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 Nehmen Sie den Datenrahmen `df_biondo`, und dann
② nimm nur die ersten 5 Zeilen, und dann
③ erstelle eine hübsche `knitr`-Tabelle, und dann

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 mache die Tabelle noch schöner mit `kableExtra`, mit Schriftgröße 20

- wir wollen normalerweise die Ausgabe von `head()`, `knitr::kable()` und `kableExtra::kable_styling()` nicht als Objekt speichern
  - und schon gar nicht als ein Objekt, das mit `df_` beginnt, was für `dataframe` steht

---

0.2 Problem

Zwei Beispiele für dasselbe Problem

```
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 Lösung 1

Speichern Sie keine `knitr`-Tabelle, wenn Sie wirklich einen `Datenrahmen` (d.h., `df_...`) speichern wollen. Speichern Sie stattdessen zuerst die `df`, und geben Sie die `df` in einem anderen Codeabschnitt als formatierte Tabelle aus.

```
# 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 Lösung 2

Obwohl `pivot_longer()` funktionierte, waren die Argumente für `cols =` nicht ganz richtig. Wir wollen hier `c()` verwenden, um die relevanten Spalten aufzulisten (und nicht eine Bedingung verwenden). Außerdem müssen die Spaltennamen nicht in Anführungszeichen gesetzt werden, da sie bereits bekannte Entitäten sind.

```
# 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 Lösung 1

Wir wollen Zeilen `filtern()`, nicht Spalten auswählen (i.e., `select()`).

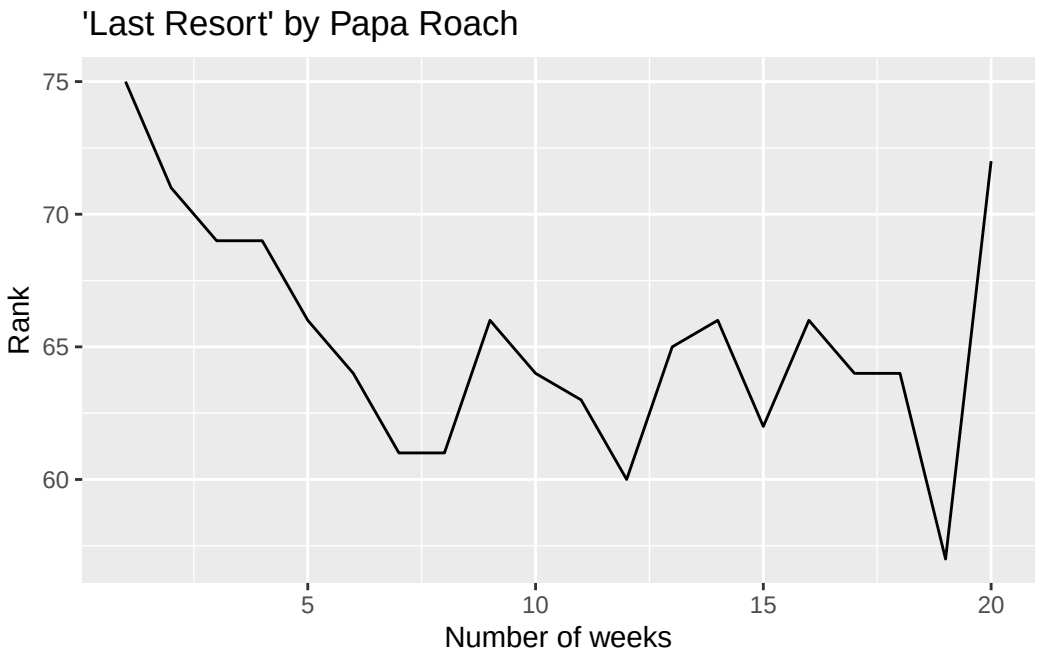
```
df_billboard_tidy %>%
  filter(track == "Last Resort") %>%
  head()
```

```
# 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 Lösung 2

Die Funktion `geom_density()` erfordert, dass es kein ästhetisches `y` gibt (weil dies immer die Dichte ist). Wir wollen `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

Heute werden wir...

- die Maße der zentralen Tendenz (wieder) kennenlernen
- Streuungsmaße (neu) kennenlernen
- lernen, wie man die Funktion `summarise()` von `dplyr` benutzt
- lernen, wie man Zusammenfassungen `.by` Gruppe erstellt

Lust auf mehr?

Ch.4, Section 4.5 Groups in Wickham et al. (o. J.)

1 Einrichtung

`Session > Restart R` um mit einer neuen Umgebung zu beginnen.

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

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

2 Deskriptive Statistik

- Die deskriptive Statistik beschreibt die zentrale Tendenz, die Variabilität und die Verteilung der Daten.
- manchmal auch "zusammenfassende" Statistik genannt, weil sie die beobachteten Daten *zusammenfasst*.

2.1 Anzahl der Werte (n)

- wichtige Informationen bei der Zusammenfassung von Daten
  - Wenn wir mehr Daten haben (höher n), haben wir mehr Vertrauen in die Schlussfolgerungen, die wir aus unseren Daten ziehen, weil wir mehr Beweise haben.
  - wird auch zur Berechnung einiger deskriptiver Statistiken verwendet

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

```
[1] 3
```

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

- die Funktion "Länge()" gibt an, wie viele (horizontale) Werte ein Objekt enthält
  - Wenn das Objekt ein Datenrahmen ist (statt eines Vektors wie "Werte"), sagt sie uns, wie viele *Spalten* wir haben.

```r
length(df_flights)
```

```
[1] 19
```

- Um die Anzahl der Werte (d.h. Beobachtungen/Zeilen) in einem Datenrahmen zu zählen, können wir verwenden

	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()` (Basis-R-Syntax), oder
  - `n()` (`dplyr`-Syntax), das werden wir später noch sehen

```
nrow(df_flights)
```

```
[1] 336776
```

2.2 Maße der zentralen Tendenz

- ziemlich genau das, was wir für *numerische* Variablen mit der Funktion `summary()` erhalten

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

2.2.1 Durchschnitt (μ)

- `mean` = Mittelwert, Durchschnitt
- die Summe aller Werte geteilt durch die Anzahl der Werte

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

- Wir können den Mittelwert leicht von Hand berechnen, wenn wir nur wenige Werte haben

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

```
[1] 2
```

- wir können die Werte auch als Vektor (eine Liste von Werten derselben Klasse) speichern
- und dann die Funktion `mean()` verwenden, um ihren Mittelwert zu berechnen

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

```
[1] 2
```

- oder wir können die Funktion `mean()` auf eine Variable in einem Datenrahmen anwenden
  - Verwendung des Zeichens `$`, um anzugeben, dass eine Spalte aus einem Datenrahmen ausgewählt werden soll

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

```
[1] 1039.913
```

- `df_flights$distance` ist vergleichbar mit `df_flights %>% select(distance)`

2.2.2 Median

- `median` = Median, mediane Wert; der Wert in der Mitte des Datensatzes
- Wenn Sie alle Ihre Werte in aufsteigender (oder absteigender) Reihenfolge aneinanderreihen, ist der mittlere Wert der Median
  - Wenn Sie z. B. 5 Werte haben, ist der 3. Wert der Median
  - bei 6 Werten ist der Mittelwert aus dem 3. und 4. Wert der Median
- 50% der Daten liegen unter diesem Wert, 50% darüber

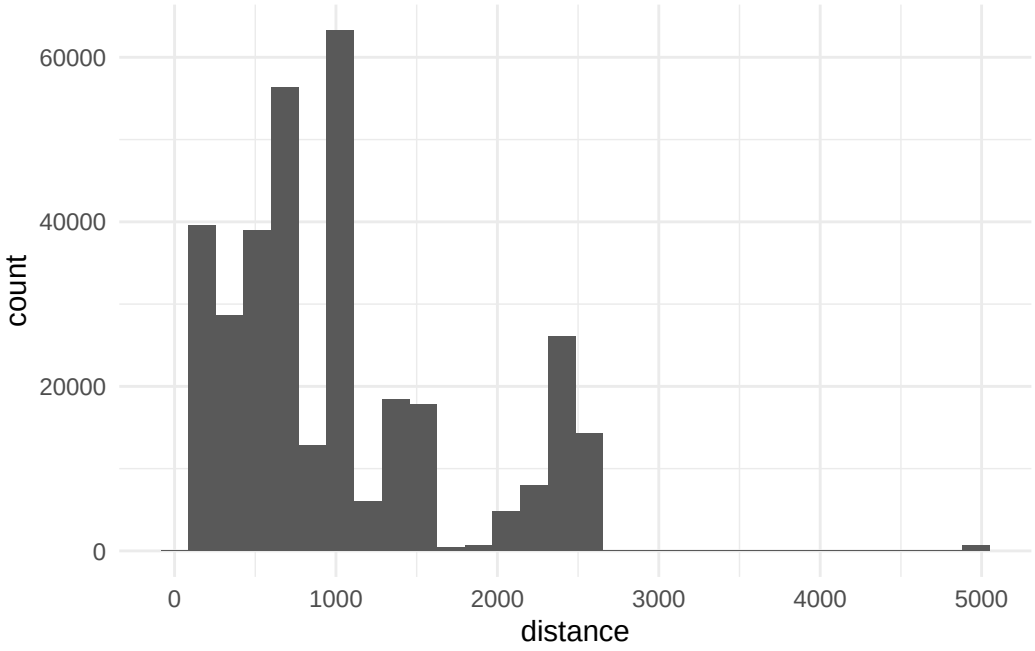
```
median(df_flights$distance)
```

```
[1] 872
```

2.2.3 Modalwert

- `mode` = Modalwert; der Wert, der am häufigsten in einem Datensatz vorkommt
- Es gibt keine R-Funktion zur Bestimmung des "Modus", aber wir können ihn mit einem Histogramm visualisieren

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



2.3 Maße der Streuung

- Maße der zentralen Tendenz beschreiben die Mitte der Daten (normalerweise)
- Streuungsmaße beschreiben die Verteilung der Datenpunkte

2.3.1 Wertebereich

- `range` = Wertebereich
  - kann sich auf den höchsten und den niedrigsten Wert beziehen, oder
  - die Differenz zwischen höchstem und niedrigstem Wert

- `max()` und `min()` den höchsten und den niedrigsten Wert ausdrucken

```
max(values)
```

```
[1] 3
```

```
min(values)
```

```
[1] 1
```

- `range()` druckt den niedrigsten und den höchsten Wert

```
range(values)
```

```
[1] 1 3
```

- können wir die Differenz zwischen diesen Werten berechnen:

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

```
[1] 2
```

2.3.2 Standardabweichung (`sd` or σ)

- ein Maß dafür, wie gestreut die Daten *im Verhältnis zum Mittelwert* sind
  - eine niedrige Standardabweichung bedeutet, dass die Daten um den Mittelwert herum gruppiert sind (d. h. es gibt eine geringere Streuung)
  - eine hohe Standardabweichung bedeutet, dass die Daten stärker gestreut sind
- Die Standardabweichung wird sehr oft angegeben, wenn der Mittelwert angegeben wird.
- um `sd` zu berechnen
  - die Quadratwurzel ($\sqrt{}$) der Summe der quadrierten Wertabweichungen vom Mittelwert ($(x-\mu)^2$) geteilt durch die Anzahl der Beobachtungen minus 1 ($n-1$)

```
sd(values)
```

```
[1] 1
```

- unsere Werte (x) sind:

```
values
```