



# **I R-Manual**

## **1 R und R-Studio installieren**

- 1.1 R Download
- 1.2 R-Studio Download
- 1.3 YouTube Video

## **2 Der Aufbau von R-Studio**

- 2.1 Benutzeroberfläche
- 2.2 Voreinstellungen
- 2.3 Packages installieren

## **3 Erste Schritte**

- 3.1 Operationen und Funktionen
- 3.2 Variablen definieren
- 3.3 Daten importieren

## **4 Datenanalyse**

- 4.1 Deskriptive Statistik
- 4.2 Neue Variablen erstellen
- 4.3 Beobachtungen sortieren und filtern

## **5 Graphiken**

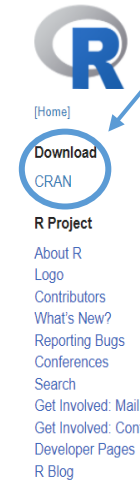
- 5.1 Punkt- und Liniendiagramm
- 5.2 Säulen-/Balkendiagramm

# 1 R und R-Studio installieren

## 1.1 R Download

- ➔ [www.r-project.org](https://www.r-project.org) ➔ Download: CRAN
- ➔ Unter “Germany” eine Domain zum Download auswählen (egal welche)
- ➔ Download für Windows/Mac/Linux auswählen
- ➔ Auf „install R for the first time“ klicken

<https://www.r-project.org>



### The R Project for Statistical Computing

#### Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred CRAN mirror.

If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

#### News

- **R version 4.2.0 (Vigorous Calisthenics) prerelease versions** will appear starting Tuesday 2022-03-22. Final release is scheduled for Friday 2022-04-22.
- **R version 4.1.3 (One Push-Up)** has been released on 2022-03-10.
- **R version 4.0.5 (Shake and Throw)** was released on 2021-03-31.
- Thanks to the organisers of useR! 2020 for a successful online conference. Recorded tutorials and talks from the conference are available on the [R Consortium YouTube channel](#).
- You can support the R Foundation with a renewable subscription as a [supporting member](#)

R for Windows

Subdirectories:

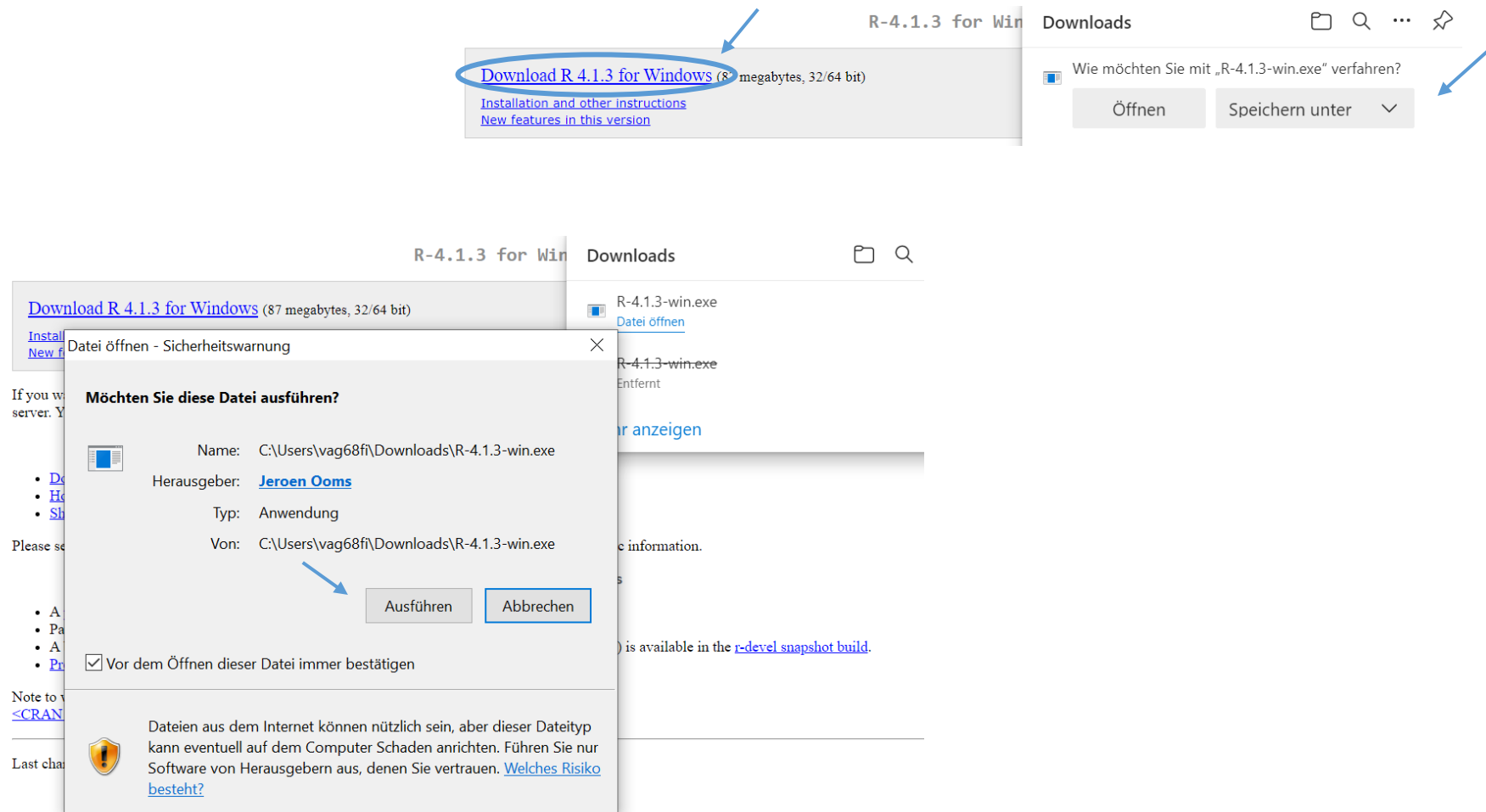
<p><a href="#">base</a></p> <p><a href="#">contrib</a></p> <p><a href="#">old contrib</a></p> <p><a href="#">Rtools</a></p>	<p>Binaries for base distribution. This is what you want to <b>install R for the first time</b>.</p> <p>Binaries of contributed CRAN packages (for R &gt;= 3.4.x).</p> <p>Binaries of contributed CRAN packages for outdated versions of R (for R &lt; 3.4.x).</p> <p>Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.</p>
---	---

Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

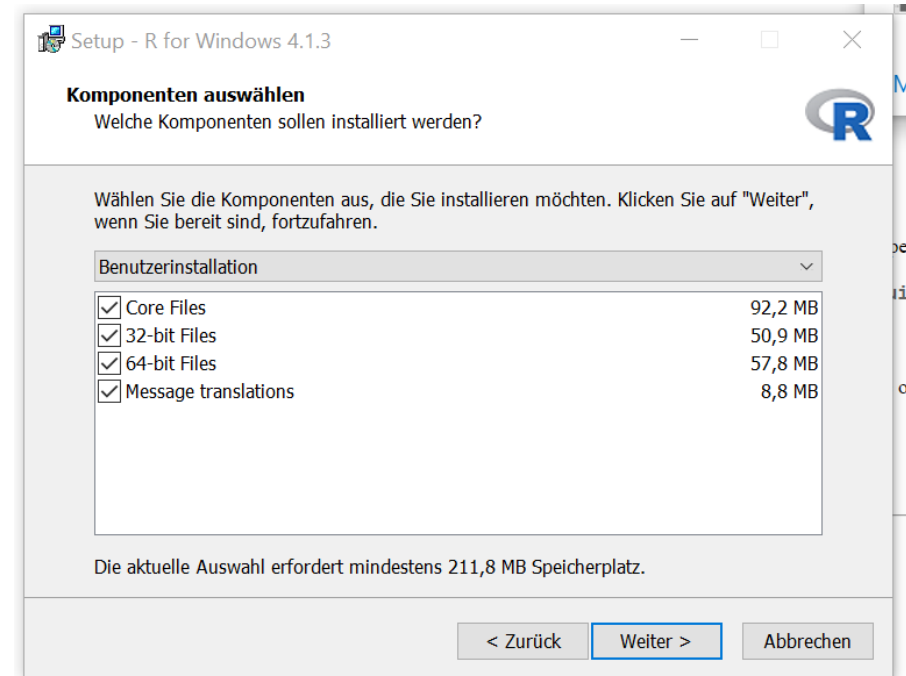
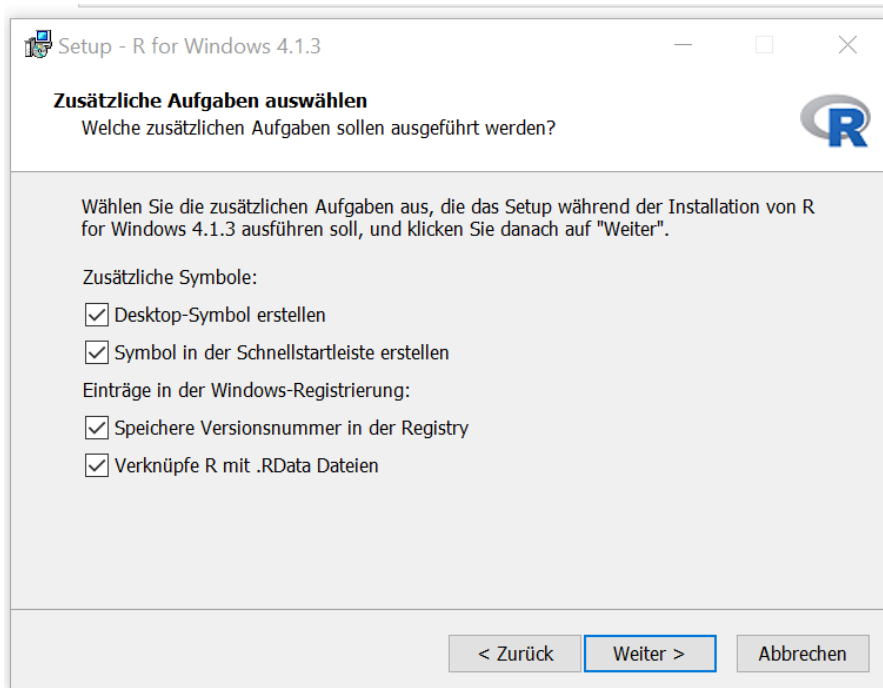
You may also want to read the [R FAQ](#) and [R for Windows FAQ](#).

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

- ➔ Auf „Download R 4.1.3 for Windows/Mac/Linux“ klicken
- ➔ Speichern und Ausführen ➔ Download beginnt



- ➔ Nach Downloadstart erscheint ein Installationsassistent: Komponenten, die installiert werden sollen, auswählen (alle) ➔ weiter
- ➔ Ordner zum Speichern des Programms wählen (Vorauswahl meistens passend) ➔ weiter
- ➔ Zusätzliche Aufgaben auswählen: alles optional, aber sehr hilfreiche (alle vier anklicken)



## 1.2 R-Studio Download

- ➔ [www.rstudio.com/products/rstudio/](http://www.rstudio.com/products/rstudio/) ➔ RStudio Desktop auswählen (Open-Source Edition)
- ➔ Installation für das passende Betriebssystem auswählen und Download starten

RStudio Desktop Open Source License <b>Free</b> <a href="#">Learn more</a>	RStudio Desktop Pro Commercial License <b>\$995</b> /year <a href="#">Learn more</a>	RStudio Server Open Source License <b>Free</b> <a href="#">Learn more</a>	RStudio Workbench Commercial License <b>\$4,975</b> /year (5 Named Users) <a href="#">Evaluation</a>   <a href="#">Learn more</a>
<a href="#">DOWNLOAD</a>	<a href="#">BUY</a>	<a href="#">DOWNLOAD</a>	<a href="#">BUY</a>

RStudio Desktop 2022.02.1+461 - [Release Notes](#)

1. Install R. [RStudio requires R 3.3.0+](#)
2. Download RStudio Desktop. Recommended for your system:

[DOWNLOAD RSTUDIO FOR WINDOWS](#)  
2022.02.1+461 | 177.27MB

Requires Windows 10/11 (64-bit)



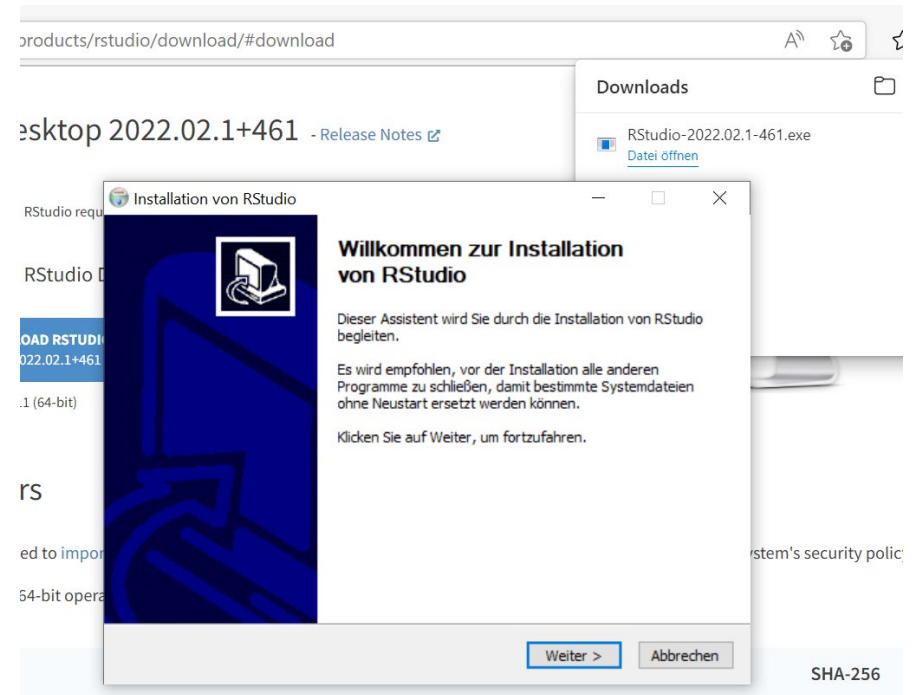
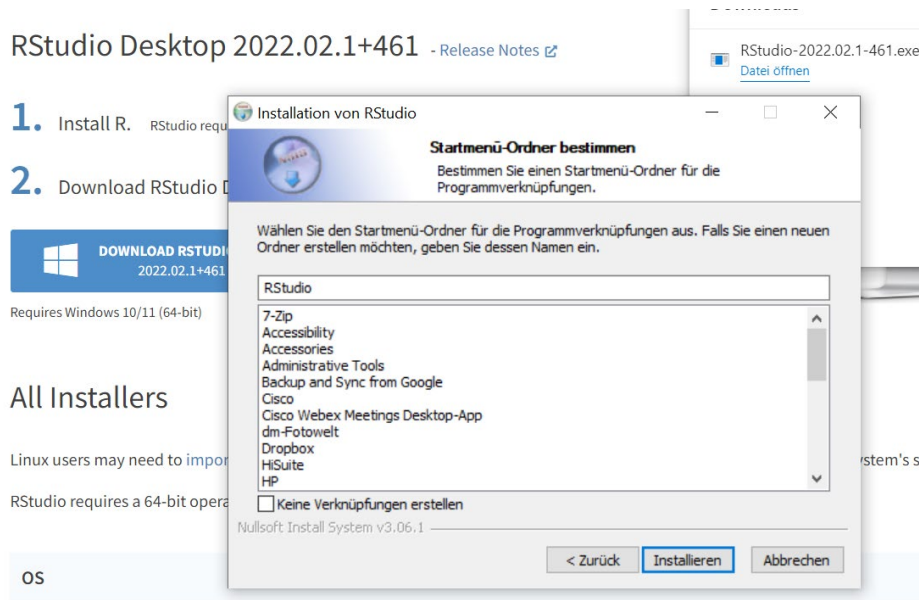
### All Installers

Linux users may need to [import RStudio's public code-signing key](#) prior to installation, depending on the operating system's security policy.

RStudio requires a 64-bit operating system. If you are on a 32 bit system, you can use an older version of RStudio.

OS	Download	Size	SHA-256
Windows 10/11	<a href="#">RStudio-2022.02.1-461.exe</a>	177.27 MB	b14149b1
macOS 10.15+	<a href="#">RStudio-2022.02.1-461.dmg</a>	217.25 MB	5b268cfa
Ubuntu 18+/Debian 10+	<a href="#">rstudio-2022.02.1-461-amd64.deb</a>	128.58 MB	d5aaa02f

- ➔ Speichern und Ausführen ➔ Download beginnt
- ➔ Nach Downloadstart erscheint ein Installationsassistent: Startmenü-Ordner bestimmen (Vorauswahl meist passend ➔ auf Installieren klicken



## 1.3 YouTube-Video zur Installation von R und R-Studio

[R installieren \(Windows\) und RStudio installieren - YouTube](#)



## 2 Der Aufbau von R-Studio

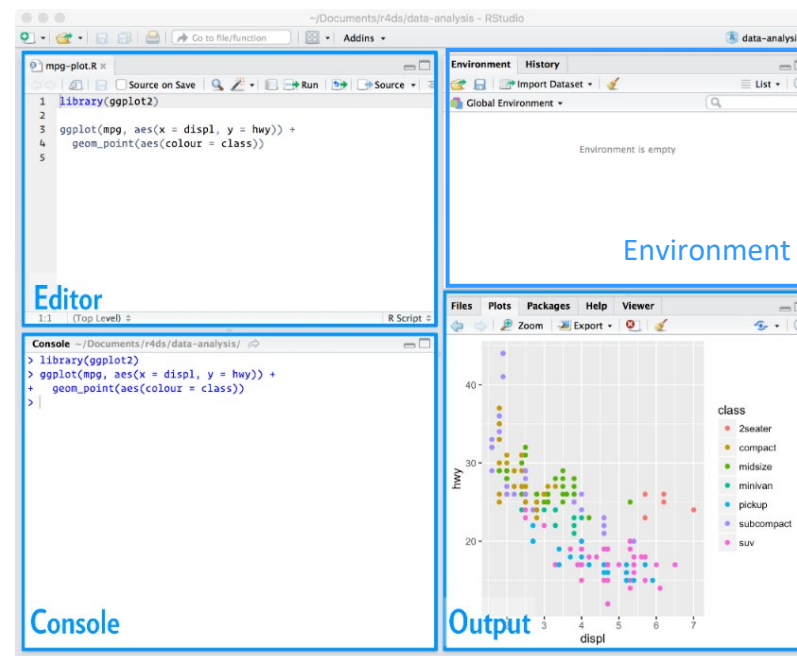
### 2.1 Benutzeroberfläche

**Konsole:** hier kann der R Code eingegeben werden, der dann von R ausgeführt wird.

**Editor / R Script:** hierbei handelt es sich um eine Textdatei, in die R-Befehle eingetragen werden können, die dann zur Ausführung an die Konsole geschickt werden. Befehle werden über die Tastenkombination *Strg* + *Enter* ausgeführt. Zeilen, die mit einem *#* beginnen, werden nicht ausgeführt.

**Environment:** hier werden alle definierten Variablen, Datensätze und Funktionen angezeigt (Abbildung rechts oben).

**Output:** hier wird der generierte Output, wie zum Beispiel Graphiken, angezeigt.





## 2.2 Voreinstellungen

Um beim Neustart von R immer ein leeres Datenfenster (Environment) vorzufinden, ist folgende Änderung der Voreinstellungen nötig:

1. R-Studio öffnen
2. Im Auswahlménü oben auf „Tools“ klicken → untersten Reiter „Global Options“ anwählen

3. The image shows two side-by-side screenshots of the 'Workspace' section in R Studio's Global Options dialog. A blue arrow points from the left screenshot to the right one. In the left screenshot, the checkbox 'Restore .RData into workspace at startup' is checked, and the dropdown 'Save workspace to .RData on exit:' is set to 'Always'. In the right screenshot, the checkbox is unchecked, and the dropdown is set to 'Never'.

## 2.3 Packages installieren

Packages stellen zusätzliche Funktionen zur Verfügung, welche nicht in der Basisausstattung von R enthalten sind. In dieser Veranstaltung verwenden wir eine Sammlung von Packages (tidyverse) zur Datenmanipulation, zum Importieren von Datenfiles und für Grafiken. Packages werden mit der Funktion `install.packages()` installiert und mit der Funktion `library()` geladen:

```
install.packages("tidyverse")  
library(tidyverse)
```

## 3 Erste Schritte

### 3.1 Operatoren und Funktionen

#### Arithmetische Operatoren

+	Addition
-	Subtraktion
*	Multiplikation
/	Division
^	Potenz

#### Numerische Funktionen

<code>sqrt(x)</code>	Quadratwurzel
<code>log(x)</code>	Natürliche Logarithmus
<code>exp(x)</code>	Exponentialfunktion
<code>abs(x)</code>	Betrag
<code>round(x, digits=n)</code>	Runden auf $n$ Nachkommastellen

#### Logische Operatoren

<code>&lt;(=)</code>	Kleiner (gleich)
<code>&gt;(=)</code>	Größer (gleich)
<code>==</code>	Gleich
<code>!=</code>	Ungleich
<code>x   y</code>	x ODER y
<code>x &amp; y</code>	x UND y

#### Statistische Funktionen

<code>mean(x)</code>	Mittelwert
<code>sd(x)</code>	Standardabweichung
<code>var(x)</code>	Varianz
<code>sum(x)</code>	Summe
<code>min(x)</code>	Minimum
<code>max(x)</code>	Maximum

## 3.2 Variablen definieren

Um die Resultate von Berechnungen zu speichern, können in R Variablen definiert werden, und diesen einen Wert zugewiesen werden. Variablen werden in R zum Beispiel so definiert:

```
my_var <- 4
```

<- ist hier ein spezieller Zuweisungspfeil, und besteht aus einem < Zeichen und einem -. `my_var` ist der Name der Variable. Dieser besteht aus Buchstaben, Zahlen und den Zeichen `_` und/oder `.`. Ein Variablenname muss mit einem Buchstaben beginnen, und darf keine Leerschläge enthalten.

## 3.3 Daten importieren

**CSV Dateien:** dieser Dateityp kann mit der Funktion `read_csv()` importiert werden. Als Argument der Funktion muss der Pfad der Datei angegeben werden:

```
my_data <- read_csv("my_data.csv")
```

Ein Data Frame in R kann mit der Funktion `write_csv()` als csv Datei gespeichert werden. Als erstes Argument der Funktion wird der Name des Data Frames angegeben und als zweites Argument der Pfad, unter dem die Datei abgespeichert wird:

```
write_csv(my_data, file = "my_data.csv")
```

**RData Dateien:** dieser Dateityp kann mit der Funktion `load()` geladen werden:

```
load(file = "my_data.RData")
```

Ein Data Frame in R kann mit der Funktion `save()` als RData Datei gespeichert werden:

```
save(my_data, file = "my_data.RData")
```

## 4 Datenanalyse

### 4.1 Deskriptive Statistik

Mit der Funktion `summary()` werden deskriptive Statistiken für alle Variablen im Datensatz ausgegeben. Diese umfassen Minimum, Maximum, Quartile, Median und Mittelwert. Als Argument der Funktion muss der zu verwendende Datensatz angegeben werden:

```
summary(my_data)
```

Mit der Funktion `summarize()` können für einzelne Variablen deskriptive Kennzahlen berechnet werden. Als erstes Argument der Funktion muss der zu verwendende Datensatz angegeben werden und als zweites Argument die Statistik, die berechnet werden soll. Hierbei kann jede statistische Funktion aus Abschnitt 3.1 verwendet werden. Zum Beispiel wird im Datensatz „my\_data“ der Mittelwert der Variable „var1“ wie folgt berechnet:

```
summarize(my_data, var1_mean = mean(var1))
```

Die Korrelation zweier Variablen kann mit der Option `cor()` berechnet werden:

```
summarize(my_data, var1_var2_cor = cor(var1, var2))
```

### 4.2 Neue Variablen erstellen

Neue Variablen können mit der Funktion `mutate()` aus schon bestehenden Variablen in einem Datensatz gebildet werden. Als erstes Argument der Funktion wird der zu verwendende Datensatz angegeben und als zweites Argument der Befehl, anhand dessen die neue Variable erstellt wird. Zum Beispiel wird im Datensatz „my\_data“ eine Variable „var3“ als Summe aus den bereits vorhandenen Variablen „var1“ und „var2“ wie folgt erstellt:

```
my_data <- mutate(my_data, var3 = var1 + var2)
```

## 4.3 Beobachtungen sortieren und filtern

**Sortieren:** Mit der `arrange()` Funktion können Beobachtungen in aufsteigender oder in absteigender Reihenfolge sortiert werden. Als erstes Argument der Funktion wird der zu verwendende Datensatz angegeben und als zweites Argument die Variable, anhand derer der Datensatz sortiert werden soll. Zum Beispiel wird der Datensatz „my\_data“ in aufsteigender Reihenfolge nach der Variablen „var1“ wie folgt sortiert:

```
my_data <- arrange(my_data, var1)
```

Der Datensatz wird in absteigender Reihenfolge nach der Variablen „var1“ mit dem Zusatz `desc()` sortiert:

```
my_data <- arrange(my_data, desc(var1))
```

**Filtern:** Beobachtungen werden mit der Funktion `filter()` ausgewählt, d.h. damit können Beobachtungen ausgewählt werden, welche gewisse Bedingungen erfüllen. Als erstes Argument der Funktion wird der zu verwendende Datensatz angegeben und als zweites Argument die Bedingung, anhand derer der Datensatz gefiltert werden soll. Zum Beispiel werden aus dem Datensatz „my\_data“ alle Beobachtungen, für die die Variable „var1“ den Wert 2 annimmt, wie folgt ausgewählt:

```
filter(my_data, var1 == 2)
```

## 5 Graphiken

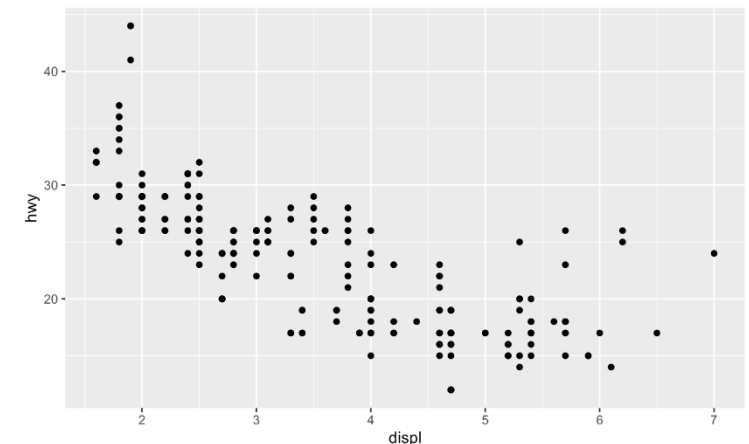
Graphiken können mit dem Package `ggplot2` erstellt werden. Hierfür wird mit der Funktion `ggplot()` ein Plot-Objekt erstellt. Mit der Funktion `aes()` wird definiert, welche Variablen auf der X-, bzw. Y-Achse dargestellt werden sollen. Die Darstellung der Graphik z.B. als Punkt-, Linien- oder Balkendiagramm, wird über die Funktionen `geom_point()`, `geom_line()` oder `geom_bar()` festgesetzt.

## 5.1 Punkt- und Liniendiagramme

**Punkt diagramme** werden mit der Funktion `geom_point()` erstellt. Innerhalb der Funktion `ggplot()` wird als erstes Argument der Datensatz, für den die Graphik erstellt werden soll, spezifiziert (im nachfolgenden Beispiel wird der Datensatz `my_data` verwendet). Als zweites Argument werden die Variablen, die auf der X- bzw. Y-Achse dargestellt werden sollen, definiert (im nachfolgenden Beispiel wird die Variable `displ` auf der X-Achse und die Variable `hwy` auf der Y-Achse abgebildet). Die Funktion `geom_point()` spezifiziert, dass ein Punkt diagramm erstellt werden soll:

```
ggplot(data = my_data, mapping = aes(x = displ, y =  
hwy)) +  
  geom_point()
```

Anwendungsbeispiele: geeignet, um Verteilungen zu einem bestimmten Zeitpunkt darzustellen. Viele Beobachtungspunkte werden in einen Zusammenhang mit einer anderen bestimmten Variablen gebracht (z.B. BIP verschiedener Länder und Handelsvolumen).



**Liniendiagramme** werden mit der Funktion `geom_line()` erstellt. Innerhalb der Funktion `ggplot()` wird als erstes Argument der Datensatz, für den die Graphik erstellt werden soll, spezifiziert. Als zweites Argument werden die Variablen, die auf der X- bzw. Y-Achse dargestellt werden sollen, definiert. Die Funktion `geom_line()` spezifiziert, dass ein Liniendiagramm erstellt werden soll:

```
ggplot(data = my_data, mapping = aes(x = displ, y = hwy)) +  
  geom_line()
```

Anwendungsbeispiele: geeignet, um Veränderungen über einen bestimmten Zeithorizont darzustellen; indexierte Darstellung (Ausgangspunkt = Wert eines bestimmten Basisjahrs), z.B. Handelsvolumen eines Landes von 1950-2020.

## 5.2 Säulen-/Balkendiagramme

**Säulen-/Balkendiagramme** werden mit der Funktion `geom_bar()` erstellt. Innerhalb der Funktion `ggplot()` wird als erstes Argument der Datensatz, für den die Graphik erstellt werden soll, spezifiziert (im nachfolgenden Beispiel wird der Datensatz `my_data` verwendet). Als zweites Argument werden die Variablen, die auf der X- bzw. Y-Achse dargestellt werden sollen, definiert (im nachfolgenden Beispiel wird die Variable `cut` auf der X-Achse und die Variable `freq` auf der Y-Achse abgebildet). Die Funktion `geom_bar()` spezifiziert, dass ein Säulen-/Balkendiagramm erstellt werden soll.

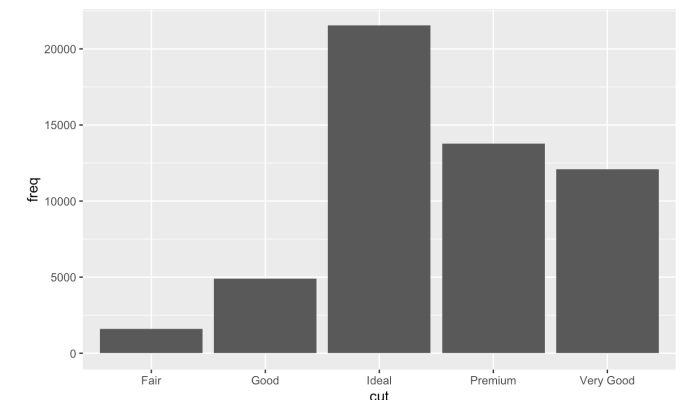
### Säulendiagramm:

```
ggplot(data = my_data, mapping = aes(x = cut, y = freq)) +  
  geom_bar(stat = "identity")
```

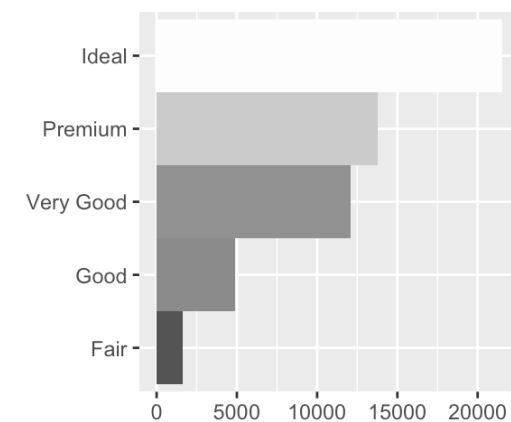
### Balkendiagramm:

```
ggplot(data = my_data, mapping = aes(x = cut, y = freq)) +  
  geom_bar(stat = "identity") + coord_flip()
```

Anwendungsbeispiele: geeignet für wenige Ausprägungsmerkmale und direkten Vergleich zwischen bestimmten Gruppierungen (z.B. Handelsvolumen der G7 Staaten im Vergleich), horizontale (Balken) und vertikale (Säulen) Darstellung möglich.



Säulendiagramm



Balkendiagramm

## Literatur

Grolemund, Garrett (2014): Hands-On Programming with R. 1st Edition. O'Reilly Media, Inc.

Wickham, Hadley und Garrett Grolemund (2017): R for Data Science. 1st Edition. O'Reilly Media, Inc.

Online Version: <http://r4ds.had.co.nz>