiederum, dass eine Zeichenfolge miteinander kombiniert werden soll. Da wir `"koordinaten_"` an die erste Stelle gesetzt haben, wird diese Zeichenfolge automatisch an den Beginn der Variable gesetzt. Würden wir dies mit `.x` vertauschen, dann würde diese Zeichenfolge nach den bisherigen Variablennamen (d. h. Dim_1 bis Dim_10) angehängt werden. Wenn Sie nun das Objekt mit dem Namen `koordinaten` ausführen oder zum Beispiel `view(koordinaten)` oder auch `colnames(koordinaten)` eingeben, dann sehen Sie, dass Sie die Dimensionsnamen erfolgreich geändert haben. Insgesamt (und leicht neu angeordnet) lautet die Syntax, mit der Sie die beiden Teildatensätze erstellen und die Variablennamen verändern wie in Code 9.19.

Code 9.19 Zusammengeführter Code zur Erstellung der Datensätze und zur Veränderung der Variablennamen

```
# Erstellung eines Datensatzes, der die Position Wörtern auf Dimensionen
beinhaltet_____#####
koordinaten <- as.data.frame(res$col$coord)
colnames(koordinaten) <- gsub(" ", "_", colnames(koordinaten))
koordinaten <- koordinaten %>% rename_all(~paste0("koordinaten_", .x))

# Erstellung eines Datensatzes, der den Beitrag von Wörtern zu Dimensionen
beinhaltet_____#####
beitrag <- as.data.frame(res$col$contrib)
colnames(beitrag) <- gsub(" ", "_", colnames(beitrag))
beitrag <- beitrag %>% rename_all(~paste0("beitrag_", .x))
```

Um die beiden Datensätze zusammenzuführen, können Sie, wie Sie in Code 9.20 sehen, den `cbind()`-Befehl benutzen. Das heißt, dass zwei oder mehr Datensätze zusammengeführt werden, die die gleichen Zeilennamen haben. Dies ist bei uns möglich, weil wir die Wörter, die in den Texten vorliegen, in beiden Fällen als Zeilennamen durch `as.data.frame()` definiert haben. Wenn wir die beiden Datensätze beispielsweise in ein neues Objekt namens `woerter_df` speichern wollen, dann müssen wir die folgende Zeile ausführen.

Code 9.20 Aneinanderbinden beider Datensätze mittels des cbind()-Befehls

```
## Koordinaten und Beiträge zusammenführen
woerter_df <- cbind(koordinaten, beitrag)
```

Darüber hinaus können Sie die Daten neu arrangieren, die Sie dem neuen Datensatzobjekt übergeben haben (Code 9.21). Sie können die Wörter beispielsweise absteigend nach deren Beitrag zur Varianzaufklärung in der ersten Dimension anordnen. Hierfür können Sie auf die `arrange()`- und `desc()`-Befehle zurückgreifen.

Code 9.21 Sortieren des Datensatzes anhand der Beiträge zur Varianzaufklärung von Wörtern auf Thema `1`

```
## Datensatz ordnen. Regel: Absteigender Beitrag der Wörter zur Erklärung
## von Dimension 1
woerter_df <- woerter_df %>% arrange(desc(beitrag_Dim_1))
```

Wenn Sie diese Zeile betrachten, dann sehen Sie zwei Dinge. Erstens, dass wir eine Verschachtelung der beiden Befehle auf der rechten Seite des Befehls haben. Die Verschachtelung von `arrange(desc())` bewirkt, dass ein Datensatz oder ein Objekt, das hierdurch angesteuert wird, in einer absteigenden Reihenfolge angeordnet werden soll. Zweitens haben Sie sicherlich die Zeichenfolge %>% bemerkt, die von `tidyverse` verwendet wird und R anzeigt, dass etwas innerhalb eines Datensatzes angesteuert werden soll. Konkret sagen wir dem Programm, dass wir innerhalb des Objektes `woerter_df` eine Sortierung nach den Werten vornehmen sollen, die in der Spalte `beitrag_Dim_1` gespeichert sind, und dass diese Sortierung nach absteigenden Werten geschieht. Das auf die Weise sortierte Objekt wird wieder `woerter_df` mit dem Pfeil <- zugewiesen.

Zuletzt können Sie die Teildatensätze, die Sie nun nicht mehr benötigen, mit dem `rm()`-Befehl (als Abkürzung für Englisch: *remove*) und Ihren Datensatz im Tabellenformat mit `write.csv()` abspeichern (Code 9.22). Dabei müssen sie zu-

erst den Datensatz angeben, den Sie speichern wollen und dann in Anführungszeichen den Namen der csv-Datei, die gespeichert werden soll.

Code 9.22 Entfernen überflüssiger Objekte und Speichern des Datensatzes

```
## entfernen überflüssiger Variablen aus dem Environment
rm(koordinaten)
rm(beitrag)

## Datensatz auf dem Desktop abspeichern

setwd("C:/Users/[Speicherpfad]") # [Speicherpfad] durch einen Pfad Ihrer Wahl
ersetzen
write.csv(woerter_df, "Koordinaten_und_Erklaerungspotential_von_Woertern_
CA.csv")
```

9.6 Schlussworte

In dem vorliegenden Kapitel haben wir Ihnen gezeigt, wie Sie R und RStudio installieren, wie Sie mit letzterem umgehen, Pakete installieren und laden können. Wir haben Ihnen auch gezeigt, welche Schritte nötig sind, um die Daten für eine Korrespondenzanalyse aufzubereiten, wie Sie diese visualisieren und die Abbildungen und statistischen Ausgaben zu interpretieren sind. Sie haben gesehen, wie Sie Themen mithilfe der Korrespondenzanalyse und den daraus folgenden Gegensätzen zwischen Wörtern und Texten innerhalb Ihres erzeugten Raumes interpretieren können, ohne die Texte zuvor eingehend gelesen haben zu müssen. Zuletzt haben wir Ihnen demonstriert, wie Sie die Ergebnisse in eine Tabellendatei (csv) abspeichern können. Allerdings haben Sie auch bemerkt, dass die Analyseform durch qualitative Interpretationen eines Teiles des Textkorpus ergänzt werden sollte.

Wir sind dabei auf der reinen Textebene verblieben und haben keine zusätzlichen Informationen wie zum Beispiel die soziale Herkunft, Alter oder Geschlecht der Studierenden genutzt, um die Themen an Personen oder soziale Gruppen rückzubinden. Sie können und sollten solche Zusatzinformationen als passive Variablen in Ihren Analysen verwenden, sofern Sie eine Vermutung haben, mit welchen Lebenswelten die Themen in den von Ihnen analysierten Kommunikationen assoziiert sind und in welchen Erfahrungen diese Themen verankert sind. Auf diese Weise können Sie Aussagen aus der Forschung zu sozialer Ungleichheit, eventuell auch der Sozialisationsforschung, Armuts- oder Elitenforschung in Ihre Analysen miteinbeziehen und damit über den Text hinausgehen. Ferner können

Sie, sofern Sie die Analyse Ihres Textes mit Daten aus Fragebögen kombinieren, auch fehlende Aussagen als passive Variablen in Ihre Textanalyse einbeziehen. Wir können nämlich vermuten, dass die Bereitschaft, Fragen zu beantworten oder auf bestimmte Themen einzugehen, von der sozialen Herkunft und der Relation zwischen Interviewer*in (also Ihnen und Ihrer sozialen Herkunft und Erfahrungen) und Interviewpartner*in abhängen. Das ist eine Konsequenz der eingangs vorgestellten, von Bourdieu maßgeblich entwickelten Herangehensweise an soziale Phänomene, die wir hier auf Texte bezogen haben.

10. Sentiment-Analyse als induktiv-quantitative Inhaltsanalyse

In diesem Kapitel bieten wir Ihnen eine Einführung in die Sentiment-Analyse in R und Python. Hierfür gehen wir zunächst auf die Grundlagen der Sentiment-Analyse ein und bieten Ihnen einen historischen Abriss über deren Entwicklung. Danach zeigen wir Ihnen Aufbereitungsschritte und Durchführung in R und stellen dabei die benötigten Pakete vor. Anschließend fahren wir mit der Aufbereitung für die und die Durchführung der Sentiment-Analyse in Python fort. Zuletzt zeigen wir Ihnen, wie Sie die Sentiment-Werte für unterschiedliche Gruppen vergleichen können. In dem vorliegenden Kapitel werden circa zwölf Seiten für die Vorführung von Code in R und Python verwendet.

10.1 Einleitung

Bei der Sentiment-Analyse handelt es sich um ein Verfahren des überwachten maschinellen Lernens, das eine automatisierte Form der deduktiven quantitativen Textanalyse darstellt. Zur Erinnerung: Die induktiv-geleitete qualitative Inhaltsanalyse bezeichnet in der empirischen Sozialforschung eine besonders stark auf das Text- und gegebenenfalls dazugehörige Datenmaterial (z. B. Icons und Bilder von Social Media-Plattformen) ausgerichtete methodische Auswertung. In der Methodenliteratur bezeichnet der Begriff induktiv die Entwicklung von Kategorien und Kodes aus dem empirischen Material heraus – Reichertz (2016, S. 138) definiert Induktion als „sichere Ableitung" –, welche dann die qualitative Inhaltsanalyse strukturieren. Im Gegensatz zur Induktion ist die deduktiv-qualitative Inhaltsanalyse eine Methode, die Datenmaterial (z. B. Gruppendiskussionen, Interviews, Parteiprogramme, Tweets und Zeitungsartikel) anhand eines aus Theorie(n) und vorherigen empirischen Untersuchungen erstellten (= Deduktion) Kategoriensystems analysiert (siehe Kapitel 5) – nur dass wir im vorliegenden Falle kein Kategoriensystem haben, sondern Wörter und Phrasen, die zuvor durch Forscher*innen mit Sentiments, d. h. mit Emotionen, verknüpft wurden.

Im vorangegangenen Kapitel haben wir uns mit der Korrespondenzanalyse auseinandergesetzt. Dabei handelt es sich um ein Verfahren, mit dessen Hilfe Sie Themendimensionen, also in Wörter zerlegte Texte, identifizieren können. Dabei werden auf einer Dimension immer zwei entgegengesetzte Themen identifiziert. Die Wörter, die nah beieinander auf diesen Dimensionen sind, treten in der Regel auch in den Texten gemeinsam auf. Weiterhin haben wir mittels der Korrespon-

denzanalyse die Möglichkeit, sowohl Texte als auch Wörter gleichzeitig in diesen Themendimensionen zu verorten. Das wiederum bietet uns die Möglichkeit, für die Themen typische Texte zu identifizieren und im Anschluss qualitativ auszuwerten. Darüber hinaus haben wir Ihnen eine Einführung in RStudio und die nötigen Pakete gegeben, ehe wir die Schritte von der Datenaufbereitung, über die Datenanalyse, Visualisierung, Interpretation und das Abspeichern der Ergebnisse gegangen sind. Nun wenden wir uns der Sentiment-Analyse zu, mit der uns eine im stärkeren Maße quantitative, automatisierte Methode der Textanalyse zur Verfügung steht. Aufgrund der breiten Verfügbarkeit von Paketen zur Sentiment-Analyse in R und Python werden wir eine Sentiment-Analyse in beiden Programmiersprachen durchführen. Beginnen werden wir dabei in R und uns dann zunächst die Grundlagen in Python ansehen, ehe wir die einzelnen Schritte durchgehen, die Sie für die erfolgreiche Durchführung einer Sentiment-Analyse gehen müssen.

Sentiment-Analyse, auch *Opinion Mining* genannt, bezeichnet Techniken aus dem Bereich der Computerlinguistik, mit denen positive oder negative Stimmungen, Subjektivität oder bestimmte Emotionen aus Textdateien herauspräpariert werden können. Historisch gesehen, lassen sich die ersten Versuche, Meinungen und Stimmungen zu erfassen, bis in die 1940er Jahre zurückverfolgen und hatten zum Ziel, Meinungen zum 2. Weltkrieg und dessen Auswirkungen zu erfassen (Mäntylä et al. 2018, S. 20). Diese Entwicklung wurde in den 1960er Jahren in den Computerwissenschaften aufgegriffen und seit den 1990ern erfolgreich Techniken entwickelt, um die Stimmungslagen in großen Textmengen erkennen zu können. Den Durchbruch schaffte die Sentiment-Analyse nach dem Jahr 2004 und Studien fokussieren seitdem Themen des gesellschaftlichen und politischen Wandels, der (nationalen) Sicherheit und Terrorismus, Wirtschaft und Finanzmarkt, Reisen, Medizin und Gesundheit, Unterhaltung und Sport, Sprache und Literatur oder menschliche Interaktion (ebd., S. 25 f.).

Sie werden sich nun sicherlich gefragt haben, warum Sie sich für die Sentiment-Analyse interessieren sollten und bei welchen Forschungsfragen Ihnen der Einsatz dieser Analyseform helfen kann. Stellen Sie sich vor, dass Sie Alltagsdiskriminierung von Migrant*innen untersuchen möchten. Sie wissen ferner, dass es auf Basis des Herkunftslandes oder der Länder, in denen Migrant*innen gearbeitet haben, auch positive Diskriminierung von offizieller Seite (Behörden) oder im Alltag geben kann (Schneider et al. 2020; Yilmaz 2019). Doch wie genau äußern sich Formen der Diskriminierung im Alltag? Welche Wörter und Sentiments (positiv/negativ, Angst, Fremdheit) weisen Kommunikationen auf, in denen Personen oder Gruppen diskriminiert werden? Alternativ könnten Sie Debatten über sensible Themen oder die Dynamik von Mobbing auf Social Media-Plattformen untersuchen und eine Anatomie der Debatte oder der Mobbingdynamiken je nach Sprecher*innen oder Adressat*innen getrennt vornehmen. Zuletzt könnten Sie die Ausschläge oder Veränderungen in den Stimmungs-

lagen nutzen, um Stichproben für qualitative Textanalysen zu ziehen, die in den Zeitraum dieser Veränderungen fallen. Auf diese Weise können Sie das hier vorgestellte Verfahren nutzen, um Dynamiken sozialer Phänomene qualitativ besser fassen zu können.

10.1.1 Schritt für Schritt-Ablauf einer Sentiment-Analyse

Um eine Sentiment-Analyse durchzuführen, müssen Sie wie in Kapitel 9 im Falle der Korrespondenzanalyse zunächst R oder Python installieren. Anschließend müssen Sie die Programmpakete installieren und diese mitsamt der Daten in Ihre Programmieroberfläche laden. Das betrifft auch die Lexika, in denen Wörter zu Sentiment-Wertungen zugeordnet sind. Danach bereinigen Sie die Texte, indem Sie alle Wörter in Kleinschreibung übersetzen, Satzzeichen entfernen und zuletzt nur diejenigen Wörter im Korpus behalten, die auch im Sentiment-Lexikon enthalten sind. Danach erkunden wir den Datensatz und analysieren, was die häufigsten Wörter sind, die Sentiments ausdrücken und inwiefern bestimmte Sprecher*innen häufiger als andere auftreten. Danach führen wir die Sentiment-

Abbildung 10.1 Schritt für Schritt-Vorgehen bei der Sentiment-Analyse

Schritt 1	**Vorbereitung und Installation** 1. R und RStudio oder Python, Anaconda und Spyder installieren 2. Pakete installieren 3. Pakete und Dateien in RStudio oder Spyder laden
Schritt 2	**Datenaufbereitung** 1. Entfernen von Datenartefakten 2. Stopwords entfernen 3. Text in Kleinschreibung konvertieren
Schritt 3	**Datenanalyse** 1. Erkundung der häufigsten Wörter mit Sentiment-Wert 2. Erkundung der Verteilung der Texte auf Sprecher*innen 3. Ermittlung der Sentiments des Gesamtdatensatzes 4. Analyse getrennt nach Gruppen
Schritt 4	**Daten verfügbar machen** 1. Analyseergebnisse visualisieren 2. Überführung der Analyseergebnisse in einen Datensatz 3. Speichern der Abbildungen und Datensätze
Schritt 5	Ergebnisse zusammenfassen und Erstellung einer Präsentation, Studienarbeit und/oder Publikation
Schritt 6	Sichere Archivierung der Daten und, wenn möglich, Aufbereitung zur Nachnutzung

Analyse durch, interpretieren gemeinsam die ausgegebenen Werte, erstellen Abbildungen und führen im Anschluss Gruppenvergleiche durch und testen, ob es Unterschiede in Sentiments zwischen Gruppen gibt.

10.1.2 Freude? Angst? Welches Gefühl möchten Sie erforschen?

Um eine Sentiment-Analyse durchzuführen, müssen Sie im Vorfeld der Untersuchung festlegen, ob Sie positive oder negative Stimmungen und Meinungen, den Grad subjektiver Aussagen oder verschiedene Emotionen (wie z. B. Freude, Angst, Ekel) messen möchten. Wenn Sie sich für eine Vorgehensweise entschieden haben, dann müssen Sie zunächst ein Lexikon mit Wörtern erstellen, die für Ihre Analyse geeignet sind. Im nächsten Schritt müssen Sie diese Wörter bewerten. Dies kann auf einer Skala z. B. von −10 bis +10 (negativ bis positiv) in ganzzahligen Schritten erfolgen oder kategorial, indem Sie einen Begriff als positiv, negativ etc. kategorisieren. Sie können die Begriffe dann Satz für Satz auf Basis der einzelnen zugewiesenen Wörter auszählen und z. B. einen Mittelwert berechnen, der angibt, wie positiv oder negativ die im Satz enthaltene Stimmung ist. Alternativ können Sie einen Schwellenwert definieren, mit dessen Hilfe Sie unterscheiden können, ob ein untersuchter Satz sehr negativ, negativ, neutral, positiv oder sehr positiv in seiner Grundstimmung ist. Ein Satz könnte beispielsweise sehr negativ bewertet werden, wenn 50 % oder mehr Wörter, die in dem Satz vorkommen und zugleich in Ihrem Lexikon enthalten sind, als negativ kategorisiert werden, oder einen Durchschnittswert von weniger als −6 auf einer Skala von −10 bis +10 aufweisen. Neben diesen einfachen Auszählungen können Sie auch in Erwägung ziehen, ein System aufzustellen, das Idiome (d. h. gleiche Bedeutungen von Wörtern), Phrasen oder die Position von Wörtern zueinander beinhaltet, um z. B. Negationen zu erkennen.[1]

Bevor Sie aber beginnen, eigene Lexika zu erstellen und Regelsysteme zu definieren, wann und wie eine im Text mithilfe des Algorithmus der Sentiment-Analyse erkannte Meinung oder Stimmungslage bewertet wird, können Sie auch auf bereits fertige Lexika zurückgreifen. Im vorliegenden Falle werden wir das R-Paket `tidytext` (Silge und Robinson 2016) verwenden, in dem drei dieser Lexika gespeichert sind. Für die Einführung werden wir uns auf ein Lexikon konzentrieren, bei dem die Sentiments im Bereich von −5 bis +5 erfasst werden und das auf einfachen Wortnennungen beruht. Im Falle von Python werden wir

1 Dies geschieht über ein sogenanntes *part of speech-tagging* (siehe Manning 2011, für einen Überblick und potenzielle Probleme bei diesem Verfahren). Bei diesem Vorgang wird einem Wort eine Wortart zugewiesen und es wird mit der Stelle eines Wortes innerhalb des untersuchten Satzes verknüpft.

auf das Paket *vaderSentiment* zurückgreifen, das uns syntaxbasierte positive, neutrale und negative Sentiment-Werte zurückgibt und daraus einen *Score* errechnet, der kontinuierliche Werte von −1 bis +1 annehmen kann. Hauptaugenmerk der Kapitel 10.2. und 10.3. liegt auf dem Dreischritt Aufbereitung, Messung und Visualisierung des durchschnittlichen Sentiments von Texten, wie auch der nach Gruppen getrennten Analyse von Sentiments. Wir lernen dabei ebenfalls Maßzahlen und Möglichkeiten kennen, mit deren Hilfe wir die Sentiments dieser Gruppen miteinander vergleichen können. Damit können Sie die weiter oben aufgeworfenen Fragen adressieren, indem Sie beispielsweise Migrantengruppen miteinander vergleichen oder die Sentiments der Kommunikation relevanter Sprecher*innen bei (Online-)Debatten zusammenfassen.

10.1.3 Datengrundlage

Als Datengrundlage verwenden wir eine selbst erstellte Stichprobe von 10 000 Bewertungen von Filmen und Serien, die mindestens fünf Rezensionen auf Amazon erhalten haben. Dabei wurden als Kontrast Produktkritiken gewählt, die nur einen Stern aufweisen (= niedrigste Bewertung) und solche, die fünf Sterne aufweisen (= höchste Bewertung). Die Grundlage hierfür stellt eine Json-Datei dar, die von Julian McAuley bereitgestellt wurde und unter *https://jmcauley.ucsd.edu/data/amazon* heruntergeladen werden kann. Eine Json-Datei ist ein Datensatz, der auf der JavaScript-Sprache beruht und es ermöglicht, eine Struktur in Webdaten zu erzeugen und sie damit in andere Dateiformate und Programmiersprachen übersetzbar zu machen. Der Datensatz umfasst dabei Variablen wie die Identifikationsnummer der Reviewer*innen *(reviewerID),* die Identifikationsnummer eines Produktes *(asin),* der Nickname der Reviewer*innen *(reviewerName),* wie viele weitere Konsument*innen das Review als hilfreich erachteten *(helpful),* den Text der Produktkritik *(reviewText),* die Bewertung von 1 bis 5 Sterne *(overall),* eine Zusammenfassung der Kritik *(summary),* den Erfassungszeitpunkt der Kritik in Unix-Zeit gezählt in Sekunden ab 00:00 Uhr des 1. Januars 1970 *(unixReviewTime)* sowie das Datum und Uhrzeit *(reviewTime).*

10.2 Sentiment-Analyse in R

Um eine Sentiment-Analyse in R durchzuführen, müssen wir 1) die Daten bereinigen, danach 2) Sentiment-Scores berechnen, ehe wir 3) die Ergebnisse visualisieren und im Anschluss 4) abspeichern können. Wie bei der Korrespondenzanalyse, werden wir wieder RStudio verwenden. Sollten Sie weder R noch RStudio installiert haben noch die Grundlagen von R kennen, dann können Sie in Kapitel 9.3 nachschlagen, wie RStudio installiert werden kann und wie die Grund-

lagen der Programmierung in R funktionieren. In unserer Sentiment-Analyse werden wir die Sentiment-Scores getrennt nach Reviewer*innen und gruppiert nach 1*- und 5*-Bewertungen in unserem kleinen Teildatensatz berechnen und visualisieren.

10.2.1 Verwendete Pakete in RStudio

In den folgenden Unterkapiteln verwenden wir `tidytext`, `textdata`, `tidyr`, `dplyr`, `stopwords`, `car` und `ggplot2`, um Sentiment-Analysen in RStudio durchzuführen. Das Paket `tidytext` beinhaltet Befehle, mit deren Hilfe Sie Texte einlesen, bereinigen sowie Inhalte filtern können (Silge und Robinson 2016). Das Paket `textdata` beinhaltet verschiedene Sentiment-Lexika, mit deren Hilfe wir den einzelnen Wörtern in den Amazon-Reviews Sentiment-Scores zuweisen können. In den Paketen `tidyr` (Wickham und Wickham 2017) und `dplyr` (Wickham 2014) finden Sie Befehle, mit deren Hilfe Sie einen Textkorpus für Ihre Textanalysen aufbereiten können. Das Paket `stopwords` beinhaltet – wie der Name schon sagt – Stopword-Listen, mit deren Hilfe Sie Wörter aus Texten entfernen können, die keinen inhaltlichen Mehrwert aufweisen. Das Paket `car` stellt Tests bereit, die wir später für Gruppenvergleiche verwenden werden. Es ist aber eher darauf ausgerichtet, Regressionsdiagnostiken zu berechnen, also Maßzahlen, die Sie für die und nach der Durchführung von Regressionsanalysen benötigen.[2] Das Paket `ggplot2` stellt viele Befehle zur Erzeugung von Grafiken bereit, mit deren Hilfe Sie die Ergebnisse visualisieren können.

Wie im vorangegangenen Kapitel erläutert und in Code 10.1 aufgeführt, installieren Sie die Pakete mittels `install.package()` und lesen diese mittels des `library()`-Befehls in RStudio ein. Den Datensatz in Tabellenformat (mit Dateiendung csv) können Sie in Ihre R-Sitzung importieren, indem Sie zunächst den Ordner mittels des `setwd()`-Befehls anwählen, in dem die Daten gespeichert sind. Anschließend können Sie die Datentabelle mittels `read.csv()` in Ihre R-Sitzung hineinladen. Bedenken Sie bitte, dass Sie den Ausdruck `[DATEIPFAD]` im gleich folgenden R-Code durch den entsprechenden Dateipfad auf Ihrem Ordner ersetzen. Alternativ klicken Sie auf „Import Dataset“ in Ihrem *Environments- und History*-Panel und suchen die Datei in dem in RStudio eingebetteten Dateibrowser (siehe Abbildung 9.2, Panel 4).

---

2 Bei der Regressionsanalyse handelt es sich um ein statistisches Verfahren, mit dessen Hilfe die Werte einer abhängigen Variablen (z. B. Sentiment-Score) auf eine Reihe erklärender Variablen (z. B. Alter, Geschlecht) geschätzt werden sollen. Es handelt sich dabei um ein hypothesentestendes Verfahren, bei dem Aussagen wie „je mehr x, desto mehr y“ oder „je mehr x, umso wahrscheinlicher tritt y auf/ein“ getestet werden sollen.

Code 10.1 Einlesen der Pakete und Dateien in R

```
# Einlesen der Pakete -------------------------------------------------------
library(tidytext)
library(textdata)
library(tidyr)
library(dplyr)
library(stopwords)
library(car)
library(ggplot2)

# Dateien einlesen ----------------------------------------------------------
setwd("C:/[DATEIPFAD]/Daten_Sentiment Analyse/")
data <- read.csv("Amazon_Movies_and_TV_5_Sample.csv")
```

10.2.2 Datenbereinigung

Im Folgenden bereiten wir die Daten so vor, dass wir eine Sentiment-Analyse durchführen können. Hierzu werden wir zuerst potenzielle Datenartefakte löschen. Datenartefakte sind Fehler in der Datenstruktur, die beispielsweise durch eine vorangegangene Bearbeitung in anderen Programmpakten wie SPSS oder Stata oder auch unterbrochene Datenspeicherprozesse hervorgerufen werden.

Nachdem wir die Daten geladen haben (die Reviews der Filme von Amazon), folgen die Bereinigungsschritte. Dies beginnt mit der Selektion von Variablen. Wie bei der Korrespondenzanalyse (siehe Kapitel 9), kann es sein, dass R beim Einlesen von csv-Dateien (alternativ Excel-Dateien) auf Spalten ohne Variablennamen trifft, beispielsweise wenn Sie zuvor eine Auswertung oder Aufbereitung in Python gemacht haben.[3] Wenn Sie eine Datei mit fehlenden Spaltennamen in R laden, dann vergibt R automatisch den Variablennamen X, wenn mehrere Variablen ohne Namen vorhanden sind, dann wird X.1, X.2 usw. vergeben. Diese wollen wir entfernen. Wie in Kapitel 9.3 (in Code 9.8) ausführlicher dargestellt, können wir diese unerwünschten Variablen mittels einer Verkettung des `select()`-Befehls und `-starts_with("X")` entfernen. Vergessen Sie dabei auch nicht, den Datensatz mit %>% anzusteuern und nach der Ausführung der Befehle mit <- an

3 Python tendiert dazu, Indexnummern auszugeben, mit deren Hilfe Sie die einzelnen Datenzeilen identifizieren können. In Python selbst muss die dazugehörige Spalte nicht benannt werden, da sie automatisch als Index erkannt wird. Beim Exportieren der Datei kann es allerdings passieren, dass er den Index ohne einen Variablennamen an eine csv-Datei oder eine Excel-Datei übergibt.

ein neues Objekt in R zu übergeben, damit das Entfernen der unnötigen Spalten gelingt.

Code 10.2 Entfernen von Datenartefakten

```
# Bereinigung der Daten ------------------------------------------------
## Datenzeilenartefakte entfernen
data <- data %>% select(-starts_with("X"))
```

Da in jeder Zeile ein Review ist und wir den Reviews später pro Wort Sentiment-Werte vergeben und somit den Datensatz wie eine Ziehharmonika auseinanderziehen, vergeben wir im nächsten Schritt eine Nummer für jedes Review. Ziel dieses Vorgangs ist, dass pro Zeile nur genau ein Wort stehen soll, dem später eindeutig ein Sentiment-Wert zugewiesen werden kann. Dabei handelt es sich um das sogenannte `tidytext`-Format, das dazu dienen soll, Ihnen den Überblick über Ihre Textdaten zu behalten. Hierfür nutzen wir den `mutate()`-Befehl, mit dessen Hilfe wir dem Datensatz neue Variablen hinzufügen und alte Variablen beibehalten können. Wir werden eine neue Variable mit Namen `review_number` erstellen, die eine fortlaufende Nummer von 1 bis zur Höchstzahl der Reviews in unserem Datensatz haben soll. Das können wir mit dem `seq()`-Befehl erreichen. Sie können als Obergrenze `10000` eingeben, da unser Datensatz insgesamt 10 000 Reviews beinhaltet. Falls Sie die Anzahl der Zeilen nicht wissen, dann können Sie diese mit `nrow()` abfragen. In die Klammern müssen Sie dann aber den von Ihnen geöffneten Datensatz angeben, dessen Zeilenanzahl Sie abfragen möchten, in unserem Falle also `nrow(data)`.[4] Diese übergeben wir den `seq()`-Befehl als Obergrenze der Zahlen, die in die neue Variable übergeben werden. Der ganze Vorgang wird in Code 10.3 zusammengefasst.

Code 10.3 Generierung einer fortlaufenden Nummer für die einzelnen Film- und Serienkritiken des Datensatzes

```
## Laufende Nummer für Reviews generieren
data <- data %>% mutate(review_number = seq(1,nrow(data)))
```

4 Möchten Sie eine oder mehrere neue Variablen generieren und die alten Variablen zugleich aus dem Datensatz entfernen, dann können Sie dies mittels des `transmute()`-Befehls erreichen. Sie könnten Beispielsweise einen Datensatz, der nur aus den Review-Nummern besteht, mit `data %>% transmute(review_number = seq(1,nrow(data)))` erzeugen.

10.2.2.1 Satzzeichen entfernen und in Kleinschreibung umsetzen

Nun müssen wir die Satzzeichen entfernen, die Wörter einheitlich klein schreiben, um eventuelle Fehler bei der Groß- und Kleinschreibung auszubessern und die Stopwords entfernen. Beginnend beim Entfernen der Satzzeichen, greifen wir auf den `gsub()`-Befehl zurück, der uns erlaubt, Zeichen und Zeichenketten zu ersetzen. Zur Erinnerung: Dieser Befehl hat drei Bestandteile. Zuerst geben Sie die Zeichenkette oder den regulären Ausdruck (Englisch: *regular expression*) an, den Sie ersetzen möchten. Dann geben Sie die Zeichenkette an, die Sie ersetzen wollen. An der zweiten Stelle folgt eine Zeichenkette, die anstelle der Zeichen eingesetzt wird, die an erster Stelle im Befehl aufgeführt sind. Zuletzt geben Sie den Text an, bei dem der Austausch der Zeichen und Zeichenketten durchgeführt werden soll. Wenn Sie zum Beispiel `gsub("quali","qual",["TEXT"])` angegeben haben, dann ersetzen Sie alle „quali" durch „qual" in Ihrem Text, unabhängig davon, ob es einzelne Wörter oder Bestandteile eines Ausdrucks wie „qualitative Inhaltsanalyse" sind.

Sie werden sich aber nun sicherlich gefragt haben, wie Sie R sagen, dass es alle Satzzeichen entfernen soll. Dies erreichen wir, indem wir `"[[:punct:]]"` an der ersten Stelle des `gsub()`-Befehls angeben. Dieser reguläre Ausdruck ist eine Abkürzung des englischen Wortes *punctuation,* übersetzt Satzzeichen. Wenn diese gelöscht werden sollen, die Wörter aber noch ein Leerzeichen zwischen ihnen aufweisen sollen, dann müssen Sie an zweiter Stelle `" "`, angeben. Insgesamt würde der Befehl nun `gsub("[[:punct:]]", " ", [TEXT])` lauten und `[TEXT]` könnte ein beliebiges Textobjekt sein.

In unserem Falle möchten wir, dass wir alle Texte verändern, die in der Variable reviewText enthalten sind. Entsprechend müssen wir den `gsub()`-Befehl im `mutate()`-Befehl verschachteln und unseren Datensatz mit `data %>%` auswählen. Wir werden nun beispielhaft die reviewText-Variable mit den um die Satzzeichen bereinigten Textdaten überschreiben. Innerhalb des `mutate`-Befehls können wir die Satzzeichen innerhalb der Reviews verändern, indem wir an die letzte Stelle des `gsub()`-Befehls `reviewText`, also den Variablennamen, angeben, wie Sie in Code 10.4 erkennen können.

Code 10.4 Entfernen von Satzzeichen aus den Texten mittels regulärer Ausdrücke

```
# Satzzeichen entfernen
data <- data %>% mutate(reviewText = gsub("[[:punct:]]", " ", reviewText))
```

Nun trennen wir die Sätze in einzelne Wörter auf, von denen je eines in einer Datenzeile zwischengespeichert wird. Dies erreichen wir, indem wir den Befehl

`separate_rows()` aus dem `tidyr`-Paket verwenden und in diesem die Variable angeben, in der unsere Reviews gespeichert sind. Darüber hinaus geben wir die `option convert = TRUE` an, um Ziffern (z. B. wenn eine Jahreszahl im Text vorhanden ist) in Wörter umzuwandeln. Damit wir den Überblick in weiteren Arbeitsschritten behalten und uns daran erinnern, dass wir keine ganzen Texte, sondern nurmehr ein Wort pro Zeile haben, benennen wir an dieser Stelle die in Wörter aufgetrennten Rezensionen in die variable `word` um. Das kann durch den `rename()`-Befehl erreicht werden, wie Sie in Code 10.5 erkennen können. Hierfür steuern wir in den Klammern zunächst den Datensatz an und geben dann, durch ein Komma getrennt, zunächst den neuen Namen der Variable an, die wir umbenennen möchten, ehe der alte Name durch ein = getrennt folgt. Wenn wir im Anschluss alle Wörter klein geschrieben haben wollen, die sich in der neuen Variable befinden, dann können wir diese mittels `tolower()` verändern, welche wir innerhalb des bereits bekannten `mutate()`-Befehls ausführen.

Code 10.5 Auftrennen des Textes in einzelne Wörter pro Zeile und Umsetzen der Wörter in Kleinschrift

```
data <- data %>% separate_rows(reviewText, convert = TRUE)
data <- rename(data, word = reviewText)
# Wörter kleinschreiben
data <- data %>% mutate(word = tolower(word))
```

10.2.2.2 Stopwords und Wortartefakte entfernen

Nun müssen wir nur noch die Stopwords entfernen und Wortartefakte mit genau einem Zeichen filtern, ehe wir die eigentliche Sentiment-Analyse durchführen können. Bei einem Wortartefakt handelt es sich um fehlerhafte Wörter, wie beispielsweise Wortschnipsel, die durch das Entfernen von Stopwords und Interpunktionen entstehen. Ein Beispiel hierfür wäre, wenn aus einem „it's" ein „it" und „s" wird. Letzteres wäre ein Wortartefakt. Hierfür laden wir zunächst die Stopword-Liste mittels `stopwords()` aus dem gleichnamigen Paket in unsere R-Sitzung hinein und übergeben Sie einem Objekt als DataFrame, welches wir `stopwoerter` nennen werden. Zur Erinnerung: Sie können eine Liste oder ein anderes Objekt als Datensatz definieren, indem Sie `as.data.frame()` verwenden. Da der `as.data.frame()`-Befehl bei der übergebenen Liste den Befehl als Variablenname verwendet, verändern wir diesen zunächst mittels des `rename()`-Befehls. Da wir diese Liste zum Entfernen der Stopwords in `data` verwenden möchten, in dem die Reviews Wort für Wort gespeichert sind, nennen wir die Variable in unserem `stoppwoerter`-Datensatz in `word` um. Als Hinweis: Nicht

die Datei, sondern die Variable heißt `word`, genauso wie im Datensatz `data`. Das hat den Vorteil, dass wir die Datensätze leichter zusammenführen können, weil Sie einen Variablennamen teilen, über den diese Zusammenführung möglich ist. Bis hierhin sieht der Code 10.6. zum Einladen der Stopwords und Umbenennen des Variablennamens wie folgt aus.

Code 10.6 Befehle zum Laden und Entfernen der Stopwords

```
# Stopwörter entfernen
stoppwoerter <- as.data.frame(stopwords())
stoppwoerter <- stoppwoerter %>% rename(word = "stopwords()")
```

Nun werden wir die Stopwords aus dem Datensatz filtern. Dies erreichen wir, indem wir `data` mit %>% ansteuern und den `anti_join()`-Befehl folgen lassen. Diesen Befehl übergeben wir unseren `stopwoerter`-Datensatz und geben mit der Option `by="word"` an, dass wir die Inhalte der Variablen namens word in beiden Datensätzen zusammenführen wollen. Der Befehl `anti_join()` negiert dabei die Zusammenführung. Anders ausgedrückt: Hier werden nur die Wörter in unserem Review-Datensatz behalten, die *keine* Stopwords sind. Zuletzt filtern wir alle Wörter heraus, die nur ein Zeichen lang sind und entledigen uns damit der Stopwords und eventuell übrig gebliebener Wortschnipsel. Dies können wir wieder machen, indem wir den `filter()`-Befehl benutzen und hier eine logische Abfrage angeben, die aussagt, dass wir alle Wörter behalten möchten, deren Zeichenlänge größer als 1 ist, also die Wortartefakte entfernen. Wie lang beispielsweise ein Wort oder eine Zeichenkette ist, können wir mittels des Befehls `nchar()` abfragen. Dies ist eine Kurzform von *number of characters,* also Zeichenzahl im Deutschen. Wenn wir die in `data` gespeicherten Daten wie schon zuvor mit %>% übergeben, dann können wir die Aussortierung von Wörtern mit nur einem Zeichen mit `filter(nchar(word) > 1)` realisieren. Die Filterung der Stopwords und Wörter mit einem (oder keinem) Zeichen können Sie – ohne Kommentar – in zwei Zeilen wie in Code 10.7 realisieren.

Code 10.7 Filtern aller Stoppwörter aus den Reviewtexten und Löschen aller Wortartefakte

```
data <- data %>% anti_join(stoppwoerter, by="word")
# Verbliebene Wortartefakte herausfiltern
data <- data %>% filter(nchar(word) > 1)
```

10.2.2.3 Erste Erkundung des Datensatzes nach den Bereinigungsprozessen

Schauen wir uns nun den Datensatz an, den wir erzeugt haben und betrachten die Wörter und Verteilung der Kommunikation auf die einzelnen Reviewer*innen, um ein Gefühl für die Daten zu bekommen, die wir für die Sentiment-Analyse verwenden. Beginnen wir dabei, wie in Code 10.8 demonstriert, mit der Größe unseres Textkorpus. Dieser besteht aus einzelnen Wörtern sowie einzigartigen Wörtern (= unterschiedlichen Wörtern). In ersterem Falle können Sie den `nrow()`-Befehl nutzen, um die Gesamtzahl der Wörter aufzurufen, aus denen Ihr Korpus besteht. Um die Anzahl unterschiedlicher Wörter zu zählen, müssen wir im zweiten Falle die Befehle `lenght(unique())` ineinander verschachteln.

Code 10.8 Auszählung der Korpusgröße und der einzigartigen Wörter innerhalb des Korpus

```
# Anzahl der Wörter berechnen
nrow(data$word)
length(unique(data$word))

# Corpus besteht aus 878.033 Wörtern (ohne Stopwörter) und 46.691 einzelnen
Wörtern
```

Darüber hinaus hilft es, sich die häufigsten Wörter eines Textkorpus auszugeben, um grob überblicken zu können, um welche Inhalte es am häufigsten geht. Um die häufigsten Wörter in Ihrem Datensatz auszuzählen und als Balkendiagramm darzustellen, können Sie zunächst die Worthäufigkeiten mittels des `count()`-Befehls auszählen. Die Logik ist auch hier, wieder Ihren Datensatz anzusteuern, in dem die Reviews bzw. Wörter gespeichert sind und dann mittels der Option `sort = TRUE` absteigend nach Anzahl des Auftretens im ganzen Datensatz zu sortieren. Um mit den Daten und Variablen nicht durcheinanderzukommen, übergeben wir die Wortauszählungen an ein neues Objekt. Wir können zwar theoretisch alle Wörter in Ihre Abbildung aufnehmen, da die meisten Wörter aber eher selten auftreten, würden Sie eine sehr langgezogene, unleserliche Grafik erzeugen, bei der wir die meisten Balken nicht erkennen würden. Daher behalten wir in unserem Falle nur die 25 Wörter für die Visualisierung, die am häufigsten über alle Rezensionen hinweg vorgekommen sind. Mittels `head([Objekt], n = 25)` können Sie erreichen, dass nur die ersten 25 Ein-

tragungen eines beliebigen Objektes (z. B. eine Liste, ein Datensatz) übrigbleiben.[5] Für die Auswahl können Sie Code 10.9 verwenden.

Code 10.9 Auswahl der 25 häufigsten Wörter Ihres Datenkorpus

```
## Auszählung und Visualisierung der häufigsten 25 Wörter
auszaehlung <- data %>% count(word, sort = TRUE)
auszaehlung <- head(auszaehlung, n = 25)
```

Nun erzeugen wir eine Abbildung mithilfe von Befehlen, die im `ggplot2`-Paket enthalten sind, und wie in Code 10.10 aufgeführt. Hierfür werden wir zunächst eine Art Leinwand für die Abbildung öffnen, danach festlegen, was auf die X-Achse (von links nach rechts) und die Y-Achse (von oben nach unten) abgebildet werden soll und übergeben die Werte. Hiernach sagen wir R, dass es ein Balkendiagramm erstellen soll und geben zusätzliche Optionen an. Ein lesbares Balkendiagramm können Sie mit den folgenden Codezeilen programmieren und abspeichern.

Code 10.10 Erstellung und Abspeichern eines Balkendiagramms, das die 25 häufigsten Wörter Ihres Datenkorpus enthält

```
plot_auszaehlung <- ggplot(data=auszaehlung,
    aes(x=reorder(word,-n), y=n)) +
 geom_bar(stat="identity", fill="steelblue") +
 theme(axis.text.x = element_text(angle=90)) +
 xlab("Häufigste Wörter") +
 ylab("Anzahl")

png("auszaehlung_Sentiment Analyse.png",
  width = 1600, height = 1000, units = "px", pointsize=300, res = 150)
print(plot_auszaehlung)
dev.off()
```

Nun gehen wir nochmal gemeinsam auf die einzelnen Bestandteile dieses Codes ein. Der `ggplot()`-Befehl ist dazu da, eine Art Leinwand für Ihre Grafik zu erstellen. Hier geben wir zunächst an, auf welche Daten wir zurückgreifen möchten. Danach legen wir mittels eines durch ein Komma getrenntes `aes()` fest, wie die

5 Um die letzten 25 Einträge eines Objektes anzuwählen, können Sie `tail([Objekt], n = 25)` eingeben.

X- und Y-Achsen ausgestaltet werden sollen. Innerhalb von `aes` wählen Sie nun die Variablen an, die Sie aufrufen wollen. Auf der X-Achse möchten wir Balken getrennt nach den Wörtern im Datensatz erzeugen, die Y-Achse soll die Anzahl der Nennungen der entsprechenden Begriffe abbilden. Wir haben die Wörter in der Variable `word`, die Auszählung in der Variable `n` gespeichert. Bislang haben wir somit `ggplot(data=auszaehlung, aes(x = word, y = n))`.

Würden wir nun aber fortfahren und ein Balkendiagramm erzeugen, dann würde uns RStudio eine Grafik ausgeben, bei der Werte auf der X-Achse alphabetisch geordnet sind. Wir möchten aber, dass sie nach absteigender Anzahl angeordnet sind. Daher übergeben wir dem `x` in `aes()` den Befehl `reorder(word, -n)`, um das zu erreichen. Wir müssen `word` angeben, weil diese Variable die Wörter enthält, die auf der X-Achse vorliegen. Dadurch, dass wir das Minuszeichen vor der Zählvariable, also `-n`, angeben, sagen wir RStudio, dass wir die Wörter in absteigender Reihenfolge sortieren möchten. Diese Reihenfolge wird durch die Werte bestimmt, die in unserer Variable `n` gespeichert sind. Dadurch, dass wir `ggplot(data=auszaehlung, aes(x = reorder(word,-n), y = n))` angegeben haben, haben wir nun unsere Leinwand formatiert, auf die wir das Balkendiagramm zeichnen werden. Das Balkendiagramm selbst rufen wir mittels `geom_bar()` auf, was wir in der Folgezeile getrennt durch ein + nach unserem `ggplot()`-Befehl einführen. Hier können wir nun angeben, wie unser Balkendiagramm aussehen soll. Wir müssen zunächst `stat="identity"` angeben, damit unser Balkendiagramm die Werte für jedes einzelne Wort abbildet. Wie wir nach der Sentiment-Analyse sehen werden, können wir statt „identity" auch „density" angeben, um eine Kerndichteschätzung[6] unserer Werte zu erhalten. Erstere Option eignet sich somit für einzelne, klar abgegrenzte Werte bzw. Gruppenauszählungen, letzteres eher für kontinuierliche Werte. Wir können die Farbe verändern, indem wir die Option `fill=""` spezifizieren. In unserem Falle haben wir uns für `fill="steelblue"` entschieden.[7] Wenn Sie hingegen die Farbe der Umrandung verändern möchten, dann können Sie dies mit `color =""` realisieren.

Drei weitere Optionen müssen wir noch einstellen, damit unser Balkendiagramm lesbar wird: Erstens sollten wir die Richtung des Textes ändern, in dem unsere 25 Wörter angegeben sind. Um die Richtung des Textes zu ändern, haben wir zunächst das generelle Aussehen des Graphen mit `theme()` angesteuert. Danach haben wir mit `axis.text.x =` angegeben, dass wir die Beschriftung der

---

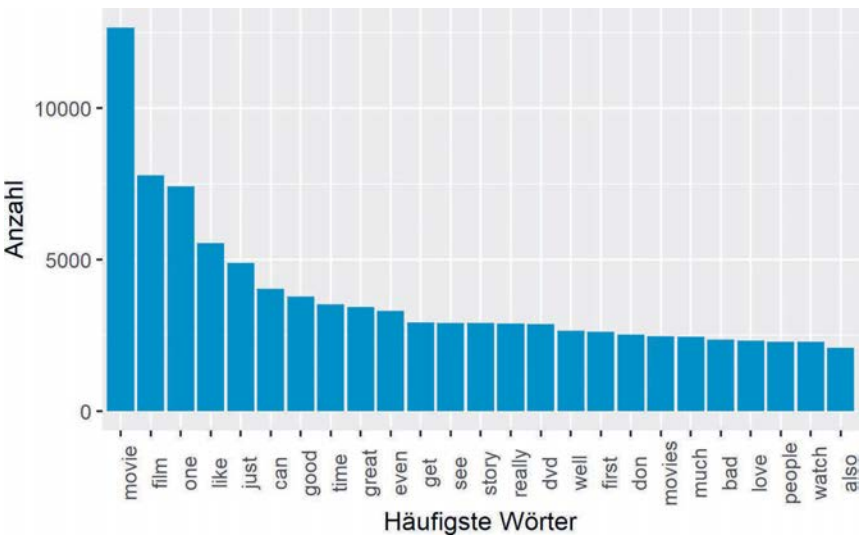
6 Unter einer Kerndichteschätzung versteht man ein statistisches Verfahren, mit dessen Hilfe Sie bei metrischen Verfahren mit vielen Ausprägungen Wahrscheinlichkeitsverteilungen berechnen können. Beispielsweise können Sie damit angeben, wie wahrscheinlich es ist, einem Sentiment-Wert zwischen 0,5 und 0,6 in Ihrem Datensatz zu begegnen, wenn die Werte zwischen –1 und +1 liegen und kontinuierlich sind, das heißt unendlich viele Ausprägungen zwischen –1 und +1 haben können.

7 Eine Übersicht über die Farben finden Sie beispielsweise unter *http://sape.inf.usi.ch/quick-reference/ggplot2/colour.*

einzelnen Striche auf der X-Achse, in unserem Falle die Wörter, auswählen möchten. Hier geben wir `element_text(angle=90)` an, um die Wörter anzuwählen und mittels `angle=90` um 90 zu drehen. Zweitens müssen wir die Beschriftungen der X- und Y-Achsen spezifizieren. Dazu können wir mittels `xlab()` und `ylab()` die X- bzw. Y-Achsenbeschriftung verändern. Für die X-Achse fügen wir die Beschriftung „Häufigste Wörter" und für die Y-Achse „Anzahl" ein.

Wir speichern das Bild als png-Datei, indem wir den gleichnamigen `png()`-Befehl aufrufen und hier zunächst den Namen angeben, unter dem wir unsere Abbildung speichern wollen und dann die Breite `(width =)` und Höhe `(height =)` in Pixeln `(units =)`. Mittels der Optionen `pointsize` und res können Sie die Auflösung der Grafik anpassen, damit diese nach dem Export auch scharf in ein anderes Dokument (z. B. ein Word-Dokument) eingefügt werden kann. Das Ergebnis finden Sie in Abbildung 10.2.

Abbildung 10.2 Balkendiagramm mit den 25 häufigsten Wörtern des Textkorpus



Wir erkennen hier, dass neben den Wörtern, die auf Filme hindeuten, eher positiv konnotierte Wörter (like, good, great, love), aber nur ein negatives Wort (bad) unter den 25 häufigsten Wörtern sind. Daneben gibt es viele Wörter, die eher als Füllwörter gewertet werden können (just, well, also). Das deutet darauf hin, dass wir entweder Wörter aus unserem Korpus der Rezensionen löschen sollten, um eine Sentiment-Analyse durchzuführen oder eine kontextsensitive Analyse durchführen, bei der auch Signalwörter oder der Syntax der Sätze hinzugezogen wird, um die Sentiment-Scores zu ermitteln.

Darüber hinaus kann es interessant sein zu prüfen, ob es Personen gibt, die die Kommunikationen dominieren. Bei diesen Personen müssten wir dann viele Kommunikationen bzw. Texte subsumiert finden. In unserem Falle sind es die Reviewer*innen, die mal mehr, mal weniger häufig mit Kritiken in unserem Datensatz vertreten sind. Um diese Auszählung durchzuführen, erstellen wir eine neue Variable mit den Namen `"no_reviews"`, die dies auszählen soll. Hierfür nutzen wir den in `dplyr` gespeicherten `add_count()`-Befehl. In unserem Falle übergeben wir diesem Befehl die Variable, deren Ausprägungen ausgezählt werden sollen. Dies ist in unseren Fall die `reviewerID`, deren Ausprägungen eine Zahlen- und Zeichenfolge ist, die den Namen des Reviewers symbolisieren soll. Mittels der Option `name = "no_reviews"` können wir eine neue Variable generieren. Anschließend können wir mittels des `summary()`-Befehls die Lagemaße, mittels `sd()` die Standardabweichung und `var()` die Varianz ermitteln. Code 10.11 zeigt Ihnen, wie Sie diese Auszählung und die Sichtung der Lage- und Streuungsmaße ermitteln.

Code 10.11 Ermittlung der Lage- und Streuungsmaße für die Anzahl verfasster Kritiken pro Reviewer*in

```
## Anzahl der Reviews pro Reviewer generieren
data <- data %>% add_count(reviewerID, name = "no_reviews")
summary(data$no_reviews)
sd(data$no_reviews)
var(data$no_reviews)
```

Halten wir aber noch einen Moment inne und führen uns vor Augen, was Lage- und Streuungsmaße sind. Bei Lagemaßen handelt es sich um statistische Maßzahlen, die wichtige, markante Punkte in Ihrer Variablen anzeigen. Dazu zählen Minimum, Quartils- bzw. Quantilswerte, Median, Mittelwert und Maximum. Das Minimum zeigt Ihnen den kleinsten, das Maximum den größten Wert an, der in Ihrer Variablen erreicht wird. In unserem Falle beantwortet das die Fragen, wie viele Reviews eine Person in unserer Stichprobe mindestens oder höchstens verfasst hat. Die Quartile berechnen die Werte, die von 25 % der Personen (1. Quartil), 50 % (2. Quartil) oder 75 % (3. Quartil) nicht überschritten werden, die im Datensatz vorkommen. Das 2. Quartil oder auch 50 %-Perzentil ist der Median. Zuletzt zeigt der Mittelwert an, wie viele Reviews eine Person im Durchschnitt geschrieben hat, die in unseren Datensatz aufgenommen wurde. Da der Mittelwert durch wenige sehr hohe Zahlen (= Ausreißer) sehr stark beeinflusst (= verzerrt) werden kann, ist es immer ratsam neben dem Mittelwert auch den Median anzugeben, wenn Sie eine Variable untersuchen.

Neben den Lagemaßen gibt es auch Streuungsmaße, die Ihnen anzeigen, wie

weit und gleichmäßig die Werte einer Variablen verteilt sind. Konzentrieren sich die Werte Ihrer Beobachtungen um den Mittelwert herum, dann ist die Streuung gering. Liegen die Werte hingegen weit auseinander und verteilen sich sehr weit vom Mittelwert entfernt, dann ist die Streuung groß. Diese Streuung wird Varianz genannt, welche die Summe der quadrierten Abstände zwischen Ihren beobachteten Werten und dem Mittelwert darstellt. Wir quadrieren die Werte, weil es auch „negative" Abstände geben kann, beispielsweise, wenn Sie einen Mittelwert von 3 haben, eine Beobachtung aber nur einen von 1. Das würde bedeuten, dass Sie einen Wert von –2 bekommen, wenn sie den Mittelwert von der Beobachtung abziehen würden. Summiert man alle Werte auf, dann würde eine Abweichung von 0 herauskommen, weswegen wir gerade die Abstände quadrieren müssen! Die Standardabweichung ist nichts anderes als die Wurzel der Varianz. Dabei gilt, dass in einem Abstand von ±2 Standardabweichungen 95 % Ihrer beobachteten Fälle verortet sind. Im Falle von ±3 Standardabweichungen sind es sogar 99 % der Fälle. Statistisch gesprochen handelt es sich hier um Konfidenzintervalle. Das heißt, dass Sie sich „sicher" sein können, dass Ihr berechneter Durchschnittswert in 95 % (oder 99 %) der Fälle im angegebenen Bereich um den Mittelwert verortet werden kann.

Wenn Sie die Datenzeilen anwählen und mit „Strg + Enter" ausführen, dann sagt Ihnen die Ausgabe, dass im Durchschnitt 1,716 Reviews pro Reviewer*in – minimal ein Review pro Reviewer*in, maximal 25 pro Reviewer*in – im Datensatz vorliegen. Der Median und die Quartilswerte sagen, dass sehr viele Reviewer*innen im Datensatz insgesamt wenig Reviews geschrieben haben. Die Standardabweichung und Varianz zeigen Ihnen, wie stark die Review-Anzahl um den Mittelwert herum streut. Wenn Sie in RStudio z. B. `2*sd[„variable"]` angeben, dann wüssten Sie, in welchem Bereich sich 95 % der Reviews befinden (= Mittelwert ±2*Standardabweichung). Zum Vergleich können Sie in Output 10.1 das Ergebnis Ihrer bisherigen Berechnungen betrachten.

Output 10.1 Ausgabe der Lage- und Streuungsmaße der Anzahl verfasster Kritiken pro Reviewer*in

```
> summary(data$no_reviews)
Min.  1st Qu. Median  Mean  3rd Qu.  Max.
1.000 1.000  1.000   1.716  1.000   25.000
> sd(data$no_reviews)
[1] 2.273739
> var(data$no_reviews)
[1] 5.169888
```

10.2.3 Durchführung der Sentiment-Analyse

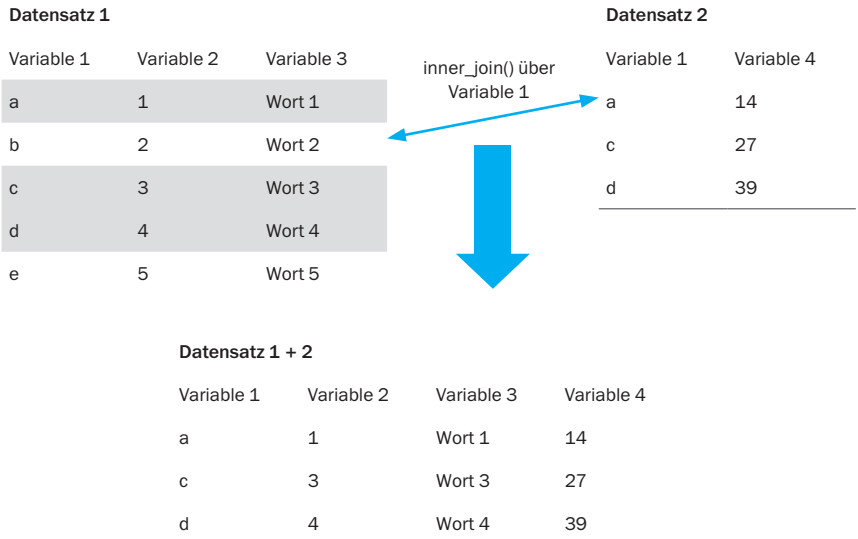
Wie schon angedeutet müssen wir, um eine Sentiment-Analyse auf Wortbasis durchzuführen, nur die Wörter in unserem Datensatz namens `data` beibehalten, die auch im Sentiment-Lexikon auffindbar sind. Das erreichen wir, indem wir zunächst wieder unseren Datensatz mit dem üblichen %>% ansteuern und dann einen `inner_join()`-Befehl ausführen. Dieser im Paket `dplyr` enthaltene Befehl ist der Programmiersprache SQL (Standard Query Language) entlehnt, mit dessen Hilfe Sie große Datensätze und Datenbanken verbinden können. Ganz konkret bewirkt dieser Befehl, dass Sie zwei Datensätze auf Basis einer in beiden Datensätzen geteilten Variable zusammenführen. Diese Variable beinhaltet in unserem Falle die Wörter der Rezensionen und Wörter mit Sentiment-Bewertungen aus einem Sentiment-Lexikon. Schauen wir uns nun gemeinsam an, wie wir das Sentiment-Lexikon mit positiven oder negativen Sentiment-Scores mit unserem Datensatz kombinieren können. Als allererstes laden wir das Lexikon mittels `get_sentiments("afinn")` in unsere Sitzung. Es verfügt über zwei Variablen: `word` und `value`. Da wir unsere Wortvariable in `data` ebenfalls in `word` umbenannt haben, prüft R nun über die Variable `word`, welche Wörter in beiden Datensätzen vorliegen. Dann löscht der Befehl alle Zeilen, die nicht in beiden Datensätzen vorliegen und fügt die restlichen Variablen beider Datensätze zusammen, die in den beibehaltenen Zeilen enthalten sind. Das Schema in Abbildung 10.3 (nächste Seite) verbildlicht diesen Vorgang, während die dazugehörige Codezeile 10.12 Ihnen das Handwerkszeug gibt, um die Sentiment-Scores und die Review-Daten in einen neuen Datensatz zusammenzuführen, den wir `sentiment_df_wort_basis` genannt haben.

Code 10.12 Filtern der Wörter ohne Entsprechung im Sentiment-Lexikon mittels inner_join()

```
# Durchführung der eigentlichen Sentiment-Analyse --------------------------
## Filtern der Wörter (nach Sentiment)
sentiment_df_wort_basis <- data %>% inner_join(get_sentiments("afinn"))
```

Nach der Zusammenführung unseres Datensatzes und des Sentiment-Lexikons können wir die Sentiments pro Reviewer*in berechnen. Zwei Schritte sind hierfür noch nötig. Erstens die Zusammenfassung der Sentiment-Scores pro Reviewer*in und zweitens die Berechnung des Durchschnittswertes aller Sentiments pro Reviewer*in. Das machen wir, da sich, wie wir weiter oben gesehen haben, Reviewer*innen im Datensatz befinden, die mehrere Rezensionen (maximal 25) geschrieben haben. Zudem dürfen wir nicht vergessen, dass pro Rezension mehrere Wörter mit Sentiment-Scores verknüpft worden sind. Entsprechend müssen

Abbildung 10.3 Schematische Darstellung des Vorgehens bei der Datenintegration mittels inner_join()-Befehl



wir auch diese Werte zusammenfassen und mitteln. In diesem Zuge zeigen wir Ihnen, wie Sie aufeinander folgende Datenmanipulationen Ihres R-Datensatzes vornehmen können. Betrachten wir nun gemeinsam Code 10.13, um einen Überblick über die Befehlsstruktur zu bekommen und gehen diese dann Schritt für Schritt durch.

Code 10.13 Berechnung des durchschnittlichen Sentiment-Wertes pro Reviewer*in

```
## Berechnung des Sentiments pro Reviewer
reviewer_sentiment <- sentiment_df_wort_basis %>%
 group_by(reviewerID) %>%
 summarise(sentiment = mean(value))
```

Wie Sie bemerken, haben wir unseren Datensatz wie in tidyverse mit %>% angesteuert und dann einen Befehl aufgerufen. Danach haben wir aber wieder das %>% und den nächsten Befehl eingegeben! Beachten Sie dabei, dass die Befehle zur Veränderung Ihrer Variablen in der Reihenfolge aufgerufen werden, in der Sie durch %>% voneinander abgetrennt sind. In unserem Falle wollen wir zunächst unsere Daten gruppieren – das machen wir durch den `group_by()`-Befehl. Hier

sollten Sie die Variable in den Klammern angeben, über die Sie die Werte zusammenfassen wollen. Das ist in unserem Falle die Variable `reviewerID`. Hiernach wollen wir den Mittelwert der Sentiment-Scores pro Reviewer*in berechnen. Das können wir mittels des `summarise()`-Befehls erreichen. Hier müssen wir angeben, welche Variable für die Berechnung herangezogen wird. Das ist bei uns der in `value` gespeicherte Sentiment-Wert. Danach geben wir an, dass wir den Mittelwert mittels `mean()`-Befehl berechnen wollen. Wenn Sie die Codezeilen in RStudio anwählen und mittels „Strg + Enter“ ausführen, dann erhalten Sie Output 10.2.

Output 10.2 Durchschnittliches Sentiment pro Reviewer*in

```
> reviewer_sentiment
# A tibble: 8,289 x 2
   reviewerID                      sentiment
   <chr>                               <dbl>
 1 A010234914H939HO67HYO                   3
 2 A02296971ID4CDE9LRKK2                -2.5
 3 A02755422E9NI29TCQ5W3                   3
 4 A05240112SHI9MDAJ1FIP                   3
 5 A06715701VPDNPM5R1AKN                   2
 6 A072589926D14ZPAPX61S                   1
 7 A100K3KEMSVSCM                       1.33
 8 A100V5QEICGPDA                     -0.125
 9 A102B8D74H64TO                        0.8
10 A102RDJLOHWSOW                     -0.571
# ... with 8,279 more rows
```

Nun werden wir die mittleren Sentiment-Scores, getrennt nach 1*- und 5*-Bewertung pro Reviews, berechnen. Wie zuvor schon, gruppieren wir unsere Werte und berechnen die Mittelwerte der Sentiments pro Rezension (Code 10.14). Das Einzige, was wir an dieser Stelle ändern müssen, ist die Variable. Wir nutzen an dieser Stelle die Variable `asin` zur Gruppierung und erhalten einen Datensatz, der aus den Identifikationsnummern der Rezensionen und den durchschnittlichen Sentiment-Werten der jeweiligen Rezension besteht. Generieren wir nun einen Datensatz, dem wir den Namen `mean_sentiment_per_star` geben.

Code 10.14 Berechnung der durchschnittlichen Sentiment-Werte für 1*- und 5*-Bewertungen

```
## Berechnung des durchschnittlichen Sentiments getrennt nach 1*- und
5*-Bewertung
mean_sentiment_per_star <- sentiment_df_wort_basis %>%
 group_by(asin) %>%
 summarise(sentiment = mean(value))
```

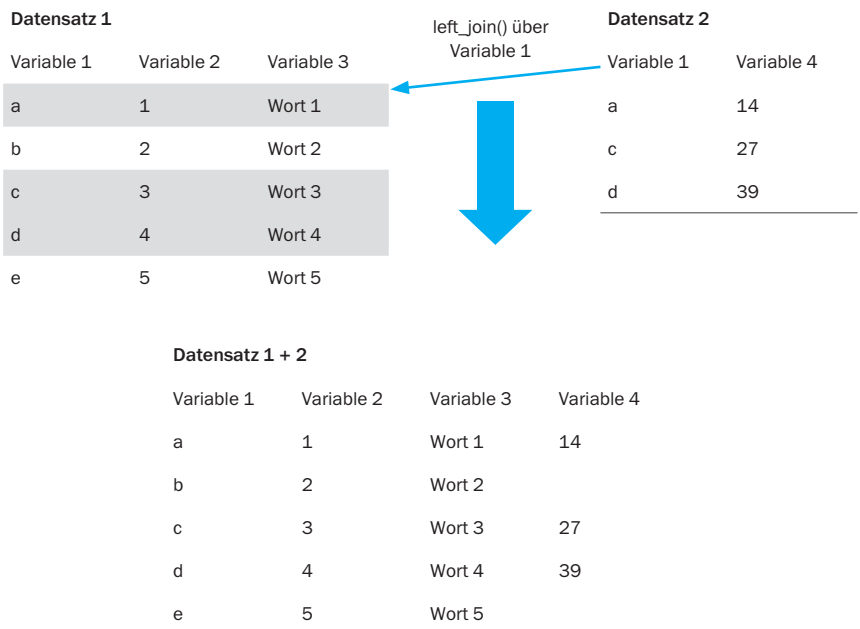
Durch diese Berechnung ist aber noch etwas ganz Entscheidendes aus dem Datensatz entfernt worden. Nämlich die Bewertungen, nach denen wir unsere Sentiment-Analysen gruppieren wollen! Entsprechend müssen wir diese Bewertungen wieder an unseren Datensatz spielen, damit wir die Sentiment-Scores getrennt nach Gruppe berechnen können. Wir können unsere Datensätze über den Befehl `left_join()` kombinieren. Dieser Befehl bewirkt, dass wir an einen bereits vorhandenen Datensatz (im übertragenen Sinn die „linke" Seite) die verfügbaren Variablen eines anderen Datensatzes („rechte" Seite im übertragenen Sinn) anfügen. Hierfür müssen wir eine oder mehrere Zeilen angeben, über die diese Zusammenführung erfolgen soll. Exemplarisch ist die Logik des `left_join()`-Befehls in Abbildung 10.4 aufgeführt. Entsprechend steuern wir nun den Datensatz `mean_sentiment_per_star` mit %>% an und teilen RStudio mit, dass wir unseren Ausgangsdatensatz `data` anfügen möchten. Die Zusammenführung erfolgt über die Identifikationsnummern der Rezensionen, die in der `by =` Option angegeben werden. Wenn wir diese Zusammenführung an unseren Datensatz mit Namen `mean_sentiment_per_star` mit `<-` übergeben möchten, dann können wir die folgende Codezeile 10.15 verwenden.

Code 10.15 Übergeben der Sentiment-Werte an den Ausgangsdatensatz mittels left_join()

```
## Sternbewertung hinzunehmen
mean_sentiment_per_star <- mean_sentiment_per_star %>% left_join(data, by =
"asin")
```

Nun stoßen wir aber auf das Problem, dass wir einen Datensatz mit sehr vielen Zeilen generiert haben, in dem alle Wörter, die zu Beginn unserer Untersuchung vorhanden waren und jeweils in den Zeilen abgelegt wurden, wieder in den zusammengeführten Datensatz hineinkopiert wurden. Da wir aber nur einen Sentiment-Wert pro Rezension benötigen und darüber hinaus die Sternbewertung und die Review ID, müssen wir zuerst die Variablen auswählen, die

Abbildung 10.4 Schematische Darstellung der Datenzusammenführung mittels left_join()



wir für unsere Analysen benötigen und dann alle Dopplungen entfernen. Wir haben nämlich mit dem `left_join()`-Befehl erreicht, dass im Datensatz mit allen Wörtern (800 988 Zeilen!) alle Durchschnittswerte der Sentiments gespeichert wurden. Ersteres realisieren wir, indem die Namen der im Datensatz befindlichen Variablen aufgerufen werden, die wir behalten wollen (siehe dazu auch Kapitel 10.2.2). Zur Erinnerung: Sie können Variablen selektiv im Datensatz ansteuern, indem Sie den Datensatznamen angeben und dann die Variablennamen durch den `c()`-Befehl in eckigen Klammern öffnen und die Variablennamen getrennt und in Anführungszeichen in die runden Klammern des `c()`-Befehls schreiben. Nun müssen Sie noch diesen Aufruf in den `unique()`-Befehl schreiben. Dieser Befehl sagt R, dass er nur die einzigartigen Werte in dem übergebenen Objekt behalten soll: In unserem Falle der Ausschnitt aus unserem Datensatz, der die Reviewer*innen ID, den Sentiment-Wert und die Bewertung mit 1* oder 5* beinhaltet (Code 10.16).

Code 10.16 Erstellung eines Datensatzes mit einzigartigen Wertekombinationen aus Nutzer*innen ID, Sentiment-Wert und 1*- und 5*-Bewertungen

```
mean_sentiment_per_star <- unique(mean_sentiment_per_
star[c("asin","sentiment","overall")])
```

Nun müssen Sie RStudio angeben, dass die 1*- und 5*-Bewertungen als Gruppen gehandhabt werden sollen (Code 10.17). Die Wertungen sind ganzzahlig (integer) und würden von R damit sonst nicht als Variable erkannt werden. Um die Bewertungen als Gruppen zu charakterisieren, müssen Sie die Variable innerhalb Ihres Datensatzes mittels `mean_sentiment_per_star$overall` auswählen und in den `factor()`-Befehl einfügen. Dieser Befehl ermöglicht es Ihnen, eine numerische Variable mit abzählbar vielen Ausprägungen in eine nominalskalierte Variable zu übersetzen, in der die einzelnen Gruppen enthalten sind.

Code 10.17 Übersetzung der als Zahlen gespeicherten 1*- und 5*-Bewertungen in eine Gruppenvariable

```
mean_sentiment_per_star$overall <- factor(mean_sentiment_per_star$overall)
```

Nun lassen wir uns die deskriptiven Statistiken der Sentiments unserer Rezensionen ausgeben. Im ersten Falle wollen wir die deskriptiven Statistiken der Sentiments unserer Reviewer*innen ermitteln, im zweiten Falle getrennt nach 1*- und 5*-Bewertungen. Ohne Gruppenvergleich können Sie den `summary()`-Befehl verwenden und die Sentiment-Variable ansteuern. Im zweiten Fall müssen wir, um die Gruppen zusätzlich ansteuern zu können, den `tapply()`-Befehl verwenden. Dieser Befehl sagt R, dass eine Funktion, in unserem Falle `summary()`, auf eine Variable ausgeführt werden soll, die ihrerseits in Gruppen aufgeteilt ist. Entsprechend würde der allgemeine Syntax `tapply([Variable], [Gruppenvariable], [BEFEHL])` lauten. Anstelle der `[Variable]` geben Sie `mean_sentiment_per_star$sentiment` an, da hier die Grundlage für die Berechnung, also die Sentiment-Werte, angelegt sind. Die Gruppennamen sind in `mean_sentiment_per_star$overall` gespeichert. Zuletzt folgt der Befehl, den Sie ausführen wollen, der aber keine runden Klammern benötigt, weil er mittels `tapply()` bereits aufgerufen wird, wie in Code 10.18 demonstriert wird.

Code 10.18 Berechnung von Lage- und Streuungsmaßen getrennt nach Gruppen mittels tapply()

```
summary(reviewer_sentiment$sentiment)
tapply(mean_sentiment_per_star$sentiment,
    mean_sentiment_per_star$overall,
    summary)
```

Wenn wir beide Zeilen markieren und mit „Strg + Enter“ ausführen, erhalten wir Output 10.3.

Output 10.3 Ausgabe der Lagemaße für den gesamten Datensatz und getrennt nach 1*- und 5*-Bewertungen

```
> summary(reviewer_sentiment$sentiment)
  Min. 1st Qu. Median  Mean 3rd Qu.  Max.
-4.0000 -0.1212 0.8000 0.8314 2.0000 4.5000
> tapply(mean_sentiment_per_star$sentiment,
+    mean_sentiment_per_star$overall,
+    summary)
$`1`
  Min. 1st Qu. Median  Mean 3rd Qu.  Max.
-4.0000 -0.5333 0.1163 0.1197 0.8022 4.0000

$`5`
  Min. 1st Qu. Median  Mean 3rd Qu.  Max.
-4.0000 0.5849 1.4118 1.3806 2.2500 4.5000
```

Auf den ersten Blick sehen wir, dass unterschiedliche Werte bei den Lagemaßen, bestehend aus Minimum, Quartilen, Median, Mittelwert und Maximum vorliegen (siehe Kapitel 10.2.2 für eine Erläuterung). Zum Beispiel sehen wir, dass der Median bei den nach Reviewer*innen gemittelten Sentiment-Scores bei 0.8, im Falle der 1*-Bewertung bei 0.1163 und bei den 5*-Bewertungen bei 1.4118 liegt. Der Mittelwert zeigt ähnliche Werte an (0.8314 bei den Reviewer*innen, 0.1197 bei 1*-Rezensionen und 1.3806 bei den 5*-Reviews). Wenn wir davon ausgehen, dass ein Sentiment von 0 eine neutrale Stimmungslage andeutet, dann sagen unsere Werte aus, dass die 5*-Bewertungen Wörter enthalten, die insgesamt auf eine positive Stimmungslage hindeuten. Diese Aussage wird nicht nur durch die Mittelwerte und Median, sondern auch durch die Quartilswerte gestützt. Eine genauere Sichtung sollten Sie später allerdings bei den Texten vornehmen, die un-

gewöhnlich positive Sentiments aufweisen, obwohl sie eine 1*-Bewertung haben oder besonders negatives Sentiment haben, obwohl sie eine 5*-Bewertung aufweisen. Dadurch können Sie in Erfahrung bringen, woran es beispielsweise liegt, dass gute Rezensionen negative Sentiments haben und vice versa. Hier könnte unser Algorithmus etwas gemessen haben, was nicht direkt mit den Bewertungen der Filme zu tun hat. Vielleicht handelt es sich bei den 1*-Bewertungen mit einem hohen Wert über 4 um Liebesfilme, die durch eher positive Wörter wie „love" auffallen, während Horrorfilme eher mit negativen Wörtern wie „terrifying" auffallen dürften, die hier sogar als Gütesiegel gemeint sein können. Eventuell wäre es in letzterem Falle besser, andere Emotionen wie Angst zu messen, die hier ein besonders positives Filmerlebnis anzeigen könnten. Auf jeden Fall sollten Sie eine qualitative Analyse folgen lassen, um diese Frage zu klären.

Zuletzt betrachten wir noch die Standardabweichung für beide Gruppen. Hierfür müssen Sie, wie in Code 10.19 den `summary()`-Befehl durch den `sd()`-Befehl ersetzen und erhalten Output 10.4.

Code 10.19 Ermittlung der Standardabweichung für den gesamten Datensatz und getrennt nach 1*- und 5*-Bewertungen

```
sd(reviewer_sentiment$sentiment)
tapply(mean_sentiment_per_star$sentiment,
    mean_sentiment_per_star$overall,
    sd)
```

Output 10.4 Ausgabe der Standardabweichungen für den gesamten Datensatz und getrennt nach 1*- und 5*-Bewertungen

```
> sd(reviewer_sentiment$sentiment)
[1] 1.390838
> tapply(mean_sentiment_per_star$sentiment,
+    mean_sentiment_per_star$overall,
+    sd)
    1    5
1.104914 1.109018
```

10.2.3.1 Visualisierung der Sentiment-Analysen

Um den eben genannten ersten Schritt zu gehen und diese Berechnungen mit Leben zu füllen, können Sie mithilfe des `ggplot2`-Paketes sowie einigen Befeh-

len Abbildungen erzeugen. Im Unterschied zu den vorangegangenen Grafiken, die wir in RStudio für die Vorauswertung der Sentiment-Analyse und die Korrespondenzanalyse erzeugt haben, wollen wir diesmal Histogramme erzeugen. Unter einem Histogramm wird ein spezieller Typus von Balkendiagrammen verstanden, der zur Darstellung kontinuierlicher Variablen mit sehr vielen Werten geeignet ist.[8] Bei einem Histogramm werden Spannbreiten festgelegt, in der die Häufigkeit des Auftritts bestimmter Werte wie zum Beispiel ein Sentiment-Score zwischen −1 und −0.9 gezählt wird. Darüber hinaus wollen wir ein Dichteplot erzeugen, bei dem eine Kerndichteschätzung vorgenommen wird. Das klingt auf den ersten Blick kompliziert, folgt aber in etwa der gleichen Logik wie das Histogramm. Nur dass wir hier einen Wert festlegen (den Kern, z. B. ein Sentiment-Score von −0.95), um den herum in einem zuvor festgelegten „Fenster" (z. B. für Werte zwischen −1 und −0.9) alle Beobachtungen zusammengefasst und in eine Wahrscheinlichkeit überführt werden, mit der die Werte in diesem Fenster beobachtet werden können. Dadurch erhalten wir keine Balken, sondern eine Art Kurve, die wir mithilfe der folgenden Codezeilen 10.20 neben Histogrammen erzeugen und als `abbildung_1` und `abbildung_2` in Ihren R-Workspace zwischenspeichern können.

Code 10.20 Erstellung von Histogrammen mit Mittelwert und Histogrammen und Kerndichteschätzungen

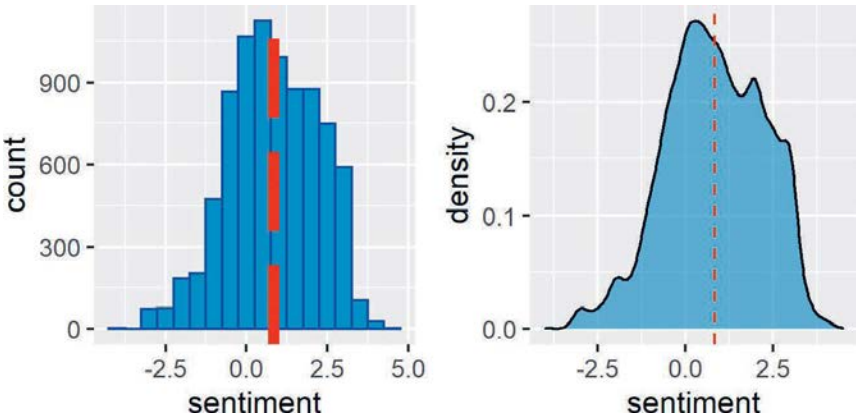
```
# Visualisierung der Sentiments ------------------------------------------
## Histogramm mit Mittelwert
abbildung_1 <- ggplot(data=reviewer_sentiment, aes(x=sentiment)) +
 geom_histogram(color="blue", fill="steelblue", binwidth = 0.5) +
 geom_vline(xintercept = mean(reviewer_sentiment$sentiment),
       color="red", linetype=5, size=2)

## Histogramm mit Mittelwert und Kerndichteschätzung
abbildung_2 <- ggplot(data=reviewer_sentiment, aes(x=sentiment)) +
 geom_density(alpha= 0.7, color = "blue", fill="steelblue") +
 geom_vline(xintercept = mean(reviewer_sentiment$sentiment),
       color="red", linetype=2)
```

8 Eine kontinuierliche Variable verfügt über unendlich viele Ausprägungen. Zum Beispiel kann unser Sentiment-Score einen beliebigen Wert zwischen −1 und +1 annehmen. Wenn wir den Wert auf vier Nachkommastellen runden würden, dann würde man je tausend Werte zwischen −1 und 1 einen für 0 und weitere tausend zwischen 0 und 1 haben. Ohne Rundung würde man niemals alle Werte zählen können, da beliebig viele Nachkommastellen bei den Werten möglich wären. Daher sagt man auch, es handle sich bei einer kontinuierlichen Variablen um eine Variable mit überabzählbar vielen Ausprägungen.

Wie bereits im Unterkapitel 10.2.2 rufen wir in diesen Codezeilen wieder unsere Leinwand mittels des `ggplot()`-Befehls auf. Hier geben Sie zunächst mit der `data =`-Option den Datensatz an, in dem die Variable befindlich ist, deren Verteilung dargestellt werden soll. Danach setzen Sie die X- und Y-Achsen-Ästhetik mit der `aes()`-Option. In unserem Falle brauchen wir nur die X-Achse anzugeben, da die Y-Achse durch die Werte unseres Histogramms bzw. Dichteplots festgelegt werden. Um ein Histogramm aufzurufen, müssen Sie nach dem `ggplot()`-Befehl zunächst ein + am Ende Ihrer Codezeile einfügen und in der nächsten Zeile `geom_histogram()`. Hier geben Sie mit der `binwidth`-Option an, wie groß die Bereiche sein sollen, auf deren Basis die Balken des Histogramms gezeichnet werden. Wir färben darüber hinaus die Umrandung der Balken mit `color = "blue"` sowie die Balken selbst mittels `fill = "steelblue"` ein. Zuletzt erzeugen wir eine senkrechte Linie mittels `geom_vline()` in der nächsten Codezeile (ebenfalls durch ein + am Ende der vorigen Zeile abgegrenzt) und übergeben den Mittelwert unseres Sentiment-Scores mit der Option `xintercept = mean(reviewer_sentiment$sentiment)`. Wenn Sie den Mittelwert einer anderen Variablen als X-Wert angeben möchten, von dem aus eine senkrechte Linie in Ihre Grafik gezeichnet wird, dann müssen Sie den Variablennamen (und den Datensatznamen) entsprechend anpassen. Wir färben die Linie mit `color = "red"` rot und ändern mit `linetype` den Linientypus und mit der `size`-Option die Größe dieser senkrechten Linie. Damit wären alle Befehle und Optionen erklärt, die `abbildung_1` auszeichnen. Um ein Density-Plot zu erzeugen, können Sie den `geom_histogram()`-Befehl durch den `geom_density()`-Befehl austauschen. Hier haben sie die gleichen Möglichkeiten zur Einstellung des Aussehens (in unserem Falle `color` und `fill`). Zusätzlich dazu haben wir mit `alpha = 0.7` festgelegt, dass unser Dichteplot leicht transparent ist. Je niedriger die Werte bei dieser Option sind, desto durchsichtiger ist nachher die Abbildung. Wenn Sie nun zuerst `abbildung_1` anwählen und „Strg + Enter“ drücken und dann `abbildung_2` und diesen Vorgang wiederholen, dann erhalten Sie die Grafiken, die in Abbildung 10.5 enthalten sind.

Zuletzt wollen wir diese Abbildungen noch speichern. Auch in diesem Falle können wir das, indem wir zunächst über `png()` den Dateinamen und die Eigenschaften der Abbildung angeben, die wir speichern wollen. Dann zeichnen wir mit `print()` das Bild und schließen Selbiges mittels des `dev.off()`-Befehls. Wie Sie beide Bilder speichern können Sie Code 10.21 entnehmen. Darüber hinaus können Sie gegebenenfalls Ihre Bildgrößen und Auflösungen anpassen.

Abbildung 10.5 Visualisierung des Histogramms und des Kerndichteplots der Sentiment-Werte des gesamten Datensatzes



Code 10.21 Erstellung und Speichern der Histogramme und Dichteplots

```
# Abbildungen aufrufen und speichern ------------------------------------
png("Abbildung_1_Sentiment_Barplot.png", width= 800, height = 800,
    res = 300)
print(abbildung_1)
dev.off()

png("Abbildung_2_Sentiment_Barplot_Distplot.png", width= 800, height = 800,
    res = 300)
print(abbildung_2)
dev.off()
```

Nun erstellen wir mit Code 10.22 eine Grafik für die Bewertungen der 1*- und 5*-Rezensionen, die sowohl ein Histogramm und einen Dichteplot für beide Gruppen beinhaltet. Dabei gehen wir fast genauso vor, wie wir es bei den beiden vorangegangenen Grafiken gemacht haben. Wir müssen allerdings ein paar Änderungen vornehmen, damit wir eine Abbildung nach Gruppen getrennt generieren können, die beide Plottypen und je eine Mittelwertlinie pro Gruppe aufweist. Wir zeigen Ihnen zunächst den Code und erläutern Ihnen dann die Änderungen.

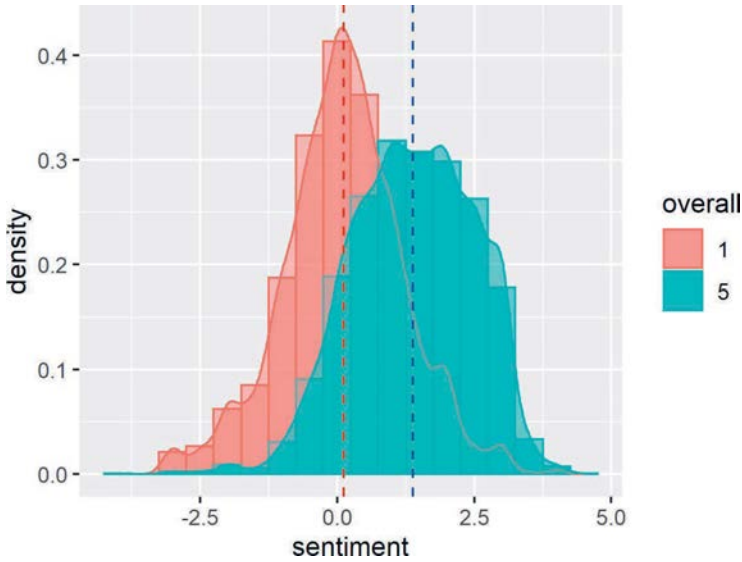
Code 10.22 Erstellung eines Histogramms mit überlagerndem Kerndichteplot für 1*- und 5*-Bewertungen

```
## Histogramm getrennt nach Sternbewertung
abbildung_3 <- ggplot(data=mean_sentiment_per_star, aes(x = sentiment, y =
..density.., color = overall, fill = overall)) +
geom_histogram(position = "identity", binwidth = 0.5, alpha = 0.5) +
  geom_density(alpha= 0.5) +
  geom_vline(xintercept = tapply(mean_sentiment_per_star$sentiment,
                                 mean_sentiment_per_star$overall,
                                 mean),
             color=c("red","blue"), linetype=2)
```

Zu Beginn müssen wir in unserem `ggplot()`-Befehl in den `aes()`-Einstellungen einige zusätzliche Optionen angeben. Neben der Angabe, dass auf der X-Achse die Sentiment-Scores abgebildet werden, fügen wir hier noch `color = overall` und `fill = overall` hinzu. Wir müssen dies an dieser Stelle machen, weil wir dadurch RStudio beibringen, dass es auf Füllungen für Balken (und Umrandungen) in den Farben unterschiedlicher Gruppen zurückgreifen soll. Dadurch erkennt RStudio auch automatisch, dass Sie eine Abbildung zeichnen möchten, in der unterschiedliche Gruppen abgebildet sind. Diese Gruppen haben wir in unserem Datensatz in der `overall`-Variable gespeichert. Darüber hinaus geben wir zusätzlich die Option `aes(y=..density..)` an. Dadurch sagen wir R, dass wir die Dichtefunktion als Grundlage für unser Balkendiagramm und den Density-Plot verwenden wollen. Keine Sorge, wenn Sie wie zuvor im `geom_histogram()`-Befehl die Option `position = "identity"` angeben, erhalten wir wieder Balken. Wir setzen noch zusätzlich einen Wert von `alpha = 0.5` ein, damit wir später den Dichteplot hinter dem Histogramm noch erkennen können. Hiernach folgt der `geom_density()`-Befehl, ehe wir die beiden vertikalen Linien mit dem `geom_vline()`-Befehl aufrufen. Hier können wir je einen Mittelwert für beide Gruppen durch `xintercept = tapply()` übergeben. Wie Sie weiter oben gesehen haben, müssen wir im `tapply()`-Befehl zunächst die Variable für unsere Sentiment-Scores übergeben, ehe die Variable folgt, in der die Gruppen gespeichert sind. Zuletzt folgt der Befehl, den wir ausführen möchten, allerdings ohne runde Klammern. In unserem Fall wäre das der `mean()`-Befehl. Zuletzt müssen wir zwei Farben für die Striche übergeben, die wir mittels `color=c("red","blue")` in `geom_vline()` festlegen. Vergessen Sie bei dem ganzen Vorgang aber nicht, alle Befehle durch ein + und einen Zeilenumbruch voneinander zu trennen! Wenn Sie nun `abbildung_3` markieren und mit „Strg + Enter“ ausführen, erhalten Sie Abbildung 10.6.

Wie Sie sehen, deutet sich auch grafisch ein Unterschied im Sentiment-Score

Abbildung 10.6 Histogramm mit überlagerndem Kerndichteplot für 1*- und 5*-Bewertungen



zwischen den Rezensionen mit 1* und 5* an, den wir auch durch die Betrachtung unserer Lage- und Streuungsmaße vermuten konnten. Sie erkennen dies einerseits daran, dass die Verteilung der 5*-Rezensionen im Vergleich zu den 1*-Rezensionen weiter nach rechts verschoben und ein größerer Teil der Dichtefunktion und der Balken des Histogramms weit in die positiven Bereiche verschoben ist. Auch die Mittelwerte liegen, in Anbetracht der Werteskala von −5 bis +5, relativ weit auseinander. Zuletzt speichern wir unsere Grafik mit einer Kombination aus den `png()`, `print()` und `dev.off()`-Befehlen wie folgt und erkunden dann statistisch, ob sich die Sentiments der 1*- und 5*-Rezensionen systematisch unterscheiden.

Code 10.23 Erstellung eines Histogramms mit überlagerndem Kerndichteplot für 1*- und 5*-Bewertungen

```
# Abbildungen aufrufen und speichern ----------------------------------------
png("Abbildung_3_Sentiment_Bewertungen_Distplot.png", width= 1500, height =
    1500, res = 300)
print(abbildung_3)
dev.off()
```

10.2.3.2 Gruppenvergleiche mittels statistischer Mittelwertdifferenztests

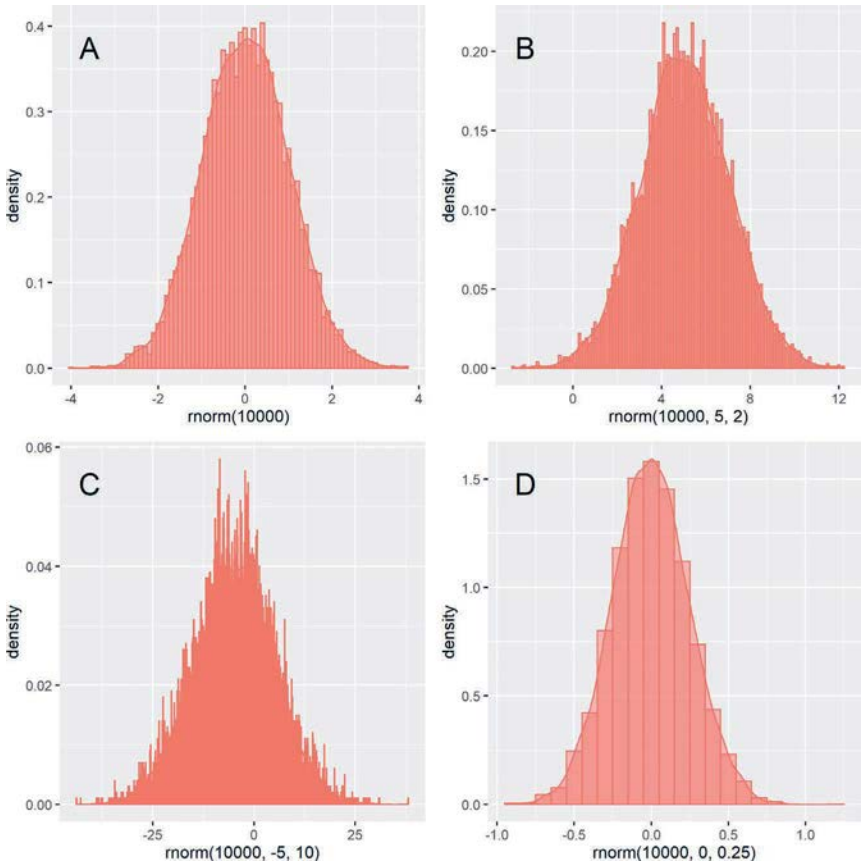
Wie sie anhand der Abbildung 10.6 vermuten können, unterscheiden sich die Sentiment-Werte der 1*- und 5*-Rezensionen im Durchschnitt. Um auf Nummer sicher zu gehen, werden wir, wie weiter oben angekündigt, einen Mittelwertdifferenztest durchführen.

Um diese Unterschiede herauszuarbeiten, bietet uns die Statistik eine Reihe von Tests an, mit denen wir zwei oder mehr Gruppen miteinander vergleichen. Um einfache Gruppenvergleiche durchzuführen, können Sie auf t-Test, Welsh-Tests oder Wilcoxon-Tests zurückgreifen, sofern die Annahmen für die einzelnen Tests erfüllt sind (Fay und Proschan 2010). Alle drei Tests prüfen, ob gemessene Werte zwischen Gruppen im Durchschnitt unterschiedlich sind. In unserem Falle: Ob die gemessenen Sentiments von 1*- und 5*-Bewertungen auf Amazon im Durchschnitt verschieden sind. Sowohl t-Test als auch Welsh-Test benötigen dabei metrische Variablen, der Wilcoxon-Test hingegen ordnet die Werte Rängen zu, wobei die Rangnummer von niedrigen Werten (z. B. Sentiment −5 = Rang 1) bis hohen Werten (Sentiment +5 = höchster Rang) reicht. Da wir für das Verständnis der t- und Welsh-Tests das Vorliegen einer Normalverteilung unserer Variablen testen wollen, müssen wir Ihnen kurz demonstrieren, wie eine solche Normalverteilung aussieht (Abbildung 10.7).

Panel A aus Abbildung 10.7 zeigt eine Standardnormalverteilung mit einem Mittelwert von 0 und einer Standardabweichung von 1, Panel B eine mit Mittelwert von 5 und Standardabweichung von 2, Panel C eine mit Mittelwert von −5 und Standardabweichung von 10 und Panel D eine mit Mittelwert = 0 und Standardabweichung von 0,25. Sie sehen hieran, dass es immer einen mehr oder minder ausgeprägten Gipfel gibt, der beim Mittelwert (auch Erwartungswert bei Zufallsstichproben) liegt. Je größer die Standardabweichung, also die Streuung um den Mittelwert, ist, desto weiter gruppieren sich die potenziellen Beobachtungswerte um diesen Gipfel herum, wie Sie anhand der verschiedenen X-Achsen der Panels sehen können. Die Kurve repräsentiert die Konzentration bestimmter Werte in bestimmten Abschnitten in Ihrer Verteilung (z. B. zwischen 0 und 0,1 auf der X-Achse). Die Logik bei Gruppenvergleichen ist, dass es, je weiter am Rand dieser Verteilung die Differenz Ihrer beiden Gruppen liegt, desto unwahrscheinlicher ist, dass die beiden Mittelwerte gleich sind. Das wird durch die niedrigen Werte an diesen Stellen verdeutlicht.

Kommen wir nun wieder zu den Eigenschaften der Gruppenvergleichstests zurück. Die Durchführung von t-Tests ist am voraussetzungsvollsten. Um einen t-Test durchzuführen, müssen die von Ihnen getesteten Variablen metrisch sein, bei beiden Gruppen müssen die Werte einer Normalverteilung folgen und Varianzen der Gruppen, d. h. die Streuung um den Mittelwert, muss bei beiden Gruppen gleich sein. Um einen Welsh-Test durchzuführen, muss die getestete Variable metrisch sein und die Streuung um die Mittelwerte beider Gruppen

Abbildung 10.7 Exemplarische Darstellungen von verschiedenen Normalverteilungen



(z. B. durchschnittliches Sentiment) muss einer Normalverteilung folgen. Die Varianzen zwischen den beiden Gruppen müssen nicht gleich sein. Sollte aber eine der ersten beiden Annahmen verletzt sein, dann können Sie noch immer den Wilcoxon-Test durchführen, da dieser ohne diese Annahmen auskommt. Um die Annahme der Normalverteilung der Werte in beiden Gruppen zu testen, können Sie den Shapiro-Wilk-Test nutzen. Um unsere Gruppen auf Varianzgleichheit zu testen, können Sie den Levene-Test durchführen.

Ganz grundsätzlich geht der Shapiro-Wilk-Test davon aus, dass bei den Werten, die wir betrachten, eine Normalverteilung vorliegt. Diese Annahme, die wir statistisch testen, ist unsere Nullhypothese (H0), ergo, dass unsere Sentiment-Werte in der jeweiligen Gruppe normalverteilt sind. Die Gegenhypothese (H1) hierzu lautet, dass unsere Sentiment-Werte nicht normalverteilt sind. Dass die

Hypothesen so formuliert wurden, hat den Grund, dass wir die Wahrscheinlichkeit möglichst geringhalten wollen, dass wir fälschlicherweise von einer Normalverteilung der Variablen ausgehen, d. h. die Nullhypothese fälschlicherweise ablehnen. Dabei handelt es sich um den Alpha-Fehler. Schlimmer wäre ein Beta-Fehler, d. h. hier, dass wir fälschlicherweise die Nullhypothese ablehnen würden. Wenn wir fehlerhafterweise davon ausgehen, dass keine Normalverteilung der Sentiments vorliegt, dann können wir noch immer den Wilcoxon-Test durchführen. Umgekehrt würden wir fälschlicherweise einen t-Test oder Welsh-Test durchführen und zu falschen Schlussfolgerungen kommen. Idealerweise sollte dieser Wert möglichst gering, höchstens aber bei 0.05 liegen, damit Sie einigermaßen sicher sind, dass das Sentiment keiner Normalverteilung folgt.[9]

Der Shapiro-Wilk-Test lässt sich in RStudio durch Eingabe des Befehls `shapiro.test([VARIABLE])` durchführen. Ersetzen Sie hier `[VARIABLE]` durch den Variablennamen, bei der Sie feststellen möchten, ob sie normalverteilt ist. Wenn wir feststellen wollen, ob unsere Sentiment-Werte pro Rezension einer Normalverteilung folgen, dann könnten wir `shapiro.test(mean_sentiment_per_star$sentiment)` eingeben und erhielten unsere Werte. Nun wollen wir aber die Werte für beide Gruppen prüfen. Also fügen wir den Befehl, um diesen Test in RStudio durchzuführen, wieder in den `tapply()`-Befehl ein, markieren die Zeilen aus Code 10.24 und drücken „Strg + Enter“, um den Befehl auszuführen.

Code 10.24 Durchführung eines Shapiro-Wilk-Tests für 1*- und 5*-Bewertungen, um eine vorliegende Normalverteilung zu prüfen

```
## Shapiro-Wilk-Test
tapply(mean_sentiment_per_star$sentiment,
       mean_sentiment_per_star$overall,
       shapiro.test)
```

Output 10.5 Ausgabe des Shapiro-Wilk-Tests für 1*- und 5*-Bewertungen (Fortsetzung nächste Seite)

```
> ## Shapiro-Wilk-Test
> tapply(mean_sentiment_per_star$sentiment,
+        mean_sentiment_per_star$overall,
```

9 Der Signifikanzwert, auch p-Wert genannt, zeigt dabei die Wahrscheinlichkeit an, die Nullhypothese fälschlicherweise anzunehmen. Damit zeigt er zugleich die Wahrscheinlichkeit für einen Alpha-Fehler an. Ist dieser Wert bei 0.05, dann bedeutet dies, dass wir in 5 % der Fälle fälschlicherweise die Nullhypothese ablehnen.

```
+        shapiro.test)
$`1`

        Shapiro-Wilk normality test

data:  X[[i]]
W = 0.99312, p-value = 4.34e-12


$`5`

        Shapiro-Wilk normality test

data:  X[[i]]
W = 0.98627, p-value < 2.2e-16
```

Der Output ist in zwei Gruppen geteilt, die mit `$`1`` und `$`5`` unterteilt sind. Wir betrachten hier die p-Werte, die mit $4.34*10^{-12}$ für die Rezensionen mit 1* und $2.2*10^{-16}$ für die Rezensionen mit 5* sehr gering sind. Das e in der Ausgabe zeigt Ihnen an, dass die Zahlen hiernach im Exponent stehen. Zur Erinnerung: H0 lautet, dass die Sentiment-Werte normalverteilt sind; das wäre hier wünschenswert gewesen, weil wir dann den T- oder Welsh-Test anwenden könnten. Unsere Tests sagen aber aus, dass es unwahrscheinlich ist, eine Normalverteilung vorliegen zu haben. Damit ist die erste Annahme verletzt und wir könnten nurmehr einen Wilcoxon-Rangsummentest durchführen, um Unterschiede in den Durchschnittswerten beider Gruppen zu ermitteln.

Gehen wir allerdings an dieser Stelle davon aus, dass die Normalverteilung unserer Sentiments eingehalten wird. Dann müssten wir die beiden Gruppen, die wir testen wollen, auf Varianzgleichheit testen. Varianzgleichheit bzw. Varianzhomogenität können wir mittels des Levene-Tests prüfen. Er folgt prinzipiell der gleichen Logik wie der Shapiro-Wilk-Test und geht davon aus, dass die Varianzen zwischen zwei Gruppen gleich sind (H0) bzw. ungleich sind (H1). Der Levene-Test ist im `car`-Paket gespeichert. Sie können ihn mittels `leveneTest()` aufrufen, müssen hier aber zunächst Ihr Modell spezifizieren und dann Ihren Datensatz mit der Option `data =` an den Befehl übergeben. In unserem Fall lautet die Modellspezifikation `sentiment ~ overall`, also wurde zunächst die Variable angegeben, in der die metrischen Werte sind, auf deren Basis die Varianzen berechnet werden, ehe durch eine Tilde (Alt Gr + Sterntaste) abgetrennt die Variable angegeben wird, in der wir die Unterscheidung in Gruppen, d. h. 1*-Rezensionen und 5*-Rezensionen, getroffen haben. Die Tilde ~ signalisiert in R, dass hier Variablen zugeordnet werden. Der Befehl ist vollständig in Code 10.25 aufgeführt und Sie erhalten Output 10.6, wenn Sie diese Zeilen markieren und ausführen.

Code 10.25 Durchführung des Levene-Tests, um Varianzgleichheit zwischen 1*- und 5*-Rezensionen zu prüfen

```
## Levene-Test
leveneTest(sentiment ~ overall,
  data = mean_sentiment_per_star)
```

Output 10.6 Ausgabe des Levene-Tests für Varianzgleichheit zwischen 1*- und 5*-Rezensionen

```
Levene's Test for Homogeneity of Variance (center = median)
        Df F value   Pr(>F)
group    1  16.407 5.16e-05 ***
      7586
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Die für Sie wichtigste Ausgabe ist im Feld `Pr(>F)`, das in unserem Falle als Wahrscheinlichkeit gedeutet werden kann, dass die Varianzen nicht gleich sind, aber irrtümlicherweise für gleich gehalten werden. Dieser Wert ist wieder sehr gering. Die drei Sterne rechts von dem Wert $5.16*10^{-5}$ zeigen an, dass Sie in weniger als einem von 1000 Fällen fälschlicherweise von einer Varianzgleichheit zwischen 1*- und 5*-Rezensionen ausgehen würden. Diese Irrtumswahrscheinlichkeiten finden Sie unten in der Ausgabe in der Zeile, die mit `Signif. codes` beginnt. Beide Tests deuten somit darauf hin, dass wir lediglich einen Wilcoxon-Rangsummentest verwenden können, um die beiden Gruppen miteinander zu vergleichen. Der Vollständigkeit halber werden wir aber auch die Befehle für den t-Test und den Welsh-Test aufführen, aber nur die Ausgabe des Wilcoxon-Tests interpretieren.

Jeder der drei Tests folgt dem gleichen Aufbau wie der Levene-Test. Das bedeutet, dass Sie erst einmal ein Modell spezifizieren und dann den Datensatz übergeben müssen, in dem die Variablen für Ihren Test enthalten sind. Dabei können Sie sowohl den t-Test als auch den Welsh-Test mit dem Befehl `t.test()` in RStudio durchführen. Die einzige Option, die beide Tests voneinander trennt, ist die `var.equal`-Option. Wird hier `FALSE` angegeben, dann führt RStudio einen Welsh-Test durch, sonst wird ein t-Test durchgeführt. Den Wilcoxon-Rangsummentest führen Sie einfach mit `wilcox.test()` aus (Code 10.26).

Code 10.26 Durchführung von t-Test, Welsh-Test und Wilcoxon-Rangsummentest in R

```
## Mittelwertdifferenztests
#t-Test
t.test(sentiment ~ overall,
       data = mean_sentiment_per_star)

#Welsh-Test
t.test(sentiment ~ overall,
       data = mean_sentiment_per_star, var.equal=FALSE)

#Wilcoxon-Rangsummentest
wilcox.test(sentiment ~ overall,
            data = mean_sentiment_per_star)
```

Wenn Sie nun die Codezeilen markieren, in dem der `wilcox.test()`-Befehl enthalten ist und diese ausführen, erhalten Sie Output10.7.

Output 10.7 Ergebnis des Wilcoxon-Rangsummentests

```
> wilcox.test(sentiment ~ overall,
+             data = mean_sentiment_per_star)

        Wilcoxon rank sum test with continuity correction

data:  sentiment by overall
W = 2972437, p-value < 2.2e-16
alternative hypothesis: true location shift is not equal to 0
```

Hier sehen Sie auch wieder den p-Wert, der mit $2.2*10^{-16}$ sehr gering ist. Zudem wird Ihnen ein Text ausgegeben, in dem ausgesagt wird, dass die 1*- und 5*-Rezensionen systematische Rangunterschiede aufweisen. In Kombination mit den deskriptiven Statistiken und Abbildung 10.6 können wir nun mit einer hohen Sicherheit davon ausgehen, dass 5*-Rezensionen systematisch positiver sind bzw. positiveres Vokabular nutzen, als dies im Falle der 1*-Rezensionen der Fall ist.

10.3 Sentiment-Analyse in Python

Wenden wir uns nun der Sentiment-Analyse in Python zu. Python ist eine einsteigerfreundliche Programmiersprache, die auf Einfachheit und Übersichtlichkeit der Programmierung ausgelegt ist. Sie ist weit verbreitet und bietet Pakete wie beispielsweise `scikit-learn` an, die auch unerfahrenen Nutzer*innen den Einstieg in Techniken des maschinellen Lernens erleichtern (Hao und Ho 2019). Sie zählt zu den am schnellsten wachsenden Programmiersprachen, sowohl was die Anzahl der Pakete mit Befehlen als auch was die Nutzer*innenzahlen angeht (Srinath 2017). Aufgrund der großen Community, der Einsteigerfreundlichkeit und der Übersichtlichkeit von Python lohnt es sich, gemeinsam einen Blick in diese Programmiersprache zu werfen und diese erst am Beispiel der Sentiment-Analyse und dann des Topic Modelings kennenzulernen.

10.3.1 Python, Spyder und Packages

Ähnlich wie bei R benötigen wir sowohl ein Programm, in dem wir Python ausführen, als auch eine grafische Oberfläche, die Ihnen die Bedienung erleichtert. Als Nutzeroberfläche werden wir Spyder verwenden, aber es gibt auch andere Nutzeroberflächen wie Pydev oder Pycharm, die Sie ebenfalls verwenden können. Hierfür werden wir zunächst die Anaconda-Distribution von Python herunterladen und im Anschluss Spyder. Zur Unterscheidung: Python ist die Programmiersprache, die wir verwenden werden, Anaconda die Arbeitsumgebung, mit deren Hilfe Sie Ihre Programmprojekte und Programmpakete verwalten können, und Spyder ist die grafische Oberfläche (ähnlich wie RStudio).

Um Python und Anaconda zu installieren, gehen Sie zunächst einfach auf *www.anaconda.com* und wählen oben im Reiter „Products“ an. Im neuen Fenster, das sich öffnet, wählen Sie die kostenfreie Individual Edition und steuern den Download Link an, der sich auf der Seite befindet. Hier wird die Edition für Windows-Nutzer*innen besonders hervorgehoben. Darunter sehen Sie aber noch die Fläche mit dem Titel *Get Additional Installers.* Hier wählen Sie den Apfel an, wenn Sie einen Apple-Rechner besitzen und Tux (den Pinguin), wenn Sie ein Linux-Betriebssystem (z. B. Kubuntu) verwenden. Laden Sie nun Anaconda herunter und starten Sie die Installationsdatei.

Nachdem Sie Anaconda installiert haben, fahren Sie mit Spyder fort. Hierfür gehen Sie auf *www.spyder-ide.org* und scrollen entweder die Seite weiter herunter oder klicken oben im Reiter auf „Download“. Dann werden Sie zum Seitenabschnitt weitergeleitet, auf dem sich die Installationsdateien befinden. Hier finden Sie sofort den „Download“-Button für die Windows Installation. Wenn Sie Mac-Nutzer*in sind, dann können Sie auf den Link *installation instructions* gehen und dort die Mac-Version herunterladen. Diese Anleitung finden Sie auch

unter *https://docs.spyder-ide.org/current/installation.html.* Linux-User*innen gehen in die Console und geben `sudo apt install spyder3` ein. Wenn Sie die Installationsdatei heruntergeladen haben, dann führen Sie diese aus und befolgen die Installationsschritte. Danach können Sie über das Startmenü Spyder starten.

Neben Spyder wurde Ihnen aber auch eine eigene Konsole installiert, das sogenannte `Anaconda prompt`. Bei einer Konsole handelt es sich um eine textbasierte Schnittstelle zwischen dem Computer und Ihrem Betriebssystem, in unserem Falle der Programmiersprache Python. Das `Anaconda prompt` wartet dabei mit eigenen Befehlen zur Verwaltung Ihrer Python-Programmierumgebung und Ihrer Programmpakete auf. Hierüber können Sie später Pakete herunterladen und updaten. Wenn Sie beispielsweise Pakete installieren oder updaten, d. h. auf die neueste Version bringen möchten, dann bietet Ihnen die Konsole zwei Möglichkeiten. Einmal mittels `pip` *(package installer for python)* und einmal mittels der Eingabe von `conda`. Wir benötigen auf jeden Fall das Paket `vaderSentiment` (VADER = Valence Aware Dictionary and Sentiment-Reasoner) (Hutto und Gilbert 2014) für die Sentiment-Analyse englischer Texte und `seaborn` (Waskom 2021) zur Visualisierung von Grafiken. Diese beiden Pakete werden – ebenso wie die anderen verwendeten Pakete – weiter unten gesondert erklärt.

Gehen Sie nun auf Ihr Startmenü (Windows-Symbol bzw. Apfel-Symbol beim Mac) und geben „Anaconda Prompt“ in die Suche ein. Klicken Sie auf das Symbol, um die Anaconda-Konsole zu öffnen. In Linux entfällt dies, weil Sie die folgenden Befehle auch über die systemeigene Konsole starten können und Linux-Betriebssysteme in der Regel über eine vorinstallierte Python-Version verfügen. Geben Sie nun zur Installation des ersten Paketes einen der folgenden Befehle ein.

Code 10.27 Installation von Paketen im Anaconda prompt am Beispiel des vaderSentiment-Paketes

```
pip install vaderSentiment
conda install vaderSentiment
```

Wenn Sie hingegen Pakete auf die neueste Version bringen möchten, dann können Sie dies wie in Code 10.28 in Ihrer Konsole durchführen, müssen aber dann das jeweilige Paket auswählen.

Code 10.28 Update von Paketen im Anaconda prompt am Beispiel des vaderSentiment-Paketes

```
pip install vaderSentiment --upgrade
conda upgrade vaderSentiment
```

Sie sehen hier den Unterschied in der Befehlsstruktur für die Anaconda Konsole. Im Falle von `pip` können Sie mit zwei Minuszeichen Optionen eingeben, während Sie im Falle des Anaconda-Befehls `install` durch `upgrade` austauschen können. `Pip` hat den Vorteil, dass es die Pakete weitaus schneller als `conda` installiert, Sie aber auf `conda` zurückgreifen können, falls der Installationsweg über `pip` nicht funktioniert.

10.3.2 Spyder-Benutzeroberfläche

Die Benutzeroberfläche von Spyder teilt sich ähnlich wie bei RStudio in mehrere Bereiche auf (siehe Abbildung 10.8). Dazu gehören die Menüs (1), der Editor (2), der Variable Explorer (3) und die Ipython Console (4). Da es sich bei Python im Gegensatz zu R um eine Programmiersprache handelt, die nicht für Statistiker*innen erstellt wurde, weisen sowohl die Menüs als auch der Variable Explorer und die Ipython Console mehr Optionen auf, die ganz spezifisch auf Programmierung und deren Ausbesserung ausgelegt sind.

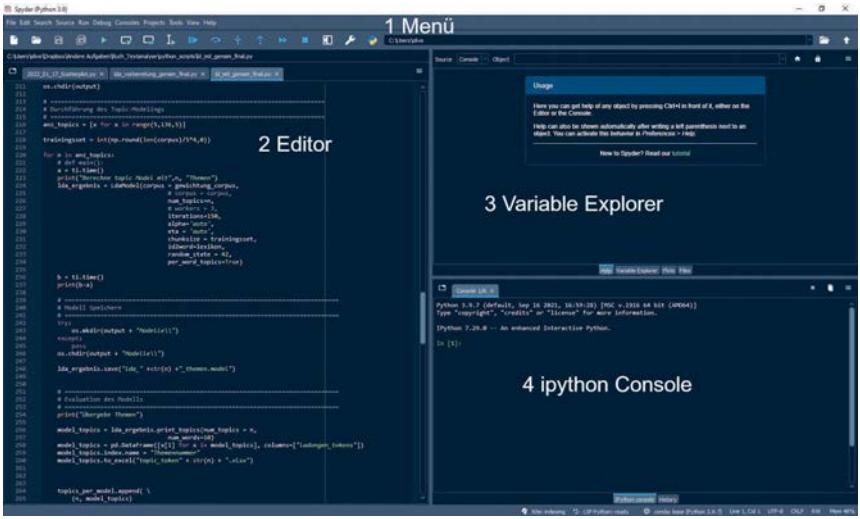
Beginnend bei den Menüs (1) sieht man eine ähnliche Aufteilung wie bei RStudio in Reiter und Buttons. Im Reiter finden sich die ausklappbaren Schaltflächen „File“, „Edit“, „Source“, „Run“, „Debug“, „Consoles“, „Projects“, „Tools“, „View“ und „Help“.[10] Wie bereits bei RStudio gehen wir die Reiter und Buttons von links nach rechts durch.

Unter dem Menüpunkt „File“ können Sie neue Python-Skripte erstellen, öffnen, speichern oder zwischen den einzelnen, von Ihnen bereits geöffneten Skripten hin- und herwechseln. Der Unterpunkt „New file“ erstellt ein neues Python-Skript. Daneben können Sie eine beliebige Datei (vorzugsweise aber Python-Skriptdateien mit der Endung .py) mit „open“ öffnen, während „open last closed“ die letzte von Ihnen in Spyder geschlossene Skriptdatei wieder öffnet. Darunter befindet sich ein sich seitlich öffnender Reiter mit dem Namen „open recent“, mit dessen Hilfe Sie eine von Ihnen definierbare Anzahl an zuletzt geöffneten Python-Skripts auflisten und wieder öffnen können.[11] Hierunter folgen „print preview“ und „print“. Falls Sie sich entschließen, Ihr Skript zu drucken, dann zeigt Ihnen „print preview“ eine Vorschau an, wie Ihr Skript auf dem Papier aussehen würde, während „print“ direkt zum Druckdialog Ihres Druckers weiterleitet. Der Menüpunkt „File switcher“ öffnet ein Suchfenster und bietet Ihnen die Möglichkeit, zwischen den von Ihnen geöffneten Python-Skripten hin- und

10 Wenn Sie die Alt-Taste drücken, sehen Sie, dass Buchstaben im Menü unterstrichen werden. Wenn Sie die entsprechende Taste im Anschluss drücken (z. B. F für File), dann öffnen Sie bequem per Tastatur die Menüs und können in Ihnen navigieren.

11 In diesem Untermenü finden Sie auch den Punkt „maximum number of recent files“, der es Ihnen ermöglicht, die Anzahl der Dateien, die angezeigt werden und zuletzt geöffnet waren, zu verändern.

Abbildung 10.8 Aufbau der grafischen Oberfläche von Spyder



herzuwechseln. Dies können Sie auch in der Leiste oberhalb des Editors machen, in welchem Sie die einzelnen Python-Skripte geöffnet haben. Sollten Sie aber ein großes Programmierprojekt begonnen haben und den Überblick über Ihre Skripte verloren haben, dann können Sie es mit dieser Funktion einfacher wiederfinden. Zuletzt können Sie mit „Restart" Spyder neu starten und mit einem Klick auf die „Quit"-Schaltfläche beenden.

Daneben gibt es den „Edit"-Reiter. In diesem finden Sie Schaltflächen, mit denen Sie Arbeitsschritte rückgängig machen oder wiederholen können *(undo* und *redo)*. Daneben gibt es hier die Möglichkeit, markierten Code *(cut)* auszuschneiden bzw. zu kopieren *(copy)* und an eine andere Stelle im Skript einzufügen *(paste)*. Mit der Schaltfläche „select all" können Sie zudem Ihren ganzen Code markieren. Ferner werden Möglichkeiten zur Kommentierung Ihres Codes aufgelistet. So ermöglicht Ihnen die Schaltfläche „comment/uncomment" eine oder mehrere angewählte Codezeilen zu kommentieren bzw. auszukommentieren. Auskommentieren meint, dass Sie Befehle mittels eines am Beginn der Zeile gesetzten Kommentarzeichens (hier: Raute / #) Ihrem Programm (hier: Python) angeben, dass der darauf folgende Befehl nicht auszuführen ist. Das ist hilfreich, wenn Sie beispielsweise Fehler im Programmcode beseitigen möchten oder für einen Arbeitsschritt einen Befehl gerade nicht benötigen. Das ist insbesondere dann sinnvoll, wenn dieser Arbeitsschritt, wie im Falle des Topic Modeling in Kapitel 11, Stunden, Tage oder sogar Wochen für die Berechnung benötigt.

Wie bei RStudio gilt, dass alles in einer Zeile, was einer Raute folgt, nicht ausgeführt wird, wenn Sie den Code durchlaufen lassen. Wie bei R können Sie

in Python einen Kommentar auch manuell durch das Hinzufügen eines Raute-Zeichens # beginnen. Daneben gibt es aber noch die Optionen „Add block comment“ und „Remove block comment“. Ersteres erstellt einen mehrzeiligen Kommentar mit spezieller Formatierung und Letzteres entfernt diesen Kommentar, dessen Struktur wie in Code 10.29 aussieht.

Code 10.29 Im Spyder-Editor eingefügter Kommentarblock

```
# ============================================================================
#
# ============================================================================
```

Wie Sie sehen, werden die Zeichen so angeordnet, dass es wie eine Kapitelüberschrift wirkt. Dies erleichtert es Ihnen, sich in Ihrem Code zurechtzufinden, wenn Sie ein längeres Skript schreiben und Ihren Code in verschiedene Sinnabschnitte (wie z. B. Daten laden, Daten bereinigen, Sentiment-Analyse durchführen) unterteilen.

Die Suchen- und Ersetzen-Funktionen sind im dritten Reiter „Search“ aufgelistet. Dabei können Sie sowohl in einem einzelnen geöffneten Code suchen, aber auch gezielt nach Dateien suchen, die in dem aktuell von Ihnen definierten Dateipfad gespeichert sind *(Find in files)*.

Daneben finden Sie den „Source“-Reiter, in dem Optionen zur Anzeige und Verbesserung Ihres im Editor (2) aufgeführten Python-Skripts aufgelistet sind. Daneben findet sich der „Run“-Reiter, indem die Möglichkeiten aufgeführt sind, mit deren Hilfe Sie Ihren gesamten Code bzw. Codezeilen ausführen können. Hier finden Sie ganz oben die Schaltfläche „run“, mit dessen Hilfe Sie das Gesamtskript ausführen können. Darüber hinaus gibt es „run cell“, mit dessen Hilfe Sie eine von Ihnen definierte Codezelle ausführen können. Diese grenzen Sie mit der Zeichenfolge **#%%** voneinander ab, was die gezielte Ausführung einer zusammenhängenden Programmcodezeile ermöglicht. „Run cell and advance“ bedeutet, dass Sie eine von Ihnen angewählte Codezelle ausführen, Ihr Cursor dann automatisch zur nächsten definierten Zelle springt. „Re-run last cell“ führt eine Datenzelle aus, die in Ihrem Skript direkt vor Ihrer Datenzelle kommt, die Sie mit Ihrem Cursor angewählt haben. Wir kommen hierauf zurück, wenn wir den Code für die Sentiment-Analyse Schritt für Schritt aufbauen. Daneben gibt es die Möglichkeit, eine Codezeile oder von Ihnen ausgewählte Codeabschnitte (linksklick gedrückt halten und dann mit der Maus über die Codezeilen ziehen und dann die linke Maustaste loslassen) durch die Auswahl von „run selection or current line“ auszuführen.

Daneben befindet sich der „Debug“-Reiter, in dem Funktionen aufgelistet sind, die Ihnen bei der Fehlersuche und Fehlerbehebung Ihres Codes behilflich

sind, „Consoles“, mit dem Sie neue Ipython Consolen (4) öffnen oder schließen können, um Codes parallel auszuführen. Es folgt der „Projects“-Reiter, in dem Sie neue Coding-Projektordner auf Ihrer Festplatte anlegen, zwischen diesen hin- und herwechseln oder diese löschen können. Unter Tools können Sie einerseits das Aussehen von Python festlegen, andererseits Arbeitspfade festlegen und bei Bedarf wechseln, in denen Sie z. B. weitere Pakete oder Python-Skripte gespeichert haben. Unter View finden Sie Optionen, mit denen Sie die Python-Oberfläche nach Ihren Bedürfnissen anpassen können. Zuletzt finden Sie unter dem „Help“-Reiter den Zugang zur Online-Hilfe sowie Tutorials, mit deren Hilfe Sie zentrale Funktionen von Python oder bestimmte Aspekte der Programmierung in Python nochmals nachvollziehen können.

Darunter befinden sich einige Buttons, die ebenfalls von links nach rechts erläutert werden. Die einzelnen Funktionsbereiche, auf die die Buttons Bezug nehmen, werden dabei durch zwei senkrechte Striche abgetrennt. Ganz links findet sich ein Dokument mit einer oben rechts befindlichen Falte. Wenn Sie auf diesen Button klicken, können Sie ein neues Skriptdokument erstellen. Das rechts daneben befindliche Aktenordnersymbol ermöglicht Ihnen das Öffnen von Dateien. Das Diskettensymbol und Diskettensymbole rechts daneben ermöglichen Ihnen, eine bzw. alle geöffneten Skripte auf einmal zu speichern. Der Listen-Button rechts daneben ist die grafische Repräsentation des „file switchers“. Das heißt, dass Sie zwischen Ihren einzelnen geöffneten Python-Codedateien hin- und herwechseln und nach einzelnen geöffneten Codedateien suchen können, wenn Sie diesen Button betätigen. Der @-Button rechts daneben lässt Sie nach spezifischen Symbolen in den von Ihnen geöffneten Dateien suchen.

Das grüne Dreieck führt den Code Ihres gesamten Skriptes aus, während das nächste Symbol, das einen grünen Kreis mit einem senkrechten Strich vor einem durchsichtig-gelben Hintergrund darstellt, eine Codezelle ausführt. Das Symbol rechts daneben (das einen roten, eckigen Pfeil beinhaltet) führt die aktuell ausgewählte Codezelle aus und springt dann zur nächsten Zelle. Ein Linksklick auf den danebengelegenen weißen Pfeil mit dem Schriftzeichen (sieht in etwa aus wie eine römische Eins) führt die von Ihnen angewählten Codezeilen aus. Der weiße, einen dreiviertel Kreis ausfüllende Pfeil mit zusätzlichem grünen Pfeil am unteren rechten Ende führt die Codezelle aus, die sich vor der von Ihnen angewählten Codezelle befindet.

Die nun folgenden, blauen Symbole sind zum Debuggen Ihres Python-Codes. Das bedeutet, dass Sie einen Modus aktivieren, in dem Ihre Codezeilen mit zusätzlichen Funktionen (wie z. B. Stopppunkten, an denen der Durchlauf Ihres Codes unterbrochen wird) ausgeführt werden. Danach folgen Buttons, mit deren Hilfe Sie Spyder konfigurieren können, sowie ein Dateibrowser, mit dessen Hilfe Sie den aktuellen Arbeitspfad angeben können.

Darüber hinaus können Sie Funktionen, die in den Buttons und in den Reitern „File“, „Edit“ und „Source“ aufgeführt sind, auch bequem durch das Drü-

Tabelle 10.1 Übersicht über die wichtigsten Tastenkürzel, die Ihnen die Arbeit mit Ihrem Skript erleichtern

Funktion	Tastenkombination
Onlinehilfe aufrufen	F1
Neue Skriptdatei erstellen	Strg + N
Skriptdatei öffnen	Strg + O
Datei speichern	Strg + S
Datei unter neuem Namen speichern	Strg + Shift + S
Arbeitsschritt rückgängig machen	Strg + Z
Arbeitsschritt nochmals durchführen	Strg + Shift + Z
Ausschneiden	Strg + X
Kopieren	Strg + C
Einfügen	Strg + V
Alles auswählen	Strg + A
Skriptzeile kommentieren/auskommentieren	Strg + 1
Kommentarblock hinzufügen	Strg + 4
Kommentarblock entfernen	Strg + 5
Gesamtes Skript ausführen	F5
Zeilenauswahl ausführen	F9
In einer Programmierzelle befindliche Programmzeilen ausführen	Strg + Enter

cken von Tastenkombinationen ausführen. Die wichtigsten für Ihren Einstieg in Python sind in Tabelle 10.1 aufgeführt.

Wie in RStudio hat der Editor die Funktion, Code zu schreiben und auszuführen. Den größten Teil des Bildschirms nimmt dabei der Editor selbst ein, darüber sehen wir die geöffneten Python-Dateien. Mithilfe des kleinen Aktenordnersymbols links von den Dateien können Sie durch die geöffneten Dateien browsen und die Codedatei anwählen, die Sie bearbeiten wollen. Rechts davon finden sich Buttons mit Pfeilen, mit deren Hilfe Sie (wenn Sie viele Python-Codedateien geöffnet haben) manuell nach den jeweiligen Dateien suchen können. Ganz rechts befindet sich ein Button mit drei Strichen. Wenn Sie auf diesen Button klicken, dann öffnet sich ein Menü, mit dessen Hilfe Sie die Fenster noch weiter konfigurieren können (z. B., ob Sie zwei Fenster mit Code parallel geöffnet haben möchten).

Der Variable Explorer (3) vereint – RStudio sehr ähnlich – mehrere Fenster, mit deren Hilfe Sie sich in den Daten beim Programmieren zurechtfinden kön-

nen. Hier sehen Sie am unteren Rand dieses Teilfensters eine Leiste, auf der „Variable explorer", „Help", „Profiler", „Code analysis", „Plots", „Files" und „Online help" steht.

Der „Variable explorer" bietet Ihnen eine grafische Übersicht über die Objekte,[12] die Sie in Python geladen und definiert haben. Die Ansicht ist wie eine Art Excel-Datentabelle aufgebaut und listet den Namen Ihrer Objekte/Daten *(name),* deren Typ *(type),* Größe *(size)* und Werte *(value)* auf. Name und Typus (z. B. DataFrame, String, Integer) sind selbsterklärend. Die Spalte Size zeigt Ihnen an, wie viele Elemente in einem Objekt gespeichert sind – zum Beispiel 1, wenn es sich um genau eine Zeichenkette oder Zahl handelt, 3, wenn es sich um eine Liste mit drei Elementen handelt. Wenn hier Zahlen in runden Klammern auftauchen, zum Beispiel (100,9), dann heißt das, dass das betreffende Objekt über 100 Zeilen und neun Spalten verfügt. Die dazugehörigen Werte werden (in Auszügen) in der „value"-Spalte angezeigt. Diese werden zusätzlich mit unterschiedlichen Farben unterlegt, wenn es sich um unterschiedliche Objekttypen handelt. Oberhalb dieser Auflistung sehen Sie wieder eine Leiste mit Buttons. Der ganz linke mit einem Pfeil, der auf eine graue Zeile zeigt, ermöglicht Ihnen das Laden von Data-Frames, z. B. einer csv-Datei, Code oder einzelnen Texten. Das Diskettensymbol rechts daneben ermöglicht Ihnen das Speichern von Dateien; das Diskettensymbol mit Stift darunter erlaubt das Speichern von Dateien mit veränderten Namen. Das Radiergummisymbol rechts daneben erlaubt es Ihnen, eine Auswahl an Objekten und Variablen zu löschen. Wenn Sie auf das Lupensymbol klicken, dann erscheint eine Suchzeile am unteren Rand des Variable Explorers. Hier können Sie Variablennamen oder Variablentypen in einer Suchzeile suchen. Spyder blendet dann automatisch die Variablennamen und Objekttypen aus, die nicht mit der Suchanfrage übereinstimmen. Der kreisförmige Pfeil erneuert zuletzt die Variablenansicht.

Das „Help"-Fenster bietet Ihnen, wie der Name schon sagt, Hilfe zu den einzelnen Objekten an, die Sie im Editor oder in der Console befindlich sind. Anders als bei R können Sie hier durch das Markieren von Objekten und das Drücken der Tastenkombination „Strg + I" die Hilfe aufrufen. Alternativ geben Sie oben unter Source an, ob das Objekt, zu dem Sie Hilfestellung erfragen möchten, im

12 Bei einem Objekt handelt es sich in Python um eine Datenstruktur oder eine Ausgabe einer Funktion mit bestimmten im Programmcode vorgegebenen Eigenschaften. Ein Objekt in Python können Sie sich wie ein physisches Objekt, beispielsweise einen Stuhl, vorstellen. Ein Stuhl (Objekt) hat in der Regel vier Beine (Eigenschaft), man kann auf ihm sitzen (Funktion), hat eine Lehne (Eigenschaft), eine Farbe (Eigenschaft) und eine Größe (Eigenschaft). Darüber hinaus stehen Objekte im Verhältnis zueinander, wodurch deren Eigenschaften mitbestimmt werden können. So steht ein Stuhl im Verhältnis zu einem Tisch (Objekt mit eigenen Eigenschaften) und ist innerhalb eines Raumes (Objekt) positioniert. Dadurch, dass Objekte in Python auf diese Art strukturiert sind, folgen Sie einer baumartigen Ordnung, die Sie im übertragenen Sinn entlanggehen können, um auf immer feinere Eigenschaften Ihrer Objekte zuzugreifen und sie für weitere Analysen nutzbar zu machen.

Editor oder der Console geöffnet ist und geben dessen Namen anschließend in der Suchleiste „Object“ ein. Rechts finden Sie ein kleines, nach unten ausgerichtetes Dreieck. Wenn Sie hierauf klicken, dann öffnet sich eine Liste der Objekte, aus denen Sie das auswählen können, zu dem Sie Hilfe erhalten wollen.

Der Profiler ermöglicht es Ihnen, Python-Dateien zu laden und auszuwerten, wie lange Ihr Code für das Ausführen Ihrer Analyseschritte (oder Funktionen) benötigt und wie häufig die einzelnen Funktionen aufgerufen werden. Sie können zudem die Zeit, die Ihr Code benötigt, und die Anzahl der Funktionsaufrufe zwischen Python-Dateien vergleichen.

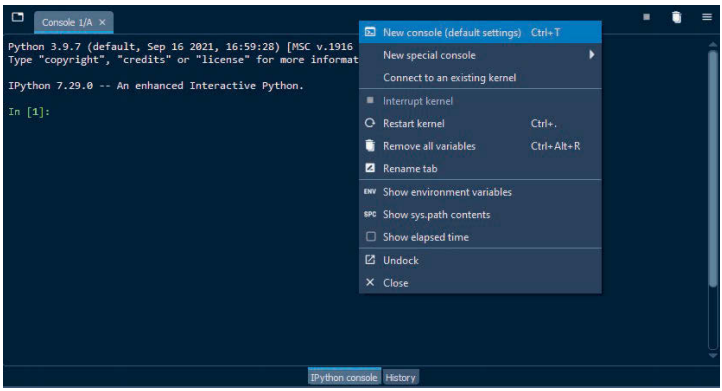
Der Reiter „Code analysis“ prüft, ob Sie Ihren Code lesbar und gemäß der Programmierkonventionen verfasst haben und zeigt Ihnen an, an welchen Stellen Sie Ihren Code nachbessern können, damit dieser übersichtlicher wird. Den Code selbst können Sie analysieren lassen, indem Sie entweder F8 drücken oder im „Code analysis“-Fenster den dreieckigen, grünen Button mit der Aufschrift „Analyze“ betätigen.

Im „Plot“-Fenster werden Ihnen – wie im RStudio auch – die Grafiken angezeigt. Sie können aber im Menü unter „Tools → Preferences“ (oder Schraubenzieher-Button) einstellen, ob Sie Ihre Grafiken in diesem Fenster, in der Konsole oder in einem separaten Fenster angezeigt haben möchten. Hierfür klicken Sie in Preferences auf „Ipython console“ und wählen im Punkt „Graphics backend“ beispielsweise Qt5 aus, wenn Sie die Grafiken in einem separaten Fenster öffnen wollen.

Das „Files“-Fenster zeigt Ihnen eine Übersicht über die angewählten und geöffneten Dateien und Ordner an, auf die Sie während Ihrer aktuellen Python-Sitzung zurückgreifen. Der „Online help“-Reiter bietet Ihnen einen Überblick über die von Ihnen installierten Pakete (hier *Modules* genannt) in Form einer Liste an. Sie können auf jedes Element dieser Liste klicken und erhalten Beschreibungen zu den einzelnen Paketen. Wenn Sie ein Paket angewählt haben, dann werden Ihnen entweder direkt Befehle angeführt, die in diesem Paketen enthalten sind, oder die Sammlungen von Funktionen, in denen wiederum Befehle gespeichert sind (z. B. *matplotlib* → *pyplot*).

Die Ipython Console (Abbildung 10.9) ist die unmittelbare Schnittstelle zwischen Ihrer grafischen Oberfläche und Ihrer Programmierumgebung. Hier können Sie Befehle eingeben und durch das Drücken von Enter ausführen. Zugleich gibt Ihnen die Ipython Console Fehlermeldungen und Mitteilungen aus, die Sie in Ihren Code inkludieren. Dies kann beispielsweise mittels des `print()`-Befehls geschehen. Die Ipython Console hat zwei Reiter, „Ipython console“ und „History“. Ersteres ist die aktive Schnittstelle, Letzteres bietet Ihnen eine Übersicht über die von Ihnen ausgeführten Codezeilen und Befehle. Ein Unterschied zu R ist, dass Sie mehrere *Consoles* gleichzeitig geöffnet haben können. Auf diese Weise können Sie mehrere Codes parallel ausführen. Diese Möglichkeit ist praktisch, wenn Sie beispielsweise aufwändige Topic Models berechnen, zugleich aber

Abbildung 10.9 Konsolenmenü und Verwaltung der Konsole in Spyder



noch weitere Daten aufbereiten wollen.[13] Wenn Sie hier ihre rechte Maustaste klicken, dann öffnet sich ein Menü, in dem Sie neue Konsolen *(New console)* oder spezielle neue Konsolen (z. B. Python mit Erweiterungen für die Programmiersprache C) öffnen können. Sie können aber auch die aktuelle Ausführung Ihrer Befehle anhalten *(Interrupt kernel),* Ihre Konsole neustarten *(Restart kernel),* alle Variablen aus Ihrer Programmierumgebung entfernen *(Remove all variables),* die aktuelle Konsole umbenennen *(Rename tab)* sowie Informationen zu Systemvariablen und Dateipfade betrachten, über die Spyder aktuell ausgeführt wird.

10.3.2.1 Benötigte und empfohlene Pakete für die Sentiment-Analyse

Um die Sentiment-Analyse durchzuführen, die Ergebnisse zu visualisieren und Gruppenvergleiche durchzuführen, benötigen wir die Pakete (auch *libraries* genannt) *vaderSentiment, pandas, re, os, numpy, scipy, matplotlib* und *seaborn.* Die Pakete *re* und *os* sind standardmäßig vorinstalliert. Sie können Pakete installieren, indem Sie in die Anaconda Console (Startmenü → Anaconda Prompt) gehen und `pip install [Paketname]` eingeben und Enter drücken. Um beispielsweise *pandas* zu installieren, müssen Sie `pip install pandas` eingeben und die Enter-Taste drücken. Doch wozu genau werden wir die einzelnen Pakete verwenden?

13 Beachten Sie dabei aber, dass Sie nur begrenzten RAM-Speicher zur Verfügung haben! Quantitative Textanalysen verbrauchen recht schnell einen Großteil Ihres Arbeitsspeichers. Um zu prüfen, wie viel Ram-Speicherplatz Sie zur Verfügung haben, drücken Sie in Windows „Strg + Alt + Entf“ und öffnen den Task-Manager. Hier sehen Sie unter Arbeitsspeicher, wie ausgelastet Ihr RAM-Speicher ist. Alternativ können Sie auch Task Manager in die Windows-Suche eingeben, um zum gleichen Fenster zu gelangen.

Das Paket *vaderSentiment* (Hutto und Gilbert 2014) verfügt erstens über die notwendigen Befehle für die Durchführung der Sentiment-Analyse. Es beinhaltet zweitens eine Liste mit Begriffen, die positiv oder negativ konnotiert sind. Drittens verfügt dieses Paket über ein Regelwerk, mit dessen Hilfe die Sentiment-Werte abhängig von Ihrer Einbettung im Satz modifiziert werden können.

Für die Arbeit mit den Daten selbst und deren Bereinigung verwenden wir die Pakete *os, pandas* und *re.* Das Paket *os* ist die Kurzform von *operating system* oder auf Deutsch Betriebssystem. In diesem Paket befinden sich Befehle, um Dateipfade zu erstellen, zu wechseln oder zu durchsuchen. Da Sie in Python mehr als einen Datensatz gleichzeitig geöffnet haben können, ist dieses Paket sehr hilfreich, um die Daten flexibel und automatisiert aufzulisten und anzusteuern. Das Paket *pandas* (McKinney 2011) benötigen wir, um Datensätze einzulesen (z. B. Excel-Dateien oder csv-Dateien), die Daten nach Bedarf zu verändern, miteinander zu kombinieren, Spalten und Zeilen umzubenennen oder in Python erzeugte Daten als strukturierten Datensatz zu speichern. Weiterhin werden wir *re* nutzen, um unsere Texte aufzubereiten. Die Kurzform *re* steht für *regular expression,* zu Deutsch regulärer Ausdruck, und bietet eine Reihe Befehle zum Finden, Ersetzen oder Auftrennen von Buchstabenfolgen. Wie Sie sehen werden, können mithilfe dieses Paketes Sätze aufgetrennt, Satzzeichen entfernt oder Wörter aus Analysen ausgeschlossen werden, die entweder wenig trennscharf sind (z. B. zu häufig oder zu selten in Texten vorkommen) oder keinen zusätzlichen Mehrwert für die Interpretation des Textes haben (beispielsweise Stoppwörter).

Für statistische Berechnungen werden wir die Pakete *numpy* (Oliphant 2006) und *scipy* (Virtanen et al. 2020) nutzen. *Numpy* ist ein Paket, das darauf ausgelegt ist, mathematische Berechnungen an Matrizen und Arrays möglichst schnell durchzuführen. Das Paket *scipy* beinhaltet Sammlungen von Funktionen, die ihrerseits Befehle zur Ausführung statistischer Berechnungen, *machine learning,* Bild- und Signalverarbeitung oder zur Verarbeitung von Geodaten beinhaltet.

Um die Ergebnisse zu visualisieren, werden wir *matplotlib* (Hunter 2007) und *seaborn* (Waskom 2021) verwenden. Während *matplotlib* Grafiken erstellt und Möglichkeiten bietet, diese interaktiv zu gestalten, ist das Paket *seaborn* auf die Erstellung statistischer Grafiken spezialisiert.

10.3.2.2 Importieren von Paketen und grundlegende Struktur von Befehlen und Funktionen

Wie bei RStudio müssen Sie in Spyder zunächst angeben, mit welchen Paketen und Funktionssammlungen Sie arbeiten möchten. Grundsätzlich können Sie ganze Pakete, Funktionssammlungen oder einzelne Befehle laden. Sie müssen die Pakete, Funktionssammlungen oder Befehle importieren, ehe Sie die von Ihnen gewünschten Berechnungen durchführen können. Wenn Sie ganze Pakete in Ihre

aktuelle Sitzung laden möchten, dann können Sie dies mittels `import [Paket]` machen. Beachten Sie aber, dass Sie anders als in RStudio das Paket adressieren müssen, in dem die Befehle gespeichert sind, und dann erst den Befehl eingeben können. Wenn Sie beispielsweise einen Mittelwert durch den im *numpy* befindlichen Befehl `mean()` berechnen möchten, dann müssten Sie `numpy.mean([Daten])` eingeben und `[Daten]` durch eine Variable in Ihrem Datensatz oder ein Objekt ersetzen, in dem Zahlen (Integers oder *floating point numbers*) gespeichert sind. Damit Sie nicht immer den ganzen Namen des Pakets angeben müssen, bietet Ihnen Python die Möglichkeit an, dem Paket in Ihrer Sitzung einen eigenen Namen zu geben. Wenn Sie *numpy* beispielsweise als np importieren wollen, dann modifizieren Sie einfach den Import-Befehl. Entsprechend lautet dieser dann `import numpy as np`. Sie können zuletzt innerhalb einer Zeile viele Pakete importieren. Hierzu können Sie die Zeile mit `import` beginnen und die zu ladenden Pakete mit Kommata abtrennen, wie Code 10.30 zeigt. Um den Befehl auszuführen, markieren Sie die Zeilen und drücken F9.

Code 10.30 Befehle zum Importieren von Paketen in Python

```
# ================================================================================
# Pakete einlesen
# ================================================================================
import os, pandas as pd, re, numpy as np, seaborn as sbs
```

Auf die gleiche Art können wir Funktionssammlungen importieren. Einige Pakete kommen mit so vielen Befehlen daher, dass sich die Entwickler entschlossen haben, die Befehle zu gruppieren und in eine Art Unterordner innerhalb des Paketes zu geben. Gleich drei unserer Pakete, die wir zur Vorbereitung und Durchführung der Sentiment-Analyse benötigen, weisen diese Struktur auf. Dabei handelt es sich um *matplotlib, scipy* und *vaderSentiment.* Die Befehle zur Erzeugung unserer Grafiken befinden sich in der Funktionssammlung pyplot innerhalb des *matplotlib*-Paketes. Um diesen zu laden, können Sie entsprechend `from matplotlib import pyplot` eingeben. Da wir allerdings hier mit Abkürzungen der Paketnamen arbeiten wollen, nennen wir es `plt` (Code 10.31).

Code 10.31 Befehl zum Importieren einer Funktionssammlung aus dem matplotlib-Paket

```
from matplotlib import pyplot as plt
```

Eine zweite Möglichkeit ist, dass Sie diese Funktionssammlung mittels `import` hineinladen können. Hierfür müssen Sie diese Sammlung mit einem Punkt von dem Paket abtrennen. Im Falle von `pyplot` würde die Struktur dieses Befehles `import matplotlib.pyplot as plt` lauten, sofern wir diese Funktionsansammlung als `plt` einladen möchten. Wir möchten nun aber darüber hinaus die `stats`-Funktionssammlung aus `scipy` als `st` in die aktuell geöffnete Sitzung laden (Code 10.32).

Code 10.32 Import der Statistik-Funktionssammlung aus dem Paket scipy

```
import scipy.stats as st
```

Daneben können Sie auch einzelne Befehle aus einem Paket in Ihre Sitzung hineinladen, wenn Sie nur einen oder wenige durch Kommata getrennte Befehle benötigen, die in einem Paket gespeichert sind. Hierfür beginnen Sie die Zeile mit `from`, gefolgt vom Namen des Paketes, ehe Sie `import` und den Befehlsnamen angeben. Die abstrakte Struktur ist `from [Paketname] import [Befehl]`. Wenn wir den Befehl zur Berechnung des Mittelwertes aus *numpy* importieren wollen, dann müssten wir entsprechend `from numpy import mean` eingeben. Sie können diese Eingabe auch mit dem `as`-Kommando kombinieren, falls Sie den Befehl nach dem Einlesen umbenennen wollen. So werden wir aus dem *vaderSentiment*-Paket nur den SentimentIntensityAnalyzer in unsere Sitzung laden. Dieser ist aber in einer Funktionssammlung enthalten, die wieder *vaderSentiment* heißt. Entsprechend kombinieren wir die beiden Möglichkeiten zum Laden von Funktionssammlungen, um an den Befehl zu gelangen. Wir beginnen mit `from`, steuern dann die Funktionssammlung in `vaderSentiment` an und laden den Befehl in unsere Sitzung mittels `import`, wie Code 10.33 zeigt.

Code 10.33 Import des SentimentIntensityAnalyzer-Befehls aus dem vaderSentiment-Paket

```
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
```

10.3.3 Ausführen von Befehlen und Überblick über verschiedene Datentypen

Um einen Befehl auszuführen, müssen Sie folgende drei Schritte beachten. Erstens müssen Sie das entsprechende Paket adressieren, in dem dieser Befehl gespeichert ist. Zweitens setzen Sie einen Punkt, der den Befehl abtrennt. Drittens schreiben

Sie den Namen des Befehls und setzen `()` dahinter. In diesen Klammern können die Daten stehen, mit deren Hilfe der Befehl beispielsweise Berechnungen durchführt. Hier können auch, getrennt durch Kommata, Optionen festgelegt werden, z. B. wie ein Befehl durchgeführt wird. Einige Befehle brauchen allerdings auch keine weitere Eingabe, da sie dazu gedacht sind, Funktionen aufzurufen oder Einstellungen zu setzen. In unserem Falle wäre dies der `set()`-Befehl des `seaborn`-Paketes. Dieser bewirkt, dass unsere Abbildungen, die wir erstellen werden, automatisch wie die Grafiken aussehen, die wir in RStudio mittels `ggplot2` erstellen können. Dies sieht dann wie folgt aus.

Code 10.34 Aufruf eines Befehls am Beispiel des set()-Befehls aus dem seaborn-Paket

```
# Einstellung, dass die Grafiken wie ggplot2-Grafiken aus R aussehen.
sbs.set()
```

Befehle benötigen normalerweise Eingaben, die in unterschiedlichen Datentypen erfolgen. Zudem können einzelne Befehle auch verschiedene Datentypen wieder ausgeben. Python kennt dabei Zeichenfolgen (Strings), ganze Zahlen (Integers), reelle Zahlen mit Nachkommastellen *(floating point numbers)* oder komplexe Zahlen *(complex numbers)* als einfache Datentypen – einfach deswegen, weil sie nicht weiter verschachtelt sind und keine Unterebenen mit eigenen Datentypen beinhalten. Zu Datentypen mit Unterebenen zählen Listen, Tupeln, Sets und Diktionäre.[14] Bei einer Liste handelt es sich um eine sortierbare Abfolge von Elementen. Diese Elemente können beispielsweise andere Listen, Zahlen oder Strings sein. Listen erlauben es, verschiedene Datentypen miteinander zu kombinieren. Listen weisen ferner die Möglichkeit auf, dass die einzelnen Inhalte veränderlich sind. Das bedeutet beispielsweise, dass Sie, wenn Sie die Zahl 42 an der ersten Stelle Ihrer Liste haben, durch einen Zugriff auf dieses erste Listenobjekt zur Zeichenfolge „zweiundvierzig" abändern können. Listen sind damit sehr flexible Datentypen, benötigen aber auch relativ viel Speicherplatz – was bei langen und großen Listen wie bei der Textanalyse zur baldigen Überlastung Ihres Arbeitsspeichers führen kann. Tupeln teilen alle Eigenschaften von Listen. Es ist allerdings nicht möglich, die Inhalte von Tupeln abzuändern.[15] Ein Diktionär ist wie ein Telefonbuch, bei dem bestimmten Namen *(keys)* Werte *(values)* zugeord-

14 Daneben gibt es auch in anderen Paketen wie *numpy* weitere Datenformate. Dazu zählen Arrays, Matrizen oder DataFrames, also die klassischen Datensätze, denen Sie in anderer Statistiksoftware häufiger begegnen werden.

15 Sollten Sie dennoch versuchen, ein Objekt innerhalb Ihrer Tupel zu verändern, dann erzeugen Sie den Fehler `TypeError: 'tuple' object does not support item assignment`.

net werden. Sets sind ungeordnete Mengen einzigartiger Objekte, d. h., wenn Sie ein Set aus allen Wörtern in Ihrem Datensatz bilden würden, dann würden nur die Wörter aufgeführt werden, die mindestens einmal vorkommen. Diktionäre sind Aneinanderreihungen verschiedener Objekte, denen eine bestimmte Stelle in Ihrem Arbeitsspeicher zugewiesen wird. Sie können nicht geordnet werden und verbrauchen relativ wenig Speicherplatz. Sie dienen als Grundlage, um DataFrames zu erstellen. Code 10.35 zeigt Ihnen, wie die einzelnen Datentypen in Ihrem Editor erstellt und mittels = an Variablen übergeben werden können. Vergessen Sie dabei nicht, die Codezeilen zu markieren und mittels F9 auszuführen.

Code 10.35 Übersicht über die in Python direkt integrierten Datentypen

```
liste = ["das", "ist", "eine", "liste",42,1.0, ["Liste", "in", "der", "Liste"]]
tupel = ("hallo", 123, 1.2, ("Tupel", "in", "der", "Tupel"))
diktionär = {"Telefonnummern": ["Vorwahl_Nummer","Vorwahl2_Nummer2"],
             "Farben": ("rot","gelb","grün"),
             "Zahlen": [1,2,3]}
```

Nach der Ausführung können Sie die ganzen Variablen aufrufen, indem sie `liste, tupel` oder `diktionär` in Ihrem Editor eintragen, markieren und F9 drücken. Alternativ geben Sie die Variablennamen in Ihre Konsole ein und drücken Enter und erhalten als Ausgabe alle Elemente, die Sie in der Liste, der Tupel oder dem Diktionär gespeichert haben, je nachdem, welchen Datentypus Sie angesteuert haben.

Um die einzelnen Elemente anzusteuern, können Sie bei einer Liste oder Tupel ein `[]` hinter Ihrem Objekt anfügen. Python vergibt dabei sogenannte Indexnummern, mit deren Hilfe Sie auf einzelne Objekte innerhalb Ihrer Variablen zurückgreifen können. Um die Gesamtzahl der Objekte in Ihrer Variablen zu ermitteln, geben Sie `len([VARIABLE])` ein, beispielsweise `len(liste)`, um die Anzahl der in der Liste gespeicherten Objekte zu ermitteln. Beachten Sie dabei, dass Sie, um bei unserer Liste die erste Eintragung aufzurufen, `liste[0]` aufrufen müssen, da Python immer bei 0 statt 1 mit der Durchnummerierung der Elemente beginnt. Wenn Sie umgekehrt das letzte Element Ihrer Liste aufrufen wollen, dann geben Sie `liste[-1]` an. Da wir hier eine Liste innerhalb der Liste gespeichert haben, können wir mit der gleichen Logik auf die einzelnen Elemente zurückgreifen. Um auf das erste Element innerhalb Ihrer Liste zurückzugreifen, geben Sie bitte `liste[-1][0]` ein. Das sagt Python, dass wir zunächst das letzte Objekt in dieser Liste und dann das erste Objekt ansteuern wollen, das sich eine Ebene tiefer befindet. Wenn Sie dies bei dem ersten Objekt, dem Wort „das“, machen würden, dann würden Sie mit `liste[0][0]` den ersten Buchstaben ansteuern. Wollen wir die ersten drei Elemente von liste aufrufen, dann geben

wir `liste[:3]` an. Das `:` zeigt Python an, dass eine aufeinanderfolgende Reihe von Elementen angesteuert werden soll. Wenn Sie einen Wert links von diesem Doppelpunkt angeben, dann handelt es sich um das Objekt mit dem niedrigsten Indexwert, das Sie ansteuern wollen. Wenn Sie diese Zahl weglassen, dann beginnt Python automatisch beim niedrigsten Wert Ihrer Liste etc. Rechts davon ist die Obergrenze, die nicht überschritten werden soll. Wir sagen also Python, dass wir alle Elemente unserer Liste bei der ersten Eintragung (Index = 0) beginnen und bei dem dritten Objekt mit Indexwert = 2 enden lassen. Bei einer Tupel ist es genauso, sodass wir hier nicht gesondert auf das Ansteuern von Elementen innerhalb der Tupel eingehen.

Bei einem Diktionär ist es etwas anders. Sie können alle Eintragungsnamen *(keys)* in unserem Falle mittels `diktionär.keys()`, alle Werte mittels `diktionär.values()` aufrufen. Die Kombination aus Eintragungsnamen (hier: Telefonnummern, Farben, Zahlen) und den dazugehörigen Werten können Sie mittels `diktionär.items()` aufrufen. Um einzelne Werte anzusteuern, die zu den Eintragungsnamen gehören, geben Sie den Eintragungsnamen in eckigen Klammern und in Anführungszeichen hinter dem Namen Ihres Diktionärs ein. Wenn wir auf die Farbenwerte zurückgreifen möchten, die im Diktionär gespeichert sind, geben Sie entsprechend `diktionär["Farben"]` an. Damit steuern Sie die Tupel an, in der die Farben gespeichert sind.

Wir sind so lange auf das Ansteuern von Daten eingegangen und lassen die Veränderung von Werten an geeigneter Stelle folgen, weil Sie ein Verständnis dafür benötigen, wie Sie mit den einzelnen Datentypen umgehen können, denen Sie bei Ihrer Sentiment-Analyse, aber auch bei anderen Verfahren in Python begegnen werden. Das ist umso wichtiger, weil bestimmte Ein- und Ausgaben in den Sentiment-Analysen oder in Topic Models als Tupeln, Diktionäre oder Listen ausgegeben werden und jeweils anders in Ihre Datensätze integriert werden können.

10.3.4 Einlesen von Daten in Spyder

Nun wollen wir die Daten in die Python-Sitzung einlesen, die wir für die Durchführung der Sentiment-Analyse benötigen. Hierfür gibt es zwei Möglichkeiten. Sie können erstens die Daten mittels des Importieren-Buttons im *Variable explorer* importieren. Sie können aber auch die Daten zweitens direkt über den entsprechenden Befehl laden, der im *pandas*-Paket enthalten ist. Hierfür müssen Sie zuerst den Dateipfad festlegen, in dem Sie Ihren vorbereiteten Datensatz gespeichert haben. Dies können Sie, indem Sie den Dateipfad einer Variablen (z. B. *path*) übergeben und dann den Ordner als Arbeitspfad mittels des `os.chdir(path)`-Befehls festlegen. Dabei ist `path` der zuvor gespeicherte Dateipfad. Der Befehl ist wie in Code 10.36 folgt aufgebaut.

Code 10.36 Verändern Ihres Dateipfades in Python

```
path = "C:\\[IHR DATEIPFAD]\\"
os.chdir(path)
```

Sie sehen, dass Ihr Dateipfad statt mit einem `\` mit `\\` abgetrennt ist. Sie werden sich wohl schon gefragt haben, warum das so ist. In Python ist `\` als ein Zeichen definiert, das andere Operationen einführt. Zum Beispiel verwendet Python `\n` als Zeichenkette, die eine neue Zeile (z. B. bei der Ausgabe) definiert. Um die Datei zu laden, in der die Amazon-Rezensionen von Filmen und Serien gespeichert sind, nutzen wir den `read_csv()`-Befehl. Wären die Textdaten in einer Excel-Datei gespeichert, dann könnten Sie den `read_excel()`-Befehl verwenden. Sie weisen die Datei einer Variablen zu, indem Sie `=` verwenden und links von diesem Zeichen die Variable, rechts davon den Befehl wie in Code 10.37 schreiben.

Code 10.37 Befehl zum Einlesen einer Tabellendatei als pandas-DataFrame

```
df = pd.read_csv("Amazon_Movies_and_TV_5_Sample.csv")
#%%
```

Zuletzt sehen Sie die Zeichenfolge **`#%%`**. Damit deuten wir Python an, dass wir eine Codezelle definieren, die von der ersten Zeile bis zu dieser Zeichenfolge reicht. Eine Codezelle ist ein abgegrenztes, sich über mehrere Zeilen erstreckendes Codestück, in dem eine Abfolge von Befehlen aufgeführt ist. Codezeilen dienen dazu, Ihren Code in Sinnabschnitte einzuteilen und diese Einheiten separat auszuführen. Das hat den Vorteil, dass Sie bestimmte Berechnungen (z. B. die eigentliche Durchführung Ihrer Sentiment-Analyse) getrennt voneinander und je nach Bedarf durchführen können. Weiterhin helfen solche Abtrennungen, eventuelle Fehler im Code leichter zu entdecken und zu beheben. Um eine Codezelle auszuführen, können Sie in eine beliebige Zeile klicken, die durch diese Zelle innerhalb der Codezelle abgedeckt wird, und mittels „Strg + Enter“ die ganze Zelle ausführen oder durch „Strg + Shift + Enter“ die Zelle ausführen und zur nächsten Zelle springen. Der gesamte Code, mit dessen Hilfe Sie die Pakete importieren und Daten einlesen können, lautet wie in Code 10.38 demonstriert.

Code 10.38 Voller Code zum Einlesen unserer Programmpakete und Daten in Python

```
# ============================================================================
# Pakete einlesen
# ============================================================================
import os, pandas as pd, re, numpy as np, seaborn as sbs
from matplotlib import pyplot as plt
import scipy.stats as st
from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

# Einstellung, dass die Grafiken wie ggplot2-Grafiken aus R aussehen.
sbs.set()

# ============================================================================
# Daten einlesen
# ============================================================================
path = "C:\\[IHR DATEIPFAD]\\"
os.chdir(path)
df = pd.read_csv("Amazon_Movies_and_TV_5_Sample.csv")
#%%
```

10.3.5 Daten aufbereiten

Die nächste Hürde, die wir nehmen müssen, um eine Sentiment-Analyse durchzuführen, besteht darin, die Textdaten aufzubereiten. Hierfür müssen wir folgende Schritte durchführen.

1. Wir trennen den Text in einzelne Sätze auf.
2. Wir transformieren alle Wörter in Kleinschreibung *(lowercase)* und entfernen alle Einzelbuchstaben, die in den Daten vorhanden sind.
3. Wir entfernen alle Sonderzeichen und überflüssigen Leerzeichen (z. B. am Anfang einer Zeichenfolge oder doppelte Leerzeichen).
4. Wir erstellen eine neue Variable im Datensatz, in der die aufbereiteten Einzelsätze als Liste gespeichert sind, und entfernen überflüssige Variablen aus dem RAM-Speicher unseres PCs.

Um diese vier Schritte erfolgreich durchzuführen, benötigen wir sowohl die Befehle `re.split()` und `re.sub()`, die Funktionen `.lower()` und `.append()`, Listenabgleiche, eine `for`-Schleife und logische Abfragen (`if-else`). Diese wer-

den in der Folge an geeigneter Stelle bei den einzelnen Schritten eingeführt und detailliert besprochen.

10.3.5.1 Auftrennen von Sätzen mittels re.split()

In unserer Datei sind die Reviews jeweils als Zeichenkette, Strings genannt, in der Variablen `reviewText` gespeichert. Wir können den Inhalt dieser Variable grundlegend ansteuern, indem wir diese entweder mit einem Punkt von dem Objekt trennen, in dem unser Datensatz gespeichert ist, oder das Objekt mit eckigen Klammern und Anführungszeichen ansteuern. Wenn wir unseren Datensatz als `df` in unsere Python-Sitzung geladen haben, dann können wir die Bewertungstexte mit `df.reviewText` oder `df["reviewText"]` aufrufen. Wenn Sie diesen Code in eine Befehlszeile schreiben, markieren und mit F9 ausführen, dann werden Ihnen in der Ipython Console standardmäßig die ersten und letzten fünf Datenzeilen angezeigt.

Der Befehl aus dem `vaderSentiment`-Paket, mit dessen Hilfe die Sentiments errechnet werden sollen, versucht stets auf Basis des gesamten Strings einen Stimmungswert zu ermitteln, der positive oder negative Stimmung bzw. einen zusammengesetzten Wert aus beiden Stimmungslagen beinhaltet. Daher müssen wir diese Zeichenketten in einzelne Sätze auftrennen, um Verzerrungen bei der Ermittlung der Sentiments zu vermeiden. Doch wie können wir das in Python erreichen?

Wir wissen, dass Sätze in einer natürlichen Sprache normalerweise durch Satzzeichen und Zeilenumbrüche voneinander getrennt werden. Das heißt, dass wir in einer Zeichenkette nach einem Punkt, einem Ausrufezeichen, einem Fragezeichen oder einem Zeilenumbruch suchen und die Zeichenfolge an diesen Stellen auftrennen müssen. Hierfür müssen wir ein Muster definieren, das die Satzzeichen beinhaltet, die als Trennzeichen für unseren String fungieren. Dieses Muster leitet dann unsere Suche in Python an und hilft unserem Programm, die einzelnen Sätze aufzutrennen. Wir wollen Python beibringen, dass es einen String, d. h. eine Zeichenfolge, aufteilen soll, wenn Satzabschlusszeichen im Textstring gefunden werden.

Hierfür definieren wir eine Variable namens `pattern`, in der wir dieses Muster abspeichern (Code 10.39). Sie können diese Variable auch anders nennen. Wichtig ist nur, dass Sie das Muster, das Sie hier definiert haben, in der Folge auch abrufen. Um ein Muster zu definieren, das nach Satzzeichen und Zeilenumbrüchen sucht, können Sie die folgende Datenzeile ausführen.

Code 10.39 Definition eines regulären Ausdrucks zum Auffinden von Satzabschlusszeichen

```
# ============================================================================
# Daten Bearbeiten
# ============================================================================

pattern = r"(\.|!|\?|\n)"
```

Wie Sie bestimmt schon gesehen haben, besteht die Zuweisung aus drei Bestandteilen. Zuerst sehen wir ganz links die Variable, der wir unsere Zeichenfolge zuweisen wollen. Danach erfolgt die Zuweisung durch `=` , während sich auf der rechten Seite ein Objekt befindet, das zugewiesen werden soll. Dieses Objekt sieht auf den ersten Blick eher kryptisch aus. Sehen wir es uns daher genauer an.

Das `r` sagt aus, dass ein regulärer Ausdruck, eine `regular expression`, folgt (siehe Box 10.1 für die Definition). Dieser Ausdruck steht in Anführungszeichen. Wenn Sie also in einem Text nach dem Ausdruck „quantitative Textanalyse“ suchen wollen, dann könnte hier also `r"quantitative Textanalyse"` stehen. Danach sehen Sie Zeichen, die durch runde Klammern eingeschlossen werden. Diese runden Klammern sagen uns, dass wir nach den Zeichen suchen, die in dieser Klammer stehen. Die senkrechten Striche in der Klammer `|` zeigen ein logisches *oder* an. Wenn Sie zum Beispiel nach den Zeichenketten „quantitative Textanalyse“ oder „qualitative Textanalyse“ in einem String suchen möchten, dann würde Ihr Muster entsprechend `r"(quantitative Textanalyse|qualitative Textanalyse)"` lauten. Sie können dieses Symbol durch die Tastenkombination Alt Gr + < (die Taste links vom y auf Ihrer Tastatur) erzeugen. In den Klammern finden Sie einen Punkt, ein Ausrufezeichen, ein Fragezeichen und das Symbol für einen Zeilenumbruch. Wir müssen `\.` schreiben, da der Punkt bei den regulären Ausdrücken für „beliebiges Zeichen“ steht. Ein Punkt allein würde alle Buchstaben (klein- und großgeschrieben), Zahlen, Sonderzeichen, aber auch Leerzeichen adressieren! Würden Sie später nur nach einem Punkt suchen, dann würden Sie ausnahmslos jedes Zeichen voneinander abtrennen und einzelne Buchstaben herausbekommen. Das `?` muss ebenfalls mit `\` eingeleitet werden, da es ansonsten Python sagen würde, dass es nach einer zuvor festgelegten, sich potenziell direkt hintereinander wiederholenden Zeichenkette Ausschau halten soll. Zuletzt signalisiert das `\n` einen Zeilenumbruch. Eine Übersicht über die einzelnen speziellen Zeichen bietet Ih-

Box 10.1: Regular Expression
Unter einer Regular Expression wird eine Abstraktion von einem oder mehreren Suchtermini verstanden, mit dessen Hilfe die Suche (und das Ersetzen) von Ausdrücken innerhalb von Zeichenketten *(strings)* oder Texten ermöglicht wird.

nen die offizielle Anleitung zum `re`-Paket, die Sie unter *https://docs.python.org/3/library/re.html* aufrufen können.

Um nun diese Trennung vorzunehmen, verwenden wir den Befehl `re.split()`. In diesem Befehl geben wir zunächst das Muster an, das wir nutzen wollen, um eine Zeichenfolge zu trennen. Dann folgt die Zeichenfolge nach einem Komma. Hier empfiehlt es sich, nicht die ganze Zeichenfolge anzugeben, sondern Sie zuvor einer Variablen separat zu übergeben. Grundsätzlich hat der Befehl daher den Aufbau `re.split([Muster],[Zeichenfolge])`. Testen wir diesen Befehl an einer Zeichenfolge, die wir in das Objekt `saetze`[16] speichern. Um das Ergebnis auszugeben, können wir unseren `re.split()`-Befehl in einen `print()`-Befehl schreiben oder das Ergebnis unseres Trennvorgangs in eine eigene Variable übergeben und dann die Variable mit `print()` in der Ipython Console wie in Code 10.40 ausgeben lassen.

Code 10.40 Auftrennen von Sätzen nach einem zuvor definierten regulären Ausdruck

```
saetze = """Dieses Buch ist dazu da, um Verfahren der quantitativen Textanalyse
zu lernen. Hier steht also ganz viel drin! Wie viel?


Das weiß ich nicht, das bleibt Ihnen überlassen.
"""
print(re.split(pattern, saetze))
```

Wir erhalten, wenn wir nun diesen Code markieren und mit F9 ausführen, folgenden Output in unserer Ipython Console.

Output 10.8 Aufgetrennter Textstring nach Anwendung des split()-Befehls und Anwendung eines Patterns mit runden Klammern

```
Out[0]:
['', '\n', 'Dieses Buch ist dazu da, um Verfahren der quantitativen Textanalyse
zu lernen',
 '.', ' Hier steht also ganz viel drin', '!', ' Wie viel', '?', '', '\n', '',
'\n', 'Das weiß ich nicht, das bleibt Ihnen überlassen', '.', '', '\n', '']
```

16 Die drei Anführungszeichen am Anfang und am Ende dieser Zeichenfolge sagen Python, dass es sich um eine String-Variable handelt, die sich über mehrere Zeilen im Editor erstreckt.

10.3.5.2 Listenobjekte und Listenabgleiche

Der Output sagt uns, dass Sie ein Listenobjekt erzeugt haben, in dem Buchstabenketten vorliegen und durch Kommata getrennt werden. Nochmals zur Erinnerung: Listenobjekte werden durch eckige Klammern [ ] begrenzt und können verschiedene Objekttypen, etwa Zeichenfolgen, Zahlen, andere Listen, Tupeln, Diktionäre, Matrizen oder DataFrames, beinhalten. Wie Sie sehen, erzeugen Sie einzelne Objekte, die zum Teil leer sind, zum Teil Sätze enthalten und zum Teil die Satzzeichen. Das liegt daran, dass die runden Klammern Python mitteilen, dass alle aufgetrennten Teile des Strings, inklusive der Zeichen, behalten werden sollen. Wir wollen aber diese Zeichen nicht behalten, weil Sie für die Sentiment-Analyse nicht von Bedeutung sind. Hierfür können wir die runden Klammern im Muster durch eckige ersetzen. Wenn wir dies machen, dann erhalten wir, wenn wir das resultierende Muster an den `re.split()`-Befehl übergeben, alles markieren und dann mit F9 ausführen, das in Code 10.41 und Output 10.9 dargestellte Ergebnis.

Code 10.41 Definition eines Patterns mit eckigen Klammern, das gefundene Trennzeichen aus dem Text löscht

```
pattern2 = r"[\.|!|\?|\n]"
print(re.split(pattern2, saetze))
```

Output 10.9 Aufgetrennter Textstring nach Anwendung des split()-Befehls und Anwendung eines Patterns mit eckigen Klammern

```
Out[1]:
['', 'Dieses Buch ist dazu da, um Verfahren der quantitativen Textanalyse zu
lernen',
' Hier steht also ganz viel drin', ' Wie viel', '', '', 'Das weiß ich nicht,
das bleibt Ihnen überlassen', '', '']
```

Am Output erkennen Sie, dass Sie nun eine Liste erhalten, in der sowohl die Sätze als auch leere Elemente (mit Länge = 0) erhalten sind. Letztere wollen wir später nicht analysieren – sie würden ein neutrales Sentiment ergeben und damit den gesamten Wert (positive/negative Stimmungslage) der Aussage verzerren. Entsprechend müssen wir diese Listenelemente aus der Liste entfernen. Das erreichen wir über einen Listenabgleich (Englisch: *list comprehension*). Listenabgleiche sind ein mächtiges Instrument, mit dessen Hilfe Sie in Python sehr schnell sehr viele Daten nach einer von Ihnen vorgegebenen Regel bearbeiten können. Um einen Listenvergleich zu machen, werden wir zunächst das Ergebnis

des `re.split()`-Befehls in eine eigene Variable übergeben. Die folgenden fünf Zeilen (Code 10.42) demonstrieren dabei, wie 1) zunächst die Übergabe der erzeugten Liste in eine Variable erfolgt, dann 2) jedes Element der Liste abgerufen wird, dann 3) die Anzahl der Zeichen jedes Elementes angezeigt werden, 4) nur die Elemente der Liste behalten werden, die nicht leer sind, d. h. deren Zeichenzahl größer als 0 ist und 5) die Elemente behalten und in Kleinschreibung transferiert werden, die nicht leer sind.

Code 10.42 Beispiele für Befehle und Veränderungen von Daten, die durch einen Listenabgleich in Python erfolgen können

```
satzliste = re.split(pattern, saetze)
[wort for wort in satzlist]
[len(wort) for wort in satzliste]
[wort for wort in satzliste if len(wort) > 0]
[wort.lower() for wort in satzliste if len(wort) > 0]
```

Führen wir nun Zeile für Zeile durch Markieren und anschließendem Drücken von F9 aus, so erhalten wir.

Output 10.10 Ausgabe der verschiedenen Beispiele für den Listenabgleich (Fortsetzung nächste Seite)

```
In [2]: [wort for wort in satzliste]
Out[2]: ['Dieses Buch ist dazu da, um Verfahren der quantitativen Textanalyse
zu lernen', ' Hier steht also ganz viel drin', ' Wie viel', '', '', 'Das weiß
ich nicht, das bleibt Ihnen überlassen', '', '']

In [3]: [len(wort) for wort in satzliste]
Out[3]: [77, 31, 9, 0, 0, 47, 0, 0]

In [4]: [wort for wort in satzliste if len(wort) > 0]
Out[4]: ['Dieses Buch ist dazu da, um Verfahren der quantitativen Textanalyse
zu lernen',
 ' Hier steht also ganz viel drin',
 ' Wie viel',
 'Das weiß ich nicht, das bleibt Ihnen überlassen']

In [5]: [wort.lower() for wort in satzliste if len(wort) > 0]
Out[5]: ['dieses buch ist dazu da, um verfahren der quantitativen textanalyse
zu lernen',
```

```
' hier steht also ganz viel drin',
' wie viel',
'das weiß ich nicht, das bleibt ihnen überlassen']
```

Sie erkennen in den vier Listenabgleichen sicherlich Gemeinsamkeiten. In allen eckigen Klammern stehen „`wort, for, in` und Ihre Variable“, in der Sie die Sätze gespeichert haben. Dabei handelt es sich bei `wort` um einen Iterator. Ein Iterator ist eine Art Zeiger, der nacheinander auf die einzelnen Bestandteile oder auch Elemente Ihrer Liste zeigt. Sie können sich einen Iterator wie einen Uhrzeiger vorstellen, der erst auf 1:00 Uhr, dann auf 2:00 Uhr, zuletzt auf 12:00 Uhr zeigt. Sie können `wort` durch ein `x` oder eine andere beliebige Zeichenfolge ersetzen. Das `for` zeigt an, dass Element für Element der Liste angesteuert wird und mit diesem etwas gemacht werden soll, es z. B. in Kleinbuchstaben transformiert wird. Im ersten Fall bewirkt dies nur, dass Element für Element der Liste übernommen wird. Im zweiten Fall soll Ihnen die Anzahl der verwendeten Zeichen angezeigt werden, die durch den `len()`-Befehl abgerufen wird.[17]

Im dritten Listenabgleich haben wir mit `if` eine logische Abfrage eingeführt. Genau genommen sagen wir Python hier, dass nur die Elemente unserer Liste beibehalten werden sollen, deren Zeichenzahl größer als null sind. Zuletzt sagen wir mit der durch einen Punkt getrennten Funktion `text.lower()`, dass wir die einzelnen Texte in der Liste in Kleinschreibung umsetzen möchten. In Gänze bedeutet der vierte Listenabgleich, dass wir a) nur die Sätze beibehalten wollen, die mindestens über ein Zeichen verfügen und b) diese Sätze in Kleinschreibung umgesetzt werden.

10.3.5.3 for-Schleifen und if-else-Abfragen

Wenden wir uns nun mit dem Wissen, das wir erworben haben, den Sätzen zu, die im Datensatz in der Variable `reviewText` gespeichert sind. Sie können die einzelnen Texte mit einem Listenabgleich (z. B. `[text for text in df.reviewText]`) aufrufen. Damit können wir zwar auf den einzelnen Text, nicht aber auf die einzelnen Worte oder Textabschnitte zurückgreifen. Das gilt auch für die Bereinigung von Textstrings. Da unser Ziel aber ist, die oben an einem zum Test verfassten String durchgeführten Bereinigungsschritte anzuwenden und wir dies bereits durch einen Listenabgleich realisiert haben, schlagen wir einen anderen

17 Sie können diesen Befehl auch außerhalb des Listenabgleiches verwenden. Wenn Sie zum Beispiel `len(saetze)` eingeben, dann erhalten Sie 171 zurück, weil die ursprüngliche Zeichenfolge aus insgesamt 171 Zeichen besteht. Daran erkennen Sie bereits, dass es auch in Python möglich ist, Befehle und Anfragen miteinander zu kombinieren.

Weg ein, um alle Bewertungen zu adressieren, die in der Variablen `reviewText` gespeichert sind. Wir können das mit einer `for`-Schleife machen. Diese ruft nacheinander alle in der Variablen befindlichen Objekte auf und führt dann den Befehl aus, der in der Schleife steht. Wenn sie beispielsweise die ersten hundert Zeichen jeder Bewertung aufrufen wollen, die in unserem Datensatz gespeichert ist, dann schreiben Sie folgenden Code.

Code 10.43 Beispiel einer for-Schleife in Python

```
for text in df.reviewText:
    print(text[:100])
```

Wie beim Listenabgleich sehen Sie auch hier wieder einen Iterator, den wir `text` genannt haben. Er ist nur ein Platzhalter für jeden einzelnen Text, den Sie nacheinander in der Variablen ansteuern möchten. Das `in` sagt Python, dass nun angegeben wird, in welcher Variablen es genau nachzusehen hat. Danach folgt die Variable, durch die der Iterator durchgehen soll. Der Doppelpunkt sagt aus, dass in der folgenden Zeile die Befehle folgen, die in der Schleife enthalten sind. Jeder Befehl, den die Schleife ausführen soll, ist mit einem Tabulator-Abstand eingerückt. Dieser Abstand signalisiert, dass es sich um Befehle handelt, die ausschließlich in dieser Schleife ausgeführt werden sollen. Das `[:100]` sagt, dass der Text vom ersten bis zum hundertsten Zeichen des text-Objektes ausgegeben werden soll.[18] Wenn Sie die Texte vom hundertsten bis zweihundertsten Zeichen angezeigt haben möchten, dann geben Sie entsprechend `[100:201]` an. Wir geben 201 an, weil Python immer bis einen Wert unter der angegebenen Obergrenze zählt und die Obergrenze ausschließt. Zugleich beginnt die Zählung immer bei Wert 0 und nicht bei 1 (wie wir weiter oben im Hinblick auf die Indexwerte der Objekte in den Listen ausgeführt haben).

Falls Sie nur lange Texte mit mindestens 300 Zeichen angezeigt bekommen möchten, dann können Sie noch eine logische Abfrage in die Schleife einführen. In unserem Falle wäre dies eine `if-else`[19]-Abfrage. Diese teilt Python mit, dass es unterschiedliche Dinge tun soll, falls diese Bedingung zutrifft *(if)* und für die Fälle, in denen diese Bedingung nicht zutrifft *(else)*. Diese werden ihrerseits mit

18 Beachten Sie dabei, dass Python einer anderen Zählweise als R folgt. In Python beginnt man beim nullten Element, in R beim ersten. Das heißt, wenn Sie das erste Element einer Liste in Python anwählen wollen, dann müssten Sie `Listenobjekt[0]` angeben, in R `Listenobjekt[1]`, sofern Sie Ihre Liste Listenobjekt genannt haben.

19 Es gibt noch die Möglichkeit, weitere Bedingungen zu setzen. Hierfür nutzen sie dann `elif`, also eine kurze Form von else if. Diese Bedingungen werden zwischen die erste if-Abfrage und die abschließende else-Abfrage aufgeführt.

Punkten am Ende der Zeile abgegrenzt und führen dazu, dass eine weitere, durch einen Tabulatorabstand generierte Zeile wie in Code 10.44 entsteht.

Code 10.44 Beispiel einer logischen if-else-Abfrage innerhalb einer for-Schleife

```
for text in df.reviewText:
    if len(text) > 300:
        print(text[:100])
    else:
        continue
```

Wie Sie sehen, können wir Schleifen und logische Abfragen miteinander verknüpfen. Damit Sie den Überblick über die Ebenen und damit die Verschachtelungen behalten, auf denen diese Abfragen und Schleifen durchgeführt werden, werden immer neue Tabulatorabstände erzeugt. Das Wort `continue` zeigt bei der Zeile nach `else` an, dass der Iterator einfach mit der nächsten Objektzeile weitermacht. Wir können aber auch Listenabgleiche in Schleifen schreiben und unsere Texte auf diese Weise auftrennen und bereinigen (Code 10.45). Darüber hinaus können wir auch das Ergebnis dieser Listenabgleiche an eine Liste anhängen und diese in eine neue Variable speichern.

Code 10.45 Bereinigung Ihrer Texte durch Listenabgleiche in einer for-Schleife

```
# ============================================================================
# Daten Bearbeiten
# ============================================================================
pattern = r"[\.|!|\?|\n]"

cleaned_texts = []
for text in df.reviewText:
    review_text = re.split(pattern,text)
    review_text = [x.lower() for x in review_text]
    review_text = [x for x in review_text if len(x) > 0]
    review_text = [re.sub("(^ +|  +|[^a-zA-Z1-9 ])","",x) for x in \
                   review_text] # \ bedeutet: Zeilenumbruch eines Befehls
    cleaned_texts.append(review_text)

df["bereinigte_texte"] = cleaned_texts
del(cleaned_texts, review_text)
#%%
```

Auch hier definieren wir als erstes die Variable `pattern`, in der wir angeben, welches Muster wir zum Trennen unserer Texte heranziehen wollen. Danach definieren wir ein Listenobjekt namens `cleaned_texts` und übergeben diesem eine leere Liste. Hiernach rufen wir eine for-Schleife aus, die alle Bereinigungsschritte nacheinander durchführt. Damit dies möglich wird, definieren wir eine neue Variable namens `review_text`, die nur in der Schleife verwendet wird und bei jedem Ansteuern eines neuen Textes in `df.reviewText` (reviewText ist der Variablenname) neu erstellt wird. Dieser Variable übergeben wir zunächst die Liste, in der wir das jeweilige Review Satz für Satz auftrennen. Danach steuern wir diese neue Liste an und überschreiben sie mit den durch `x.lower()` klein geschriebenen Sätzen. Danach überscheiben wir unsere Variable abermals und behalten nur diejenigen Elemente der Liste, die nicht leer sind, also eine Zeichenzahl größer null haben. Zuletzt ersetzen wir die durch die Trennung entstandenen Leerzeichen am Anfang der Zeichenketten, angedeutet durch `^ +`, doppelte Leerzeichen, und geben durch die Kombination aus `[^a-zA-Z1-9 ]` an, dass alle normalen Buchstaben (Groß- und Kleinschreibung) sowie einfache Leerzeichen in unseren Strings verbleiben sollen. Durch diese Zeichenfolge können wir alle Sonderzeichen aus unseren Texten entfernen. An der letzten Stelle der Schleife bewirkt die `append()`-Funktion, dass wir unserem Listenobjekt `cleaned_texts`, den in der Schleife bereinigten Text, hier als `review_text` vorliegend, anhängen.

Danach definieren wir eine neue Variable für unseren Datensatz und nennen diese `bereinigte_texte`. Diese Variable stellt eine neue Spalte im Datensatz bereit.[20] Dieser übergeben wir nun die Liste. Wenn Sie in den Variable Explorer (rechts oben) schauen, dann bemerken Sie sicherlich, dass sich beim Datensatz unter Size nun statt `(10000,9)` `(10000,10)` in der Zeile befindet, in der die Informationen unseres Datensatzes angezeigt werden. Das bedeutet, dass nun 10 000 Zeilen und 10 Spalten in unserem Datensatz vorliegen. Sie können kontrollieren, ob unsere Befehle erfolgreich ausgeführt wurden, indem Sie die ersten fünf Texte mit `df.bereinigte_texte.head()`, die letzten mittels der `tail()`-Funktion aufrufen.[21] Das erspart Ihnen Zeit, weil Sie nur einen ganz kleinen Teil Ihres Datensatzes aufrufen – und ein ganzer Datensatz mit sehr viel Text sehr groß ist und sehr lange Zeit zum Aufrufen benötigt! Zuletzt löschen wir die neuen, nun überflüssig gewordenen Variablen mittels des `del()`-Befehls aus dem

20 Bedenken Sie, dass Sie, wenn Sie Spalten hinzufügen möchten, stets Listen hinzufügen. Im Falle von Zeilen müssten Sie Tupeln übergeben, die in Python durch () eingeklammert sind.

21 Wenn Sie mehr als fünf bereinigte Texte anzeigen lassen möchten, die am Beginn bzw. Ende ihres Datensatzes vorhanden sind, dann geben Sie an dieser Stelle `df.bereinigte_texte.head(n=10)` bzw. `df.bereinigte_texte.tail(n=25)` ein. Das n ist eine Option der jeweiligen Funktion und sagt Python, wie viele Texte Sie auswählen möchten. Im ersten Falle wählen Sie die ersten zehn Texte an, in letzterem 25. Dieses Vorgehen kann nützlich sein, wenn Sie einen von Ihnen geschriebenen Code zeitsparend an wenigen Fällen ausprobieren wollen.

Speicher. Wenn Sie mit wenigen Daten arbeiten und ansonsten wenige Variablen definieren, dann müssen Sie diesen Schritt nicht machen. Im Falle von großen Datensätzen, die schnell Ihren Rechner überlasten können, empfiehlt es sich aber, die Daten aus dem Arbeitsspeicher zu entfernen, die Sie nicht mehr benötigen.

10.3.6 Ausführung der Sentiment-Analyse

Kommen wir nun zur eigentlichen Sentiment-Analyse. Sie werden sehen, dass wir für die Durchführung der Sentiment-Analyse auf die gleiche Logik zurückgreifen werden, die bereits für die Vorbereitung und Bereinigung des Textes angewandt wurde. Auch hier werden wir eine for-Schleife, einen Listenabgleich und die `append()`-Funktion verwenden. Zu diesen kommt der Abruf eines Sentiment-Lexikons mitsamt der Zuweisung eines Sentiment-Scores zu den jeweiligen Rezensionen. Wie Code 10.46 zeigt, benötigen nur weitere sieben Zeilen Code, um diese Analyse durchzuführen. Dabei folgt der Code sieben Schritten.

1. Wir laden die Funktion `SentimentIntensityAnalyzer()` aus dem *vaderSentiment*-Paket und übergeben Sie einem Objekt, das wir auf unsere bereinigten Texte anwenden.
2. Wir generieren eine leere Liste, in die wir die Sentiments übergeben möchten.
3. Wir schreiben eine for-Schleife, die die bereinigten Texte nacheinander ansteuert, die im Datensatz gespeichert sind.
4. Wir ermitteln die Sentiment-Scores der Reviews Satz für Satz und errechnen den durchschnittlichen Wert.
5. Wir fügen den durchschnittlichen Wert der Liste hinzu.
6. Wir übergeben diese Liste dem Datensatz als neue Variable.
7. Wir bereinigen den Datenspeicher.

Code 10.46 Durchführung der Sentiment-Analyse in Python (Fortsetzung nächste Seite)

```
# ============================================================================
# Durchführung der Sentiment-Analyse
# ============================================================================

analyzer = SentimentIntensityAnalyzer()

sentiments = []
for text in df.cleaned_texts:
    sentiment = [analyzer.polarity_scores(sentence)["compound"] for sentence
in text]
```

```
    sentiments.append(np.mean(sentiment))

df["sentiments"] = sentiments
del(sentiments,sentiment)

df.to_csv("Sentiment_Analyse_Amazon_Movies_and_TV_5_Sample.csv")
#%%
```

Beachten Sie dabei, dass wir den Text innerhalb des Listenabgleichs Satz für Satz durchgehen. Die Funktion `analyzer.polarity_scores()` berechnet Ihnen den Sentiment-Score. Dabei wird ein Diktionär erstellt, der einen negativen Score, einen neutralen Score, einen positiven Score sowie einen aus den drei Scores zusammengesetzten Score enthält, der von –1 bis +1 reicht. Wenn wir zum Beispiel den Satz `'but there was nothing funny or original here'` in die Klammern des `analyzer.polarity_scores()` einsetzen, dann erhalten wir die folgende Ausgabe.

Output 10.11 Beispielhafte Ausgabe der Sentiments nach der Analyse eines Satzes

```
{'neg': 0.481, 'neu': 0.519, 'pos': 0.0, 'compound': -0.6759}
```

An den geschwungenen Klammern sehen Sie, dass es sich um ein Diktionär handelt. Die Texte in Anführungszeichen sind die `keys`, die Werte selbst die `values` eines Diktionärs. Wenn wir nun auf die negativen Werte zurückgreifen möchten, die als `'neg'` adressiert werden können, dann können wir entweder das ausgegebene Diktionär in eine neue Variable übergeben, z. B. `negative_sentiment`, und den Wert durch das Hinzufügen von `['neg']` aufrufen. Alternativ können Sie `['neg']` auch direkt hinter den Befehl schreiben. Letztere Variante haben wir im Listenvergleich angewandt. Hier teilen wir Python mit, dass wir den zusammengesetzten compound-Wert unserer Sentiment-Analyse Satz für Satz berechnen möchten. Dabei generieren wir eine Liste, die wir der Variable `sentiment` übergeben. In der nächsten Zeile berechnen wir den Mittelwert aus allen zusammengesetzten Sentiment-Werten, in dem wir den `mean()`-Befehl aus dem `numpy`-Paket verwenden und auf die Liste beziehen, in der die Sentiment-Werte gespeichert sind. Diese fügen wir der eingangs definierten Liste `sentiments` mittels der `append()`-Funktion hinzu. Wir übergeben diese Liste dem Datensatz als Variable mit gleichem Namen, ehe wir sowohl die Liste aller durchschnittlichen Sentiments und die Liste löschen, in der wir die Sentiment-Werte pro Text bestimmt haben.

Als zusätzlichen Schritt geben wir an, dass wir die Bewertungen speichern möchten. Hierfür bietet das Paket *pandas* eine Reihe von Funktionen an, die Sie – wiederum durch einen Punkt getrennt – im Anschluss an Ihren Datensatz aufrufen können. Diese Funktionen werden mit einem `to_` eingeleitet, wonach Sie die Dateiform angeben können. In unserem Falle wollen wir die Ergebnisse unserer Sentiment-Analyse als csv-Datei speichern. Hierfür geben wir `df.to_csv("[NAME DER CSV-DATEI].csv")` ein. Es besteht aber auch die Möglichkeit, diese Datei als Excel-Datei, als Json-Datei, als SQL-Datenbank, als Stata-Datei, als SAS oder SPSS-Datei zu exportieren. Auf diese Weise könnten Sie dann zusätzliche Berechnungen in anderen, Ihnen bereits bekannten statistischen Paketen durchführen.

10.3.6.1 Sentiment-Werte mit numpy, scipy, matplotlib und seaborn erkunden

Nachdem Sie die Sentiment-Analyse durchgeführt und die Ergebnisse in Ihrem Datensatz gespeichert haben, empfiehlt es sich, die Ergebnisse zu erkunden. Hierfür werden wir in der Folge Maßzahlen der deskriptiven Statistik für den ganzen Datensatz und einzelne Gruppen verwenden, Grafiken erstellen sowie statistische Tests verwenden, die Gruppenvergleiche ermöglichen.

Deskriptive Statistik

Einen guten Einstieg in die Analyse der Daten bieten Ihnen die Lage- und Streuungsmaße, von denen Sie bestimmt schon einmal in Ihrer Statistikvorlesung gehört haben. Dabei handelt es sich um Minimum, Median, Mittelwert, Maximum und Standardabweichung. Falls Sie nochmals einen Überblick über Lage- und Streuungsmaße benötigen, können Sie in Kapitel 10.2.2 nachschlagen. All diese Werte können wir durch Befehle generieren, die im Paket `pandas` gespeichert sind. Dabei müssen Sie darauf achten, dass Sie die Daten, deren Werte Sie berechnen möchten, in den runden Klammern ansteuern. Tabelle 10.2 listet die entsprechenden Befehle auf.

Wenn Sie sich diese Maßzahlen in der Ipython Console anzeigen lassen möchten, dann können Sie diese in den `print()`-Befehl schreiben und mit F9 und vorhergehenden Anwählen oder „Strg + Enter“ ausführen. Um in unserem Falle die deskriptiven Maßzahlen für die Sentiment-Scores zu berechnen und anzeigen zu lassen, können Sie Code 10.47 verwenden. Im `print()`-Befehl lassen wir uns einen zuvor durch ein Komma getrennten Textstring (in Anführungszeichen) mitsamt der Ausgabe unseres `numpy`-Befehles anzeigen. Auf diese Weise gehen wir sicher, dass wir bei der Anzeige auch genau wissen, welcher Wert das Minimum etc. darstellt.

Tabelle 10.2 Numpy-Befehle für den Aufruf ausgewählter Lage- und Streuungsmaße

Maßzahl	Befehl
Minimum	numpy.min()
Median	numpy.median()
Mittelwert	numpy.mean()
Maximum	numpy.max()
Standardabweichung	numpy.std()

Code 10.47 Berechnung und Ausgabe der deskriptiven Statistiken für die errechneten Sentiment-Werte

```
# ================================================================================
# Einfache statistische Berechnungen
# ================================================================================

# Ausgabe der Punkt- und Streuungsmaße des Gesamtdatensatzes
print("Minimum:", np.min(df.sentiments))
print("Median:", np.median(df.sentiments))
print("Mittelwert:", np.mean(df.sentiments))
print("Maximum", np.max(df.sentiments))
print("Standardabweichung:", np.std(df.sentiments))
```

Neben der reinen Beschreibung der Ergebnisse unserer Sentiment-Analysen könnte uns aber auch ein Vergleich zwischen Gruppen oder die Veränderungen von Stimmungen im Zeitverlauf interessieren. Wir könnten beispielsweise fragen, ob die Kritiken immer negativer ausfallen, je länger ein*e Nutzer*in Filme und Serien bewertet. Wir könnten auch fragen, ob bestimmte Filmgenres im Zeitverlauf besser oder schlechter bewertet werden. Vielleicht gibt es aber auch eine Wertungsinflation und 5*-Bewertungen werden immer häufiger, auch wenn zugleich die Wörter, die negative Kritik ausdrücken sollen, häufiger werden. Wenden wir uns nun wieder dem Vergleich der Reviews mit 1*- und 5*-Bewertung zu.

Wenn Sie nun die deskriptiven Statistiken für eine dieser Gruppen mittels des `print()`-Befehls anzeigen lassen möchten, dann können Sie die 1*- und 5*-Bewertungen auswählen, indem Sie diese mittels einer Abfragebedingung selektieren und im Anschluss die oben angegebenen *numpy*-Berechnungen durchführen. Eine Selektion funktioniert ähnlich wie in RStudio mit einer ecki-

gen Klammer. Hinzu kommt eine logische Abfrage, mit der Sie die Auswahlkriterien für eine oder mehrere Variablen definieren (siehe Box 10.2).

Box 10.2: Übersicht über die logischen Operatoren in Python

Abfrage	Operator
Gleich	==
Kleiner	<
Kleiner gleich	<=
Größer	>
Größer gleich	>=
Nicht	!=
Und	&
Oder	|

In unserem Falle wollen wir untersuchen, inwiefern sich die Bewertungen von Film- und Serienreviews systematisch zwischen Reviews mit einem Stern und fünf Sternen unterscheiden. Wenn Sie nur die Fälle im Datensatz auswählen wollen, die fünf Sterne haben, dann können Sie (sofern Ihr Datensatz unter dem Namen `df` abgespeichert ist) `df[df.overall == 5]` wie in Code 10.48 eingeben. Das `df[]` sagt Python, dass nun eine Selektion folgt. Diese Selektion kann verschiedene Variablen beinhalten, die als Liste übergeben werden (z. B. `df[["overall","sentiments"]]`, um einen Teildatensatz zu erstellen), andererseits kann hier die eben angegebene logische Bedingung eingefügt werden, um nur bestimmte Zeilen zu behalten. Wenn Sie nach der Auswahl der Reviews mit Fünf-Sternebewertung nur die Sentiment-Werte ansteuern möchten, dann fügen Sie `.sentiment` hinzu, sodass insgesamt die Abfrage `df[df.overall == 5].sentiment` entsteht.[22] Sie können diese Werte einer neuen Variablen übergeben und diese im Anschluss mittels der `numpy`-Befehle erkunden oder diese Selektion verschachtelt in die `numpy`-Befehle einbetten. Dies würde dann wie folgt aussehen.

Code 10.48 Berechnung der Lage- und Streuungsmaße und Ausgabe mittels des print()-Befehls

```
# Ausgabe der Punkt- und Streuungsmaße für alle Reviews mit 5-Sternberwertung
print("Minimum:",np.min(df[df.overall == 5].sentiments))
print("Median:",np.median(df[df.overall == 5].sentiments))
print("Mittelwert:", np.mean(df[df.overall == 5].sentiments))
print("Maximum",np.max(df[df.overall == 5].sentiments))
print("Standardabweichung:",np.std(df[df.overall == 5].sentiments))
```

22 Sie können auch nach mehreren Bedingungen selektieren. Wenn Sie zum Beispiel alle Reviews mit Bewertungen mit weniger als 3* sowie mit 5* untersuchen wollen, dann können Sie diese Abfragen in Klammern setzen und mit einem logischen Und (`&`) bzw. einem Oder (`|`) miteinander verknüpfen. Im vorliegenden Falle würde eine solche Selektion `df[(df.overall < 3) | (df.overall == 5)]` lauten.

Um den Überblick zu behalten, können Sie wie in Code 10.48 demonstriert auch einen kleinen Datensatz erstellen, in den Sie Lage- und Streuungsmaße hineinspeichern. Wir legen Ihnen dies ans Herz, falls Sie einmal große Datensätze mit vielen Variablen und Gruppen analysieren müssen, wie dies beispielsweise beim Sozioökonomischen Panel der Fall ist. Im Folgenden erstellen wir einen Datensatz, dessen Spalten die Lage- und Streuungsmaße beinhaltet, während die Zeilen einmal die Werte für den gesamten Datensatz, einmal für die Reviews mit 1*-Bewertung und einmal für die Reviews mit 5*-Bewertung enthalten. Hierfür wollen wir zunächst die Spaltennamen bzw. Variablennamen festlegen. Diese speichern wir in einer einfachen Liste und nennen die neue Variable Spaltennamen. Ferner können wir die Zeilennamen auch schon festlegen, die wir später an den Datensatz übergeben wollen. Auch hierfür kann wieder eine Liste mit Namen übergeben werden. Beides zusammengenommen lässt sich in den folgenden Codezeilen realisieren.

Code 10.49 Anlegen der Zeilen- und Spaltennamen für einen eigenen Datensatz

```
# Generierung eines Datensatzes mit deskriptiven Statistiken
spaltennamen = ["Minimum","Median","Mittelwert","Maximum","Standardabweichung"]
zeilennamen = ["Gesamt","ein-Stern","fünf-Sterne"]
```

Danach erstellen wir eine Variable, die die Zeilen mit den Werten beinhalten wird und die, befüllt mit diesen Werten, an einen *pandas*-DataFrame übergeben wird. Um einen Datensatz zu generieren, können Sie eine Liste mit Tupeln befüllen. Die Liste selbst klammert den ganzen Datensatz ein, die Tupeln beinhalten dann Ihrerseits die Werte für die einzelnen Zeilen. Eine Tupel können Sie mit `()`, d. h. ohne separaten Befehl davor, erzeugen und mit Werten befüllen. Wenn Sie die Zellen einer Zeile mit den Worten „Zelle1“, „Zelle2“ und „Zelle3“ befüllen möchten, dann müssen Sie entsprechend `[("Zelle1", "Zelle2", "Zelle3")]` angeben. Um eine weitere Zeile mit den Zahlen eins bis drei hinzuzufügen, müsste dieses Objekt die Struktur `[("Zelle1", "Zelle2", "Zelle3"), (1,2,3)]` aufweisen. Achten Sie darauf, dass jedes Tupel die gleiche Anzahl an Elementen hat, da sonst der Datensatz später nicht erstellt werden kann![23] Beginnen wir damit, die erste Zeile mit allen Werten, wie in Code 10.50 angeführt, anzulegen.

23 Sie könnten dies mit einem Listenabgleich kontrollieren, bei dem sie die Anzahl mittels des `len()`-Befehls abfragen. Zum Beispiel: `[len(t) for t in {Liste mit Datenzeilen}]`. Dabei steht der Iterator t als Platzhalter für Tupel, während Sie den letzten Teil der Listenabfrage durch den Namen Ihrer Liste ersetzen müssen.

Code 10.50 Festlegen der ersten Zeile eines neuen Datensatzes, der Lage- und Streuungsmaßen der Sentiment-Analyse des Gesamtkorpus enthält

```
# Festlegung der ersten Zeile
zeilen = [(np.min(df.sentiments), np.median(df.sentiments), \
                    np.mean(df.sentiments),np.max(df.sentiments), \
                    np.std(df.sentiments))]
```

Nun wollen wir die Werte der Einzelgruppen ergänzen. Hierfür müssen wir zunächst einen Teildatensatz erstellen, in dem zunächst alle 1*-Bewertungen und danach alle 5*-Bewertungen aufgeführt sind. Diese wollen wir im nächsten Schritt als Tupel der Liste hinzufügen, der wir den Namen `zeilen` gegeben haben. Um die Werte für alle in overall gespeicherten Gruppen zu erhalten, können wir eine for-Schleife schreiben, die die Lage- und Streuungsmaße der Gruppen berechnet und dann als Tupel der Liste hinzufügt, aus der wir den Datensatz generieren wollen. Da unser Datensatz insgesamt 10 000 Zeilen lang ist, aber nur zwei Werte besitzt, wollen wir nicht, dass unsere Schleife auch 10 000 mal durchläuft. Das würde einerseits die Berechnungszeit stark erhöhen, andererseits hätten Sie entsprechend viele Duplikate in Ihren Daten, die Sie nachher nochmals entfernen müssten. Bei noch größeren Datenmengen würden Sie eine extrem große Datenmatrix erzeugen und im Arbeitsspeicher Ihres Rechners zwischenspeichern, was Python mit einem Fehler, einem Streik (es wird nicht mehr weiterberechnet) oder einem Systemabsturz quittieren würde. Ihr PC hat also auch Möglichkeiten zur Hand, passiven Widerstand gegen Ihre Eingaben zu leisten.

Um uns auf diese zwei Werte zu beschränken, können wir `list(set(df.overall))` eingeben. Der `set()`-Befehl extrahiert Ihnen alle einzigartigen Werte. Durch die Verschachtelung mit dem `list()`-Befehl werden diese Werte in eine Liste übersetzt, die Ihre for-Schleife dann wie in Code 10.51 ansteuern kann. Danach erstellen Sie einen Teildatensatz mittels einer logischen Abfrage nach dem gleichen Muster wie bei der Ausgabe der Lage- und Streuungsmaße für die Reviews mit 5*-Bewertung. Die Struktur dieser Schleife lautet wie folgt.

Code 10.51 Erstellen einer neuen Liste für den Datensatz

```
# Hinzufügen weiterer Zeilen
for bewertung in list(set(df.overall)):
    df_part = df[df.overall == bewertung]
    zeilen.append(
        (np.min(df_part.sentiments), np.median(df_part.sentiments), \
                    np.mean(df_part.sentiments),np.max(df_part.sentiments), \
                    np.std(df_part.sentiments)))
```

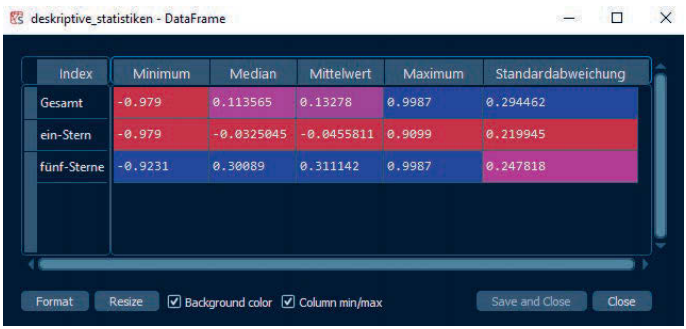
Wie in Code 10.52 erstellen wir einen neuen *pandas*-DataFrame mit `pd.DataFrame()`. Die Optionen, die wir zur Spezifizierung des Datensatzes benötigen, sind `data`, `columns` und `index`. In `data` übergeben sie mit einem `=` die Datenzeilen, in `columns` die Spaltennamen und in `index` die Zeilennamen. Vergessen Sie im Anschluss nicht, überflüssige Variablen mit `del()` aus dem Speicher zu entfernen, damit Ihr Speicher nicht vollläuft.

Code 10.52 Erstellung eines pandas-DataFrames mit Lage- und Streuungsmaßen der Sentiment-Analyse

```
deskriptive_statistiken = pd.DataFrame(data = zeilen, \
                                       columns = spaltennamen, \
                                       index = zeilennamen)
del(df_part,zeilen,spaltennamen)
```

Wenn Sie oben rechts im Variable Explorer auf den neuen Datensatz (hier als Objekt `deskriptive_statistiken` gespeichert) doppelklicken, dann bekommen Sie in etwa die Ansicht aus Abbildung 10.10, die Ihnen die Lage- und Streuungsmaße für den gesamten Datensatz sowie die beiden Teilgruppen anzeigt, für die wir die Analyse durchgeführt haben. Dabei sind niedrige Werte in Rot, hohe Werte in Blau hervorgehoben.

Abbildung 10.10 Ansicht des erzeugten Datensatzes mit Lage- und Streuungsmaßen



Grafiken erstellen

Wir können Gruppenunterschiede auch mittels Grafiken visualisieren. Dafür benötigen wir Befehle, die sich in den Paketen *matplotlib* und *seaborn* befinden. In Anlehnung an die Grafiken, die wir in RStudio erzeugt haben, werden wir eine Grafik mit je einem Barplot und Kerndichteschätzungen für alle Bewertungen und pro Gruppe (1*-Reviews und 5*-Reviews) erstellen. Wir gehen dabei so vor, dass wir zuerst einen Grafikcontainer erstellen, in dem wir danach unsere Grafik wie auf eine Leinwand malen werden. Im Anschluss erstellen wir je eine Grafik pro Gruppe, ehe eine Legende folgt. Als letzten Schritt werden wir die Grafik als png-Datei mit einer hohen Bildauflösung abspeichern. Die erstellte Grafik mit den drei Dichteplots und Histogrammen sehen Sie bei Abbildung 10.11.

Unseren Grafikcontainer erstellen wir mit dem Befehl `figure()`, der in `matplotlib pyplot` abgelegt ist. Wir können in den Klammern die Größe festlegen, indem wir die Option `figsize` eingeben und nach einem `=` in eckigen Klammern die Breite und Höhe angeben. Standardmäßig wird die Größe in Inches angegeben.[24] Danach können wir die Histogramme und Dichteplots mit dem `distplot()`-Befehl aufrufen, der im `seaborn`-Paket gespeichert ist. In diesem Befehl gibt man zunächst die Daten an, die man im Graph angezeigt hat. In unserem ersten Fall sind diese Grafiken in df.sentiments gespeichert. Darüber hinaus können wir die Details des Histogramms und Dichteplots in den Optionen angeben. Im gleich folgenden Codebeispiel stellen wir durch `bins=50` ein, dass das Histogramm 50 Balken umfassen soll. Mit der `label`-Option können wir dem Plot einen Namen geben. Sofern wir einen Namen vergeben, kann dieser später in der Legende aufgegriffen werden. Zuletzt wollen wir eine distinkte Farbe für jedes Histogramm und dazugehörigen Distanzplot vergeben. Das machen wir mittels der color-Option.[25] Daneben gibt es auch Farbpaletten, die Sie separat aufrufen können. Bevor wir die Grafik abspeichern, fügen wir noch mit `plt.legend()` eine Legende hinzu, die ohne Optionen automatisch in eine möglichst nicht durch Grafiken ausgefüllte Ecke des Grafikcontainers platziert wird. Damit die Grafik nicht übermäßig große, weiße Abstände an den Rändern lässt, können wir sie mit dem `tight-layout()`-Befehl zurechtziehen. Der volle Code 10.53, mit dem Sie Abbildung 10.4 erzeugen können, lautet wie folgt.

---

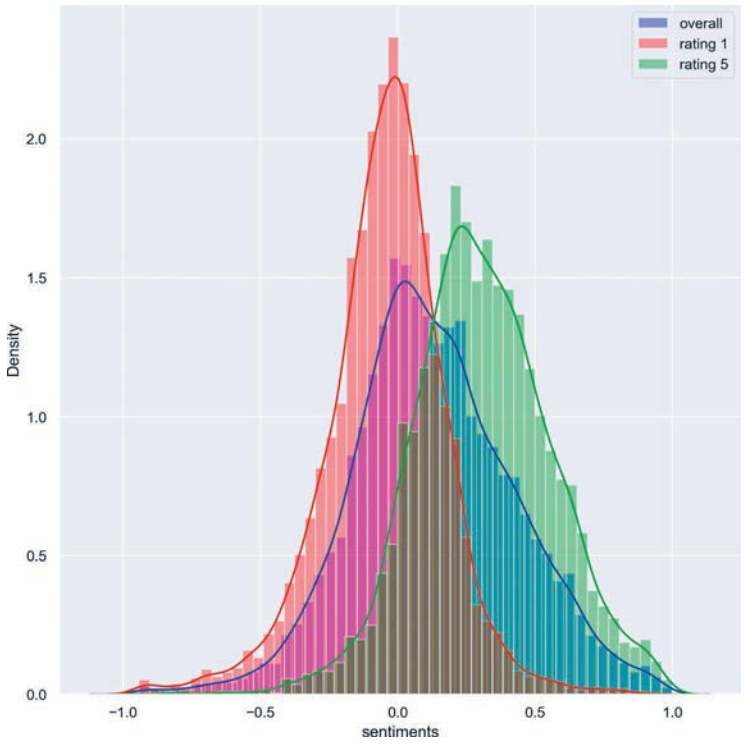
24 Sie können die Größe umrechnen, indem Sie eine Variable definieren, diese `cm` nennen und ihr den Wert `1/2.54` übergeben. Entsprechend würden sie `plt.figure(figsize=-[8*cm,8*cm])` eingeben, um eine Grafik mit den Maßen 8cm * 8cm zu erhalten.

25 Die Website *https://matplotlib.org/stable/gallery/color/named_colors.html* bietet Ihnen einen Überblick über die einzelnen Farben und deren Namen.

Code 10.53 Erstellung des Histogramms und Dichteplots mittels Befehlen aus der matplotlib und seaborn

```
# ============================================================================
# Erstellen der Grafiken mittels matplotlib und seaborn
# ============================================================================

plt.figure(figsize=[8,8])
sbs.distplot(df.sentiments, bins=50, label="overall", color="blue")
sbs.distplot(df[df.overall == 1].sentiments, bins=50, label="rating 1",
color="red")
sbs.distplot(df[df.overall == 5].sentiments, bins=50, label="rating 5",
color="green")
plt.legend()
plt.tight_layout()
```

Abbildung 10.11 Histogramme und Dichteplots der Sentiments getrennt nach gesamten Datensatz sowie 1*- und 5*-Bewertungen



Um die Abbildung zu speichern, empfiehlt es sich, Python zunächst auf einen von Ihnen vordefinierten Ordner zugreifen zu lassen, in dem Sie Ihre Abbildungen speichern möchten, dann die Grafik zu exportieren und zuletzt zu schließen. Auf diese Weise behalten Sie sowohl auf Ihrer Festplatte als auch im „Plot"-Fenster des Variable Explorers die Übersicht. Den Dateipfad können Sie mit `os.chdir()` verändern. Den Pfad haben wir einfach output genannt, damit unsere Grafiken von Daten und Codedateien getrennt gespeichert werden. In der `matplotlib` findet sich der Befehl `savefig()`, mit dem Sie die Grafik auf Ihre Festplatte speichern können. Hierfür geben Sie im Anschluss an den Grafikcontainer, den Sie erstellt haben, in den Klammern den gewünschten Namen Ihrer Abbildung ein. Wenn Sie eine höhere Auflösung wünschen – was von Druckereien beispielsweise beim Druck Ihrer Abschlussarbeit verlangt wird – dann können Sie die Größe mit der dpi-Option einstellen. Die Abbildung 10.4 weiter oben wurde mit einem dpi-Wert von 300 abgespeichert. Zuletzt schließen Sie den Grafikcontainer mittels der `close()`-Funktion. Dieser Speichervorgang kann wie folgt in Code 10.54 ausgedrückt werden.

Code 10.54 Speichern einer Grafik durch Befehle der matplotlib

```
os.chdir(output)
plt.savefig("Abbildung_Histogramme_und_Densityplot.png", dpi=300)
plt.close()
```

Gruppenvergleiche mit statistischen Maßzahlen

Zwar deuten sowohl die Grafiken als auch die deskriptiven Statistiken darauf hin, dass es auch bei der Sentiment-Analyse in Python einen Unterschied im Sentiment der Rezensionen mit 5* und 1* gibt. Dennoch können wir, wie im Falle der Sentiment-Analyse in R, nicht einfach davon ausgehen, dass ein wesentlicher Unterschied in der Stimmung der Rezensionen zwischen den beiden Gruppen besteht.

Wie im Kapitel 10.2.3 (Mittelwertdifferenztests) für RStudio ausgeführt, können wir entweder t-Test, Welsh-Tests oder Wilcoxon-Tests verwenden, um Mittelwertdifferenzen zwischen Gruppen zu ermitteln. Schlagen Sie bitte in dem Kapitel nach, wenn Sie noch weitere Erläuterungen lesen möchten. Im Falle der ersten beiden Tests müssen die geprüften Variablen metrisches Skalenniveau aufweisen. Dabei müssen die Werte der betrachteten Variablen (Sentiment-Scores) sowohl beim t-Test als auch beim Welsh-Tests einer Normalverteilung folgen. Beim t-Test muss zudem Varianzgleichheit zwischen den Gruppen vorliegen. Sind diese Voraussetzungen nicht gegeben, können Sie den Wilcoxon-Test anwenden. Dieser hingegen bemüht nur die Rangdifferenzen der Werte zwischen beiden Gruppen, wobei die Ränge aufsteigend von 1 beim niedrigsten und N

(= höchster Rang) beim höchsten Wert sind. Diesen Rängen werden die einzelnen Gruppen zugewiesen und dann geprüft, ob bei einer der Gruppen systematisch höhere oder niedrigere Rangwerte zugewiesen werden.

Beginnen wir nun damit, die Annahmen zu prüfen. Da wir kontinuierliche Werte für den Sentiment-Score haben, die von −1 bis +1 reichen, können wir die erste Annahme als erfüllt betrachten. Nun wollen wir aber auch wissen, ob die Sentiment-Werte beider Gruppen einer Normalverteilung folgen. Dies können wir durch die Anwendung des Shapiro-Wilk-Tests prüfen, der in der Python Library `scikit-learn` in der Bibliothek `stats` gespeichert ist und in unserem Falle durch `st.shapiro()` aufgerufen werden kann. In die runden Klammern fügen Sie einfach das Objekt oder die Variable eines Datensatzes ein, die Sie testen wollen, beispielsweise `st.shapiro(df.sentiments)` für alle Sentiment-Scores oder `st.shapiro(df[df.overall == 5].sentiments)` für die Sentiment-Scores der 5*-Bewertungen. Als Ausgabe bekommen Sie ein Objekt, in dem an erster Stelle die Teststatistik, an zweiter Stelle der Signifikanzwert ausgegeben wird. Doch warum ist dieser so wichtig und was bedeutet er in unserem Falle?

Ganz grundsätzlich geht der Shapiro-Wilk-Test davon aus, dass bei den Werten, die wir betrachten, eine Normalverteilung vorliegt. Diese Annahme, die wir statistisch testen, ist unsere Nullhypothese (H0), ergo, dass unsere Sentiment-Werte in der jeweiligen Gruppe normalverteilt sind. Die Gegenhypothese (H1) hierzu lautet, dass unsere Sentiment-Werte nicht normalverteilt sind. Dass die Hypothesen so – und nicht andersherum – formuliert wurden, hat den Grund, dass wir die Wahrscheinlichkeit möglichst geringhalten wollen, dass wir fälschlicherweise von einer Normalverteilung der Variablen ausgehen, d. h. die Nullhypothese fälschlicherweise ablehnen. Dabei handelt es sich um den Alpha-Fehler. Schlimmer wäre ein Beta-Fehler, d. h. hier, dass wir fälschlicherweise die Nullhypothese ablehnen würden. Wenn wir fehlerhafterweise davon ausgehen, dass keine Normalverteilung der Sentiments vorliegt, dann können wir noch immer den Wilcoxon-Test durchführen. Umgekehrt würden wir fälschlicherweise einen t-Test oder Welsh-Test durchführen und zu falschen Schlussfolgerungen kommen. Der Signifikanzwert, auch p-Wert genannt, zeigt dabei die Wahrscheinlichkeit an, die Nullhypothese fälschlicherweise anzunehmen. Damit zeigt er zugleich die Wahrscheinlichkeit für einen Alpha-Fehler an. Ist dieser Wert bei 0.05, dann bedeutet dies, dass wir in 5 % der Fälle fälschlicherweise die Nullhypothese ablehnen. Idealerweise sollte dieser Wert möglichst gering, höchstens aber bei 0.05 liegen, damit Sie einigermaßen sicher sind, dass das Sentiment keiner Normalverteilung folgt.

Technisch gesehen wollen wir die Werte für beide Gruppen testen und in der Console ausgeben lassen, ob unsere Sentimentwerte für die einzelnen Gruppen einer Normalverteilung folgen. Hierzu schreiben wir zunächst eine Schleife, die zunächst alle Bewertungsgruppen, in unserem Falle Bewertungen mit 1* und 5*, ansteuert. Das können Sie wieder über die `list(set())`-Befehlsverschachte-

lung erreichen und diese Werte an eine for-Schleife übergeben. Dann prüfen wir, ob die p-Werte kleiner als 0.05 sind. Dies erreichen wir, in dem wir den Shapiro-Wilk-Test durchführen und nach dem Befehl eine `[1]` eingeben, da unter dieser „Adresse" im Testergebnis die p-Werte gespeichert sind. Diese Prüfung können wir in eine if-else-Abfrage schreiben und anschließend in einem `print()`-Befehl angeben, ob dieser p-Wert unterschritten wurde oder nicht. Dieser Logik folgend, geben wir in Python Code 10.55 ein.

Code 10.55 Durchführung eines Shapiro-Wilk-Tests in Python und Ausgabe der Ergebnisse

```
# ============================================================================
# Test auf Normalverteilung
# H0: Verteilung folgt Normalverteilung
# H1: Verteilung folgt keiner Normalverteilung
# Wenn p < 0.05, dann liegt keine Normalverteilung vor
# ============================================================================

for i in list(set(df.overall)):
    if st.shapiro(df[df.overall == i].sentiments)[1] < 0.05:
        print("Normalverteilung für", str(i),"*-Bewertung wird nicht
              eingehalten")
    else:
        print("Normalverteilung für", str(i), *-Bewertung "wird eingehalten")
```

Wichtig ist, dass Sie im `print()`-Befehl den Iterator (hier: `i`) separat als Zeichenfolge bzw. String mit `str(i)` ansteuern, da Sie sonst einen Fehler erhalten. Das liegt daran, dass die Bewertungen in einem ganzzahligen Format, also als Integer in unserem Datensatz gespeichert werden, der `print()`-Befehl aber nur Zeichenfolgen ausgeben kann. Wenn wir diese Befehlszeilen ausführen, erhalten wir einen Output, der uns klar zu verstehen gibt, dass die Werte in den beiden Gruppen keiner Normalverteilung folgen und wir somit den Wilcoxon-Test verwenden sollten.

Output 10.12 Ausgabe der Ergebnisse des Shapiro-Wilk-Tests

```
Out[8]:
Normalverteilung für 1 *-Bewertung wird nicht eingehalten
Normalverteilung für 5 *-Bewertung wird nicht eingehalten
```

Gehen wir allerdings an dieser Stelle davon aus, dass die Normalverteilung unserer Sentiments eingehalten wird. Dann müssten wir die beiden Gruppen, die wir testen wollen, auf Varianzgleichheit testen. Varianzgleichheit bzw. Varianzhomogenität können wir mittels des Levene-Tests prüfen. Der Levene-Test ist, wie der Shapiro-Wilk-Test, im Paket `scikit-learn` und hier in der `stats`-Bibliothek aufzufinden und in unserem Falle mit `st.levene()` aufrufbar. Er folgt im Prinzip der gleichen Logik wie der Shapiro-Wilk-Test und geht davon aus, dass die Varianzen zwischen zwei Gruppen gleich (H0) bzw. ungleich sind (H1). Anders als beim Shapiro-Wilk-Test müssen wir hier aber die beiden Gruppen, die wir gegeneinander testen wollen, in den Befehl eingeben. Somit lautet dessen Grundstruktur `st.levene([GRUPPE1], [GRUPPE2])`. Sie müssen die beiden Gruppen durch die Sentiment-Werte der 1*- und 5*-Bewertungen ersetzen, um diesen Test durchzuführen. Als Ausgabe erhalten Sie wieder die Teststatistik und den p-Wert. Ersteren können Sie mit `st.levene([GRUPPE1], [GRUPPE2])[0]` und Letzteren durch Aufruf von `st.levene([GRUPPE1], [GRUPPE2])[1]` ansteuern. Wenn wir eine Ausgabe des Tests mittels eines print()-Befehles schreiben wollen, dann können wir uns an den Codezeilen des Shapiro-Wilk-Tests orientieren. Da wir einen direkten Vergleich haben, benötigen wir hierfür nur eine `if-else`-Abfrage, nicht aber die `for`-Schleife, wie Sie anhand Code 10.56 erkennen können.

Code 10.56 Code zur Durchführung des Levene-Tests in Python

```
# ============================================================================
# Levene-Test auf Varianzhomogenität
# H0: Varianzen zwischen zwei Gruppen sind gleich
# H1: Varianzen sind ungleich zwischen zwei Gruppen
# ============================================================================

if st.levene(df[df.overall == 1].sentiments, df[df.overall == 5].sentiments)
[1] < 0.05:
    print("Varianzen der beiden Gruppen sind gleich")
else:
    print("Varianzen der beiden Gruppen sind nicht gleich")
```

Sie erhalten nach der Ausführung des Codes folgende Ausgabe.

Output 10.13 Ausgabe des Levene-Tests auf Varianzgleichheit zwischen 1*- und 5*-Bewertungen

```
Out[9]:
Varianzen der beiden Gruppen sind gleich
```

Die Ausgabe zeigt Ihnen an, dass die Varianzen der beiden Gruppen gleich sind und somit die letzte der drei Annahmen erfüllt ist. Da allerdings die zweite Annahme (Normalverteilungsannahme) nicht erfüllt ist, können wir nur einen Wilcoxon-Rangsummentest durchführen, um die Unterschiede in den Sentiment-Werten zu ermitteln.

Zuletzt werden wir noch einen t-Test, einen Welsh-Test und den Wilcoxon-Test durchführen (Code 10.57). Jeder dieser Tests erzeugt Ihnen, analog zu den beiden vorangegangenen Tests, zunächst eine Teststatistik und dann einen p-Wert. Dabei teilen sich t-Test und Welsh-Test einen Befehl, nämlich `st.ttest_ind([GRUPPE1], [GRUPPE2])`. Wie Sie sehen, ist auch dieser Test in der Bibliothek `stats` in `scikit-learn` abgelegt und folgt dem Aufbau des Levene-Tests. Der Unterschied hier ist die Option `equal_var`, was eine Kurzform von *equal variance* (oder Varianzgleichheit auf Deutsch) ist. Wenn Sie dem Test mit `True` angehen, dass die Varianzen der betrachteten Verteilungen gleich sind, dann führen Sie einen t-Test durch. Geben Sie hier `False` an, dann führen Sie einen Welsh-Test aus. Wenn Sie einen Wilcoxon-Test durchführen wollen, dann geben Sie `st.wilcoxon([Gruppe1], [Gruppe2])` an und ersetzten die eckigen Klammern und die Gruppen durch die Variablen bzw. die Gruppen, die Sie gegeneinander testen möchten. Bei allen Tests würden p-Werte < 0.05 auf den Vergleich der Sentiments zwischen den 5* und 1* Bewertungen übertragen bedeuten, dass die Sentiment-Werte beider Gruppen mit hoher Wahrscheinlichkeit verschieden sind. Wenn wir die Fehlerwahrscheinlichkeit mit einem `print()`-Befehl ausgeben wollen, dann können wir das Ergebnis des geeigneten Tests einer Variablen übergeben (hier folgend: Wilcoxon) und diese dann in einer Print-Ausgabe ansteuern, wie die folgende Codeübersicht anzeigt. Bedenken Sie aber, dass Sie die Ergebnisse Ihrer Tests mit einem Gleichzeichen auch an neue Variablen (z. B. eine Liste, ein DataFrame) übergeben und damit weiterarbeiten können.

Code 10.57 Durchführung der Mittelwertdifferenztests zwischen Gruppen: t-Test, Welsh-Test und Wilcoxon-Rangsummentest (Fortsetzung nächste Seite)

```
## Dennoch testweise Durchführung des T-Tests und des Welsh-Tests
#T-Test
st.ttest_ind(df[df.overall == 1].sentiments,
```

```
            df[df.overall == 5].sentiments,
            equal_var=True)

#Welsh-Test
st.ttest_ind(df[df.overall == 1].sentiments,
            df[df.overall == 5].sentiments,
            equal_var=False)

# Wilcoxon rank-sum test
wilcoxon = st.wilcoxon(df[df.overall == 1].sentiments,
            df[df.overall == 5].sentiments)

print("Alpha-Fehlerwahrscheinlichkeit, dass der Mittelwert"\
      ,"der beiden Gruppen gleich ist"\
      ,"beträgt:", wilcoxon[1])
```

Wenn Sie die Zeilen markieren und mit F9 ausführen, dann teilt Ihnen Ihre Ipython Console nun freundlich mit, dass die Wahrscheinlichkeit, dass die Mittelwerte beider Gruppen gleich sind, wir sie aber fälschlicherweise als unterschiedlich interpretieren, gleich null ist. Das deutet Ihnen an, dass die Mittelwerte beider Gruppen unterschiedlich sind. In Kombination mit den deskriptiven Statistiken und den Histogrammen und Dichteplots können Sie nun sicher davon ausgehen, dass 5*-Bewertungen ein durchschnittlich positiveres Sentiment aufweisen als 1*-Bewertungen von Filmen und Fernsehserien.

Output 10.14 Ausgabe des Wilcoxon-Rangsummentests mittels des print()-Befehls

```
Alpha-Fehlerwahrscheinlichkeit, dass der Mittelwert der beiden Gruppen gleich
ist, beträgt: 0.0
```

10.4 Zusammenfassung und abschließende Worte

Dieses Kapitel hatte das Ziel, Ihnen zu zeigen, wie die Daten aufbereitet werden müssen, um eine Sentiment-Analyse durchzuführen und wie diese im Anschluss ausgeführt wird. Ferner wurde Ihnen demonstriert, wie Sie Vergleiche von Sentiments zwischen Gruppen berechnen können und worauf Sie hierbei zu achten haben. Bedenken Sie dabei aber stets, dass dies nur der erste Schritt ist, den wir hier diskutiert haben. In der Regel werden Sentiment-Analysen für die

Veränderungen von Stimmungen im Zeitverlauf (beispielweise in Bezug auf den medialen Diskurs um Windenergie in deutschen Zeitungen bei Dehler-Holland et al. 2021), der positiven bzw. negativen Kontextualisierung verschiedener Gruppen (Heidenreich et al. 2020) oder der Positionierung von politischen Akteuren innerhalb (parlamentarischer) Debatten verwendet (Abercrombie und Batista-Navarro 2020). Das gleiche kann auch in Kombination mit Topic Modeling (siehe Kapitel 11) verwendet werden, um die positive/negative Haltung von Akteur*innen oder Akteursgruppen zu bestimmten Themen (und das im Zeitverlauf!) zu modellieren.

Der vorliegende Datensatz wurde aus Übungsgründen sehr eingeschränkt. Dennoch könnten Sie beispielsweise prüfen, ob sich einzelne Reviewer*innen, die häufiger Rezensionen schreiben, durch besonders positive oder negative Kritiken auszeichnen. Alternativ können Sie ermitteln, ob Reviewer*innen Filme unterschiedlicher Genres präferieren (z. B. anhand der Anzahl der Kritiken pro Genre) und ob sie Filme in Genres, mit denen sie vertraut sind, kritischer und damit auch negativer beschreiben. Zudem können Sie nach Zeiträumen vergleichen, ob Filme und Serien zu Beginn, also kurz nach deren Veröffentlichung, oder später bessere Rezensionen bekommen.

Jedoch sollten Sie darauf achten, dass Sentiment-Analysen und die Zuordnung zu positiven oder negativen Sentiments nicht perfekt sind. Sie können nur mit einer gewissen Sicherheit sagen, dass es sich um positive bzw. negative Sentiments handelt. Bei genauerer Betrachtung der deskriptiven Statistiken sowie der Histogramme können Sie sehen, dass es 1*-Bewertungen gab, die sehr positive Sentiment-Werte hatten, und 5*-Bewertungen, denen extrem negative Sentiments zugewiesen wurden. Aufgrund dieser Unsicherheiten sollten Sie die Ergebnisse mit gebotener Vorsicht interpretieren und die Texte, in denen Anomalien vorkommen, qualitativ sichten! Zudem können Sie, wenn Sie den Verlauf von Sentiments über die Zeit hinweg betrachten, den Verlauf selbst oder Ausschläge ins Positive oder Negative dazu nutzen, um Texte für eine qualitative Untersuchung zu verwenden, die Ihnen genaueren Aufschluss darüber gibt, was in diesen Zeiträumen geschehen ist. Gleiches gilt für Gruppen, z. B. wenn positiver über bestimmte Migrant*innengruppen als über andere berichtet oder auf Social Media-Plattformen wie Twitter gesprochen wird. Dadurch, dass die Sentiment-Analyse sehr weit vom Text entfernt ist und die Stimmungslage auf eine einfache Maßzahl reduziert wird, verbleibt sie auf der manifesten Ebene der Interpretation. Um den latenten Sinngehalt der Kommunikationen zu erschließen, benötigen Sie entsprechende Verfahren der qualitativen Inhaltsanalyse.

11. Topic Modeling mittels Latent Dirichlet Allocation

Dieses Kapitel hat das Ziel, Ihnen eine Einführung in das Topic Modeling mit dem Python-Paket gensim zu geben. Wir fokussieren uns dabei auf die Latent Dirichlet Allocation (LDA) als eine Methode des Topic Modelings und zeigen Ihnen am Beispiel eines Korpus aus Filmskripten, wie Sie die Daten aufbereiten müssen, eine LDA technisch umgesetzt wird, welche Kriterien Sie für die Modellauswahl anlegen, wie Sie die Themen interpretieren und wie Sie das Modell visualisieren können. In diesem Kapitel werden circa 16 Seiten für die Syntax aufgewandt und Vorgehen sowie Output in neun Abbildungen und 18 Tabellen visualisiert.

11.1 Einleitung

In diesem Kapitel wenden wir uns nun dem Topic Modeling zu. Darunter werden Verfahren gefasst, mit deren Hilfe Sie potenziell vorliegende Themen innerhalb eines großen Textkorpus finden können, ohne alle Texte im Korpus gesichtet oder bearbeitet zu haben. Dennoch sollten Sie ohne gute Kenntnis des Untersuchungsgegenstandes kein Topic Modeling durchführen, denn ohne dessen Kenntnis können Sie zwar die Auswertungstechnik anwenden (z. B. weil ein Datenkorpus verfügbar ist), jedoch nicht die Themen des Topic Modeling verstehen, interpretieren und Dritten verständlich machen (siehe Kapitel 2.1.2). Entsprechend der Anforderungen des Topic Modeling werden wir Ihnen in diesem Kapitel die Sachkenntnis des Untersuchungsgegenstandes (z. B. für die systematische Vorbereitung der Untersuchung) bereitstellen, mit Code-Beispielen das Programmieren in Python vorstellen und rudimentär die notwendigen statistischen Hintergründe der Auswertungstechnik darlegen.

Aus dem Portfolio der Topic Modeling-Verfahren (z. B. latente Semantikanalyse, Structural Topic Model, Correlated Topic Model) haben wir für die Einführung die *Latent Dirichlet Allocation* als weit verbreitetes Topic Modeling-Verfahren ausgesucht (Jelodar et al. 2019). Die Latent Dirichlet

Box 11.1: Sprachdurcheinander
In vorherigen Kapiteln haben wir Sie darauf hingewiesen, dass der Soziolekt, d. h., die Umgangs- oder Fachsprache von Soziolog*innen und Sozialwissenschaftler*innen, Ihnen anfangs ungewohnt erscheinen mag, dass er jedoch die Kommunikation erleichtert. In diesem Kapitel möchten wir Sie darauf vorbereiten, dass Sie ein Deutsch-Englisch-Sprachdurcheinander antreffen werden, denn die Fachtermini und Code-Befehle sind auf Englisch. Davon sollten Sie sich nicht abschrecken lassen, denn schon Ihr (unterstelltes) Schulenglisch reicht aus, um beispielsweise im Begriff „probabilistisch" das englische *probability*, also Wahrscheinlichkeit, erkennen zu können. Das vollständige Übersetzen von englischen Fachtermini erscheint uns auch nicht sinnvoll, da Sie sonst nicht anschlussfähig kommunizieren können (z. B. auf einschlägigen Online-Plattformen).

Allocation, kurz LDA, wurde von Blei, Ng und Jordan (2003) mit dem Ziel entwickelt, Themen in großen Textkorpora unter Verwendung eines statistischen Modells zu ermöglichen. Dabei handelt es sich um ein probabilistisches Modell. Probabilistisch bedeutet, dass jedes Token (= Worteinheit oder Wortkette) eines Textkorpus jedem Thema mit einer bestimmten Wahrscheinlichkeit zugeordnet wird.[1]

Das statistische Verfahren, das der LDA zugrunde liegt, basiert auf der sogenannten „bag-of-words"-Annahme. Die „bag-of-words"-Annahme sagt aus, dass Wörter innerhalb von Texten austauschbar sind und die Reihenfolge ignoriert werden kann, in der die Worte in unterschiedlichen Texten im zu analysierenden Textkorpus auftreten (Blei et al. 2003, S. 994). Folglich reicht das gemeinsame (mehr oder minder häufige) Auftreten von Wörtern innerhalb einer großen Anzahl von Texten aus, um Themen zu identifizieren. In anderen Worten: Die „bag-of-words"-Annahme beruht auf einer Nachbarschaftsannahme von häufig im Textkorpus miteinander auftretenden Worten. Wie die „bag-of-words"-Annahme erkennen lässt, kann und wird ohne menschliches Zutun kein *machine learning* (maschinelles Lernen) stattfinden und damit auch keine nachvollziehbare LDA von einem umfangreichen Textkorpus erfolgen. Beispielsweise müssen wir im Vorfeld bestimmen, welche Anzahl an Themen die LDA in den Texten erkennen soll. In der Folge erstellt ein Topic Model eine Zuordnung von Wörtern zu Themen sowie von Texten zu Themen. Dabei gehen wir davon aus, dass die durch dieses Verfahren gefundenen Themen die latenten Sinnstrukturen von Texten offenlegen, vorausgesetzt, es werden hinreichend viele Texte analysiert. Somit gilt es, einerseits viele Topic Models mit unterschiedlichen Themenzahlen zu berechnen und die Themen andererseits qualitativ zu interpretieren. Erst dadurch können wir erkennen, ob die von der LDA gefundenen Themen nachvollziehbar sind und latente Inhalte offenlegen. Latente Inhalte offenlegen bedeutet, dass Sinnstrukturen offenbar werden, welche nicht auf den ersten Blick erkennbar, also manifest sind (siehe Tabelle 2.1).

11.1.1 Schritt für Schritt-Ablauf des Topic Modelings

Wie in den vorhergehenden Kapiteln ist auch dieses Kapitel in mehrere Abschnitte aufgegliedert, die den einzelnen Arbeitsschritten in Abbildung 11.1 folgen. Da wir Ihnen bereits im vorangegangenen Kapitel gezeigt haben, wie Sie Python, Anaconda und Spyder installieren, stellen wir hier nur in Kürze die Daten und die für die Auswertung benötigten Pakete vor (siehe zu Copylefts und Copyrights Kapitel 1.3).

---

1 Diese Zuordnung betrifft einzelne Wörter (= Unigramme) und Wortketten (= Bigramme im Falle von zwei Wörtern, N-Gramme für drei und mehr Wörter) gleichermaßen.

Abbildung 11.1 Schritt für Schritt-Vorgehen für die Durchführung einer Latent Dirichlet Allocation

Schritt 1	**Vorbereitung und Installation** 1. R und RStudio oder Python, Anaconda und Spyder installieren 2. Pakete installieren 3. Pakete und Dateien in RStudio oder Spyder laden
Schritt 2	**Aufbereiten der Daten** 1. Tokenisieren der Texte 2. Festlegen der Stopwords 3. Selektion von Wortarten 4. Auftrennen langer Texte in Abschnitte 5. Entfernung von Eigennamen mittels eigener Stopword-Liste 6. Lexikon erzeugen (= Wort-Vektor-Verbindung erzeugen) 7. Datenkorpus im Bag of Words-Format erzeugen 8. Gegebenenfalls Korpus gewichten (tfidf)
Schritt 3	**Durchführung der Latent Dirichlet Allocation** 1. Festlegung der Themenzahl und Größe des Trainingssets 2. Erstellung von Modellen und Finetuning der Modellparameter
Schritt 4	**Modelle für die weitere Analyse auswählen** 1. Kohärenzmaße vergleichen 2. Texte qualitativ sichten, die charakteristisch für die durch den Algorithmus gefundenen Themen sind 3. Gegebenenfalls Stopwords ergänzen, Korpus gewichten und danach Schritte 12 & 13 wiederholen sowie Modellparameter verändern
Schritt 5	**Visualisierung der Ergebnisse** 1. Erstellen einer interaktiven HTML-Datei 2. Erstellen einer Wort x-Themen-Heatmap 3. Erstellung einer Text x-Themen-Heatmap 4. Erstellen einer Grafik, um Zeittrends darzustellen
Schritt 6	Ergebnisse zusammenfassen und Erstellung einer Präsentation, Studienarbeit und/oder Publikation
Schritt 7	Sichere Archivierung der Daten und, wenn möglich, Aufbereitung zur Nachnutzung

11.1.2 Daten und Forschungsfrage

Der Textkorpus, den wir als Beispiel verwenden, besteht aus 626 Filmskripten, in denen Wissenschaft thematisiert wird oder Wissenschaftler*innen dargestellt werden. Der Datenkorpus besteht aus nicht einheitlich strukturierten Textdateien (diese erkennen Sie an der Endung .txt). Unstrukturierte Daten müssen wir für die Auswertung vorbereiten, deshalb sollten Sie dafür viel Zeit einplanen. Sie müssen die Daten einlesen, bereinigen und dann für eine weitere, automatisierte Auswertung nutzbar machen.

Die 626 Filmskripte haben einen Umfang von etwa 39 000 Seiten in Word, welche wir für eine beispielhafte Metaanalyse mit Topic Modeling verwenden. Die Metaanalyse gehört laut Franco Moretti (2000; 2013) zu den inhaltsanalytischen Methoden des *distant reading* (auf Deutsch etwa: quantitatives Fernlesen), welches er von *close reading* (auf Deutsch etwa: qualitatives Nahlesen) unterscheidet. Das Fernlesen von Filmen zielt auf die Analyse abstrakter, also Filmen übergeordneter Themen. Diese Themen wiederum ermöglichen uns mithilfe von Topic Modeling die Beantwortung der Forschungsfrage, welches gesellschaftliche Bild von Wissenschaft(ler*innen) im Hollywoodfilm transportiert wird.

Warum ist diese Forschungsfrage soziologisch interessant? Laut dem Autor des Buches *Jurassic Park,* Michael Crichton (1999), existieren zumindest drei Gründe für das Interesse der Filmbranche in Hollywood an Wissenschaft: Erstens bieten Wissenschaftler*innen interessante Lebensgeschichten und deren Lebensweise Material für Handlungsabläufe. Zweitens beinhalten filmische Abbildungen von Wissenschaftler*innen die Möglichkeit, (un-)menschliche Aspekte von Gesellschaft zu thematisieren. Drittens bedeutet Wissenschaft eines der großen Abenteuer der Gegenwart. Weniger romantisch, soziologisch ausgedrückt, ermöglichen Filme die „Gesellschaftsanalyse, die uns direkt zu den gesellschaftlichen Konflikten, Sinnstrukturen und Ideologien führt, die unser Handeln prägen" (Mai und Winter 2006, S. 14). Für die Gesellschaftsanalyse werden in der Literatur Filme als spezifische Ressource von gesellschaftlichem Wissen betrachtet, welche der Gesellschaft einen anthropologischen bzw. anthropozentrischen Spiegel tatsächlicher und vorgestellter sozialer Realitäten vorhält (z. B. Denzin 2003; Morin 2005). Die Diversität gesellschaftlicher Vorstellungen und Abbildungen (von Wissenschaft) kommt dabei auch in Filmen über Wissenschaft und mit Wissenschaftler*innen (als Rollenrepräsentation) vor. Rollen-Stereotype (Schweinitz 2006) spiegeln dabei nicht nur die ambivalente (gesellschaftliche) Betrachtung von Wissenschaft, sondern auch die stereotype Darstellung des weißbärtigen, glatzköpfigen, Brille tragenden Naturwissenschaftlers wider (z. B. Frayling 2005; Tudor 1989; Weingart 2003; Weingart et al. 2006).

11.2 Topic Modeling mit Python

11.2.1 Struktur und Tücken von Filmskripten als Datenmaterial

Bei der Erstellung eines Textkorpus aus Filmskripten ist zu bedenken, dass die Skripte und Filme geistiges Eigentum und damit urheberrechtlich geschützt sind. Filmskripte folgen einem szenischen Aufbau. Das heißt, dass für Dialoge (und Protagonist*innen), Regieanweisungen und Beschreibungen der Szenarien und des Bühnenbildes vorliegen. Dies stellt die Struktur eines Filmskriptes dar. Inhalt sind Handlungen und Dialoge. Betrachten wir nun als Beispiel zur Erklärung von

Struktur und Inhalt von Filmskripten einen Auszug aus dem Film *Indiana Jones and the Temple of Doom* (1984), dessen Geschichte von George Lucas und dessen Drehbuch von Willard Huyck und Gloria Katz geschrieben wurden (Tabelle 11.1). Zu Erklärungszwecken wurde der Ausschnitt mit Abschnittsnummern versehen. Die Abschnittsnummern unterscheiden durch farbliche Markierung Text für die Schauspieler*innen und Drehbuch- bzw. Regieanweisungen (schwarz hinterlegte Absatzzahlen). In Tabelle 11.1 sind zehn von 30 Absätzen Drehbuch- bzw. Regieanweisungen, welche jedoch sehr wertvoll sind, um uns ins Bild zu setzen. Dazu gehören auch Hinweise bei den Filmfiguren, beispielsweise zum Gesichtsausdruck „(disgusted)" in Absatz 5. Zentrale textuelle Merkmale eines Filmskripts sind weiter die Erwähnung der sprechenden Filmfiguren, im Beispiel Dr. Indiana Jones, ein Archäologe, und Willi, die Tänzerin.

Die Szene wurde nicht nur wegen der hervorragenden Eignung als Ankerbeispiel für die Rolle der Wissenschaft in Hollywoodfilmen ausgewählt. Bei genauer Betrachtung des Textes entdecken wir im Text einige Fehler. Diese Fehler können zu Verzerrungen in der Themenzuordnung, falsche Auswahl der Themenzahl, zu Fehlinterpretationen und verlängerten Berechnungszeiten führen. Entsprechend müssen wir umfangreiche Bereinigungsschritte und damit Datenaufbereitungen *(pre-processing)* durchführen, um möglichst viele Fehlerquellen auszuschließen. Beispiele für Fehler aus Tabelle 11.1, die in Summe zu Problemen beim Topic Modeling führen können, lauten:

- Zeile 7: Statt „face" steht im Satz „… her hace lights up …"
- Zeile 9: Ergänzend zu Indiana wird in Anweisungen von „Indy" geschrieben
- Zeile 11: Bindestrich „prin-cess"
- Zeile 21: Das „k" fehlt in „talking" im Satz „You're taling to …"
- Zeile 26: Bindestrich in „get-ting"
- Zeile 29: Im Satz „Don't catch cold" müsste vermutlich nach „catch" ein „a" eingefügt werden.

Das fehlende „a" ist kein Hindernis für die anstehenden Auswertungen. Jedoch haben der Rechtschreibfehler bei „hace" und der Bindestrich in „prin-cess" Folgen. Die Folgen sind, dass beide Nomen bei der automatisierten Erfassung des Textes nicht berücksichtigt werden. Dabei handelt es sich hierbei nur um zwei Nomen, was bei mehreren tausend Seiten Text quantitativ eher nicht ins Gewicht fällt. Selbst bei Einbeziehung machen die durch fehlerhafte Wörter resultierenden Verluste weniger als 1 % der Wörter aus (Tabelle 11.1 enthält 361 Wörter). Im Beispielabschnitt könnten wir die erkannten Fehler korrigieren, das wäre schnell gemacht. Doch alle Fehler per Hand zu korrigieren, wäre bei einem Skript mit 116 Word-Seiten und 33 344 Wörtern sehr aufwendig. Ebenso zeitaufwändig ist es aber, ein Programmskript zu schreiben, welches die Korrekturen der über 600 Filmskripten automatisch durchführt. Fragen Sie sich daher immer, ob der

Tabelle 11.1 Struktur Filmskript am Beispiel der Filmszene *I'm a scientist. I like doing research on certain nocturnal activities*

Absatz	Text und Drehbuch- bzw. Regieanweisungen
1	*Indiana:* I better see how Willie is.
2	-- Short Round shakes his head scornfully as Indy crosses the hall and knocks on another door. After a moment, the door opens and Willie is standing there in a tempting nightgown.
3	*Indiana:* I brought you something.
4	-- He holds up something wrapped in a piece of silk.
5	*Willie:* (disgusted) Not leftovers?
6	*Indiana:* No -- real food.
7	-- Willie opens the bundle suspiciously -- then her hace lights up as she examines the breads and fruits inside.
8	*Willie:* Oh, it is real food … it's beautiful.
9	-- She bites happily into a piece of fruit -- its juice runs down her chin and Indy wipes it off gently with his hand. The mouth deliberately seductive and Willie is not displeased.
10	*Willie:* (Cont'd) You're nice. Listen, I'm taking applications -- how'd you like to be my palace slave?
11	*Indiana:* (smiling) Wearing your jewels to be, prin-cess?
12	-- Indy touches her necklace -- then his hand caresses her neck and ear. She shivers slightly and speaks softly.
13	*Willie:* Yeah -- and nothing else. (smiling) That shock you?
14	*Indiana:* (shaking his head) I'm a scientist. I like doing research on certain „nocturnal activities"
15	-- She smiles and puts a grape to his lips. He opens his mouth takes it and chews it.
16	*Willie:* You mean like love rituals …
17	-- He swallows the grape and they move toward each other slowly to kiss, revealing the passion that's simmering.
18	*Indiana:* And mating customs …
19	-- They kiss again more heatedly.
20	*Willie:* Primitive sexual practices?
21	*Indiana:* You're taling to an authority in that area.
22	-- They kiss again hungrily
23	*Willie:* You're dying to come into my room, aren't you?
24	*Indiana:* You want me so bad, why don't you invite me?
25	*Willie:* Too proud to admit you're crazy about me, Dr. Jones?
26	*Indiana:* I think you're too used to get-ting you own way, Willie …
27	-- They kiss yet again -- and Indy breaks it off, just to show he's still in control. He backs away toward his room.
28	*Willie:* (watching him) We'll see who gives in first -- I'll leave my door open.
29	*Indiana:* Don't catch [a] cold.
30	*Willie:* Dr. Jones -- ?

Quelle: *Indiana Jones and the Temple of Doom* (1984)

Zeitaufwand, der für die Bereinigung notwendig ist, gerechtfertigt ist, oder ob Sie die fehlerbehafteten Wörter aus Ihrem Datenkorpus löschen wollen. Vor allem, wenn wir bedenken, dass unser Korpus mit einer Fehlerwahrscheinlichkeit von < 1 % belastet ist.

11.2.2 Fehlerbereinigung

Beenden wir die unterhaltsame Exkursion in die Archäologie und Historie von Filmen – Sie merken, dass bei Datenbereinigung großes Ablenkungspotenzial besteht –, und wenden uns wieder der wissenschaftlichen, inhaltsanalytisch-quantitativen Erforschung von Wissenschaft im Hollywoodfilm zu. Im vorigen Kapitel 11.2.1 wurden schon die tragbaren Probleme von Fehlern in den Filmskripten angesprochen. Ist grundsätzlich eine etwa einprozentige Fehlerwahrscheinlichkeit für eine induktiv-quantitative Inhaltsanalyse tragbar, so müssen und können manche Fehlerquellen für das Topic Modeling entfernt werden. Das Entfernen von Fehlerquellen erfolgt dabei in allen Phasen des Forschungsprozesses. Initial bei der händischen Sichtung einiger Texte, beim Einlesen und Aufbereiten der Daten, nach den Durchläufen der Topic Models sowie bei der Visualisierung und Interpretation. Nachdem die Fehler entdeckt wurden, insbesondere nach der Durchführung von Topic Models, müssen diese Fehler in der Datenaufbereitungsroutine berücksichtigt und die Modelle neu berechnet werden.

Ergänzend zu in Tabelle 11.1 erkannten Fehlern wie „hace“ (statt *face*) und „taling“ (statt *talking*) und Bindestrichen in Worten (z. B. „prin-cess“ und „get-ting“) werden uns in den *runs* viele unvollständige Worte (z. B. „ror“, „dio“ und „yow“), uneindeutige Geräusche (z. B. „uhhh“) und Symbole (z. B. „==“ und „===“) angezeigt. Diese offensichtlichen Fehler können wir direkt in die Stoppwortliste (im Englischen umgangssprachlich *stop-list*) übernehmen (siehe Box 11.2). Die Stoppwortliste ist für die Arbeit mit dem Programm Python und hier besonders mit dem Paket gensim wichtig, da die Stopwords anzeigen, dass bei der Ausführung des Algorithmus, mit dessen Hilfe die Themen berechnet werden, die angegebenen Worte nicht berücksichtigt werden sollen. Das gilt unabhängig von der Anzahl der Themen, die wir berechnen wollen.

In den ersten Durchläufen kreiert der Topic Modeling-Algorithmus vor allem Namenstopics (z. B. Indiana und Willie). Dies ist zu erwarten, da in den Filmskripten der Sprecher*innenwechsel fortwährend kenntlich gemacht wird (siehe Tabelle 11.1), und sich alles um die Hauptdarsteller*innen dreht. Die Durchläufe sind jedoch nicht umsonst, da wir auf diese Weise die Namen der Protagonisten identifizieren und diese der Liste der Stoppwörter hinzufügen können (siehe Box 11.2). Das geschieht allerdings über viele Durch-

Box 11.2: Beispiele Stoppwortliste
moguy, norther, eame, turnbull, krupp, nomi, aulon, aramis, aismero, malfoy, kalina, emdash, ===, pitts, sciama, coldenrod, thas, walla, ror, uhhh, dio, yow, mcu, quadhole usw.

läufe hinweg, denn wenn die Namen der Hauptcharaktere nicht berücksichtigt werden, dann sammeln sich die nächsthäufigen Namen der Nebendarsteller*innen in den Topics. Die Stoppliste der Namen erstellen wir, da wir die latenten Inhalte der Filmskripte und nicht die Filmcharaktere analysieren wollen.[2]

Wenn Sie diese „manuelle" Form der Namenssuche umgehen wollen, könnten Sie auch ein Webscraping-Skript (siehe Kapitel 8.2.6) programmieren, das z. B. bei der *Internet Movie Database*[3] die Namen von Filmcharakteren einsammelt und daraus eine Stoppliste erstellt. Allerdings ist nicht gesagt, dass diese dann vollständig ist. Zudem bedürfte auch diese einer genauen Kontrolle und kann ebenso viele Probleme erzeugen und ebenso lange dauern wie das manuelle Vorgehen. Daher sollten Sie auch an dieser Stelle abwägen, welches Vorgehen den größeren Aufwand bedeutet.

In dem vorliegenden, sehr besonderen Fall der Filmskripte waren nach mehr als 20 Durchläufen immer noch Namen von Filmfiguren in den Topics zu entdecken, da Filmfiguren mitunter ausgefallene Namen haben. Manche sind durch die gute Kenntnis des Untersuchungsgegenstandes einfach erkennbar, andere jedoch erst bei näherer Betrachtung auffindbar. Beispielsweise konnte

- *bumblebe[e]* als Hummel weder plausibilisiert noch verifiziert werden, und stellte sich nach Blick in die Filmskripte als der *Autobot Bumblebee* (Transformers 2007) heraus.
- Ebenso stellte sich das *top word kingsfield* als Name einer Filmfigur aus *The Paper Chase* (1973) heraus.

Problematisch wurde beispielsweise die Identifikation vieldeutiger Namen. Das bedeutet, dass sowohl Gegenstände, Tiere, Fabelwesen oder mythologische Geschöpfe als auch Charaktere aus den Filmen gleichermaßen gemeint sein können. Besonders auffällig ist dieses Problem in unserem Korpus bei den *X-Men*-Filmen. Hier stellen sich besonders viele Wörter bei genauerer Betrachtung als Namen von Filmfiguren heraus. Beispiele hierfür sind:

- *cyclop[s]* = Einäugige*r (Topic 15; Tabelle 11.2)
- *magneto* = Magnetzünder (Topic 6; Tabelle 11.2)
- *raven* = Rabe (Topic 31; Tabelle 11.2)
- *rogu[e]* = bösartig, gefährlich (Topic 9, 11 und 14; Tabelle 11.2)
- *storm* = Sturm (Topics 2 und 15; Tabelle 11.2)

---

2 Falls Sie analysieren wollen, was einzelne Charaktere aussagen und eventuell zu wem diese etwas sagen, sind andere Verfahren wie zum Beispiel die soziosemantische Netzwerkanalyse geeignet (Basov und Kholodova 2021).

3 IMDB, siehe *www.imdb.com*.

Besonders schwierig ist die Identifikation dieser Charakternamen, da Tiernamen wiederholt als *top words* auftauchen (z. B. *cat, rabbit, raptor, rat, turtle* und *toad;* Tabelle 11.2). Dadurch wird beispielsweise *raven* nicht sofort identifizierbar. Gleiches gilt für den Namen der *X-Men*-Filmfigur *Magneto,* welcher zwar im Übersetzungsprogramm mit Magnetzünder eine sinnige, jedoch unwahrscheinliche Bedeutung erhält. In Ermangelung anderer Fundstellen außerhalb der *X-Men*-Filme kann der Name *Magneto* ebenso wie der Name *Cyclops* auf die Stoppwortliste aufgenommen werden.

Im Gegensatz zu den eindeutigen Fällen *Cyclops* und *Magneto* können die Namen der Filmfiguren *Rogue* und *Storm* nicht ohne erhebliche Verluste an Informationen auf die Stoppwortliste aufgenommen werden. Zwar taucht *Storm* zusammen mit *Cyclops* im Topic 15 auf, jedoch ohne weitere *X-Men* ist *storm* als Sturm ein nachvollziehbares *top word* in Topic 2 in Tabelle 11.2). Dasselbe gilt noch vielmehr für *rogue,* welches in der Bedeutung „bösartig", „gefährlich" sehr gut in die Topics 9, 11 und 14 passt. Insgesamt zeichnet sich ab, dass Wissenschaft sowie Wissenschaftler*innen im Hollywoodfilm häufig in Zusammenhang mit Gewalt, d. h. mit Monstern (z. B. Topic 6, *colossus,* Topic 7, und *beast,* Topic 19) und Militär (z. B. *captain,* Topic 19, und *sargent,* Topic 7) steht. Entsprechend wurden beide Namen der *X-Men*-Filmfiguren nicht der Stoppwortliste hinzugefügt. Diese inhaltsorientierte Entscheidung bedeutet, dass wir bei der Datenauswertung stets überprüfen müssen, ob der Sinn der Topwörter *rogue* und *storm* in einem Topic unabhängig von den Filmfiguren ist. Diese Expert*inneneinschätzung müssen wir als Forscher*innen übernehmen, das kann (gegenwärtig) keine Software durch maschinelles Lernen übernehmen. Auch wenn Ihnen das im Moment als ein Sonderproblem der Filmskripte erscheinen mag, sollten Sie sich merken, dass jeder Korpus seine eigenen spezifischen Probleme der Datenbereinigung haben wird.

Neben der Problematik mit den Namen haben wir in dem Textkorpus der Filmskripte weitere Fehler gefunden, die wir in Tabelle 11.2 durch die Verwendung von eckigen Klammern „[]" kenntlich gemacht haben. Diese potenziellen Fehlerquellen und Ergänzungsnotwendigkeiten sind durch das *Stemming* selbstverschuldet (siehe Kapitel 11.3.5). Ohne das Stemming wäre jedoch die Identifikation von Wörtern bzw. ihres Wortstammes für die gensim Software schwieriger und das zu erwartende Ergebnis wäre schlechter. Folglich müssen wir die Wörter wieder vervollständigen, was beim Stemming teilweise einfach ist durch Ergänzung von:

- abgeschnittenen Buchstaben „e" am Ende der Wörter (z. B. *turtl[e]* und *gon[e]* in Topic 3 sowie *jungl[e]* und *creatur[e]* in Topic 18, Tabelle 11.2);
- abgeschnittenen Endungen wie in Topic 11 bei *termin[ate]* und *presid[ent],* wobei es für das grundlegende Verständnis eines *top words* unerheblich ist, ob Ihre Wahl beispielsweise auf *termin[ate], termin[ator]* oder *termin[ation]* gefallen ist;

- umgewandelte Buchstaben „i“ in „y“, wie bei *mummy* (wobei erst einmal unklar bleibt, ob es sich dabei um die Mumie oder Mutti handelt; Topic 3).
- falsch erkannten Buchstaben „l“ bei *iye,* was als *lye* mit Bedeutung „Beize“, „Lauge“ ein Artefakt aus dem Filmskript *The Shaggy Dog* (2006) ist.

Sie sehen, dass die Fehlerkorrektur und Wortvervollständigungen bei einem heterogenen Textkorpus eine kontinuierliche Arbeit mit den Textdaten bedeutet. Bei der Interpretation der Ergebnisse müssen wir dann berücksichtigen, dass wir einige Topwörter nicht zweifelsfrei bereinigen bzw. erkennen konnten. Zum Beispiel

- ist *revis[ion]* als Überprüfung ein erwartbares Wort im Filmkorpus jedoch auch eine Regieanweisung in Filmskript *Thor* (2011);
- kann *rec[ord]* sowohl Regieanweisung sein als auch eine (Film-)Aufnahme im Film;
- taucht *med[ic(ine)/ium]* mit Medizinbezug und als Regieanweisungen für *medium close* usw. im Filmskript *The Omega Man* (1971) auf;
- wurde *auto[mobile]* als Bedeutung gefunden in Filmskript *The Visitor* (2007) jedoch *autobot* im Filmskript *Transformers* (2007);
- sind die Bedeutungen für *cha[nce/nge]* sowohl als Chance als auch als Wandel beide plausibel und müssen daher pro Topic spezifisch interpretiert werden;
- erscheint *sew[er]* als Abwasserkanal naheliegend, denn bei vorgenommener Suche nach Nomen und Adjektiven sollten/können keine Verben vorkommen (ist gleichbedeutend mit Ausschluss von *to sew* als nähen);
- konnte *semi[-trailer]* für Lastwagenanhänger im Filmskript *Tremors* (1990) gefunden werden.

Haben wir Fehler erkannt, welche nur in einem Filmskript vorkommen, kann sich der Aufwand der manuellen Fehlerkorrektur lohnen, wenn der Textkorpus kleiner ist. Bei dem hier vorliegenden sehr großen Korpus lohnt sich der Aufwand allerdings nicht, deshalb haben wir für die vorliegende Analyse die stark fehlerhaften Filmskripte aus dem Datenkorpus entfernt – es sind immer noch mehr als 600 Filmskripte. Ist es inhaltlich notwendig, wenn das Löschen einzelner Dokumente keine Option ist, so wäre ergänzend zur Stoppwortliste die Lemmatisierung eine Lösung. Bei der Lemmatisierung (siehe Kapitel 11.3.5) handelt es sich um einen automatischen Schritt bei der Datenaufbereitung für das Topic Modeling. Hierbei werden Wörter auf den Wortstamm reduziert. Lemmatisierung kann aber auch eingesetzt werden, um dem Programm mitzuteilen, welche Wörter es als gleiche Wörter zu verstehen hat (siehe Box 11.3) oder um wie beim Beispiel „prin-cess“ (siehe Box 11.3 Lemma 2) weitere mögliche Fehlerquellen einzuberechnen und die Lemmaliste zu ergänzen.

Selbstverständlich könnten wir noch viel mehr Zeit in die Datenberei-

Tabelle 11.2 Beispiele für problematische Wörter bei der Auswertung von Topic Models (Fortsetzung nächste Seite)

Topic	Benennung	Top words									
2		ditch [= Graben, notgewassert]	tunnel	judg[e; Richt-er*in, Sachv-erständige*r]	fusion [Schmelzen, Verbindung]	agent [Agent, Wirkstoff]	storm [Sturm, Filmcharakter]	church	mama	pearl [Perle, Juwel]	air
3		mummy [Mumie, Mutti]	gramp [Opa]	cage	turtl[e]	gon[e]	toot [hupen]	chairman	toad [Kröte, Ekel]	date	guy
6		magneto [Magnetzünder, Filmfigur]	bumblebe[e; Hummel, Filmfigur]	dutch	villag[e/r]	burgomeist[er; Eismöve]	med[ic(ine)/ium?]	monster	gon[e]	scale [Ausmaß, Waage]	agent [Agent, Wirkstoff]
7	Konflikt mit Riese/(kommunistisch.) Diktator	dictat[or]	iye [Iye; Lauge, Beize]	colossu[s; Koloss, Riese]	play	actually [tatsächlich]	blous[e; Bluse, Uniformjacke]	communist	gon[e]	sgt [sargent; Feldwebel]	fingerprint
9	Beruhigung der Liebsten bei nahender Gefahr	rogu[e; bösartig, gefährlich, Filmfigur]	lull [beruhigen/lullaby?]	love	hum [summen]	dog	mansion	foot	grass	raptor [Raptor, Raubvogel, Kampfflugzeug]	Claw[s]
11	feurige Zerstörung/Bedrohung	termin[ate]	rogu[e; bösartig, gefährlich, Filmfigur]	presid[ent]	bot [Roboter/Webcrawler, Kinderpopo]	torch [Fackel, Brandstiftung]	frenzy [Raserei, Wahnsinn]	gun	fire	Agent [Agent, Wirkstoff]	subway
12		kennel [Hundehütte, Zwinger]	dog	rec[ord?]	hallelujah	captain	hoop [Reifen, Korb]	monster	hovercraft	crypt [Gruft, Krypta]	flashback [Rückblende]

Topic	Benennung	Top words									
14		professor	formula	judg[e; Richter*in, Sachverständige*r]	rogu[e; bösartig, gefährlich, Filmcharakter]	pleas [Bitten, Vorwände]	Cure [Heilung, Heiler*in]	book	gon[e]	anybody	music
15		cyclop[es; Einäugige, Filmfigur]	pirat[e]	machin[e]	storm [Sturm, Filmfigur]	policeman	rat	manhattan	paus[e]	floor	ice
18		revis[ion; Überprüfung]	dog	realm [Bereich, Königreich]	child	beast	scientist	jungl[e]	rabbit	creatur[e]	beat [erschöpft, erschlagen]
19	Gemeinsam gegen Bestie/ Untier vorgehen	coal[ition]	beast	ranger	courtesy [Höflichkeit, Aufmerksamkeit]	disc [Datenscheibe, Schallplatte]	sew[er: Abwasserkanal]	hunter	captain	code	plane
26		kingsfield [Filmcharakter]	swamp	tourist	street	wheelchair	raptor [Raptor, Raubvogel, Kampfflugzeug]	bomb	creatur[e]	wasp	chancellor
30		magenta [Farbe violett]	guest	bass [Barsch, Bass]	stallion [Hengst, Stecher]	sonnet	camel	drain [entleeren, austrocknen]	brain	doctor	sea
31		raven [Rabe; Filmfigur]	travel	plane	world	machin[e]	gon[e]	marin[e]	effect	beat	team

Box 11.3 Beispiele für Lemma

Lemma 1	
hace	face
taling	talk
talking	talk
usw.	
Lemma 2	
prin-cess	princess
princess'	princess
prince	prince
prin-ce	prince
usw.	

nigung stecken. Allerdings müssen wir uns dabei immer vor Augen führen, dass Big Data-Auswertungen ein chaotisches Geschäft sind (im Englischen: *messy*) und selbst einen Schlussstrich setzen, also entscheiden, wann die Daten gut genug sind. Obwohl wir uns noch im Stadium der Datenbereinigung befinden, zeichnen sich in Tabelle 11.2 schon gut interpretierbare Topics ab, beispielsweise:

- Topic 9, in dem die Topwörter *lull, love, hum, dog* und *mansion* auf die „Beruhigung der Liebsten bei nahender Gefahr" durch einen durch das *grass* sich nähernden *raptor* mit spitzen *claws* hindeuten.
- Topic 11, welches wir schon recht deutlich als „feurige Zerstörung/Bedrohung" klassifizieren könnten, oder
- Topic 19 als „gemeinsam gegen Bestie/Untier vorgehen".

Die genannten Beispiele zeigen, dass wir uns bei der Datenbereinigung bereits (teilweise) in der Datenanalyse befinden. Daher ist es stets ratsam, die Erkenntnisse in das Forschungstagebuch einzutragen (siehe Box 4.2 und Kapitel 4.3.3). Selbstverständlich können wir in der Regel während der Datenbereinigung jedoch aus der Mehrzahl der Topics keinen Sinn erschließen, wie beispielsweise den Topics 3, 6, 12, 26 und 30. In der Regel bleiben mit zunehmender Themenzahl auch immer mehr Themen übrig, die nicht interpretierbar sind. Sie sollten aber versuchen, die Anzahl nicht interpretierbarer Themen möglichst gering zu halten (zur Not auch durch die Auswahl eines Modells mit geringer Themenzahl und hoher, aber nicht höchster Kohärenz, wie wir in Kapitel 11.4.3 darlegen werden).

11.2.3 Benötigte Pakete

Wenden wir uns nun dem technischen Aspekt des Topic Modelings in Python zu. Um eine LDA in Python durchführen zu können, müssen wir die Pakete `nltk`, `gensim`, `pyLDAvis`, `re`, `os` und `pandas` in unsere Spyder-Sitzung importieren (siehe Kapitel 10.3.1). Darüber hinaus importieren wir die `pyplot`-Bibliothek aus dem `matplotlib`-Modul und `pyLDAvis` zur Erzeugung interaktiver HTML-Dateien. Nachdem die genannten fünf Pakete bereits ausführlich in Kapitel 10.3.1 erklärt wurden, beschränken wir uns hier auf die Vorstellung der neuen Pakete.

Das Paket `nltk` (Loper und Bird 2002) steht für *natural language toolkit* und beinhaltet Befehle, die im Bereich der Computerlinguistik angewandt werden. Es enthält Textkorpora (abrufbar mit `nltk.download()`) mit deren Hilfe Sie zum

Beispiel Stoppwörter entfernen können, oder *part-of-speech tagger,* mit deren Hilfe Sie Wortarten erkennen können. Darüber hinaus beinhaltet das `nltk`-Modul Befehle, um Wortstämme und Lemmata (also Grundformen) von Wörtern zu erkennen oder den Text zu *tokenisieren,* d. h. ihn in einzelne Wörter und Wortketten aufzutrennen. Von Interesse für unseren Anwendungsfall sind `nltk.stem` und `nltk.corpus`. Erstere ist für uns interessant, weil sie Befehle enthält, die uns die Lemmatisierung und das Stemming von Wörtern erlauben. Letztere werden wir verwenden, um Stoppwörter in unsere Sitzung zu importieren und auf die Texte anzuwenden. Zur Erinnerung: Stoppwörter sind Wörter ohne zusätzlichen Sinngehalt bzw. Wörter, die unsere Analyse verzerren würden und deshalb aus dem Korpus ausgeschlossen werden können.

Das Paket `gensim` beinhaltet Bibliotheken und Befehle, die Ihnen die Durchführung und Auswertung von Topic Models in Python erst ermöglichen. Für unseren Bedarf sind die Bibliotheken `gensim.models` und `gensim.corpora` relevant. Erstere Bibliothek beinhaltet die Befehle, mit der Sie die LDA durchführen können. Darüber hinaus beinhaltet sie Algorithmen, mit deren Hilfe Sie die Kohärenz Ihrer Themen berechnen und die Themen zu Wörtern und Texten zuweisen können. Das wiederum hilft Ihnen dabei, das passendste Modell zur weiteren Interpretation auszuwählen. In `pyLDAvis` sind Befehle implementiert, mit deren Hilfe Sie eine interaktive HTML-Datei erzeugen können, um die Themen besser interpretieren zu können.

11.2.4 Pakete und Daten einlesen

Wie bei den Verfahren der Korrespondenzanalyse (siehe Kapitel 9) und Sentiment-Analyse (siehe Kapitel 10) müssen wir zuerst die Pakete, Bibliotheken, Befehle und Daten in unsere Python-Sitzung einlesen. Diese laden wir wieder mittels des `import`-Moduls. Da wir viele Pakete, Bibliotheken und Befehle laden werden, empfehlen wir Ihnen, damit Sie den Überblick in Ihrem Code behalten, Kommentare einzufügen, welche Funktionen die einzelnen Pakete in Ihrem Code einnehmen oder für welche Arbeitsschritte Sie diese Pakete nutzen können. Die einzelnen Befehle werden Sie an gegebener Stelle im Arbeitsprozess kennenlernen. Beginnen wir mit dem Import der uns bereits bekannten Pakte *re, os, pandas* und *numpy.*

Code 11.1 Befehlszeile zum Import von re, os, pandas und numpy

```
import re, os, pandas as pd, numpy as np
```

Nachdem Sie die Zeile mit dem `import`-Befehl angewählt haben, können Sie diese mit F9 ausführen. Mittels dieses Befehls laden wir das `nltk`-Paket und ei-

nige Befehle in unsere Python-Sitzung, die wir für die Datenaufbereitung benötigen. Der erste dieser Befehle lautet `stopwords` und ist in der Bibliothek `nltk.corpus` gespeichert. Wie der Name des Befehls andeutet, beinhaltet der Befehl die gängigen Stopword-Listen in verschiedenen Sprachen (z. B. Deutsch und Englisch). Die `nltk.stem` zugehörigen Befehle `PorterStemmer` und `WordNetLemmatizer` ermöglichen uns, die einzelnen eingelesenen Wörter auf Ihre Wortstämme *(Stemming)* und Grundformen (Lemmatisierung) zu reduzieren. Da wir englischsprachige Filme untersuchen wollen, es aber britisches und US-amerikanisches Englisch mit unterschiedlichen Schreibweisen gibt, ist es empfehlenswert, die Wörter zunächst zu lemmatisieren und dann zu stemmen. Die Codezeilen für den Import der Pakete und Befehle sind, wie in Code 11.2 aufgeführt.

Code 11.2 Befehle zum Import des nltk-Paketes, von Stopwords, dem Stemmer- und Lemmatisierungs-Algorithmus

```
# NLP-Pakete importieren
import nltk
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer
from nltk.stem.wordnet import WordNetLemmatizer
```

Nun können Sie die Dateipfade definieren, in denen einerseits die Filmskripte gespeichert sind, und andererseits Ihre Ergebnisse gespeichert werden sollen. Sie können die Pfade manuell in ihrem Windows-Explorer erstellen oder mittels eines Befehls in Python erzeugen, der im `os`-Paket enthalten ist.[4] Führen Sie die Befehle durch Markieren der Datenzeilen und anschließendem Drücken von F9 aus.

Code 11.3 Festlegung der Dateipfade, in denen die Filmskripte und Stopword-Listen gespeichert sind, sowie des Ordners, in dem die Ergebnisse der LDA gespeichert werden sollen

```
root = "C:[IHR PFAD]\\Filmskripte\\"
path_stopwords = root + "ORDNER_MIT_STOPWORDS\\"
path_filme = root + "ORDNER_MIT_WISSENSCHAFTSFILMEN\\"
output = root + "Output\\"
```

4 Der Befehl lautet `os.mkdirs()`. Wenn sie einen Pfadnamen, zum Beispiel „C:\\Benutzer\\Desktop\\Testordner\\“ angeben, dann erstellt Ihnen Python diesen Dateipfad auf Ihrer Festplatte.

11.2.4.1 Texte mit open() und read() einlesen

Nachdem wir die Dateipfade festgelegt haben, können wir damit beginnen, die Filmskripte nacheinander in unsere Python-Sitzung zu importieren. Jedes Filmskript wurde dabei einzeln als Textdatei auf der Festplatte im Ordner „Filmskripte“ gespeichert. Das heißt, wir haben Dateien mit einer .txt-Endung, die wir Zeile für Zeile einlesen müssen. Das Einlesen von Texten besteht aus zwei Teilen. Erstens muss die Datei geöffnet werden und nach dem Öffnen Ihrer Python-Sitzung übergeben werden. Der erste Schritt lässt sich mittels des `open()`-Befehls erreichen. Dieser lädt die Textdatei in unsere Python-Sitzung. Eine geöffnete Datei ist aber noch keine eingelesene Datei. Genauso wie Sie ein Buch erst aufklappen und den darin enthaltenen Text Zeile für Zeile lesen, um diesen zu verstehen, muss auch Python den Text lesen, was mittels der Funktion `.read()` geschieht. Um einen Text komplett einzulesen, müssen sie folglich nacheinander die `open().read()`-Funktionen eingeben und innerhalb des `open()`-Befehls den Namen der Textdatei angeben, die eingelesen werden soll.

Zusätzlich müssen wir die Spracheinstellung *(encoding)* der Zeichen (Buchstaben, Zahlen, Satzzeichen, Sonderzeichen) als Option angeben, auf die beim Einlesen zurückgegriffen werden soll. Diese *encoding*-Vorgabe können Sie sich wie eine Sprache vorstellen. Wenn Sie versuchen würden, einen Text auf Deutsch zu lesen, der in Englisch oder Japanisch geschrieben ist, dann würden Sie die Bedeutung des Textes nicht verstehen und die Buchstaben und Symbole nicht interpretieren können. Analog müssen sie Ihrem Computer mit dem Einlesen von Schriftzeichen vorgeben, in welchem Zeichensatz ein Text vorliegt. Standardmäßig verwenden heutige Computer die utf-8 Sprachkodierung. Die hierzu passende Option ist `encoding = "utf8"`, mit der Sie Python angeben, dass es die Textdatei mit dieser Zeichenkodierung öffnen soll. Um eine Textdatei in Gänze zu laden, reicht es für gewöhnlich aus, die Codezeile aus Code 11.4 einzugeben, zu markieren und dann mittels F9 auszuführen.

Code 11.4 Einlesen einer Textdatei durch die Verwendung von open().read()

```
open("[NAME DER DATEI].txt", encoding="utf8").read()
```

Es ist jedoch möglich und sehr wahrscheinlich, dass Sie auf Texte mit ganz speziellen Sonderzeichen (z. B. das ß und µ) stoßen, die mit dem oben genannten Code nicht eingelesen werden. Für diese Spezialfälle kann der `open()`-Befehl durch einige Zusatzfunktionen und eine Zusatzoption ergänzt werden. Innerhalb des `open()`-Befehls müssen wir hierfür `encoding = "utf8"` durch `errors = "ignore"` ersetzen. Würde ein Text mittels des utf8-Schriftsatzes eingelesen werden, würde dieser einen Fehler erzeugen. Dadurch, dass wir den Fehler igno-

rieren, eröffnen wir Python die Möglichkeit, einen „Umweg“ zu nehmen, um den Text doch einzulesen und in eine für Python lesbare Schrift zu übersetzen. Dies geschieht über `encode()` und `decode()`. Mittels `encode()` können Sie die Sprachkodierung einer Datei angeben, `decode()` versucht dann, die korrekten Zeichen wiederzugeben. In den beiden letztgenannten Funktionen müssen wir dann jeweils `utf-8` angeben, damit der für uns korrekte Zeichensatz erstellt wird. Sie können sich das wie einen Lateintext aus Ihrer Schulzeit vorstellen. Sie öffnen das Buch mit den Texten (z. B. Caesar oder Tacitus), sehen den Text und verstehen diesen erst einmal gar nicht. Das ist sinnbildlich der `open([TEXT], errors = "ignore")`-Befehl. Nun nehmen Sie ein Wörterbuch zur Hand und beginnen, die Bedeutung des Textes zu entziffern. Das ist mit der `decode()`-Funktion gleichzusetzen. Zuletzt übersetzen Sie den Text und schreiben ihn in ein Heft. Das ist mit der `encode()`-Funktion gemeint. Eine problematische Textdatei können Sie somit mit der in Code 11.5 aufgeführten Codezeile öffnen und einlesen.

Code 11.5 Codebeispiel zum Öffnen und Einlesen von Textdateien mit Sonderzeichen

```
open("[NAME DER DATEI].txt", errors="ignore").read().encode("utf-8").
decode('utf-8')
```

Um beide Vorgänge zum Öffnen und Einlesen von Textdateien miteinander zu verbinden, können Sie eine `try`- und `except`-Abfrage verwenden, wie sie in Code 11.6 aufgeführt ist. Damit teilen Sie Python mit, dass es zunächst versuchen soll, eine oder mehrere Codezeilen auszuführen. Sofern diese Codezeile bzw. Codezeilen einen Fehler produzieren, geben Sie mittels `except` an, dass hier eine Ausnahme hinzugefügt und alle Befehle ausgeführt werden sollen, die durch diese Ausnahme definiert sind. Andernfalls wird die Ausführung Ihres Codes gestoppt. Anders als bei `if-else`-Abfragen, die auch ohne ein else auskommen, muss auf ein `try:` zwangsläufig ein `except:` folgen. Wenn Sie in diesem Kontext nur `except:` angeben, dann sagen Sie Python, dass im Falle jeglicher Fehlermeldung die folgenden Zeilen ausgeführt werden sollen. Sie können nach `except` aber auch Fehlermeldungen spezifizieren, welche vorliegen müssen, damit die Ausnahmeregelung dennoch ausgeführt wird. Seien Sie aber vorsichtig mit Ausnahmeregelungen. Ausnahmeregelungen können sehr schnell dazu führen, dass Ihr Code fehlerhafte Dateien oder Funktionen ausführt, die in der Folge mehr Ausnahmen und Fehler nach sich ziehen können. Versuchen Sie daher, mit so wenig Ausnahmeregelungen wie möglich auszukommen, in unserem Falle: Dass Sie mit möglichst wenig Zeilen Programmcode möglichst viele Einlesefehler von Filmskriptdateien ausschließen können!

Code 11.6 try-except-Abfrage zum Öffnen und Einlesen von Textdateien ohne und mit Sonderzeichen

```
try:
    lines = open([NAME DER DATEI].txt, encoding="utf8").read()
except:
    lines = open([NAME DER DATEI].txt, errors="ignore").read().encode("utf-8").
decode('utf-8')
```

11.2.4.2 Erstellung fortlaufender Texte mit dem sub()-Befehl und der lower()-Funktion

Die Textdateien, die eingelesen werden, sind noch nicht bereinigt und müssen in eine einheitliche, fortlaufende Fließtextform gebracht werden, damit wir den Text bereinigen können. Text bereinigen bedeutet, dass wir in den Dateien die Zeilenumbrüche, Tabulatorsprünge, Groß- und Kleinschreibung entfernen müssen, welche die Analyse erschweren oder verunmöglichen können. Die Textbereinigung werden wir mit dem in Kapitel 10.3.5 bereits beschriebenen `re.sub()`-Befehl mit anschließender `.lower()`-Funktion transformieren, wie in Code 11.07 dargestellt. Der `re.sub()`-Befehl ersetzt eine Zeichenfolge innerhalb eines Textes, und die `.lower()`-Funktion transformiert den gesamten Text in Kleinschreibung.

Beim exemplarischen Sichten ausgewählter Textdateien haben wir festgestellt (Kapitel 11.1.2), dass sich tatsächlich Zeilenumbrüche (`\n`), Tabstops (`\t`), aber auch mehrfache Leerzeichen und Sonderzeichen wie `=======` in den Filmskripten finden. Wie wir bei der Einführung regulärer Ausdrücke (siehe auch Kapitel 10.3.5.1) bereits beschrieben haben, lautet die Zeichenfolge für die Exklusion von Sonderzeichen `[^A-Za-z0-9 ]`. Entsprechend müssen wir Python anzeigen, dass entweder der Zeilenumbruch, Tabstopps, mehrfache Leerzeichen und alle Zeichen, die weder im Alphabet enthalten noch Zahlen sind, aus den jeweiligen Texten entfernt werden sollen. Bitte vergessen Sie nicht, den Platzhalter `[TEXTDATEI]` durch den Namen der Textdatei zu ersetzen, die Sie durch Ausführen der Befehle aus Code 11.4, 11.5 oder 11.6 eingelesen haben.

Code 11.7 Entfernen von Sonderzeichen einer Textdatei und Transformation des Textes in Kleinschreibweise

```
re.sub("(\n|\t|  |[^A-Za-z0-9 ])"," ",[TEXTDATEI]).lower()
```

11.2.4.3 Programmieren eines eigenen Befehls zum Laden der Daten

Um die Filmskripte nacheinander in Ihre Sitzung zu laden, können wir eine `for`-Schleife schreiben, wie im Kapitel 10.3.5.3 ausführlich dargestellt wird. Viel besser ist es allerdings, einen Befehl zu programmieren. Dieses Vorgehen lohnt sich insbesondere dann, wenn Sie häufig die im Befehl enthaltenen Codezeilen innerhalb Ihres Python-Skriptes ausführen müssen. Würden wir keinen Befehl definieren und müssten viele `for`-Schleifen in unser Python-Programmskript einbauen, dann würde unser Programmcode lang und potenziell fehlerbehafteter – und das möchten wir dringend vermeiden. Denn es ist einfacher, drei Zeilen Programmcode zu prüfen, als hunderte von Zeilen auf Fehler zu durchkämmen.[5]

Bevor wir einen Befehl schreiben, muss bestimmt werden, welche Aufgabe dieser Befehl ausführen soll, was dessen Ziel ist, und wie exakt dieses Ziel erreicht werden kann. In unserem Falle ist die Aufgabe des Befehls, Filmskripte einzulesen und, sofern das nicht unmittelbar funktioniert, auf die `encode().decode()`-Funktionen zurückzugreifen. Im Anschluss an die Befehlsausführung soll die eingelesene Textdatei um Sonderzeichen bereinigt werden. Der Befehl für das Python-Programmskript definiert folgende Schritte.

1. Beginne die Funktion und übergebe dem Befehl eine Textdatei.
2. Ist die Textdatei mittels `open("[NAME DER DATEI].txt", encoding="utf8").read()` einlesbar?
   a) Versuche, die Datei einzulesen (mit `try` eingeleitet).
   b) Wenn der Versuch scheitert, dann führe den Ersatzbefehl aus (mit `except` eingeleitet).
3. Bereinige die Textdatei um Sonderzeichen.
4. Übergib den eingelesenen und um Sonderzeichen bereinigten Text zurück an die Python-Konsole.

Der erste Programmierschritt eines Python-Skriptes wird durch den Ausdruck `def [BEFEHL]()` eingeleitet. Hier können Sie den Namen des Befehls angeben und danach in den runden Klammern definieren, welche Objekte (z. B. Filmskripttexte und Jahreszahlen) übergeben und durch den Befehl bearbeitet werden sollen. Wenn Sie mehr als eine Variable, zum Beispiel Texte oder Filmtitel, benötigen, um den Befehl auszuführen, dann müssen Sie die Variablen mit einem

---

5 Sollten Sie sich dazu entschließen, ein eigenes Programmpaket zu schreiben, das diese Befehle für Sie ausführt und auch für zukünftige Projekte geeignet ist, dann müssen Sie diese Befehle und/oder Funktionen, die Sie schreiben, bündeln. Hierzu können Sie beispielsweise Lehrbücher zum objektorientierten, funktionalen oder prozeduralen Programmieren zu Rate ziehen. Im Sinne der Open Source-Bewegung empfehlen wir zudem, dass Sie Ihr Projekt auf GitHub inklusive einer Dokumentation anderen Nutzer*innen zur Verfügung stellen.

Komma voneinander abtrennen. Beachten Sie bitte, dass Sie Platzhalternamen für die innerhalb des Befehls verwendeten Variablen angeben sollten und keine Variablennamen wählen, die bereits im Python-Programmskript verwendet werden. Auf diese Weise stellen Sie sicher, dass der Befehl unabhängig von dem Python-Programmskript funktioniert, in dem die Variablen vorkommen. Der Befehl selbst wird danach mit einem Doppelpunkt eingeführt und hat damit die in Codebeispiel 11.8 aufgeführte Grundstruktur.

Code 11.8 Einleitung der Definition eines eigenen Befehls in Python

```
def [NAME_DES_BEFEHLS](VARIABLE1,VARIABLE2,usw…):
```

Wenn wir auf Enter drücken, um eine neue Codezeile zu beginnen, dann rückt Python unseren Code automatisch ein und zeigt dadurch, dass wir begonnen haben, einen Befehl zu schreiben. Dieser Vorgang in Python ist uns schon im Falle von Schleifen und logischen Abfragen begegnet (siehe Kapitel 10.3.5). Schleifen, logische Abfragen, aber auch Befehlsdefinitionen benötigen diese Einrückung, um zu erkennen, was in der jeweiligen Schleife oder im Befehl ausgeführt werden soll bzw. was genau geschehen soll, wenn eine logische Bedingung erfüllt oder nicht erfüllt ist.

Ein Code-Befehl besteht aus mehreren Bestandteilen. Erstens aus einer Beschreibung, damit andere Nutzer*innen wissen, was wir mit dem Befehl bezwecken, welche Variablen wir an den Befehl übergeben sollen und zuletzt, was der Befehl ausgibt. Diese Beschreibung wird mit jeweils drei Anführungszeichen `"""` eingeleitet und geschlossen. Python gibt Ihnen dabei einige Zeilen vor, die einem Standardbeschreibungsformat folgen und die Sie ausfüllen können. Danach folgt, zweitens, der Code selbst (Schritte 2 und 3), ehe zuletzt mittels `return()` das Ergebnis der im Befehl ablaufenden Vorgänge ausgegeben wird (Schritt 4 im Ablaufschema). Verwechseln Sie dabei nicht `return()` mit `print()`! Während `return()` das Ergebnis einer Funktion an den Speicher ihres PCs übergibt, können Sie mit `print()` einen Text in Ihrer Ipython Console anzeigen lassen. Letzteres ergibt Sinn, wenn Sie einsehen möchten, bei welchem Schritt Ihr Befehl ist, was das Zwischenergebnis ist oder was am Ende des ganzen Vorgangs berechnet wurde.

Grundsätzlich können Sie innerhalb von Befehlen angeben, dass Schleifen oder logische Abfragen wie eine `if-else`-Abfrage verwendet werden sollen. Mit Blick auf unsere Beispieldaten arbeiten wir mit `try` und `except`, da diese Abfragen Fehler (z. B. Fehler beim Einlesen der Dateien) abfangen und umgehen können. Wenn Sie nämlich eine Datei mit Sonderzeichen einlesen wollen, die nicht durch die `read()`-Funktion allein geladen werden kann, wird Ihnen Python einen Fehler ausgeben. Wenn Sie jedoch diese Funktion mit `try:` einleiten

und dann eingerückt in die nächste Zeile einfügen, dann teilen Sie Python mit, dass es zunächst einmal diesen Befehl ausprobieren soll, um Ihre Datei zu laden. Sollte aber festgestellt werden, dass ein Fehler vorliegt, dann benötigt Python eine Alternative, um den Vorgang abzuschließen. Diese Alternative wird mit `except:` eingeleitet, wonach in der Folgezeile eingerückt der Programmcode der Befehle und Funktionen folgt, mit deren Hilfe zuerst die Sprachkodierung angegeben wird, die bewirkt, dass eine Datei gelesen und dekodiert wird. In beiden Fällen sollten Sie die geladene Datei einer Variable übergeben, die wir zuletzt mit dem `return([VARIABLENNAME])`-Befehl wie in Code 11.10 ausgeben.

Darüber hinaus wäre es wünschenswert, dass Sie anderen Nutzer*innen mitteilen, was die Ausführung dieses Befehls bewirkt, welche Variablen oder Objekte als Input dienen und welche Variablen bzw. Objekte ausgegeben werden. Sie können dies realisieren, indem sie drei Anführungszeichen der def-Zeile eingerückt folgen lassen. Damit sagen Sie Python, dass die folgenden Zeilen nur Text beinhalten, der als Hilfetext aufgerufen werden soll (wenn Sie z. B. Ihren Befehl in der Ipython Console schreiben, markieren und „Strg + i“ drücken). Bei Befehlen ergänzt Ihnen Python die drei von Ihnen angegebenen Zeilen wie in Code 11.9 aufgeführt.

Code 11.9 Beispiel für die Eingabemöglichkeit von Hilfetext für selbst geschriebene Befehle

```
    """

    Returns
    -------
    None.

    """
```

Beachten Sie bitte, dass Sie in der Codezeile, in der Sie den Befehlsnamen definiert haben, eine Variable (z. B. `"text"`) definieren, die Sie in Ihrer Funktion ansteuern. Wir kombinieren nun die in den Codebeispielen 11.6 bis 11.9 aufgeführten Programmcodezeilen und enden mit einer `return()`-Ausgabe des fertig eingelesenen Textobjektes. Der gesamte Befehl, der Ihnen nacheinander alle Filmskripte einlesen wird und Sonderzeichen bereinigt, hat somit die in Code 11.10 abgebildete Struktur.

Code 11.10 Beispielhafter Befehl zum Einlesen von Textdateien und Bereinigen um Sonderzeichen

```
def text_laden(text):
    """
    Diese Funktion hilft Ihnen dabei, Filmskripte einzulesen und
    nach dem Einlesen in Sätze aufzutrennen und einen ersten
    Bereinigungsschritt zu machen.

    Parameters
    ----------
    text = Filmskript, das von Ihnen übergeben wird

    Returns
    -------
    Gibt einen bereinigtes und nach Zeilen und Absätzen getrenntes
    Listenobjekt aus, bei dem eine Zeile einen Satz darstellt.
    """

    try:
        lines = open(text, encoding="utf8").read()
    except:
        lines = open(text, errors="ignore").read().encode("utf-8").
        decode('utf-8')

    lines = re.sub("(\n|\t|  |[^A-Za-z0-9 ])"," ",lines).lower()

    return(lines)
```

11.2.4.4 Generierung eines pandas-DataFrames mittels for-Schleife und dem definierten Befehl

Um nun die Filmskripte zu laden und einzulesen, können Sie zunächst mit dem `os.chdir()`-Befehl den Dateipfad auswählen, in dem diese Filmskripte gespeichert sind. Danach lassen Sie Python durch die Verwendung eines Listenabgleichs und `os.listdir()` (= erkennt alle Dateinamen in einem zuvor angegebenen Ordner) eine Liste übergeben. Im Anschluss lesen Sie die Daten mithilfe unseres Befehls in Code 11.10 ein und kombinieren die Textdaten mit weiteren Informationen wie dem Filmtitel, Genre und Erscheinungsjahr, was im Folgenden erklärt wird.

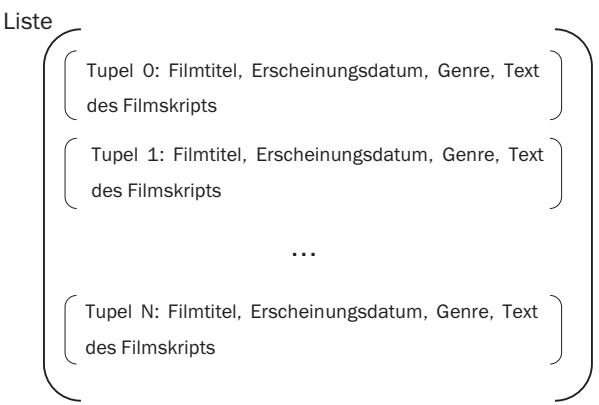
Die Filmskripte einzulesen und in ein einheitliches Textformat zu bringen, ist bereits ein erster wichtiger, aber nicht der letzte Schritt zur automatisierten Analyse der Texte. Wollen wir aber Analysen durchführen, die es uns erlauben, verschiedene Zeiträume oder Genres miteinander zu vergleichen, dann benötigen wir einen systematisierten Datensatz, der nicht nur die Texte, sondern auch das Veröffentlichungsjahr, Titel und Genre beinhaltet. Diese Daten können beispielsweise in einer separaten Excel-Tabellendatei gespeichert werden. Folglich müssen wir dafür Sorge tragen, dass die Texte und die sogenannten Metadaten verknüpft werden können. Hierfür verwenden wir ergänzend zum jeweiligen Filmnamen eine Identifikationsnummer für jedes Filmskript, die ebenfalls in der Tabellendatei enthalten ist, in der die Metadaten gespeichert sind. Der Trick bzw. die Vorarbeit ist, dass bei der Datensammlung die Dateien der Filmskripte so gespeichert wurden, dass nach einheitlichem Muster mit Unterstrichen eine Identifikationsnummer im Dateinamen vorkommt (z. B. wf0816_ThefifthElement_1997.txt).[6] Die Buchstaben- und Zahlenfolge der Identifikationsnummer können wir mittels eines regulären Ausdrucks (regex) extrahieren, der alle Zeichen nach dem ersten Unterstrich entfernt. Wie in Codebeispiel 11.11 weiter unten definiert, wollen wir für die Auswertung zudem den Namen des Filmes erhalten. Hierfür müssen wir einen Befehl formulieren, der alle Zeichen bis zum ersten Unterstrich und ab dem zweiten Unterstrich entfernt.

Damit wir die Texte der Filmskripte, die im Variablennamen enthaltene Identifikationsnummer und den Titel mit den Informationen zu Genre und Jahr der Veröffentlichung in Bezug setzen können, müssen wir nun Filmskript, Identifikationsnummer und Filmtitel in ein *pandas*-DataFrame-Objekt umwandeln. Hierzu gehen wir mithilfe einer Schleife Textdatei nach Textdatei durch und fügen Filmskript, Identifikationsnummer und Titel als Tupel einer Liste hinzu. Ein Tupel ist analog zu einer Zeile in einer Tabelle bzw. in einem geordneten Datensatz zu verstehen. Eine Liste bestehend aus Tupeln zeigt *pandas* an, dass es sich um ein potenzielles DataFrame-Objekt handelt. Dieses Tupel können wir mit `append.(FUNKTIONSNAME[VARIABLE])` der Liste übergeben, die als Grundlage für unseren *pandas*-DataFrame fungiert. Abbildung 11.2 verdeutlicht den Aufbau dieser ersten Vorstrukturierung, die nötig ist, um ein *pandas*-DataFrame erstellen zu können.

Wenn die Schleife (siehe Code 11.11) erfolgreich durchgelaufen ist, dann hat unsere Liste mit den Filmskriptinformationen den gleichen Aufbau wie in Abbildung 11.2 dargestellt. Dazu müssen wir die Tupel-Liste durch den Aufruf von `pd.DataFrame([DATEN])` in einen Datensatz transformieren. Abgeschlossen

6 An dieser Stelle möchten wir nochmals anmerken, dass es leichter ist, strukturierte oder semistrukturierte Daten einzulesen und aufzubereiten. Deswegen empfehlen wir an dieser Stelle, Ihre Daten im Vorfeld (sofern möglich) bereits in eine einfach einzulesende Form zu bringen.

Abbildung 11.2 Aufbau eines pandas-DataFrames bestehend aus einer Liste mit Tupeln



wird der Vorgang durch Eingabe von „=“, wodurch der Datensatz als neue Variable dem Python-Programmskript übergeben wird.

Wie Sie in Abbildung 11.2 erkennen, haben wir zwar mit der Liste ein Objekt erzeugt, das die Daten der einzelnen Filme beinhaltet und sortiert hat. Für die Erstellung eines DataFrame müssen wir jedoch noch fortlaufende Zeilennummern und Variablen bzw. Spaltennamen setzen. Dazu müssen wir in den Optionen des `pd.DataFrame()`-Befehls in der Option `columns=[]` angeben, wie die einzelnen Spalten des Datensatzes heißen sollen. Für die eindeutige Namensgebung der Spalten verwenden wir „idno“ für die Identifikationsnummer (weil dies der gleiche Variablenname wie im Excel-File mit den Metadaten ist), „Titel“ für den Filmtitel und „Text“ für den Text des Filmskripts. Das können Sie machen, indem Sie eine Liste mit den Spaltennamen der Option `columns` übergeben.

Hiernach müssen wir den Datensatz mit den Genre- und Veröffentlichungsinformationen mittels des `pd.read_csv()`-Befehls laden und zuletzt mit dem Befehl `pd.merge()` zusammenführen. Der Befehl `pd.merge()` ist so aufgebaut, dass die Datensätze ausgewählt werden, die zusammengeführt werden sollen. Danach können wir bei Optionen definieren, ob beispielsweise mit `how="left"`[7]

7 Andere Möglichkeiten wären `how="right"`, `"inner"`, `"outer"` oder `"cross"`. Bei `"right"` wird der 2. Datensatz als Ausgangspunkt für die Zusammenführung verwendet. Bei `"outer"` führt der `merge()`-Befehl nur die Datenreihen zusammen, die in beiden Datensätzen vorkommen. Die `"inner"-join`-Option ist die Standardeinstellung und führt dazu, dass nur die Schnittmenge der in beiden Datensätzen enthaltenen Daten zusammengeführt werden. Wenn zum Beispiel im ersten Datensatz Werte für *Indiana Jones, Star Wars* und *Krieg der Welten* enthalten sind, im zweiten aber nur für *Indiana Jones,* dann wird nur die Datenzeile mit *Indiana Jones* und den entsprechenden Werten zusammengeführt, die bei den Variablen aufgelistet sind.

der zweite Datensatz an den ersten angefügt werden soll, und mit dem Befehl `on="[VARIABLENNAME]"` definieren, über welchen Spaltennamen Sie die Dateien zusammenführen wollen. Sie können auch separate Spaltennamen für den ersten (`left_on`) und rechten Datensatz (`right_on`) festlegen. Hierfür ist jedoch erforderlich, dass gleiche Werte in beiden Spaltennamen vorliegen, um die Datensätze zusammenzuführen (z. B. haben wir den Filmtitel „Indiana Jones" im Vorfeld den Variablennamen `title` im ersten und `Titel` im zweiten Datensatz gegeben, das müssen wir nun beachten).

In Code 11.11 sind alle in diesem Kapitel 11.2.4.4 beschrieben Programmcodes zusammengefasst, welche Sie zum Einlesen, bereinigen und zusammenführen der Filmskripte und Metadaten benötigen.[8] An dieser Stelle möchten wir Sie noch darauf hinweisen, dass in Python importierte Texte sehr viel Speicherplatz einnehmen können. Daher ist es erforderlich, dass Sie regelmäßig Ihren Arbeitsspeicher leeren.

Code 11.11 Beispielcode zum Einlesen aller Wissenschaftsfilm-Textdateien und Metadaten durch die Verwendung einer Schleife und des selbst geschriebenen Befehls (Fortsetzung nächste Seite)

```
# ============================================================================
# Einlesen der Wissenschaftsfilme
# ============================================================================
os.chdir(path_filme)

files = [file for file in os.listdir()]

data = []
for f in files:
    data.append((re.sub("_.+_.+","",f),
                 re.sub("(wf[0-9]+_|_[0-9]{4}.txt)","",f),
                 text_laden(f)))

data = pd.DataFrame(data, columns= ["idno","Titel","Text"])
# ============================================================================
# Einlesen der Metadaten
# ============================================================================
os.chdir(path_stopwords_and_metadata)
```

8 Man könnte theoretisch die for-Schleifen und die Erstellung der Texte in einzelne Datensätze ebenfalls in die Funktionsdefinition mitaufnehmen.

```
metadata = pd.read_csv("metadata_new.csv", sep=";")
del(metadata["Unnamed: 0"])

df = pd.merge(data,metadata, on="idno")

del(data,metadata)
```

11.3 Aufbereitung der Daten für die Analyse

Nachdem wir die Textdateien nun erfolgreich eingelesen haben, können wir uns dem nächsten Schritt zuwenden: Der Aufbereitung der Textdateien. Wir müssen die Texte aufbereiten, damit wir bei unserer eigentlichen Analyse im Topic Model kein Gestammel (z. B. uaargh, tis), Wortartefakte (hes oder 's anstatt he's) oder Wörter ohne tiefere Bedeutung (this, is) in unseren Themen vorfinden. Von der Beispielanalyse ausschließen möchten wir auch Lautmalereien und Füllwörter, welche nur bei einer qualitativen Tiefenanalyse der Texte eine Bedeutung haben können und damit analysiert werden sollten. Bei über sechshundert Filmskripten, die ihrerseits hunderte Seiten lang sind, ist eine Tiefenanalyse jedoch sehr aufwändig. Da eine qualitative Analyse aller Filme sehr bzw. zu lange dauern würde, müssten Sie für eine vertiefende Analyse eine gut begründete Auswahl an Filmen treffen, wie in Kapitel 2.2.3 beschrieben wurde.

In den folgenden Abschnitten zeigen wir Ihnen, wie wir unsere Filmskripte in mehreren Schritten aufbereiten.

1. Text tokenisieren,
2. Stopwords festlegen,
3. Stopwords entfernen,
4. Auswahl der Worte systematisieren auf bestimmte Wortarten einschränken,
5. englische Schreibweisen (American and British English) zusammenführen,
6. reduzieren der Wörter mittels der Techniken der Lemmatisierung und des Stemmings,
7. Filmskripte für Analyse in kleinere Teilabschnitte aufteilen,
8. Wort-Id-Lexikon erstellen,
9. Korpus in ein *bag of words*-Format umwandeln,
10. Gewichtung von einzelnen Worten in Abhängigkeit von Länge der Texte und Gesamtkorpus (tfidf).

11.3.1 Text tokenisieren

Beginnen wir nun mit der Tokenisierung der Texte. Tokenisierung bedeutet, dass wir den Textstring (= Zeichenfolge) in einzelne Wörter, Token (= Merkmale) genannt, auftrennen. Das Ergebnis dieses Vorgangs ist die Erstellung einer Liste, in der alle Wörter, aber auch Satz- und Sonderzeichen eines Textes enthalten sind. Für das Tokenisieren verwenden wir aus dem nltk-Paket den `word_tokenize()`-Befehl. Diesen Befehl können Sie über die Bibliothek `tokenize` aufrufen. Die in Code 11.12 dargestellte Befehlszeile erfasst den Text, den Sie in seine Zeichenbestandteile, d. h. in Einzelwörter in der Liste, zerlegen möchten.

Box 11.4: Weitere Materialien und Informationen online
Wir stellen Ihnen eine volle Liste der genutzen Fremdwörter wie „Textstring" oder „Token" mit Erklärungen auf dem Blog *https://sozmethode.hypotheses.org/category/topic-modeling* zur Verfügung.

Code 11.12 Beispielcode für die Tokenisierung eines Textes in einzelne Wörter

```
nltk.tokenize.word_tokenize(text)
```

Unser Datensatz mit Namen df (= *dataframe*) enthält mehrere hundert Texte, welche Sie als ein Listenobjekt erstellen müssen (z. B. `tokenisierte_texte`). Hierfür müssen Sie die einzelnen Filmskripte mithilfe einer `for`-Schleife ansteuern, und diese der zuvor erstellten Liste mittels der `.append()`-Funktion zuordnen. Nachdem diese Schleife durchgelaufen ist, können Sie dem Datensatz das neue Listenobjekt als Variable übergeben (Code 11.13).

Code 11.13 Beispielcode für die Tokenisierung eines Textes in einzelne Wörter

```
tokenisierte_texte = []

for text in df.Texte:
   tokenisierte_texte.append(nltk.tokenize.word_tokenize(text))

df["texte_tokenisiert"] = tokenisierte_texte
```

11.3.2 Festlegen der Stopwords

Im zweiten Schritt der Datenaufbereitung entfernen wir die Stopwords (= Stoppwörter). Stopwords dienen der Sortierung nach für unsere Untersuchung rele-

vanten und nicht relevanten Wörtern. Nicht relevant bedeutet, dass die Wörter nicht für die Beantwortung unserer Forschungsfrage hilfreich sind. Beispielsweise müssen wir zuerst zwischen Wörtern wie „ist“, „und“ „das“ usw. und Eigennamen wie „Magneto“, „Indiana Jones“ sowie Ausrufen und Lautmalereien wie „aaaah“ unterscheiden. Diese Wörter sind nicht geeignet, um Themen voneinander zu unterscheiden – es sei denn, Sie möchten Themen wie „Lautmalerei“ oder „Protagonisten aus *Indiana Jones*“ oder „Protagonisten aus der *X-Men*-Reihe“ als Thema in Ihren Daten für eine vertiefte Untersuchung entdecken. Für den erstgenannten Fall bietet die `nltk.corpus`-Bibliothek eine große Auswahl an Stopwords an. Die Bibliothek enthält Wörter und Wortendungen, die in der jeweiligen Sprache am gebräuchlichsten sind oder keine zusätzlichen Informationen für die Interpretation von Themen enthalten. Die Bibliothek `nltk.corpus` enthält den Befehl `stopwords()` importieren. Wieder müssen wir Python durch den Befehl `stopwords.words('[SPRACHE]')` angeben, welche Sprache die Stopword-Liste hat, und mit „=“ einer Variable zuordnen. Da wir englischsprachige Filmskripte analysieren, ersetzen wir `[SPRACHE]` durch `english` und übergeben die Stopword-Liste an eine neue Variable, die wir hier `stopw` nennen (Code 11.14). Wenn wir beispielsweise die Stopwords einlesen und die ersten zehn Stopwords aus dieser Liste mit `stopw[:10]` aufrufen, dann erhalten wir Output 11.1.

Code 11.14 Einlesen englischsprachiger Stopwords aus dem nltk-Paket

```
## Einlesen der Stopwords aus dem nltk-Paket
stopw = stopwords.words('english')
```

Output 11.1 Beispielhafte Ausgabe der ersten zehn Stopwords, die im Stopword-Korpus des nltk-Paketes enthalten sind

```
In[4]: stopw[:10]
Out[4]: ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you',
"you're"]
```

Einige weitere Stopwords, zum Beispiel Charakternamen, sind zwecks Übersicht in einer eigenen Stopword-Liste enthalten (siehe Kapitel 11.2.4.4). Diese Stopword-Liste ist eine Textdatei, welche wir unter dem Namen `stopwords_en_project.txt` gespeichert haben. Die Datei `stopwords_en_project.txt` muss wie die Filmskript-Dateien im Python-Programmskript geöffnet werden (Code 11.15).

Die zusätzliche Textdatei der Stopwords öffnen wir mit der uns bekannten `open().read()`-Kombination. Da Stopwords nach Zeilen getrennt sind, muss

mit dem `re.split()`-Befehl im Python-Programmskript vorgegeben werden, den Zeilenumbruch (`\n`) von Zeichenfolgen (= Strings) aufzutrennen. Die Auftrennung eines Textstrings erzeugt eine Liste bestehend aus Zeichen und Wörtern, welche in einer Zeile der Stopword-Listen enthalten sind. Die neue Liste erzeugt ein String-Objekt, welches wir jeweils einzelnen als n neue Stopwords in einer Schleife an die Liste mit den in Code 11.14 eingelesenen Stopwords anfügen. Die hierfür notwendigerweise zu schreibende for-Schleife muss die `.append([NEUES STOPWORD])`-Funktion ausführen, wobei `[NEUES STOPWORD]` durch einen von Ihnen gewählten Begriff ersetzt werden muss.

Nach der gleichen Logik können wir Wörter anfügen, die sich während der Durchgänge der LDA als nicht relevant, d. h. wenig trennscharf oder als zusätzliche Stopwords wie z. B. Eigennamen, definiert haben. Bei der Aufarbeitung großer Datenmengen müssen Sie davon ausgehen, dass Überschneidungen zwischen den Stopword-Listen bestehen können. Entsprechend müssen wir, um den Befehl abzuschließen, alle Doppelungen aus unserer Stopword-Liste entfernen. Doppelungen werden mittels des `unique()`-Befehls aus der `numpy`-Bibliothek entfernt. Im `unique()`-Befehl fügen wir den Namen unserer Stopword-Liste in die Klammern ein. Insgesamt erhalten wir eine Programmsyntax, wie sie in Code 11.15 aufgeführt ist.

Code 11.15 Beispielcode zum Einlesen einer im Vorfeld erstellten Stopword-Liste im txt-Format

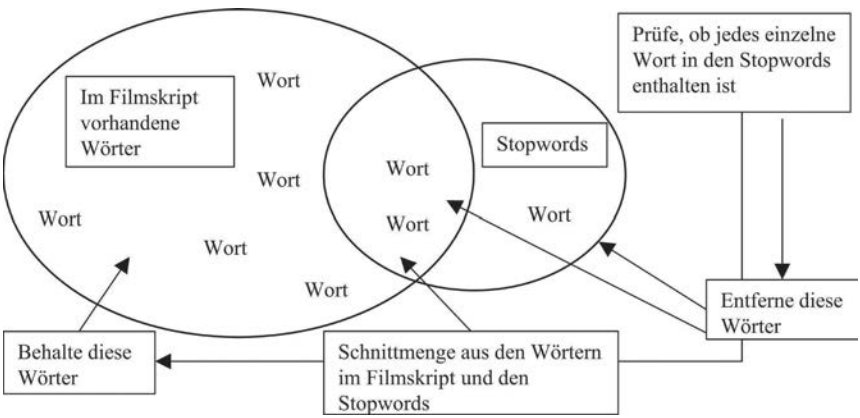
```
os.chdir(path_stopwords)
film_stopwords = re.split("\n",open("stopwords_en_project.txt").read())

for word in film_stopwords:
    stopw.append(word)

additional_stopwords = [] # Hier Stopwords nach den Durchläufen der LDA
hinzufügen.
for a in additional_stopwords:
    stopw.append(a)

stopwoerter = np.unique(stopw)
del(stopw, film_stopwords)
```

11.3.3 Entfernen von Stopwords und Wortfragmenten

Im Datenaufbereitungsvorgang haben wir bisher die Stopword-Liste, die spezifisch auf unseren Untersuchungsgegenstand Filmskripte angepasst wurde, aus der `nltk.corpus`-Bibliothek geladen und durch weitere Stopwords ergänzt. Zum Entfernen dieser Wörter aus dem Textkorpus, welche als Tokens in den Texten erfasst (Kapitel 11.3.1) und in Stopword-Listen definiert wurden (Kapitel 11.3.2), müssen wir einen Listenabgleich durchführen. Mit dem Listenabgleich prüfen wir, ob die in einem Filmskript enthaltenen Wörter Teil unserer Stopword-Liste sind, und aus jedem Filmskript entfernt werden müssen.

Theoretisch können wir mit einem Listenabgleich jedes Element einer Liste aufrufen. Der Aufruf von getrennten Elementen erfolgt über den Befehl `[x for x in LISTE]`, wobei das `x` für jedes Listenelement – hier ein Wort als String – steht. Der `x for x`-Befehl definiert für Python, jedes Wort exakt so auszugeben, wie es im Objekt mit dem Namen `LISTE` vorliegt. Angenommen, wir führen den Listenabgleich durch, um nur bestimmte Elemente aus der anderen Liste zu behalten, dann müssen wir am Ende unseres Listenabgleiches die Bedingung `if x in [ZWEITER_LISTE]` schreiben. Das Schema in Abbildung 11.3 verdeutlicht die Logik, die dem Listenabgleich zugrunde liegt.

Abbildung 11.3 Schematische Darstellung der Logik eines Listenabgleichs zwischen den in einem Filmskript enthaltenen Wörtern und der Stopword-Liste



Methodisch erfordert unser Listenabgleich zwischen tokenisierten Texten und Stopwords, dass die Elemente, die in der Liste verbleiben, nicht in der zweiten Liste mit den Stopwords stehen. Dazu definieren wir einen Listenabgleich, in dem die Abfrage `if not x in stopwoerter` enthalten ist. Ergänzen wir den

Befehl um das gleichzeitige Entfernen von Wortfragmenten und Sonderzeichen, so verwenden wir den regulären Ausdruck `[^A-Za-z0-9 ]` aus dem `re`-Paket (Kapitel 10.3.5.1). Um beide Befehle zu kombinieren, verwenden wir den Befehl `re.sub()`, sodass beim Listenabgleich nur die Worte beibehalten werden, die nicht in der Stopword-Liste sind (Code 11.16). Sie möchten diese Datenaufbereitung jedoch nicht nur für einen Text durchführen, sondern benötigen erneut eine for-Schleife, damit die Stopwords aus allen Texten entfernt werden? Bitte vergessen Sie nicht, den Text entweder einem neuen Objekt durch „=“ zu übergeben oder es an eine neue Liste mittels der `append()`-Funktion anzufügen.

Code 11.16 Entfernen der Stopwords und Sonderzeichen aus einem Filmtext mittels Listenabgleich

```
[re.sub("[^A-Za-z0-9]","", word) for word in text if word not in stopwoerter]
```

Aufgrund der systematischen, jedoch schematischen Vorgehensweise beim maschinellen Lernen müssen Sie davon ausgehen, dass Wortfragmente übrigbleiben oder Wörter falsch getrennt wurden. So könnte es sein, dass aus „it’s“ ein „it“ und „s“ wurde, und beide als Wörter als Listenelement nach der bisherigen Datenbereinigung existieren. Potenziell problematisch ist ebenso, dass ganze Tokens entfernt wurden, was Sie an der Leerstelle in der Liste erkennen. Leerstellen haben standardisiert in einer Zeile die „Länge“ von 0, wohingegen Wortfragmente eine „Länge“ von 1 haben. Folglich können wir die Liste weiter bereinigen, indem wir Tokens mit einer Länge von kleiner 2 aussortieren. Grundsätzlich können Sie die Länge eines Wortes oder eines anderen Objektes in Python mit dem Befehl `len()` ermitteln. Wenn Sie beispielsweise die Länge des Wortes „Liste“ mit dem Befehl `len("Liste")` abfragen, dann erhalten Sie als Ausgabe in der Ipython Console die Zahl `5`.

Für den Listenabgleich schreiben wir folglich eine logische Abfrage, welche prüft, ob ein Wort länger als ein oder nicht kürzer als zwei Zeichen ist. Code 11.17 stellt die erste, Code 11.18 die zweite der Varianten der logischen Abfrage dar, es bleibt Ihnen überlassen, welche Variante des Listenabgleichs Sie wählen. Zum Übertragen der Listenabgleiche auf alle Texte müssen Sie auch hier wieder eine `for`-Schleife definieren (siehe Code 11.23 weiter unten), da die tokenisierten Texte als Liste innerhalb einer Variablen vorliegen, und ein Listenabgleich nur mit je einer Liste (= tokenisiertem Text) und der Stopword-Liste durchgeführt werden kann.

Code 11.17 Variante 1 zur Aussortierung von Wörtern aus einem Filmskript, die maximal einen Buchstaben haben

```
[word for word in text if len(word) > 1]
```

Code 11.18 Variante 2 zur Aussortierung von Wörtern aus einem Filmskript, die maximal einen Buchstaben haben

```
[word for word in text if not len(word) < 2]
```

11.3.4 Beschränkung der Wörter auf Nomen, Verben und Adjektive

Nachdem wir eben gelernt haben, mithilfe der Listenabgleiche sowohl Stopwords als auch Wortfetzen aus unseren Texten zu entfernen, müssen im nächsten Schritt der Textaufarbeitung für die Inhaltsanalyse bestimmte Wortarten aus den jeweiligen Texten entfernt werden. Zur Beantwortung unserer Forschungsfrage benötigen wir nur die Wortarten Nomen, Verben und Adjektive. Die Reduktion von Texten auf bestimmte Wortarten ermöglicht es, die Themenkohärenz zu erhöhen (Martin und Johnson 2015). Bei einem (sehr) großen Textkorpus erhöht die Reduktion von Texten die Wahrscheinlichkeit klar umrissene Topics zu erhalten. Ohne Themenkohärenz ist es sehr wahrscheinlich, dass sich einzelne oder eine Vielzahl von Topics überlappen können und einen Großteil Ihrer Ergebnisse des Topic Models nicht interpretierbar machen.[9]

Technisch wird die Reduktion mit dem `pos_tag()`-Befehl aus dem nlkt-Paket durchgeführt. Die Abkürzung `pos` steht für „part of speech", also Wortart, und tag ist die Abkürzung für „tagging", d. h. markieren oder etikettieren. Zur Verdeutlichung des Vorganges wird in Code 11.19 diesem Befehl eine Liste mit Tokens übergeben. Wenn Sie den Befehl durch markieren und F9 drücken ausführen, dann erhalten Sie die in Output 11.2 dargestellte Liste mit Tupeln als Ausgabe, die sowohl die jeweiligen Wörter/Token als auch eine Etikettierung der Wortart enthält.

Die Etikettierung der Ausgabewerte wird durch DT (= Englisch: „determiner" für Bestimmungswort), VBZ (= Verb in der dritten Person singulär im Präsens),

9 Das lässt sich häufig aber erst nach den ersten Durchgängen der LDA feststellen. In unserem Fall ergaben Themen, die nicht auf Nomen, Verben und Adjektive beschränkt wurden, häufig keinen Sinn. Auch wenn wir die Abschnitte der Filmskripte im Nachhinein gelesen haben, die stark mit den jeweiligen Themen assoziiert waren, so ergab sich dennoch in vielen Fällen, dass diese Themen nahezu uninterpretierbar waren.

NN (= Nomen im Singular) und der Punkt am Ende deutet auf ein Satzzeichen hin. Grundsätzlich ist anzumerken, dass der `pos_tag()`-Befehl am besten für englische Texte funktioniert und in anderen Sprachen wie Deutsch oft fehlerhafte Wort-Zuweisungen beim Tagging trifft.

Code 11.19 Funktionsweise des pos_tag()-Befehls aus dem nlkt-Paket

```
nltk.pos_tag(["this","is","a","test","!"])
```

Output 11.2 Ausgabe des in nltk enthaltenen pos_tag()-Befehls

```
Out[0]: [('this', 'DT'), ('is', 'VBZ'), ('a', 'DT'), ('test', 'NN'), ('!',
'.')]
```

Mit diesem grundsätzlichen Verständnis des *part of speech-tagging* wenden wir uns nun dem großen Textkorpus zu. Um Wörter in einem Text auszuwählen, definieren wir zunächst die Wortarten, die wir für die weiteren Analysen in unserem Datensatz beibehalten wollen. Diese überführen wir in eine Liste, die wir mit `pos_types` benannt haben. In diese Liste überführen wir alle *part of speech-tags,* die Nomen (beginnend mit `NN`), Adjektive (beginnend mit `JJ`) und Verben (beginnend mit `V`) umfassen. Der Code 11.20 definiert, wie dieser Listenabgleich erfolgen soll.

Code 11.20 Entfernen von Wortarten durch einen Listenabgleich zwischen einem Filmskript und einer Liste, in der die Wortarten aufgeführt sind

```
pos_types = ["NN", "NNS", "NNP", "NNPS",
             "JJ", "JJR", "JJS",
             "VB", "VBD", "VBG", "VBN", "VBP", "VBZ"]

[word[0] for word in nltk.pos_tag(text) if word[1] in pos_types]
```

Am Beispiel des Listenabgleichs in Code 11.20 möchten wir in Erinnerung rufen, welches Schema mit `word[0]` im vorderen Teil und `word[1]` im hinteren Teil des Abgleiches in Python verwendet wird. Da `word` die Tupel (= Zeile in einem geordneten Datensatz) bestehend aus Wort und Wortart ansteuert, die durch den `pos_tag()`-Befehl erzeugt wurde, müssen wir in unserem Listenabgleich festlegen, welchen Teil der Tupel wir behalten wollen, und welchen Teil der Tupel wir mit den Wortarten abgleichen wollen, die in `pos_types` enthalten sind. Be-

trachten wir die in Output 11.2 dargestellten Tupeln, dann sehen Sie, dass das Wort (z. B. „is“) die erste Stelle der Tupel und die Wortart (z. B. „VBZ“) an der zweiten Stelle platziert ist. Wenn Sie das erste Element einer Tupel, Liste etc. ansteuern möchten, dann müssen sie `[0]` hinter die entsprechende Tupel, Liste oder andere Datentypen angeben. Erinnern Sie sich bitte daran, dass Python bei 0 zu zählen beginnt, das erste Element eines Objektes somit auch mit 0 indiziert ist. Das zweite Element, in unserem Falle die Wortart, ist an Stelle `[1]`. Somit prüfen wir, ob das zweite Element der Tupel, also die zugewiesene Wortart, in unserer Liste der Wortarten enthalten ist. Wenn die Wortart vorhanden ist, dann wird das an Stelle `[0]` befindliche Wort beibehalten und einer Liste zugefügt. Wenn die Wortart nicht in unserer Liste ist, dann wird das Wort auch nicht in die Liste aufgenommen. Auf diese Weise können Sie bei je einem Filmskript die Worte auf Nomen, Adjektive und Verben reduzieren.

11.3.5 Lemmatisierung und Stemming der Filmskripte

Nachdem wir den Text der Filmskripte auf Tokens reduziert haben, die Nomen, Adjektive oder Verben sind, müssen wir diese nun lemmatisieren und stemmen (Schritt 6 der Datenaufbereitung). Zur Erinnerung: Wir lemmatisieren die Tokens, d. h. wir reduzieren Worte als Zeichenketten auf ihre Grundform, um unterschiedliche Schreibweisen ein und desselben Wortes im amerikanischen und britischen Englisch zusammenzuführen. Wir stemmen anschließend die Tokens, um die Worte auf den Wortstamm zu reduzieren und unterschiedliche Tempora, Einzahl und Mehrzahl zusammenzufassen. Dadurch verringern wir die Anzahl unterschiedlicher Tokens weiter und stellen sicher, dass Tokens mit dem exakt gleichen Inhalt (z. B. rennen, rannte, gerannt) nicht mehrfach vorkommen und die Trennschärfe zwischen Themen verringern (Stichwort: Themenkohärenz).

Um ein Filmskript zu lemmatisieren, ist die Zuweisung des Lemmatisierungs-Algorithmus zu einer Variable und ein Listenabgleich nötig. Zunächst rufen wir den Lemmatisierungs-Algorithmus `WordNetLemmatizer()` auf und übergeben ihn mit „`=`“ der Variable `wnl`. Wir verwenden `wnl`, um nicht den vollen Befehl und darin enthaltene Funktionen wiederholt eingeben zu müssen. In unserem Falle muss dieser Schritt nur einmal durchgeführt werden, da dieser Algorithmus eine Art Wörterbuch und Regelwerk darstellt, unter welchen Bedingungen Worte zu einer Grundform reduziert und zusammengeführt werden (siehe Code 11.21 und 11.22). Um einen Text zu lemmatisieren, müssen Sie die Funktion `wnl.lemmatize()` aufrufen und ein Token in die Klammer eintragen. Wenn Sie zum Beispiel „walking“ oder „walked“ an die Funktion übergeben, dann wird Ihnen das lemmatisierte „walk“ ausgegeben. Sie können nun eine Liste mit den Tokens eines Filmskriptes lemmatisieren, indem Sie einen Listenabgleich schreiben. Im ersten Teil des Listenabgleiches definieren Sie, dass ein Wort dieser Liste

durch dessen Wortstamm ersetzt wird (Code 11.21). Der zweite Teil des Befehls übergibt die lemmatisierten Worte einer Liste mit dem Namen `"text"`, damit wir die lemmatisierten Wörter durch Stemming auf den Wortstamm reduzieren können.

Code 11.21 Lemmatisierung und Stemming der Tokens innerhalb eines Textes

```
wnl = WordNetLemmatizer()
text = [wnl.lemmatize(word) for word in text]

ps = PorterStemmer()
text = [ps.stem(word) for word in text]
```

Für das Stemming benötigt es ebenfalls zwei Skriptzeilen. Zeile 1 lädt den Stemming-Algorithmus und Zeile 2 führt den Listenabgleich durch, um die Tokens in einer Liste zu stemmen. Um den Stemming-Algorithmus in unsere Python-Sitzung zu laden rufen Sie den `PorterStemmer()`-Befehl auf und übergeben die gestemmten Tokens einer Variable. Wir nennen diese Variable `ps`, was ein Kürzel des Stemming-Algorithmus ist. Sie müssen sich den Stemming Algorithmus wie ein Buch vorstellen, das ein Grammatik-Regelwerk beinhaltet. Dieses Buch müssen Sie auch bloß einmal aufschlagen, um die Regeln nachlesen zu können. Gleiches gilt für den Lemmatisierungs-Algorithmus, der, um bei diesem Sinnbild zu bleiben, ein eigenes Regelwerk darstellt, das Sie aufschlagen können.

11.3.6 Schritte 1 bis 6 der Datenaufbereitung in einem Python-Programmskript zusammenfassen

Für Ihr Lernen und Verstehen der Programmierschritte ist die Einzeldarstellung in den Kapiteln 11.3.1 bis 11.3.5 notwendig. Im Forschungsalltag würden Sie die bisher vorgestellten Schritte der Datenaufbereitung jedoch in einem zusammenfassenden Befehl verwenden (Code 11.22). Wir nennen den Befehl `text_bereinigen()`, und definieren für Python, dass einerseits die Texte und andererseits Stopwords übergeben werden sollen, damit der Befehl ausgeführt werden kann. Den Befehl leiten wir mit `def text_bereinigen(text, stop_woerter):` ein, wodurch sich der Befehl auf ein Objekt namens `text` und eines namens `stop_woerter` bezieht und am Ende einen bereinigten Text ausgibt.

Code 11.22 Zusammenführen der einzelnen Bereinigungsschritte in einen eigens definierten Befehl (Fortsetzung nächste Seite)

```
def text_bereinigen(text, stop_woerter):
    """

    Diese Funktion ist dazu gedacht, die Texte zu bereinigen.
    Sie müssen hierfür einen Text in Listenformat übergeben,
    wobei jedes Listenobjekt eine Zeile im Filmskript darstellt.

    Zusätzlich zum Text müssen Sie eine Liste aller Stopwords
    (stop_woerter) übergeben, damit diese Funktion ausgeführt werden kann.

    Diese Funktion geht dabei wie folgt vor:

        1) der Text wird tokenisiert
        2) Stoppwörter werden entfernt
        3) Nur Nomen, Verben, Adjektive und Adverbien werden selektiert
        4) Die Wörter werden lemmatisiert
        5) Die Wörter werden gestemmt
        6) Alle Wörter, die weniger als drei Buchstaben haben, werden entfernt
        7) Alle "Sätze", die nur aus einem Wort bestehen, werden entfernt.


    Returns
    -------
    Gibt ein Listenobjekt aus, bei dem alle Wörter in dieser Liste bereinigt
    wurden.

    """

    # Typenliste für Wörter erstellen:

    pos_types = ["NN", "NNS", "NNP", "NNPS",
                 "JJ", "JJR", "JJS",
                 "VB", "VBD", "VBG", "VBN", "VBP",
                 "VBZ"]

    # Stemmer laden
    ps = PorterStemmer()
    wnl = WordNetLemmatizer()
```

```
    t = nltk.tokenize.word_tokenize(text)

    t = [re.sub("[^A-Za-z0-9]", "", word) for word in t if word not in
stopwoerter]
    t = [word[0] for word in nltk.pos_tag(t) if word[1] in pos_types]
    t = [wnl.lemmatize(word) for word in t]
    t = [ps.stem(word) for word in t]
    t = [word for word in t if len(word) > 1]

    return(t)
```

Die Zusammenfassung in Code 11.22 der bisherigen Codebeispiele aus den Kapiteln 11.3.1 bis 11.3.5 ermöglichen die Datenaufbereitung eines Filmskriptes für die Inhaltsanalyse. Um alle 626 Filmskripte zu bereinigen, die in unserem in Programmcode 11.11 erstellten Datensatz (`df`) in der Variable Text gespeichert sind, und einer neuen Variablen zu übergeben wenden wir Programmcode 11.23 an. Wie Sie sehen, benötigen wir wenige Zeilen und eine neu definierte Liste für die Bereinigung unserer Filmskripte.[10] Diese Liste können wir unserem Datensatz als neue Variable übergeben.

Code 11.23 Codezeilen zur Ausführung des Befehls zur Textbereinigung

```
data_cleaned = []

for df.Text:
    data_cleaned.append(text_bereinigen(text,stopwoerter))

df["bereinigte_texte"] = data_cleaned
```

11.3.7 Wie die Maschine lernt, Wissenschaft auszusprechen

Die bereinigten Texte bilden nun die Grundlage für das sogenannte *machine learning* (auf Deutsch: maschinelles Lernen) in Python. Dazu muss Python lernen, welche Schlagwörter Wissenschaft abbilden. Stellen Sie sich vor, *Indiana Jones* schlägt sich durch den Dschungel, um eine Götzenfigur zu finden. Stellen Sie sich weiter vor, im Film *Jurassic Park* flüchten die Wissenschaftler*innen vor

10 Da diese Rechenschritte sehr aufwändig sind, kann es sein, dass Sie mehrere Stunden dafür benötigen.

den Raptoren durch den Dschungel. Die Maschine muss nun anhand der Filmskripte lernen, die Wissenschaftler*innen von den vielen Bäumen des Dschungelwaldes und verfolgenden Dinosauriern usw. zu unterscheiden. Wie wir, so „lernt" auch der Algorithmus, auf dem das Topic Modeling fußt, besser, wenn nicht direkt das große Ganze – im Beispielfall 626 Filmskripte mit ungefähr 39 000 Seiten Textumfang – präsentiert wird, sondern die Texte nacheinander durchgearbeitet werden können. Dazu ist es nötig, die einzelnen Texte in kleinere Bestandteile aufzuteilen und Textstellen zu selektieren, in denen Wörter vorkommen, die mit Wissenschaft in Bezug gesetzt werden können.

Um den Algorithmus zu trainieren müssen wir die Filmskripte entweder auf Absatz- oder Satzlänge kürzen und diese Absätze oder Sätze nutzen (Guo et al. 2021). Durch den Fokus auf einzelne, kleine Textausschnitte lernt die Maschine Themenzuordnungen, die nachfolgend für die quantitativ-induktive Inhaltsanalyse des Gesamttextkorpus angewendet werden kann. Erfahrungsbasiert empfehlen wir Ihnen, die Filmskripte in Pakete zu je 100 Wörter aufzuteilen und aus diesen Textbestandteilen dann jeweils die Themen zu extrahieren. Aufgrund der großen Heterogenität des Textkorpus der Filmskripte zeigte sich, dass trotz mehrerer Testdurchläufe der LDA, die kleinteilige Aufteilung in Themenblöcke zu je 100 Wörtern wenig trennscharfe Topics generierte, d. h. die Darstellung von Wissenschaft in den Filmen unterrepräsentiert ist und wir wenig bis keine empirischen Ergebnisse für die Beantwortung unserer Forschungsfrage erhalten. In einem solch schwierigen Fall müssen wir zunächst in den tokenisierten Texten prüfen, wo in Filmskripten Begriffe vorkommen, die mit Wissenschaft (z. B. Fachdisziplinen, Laboratorien und Studium) verknüpft sind. Nach mehrfachem Sichten der generierten Topics und Lektüre ausgewählter Textpassagen in den Filmskripten können wir einen regulären Ausdruck (Code 11.24) schreiben, der Wortstämme von Begriffen erfasst und mit dessen Hilfe wir in den tokenisierten Filmskripten die Begriffe suchen und für die Analyse auswählen können.

Code 11.24 Regulärer Ausdruck mit Wortstämmen, die in den Filmskripts eindeutig mit Wissenschaft in Verbindung gebracht werden konnten

```
wissenschaft_muster =
r'(doct|^dr$|prof|studi|experim|univers|scien|librari|student|' + \
                            'physic|chemistry|math|engin|discov|^lab$|project)'

text = df[df.index == 0].bereinigte_texte
```

Für den Test von Code 11.24 wählen wir ein Filmskript aus, indem wir die erste Zeile unseres Datensatzes mit `df[df.index == 0]` auswählen und dann einen Text extrahieren, den wir hier exemplarisch `text` genannt haben. Das tun wir, um

innerhalb dieses Textes herausfinden, an welchen Positionen im bereinigten Text Wissenschaft oder Wissenschaftler*innen thematisiert werden. Diese Positionen von Begriffen zu Wissenschaft wollen wir sodann als Ausgangspunkt nutzen, um die Umgebung dieses Wortes zu ermitteln. Basierend auf Erfahrung definieren wir als Textumgebung einen Textausschnitt bestehend aus fünfzig Tokens vor und 49 Tokens nach dem mittels des regulären Ausdrucks gefundenen Tokens (z. B. „prof“ und „student“). Der gefundene Wortstamm stellt das hundertste Token im Ausschnitt des Filmskripts dar. Die Auswahl von 100 Tokens ermöglicht, auch arbeitsökonomisch einen anschaulichen Einblick in das Ergebnis unseres Tests zu erhalten, welcher nicht zu fokussiert (z. B. kleiner Ausschnitt mit 50 Tokens) oder zu breit ist (z. B. großer Ausschnitt mit 250 Tokens).

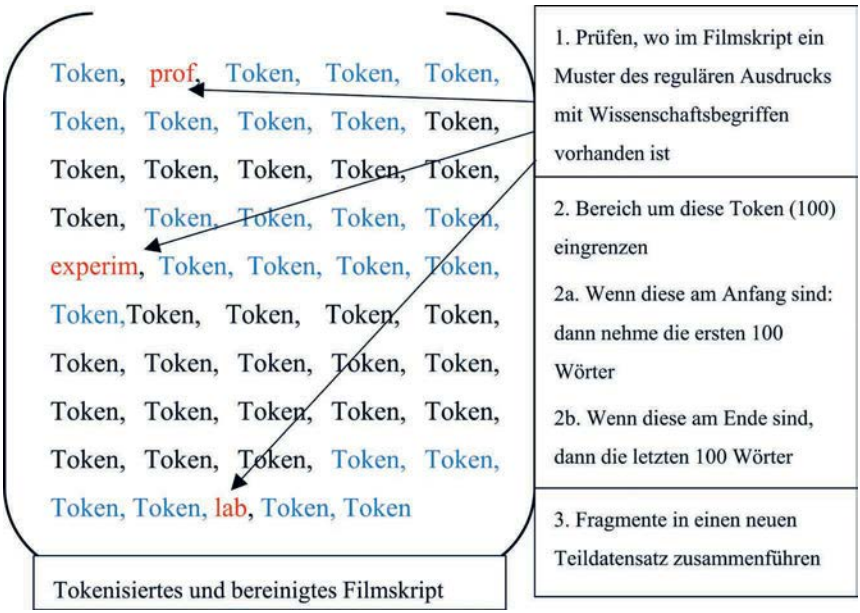
Für die Programmierung der tokenisierten Textausschnittsuche müssen grundsätzlich drei Fälle unterschieden werden. Erstens kann ein Token mit Wissenschaftsbezug ganz am Anfang, in der Mitte oder am Ende eines Textausschnittes stehen. Im ersten Fall müssen wir bedenken, dass weniger als 50 Wörter vor einem Token mit Wissenschaftsbezug stehen. Entsprechend würden wir den Textausschnitt zwischen dem ersten und bis zum hundertsten Token definieren. Im zweiten Fall können wir mit der regulär definierten Ober- und Untergrenze von ± 50 Tokens fortfahren. Der letzte Fall ist eigentlich eine Variation des ersten Falles. Hier müssen wir dann die letzten 100 Wörter eines Textes extrahieren und als Grundlage für die weiteren Auswertungen heranziehen.

Die Maschine lernt so, „Wissenschaft“ zu erkennen, und wir können die Umgebung aller mit Wissenschaft in Bezug stehenden Wortstämme als je eigene Texte in einen neuen Datensatz überführen. Der für den Test analysierte Text des Filmskripts besteht als neuer Datensatz aus Textabschnitten mit je 100 Tokens, wie in Abbildung 11.4 schematisch gestellt.

Mit dem grundsätzlichen Verständnis des Zwecks der Textausschnittbegrenzung können wir die technische Umsetzung mit Ziel „die Maschine lernt Wissenschaft zu erkennen“ angehen (Code 11.25). In Code 11.25 müssen wir zuerst eine einzelne Datenzeile aus unserem *pandas*-DataFrame herausschneiden. Im Befehl `df[df.index == 0]` wird beispielsweise durch die Zahl Null oder eine andere Zahl die Datenzeile angegeben, welche einem neuen Objekt mit Beispielnamen `d` übergeben wird. Die eckigen Klammern im `df`-Befehl ermöglichen es, gezielt einen Teil des *pandas*-DataFrames auszuwählen. Wie immer in Python-Skripten müssen in den eckigen Klammern eine oder mehrere Bedingungen definiert werden, mit deren Hilfe die Maschine eine Auswahl treffen kann. Zur Übung wollen wir nur eine Datenzeile ausschneiden, um mit dem Text der Datenzeile weiterzuarbeiten. Für die Weiterarbeit müssen wir Python über einen Indexwert angeben, welche Datenzeile ausgeschnitten werden soll.

Die ausgeschnittene Datenzeile muss nun als Liste bereinigter Tokens einem Objekt übergeben werden, welches als `text` benannt ist. Durch den Befehl `text = d.bereinigte_texte.iloc[-1]` erhalten wir Zugriff auf das bereinigte

Abbildung 11.4 Schematische Darstellung des Vorgehens zur Eingrenzung von Textstellen um relevante Tokens in einem Filmskript



Filmskript. Die Funktion `iloc[-1]` definiert, dass die letzte Zeile im Teildatensatz ausgewählt wird. Bei der Funktion `iloc[-1]` wenden wir einen Trick für einen Datensatz mit nur einer Zeile an, unabhängig davon, ob der Zeilenwert 0, 177 oder 1000 ist. Der Trick ermöglicht uns zudem, auf jeden Fall das korrekte Datenfragment des Beispiel-Filmskripts auszusuchen.

Dann erstellen wir eine Liste namens `textfragmente`, die dazu dient, die einzelnen Ausschnitte aus den Filmskripts zu speichern, in denen Begriffe vorkommen, die mit Wissenschaft zu tun haben. Wir machen das, da wir in der Regel nicht annehmen können, dass in Filmen, in denen Wissenschaft thematisiert wird, nur einmalig über Wissenschaft (z. B. in Form einer*eines Professor*in, einer wissenschaftlichen Entdeckung oder durch die Wissenschaft zu begegnenden Bedrohung) gesprochen wird.

Dazu müssen wir zunächst die Filmskripte in kleinere Textabschnitte, bestehend aus je 100 Wörtern, trennen. Das erfordert die Erstellung einer Liste mit den „Orten" der Tokens, die mit Wissenschaft zu tun haben, im bereinigten Filmskript. Damit ist gemeint, dass jedem Token eine feste Nummer zugeordnet wird, die umso niedriger ist, je weiter vorne im Filmskript das Token steht (z. B. `0` für die erste Stelle, `9` für die zehnte Stelle usw.). Um den Ort zu ermitteln, führen wir einen Listenabgleich mit zwei Iteratoren (`i und item`) durch. Iteratoren sind „Zeiger", die wie Zeiger einer Uhr auf die einzelnen Stunden oder Minu-

ten zeigen. Die dazugehörigen Befehle sind der `enumerate()`-Befehl und der `re.search()`-Befehl. Der Befehl `enumerate()` weist den einzelnen Listenobjekten – in unserem Falle Tokens – eine Nummer zu. Diese Nummer wird dem ersten Iterator `i` übergeben und ermöglicht in Verbindung mit dem zweiten Iterator `item`, das dazugehörige Token mit dem `search()`-Befehl aufzurufen. Mit dem `search()`-Befehl aus dem `re`-Paket können wir nach einem regulären Ausdruck (z. B. `"[A-Z]"` wenn Sie exakt einen Großbuchstaben des Alphabets suchen) oder einer Zeichenfolge (z. B. `"[A-Z][a-z]+"` für Worte, die mit einem Großbuchstaben beginnen und mit mindestens einem Kleinbuchstaben fortgeführt werden) im Text suchen. Beispielsweise programmieren Sie die Suche nach „Labor" in einem Objekt namens „Text" mit dem Befehl `re.search("Labor","Text")`.

Die Maschine hat nun zumindest in groben Zügen gelernt, was sie als „Wissenschaft" identifizieren kann. Nun müssen wir prüfen, ob grundsätzlich, und wenn ja, mindestens einmal Wissenschaft oder Wissenschaftler*innen in einem Filmskript adressiert werden. Das testen wir mittels des Programmcodes 11.25. Hier sehen wir im Anschluss an den `search()`-Befehl, dass eine längere Schleife mit einer if-, elif- und else-Abfrage, die durch ein `if len(list_elements) > 0` eingeführt wird. Das durch `else` eingeleitete `pass` am Ende dieser ersten Schleife definiert, dass das nächste Filmskript im Objekt auf Wissenschaftsbegriffe übergeprüft werden soll, selbst wenn im aktuellen Skript keiner dieser Begriffe gefunden wird. Diese Überprüfung stellt sicher, dass in den vorliegenden Filmskripten auch tatsächlich Wissenschaft und/oder Wissenschaftler*innen vorkommen.

Python rückt in der `if-else`-Abfrage eine `for`-Schleife ein. Die `for`-Schleife definiert, dass alle Ausschnitte in einem Filmskript, in denen Wissenschaft thematisiert wird, nacheinander angesteuert werden. Das erste `if l < 50` prüft, ob das Token (z. B. Labor) am Anfang des Filmskriptes steht. Trifft dies zu, dann wird ein Ausschnitt bestehend aus den ersten 100 Tokens ausgeschnitten. Das `elif l > len(text)-50` prüft analog hierzu, ob das mit Wissenschaft assoziierte Token zu den letzten 50 Wörtern des Filmskripts gehört. Der Befehl `len(text)` stellt sicher, dass die Gesamtzahl der Tokens eines Filmskripts, also die (bereinigte und tokenisierte) Wortzahl des Filmskripts systematisch analysiert werden. Wenn beide Bedingungen, Token mit Wissenschaftsbezug am Textanfang oder -ende, nicht zutreffen, dann stellt der Befehl `else` sicher, dass ein Textausschnitt von 100 Tokens mittels `text[l-50:l+50]` generiert wird, von denen 50 vor und 49 nach dem gefundenen, mit Wissenschaft assoziierten Begriff in unserer Liste stehen. Mithilfe der `append()`-Funktion werden die Textabschnitte anschließend in eine Liste namens `textfragmente` übernommen.

Nachdem wir die einzelnen Textstellen identifiziert und die Texte entsprechend aufgetrennt haben, ist es empfehlenswert, die Texte in einem strukturierten Format zwischenzuspeichern (z. B. csv- oder Excel-Datei). Die Texte werden durch

den letzten, außerhalb der Schleife definierten Listenabgleich `[' '.join(x) for x in df.bereinigte_texte]` in Code 11.25 zusammengeführt. Der Befehl-Teil `' '.join()` ermöglicht, dass alle Tokens mit jeweils einem Leerzeichen zwischen den Tokens zusammengeführt werden, womit die Variable `bereinigte_texte` überschrieben wird. Im Anschluss definieren wir den `df.to_csv()`-Befehl, und müssen Python den Namen für den zu speichernden Datensatz angeben. Da wir die Daten nicht aus Versehen wieder zusammenführen wollen, definieren wir sicherheitshalber einen Tabstop, in Python `\t`, als Trennung zwischen den Spalten unseres Datensatzes. Die Vorsichtmaßnahme ist empfehlenswert, weil wir mit Textdaten arbeiten, die im Zweifel Kommata enthalten, und eine csv-Datei Spalten und Zellen normalerweise durch ein Komma abtrennt.

Code 11.25 Auftrennen eines Filmskripts in kleinere Textabschnitte, bestehend aus je 100 Wörtern (Fortsetzung nächste Seite)

```
d = df[df.index == row]

text = d.bereinigte_texte.iloc[-1]

textfragmente = []

list_elements = [i for i, item in enumerate(text) \
                    if re.search(wissenschaft_muster, item)]

if len(list_elements) > 0:
    for l in list_elements:
        if l < 50:
            t = text[:100]
        elif l > len(text)-50:
            t = text[-100:]
        else:
            t = text[l-50:l+50]

        textfragmente.append(t)
        else:
            pass

    d_wiss = pd.concat([d]*len(textfragmente),ignore_index=True)
    d_wiss.bereinigte_texte = textfragmente
else:
    pass
```

```
df.bereinigte_texte = [' '.join(x) for x in df.bereinigte_texte]
df = df.drop_duplicates(subset=["bereinigte_texte"])
df.to_csv("filmdatensatz_nach_bereinigung_ohne_spacy.csv", sep="\t")
```

Als Beispiel wollen wir einen Text verfassen, in dem der Ausdruck „qualitative Forschung“ vorkommt. In diesem wollen wir, wie wir in Code 11.26 demonstrieren, nun nach diesem Ausdruck mit dem `search([SUCHMUSTER],[TEXT])`-Befehl suchen. Wenn Ihre Suche einen oder mehrere Treffer hat, dann wird Ihnen der erste Treffer angezeigt. Wenn Sie den Befehl ausführen, dann erhalten Sie als Ausgabe den „Ort“ (in Zeichennummern), an dem der gesuchte Ausdruck im Text vorhanden ist (Output 11.3). Wenn Sie hingegen einen längeren String, d. h. Text, vor sich liegen haben und alle Treffer einer Suchanfrage ausgeben möchten, dann müssen Sie `re.findall([SUCHMUSTER],[TEXT])` eingeben.[11]

Code 11.26 Verwendung des re.search()-Befehls zur Suche des Wortes „qualitativ“ in einem Text

```
Text = """ Das ist ein Text, in dem qualitative Sozialforschung betrieben
wird. Doch nicht nur das! Hier werden ebenso Methoden der quantitativen und
automatisierten Textanalyse beschrieben und mit qualitativen Methoden in
Verbindung gebracht. Damit können Sie qualitativ und quantitativ Forschen
"""

re.search("qualitativ",Text)
```

Output 11.3 Ausgabe des re.search()-Befehls

```
Out[0]: <re.Match object; span=(26, 36), match='qualitativ'>
```

11.3.8 Lexikon erzeugen und zu seltene bzw. zu häufige Wörter entfernen

Im Schritt acht von zehn der Datenaufbereitung erstellen wir ein Lexikon mit der `Dictionary()`-Funktion aus der `gensim.corpora`-Bibliothek. Dieses Lexikon

11 Eine Gesamtübersicht aller Suchanfragen und Funktionen des re-Paketes sehen Sie bitte auf *https://docs.python.org/3/library/re.html* nach.

dient dazu, den Wörtern eine einzigartige Wort-ID zuzuordnen. Wenn Sie zum Beispiel in dem Filmskript von „The Fly“ (1958/1986) den Text mit der Aussage „Dr. Delambre wurde im Teleporter mit einer Fliege gekreuzt. Delambre hat jetzt einen Fliegenkopf“ vorliegen hätten, dann würde die Dictionary-Funktion dem „Dr.“ eine 0, dem „Delambre“ eine 1, dem „wurde“ eine 2, dem „im“ eine 3, dem „Teleporter“ eine 4, dem „mit“ eine 5, dem „einer“ eine 6, der „Fliege“ eine 7, dem „gekreuzt“ eine 8, dem „hat“ eine 9, dem „jetzt“ eine 10, dem „einen“ eine 11 und dem „Fliegenkopf“ eine 12 zuordnen. Somit würde der Satz in Python eine Abfolge aus 0,1,2,3,4,5,6,7,8,2,9,10,11,12, darstellen.

Für die Erstellung des Lexikons für den Korpus wird folglich nur die Zuordnungen 0 = „Dr.“ usw. erkannt, die wichtig ist, um das weiter unten in Kapitel 11.3.8 beschriebene *bag of words*-Format zu erstellen. So ist es auch mit der Lexikon-Funktion möglich, einfach auszuzählen, wie häufig Wörter mit ID = 0, ID = 1 usw. vorkommen. Durch diese Repräsentation des Textes als sogenannter Vektor wird die Textgröße reduziert und kann damit schneller verarbeitet werden (siehe Box 11.5). Das Vektorformat repräsentiert die Grundannahme der LDA, dass nicht die Reihenfolge, sondern die Verteilung der Worthäufigkeiten über Texte hinweg latent vorliegende Themen aufzuzeigen hilft.

Box 11.5 Definition Vektor
Ein Vektor ist eine Abfolge von Zahlen, die in der Mathematik und Physik Richtungen und die Stärke einer „Kraft“ angibt, die in diese Richtungen wirkt. In unserem Fall besteht ein Vektor aus den Wörtern und den Worthäufigkeiten pro Text. Diese sind wie das sprachliche Pendant zum mathematischen Vektor.

In der Programmierung müssen wir für die Erstellung eines Wort-ID-Lexikons die `Dictionary()`-Funktion einem Objekt mit Beispielnamen `lexikon` zuordnen. In das Lexikon werden alle Ausschnitte aus den Filmskripten aufgenommen, in denen Begriffe mit Bezug zu Wissenschaft gefunden werden. Um alle Filmskripte im Korpus zu erfassen, müssen wir eine `for`-Schleife schreiben, die jeden Abschnitt systematisch nacheinander ansteuert und auf Wissenschaftsbegriffe überprüft. Um relevante Abschnitte des Filmskriptes in das Lexikon aufzunehmen, müssen wir hierfür die Funktion `lexicon.add_documents()` aufrufen und über die Klammern das Filmskript an das Lexikon übergeben.

Die Filmskript-Abschnitte liegen jedoch nicht als einfacher Text, sondern tokenisiert vor (Kapitel 11.3.1). Folglich müssen wir einen Listenabgleich durchführen und die Tokens einzeln mit dem Befehl `[token for token in text]` der for-Schleife übergeben (Code 11.27).

Code 11.27 Erstellung eines gensim-Lexikons und Hinzufügen der Filmskripte zum Lexikon

```
# ============================================================================
# Dictionary generieren
# ============================================================================

lexikon = Dictionary()
for text in df.data_cleaned:
    lexikon.add_documents([token for token in text])
```

Das Lexikon ermöglicht uns nicht nur, die für unsere Untersuchung relevanten Wörter zu identifizieren, sondern auch Wörter aus dem Korpus zu entfernen, die zu häufig oder zu selten vorkommen. Sehr häufig vorkommende Wörter führen zu (vielen) Überschneidungen und Ähnlichkeit von Themen, d. h., dass Themen nicht trennscharf erkennbar sind. Hingegen werden selten vorkommende Wörter beim Topic Modeling als eigene Themen erkannt, sofern sie gemeinsam in einem Filmskript auftreten. Zu große Trennschärfe führt dazu, dass Themen extrahiert werden, die lediglich einen Film, wenige Filme oder nur bestimmte Filme widerspiegeln, beispielsweise die *X-Men*-Filme mit filmspezifischen Worten wie *Magneto* oder *Storm* (Tabelle 11.2).[12]

Die häufigsten und seltensten Wörter aus unserem Datensatz entfernen wir aus dem `lexikon`-Objekt mit der `filter_extremes()`-Funktion (Code 11.27). In der Klammer der `filter_extremes()`-Funktion definieren wir durch `no_below` bzw. `no_above` einen Grenzwert, der alle Wörter bzw. Word-IDs aus dem Korpus entfernt, die in geringerer oder größerer Anzahl vorkommen als durch den Grenzwert festgelegt wird. Für die Festlegung des Grenzwertes müssen Sie eine forschungspragmatische, jedoch inhaltsbezogene Entscheidung treffen. Wenn Sie z. B. merken, dass Ihre Themen durch Wörter dominiert oder erst erzeugt werden, die einmalig vorkommen und keine weitere Bedeutung haben. Oder wenn Wörter so häufig vorkommen, dass sie in allen Themen auftreten. Wenn sie den Grenzwert auf zehn Worte festlegen, und alle Wörter aus dem Korpus entfernen möchten, die weniger als zehn Mal vorkommen, setzen Sie den Grenzwert entsprechend mit `no_below = 10` fest. Wenn Sie hingegen Wörter

12 Ergänzend zu inhaltlichen Aspekten reduziert die Selektion den Korpus und damit die Berechnungskapazitäten. Zum Vergleich: Ein Topic Model mit knapp 100 000 Texten und 50 000 Tokens benötigt für einen Berechnungsdurchgang mehrere Stunden. Ein Topic Model mit 500 000 Texten und 100 000 Tokens hingegen mehrere Tage. Wie in Kapitel 11.4.4 beschrieben, benötigen Sie viele Durchläufe mit unterschiedlicher Themenzahl, wodurch die Auswertungen Wochen oder Monate in Anspruch nehmen können!

aus dem Korpus entfernen wollen, die in mehr als 50 % der Texte vorkommen, dann programmieren Sie `no_above = 0.5`, für 60 % `no_above = 0.6` usw. Eine weitere Selektionsmöglichkeit ist, die 100 am häufigsten vorkommenden Wörter des Textkorpus mit der Funktion `lexikon.filter_n_most_frequent(100)` zu löschen. Der Code 11.28 zeigt Ihnen beispielhaft, wie Sie den Selektionsprozess programmieren können. Die zwei `print()`-Befehle helfen bei der Kontrolle, wie groß unser Textkorpus vor und nach der Bereinigung ist (Output 11.4).

Die `filter_extremes()`-Funktion ermöglicht uns auch zu Testzwecken oder um die Arbeitskapazitäten des Computers nicht zu überfordern, eine Begrenzung der Anzahl der aus den Texten zu extrahierenden Wörter. In der Klammer definieren Sie mit `keep_n` den zu erstellenden Textkorpus auf eine bestimmte Anzahl der am häufigsten auftretenden Wörter zu beschränken. In gensim ist der (für Sie nicht sichtbare) Standardwert mit 100 000 festgelegt. Um Platz in Ihrem Arbeitsspeicher zu sparen, können Sie den Wert auf 20 000 Wörter bzw. Tokens begrenzen, indem Sie `keep_n = 20000` eintragen.

Code 11.28 Bereinigung des Korpus um die am häufigsten und am seltensten auftretenden Wörter

```
print("Anzahl der Tokens im Lexikon vor der Bereinigung:", len(lexikon))
lexikon.filter_extremes(no_below=5, no_above=0.3, keep_n=50000)
print("Anzahl der Tokens im Lexikon nach der Bereinigung:", len(lexikon))
```

Output 11.4 Größe des Korpus bestehend aus Filmskripten vor und nach der Bereinigung

```
Anzahl der Tokens im Lexikon vor der Bereinigung: 109610
Anzahl der Tokens im Lexikon nach der Bereinigung: 22373
```

11.3.9 Korpus im „bag of words"-Format erzeugen

Im nächsten Schritt wollen wir unsere Texte in ein *bag of words*-Format übersetzen, damit die LDA die Zuordnung von Wörtern zu Wort-Identifikationsnummern (kurz: IDs) erkennt. Hierfür benötigen wir ein Lexikon, das die Zuordnung von Tokens zu den Wort-IDs kennt. Zur Erinnerung: Ein *bag of words*-Format trennt die einzelnen Wörter von ihrem Kontext und berücksichtigt nur die Worthäufigkeit im Textkorpus als Information. Im Gegensatz zu den Grundannahmen der qualitativen Textanalyse (Sprache funktioniert nicht beliebig, sondern folgt in ihrem Aufbau Regeln), folgt die LDA der Annahme, dass sich auf statistischer Ba-

sis Themen aufgrund des systematischen gemeinsamen Auftretens bzw. Fehlens von Wörtern identifizieren lassen (McFarland et al. 2013; Wieczorek et al. 2021).

Das *bag of words*-Format muss als Liste definiert werden, welche mit der in Kapitel 11.3.7 erstellten Liste des Datensatzes verglichen werden kann. In Code 11.29 wird die Liste mit `corpus` benannt. Der Liste `corpus` übergeben wir alle als Listen von Tokens vorliegenden Filmskripte, die in der Variablen `bereinigte_texte` im Datensatz gespeichert sind, und führen den Listenabgleich durch.

Code 11.29 Listen in einer Liste zusammenführen

```
corpus = [text for text in df.bereinigte_texte]
```

Sätze werden in ein *bag of words*-Format mit der Funktion `doc2bow()` im Lexikon umgewandelt.[13] Für den Listenabgleich mithilfe des in Kapitel 11.3.9 erstellten Lexikons erfolgt eine Übersetzung zwischen Token und Wort-IDs für eine Textstelle durch die `doc2bow()`-Funktion. Diese Rechenoperation muss selbstverständlich für jede Textstelle im Filmskript-Korpus-Objekt durchgeführt werden. Um den Arbeitsspeicher des Computers nicht zu überfüllen, müssen wir das `corpus`-Objekt mit der *bag-of-words*-Repräsentation unserer Filmskripte überschreiben.

Code 11.30 Erstellung eines Gesamtkorpus im bag-of-words-Format

```
corpus = [lexikon.doc2bow(text) for text in corpus]
```

11.3.10 Wörter im Korpus gewichten

Im letzten Schritt der Datenaufbereitung werden Wörter im sehr heterogenen Filmskript-Korpus gewichtet. Die Gewichtung ermöglicht es, die Trennschärfe zwischen den einzelnen Themen zu erhöhen und die Themen besser interpretieren zu können. Durch das Gewichten wird relevanten, jedoch in einem Textkorpus seltenen Wörtern ein höherer Stellenwert zugemessen. Angenommen, ein Filmskript bestehend aus 100 Wörtern enthält zehn Mal den Begriff „Professor*in," wobei der Begriff „Professor*in" im Korpus der 626 Filmskripte nur 15 mal vorliegt. Zur Beantwortung unserer Forschungsfrage nach dem gesell-

13 Die Umwandlung des Korpus bestehend aus ganz vielen Tupeln erhält dabei die Struktur `(Wort-ID, Anzahl des Vorkommens im Korpus)` die Ordnung für einen Listenabgleich (Code 11.30).

schaftlichen Bild von Wissenschaft(ler*innen) im Hollywoodfilm ist dieser Begriff jedoch zentral. Mit der Gewichtung würde der Begriff „Professor*in" gegenüber häufigen Begriffen aufgewertet. Angenommen, im selbigen Filmskript findet sich zwanzig Mal der Begriff „Studierende," welcher jedoch 1000 mal im Filmskript-Korpus vorkommt, dann ist das Wort „Studierende" zwar häufiger im Textkorpus vertreten, ist aber weniger typisch für das spezifische Filmskript als „Professor*in".

Die Gewichtungen werden im Verfahren der sogenannten tfidf-Gewichtung schrittweise (= generativ) erstellt; tfidf bedeutet ausgeschrieben *term-frequency inverse-document-frequency* (Worthäufigkeit und umgekehrte Dokumentenhäufigkeit). Um die Gewichtung durchzuführen, rufen wir in der `gensim.models`-Bibliothek den Befehl `TfidfModel` auf (siehe erste Zeile in Code 11.31). Dem `TfidfModel()`-Befehl übergeben wir das in Code 11.30 erstellte `corpus`-Objekt für eine automatische Gewichtung.[14] Das neue Objekt mit Namen `gewichtung` modifiziert den Korpus, den wir neu `gewichtung_corpus` nennen. `gewichtung_corpus` ist die Grundlage für unsere LDA, deren Durchführung in Kapitel 11.4 erklärt wird.

Code 11.31 Durchführung einer tfidf-Gewichtung des Korpus

```
from gensim.models import TfidfModel
gewichtung = TfidfModel(corpus)
gewichtung_corpus = gewichtung[corpus]
```

11.4 Durchführung einer Latent Dirichlet Allocation

Topic Modeling als statistisches Verfahren beruht auf unüberwachtem maschinellen Lernen (siehe Kapitel 8.2.2 für einen Überblick). Für die softwareunterstützte Auswertung mit Python müssen Sie daher im Vorfeld die Parameter der Modelle einstellen, d. h. die Anzahl der Themen festlegen, die Ihr Topic Model finden soll, wie die potenzielle Wahrscheinlichkeitsverteilung der Themen ist, wie die Wörter, auf deren Basis die Themen gefunden werden, innerhalb und zwi-

14 Als Option, wie genau gewichtet werden soll, können Sie `smartirs =` angeben. Danach folgen drei Buchstaben, mit denen die genaue tfidf-Gewichtung angegeben wird. Zum Beispiel `'ntc'`, wobei das n die relative Häufigkeit der Begriffe, das t die umgekehrte Häufigkeit, mit der ein Begriff im Korpus auftritt, bedeutet. Zuletzt sagt c aus, dass ein Begriff über die Wurzel der Korpusgröße, das heißt der Anzahl der Dokumente, gewichtet wird (sogenannte *cosine normalization*). Für einen Überblick über die einzelnen Optionen kann das Online-Handbuch von gensim zu Rate gezogen werden, das unter folgendem Hyperlink erreichbar ist: *https://radimrehurek.com/gensim/models/tfidfmodel.html*.

schen den Texten gewichtet werden. Darüber hinaus können Sie eine Gewichtung der Wörter über deren gehäuftes Auftreten in wenigen Texten versus geringes Auftreten in vielen Texten vornehmen (sogenannte *tfidf* oder *Term frequency inverse document frequency*). Dabei müssen Sie aber vorsichtig sein, da all diese Einstellungen das Ergebnis Ihres Topic Models stark beeinflussen können! Auch die Auswahl einer geeigneten Themenzahl stellt ein weiteres Problem dar, das sowohl von der Interpretierbarkeit der Themen als auch deren Überschneidungsfreiheit (= Kohärenz oder eng. *coherence*) abhängt. Beides hängt wiederum mit den Gewichtungen zusammen, die in Ihr Modell miteinfließen.[15]

Bevor wir mit dem auf der LDA-basierenden Topic Modeling beginnen, fassen wir an dieser Stelle zusammen, welche Schritte der Datenaufarbeitung wir erfolgreich durchgeführt haben, und welche Objekte und Variablen wir erstellt haben. Wir verfügen über einen bereinigten Textkorpus der 626 Filmskripte, wir haben ein Word-ID-Lexikon erstellt und wir haben den Textkorpus in ein *bag of words*-Format übersetzt. Zuletzt haben wir eine tfidf-Gewichtung vorgenommen. Die knappe Zusammenfassung in vier Ergebnissen verdeutlicht, warum für automatisierte Auswertungstechniken der induktiv-quantitativen Inhaltsanalyse in Abbildung 1.2 grob 70 bis 80 % der Forschungszeit für Datenarbeit veranschlagt wurden.

Die Durchführung der LDA und die Themenauswahl erfolgt in mehreren Schritten: In Kapitel 11.4.1 werden die Software-Pakete zur Durchführung der LDA erklärt. In Kapitel 11.4.2 lernen Sie, Topic Models zu berechnen und welche Optionen `gensim` bereitstellt, damit der LDA-Algorithmus Themen aus dem Textkorpus extrahiert, und wie Sie diese speichern können. In Kapitel 11.4.3 lernen Sie, wie Sie die Themenzahl Ihres finalen Topic Models auswählen können. Hierzu werden wir gemeinsam Kohärenzmaße berechnen, visualisieren, interpretieren und dann potenziell qualitativ zu analysierende Modellkandidaten bestimmen. In Kapitel 11.4.4 zeigen wir Ihnen, wie Sie die Berechnung einer Vielzahl von Topic Models mittels for-Schleife automatisieren können.

11.4.1 Benötigte Pakete laden

Die `gensim.models`-Bibliothek enthält eine Vielzahl an Befehlen und Funktionen für die LDA-Durchführung. Die Programmierung in Code 11.32 ruft die wesentlichen Funktionen von `gensim.models` auf.

- `LdaModel`-Befehl ruft den Algorithmus auf, mit dessen Hilfe Python versucht, die Themenstruktur im bereinigten Textkorpus zu erkennen.

15 Diese Gewichtungen werden uns in Kapitel 11.4.2 als Alpha- und Eta-Werte unserer LDA begegnen.

- Der `gensim.models.coherencemodel`-Befehl ermöglicht es, Topic Models mit unterschiedlicher Themenzahl über Kohärenzmaße miteinander zu vergleichen.
- Der `Dictionary`-Befehl aus der Bibliothek `gensim.corpora` ermöglicht es dem LDA-Algorithmus, die Zuordnung von Wort-IDs zu Wörtern abzurufen, d. h. die mathematische Zuordnung von Wörtern und Texten zu Themen erfolgt.[16]
- Das `pyLDAvis`-Paket enthält Befehle zur interaktiven Visualisierung der Topic Modeling-Ergebnisse.

Code 11.32 Befehlszeile für den Import der für die LDA benötigten Pakete und Bibliotheken

```
# NLP-LDA-Pakete importieren
from gensim.models import LdaModel
from gensim.models.coherencemodel import CoherenceModel as cm
from gensim.corpora import Dictionary
```

Code 11.33 Grundstruktur und Definition des LDA-Befehls aus dem gensim-Modul

```
trainingsset = int(np.round(len(corpus)/5*4,0))

lda_ergebnis = LdaModel(corpus = gewichtung_corpus,
                                num_topics=5,
                                iterations=150,
                                chunksize = trainingsset,
                                id2word=lexikon,
                                per_word_topics=True,
                                alpha='auto',
                                eta = 'auto')
```

16 Wenn Sie sich dafür entscheiden, eine LDA über mehrere Prozessoren berechnen zu lassen, dann benötigen Sie in Windows den im Paket `multiprocessing` implementierten `freeze_support`-Befehl. Dieses Paket ist für Windows-Nutzer relevant, wenn eine LDA durchgeführt werden soll, die die Berechnungen auf mehrere Prozessoren verteilt. Windows hat hier häufig das Problem, dass parallele Berechnungen zu einem „Einfrieren" der Berechnungen führen, d. h., dass Python einfach abstürzt und den Befehl nicht durchführt – und zwar ohne es zu melden!

Für die Erstellung eines Topic Models müssen wir den LDA-Algorithmus mit dem `LdaModel()`-Befehl in der `gensim.models`-Bibliothek aufrufen. Die Programmierungen in Code 11.33 definieren für Python, wie genau die Themenzuweisung stattfinden soll und was bei der Themenberechnung zu beachten ist.

Wie stets in der induktiv-quantitativen Inhaltsanalyse müssen wir uns wieder in die Daten vortasten. Das Vortasten erfolgt mithilfe eines Trainingssets. Die erste Zeile in Code 11.33 setzt die Größe des Trainingssets auf 80 % (`int(np.round(len(corpus)/5*4,0))`) des Filmskript-Korpus fest. Diese 80/20-Regel ist gängig im Bereich des *machine learning* (Géron 2019, S. 30 f.) und folgt der Argumentation, dass ein Modell einerseits präziser wird, je mehr Daten dem zugrundeliegenden Algorithmus „gezeigt" werden, auf der anderen Seite aber genügend Daten vorliegen müssen, damit die Generalisierbarkeit des Modelles geprüft werden kann. Bedenken Sie dabei, dass das Ziel beim *machine learning* in der Regel die Erkennung von Mustern und deren Vorhersage in neuen, noch unbekannten Daten ist. Sie können auch andere Anteile (z. B. 50 %) oder eine fixe Anzahl Dokumente wählen, die dem Algorithmus immer wieder zufällig „gezeigt" werden, um sie danach auf die zurückgehaltenen Dokumente anzuwenden und dabei zu prüfen, inwiefern die Themenzuordnung akkurat ist, die anhand des Trainingssets generiert wurde.

Um diese Prozentzahl zu ermitteln, brauchen Sie die Gesamtzahl der Dokumente, die Sie analysieren wollen. Hierfür rufen Sie die Anzahl der verwendeten Datenzeilen mit `len([CORPUS])` auf. Um nun auf die 80 % der Texte zu kommen, teilen wir die Anzahl der Dokumente zunächst durch fünf und multiplizieren sie dann mit vier. Sie könnten die Anzahl theoretisch auch durch zehn teilen und dann mit acht multiplizieren. Sollten Sie entsprechend 50 % der Texte analysieren wollen, dann können Sie die Anzahl der Texte durch zwei Teilen, wenn Sie hingegen genau 2000 Texte dem Algorithmus „zeigen" möchten, dann geben Sie einfach 2000 ein. Der `LdaModel()`-Befehl kann nur mit ganzen Zahlen *(integers)* arbeiten. Um Nachkommastellen bei Kalkulationsbefehlen zu vermeiden, können Zahlen mit dem `round()`-Befehl aus dem `numpy`-Modul gerundet und anschließend mit `int()` zu einem *integer* (= ganze Zahl) umgewandelt werden. Die 0 nach dem Komma im `round()`-Befehl gibt an, dass keine Nachkommastelle beim Runden erstellt werden soll. Leider gibt es für die Schätzung der Modellparameter keine exakten Vorgaben. Eine grobe Daumenregel bzw. Erfahrungswerte besagen, für die Schätzung der Modellparameter im Trainigsset eine eher hohe Zahl an Iterationen anzugeben (z. B. 100, 150, 250). Wählen Sie eine zu geringe Anzahl an Iterationen, so besteht die Gefahr, dass Ihr Topic Model ungenau wird (= schlechte Themenzuordnung von Worten und Texten). Sie sollten jedoch bedenken, dass eine sehr hohe Iterationsanzahl ebenfalls nicht immer die sichere oder gute Wahl ist. Die Berechnung des Topic Model mit unserem kleinen Datensatz (in Kapitel 11.3.9 reduzierter Textumfang der 626 Filmskripte) hat eine Berechnungszeit von etwa ein bis zwei Minuten. Bei großen Textkorpora (z. B.

deutlich mehr als 39 000 Seiten) müssen Sie davon ausgehen, dass eine Iteration mehrere Stunden in Anspruch nehmen kann.

Die Option `chunksize` definiert, wie viele Textdokumente bei jedem Schritt für das Training Ihres Modells verwendet werden. Die programmierte Standardeinstellung ist 2 000 Textdokumente. Bei größeren Textkorpora ist es empfehlenswert, diesen Wert für das Trainigsset zu erhöhen und bei kleineren Textkorpora zu verringern. Bitte beachten Sie, dass der `chunksize`-Befehl sehr viel Arbeitsspeicher von Ihrem Computer beansprucht. Sollte die Größe des Trainingssets Ihren Arbeitsspeicher „überfordern", weil Ihr Computer gleichzeitig tausende Texte im „Gedächtnis" behalten muss und diese zugleich in große Datenstrukturen zerlegt, dann müssen Sie eine kleinere Textzahl als Trainingsset wählen (z. B. weniger als 80 % des Textkorpus).

In Code 11.33 wird mit dem `id2word`-Befehlt in der Klammer von `LdaModel()` das Lexikon angegeben, in dem die „Übersetzung" zwischen Wort-IDs und den Tokens gespeichert ist. Im obigen *The Fly*-Beispiel würde „Dr." die Wort-ID 0, „Delambre" die Wort-ID 1 usw. haben, sodass Python die Tokens erkennen kann, um Wissenschaft auszusprechen. Wenn Sie diese Definition nicht angeben, dann würde Python die Wort-IDs anstelle der Tokens, die den Themen zugeordnet werden, ausgeben. Weiter wird im `random_state`-Befehl ein Ausgangswert für den Auswahl-Algorithmus angeben. Indem wir einen festen Ausgangswert für die Ermittlung der Modellparameter unseres Trainingssets setzen, können die Ergebnisse repliziert werden. Basierend auf dem Ausgangspunkt zur Ermittlung der Modellparameter berechnet der `per_word_topics`-Befehl die Wahrscheinlichkeit für jedes Wort, einem Thema zugeordnet zu werden.[17]

Zuletzt müssen wir in Code 11.33 die Dokument-Themen und die Wort-Themen-Gewichtungen definieren. Die Dokument-Themen-Gewichte können wir mit dem Befehl `alpha =`, die der Themen-Wort-Gewichte mit `eta =` vorgeben. Geben Sie höhere Alpha- und Eta-Werte an, so werden einem Filmskript oder Wort tendenziell mehrere Topics zugewiesen. Umgekehrt generieren kleine Alpha- und Eta-Werte wenige(r) Themen (z. B. `alpha = 0.01, eta = 0.001`). Um das Versuch-und-Irrtum-Spiel zu vermeiden, können wir für `alpha` und `eta` den Befehl `'auto'` eingeben (statt Werte). Mit „auto" wird definiert, für den LDA-Algorithmus die Alpha- und Eta-Werte generativ festzulegen, d. h., diese aufgrund des maschinellen Lernvorgangs während der Iterationen zu verändern und zu optimieren. Das Resultat des Topic Model-Lernvorgangs ist eine Optimierung der Extraktion von Themen und damit der Wahrscheinlichkeit, für Menschen interpretierbare Themen zu erzeugen.

Um das in Code 11.33 programmierte Topic Model mit fünf Themen zu be-

17 Damit erschöpfen sich die Befehlsoptionen für die LDA nicht. Sie reichen allerdings für unsere Zwecke aus. Eine vollständige Dokumentation der Befehlsoptionen finden Sie unter der Webadresse *https://radimrehurek.com/gensim/models/ldamulticore.html.*

rechnen und dabei 80 % des Korpus als Trainingsset zu verwenden, markieren Sie die Programmzeilen und führen diese mit F9 oder Rechtsklick → „run selection or current line“ aus. Das Ergebnis des Durchlaufs (hier mit fünf Themen) der LDA übergeben Sie, wie stets mit dem „=“-Zeichen, im Beispiel an ein neues Objekt namens `lda_ergebnis`.

Das in Code 11.33 programmierte Modell können Sie zur Wiederverwendung speichern (z. B. um Berechnungszeit für weitere Filmskriptanalysen zu sparen). Das trainierte Topic Model wird in gensim über die Funktion `.save("[Dateiname].model")` gespeichert (Code 11.34), wobei Sie den Platzhalter `[Dateiname]` durch einen sinnvollen Namen ersetzen müssen (z. B. Name `filmskripte_5.model`). Den Speicherort des Topic Models definieren Sie wieder mit dem `os.chdir()`-Befehl. Um die Übersicht zu wahren, sollten Sie gespeicherte Topic Models und (Zwischen-)Ergebnisse stets im selben Ordner speichern.

Code 11.34 Abspeichern des trainierten Modells

```
lda_ergebnis.save("lda_" +str(5) +"_themen.model")
```

11.4.2 Die Wort-Themen-Assoziationen: ein „Gefühl“ für die Daten bekommen

Im Soziolekt (= Umgangssprache von Sozialwissenschaftler*innen) bedeutet „ein Gefühl für die Daten bzw. die Themen zu bekommen“, eine vorsichtige erste Zuordnung von Tokens zu Themen vorzunehmen. Dafür betrachten wir die am stärksten mit unseren fünf Themen assoziierten zehn Tokens, die in Code 11.35 mit dem Befehl `lda_ergebnis.print_topics()` ausgeben werden. Mit dem Befehl `num_topics` wird die Anzahl auszugebender Themen und mit der Befehl `num_words` die Anzahl der Worte festlegt, die am stärksten mit den modellierten Themen assoziiert sind. Mit der Standardeinstellung `num_topics` von gensim werden Ihnen bei diesem Befehl die ersten statistisch am höchsten gewichteten zehn Wörter pro Thema angezeigt. Möchten Sie beispielsweise die ersten 20 Wörter betrachten, dann geben Sie den Befehl `topn = 20` mit Komma getrennt in der Klammer ein. Nach selbem Muster lautet die Programmierung für ein Topic Model mit 50 Themen, für die jeweils die ersten 25 Wörter ausgegeben werden sollen, dann `lda_ergebnis.print_topics(num_topics = 50, topn= 25)`. Den Befehl führen Sie wie (inzwischen) gewohnt mit F9, „Strg + Enter“ oder F5 (für einen Gesamtdurchlauf Ihres Skriptes) aus.

Code 11.35 definiert die Ausgabe von Themen mit den zehn prägnantesten Tokens je Thema. Die in Code 11.35 integrierte `print()`-Funktion generiert Output 11.5.

Code 11.35 Aufruf der zehn Tokens, die am stärksten mit den jeweiligen Topics assoziiert sind

```
model_topics = lda_ergebnis.print_topics(num_topics = 5, num_words=10)
```

Output 11.5 Ausgabe der zehn Tokens, die am stärksten mit den Themen des Topic Models assoziiert sind

```
Out[10]:
[(0,
  '0.001*"okay" + 0.001*"gon" + 0.001*"love" + 0.001*"pleas" +
0.001*"beat" + 0.001*"doctor" + 0.001*"judg" + 0.001*"dog" + 0.001*"god" +
0.001*"professor"'),
 (1,
  '0.002*"gon" + 0.002*"professor" + 0.001*"love" + 0.001*"okay" +
0.001*"everyth" + 0.001*"mayb" + 0.001*"home" + 0.001*"father" + 0.001*"feel"
+ 0.001*"doctor"'),
 (2,
  '0.002*"frock" + 0.002*"gon" + 0.001*"creed" + 0.001*"book" + 0.001*"okay" +
0.001*"love" + 0.001*"kill" + 0.001*"pleas" + 0.001*"god" + 0.001*"raptor"'),
 (3,
  '0.001*"captain" + 0.001*"travel" + 0.001*"presid" + 0.001*"stare" +
0.001*"machin" + 0.001*"beat" + 0.001*"street" + 0.001*"gun" + 0.001*"fire" +
0.001*"dog"'),
 (4,
  '0.001*"pleas" + 0.001*"offic" + 0.001*"gon" + 0.001*"kill" + 0.001*"doctor" +
0.001*"love" + 0.001*"world" + 0.001*"okay" + 0.001*"agent" + 0.001*"god"')]
```

Output 11.5 präsentiert eine Liste, die aus Tupeln besteht. Die Tupeln sind als Liste nach den Themennummern 0 bis 4 geordnet. Nach den Themennummern folgen Eintragungen wie „0.002 * professor“ in Thema Nummer/Topic 1. Die 0.002 zeigt dabei die Relevanz des Tokens für das Thema an. Relevanz bedeutet, dass mit jedem zusätzlichen Vorkommen des Tokens „professor“ in einem Text die Wahrscheinlichkeit steigt, dass der Text auch tatsächlich dem entsprechenden Thema durch die LDA zugeordnet wird. Die Werte für die tfidf-Gewichtungen (Kapitel 11.3.10) sind sehr gering und dürfen nicht mit der Angabe von Worthäufigkeit verwechselt werden. Die zehn Worte in den fünf Themen in Output 11.5 sind nicht überschneidungsfrei (z. B. „professor“ in Topics 0 und 1, „pleas“ in Topics 0, 2 und 4). Die mehrfachen Überschneidungen deuten darauf hin, dass eine höhere Anzahl Themen gewählt werden sollte, um Wort-Überschneidungen zu minimieren.

11.4.2.1 Die Text-Themenzuordnung: Das Gefühl für die Ergebnisse vertiefen

Für eine erste Zwischenanalyse der Ergebnisse kann eine Zuordnung von den Ausschnitten der Filmskripte zu den Themen mit dem `lda_ergebnis.get_document_topics()`-Befehl vorgenommen werden (Code 11.36).[18] In der Klammer müssen Sie ein Filmskript aus dem in Kapitel 11.3.9 generierten Korpus aufrufen, beispielsweise `corpus[0]`. Das in Output 11.6 abgebildete Listenobjekt besteht aus Tupeln, welche erst das Thema und dann die Wahrscheinlichkeit angegeben, welchen Themen der Textausschnitt (= `corpus[0]`) zugeordnet wird. Das Ergebnis ist, dass der Textausschnitt des Filmskripts am stärksten durch Thema 4 (Wert nahe eins) sowie geringfügig von den Themen 1 und 3 erfasst wird.

Code 11.36 Aufruf der zehn Tokens, die am stärksten mit den jeweiligen Topics assoziiert sind

```
lda_ergebnis.get_document_topics(corpus[0])
```

Output 11.6 Zuordnung des ersten Textabschnitts zu den fünf extrahierten Themen

```
Out[10]: [(1, 0.15574117), (3, 0.16045405), (4, 0.68269956)]
```

Code 11.37 wiederholt in einem Listenabgleich die Verbindung der Themen aus Output 11.6 für alle Filmskript-Textausschnitte in `corpus`. Hierfür fügen wir den `.get_documt_topics()`-Befehl in den Listenabgleich ein und ersetzen `corpus[0]` durch den Iterator `satz`. Die in Code 11.37 erstellte Liste, die aus Listen von Tupeln besteht, wird einem neuen Objekt mit „=“ zugewiesen.

Code 11.37 Zuweisung von Themen zu Dokumenten

```
themen_pro_dokument = [lda_ergebnis.get_document_topics(satz) for satz in
corpus]
```

18 Sofern Sie einen eigenen Objektnamen (z. B. *results*) definiert haben, in dem die Ergebnisse Ihrer LDA abgelegt sind, müssen Sie `lda_ergebnis` durch den entsprechenden Variablennamen austauschen.

Die Themenzuordnung kann nun für die spätere Modellauswahl getestet werden. Um die Zwischenergebnisse einer Interpretation und Prüfung der Sinnhaftigkeit der Topics zu unterziehen, müssen sowohl die Themen x Token-Matrix als auch Dokument x Themen-Matrix in einen *pandas*-DataFrame überführt und als Datentabelle exportiert werden. Die Zuweisung und der Export der Themen x Token-Zuordnung ist in Code 11.38 in vier Schritte unterteilt Dabei übergeben wir dem Objekt `model_topics` die zehn prävalenten, d. h. am stärksten auf die jeweiligen Themen ladenden Worte, indem wir

1. erneut die `.print_topics()`-Funktion aufrufen,
2. einen *pandas*-DataFrame mit dem `DataFrame()`-Befehl erstellen. In der Klammer greift der Befehl `data = ([x[1] for x in model_topics]` auf die in den jeweiligen Tupeln in `model_topics` an zweiter Stelle abgelegten Tokens und Ladungen. Mit dem Befehl `columns = ["ladungen_tokens"]` wird die Spalte in der Klammer des `pd.DataFrame` benannt. Der systematische Listenabgleich sorgt zuverlässig dafür, dass die Themenladungen für die Worte aufgerufen und an den DataFrame übergeben werden. Der columns-Befehl definiert eine Liste im `DataFrame()`, da der Befehl normalerweise an dieser Stelle eine Abfolge von Variablennamen erwartet,
3. mit dem Befehl `index.name = "Themennummer"` den Index umbenennen,
4. zuletzt den *pandas*-(pd)-DataFrame mit dem `.to_excel()`-Befehl exportieren.

Code 11.38 Export der zehn prävalentesten Wörter pro Topic in eine Excel-Datei

```
model_topics = lda_ergebnis.print_topics(num_topics = n,num_words=10)
model_topics = pd.DataFrame(data=[x[1] for x in model_topics],
columns=["ladungen_tokens"])
model_topics.index.name = "Themennummer"
model_topics.to_excel("topic_token" + str(5) + ".xlsx")
```

11.4.2.2 Die Text-Themen-Assoziation: Wie aus einem Gefühl für Themen eine Themenordnung entsteht

Nach der Ergebnissicherung in Excel der modellierten Topics erfolgt der zweite Generiere-und-Exportiere-Vorgang für die Dokument x Themen-Matrix. Code 11.39 schließt hierfür dreifach an den Listenabgleich in Code 11.38 an. Erstens wird ein Datensatz für Themen (definiert Anzahl Spalten) und Textausschnitte (definiert Anzahl Zeilen) erstellt. Zweitens werden die Zeilen, sofern sie nicht

bereits Werte für bestimmte Themen enthalten, mit Nullwerten aufgefüllt, ehe, drittens, das DataFrame-Objekt exportiert werden kann.

Wie in Code 11.38 nutzen wir für Code 11.39 den `DataFrame()`-Befehl, um das *pandas*-DataFrame-Objekt zu erstellen. Die Zellen des `pd.DataFrame` werden mit den Befehlen `index = range(len(df))` und `columns = ["topic_" + str(x) for x in range(5)]` befüllt. Befüllt werden die Zellen mit zwei Komponenten. Komponente eins ist ein Index mit Werten, die von 0 bis Anzahl der Zeilen −1 unseren Datensatz der Textausschnitte der Filmskripte umfasst. Um zu prüfen, wie viele Datenzeilen der Datensatz umfasst, können Sie entweder `range(len(df))` oder `df.index` angeben. Hier sehen wir, dass der Index von 0 bis 11 079 reicht, wir also 11 080 einzelne Ausschnitte aus den Filmskripten vorliegen haben. Für die Spalten des `pd.DataFrame` erzeugen wir durch den Listenabgleich fortlaufende Variablennamen wie `topic_0`, `topic_1` usw., die es uns ermöglichen, die Themen pro Dokument den korrekten Spalten zuzuordnen. Im DataFrame-Objekt stellt die Angabe `range(5)` sicher, dass beim Listenabgleich nur fünf Themen erzeugt werden. Selbstverständlich müssen Sie die Zahl entweder manuell ändern, wenn Sie ein Topic Model mit anderer Themenzahl definiert haben, oder durch einen Iterator (z. B. n) ersetzen, falls Sie den LDA-Algorithmus in einer for-Schleife mit flexibler Themenzahl ausführen.

Um die leeren Zellen innerhalb Ihres Datensatzes zu befüllen, nutzen wir den in Beispielcode 11.39 eingeführten `fillna(0)`-Befehl. Die 0 in der Klammer des Befehls übergibt den Wert 0 an leere Zellen. Sie können die 0 auch durch ein `-99` oder die `"NA"`-Zeichenfolge ersetzen, die anstelle des Wertes 0 in die leeren Zellen geschrieben wird.

Daneben möchten wir aber auch noch die Zuordnungswerte der Texte zu den Themen in die Zeilen füllen, die wir in unserem `themen_pro_dokument`-Objekt als Liste von Tupeln gespeichert haben (für eine Erläuterung von Tupeln siehe Kapitel 11.2.4.4). Die Befüll-Ausführung in Python folgt der Logik

a) steuere Zeile für Zeile im Listenobjekt an,
b) extrahiere alle Themennummern und kombiniere diese mit dem Präfix `topic_`, um eine Zuweisung zu den Zellen in der korrekten Spalte zu ermöglichen,
c) extrahiere alle Themenladungen für den jeweiligen Ausschnitt des Filmskripts und
d) übergebe diese Werte den passenden Zellen in der jeweiligen Datenzeile.

Schritt a erfordert die Definition einer `for`-Schleife, welche Zeile für Zeile aus der `themen_pro_dokument`-Liste aufruft. Der Einfachheit halber wird die zeilenweise Iteration mit `row` benannt. `row` übergibt dem Iterator die Zeilenzahl mit dem Befehl `in range(len(themen_pro_dokument))` an die `for`-Schleife. Um mit der `for`-Schleife die korrekten Spalten aufzurufen (Schritt b), übergeben wir

die Topics mit einem Listenabgleich an ein Objekt, das in Code 11.39 `keys` genannt wird. In Unkenntnis der potenziellen Topic-Anzahl wird ein Listenabgleich verwendet, der folgende Struktur aufweist: `["topic_" + str(x[0]) for x in themen_pro_dokument[row]]`. Der Befehl `(x[0])` steuert nun die erste Stelle in den Tupeln an, in denen die Themennummern gespeichert sind. Der `row`-Iterator ermöglicht es beim Listenabgleich, eine einzelne Zeile anzusteuern und die Werte aufzurufen. Der gleichen Logik folgt der Abruf der Themenladungen, die an der zweiten Stelle der jeweiligen Tupel gespeichert sind, welche dem Objekt `values` übergeben werden muss (Schritt c). Das zeilenweise Befüllen des Datensatzes (Schritt d) wird über die `.loc[ZEILE,SPALTE]`-Funktion gesteuert. Die `.loc[ZEILE,SPALTE]`-Funktion kombiniert die Spalteninformation in `key`, die Zeileninformation im Iterator `row` und die zu übergebenden Werte in `values`. Abschließend exportieren wir unser nun mit den Themen befülltes DataFrame-Objekt an Excel mit dem uns bereits bekannten `.to_excel()`-Befehl.

Code 11.39 Erstellung und Export eines pandas-DataFrame-Objektes

```
# Themen aus den Dokumenten extrahieren
themen_pro_dokument = [lda_ergebnis.get_document_topics(item) for item in
corpus]

# Themendatensatz erstellen und mit den korrekten Werten überschreiben
topics_df = pd.DataFrame(index = range(len(df)),
                              columns = ["topic_" + str(x) for x in range(5)])

topics_df = topics_df.fillna(0)

for row in range(len(themen_pro_dokument)):
    print("bearbeite Zeile:", row)
    keys = ["topic_" + str(x[0]) for x in themen_pro_dokument[row]]
    values = [x[1] for x in themen_pro_dokument[row]]
    topics_df.loc[row,keys] = values

topics_df.to_excel("document_topic_" + str(5) + ".xslx")
```

11.4.2.3 Durchschnittliche Themenprävalenz im Textkorpus und Themen-Alpha-Werte

Zuletzt müssen wir prüfen, wie hoch die durchschnittliche Themenprävalenz pro Dokument ist und welche Alpha-Werte den Themen im Topic Model zugewiesen

wurde. Code 11.40 berechnet die durchschnittliche Themenprävalenz im Textkorpus über die `.mean()`-Funktion. Die `.mean()`-Funktion berechnet dabei den Durchschnittswert für jede Spalte. Jede Spalte enthält Themen, die in den jeweiligen Textauszügen aus den Filmskripten vorkommen.[19]

Der Durchschnittswerte-Output wird an das Objekt mit Namen `topics_df_grouped` übergeben (df macht *pandas*-DataFrame kenntlich). Das Objekt ist eine Datenspalte, die einen Index- und dazugehörige Werte einer einzelnen Variable enthält und an `topics_df_grouped` übergeben wird, indem wir eine neue Spalte mit `model_topics["wahrscheinlichkeit"] =` definieren. Um den Alpha-Wert, d. h. die Topic-Gewichtung, aufzurufen, verwenden wir den `.alpha`-Befehl (ohne Klammern, da keine weiteren Eingaben benötigt werden) und übergeben die Alpha-Werte an eine weitere, neu definierte Spalte mit dem Befehl `model_topics["alpha"] = lda_ergebnis.alpha`. Das Ergebnis in Augenschein nehmen können Sie durch den Export als csv-Datei mit dem `.to_csv()`-Befehl. Anstelle von csv können Sie das DataFrame-Objekt als Datei auch als Excel, JSON, HTML, eine LaTex-Datei, Markdown, SQL-Datei oder in andere Formate exportieren.

Code 11.40 Berechnung und Export der durchschnittlichen Themenprävalenz pro Textausschnitt und des Alpha-Wertes der Themen

```
topics_df_grouped = topics_df.mean()
model_topics["wahrscheinlichkeit"] = topics_df_grouped.values


model_topics["alpha"] = lda_ergebnis.alpha


os.chdir(output + "Woerter_pro_Thema_Gewichtungen\\")
model_topics.to_csv("woerter_modell_mit_" + str(n) +"_Themen.csv", sep="\t")
```

11.4.3 Berechnung der Modellkohärenz

Das englische Wort *perplexity* bedeutet Ratlosigkeit, Verlegenheit und Verworrenheit. *Coherence,* deutsch Kohärenz, bedeutet Zusammenhang und Stimmigkeit. Mit Blick auf unseren Untersuchungsgegenstand konnten wir in der Datenaufbereitung Stimmigkeit bisher vor allem in Form von Wörtern bzw. Tokens und deren latenten Zusammenhang in Themen bzw. Topics in Trai-

19 Wir können aber auch die Durchschnittswerte aller Zeilen berechnen. Hierfür übergeben Sie der `.mean()`-Funktion die Option `axis=1`.

ningssets erkennen. Angesichts der Zerstückelung des Filmskripte-Korpus in Textausschnitte, welche weiter in das *bag-of-words*-Format transformiert wurden, müssen wir uns weiterhin auf statistische Werte in Form von Maßen und Zahlenwerten (Englisch: *scores*) verlassen, um Zusammenhänge in den verworrenen Daten erkennen zu können. Die Berechnung beider Werte hilft uns dabei, ein geeignetes Topic Model für die Interpretation der Ergebnisse auszuwählen.

Wie im sozialen Leben herrscht auch bei der induktiv-quantitativen Inhaltsanalyse zuerst Verworrenheit, welche es stimmig zu machen gilt. Entsprechend berechnen wir das Kohärenzmaß als `u_mass`-Coherence-Scores (Mimno et al. 2011) nach der Ermittlung von Perplexity-Werten (Hoffman et al. 2010). Die Perplexity im Filmskript-Korpus wird mit dem Befehl `lda_ergebnis.log_perplexity()` ermittelt und in der Klammer dem Korpus übergeben (Code 11.41). Der `CoherenceModel()`-Befehl kann abgekürzt als `cm()` in das Python-Skript integriert werden. Für die Berechnung der Kohärenzwerte müssen wir mit dem Befehl `model =` festlegen, dass auf der Datenbasis eines bestimmten Topic Models der Kohärenzwert berechnet wird. Das Topic Model wurde in Code 11.41 als Objekt `lda_ergebnis` definiert, welches die Themenzuordnungen der Wörter und Dokumente (= Themenausschnitte im *bag-of-words*-Format) enthält. Für die Berechnung wird der Korpus mit Namen `corpus` dem `CoherenceModel()`-Befehl mit „=“ übergeben. Das Kohärenzmaß `u_mass`-Coherence muss bei `coherence =` definiert werden.[20] Nun übergeben wir den Perplexity- und Coherence-Score an je eigene Objekte und lassen uns deren Werte mit dem `print()`-Befehl ausgeben, wie in Code 11.41 aufgeführt.

Code 11.41 Berechnung der Perplexity- und Kohärenzwerte des Modells

```
print("Berechne Gütemaße für Modell mit ",n,"Themen"
perplexity = lda_ergebnis.log_perplexity(corpus)
coherence = cm(model=lda_ergebnis, corpus = corpus, coherence='u_mass')

print("Gütemaße für Modell mit ",n,"Themen\n\tperplexity:", perplexity, \
      "\n\tu_mass coherence:", coherence)
```

20 Daneben gibt es noch weitere Kohärenzmaße, die gensim bereitstellt. Dazu zählen die `c_uci`, `c_v` und `c_npmi`-Kohärenz.

Output 11.7 Kohärenzwert und Perplexity-Wert unseres Modells mit fünf Themen

```
Gütemaße für Modell mit 5 Themen
        perplexity: -8.307157805994407
        u_mass coherence: -1.6537382255241464
```

Wie Sie in Output 11.7 sehen, erhalten wir für ein Modell mit fünf Themen eine u_mass-Coherence von –1.6537 und eine Perplexity von –8.3072. Ist das nun gut oder schlecht? Wie sollen wir dieses Modell interpretieren? Aus diesen Werten allein und den limitierten Wörtern pro Thema, die wir oben zu Beispielzwecken ausgegeben haben, lassen sich diese Fragen leider nicht beantworten. Antworten auf die Frage liefern jedoch mehrere, nacheinander automatisiert erzeugte Themenmodelle in Kapitel 11.4.4, deren Werte im Ergebniskapitel 11.5.1 exemplarisch interpretiert werden.

11.4.4 Berechnung einer Vielzahl von Topic Models mit for-Schleife

Analog zur Zusammenfassung der Schritte, die wir für die Datenaufbereitung genutzt haben (Kapitel 11.3.6) werden in diesem Kapitel die zu Lernzwecken kleinteiligen Programmierungsschritte der Kapitel 11.4.1 bis 11.4.3 zusammengefasst. Um uns in Zukunft die Arbeit zu erleichtern (und nicht permanent mit voller Aufmerksamkeit am PC sitzen zu müssen), werden wir eine `for`-Schleife schreiben und die Zuordnung von Themen zu Wörtern und Dokumenten, darüber hinaus auch die Perplexity- und Coherence-Werte für die jeweiligen Modelle berechnen, im Anschluss speichern und zusammenführen. Bevor wir die `for`-Schleife aufrufen, definieren die Themenanzahl der einzelnen Modelle. In unserem Falle handelt es sich um 5 bis 135 Themen, die in einem Abstand je fünf Themen an die Schleife übergeben werden. Hierfür setzen wir einen Iterator, also einen „Zeiger“ (wie bei einer Uhr), der auf die Themenwerte zeigt und alle Befehle nacheinander für exakt diesen Themenwert ausführt. Die Themenzahl legen wir mittels `range(5,136,5)` fest. Die letzte Zahl im `range()`-Befehl gibt die Schritte zu berechnender Topic Models an. Ohne Angabe von Intervallschritten würde Python standardmäßig Topics für jedes Modell in der Range 5, 6, 7, …, 133, 134, 135 erstellen. Die Themenzahl übergeben wir dann einer Liste, deren Werte Schritt für Schritt in der `for`-Schleife abgearbeitet werden.

Listenobjekte für die Topics pro Modell, Topics pro Text, Perplexity-Werte und Coherence-Scores sollten Sie getrennt zwischenspeichern. Topics pro Modell und Topics pro Text ermöglichen die Themenzuordnungen auf einen Blick, und Perplexity-Werte und Coherence-Scores ermöglichen den Vergleich der Modell-

güte. Diese Werte können mit dem `.append()`-Befehl anderen Listenobjekten angehängt werden, wie Sie im unteren Teil in Code 11.42 sehen.

Eine Vielzahl an Topic Models zu berechnen bedeutet eine Vielzahl and Dateien zu generieren. Entsprechend sollten Sie im Verzeichnis einen eigenen Output-Ordner angeben, in dem Gütemaße, Themen x Dokument- und Themen x Wort-Matrix gespeichert werden.

Auch für die Vielzahl an Topic Models müssen wir die Größe des Trainingssets definieren, wie weiter oben in Code 11.33 beschrieben. In der `for`-Schleife soll die Themenanzahl nicht vordefiniert iterieren, was wir mit dem Befehl `for n in anz_topics` angeben (`anz_topics` ist unsere Liste mit den Themenzahlen) und indem wir die Programmcodezeile mit einem Doppelpunkt beenden. Nach dem Doppelpunkt wurde in Code 11.42 zeilenweise geordnet, hintereinander alle von Code 11.33 bis 11.41 eingefügt. Wenn Sie den ganzen Vorgang durch Anwählen und F9, den gesamten Programmcode mit F5 oder den Codeblock durch „Strg + Enter" ausführen, dann kann die Berechnung sehr lange dauern – je nach Größe des Textkorpus und des Lexikons Stunden oder sogar Tage.[21] Nachdem die Schleife durchgelaufen ist, werden die Kohärenz- und Perplexity-Werte gespeichert, indem beide in ein `pandas-DataFrame()`-Objekt überführt und dann mit dem `.to_excel()` oder `.to_csv()`-Befehl im gewählten Format im Output-Ordner abgelegt werden.

Code 11.42 Zusammengeführter Programmcode der Ausführung einer LDA, Zuweisung von Themen zu Worten und Texten und Berechnung von Kohärenzmaßen (Fortsetzung auf nächsten Seiten)

```
topics_per_model = []
topics_per_text = []
perplexities = []
coherences = []

os.chdir(output)
# ============================================================================
# Durchführung des Topic Modelings
# ============================================================================
anz_topics = [x for x in range(5,136,5)]

trainingsset = int(np.round(len(corpus)/5*4,0))
```

21 Zum Beispiel: Für die Aufbereitung und Berechnung aller Topic Models in Wieczorek et al. (2021) benötigte ein extra dafür abgestellter Server für 528 488 Abstracts (z. B. von Zeitschriftenbeiträgen) mehrere Wochen Berechnungszeit.

```
for n in anz_topics:
    print("Berechne topic Model mit",n, "Themen")
    lda_ergebnis = LdaModel(corpus = gewichtung_corpus,
                                    # corpus = corpus,
                                    num_topics=n,
                                    # workers = 3,
                                    iterations=150,
                                    alpha='auto',
                                    eta = 'auto',
                                    chunksize = trainingsset,
                                    id2word=lexikon,
                                    random_state = 42,
                                    per_word_topics=True)

    # ================================================================================
    # Modell speichern
    # ================================================================================
    try:
        os.mkdir(output + "Modelle\\")
    except:
        pass
    os.chdir(output + "Modelle\\")

    lda_ergebnis.save("lda_" +str(n) +"_themen.model")

    # ================================================================================
    # Evaluation des Modells
    # ================================================================================
    print("Übergebe Themen")

    model_topics = lda_ergebnis.print_topics(num_topics = n,
                                    num_words=10)
    model_topics = pd.DataFrame([x[1] for x in model_topics],
columns=["ladungen_tokens"])
    model_topics.index.name = "Themennummer"
    model_topics.to_excel("topic_token" + str(n) + ".xlsx")



    topics_per_model.append( \
        (n, model_topics)
        )
```

```
    # Themen aus den Dokumenten extrahieren
    themen_pro_dokument = [lda_ergebnis.get_document_topics(item) for item in
corpus]
    # topics_per_text.append((n, themen_pro_dokument))

    # Themendatensatz erstellen und mit den korrekten Werten überschreiben
    topics_df = pd.DataFrame(index = range(len(df)),
                             columns = ["topic_" + str(x) for x in range(n)])

    topics_df = topics_df.fillna(0)

    for row in range(len(themen_pro_dokument)):
        print("bearbeite Zeile:", row)
        keys = ["topic_" + str(x[0]) for x in themen_pro_dokument[row]]
        values = [x[1] for x in themen_pro_dokument[row]]
        topics_df.loc[row,keys] = values

    topics_df.to_excel("document_topic_" + str(n) + ".xslx")

    ## Durchschnittliche Wahrscheinlichkeit berechnen,
    ## ein Topic im Korpus anzutreffen
    topics_df_grouped = topics_df.mean()
    model_topics["wahrscheinlichkeit"] = topics_df_grouped.values

    ## Übergeben der Modellgewichtungen
    model_topics["alpha"] = lda_ergebnis.alpha

    try:
        os.mkdir(output + "Woerter_pro_Thema_Gewichtungen\\")
    except:
        pass

    os.chdir(output + "Woerter_pro_Thema_Gewichtungen\\")
    model_topics.to_csv("woerter_modell_mit_" + str(n) +"_Themen.csv",
sep="\t")

    print(topics_df_grouped)


    del(topics_df, topics_df_grouped)

    print("Berechne Gütemaße für Modell mit ",n,"Themen")
```

```
    perplexity = lda_ergebnis.log_perplexity(corpus)
    coherence = cm(model=lda_ergebnis, corpus = corpus, coherence='u_mass').
get_coherence()

    print("Gütemaße für Modell mit ",n,"Themen\n\tperplexity:", perplexity, \
          "\n\tu_mass coherence:", coherence)

    perplexities.append(perplexity)
    coherences.append(coherence)

# ================================================================================
# Datensatz exportieren
# ================================================================================
os.chdir(output)
masszahlen = (perplexities, coherences)
pd.DataFrame(masszahlen, index=["Perplexity", "u_mass Coherence"],
             columns = anz_topics).to_csv("Modellgüte.csv", sep="\t")
```

11.5 Auswertung der Topic Models und Interpretation der Ergebnisse

Nachdem wir nun die technische Seite der LDA, d. h. Vorbereitung und Durchführung, gemeistert haben, können die Ergebnisse ausgewertet und interpretiert werden. Die zu interpretierenden Topic Models wählen wir auf Basis der visualisierten Kohärenz- und Perplexitätsmaße pro Topic Model. Für die in Kapitel 11.5.1 ausgewählten Beispiel-Topic Models werden in Kapitel 11.5.2 die Topwörter und Themenzuordnungen zu den Texten analysiert und diskutiert. Sie werden erkennen, dass Sie ohne Sachwissen über das analysierte Thema weder Themen einordnen noch qualifizierte Vermutungen (≠ Raten) äußern können, was die vom LDA-Algorithmus extrahierten Themen wohl bedeuten mögen. Die Sichtung der Topics basiert auf einem Überblick der Themenstruktur in den entsprechenden Filmen, im Textkorpus sowie der Entwicklung von Topics im Zeitverlauf. Der Überblick wird visualisierend unterstützt durch Heatmaps (Thema x Film), Intertopic Distance Maps (Themen auf zwei Dimensionen reduziert) und Zeitkurven von im Filmskriptkorpus prävalenten Themen. Die Kapitel 11.5.3 bis 11.5.5 adressieren die qualitative Annäherung an die Themen und zeigen auf, welche Fallstricke bei der automatischen Generierung von Topics bestehen bleiben. In Kapitel 11.5.5 wird exemplarisch dargestellt, wie sie mittels *distance reading* (Moretti 2000; 2013) über Texte im Filmskriptkorpus hinweg Topics plausibilisieren können.

11.5.1 Perplexity- und Coherence-Scores: die softwaregesteuerte Maschine hilft beim Lesen, ein interpretierbares Modell wählen wir aus

Mit Code 11.42 generieren Sie eine Vielzahl an Kohärenz- und Perplexity-Werten. Anstatt in Tabellen mit einer Vielzahl an Kohärenz- und Perplexity-Werten zu suchen, kann die Auswahl eines oder mehrerer geeigneter Topic Models für die Interpretation der Themen visualisiert werden. Die Visualisierung beider Werte für die jeweilige Themenzahl bedeutet, eine Abbildung mit einer geteilten X-Achse, aber zwei unterschiedlichen Y-Achsen zu erzeugen (Code 11.43 und Abbildung 11.5).

In Code 11.43 wird Art Leinwand `fig` erstellt, auf der wir mehrere Bilder übereinanderlegen möchten. Das erste dieser Bilder nennen wir `plot1`, über das wir noch ein zweites Bild `plot2` legen möchten. Um auf `fig`, `plot1` und `plot2` zu malen, müssen wir Befehl `subplots()` aus der `matplotlib-pyplot`-library aufrufen. Diese ermöglicht es uns, auf diese Leinwand die Coherence- und Perplexity-Werte wie Pinselstriche zu malen. Mit dem Befehl `figsize =` definieren wir Breite und Höhe der Abbildung in Inches (1 Inch = 2,54 cm). Damit Python die Breite (erster Wert) und Höhe (zweiter Wert) der Abbildung erstellen kann, müssen wir entweder eine Liste oder Tupel mit zwei Elementen programmieren. Im nachfolgenden Codeabschnitt muss das „Zeichnen" der ersten Linie (= Perplexity-Wert) dem `plot1.plot()`-Befehl übergeben werden und die Themenanzahl mit `anz_topics` für die X-Achse definiert werden (aus `for`-Schleife in Code 11.40). Danach werden die in der `for`-Schleife in Code 11.40 berechneten Perplexity-Werte der ersten Y-Achse (links im Plot) übergeben.

Für eine vollständige graphische Darstellung muss mit „Pinselstrichen" in Abbildung 11.6 auch der Y-Achsentitel mit „perplexity" mit dem `label =`-Befehl angegeben werden, welcher auch in der Legende des Plots (= Abbildung) angegeben wird. Die Beschriftung der X-Achse wird mit `plot1.set_xlabel()`, die der ersten Y-Achse mit `plot1.set_ylabel()` und der Abbildungstitel mit `plot1.set_title()` festgelegt.[22]

Die u_mass-Coherence-Werte als zweites Gütemaß wird mit einem zweiten Pinselstrich (= `plot2`) als eigene Y-Achse mit geteilter X-Achse (= `twinx` mit Topic-Anzahl) mit dem Befehl `plot2 = plot1.twinx()` definiert. Sind die Themenzahl der X-Achse und die Kohärenzwerte der Y-Achse definiert, wie

22 Die Schrift für den Titel von Abbildung 11.6 wird mit den Befehlen `fontsize =` und `fontweight =` festgelegt. `fontsize =` definiert die Schriftgröße, und mit `fontweight =` wird vorgegeben, ob die Schrift fettgedruckt, kursiv und/oder unterstrichen werden soll. Soll beispielsweise die Achsenbeschriftung kursiv und fettgedruckt sein, so müssen Sie bei `style = "italic"` und `weight = "bold"` als separate Befehle angeben.

im vorigen Absatz für `plot1`, kann die Legendenbeschriftung mit der `label =` -Befehl eingefügt werden.

Die Visualisierung der u_mass-Coherence-Werte können wir unterstützen, indem die zu pinselnde Linie mit dem Befehl `color = "red"` rot eingefärbt wird. Andere Farben geben Sie mit `color = "blue", color = "green"` usw. vor. Wie bei Plot 1, erfolgt die Beschriftung der Y-Achse am rechten Rand von Abbildung 11.5 mit dem Befehl `plot2.set_ylabel()`. Titel und X-Achsenbezeichnung wurde bereits bei der Erklärung zu Plot 1 festgelegt, und müssen nicht erneut definiert werden, da beide Plots diese Beschriftung teilen.

Um eine Legende für beide geplottete Linien zu erstellen, werden der Leinwand die Kohärenz- und Perplexity-Werte mit dem Befehl `fig.legend()` entnommen. Die vollständige Abbildung wird mit dem `plt.savefig()`-Befehl gespeichert – und, falls Sie mögen, Python mit dem `plt.close()`-Befehl geschlossen. Nun sollte sich eine Abbildung 11.5 analoge Abbildung in dem von Ihnen angegebenen Output-Ordner auf Ihrer Festplatte befinden.

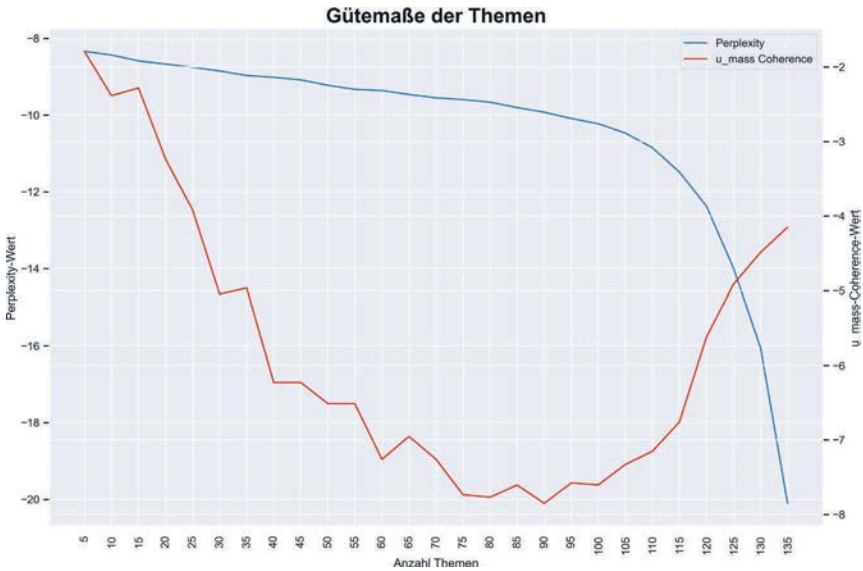
Code 11.43 Programmcodezeilen zur Erzeugung eines Plots mit beiden Gütemaßen für die jeweilige Themenzahl

```
fig, plot1 = plt.subplots(figsize = [12,8])
plot1.plot(anz_topics,perplexities, label="Perplexity")

plot1.set_xlabel("Anzahl Themen")
plot1.set_ylabel("Perplexity-Wert")
plot1.set_title("Gütemaße der Themen", fontsize=20, fontweight="bold")

plot2 = plot1.twinx()
plot2.plot(anz_topics,coherences, label="u_mass Coherence", color="red")
plot2.set_ylabel("u_mass-Coherence-Wert")
fig.legend()
plt.savefig("Modellgüte.png",dpi =300)
plt.close()
```

Bei der Auswahl von Topic Models für die Inhaltsanalyse können Sie grob folgende Daumenregel anwenden.

1. Je kleiner die Themenanzahl, desto höher aggregiert werden die Themen durch Wörter erfasst. Beispielsweise fassen die Topic Models mit einer Anzahl von fünf bis zu zumindest 20 Themen den Korpus mit 626 Filmmanuskripten stark zusammen.
2. Je höher die Themenanzahl, desto differenzierter werden die Themen durch

Abbildung 11.5 Coherence- und Perplexity-Scores für 5 bis 135 Themen



Wörter erfasst. Wählen wir jedoch eine zu hohe Anzahl Themen (z. B. < 135), so müssen wir mit sehr vielen Überschneidungen, d. h. (teilweise) wenig trennscharfen Themen rechnen.

Für die von Ihnen bzw. uns zu treffende Entscheidung erhalten wir in Abbildung 11.5 verschiedene Angaben: Am linken Rand die Perplexity-Werte, am rechten Rand die u_mass-Coherence-Werte und die Anzahl Themen bzw. Topics in Fünfer-Schritten. Idealerweise suchen wir hierfür in Abbildung 11.5 nach einem Topic Model, welches das globale Minimum von Perplexity- und Kohärenzwerten auf sich vereint. Wird kein globales Minimum, d. h. der geringste Wert, ersichtlich, so helfen lokale Minima (z. B. erkennbar als „Zacken" in der roten u_mass-Coherence-Linie) bei der Entscheidung für ein oder mehrere zu analysierende Topic Models. Von dem bzw. den ausgewählten Topic Models müssen die Wortlisten der einzelnen Themen gesichtet und über die Betrachtung von Ausschnitten der Filmskripts plausibilisiert werden. Abbildung 11.5 deutet darauf hin, dass wir ein Modell mit 90 Themen bevorzugen sollten, da hier die Kohärenz den geringsten Wert (= Abweichung vom Nullwert und eine perfekte Thementrennbarkeit) aufweist. Zugleich ist die Perplexität recht gering, was Sie am visualisierten Abstand zwischen Plot 1 und Plot 2 erkennen.[23] Die Perplexity

23 In der Regel nähert sich der Perplexitätswert kontinuierlich der Null an, was sich nach Blei, Ng und Jordan (2003) dadurch erklären lässt, dass der Wert auf Basis einer Genauigkeits-

(Plot 1) deutet weiter darauf hin, dass Topic Models mit noch höherer Themenanzahl Themen noch differenzierter voneinander zu unterscheiden helfen könnten (ja, Konjunktiv). Der Perplexity „widersprechen" jedoch die Kohärenzwerte, welche auf relativ trennscharfe Themen bei Modellen ab mindestens 40 Themen hinweisen. Die von Ihnen/uns vorzunehmende Abwägung sollte daher grob von folgenden drei Fragen geleitet werden.

1. Kann ich gemäß den Gütemaßen die beispielsweise 90 Themen überwiegend sinnvoll interpretieren?
2. Wie viele durch den LDA-Algorithmus entdeckte Themen sind Artefakte?
3. Wie viele Themen von n-Themen sind insgesamt durch uns Forscher*innen interpretierbar?

11.5.2 Sichtung der Themen und Visualisierung über Themen, Texte und Zeitpunkte hinweg

Mit den Gütemaßen als grobe Orientierung müssen wir die Ergebnisse unterschiedlicher Topic Models betrachten, um mit Inhaltswissen und durch sachbezogene Plausibilisierung die zu analysierenden Topic Modele auszuwählen. Geleitet von den drei Fragen bietet die Perplexity-Gütemaß-Linie für qualifiziertes Raten bzw. naturwissenschaftliches *trial-and-error* ein gutes Vorgehen. Für die folgende exemplarische Analyse der Topics haben wir ein vergleichendes Vorgehen gewählt, bei dem wir das 30 Topic Model mit dem 80 Topic Model vergleichen. Zwar hat das Modell mit 90 Themen die „beste" Kohärenz, es erkauft sich diese Kohärenz allerdings mit sehr vielen nicht interpretierbaren Themen. Daher haben wir manuell geprüft, wie viele Themen bei anderen, ähnlich trennscharfen Modellen gut interpretierbar sind. Hier fiel unsere Wahl auf 80 Themen.

Für die induktiv-quantitative Inhaltsanalyse erstellt der Programmcode aus Code 11.42 pro Modell zwei csv-Dateien mit den folgenden, für die Themenanalyse relevanten Inhalten.

1. *Top words* (Topwörter), d. h. die prävalentesten Tokens pro Topic mit Angabe der Wahrscheinlichkeit (Tabelle 11.9),
2. Die fünf häufigsten Filme pro Topic mitsamt erweiterter Stichwortliste (Tabellen 11.10, 11.11 und 11.12).

---

abschätzung eines aus den Daten heraus generierten Trainingssets ergibt. Dieses Verfahren mit mehreren hundert Modellen lässt sich aber in der Regel schlecht durch den Menschen interpretieren.

Um einen ersten Überblick zu erhalten, ist es empfehlenswert, dass wir die Wahrscheinlichkeitsangaben aus der *top word*-Ausgabe löschen (siehe Box 11.6). Die *top words* in Tabelle 11.3 (nächste Seite) geben uns einen sehr groben Überblick zu den maschinell modellierten Topics. Zur Vereinfachung wurden die *top words* bereits aufgrund der Erkenntnisse der Datenbereinigung (siehe Kapitel 11.3) zusammengeführt und händisch ergänzt, um Ihnen die Interpretation zu erleichtern. Topic 19 *auto[+]* kennzeichnet beispielsweise, dass es sich um den Wortstamm von *automobile* oder *automated* handelt. Der erste Eindruck, sofern wir mit der Brille der Fragestellung nach dem gesellschaftlichen Bild von Wissenschaft(ler*innen) in Hollywoodfilmen herangehen, ist die Identifikation der *top words professor* und *doctor* (z. B. schwarze Hervorhebung in Tabelle 11.1). Weiter auffällig sind die verschiedenen *top words* von Tieren und Untieren (z. B. *ant* und *beast,* dunkelgraue Hervorhebung), Höflichkeiten oder Bitten (z. B. 11 Mal *pleas[e]*), hellgraue Hervorhebung) und viele *top words* für gewaltverheißende Gestalten wie *goon* und *thug* und *guard* sowie Gewaltvokabular wie *kill, termin[ate]* und *tortur[e],* was durch *gon[e]* (12 Mal fett hervorgehoben) mit der möglichen Bedeutung als verstorben interpretiert werden kann (alternative Bedeutung: Filmfigur ist gegangen). In diesem Zusammenhang und basierend auf unserem Sachwissen (siehe Kapitel 11.1.2) sind *top words* wie *mama* und *mother* sowie *father* und *grandfath[er]* (auch als Kosename *gramp*) weniger als Hinweis auf friedliche Familienbeschreibungen, denn als Notwendigkeit diese zu schützen zu deuten.

Box 11.6: Weitere Materialien und Informationen online
Auf dem Blog finden Sie unter *https://sozmethode.hypotheses.org/category/topic-modeling* eine Anleitung wie Sie in neun Schritten die Tabellen in Excel anpassen und in Word einfügen können.

Weitere Hilfe beim Verstehen und einen Überblick zu gewinnen, bieten die Visualisierungen von Topic Models als Intertopic Distance Map (Abbildungen 11.7 und 11.8), deren Ordnung für die Gruppierung in Tabelle 11.3 ausschlaggebend ist, und die sogenannten Heatmaps (Abbildung 11.6). Diese erfüllen auch den Zweck der Ergebnisdarstellung. Beide zeigen die Themenstruktur des Korpus an, wobei die Heatmap die Themenverteilung pro Filmskript zusammenfasst. Die Intertopic Distance Map hilft die mögliche Bedeutung der tokenisierten Wörter für und über Topics hinweg in einem Modell zu analysieren. Für die Ergebnisdarstellung der Themen wird als dritte Visualisierung die Entwicklung der Themen über Zeit vorgestellt (Abbildung 11.9).

11.5.2.1 Heatmap erstellen und interpretieren

Wenden wir uns nun der Visualisierung der Themenstrukturen über Filmgrenzen hinweg zu. Hierfür verwenden wir eine Heatmap. In der Heatmap gilt, dass je heller (= größere Hitze) die Farben sind, desto prävalenter (= von höherer

Tabelle 11.3 Übersicht der 30 Topics mit den je zehn häufigsten Worten (*top words;* Quadranteneinteilung nach Abbildung 11.8; o = oben, u = unten, li = links, re = rechts) (Fortsetzung nächste Seite)

Quadrant	Topic	Top words									
A (o li)	0	manicur[e]	mumm[y]	towni[e]	church	goon	love	bless	doctor	beast	west
A	1	**gon[e]**	professor	cantrel	love	buddy	dog	talk[er]	student	ditch	brain
A	2	rogu[e]	storm	ice	presid[ent]	guard	world	editor	agent	professor	offic[e/r]
A	3	raptor	**gon[e]**	hudson	professor	okay	wraith	pleas[e]	love	world	offic[e/r]
A	4	curat[or]	forev[er]	ant	tep[al]	question-nair[e]	doctor	data	ballroom	pleas[e]	cleaver
A	6	raptor	basket	**gon[e]**	spain	choir	knapsack	chairman	pleas[e]	pathway	mother
A	7	casket	rat	judg[e]	subway	juri[dic/al]	dictat[or]	**gon[e]**	nurs[e]	pleas[e]	ever
A	9	travel	presid[ent]	machin[e]	fire	monster	rock	gun	creatur[e]	tunnel	ground
A	10	frock	mutant	street	cop	termin[ate]	**gon[e]**	men	love	music	storm
B (u li)	12	**gon[e]**	world	presid[ent]	okay	kelp	pleas[e]	kill	coal[ition]	god	scientist
B	5	coach	superstit[ion]	book	pleas[e]	**gon[e]**	tiger	friend	glass	mother	agent
B	8	copper	villag[e]	grunt	music	child	kill	scream	dog	water	okay
D (u re)	11	bee	mama	storeroom	love	corn	music	field	rabbit	captain	offic[e/r]
D	14	rat	money	thug	turtl[e]	anybody	beat	pleas[e]	cat	**gon[e]**	mother
D	17	humve[e]	coal[iton]	pharaoh	seaplan[e]	lighthous	mutant	punt	pleas[e]	west	rogu[e]
D	18	**gon[e]**	drainag[e]	presid[ent]	pilot	father	pleas[e]	chip	helm[et]	okay	fusion

Quadrant	Topic	Top words									
C (o re)	23	creed	prank	toot	spyglass	barber	chairman	skinhead	snip[er]	oakland	plant
C	13	manhattan	tortur[e]	love	captain	doctor	pleas[e]	[all-]told[-off]	mind	**gon[e]**	judg[e]
C	15	watch[-]it	rogu[e]	owl	binocular	stare	ridg[e]	rat	theory	salmon	waiter
C	16	gulli[ble	goate[e]	rope	plant	auto[+]	disk	sentry	captain	cabl[e]	command[er]
C	19	coal[ition]	moloch	west	presid[ent]	cop	senat	throne	offic[e/r]	seldom	son
C	20	captain	amen	camel	beast	congressman	**gon[e]**	hatch	mayb[e]	vice	fireman
C	21	street	machin[e]	music	dog	travel	beat	love	paus[e]	world	gun
C	22	agent	tornado	termin[ate]	travel	bed	radio	dog	rabbit	street	stare
C	24	marin[e]	cart	termin[ate]	bot	coach	rocket	dog	conductor	music	fall
C	25	stallion	artwork	api	bulldoz[er]	sea	command[er]	island	condens[ation]	effect	shrimp
C	26	machin[e]	offic[e/r]	hotel	street	control	travel	corridor	build	floor	director
C	27	dog	nightcrawl[er]	agent	raven	**gon[e]**	fingerprint	storm	seamstress	kill	captain
C	28	magenta	shredder	clan	presid[ent]	scientist	growl	cello	fire	machin[e]	foot
C	29	pussycat	bailiff	doctor	grandfath[er]	friend	love	renegad[e]	embryo	pleas[e]	mean

Bedeutung) ist das jeweilige Thema in einem Film. Häufige Wörter von heißen Themen in Filmskripten sind beispielsweise „professor", „buddy", „ditch", „brain", „student". Dabei arrangieren wir eine Heatmap so, dass in jeder Zeile ein Film, in jeder Spalte die Themenabdeckung steht. Senkrechte, helle Linien zeigen, dass Themen über Filme hinweg prävalent sind, d. h. stark vertreten sind. Wenn Sie Abbildung 11.6 betrachten, dann sehen sie pro Film immer mehrere Linien in einer Zeile. Das spiegelt den Themenmix des Filmes wider, der in unserem Falle irgendwo zwischen fünf und sieben Themen pro Film darstellt. Die gleiche Farbgebung zeigt, dass filmübergreifend die Themen 1, 9 und 21, und zum Teil auch 3, 4, 16 und 26 gehäuft vorkommen. Die anderen Themen scheinen eher randständig zu sein und selten vorzukommen.

Oberflächlich betrachtet präsentieren die Themen 1, 9 und 21 Wissenschaft über (mutige bis verrückte) Professoren,[24] die mit dem (reisenden) Präsidenten kooperieren, um Monster zu bekämpfen (Thema 9). Die Szenenbeschreibungen (womöglich Thema 21) werden durch Tiernamen (dog, cat, turtle, Thema 16), Verrat und Gangster (thug, rat) und teilweise auch Querverweisen auf das Militär (officer, captain) oder Orte (hotel, Thema 26) ergänzt, was allerdings inhaltlich noch weiter auszuarbeiten ist.

Die Programmierung der Heatmap in Abbildung 11.6 wird in Code 11.44 abgebildet. Die Heatmap basiert auf einem Listenabgleich von Themen pro Textausschnitt des 30er Topic Models und deren Mittelwerte pro Film. Wie bereits in früheren Aufbereitungs- und Analyseschritten nutzen wir eine `for`-Schleife, um eine Liste mit Werten (in unserem Falle Themenzuordnung pro Text) aufzufüllen. Dieser leeren Liste geben wir den Namen `text_topics = []`, der nacheinander die Themen pro Text übergeben werden (Code 11.44). Die Themen werden mit der `.get_document_topics()`-Funktion aus dem Korpus extrahiert und mittels der `.append()`-Funktion der Liste hinzugefügt. Auf diese Weise erhalten wir eine Liste, die aus Tupeln besteht. Jede Tupel enthält wiederum eine Kombination aus Themennummer und Themenprävalenz der Filmskripte, die in `corpus` enthalten sind.

Die `text_topics`-Liste ist die Basis für einen Dokument x Topic-Datensatz. Hierfür müssen wir die Themen aus den Tupeln in separate *pandas*-DataFrame-Objekte (je eines pro Datenzeile) überführen. Tabelle 11.4 verdeutlicht beispielhaft die Struktur, in der die mit dem `.get_document_topics()`-Befehl extrahierten Themen mit dem `DataFrame()`-Befehl in ein *pandas*-Datensatzobjekt übersetzt werden.

24 Ja, die Protagonisten in Hollywood-Wissenschaftsfilmen sind überwiegend bis ausschließlich männlich.

Abbildung 11.6 Heatmap der Verteilung der 30 Themen auf die im Topic Model analysierten Filme

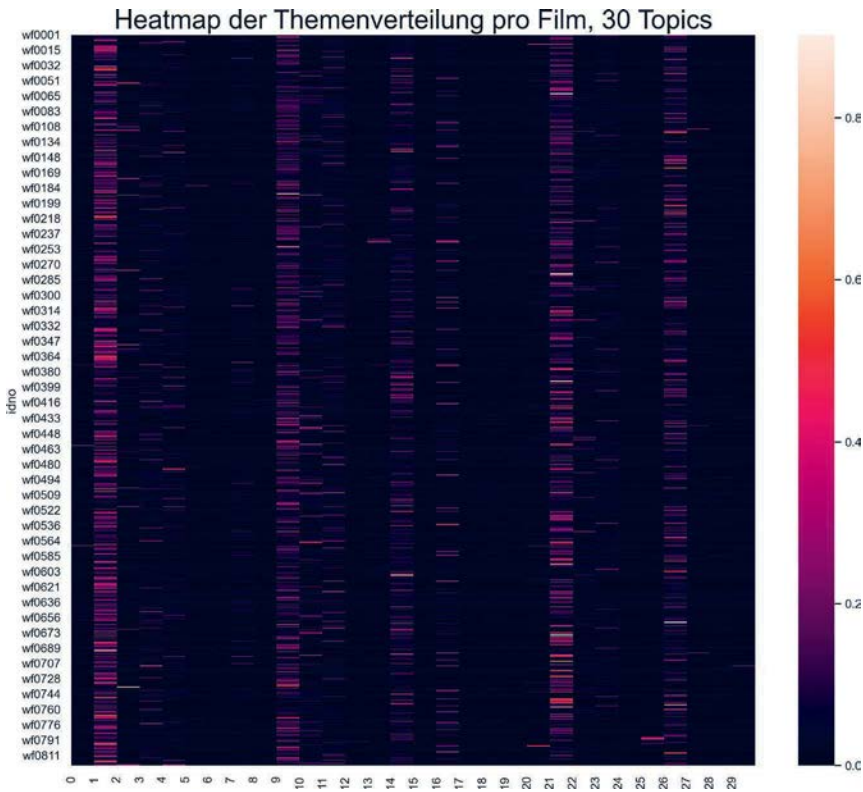


Tabelle 11.4 Beispielhaftes Aussehen eines Datensatzes, der aus der Zuordnung von Themen und deren Prävalenzen bei einem Ausschnitt aus den Filmskripten generiert wurde

	0	1
0	1	0.3
1	4	0.1
2	10	0.6

In Tabelle 11.4 befinden sich die Prävalenzen in den Spalten statt in den Zeilen und die Themennummern stellen die Zeilen- statt die Spaltennamen dar. Der fertige Datensatz muss auch in den Zeilen eine fortlaufende Identifikationsnummer aufweisen, damit wir im Datensatz die Filmnummern pro Zeile und das Erscheinungsjahr ergänzen können, um sowohl die Heatmap mit den Themenverteilungen pro Film als auch die Themenprävalenz pro Jahr zu berechnen. Wie in Tabelle 11.5 dargestellt, müssen wir daher die Struktur von Tabelle 11.4 kippen, um die Themennummern und Themenprävalenzen in die Spalten zu verlegen. Kippen oder neudeutsch transponieren (Englisch: *transpose*) erfolgt mit dem `.T`-Befehl. In einem weiteren Transponierschritt müssen die Themennummern aus der ersten Zeile entfernt und, wie in Tabelle 11.5 dargestellt, die Themenprävalenz in einer Datenzeile (= einem Text) dargestellt werden. Da wir die die Texte nacheinander, d. h. Reihe für Reihe, ansteuern werden und den Iterator hierfür `row` genannt haben, haben wir die Zeile in Tabelle 11.6 auch row genannt, um diese Verknüpfung zu verdeutlichen.

Tabelle 11.5 Datensatz (Text und Thema) nach Transponierung der Zeilen und Spalten

	0	1	2
0	1	4	10
1	0.3	0.1	0.6

Tabelle 11.6 Datensatz (ein Text und Themen) nach der der weiteren Ausbesserung

	1	4	10
row	0.3	0.1	0.6

Die in Tabellen 11.4 bis 11.6 dargestellten Datenmanipulationen sind in Code 11.44 wie folgt programmiert. Die Zuordnung von Themen und Themenwerten erfolgt im Zweischritt, d. h. Sie behalten aus der ersten Zeile den Spaltennamen und löschen danach die Zeile. Technisch übernimmt der Befehl `.columns = [int(x) for x in [TEILDATENSATZ].iloc[0]]` die Werte aus der ersten Zeile als ganze Zahl, als sogenanntes Integer. Die erste Zeile wird mit dem `.drop(index = 0)`-Befehl gelöscht. In der Klammer definiert `index = 0` einen ganz spezifischen Teil des Datensatzes, hier die zu löschende erste Zeile. Den

neuen Zeilennamen definieren wir mit dem `.rename()`-Befehl. In der Klammer wird wieder mit dem Befehl `index =` definiert, welches Element im *pandas*-DataFrame-Objekt umbenannt werden soll. Statt `rows =` könnten Sie mit `columns =` Spaltennamen umbenennen.

In Code 11.44 wurde eine `for`-Schleife programmiert, da alle Themenzuordnungen pro Textausschnitt in solche Teildatensätze transponiert werden sollen, welche in der Liste mit Namen `document_topic_matrix` wieder zusammengefügt werden. Das Iterieren gemäß der „Anzahl" der Elemente in der `texttopics`-Liste mit der `range(len())`-Befehlsverschachtelung erlaubt uns, direkt die Zeilenwerte an die Schleife zu übergeben. Als Zeilennummer definiert, ermöglicht uns der Datensatz eine Zusammenführung von Themen mit den Metadaten (Filmtitel und Jahr der Veröffentlichung) mit dem `.append()`-Befehl. Ist die `for`-Schleife durchgelaufen, so fügen wir die Liste mit dem `concat()`-Befehl zu einem einheitlichen Datensatz zusammen, was in Fachsprache „konkarieren" für Zusammenfügung einer Zeichen- bzw. Informationskette heißt. Nichtfachsprachlich ausgedrückt wurde im Ergebnis das `document_topic_matrix`-Listenobjekt (bestehend aus einzelnen Datensätzen) durch einen einheitlichen Datensatz ersetzt.

Der vereinheitlichte Datensatz enthält jedoch viele leere Zellen, wie in Beispiel-Tabelle 11.7 dargestellt. Die leeren Zellen sind durch den `concat()`-Befehl entstanden, der zwar Themenwerte als Spaltennamen und die Themenprävalenzen an die richtigen Stellen im Datensatz übergeben hat; jedoch enthalten im Datensatz mit 30 Themenzeilen (mit enthaltenen Werten!) nicht alle Zellen Werte (= schwarze Flächen für keine Hitze in Abbildung 11.6), beispielsweise bei den seltenen Themen 6 und 29. Die Zeilen mit N und N – 1 zeigen dabei die letzten beiden Zeilen eines potenziellen Datensatzes an. Das N wird hierbei verwendet, da es in der Statistik ein Platzhalter für einen Wert (z. B. eine Obergrenze) darstellt.

Tabelle 11.7 Struktur des zusammengeführten Beispieldatensatzes mit 30 Themen auf Basis von Code 11.42

	0	1	2	3	4	5	6	7	8	9	…	29
0		0.3		0.1						0.6	…	
1	0.9		0.1								…	
2					0.25	0.25	0.1	0.05	0.1		…	0.15
…	…	…	…	…	…	…	…	…	…	…	…	…
N – 1		0.9								0.1	…	
N									0.2		…	0.8

Für die Berechnung von Durchschnittswerten und die Erstellung einer Heatmap benötigt Python jedoch Werte. Die leeren Zellen steuern wir im Datensatz an und definieren mit dem `.fillna()`-Befehl durch Eintrag einer Null in der Klammer, dass alle fehlenden Werte durch 0 ersetzt werden. Statt einer Null können theoretisch andere Zeichenketten in die Klammern eingetragen werden, jedoch ohne das Ergebnis zu verfälschen. Alternative Einträge wären 99, „fehlender Wert" oder „NA", welche beim nächsten Öffnen des Datensatzes (z. B. über Doppelklick in Ihrem „Variable-Explorer"-Fenster oben rechts in Python) im Datensatz als Themenwerte „-99", „NA" oder „fehlender Wert" in den jeweiligen Zellen erscheinen.

Der mit Code 11.42 erstellte Datensatz enthält bisher die Merkmale Themen pro Filmabschnitt. Diese Merkmale müssen für weitere Auswertungen mit den Informationen Filmname oder dessen Identifikationsnummer und Erscheinungsjahr verbunden werden. In den Programmierungen haben wir darauf geachtet, dass die Reihenfolge und die Zeilenzahl des mit Code 11.44 erstellten Datensatzes identisch mit dem Datensatz der LDA ist (siehe Kapitel 11.4.2.2). Anschließend wird mit dem Befehl `document_topic_matrix.groupby(by="idno").mean()` ein nach Identifikationsnummer gruppierter Datensatz, in dem die Prävalenzen auf die Filme gemittelt sind, erstellt. Der `groupby()`-Befehl gruppiert die Werte der Variable `idno`, und die Durchschnittswerte werden mit dem Befehl `.mean()` berechnet. Das Ergebnis dieses Vorgangs ist eine Liste, die wir dann an eine Variable mit dem Namen `films` übergeben.

In der anschließenden Zeile erstellt das Objekt `films_ohne_jahre`, um die Datenspalten mit Information Themenprävalenzen (ohne Jahre) auszuwählen zu können. Der Aufruf der Liste mit Namen `films` definiert die spezifische Auswahl von Daten aus dem entsprechenden Objekt mithilfe des Listenabgleichs `[x for x in films.keys() if type(x) == int]`.

Das Erstellen einer Heatmap basiert auf dem Prinzip, Leinwand (= `figure()`-Befehl der `matplotlib.pyplot`-Bibliothek) mit gemittelten Themenwerten pro Film als Farben (= `heatmap()`-Befehl aus `seaborn`-Paket ausführen) zusammenzubringen. Zum „Bauen" der Heatmap in Abbildung 11.6 werden folgende Befehle verwendet.

- `title()` definiert die Überschrift;
- `xticks()`verändert Striche auf der X-Achse, wobei
  - `ticks` = die Anzahl und Abstände der einzelnen Striche auf der X-Achse,
  - `labels` = die Beschriftung und
  - `rotation` = den Winkel, in dem die Beschriftung der X-Achse angebracht wird, definiert;
- `tight_layout()` passt die eingefärbte Grafik der Leinwand an;
- `savefig()` speichert die Heatmap;
- `close()` schließt das „Bauen" die Leinwand ab.

Code 11.44 Code zur Generierung der Heatmap mit dreißig Themen pro Filmskript (Fortsetzung nächste Seite)

```
## Output-Pfad festlegen
os.chdir(output + "Visualisierungen\\")

text_topics = []
for text in corpus:
    text_topics.append(ergebnis.get_document_topics(text))

document_topic_matrix = []

for row in range(len(text_topics)):
    d = pd.DataFrame(text_topics[row]).T
    d.columns = [int(x) for x in d.iloc[0]]
    d = d.drop(index=0)
    d = d.rename(index = {d.index[0] : row})
    document_topic_matrix.append(d)

document_topic_matrix = pd.concat(document_topic_matrix)

document_topic_matrix = document_topic_matrix.\
    reindex(sorted(document_topic_matrix.columns), axis=1)

document_topic_matrix = document_topic_matrix.fillna(0)

## Merkmale ergänzen
document_topic_matrix["idno"] = df.idno

## Filme gruppieren und die durchschnittliche Themenprävalenz pro Film
ermitteln
films = document_topic_matrix.groupby(by="idno").mean()

## Themen selektieren
films = films[[x for x in films.keys() if type(x) == int]]

plt.figure(figsize=[12,12])
sbs.heatmap(films)
plt.title("Heatmap der Themenverteilung pro Film, 30 Topics", size=24)
plt.xticks(ticks=films.keys(),
           labels= films.keys(),
           rotation=90)
```

```
plt.tight_layout()
plt.savefig("heatmap_30.png", dpi = 300)
plt.close()
```

11.5.2.2 Intertopic Distance Map erstellen und interpretieren

Ergänzend zu den visuellen Informationen aus der Heatmap erhalten wir mit einer Intertopic Distance Map Informationen darüber, 1) welche Themen mit höherer Wahrscheinlichkeit gemeinsam auftreten, 2) welches die Wörter sind, die am häufigsten bei diesen Themen auftreten, und 3) welche Wörter (fast) ausschließlich bei den jeweiligen Themen vorliegen (siehe auch Box 11.7). Mathematische Grundlage der Intertopic Distanzberechnung ist die multidimensionale Skalierung, die Ähnlichkeiten als gemeinsames Auftreten von Wörtern und Themen in Texten in Distanzen umgerechnet. Die Distanzen werden dann auf eine zweidimensionale Fläche projiziert, die in Abbildung 11.7 auf der linken Seite zu sehen ist. Die Themenverteilung bietet Hinweise auf potenziell vorliegende latente Dimensionen.[25]

Box 11.7: Weitere Materialien und Informationen online
Die interaktive Intertopic Distance Map können Sie unter *https://sozmethode.hypotheses.org/methodenbuch* betrachten.

Die Programmierung der interaktiven Intertopic Distance Map enthält Programmcode 11.45. Die Intertopic Distance Map wird als interaktive HTML-Datei wie eine Webseite programmiert, die mit den Ergebnissen des 30er Topic Models gefüllt wird. Grundlage ist eine Art leerer „Notizblock". Der Notizblock wird mit dem `enable_notebook()`-Befehl (`pyLDAvis`-Paket in Python) angelegt. Im Notizblock wird mit dem `gensim_models.prepare()`-Befehl eine interaktive Grafik mit dem Namen `vis` generiert.[26] Der `gensim_models.prepare()`-Befehl benötigt für die Grafikerstellung das Ergebnis unserer LDA (`ergebnis`), den Textkorpus (`corpus`) und das `dictionary` mit der Zuordnung von Token zu

25 Alternativ können Sie auch eine Hauptkomponentenanalyse (auch: Faktorenanalyse) oder Clusteranalyse durchführen, um die Themen auf unterliegende Dimensionen zurückzuführen, die eventuell auf einer abstrakteren Ebene interpretiert werden können und Schlüsse über die Themenverteilung, wie sie in Ihren Daten vorliegt, erlauben.

26 Dieser Befehl hat eine weitere, hier nicht aufgeführte Option namens `mds =` . Hier können Sie angeben, auf welcher Basis Sie die Distanzen der Intertopic Distance Map berechnen möchten. Sie können `pcoa`, `mmds` und `tsne` angeben. `pcoa` ist die reguläre Berechnung einer multidimensionalen Skalierung. `mmds` nimmt an, dass alle Variablen metrisch skaliert sind. `tsne` verwendet den *distributed stochastic Neighbor-embedding-Algorithmus.* Dabei sind die mathematischen Grundlagen je verschieden. Deren Herleitung würde den gegebenen Rahmen übersteigen und wäre nur durch die Verwendung sehr vieler Formeln möglich.

Wort-ID (`trained_model` aus Kapitel 11.3.8). Den HTML-Notizblock öffnen wir mit dem `display()`-Befehl, um das `vis`-Objekt zu übergeben, und speichern das Ergebnis mit dem `save_html()`-Befehl, der zusätzlich zum `vis`-Objekt den Dateispeicherpfad und Dateinamen enthält.

Code 11.45 Erstellen und Speichern einer interaktiven Intertopic Distance Map

```
pyLDAvis.enable_notebook()
vis = pyLDAvis.gensim_models.prepare(ergebnis, corpus,
                                     dictionary=trained_model.id2word)

pyLDAvis.display(vis)

pyLDAvis.save_html(vis, "interaktive_Grafik_" + str(30) + "_Themen.html")
```

Die gespeicherte interaktive HTML-Datei muss im Browser (z. B. Firefox) geöffnet werden. Wie in Abbildung 11.8 sehen Sie in der Intertopic Distance Map eine „Landkarte" (= map), welche die Topics nach Prävalenz absteigend anzeigt. In Abbildung 11.8 werden die Wörter, auch als *(top) terms* bezeichnet, des ausgewählten Themas dargestellt (Topic 1).

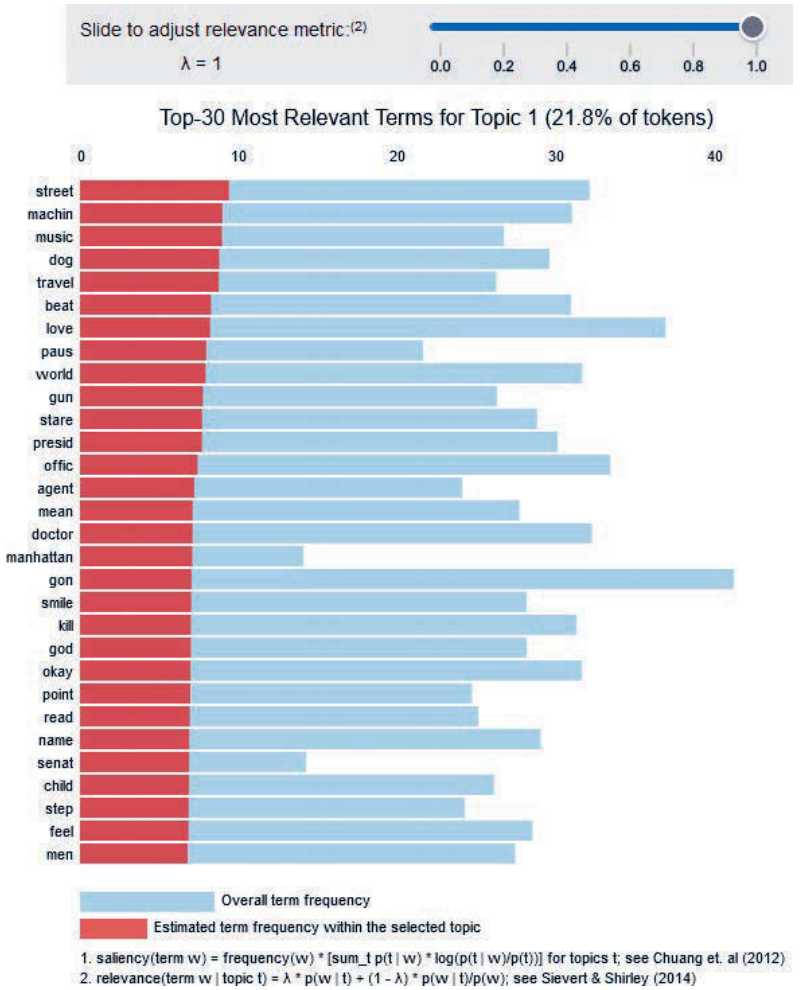
In Abbildung 11.7 werden in einem Koordinatensystem die 30 Topics des Modells gemäß Distanz zwischen den 30 Topics als Landkarte angezeigt. Die Größe eines Kreises bildet ab, wie viele Tokens im Datensatz, gemessen an der Gesamtzahl der im Korpus befindlichen Tokens, durch das jeweilige Thema erfasst werden. Sie können das Thema wechseln und die Tokens für andere Themen betrachten, indem Sie entweder mit der Maus auf einen der Kreise klicken oder in der Menüleiste über der Intertopic Distance Map eine Themenzahl zwischen 0 und 29 eingeben. Um zum Topic mit der nächsthöheren Prävalenz zu gelangen (= erkennbar an den niedrigen Topic-Nummern), können Sie alternativ auf die Schaltfläche `previous topic` klicken oder, um zum nächsten, weniger prävalenten Topic zu springen, auf `next topic` klicken. Mit `clear topic` löschen Sie Ihre Auswahl.

Die Kreisfarbe des ausgewählten Themas, in Abbildung 11.7 Topic 1, wechselt von blau auf rot (Standardeinstellung), wenn Sie diesen Kreis durch Klicken Ihrer Maustaste anwählen. Für Topic 1 werden die 30 *most relevant terms* angezeigt. Im Balkendiagramm in Abbildung 11.8 wird die Relevanz der am stärksten mit Thema 1 assoziierten 30 Wörter (roter Balkenteil) und die Häufigkeit (blauer Balkenteil) angezeigt, mit der der Begriff im gesamten Korpus vorkommt. Über dem Balkendiagramm sehen Sie einen dunkelblauen Schieberegel, mit dem Sie einstellen können, wie spezifisch die angezeigten Tokens sein sollen, d. h. wie ausschließlich diese in einem Thema vorkommen sollen. Ein mit dem griechischen

Abbildung 11.7 Interaktive Intertopic Distance Map: Verortung der Themen auf zwei Themendimensionen

Lambda-Zeichen (λ) gekennzeichneter Wert von 1 zeigt die Tokens an, die am häufigsten im jeweiligen Thema, hier Topic 1, auftreten. Der Wert von 0 weist Tokens aus, die ausschließlich beim ausgewählten Topic 1 vorkommen. Wir haben in Abbildung 11.8 den Regler auf λ = 1 gesetzt, sodass die häufigsten vorkommenden Wörter angezeigt werden. In der Überschrift wird der Anteil mit dem Hinweis kenntlich gemacht, dass Topic 1 45,8 % aller Tokens aus dem `dictionary` abdeckt. Beispielsweise ist das Token „street“ (erster Balken im Diagramm) knapp 2 800-mal im Korpus enthalten, hiervon entfallen circa 1400 Nennungen auf das erste Thema.

Abbildung 11.8 Interaktive Intertopic Distance Map: Übersicht über die Worte, die am stärksten auf die Themen laden und Schieberegler zur Einstellung der Spezifität der vorkommenden Worte



Im Browser können Sie mit der Maus ein Token anklicken oder den Mauspfeil darüber halten, wenn Sie einen visuellen Eindruck für die Häufigkeit von „street" oder einem anderen Token in den 30 Themen erhalten wollen. In der interaktiven Intertopic Distance Map (linke Bildhälfte) werden dadurch Themenkreise vergrößert, in denen das Wort häufig auftritt. Zugleich verkleinern sich die Themenkreise, in denen das Wort selten vorkommt.

Für die Interpretation sollten Sie systematisch zuerst alle Themen nacheinander betrachten, und daraufhin die häufigsten themenspezifischen Worte

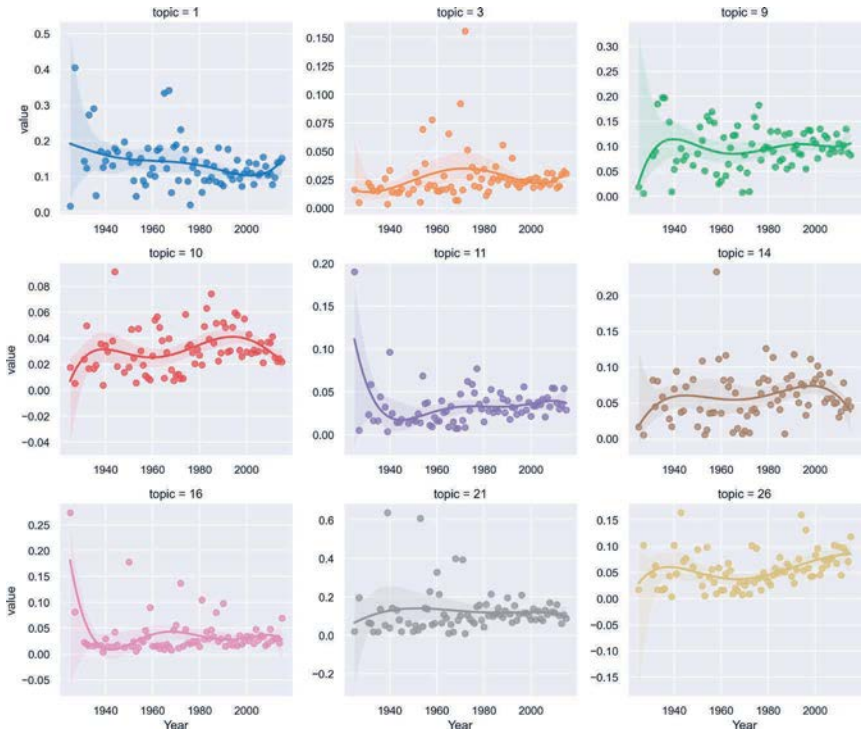
analysieren. Dieselbe Erkenntisstrategie von Token-Bedeutung für den Korpus gefolgt von Token-Bedeutung für das Thema können Sie auch für die immer spezifischer werdende Analyse von Worten für ein Thema verfolgen. Entsprechend verschieben Sie den λ-Schieberegler Schritt für Schritt zu einem niedrigeren Wert (z. B. 0.8, 0.5, 0.3 und dann 0).

11.5.2.3 Themenvorkommen im Zeitverlauf

In Kapitel 11.5.2 wurde einleitend die Doppelfunktion von Intertopic Distance Map und Heatmap mit Ergebnisdarstellung und mit dem Gewinnen eines Gefühls für bzw. eines visuellen Überblicks über die Daten beschrieben. Der visuell unterstützte Einblick in die Themenverteilung pro Filmskript mit der Heatmap (Abbildung 11.6) und die in der Intertopic Distance Map (Abbildungen 11.7 und 11.8) abgebildete statistische Bedeutung der tokenisierten Wörter für und über Topics hinweg zeigt Ihnen, wie aufwendig Datenanalyse ist. Wir erwähnen die Datenanalyse an dieser Stelle, um Sie daran zu erinnern, dass Sie den erheblichen Lernaufwand des Programmierens für die Unterstützung der quantitativ-induktiven Inhaltsanalyse vielschichtiger Textdaten machen, wie z. B. einem Korpus an Filmskripten. Wie Datenanalyse und Ergebnisdarstellungen Hand in Hand gehen, wird (hoffentlich) auch bei der nun folgenden Untersuchung der Entwicklung von Themen über Zeit gut erkennbar. Als dritte hier vorgestellte Möglichkeit, die mit dem 30er Topic Model generierten Themen zu erschließen, können über die zeitliche Struktur Themenkonjunkturen greifbar gemacht werden.

In Abbildung 11.9 werden Themenkonjunkturen von überdurchschnittlich häufig im Datensatz auftretenden Beispielthemen dargestellt. In Abbildung 11.9 wird im Beispiel oben links bei Thema 1 (Tokens professor, gene, brain, student, talk usw.) ersichtlich, dass Topic 1 in vor 1940 erschienenen Filmen stärker vertreten war, und in den 2000er Jahre-Filmen tendenziell dessen Relevanz (= Prävalenz) abgenommen hat. Ergänzend zur Betrachtung des gesamten Zeitraumes erkennen wir, dass nach 2009 der Anteil des Themas 1 in den Filmskripten pro Jahr wieder zugenommen hat, was wir als Relevanzzuwachs von Thema 1 in unseren Daten interpretieren können. Wir sehen zudem Ausreißer zwischen 1960 und 1970, was bedeutet, dass zumindest in zwei Jahren dieses Jahrzehnts dieses Thema besonders häufig im Korpus der Filmskripte enthalten ist. Wir können diese Information nun auf zweierlei Weise nutzen. Erstens, indem wir auf Basis der Hochpunkte und Ausreißer Texte aus den entsprechenden Jahren auswählen und in einer inhaltlichen Analyse sichten. Hierdurch können wir ermitteln, was der Grund für die Ausreißer ist oder inwiefern diese Themen anders kontextualisiert werden. Das würde uns erlauben, den Sinngehalt der Texte und Themen besser zu verstehen. Oder wir können diese Abbildung im Anschluss an die Inhaltsanalysen nutzen, um Thementrends zu erläutern und Gründe zu

Abbildung 11.9 Visualisierung der Prävalenz der überdurchschnittlich häufig auftretenden Themen im Zeitverlauf



finden, warum die Themen an Attraktivität gewonnen oder verloren haben. Unerlässlich für die inhaltsanalytische Interpretation der Entwicklungen sind Kontextinformationen und Sachwissen. Dazu mehr ab Kapitel 11.5.3.

Die Programmierung von Abbildung 11.9 ist in Code 11.46 zusammengefasst. In Code 11.46 verwenden wir den *pandas*-DataFrame `document_topic_matrix` aus Code 11.45. Mit `document_topic_matrix["Year"] = df.Year` übergeben wir das Erscheinungsjahr der Filme als eigene Variable. In der anschließenden Zeile erstellt das Objekt `films_by year`, um die Datenspalten mit Information und Themenprävalenzen (mit Jahren) auszuwählen zu können. Der Befehl `films[]` definiert die spezifische Auswahl von Daten aus dem entsprechenden Objekt mithilfe des Listenabgleichs `[x for x in films.keys() if type(x) == int]`.

Nun müssen die Themen nach dem Erscheinungsjahr der Filme gruppiert und die durchschnittliche Themenprävalenz pro Jahr berechnet werden. Mithilfe der Funktionen `groupby(by = "Year").mean()` können wir einen Datensatz gruppieren und dann die Durchschnittswerte dieser Gruppen berechnen.

Das `by = "Year"` zeigt dabei an, welche Variable innerhalb des Datensatzes für die Gruppierung verwendet werden soll. Wenn Sie an dieser Stelle eine Liste angeben, dann wird die Gruppierung über mehrere Variablen hinweg durchgeführt, beispielsweise `by = ["Year","Title"]` würde bewirken, dass nach Jahren und Titel gruppiert werden würde – was natürlich voraussetzt, dass es mehrere Filme gleichen Titels gibt, die in unserem Datensatz vorliegen. Die Werte fügen wir dem Objekt `films_by_year` hinzu, indem wir sie mit einem = übergeben.

Die `groupby()`-Funktion speichert die Erscheinungsjahre als Index-Wert statt als eigene Variable. Die Variable erstellen wir mit dem `reset_index()`-Befehl. Umwandeln müssen wir auch die Themen vom numerischen Integer (= ganzen Zahlen) als für die Visualisierung notwendige Zeichenkette (String). Dazu konvertieren wir den Typus der Spaltennamen mithilfe des Listenabgleichs `[str(x) for x in films_by_year.columns]`, und übergeben die Zeichenkette zurück an die Spalten mit `films_by_year.columns =`.

In Abbildung 11.9 sind nur überdurchschnittlich häufig im Filmkorpus vorkommende Topics abgebildet. Den Wert für überdurchschnittlich häufig errechnen wir als Durchschnittswert pro Thema über alle Jahre hinweg mit dem `.mean()`-Befehl. Das Ergebnis besagt, dass jedes Thema mindestens in 3,3 % der Filmausschnitte vorkommen sollte, was als 1/30 für Python als Wert definiert werden muss. Die Auswahl der häufigsten Topic-Werte aus dem Datensatz wird durch eckige Klammern `[]` angeben. In den eckigen Klammern wird der `mean()`-Befehl um das Größerzeichen > und Schwellenwert 1/30 ergänzt. Der außerhalb der eckigen Klammern anschließende `.index`-Befehl spezifiziert, dass die Indexwerte des Datensatzes (= Themennummern) beibehalten werden. Um die Auswahl der überdurchschnittlich häufigen Themen (= *topic selection*) als Liste für einen Listenabgleich vorzubereiten, müssen wir den Code in die Klammer des `list()`-Befehl einbetten.

Für die Visualisierung müssen wir die Daten in die richtige „Form" für den `seaborn`-Befehl bringen, welcher Topic 1, Topic 3, 9 usw. als einzelne Kacheln der Abbildung 11.9 einzeichnet. Der `seaborn`-Befehl benötigt eine Datentabelle in einer Form, wie sie beispielhaft in Tabelle 11.8 abgebildet ist. Sie sehen im Beispielausschnitt, dass die Prävalenzen untereinander nach Thema und Jahr sortiert werden müssen.

Analog zum `seaborn`-Befehl können mit dem `melt()`-Befehl aus der `pandas library` dem Datensatz Informationen zum Erstellen eines gruppierten Plots übergeben werden. Für die Gruppierung müssen die `id_vars = "Year"` und die `value_vars`, die die Themenprävalenzen der Topics im Datensatz enthalten, unterschieden werden. Die überdurchschnittlich häufigen Themen trennen wir aus dem Datensatz Filme nach Erscheinungsjahr mit der Definition des Listenabgleichs `[x for x in films_by_year.keys() if not re.search ("Year",x)]` heraus. Die neue Datenstruktur wird als Objekt `films_by_year_grouped` benannt. Für den Plot werden nur die überdurchschnittlich häufig auf-

Tabelle 11.8 Aussehen eines zusammengeführten Beispieldatensatzes mit 30 Themen auf Basis des Codes 11.42

Jahr	Thema	Prävalenz
1920	0	0.8
1921	0	0.65
…	…	…
2017	29	0.2
2018	29	0.1
2019	29	0.25

tretenden Themen als Variablen durch Aufruf der Liste `topic_selection` im Befehl `films_by_year_grouped.variable.isin(topic_selection)]` berücksichtigt. Der Backslash \ gibt in Python an, dass Befehle länger als eine Codezeile geschrieben wurden und auszuführen sind. Der Befehl `isin()` vergleicht, ob eine Variable (= die Themenzahl) in der Liste mit den überdurchschnittlich häufigen Topics im Datensatz enthalten ist. Für die Abbildung wird der Name `variable`, der durch den `melt()`-Befehl erzeugt wurde, in `"topic"` umbenannt, damit in Abbildung 11.9 `topic =` statt `variable =` die einzelnen Kacheln betitelt. Im `rename()`-Befehl `columns = {[ALTER NAME] : [NEUER NAME]}` ersetzen wir in Code 11.46 hierfür `[ALTER NAME]` durch `variable` und `[NEUER NAME]` durch `topic`.

Die Überschrift und Daten müssen nun auf die Leinwand der Kacheln mit dem `FacetGrid()`-Befehl gezeichnet werden. Grundsätzlich wird mit `FacetGrid()` eine Leinwand mit einer Reihe an Kacheln generiert. Zum Bemalen der Kachelleinwände muss dem Befehl der Datensatz `films_by_year_grouped_reduced` übergeben werden. Die Variable `col = "topic"` definiert, dass die Prävalenzen der Topics als Grundlage für die Färbung der einzelnen Kachelleinwände verwendet werden. Da eine Word- oder Druckseite nicht unbegrenzt breit ist, muss die eine Reihe der Kacheln mit dem Befehl `col_wrap = 3` auf drei Kacheln pro Zeile in Abbildung 11.9 begrenzt werden. Würden mehr als sechs überdurchschnittlich häufige Topics existieren, so würde eine dritte, vierte usw. Zeile mit Kacheln eröffnet werden. Die unterschiedlichen Farben der Kacheln in Abbildung 11.9, eine andere Farbe für jedes dargestellte Thema, werden mit dem Befehl `hue = "topic"` erzeugt.

Ebenfalls mit Blick auf mögliche Präsentations- und Publikationsformate muss die Bildgröße für Abbildung 11.9 gewählt werden. Die Höhe von Abbildung 11.9 ist durch `height = 4` (Inches) vorgegeben, und mit `aspect = 1` wird spezifiziert, dass das Bild genauso breit wie hoch sein soll. `aspect`-Werte

größer als 1 verbreitern das Bild, und Werte kleiner als 1 bewirken, dass das Bild schmaler wird. Abbildung 11.9 besteht jedoch aus einem Bild, welches sechs Kacheln als kleine Bilder enthält. Mit dem Befehl `sharex = False` und `sharey = False` erhält jede Teilgrafik eine eigene X- und Y-Achse, um die Topic-Verläufe über Zeit abbilden zu können.[27] Die Bild-mit-sechs-Bildern-Leinwand wird als Objekt `g` (kurz für Grafik) definiert.

Diese Leinwand mit den Kacheln muss anschließend mit dem `.map()`-Befehl vervollständigt werden. Die Kacheln erhalten mit dem `regplot`-Befehl die Abbildung der zeitlichen Verläufe als Regressionsgerade. Statt einer Geraden generiert der `order =`-Befehl gebogene Linien. `order =` verlangt die Angabe einer gerade Zahl (Integer), welche die Potenz der zu zeichnenden Linie angibt. Wenn Sie z. B. 2 angeben, dann wird die Linie eine U-Form aufweisen (= ein Maximum oder Minimum); bei 3 werden zwei Extremwerte (Minimum und Maximum) mit Sattelpunkt eingezeichnet. Für Abbildung 11.9 wurde der Wert 5 gewählt, der maximal zwei Hochkonjunkturen pro Thema und zwei Zeitpunkte ausgeben kann, an der die Themen randständig waren. Anders ausgedrückt, stellen wir auf diese Weise die „Kurvigkeit" unserer Linie ein. Speichern können wir die fertiggestellte Abbildung mit dem `.savefig()`-Befehl mitsamt Dateipfad, Namen der Datei und der Auflösung (z. B. `dpi = 300`).

Code 11.46 Erstellen von Themenkonjunkturen für die überdurchschnittlich häufig auftretenden Topics im Datensatz des 30er-Topic Models (Fortsetzung nächste Seite)

```
# ============================================================================
# Themen pro Jahr in Pandas-DataFrame aufnehmen
# ============================================================================

## Eigenschaften auffüllen
document_topic_matrix["Year"] = df.Year

## Themen selektieren (mit Jahreszahl)
films_by_year = films[[x for x in films.keys() if type(x) == int]]

# ============================================================================
# Visualisierung von Themen pro Jahr
# ============================================================================
```

27 Es kann aber auch manchmal erwünscht sein, dass die Werte auf den Achsen der gleichen Metrik folgen, damit eine Vergleichbarkeit zwischen den Werten (z. B. insgesamte Themenprävalenz auf der Y-Achse) hergestellt werden kann.

```
films_by_year = document_topic_matrix.groupby(by="Year").mean()
films_by_year = films_by_year.reset_index()
films_by_year.columns = [str(x) for x in films_by_year.columns]

# ========================================================================
# Erstelle einen gruppierten Plot
# ========================================================================
topic_selection = list(films.mean()[films.mean() > 1.0/30].index)
topic_selection = [str(x) for x in topic_selection]

films_by_year_grouped = pd.melt(films_by_year,
        id_vars = "Year",
        value_vars = [x for x in films_by_year.keys() if not
re.search("Year",x)])


films_by_year_grouped_reduced = films_by_year_grouped\
    [films_by_year_grouped.variable.isin(topic_selection)]

films_by_year_grouped_reduced = films_by_year_grouped_reduced.\
    rename(columns = {"variable" : "topic"})

g = sbs.FacetGrid(films_by_year_grouped_reduced, col="topic",
                  col_wrap=3, hue="topic", height=4, aspect=1,
                  sharex=False, sharey = False)

g.map(sbs.regplot,"Year", "value", order=5)
g.savefig("topics_im_Zeitverlauf.png", dpi=300)
plt.close()
```

11.5.3 Nächste Schritte zum Verständnis der Daten: Themen verstehen

Nun, da Sie wissen, dass Sie sich auch auf schöne bunte Ergebnisdarstellungen aufgrund Ihrer Mühen bei der Datenanalyse freuen können, können wir ans Eingemachte gehen. Die in Wissenschaftstexten fremde, umgangssprachliche Wortverwendung des „Eingemachten“ soll ausdrücken, dass stets ein weiter empirischer Weg und intensive Datenarbeit zum Verstehen der latenten Inhalte der LDA notwendig ist. Dieser Weg beginnt bei der Ausgabe der *top words* oder auch prävalentesten Worte pro Topic mit Angabe der Wahrscheinlichkeit (Ta-

belle 11.9) und schlängelt sich dann entlang der Ausgabe der fünf häufigsten Filme pro Topic mitsamt erweiterter Stichwortliste.

Wie in Tabelle 11.9 als Auszug abgebildet, erhalten wir folgende Informationen:

- Topic
- Wahrscheinlichkeit: Angabe im Englischen mit Punkt statt der deutschen Konvention des Kommas
- Top words

In Tabelle 11.9 sehen wir, dass viele *top words* auf den ersten Blick fehlerhaft und unvollständig sind. Wie bereits oben in Kapitel 11.3 bei Datenbereinigung erklärt, liegen die Ursachen im *stemming* (z. B. Wortendung bei *mummi* (Topic 1, *top word* 2) auf „i" statt auf „y") und der Falscherkennung von Buchstaben (z. B. bei *taik* (Topic 2, *top word* 7) statt vermutlich *talk*).

Die Daten in Tabelle 11.9 ergeben von sich aus keinen tieferen Sinn. Da Daten niemals selbsterklärend sind, liegt es an uns, oder an Ihnen und Ihren (Studien-) Kolleg*innen, den Daten Sinn beizumessen und ihnen für Rezipient*innen Bedeutung zu geben. Sinn und Bedeutung beimessen können wir den Daten nur, wenn wir über hinreichende Kenntnisse des Untersuchungsgegenstandes verfügen. Das sind im vorliegenden Fall Filme und das Thema Wissenschaft, das in diesen Filmen behandelt wird. Folglich ist es auch ein Trugschluss, zu glauben, dass, bloß weil Sie die Methode beherrschen, Sie zu jedwedem Thema eine LDA bzw. Topic Modeling Auswertung durchführen könnten – *the fallacy of the data analyst.*

Zudem erfolgt die beispielhafte Untersuchung auf Medium Data oder Big Data (für eine Unterscheidung siehe Riebling 2018). Das heißt, wir verfügen über eher wenig dynamische Daten, zudem *a lot of data* und damit sehr viele Textdaten für die Analyse. Beispielsweise besteht das Filmskript von *Indiana Jones and the Temple of Doom* (1984; siehe Tabelle 11.1) aus 33 344 Wörtern (116 Word-Seiten). Der gesamte Korpus an 626 Filmskripten umfasst über 39 000 Seiten.

Wie immer bei einer Inhaltsanalyse können wir den Inhalt nicht analysieren und Dritten verständlich machen, sofern wir diesen nicht selbst verstehen. Sofern die *top words* in Tabelle 11.9 nicht bekannt oder eindeutig verständlich sind, wurden die Informationen plausibilisiert, indem zwei Informationsquellen verwendet wurden.

- Online-Lexikon, beispielsweise *www.dict.cc.*
- Online-Enzyklopädie *https://en.wikipedia.org* (auf Englisch, da Textdaten in englischer Sprache vorliegen).

Tabelle 11.9 Auszug der 30 Topics mit Wahrscheinlichkeit/Vorkommen *top words*

	Wahrscheinlichkeit/top words											
Topic	1	2	3	4	5	6	7	8	9	10		
1	0.003* manicur	+ 0.002* mummi	+ 0.002* towni	+ 0.002* church	+ 0.002* goon	+ 0.002* love	+ 0.002* bless	+ 0.002* doctor	+ 0.001* beast	+ 0.001* west	0.01051741659125172	0.031346645
2	0.003* gon	+ 0.002* professor	+ 0.002* cantrel	+ 0.002* love	+ 0.002* buddi	+ 0.002* dog	+ 0.002* taik	+ 0.002* student	+ 0.002* ditch	+ 0.001* brain	0.03824101732833654	0.03108848
3	0.006* rogu	+ 0.002* storm	+ 0.002* ice	+ 0.002* presid	+ 0.002* guard	+ 0.001* world	+ 0.001* editor	+ 0.001* agent	+ 0.001* professor	+ 0.001* offic	0.008229879965914904	0.03046129
4	0.002* raptor	+ 0.002* gon	+ 0.002* hudson	+ 0.002* professor	+ 0.002* okay	+ 0.002* wraith	+ 0.002* pleas	+ 0.002* love	+ 0.002* world	+ 0.001* offic	0.00366307975504117	0.031363178
5	0.002* gon	+ 0.002* world	+ 0.002* presid	+ 0.002* okay	+ 0.002* kelp	+ 0.002* pleas	+ 0.001* kill	+ 0.001* coal	+ 0.001* god	+ 0.001* scientist	0.020233414805366487	0.029752707
6	0.005* curat	+ 0.002* forev	+ 0.002* ant	+ 0.002* tep	+ 0.002* question-nair	+ 0.002* doctor	+ 0.002* data	+ 0.001* ballroom	+ 0.001* pleas	+ 0.001* cleaver	0.0010012759983418543	0.029500745

Mit den beiden Informationsquellen konnten folgende Informationen gewonnen werden. Damit konnten diverse *top words* zwar verständlich gemacht werden, jedoch sind noch nicht alle *top words* in Tabelle 11.9 eindeutig und ergeben Sinn.

- Topic 1/top word 3: towni[e] = abschätzig für Stadtmensch
- Topic 1/top word 5: goon = Schlägertyp, Rowdy (unwahrscheinlich: gooney = Schwarzfußalbatros; gooner = Fan des Fußballklubs Arsenal London)
- Topic 2/top word 3: cantrel['s pentalogy] = Fehlbildung von Herz, Bauchwand, Thorax usw.; Cantrel Peak = Berg in der Antarktis
- Topic 2 und 3/top word 10: offic[e/r] = Büro, Beamte*r, Polizist*in
- Topic 3/top word 7: editor = Redakteur*in bei Zeitung und von Software
- Topic 4/top word 1: raptor = Greifvogel, Raptor (Dinosaurier), Name von Kampfflugzeug
- Topic 4/top word 3: hudson = häufiger Familien- und Ortsname, Fluss in den USA (z. B. durchfließt New York), Hudson Bay = Randmeer in Kanada
- Topic 4/top word 6: wraith = Gespenst, Geistererscheinung
- Topic 6/top word 3: ant = Ameise, Ameisen-…(unwahrscheinlich: Akteur*innen-Netzwerk-Theorie)
- Topic 6/top word 4: tep = total endoprosthesis (Medizin), tep[al] = Blütenblatt
- Topic 6/top word 10: cleaver = Hackbeil, Spalter (Zahnmedizin und Archäologie)

Der spezifische Sinn von *top words* mit mehreren plausiblen Bedeutungen (z. B. *editor, raptor* und *offic[e/r]*) muss jeweils für das entsprechende Topic aus dem inhaltlichen Kontext beigemessen werden. Hier können wir uns wieder nicht auf die Maschinen-, sondern nur die Menschenlesbarkeit zur Verständnisschaffung verlassen.

Aufgrund der Sachkenntnis zu Film und Wissenschaft erscheint uns das *top word* „ant" auffällig (ja, Ihnen auch). Einerseits gibt es diverse Filme, in denen Ameisen vorkommen. Andererseits wissen wir, dass die Topic Modeling Software aufgrund von Trennzeichen in Filmskripten (siehe Tabelle 11.1) Fehler generiert, indem Wörter unvollständig sind oder als maschinengelerntes Artefakt entstehen. Folglich könnte *ant* ein Wortteil sein von:

- ant-enna
- (moose) ant-ler
- quadr-ant
- pl-ant
- mut-ant
- ignor-ant
- serv-ant

Jetzt fragen Sie sich, wie Sie ohne umfassende Kenntnisse des Untersuchungsgegenstandes auf diese Wörter kommen könnten? Einfach: Wir verwenden die Suchfunktion in Excel (ermöglicht das Öffnen der csv-Datei), wobei die Suche bei Kopie der Schlagworte pro Film in eine Word-Datei noch einfacher ist. *Ant* ist ein *top word* in Topic 6 (Tabelle 11.9), welches jedoch zweimal als Teil des jeweils vollständigen Wortes *want* (auf Deutsch: Bedürfnis, Wunsch) vorkommt (Tabelle 11.10, nächste Seite; siehe Filme *The Rocky Horror Picture Show* und *Teachers' Pet*). Diese Erkenntnis erleichtert das Verstehen (noch) nicht, was bei der Betrachtung latenter Inhalte jedoch zu erwarten ist. Auch die fünf angegebenen Filme sind weder auf den ersten noch auf den zweiten Blick als Ameisenfilme zu identifizieren.

- *Twister* (1996): Wissenschaftler*innen auf der „Jagd" nach Wirbelstürmen, um ein neues Wetterwarnsystem zu schaffen und ihre Beziehung zu retten.[28]
- *The Man from Earth* (2007): Abschiedsparty eines Professors, welche als mysteriöses Verhör abläuft.[29]
- *The Rock* (1996): Biochemiker verhindert Anschlag eines abtrünnigen Generals mit Nervengas auf San Francisco.[30]
- *The Rocky Horror Picture Show* (1975): Aufgrund einer Autopanne ist ein Paar dem verrückten Wissenschaftler Dr. Frank-N-Furter ausgesetzt.[31]
- *Teachers' Pet* (1958): (Liebes-)Spannungsgeladener Film um einen Zeitungsredakteur *(editor)* und eine Professorin eines Journalismus Colleges.[32]

11.5.4 Daten noch besser verstehen

Dennoch sollten wir die Ameisen noch nicht aufgeben, denn im Korpus befindet sich der Ameisenfilm: *Them!* (1954). Auch wenn wir den Film nicht kennen und keine Zeit haben, jeden der 625 Filme selbst zu sehen, so können wir uns im Internet Hilfe holen. Bei der *Internet Movie Database* (IMDB) können User*innen Filmzusammenfassungen und -rezensionen einstellen, welche in der Regel einen knappen und guten Überblick zum Film ergeben, wie von Claudio Carvalho aus Rio de Janeiro, Brasilien.

> „In the New Mexico desert, Police Sgt. Ben Peterson and his partner find a child wandering in the desert and soon they discover that giant ants are attacking the locals. FBI

28 Siehe: *www.imdb.com/title/tt0117998/?ref_=nv_sr_srsg_0* (22.11.2021).
29 Siehe: *www.imdb.com/title/tt0756683/?ref_=nv_sr_srsg_0* (22.11.2021).
30 Siehe: *www.imdb.com/title/tt0117500/?ref_=nv_sr_srsg_0* (22.11.2021).
31 Siehe: *www.imdb.com/title/tt0073629/?ref_=nv_sr_srsg_0* (22.11.2021).
32 Siehe: *www.imdb.com/title/tt0052278/?ref_=nv_sr_srsg_0* (22.11.2021).

Tabelle 11.10 Schlagworte der fünf Filme zu Topic 6 (30 Topics *run*)

Topic	Film	Year	Keywords (Schlagworte)*
6	Twister	1996	wow take rock music human fade crew approach yeah realli idea yeah well realiz got got approv warn system lab got analysi data yeah got model data need run analysi run run think run lab alway thing crew walk got sensor work comput went got data come ear twister ever record hey check sky sky know think seen music credit come respect wind view credit start fade cloud fast motion
6	The Man from Earth	2007	seen vista come end seen star collid heard ocean roar know mean friend noth last forev alway heard thing end know noth last forev mayb thing forev feel forev feel seen men take world hand chang point view felt shake seen men take stand fight left noth last forev alway heard thing end know noth last forev mayb thing forev feel forev feel noth last forev alway heard thing end know noth last forev mayb thing forev feel forev feel forev feel
6	The Rock	1996	sprinkler sprinkler pipe knock drop trickl head someth block pipe name gon give build ling get atropin terrifi jerk cabinet reveal inch needl syrinx ling die die inject diffus continu devic goddamn take fumbl syring get away needl ling inject lonner goddamn water pipe knock cough lonner come come sink knee hold syring heart kari style hand trembl concentr devic calm unsettl hand grab instrument perform precis deft function snip wire splice ling sweat second
6	The Rocky Horror Picture Show	1975	lift hall figur turn throw lift cage zoom death mask film chang colour howev evid red mouth host song see met handyman brought knock thought candyman ballroom get strung stride ballroom throne judg book cover much hell give scream let cloak fall throne reveal transvestit attir guest scream slowli move leg kick let show mayb play groovi circl forc ballroom want someth abysm take reef movi clumsili attempt circl ignor greet guest glad caught home use
6	Teachers Pet	1958	ago learn anyth come sell advertis space use sell take stori save make rehash evid disclos accomplic sala alon kill partner alway poverti bitter despair sala pull trigger load thank wait want dame stone daughter combin got know find someth

* Falls länger, wurden Schlagwortlisten aus Platzgründen auf circa 80 Wörter gekürzt

agent Robert Graham teams up with Ben and with the support of Dr. Harold Medford and his daughter Dr. Patricia ‚Pat' Medford, they destroy the colony of ants in the middle of the desert. Dr. Harold Medford explains that the atomic testing in 1945 developed the dangerous mutant ants. But they also discover that two queen ants have flown away to Los Angeles and they are starting a huge colony in the underground flood control tunnels of that city. When a mother reports that her two children are missing, the team begin searching for them. Will they arrive in time to save the children and destroy the colony?"[33]

33 Siehe *www.imdb.com/title/tt0047573/?ref_=nv_sr_srsg_10* (22.11.2021).

Der Film *Them!* bzw. die Ameisen sind jedoch insgesamt nicht von großer latenter Bedeutung im Korpus, weder bei den 30 Topics (Tabelle 11.11) noch bei der detaillierteren Darstellung mit 80 Topics (Tabelle 11.12).

Tabelle 11.11 Schlagworte der fünf Filme zu Topic 1 (30 Topics *run*)

Topic	Titel	Year	Text
1	Them!	1954	other explod nobodi know enter age open world eventu find world nobodi predict
1	Village of the Damned	1995	know well let know arrang father brick wall right sight still bit clear rememb dress hair longer caught london let stay come know thing nearest let drive noth feel useless still see child friday happen especi happen think get troubl tell succeed want road know silli afraid harm strang trust even seem emot cours right ring morn morn brick wall think wall sorri mistress ask take care manner speech matter even child tonight talk energi discoveri year ago complet chang
1	The Man with Two Brains	1983	awak wait want see want foot husband awak see awak never told eater gain much fat fat
1	The Lost World Jurassic Park	1997	eagerli leap chest open scream cemeteri snow fall midwint sky cemeteri group fifti mourner group coffin festoon cascad flower array frame minist read mourner wipe away tear stand distanc group right besid blank sunburn place winter set turn look shoulder sixteen love blond hair notic recogn nudg year turn break welcom ceremoni break walk stand kiss cheek extend shake daughter hello glad came sorri grandfath thank day think decid peopl know island
1	The Theory of Everything	2014	petal travel upward stumbl fell hard revers lift ground beheld firework revers firework implod hole knock tea whilst work desk revers liquid leap ball danc bridg continu theori everyth continu meet parti stand see smile freez jump space travel forward space matter travel speed approach point space singular hole disappear insert card card brief histori sold copi worldwid year plan retir continu theori everyth declin offer knighthood card phd poetri happili marri friend grandchild end credit end

Tabelle 11.12 Schlagworte der fünf Filme zu Topic 23 (80 Topics *run*) (Fortsetzung nächste Seite)

Topic	Titel	Year	Text
23	The Lost World Jurassic Park	1997	mombassa cancel point stream run nearbi carnivor hunt stream bed want set base camp eat peopl bar think heard find spot rememb herbivor risk sigh work put pull asid want run littl camp trip condit charg check tell case scotch condit fee keep want exchang servic right hunt tyrannosaur male busi condit ahead set camp right swamp care safari dentist listen idea okay els say okay lad jungl jungl foliag shiver quak final fall hunter convoy roar jungl stand vehicl speedbird wave forward driver wheel seat

Topic	Titel	Year	Text
23	Eternal Sunshine of the Spotless Mind	2004	seem hold heart shape box platform track empti continu continu almost train pull platform break crowd lurch stair hurri overpass stair train stop door get train apart say goodby call right yeah tomorrow tonight test line exit stay watch apart enter drop overcoat chair dial hello beat call work today said home sick know take think continu continu yeah tri home get messag got think messag volum yeah machin cheer told realli
23	The Lost World Jurassic Park	1997	tent equip tent deton seri explos other knock ground seri concess blast drag four char burn tire slowli spin foot look charg jungl break jungl clear blind see tyrannosaur chain stake bastard bleat pain leg hang tug pull stake ground camp survey destruct fire spread tent tongu flame flap air anim gone personnel scatter breathless smear smoke stagger name hold snipe padlock anim cage alon jungl race ridg trail aav park burst god
23	Saw	2004	workshop suddenli rush distract grab sit floor bathroom still hold eye memori wait exhal sharpli look stand stare someth seem bit tone address know turn care work yeah know paus instinct tone far believ word instinct yeah know look offens say know speak calmli anger lie surfac els tell well let see birthday friend stab nail tell tell girlfriend lower hand fuck feminist punk broke thought told toenail slightli fed stop knew turn light turn away
23	Them!	1954	other explod nobodi know enter age open world eventu find world nobodi predict

Tabelle 11.13 Schlagworte der fünf Filme zu Topic 48 (80 Topics *run*)

Topic	Titel	Year	Text
48	Torn Curtain	1966	shoot basket thought insid ask minut switch said ballerina describ caught refuge costum basket trip got watch close say say hello
48	Them!	1954	think chase wind mayb sent report drank breakfast well call hey wait minut right mayb keep circl hey hey honey hey minut honey name belong code ahead trailer mile road see anybodi around better check okay matter know sunstrok sunburn sun long look shock spot trailer ahead mayb mayb use wake somebodi check blood happen morn check traffic accid cave anyth footprint tire mark found pick scatter sugar yeah someth
48	Raiders of the Lost Ark	1981	swordsman groin street agent lead arab carri basket head basket fasten close make place street cut far bazaar wedg inch scream bazaar heard look squar basket escort disappear build arab rise flash fall ankl frantic push panick mass human direct basket gone chase intercut move basket guy move basket street alley passageway peopl alway seem corner catch glimps basket disappear corner fight human riptid final
48	The Island	2005	furrow appear brow recoil horror perhap rememb level came come spur action follow hall still movement nascent stir footfal rippl nightmar chamber stopgun nose follow move silenc hone much pass signal censor split track advanc intercut foundat chamber censor make strobe armatur mechan move tank angl project fire laser burst sear crosshatch brow foundat chamber conting move row foundat tank nascent cabl plug ear
48	Iron Man	2008	stark honestli expect believ suit conveni appear fact know confus question offici stori entir make accus insinu superhero never said superhero well outlandish hero type clearli list charact defect larg stick card yeah truth iron

11.5.5 Datenbasierte Entscheidungen für die Analyse treffen

Tabellen 11.11, 11.12 und 11.13 zeigen detaillierter als die *top words* (Tabelle 11.9), dass die Ameisen nicht von latenter Bedeutung sind. Vielmehr deuten die anderen um *ant* angesiedelten *top words* – sowohl bei 30 als auch 80 Topics – darauf hin, dass *ant* ein maschinell erzeugtes Artefakt ist. Wider der Maschinen- und mit der Menschenlesbarkeit bestehen nun zwei Möglichkeiten, mit der empirischen Erkenntnis umzugehen. Entweder wir entscheiden uns für die Variante

a) Topic-Mutismus (lateinisch *mutus* = Stummheit; im Englischen: *mute*) und berücksichtigen alle Topics mit *ant* als *top word* nicht für die Auswertung, oder
b) *top word*-Mutismus und ignorieren *ant* als *top word* beim Verstehen der entsprechenden Topics.

Wie so oft in der (quantitativen) empirischen Sozialforschung gibt es hierbei keine richtige und falsche Entscheidung, sondern wir müssen inhaltsbezogen eine Entscheidung treffen. Im vorliegenden Fall würde für a) Topic-Mutismus sprechen, dass Topic 6 vom 30er *run* (Tabelle 11.14) schwierig zu verstehen ist und folglich auch nach Menschenlesbarkeit keinen eindeutigen Sinn ergibt. Der inhaltliche Zusammenhang von *ant* mit *curat[or]* (= Museumsdirektor*in, (Nachlass-)Verwalter*in) ist eher schwer herzustellen, auch ohne die detaillierten Schlagworte in Tabelle 11.9 zu berücksichtigen. Weiter befindet sich mit *forev[?]* ein schwierig zu deutendes *top word* an zweiter Stelle, da Adverbien in der Wortsuche ausgeschlossen sind und die Bedeutung *forevacuum pump* höchst unwahrscheinlich erscheint im Verein mit den anderen *top words.*

Ein weiteres inhaltliches Argument für Stummschalten (Topic-Mutismus) ist, dass *ant* in Topic 26 des 80er *run* das erstgenannte *top word* ist, welches die höchste Ladung (= maschinengelernte Bedeutung) aufweist. Wäre *ant* (jeweils) weniger bedeutsam (z. B. ab fünfter Stelle) oder gar das letztgenannte Wort, so wäre die inhaltliche Bedeutung gering, und wir könnten uns für *top word*-Mutismus entscheiden. Eine Entscheidung für die Variante b) müsste jedoch auch das Kriterium erfüllen, dass wir das Topic 26 verstehen und diesem für Dritte Sinn beimessen können. Es erscheint jedoch zu viel Phantasie nötig, um auf Auszeichnungen hindeutende *top words (medal* und *honor)* mit *museum, blip* (= Markierung im Computer, Radarzeichen), *gurgl[e/ing]* (= Glucksen, Lallen, plätschern) und *presi[dential] campaign* unter einen thematischen Hut mit *monster* und *wasp* zu bringen.

Methodisch sprechen also zumindest folgende Aspekte für den Ausschluss von *ant* und damit dem Topic-Mutismus aller Topics mit *ant:*

- *ant* scheint ein Artefakt zu sein (z. B. aufgrund von Wörtertrennungen);

- *ant* konstruiert inhaltlich latente Scheinzusammenhänge, welche daran zu erkennen sind, dass weder beim höher aggregierten Topic Modeling mit 30 Topics noch bei der detaillierteren Auswertung mit 80 Topics inhaltlicher Sinn erschließbar ist;
- *ant[s]* scheinen selbst im Ameisenfilm *Them!* vorwiegend als *beast[s]* und *creature[s]* adressiert zu werden (Tabelle 11.14, Topic 23).

Wir sehen, wie viel Inhaltswissen und Sachkenntnis des Untersuchungsgegenstandes notwendig ist, um ein artifizielles *top word* erkennen und es dann ausschließen zu können. An den Ausführungen wird erneut deutlich, dass Big Data *a lot of data* für die sozialwissenschaftliche Analyse heißt. Hierbei zeigt sich, dass bei der Datenauswertung die Datenbereinigung (siehe Kapitel 11.3) nicht abgeschlossen ist, sondern vor der Analyse noch notwendige inhaltsanalytische Überprüfungen vorgenommen werden müssen (Phase „Ausbesserungen" in Abbildung 1.2). Trotz des Verlustes von *ant* bzw. bestimmter Topics (Tabelle 11.13) liegt der immense Gewinn des zeitaufwendigen und systematischen Vorgehens bei der Überprüfung darin, dass wir das empirische Material punktuell vertieft kennenlernen.

Selbstverständlich können wir in dieser Ausführlichkeit nicht alle fragwürdigen *top words* überprüfen. Auch hier greift die *About Schmidt* (2002) Regel: Besser geht's nicht! Die ausführlichen Erklärungen zu *ant* sollen Ihnen verdeutlichen, dass Sie basierend auf Ihrer Sachkenntnis des Untersuchungsgegenstandes guten Gewissens Topics stummschalten können, d. h. nicht für die Inhaltsanalyse berücksichtigen können (Tabellen 11.14 und 11.15). Das Beispiel zeigt Ihnen jedoch auch, wie Sie Zweifelsfälle anhand der durch Topic Modeling erstellten Textdaten methodisch klären und für Dritte inhaltssystematisch erklären können. Weiter müssen Sie sich stets vor Augen halten, dass Sie auch Big Data induktiv-quantitativ inhaltsanalytisch auswerten. Angesichts der Datenmenge bedienen Sie sich der Verstehensheuristik des *distant reading* (Moretti 2013), dessen spezifische Qualitäten der Wissensgenese Moretti (2000, S. 57 f.) wie folgt beschreibt.

> „Distant reading: where distance, let me repeat it, *is a condition of knowledge:* it allows you to focus on units that are much smaller or much larger than the text: devices, themes, tropes – or genres and systems. And if, between the very small and the very large, the text itself disappears, well, it is one of those cases when one can justifiably say, less is more. If we want to understand the system in its entirety, we must accept losing something".

Die spezifischen Qualitäten des Wissens von *distant reading* als Fernlesen erfordern jedoch auch, dass wir bei der Erklärung, Deutung und Interpretation, also bei der Sinnstiftung von Topics, aufgrund der potenziellen Fehleranfälligkeit sehr vorsichtig vorgehen müssen. Wie das Zitat von Moretti (2000) betont, zielt das

Tabelle 11.14 *Ant* als Beispiel für problematische *top words* (30 und 80 Topics *run*)

Run	Topic	Top words									
30											
	6	curat[or]	forev[?]	ant	tep[al]	question-nair[e]	doctor	data	ballroom	pleas	cleaver
80											
	26	ant	medal	honor	gurgl[e/ing]	museum	blip	campaign	presid[ent]	monster	wasp

Tabelle 11.15 Latente Inhalte von Filmen am Beispiel des Films *Them!* (30 und 80 Topics *run*)

Run	Topic	Top words									
30											
	1	manicur[e]	mumm[y]	towni[e]	church	goon	love	bless	doctor	beast	west
80											
	23	beast	presid[ent]	missil[e]	creatur[e]	rat	scientist	scream	machin[e]	body	beat
	48	steam	saleswoman	mose	gramp	basket	costum[e]	termin[ate]	inhibitor	deterr[ence]	counterman

spezifische Wissen von *distant reading* auf die Erkenntnis von Ideen *(devices),* Themen und Bildern der Sprache *(tropes),* welche spezifisch für ein Filmgenre sein können und/oder für das System Wissenschaft im Film. Genau für das Fernlesen von Film *(distant reading of film)* bieten die vom Topic Modeling ausgegebenen Daten, insbesondere die Abstrahierungen durch *top words* (Tabelle 11.9), eine gute Basis.

11.5.6 Latente Inhalte an den Beispielen Them! und X-Men erklären, deuten und interpretieren

Trotz des prominenten manifesten Inhalts der durch Atomtests in der Wüste New Mexikos mutierten Ameisen im Film *Them!* weist Topic 1 (*run* mit 30 Topics) in Tabelle 11.11 auf andere latente Inhalte und damit maschinell-errechnete Sinnzusammenhänge hin. Die Stichworte in Tabelle 11.15 skizzieren einen Handlungszusammenhang, in welchem in ländlicher Idylle Frauen stereotyp der Nagelpflege nachgehen, die Kirche (noch) im Dorf steht und sich abschätzig über die Städter *(towni[es])* unterhalten wird, welche die große Gefahr noch nicht erahnen (z. B. *Them!, Village of the Damned* und *Jurassic Park: The Lost World*). Sind die Stichworte ausreichend und informativ genug, um ein analoges Weltraumszenario des Films *The Theory of Everything* zu erfassen, so reicht unsere Phantasie nicht aus, um Sinn aus den wenigen Stichworten zu *The Man with Two Brains „awak wait want see want foot husband awak see awak never told eater gain much fat fat"* zu schaffen. Die bei der IMDB angezeigte Rezension von george.schmidt (vom 11. Apr. 2003) lässt darauf schließen, dass das sich Verlieben der Ehefrau in ein Gehirn in einem Glas als latenter Inhalt ebenso das gewohnte Leben im (Raumschiff-)Heim stört wie mutierte Ameisen und Raptoren.

> „Paging Dr. Hfuhruhurr
> *The Man with Two Brains* (1983) *** 1/2 Steve Martin, Kathleen Turner, David Warner, Paul Benedict, Merv Griffin, Sissy Spacek (voice only).
> Martin is hilarious as brilliant neuro surgeon of the screw-top transplant, Dr. Hfuhruhurr (pronounced as it sounds!) who has brains on his mind and a ‚devil woman' wife (the sultry Turner loving every minute of it) as he falls in love with a brain in a jar (voice supplied by the melancholic Spacek). Part Mad Scientist spoof, part Marx Bros./3 Stooges dialogue all parts brilliantly goofy and some wickedly funny moments. Best line: ‚Into the mud Scum Queen!' Get that cat out of here!! Merv Griffin has a cameo in one truly stunning plot twist".[34]

34 Beste Rezension (IMDB Angabe aufgrund von User*innen *thumbs-up*), siehe: *www.imdb.com/title/tt0085894/?ref_=nv_sr_srsg_0* (22. 11. 2021).

Die ambivalente Darstellung des verrückten Wissenschaftlers in *The Man with Two Brains* ist nicht in den anderen Filmen vorzufinden, in denen Wissenschaftler*innen gegen Gefahren kämpfen und wenn nötig vor der Gefahr fliehen. Basierend auf den diversen empirischen Informationen könnten wir Topic 1 mit „gefährdete heimische Idylle" benennen (Tabelle 11.17). Das ist vielleicht nicht die beste Benennung, sie fasst jedoch gut erkennbar und hinreichend unterscheidbar die wesentlichen latenten Inhalte zusammen.

Im Gegensatz zum Topic Modeling *run* mit 30 Topics sind die latenten Inhalte und damit auch die Filmsequenzen beim *run* mit 80 Topics viel differenzierter. Für die Deutung und Interpretation der Topics hat dies weder Vor- noch Nachteile. Das ist gut belegbar an den Beispielen der Topics 23 und 48 des 80 Topics *run*. In Tabelle 11.12 weist Topic 23 sehr unterschiedliche filmische Sequenzen aus, welche verdeutlichen, dass verschiedene *beast[s]* und *creatur[es]* gewaltsam (z. B. mit *missil[e]*) bekämpft werden müssen. In den Topic 23 am wahrscheinlichsten repräsentierenden fünf Filmen tritt dabei der politisch-militärisch-wissenschaftliche Komplex in unterschiedlichen Konstellationen auf, mal personifiziert durch *presid[ent]* und *scientist* (z. B. *Them!*) und mal als Militär-Wissenschafts-Kombination (z. B. *The Lost World: Jurassic Park*), spielt jedoch in *Eternal Sunshine of the Spotless Mind* keine Rolle. Auch die detaillierteren Stichworte in Tabelle 11.12 lassen recht eindeutig den Schluss zu, dass es sich bei Topic 23 um Handlungssequenzen handelt, in denen gefährliche Kreaturen im Dschungel *(The Lost World: Jurassic Park)* oder in der Wüste *(Them!)* oder im Badezimmer/Haus *(Saw)* oder im (nur scheinbar) gelöschten Gedächtnis *(Eternal Sunshine of the Spotless Mind)* Chaos und Vernichtung verursachen.

Hingegen erschließen sich uns die latenten Inhalte von Topic 48 nicht unmittelbar aus den *top words* in Tabelle 11.13. Hier ermöglicht die in Tabelle 11.13 abgebildete Kontextualisierung die Identifikation einer erst ruhigen Frühstücks- *(Them!)* und Basarszene *(Raiders of the Lost Ark),* welche durch das rasche Verpacken, insbesondere von Gegenständen in Körbe *(basket* in *Raiders of the Lost Ark* und *Torn Curtain)* hektisches Treiben in Form von Verfolgungsjagden ausbricht. Das in die Rüstung ein- und wieder auspacken bei *Iron Man* wird hierbei von LDA scheinbar als äquivalente Handlung erfasst, ebenso wie das alptraumhafte und gewaltsame Entpacken der dystopischen Bedeutung von *The Island.*

Zurück zum 30er Topic Modeling *run* und den *X-Men* (übrigens *men* für Menschen). Beispielsweise kann Topic 3 (top words: rogu[e], storm, ice, presid[ent], guard, world, editor, agent, professor und offic[e/r], Tabelle 11.2) gut in Verbindung mit den fünf Filmen *The Lawnmower Man* (1992), *Arctic Blast* (2010), *The Peacemaker* (1997), *Splice* (2009) und *X – The Man with the X-ray Eyes* (1963) nachvollzogen werden. Im Gegensatz zu Topic 3, offenbaren bei Topic 28 die den realen Naturphänomenen gleichen Namen aus der *X-Men*-Filmreihe sich als mitgeschleppter analytischer Ballast aufgrund der Entscheidung bei der Datenbereinigung (Kapitel 11.3). Insbesondere der Film *X-Men 2* (1991) dominiert

bereits unter den top fünf Filmen *The Raven* (1935), *Jurassic Park: The Lost World* (1997), *Twister* (1996) und *Tremors 3: Back to Perfection* (2001) die *top words*. Dabei wird das *top word raven* auch in den knappen Stichworten *„think littl gentli raven huh"* dem gleichnamigen Film zuortbar, und *dog* erweist sich als ein thematisch irrelevantes Nebenhecheln von Tyrannosaurus-, also Raptorenfutter mit nachfolgendem Durstlöschen aus dem Pool des Hauses (Ort: San Diego) in *Jurassic Park: The Lost World*. Dennoch ist *Raven Darkholme* eine Filmfigur aus der *X-Men*-Filmreihe, welche in *X-Men 2* dazu beiträgt den Mutanten *nightcrawler* (= Regenwurm) zu stellen, welcher sich in das Weiße Haus eingeschlichen hat, um den US-Präsidenten zu töten. Die Topic 28 zugeordneten Stichworte zeigen dabei, dass es sich bei *storm* sowohl die Filmfigur als auch den (Wirbel-)Sturm handelt, und beim Topic Modeling beispielsweise der in Tabelle 11.16 prominent vorkommende Name *magneto* aufgrund seines Vorkommen auf der Stoppliste ignoriert wurde. Dennoch zeigen die Topic-Beispiele, dass eine Konzentration von Daten, im vorliegenden Fall aufgrund des Vorkommens mehrere Filme der *X-Men*-Franchise in einem insgesamt aus singulären Filmen bestehenden Datenkorpus zu erheblichen analytischen Problemen führt.

Tabelle 11.16 Überlagerung der Bedeutung von *X-Men 2* (1991) in Topic 28

Top words Topic 28	Stichworte (begrenzt auf ca. 80 Wörter)
dog nightcrawl[er] agent raven gon[e] fingerprint storm seamstress [auch: dressmaker] kill captain	storm nightcrawl reappear find platform illus nightcrawl stare billion light rotat hum get light brighten nightcrawl finish prayer heaven clear hum echo tug chain feel effect dark open scream noth come hear magneto see continu douahertv continu lift see grin magneto seem run whip chain wrap throat tighten magneto word never happen helicopt ground moment later magneto sit push button propel begin pull stick glanc turn match eye narrow curious figur stand alon helicopt magneto look advic answer magneto reach open magneto extend

Überhaupt keine Anhaltspunkte in den Textdaten sind zum *top word seamstress* zu finden, wobei bei den 80 Topics *seamstress* das *top word* von Topic 44 ist und die fünf häufigsten Plotausschnitte einzig aus dem Film *Demolition Man* (1983) stammen, wobei der Blick ins Filmskript nur zwei Vorkommen von *seamstress* offenbart, wie in Tabelle 11.17 abgebildet – ein weiterer unübertrefflich großer Filmmoment dargebracht vom US-Schauspieler Silvester Stallone (bitte um Entschuldigung für die unwissenschaftliche Bewertung):

Am Beispiel von Topic 28 sehen wir, dass zwischen dem latenten Maschinen- und Menschenlesen von *top words* kein zwingender Zusammenhang besteht. Die Maschinenlesart wurde abgelenkt von Nicht-Unterscheidbarkeiten von Namen

Tabelle 11.17 Auszug zu *top word seamstress* aus dem Film *Demolition Man* (1983)

Absatz	Text und Regieanweisungen in Filmskript
1	[Lieutenant Lenina] *Huxley* [mit regem Bezug zu *1984* Aldous Huxley; als Film ebenfalls im Korpus]: (chuckling) It was your rehab training. For each inmate the computer draws up a skill or trade which best suits their genetic disposition. It implants the knowledge and desire to carry out whatever training was assigned.
2	*Spartan:* I'm a ‚seamstress?' Seamstress. Great. How come I come out of cryoprison and I'm Betsy fucking Ross and Phoenix comes out and he can access computers, operate all vehicles, find the locations of every damn thing in the city? (he has a thought) Can you get me Phoenix's rehab program?
3	[Regieanweisung] Huxley punches madly away. An *access denied* sign flashes on the screen, cutting her off. Lenina gets into a little more furious *computer* playing until she gets a violent *beep.* A *sweet female computer voice chirps* along with corresponding printed information.
4	*Sweet female computer* (V. O.): Phoenix, Simon. Rehabilitation skills; Urban combat kill, torture methodology, computer override authorization, violent …
5	*Spartan:* Who develops the rehab programs? Attila the Hun?

(Tabelle 11.15, 80er Durchlauf) und der Seltenheit eines *top words* und (zufällig oder unerklärlich systematisch häufigen) benachbarten Worten (Tabelle 11.16). Im Ergebnis müssen wir einen Topic-Mutismus festhalten, d. h. Topic 28 von der weiteren Analyse ausschließen. Folglich können wir nicht *per se* von einem Maschine-Mensch-Gleichschritt des Verstehens ausgehen, insbesondere bei einem heterogenen Textkorpus wie Wissenschaftsfilmskripten. Eine Daumenregel hierbei ist, dass Sie heterogene Texte dann gut interpretieren können, wenn das Verhältnis zwischen gut interpretierbaren zu nicht interpretierbaren Themen etwa 3 zu 2 ist. Bei homogeneren, besser auszulesenden und interpretierbareren Daten (z. B. Twitter-Daten) ist ein deutlich besseres Verhältnis von etwa 3 zu 1 zwischen interpretierbaren und nicht interpretierbaren Themen zu erwarten.

11.6 Zusammenfassung und abschließende Worte

In diesem Kapitel haben wir Ihnen demonstriert, wie Sie von der Aufbereitung eines Textkorpus über die Anwendung von Topic Models (LDA) hin zu einer *distant reading*-Interpretation von Themen und Texten gelangen können. Dabei haben Sie die Tücken, aber auch die Chancen von Topic Modeling-Ansätzen kennengelernt. Obwohl wir mit einigen hundert Filmskripten einen vergleichsweise kleinen Korpus analysiert haben, konnten wir dennoch einige Themen unter Hinzunahme unseres Expert*innenwissens rekonstruieren. Diese Themen adressierten den militärisch-wissenschaftlichen Komplex, das Bekämpfen von

Tabelle 11.18 30 Topics Auszug für Beispiele (fehlgeschlagene) Benennung Topics

Topic	Themen	Top words									
1	Gefährdete heimische Idylle	manicur[e]	mumm[y]	towni[e]	church	goon	love	bless	doctor	beast	west
2	Operation konnte geliebtes Wesen nicht retten	gon[e]	professor	cantrel	love	buddy	dog	talk[er]	student	ditch	brain
3	(Bericht über) Bedrohung durch gefährliche Naturgewalten	rogu[e]	storm	ice	presid[ent]	guard	world	editor	agent	professor	offic[e/r]
6	–	curat[or]	forev[?]	ant	tep[al]	question-nair[e]	doctor	data	ballroom	pleas	cleaver
28	–	dog	night-crawl[er]	agent	raven	gon[e]	fingerprint	storm	seam-stress	kill	captain

Monstern und anderen Bedrohungen oder zeigten eine Vorstadtidylle oder Regieanweisungen oder aber konkrete Orte (wie das Hotel) auf.

Stellen Sie sich vor, dass Ihnen ein Korpus bestehend aus Millionen von Nachrichten, z. B. Zeitungen, Forenbeiträgen, wissenschaftlichen Texten, Parlamentsdebatten oder Chatprotokollen vorliegen. Mit steigender Anzahl der in einen Korpus aufgenommenen Texte und entsprechender Wortfilterung können Sie auf die demonstrierte Art und Weise Themenkonjunkturen, Diskursstrukturen oder ganze Debatten im Zeitverlauf rekonstruieren. Sie können auch die Sentiment-Analyse aus Kapitel 10 anwenden, um dann die emotionale Geladenheit von Argumenten und Themen im Zeitverlauf aufzuzeigen. Alternativ können Sie eine Korrespondenzanalyse durchführen, um die Themenstruktur mit weiteren Informationen, z. B. der Sprecher*innen oder der sozialen Umstände, aus denen heraus diese eine Aussage tätigten, mit in die Analyse aufnehmen.

Auf jeden Fall haben Sie gemerkt, dass trotz der Einordnung der LDA als eine digitale, automatisierte Form der quantitativen Textanalyse noch enorm viel qualitative Interpretationsleistung abverlangt wird, ehe die Themen sinnvoll interpretiert und an Texte zurückgebunden werden können. Wir können sogar noch einen Schritt weitergehen und behaupten, dass eine Trennung in qualitative und quantitative Methoden eher artifiziell ist und einem Dialog über Erklärung und Verstehen sozialer Phänomene eher im Weg steht. Erst die qualitative Anreichung der quantitativen Daten oder die quantitative Einordnung qualitativ-verstehender Urteile über Textinhalte (und eventuell angeschlossener, weiterer Methoden wie der Regressionsanalyse) können ein gesamtheitliches Bild sozialer Phänomene ermöglichen. Wir können mit dieser Methodenkombination, zu der auch unserer Meinung nach die LDA gehören sollte, je nach Geschmack eine neue, digitalisierte Art computergestützter Ethnographie mit quantitativ-verallgemeinerndem Einschlag oder eine quantitativ-erklärend-verstehende Forschung mit einem stark qualitativ-interpretierenden Kern durchführen. Wie Sie die Methode letztlich verwenden und an welcher Stelle im Forschungsprozess Sie mit anderen Methoden kombiniert werden kann, ist Ihrer Entscheidung überlassen.

12. Die Schlussworte: keine Angst vor Daten, Software und Interpretation

Nun sind Sie am Ende des Buches angelangt und haben einen Überblick über unterschiedlichste Verfahren der qualitativen und quantitativen Inhaltsanalyse erhalten und wie diese digital unterstützt, teilautomatisiert und vollautomatisiert umgesetzt werden können. Wir haben Ihnen gezeigt, welche methodengeleiteten Auswertungstechniken bzw. Verfahren und Softwareprogramme Sie verwenden können, um die jeweiligen Verfahren der Inhaltsanalyse durchführen zu können.

Unser Anliegen war es, Ihnen die ausgewählten Auswertungstechniken der Inhaltsanalyse in einem Buch in möglichst einfacher, gut nachvollziehbarer Art vorzustellen und anwendungsorientiert zu vermitteln. Wir haben Ihnen in der Einleitung einen Überblick geliefert, wie Sie eine Entscheidung treffen können, welches Verfahren sich für welche Fragestellung eignet und welche Daten Sie dafür benötigen (Abbildung 1.1). Geklärt wurde ferner in Kapitel 2, welche Formen der Kommunikation es gibt und wie sich diese mit qualitativen und quantitativen Verfahren untersuchen lassen. Bitte bedenken Sie hierbei auch, dass die unterschiedlichen Verfahren stets auch mit den Möglichkeiten und Grenzen von Datentypen sowie deren Verfügbarkeit zusammenhängen (Kapitel 3).

Am Ende erscheint es uns zentral, noch einmal vier Punkte hervorzuheben, die Ihnen als Orientierung für Ihre Forschung und die Anwendung textanalytischer Verfahren dienen können.

1. Wie wir aufgezeigt haben, benötigen Sie für die verschiedenen Verfahren unterschiedlich viel Zeit für die einzelnen Verfahrensschritte (Abbildung 1.2). So verbringen Sie beispielsweise für die qualitative induktive Inhaltsanalyse am meisten Zeit mit der Analyse (Entwicklung des Kategoriensystems und Interpretation der Daten) und weniger Zeit mit der Datenerhebung und -bereinigung. Denn Sie werden nur wenige Daten analysieren, da die Analyse zeitaufwendig ist und es bei diesem inhaltsanalytischen Verfahren darum geht, bei der Interpretation der Daten in die Tiefe zu gehen (z. B. einer Einzelfallnarration in Kapitel 4). Beim Verfahren des Topic Modeling wiederum verwenden Sie viel Zeit auf die Datenbereinigung und Programmierung und relational weniger Zeit auf die Analyse, welche statistisch und visuell unterstützt wird (z. B. Kapitel 11.5.2). Am Ende sind aber alle Verfahren empirischer Sozialforschung ähnlich zeitintensiv und arbeitsaufwendig, der Zeit- und Arbeitsaufwand verteilt sich nur unterschiedlich.
2. Auch wenn Sie bei manchen Verfahren weniger Zeit für die Analyse aufwenden, so sind doch alle Verfahren interpretative Verfahren. Das heißt Sie erhal-

ten nicht am Ende eine Zahl, die alles aussagt[1] und allgemeinverständlich ist. Vielmehr müssen die Ergebnisse, die bei allen Verfahren entstehen, interpretiert werden. Interpretation (lateinisch *interpretatio,* auf Deutsch: Auslegung, Übersetzung und Erklärung) bedeutet im allgemeinen Sinne das Verstehen oder die subjektiv als plausibel angesehene Deutung von etwas Gegebenem oder wenigstens von etwas Vorhandenem. Das kann z. B. eine Aussage im Text, ein Kunstwerk oder eine soziale Situation sein. Diese Interpretationen müssen transparent und intersubjektiv nachvollziehbar sein. Selbst bei manifesten Inhalten können wir nicht davon ausgehen, dass Sie und Dritte dasselbe verstehen. Vor allem wenn Sie latente Inhalte, also solche die nicht sofort ersichtlich sind, rekonstruieren wollen, müssen die Interpretationen in den Daten auffindbar und für Dritte transparent nachvollziehbar sein.

3. Uns war es in diesem Buch leider nicht möglich, Beispiele für alle Textformen heranzuziehen. Vielmehr haben wir die unterschiedlichen Verfahren an wenigen empirischen Daten dargestellt, um Ihnen eine Vergleichbarkeit zu ermöglichen. Grundsätzlich können für die Inhaltsanalyse aber nahezu alle Textformen herangezogen werden. Besonders häufig werden Interviews im qualitativen Spektrum verwendet, während Twitterdaten am automatisierten, quantitativen Ende des Spektrums dominieren. Dass wir diese Texttypen nicht herangezogen haben, soll keine Beurteilung sein! So könnten Sie beispielsweise in einer Sekundärauswertung einer großen Interviewstudie oder einer Diskursanalyse von Tweets sowohl induktiv-quantitative vollautomatisierte Verfahren als auch eine induktive oder deduktive qualitative Inhaltsanalyse durchführen, wenn Sie aus der großen Datenmenge eine Stichprobe bzw. Sample ziehen (das natürlich gut begründet sein muss; Kapitel 3.2).
4. Die vorigen drei Punkte zeigen, dass methodengeleitete empirische Sozialforschung sowohl Wissen *(know-that)* und arbeitsintensives Erfahrungsammeln (für *know-how*) als auch Kreativität (z. B. bei der Dateninterpretation) erfordert. Die genannten drei Punkte verdeutlichen auch, dass es nur wenige inhaltsanalytische Verfahren gibt, die aufgrund geringer Datenmengen ohne Softwareanwendung durchgeführt werden sollten oder können. Beispielsweise können Sie eine induktiv-qualitative Inhaltsanalyse wie in Kapitel 4 ohne MAXQDA durchführen. Durch die Verwendung von MAXQDA oder einer ähnlichen Auswertungssoftware bauen Sie jedoch einen systematischen

---

1 Zur Veranschaulichung hier ein kleiner Exkurs zum Buch „Per Anhalter durch die Galaxis" von Douglas Adams. In diesem Buch wird beschrieben, wie ein Supercomputer geschaffen wurde, der die Antwort auf die Frage „nach dem Leben, dem Universum und dem ganzen Rest" liefern sollte. Nach Millionen Jahren Rechenzeit kommt der Supercomputer zu dem Ergebnis, dass die Antwort auf alle Fragen 42 ist. Doch damit ist niemandem geholfen, denn 42 sagt nichts aus. Die Zahl 42 muss interpretiert werden, wozu es aber weiterer Informationen bedarf.

Wissensspeicher auf und können gut nachvollziehen und nachvollziehbar machen, wie Sie bei der Datenanalyse vorgegangen sowie zu Ergebnissen und deren Interpretationen gelangt sind. Bei allen anderen in diesem Buch vorgestellten inhaltsanalytischen Methoden kommen Sie ohne Softwareunterstützung nicht aus. Vorgestellt haben wir nur eine Auswahl mit AntConc, MAXQDA, Python, R/RStudio und VosViewer (siehe zu Copylefts und Copyrights Kapitel 1.3). Ob im Studium oder während der Doktorarbeit haben Sie überdies hinreichend Zeit, um sich die eine oder andere Software anzueignen. So anspruchsvoll die Begriffe zu erlernen und erschlagend die Programmierungen im ersten Augenblick sein mögen, so sehr zeigt die Erfahrung in der Lehre immer wieder, dass aller Anfang zwar schwer ist, wenn Sie aber einmal gelernt haben, ein Softwareprogramm anzuwenden, Sie sich rasch(er) auch in weitere Auswertungssoftware einarbeiten können. Zudem finden Sie auf den Webseiten der Anbieter*innen und an anderen Orten im Internet viele Manuals und Einführungsvideos zu Softwareanwendung.

Mit Blick auf den Arbeitsmarkt und einen Beruf, den Sie gerne ergreifen möchten, sind jetzt und in Zukunft spezifische Softwarekenntnisse als hoch einzuordnen, und bedeutsam ergänzend zu grundlegenden Methoden-, Theorie- und interessensspezifischen Sachkenntnissen sowie Analysefähigkeiten von empirischen Sozialforscher*innen. Für Sozialwissenschaftler*innen, egal, ob Sie Erziehungswissenschaftler*in, Psycholog*in, Politolog*in oder Soziolog*in sind, gilt jedoch, dass Sie nicht als Spezialist*in in einem der genannten Bereiche ausbildet werden, denn Informatiker*innen und sogenannte Data Analysts werden in der Regel stets besser programmieren können als Sie. Folglich gilt es für Sie, ein spezifisch sozialwissenschaftliches „Kompetenz-Paketangebot" zu erlernen, und auf dem Arbeitsmarkt anzubieten. Selbstverständlich sind die genannten sozialwissenschaftlichen Kenntnisse und Kompetenzen auch über den Arbeitsmarkt hinaus von gesellschaftlicher Relevanz, zum Beispiel zur persönlichen Orientierung und Unterstützung von Mitmenschen in der Wissensgesellschaft.

12.1 Keine Datenanalyse ohne Interpretation

Da es sich hier um ein Einführungsbuch für die unterschiedlichen Spielarten der Inhaltsanalyse handelt, haben wir darauf verzichtet, in den einzelnen Kapiteln lange Interpretationstexte zu schreiben. Denn diese sind stark von dem Forschungsfeld und den darin bereits vorhandenen Forschungsergebnissen, von der Forschungsfrage, die sich aus den bereits vorhandenen Forschungsergebnissen ableitet, der theoretischen Brille und den vorhandenen Daten abhängig. Wie wir in Kapitel 4 (induktiv-qualitative Inhaltsanalyse) und Kapitel 5 (deduktiv-qualitative Inhaltsanalyse) aufgezeigt haben, können je nach Fragestellung, Verfahren

und theoretischer Brille aus den gleichen Daten unterschiedliche Erkenntnisse gewonnen werden. Insofern ist es uns wichtig, an dieser Stelle noch einmal darauf hinzuweisen, dass wir Ihnen mit diesem Methodenbuch nur in begrenztem Umfang eine Anleitung geben können, wie Sie die Daten aufbereiten und analysieren können, aber nicht, wie Sie die dann gewonnenen Ergebnisse interpretieren können.

Auf drei Herausforderungen möchten wir an dieser Stelle jedoch noch einmal hinweisen: Erstens ist es für die Interpretation Ihrer Ergebnisse zentral, dass Sie den Kontext der Daten im Blick behalten, die Daten also kontextabhängig interpretieren. Dies gilt für alle Verfahren der Inhaltsanalyse. Zwei Beispiele: In der Autoethnographie (siehe Kapitel 4 und 5) sind Beschreibungen vorhanden, die Sie ohne die Kontextinformationen, wie das Material entstanden ist, nicht einordnen und damit nicht kodieren könnten. Es fallen Begriffe wie „Moodle", „Zoom", „Ersti" usw. Begriffe, die wir derzeit alle kennen und einordnen können, die aber wahrscheinlich in ein paar Jahren erklärungsbedürftig sind. Sie kennen die Begriffe auch nur, da Sie sich selbst wahrscheinlich im Hochschulkontext bewegen, also den Erfahrungsraum, den Kontext implizit teilen, ohne darüber nachzudenken. Wäre dies anders, müssten Sie für sich die Aussagen und Begriffe ebenfalls nachschlagen und die Bedeutung rekonstruieren, bevor Sie sie kodieren können. Das Beispiel zeigt, dass Sie die Informationen, die in den Autoethnographien enthalten sind, einordnen können, weil Sie sie persönlich verstehen. Hätten wir aber Beispiele gewählt, die Sie nicht auf Anhieb verstehen könnten, wären die Kontextinformationen zur Entstehung des Materials noch entscheidender. Nehmen Sie beispielsweise an, es wären Autoethnographien vom Alltag einer*eines Büroangestellten. Da könnte es ohne Kontext eine Weile dauern, bis Sie herausgefunden haben, wovon die Person spricht (z. B. womöglich ihre*seine Studienerfahrung im Rückblick).

Auch für die teil- und vollautomatisierten Verfahren sind Kontextinformationen sehr bedeutend. Es macht einen großen Unterschied, ob Sie strukturierte, mit Verweisen auf andere Textdaten gespickte Twitter-Daten analysieren, durch Webcrawling Daten von Webseiten heruntergeladen haben oder einen Dokumentenkorpus wie beispielsweise digitalisierte Bücher (z. B. vom Projekt Gutenberg), Filmskripte, Liedtexte oder Online-Zeitungsartikel für Ihre Forschung verwenden. Bedenken Sie dabei stets, dass diese Daten in mehr oder minder strukturiertem Format existieren, und Sie bei geringer einheitlicher Strukturierung der Dokumente viel Zeit für die Aufbereitung verwenden müssen, damit Sie die Verfahren des Topic Modeling überhaupt anwenden können! Zudem erhalten Sie beim Topic Modeling in der Regel Themen, mit denen Sie unter Umständen im ersten Moment wenig anfangen können. Auch hier ist es dann entscheidend, die jeweiligen Textstellen als Kontextinformationen heranzuziehen und für die Interpretation der Topics zu nutzen, was ebenso für die Korrespondenzanalyse und Sentiment-Analyse zutrifft.

Zweitens: Haben Sie keine Angst vor unklaren bzw. nicht eindeutigen Ergebnissen. In der Forschung ist das der Standard bzw. Ausgangspunkt von Datenarbeit. Oftmals werden derzeit aber nicht signifikante (= unter statistischer Wahrscheinlichkeit) Ergebnisse in der quantitativen Forschung von der Veröffentlichung ausgeschlossen. Unklare Ergebnisse sind auch Ergebnisse, wenn Sie Folgendes beachten: Erstens müssen Sie sicherstellen, dass Ihr Sampling gut gewählt war (siehe Kapitel 1.2). Unklare Ergebnisse können an der falschen Personenauswahl bei Interviews oder an falschen Daten für Ihre Forschungsfrage liegen. Wenn Sie das ausschließen können, dann muss zweitens geklärt werden, ob es Fehler bei der Anwendung der Methode gab. Wenn das nicht der Fall ist, dann ist zu klären, wie die unklaren Ergebnisse interpretiert werden können. Hier raten wir Ihnen auf theoretische Konzepte zurückzugreifen und bereits durchgeführte Studien zu Ihrem Forschungsthema heranzuziehen. Oftmals standen andere Sozialwissenschaftler*innen vor ähnlichen Herausforderungen, und Sie können in deren Forschung Ansätze für Erklärungen Ihrer empirischen Daten finden.

Drittens kann es vorkommen, dass Sie bei der qualitativen Inhaltsanalyse vor Textpassagen stehen werden, die vage Aussagen beinhalten und deshalb sehr schwierig zu verstehen, zu kategorisieren und somit zu kodieren sind. Hier müssen Sie zunächst entscheiden, ob die Textpassagen trotz der vagen Aussagen Informationen mit Bezug zu Ihrer Forschungsfrage enthalten und deshalb kodiert werden müssen. Wenn die Textpassagen (mit großer Sicherheit) nicht Ihre Forschungsfrage beantworten helfen, ist es sinnvoll, diese nicht zu kodieren. Im Zweifel, also bei eindeutiger Uneindeutigkeit, raten wir Ihnen vor dem Kodieren, die entsprechenden Textpassagen in einer Gruppe zu besprechen, zumindest auszugsweise, damit Sie ausschließen können, dass Sie aufgrund Ihres Vorwissens und unreflektierter Vorannahmen ausschließlich Ihre subjektive Sichtweise in die Textpassage hineininterpretieren und diese damit nicht gegenstandsangemessen kodieren. Wenn Sie sich bei Ihrer Interpretation unsicher sind oder die Textpassagen vieldeutig sind, dann ist es zulässig, ja sogar gute wissenschaftliche Praxis, die betreffenden Textpassagen als unklar zu markieren und mehrere Interpretationen als Möglichkeit anzubieten. Da unseres Erachtens mit der qualitativen Inhaltsanalyse keine Häufigkeitsauszählungen gemacht werden sollten, ist es mit entsprechendem Verweis zulässig, eine Textpassage in unterschiedlichen Kategorien zu kodieren.

12.2 Analysedreischritt: Kontext-Verstehen, Inhalte-Verstehen und dem Publikum verständlich machen

Zum Abschluss möchten wir noch auf den Analysedreischritt aus Kapitel 2 zurückkommen. Dieser besteht aus „Kontext-Verstehen, Inhalte-Verstehen und dem Publikum verständlich machen". Die Beschreibung der unterschiedlichen Verfahren der Inhaltsanalyse sollte Ihnen gezeigt haben, wie sich der Analysedreischritt in den Verfahren unterscheidet. Als erster Schritt, dem „Kontext-Verstehen", gilt es das soziale Phänomen anhand der sozialen, objektbezogenen, räumlichen und zeitlichen Aspekte zu erfassen. Sie benötigen bei den vorgestellten Verfahren der Inhaltsanalyse unterschiedlich lange, um die manifesten Inhalte, also die Inhalte, die ohne Interpretation zu verstehen sind, zu extrahieren.

Um die latenten Inhalte, also die Inhalte, die rekonstruiert werden müssen, zu verstehen, bedarf es dann des zweiten Schritts, des „Inhalte-Verstehens". Hier werden durch Erklären, Deuten, Interpretieren und Schlüsse ziehen die latenten Inhalte extrahiert, die plausibel und für Dritte gut nachvollziehbar sein müssen und mit dem untersuchten sozialen Phänomen im spezifischen Kontext in Zusammenhang stehen.

Das im vorigen Absatz dargestellte Vorgehen im Analyseschritt 2b knüpft nahtlos an den Analyseschritt 3 für andere Wissenschaftler*innen als Adressat*innen bzw. Publikum der Analyse an (Tabelle 2.1). Zur Vereinfachung des „Publikum-Verstehens" für die Wissenschaft ist es hilfreich, die gängigen fachwissenschaftlichen Begriffe zu verwenden – allerdings ohne beispielsweise soziologischer klingen zu wollen als Soziologieprofessor*innen. Ein gewisses Maß an einschlägiger Sprache ist auch in Publikationen empfehlenswert, welche ein Fach- oder Expert*innenpublikum ansprechen sollen. Fremdwort- und Fachjargon-Abstinenz ist empfehlenswert, wenn das Publikum unspezifischer ist, beispielsweise Leser*innen einer Zeitung. Um den Zeitungsleser*innen das soziale Phänomen verständlich zu machen, d. h. ihr Interesse zu wecken, müssen jedoch sämtliche Aspekte der Analyseschritte 1 und 2 berücksichtigt werden.

An unseren Ausführungen sollte deutlich geworden sein, dass die Ergebnisse der Inhaltsanalyse immer abhängig sind von der Forschungsfrage, dem Kontext der Forschung, den Forschungsdaten und den angewandten Verfahren. Insofern unterscheidet sich die Reichweite der Aussagen, die Sie mit Ihren Ergebnissen machen können, also wie übertragbar Ihre Ergebnisse auf andere Fälle, andere Kontexte oder auch andere Forschungsmethoden sind. Grundsätzlich größere Reichweite haben theoretische, d. h. klar mit der Empirie verbundene, jedoch abstrahierte Darstellungen eines sozialen Phänomens und sozialer Realität. Als Theorie formuliert ist es nachrangig, ob diese Theorie auf einem Fall oder vielen Fällen beruht. Die Reichweite bzw. Verallgemeinerbarkeit von Aussagen nimmt auch mit der Datenmenge zu, da mehr Fälle in der Regel eine breitere Repräsentation eines sozialen Phänomens bedeuten. Bei großer Reichweite

Box 12.1: Beitrag Anwendungsbeispiele
Wir möchten Sie gerne ermutigen Ihre eigenen Anwendungsbeispiele, die durch Nutzung dieses Buches entstanden sind, auf dem Blog *https://sozmethode.hypotheses.org/methodenbuch* zu veröffentlichen. Was veröffentlicht werden kann und wie das Prozedere ist, erfahren Sie auf dem Blog. Wir freuen uns auch über Weiterentwicklungen der Verfahren oder Codes und sind sehr an einem Austausch mit Ihnen interessiert.

herrscht eine große Übertragbarkeit vor, bei geringer Reichweite eine geringe Übertragbarkeit.

Denken Sie aber bitte nicht, dass Ihre Forschungsergebnisse unbedeutend sind, nur, weil sie sehr spezifisch sind, nur einen Fall betreffen, und eine geringe Reichweite haben! Insofern möchten wir alle ermutigen, uns methodische Reflexionen zu Ihrer Anwendung der Inhaltsanalyse zukommen lassen, die wir gerne, nach Prüfung, auf dem Blog *sozmethode* veröffentlichen. Denn methodische Reflexionen aus unterschiedlichen Forschungskontexten kann für andere sehr hilfreich sein. Also bitte: Seien Sie mutig und melden sich bei uns!

Literatur

Abercrombie, G. & Batista-Navarro, R. (2020): Sentiment and position-taking analysis of parliamentary debates: A systematic literature review. *Journal of Computational Social Science,* 3, S. 245–270.

Akremi, L. (2019): Stichprobenziehung in der qualitativen Sozialforschung. In: Baur, N. & Blasius, J. (Hrsg.), *Handbuch Methoden der empirischen Sozialforschung.* Wiesbaden: Springer, S. 313–331.

Akremi, L., Baur, N., Knoblauch, H. & Traue, B. (Hrsg.) (2018): *Handbuch interpretativ forschen.* Weinheim, Basel: Beltz Juventa.

Andreotta, M., Nugroho, R., Hurlstone, M. J., Boschetti, F., Farrell, S., Walker, I. & Paris, C. (2019): Analyzing social media data: A mixed-methods framework combining computational and qualitative text analysis. *Behavior research methods,* 51(4), S. 1766–1781.

Armat, M. R., Assarroudi, A., Rad, M., Sharifi, H. & Heydari, A. (2018): Inductive and Deductive: Ambiguous Labels in Qualitative Content Analysis. *Qualitative Report,* 23, S. 219–221.

Arndt, C., Ladwig, T. & Knutzen, S. (2020): *Zwischen Neugier und Verunsicherung – Interne Hochschulbefragungen von Studierenden und Lehrenden im virtuellen Sommersemester 2020.* https://doi.org/10.15480/882.3090

Autor:innengruppe AEDiL (2021): *Corona-Semester reflektiert Einblicke einer kollaborativen Autoethnographie.* Bielefeld: wbv.

Badjatiya, P., Gupta, S., Gupta, M. & Varma, V. (2017): Deep learning for hate speech detection in tweets. In: Barrett, R., Cummings, R., Agichtein, E. & Gabrilovich, E. (Hrsg.), *Proceedings of the 26th International Conference on World Wide Web.* Genf: International World Wide Web Conferences Steering Committee, S. 759–760.

Bail, C. A. (2014): The cultural environment: Measuring culture with big data. *Theory and Society, 43*(3/4), S. 465–482.

Basov, N. & Kholodova, D. (2021): Networks of context. Three-layer socio-cultural mapping for a Verstehende network analysis. *Social Networks,* 69, S. 84–101.

Bauernschmidt, S. (2020): Konturen kulturwissenschaftlicher Inhaltsanalyse. *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research,* 21(1).

Baur, N. & Blasius, J. (2018): *Handbuch Methoden der empirischen Sozialforschung* (2. Auflage). Wiesbaden: Springer.

Baur, N., Graeff, P., Braunisch, L. & Schweia, M. (2020): The Quality of Big Data. Development, Problems, and Possibilities of Use of Process-Generated Data in the Digital Age. *Historical Social Research/Historische Sozialforschung,* 45(3), S. 209–243.

Beck, K. (2010): Soziologie der Online-Kommunikation. In: Schweiger, W. & Beck, K. (Hrsg.), *Handbuch Online-Kommunikation.* Wiesbaden: VS Verlag, S. 15–35.

Benvenuto, N. & Piazza, F. (1992): On the complex backpropagation algorithm. *IEEE Transactions on Signal Processing,* 40(9), S. 967–969.

Bick, W. & Müller, P. J. (1984): Sozialwissenschaftliche Datenkunde für prozeßproduzierte Daten: Entstehungsbedingungen und Indikatorenqualität. In: Bick, W., Mann, R. & Müller, P. J. (Hrsg.), *Sozialforschung und Verwaltungsdaten.* Stuttgart: Klett-Cotta, S. 123–159.

Blasius, J. & Schmitz, A. (2013): Sozialraum- und Habituskonstruktion. Die Korrespondenzanalyse in Pierre Bourdieus Forschungsprogramm. In: Lenger, A., Schneickert, C. & Schumacher, F. (Hrsg.), *Pierre Bourdieus Konzeption des Habitus: Grundlagen, Zugänge, Forschungsperspektiven.* Wiesbaden: Springer, S. 201–218.

Blei, D. M. (2012): Probabilistic topic models. *Communications of the ACM,* 55(4), S. 77–84.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003): Latent dirichlet allocation. *The Journal of Machine Learning Research,* 3(0), S. 993–1022.

Bohnsack, R., Krüger, H.-H. & Pfaff, N. (2013): Einleitung: Rekonstruktive Milieuforschung. *Zeitschrift für Qualitative Forschung,* 14(2), S. 171–178.

Bonn, A., Richter, A., Vohland, K., Pettibone, L., Miriam, B., Feldmann, R., Goebel, C., Grefe, C., Hecker, S., Hennen, L., Hofer, H., Kiefer, S., Klotz, S., Kluttig, T., Krause, J., Küsel, K., Liedtke, C., Mahla, A., Neumeier, V., PremkeKraus, M., Rillig, M. C., Röller, O., Schäffler, L., Schmalzbauer, B., Schneidewind, U., Schumann, A., Settele, J., Tochtermann, K., Tockner, K., Vogel, J., Volkmann, W., von Unger, H., Walter, D., Weisskopf, M., Wirth, C., Witt, T., Wolst, D. & Ziegler, D. (2016): *Grünbuch Citizen Science Strategie 2020 für Deutschland.* Berlin: Projekt „Bürger schaffen Wissen – Wissen schafft Bürger" (GEWISS), https://www.buergerschaffenwissen.de/sites/default/files/assets/dokumente/gewiss-gruenbuch_citizen_science_strategie.pdf

Borchardt, A. & Göthlich, S. E. (2009): Erkenntnisgewinnung durch Fallstudien. In: Albers, S., Klapper, D., Konradt, U., Walter, A. & Wolf, J. (Hrsg.), *Methodik der empirischen Forschung.* Wiesbaden: Gabler, S. 33–48.

Bosco, C., Patti, V., Bogetti, M., Conoscenti, M., Ruffo, G. F., Schifanella, R. & Stranisci, M. (2017): Tools and resources for detecting hate and prejudice against immigrants in social media. In: AISB Consortium (Hrsg.), *Proceedings of First Symposium on Social Interactions in Complex Intelligent Systems (SICIS), AISB Convention 2017, AI and Society.* Bath: AISB Consortium, S. 79–84.

Bourdieu, P. (2010): *Homo academicus.* Frankfurt am Main: Suhrkamp.

Bourdieu, P. (2012): *Praktische Vernunft: Zur Theorie des Handelns.* Frankfurt am Main: Suhrkamp.

Bourdieu, P. (2016): *Die feinen Unterschiede: Kritik der gesellschaftlichen Urteilskraft* (25. Auflage). Frankfurt am Main: Suhrkamp.

Braxton, J. M., Milem, J. F. & Sullivan, A. S. (2000): The Influence of Active Learning on the College Student Departure Process. Toward a Revision of Tinto's Theory. *The Journal of Higher Education, 71*(5), S. 569–590.

Breiger, R. L., Wagner-Pacifici, R. & Mohr, J. W. (2018): Capturing distinctions while mining text data: Toward low-tech formalization for text analysis. *Poetics,* 68, S. 104–119.

Bremer, H. & Teiwes-Kügler, C. (2013): Habitusanalyse als Habitus-Hermeneutik. *ZQF – Zeitschrift für Qualitative Forschung,* 14(2), S. 199–219.

Browne, M. W. (2000): Cross-validation methods. *Journal of mathematical psychology,* 44(1), S. 108–132.

Buneman, P. (1997): Semistructured data. In: Mendelzon, A. & Özsoyoglu, Z. (Hrsg.), *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems.* Tucson: Accociation for Computing Machinery, S. 117–121.

Burzan, N. (2016): *Methodenplurale Forschung: Chancen und Probleme von Mixed Methods.* Weinheim, Basel: Beltz Juventa.

Cao, K., Liu, Y., Meng, G. & Sun, Q. (2020): An overview on edge computing research. *IEEE Access,* 8, S. 85714–85728.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L. & Blei, D. M. (2009): Reading Tea Leaves: How Humans Interpret Topic Models. In: Bengio, Y., Schuurmans, D., Lafferty, J. D., Williams, C. K. I. & Culotta, A. (Hrsg.), *Proceedings of the 22nd International Conference on Neural Information Processing Systems (NIPS).* Red Hook: Curran Associates, S. 288–296.

Charmaz, K. C. (2011): Den Standpunkt verändern: Methoden der konstruktivistischen Grounded Theory. In: Mey, G. & Mruck, K. (Hrsg.), *Grounded Theory Reader.* Wiesbaden: VS Verlag, S. 181–205.

Chen, P. P.-S. (1976): The entity-relationship model. Toward a unified view of data. *ACM transactions on database systems (TODS),* 1(1), S. 9–36.

Chomsky, N. (2009): *Syntactic structures.* Berlin: de Gruyter.

Collins, R. (1990): Conflict Theory and the Advance of Macro-Historical Sociology. In: Ritzer, G. (Hrsg.), *Frontiers of Social Theory. The New Syntheses.* New York: Columbia University Press, S. 68–87.

Cosentino, V., Izquierdo, J. L. C. & Cabot, J. (2017): A systematic mapping study of software development with GitHub. *IEEE Access, 5,* S. 7173–7192.

Coser, L. (1957): Social conflict and the theory of social change. *The British Journal of Sociology, 8*(3), S. 197–207.

Courtois, C., Slechten, L. & Coenen, L. (2018): Challenging Google Search filter bubbles in social and political information. Disconforming evidence from a digital methods case study. *Telematics and Informatics,* 35(7), S. 2006–2015.

Creswell, J. W. & Plano Clark, V. L. (2011): *Designing and conducting mixed methods research* (2. Auflage). Los Angeles: Sage.

Crichton, M. (1999): Ritual Abuse, Hot Air, and Missed Opportunities. *Science,* 283(5407), S. 1461–1463.

Curran, B., Higham, K., Ortiz, E. & Vasques Filho, D. (2018): Look who's talking: Two-mode networks as representations of a topic model of New Zealand parliamentary speeches. *PLOS ONE, 13*(6), e0199072.

Daenekindt, S. & Huisman, J. (2020): Mapping the scattered field of research on higher education. A correlated topic model of 17,000 articles, S. 1991–2018. *Higher Education,* 80(3), S. 571–597.

Dahm, G. & Lauterbach, O. (2016): Measuring Students' Social and Academic Integration. Assessment of the Operationalization in the National Educational Panel Study. In: Blossfeld, H.-P., von Maurice, J., Bayer, M. & Skopek, J. (Hrsg.), *Methodological Issues of Longitudinal Surveys. The Example of the National Educational Panel Study.* Wiesbaden: Springer, S. 313–329.

Davies, M. (2010): The Corpus of Contemporary American English as the first reliable monitor corpus of English. *Literary and linguistic computing,* 25(4), S. 447–464.

Dehler-Holland, J., Schumacher, K. & Fichtner, W. (2021): Topic Modeling Uncovers Shifts in Media Framing of the German Renewable Energy Act. *Patterns,* 2(1), S. 100169.

Deng, L. & Liu, Y. (2018): *Deep Learning in Natural Language Processing.* Singapore: Springer Singapore.

Denzin, N. (2004): Reading Film. In: Flick, U., von Kardorff, E. & Steinke, I. (Hrsg.), *A Companion to Qualitative Research.* London: Sage, S. 237–242.

Devi, M. D. & Saharia, N. (2020): Exploiting Topic Modeling to classify sentiment from lyrics. In: Bhattacharjee, A., Borgohain, S., Verma, G., Gao, X. (Hrsg.), *Proceedings of the 2nd International Conference on Machine Learning, Image Processing, Network Security and Data Sciences.* Wiesbaden: Springer, S. 411–423.

Diaz-Bone, R. (2006): Zur Methodologisierung der Foucaultschen Diskursanalyse. *Forum qualitative Sozialforschung,* 7(1).

Diaz-Bone, R. (2015): Gütekriterien der quantitativen Sozialforschung. In: Diaz-Bone, R. & Weischer, C. (Hrsg.), *Methoden-Lexikon für die Sozialwissenschaften.* Wiesbaden: Springer, S. 169.

Diaz-Bone, R. (2019): Formen des Schließens und Erklärens. In: Baur, N. & Blasius, J. (Hrsg.), *Handbuch Methoden der empirischen Sozialforschung.* Wiesbaden: Springer, S. 49–66.

Diaz-Bone, R., Horvath, K. & Cappel, V. (2020): Social Research in Times of Big Data. The Challenges of New Data Worlds and the Need for a Sociology of Social Research. *Historical Social Research/Historische Sozialforschung,* 45(3), S. 314–341.

Dillon, L., Neo, L. S. & Freilich, J. D. (2020): A comparison of ISIS foreign fighters and supporters social media posts: An exploratory mixed-method content analysis. *Behavioral Sciences of Terrorism and Political Aggression,* 12(4), S. 268–291.

DiMaggio, P., Nag, M. & Blei, D. (2013): Exploiting affinities between topic modeling and the sociological perspective on culture: Application to newspaper coverage of US government arts funding. *Poetics,* 41(6), S. 570–606.

Dinov, I. (2018): Black box machine-learning methods: Neural networks and support vector machines. In: Dinov, I. (Hrsg.), *Data Science and Predictive Analytics.* Wiesbaden: Springer, S. 383–422.

Dolata, U. (2018): *Privatisierung, Kuratierung, Kommodifizierung: Kommerzielle Plattformen im Internet.* SOI Discussion Paper. https://www.sowi.uni-stuttgart.de/dokumente/forschung/soi/soi_2018_4_Dolata.Kommerzielle.Plattformen.im.Internet.pdf

Duriau, V. J., Reger, R. K. & Pfarrer, M. D. (2007): A Content Analysis of the Content Analysis Literature in Organization Studies. Research Themes, Data Sources, and Methodological Refinements. *Organizational Research Methods,* 10(1), S. 5–34.

Ebbinghaus, B. (2005): When Less is More. Selection Problems in Large-N and Small-N Cross-National Comparisons. *International Sociology,* 20(2), S. 133–152.

Ebbinghaus, B. (2009): Vergleichende Politische Soziologie: Quantitative Analyse- oder qualitative Fallstudiendesigns? In: Kaina, V. & Römmele, A. (Hrsg.), *Politische Soziologie.* Wiesbaden: VS Verlag, S. 481–501.

Edelmann, A. & Mohr, J. W. (2018): Formal studies of culture: Issues, challenges, and current trends. *Poetics,* 68, S. 1–9.

Ellis, C., Adams, T. E. & Bochner, A. P. (2010): Autoethnography. An Overview. *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research,* 12(1).

Elmasri, R., Weeldreyer, J. & Hevner, A. (1985): The category concept: An extension to the entity-relationship model. *Data & Knowledge Engineering,* 1(1), S. 75–116.

Elo, S. & Kyngäs, H. (2008): The qualitative content analysis process. *Journal of Advanced Nursing,* 62(1), S. 107–115.

Fairclough, N. (2013): *Critical discourse analysis: The critical study of language.* New York, London: Routledge.

Fawcett, P., Jensen, M. J., Ransan-Cooper, H. & Duus, S. (2019): Explaining the „ebb and flow" of the problem stream: Frame conflicts over the future of coal seam gas („fracking") in Australia. *Journal of Public Policy,* 39(3), S. 521–541.

Fay, M. P. & Proschan, M. A. (2010): Wilcoxon-Mann-Whitney or t-test? On assumptions for hypothesis tests and multiple interpretations of decision rules. *Statistics surveys,* 4(1), S. 1–39.

Feyerabend, P. (1980): *Erkenntnis für freie Menschen.* Frankfurt am Main: Suhrkamp.

Findler, N. (1979): A heuristic information retrieval system based on associative networks. In: Findler, N. (Hrsg.), *Associative Networks.* London, Oxford, Boston, New York, San Diego: Academic Press, S. 305–326.

Fink, A. (2019): *Conducting Research Literature Reviews. From the Internet to Paper.* Los Angeles: Sage.

Fleck, L. (2019): *Entstehung und Entwicklung einer wissenschaftlichen Tatsache. Einführung in die Lehre vom Denkstil und Denkkollektiv.* Frankfurt am Main: Suhrkamp.

Flick, U. (2019): Gütekriterien qualitativer Sozialforschung. In: Baur, N. & Blasius, J. (Hrsg.), *Handbuch Methoden der empirischen Sozialforschung.* Wiesbaden: Springer, S. 473–488.

Foucault, M. (2017): *Sicherheit, Territorium, Bevölkerung: Vorlesung am Collège de France, 1977–1978.* Frankfurt am Main: Suhrkamp.

Frayling, C. (2005): *Mad, Bad and Dangerous: The Scientist and the Cinema.* London: Reaktion.

Früh, W. (2011): *Inhaltsanalyse: Theorie und Praxis* (7., überarbeitete Auflage). Konstanz: UVK.

Früh, W. (2017): *Inhaltsanalyse: Theorie und Praxis* (9., überarbeitete Auflage). Konstanz: UVK.

Fuhse, J., Stuhler, O., Riebling, J. & Martin, J. L. (2020): Relating social and symbolic relations in quantitative text analysis. A study of parliamentary discourse in the Weimar Republic. *Poetics,* 78, S. 101363.

Geertz, C. (2002 [1983]): *Dichte Beschreibung. Beiträge zum Verstehen kultureller Systeme.* Frankfurt am Main: Suhrkamp.

Ghanem, B., Karoui, J., Benamara, F., Rosso, P. & Moriceau, V. (2020): Irony Detection in a Multilingual Context. In: Jose, J. M., Yilmaz, E., Magalhães, J., Castells, P., Ferro, N., Silva, M. J. & Martins, F. (Hrsg.), *Advances in Information Retrieval.* Wiesbaden: Springer, S. 141–149.

Giraudel, J. & Lek, S. (2001): A comparison of self-organizing map algorithm and some conventional statistical methods for ecological community ordination. *Ecological Modelling,* 146(1–3), S. 329–339.

Glaser, B. G. & Strauss, A. L. (2008): *Grounded theory: Strategien qualitativer Forschung* (1. Nachdruck der 2., korrigierten Auflage). Bern: Huber.

Gläser, J. & Laudel, G. (2010): *Experteninterviews und Qualitative Inhaltsanalyse.* Wiesbaden: VS Verlag.

Gold, Z. & Latonero, M. (2017): Robots Welcome: Ethical and Legal Considerations for Web Crawling and Scraping. *Washington Jounral of Law, Technology & Arts,* 13(3), S. 275.

Graeff, P. & Baur, N. (2020): Digital Data, Administrative Data, and Survey Compared: Updating the Classical Toolbox for Assessing Data Quality of Big Data, Exemplified by the Generation of Corruption Data. *Historical Social Research,* 45(3). S. 244–269.

Graneheim, U. H. & Lundman, B. (2004): Qualitative content analysis in nursing research: Concepts, procedures and measures to achieve trustworthiness. *Nurse Education Today,* 24(2), S. 105–112.

Graneheim, U. H., Lindgren, B.-M. & Lundman, B. (2017): Methodological challenges in qualitative content analysis: A discussion paper. *Nurse Education Today,* 56, S. 29–34.

Grimmer, J. & Stewart, B. M. (2013): Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts. *Political Analysis,* 21(3), S. 267–297.

Grosman, J. & Reigeluth, T. (2019): Perspectives on algorithmic normativities. Engineers, objects, activities. *Big Data & Society,* 6(2), S. 1–12.

Grothe-Hammer, M. & Kohl, S. (2020): The decline of organizational sociology? An empirical analysis of research trends in leading journals across half a century. *Current Sociology,* 68(4), S. 419–442.

Grudniewicz, A., Moher, D., Cobey, K. D., Bryson, G. L., Cukier, S., Allen, K., … Lalu, M. M. (2019): Predatory journals. No definition, no defence. *Nature,* 576(7786), S. 210–212.

Guo, C., Lu, M. & Wei, W. (2021): An Improved LDA Topic Modeling Method Based on Partition for Medium and Long Texts. *Annals of Data Science,* 8(2), S. 331–344.

Guy, G. (2013): The cognitive coherence of sociolects: How do speakers handle multiple sociolinguistic variables? *Journal of pragmatics,* 52, S. 63–71.

Häder, M. & Häder, S. (2019): Stichprobenziehung in der quantitativen Sozialforschung. In: Baur, N. & Blasius, J. (Hrsg.), *Handbuch Methoden der empirischen Sozialforschung.* Wiesbaden: Springer, S. 333–348.

Händel, M., Bedenlier, S., Gläser-Zikuda, M., Kammerl, R., Kopp, B. & Ziegler, A. (2020a): *Do Students have the Means to Learn During the Coronavirus Pandemic? Student Demands for Distance Learning in a Suddenly Digital Landscape.* PsyArXiv. https://doi.org/10.31234/osf.io/5ngm9

Händel, M., Stephan, M., Gläser-Zikuda, M., Kopp, B., Bedenlier, S. & Ziegler, A. (2020b): *Digital readiness and its effects on higher education student socio-emotional experiences in the context of COVID-19 pandemic.* PsyArXiv. https://doi.org/10.31234/osf.io/b9pg7

Hao, J. & Ho, T. K. (2019): Machine learning made easy: A review of scikit-learn package in python programming language. *Journal of Educational and Behavioral Statistics,* 44(3), S. 348–361.

Hartmann, M. (2010): Die Exzellenzinitiative und ihre Folgen. *Leviathan,* 38(3), S. 369–387.

Harzing, A.-W. (2019): Publish or Perish (Version 7) [Windows]. London: Harzing.com. https://harzing.com/resources/publish-or-perish

Heiberger, R. H. & Riebling, J. R. (2016): Installing computational social science: Facing the challenges of new information and communication technologies in social science. *Methodological Innovations,* 9, S. 1–11.

Heidenreich, T., Eberl, J.-M., Lind, F. & Boomgaarden, H. (2020): Political migration discourses on social media: A comparative perspective on visibility and sentiment across political Facebook accounts in Europe. *Journal of Ethnic and Migration Studies,* 46(7), S. 1261–1280

Hjellbrekke, J. (2019): *Multiple correspondence analysis for the social sciences.* New York: Routledge.

Hoffman, M., Bach, F. R. & Blei, D. M. (2010): Online Learning for Latent Dirichlet Allocation. In: Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R. & Culotta, A. (Hrsg.), *Proceedings of the 21st Conference on Advances in Neural Information Processing Systems 23.* Red Hook: Curran Associates, S. 856–864.

Holton, J. (2007): The Coding Process and Its Challenges. In: Bryant, A. & Charmaz, K. (Hrsg.), *The Sage Handbook of Grounded Theory.* London: Sage, S. 265–289.

Hsieh, H.-F. & Shannon, S. E. (2005): Three Approaches to Qualitative Content Analysis. *Qualitative Health Research,* 15(9), S. 1277–1288.

Hu, Y., John, A., Wang, F. & Kambhampati, S. (2012): Et-lda: Joint topic modeling for aligning events and their twitter feedback. In: Association for the Advancement of Artificial Intelligence (Hrsg.), *Proceedings of the 26th AAAI Conference on Artificial Intelligence.* AAAI Press: Toronto, S. 59–65.

Hunter, J. D. (2007): Matplotlib: A 2D graphics environment. *Computing in science & engineering,* 9(3), S. 90–95.

Hutter, S. (2020): Quantitative Inhaltsanalyse. In: Wagemann, C., Goerres, A. & Siewert, M. B. (Hrsg.), *Handbuch Methoden der Politikwissenschaft.* Wiesbaden: Springer, S. 837–859.

Hutto, C. & Gilbert, E. (2014): Vader: A parsimonious rule-based model for sentiment analysis of social media text. In: Adar, E. & Resnick, P. (Hrsg.), *8th International Conference on Weblogs and Social Media (ICWSM-14).* AAAI Press: Ann Arbor, S. 216–225.

Jaton, F. (2021): Assessing biases, relaxing moralism. On ground-truthing practices in machine learning design and application. *Big Data & Society,* 8(1), S. 1–15.

Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y. & Zhao, L. (2019): Latent Dirichlet allocation (LDA) and topic modeling: Models, applications, a survey. *Multimedia Tools and Applications,* 78(11), S. 15169–15211.

Jivani, A. G. (2011): A comparative study of stemming algorithms. *International Journal for Computer Applications in Technology,* 2(6), S. 1930–1938.

Jones, K. (1994): Natural language processing: a historical review. In: Zampolli, A., Calzolari, N. & Palmer, M. (Hrsg.), *Current issues in computational linguistics:* in *honour of Don Walker.* Berlin, Heidelberg: Springer, S. 3–16.

Kelle, U. (2008): *Die Integration qualitativer und quantitativer Methoden in der empirischen Sozialforschung. Theoretische Grundlagen und methodologische Konzepte.* Wiesbaden: Springer.

Kelle, U. (2019): Mixed Methods. In: Baur, N. & Blasius, J. (Hrsg.), *Handbuch Methoden der empirischen Sozialforschung* (2. Auflage). Wiesbaden: Springer, S. 159–172.

Kitchin, R. & McArdle, G. (2016): What makes Big Data, Big Data? Exploring the ontological characteristics of 26 datasets. *Big Data & Society, 3*(1), S. 1–10.

Knoblauch, H., Baur, N., Traue, B. & Akremi, L. (2018): Was heißt „Interpretativ forschen"? In: Akremi, L., Baur, N., Knoblauch, H. & Traue, B. (Hrsg.) (2018): *Handbuch interpretativ forschen.* Weinheim, Basel: Beltz Juventa, S. 9–36.

Kozlowski, A. C., Taddy, M. & Evans, J. A. (2019): The geometry of culture: Analyzing the meanings of class through word embeddings. *American Sociological Review,* 84(5), S. 905–949.

Kracauer, S. (1952): The Challenge of Qualitative Content Analysis. *Public Opinion Quarterly,* 16(4), S. 631–642.

Krebs, D. & Menold, N. (2019): Gütekriterien quantitativer Sozialforschung. In: Baur, N. & Blasius, J. (Hrsg.), *Handbuch Methoden der empirischen Sozialforschung* (2. Auflage). Wiesbaden: Springer, S. 489–504.

Kreulich, K., Lichtlein, M., Zitzmann, C., Bröcker, T., Schwab, R. & Zinger, B. (2020): *Hochschullehre in der Post-Corona-Zeit. Studie der bayerischen Hochschulen für angewandte Wissenschaften.* Forschungs- und Innovationslabor Digitale Lehre (FIDL), https://w3-mediapool.hm.edu/mediapool/media/baukasten/img_2/fidl/dokumente_121/FIDLStudiePostCoronaGesamt.pdf.

Krippendorff, K. (2004): *Content Analysis: An Introduction to Its Methodology.* Thousand Oaks, California: Sage.

Krotov, V. & Tennyson, M. (2018): Research note: Scraping financial data from the web using the R language. *Journal of Emerging Technologies in Accounting,* 15(1), S. 169–181.

Krotov, V., Johnson, L. & Silva, L. (2020): Tutorial: Legality and Ethics of Web Scraping. *Communications of the Association for Information Systems,* 47(1), S. 22.

Kuckartz, U. (2018): *Qualitative Inhaltsanalyse. Methoden, Praxis, Computerunterstützung* (4., überarbeitete Auflage). Weinheim, Basel: Beltz.

Kupietz, M., Belica, C., Keibel, H. & Witt, A. (2010): *The German Reference Corpus DeReKo: A primordial sample for linguistic research. Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC'10).* Valletta: European Language Resources Association, S. 1848–1854.

Lamnek, S. & Krell, C. (2016): *Qualitative Sozialforschung. Mit Online-Material* (6., überarbeitete Auflage). Weinheim, Basel: Beltz.

Lauer, C., Brumberger, E. & Beveridge, A. (2018): Hand collecting and coding versus data-driven methods in technical and professional communication research. *IEEE Transactions on Professional Communication,* 61(4), S. 389–408.

Laver, M., Benoit, K. & Garry, J. (2003): Extracting Policy Positions from Political Texts Using Words as Data. *The American Political Science Review,* 97(2), S. 311–331.

Lê, S., Josse, J. & Husson, F. (2008): FactoMineR: an R package for multivariate analysis. *Journal of statistical software,* 25(1), S. 1–18.

Leifeld, P. (2020): Policy Debates and Discourse Network Analysis: A Research Agenda. *Politics and Governance,* 8(2), S. 180–183.

Lersch, E., Stöber, R. (2008): Quellenüberlieferung und Quellenrecherche. In: Arnold, K., Behmer, M. & Semrad, B. (Hrsg.), *Kommunikationsgeschichte. Positionen und Werkzeuge; ein diskursives Hand- und Lehrbuch.* Berlin: Lit, S. 289–322.

Lin, Y., Michel, J., Aiden, E., Orwant, J., Brockman, W. & Petrov, S. (2012): Syntactic annotations for the google books ngram corpus. In: Li, H., Lin, C.-Y., Osborne, M. L, Lee, G. & Park, J. (Hrsg.), *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics.* Jeju: Association for Computational Linguistics, S. 169–174.

Liu, B. (2012): Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies,* 5(1), S. 1–167.

Loper, E. & Bird, S. (2002): *NLTK: The natural language toolkit.* arXiv preprint cs/0205028.

Lörz, M., Marczuk, A., Zimmer, L., Multrus, F. & Buchholz, S. (2020): Studieren unter Corona – Bedingungen: Studierende bewerten das erste Digitalsemester. *DZHW Brief.* https://doi.org/10.34878/2020.05.DZHW_BRIEF

Luhmann, N. (1984): *Soziale Systeme. Grundriß einer allgemeinen Theorie.* Frankfurt am Main: Suhrkamp.

Maeße, J. & Sparsam, J. (2017): Die Performativität der Wirtschaftswissenschaft. In: Maurer, A. (Hrsg.), *Handbuch der Wirtschaftssoziologie.* Wiesbaden: Springer, S. 181–195.

Mai, M. & Winter, R. (2006): Kino, Gesellschaft und soziale Wirklichkeit. Zum Verhältnis von Soziologie und Film. In: Mai, M. & Winter, R. (Hrsg.), *Das Kino der Gesellschaft – die Gesellschaft des Kinos: interdisziplinäre Positionen, Analysen und Zugänge.* Köln: Halem, S. 7–23.

Mannheim, K. (1980): *Strukturen des Denkens.* Frankfurt am Main: Suhrkamp.

Manning, C. D. & Schütze, H. (1999): *Foundations of Statistical Natural Language Processing.* Cambridge: MIT Press.

Manning, C. D. (2011): Part-of-speech tagging from 97 % to 100 %: Is it time for some linguistics? In: Gebulkh, A. (Hrsg.), *Proceedings of the 12th International Conference on Computational Linguistics and Intelligent Text Processing(CICLing).* Berlin, Heidelberg: Springer, S. 171–189.

Mäntylä, M. V., Graziotin, D. & Kuutila, M. (2018): The evolution of sentiment analysis – A review of research topics, venues, and top cited papers. *Computer Science Review,* 27, S. 16–32.

Martin, F. & Johnson, M. (2015): More efficient Topic Modeling through a noun only approach. In: Hachey, B. & Webster, K. (Hrsg.), Proceedings of the Australasian Language Technology Association Workshop 2015. Parramatta: Association for Computing Machinery, S. 111–115.

Martín-Martín, A., Thelwall, M., Orduna-Malea, E. & Delgado López-Cózar, E. (2021): Google Scholar, Microsoft Academic, Scopus, Dimensions, Web of Science, and OpenCitations' COCI: A multidisciplinary comparison of coverage via citations. *Scientometrics,* 126(1), S. 871–906.

MAXQDA (2020): *MAXQDA 2020 Manual.* www.maxqda.de/download/manuals/MAX2020-Online-Manual-Complete-DE.pdf

Mayring, P. (2010): *Qualitative Inhaltsanalyse: Grundlagen und Techniken* (11., aktualisierte und überarbeite Auflage). Weinheim, Basel: Beltz.

Mayring, P. (2015): Qualitative Content Analysis: Theoretical Background and Procedures. In: Bikner-Ahsbahs, A., Knipping, C., Presmeg, N. (Hrsg.), *Approaches to Qualitative Research in Mathematics Education.* Wiesbaden: Springer, S. 365–380.

McKinney, W. (2011): pandas: A foundational Python library for data analysis and statistics. *Python for high performance and scientific computing,* 14(9), S. 1–9.

Meuser, M. & Nagel, U. (2002): ExpertInneninterviews – Vielfach erprobt, wenig bedacht. In: Bogner, A., Littig, B. & Menz, W. (Hrsg.), *Das Experteninterview.* Wiesbaden: VS Verlag, S. 71–93.

Miles, M. B. & Huberman, A. M. (1994): *Qualitative data analysis: An expanded sourcebook* (2. Auflage). Thousand Oaks, California: Sage.
Miller, G. A. (1995): WordNet. A lexical database for English. *Communications of the ACM,* 38(11), S. 39–41.
Mills, K. A. (2018): What are the threats and potentials of big data for qualitative research? *Qualitative Research,* 18(6), S. 591–603.
Mimno, D., Wallach, H. M., Talley, E., Leenders, M. & McCallum, A. (2011): Optimizing semantic coherence in topic models. In: Merlo, P., Barzilay, R. & Johnson, M. (Hrsg.), *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing.* Stroudsburg: Association for Computational Linguistics, S. 262–272.
Misoch, S. (2015): *Qualitative Interviews.* Berlin: de Gruyter.
Mohr, J. W. & Bogdanov, P. (2013): Introduction – Topic models. What they are and why they matter. *Poetics,* 41(6), S. 545–569.
Mohr, J. W. (1998): Measuring Meaning Structures. *Annual Review of Sociology,* 24(1), S. 345–370.
Mohr, J. W., Wagner-Pacifici, R. & Breiger, R. L. (2015): Toward a computational hermeneutics. *Big Data & Society,* 2(2), S. 1–8.
Molina, M. & Garip, F. (2019): Machine learning for sociology. *Annual Review of Sociology,* 45, S. 27–45.
Moretti, F. (2000): Conjectures on World Literature. *New Left Review,* 1, S. 54–68.
Moretti, F. (2013): *Distant reading.* London, New York: Verso.
Morin, E. (2005): *The Cinema, or the Imaginary Man.* Minneapolis: University of Minnesota Press.
Morris, R. (1994): Computerized content analysis in management research. A demonstration of advantages & limitations. *Journal of Management,* 20(4), S. 903–931.
Müller, L. & Braun, E. (2018): Student Engagement. *Zeitschrift für Erziehungswissenschaft,* 21(3), S. 649–670.
Münch, R. (1986): *Die Kultur der Moderne: Ihre Grundlagen und ihre Entwicklung in England und Amerika* (Bd. 1). Frankfurt am Main: Suhrkamp.
Münch, R. (2007): *Die akademische Elite. Zur sozialen Konstruktion wissenschaftlicher Exzellenz.* Frankfurt am Main: Suhrkamp.
Münch, R. (2014): *Academic capitalism. Universities in the global struggle for excellence.* New York: Routledge.
Muno, W. (2009): Fallstudien und die vergleichende Methode. In: Pickel, S., Pickel, G., Lauth, H.-J. & Jahn, D. (Hrsg.), *Methoden der vergleichenden Politik- und Sozialwissenschaft.* Wiesbaden: VS Verlag, S. 113–131.
Munoz-Najar Galvez, S., Heiberger, R. & McFarland, D. (2019): Paradigm Wars Revisited. A Cartography of Graduate Research in the Field of Education (1980–2010). *American Educational Research Journal,* 57(2), S. 612–652.
Murphy, K. P. (2012): *Machine learning: A probabilistic perspective.* Cambridge: MIT press.
Myrtveit, I., Stensrud, E. & Shepperd, M. (2005): Reliability and validity in comparative studies of software prediction models. *IEEE Transactions on Software Engineering,* 31(5), S. 380–391.
Napier, K. & Shamir, L. (2018): Quantitative Sentiment Analysis of Lyrics in Popular Music. *Journal of Popular Music Studies,* 30(4), S. 161–176.
Neuendorf, K. A. (2002): *The Content Analysis Guidebook.* Thousand Oaks, California: Sage.
Newman, M. L., Pennebaker, J. W., Berry, D. S. & Richards, J. M. (2003): Lying Words: Predicting Deception from Linguistic Styles. *Personality and Social Psychology Bulletin,* 29(5), S. 665–675.

Nguyen, T. H. & Shirai, K. (2015): Topic modeling based sentiment analysis on social media for stock market prediction. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Procession,* S. 1354–1364.

Nicolae, S., Endreß, M., Berli, O. & Bischur, D. (Hrsg.) (2019): *(Be)Werten: Beiträge zur sozialen Konstruktion von Wertigkeit.* Wiesbaden: Springer.

Niehr, T. (2014): *Einführung in die Politolinguistik. Gegenstände und Methoden.* München: UTB.

Niyogi, P., Smale, S. & Weinberger, S. (2011): A topological view of unsupervised learning from noisy data. *SIAM Journal on Computing,* 40(3), S. 646–663.

Obar, J. A. & Oeldorf-Hirsch, A. (2020): The biggest lie on the internet: Ignoring the privacy policies and terms of service policies of social networking services. *Information, Communication & Society,* 23(1), S. 128–147.

Obermeyer, Z. & Emanuel, E. J. (2016): Predicting the future – Big data, machine learning, and clinical medicine. *The New England Journal of Medicine,* 375(13), S. 1216.

Oevermann, U. (2013): Objektive Hermeneutik als Methodologie der Erfahrungswissenschaften von der sinnstrukturierten Welt. In: Langer, P. C., Kühner, A. & Schweder, P. (Hrsg.), *Reflexive Wissensproduktion. Anregungen zu einem kritischen Methodenverständnis in qualitativer Forschung.* Wiesbaden: Springer, S. 69–98.

Oliphant, T. (2006): *A guide to NumPy* (Bd. 1). https://ecs.wgtn.ac.nz/foswiki/pub/Support/ManualPagesAndDocumentation/numpybook.pdf

Orduna-Malea, E., Martín-Martín, A. & Delgado López-Cózar, E. (2017): Google Scholar as a source for scholarly evaluation. A bibliographic review of database errors. *Revista Española de Documentación Científica,* 40(4), e185.

Östling, J. (2020): Humboldt's University: The History and Topicality of a German Tradition. In: Engwall, L. (Hrsg.), *Missions of Universities.* Wiesbaden: Springer, S. 63–80.

Otter, D. W., Medina, J. R. & Kalita, J. K. (2020): A survey of the usages of deep learning for natural language processing. *IEEE Transactions on Neural Networks and Learning Systems,* 32(2), S. 604–624.

Ozduzen, O., Korkut, U. & Ozduzen, C. (2020): ‚Refugees are not welcome': Digital racism, online place-making and the evolving categorization of Syrians in Turkey. *new media & society,* 23(11), S. 3349–3369.

Pang, B. & Lee, L. (2008): Opinion Mining and Sentiment Analysis. *Foundations and Trends® in Information Retrieval,* 2(1/2), S. 1–135.

Papilloud, C. & Hinneburg, A. (2018): *Qualitative Textanalyse mit Topic-Modellen.* Wiesbaden: Springer.

Parsons, T. (1968): *The Structure of Social Action.* New York: Free Press.

Pennebaker, J. W., Boyd, R. L., Jordan, K. & Blackburn, K. (2015): *The development and psychometric properties of LIWC2015.* Austin, Texas: University of Texas at Austin.

Petticrew, M. & Roberts, H. (2008): *Systematic Reviews in the Social Sciences. A Practical Guide.* Hoboken: Wiley.

Plath, W. (1967): *Multiple path analysis and automatic translation.* Amsterdam, The Netherlands: North-Holland.

Popping, R. (2000): *Computer-assisted text analysis.* London und Thousand Oaks, California: Sage.

Powers, E. (2017): My news feed is filtered? Awareness of news personalization among college students. *Digital Journalism,* 5(10), S. 1315–1335.

Prasad, B. D. (2019): Qualitative Content Analysis: Why is it Still a Path Less Taken? *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research,* 20(3).

Przyborski, A. & Wohlrab-Sahr, M. (2014): *Qualitative Sozialforschung: Ein Arbeitsbuch.* München: de Gruyter.

Rabe-Hesketh, S. & Skrondal, A. (2012): *Multilevel and longitudinal modeling using Stata* (3. Auflage). College Station, Texas: Stata Press Publication.

Reichertz, J. (2016): *Qualitative und interpretative Sozialforschung. Eine Einladung.* Wiesbaden: Springer.

Resnik, P., Garron, A. & Resnik, R. (2013): Using topic modeling to improve prediction of neuroticism and depression in college students. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing,* S. 1348–1353.

Riebling, J. R. (2018): The medium data problem in social science. In: Stuetzer, C. M., Welker, M. & Egger, M. (Hrsg.), *Computational Social Science in the Age of Big Data. Concepts, Methodologies, Tools, and Applications.* Köln: Halem, S. 76–100.

Roberts, M. E., Stewart, B. M., Tingley, D., Lucas, C., Leder-Luis, J., Gadarian, S. K., … Rand, D. G. (2014): Structural Topic Models for Open-Ended Survey Responses. *American Journal of Political Science, 58*(4), S. 1064–1082.

Rössler, P. (2010): *Inhaltsanalyse.* Konstanz: UVK.

Rössler, P. (2017): *Inhaltsanalyse* (3., völlig überarbeitete Auflage). Konstanz: UVK.

Sbalchiero, S. & Eder, M. (2020): Topic modeling, long texts and the best number of topics. Some Problems and solutions. *Quality & Quantity,* 54(4), S. 1095–1108.

Schank, R. & Tesler, L. (1969): A conceptual dependency parser for natural language. In: *Association for Computing Machinery, International Conference on Computational Linguistics COLING 1969.* Sånga Säby: Association for Computing Machinery, Preprint No. 2.

Schmidhuber, J. (2015): Deep learning in neural networks: An overview. *Neural networks,* 61, S. 85–117.

Schneider, G., Segadlo, N. & Leue, M. (2020): Forty-Eight Shades of Germany: Positive and Negative Discrimination in Federal Asylum Decision Making. *German Politics,* 29(4), S. 564–581.

Schneijderberg, C. & Götze, N. (2020): *Organisierte, metrifizierte und exzellente Wissenschaftler*innen. Veränderungen der Arbeits- und Beschäftigungsbedingungen an Fachhochschulen und Universitäten von 1992 über 2007 bis 2018.* Zenodo. https://doi.org/10.5281/zenodo.3949756

Schneijderberg, C. & Götze, N. (2021): Academics' Societal Engagement in Cross-country Perspective: Large-n in Small-n Comparative Case Studies. *Higher Education Policy,* 34(1), S. 1–17.

Schneiker, A., Dau, M., Joachim, J., Martin, M. & Lange, H. (2019): How to analyze social media? Assessing the promise of mixed-methods designs for studying the Twitter feeds of PMSCs. *International Studies Perspectives,* 20(2), S. 188–200.

Schreier, M. (2012): *Qualitative content analysis in practice.* Los Angeles: Sage.

Schreier, M., Stamann, C., Janssen, M., Dahl, T. & Whittal, A. (2019): Qualitative Content Analysis: Conceptualizations and Challenges in Research Practice – Introduction to the FQS Special Issue „Qualitative Content Analysis I". *Forum Qualitative Sozialforschung/ Forum: Qualitative Social Research,* 20(3).

Schulz, W. (2009): Kommunikationsprozess. In: Noelle-Neumann, E., Schulz, W. & Wilke, J. (Hrsg.), *Fischer Lexikon Publizistik, Massenkommunikation* (2. Auflage). Frankfurt am Main: Fischer Taschenbuch, S. 169–199.

Schwarz-Friesel, M. & Consten, M. (2014): *Einführung in die Textlinguistik.* Darmstadt: WBG.

Schweinitz, J. (2006): Film und Stereotyp. Eine Herausforderung für das Kino und die Filmtheorie. Berlin: Akademie.

Schwemmer, C. & Wieczorek, O. (2020): The Methodological Divide of Sociology. Evidence from Two Decades of Journal Publications. *Sociology,* 54(1), S. 3–21.
Seaver, N. (2019): Captivating algorithms: Recommender systems as traps. *Journal of Material Culture,* 24(4), S. 421–436.
Seawright, J. & Gerring, J. (2008): Case Selection Techniques in Case Study Research: A Menu of Qualitative and Quantitative Options. *Political Research Quarterly,* 61(2), S. 294–308.
Short, J. C. & Palmer, T. B. (2008): The Application of DICTION to Content Analysis Research in Strategic Management. *Organizational Research Methods,* 11(4), S. 727–752.
Short, J. C., Broberg, J. C., Cogliser, C. C. & Brigham, K. H. (2010): Construct Validation Using Computer-Aided Text Analysis (CATA). An Illustration Using Entrepreneurial Orientation. *Organizational Research Methods,* 13(2), S. 320–347.
Silge, J. & Robinson, D. (2016): tidytext: Text mining and analysis using tidy data principles in R. *Journal of Open Source Software,* 1(3)–37, S. 1–3.
Silva, B. C. & Proksch, S.-O. (2021): Politicians unleashed? Political communication on Twitter and in parliament in Western Europe. *Political Science Research and Methods,* S. 1–17.
Smelser, N. J. (2003): On Comparative Analysis, Interdisciplinarity and Internationalization in Sociology. *International Sociology,* 18(4), S. 643–657.
Solan, Z., Horn, D., Ruppin, E. & Edelman, S. (2005): Unsupervised learning of natural languages. *Proceedings of the National Academy of Sciences,* 102(33), S. 11629–11634.
Srinath, K. (2017): Python – the fastest growing programming language. *International Research Journal of Engineering and Technology (IRJET),* 4(12), S. 354–357.
Stamann, C., Janssen, M. & Schreier, M. (2016): Qualitative Inhaltsanalyse – Versuch einer Begriffsbestimmung und Systematisierung. *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research,* 17(3).
Steigleder, S. (2008): *Die strukturierende qualitative Inhaltsanalyse im Praxistest.* Marburg: Tectum.
Steinhardt, I. & İkiz-Akıncı, D. (2020): *Digitale Bildungspraktiken von Studierenden (DEPS). Daten- und Methodenbericht zur Studie DEPS.* Kassel/Hannover: DZHW. https://metadata.fdz.dzhw.eu/public/files/data-packages/stu-dps2018$-1.0.0/attachments/dps2018_Data-Methods_Report.pdf
Steinhardt, I. (2015): *Lehre stärkt Forschung: Studiengangentwicklung durch ProfessorInnen im Handlungssystem Universität.* Wiesbaden: Springer.
Steinhardt, I. (2021): Students in the spotlight. Using collaborative autoethnography to build a community of learning in the Corona crisis. *ISA Pedagogy Series,* 1(1), S. 42–59.
Steinhardt, I., Fischer, C., Heimstädt, M., Hirsbrunner, S. D., İkiz-Akıncı, D., Kressin, L., … Wünsche, H. (2020): *Das Öffnen und Teilen von Daten qualitativer Forschung: Eine Handreichung.* Berlin: Weizenbaum Institute for the Networked Society – The German Internet Institute. https://doi.org/10.34669/wi.ws/6
Steinhardt, I., Schneijderberg, C., Götze, N., Baumann, J. & Krücken, G. (2017): Mapping the quality assurance of teaching and learning in higher education. The emergence of a specialty? *Higher Education,* 74(2), S. 221–237.
Stier, S., Bleier, A., Lietz, H. & Strohmaier, M. (2018): Election campaigning on social media: Politicians, audiences, and the mediation of political communication on Facebook and Twitter. *Political communication,* 35(1), S. 50–74.
Stier, S., Posch, L., Bleier, A. & Strohmaier, M. (2017): When populists become popular: Comparing Facebook use by the right-wing movement Pegida and German political parties. *Information, Communication & Society,* 20(9), S. 1365–1388.
Stock, W. G. (2000): Was ist eine Publikation? Zum Problem der Einheitenbildung in der Wissenschaftsforschung. In: Fuchs-Kittowski, K., Laitko, H., Parthey, H. & Umstätter, W.

(Hrsg.), *Wissenschaft und Digitale Bibliothek. Wissenschaftsforschung Jahrbuch 1998.* Berlin: Gesellschaft für Wissenschaftsforschung, S. 239–282.

Stone, P. J., Dunphy, D. C. & Smith, M. S. (1966): *The general inquirer: A computer approach to content analysis.* Oxford: MIT Press.

Strauss, A. L. (2007): *Grundlagen qualitativer Sozialforschung: Datenanalyse und Theoriebildung in der empirischen soziologischen Forschung* (unveränderter Nachdruck der 2. Auflage). München: Fink.

Strong, C. (2014): The challenge of „Big Data": What does it mean for the qualitative research industry? *Qualitative Market Research* 17(4), S. 336–342.

Strübing, J., Hirschauer, S., Ayaß, R., Krähnke, U. & Scheffer, T. (2018): Gütekriterien qualitativer Sozialforschung. Ein Diskussionsanstoß. *Zeitschrift für Soziologie* 47(2). S. 83–100.

Suri, H. (2011): Purposeful Sampling in Qualitative Research Synthesis. *Qualitative Research Journal,* 11(2), S. 63–75.

Tashakkori, A. & Teddlie, C. (Hrsg.) (2003): *Handbook of mixed methods in social & behavioral research.* Thousand Oaks, California: Sage.

Tausczik, Y. R. & Pennebaker, J. W. (2010): The Psychological Meaning of Words. LIWC and Computerized Text Analysis Methods. *Journal of Language and Social Psychology,* 29(1), S. 24–54.

Tennant, J. (2020): Web of Science and Scopus are not global databases of knowledge. *European Science Editing,* 46, e51987.

Thelwall, M. & Stuart, D. (2006): Web crawling ethics revisited: Cost, privacy, and denial of service. *Journal of the American Society for Information Science and Technology,* 57(13), S. 1771–1779.

Thelwall, M., Buckley, K. & Paltoglou, G. (2012): Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology,* 63(1), S. 163–173.

Tilly, C. (1984): *Big Structures, Large Processes, Huge Comparisons.* New York: Sage.

Tinto, V. (1975): Dropout from Higher Education: A Theoretical Synthesis of Recent Research. *Review of Educational Research,* 45(1), S. 89–125.

Tinto, V. (1997): Classrooms as Communities. Exploring the Educational Character of Student Persistence. *The Journal of Higher Education,* 68(6), S. 599–623.

Traus, A., Höffken, K., Thomas, S., Mangold, K. & Schröer, W. (2020): *Stu.diCo. – Studieren digital in Zeiten von Corona.* https://dx.doi.org/10.18442/150

Tudor, A. (1989): Monsters and Mad Scientists. A Cultural History of the Horror Movies. Oxford: Blackwell.

van Atteveldt, W., van der Velden, M. A. & Boukes, M. (2021): The Validity of Sentiment Analysis: Comparing Manual Annotation, Crowd-Coding, Dictionary Approaches, and Machine Learning Algorithms. *Communication Methods and Measures,* 15(2), S. 121–140.

Vatin, F. (2013): Valuation as Evaluating and Valorizing. *Valuation Studies,* 1(1), S. 31–50.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., … Bright, J. (2020): SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature methods,* 17(3), S. 261–272.

Vormbusch, U. (2012): *Die Herrschaft der Zahlen. Zur Kalkulation des Sozialen in der kapitalistischen Moderne.* Frankfurt am Main, New York: Campus.

Wang, C. & Blei, D. M. (2011): Collaborative topic modeling for recommending scientific articles. *KDD '11: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining,* S. 448–456.

Waskom, M. L. (2021): Seaborn: Statistical data visualization. *Journal of Open Source Software,* 6(60):3021.

Weber, R. (1990): *Basic Content Analysis.* Thousand Oaks, California: Sage.
Weingart, P. (2006): Chemists and their Craft in Fiction Film. *HYLE – International Journal for Philosophy of Chemistry,* 12(1): S. 31–44.
Weingart, P., Muhl, C. & Pansegrau, P. (2003): Of power maniacs and unethical geniuses: Science and scientists in fiction film. Public Understanding of Science 12(3): S. 279–288.
Wickham, H. & Wickham, M. H. (2017): *Package ‚tidyr'. Easily Tidy Data with ‚spread' and ‚gather ()' Functions.* https://mran.revolutionanalytics.com/web/packages/tidyr/tidyr.pdf
Wickham, H. (2014): Data manipulation with dplyr. *The Journal of Open Source Software,* 4(43):1686.
Wieczorek, O., Unger, S., Riebling, J., Erhard, L., Koß, C. & Heiberger, R. (2021): Mapping the field of psychology. Trends in research topics 1995–2015. *Scientometrics,* 126, S. 9699–9731.
Willke, H. (2004): *Einführung in das systemische Wissensmanagement.* Heidelberg: Carl-Auer-Systeme.
Wolff, S. (2000): Dokumenten- und Aktenanalyse. In: Flick, U., von Kardorff, E. & Steinke, E. (Hrsg.), *Qualitative Forschung. Ein Handbuch.* Reinbek: Rowohlt, S. 502–514.
Yao, J. & Shepperd, M. (2020): Assessing software defection prediction performance: Why using the Matthews correlation coefficient matters. *Proceedings of the Evaluation and Assessment in Software Engineering.* EASE 2020, April 15–17, 2020, Trondheim, Norway, S. 120–129.
Yilmaz Sener, M. (2019): Perceived discrimination as a major factor behind return migration? The return of Turkish qualified migrants from the USA and Germany. *Journal of Ethnic and Migration Studies,* 45(15), S. 2801–2819.
Zainuddin, N., Selamat, A. & Ibrahim, R. (2018): Hybrid sentiment classification on twitter aspect-based sentiment analysis. *Applied Intelligence,* 48(5), S. 1218–1232.
Zitt, M. & Bassecoulard, E. (2006): Delineating complex scientific fields by an hybrid lexical-citation method: An application to nanosciences. *Information Processing & Management,* 42(6), S. 1513–1531.

Autor*innenvorstellung

Christian Schneijderberg ist promovierter Soziologe. Seit 2009 forscht er am International Center for Higher Education Research (INCHER) und lehrt im Fach Soziologie an der Universität Kassel. Im akademischen Jahr 2020/21 hat Christian Schneijderberg die Professur „Soziologie, Methoden und Techniken der empirischen Sozialforschung" in der Soziologie der RWTH Aachen vertreten. Als empirischer Sozialforscher und Methodenpragmatist verwendet Christian Schneijderberg qualitative, quantitative und Mixed Methods bzw. methodenplurale Forschungsdesigns. Die Methodenwahl hängt jeweils von Untersuchungsgegenstand, Erkenntnisinteresse und Daten (z. B. aus Datenbank, Ethnographie, Fragebogenerhebung, Gruppendiskussionen und Interviews) ab.

Oliver Wieczorek ist promovierter Soziologe und seit 2022 am International Center for Higher Education Research (INCHER) in Kassel angestellt. Zuvor war er an der Zeppelin Universität Friedrichshafen und an der Universität Bamberg in mehreren Projekten in den Bereichen der Wissenschafts- und Hochschulforschung sowie kritischen Bildungsforschung gemeinsam mit Prof. Münch beteiligt. Seine Spezialisierung im methodischen Bereich liegt auf quantitativen Analyseverfahren, zu denen die quantitative Textanalyse, Netzwerkanalyse sowie verschiedene Spielarten der Regressions- und Ereignisanalyse zählen.

Isabel Steinhardt ist Professorin für Bildungssoziologie an die Universität Paderborn. Zuvor war Isabel Steinhardt wissenschaftliche Mitarbeiterin in der Soziologie und dem International Center for Higher Education Research (INCHER) der Universität Kassel. In Kassel hat sie Methodenberatungen im Kompetenzzentrum für empirische Forschungsmethoden angeboten und in diesem Zuge 2016 den Blog „Sozialwissenschaftliche Methodenberatung" gegründet. Ihre Forschungsthemen liegen neben den qualitativen Methoden in der Bildungsforschung, Praxistheorie, Digitalisierung/Digitalität sowie Open Science.

Udo Kuckartz | Stefan Rädiker
Qualitative Inhaltsanalyse.
Methoden, Praxis, Computerunterstützung
5. Aufl. 2022, 274 Seiten, broschiert
ISBN: 978-3-7799-6231-1
Auch als E-BOOK erhältlich

Dieses Lehrbuch bietet eine methodisch fundierte, verständliche und anwendungsbezogene Anleitung zur inhaltsanalytischen Auswertung qualitativer Daten.
Dabei werden drei Varianten qualitativer Inhaltsanalyse ausführlich vorgestellt:

- die inhaltlich strukturierende,
- die evaluative und
- die typenbildende qualitative Inhaltsanalyse.

Dieses Buch ist ein wertvoller Begleiter für die wissenschaftliche Forschungspraxis in vielen Disziplinen.

www.beltz.de
Beltz Juventa · Werderstraße 10 · 69469 Weinheim