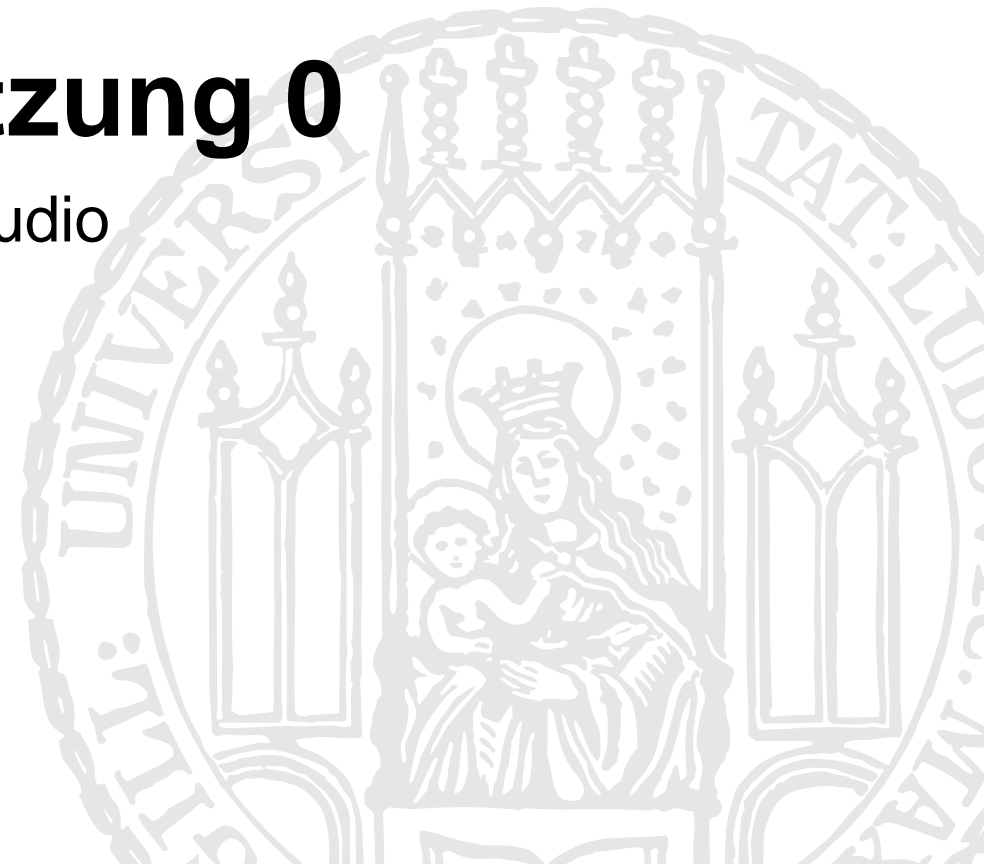


Datenanalyse – Sitzung 0

Einführung in R und RStudio

Institut für Kommunikationswissenschaft und Medienforschung
Ludwig-Maximilians-Universität München





- Einstieg in die Datenauswertung mit R
- Ergänzung und Nachbereitung der Vorlesung
*Einführung in die Statistik für Kommunikationswissenschaftler*innen*
- Praktische Anwendung der gelernten Theorie unter Benutzung von R
- Vorbereitung auf die Klausur

Ablauf der Sitzung

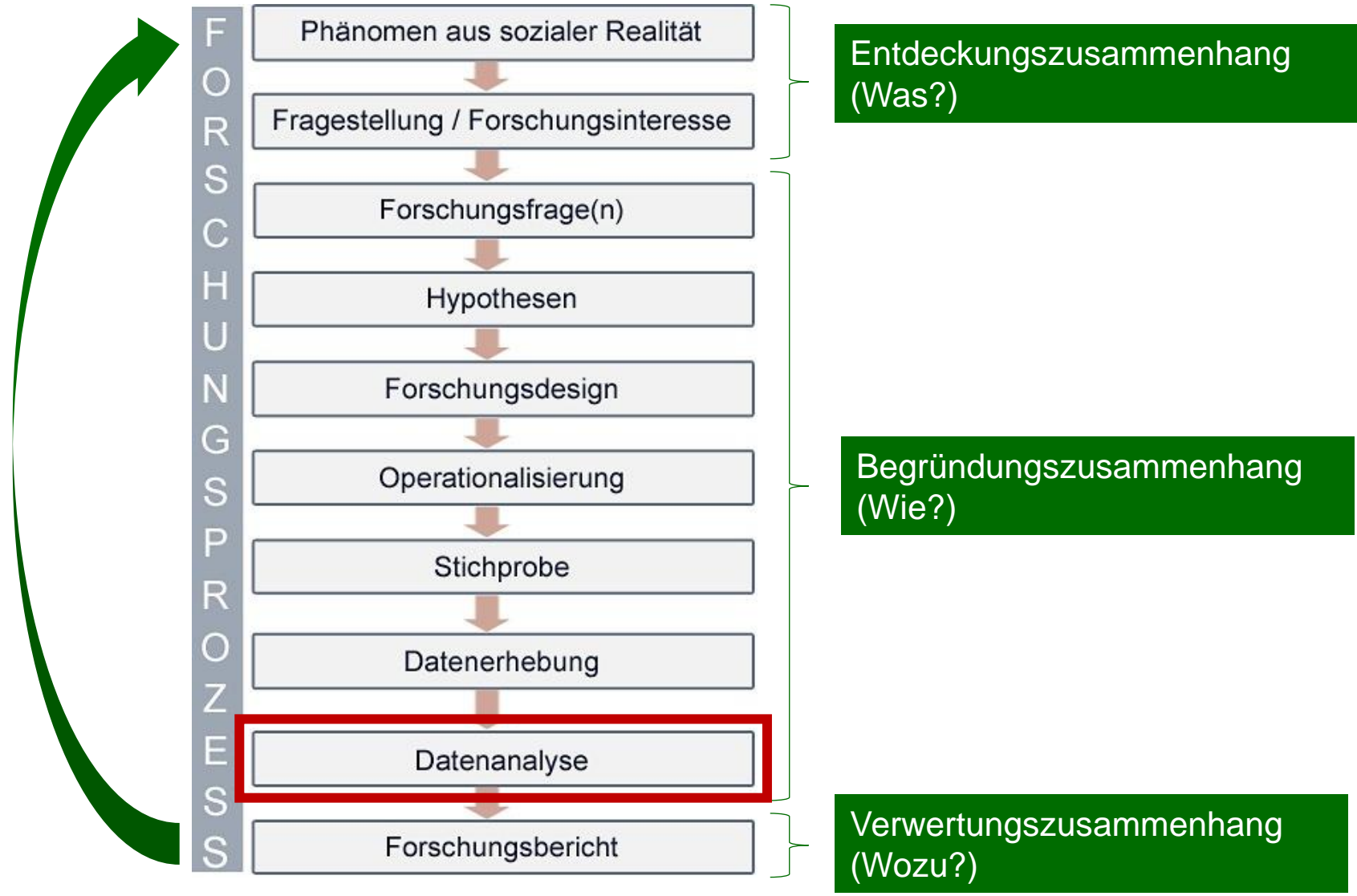
1. Einführung
2. Getting staRted: R und die RStudio-Oberfläche
3. Erste praktische Schritte in R: R als Taschenrechner nutzen
4. R zum Erstellen von Datensätzen verwenden
5. Fälle und Variablen in Dataframes auswählen
6. Wiederholung: Skalenniveaus
7. Wo bekomme ich zusätzliche Hilfe am IfKW?
8. Ergänzende Hilfen und Literatur

1. EINFÜHRUNG

Was ist Datenanalyse und wozu braucht man sie eigentlich?

Um interessante Fragen beantworten zu können, benötigt man als Wissenschaftler*in zwei Dinge:

- ① Daten
- ② und eine Erklärung für diese Daten



Was ist Datenanalyse und wozu braucht man sie eigentlich?

Datenanalyse ist

- die Anwendung **statistischer Methoden**,
- mit denen aus vorliegenden Einzeldaten zusammenfassende Informationen (**Kenngößen**) gewonnen
- und tabellarisch oder grafisch **dokumentiert** werden.



2. GETTING STARTED: R UND DIE RSTUDIO-OBERFLÄCHE

R – was ist das? Die R-Statistiksoftware

- Ein weit verbreitetes Open-Source-Programm für die Auswertung quantitativer Daten (mittlerweile der Standard in vielen Branchen)
- Ermöglicht komplexe Datenmanipulationen, Analysen und visuelle Darstellungen
- Wer R installiert, installiert automatisch das Standard RGUI (GUI = **G**raphical **U**ser **I**nterface bzw. „grafische Bedienoberfläche“).
 - Achtung: Auch wenn R prinzipiell bereits über dieses GUI bedienbar ist, gibt es noch bessere Bedienoberflächen, die einem die Arbeit mit R weiter erleichtern. Eine solche alternative Bedienoberfläche (RStudio) besprechen wir auf der übernächsten Folie.

R – Wo sind R-Kenntnisse wichtig?

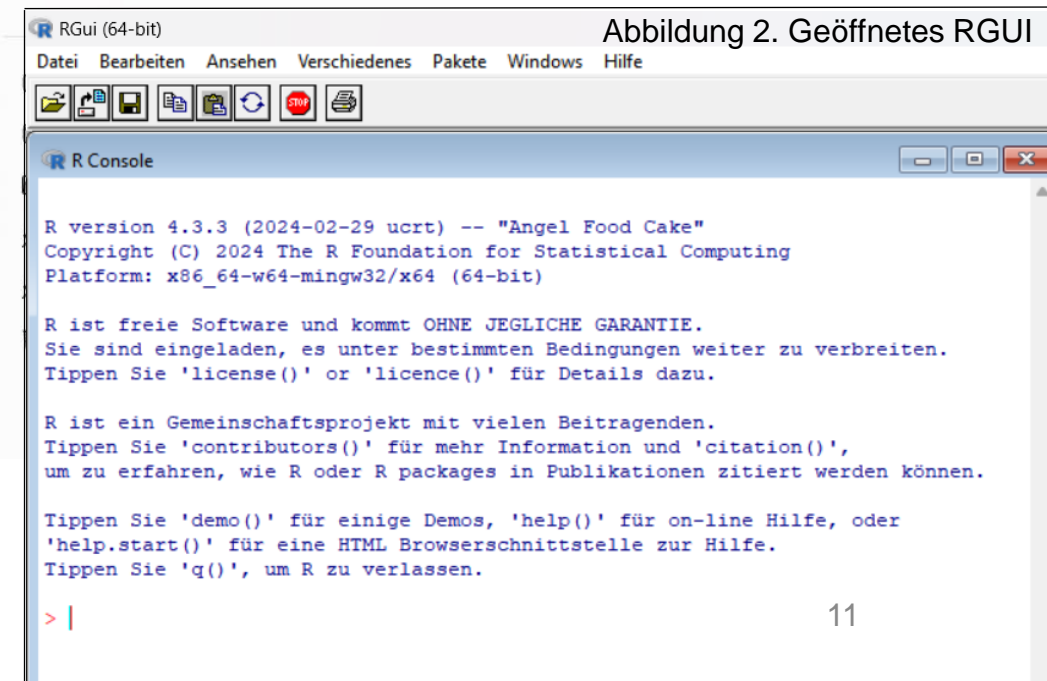
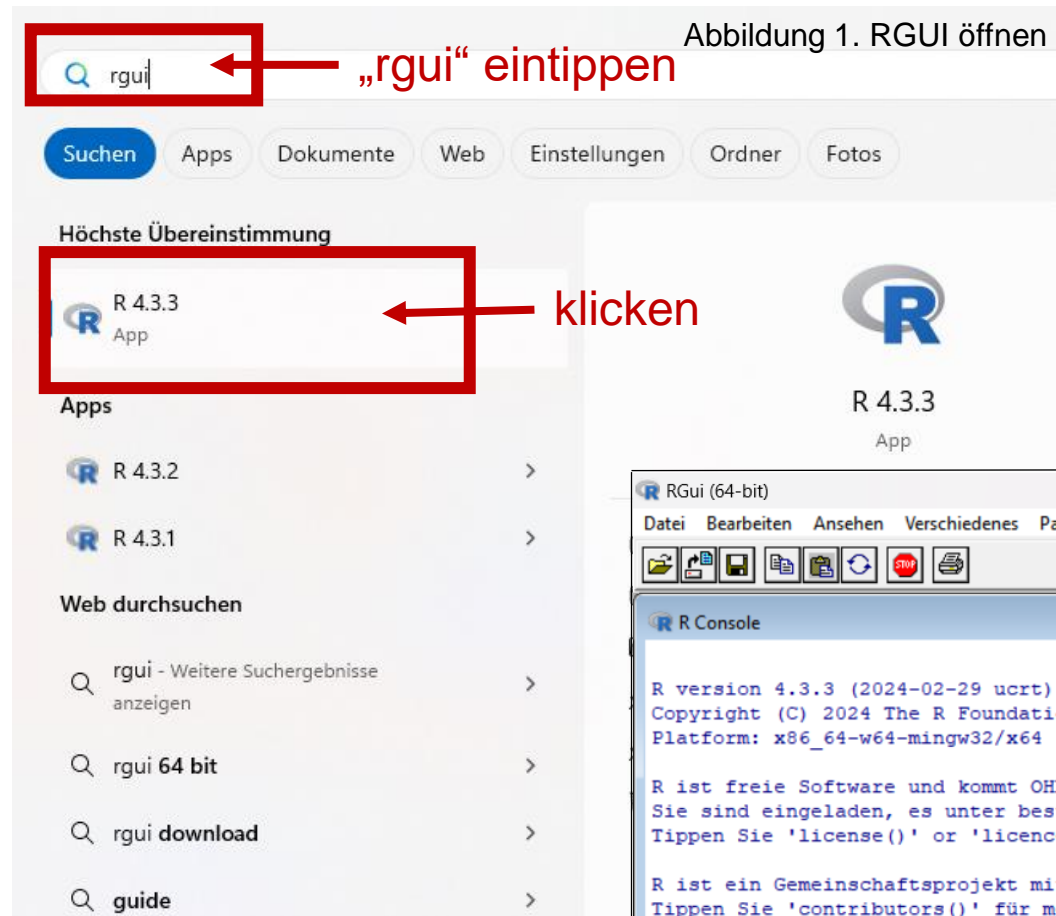
Kurzfristig: in der anstehenden Statistik-Klausur

Langfristig:

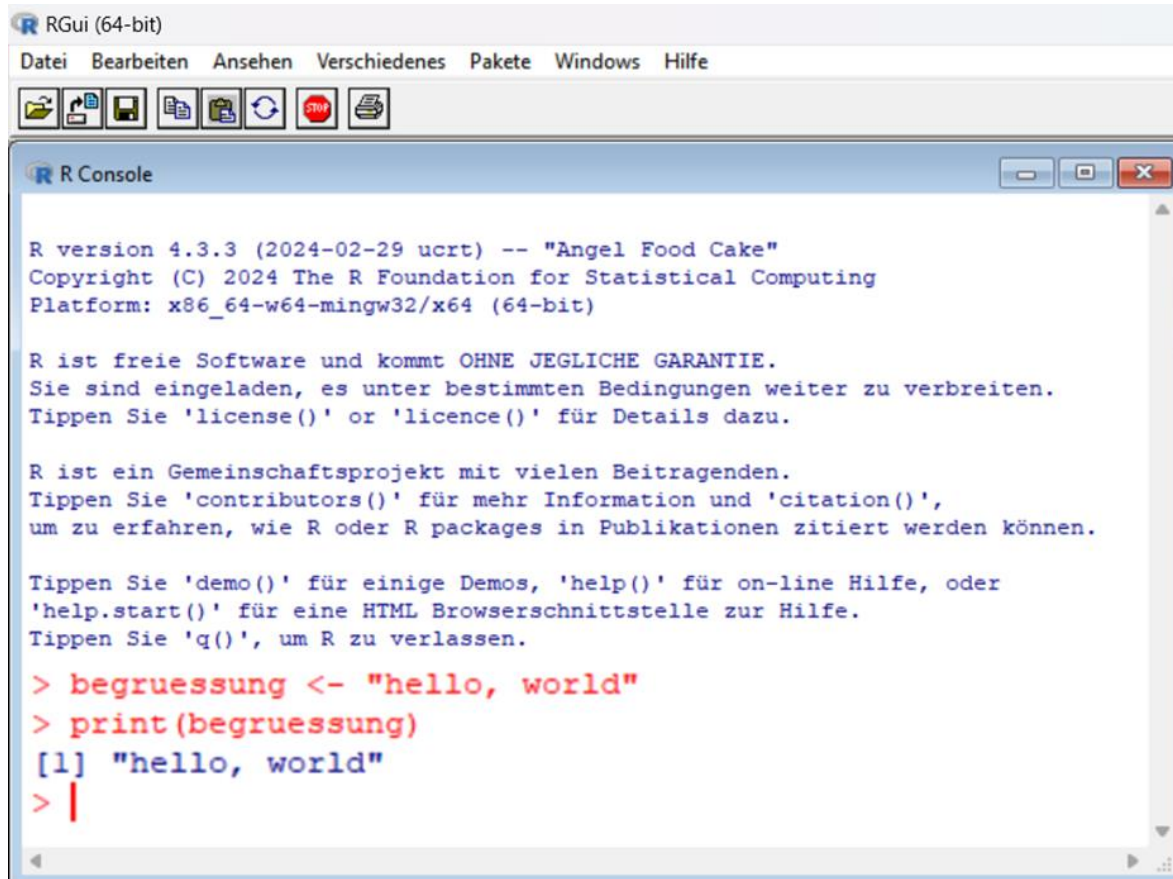
- In künftigen Lehrveranstaltungen (bei empirischen Forschungsprojekten)
- Im Rahmen einer empirischen BA-/MA-Arbeit
- Im Berufsleben – insbesondere in Branchen wie Markt-, Meinungs-, Sozial- und PR-Forschung, wo Unternehmen oft Hunderte bis Tausende von Euros für R-Kurse pro Mitarbeiter*in ausgeben. Sowohl der Umgang mit Statistik als auch der Umgang mit R steigern Ihren Wert auf dem Arbeitsmarkt deutlich.

Wie rufe ich die Standard RGUI auf?

- Auf Windows reicht es aus, wenn Sie “rgui” in die Suchfunktion eingeben.
- Unter den Suchtreffern finden Sie alle R-Installationen, die sich aktuell auf Ihrem Gerät befinden.
- Klicken Sie auf das RGUI derjenigen R-Installation, die Sie öffnen möchten.
- Wie Sie erkennen können, sieht das RGUI etwas altmodisch und wenig benutzerfreundlich aus.



R – wie funktioniert das?



```

RGui (64-bit)
Datei Bearbeiten Ansehen Verschiedenes Pakete Windows Hilfe

R Console

R version 4.3.3 (2024-02-29 ucrt) -- "Angel Food Cake"
Copyright (C) 2024 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R ist freie Software und kommt OHNE JEGLICHE GARANTIE.
Sie sind eingeladen, es unter bestimmten Bedingungen weiter zu verbreiten.
Tippen Sie 'license()' or 'licence()' für Details dazu.

R ist ein Gemeinschaftsprojekt mit vielen Beitragenden.
Tippen Sie 'contributors()' für mehr Information und 'citation()',
um zu erfahren, wie R oder R packages in Publikationen zitiert werden können.

Tippen Sie 'demo()' für einige Demos, 'help()' für on-line Hilfe, oder
'help.start()' für eine HTML Browserschnittstelle zur Hilfe.
Tippen Sie 'q()', um R zu verlassen.

> begruessung <- "hello, world"
> print(begruessung)
[1] "hello, world"
> |
  
```

Zum Beispiel weist dieser Befehl die Buchstaben “hello, world“ einem R-Objekt namens *begruessung* zu, indem der *Zuweisungsoperator* <- verwendet wird. Im Anschluss können Sie den zugewiesenen Wert mittels des Funktionsaufrufs „*print()*“ anzeigen.

- In R weisen Sie Werte (zum Beispiel einzelne Zahlen / Buchstaben, mehrere Zahlen / Buchstaben oder ganze Datendateien) R-Objekten zu, um mit ihnen weiterzuarbeiten.

Merken Sie sich:

- Alles, was existiert, ist ein R-Objekt
(hier: *begruessung*)
- Alles, was passiert, ist ein Funktionsaufruf
(hier: *print()*)

RStudio – was ist das?



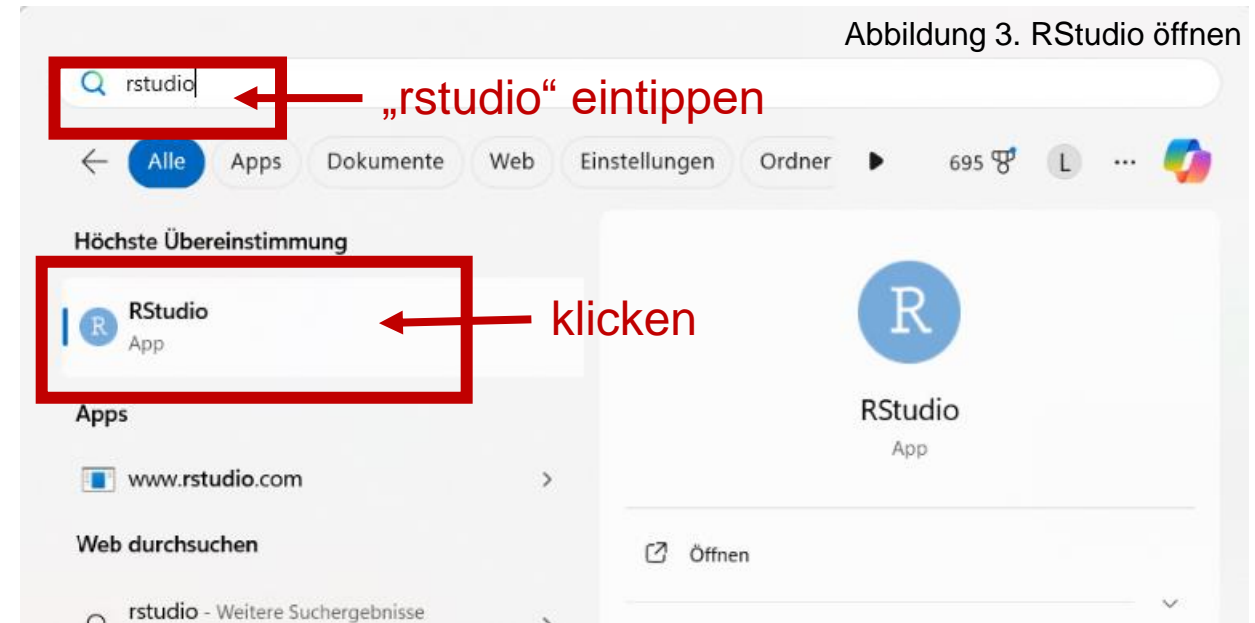
RStudio GUI

- Eine grafische Bedienoberfläche für R, die das Arbeiten mit R im Vergleich zum Standard RGUI erleichtert
- Was heißt das konkret: RStudio hat mehr Drop-Down Menüs, kann verschiedene Fenster gleichzeitig offen haben und bietet Optionen, um sich ein personalisiertes Layout zu erstellen
- Beachten Sie: R und RStudio müssen immer separat installiert werden – erst R, dann RStudio

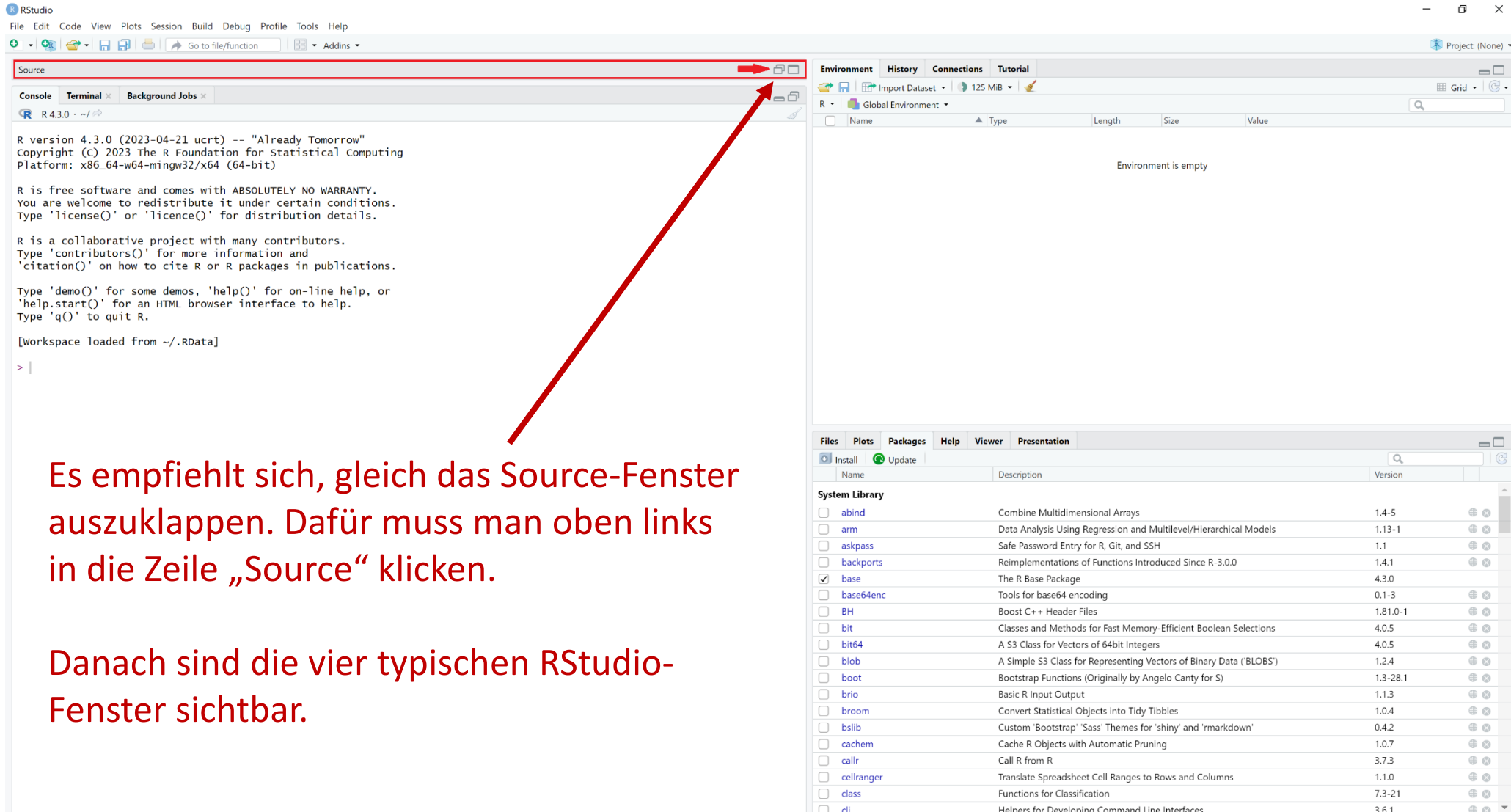
RStudio – das erste Mal öffnen

RStudio wird durch einen Klick auf das RStudio-Icon geöffnet

- Beim ersten Öffnen von RStudio erscheinen drei Fenster – Ein großes auf der linken Seite und zwei kleinere auf der rechten Seite
- Der spezifische Aufbau von RStudio kann sich je nach Version und Betriebssystem unterscheiden



RStudio-Oberfläche – nach dem ersten Öffnen



The screenshot shows the RStudio interface with the following components:

- Source window:** Contains the R version information and license text. A red arrow points to its icon in the top-left corner.
- Console:** Shows the R version 4.3.0 (2023-04-21 ucrt) and the license text.
- Environment:** Shows the Global Environment with the message "Environment is empty".
- Packages:** Shows a list of installed and available packages, including the System Library.

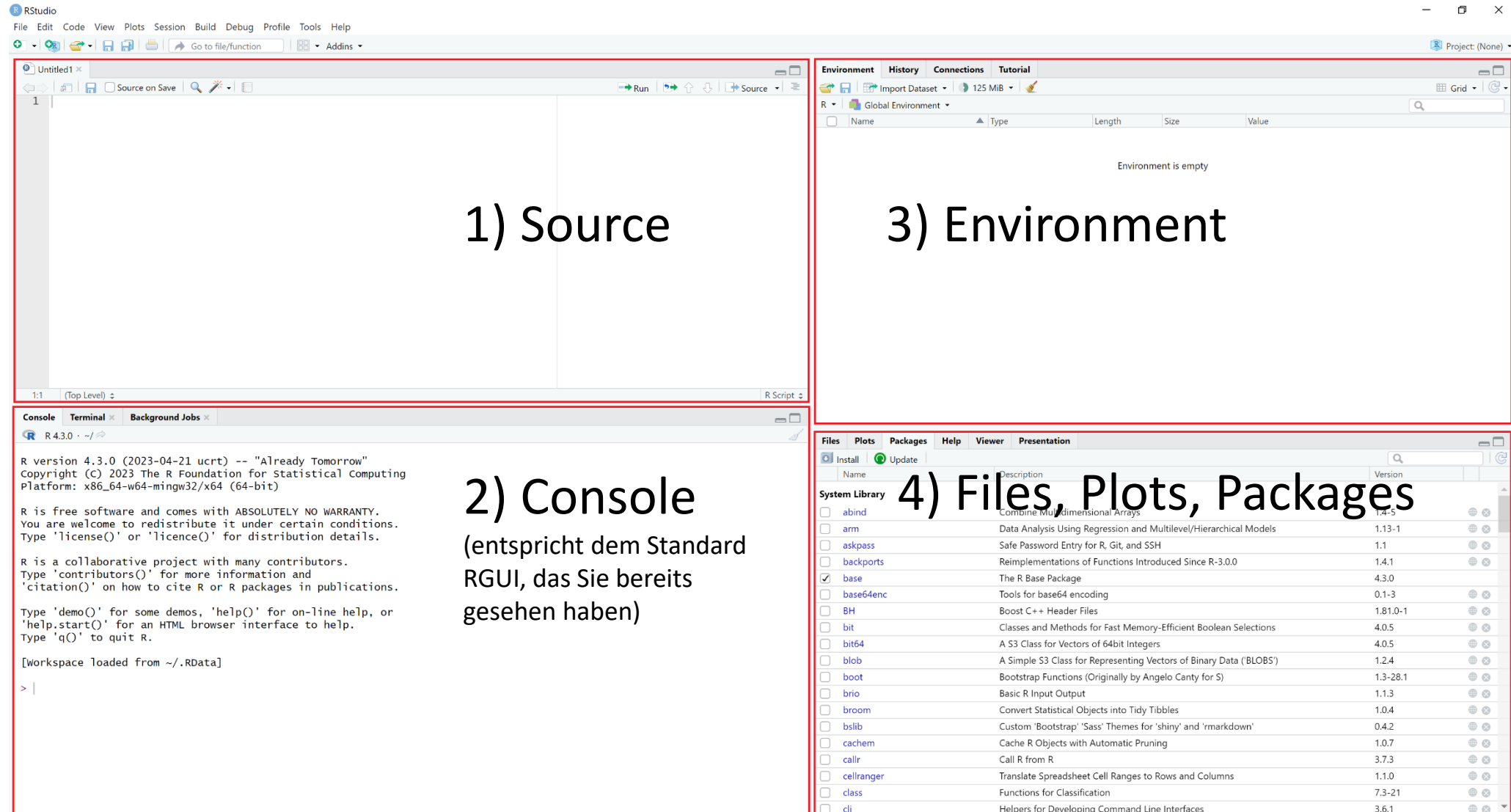
System Library

Name	Description	Version
<input type="checkbox"/> abind	Combine Multidimensional Arrays	1.4-5
<input type="checkbox"/> arm	Data Analysis Using Regression and Multilevel/Hierarchical Models	1.13-1
<input type="checkbox"/> askpass	Safe Password Entry for R, Git, and SSH	1.1
<input type="checkbox"/> backports	Reimplementations of Functions Introduced Since R-3.0.0	1.4.1
<input checked="" type="checkbox"/> base	The R Base Package	4.3.0
<input type="checkbox"/> base64enc	Tools for base64 encoding	0.1-3
<input type="checkbox"/> BH	Boost C++ Header Files	1.81.0-1
<input type="checkbox"/> bit	Classes and Methods for Fast Memory-Efficient Boolean Selections	4.0.5
<input type="checkbox"/> bit64	A S3 Class for Vectors of 64bit Integers	4.0.5
<input type="checkbox"/> blob	A Simple S3 Class for Representing Vectors of Binary Data ('BLOBS')	1.2.4
<input type="checkbox"/> boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-28.1
<input type="checkbox"/> brio	Basic R Input Output	1.1.3
<input type="checkbox"/> broom	Convert Statistical Objects into Tidy Tibbles	1.0.4
<input type="checkbox"/> bslib	Custom 'Bootstrap' 'Sass' Themes for 'shiny' and 'rmarkdown'	0.4.2
<input type="checkbox"/> cachem	Cache R Objects with Automatic Pruning	1.0.7
<input type="checkbox"/> callr	Call R from R	3.7.3
<input type="checkbox"/> cellranger	Translate Spreadsheet Cell Ranges to Rows and Columns	1.1.0
<input type="checkbox"/> class	Functions for Classification	7.3-21
<input type="checkbox"/> cli	Helpers for Developing Command Line Interfaces	3.6.1

Es empfiehlt sich, gleich das Source-Fenster auszuklappen. Dafür muss man oben links in die Zeile „Source“ klicken.

Danach sind die vier typischen RStudio-Fenster sichtbar.

RStudio-Oberfläche – die vier Fenster



The screenshot shows the RStudio interface with four windows highlighted by red boxes and numbered 1 to 4:

- 1) Source:** The top-left window showing a script editor with a single line of code.
- 2) Console:** The bottom-left window showing the R command prompt with the R version 4.3.0 startup message and a prompt for user input.
- 3) Environment:** The top-right window showing the Environment pane, which is currently empty.
- 4) Files, Plots, Packages:** The bottom-right window showing the Files, Plots, and Packages pane, displaying a list of installed R packages.

1) Source

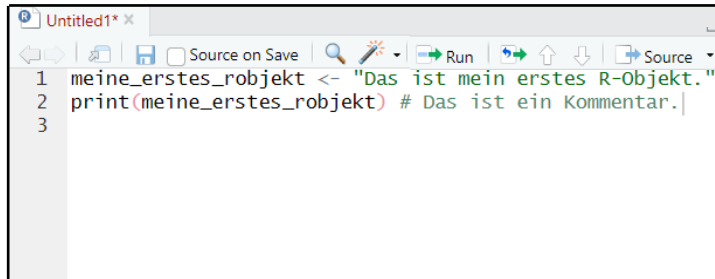
3) Environment

2) Console
(entspricht dem Standard
RGUI, das Sie bereits
gesehen haben)

4) Files, Plots, Packages

RStudio-Oberfläche – die vier Fenster

Erstens: Source

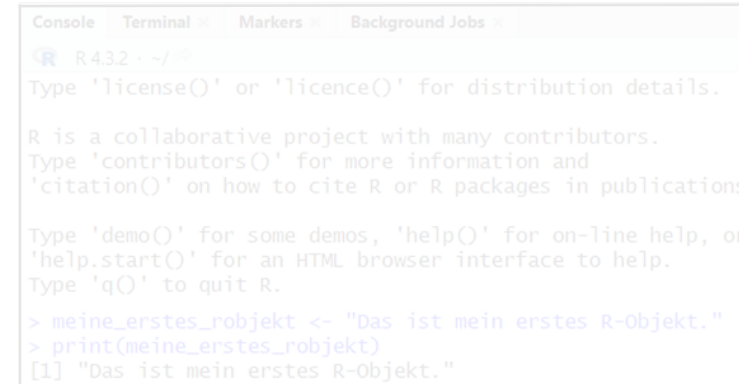


```

1 meine_erstes_robjekt <- "Das ist mein erstes R-Objekt."
2 print(meine_erstes_robjekt) # Das ist ein Kommentar.
3

```

Zweitens: Console



```

R 4.3.2 ~/>
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> meine_erstes_robjekt <- "Das ist mein erstes R-Objekt."
> print(meine_erstes_robjekt)
[1] "Das ist mein erstes R-Objekt."

```

Was passiert in der Source?

Das ist das „Atelier von R“, d.h. der Ort, an dem Sie Ihre Befehle in einem **Skript** entwickeln („coden“), z.B. wenn Sie etwas berechnen.

- Coden: Zuweisen von Werten zu benannten R-Objekten.
- Kommentieren: Möglichkeit, den eigenen Code zu kommentieren für bessere Verständlichkeit. Nutzung von Hashtags (#) für einzelne Zeilen (*hier: # Das ist ein Kommentar*).
- Ausführen: Ausführung von markiertem Code durch Klick auf Run oder Nutzung von Tastenkombinationen (Shortcut: Strg + Enter).

Was passiert in der Console?

Das ist die „Bühne von R“, d.h. der Ort, an dem Sie Ihre Ergebnisse des Source-Fensters sehen, z.B. die Ergebnisse Ihrer Berechnungen.

- Code ausführen: Zeigt, was passiert, wenn der Code aus dem Source-Fenster ausgeführt wird. Hier werden auch Fehlermeldungen angezeigt (*hier: in blauer Schrift*).
- Anzeigen der Ergebnisse: Zeigt die Ausgabe von R-Objekten (*hier: in schwarzer Schrift*).
- Interaktion: Erlaubt direkte Eingabe und Ausführung von R-Code (z.B. um Code zu testen). (*hier nicht dargestellt*)

RStudio-Oberfläche – die vier Fenster

Erstens: Source



```

1 meine_erstes_robjekt <- "Das ist mein erstes R-Objekt."
2 print(meine_erstes_robjekt) # Das ist ein Kommentar.
3

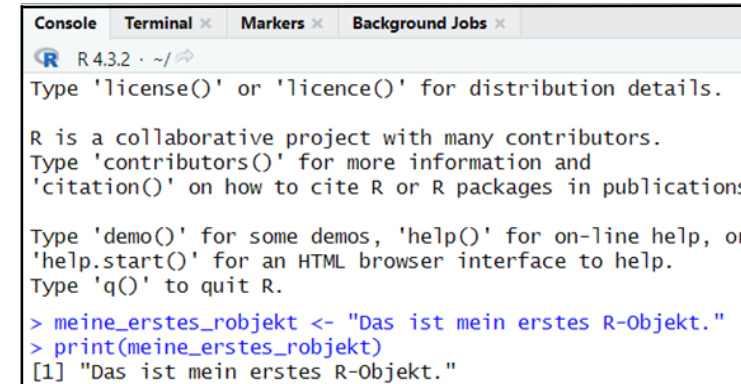
```

Was passiert in der Source?

Das ist das „Atelier von R“, d.h. der Ort, an dem Sie Ihre Befehle in einem Skript entwickeln („coden“), z.B. wenn Sie etwas berechnen.

- Coden: Zuweisen von Werten zu benannten R-Objekten.
- Kommentieren: Möglichkeit, den eigenen Code zu kommentieren für bessere Verständlichkeit. Nutzung von Hashtags (#) für einzelne Zeilen (*hier: # Das ist ein Kommentar*).
- Ausführen: Ausführung von markiertem Code durch Klick auf Run oder Nutzung von Tastenkombinationen (z.B. Strg+Enter).

Zweitens: Console



```

R 4.3.2 · ~/
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> meine_erstes_robjekt <- "Das ist mein erstes R-Objekt."
> print(meine_erstes_robjekt)
[1] "Das ist mein erstes R-Objekt."

```

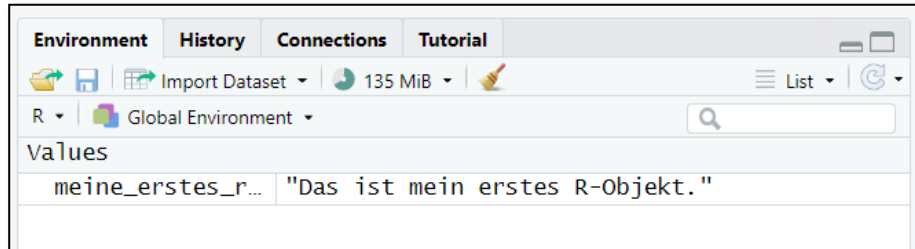
Was passiert in der Console?

Das ist die „Bühne von R“, d.h. der Ort, an dem Sie Ihre Ergebnisse des Source-Fensters sehen, z.B. die Ergebnisse Ihrer Berechnungen.

- Code ausführen: Zeigt, was passiert, wenn der Code aus dem Source-Fenster ausgeführt wird. Hier werden auch Fehlermeldungen angezeigt (*hier: in blauer Schrift*).
- Anzeigen der Ergebnisse: Zeigt die Ausgabe von R-Objekten (*hier: in schwarzer Schrift*).
- Interaktion: Erlaubt direkte Eingabe und Ausführung von R-Code (z.B. um Code zu testen). (*hier nicht dargestellt*)

RStudio-Oberfläche – die vier Fenster

Drittens: Environment

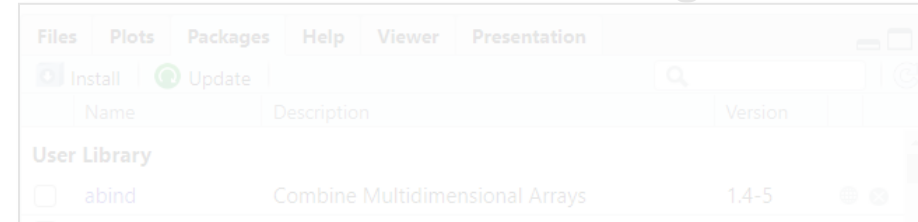


Was passiert in der Environment?

Das ist das „Lagerhaus bzw. Gedächtnis von R“, d.h. der Ort, an dem alle R-Objekte abgelegt und sortiert werden.

- R-Objekte lagern und ansehen: Ermöglicht Inspektion von R-Objekten. Entweder durch das Aufklappen eines Datensatzes (*hier nicht dargestellt: blauer Pfeil*) oder durch Doppelklick auf den Namen des R-Objekts.

Viertens: Files, Plots, Packages

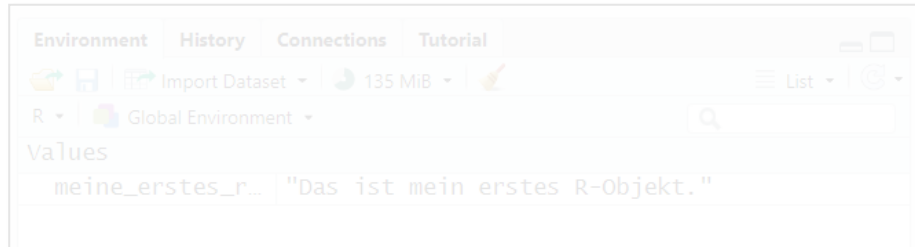


Was passiert in Files, Plots, Packages?

Hier findet sich der wichtige Rest. Es können Daten visualisiert, Hilfe kann gefunden, Packages aktualisiert werden (mehr zu all dem später in diesem Seminar).

RStudio-Oberfläche – die vier Fenster

Drittens: Environment

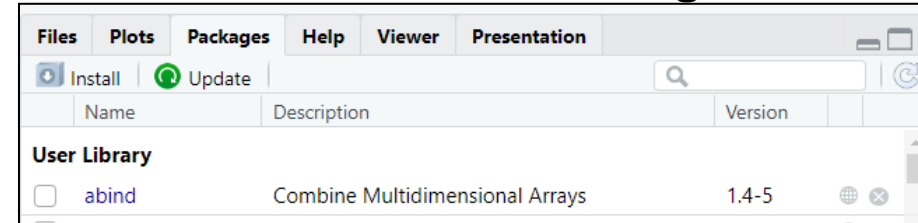


Was passiert in der Environment?

Das ist das „Lagerhaus bzw. Gedächtnis von R“, d.h. der Ort, an dem alle R-Objekte abgelegt und sortiert werden.

- R-Objekte lagern und ansehen: Ermöglicht Inspektion von R-Objekten. Entweder durch das Aufklappen eines Datensatzes (*hier nicht dargestellt: blauer Pfeil*) oder durch Doppelklick auf den Namen des R-Objekts.

Viertens: Files, Plots, Packages



Was passiert in Files, Plots, Packages?

Hier findet sich der wichtige Rest. Es können Daten visualisiert, Hilfe kann gefunden, Packages aktualisiert werden (mehr zu all dem später in diesem Seminar).

3. ERSTE PRAKTISCHE SCHRITTE IN R: R ALS TASCHENRECHNER NUTZEN

Ein wichtiges Wort vorweg

Die Rechtschreibung und Grammatik in R folgt strengen Regeln (im Fachjargon sagen wir: R ist „case-sensitive“). Wenn Sie diese Regeln nicht exakt beachten, dann werden Sie häufiger eine Fehlermeldung erhalten.

- R achtet auf Groß- und Kleinschreibung. Beispiel: `print()` \neq `Print()`.
- R achtet auch auf Leerzeichen und Textzeilenumbrüche. Beispiel: `print()` \neq `print ()`.
- R achtet auf die Nutzung von Anführungszeichen. Beispiel: `print(5)` \neq `print("5")`.
- R nutzt englische, gerade Anführungszeichen. Beispiel: `print("5")` \neq `print(`5`)`.

Die meisten Fehlermeldungen in R entstehen, weil eine der obigen Regeln missachtet wird. **Achten Sie darauf, den vorgeführten Code sauber zu übernehmen. Ganz konkret: Achten Sie darauf, dass Leerzeichen, Groß- und Kleinschreibung sowie Anführungszeichen korrekt gesetzt sind.**

Nutzung von R für einfache Rechenaufgaben

- Eines der ersten Dinge, die jeder in R lernt, ist, R als Taschenrechner zu verwenden.
- Sie haben Zugang zu vielen mathematischen Operatoren in R: z.B. +, -, *, /, ^.

Lassen Sie es uns gemeinsam ausprobieren:

1. Addition: `4 7 + 5` Input im Source-Fenster (oder alternativ Console)

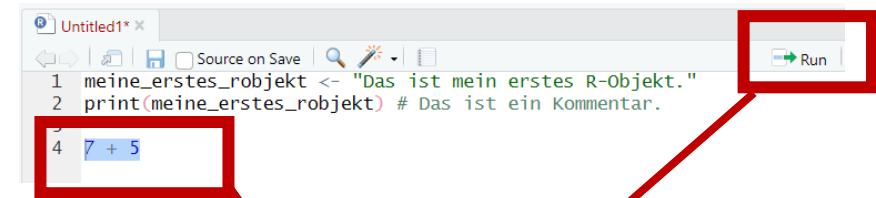
`> 7 + 5`
`[1] 12` Output im Console-Fenster

2. Subtraktion: `5 12 - 7` Input im Source-Fenster (oder alternativ Console)

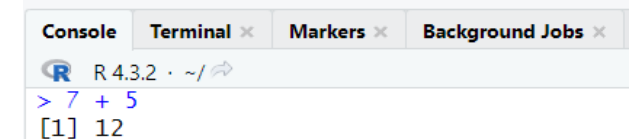
`> 12 - 7` Output im Console-Fenster
`[1] 5`

3. Exponieren: `6 3^3` Input im Source-Fenster (oder alternativ Console)

`> 3^3` Output im Console-Fenster
`[1] 27`



Um Befehle im Source-Fenster auszuführen, klicken Sie zunächst in die auszuführende Zeile und anschließend auf „Run“ (Shortcut: Strg + Enter).
 Dann sehen Sie den Output in der Console:



Nutzung von Variablen in R für Rechenaufgaben

In aller Regel werden Sie R nicht als Taschenrechner benutzen wollen, sondern damit komplexere statistische Auswertungen berechnen. Dafür müssen Sie den **Zuweisungsoperator** „<-“ nutzen lernen.

Zuweisung von Werten in R-Objekte mittels „<-“

- Mit dem Zuweisungsoperator „<-“ können Sie R-Objekten (sog. Variablen) Zahlen oder Buchstaben zuweisen
- In R können die Namen von R-Objekten (sog. Variablennamen) Zahlen, Buchstaben sowie spezielle Zeichen wie Punkt (.) und Unterstrich (_) enthalten, aber sie müssen mit Buchstaben beginnen. Vermeiden Sie zudem im besten Fall deutsche Umlaute und Sonderzeichen (wie @) in Variablennamen, um Problemen vorzubeugen.

- Gute Option für Variablennamen:

- `meine_erste_Zahl <- 3`  Merke: Der sog. „snake_case“ ist Standard für R !



- Nicht als Variablennamen erlaubt (führt zu Fehlermeldungen):

- `.meine.erste.zahl <- 3`

- `_meine_erste_zahl <- 3`

- `meine-erste-zahl <- 3`  Merke: Mit dem sog. „kebab-case“ kann R nicht gut umgehen !



Nutzung von Variablen in R für Rechenaufgaben

Nun können wir Zahlen (numerische Werte) in **Variablen** ablegen, um damit weiterzurechnen! (und mittels `print()` anzuzeigen)

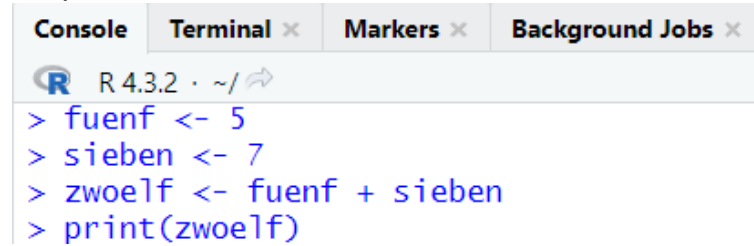
Verwendung von Variablen in Berechnungen:

- Zahlen zu Variablen zuweisen (numerische Werte in der Variablen hinterlegen):

Input im Source-Fenster:

```
17 # Rechnen mit Variablen:
18 fuenf <- 5
19 sieben <- 7
20 zweielf <- fuenf + sieben
21 print(zwoelf)
```

Output in der Console:



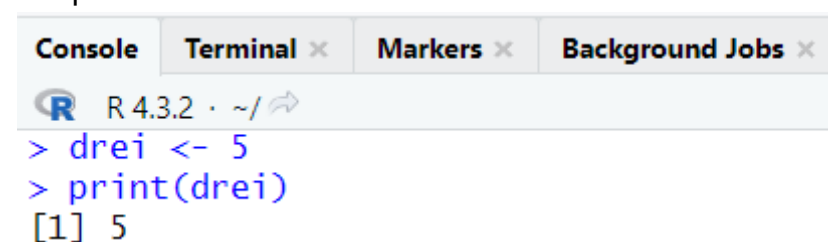
```
Console Terminal x Markers x Background Jobs x
R R 4.3.2 · ~/ ↵
> fuenf <- 5
> sieben <- 7
> zweielf <- fuenf + sieben
> print(zwoelf)
```

- Achtung, freie Wahl bei der Benennung der Variablen (Versuchen Sie aber bitte immer möglichst passende Variablennamen zu wählen):

Input im Source-Fenster:

```
23 # freie wahl bei der Benennung von Variablen:
24 drei <- 5
25 print(drei)
```

Output in der Console:



```
Console Terminal x Markers x Background Jobs x
R R 4.3.2 · ~/ ↵
> drei <- 5
> print(drei)
[1] 5
```

Nutzung von Vektoren in R für Rechenaufgaben

Man kann nicht nur eine, sondern gleich mehrere numerische Werte bzw. einen ganzen Zahlenstrahl in Variablen ablegen, um damit weiterzurechnen. Solche Variablen werden **Vektoren** genannt.

Verwendung von Vektoren in Berechnungen:

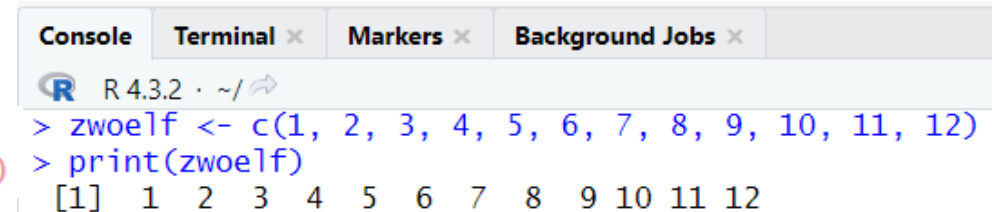
- Einen Vektor mittels `c()`-Funktion erstellen und in eine Variable ablegen:

Eselsbrücke: `c()` steht für „combine“.

Input im Source-Fenster:

```
27 # Erstellen eines Vektors aus mehreren Zahlen:
28 zwoelf <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
```

Output in der Console:



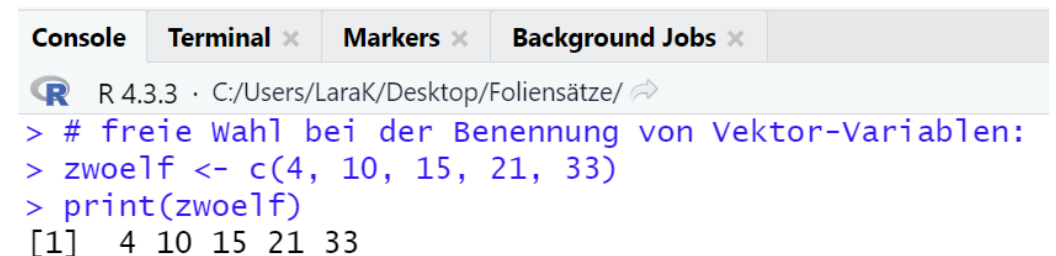
```
Console Terminal x Markers x Background Jobs x
R 4.3.2 · ~/
> zwoelf <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
> print(zwoelf)
[1] 1 2 3 4 5 6 7 8 9 10 11 12
```

- Achtung, auch hier freie Wahl bei der Benennung der Variablen (Versuchen Sie aber bitte immer möglichst passende Variablennamen zu wählen):

Input im Source-Fenster:

```
31 # freie Wahl bei der Benennung
32 zwoelf <- c(4, 10, 15, 21, 33)
```

Output in der Console:



```
Console Terminal x Markers x Background Jobs x
R 4.3.3 · C:/Users/LaraK/Desktop/Foliensätze/
> # freie Wahl bei der Benennung von Vektor-Variablen:
> zwoelf <- c(4, 10, 15, 21, 33)
> print(zwoelf)
[1] 4 10 15 21 33
```

Nutzung von Vektoren in R für Rechenaufgaben

Sie können mathematische Operationen auf Vektoren anwenden (z. B. +, -, * und /). Erstellen wir zwei Vektoren „Gewicht“ und „Größe“, die die Gewichts- und Größenmaße von 6 befragten Personen enthalten. Zum Beispiel wiegt die erste Person 60 kg und ist 1,75 m groß. Beachten Sie, dass R per Default die englische Zeichensetzung verwendet (d.h. Dezimalpunkte statt Kommata).

Verwendung von mathematischen Operatoren mit Vektoren:

- Erstellung des Gewicht- und Größe-Vektors:

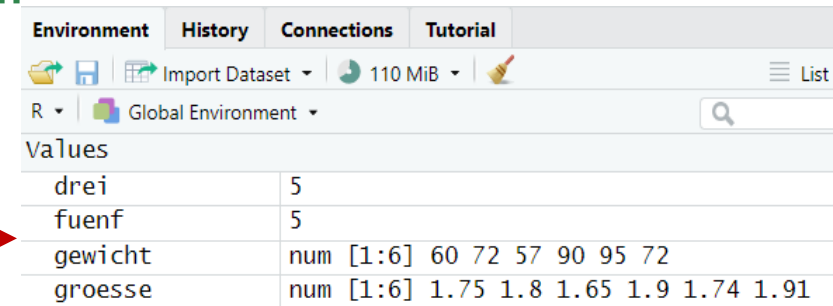
```
35 # Anwendung von mathematischen Operatoren auf Vektoren:
36 gewicht <- c(60, 72, 57, 90, 95, 72)
37 groesse <- c(1.75, 1.80, 1.65, 1.90, 1.74, 1.91)
```

- Berechnung eines neuen BMI-Vektors mittels der BMI-Formel:

```
38 BMI <- gewicht / groesse^2
39 print(BMI)
```

```
Console Terminal x Markers x Background Jobs x
R 4.3.2 · ~/
> BMI <- gewicht / groesse^2
> print(BMI)
[1] 19.59184 22.22222 20.93664 24.93075 31.37799 19.73630
```

↳ Der erste Befragte mit 1,75m und 60kg hat also einen BMI von 19,59184



Environment	History	Connections	Tutorial
R Global Environment			
Values			
drei		5	
fuenf		5	
gewicht	num [1:6]	60 72 57 90 95 72	
groesse	num [1:6]	1.75 1.8 1.65 1.9 1.74 1.91	

↑
In der Environment werden nun die numerischen (*num*) Vektor-Variablen mit 6 Elementen (*[1:6]*) angezeigt

Elemente in einem Vektor auswählen

Im vorangegangenen Beispiel des BMI-Vektors bekommen wir den BMI von allen 6 Befragten angezeigt. Wir können aber auch zielgerichtet nur einen einzigen Wert von einem oder mehreren Befragten abrufen, der uns besonders interessiert.

Selektieren von Elementen eines Vektors mittels Indexierung per Klammer []:

- Auswahl des BMIs des ersten Befragten:

```
41 # Auswahl von Elementen eines Vektors
42 BMI[1] # BMI des ersten Befragten
```

```
Console Terminal x Markers x Background Jobs x
R R 4.3.2 · ~/ ↵
> BMI[1] # BMI des ersten Befragten
[1] 19.59184
```

- Auswahl der BMIs aller Befragten außer dem ersten Befragten mittels „-“:

```
43 BMI[-1] # alle BMIs außer vom ersten Befragten
```

```
R R 4.3.2 · ~/ ↵
> BMI[-1] # alle BMIs außer vom ersten Befragten
[1] 22.22222 20.93664 24.93075 31.37799 19.59184
```

- Auswahl des BMIs des zweiten bis vierten Befragten mittels „:“:

```
44 BMI[2:4] # BMI von Befragten 2, 3 und 4
```

```
Console Terminal x Markers x Background Jobs x
R R 4.3.2 · ~/ ↵
> BMI[2:4] # BMI von Befragten 2, 3 und 4
[1] 22.22222 20.93664 24.93075
```

Numerische und Character-Vektoren

Ein Vektor muss keine numerischen Werte enthalten, sondern kann auch ein sogenannter „Character-Vector“ sein. Dieser enthält Buchstaben (manchmal als „String“ bezeichnet) und wird eindeutig mit geraden Anführungszeichen markiert: " "

– und zwar in allen (!) Fenstern, also in der Source, Environment und in der Console.

Erzeugen eines Character-Vektors:

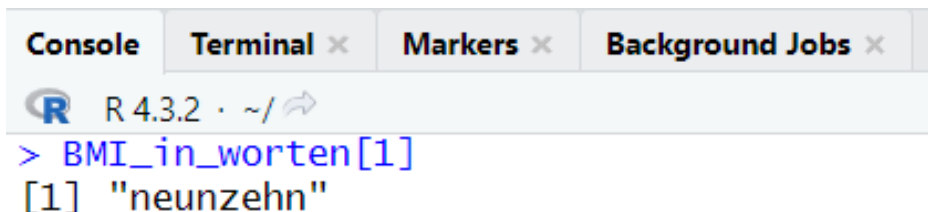
- Hier für das Beispiel unseres BMI-Vektors:

```
46 # Erzeugen eines Character-Vektoren:
47 BMI_in_worten <- c("neunzehn", "zweiundzwanzig", "zwanzig", "vierundzwanzig", "einunddreissig", "neunzehn")
48 print(BMI_in_worten)
```

```
> # Erzeugen eines Character-Vektoren:
> BMI_in_worten <- c("neunzehn", "zweiundzwanzig", "zwanzig", "vierundzwanzig", "einunddreissig", "neunzehn")
> print(BMI_in_worten)
[1] "neunzehn"      "zweiundzwanzig" "zwanzig"          "vierundzwanzig" "einunddreissig" "neunzehn"
```

- Auch hier lassen sich einzelne Elemente auswählen:

```
49 BMI_in_worten[1]
```



```
Console Terminal x Markers x Background Jobs x
R 4.3.2 · ~/
> BMI_in_worten[1]
[1] "neunzehn"
```

Wichtige Take-Aways

- **Mathematische Operatoren:** verwenden Sie $+$, $-$, $*$, $/$, $^$
- **Anwendung:** wenden Sie mathematische Operatoren auf Zahlen ($7 + 5$) Variablen (`zwoelf <- sieben + fuenf`) und Variablen an, die Vektoren enthalten (`BMI <- gewicht / groesse^2`)
- **Erstellung von Vektoren:** mittels combine-Funktion `c()`
- **Auswählen von Elementen aus einem Vektor:** mittels eckiger Klammern `[]`



4. ZWEITE PRAKTISCHE SCHRITTE IN R: R ZUM ERSTELLEN VON DATENSÄTZEN VERWENDEN

Vektoren zu einem Datensatz zusammenfügen

Mittels `cbind()` lassen sich Vektoren zu einem Datensatz zusammenfügen. Datensätze nennen wir in R **“Dataframe”**.

Zusammenfügen von Vektoren zu einem Dataframe mittels `cbind()`:

- `cbind()` mit allen zu kombinierenden Vektoren füllen:

Eselsbrücke: `cbind()` steht für column bind

Input im Source-Fenster:

```
51 # Vektoren zu einem Datensatz zusammenfügen:
52 data <- cbind(gewicht, groesse, BMI, BMI_in_worten)
53 print(data)
```

Output in der Console:

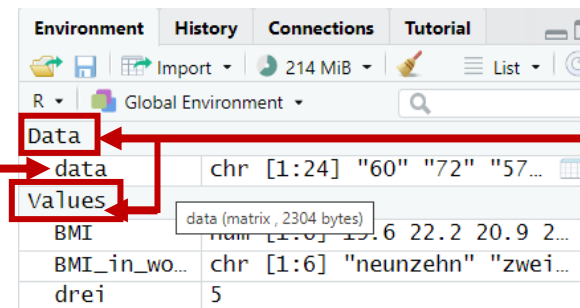
```
> data <- cbind(gewicht, groesse, BMI, BMI_in_worten)
> print(data)
      gewicht groesse BMI BMI_in_worten
[1,] "60"      "1.75" "19.5918367346939" "neunzehn"
[2,] "72"      "1.8"  "22.2222222222222" "zweiundzwanzig"
[3,] "57"      "1.65" "20.9366391184573" "zwanzig"
[4,] "90"      "1.9"  "24.9307479224377" "vierundzwanzig"
[5,] "95"      "1.74" "31.3779891663364" "einunddreissig"
[6,] "72"      "1.91" "19.7363010882377" "neunzehn"
```

- Sie können Datensätze auch mittels Doppelklick in der Environment betrachten:

	gewicht	groesse	BMI	BMI_in_worten
1	60	1.75	19.5918367346939	neunzehn
2	72	1.8	22.2222222222222	zweiundzwanzig
3	57	1.65	20.9366391184573	zwanzig
4	90	1.9	24.9307479224377	vierundzwanzig
5	95	1.74	31.3779891663364	einunddreissig
6	72	1.91	19.7363010882377	neunzehn

Geöffneter Datensatz

Wir können Datensätze mit einem Doppelklick auf ihren Namen öffnen.



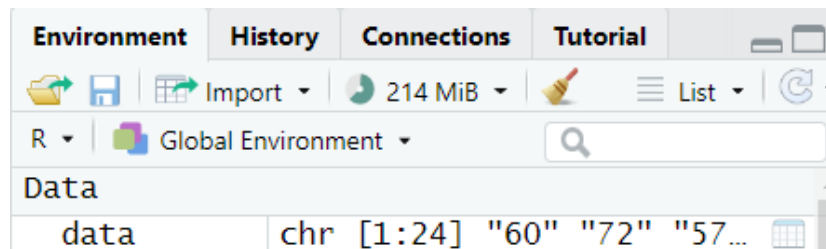
In der Environment, das Lagerhaus von R, existiert nun ein neuer Raum: „Data“, in dem Dataframes abgelegt werden. Alle Vektoren und numerischen Werte werden dagegen weiterhin unter „Values“ abgelegt.

Exkurs: Besonders „schöne“ Datensätze mittels `as_tibble()`

Mittels `as_tibble()` können Datensätze mit Zusatzinformationen (z.B. Zahl der enthaltenen Befragten) und schöner formatiert angezeigt werden. Diese Datensätze nennen wir in R **“Tibbles”**.

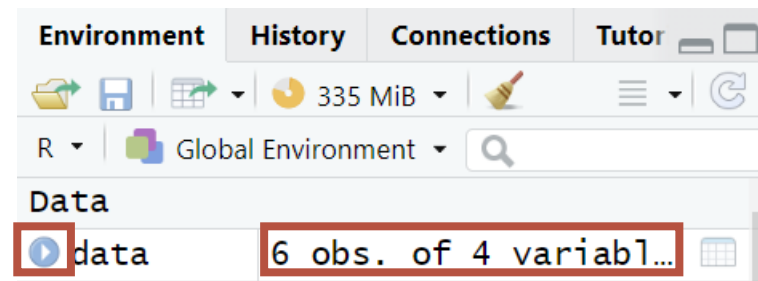
Dataframe - nur `cbind()`:

```
> data <- cbind(gewicht, groesse, BMI, BMI_in_worten)
> print(data)
      gewicht groesse BMI BMI_in_worten
[1,] "60"      "1.75"  "19.5918367346939" "neunzehn"
[2,] "72"      "1.8"    "22.2222222222222" "zweiundzwanzig"
[3,] "57"      "1.65"   "20.9366391184573" "zwanzig"
[4,] "90"      "1.9"    "24.9307479224377" "vierundzwanzig"
[5,] "95"      "1.74"   "31.3779891663364" "einunddreissig"
[6,] "72"      "1.91"   "19.7363010882377" "neunzehn"
```



Tibble - `cbind()` + `as_tibble()`:

```
> data <- cbind(gewicht, groesse, BMI, BMI_in_worten)
> data <- as_tibble(data)
> print(data)
# A tibble: 6 x 4
  gewicht groesse BMI BMI_in_worten
  <chr>    <chr> <chr> <chr>
1 60      1.75  19.5918367346939 neunzehn
2 72      1.8   22.2222222222222 zweiundzwanzig
3 57      1.65  20.9366391184573 zwanzig
4 90      1.9   24.9307479224377 vierundzwanzig
5 95      1.74  31.3779891663364 einunddreissig
6 72      1.91  19.7363010882377 neunzehn
```



Übung 1a

Legen Sie nun selbst einen neuen Dataframe an.

Folgende Daten von 3 Befragten liegen in einem Fragebogen vor:

gruppe	alter
Experimentalgruppe	55
Kontrollgruppe	21
Experimentalgruppe	34



Wie kann die Umsetzung in einem Dataframe aussehen?

Legen Sie die Daten in einem fiktiven Dataframe in R mittels `c()` und `cbind()` an.

Lösung für Übung 1a

Dateneingabe

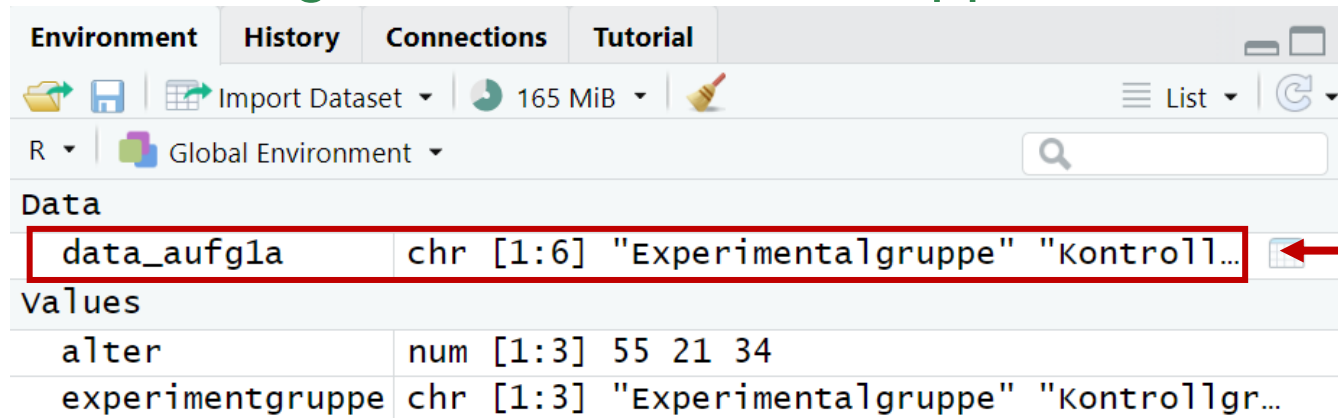
```
55 # Beispiellösung für Aufgabe 1a:  
56 gruppe <- c("Experimentalgruppe", "Kontrollgruppe", "Kontrollgruppe")  
57 alter <- c(55, 21, 34)  
58 data_aufg1a <- cbind(gruppe, alter)  
59 print(data_aufg1a)
```

Output in der Console:

```
> experimentgruppe <- c("Experimentalgruppe", "Kontrollgruppe", "Kontrollgruppe")  
> alter <- c(55, 21, 34)  
> data_aufg1a <- cbind(experimentgruppe, alter)  
> print(data_aufg1a)  
      experimentgruppe      alter  
[1,] "Experimentalgruppe" "55"  
[2,] "Kontrollgruppe"     "21"  
[3,] "Kontrollgruppe"     "34"
```

Lösung für Übung 1a

Betrachtung der Daten mittels Doppelklick in der Environment



Environment | History | Connections | Tutorial

Import Dataset | 165 MiB | List

R | Global Environment

Data

data_aufg1a	chr [1:6]	"Experimentalgruppe" "Kontrollgr..."
-------------	-----------	--------------------------------------

Values

alter	num [1:3]	55 21 34
experimentgruppe	chr [1:3]	"Experimentalgruppe" "Kontrollgr..."

Datensatz mit einem Doppelklick öffnen.

Ergebnis:

	gruppe	alter
1	Experimentalgruppe	55
2	Kontrollgruppe	21
3	Kontrollgruppe	34

Hier noch der gesamte Datensatz für Übung 1a

- Das Experiment war etwas umfangreicher.
- Damit Sie nicht die gesamten Daten abtippen müssen, finden Sie in Ihrem R-Skript die übrigen Fälle und Variablen.

Hinweis: Die Variablen *mng_scholz* und *mng_soeder* enthalten die Meinung der Befragten zu diesen beiden Politikern auf einer Skala von 1 (= sehr schlecht) bis 5 (= sehr gut).

Der gesamte Datensatz:

	gruppe	alter	geschlecht	mng_scholz	mng_soeder
1	Experimentalgruppe	55	maennlich	1	2
2	Kontrollgruppe	21	maennlich	3	1
3	Kontrollgruppe	34	maennlich	4	5
4	Experimentalgruppe	64	weiblich	3	4
5	Experimentalgruppe	19	weiblich	5	4
6	Kontrollgruppe	47	weiblich	2	2
7	Kontrollgruppe	87	maennlich	1	2
8	Experimentalgruppe	25	maennlich	3	1
9	Experimentalgruppe	18	maennlich	4	5
10	Kontrollgruppe	63	weiblich	3	4
11	Kontrollgruppe	29	weiblich	5	4
12	Experimentalgruppe	45	weiblich	2	2

5. DRITTE PRAKTISCHE SCHRITTE IN R: FÄLLE UND VARIABLEN IN DATAFRAMES AUSWÄHLEN

Fälle und Variablen in Datensätzen auswählen

Mittels eckiger Klammern [] lassen sich nicht nur Elemente in Vektoren auswählen, sondern auch Fälle (= Zeilen) und Variablen (= Spalten) in einem Dataframe.

Die Zahl vor dem Komma zeigt die Zeile an, die ausgewählt wird, die Zahl nach dem Komma die Spalte.

Wählen Sie den zweiten Fall (Zeile 2) und die erste Spalte aus (Gewichtsvariable):

- Wählen Sie das Gewicht der/s zweiten Befragten aus (er/sie wiegt 72kg):

Input im Source-Fenster:

```
81 # Fälle und Variablen aus einem Datensatz auswählen:
82 data[2, 1] # wählt den zweiten Fall in der ersten Spalte aus
```

```
Console Terminal x
R R 4.3.2 · ~/ ↵
> data[2, 1]
gewicht
"72"
```

- Wählen Sie das Gewicht aller Befragten aus, indem die Zahl vor dem Komma leergelassen wird:

Input im Source-Fenster:

```
84 # Alle Fälle für eine Variable in einem Datensatz
85 data[ , 1]
```

```
Console Terminal x Markers x Background Jobs x
R R 4.3.2 · ~/ ↵
> data[ , 1]
[1] "60" "72" "57" "90" "95" "72"
```

- Zeigen Sie alle Werte der/s zweiten Befragten an:

Input im Source-Fenster:

```
87 # Alle Variablen
88 data[2, ]
```

```
Console Terminal x Markers x Background Jobs x
R R 4.3.2 · ~/ ↵
> data[2, ]
gewicht          groesse          BMI          BMI_in_worter
"72"            "1.8" "22.22222222222222" "zweiundzwanzig"
```

Übung 1b

Was denkt der vierte Befragte über Olaf Scholz?

Lassen Sie sich alle eingetragenen Werte für den fünften Befragten ausgeben.

Zum Knobeln & Googeln: Wie können Sie sich gleichzeitig alle Angaben für den vierten und den fünften Befragten anzeigen lassen?



Lösen Sie die Aufgabe mittels eckiger Klammern [].

Lösung für Übung 1b

Was denkt der vierte Befragte über Olaf Scholz?

```
92 data_aufg1a[4, 4] # Meinung zu Scholz des vierten Befragten → > data_aufg1a[4, 4]
                                     Output mng_scholz
                                                "3"
```

Lassen Sie sich alle eingetragenen Werte für den fünften Befragten ausgeben.

```
93 data_aufg1a[5, ]
      ↓ Output
> data_aufg1a[5, ]
  experimentgruppe      alter      geschlecht      mng_scholz      mng_soeder
"Experimentalgruppe"    "19"      "weiblich"      "5"              "4"
```

Zum Knobeln & Googeln: Wie können Sie sich gleichzeitig alle Angaben für den vierten und den fünften Befragten anzeigen lassen?

```
94 data_aufg1a[4:5, ] # alle werte des vierten & fünften Befragten
      ↓ Output
> data_aufg1a[4:5, ]
  experimentgruppe      alter      geschlecht      mng_scholz      mng_soeder
[1,] "Experimentalgruppe" "64"      "weiblich"      "3"              "4"
[2,] "Experimentalgruppe" "19"      "weiblich"      "5"              "4"
```

Fälle & Variablen in Datensätzen auswählen mittels `$`

Spalten in einem Dataframe können auch direkt mit ihrem Namen mittels `$` ausgewählt werden.

Zudem lassen sich `$` und `[]` zur Auswahl konkreter Werte kombinieren.

Wählen Sie den zweiten Fall (Zeile 2) und die erste Spalte aus (GewichtsvARIABLE):

- Wählen Sie das Gewicht des zweiten Befragten aus (er/sie wiegt 72kg):

Input im Source-Fenster:

```
99 data$gewicht[2]
```

Output in der Console:

```
> data$gewicht[2]
[1] "72"
```

- Wählen Sie das Gewicht aller Befragten aus:

Input im Source-Fenster:

```
102 data$gewicht
```

Output in der Console:

```
> data$gewicht
[1] "60" "72" "57" "90" "95" "72"
```

Achtung: Der `$`-Operator funktioniert nur, wenn die Daten zuvor explizit in einen Dataframe / eine Tibble umgewandelt wurden (die Umwandlung zum Tibble lernen Sie aber erst nächste Woche richtig ausführlich):

```
data <- tidyr::as_tibble(data)
```


Wichtige Take-Aways

- **Dataframes:** so werden Datensätze (= Tabellen) in R genannt
- **Fälle:** bezeichnen die Reihen / Zeilen eines Datensatzes
- **Variablen:** bezeichnen die Spalten eines Datensatzes
- **Erstellen eines Dataframes:** durch zusammengefügte Vektoren mittels `cbind()`
- **Tibbles:** “schöne” Datensätze mit Zusatzinformationen mittels `as_tibble()`
- **Auswählen von Werten aus einem Dataframe:** mittels eckiger Klammern `[]` oder einer Kombination aus `$ + []`



6. KURZE WIEDERHOLUNG: SKALENNIVEAUS

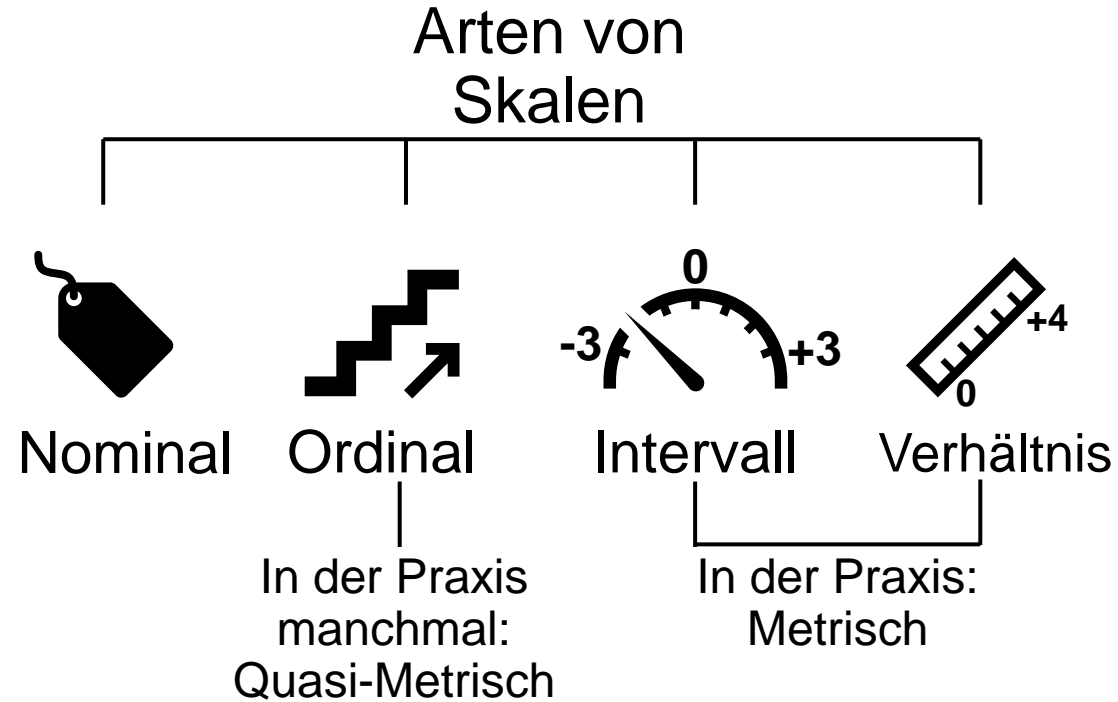
Skalenniveaus



Messniveau	Annahme	Beispiel
Nominal	Verschiedenartigkeit $A \neq B$	Geschlecht Konfession
Ordinal	sachlogische Rangfolge $A < B < C$	Schulnoten Tabelle der Bundesliga
Metrisch <ul style="list-style-type: none"> • Intervallskala • Verhältnis- bzw. Ratioskala 	Abstände messbar $B - A = D - C$ $A = x \text{ mal } B$	Temperatur in Celsius Zustimmung auf Skala Alter Einkommen

Weiterführende Hilfe als Lehrvideo: https://cast.itunes.uni-muenchen.de/clips/5SYoZMk75V/vod/high_quality.mp4

Skalenniveaus



Skalenniveaus – Beispiel „Fernsehkonsument“

→ i.d.R. bestimmt die Messung das Skalenniveau und nicht das Merkmal

Beispiel: Fernsehkonsum

	Nominales Skalenniveau	Ordinales Skalenniveau	Metrisches Skalenniveau (Verhältnisskala)
Frage:	Schauen Sie fern?	Wie oft schauen Sie fern? Bitte kreuzen Sie an, was für Sie zutrifft.	Wie viele Stunden pro Tag schauen Sie fern?
Antwortkategorien:	1 = ja 2 = nein	1 = nie 2 = gelegentlich 3 = häufig 4 = sehr häufig	offen



ÜBUNG ZUR WIEDERHOLUNG DER SKALENNIVEAUS: AUFGABE 2

Übung 2

- a) Stellen Sie sich einen Fragebogen vor, der Oktoberfest-Besucher*innen vorgelegt wird.

Bestimmen Sie für folgende im Oktoberfest-Fragebogen abgefragte Merkmale das zugehörige **Skalenniveau** (diskret oder stetig).

➤ Anzahl Besuche Oktoberfest	➤ Fernsehnutzungsdauer in Minuten
➤ Tracht (ja/nein)	➤ Interesse an Sportberichterstattung (5er-Skala)
➤ Anzahl Maß Bier	➤ Häufigkeit Kopfschmerzen

- b) Warum ist es bedeutsam, das Skalenniveau der Merkmale richtig zu identifizieren?

Lösung für Übung 2

- a) Stellen Sie sich einen Fragebogen vor, der Oktoberfest-Besucher*innen vorgelegt wird.

Bestimmen Sie für folgende im Oktoberfest-Fragebogen abgefragte Merkmale das zugehörige **Skalenniveau** (diskret oder stetig).

➤ Anzahl Besuche Oktoberfest Metrisch (Ratio, diskret)	➤ Fernsehnutzungsdauer in Minuten Metrisch (Ratio, stetig)
➤ Tracht (ja/nein) Nominal	➤ Interesse an Sportberichterstattung (5er-Skala) Ordinal/quasi-metrisch
➤ Anzahl Maß Bier Metrisch (Ratio, diskret)	➤ Häufigkeit Kopfschmerzen Ordinal/quasi-metrisch

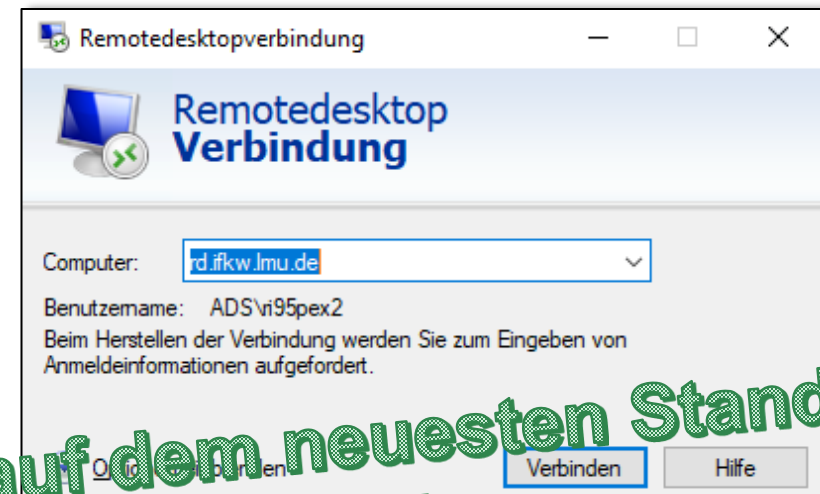
- b) Verschiedene Skalenniveaus erfordern unterschiedliche statistische Auswertungsmethoden. Das wird in den Sitzungen zu Auswertungsmethoden später noch relevant.

7. Wo BEKOMME ICH ZUSÄTZLICHE HILFE?

R UND RSTUDIO VON ZUHAUSE NUTZEN

Wie kann ich R zuhause nutzen? Alternative 1.

- Auf dem **IfKW Remote-Desktop** einloggen
(ausführliche Anleitung: <https://www.it.ifkw.uni-muenchen.de/anleitungen/remotedesktop/index.html>)
 - VPN-Client (eduVPN) installieren und Verbindung aufbauen
(hier wird die LRZ-Kennung benötigt, z.B. di12abc)
 - Programm für Remote-Zugriff installieren (bei Windows bereits vorinstalliert) und Remotedesktopverbindung aufbauen
 - Serveradresse eingeben: **rd.ifkw.lmu.de**
 - Wie im CIP anmelden
 - Zertifikat ggf. akzeptieren
- RStudio öffnen



>> Vorteil: Alles vorinstalliert, alles auf dem neuesten Stand!
Nachteil: Verbindung ist z. T. instabil.

Wie kann ich R zuhause nutzen? Alternative 2.

- Da **R** und **RStudio** kostenfrei sind, können Sie beides auch auf Ihrem eigenen Endgerät installieren und nutzen.
- Befolgen Sie dafür bitte die ausführliche Schritt-für-Schritt-Anleitung aus unserem Kompendium: <https://datenanalyse.ifkw.lmu.de/installation-von-r-und-rstudio>
 - *Achtung:* Beachten Sie die abweichende Anleitung für Windows vs. Mac OS.

>> Unsere Empfehlung: Sie werden R so häufig in Ihrem Studium benötigen, dass es sich lohnt, frühzeitig Ihre eigene Version aufzusetzen und zu individualisieren!

Tipp: Lassen Sie sich im R-Statistik-Helpdesk mit der Installation helfen, falls Probleme auftreten.

7. Wo BEKOMME ICH ZUSÄTZLICHE HILFE?

ERGÄNZENDE HILFEN UND LITERATUR

Wo bekomme ich zusätzliche Hilfe am IfKW?

Datenanalyse am IfKW




Herzlich Willkommen!

Hier finden Sie zahlreiche Tipps und Anleitungen zum Umgang mit R. Die Verfahren orientieren sich daran, wie R hier am IfKW in der Datenanalyse gelehrt wird. Wir hoffen, dass Ihnen diese Seite helfen wird.

Klicken Sie einfach auf die untenstehenden R-Sticker, um mehr zum jeweiligen Thema zu erfahren.



Zusätzlich zu diesen Folien „Datenanalyse“ steht Ihnen das umfangreiche, IfKW-eigene R-Kompendium zur Verfügung. Hier können Sie (fast) alle Lerninhalte noch einmal ausführlich nachlesen: <https://datenanalyse.ifkw.lmu.de/>

- Das Anklicken einer honigwarbenartigen Kachel führt Sie zur zugehörigen Hilfeseite.
 - Grüne Kacheln repräsentieren die Hilfeseiten für statistische Verfahren.
 - Rosafarbene und schwarze Kacheln repräsentieren die Hilfeseiten für die Installation und den Umgang mit R / RStudio.
- Durch Anklicken der  können Sie unter “zeige alle Datenanalyse-Inhalte” die honigwarbenartigen Kacheln nach Datenanalyse-Sitzungen sortieren. Nützlich zur Wiederholung!
- Auf manchen Folien werden Sie oben rechts dieses Symbol finden: . Es zeigt Ihnen an, dass Sie diese Inhalte auch  im Kompendium nachlesen können (verlinkt).

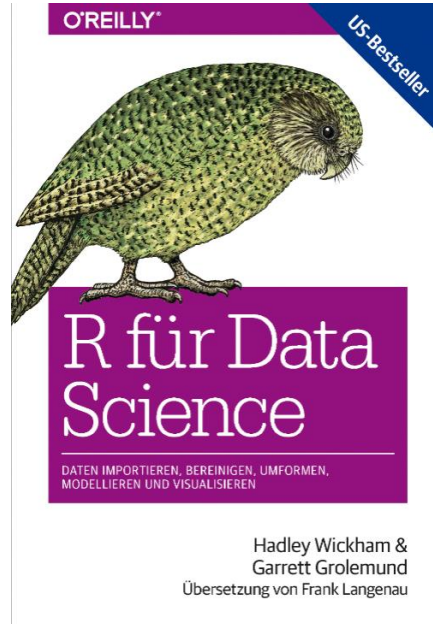
Wo bekomme ich zusätzliche Hilfe am IfKW?

- Bei Installationsproblemen, Fehlermeldungen oder Verständnisfragen steht der **R-Statistik-Helpdesk am IfKW** in der Oettingenstraße 67 (**Raum U110**) zur Verfügung. Sie können dort jederzeit unangemeldet in den Öffnungszeiten vorbeikommen und sich unterstützen lassen.
- Aktuelle Öffnungszeiten des Helpdesks finden sich via https://www.ifkw.uni-muenchen.de/organisation/statistik_helpdesk.
- Zusätzlich ist der R-Statistik-Helpdesk via E-Mail an statistik-helpdesk@ifkw.lmu.de erreichbar.
- **Achtung:** Bitte achten Sie bei Anfragen im R-Statistik-Helpdesk auf die Einhaltung der **Netiquette**. Kontaktieren Sie den Helpdesk bitte mit einem konkreten Problem / einer Frage und informieren Sie über alle Schritte, die Sie bisher zur Problemlösung selbst unternommen haben.

Ein bisschen moderne Hilfe (nicht für die Klausur)

- Mathematische Symbole und griechische Buchstaben entziffern: <http://shapecatcher.com>
- Handschriftliche Formeln erkennen und berechnen: MyScript Calculator (App für iOS und Android)
- Wissenschaftlicher Taschenrechner: <https://web2.0calc.com/>
- Methodenvisualisierungen: students.brown.edu/seeing-theory/

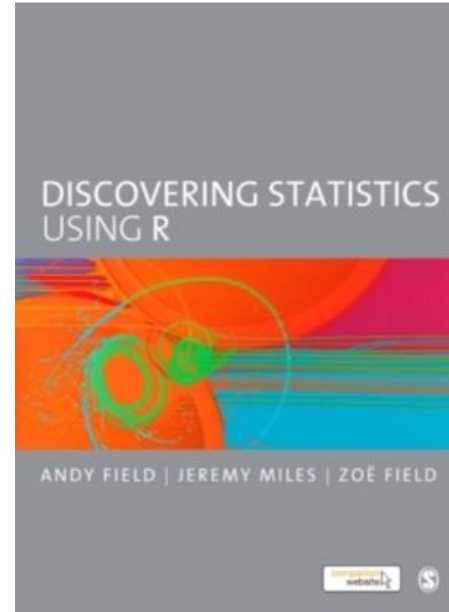
Ergänzende Literatur zum Nachschlagen



Besonderheit: Vom Macher des „tidyverse“. Open-Access im Netz unter <https://r4ds.hadley.nz/> (Englisch). Via UB ist auch der Volltext der deutschen Fassung digital erhältlich:



Stärke: Beste, praxisnahe und umfassendste Einführung in R, insbesondere in die tidyverse-Grammatik (die wir hier nutzen).



Besonderheit: Mit Witz und vielen Katzenbeispielen geschrieben. Gibt es in der Bibliothek.

Stärke: Sehr gute und leicht verständliche Erklärung der statistischen Verfahren – auch bei den komplexen Themen (z.B. Konfidenzintervalle).



Besonderheit: Von deutschen Kommunikationswissenschaftler*innen geschrieben. Open-Access im Netz unter <https://link.springer.com/book/10.1007/978-3-658-34285-2>

Stärke: Praxisnahe, deutschsprachige Einführung mit zahlreichen Best-Practice-Empfehlungen, die sich auf die KW beziehen. Auch mit tidyverse-Grammatik.