Deskriptive Statistik

Maße der zentralen Tendenz und Streuung

Daniela Palleschi

2023-06-13

Inhaltsverzeichnis

Wiederholung

Letzte Woche haben wir...

- etwas über breite und lange Daten gelernt
- breite Daten länger gemacht
- lange Daten breiter gemacht

0.1 Überprüfung

```
pacman::p_load(tidyverse,
               here)
```

```
df_biondo <- read_csv(here("daten", "biondo_etal_2021_tidy.csv"))
df_billboard <- read_csv(here("daten", "billboard.csv"))
```

```
df_biondo %>%                                                    ①
  head(n = 5) %>%                                                ②
  knitr::kable() %>%                                             ③
  kableExtra::kable_styling(font_size = 20)                      ④
```

① Take the data frame `df_biondo`, and then
② take just the first 5 rows, and then
③ create a pretty `knitr` table, and then

subj	item	tense	verb	gramm	acc	rt	t
1	1	future	representarán	1	1	840.1917	159
1	2	future	alzarán	1	1	1310.1809	64
1	3	future	centrarán	1	1	700.2674	84
1	4	future	coleccionarán	1	1	650.1856	133
1	5	future	complementarán	1	1	580.2159	140

④ make the table even nicer using `kableExtra`, with font size 20

- we usually don't want to save the output from `head()`, `knitr::kable()` and `kableExtra::kable_styling()` as an object
  - and certainly not as an object that begins with `df_`, which stands for `dataframe`

0.2 Problem

Two examples of the same issue

```
df_biondo_long <- df_biondo %>%
  pivot_longer(
    cols = ("rt" | "tt"),
    names_to = "maß",
    values_to = "ms") %>%
  head(n = 10) %>%
  knitr::kable() %>%
  kableExtra::kable_styling()
```

```
df_biondo_long <- df_biondo %>%
  pivot_longer(
    cols = c(contains("rt"), contains("tt"))
  ) %>%
  knitr::kable() %>%
  kableExtra::kable_styling(font_size = 20)
```

subj	item	tense	verb	gramm	acc	maß	ms
1	1	future	representarán	1	1	rt	840.191
1	1	future	representarán	1	1	tt	1596.000
1	2	future	alzarán	1	1	rt	1310.180
1	2	future	alzarán	1	1	tt	648.000
1	3	future	centrarán	1	1	rt	700.267
1	3	future	centrarán	1	1	tt	841.000
1	4	future	coleccionarán	1	1	rt	650.185
1	4	future	coleccionarán	1	1	tt	1337.000
1	5	future	complementarán	1	1	rt	580.215
1	5	future	complementarán	1	1	tt	1400.000

0.3 Solution 1

Don't save a `knitr` table if you really mean to save a `dataframe` (i.e., `df_...`). Instead, save the `df` first, and in a different code chunk print the `df` as a formatted table.

```
# save longer dataframe
df_biondo_long <- df_biondo %>%
  pivot_longer(
    cols = ("rt" | "tt"),
    names_to = "maß",
    values_to = "ms")
```

```
# print table of longer df
df_biondo_long %>%
  head(n = 10) %>%
  knitr::kable() %>%
  kableExtra::kable_styling(font_size = 20)
```

0.4 Solution 2

Although `pivot_longer()` worked, the arguments for `cols =` weren't quite right. We want to use `c()` to list relevant columns here (not use a conditional). Also, the column names don't need to be in quotation marks because they are already known entities.

```
# save longer dataframe
df_biondo_long <- df_biondo %>%
  pivot_longer(
    cols = c(rt,tt),
    names_to = "maß",
    values_to = "ms")
```

0.5 Problem

Einrichtung:

```
df_billboard_tidy <- df_billboard %>%
  pivot_longer(
    cols = starts_with("wk"),
    names_to = "week",
    values_to = "rank",
    values_drop_na = TRUE
  ) %>%
  mutate(week = parse_number(week))
```

Warum wird mein Titel (Last Resort von Papa Roach) nicht gefunden?

```
df_billboard_tidy %>%
  select(contains("Resort"))
```

```
ggplot(data = df_billboard_tidy,
  aes(x = week, y = rank)) +
  labs(title = "'Last Resort' by Papa Roach",
       x = "Number of weeks", y = "Rank") +
  geom_density()
```

0.6 Solution 1

We want to `filter()` rows, not `select()` columns.

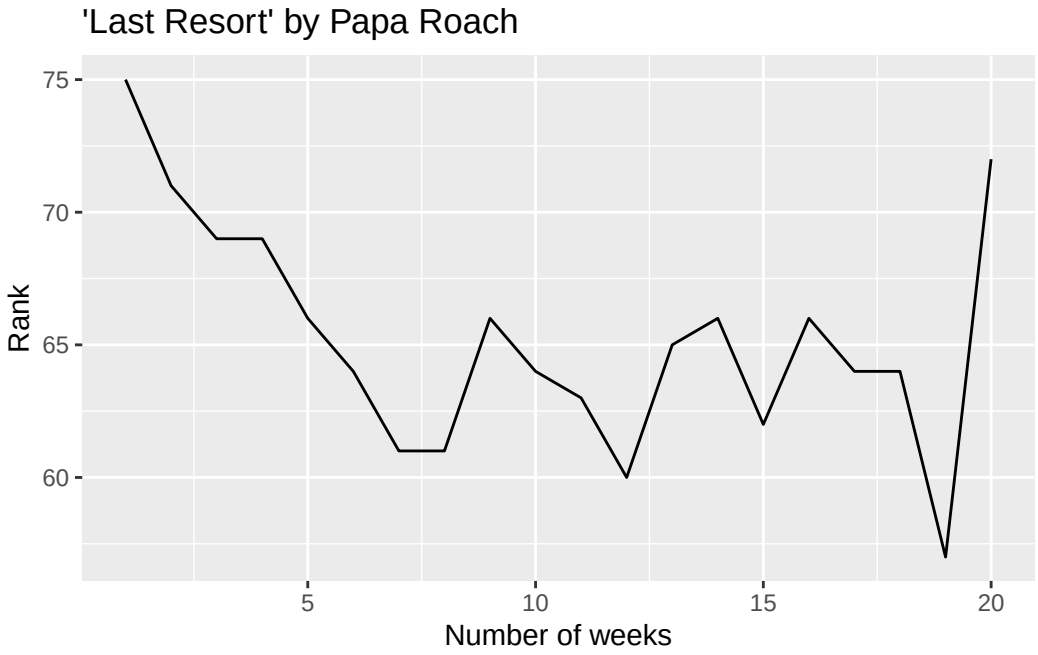
```
df_billboard_tidy %>%
  filter(track == "Last Resort") %>%
  head()
```

```
# A tibble: 6 x 5
  artist     track       date_entered  week  rank
  <chr>      <chr>       <date>       <dbl> <dbl>
1 Papa Roach Last Resort 2000-07-29       1    75
2 Papa Roach Last Resort 2000-07-29       2    71
3 Papa Roach Last Resort 2000-07-29       3    69
4 Papa Roach Last Resort 2000-07-29       4    69
5 Papa Roach Last Resort 2000-07-29       5    66
6 Papa Roach Last Resort 2000-07-29       6    64
```

0.7 Solution 2

`geom_density()` requires that there to be no `y` aesthetic (because this is always density). We want `geom_line()`.

```
df_billboard_tidy %>%
  filter(track == "Last Resort") %>%
ggplot(
  aes(x = week, y = rank)) +
  labs(title = "'Last Resort' by Papa Roach",
       x = "Number of weeks", y = "Rank") +
  geom_line()
```



Heutige Ziele

Today we will...

- (re-)learn about measures of central tendency
- (re-)learn about measures of dispersion
- learn how to use the `summarise()` function from `dplyr`
- learn how to produce summaries `.by` group

Lust auf mehr?

Ch.4, Section 4.5 (Groups) https://r4ds.hadley.nz/data-transform.html#groups Wickham et al. (o. J.)

1 Einrichtung

`Session > Restart R` to start with a fresh environment.

```
pacman::p_load(tidyverse,
               here)
```

```
df_flights <- read_csv(here("daten", "flights.csv"))
```

2 Deskriptive Statistik

- descriptive statistics describe the central tendency, variability, and distribution of the data
- sometimes called `summary` statistics, because it *summarises* the observed data

2.1 Anzahl der Werte (n)

- important information when summarising data
  - when we have more data (higher n), we have more confidence in the conclusions we draw from our data because we have more evidence
  - is also used to calculate some descriptive statistics

```
values <- c(3,1,2)
length(values)
```

```
[1] 3
```

`length()` versus `nrow()` and `n()`

- the function `length()` tells us how many (horizontal) values there are in an object
  - if that object is a data frame (instead of a vector like `values`), it tells us how many *columns* we have

```
length(df_flights)
```

```
[1] 19
```

- to count the number of values (i.e., observations/rows) in a data frame we can use

	air_time	distance
	Min. : 20.0	Min. : 17
	1st Qu.: 82.0	1st Qu.: 502
	Median :129.0	Median : 872
	Mean :150.7	Mean :1040
	3rd Qu.:192.0	3rd Qu.:1389
	Max. :695.0	Max. :4983
	NA's :9430	NA

  - `nrow()` (base R syntax), or
  - `n()` (`dplyr` syntax), we'll see this later

```
nrow(df_flights)
```

```
[1] 336776
```

2.2 Measures of central tendency

- pretty much what we get for *numeric* variables with the the `summary()` function

```
df_flights %>%
  select(air_time, distance) %>%
  summary() %>%
  knitr::kable() %>%
  kableExtra::kable_styling(font_size = 30)
```

2.2.1 Mean (μ)

- `mean` = Mittelwert, Durchschnitt
- the sum of all values divided by the number of values

$$\mu = \frac{Summe\ der\ Werte}{n}$$

- we can easily compute the mean by hand when we have only a few values

```
(3+1+2)/3
```

```
[1] 2
```

- we can also save the values as a vector (a list of values of the same class)
- and then use the function `mean()` to calculate their mean

```
values <- c(3,1,2)
mean(values)
```

```
[1] 2
```

- or we can run the `mean()` function on a variable in a data frame
  - using the `$` to indicate we want to select a column from a data frame

```
mean(df_flights$distance)
```

```
[1] 1039.913
```

- `df_flights$distance` is similar to `df_flights %>% select(distance)`

2.2.2 Median

- `median` = Median, mediane Wert; the value in the middle of the dataset
- if you line up all your values in ascending (or descending) order, the middle value is the median
  - e.g., if you have 5 values, the 3rd value is the median
  - if you have 6 values, the mean of the 3rd and 4th values are the median
- 50% of the data lie below this value, 50% above it

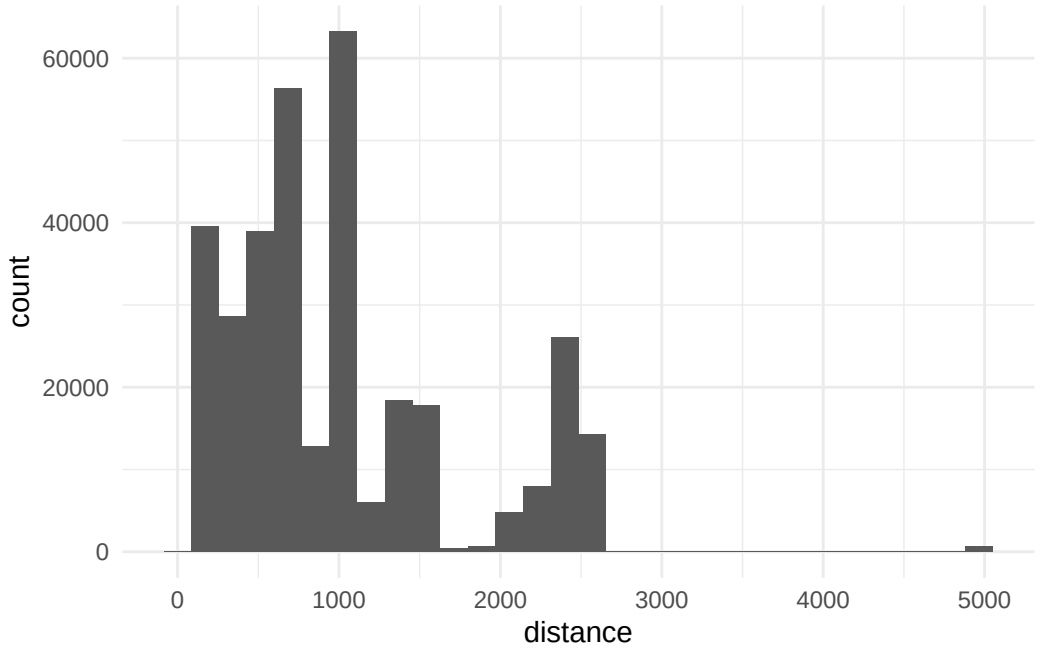
```r
median(df_flights$distance)
```

```
[1] 872
```

2.2.3 Mode

- `mode` = Modalwert; the value that occurs the most in a data set
- there's no R function to determine the `mode`, but we can visualise it with a histogram

```r
df_flights %>%
  ggplot(aes(x = distance)) +
  geom_histogram() +
  theme_minimal()
```



2.3 Measures of dispersion

- measures of central tendency describe the middle of the data (usually)
- measures of dispersion describe the spread of data points

2.3.1 Range

- `range` = Wertebereich
  - can refer to the highest and lowest values, or
  - the difference between highest and lowest value

- `max()` and `min()` print the highest and lowest values

```
max(values)
```

```
[1] 3
```

```
min(values)
```

```
[1] 1
```

- `range()` prints the lowest and highest values

```
range(values)
```

```
[1] 1 3
```

- we can calculate the difference between these values:

```
max(values) - min(values)
```

```
[1] 2
```

2.3.2 Standard deviation (`sd` or σ)

- a measure of how dispersed that data is *in relation to the mean*
  - low standard deviation means data are clustered around the mean (i.e., there is less spread)
  - high standard deviation means data are more spread out
- standard deviation is very often reported whenever mean is reported