Computergestützte Datenanalyse: DATA-Übung mit R

Tag 2 – 18.07.2025

UNSER PLAN

- **Tag 1**
  - Einführung in R und RStudio
  - „Basics“:
  - Coding Konventionen
  - Objekte, Datenimport & Co
- **Tag 2**
  - Skalenniveau
  - Troubleshooting
  - Datenaufbereitung
  - Datenvisualisierung
  - Deskriptive Statistik
- **Tag 3**
  - Inferenzstatistik I
  - Bivariate Analyse
- **Tag 4**
  - Indexbildung
  - Inferenzstatistik II
  - Abschluss

Genereller Ablauf

- Vier Tage geblockt
- Mischung aus Input- und Übungssessions
- Anwesenheitsabfrage alle 90 Minuten

Heute

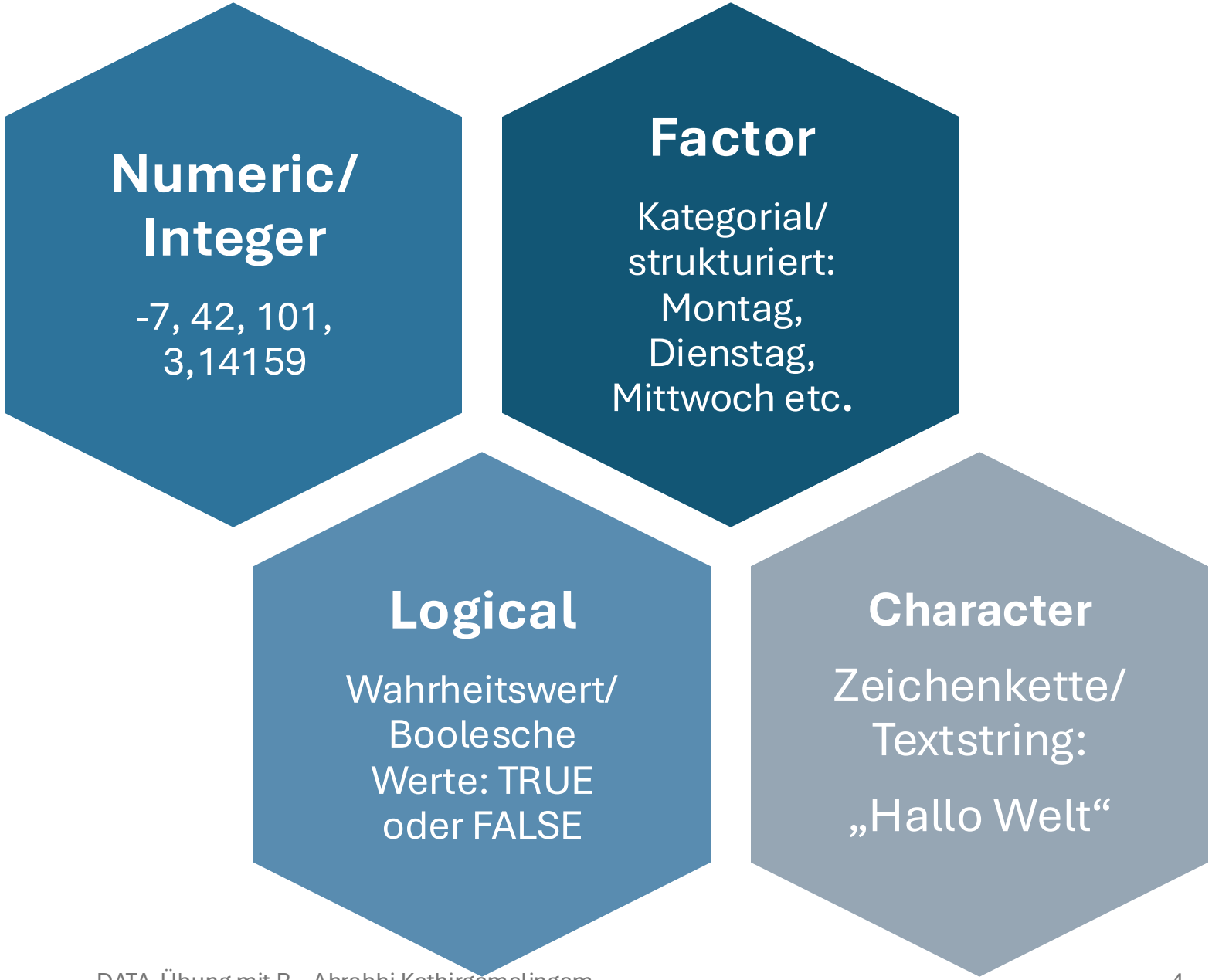
- Zwei 15 Minuten Pause
- Eine Mittagspause

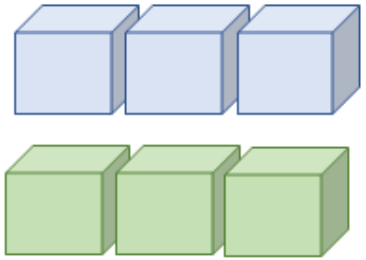
REFRESHER

- Was ist der Unterschied zwischen R und Rstudio?
- Was ist base?
- Was sind Konventionen für R? Und Good Practices?
- Was für Formate gibt es?
- Was für Klassen gibt es?

Datentypen/ Klassen

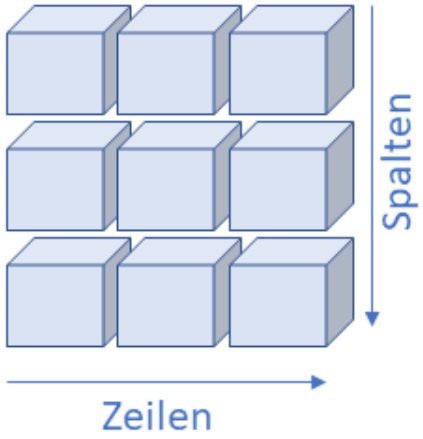
Zum Nachlesen





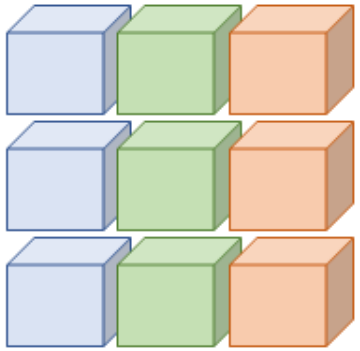
Vektor

Eine ganz einfache Liste von Werten einer Klasse!



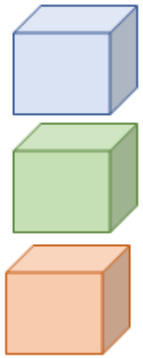
Matrix

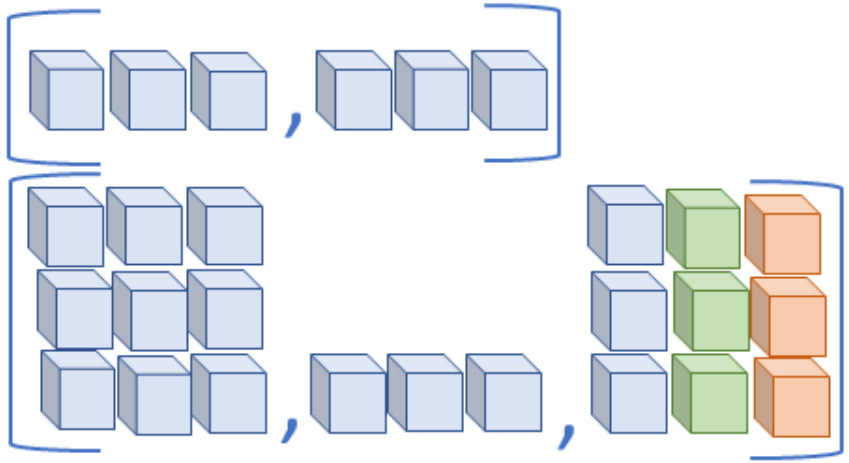
Eine Tabelle mit Werten einer Klasse!



Data Frame

Eine Tabelle, in der jede Spalte (Variable) eine andere Klasse sein kann!





Lists

Kann unterschiedliche Formate und Klassen beinhalten

Quelle: https://devopedia.org/r-data-structures

Welche Skalenniveaus kennen Sie?

Skalenniveaus

- Nominalskala: Unterschiede
- Ordinalskala: + Rangfolge
- Intervallskala: + gleiche Abstände
- Ratioskala: + Natürlicher Nullpunkt

Skalenniveaus

- Nominalskala
  - Ausprägungen eines Merkmals bedeutet nur, dass es einen Unterschied gibt
  - Keine Rangfolge
  - Zum Beispiel: Familienstand, Geschlecht, Lieblingsfarbe
  - Analysemöglichkeiten: Häufigkeiten, Kreuztabellen
  - Spezialfall von zwei Kategorien: Dichotomie

Skalenniveaus

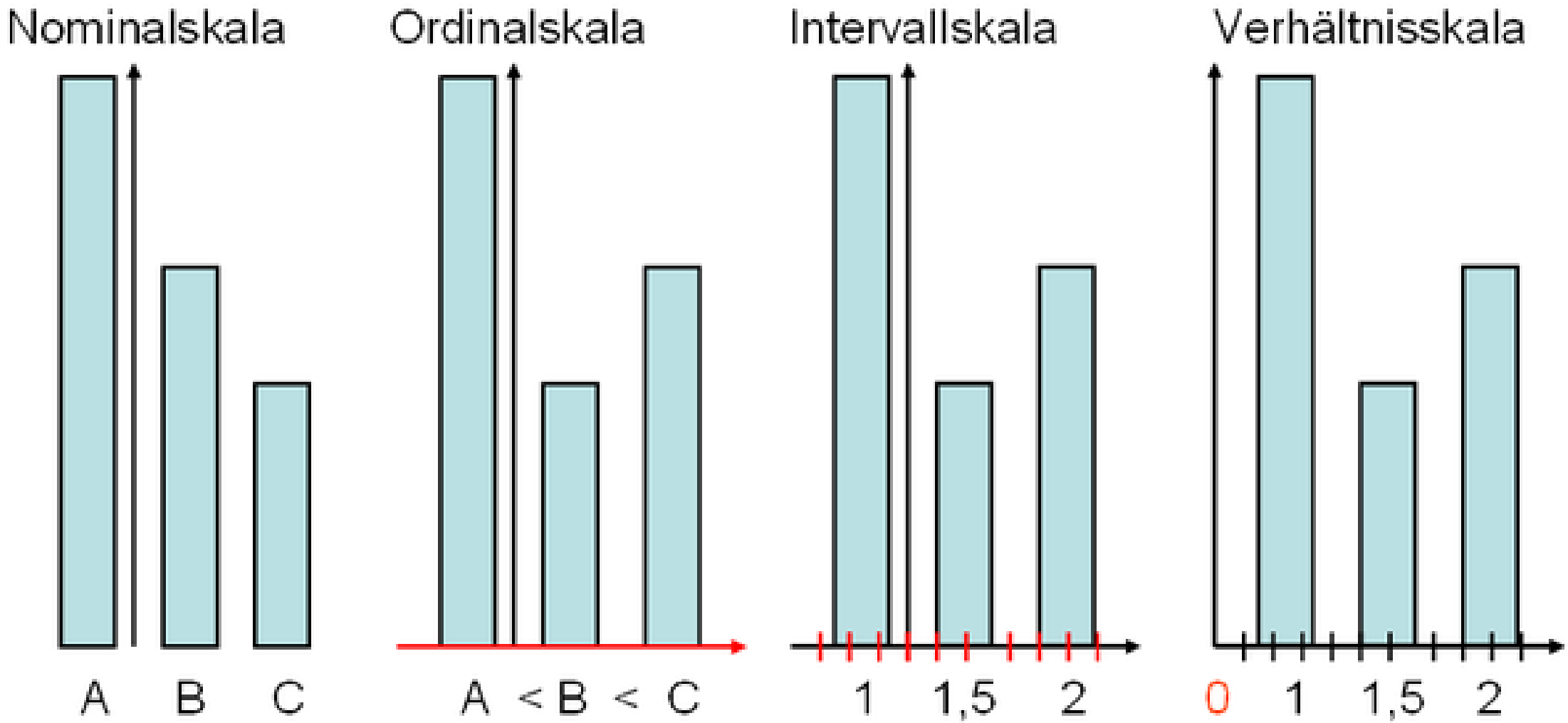
- Ordinalskala
  - Ausprägungen eines Merkmals zeigen Unterschied und die Rangfolge
  - d.h. Auskunft über ein Mehr oder Weniger des Ausmaßes einer Ausprägung
  - Zum Beispiel: Schulnoten, Zufriedenheit mit der Demokratie, Berufprestige
  - Analysemöglichkeiten: Häufigkeiten, Rangkorrelationskoeffizienten etc.

Skalenniveaus

- Intervallskala
  - Information über die Abstände zwischen den Ausprägungen eines Merkmals
  - Es gibt keinen „natürlichen“ Nullpunkt
  - Zum Beispiel: Temperatur in Grad Celsius, Geburtsjahr
  - Analysemöglichkeiten: Mittelwerte, Streuungsmaße, Korrelationskoeffizienten nach Pearson

Skalenniveaus

- Ratioskala
  - Information über das Verhältnis der Abstände zwischen den Ausprägungen eines Merkmals
  - Es gibt einen „natürlichen“ Nullpunkt
  - Zum Beispiel: Alter, Einkommen
  - Mittelwerte, Streuungsmaße, Regressionen, etc.



https://de.wikipedia.org/wiki/Skalenniveau

Sonderfall: Die Dummy-Variable

- Variable mit der (binären) Ausprägung 0 und 1
- **Beispiel 1:** Bei einer Wahlumfrage gibt eine kategoriale Variable an, welche Partei die Befragten wählen würde. Um den Anteil der SPD-Wähler*innen zu ermitteln, benutzt man eine Dummy-Variable mit den Ausprägungen:
  1 = SPD-Wählerin und 0 = keine SPD-Wählerin.
- **Beispiel 2:** Die Dummy-Variable bekommt den Wert 1, wenn die befragte Person jünger als 50 Jahre ist, und ansonsten den Wert 0.
- Dummykodierte Variablen können ebenfalls als erklärende Variablen in einer multiplen linearen Regression verwendet werden

Skalenniveaus

Aufgabe:

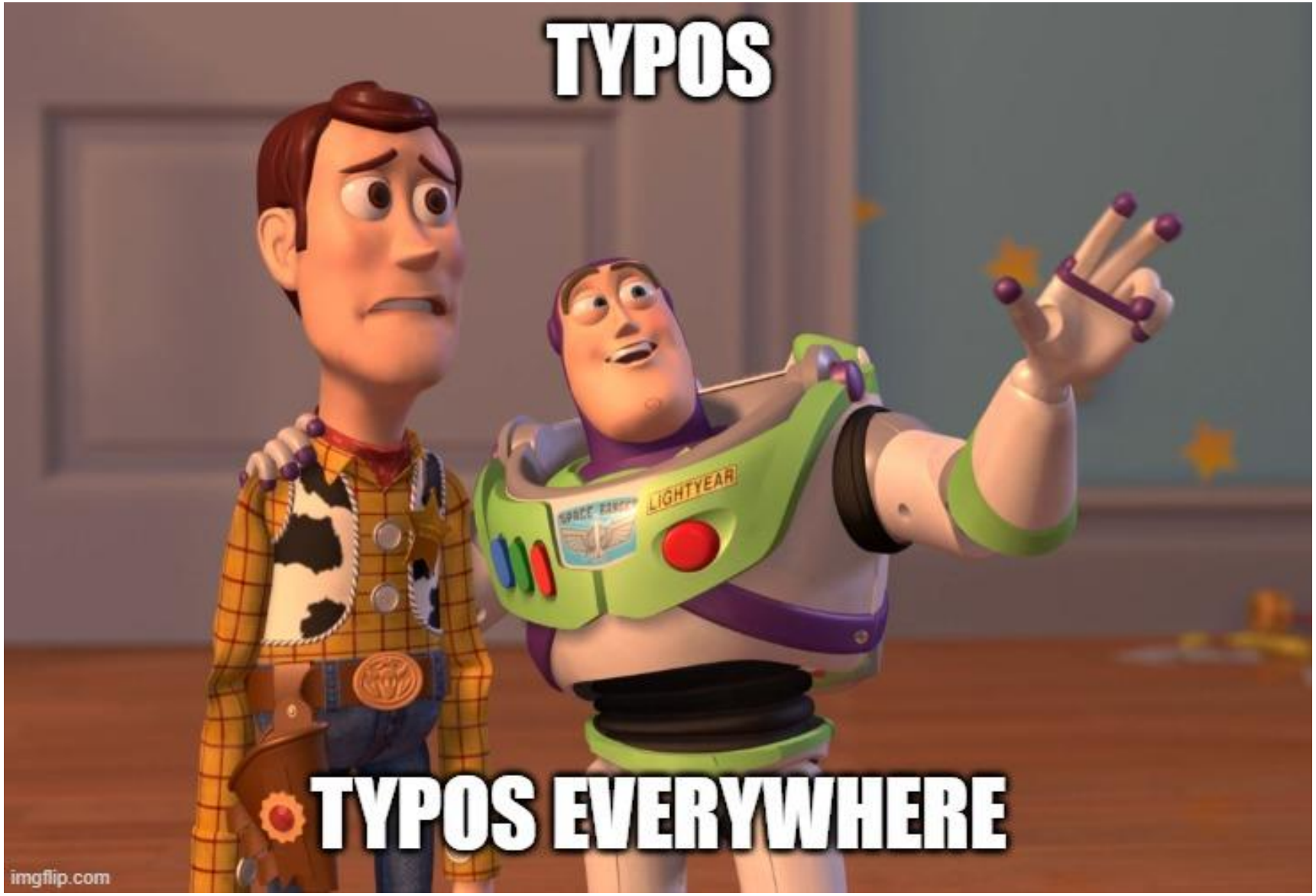
```
install.packages("rio")
library(rio)
install_formats()

setwd("Pfad zum Ordner")
allbus <– import("ALLBUS....sav")

variable.names(allbus)
allbus$german
allbus$eastwest
allbus$xr20
allbus$xt10
```

besser: attributes(allbus$german)

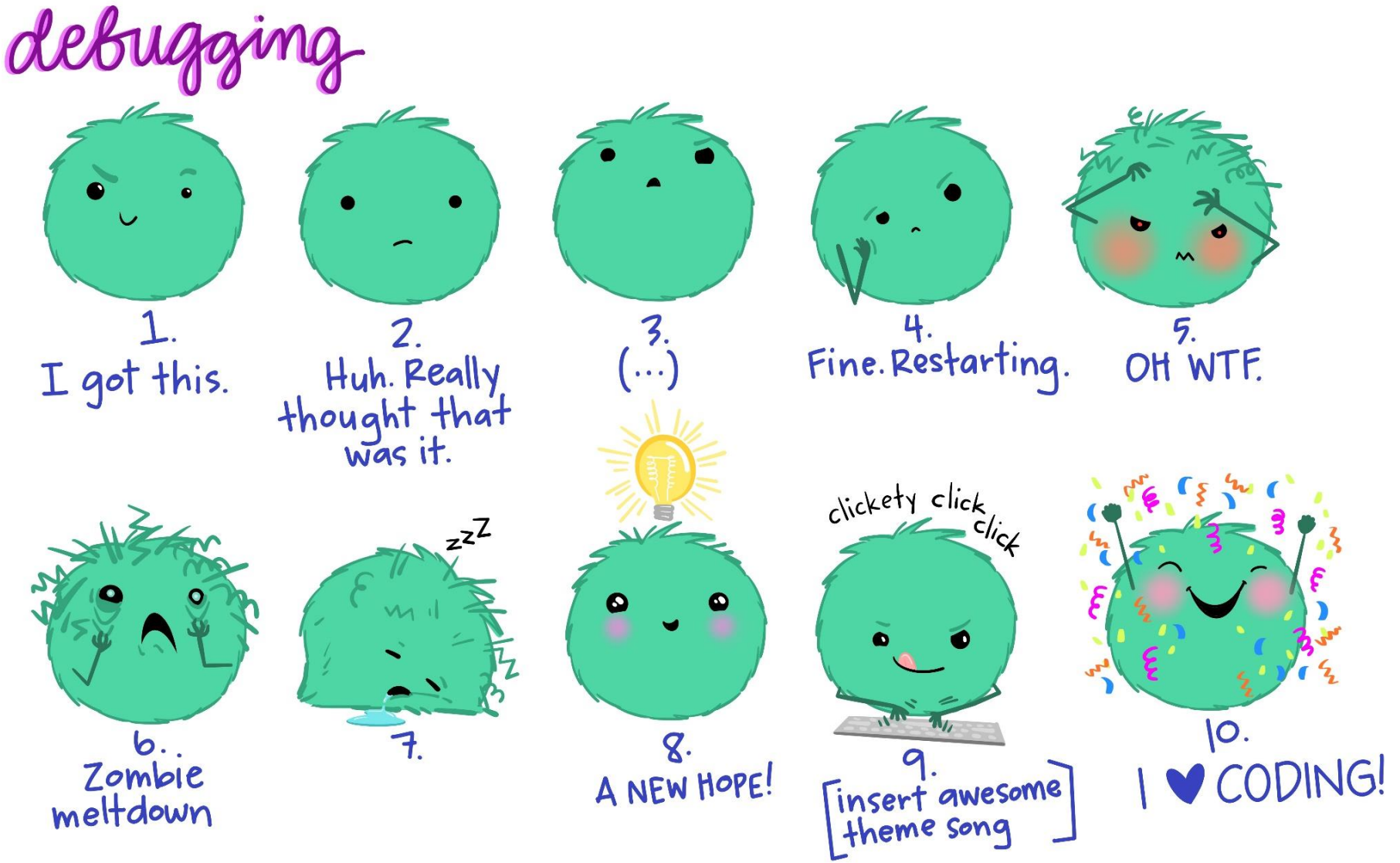
Troubleshooting


TYPOS
LIGHTYEAR
TYPOS EVERYWHERE
imgflip.com

Troubleshooting

- R gibt in der Console eine Fehlermeldung (error) aus, z.B.
  - Error: unexpected string constant in "a <- c("a", "b", "c" "d"“
  - Error: object 'b' not found
  - Error in a + b : non-numeric argument to binary operator

- Aber es gibt auch Fehler, die nicht in R getagged werden und eher inhaltlicher Natur sind!


debugging
1.
I got this.
2.
Huh. Really thought that was it.
3.
(...)
4.
Fine. Restarting.
5.
OH WTF.
6..
Zombie meltdown
zzZ
7.
8.
A NEW HOPE!
clickety click click
9.
[insert awesome theme song]
10.
I ♥ CODING!
@allison_horst

Troubleshooting

- Zu den häufigsten Fehlerquellen zählen
  - Fehlende Klammer „( )“ am Anfang oder am Ende eines Befehls
  - Fehlende Anführungszeichen " " am Anfang oder am Ende
  - Fehlendes Komma „ , “
  - Typos (R ist case-sensitive!)
  - Package nicht installiert oder aktiviert
  - Dateipfad oder Directory ist falsch
  - Variable hat die falsche Klasse
  - Das Problem sitzt i.d.R. vor dem Bildschirm
  - liegt das Problem auf der Seite des Systems, einfach neustarten

Troubleshooting

- Die Lösung liegt irgendwo zwischen Stufe 6 (Zombie meltdown) und Stufe 8 (new hope)

- Ressourcen und Strategien zur Lösungssuche
  - Fehlermeldung lesen (und versuchen zu verstehen)
  - Syntax überprüfen
  - Schrittweise vorgehen, Kommentieren!
  - Google + Stackoverflow + ChatGPT? Code + Error
  - Frisch starten!

Troubleshooting

- Weitere Ressourcen
  - Getting help with R
  - Help-Function „?“
  - Insbesondere für packages
  - R-Suchmaschine: https://rseek.org/

PAUSE

REFRESHER

- Was sind die vier Skalenniveaus?
- Was sind die häufigsten Fehler bei R?
- Was hilft, wenn wir Fehlermeldungen bekommen?

Datenaufbereitung

ALLBUS

- Allgemeine Bevölkerungsumfrage der Sozialwissenschaften (kurz: ALLBUS)
  - Seit 1980 alle zwei Jahre: Erhebung zur gesellschaftlichen Dauerbeobachtung von Einstellungen, Verhalten und sozialem Wandel in Deutschland, z.B. Mediennutzung, politische Einstellungen, soziales Kapital, Soziodemografie etc.
  - Aufgrund von Corona wurde die Befragung 2020 nicht durchgeführt
  - Weitere infos: https://www.gesis.org/allbus/inhalte-suche/studienprofile-1980-bis-2018/2018

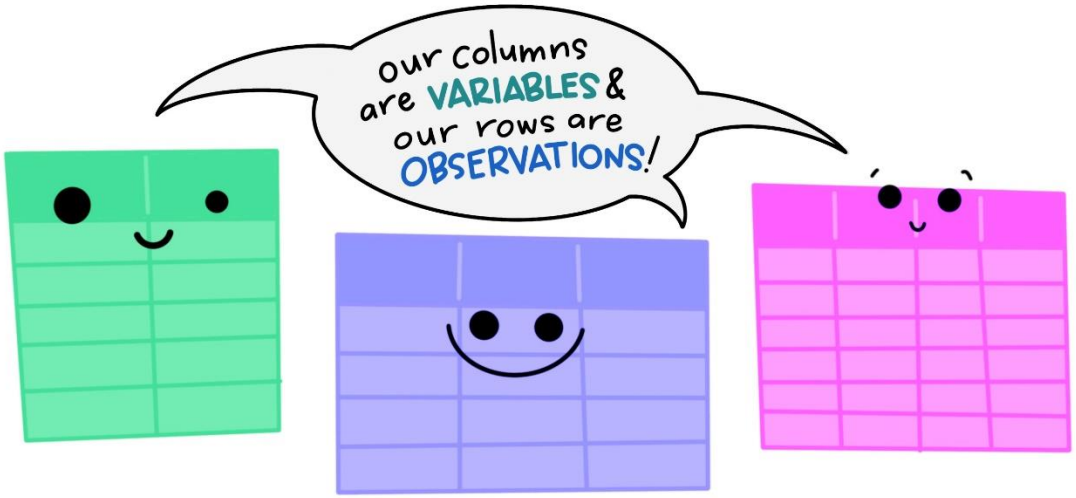
Datenaufbereitung | Data wrangling

- 80/20 Ration – 80 Prozent Datenaufbereitung und 20 Prozent Datenauswertung

- Mögliche Schritte der Datenaufbereitung
  - Subsetting und Filtern (z.B. nur Befragte aus Westdeutschland)
  - Rekodierung von Variablen (z.B. Alter und Geburtskohorten)
  - Neue Variablen erstellen (z.B. SES)
  - Missing Values
  - Neue Variablennamen vergeben (variablen_name_einer_bestimmten_variable = v1)

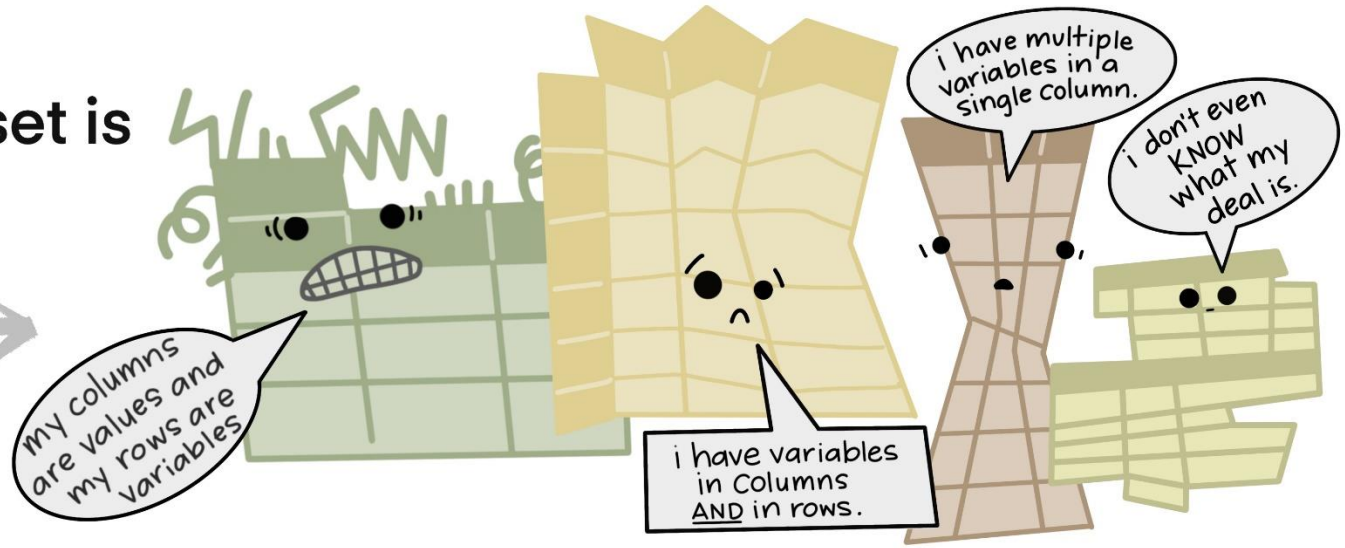
- Voraussetzung: Daten sind im „tidy“ Format
  - „wide“ vs. „long“ Format

The standard structure of tidy data means that "tidy datasets are all alike..."



"...but every messy dataset is messy in its own way."

–HADLEY WICKHAM



https://github.com/allisonhorst/stats-illustrations

Tidy data

- Tidy-Data heißt
  - Jede Variable ist in einer Spalte
  - Jede Beobachtung ist in einer Zeile
  - Jeder Wert ist in einer Zelle

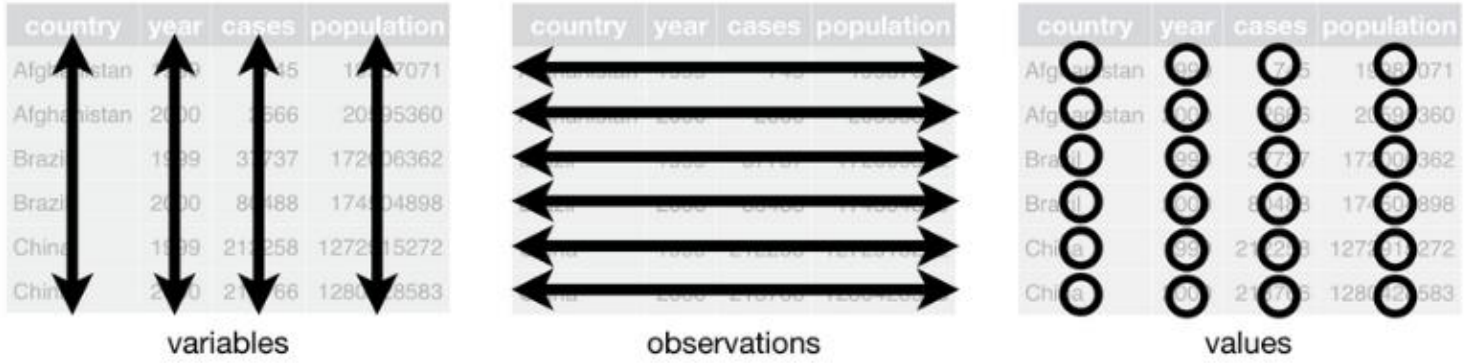


Figure 12.1: Following three rules makes a dataset tidy: variables are in columns, observations are in rows, and values are in cells. https://r4ds.had.co.nz/tidy-data.html

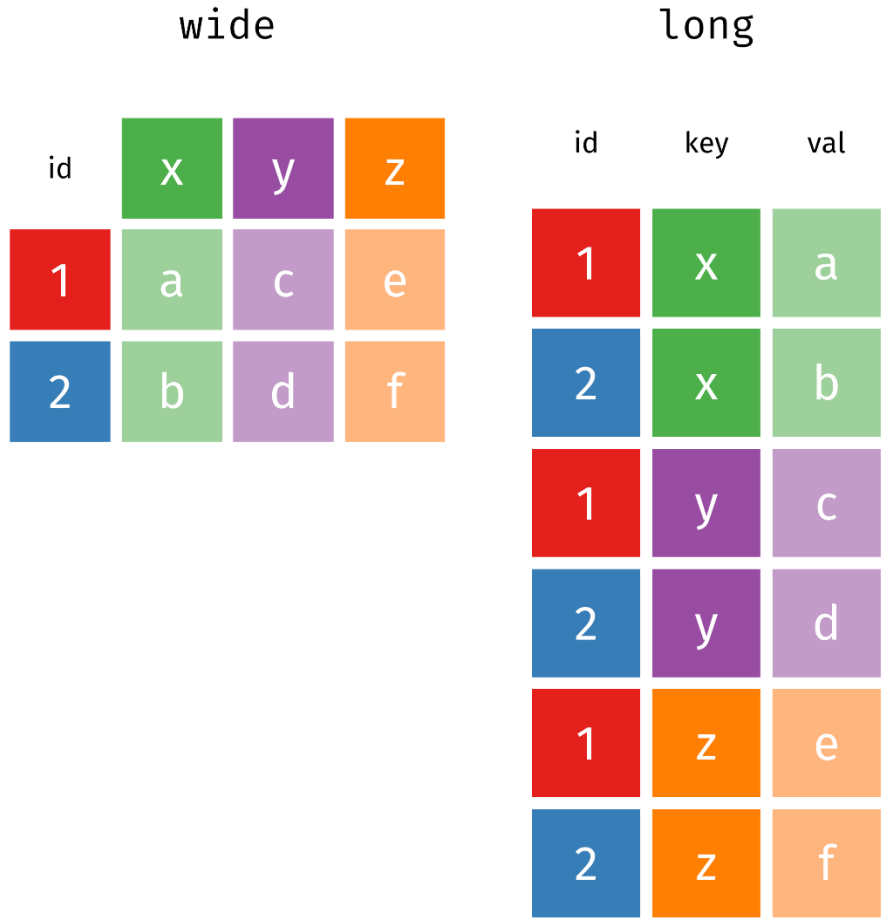
Der Aufbau von Tabellen (bzw. Data Frames) folgt dabei in der Regel einer logischen Struktur. Die Zeilen geben die Anazhl der *Beobachtungen* (oder *Fälle*) an, während die Spalten einzelne *Variablen* (oder *Merkmale*) repräsentieren.

Einheit	**Beobachtung (oder Fall)**	**Variable (oder Merkmal)**
Tabellen-Strukturmerkmal	Zeile	Spalte
Beispiele	• Respondent in einer Befragung • Zeitungsartikel in einer Inhaltsanalyse • Versuchsdurchlauf in einem Experiment	• Antwort auf eine Frage in einem Fragebogen • Quelle oder Titel eines Zeitungsartikels in einer Inhaltsanalyse • Messwert in einem Experiment

https://patrickzerrer.github.io/R_Hands_on_Book/Skript_2.1.html

„wide“ vs. „long“ Format

- Interessant, wenn mit Panel oder Aggregat-Daten gearbeitet wird
- Unterschiedliche Funktionen zum aggregieren und disaggregieren von Daten



https://github.com/gadenbuie/tidyexplain#tidy-data


tidyverse
www.rstudio.com


dplyr
forcats
ggplot2
haven
lubridate
purrr
stringr
readxl
tibble
tidyr
tidyverse

Datenaufbereitung: tidyverse

- „The tidyverse is a coherent system of packages for data manipulation, exploration and visualization that share a common design philosophy.” (Rickert, 2017)

- Installiert automatisch eine ganze Reihe weiterer Packages
  - ggplot2 (Visualisierung)
  - dplyr
  - magrittr

tidyverse

- Pipe-Operator als kleiner Zauberstab
  - df %>% Funktion
  - Shortcut strg + shift + m (windows) oder cmd + shift + m (Mac)

- Problem: lange Verkettung von Funktionen in BaseR
  sum(is.na(df$variable))

- Lösung: „dplyr“ aus dem Tidyverse-Package mit install.packages(„tidyverse“)
  Z.B.
  df$variable %>%
  is.na() %>%
  sum()

tidyverse

- Aufgabe:

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

sum(is.na(allbus$german))

allbus$german %>%
        is.na() %>%
                sum()
```

Base R vs. tidyverse

- Es gibt Pro und Contra Argumente zum tidyverse

- Im Kurs gilt: Hauptsache der Code läuft!
  - Beide „Stile“ können gemeinsam verwendet werden

Datenaufbereitung: Subsetting & Filtern

Subsetting & Filtern

- Um Bedingungen zur formulieren, nutzen wir Operatoren
  - Wir sind beispielsweise an der Analyse von Subgruppen interessiert
    - Nur Befragte unter 30 Jahren
    - Nur Befragte aus Ostdeutschland
    - Nur Befragte mit Internetanschluss
    - Nur Befragte mit Internetanschluss und über 65 Jahre

Subsetting & Filtern

- Bedingungen formulieren über logische Operatoren

Operator	Beschreibung
<	Kleiner als
<=	Kleiner als oder gleich
>	Größer als
>=	Größer als oder gleich
==	Genau gleich
!=	Nicht genau gleich
| (x | y)	Oder (x oder y)
& (x & y)	Und (x und Y)

Subsetting & Filtern mit tidyverse

filter(variable OPERATOR value)

Beispiel:
filter(age >= 18)

filter(variable OPERATOR value,
variable OPERATOR value,
...)

Operator	Beschreibung
<	Kleiner als
<=	Kleiner als oder gleich
>	Größer als
>=	Größer als oder gleich
==	Genau gleich
!=	Nicht genau gleich
| (x | y)	Oder (x oder y)
& (x & y)	Und (x und Y)

Subsetting & Filtern

- Datensatz nur mit Variablen german und ep01

```
allbus %>% german, ep01

allbus %>%
        select(german, ep01)

allbus_sub <- allbus %>%
        select(german, ep01)
```

Kommentieren mit # nicht vergessen ☺

Datei selbst wirst erst verändert, wenn wir den Datensatz abspeichern!

Subsetting & Filtern

- Datensatz nur mit Variablen german und ep01

```
allbus %>% german, ep01 (funktioniert nicht)

allbus %>%
        select(german, ep01)

allbus_sub <- allbus %>%
        select(german, ep01)
```

Kommentieren mit # nicht vergessen ☺

Datei selbst wirst erst verändert, wenn wir den Datensatz abspeichern!

Subsetting & Filtern

- Datensatz mit Personen mit dt. Staatsangehörigkeit (german == 1)

```
allbus %>% german
allbus %>% german == 1

allbus %>%
        filter(german == 1)

allbus_german <- allbus %>%
        filter(german == 1)
```

Kommentieren mit # nicht vergessen ☺

Datei selbst wirst erst verändert, wenn wir den Datensatz abspeichern!

Subsetting & Filtern

- Datensatz mit Personen mit dt. Staatsangehörigkeit

```
allbus %>% german (funktioniert nicht)
allbus %>% german == 1 (funktioniert nicht)

allbus %>%
        filter(german == 1) (nicht besonders sinnvoll)

allbus_german <- allbus %>%
        filter(german == 1)
```

Subsetting & Filtern

- Datensatz mit Personen mit dt. Staatsangehörigkeit
  + Interviewdauer über 100 Minuten

```
allbus_german <- allbus %>%
        filter(german == 1,
               xt10 >= 100)
```

Datenaufbereitung: Rekodierung

Rekodierung

- Subsetting = Datensatz verändern, Rekodierung = Variablen verändern!
- Beispiele:
  - Benennungen verändern
  - Skalenniveau transformieren (quasi-)metrisch -> Dummy-Variable
  - R-Klasse anpassen: z.B. einen Factor für ordinale Variable erstellen
  - Kategorien mit schwachen Besetzungen ggf. mit anderen Kategorien zusammenführen

Rekodierung (weiteres Beispiel)

- Generell gilt die Faustregel, dass hohe Werte auf der verwendeten Skala mit hohen Werten in der Ausprägung der Variable einhergehen sollten. Wenn das nicht gegeben ist, ist es empfehlenswert die Skala umzudrehen
  - Im ALLBUS ist eine hohe Zustimmung manchmal mit dem Wert 1 und eine niedrige Zustimmung mit dem Wert 10 kodiert. Diese Kodierung ist nicht intuitiv und bereitet uns bei späteren Analysen Probleme!

Beispiel: ALLBUS und Wahrnehmung wirtschaftlicher Lage

F001 **ep01**

Beginnen wir mit einigen Fragen zur wirtschaftlichen Lage.
Benutzen Sie für Ihre Antworten bitte die Liste.

⇨ *Liste 1 vorlegen und bis Frage 2 liegen lassen!*

Wie beurteilen Sie ganz allgemein die heutige wirtschaftliche Lage in Deutschland?

- ○ Sehr gut
- ○ Gut
- ○ Teils gut / teils schlecht
- ○ Schlecht
- ○ Sehr schlecht

- ○ Weiß nicht
- ○ KA

ep01	WIRTSCHAFTSLAGE IN DER BRD HEUTE

F001

Beginnen wir mit einigen Fragen zur wirtschaftlichen Lage. Benutzen Sie für Ihre Antworten bitte die Liste.

(Int.: Liste 1 vorlegen und bis Frage 2 liegen lassen!)

Wie beurteilen Sie ganz allgemein die heutige wirtschaftliche Lage in Deutschland?

-9 Keine Angabe

-8 Weiß nicht

1 Sehr gut

2 Gut

3 Teils gut / teils schlecht

4 Schlecht

5 Sehr schlecht

ZA5270, ep01: (N=3467) (gewichtet nach wghtpew)

Wert	Ausprägung	Missing	Anzahl	Prozent	Gült.Prozent
-9	KEINE ANGABE	M	1	0,0	
-8	WEISS NICHT	M	8	0,2	
1	SEHR GUT		622	17,9	17,9
2	GUT		1819	52,3	52,5
3	TEILS TEILS		870	25,0	25,1
4	SCHLECHT		133	3,8	3,8
5	SEHR SCHLECHT		24	0,7	0,7
	Summe		3477	100,0	100,0
	Gültige Fälle		3467		

Rekodieren mit tidyverse

Unterschiedliche Optionen

mutate() – Variable erstellen/ verändern

casewhen() - Rekodieren

```
df <- df %>%
  filter(variable_1 > Bedingung) %>%
  mutate(variable_2_r = case_when(    #neue Variable! (r für recoded)
            variable _2 == ALT ~ NEU,
            variable _2 == ALT ~ NEU,
            variable_2 == 99 ~ NA ))
```

Operator	Beschreibung
<	Kleiner als
<=	Kleiner als oder gleich
>	Größer als
>=	Größer als oder gleich
==	Genau gleich
!=	Nicht genau gleich
| (x | y)	Oder (x oder y)
& (x & y)	Und (x und Y)

Rekodierung

- Variable ep01 rekodieren

- Aktuelle Kodierung checken mit attributes(df$variable):

Sehr gut - 1
Gut – 2
Teils Teils – 3
Schlecht – 4
Sehr schlecht – 5
Usw.

- Häufigkeiten anschauen mit table(df$variable)

Rekodierung

- Variable ep01 rekodieren

- Rekodieren mit mutate() und case_when()
- Überschreiben oder nicht? Lieber: ep01_r

```
df <- df %>%
mutate(variable_2_r = case_when(
        variable _2 == ALT ~ NEU,
        variable _2 == ALT ~ NEU,
        variable_2 == 99 ~ NA ))
```

Rekodierung

- Variable ep01 rekodieren

- Lösung

```
allbus <- allbus %>%
  mutate(ep01_r = case_when(
    ep01 == 1 ~ 5,
    ep01 == 2 ~ 4,
    ep01 == 3 ~ 3,
    ep01 == 4 ~ 2,
    ep01 == 5 ~ 1))

table(allbus$ep01_r)
```

REFRESHER

- Was ist Subsetting, was ist Rekodieren?
- Was sind die Unterschiede?
- Worauf sollten wir beim Subsetten achten?
- Worauf sollten wir beim Rekodieren achten?
- Was ist tidyverse?

PAUSE

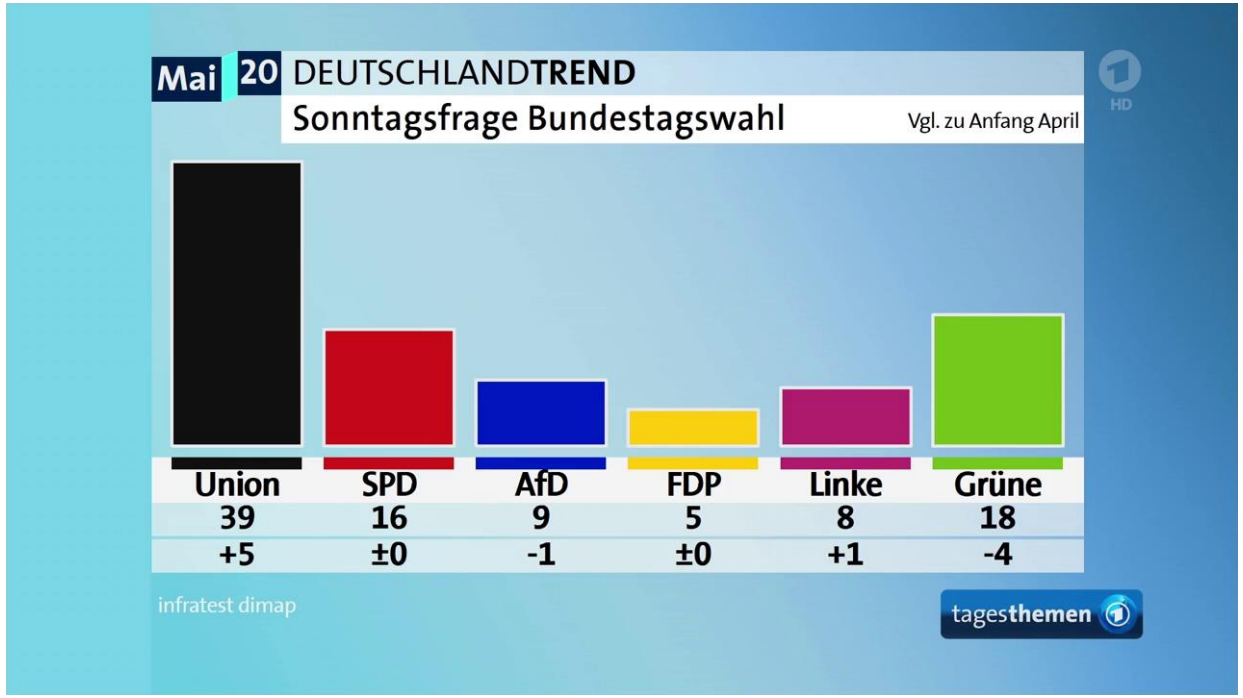
Datenvisualisierung!

Allison Horst

Wofür Daten visualisieren?

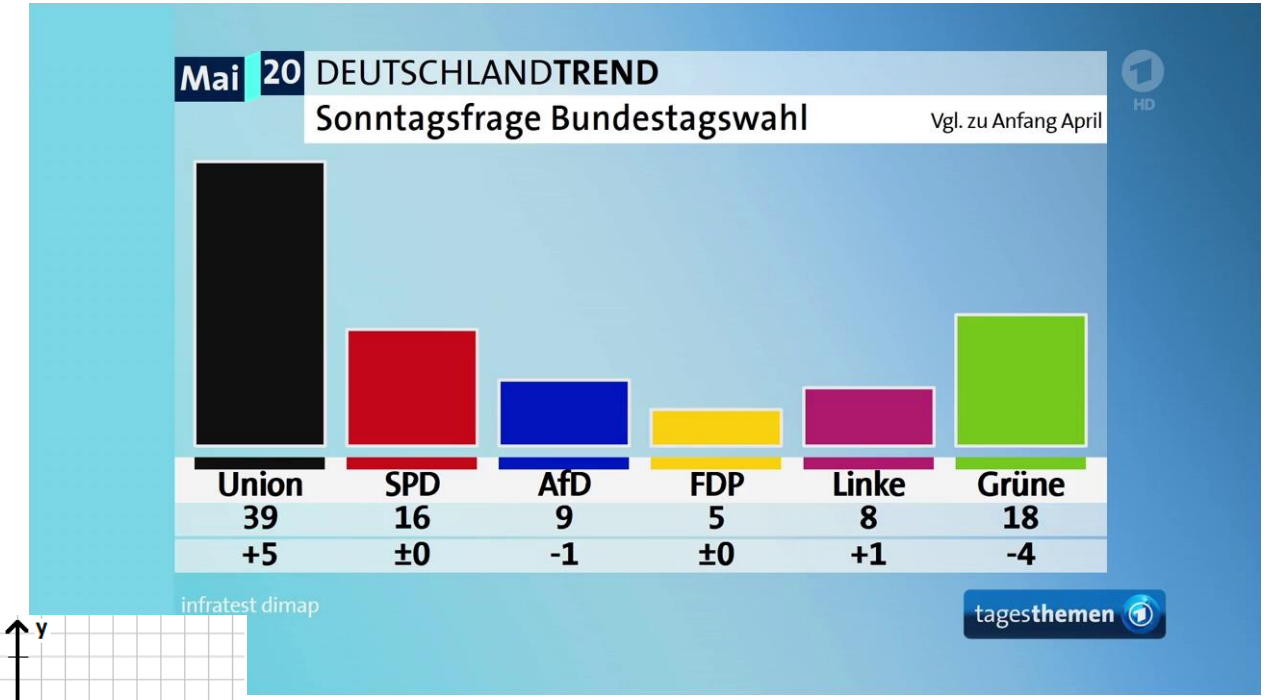
- Ergebnisse kommunizieren
  - Zusammenhänge verständlich machen!
  - Zusammenhänge zugänglich machen!
  - Visuelle Vergleiche ermöglichen!
- Davor aber:
  - Daten visualisieren zum Begreifen
  - Datenqualität überprüfen

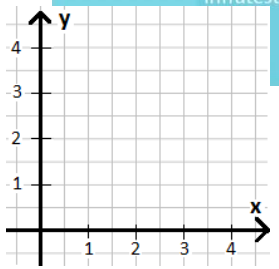
Encodings, Geometrics, Scales



- Encoding: Welche Daten werden in visuelle Elemente übertragen?
- Geometrics: Welche visuellen Elemente werden genutzt?
- Scales: Auf welchen Skalen werden die Daten abgebildet?

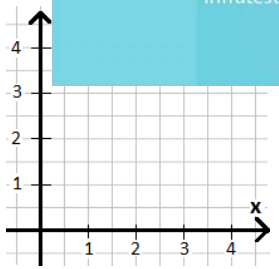
Encodings, Geometrics, Scales





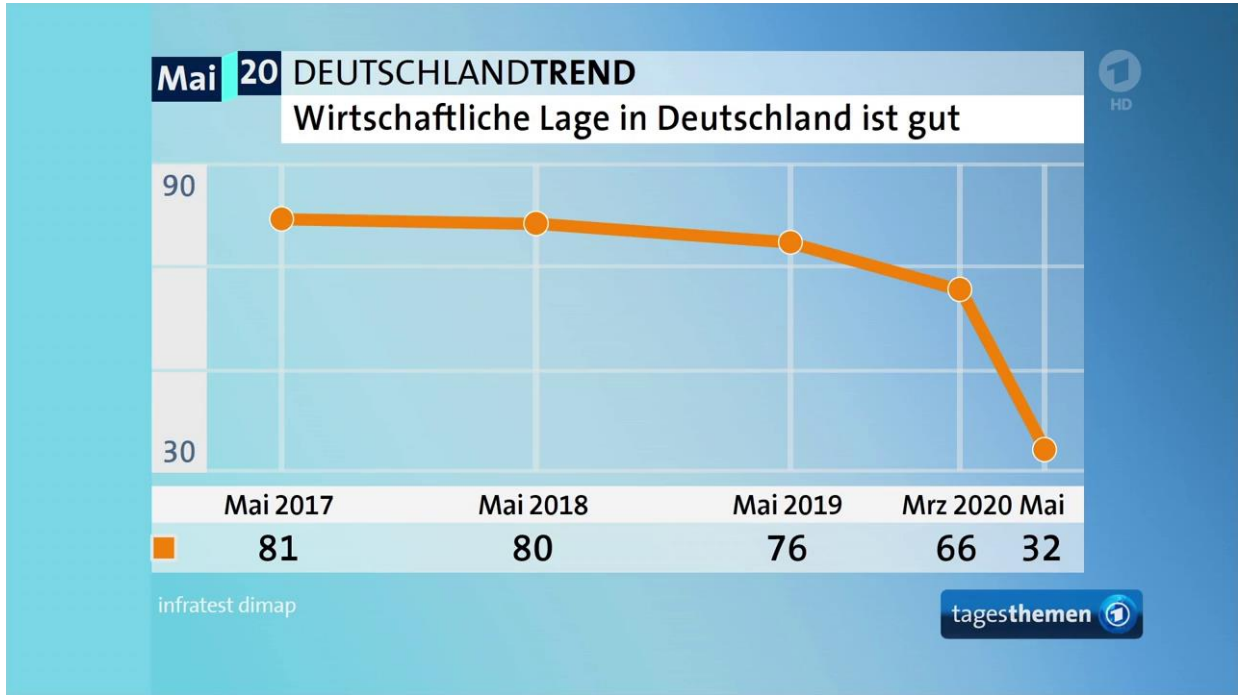
- Encoding: Wahlabsicht per Partei auf x Achse *und* Farbe!
- Geometrics: Balken (Höhe der Balken und Farbe der Balken)
- Scales:
  x-Achse: Parteien als Kategorien
  y-Achse: Wahlabsicht bis…
  Maximalwert! *(nicht 0-100%)*

Encodings, Geometrics, Scales



- Encoding:
- Geometrics:
- Scales:

Encodings, Geometrics, Scales



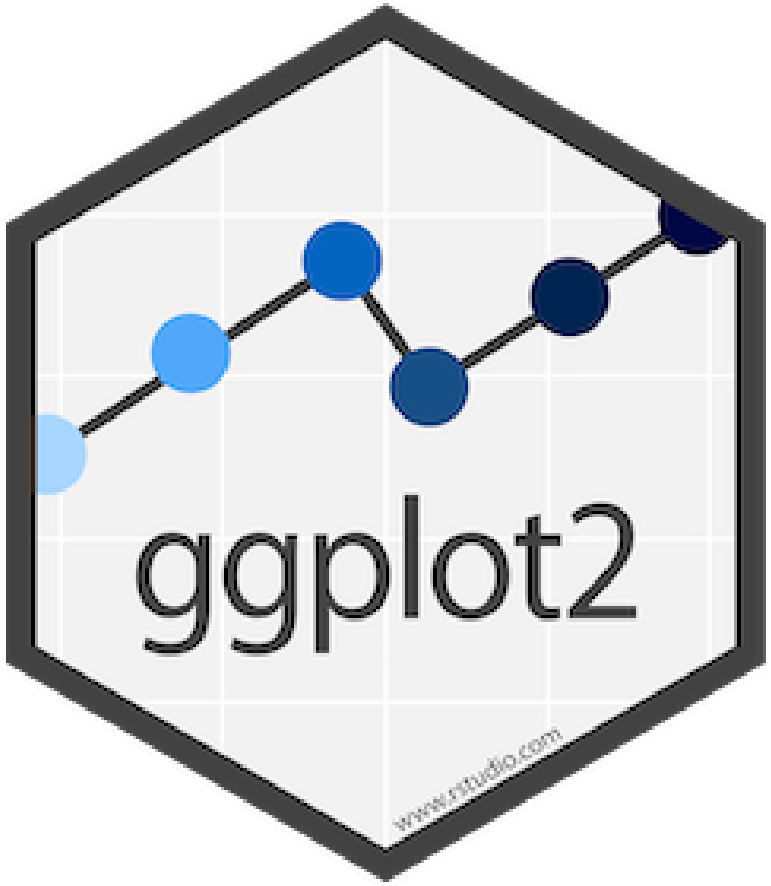
Was gefällt euch? Was gefällt euch nicht?

- Encoding: Zeit in Monaten und Zustimmung zu der Aussage
- Geometrics: Wert-Kombi als Punkte und verbindende Linie
- Scales:
  - x-Achse: Zeit (Mai 2017-2020)
  - y-Achse: Prozent, wieder nur Maximalwert (startet bei 30%)

Wofür Daten visualisieren?

- Ergebnisse kommunizieren
  - Zusammenhänge verständlich machen!
  - Zusammenhänge zugänglich machen!
  - Visuelle Vergleiche ermöglichen!
- Davor aber:
  - Daten visualisieren zum Begreifen
  - Datenqualität überprüfen


tidyverse
www.rstudio.com


ggplot2
www.rstudio.com

Datenvisualisieren mit tidyverse

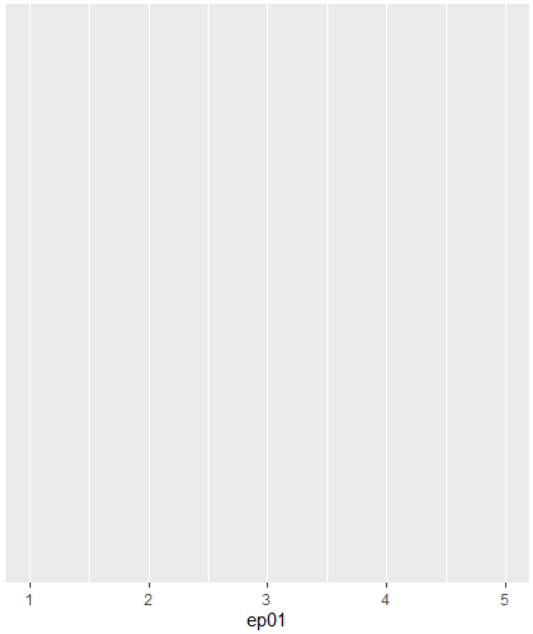
df %>% ggplot()

Encodings (aesthetics) hinzufügen:

df %>% ggplot(aes(x = variable_1, y = variable_2))

Aufgabe:

allbus %>% ggplot(aes(x = ep01_r))



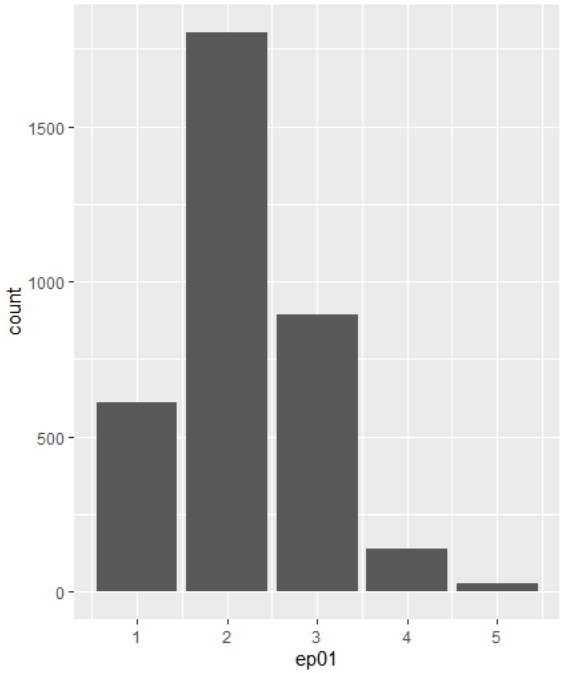
Datenvisualisieren mit tidyverse

Geometrics

```
df %>% ggplot(aes(x = variable_1, y = variable_2)) +
    geom_bar()
```

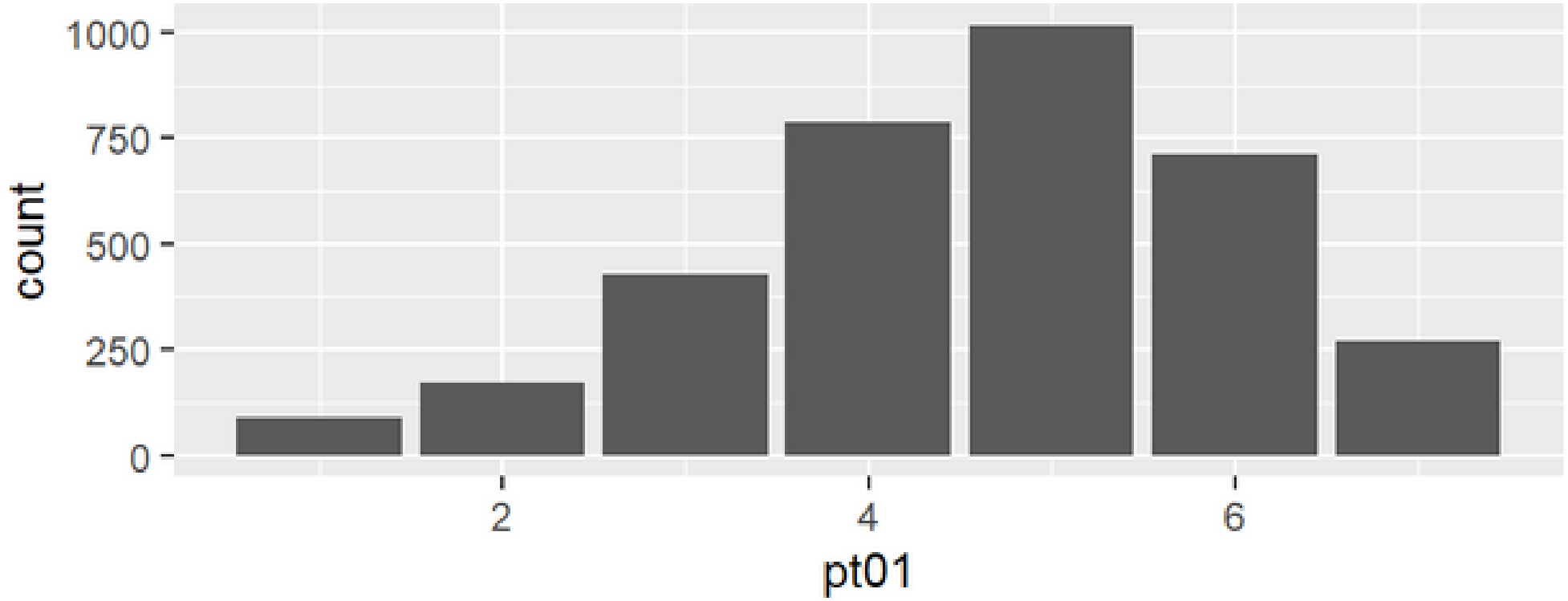
Aufgabe:

```
allbus %>% ggplot(aes(x = ep01_r)) +
    geom_bar()
```



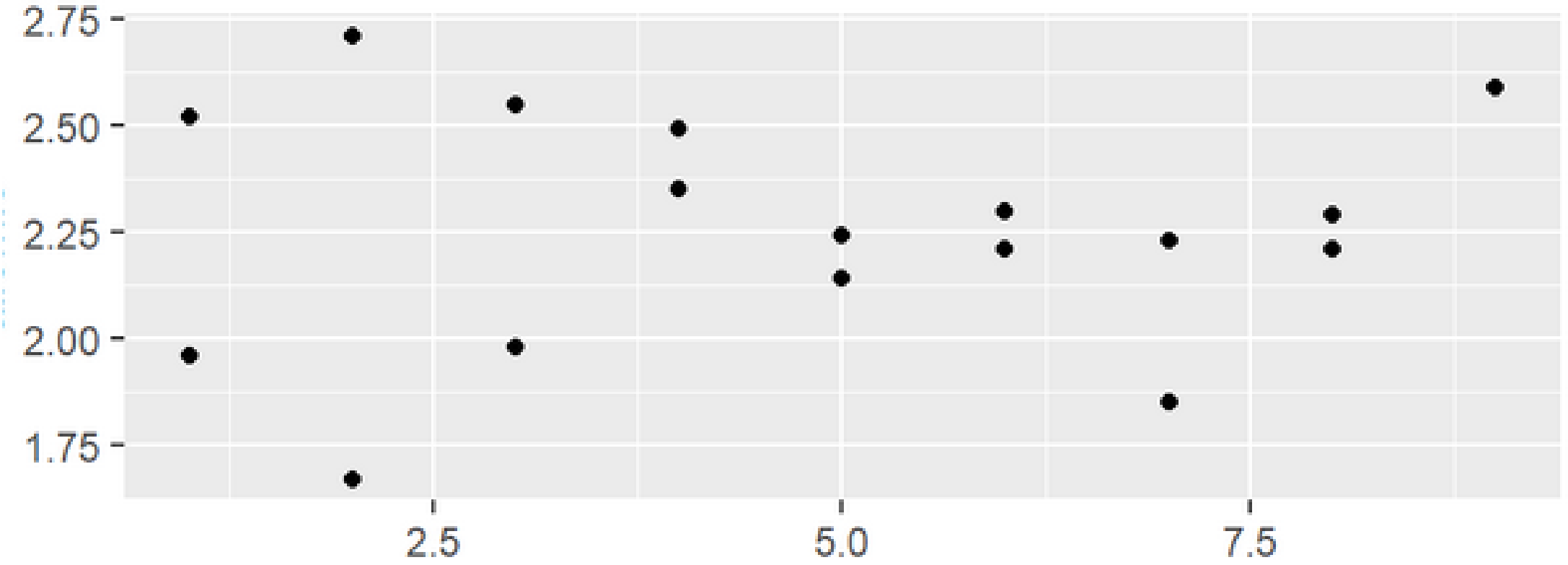
Auswahl von Darstellungen mittels ggplot2

- geom_bar()



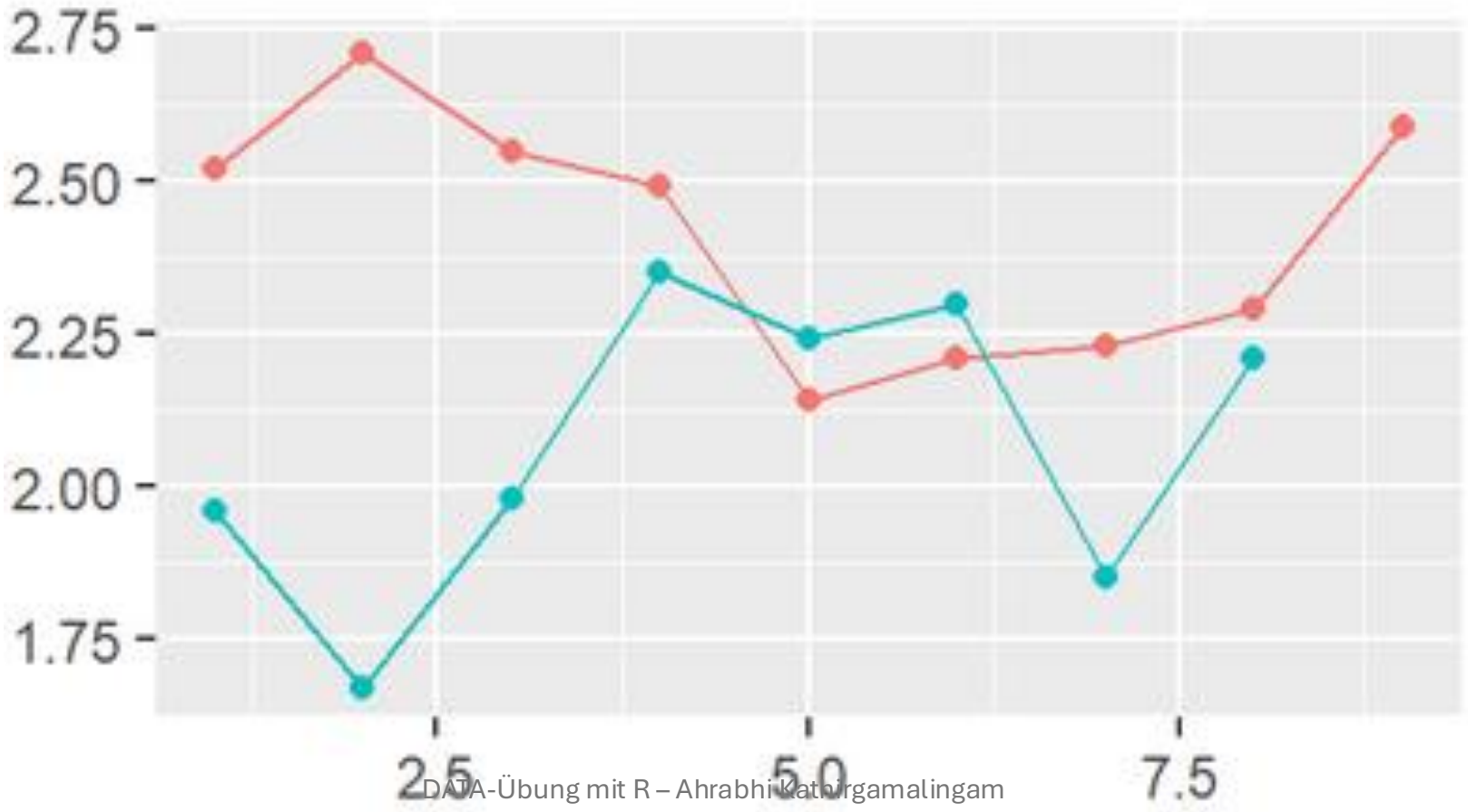
Auswahl von Darstellungen mittels ggplot2

- geom_point()



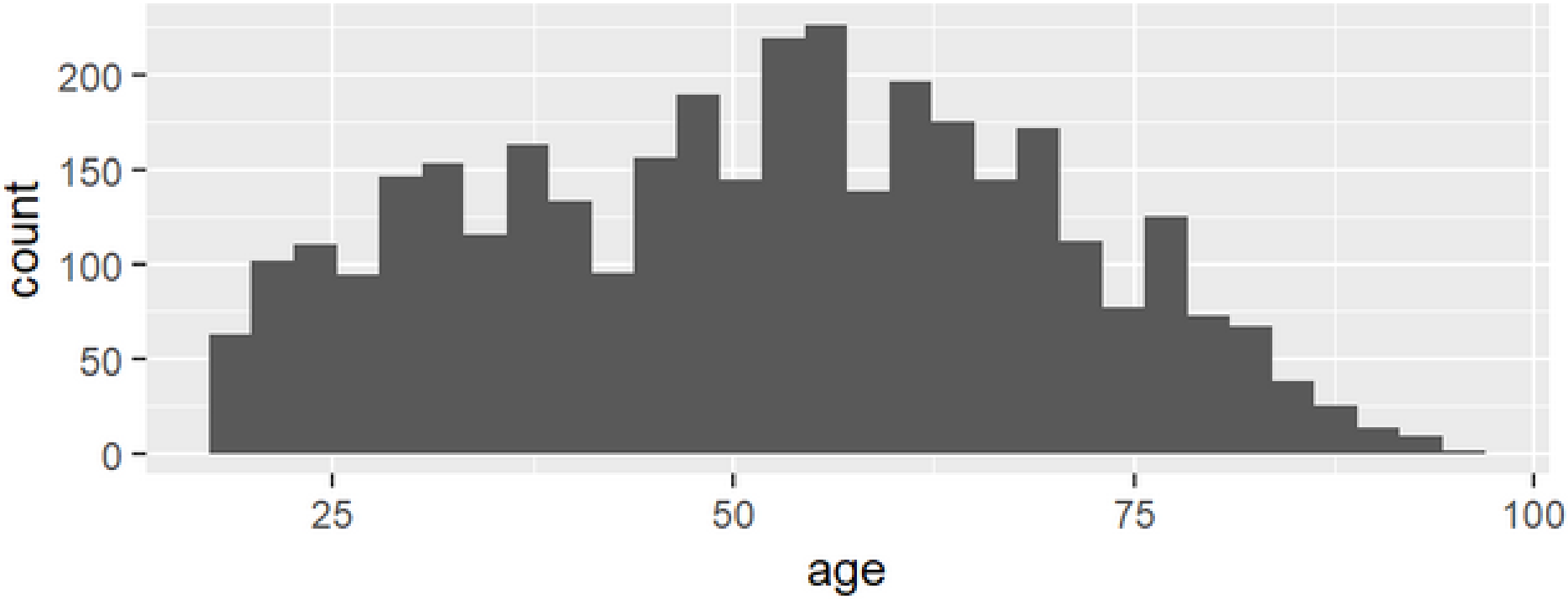
Auswahl von Darstellungen mittels ggplot2

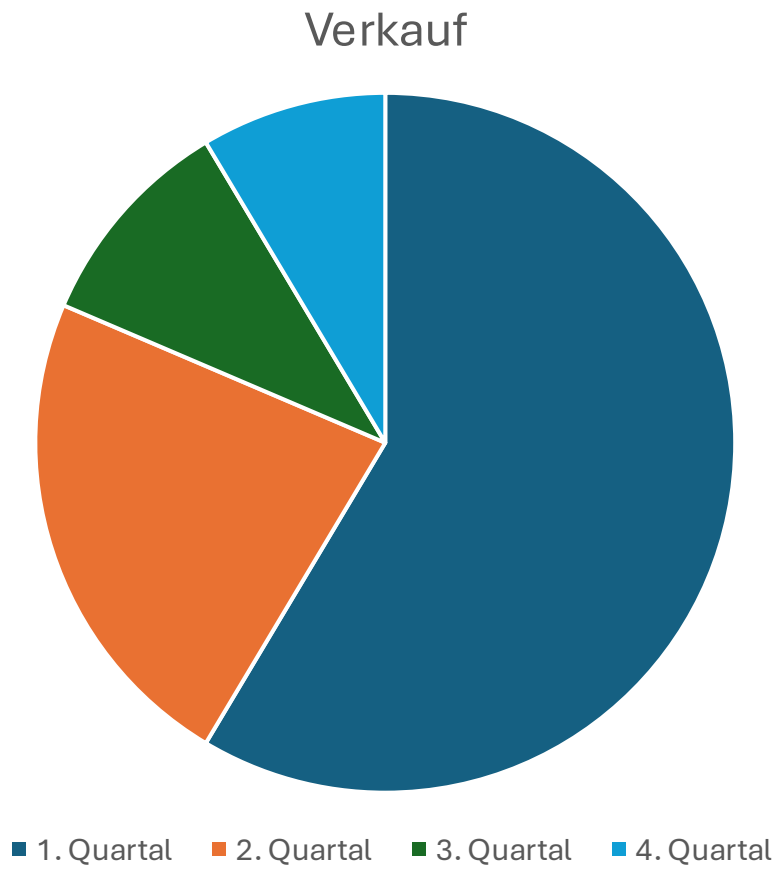
- geom_line()



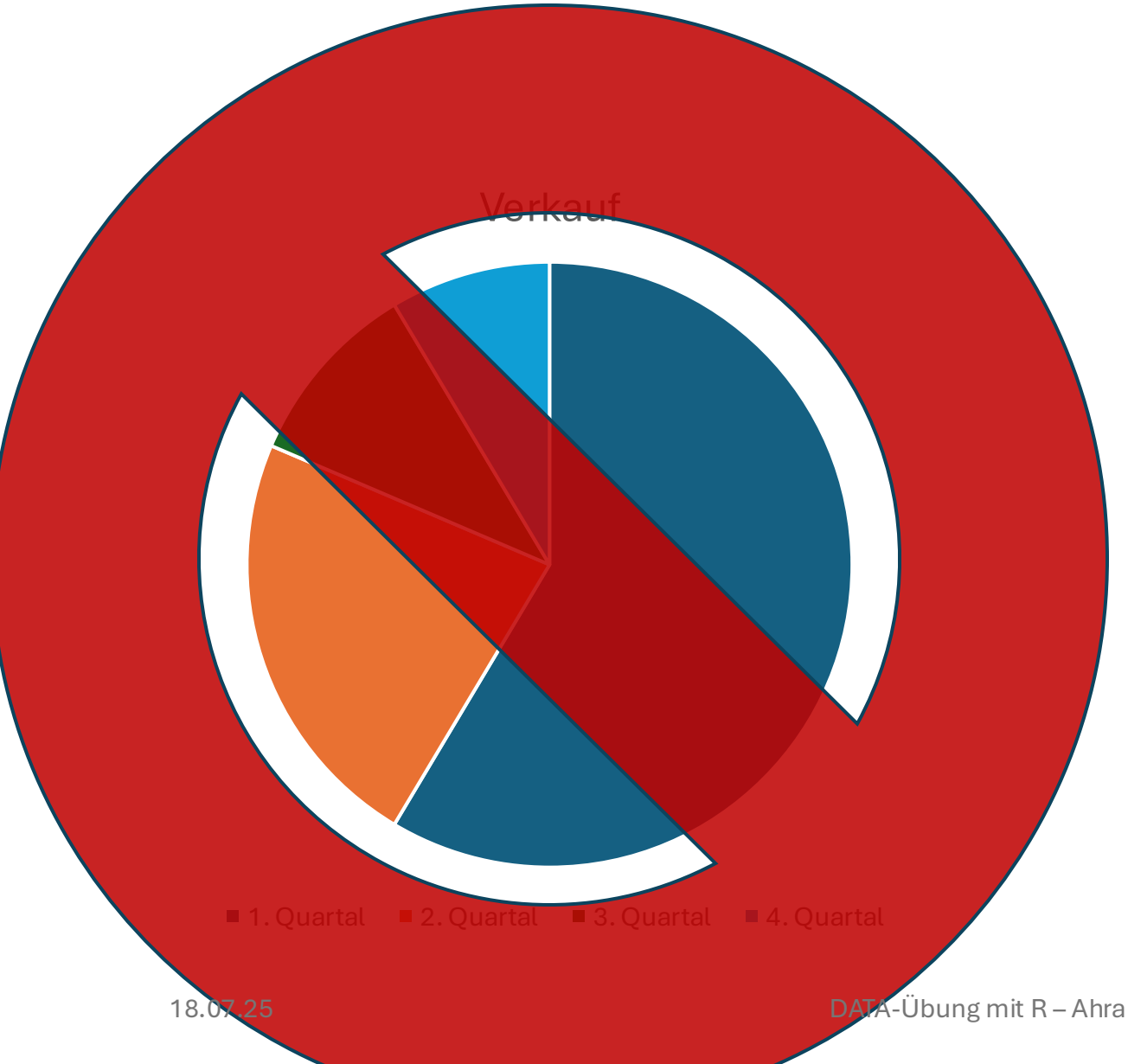
Auswahl von Darstellungen mittels ggplot2

- geom_histogramm()





Ja, Nein, Vielleicht?



- Keine klare Achse (Balken besser)
- Schlecht vergleichbar
- Proportionen werden oft falsch eingeschätzt
- Nicht üblich in Wissenschaft

DATA VISUALIZATION WITH MESSY AND UNAGGREGATED DATA

EXPECTATION

REALITY

SMITHESIS.COM

Tipps für Datenvisualisierung

- Erstmal: Ziele definieren!
- Passenden Diagrammtyp wählen
- Barrierefrei gestalten! (Kontraste, Farben, Beschriftungen)
- Achsen + Einheiten beschriften
- Farben sparsam und gezielt
- Daten nicht überladen – Fokus auf das Wesentliche!
- Leserichtungen beachten
- Titel und kurze Erklärung (Note)
- Reproduzierbar machen: über Code und nicht über zB Excel!

Datenvisualisieren mit tidyverse

Aufgabe: Unser Balkendiagramm verschönern: Beschriften, Farben, Titel, etc.

```
allbus %>% ggplot(aes(x = ep01_r)) +
        geom_bar()
```

```
allbus %>% ggplot(aes(x = ep01_r)) +
 geom_bar() +
 scale_x_continuous(breaks = c(1,2,3,4,5),
                    labels = c("?", “?", "?",
                               "?", "?"))
```

Datenvisualisieren mit tidyverse

```
allbus %>% ggplot(aes(x = ep01_r)) +
  geom_bar() +
  scale_x_continuous(breaks = c(1,2,3,4,5),
                     labels = c("sehr schlecht", "schlecht", "teils/teils",
                                "gut", "sehr gut"))
```

Datenvisualisieren mit tidyverse

```
allbus %>% ggplot(aes(x = ep01_r)) +
  geom_bar() +
  scale_x_continuous(breaks = c(1,2,3,4,5),
                     labels = c("sehr schlecht", "schlecht", "teils/teils",
                                "gut", "sehr gut")) +
  labs(x = “???",
       y = "???",
       title = "??? “)
```

Datenvisualisieren mit tidyverse

```
allbus %>% ggplot(aes(x = ep01_r)) +
  geom_bar() +
  scale_x_continuous(breaks = c(1,2,3,4,5),
                                labels = c("sehr schlecht", "schlecht", "teils/teils",
                                           "gut", "sehr gut")) +
  labs(x = “Einschätzung wirtschaftliche Lage",
         y = “Absolute Häufigkeit",
         title = “Abbildung 1: Subjektive Einschätzung der
         Wirtschaftlichen Lage “)
```

Datenvisualisieren mit tidyverse

```
plot_1 <-  allbus %>% ggplot(aes(x = ep01_r)) +
           geom_bar() +
           scale_x_continuous(breaks = c(1,2,3,4,5),
                              labels = c("sehr schlecht", "schlecht", "teils/teils",
                                         "gut", "sehr gut")) +
           labs(x = “Einschätzung wirtschaftliche Lage",
                y = “Absolute Häufigkeit",
                title = “Abbildung 1: Subjektive Einschätzung der
                    Wirtschaftlichen Lage “)
```

Datenvisualisieren mit tidyverse

```
plot_1 + coord_flip()

plot_1 + theme_minimal()

plot_1 + theme_dark()

plot_1+ theme(axis.text.x = element_text(angle = 40))
```

Troubleshooting

Skript: DATA_SoSe2025_Tag2-Troubles.R

Findet meine acht Fehler,
listet sie auf (mit # - Kommentarfunktion)
UND korrigiert sie aus!

REFRESHER

- Warum Daten visualisieren?
- Was halten wir von Tortendiagrammen?
- Worauf sollten wir beim Visualisieren achten?
- Was ist der Unterschied zwischen Nominalskala und Ordinalskala?
- Was ist der Unterschied zwischen Subsetting und Rekodieren?

Deskriptive Statistik

Deskriptivstatistik

- Beschreibung und Darstellung von Verteilungen
- Uni- und bivariate Analyse
  - Die univariate Statistik bezieht sich auf ein einziges Merkmal
  - Bivariate Analysen geben den Zusammenhang zwischen zwei Merkmalen wieder
- Häufigkeitsverteilungen, Maße der zentralen Tendenz (Lagemaße) und Streuungsmaße

Häufigkeitsverteilungen

- Absolute Häufigkeit

```
# Häufigkeitsverteilung Ost/West
table(allbus2018$eastwest)
##
##    1    2
## 2387 1090
```

Häufigkeitsverteilungen

- Relative Häufigkeit

```
prop.table(table(allbus2018$eastwest))
##
##         1         2
## 0.6865114 0.3134886
```

Häufigkeitsverteilungen

- Prozentuale Häufigkeit

```
prop.table(table(allbus2018$eastwest))*100
##
##         1         2
## 68.65114 31.34886
```

Häufigkeitsverteilungen

- Achtung: Welches Skalenniveau bzw. wie viele Ausprägungen hat die Variable?
- Häufigkeitsauszählung für die Variable „age“ table(allbus$age)
  - Darstellung unübersichtlich
  - Lösung: Alter der Befragten in Geburtskohorten (Klassifizierung -> Eigenschaften von Variablen verändern). Allerdings haben wir dann einen Informationsverlust, so gehen Unterschiede innerhalb der Klassen verloren)

Maße der zentralen Tendenz
Mean, Median, Mode

Maße der zentralen Tendenz

- Arithmetische Mittel (Mean)
  - Das arithmetische Mittel kennzeichnet den “Schwerpunkt einer Verteilung” (Diaz-Bone, 2019, S. 45)
  - Vergleich na.rm=TRUE und na.rm=FALSE

```
mean(allbus$age) funktioniert nicht!
mean(allbus$age, na.rm = TRUE)
```

- Median
  - „[Der Median]unterteilt die Reihe in zwei Hälften: die eine Hälfte der Ausprägungen ist kleiner als (oder höchstens gleich groß wie) der Median, die andere Hälfte der Ausprägungen ist größer als (oder zumindest gleich groß wie) der Median“ (Diaz-Bone, 2019, S. 45-46)

```
median(allbus$age)
median(allbus$age, na.rm = TRUE)
```

Maße der zentralen Tendenz

- Modus
  - Die häufigste Ausprägung einer mindestens nominalskalierten Variable ist der Modus. Der Modus ist ein „typischer“ Wert für eine Verteilung

```
modal_tabelle <- table(allbus2018$age)
names(modal_tabelle)[which(modal_tabelle==max(modal_tabelle))]
## [1] "55"
```

Streuungsmaße

- Varianz (s^2)
  - Maß für die Streuung einer Verteilung um ihren Mittelwert

```
var(allbus$age, na.rm = TRUE)
## [1] 311.2478
```

- Standardabweichung (s)
  - Wurzel aus der Varianz (gibt die Streuung in der ursprünglichen Maßeinheit an). Bei annähernd normalverteilten Merkmalen liegen ca. 68 % aller Werte im Bereich von ± 1 Standardabweichung um den Mittelwert

```
sd(allbus$age, na.rm = TRUE)
## [1] 17.64222
```

Siehe Lernressourcen und R – Ein Einführungsskript

Quartilabstand

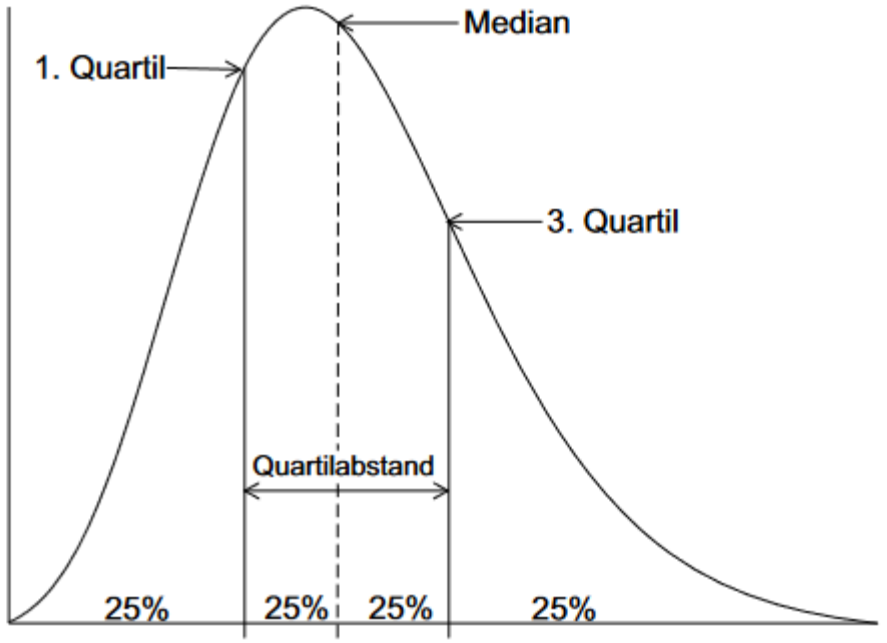


Abb. 2: Quartilabstand

Abbildung aus Weins 2010, S. 70

summary()

- Mindest, Maximalwert
- Missing Values
- Median
- Mean
- Quantile

summary(allbus$age)

Univariate Maßzahlen (Auswahl)

	Skalenniveau			
	Nominal	Ordinal	Intervall	Ratio
Modalwert	X	X	X	X
Median		X	X	X
Arithmetisches Mittel			X	X
Quartilabstand		(X)	X	X
Varianz & Standardabweichung			X	X

FRAGEN, UNKLARHEITEN, FEEDBACK?

VIELEN DANK ☺

UND BIS NÄCHSTE WOCHE!

ahrabhi.kathirgamalingam@cais-research.de