

Usable Security and Privacy Starting to Work with Data (in R)

Matthew Smith
Anna-Marie Ortloff
Florin Martius

Behavioural Security and Privacy Lab, Universität Bonn, Fraunhofer FKIE

Some slides courtesy of Dr. David Elweiler, University of Regensburg

- Intro to R – be able to use R and R Studio to import and export data, calculate common descriptive statistics ...
 - Know common measures of central tendency and spread and be able to discuss when to apply them
 - Calculate them (in R)
 - Know how to work with and manipulate dataframes
 - Know some common pitfalls when working with real data and some solutions
 - Recognize untidy data and know what format this data should be in to be tidy
 - Transform untidy data sets into the tidy data format

- Field, Andy, Miles, Jeremy & Field, Zoë (2012): Discovering Statistics Using R. Los Angeles/London/New Delhi: SAGE Publications.
- Navarro, Danielle. (2016) Learning statistics with R. Available at: <https://learningstatisticswithr.com/>
- Wickham, Hadley & Grolemund, Garret (2017) R for Data Science. Available at: <https://r4ds.had.co.nz/index.html>
- Ismay, Chester & Kim, Albert (2021) Statistical Inference via Data Science. A ModernDive into R and the Tidyverse. Available at: <https://moderndive.com/index.html>

INTRO TO R

What's R?

- Programming language
 - Specialised on statistical analysis and graphics
 - Compiling not necessary
-
- Based on S programming language
 - Ross Ihaka and Robert Gentleman started development on R in 1991
 - Since 1995 freely available under Gnu General Public License
-
- IDE: R-Studio
-
- Download instructions on eCampus (Lecture R Code)



R - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Source on Save Run Source

```

1 library(tidyverse)
2 library(utis)
3
4 # use data from https://www.ecb.europa.eu/stats/policy_and_ex
5 conversion_rate_info <- read.csv("original-data/eurofxref-his
6   select(Date, USD) %>%
7   mutate(Date = as.character(Date)) %>%
8   separate(Date, into=c("year", "month", "day"), sep="-") %>%
9   select(-month, -day) %>%
10  mutate(year = as.numeric(year)) %>%
11  group_by(year) %>%
12  summarise(USD_avg_rate=mean(USD))
13
14
15 ransomware_info <- read.csv("cleaned-data/all_data_with_judgm
16 num_participants <- nrow(ransomware_info)
17
187:1 (Top Level) R Script

```

Environment History Connections Tutorial

R Global Environment

Data

Object	Size
amounts	45 obs. of 10 variables
conversion_rate_info	23 obs. of 2 variables
crypto_amounts	5 obs. of 10 variables
infections	176 obs. of 5 variables
infections_all	3 obs. of 3 variables
infections_rec	infections_all (tbl_df, 1216 bytes) of 3 variables
infections_tim	of 2 variables
more_amount_info	176 obs. of 7 variables
non_police	83 obs. of 53 variables

Files Plots Packages Help Viewer

Install Update

Name	Description	Version
abind	Combine Multidimensional Arrays	1.4-5
AER	Applied Econometrics with R	1.2-9
ArgumentCheck	Improved Communication to Users with Respect to Problems in Function Arguments	0.10.2
askpass	Safe Password Entry for R, Git, and SSH	1.1
assertthat	Easy Pre and Post Assertions	0.2.1
backports	Reimplementations of Functions Introduced Since R-3.0.0	1.2.0
base64enc	Tools for base64 encoding	0.1-3
bayestestR	Understand and Describe Bayesian Models and Posterior Distributions	0.9.0
BH	Boost C++ Header Files	1.75.0-0

Console

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.

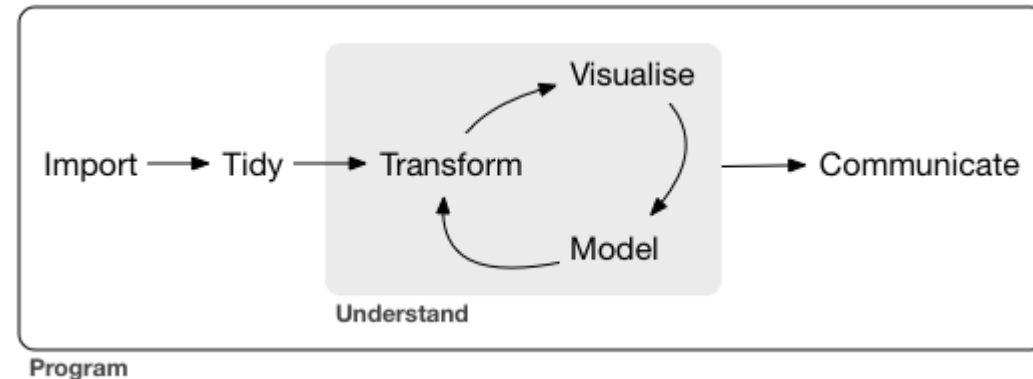
[Workspace loaded from ~/Promotion/Supervising/BAS/2021-Maike-Vossen/R/.RData]

> |

DEMO

R-script on eCampus

- Load necessary packages
- Load data into R
 - data is typically in form of csv-files
 - other file formats or database connection also possible
- Work with data
 - Prepare data
 - Analyse Data
 - Visualize Data



<https://r4ds.had.co.nz/introduction.html>

- Possibly: Export data or visualizations (so others can use them)

DESCRIBING DATA

Why would we want to describe data?

- People are very bad at judging – they are biased in many ways:
 - Particularly true of summarizing large amounts of data
- Necessary to describe data and their distribution
 - to use particular statistical techniques
 - work out, which one we should use
- Science is hard and the truth is sometimes hidden in complicated data.
- Statistics provides us with tools to untangle this data and find out the truth.

What do you mean biased?

- Simpson's paradox (University of California, Berkeley, applicants)

	Number of applicants	Percent admitted
Males	8442	44%
Females	4321	35%

- If you look at these figures you think there is a strong bias in favour of male applicants
- What if I were to say that these data actually reflect a weak bias in favour of women?

Example from Navarro, Danielle. (2016) Learning statistics with R

Let's look at the data a little more closely

Department	Males		Females	
	Applicants	Percent admitted	Applicants	Percent admitted
A	825	62%	108	82%
B	560	63%	25	68%
C	325	37%	593	34%
D	417	33%	375	35%
E	191	28%	393	24%
F	272	6%	341	7%

- Most departments have a higher admission rate for females – how can this be?
- Variation in applicants across departments
- Females tend to apply for departments with low admission rates

Bickel, P. J., Hammel, E. A., & O'Connell, J. W. (1975). Sex bias in graduate admissions: Data from Berkeley. *Science* , 187 , 398{404. 6 , 7

Here we can see this graphically

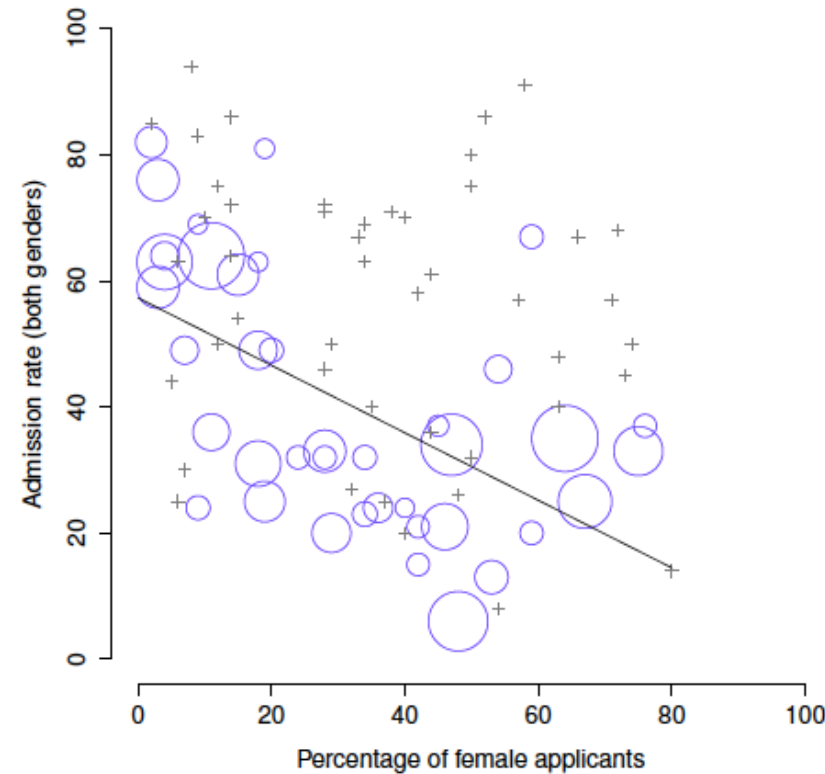


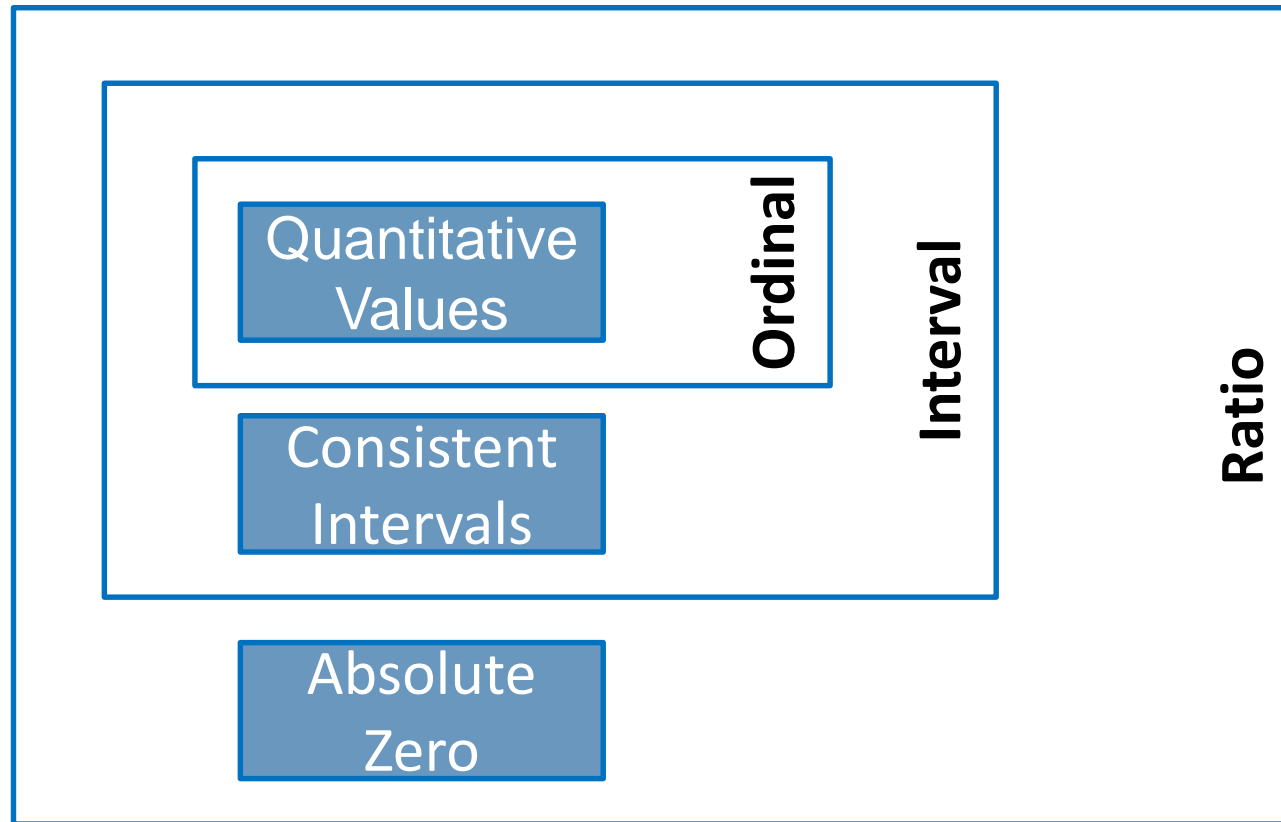
Figure 1.1: The Berkeley 1973 college admissions data. This figure plots the admission rate for the 85 departments that had at least one female applicant, as a function of the percentage of applicants that were female. The plot is a redrawing of Figure 1 from [Bickel et al. \(1975\)](#). Circles plot departments with more than 40 applicants; the area of the circle is proportional to the total number of applicants. The crosses plot department with fewer than 40 applicants.

Navarro, Danielle (2016). Learning statistics with R.

- Measures of central tendency
 - Arithmetic mean
 - Median
 - Mode
- Measures of spread
 - IQR
 - Standard Deviation
 - Variance
- Graphical descriptions (more on these later)
 - Describe distributions: e.g. Boxplots
 - Describe relationships: e.g. Scatterplots

CENTRAL TENDENCY

- Serves to describe the center of a distribution: Which single value best describes the data?
- Which measure to use depends on:
 - Context
 - The data
 - The scale of measurement
- 3 „classic“ measures
 - Mean
 - Median
 - Mode



Nominal
Scale

- **Nominal**
 - discrete, but unrelated categories, equal or unequal
 - example: day/night, type of cake (chocolate, vanilla, fruit)
- **Ordinal**
 - discrete categories that can be ordered
 - example: 5/7 point scales, small/medium/large pizza
- **Interval scale: discrete or continuous measurements where distance makes sense**
 - example: Temperature in Celsius, often ratings
- **Ratio scale: Interval + true zero (ratios make sense)**
 - example: Age, Money, Distance, Temperature in Kelvin, Lines of code

Arithmetic mean of ungrouped empirical data

- The sum of all observed values divided by the number of observations.

$$\bar{x} = \frac{1}{n} \cdot (x_1 + \dots + x_n) = \frac{1}{n} \sum_{i=1}^n x_i$$

- Defined meaningfully for continuous variables
- Sensitive to outliers

- As a starting point you need a sorted list
 $x(1) \leq \dots \leq x(i) \leq \dots \leq x(n)$
- Definition

$$x_{med} = \begin{cases} x_{(\frac{n+1}{2})} & \text{for uneven } n \\ \frac{1}{2}(x_{(n/2)} + x_{(n/2+1)}) & \text{for even } n \end{cases}$$

8	12	31	45	56
---	----	----	----	----

8	12	31	32	45	56
---	----	----	----	----	----

$$\text{Median} = 0.5 \cdot (31+32)$$

- The data should be at least ordinally scaled.
- Robust, which means less susceptible to outliers
- Easily interpretable
 - At least 50% of the data are less than or equal to x_{med}
 - At least 50% of the data are greater than or equal to x_{med}

Generalization: What are Quantiles?

- Again we need an ordered list
- p-Quantile: Every value x_p where
 - $0 < p < 1$,
 - at least proportion p of the data is less than or equal x_p and
 - at least a proportion $1-p$ is greater than or equal to x_p .
- Lower Quartile = 25%-Quantile = $x_{0.25}$
- Upper Quartile = 75%-Quantile = $x_{0.75}$

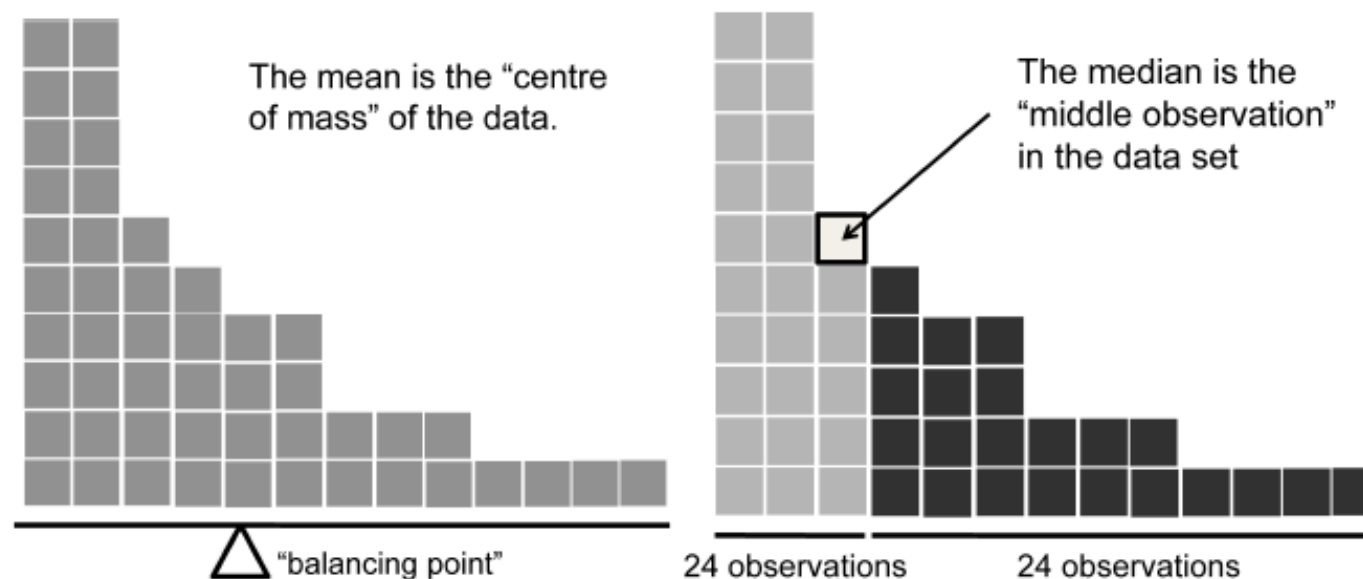
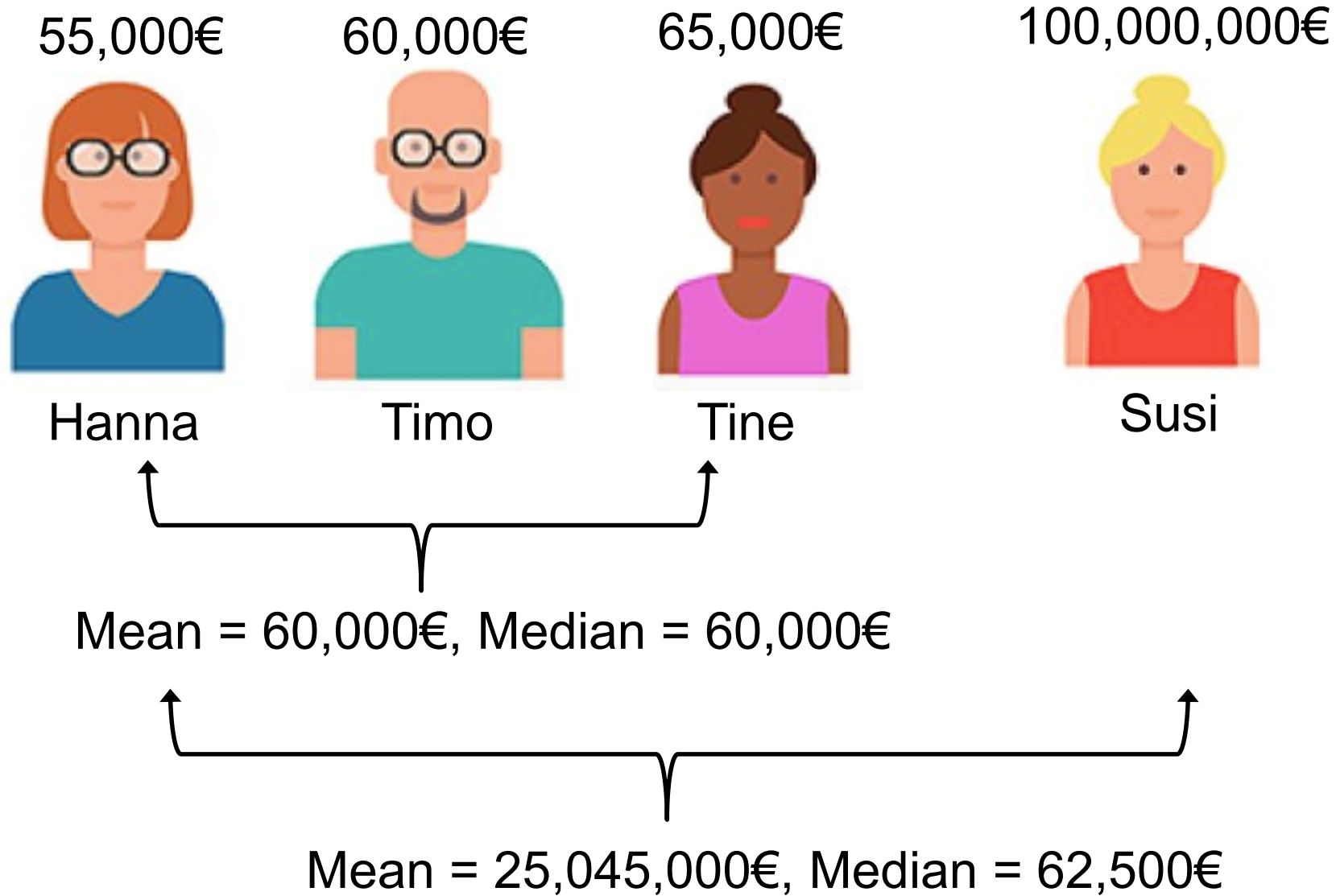


Figure 5.2: An illustration of the difference between how the mean and the median should be interpreted. The mean is basically the “centre of gravity” of the data set: if you imagine that the histogram of the data is a solid object, then the point on which you could balance it (as if on a see-saw) is the mean. In contrast, the median is the middle observation. Half of the observations are smaller, and half of the observations are larger.

Navarro, Danielle. (2016) Learning statistics with R. p118

Median vs Mean



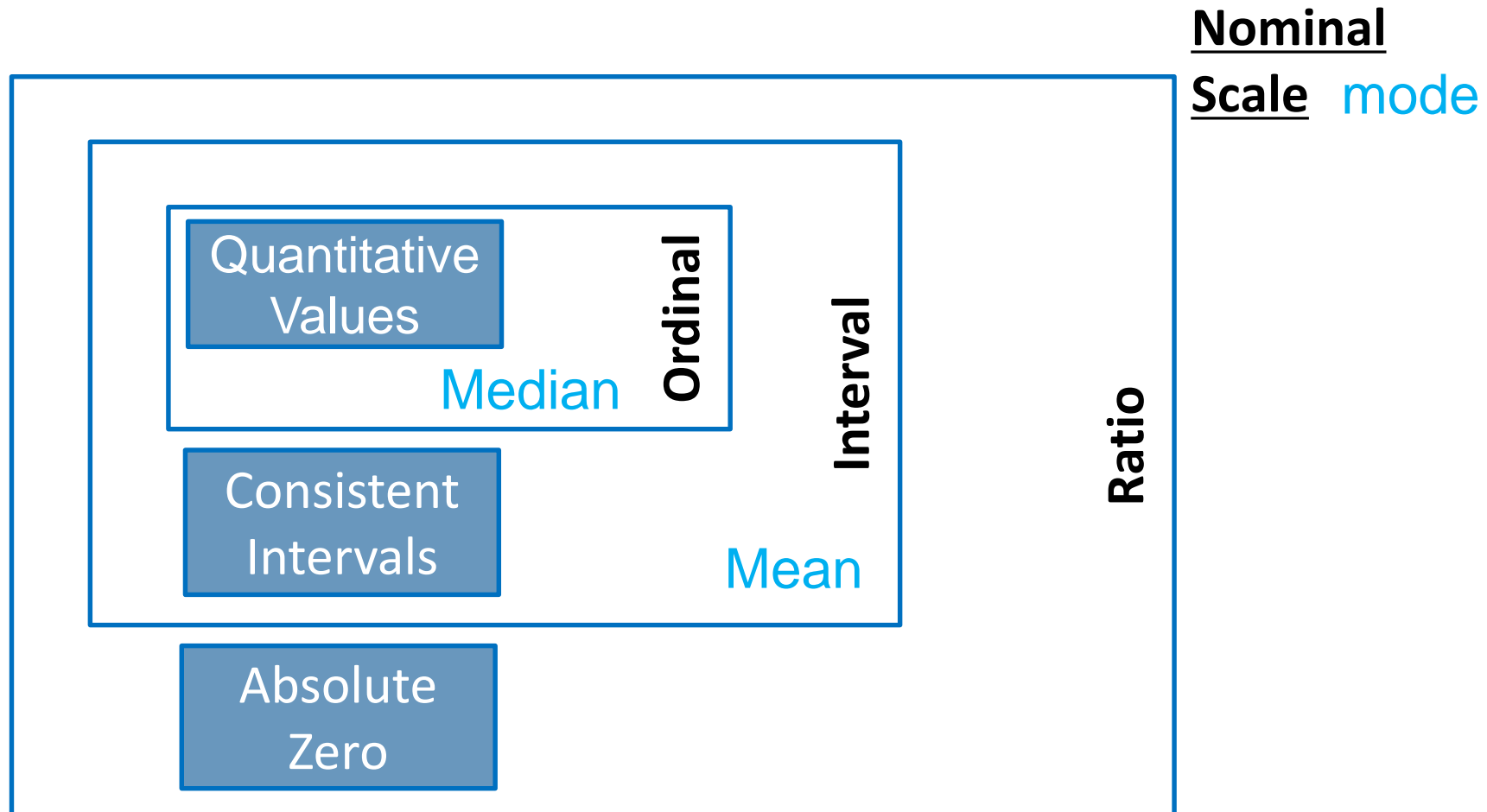
Median vs Mean

- If you are interested in looking at the **overall income** at the table: **mean**
- If you are interested in what counts as a **typical income** at the table: **median**

- Definition: The most common “number”
- The data need to be at least nominal-scaled
- Unambiguous, if the frequency distribution has a single maximum

8 12 15 21 31 31 31 45 46 46 56

When to use what?



Median vs Mean vs Mode

- If your data are **nominal scale**, mean or the median most likely inappropriate → **mode**
- If your data are **ordinal scale**, **median** is more appropriate than the mean.
- The median only makes use of the order information in your data (i.e., which numbers are bigger), but doesn't depend on the precise numbers involved (i.e. ordinal scale).
- The **mean** makes use of the precise numeric values assigned to the observations, so it's not really appropriate for ordinal data
- For **interval and ratio scale data**, both the **mean and median** are generally acceptable.
- The mean has the advantage that it uses all the information in the data (useful when you don't have a lot of data)
- The mean is very sensitive to extreme values

- **Arithmetic mean:** `mean(x)`

- **Median:** `median(x)`

- **No built-in function for mode:**

```
calc_mode <- function(x) {  
  ux <- unique(x)  
  tab <- tabulate(match(x, ux))  
  ux[tab == max(tab)]  
}  
calc_mode(x)
```

<https://stackoverflow.com/questions/2547402/is-there-a-built-in-function-for-finding-the-mode>, zuletzt abgerufen am 20. Juni 2021

- Which measure of central tendency can be used to describe the following variable:
"time to complete programming task" ?
 - ☐ Mean
 - ☐ Median
 - ☐ Mode

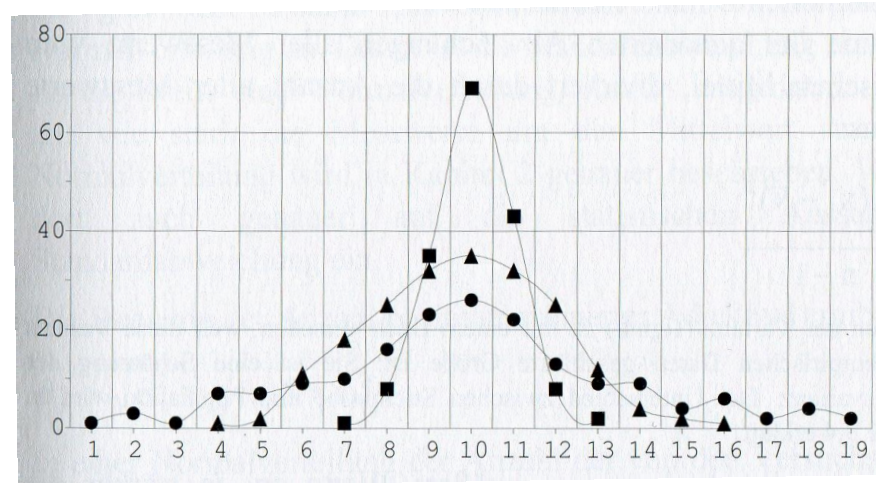


<https://pingo.coactum.de/426954>

SPREAD

Why measure variance and spread?

- Spread: „to what extent do values deviate from the average?“
- Completely different distributions can lead to the same mean



cf. Rasch et al. 2010, p. 19

- Explain differences when the characteristic values of different persons / objects, etc. change
- Explanation of difference requires quantification of difference

Describing the spread: IQR

- 2, 4, 5, 6, 8, 8, 9, 11, 12, 14
- **Range** = max – min = 14-2 = 12
- **Interquartile range (IQR)** : 75%-25% = 6
- 2, 4, **5**, 6, 8, 8, 9, **11**, 12, 14

- Sample Variance:

$$Var(X) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu)^2$$

$n =$
sample
size

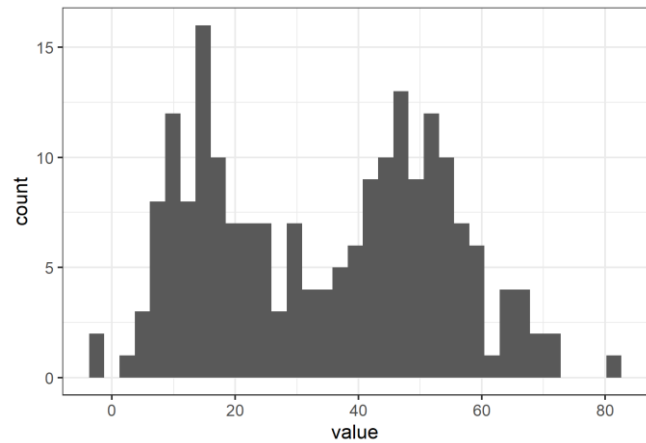
$\mu =$
sample
mean

- Standard deviation σ is the square root of the variance of X
 - Easier to interpret because of same unit

- Variance: `var(x)`
- Standard deviation: `sd(x)`
- Interquartile-Range: `IQR(x)`

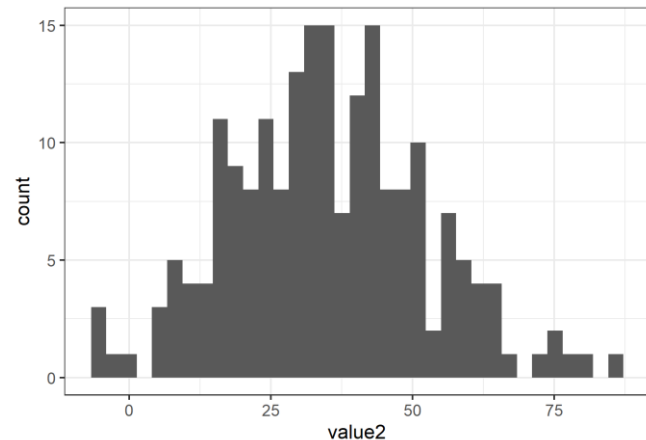
- Which of the two distributions has approximately the following parameters:
 $M=35$, $SD=18$?

- **A**



$M=34.4$
 $SD=19.0$

- **B**



$M=35.0$
 $SD=17.2$



<https://pingo.coactum.de/426954>

TYPICAL PROBLEMS

- Transforming a factor to an integer:

```
grades <- as.factor(c(1.3, 3.7, 2.5, 2.0, 3.7))  
grades_num <- as.numeric(grades)
```

- **Expected outcome:**

```
> grades_num  
[1] 1.3 3.7 2.5 2.0 3.7
```

- **Actual outcome:**

```
> grades_num  
[1] 1 4 3 2 4
```

- Factors are used to represent nominal data
- Two layers
 - Label representation as characters
 - Underlying numeric representation
- **Solution:**

```
grades_num <- as.numeric(as.character(grades))
```

Generalised: The Data Type Problem

- Unexpected results: in graphs, in stats, in summaries
 - Often solved by checking which data type is used

```
> class(grades)
[1] "factor"
> class(grades_num)
[1] "numeric"
> is.numeric(grades_num)
[1] TRUE
```

- Changing data-types (We've already seen an example)

```
> grades <- as.character(grades_num)
> class(grades)
[1] "character"
```

- Using a vector of values in a calculation (e.g. descriptive stats)

```
> data <- c(1, 3, NA, 5, 6, 2, 1, 2, 34, 5, 6, 7, 8, 8, 3, 2, NA)
> mean(data)
```

- **Expected outcome:** some numeric value, excluding the NA values

- **Actual outcome:** [1] NA

- R can't calculate mean because there are missing values
- You have to specify what to do with them

- **Solution:**

```
> mean(data, na.rm=TRUE)
[1] 6.2
```

- Loading data from csv

```
cars_df <- read.csv("cars.csv")
```

- **Expected outcome:** dataframe with 9 columns
- **Actual outcome:** big blob of data in one column

	model.mpg.cyl.disp.hp.drat.wt.num_front_passengers.num_back_passengers
1	Mazda RX4;21;6;160;110;03. Sep;Feb 62;2;2
2	Mazda RX4 Wag;21;6;160;110;03. Sep;2.875;2;3
3	Datsun 710;22. Aug;4;108;93;Mrz 85;Feb 32;2;2
4	Hornet 4 Drive;21. Apr;6;258;110;03. Aug;3.215;2;3
5	Hornet Sportabout;18. Jul;8;360;175;Mrz 15;Mrz 44;...
6	Valiant;18. Jan;6;225;105;Feb 76;Mrz 46;2;3
7	Duster 360;14. Mrz;8;360;245;Mrz 21;Mrz 57;2;3
8	Merc 240D;24. Apr;4;146.7;62;Mrz 69;Mrz 19;2;2

- **Solution:** check which separator is used and adapt either the file or the code
 - Sep = ;
 - Sep = ,


```
cars_df <- read.csv2("cars.csv")  
cars_df <- read.csv("cars.csv")
```


WORKING WITH DATA FRAMES IN R

Select subsets of rows

- **dplyr:** `filter(df, Auswahlbedingung)`
- Selection criteria:
- Comparisons with `==`, `!=`, `>`, `<`
- You can combine criteria using `&` and `|`
- Example:
- `filter(school_df, homeroom==103 & num_electives==2)`

name	homeroom	num_electives
Max	103	2
Tina	103	1
Mia	104	2



name	homeroom	num_electives
Max	103	2

Select subsets of columns

- **dplyr:** `select(df, Column names to select)`
- Syntax: Column names without quotes
- Example:
 - `select(school_df, name, num_electives)`
 - `select(school_df, -homeroom)`

name	homeroom	num_electives
Max	103	12
Tina	103	11
Mia	104	12



name	num_electives
Max	12
Tina	11
Mia	12

Inserting a new column

- **Base-R:** `df$newColumnName <- Calculation to store`
- **dplyr:** `mutate(df, newColumnName = Calculation to store)`
- **Example:**
- `mutate(school_df, num_normal_subjects = c(8, 9, 7))`

name	num_electives
Max	2
Tina	1
Mia	2



name	num_electives	num_normal_subjects
Max	2	8
Tina	1	9
Mia	2	7

Inserting a new column

- **Base-R:** `df$newColumnName <- Calculation to store`
- **dplyr:** `mutate(df, newColumnName = Calculation to store)`
- **Example:**
- `mutate(school_df, num_normal_subjects = c(8,9,7))`
- `mutate(school_df, total_subjects = num_electives + num_normal_subjects)`

name	num_electives	num_normal_subjects
Max	2	8
Tina	1	9
Mia	2	7



name	num_electives	num_normal_subjects	total_subjects
Max	2	8	10
Tina	1	9	10
Mia	2	7	9

- Lots more information here:
 - <https://nyu-cdsc.github.io/learningr/assets/data-transformation.pdf>
 - <https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf> (old version of the cheat sheet, includes tidyr)

Put this in your pipe!

- Idea

- Multiple operations are performed one after the other
- Each operation is performed on the result of the previous operation

```
school_df <- select(school_df, -homeroom)
school_df <- mutate(school_df, num_normal_subjects = c(8, 9, 7))
school_df <- mutate(school_df, total_subjects = num_electives + num_normal_subjects)
```

- In dplyr you use the Pipe-Operator %>% from the magrittr-package for this

- Example

```
school_df <- data.frame(name, homeroom, num_electives) %>%
  select(-homeroom) %>%
  mutate(num_normal_subjects = c(8, 9, 7)) %>%
  mutate(total_subjects = num_electives + num_normal_subjects)
```

TIDY DATA

Why tidy?

- Standard way to organize data
- Makes starting analyses and exploring data easier
- Tools in the tidyverse work with tidy data

What is a data set?

Often: Dataframe = data set

Sometimes data sets consist of multiple dataframes

- **Values** (e.g. numbers: 1, 3.45, strings: „good“)

Each value belongs to

- a **Variable** – measures an underlying attribute (e.g. height)
- an **Observation** – all values measured on one unit (e.g. a single participant in a study)

When is a dataset tidy?

Related to 3rd Normal Form (Databases)

1. Each variable forms a column.
2. Each observation forms a row.
3. Each type of observational unit forms a table.

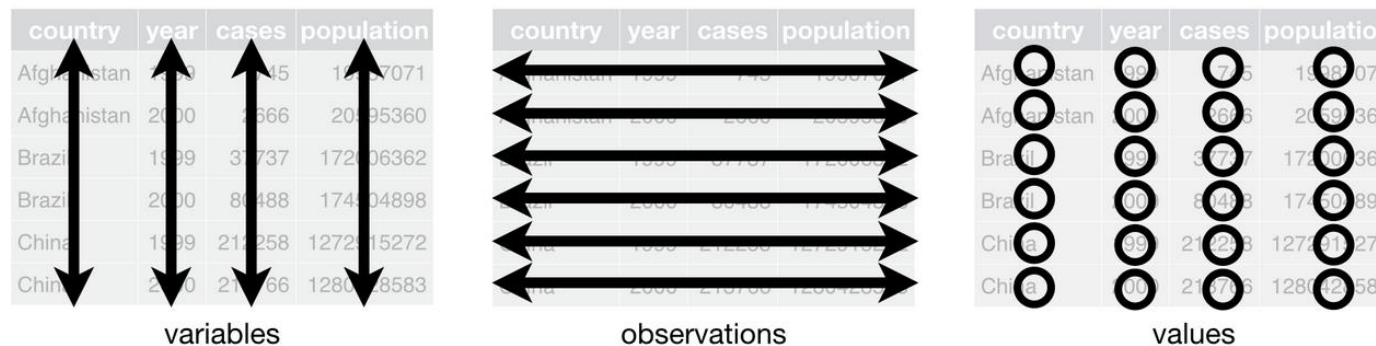


Figure 12.1: Following three rules makes a dataset tidy: variables are in columns, observations are in rows, and values are in cells.

<http://vita.had.co.nz/papers/tidy-data.pdf>, p. 4, last accessed 30.05.2021
<https://r4ds.had.co.nz/tidy-data.html>, last accessed 30.05.2021

What is an observational unit?

- Values in a dataset are collected at multiple levels
- Each of these levels is an observational unit
- Example: Study with multiple questionnaires, participant data and eye tracking data
 - Pre-study questionnaire
 - Participant data (demographics)
 - Eye tracking data
 - Post-study questionnaire

MESSY DATA

Problem I

- Column names represent values instead of variable names.
- Often used for presenting data

gender	pregnant	not_pregnant
male	0	32
female	2	30

- Actual variable names: Gender, Pregnancy Status, Frequency

Problem II

- A column represents more than one variable.

semester	course	sec_m	sec_f	info_m	info_f
SoSe19	36689	20	15	35	17
WiSe19	37890	34	10	38	27
SoSe20	36689	21	20	31	15

- E.g. Column Sec_m represents the attendance number for male cyber security students

Problem III

- Variables are represented in both rows and columns.

semester	course	measurement	sec	info	cs
SoSe19	36678	highest_grade	1.3	1.3	2.0
SoSe19	36678	lowest_grade	4.0	3.7	3.3
SoSe19	36689	highest_grade	1.0	1.7	1.3
SoSe19	36689	lowest_grade	3.7	4.0	3.7

- measurement contains the variables highest_grade and lowest_grade, while the sec, inf, and cs columns belong to a single variable major subject

- Different types of observations are represented in a single table.

participant_id	age	gender	questionnaire	q1	q2	q3
1	23	w	USE	1	4	1
1	23	w	SUS	2	7	4
1	23	w	UES	3	4	7
1	23	w	UIST	3	3	4
2	25	m	USE	3	5	2

- One type of observation concerns the questionnaires, another is the participant data (which is repeated frequently)

Problem V

- One observation is spread over several tables.
- E.g. Several datasets with the same or similar structure for different years or months

📁 2019-08	August 2019 data (#179)	7 months ago
📁 2019-09	September 2019 data (#180)	6 months ago
📁 2019-10	Add October 2019 data. (#183)	5 months ago
📁 2019-11	Add November 2019 data.	4 months ago
📁 2019-12	New tracker database (#186)	3 months ago
📁 2020-01	Add January 2020 data (#190)	2 months ago
📁 2020-02	Feb 2020 data update (#193)	29 days ago

<https://github.com/cliqz-oss/whotracks.me/tree/master/whotracksme/data/assets>

MAKE IT TIDY

Wide vs. Long data formats

ID	Alter	Gewicht	Familienstand	Beruf
P1	25	75	Ledig	Schreiner
P2	16	55	Ledig	Azubi

↔ **WIDE**

LONG

↔

ID	Variable	Value
P1	Alter	25
P1	Gewicht	75
P1	Familienstand	Ledig
P1	Beruf	Schreiner
P2	Alter	16
P2	Gewicht	55
P2	Familienstand	Ledig
P2	Beruf	Azubi

pivot_longer



- Get long format (and solve problem 1)
- Successor to `gather()`
- **Syntax:** `pivot_longer(df, cols=columns_to_pivot, names_to=name_of_the_column_for_the_current_colnames, values_to=name_of_the_column_for_the_current_values)`

```
problem1_tidy <- problem1_df %>%
  pivot_longer(cols=pregnant:not_pregnant, names_to="pregnancy_status", values_to="frequency")
```

Gender	pregnant	not_pregnant
Male	0	32
Female	2	30



Gender	pregnancy_status	frequency
Male	pregnant	0
Male	not_pregnant	32
Female	pregnant	2
Female	not_pregnant	30



- Get wide format (and solve part of problem III)
- Successor to `spread()`
- **Syntax:** `pivot_wider(df, names_from, values_from)`

course	measurement	grade	major
36678	highest grade	1.3	Sec
36678	lowest grade	4.0.	Sec
36689	highest grade	1.0	Sec
36689	lowest grade	3.7	Sec



course	major	highest_grade	lowest_grade
36678	Sec	1.3	4.0
36689	Sec	1.0	4.7



- Get wide format (and solve part of problem III)
- Successor to `spread()`
- **Syntax:** `pivot_wider(df, names_from, values_from)`

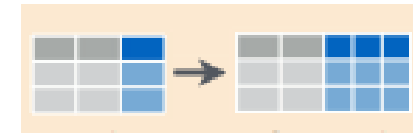
```
problem3_tidy <- problem3_df %>%
  pivot_longer(cols=sec:cs, names_to="major", values_to="grade") %>%
  pivot_wider(names_from = measurement, values_from = grade)
```

course	measurement	grade	major
36678	highest grade	1.3	Sec
36678	lowest grade	4.0.	Sec
36689	highest grade	1.0	Sec
36689	lowest grade	3.7	Sec



course	major	highest_grade	lowest_grade
36678	Sec	1.3	4.0
36689	Sec	1.0	4.7

separate



- Divide values in cells into multiple columns (and solve part of problem II)
- **Syntax:** `separate(df, col=column_to_split, into=vector_of_new_columns, sep=regex_for_place_to_split)`

Semester	course	major_gender	attendance
SoSe19	36689	info_m	20
SoSe19	36689	info_f	15
SoSe19	36689	sec_m	35
SoSe19	36689	sec_f	17



Semester	course	major	gender	attendance
SoSe19	36689	info	m	20
SoSe19	36689	Info	f	15
SoSe19	36689	Sec	m	35
SoSe19	36689	sec	f	17



- Divide values in cells into multiple columns (and solve part of problem II)
- **Syntax:** `separate(df, col=column_to_split, into=vector_of_new_columns, sep=regex_for_place_to_split)`

```
problem2_tidy <- problem2_df %>%
  pivot_longer(cols=sec_m:info_f, names_to="major_gender", values_to="attendance") %>%
  separate(col=major_gender, into=c("major", "gender"), sep="_")
```

semester	course	major_gender	attendance
SoSe19	36689	info_m	20
SoSe19	36689	info_f	15
SoSe19	36689	sec_m	35
SoSe19	36689	sec_f	17



semester	course	major	gender	attendance
SoSe19	36689	info	m	20
SoSe19	36689	Info	f	15
SoSe19	36689	Sec	m	35
SoSe19	36689	sec	f	17

What about problem IV (and V)?

- Both of them can be solved without the help of `tidyr`
- Problem IV
 - Split the dataframe into two (e.g. using `select`)
 - Use `unique` to ensure you don't have double entries of participants
- Problem V
 - Use `dplyr::bind_cols` to combine the dataframes into one

- **Central tendency** (e.g. mean, median, mode) and **spread** (IQR, variance, standard deviation) are ways to describe data and **condense** many data points into just one or two values
- R and R-Studio are helpful tools to work with and analyse data
- **Tidy data** is a convention whereby, if data is in this format, you can easily use it in many different functions. It means that each **variable** forms a column, each **observation** forms a row and each type of **observational unit** forms a table.
- Functionality in the **dplyr** and **tidyr** packages can help manipulate and wrangle datasets, so that they are ready to use in your further analyses.