Data Science für normale Anwender*innen

Prof. Dr. Nicolas Meseth

2022-01-12

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 in Kapitel 5 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.

Grundlagen

Chapter 1

Die Arbeitsumgebung

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. 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.

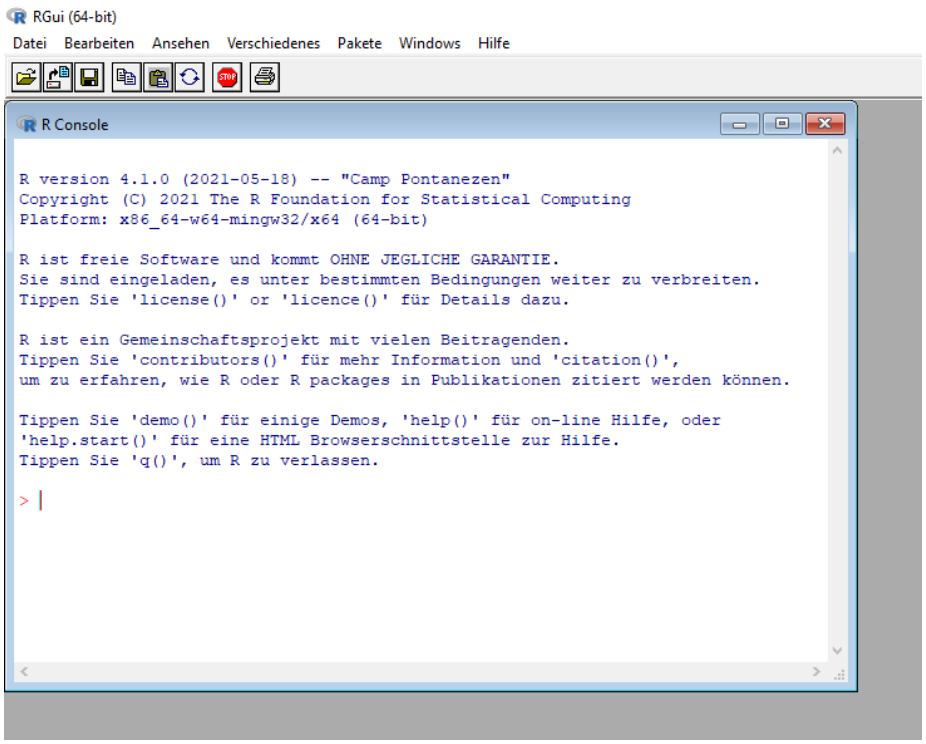

Figure 1.1: Die RGui bietet einen rudimentären Editor für R-Befehle.

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:

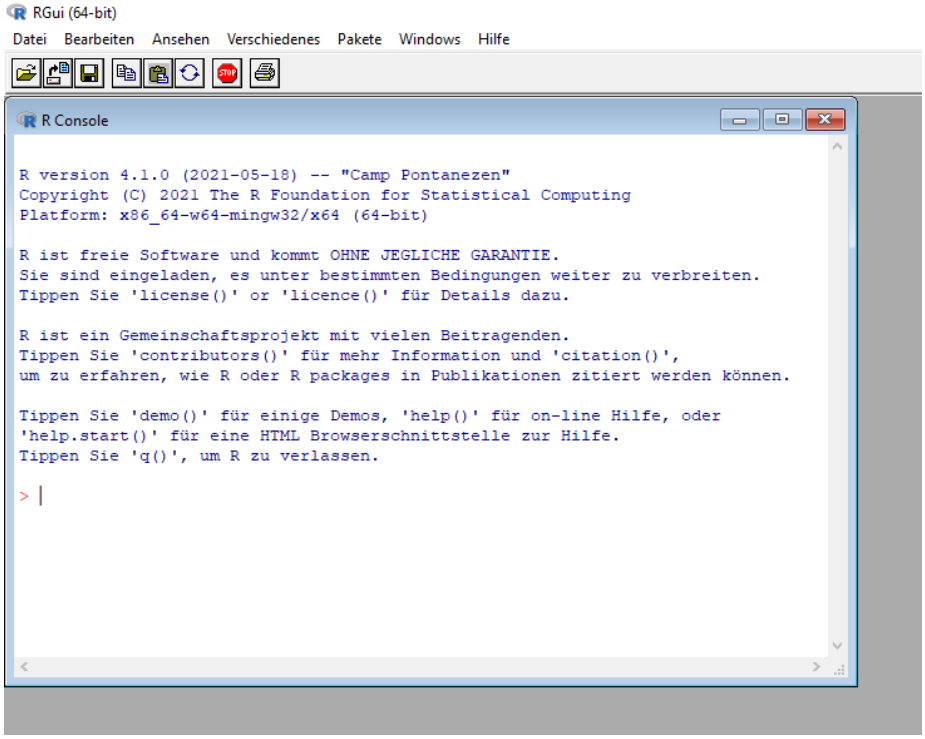

Figure 1.2: Das RStudio ist in vier Bereiche eingeteilt.

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 wollen, bietet sich die Verwendung eines R-Notebooks an. Ein R-Notebook endet auf `.Rmd` und wir können neben R-Code auch Markdown verwenden.

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

Strukturierte Daten (Tabellen)

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.

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

Wenn man so will ist der Tibble eine Weiterentwicklung des klassischen Dataframe in R. Tibbles werden im Tidyverse standardmäßig verwendet und mit dem Paket `tibble` eingeführt.

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 Werten besteht, die durch ein Trennzeichen voneinander getrennt sind. Wenn die Werte mit einem Komma voneinander getrennt sind, nennt man das Forat auch **CSV**. CSV steht für **Comma Separated Values**, was auf Deutsch so viel wie *Durch Kommata getrennte Werte* bedeutet.

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

Datensätze

Table 5.1: Datensätze in diesem Buch

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