Data Science für normale Anwender*innen

Prof. Dr. Nicolas Meseth

Aktualisiert am 13.01.2022

Contents

Übersicht

Zielgruppe

Dieses Buch gibt eine Einführung in wichtige Themen bei der Arbeit mit Daten. Wie der Titel schon ahnen lässt, adressiert dieses Buch *normale* Anwender*innen. Damit meine ich Personen, die in ihrem beruflichen (oder privaten) Alltag von Kenntnissen in der Analyse von Daten (über Excel hinaus) profitieren, diese aber nicht deren Haupttätigkeit ist.

In genau diese Zielgruppe fallen mehr als 95% der Studierenden, die an der Hochschule Osnabrück in jedem Semester an meinen Vorlesungen und Seminaren teilnehmen. Sie studieren einen Studiengang aus der Fachrichtung Agrar- oder Lebensmittelwirtschaft und haben in meinen Veranstaltungen die Möglichkeit, zusätzliche Kompetenzen zu digitalen Themen zu entwickeln. Sie arbeiten später aber nicht in IT-Berufen oder als Data Scientist. Sie sind aber dafür gerüstet, mit diesen Abteilungen (IT / Data Scientists) besser zusammenzuarbeiten und viele Dinge können sie auch in Eigenregie umsetzen. Dazu zählen beispielsweise Datenanalysen und -visualisierungen mit R.

Didaktisches Konzept

Das Buch ist in fortlaufend nummerierte Kapitel gegliedert. Man folgt meinem didaktischen Konzept, wenn man diese Reihenfolge einhält. Dieses Konzept beruht darauf, sich zuerst *hands-on* mit einem Thema auseinanderzusetzen, bevor die relevanen theoretischen Hintergründe eingeführt werden. Meiner Erfahrung nach sind theoretische Hintergründe einfacher zu verstehen, wenn das Subjekt der Betrachtung bereits in den Händen war. Der Kontext ist präsent und man weiß bei fachlichen Begriffen sofort die richtige Assoziation herzustellen.

Ein zweiter wichtiger Gedanke beim didaktischen Aufbau des Buches ist die Verwendung von Beispielen aus der Praxis. Dazu zählen insbesodere Datensätze und Fragen an diese Daten. Jedes Kapitel beinhaltet Beispiele und Übungen mit

Bezug zu Daten aus der Praxis. Die Anwendungsfälle hinter den Daten reichen von Onlineumfragen aus der Markforschung über Daten aus Laborexperimenten bis hin zu Datensätzen aus sozialen Medien wie Twitter. Alle Datensätze, die in diesem Buch verwendet werden, sind im Kapitel Datensätze gelistet.

Nicht immer ist es sinnvoll, ein neues Konzept anhand eines Datensatzes aus der Praxis einzuführen. In diesem Fall greife ich auf synthetische Datensätze zurück, um die Idee oder das Konzept möglichst plastisch darstellen zu können. Die Anwendung des eingeführten Konzepts auf Datensätze aus der Praxis erfolgt dann anschließend. Es sollte in diesem Buch kein Konzept eingeführt werden, das nicht in der Praxis relevant ist.

Das Buch kann selbstverständlich abseits des didaktischen Konzepts verwendet werden. Ich habe beim Verfassen darauf geachtet, das jedes Kapitel in sich geschlossen ist und einzeln gelesen und bearbeitet werden kann. Das gilt auch für die Übungen für die Anwendung mit R.

Erste Schritte mit R

Chapter 1

Die Arbeitsumgebung

TL;DR

- Für die Arbeit mit R benötigen wir eine Installation von R auf unserem Computer.
- R ist für alle Betriebssysteme verfügbar.
- Mit R bekommen wir auch eine einfache grafische Benutzeroberfläche (RGui) mitgeliefert. Diese bietet aber wenig Funktionen.
- Mit dem kostenlosen RStudio bekommen wir eine vollwertige integrierte Entwicklungsumgebung (IDE) für die Arbeit R.

1.1 R installieren

R ist eine Open-Source-Software und für alle gängigen Betriebssysteme verfügbar. Ladet euch zunächst die neueste Version von R für euer Betriebssystem herunter und installiert es anschließend:

- R für Windows
- R für macOS
- R für Linux

Neben der Sprache und dem Interpreter für R erhaltet ihr mit der Installation auch eine rudimentäre grafische Oberfläche mit dem Namen *RGui* (GUI = Graphical User Interface). Diese besteht aus einer einfachen Konsole, über die ihr R-Befehle eingeben und ausführen könnt.

Erweiterte Funktionen wie Autovervollständigung beim Schreiben von R-Code, ein integrierter Debugger für die Fehlersuche, eine Echtzeit-Vorschau für R-Markdown und viele andere Features mehr bietet dieses einfache Tool nicht.

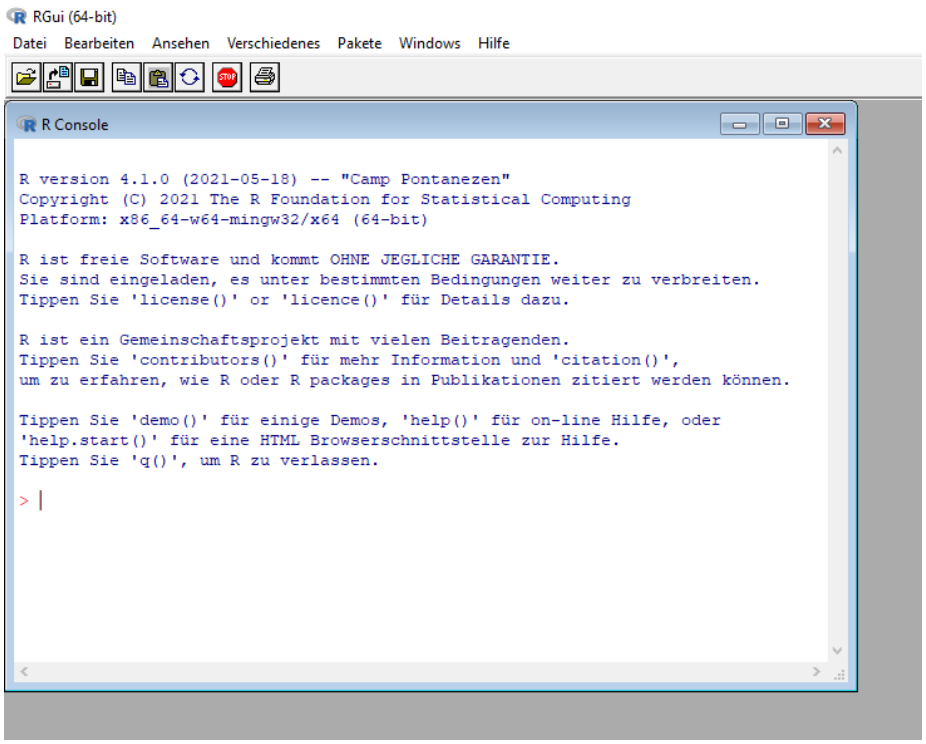

Figure 1.1: Die RGui bietet einen rudimentären Editor für R-Befehle.

Deshalb verwenden wir für die Arbeit mit R nicht die RGui, sondern das ebenfalls kostenlos nutzbare RStudio.

1.2 RStudio installieren

Das RStudio ist das Standardwerkzeug für die Arbeit mit R und bietet dafür viele nützliche Funktionen. Das RStudio ist ein sogenanntes **Integrated Development Environment (IDE)** für R. Einen schnellen Überblick über die grafische Benutzeroberfläche findet ihr in dem offiziellen RStudio Cheatsheet.

Klickt auf den Link unten und wählt RStudio für euer Betriebssystem aus. Installiert RStudio und öffnet es:

- RStudio herunterladen

Nach dem Öffnen seht ihr die Oberfläche des RStudio, die wie auf dem Screenshot unten aussieht:

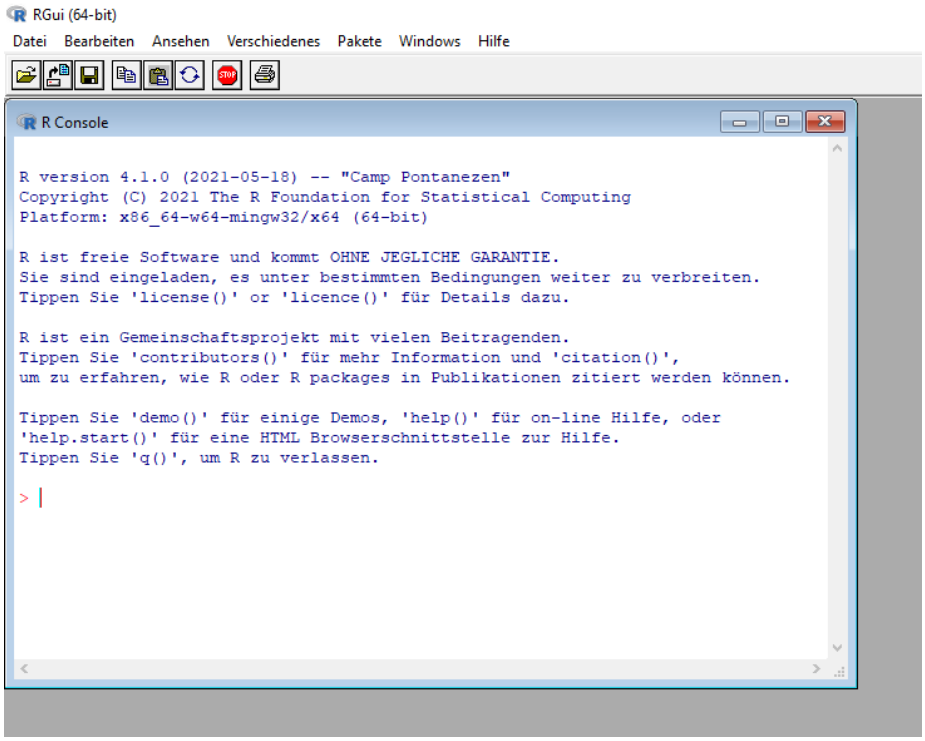

Figure 1.2: Das RStudio ist in vier Bereiche eingeteilt.

1.3 Überblick über RStudio

Das Werkzeug besteht in der Standardansicht aus vier Bereichen (s. Screenshot oben):

1.3.1 Der Skripteditor

Der wichtigste Bereich ist der Skripteditor. Hier schreiben wir unseren R-Code und speichern ihn in Dateien auf unserem Computer. Dabei unterstützt uns RStudio mit vielen nützlichen Funktionen.

Wir können in RStudio unterschiedliche Arten von Dateien verwenden, um unsere Skripte zu speichern. Die einfachste Art sind sogenannte **R-Skripte** mit der Dateiendung `.R`. Wenn wir nicht nur R-Code, sondern auch Visualisierungen und formatierten Text in einem Dokument verwenden und anzeigen wollen, bietet sich die Verwendung eines **R-Notebooks** an. Eine R-Notebook-Datei endet auf `.Rmd` (R-Markdown) und wir können neben R-Code auch Markdown verwenden. Über Markdown lernen wir zu einem späteren Zeitpunkt mehr.

1.3.2 Informationen zur aktuellen R-Umgebung

R erstellt für die Ausführung von Skripten eine sogenannte Session. In einer Session werden alle Objekte, wie die momentan verwendeten Daten, eigene Variablen oder Funktionen, im Arbeitsspeicher des lokalen Rechners gespeichert. In dem zweiten Fenster können im Tab *Environment* alle Objekte, die es in der aktuellen Session gibt, in der Übersicht und im Detail betrachtet werden. Der *History* Tab enthält eine Liste aller ausgeführten Befehle in der aktuellen Session. Daneben gibt es noch den *Connections* Tab für die Verbindung zu unterschiedlichen Datenquellen, wie Datenbanken oder Webservices, sowie den *Tutorial* Tab, in dem direkt in RStudio unterschiedliche Anleitungen angezeigt werden können.

{% hint style="info" %} Für die Anzeige von Tutorials direkt in RStudio muss das Paket `learnr` installiert werden. Ihr müsst der Installation einmalig zustimmen und das Paket installieren. {% endhint %}

1.3.3 Konsole und Terminal

Die Konsole ermöglicht, R-Befehle einzugeben und mit der Taste Enter auszuführen. Die Konsole in RStudio ist fast identisch zu der RGui. Für das schnelle Ausprobieren von Befehlen kann die Konsole nützlich sein. Für die strukturierte Arbeit mit den Daten sollten wir aber den Skripteditor verwenden, weil wir dort unsere Arbeit speichern und mit Kommentaren versehen können.

1.3.4 Dateien, Vorschau und Plots

In diesem Bereich werden Vorschauen unterschiedlicher Artefakte angezeigt. Dazu gehören gerenderte R-Notebooks, Markdown, aber auch Visualisierungen. In diesem Bereich gibt es auch einen rudimentären Datei-Explorer, um Dateien wie Skripte oder Datendateien zu finden und von dort zu öffnen.

Chapter 2

Einen Datensatz laden

Nachdem die Arbeitsumgebung eingerichtet ist, können wir mit den ersten Schritten in R beginnen. Wir steigen direkt ein und lernen, wie wir mit Daten in R arbeiten.

TL;DR

- R bietet verschiedene Möglichkeiten, um mit Daten zu arbeiten.
- Für strukturierte Daten in Tabellenform (Zeilen und Spalten) verwenden wir in R den Dataframe.
- Eine modernere Version des klassischen Dataframe ist das Tibble aus dem Tidyverse.

2.1 Der Dataframe

In R gibt es verschiedene Strukturen für die Speicherung von Daten. Darunter sind beispielsweise Vektoren, Listen oder Matritzen. Um tabellarische Daten abzubilden, die in Spalten und Zeilen organisiert sind, verwenden wir in R den sogenannten **Dataframe**.

2.2 Tibbles

Der Tibble ist eine Weiterentwicklung des klassischen Dataframe in R. Tibbles werden im Tidyverse standardmäßig verwendet und mit dem Paket `tibble` eingeführt.

2.3 Das Tidyverse

Chapter 3

Daten aus dem CSV-Format laden

3.1 Das CSV-Format

In den meisten Fällen erzeugen wir einen Dataframe oder Tibble, indem wir einen Datensatz aus einer Datenquelle laden. Häufig ist die Quelle eine einfache Texdatei, in der Daten zeilenweise gespeichert sind und jede Zeile aus mehreren einzelnen Werten besteht, die durch ein bestimmtes **Trennzeichen** voneinander getrennt sind. Wenn die Werte mit einem Komma voneinander getrennt sind, nennt man das Format **CSV**. CSV steht für **Comma Separated Values**, was auf Deutsch so viel wie *Durch Kommata getrennte Werte* bedeutet.

Unten seht ihr ein Beispiel für Daten im CSV-Format.

```
id,firstname,lastname
1,Boris,Becker
2,Steffi,Graf
3,Rafael, Nadal
```

Die erste Zeile nennen wir auch Kopfzeile oder *Header*. Sie beinhaltet die Spaltennamen. Jede nachfolgende Zeile stellt einen Datensatz oder *Record* dar. Dabei müssen die Werte in jeder Zeile mit Komma getrennt in der selben Reihenfolge wie im Header aufgeführt werden. Nur so ist eindeutig erkennbar, welcher Wert zu welcher Spalte gehört.

Manchmal fehlen Werte für eine bestimmte Spalte. In diesem Fall werden einfach zwei (oder mehrere) Kommata hintereinander gesetzt.

```
id,firstname,lastname,height,weight,gender
```

```
1,Boris,Becker,,,male
2,Steffi,Graf,175,64,female
3,Rafael,Nadal,185,,male
```

Chapter 4

Einen Datensatz erkunden

4.1 Alle Daten anzeigen

Oft ist es hilfreich, einen schnellen Blick in einen Datensatz zu werfen, um beispielsweise die Werte einer Spalte zu überprüfen. Dafür können wir in RStudio die Funktion `view()` verwenden. Der Funktion übergeben wir ein Tibble und es öffnet sich daraufhin ein neuer Tab im Skripteditor, der die Daten als Tabelle anzeigt.

```
view(food_production)
```

Chapter 5

Der Werkzeugkasten

Bei der Arbeit mit Daten ist es wichtig zu wissen, welches Werkzeug (hier: R-Paket oder Funktion) wir für welche Aufgabe verwenden. Genauso wie in einer Werkstatt, in der es eine Vielzahl an Werkzeugen gibt, die für unterschiedliche Zwecke geeignet sind.

Figure 5.1: Verschiedene Werkzeuge für unterschiedliche Aufgaben.

Die Tabelle unten listet wichtige Werkzeuge auf, die wir im weiteren Verlauf dieses Skriptes kennenlernen werden.

Paket	Funktion	Aufgabe
tibble	as.tibble	Erstellt einen modernen Dataframe für tabellarische Daten.
readr	read_csv	Lesen von tabellarischen Datenformaten wie CSV-Dateien.
dplyr	select	Auswählen von Spalten (Variablen) eines Datensatzes.
dplyr	filter	Filtern von Daten auf Basis fast beliebiger Ausdrücke.
dplyr	mutate	Hinzufügen neuer Spalten (Variablen).
dplyr	recode	Spaltenwerte neu kodieren.
dplyr	arrange	Die Reihenfolge von Zeilen verändern.
dplyr	group_by	Gruppieren von Daten.
dplyr	summarise	Zusammenfassen von Daten.
ggplot2	ggplot, aes, geom_line, geom_bar, geom_col geom_point u.v.m.	Visualisieren von Daten.

5.1 Das Paket `tibble`

Das Paket `tibble` führt das moderne Pendant zum klassischen Dataframe in R ein.

- Zur offiziellen Webseite des `tibble` Pakets

5.2 Das Paket `readr`

Das Paket `readr` beinhaltet Funktionen für das Laden von Daten aus strukturierten Datenformaten wie CSV-Dateien. Alle Funktionen zum Datenimport aus readr erzeugen automatisch einen `tibble`.

- Zur offiziellen Webseite des `readr` Pakets

5.3 Das Paket `dplyr`

Das Paket `dplyr` hat einen etwas merkwürdigen Namen. Er setzt sich aus dem Buchstaben „d“ und dem abgekürzten Wort „plyr“ zusammen. Das „d“ steht für Dataframe, während „plyr“ für den englischen Begriff „plier“ steht, was auf Deutsch „Zange“ bedeutet. Passend dazu bildet das offizielle Symbol des Pakets mehrere Zangen ab.

- Zur offiziellen Dokumentation des `dplyr` Pakets

`dplyr` liefert uns eine Vielzahl wichtiger Funktionen für die Manipulation von Daten, die in Form eines Tibble vorliegen. Eine Übersicht der Funktionen findet ihr in dem Cheat Sheet Data Transformation wit dplyr.

5.4 Das Paket `ggplot2`

`ggplot2` ist eines der umfassendsten Pakete für die professionelle Visualisierung von Daten mit R:

- Zur offiziellemn Dokumentation des `ggplot2` Pakets

Das Cheat Sheet Data Visualization with ggplot2 beinhaltet alle wichtigen Funktionen im Überblick.

Explorative Datenanalyse mit R

Chapter 6

Der explorative Analyseprozess

Dieser Abschnitt führt euch in die Grundlagen der **explorativen Datenanalyse** mit R ein. In der explorativen Datenanalyse versuchen wir einen unbekannten Datensatz mit geeigneten Verfahren kennenzulernen und schnell Muster in den Daten zu erkennen. Auf Basis dieser Muster formulieren wir **Hypothesen**. Diese Hypothesen können anschließend mit statistischen Modellen aus dem Bereich der schließenden Statistik auf ihre Gültigkeit überprüft werden. Dieser Schritt ist jedoch nicht Teil der explorativen Datenanalyse nach dem Verständnis dieses Buches.

Eine ausgezeichnete Einführung in die explorative Datenanalyse mit R gibt auch das Buch R for Data Science von Wickham and Grolemund [2016]. Das Buch ist online frei zugänglich.

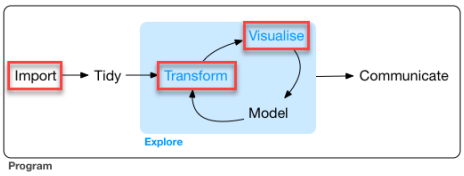

Figure 6.1: Der Datenanalyseprozess.

Wickham and Grolemund [2016] definieren den Datenanalyseprozess durch eine Abfolge bestimmter Schritte, wie in der Abbildung **??** gezeigt. In diesem Abschnitt stehen die rot markierten Schritte im Fokus.

6.1 Daten laden

Jeder Analyseprozess beginnt mit dem Laden eines Datensatzes. Dabei gibt es verschiedene Datenquellen, die in Betracht gezogen werden müssen. Ein häufig verwendetes Format sind Komma-separierte Werte (***c**omma **s**eparated **v**alues* = CSV) in einfachen Textdateien. Dieses Format steht auch hier im Vordergrund.

Gemäß der Abbildung aus Wickham und Grolemund 2016 folgt auf das Laden der Daten der Arbeitsschritt „Tidy“. Dieser ist dann notwendig, wenn die Daten nicht in der typischen Form bestehend aus Spalten und Zeilen vorliegen. Leider ist das in der Praxis oft der Fall. In diesem Kurs wird aber zunächst davon ausgegangen, dass die Daten das entsprechende Format aufweisen. Die interessierte Leserin verweise ich auf das Kapitel 12 im Buch „R for Data Science“.

6.2 Daten transformieren

Das Ziel der explorativen Datenanalyse ist die Visualisierung der Daten mit geeigneten Diagrammen, um interessante Muster sichtbar werden zu lassen. Visualisierungen benötigen häufig nur einen Teil der Daten (wenige Spalten oder bestimmte Zeilen). Auch müssen wir oft neue Spalten berechnen oder bestehende Daten aggregieren, bevor wir sie visualisieren können. Alle diese Aufgaben können wir unter dem Bereich der **Datentransformation** zusammenfassen.

Für diese Aufgaben bietet R mit dem Paket `dplyr` mächtige Funktionen. Insbesondere lernen wir in dem Abschnitt:

- Wie wir bestimmte Spalten auswählen können (dplyr-Verb: `select`).
- Wie wir Zeilen fast beliebig filtern können (dplyr-Verb: `filter`).
- Wie wir neue, berechnete Spalten hinzufügen können (dplyr-Verb: `mutate`).
- Wie wir Zeilen sortieren können (dplyr-Verb: `arrange`).
- Wie wir Zeilen zusammenfassen und gruppieren können (dplyr-Verben: `summarize` und `group_by`)

6.3 Daten visualisieren

Das wichtigste Werkzeug in der explorativen Datenanalyse ist die **Visualisierung von Daten**. In R steht uns dafür mit `ggplot2` ein leistungsfähiges Instrument zur Verfügung. Wir lernen für bestimmte Anwendungsfälle die richtigen Visualisierungen zu identifizieren und mit `ggplot2` umzusetzen.

6.4 Literatur

6.4.1 Bücher

Wickham, Hadley, and Garrett Grolemund. R for Data Science: Import, Tidy, Transform, Visualize, and Model Data. First edition, O'Reilly, 2016. Online verfügbar: https://r4ds.had.co.nz/

Wickham, Hadley. ggplot2. Springer Science+Business Media, LLC, 2016. Online verfügbar: https://ggplot2-book.org/

Kabacoff, Robert. R in Action: Data Analysis and Graphics with R. Second edition, Manning, 2015.

Sauer, Sebastian. Moderne Datenanalyse mit R: Daten einlesen, aufbereiten, visualisieren, modellieren und kommunizieren. Springer Gabler, 2019. Online verfügbar: https://link.springer.com/book/10.1007/978-3-658-21587-3

6.4.2 Online-Dokumentationen

- Die offizielle Dokumentation der Tidyverse-Bibliotheken
- Ein Tutorial für die Einführung in R mit RStudio
- Ein Tutorial zu den Grundlagen der Datenmanipulation mit R, `tidyr` und `dplyr`
- Weiterführende Anleitungen zur Datenmanipulation mit `dplyr` (Efficient Data Manipulation)
- Weiterführende Anleitungen zur Datenmanipulation mit `dplyr` (Advanced Data Manipulation)

Anhang

Datensätze

Table 6.1: Datensätze in diesem Buch

Name	**Download-Link**	**Kategorie**
Online-Umfrage zum Kaufverhalten von Orangenlimonade	Download	Marktforschung
Tweets ausgewählter deutscher Politiker:innen	Download	Soziale Medien

Bibliography

Hadley Wickham and Garrett Grolemund. *R for data science: import, tidy, transform, visualize, and model data*. O'Reilly, Sebastopol, CA, first edition edition, 2016. ISBN 9781491910399 9781491910368. OCLC: ocn968213225.