Methoden quantitativer Forschung
Einstieg in die Datenanalyse mit R

FOM DVDW 03/2021 Virtuell 9 – WiSe 2021/22

Dipl.-Math. Norman Markgraf

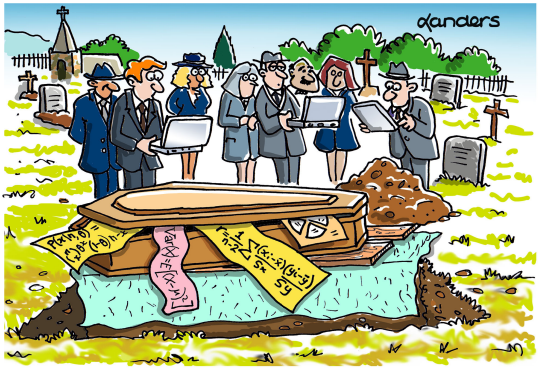
1. Crashkurs R

2. Tipps für die Darstellung mit `ggformula`

3. Datenhandling

1 Crashkurs R

Cartoon: Computereinsatz in der Statistik



„Auch wenn die Zeit für das ‚Einsetzen von Zahlen in Formeln' und das 'Abbildungen zeichnen per Hand' gekommen ist: die Ideen und Konzepte leben weiter - in unseren Computerprogrammen."[1]

[1] https://www.CAUSEweb.org/ © J. B. Landers, Überschrift K. Lübke

R Installation

1. R (https://www.r-project.org/)
2. RStudio Desktop (https://www.rstudio.com/)
3. Installation von Zusatzpaketen in RStudio:

```
install.packages("mosaic")
```

Ausführliche Installationsanleitung hier.

Code: mosaic

```
analysiere( y  # ggfs. abhängige Variable
            ~  x # unabhängige Variable(n)
            | z, # ggfs. bedingende (gruppierende) Variable(n)
            Optionen, # ggfs. weitere Optionen
            data = daten ) # Datensatz
```

`analysiere()`: Was soll R tun?[2]

Hinweis: unter macOS: `~`: `alt+n` oder `option+n`, `|`: `alt+7` oder `option+7`

[2] Befehlsübersicht hier

Zentrale Fragen

1. Was soll der Computer für mich tun?
2. Was muss der Computer dafür wissen?

```
meineanalyse(meiny ~ meinx, data = meinedaten)
```

R Basics

- R unterscheidet zwischen Groß- und Kleinbuchstaben.
- R verwendet den Punkt . als Dezimaltrennzeichen.
- Fehlende Werte werden in R durch `NA` kodiert.
- Kommentare werden mit dem Rautezeichen # eingeleitet.
- Eine Ergebniszuweisung erfolgt über `<-`.
- %>% (Paket `dplyr`) übergibt Ergebnisse.
- Hilfe zur Funktion `foo`: `?foo`

Der Aufbau einer CSV-Datei

Allgemeines Der Aufbau von CSV Dateien (**c**omma-**s**eparated **v**alues) ist sehr einfach. Ein allgemeiner Standard für das Dateiformat CSV existiert jedoch nicht. Aber im RFC 4180 grundlegend beschrieben. Die zu verwendende Zeichenkodierung ist ebenso nicht festgelegt; 7-Bit-ASCII-Code gilt weithin als der kleinste gemeinsame Nenner.

Dateiaufbau Innerhalb der Textdatei haben einige Zeichen eine Sonderfunktion zur Strukturierung der Daten.

- Ein Zeichen wird zur **Trennung von Datensätzen** benutzt. (Zumeist der Zeilenumbruch)
- Ein Zeichen wird zur **Trennung von Datenfeldern** (Spalten/Variabel) innerhalb der Datensätze benutzt. Allgemein wird dafür das **Komma** eingesetzt (`read.csv()`). In Deutschland eher das **Semikolon** (`read.csv2()`).
- Um Sonderzeichen innerhalb der Daten nutzen zu können (z. B. Komma in Dezimalzahlwerten), wird ein Feldbegrenzerzeichen (auch: Textbegrenzungszeichen) benutzt. Normalerweise ist dieser Feldbegrenzer das *Anführungszeichen* ". Wenn der Feldbegrenzer selbst in den Daten enthalten ist, wird dieser im Datenfeld verdoppelt (siehe Maskierungszeichen).

Der erste Datensatz kann ein Kopfdatensatz sein, der die Spaltennamen definiert.

Jeder Datensatz sollte laut RFC 4180, Absatz 2, Punkt 4 die gleiche Anzahl Spalten enthalten – dies wird aber nicht immer eingehalten.

CSV-Dateien einlesen (I/II)

In R stehen zwei Befehle für die beiden Varianten zur Verfügung:

Für den Fall, dass als Trennzeichen das Komma und ein Dezimalpunkt verwendet wurde:

```
read.csv
```

```
## function (file, header = TRUE, sep = ",", quote = "\"", dec = ".",
##     fill = TRUE, comment.char = "", ...)
## read.table(file = file, header = header, sep = sep, quote = quote,
##     dec = dec, fill = fill, comment.char = comment.char, ...)
## <bytecode: 0x7f90c7cb59e0>
## <environment: namespace:utils>
```

Für den Fall, dass als Trennzeichen das Semikolon und ein Dezimalkomma verwendet wurde:

```
read.csv2
```

```
## function (file, header = TRUE, sep = ";", quote = "\"", dec = ",",
##     fill = TRUE, comment.char = "", ...)
## read.table(file = file, header = header, sep = sep, quote = quote,
##     dec = dec, fill = fill, comment.char = comment.char, ...)
## <bytecode: 0x7f90c7cb2310>
## <environment: namespace:utils>
```

CSV-Dateien einlesen (II/II)

```
Teilnehmerliste_Workshop <- read.csv2(
    "~/Dropbox/FOM/R-Workshop-2021/Teilnehmendenliste.csv",
    header = FALSE  # Keine Spaltenüberschrift! -> V1, V2, V3, V4 !
)
head(Teilnehmerliste_Workshop, 5)
```

```
##                                   V1
## 1              1,Frau,Abelein,Anna
## 2            2,Herr,Augustin,Frank
## 3              3,Frau,Bausch,Sonja
## 4 4,Herr,Bernedo Schneider,Gordon
## 5               5,Herr,Dill,Arthur
```

Excel-Dateien einlesen (I/II)

```
library(readxl)  # Paket zum Einlesen von Excel-Dateien
Datei <- "~/Dropbox/FOM/R-Workshop-2021/Teilnehmendenliste.xlsx"
Teilnehmerliste_Workshop <- read_excel(Datei)
head(Teilnehmerliste_Workshop, 4)
```

```
## # A tibble: 4 x 4
##   Teilnehmendenliste ...2  ...3             ...4
##                <dbl> <chr> <chr>            <chr>
## 1                  1 Frau  Abelein          Anna
## 2                  2 Herr  Augustin         Frank
## 3                  3 Frau  Bausch           Sonja
## 4                  4 Herr  Bernedo Schneider Gordon
```

Der Befehl `read_excel()`:

```
read_excel(path, sheet = NULL, range = NULL, col_names = TRUE,
  col_types = NULL, na = "", trim_ws = TRUE, skip = 0,
  n_max = Inf, guess_max = min(1000, n_max),
  progress = readxl_progress(), .name_repair = "unique")
```

Excel-Dateien einlesen (II/II)

```
Datei <- "~/Dropbox/FOM/R-Workshop-2021/Teilnehmendenliste.xlsx"
Teilnehmerliste_Workshop_Demo <- read_excel(
  Datei,
  sheet = "Demo",
  col_names = c("Anrede", "Nachnamen", "Vornamen"),
  col_types = c("skip", "guess", "text", "text", "guess"))

head(Teilnehmerliste_Workshop_Demo, 4)
```

```
## # A tibble: 4 x 3
##   Anrede Nachnamen Vornamen
##   <chr>  <chr>     <chr>
## 1 <NA>   <NA>      <NA>
## 2 Frau   Abelein   Anna
## 3 Herr   Augustin  Frank
## 4 Frau   Bausch    Sonja
```

Vorbereitung Analyse

Dick De Veaux: How much is a Fireplace Worth?[3]

- `Preis`: Preis in $.
- `Wohnflaeche`: Wohnfläche in m^2.
- `Alter`: Alter der Immobilie in Jahren.
- `Klimaanlage`: Inwieweit eine (zentrale) Klimaanlage vorhanden ist.
- `Kamin`: Inwieweit ein Kamin vorhanden ist.
- `Heizung`: Heizsystem: Gas, Strom oder Öl.

```
# Paket laden
library(mosaic)

# URL
#daten_url <- "http://statistix.org/Data/SaratogaHouses.csv"

# Daten einlesen
#Houses <- read.csv2(daten_url)
Houses <- read.csv2(here::here("SaratogaHouses.csv"))
```

[3] Siehe auch: `?mosaicData::SaratogaHouses`

Datenvorverarbeitung

Häufig müssen Daten vor der eigentlichen Analyse vorverarbeitet werden, z. B.:

- Variablen auswählen: `select()`
- Beobachtungen auswählen: `filter()`
- Variablen verändern, neu erzeugen: `mutate()`
- Beobachtungen zusammenfassen: `summarise()`
- ...

Das Paket `dplyr`[4] bietet dazu viele Möglichkeiten.

Umfangreiche Dokumentation: http://dplyr.tidyverse.org/index.html

[4] wird mit `mosaic` installiert und geladen.

Inspizieren

```
inspect(Houses)
```

```
##
## categorical variables:
##          name     class levels    n missing
## 1 Klimaanlage character      2 1728       0
## 2       Kamin character      2 1728       0
## 3     Heizung character      3 1728       0
##                                   distribution
## 1 Nein (63.3%), Ja (36.7%)
## 2 Ja (57.2%), Nein (42.8%)
## 3 Gas (69.3%), Strom (18.2%) ...
##
## quantitative variables:
##             name   class       min          Q1      median          Q3
## ...1       Preis integer 5000.0000 145000.0000 189900.0000 259000.0000
## ...2 Wohnflaeche numeric   57.2278    120.7729    151.8488    198.6018
## ...3       Alter integer    0.0000     13.0000     19.0000     34.0000
##               max         mean          sd    n missing
## ...1 775000.0000 211966.70544 98441.39102 1728       0
## ...2    485.6931    163.04122    57.59342 1728       0
## ...3    225.0000     27.91609    29.20999 1728       0
```

Übung 1: Skalenniveau

Welches Skalenniveau hat die Variable `Heizung`?

A. Kategorial - nominal.
B. Kategorial - ordinal.
C. Numerisch - intervallskaliert.
D. Numerisch - verhältnisskaliert.

Übung 1: Skalenniveau

Welches Skalenniveau hat die Variable `Beizug`?

A. Kategorial - nominal.
B. Kategorial - ordinal.
C. Numerisch - intervallskaliert.
D. Numerisch - verhältnisskaliert.

Nominal mit drei Ausprägungen, also ***A***.

Eine Variable

meinx: kategorial

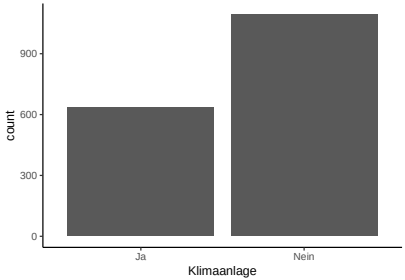
```
# Säulendiagramm
gf_bar( ~ Klimaanlage,
          data = Houses)
# Tabelle
tally( ~ Klimaanlage,
       data = Houses)
# Anteil
prop( ~ Klimaanlage,
      data = Houses,
      success = "Ja")
```

meinx: numerisch

```
# Histogramm
gf_histogram( ~ Preis,
           data = Houses)
# Kennzahlen
favstats( ~ Preis,
          data = Houses)
# Mittelwert
mean( ~ Preis,
      data = Houses)
```

Eine kategoriale Variable

```
# Säulendiagramm
gf_bar( ~ Klimaanlage, data = Houses)
```




```
# Tabelle
tally( ~ Klimaanlage, data = Houses)
```

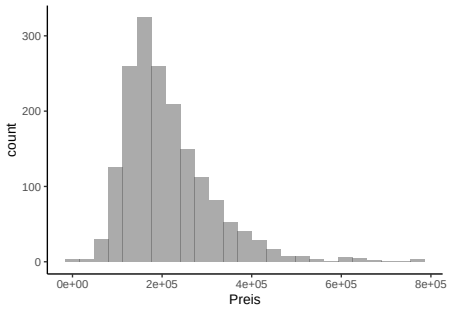
```
## Klimaanlage
##   Ja Nein
##  635 1093
```

```
# Anteil
prop( ~ Klimaanlage, success = "Ja", data = Houses)
```

```
##   prop_Ja
## 0.3674769
```

Eine numerische Variable

```
# Histogramm
gf_histogram( ~ Preis, data = Houses)
```




```
# Kennzahlen
favstats( ~ Preis, data = Houses)
```

```
##    min     Q1 median     Q3    max     mean       sd    n missing
##   5000 145000 189900 259000 775000 211966.7 98441.39 1728       0
```

```
# Mittelwert
mean( ~ Preis, data = Houses)
```

```
## [1] 211966.7
```

Zwei Variablen

meinx, meiny: kategorial

```
# Mosaikplot
mosaicplot(Kamin ~ Klimaanlage,
           data = Houses)
# Kreuztabelle
tally(Kamin ~ Klimaanlage,
      data = Houses)
# Chi-Quadrat Test
xchisq.test(Kamin ~ Klimaanlage,
            data = Houses)
```

meinx, meiny: metrisch

```
# Streudiagramm
gf_point(Preis ~ Wohnflaeche,
       data = Houses)
# Korrelation
cor(Preis ~ Wohnflaeche,
    data = Houses)
# Korrelationstest
cor.test(Preis ~ Wohnflaeche,
         data = Houses)
```

Übung 2: Nullhypothese

Wie lautet beim Chi-Quadrat Unabhängigkeitstest die Nullhypothese?

A. Die beiden Variablen sind *abhängig*. Die Verteilung der einen Variable *hängt vom Wert* der anderen Variable ab.
B. Die beiden Variablen sind ***un**abhängig*. Die Verteilung der einen Variable *hängt **nicht** vom Wert* der anderen Variable ab.

Übung 2: Nullhypothese

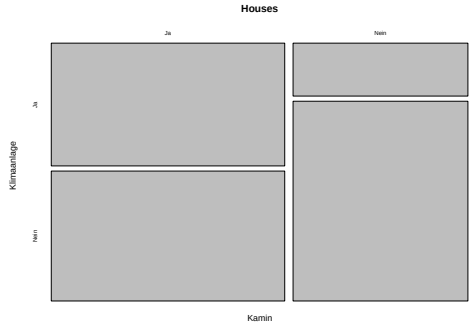
Wie lautet beim Chi-Quadrat Unabhängigkeitstest die Nullhypothese?

A. Die beiden Variablen sind abhängig. Die Verteilung der einen Variable *hängt vom Wert* der anderen Variable ab.
B. Die beiden Variablen sind **un**abhängig. Die Verteilung der einen Variable *hängt* ***nicht*** *vom Wert* der anderen Variable ab.

B: Die Nullhypothese H_0 lautet, es gibt keinen Zusammenhang, die Alternativhypothese H_A ist das Gegenteil, es gibt einen Zusammenhang.

Zwei kategoriale Variablen (I/II)

```
# Mosaikplot
mosaicplot(Kamin ~ Klimaanlage, data = Houses)
```




```
# Kreuztabelle
tally(Kamin ~ Klimaanlage, data = Houses)
```

```
##       Klimaanlage
## Kamin   Ja Nein
##   Ja   480  508
##   Nein 155  585
```

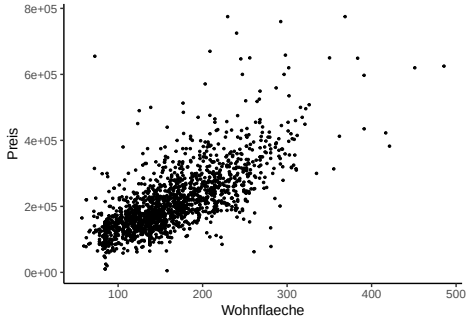
Zwei kategoriale Variablen (II/II)

```r
# Chi-Quadrat Test
xchisq.test(Kamin ~ Klimaanlage, data = Houses)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  x
## X-squared = 137.85, df = 1, p-value < 2.2e-16
##
##   480      508
## (363.07) (624.93)
## [37.34]  [21.69]
## < 6.14>  <-4.68>
##
##   155      585
## (271.93) (468.07)
## [49.85]  [28.96]
## <-7.09>  < 5.40>
##
## key:
##  observed
##  (expected)
##  [contribution to X-squared]
##  <Pearson residual>
```

Zwei numerische Variablen (I/II)

```
# Streudiagramm
gf_point(Preis ~ Wohnflaeche, data = Houses)
```




```
# Korrelation
cor(Preis ~ Wohnflaeche, data = Houses)
```

```
## [1] 0.7123902
```

Zwei numerische Variablen (II/II)

```
# Korrelationstest
cor.test(Preis ~ Wohnflaeche, data = Houses)
```

```
##
##  Pearson's product-moment correlation
##
## data:  Preis and Wohnflaeche
## t = 42.173, df = 1726, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6883589 0.7348595
## sample estimates:
##       cor
## 0.7123902
```

Zwei Gruppen

meinx: binär, meiny: kategorial

```
# Säulendiagramm
gf_bar( ~ Kamin | Klimaanlage,
          data = Houses)
# Anteile
prop(Kamin ~ Klimaanlage,
     data = Houses,
     success = "Ja")
# Anteilstest
prop.test(Kamin ~ Klimaanlage,
     data = Houses,
     success = "Ja")
```

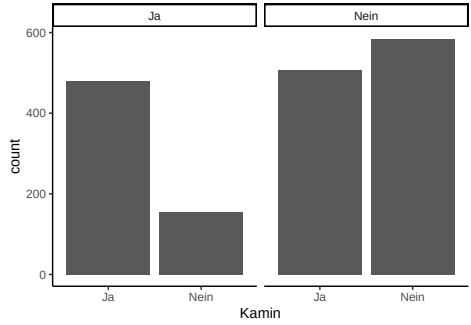
meinx: binär, meiny: numerisch

```
# Histogramm
gf_histogram( ~ Preis | Kamin,
           data = Houses)
# Mittelwerte
mean(Preis ~ Kamin,
     data = Houses)

# t-Test
t.test(Preis ~ Kamin,
       data = Houses)
```

Zwei Gruppen, kategorial (I/II)

```
# Säulendiagramm
gf_bar( ~ Kamin | Klimaanlage, data = Houses)
```




```
# Anteile
prop(Kamin ~ Klimaanlage, data = Houses, success = "Ja")
```

```
##   prop_Ja.Ja prop_Ja.Nein
##    0.7559055    0.4647758
```

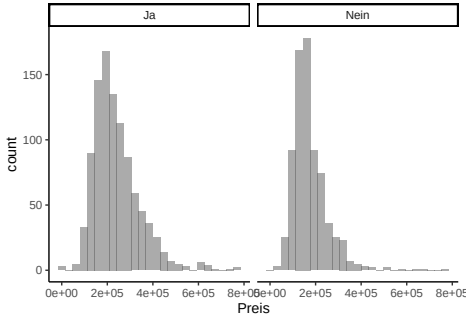
Zwei Gruppen, kategorial (II/II)

```
# Anteilstest
prop.test(Kamin ~ Klimaanlage, data = Houses, success = "Ja")
```

```
##
##  2-sample test for equality of proportions with continuity
##  correction
##
## data:  tally(Kamin ~ Klimaanlage)
## X-squared = 137.85, df = 1, p-value < 2.2e-16
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.2452697 0.3369896
## sample estimates:
##    prop 1    prop 2
## 0.7559055 0.4647758
```

Zwei Gruppen, numerisch (I/II)

```
# Histogramm
gf_histogram( ~ Preis | Kamin, data = Houses)
```




```
# Mittelwerte
mean(Preis ~ Kamin, data = Houses)
```

```
##       Ja     Nein
## 239914.0 174653.4
```

Zwei Gruppen, numerisch (II/II)

```
# t-Test
t.test(Preis ~ Kamin, data = Houses)
```

```
##
##  Welch Two Sample t-test
##
## data:  Preis by Kamin
## t = 14.971, df = 1724.7, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group Ja and
## 95 percent confidence interval:
##  56710.60 73810.61
## sample estimates:
##   mean in group Ja mean in group Nein
##           239914.0           174653.4
```

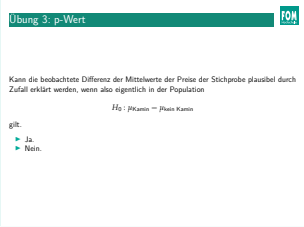
Übung 3: p-Wert

Kann die beobachtete Differenz der Mittelwerte der Preise der Stichprobe plausibel durch Zufall erklärt werden, wenn also eigentlich in der Population

$$H_0 : \mu_{\text{Kamin}} = \mu_{\text{kein Kamin}}$$

gilt.

- Ja.
- Nein.



Nein: Der p-Wert (`p-value`) ist sehr klein, d. h. die beobachtete Differenz tritt innerhalb der normalen Variation sehr selten auf. Daher würde man hier H_0 verwerfen ($\alpha = 5\%$).

Schema Hypothesentest

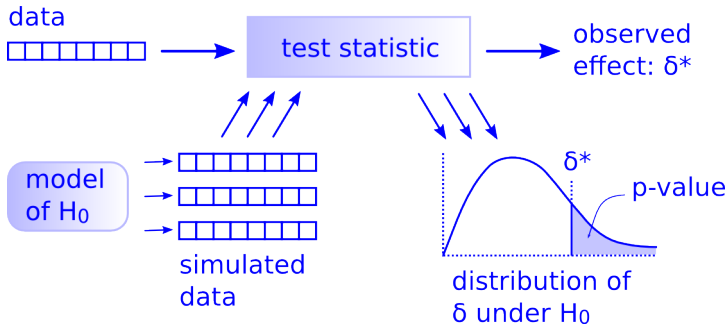


Abbildung: Quelle: Blogbeitrag Allen Downey[5]

Alternative: Verwende theoretische Verteilungsannahmen unter H_0, häufig approximativ oder asymptotisch.[6]

[5] http://allendowney.blogspot.de/2016/06/there-is-still-only-one-test.html
[6] Bspw. $t-, \chi^2-, F-$ Verteilungen.

Ablauf: Permutations- und Randomisationstest

Vorraussetzung: Zufällige Stichprobe (Permutation) oder zufällige Zuordnung (Randomisation).

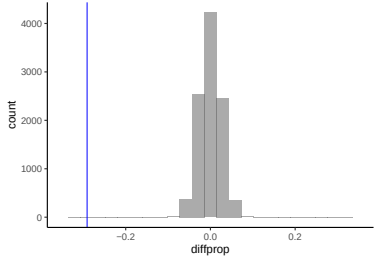
Beispiel: Zwei-Stichproben-Fall:

- Wiederhole z. B. 10000×
  - Mische die $n_1 + n_2$ Beobachtungen.
  - Ordne zufällig n_1 Beobachtungen der ersten Stichprobe zu, die restlichen der zweiten.
  - Berechne die Differenz der Mittelwerte $\bar{x}_1 - \bar{x}_2$. Analog für andere Teststatistiken, z. B. Anteil.
- Zeichne Histogramm der Verteilung der Teststatistik des Modells unter $H_0 : \mu_1 - \mu_2 = 0$. Vergleiche mit dem beobachteten Wert der Teststatistik (der Stichprobe).
- Der p-Wert ist der Anteil der zufälligen Teststatistiken, die mindestens so groß sind wie der beobachtete Wert.[7]

[7] Bei ungerichteten, zweiseitigen Tests im Absolutbetrag.

Permutationstest kategorial

```
# Reproduzierbarkeit
set.seed(2009)
# Anteilsdifferenz in Stichprobe
pdiff_est <- diffprop(Kamin ~ Klimaanlage, success = "Ja", data = Houses)
# Simuliere H_0: Permutiere Klima
Nullvtlg <- do(10000) *
  diffprop(Kamin ~ shuffle(Klimaanlage), success = "Ja", data = Houses)
# Histogramm Nullverteilung
gf_histogram( ~ diffprop, data = Nullvtlg) %>%
  gf_vline(xintercept = ~ pdiff_est, color = "blue") + xlim(-0.35, 0.35)
```

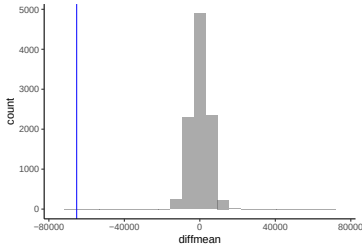



```
# p-Wert
prop( ~(abs(diffprop)>=abs(pdiff_est)), data = Nullvtlg )
```

```
## prop_TRUE
##         0
```

Permutationstest numerisch

```
# Reproduzierbarkeit
set.seed(2009)
# Mittelwertdifferenz in Stichprobe
meandiff_est <- diffmean(Preis ~ Kamin, data = Houses)
# Simuliere H_0: Permutiere Klima
Nullvtlg <- do(10000) *
  diffmean(Preis ~ shuffle(Kamin), data = Houses)
# Histogramm Nullverteilung
gf_histogram( ~ diffmean, data = Nullvtlg) %>%
  gf_vline(xintercept = ~ meandiff_est, color = "blue") +
  xlim(-75000, 75000)
```




```
# p-Wert
prop( ~(abs(diffmean)>=abs(meandiff_est)), data = Nullvtlg)
```

```
## prop_TRUE
##         0
```

Schema Bootstrap

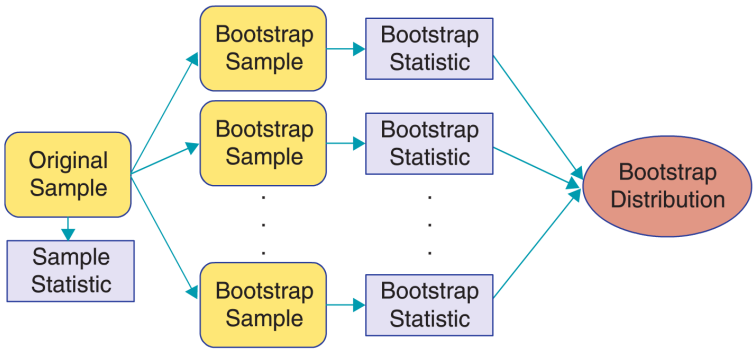


Abbildung: Quelle: Lock, Robin, Patti Frazer Lock, Kari Lock Morgan, Eric F. Lock, and Dennis F. Lock (2012): Statistics: UnLOCKing the Power of Data. Wiley.

Ablauf: Bootstrap

Vorraussetzungen:

- Zufällige Stichprobe oder zufällige Zuordnung.
- Nicht zu kleine Stichprobe.[8]

Beispiel: Bootstrap-Perzentil-Intervall[9] für eine Stichprobe:

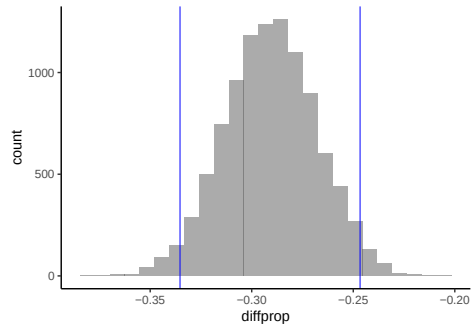
- Wiederhole z. B. 10000×
  - Ziehe mit Zurücklegen eine Stichprobe vom Umfang n aus der Originalstichprobe.
  - Berechne Statistik, z. B. Mittelwert $\bar{x}$ der Bootstrap-Stichprobe. Analog für andere Statistiken, z. B. Anteil.
- Zeichne Histogramm der Bootstrap-Verteilung der Statistik.
- Das 95 %-Bootstrap-Perzentil-Intervall sind die mittleren 95 % der Bootstrap-Verteilung.

[8] $n \geqslant 35$

[9] Es gibt weitere, teilweise exaktere Bootstrap-Methoden.

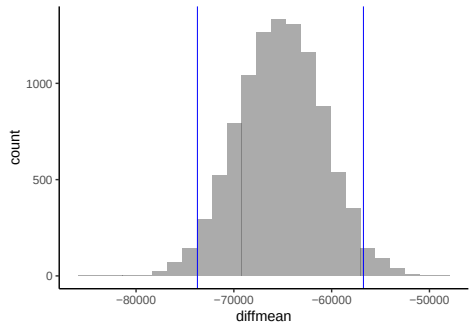
Bootstrap kategorial

```
# Reproduzierbarkeit
set.seed(2009)
# Simuliere Stichprobenziehung
Bootvtlg <- do(10000) *
  diffprop(Kamin ~ Klimaanlage, success = "Ja", data = resample(Houses))
# 95% Konfidenzintervall
ci <- quantile( ~ diffprop, probs=c(0.025, 0.975), data = Bootvtlg)
# Histogramm
gf_histogram( ~ diffprop, data = Bootvtlg) %>%
  gf_vline(xintercept = ci, color = "blue")
```



Bootstrap numerisch

```
# Reproduzierbarkeit
set.seed(2009)
# Simuliere Stichprobenziehung
Bootvtlg <- do(10000) *
  diffmean(Preis ~ Kamin, data = resample(Houses))
# 95% Konfidenzintervall
ci <- quantile( ~ diffmean, probs=c(0.025, 0.975), data = Bootvtlg)
# Histogramm
gf_histogram( ~ diffmean, data = Bootvtlg) %>%
  gf_vline(xintercept = ci, color = "blue")
```



Simulation und Resampling

- **Permutationstest**, hier: simuliere zufällige Zuordnung[10]. Simuliere Verteilung einer Statistik unter der Annahme, dass kein Zusammenhang vorliegt (Modell H_0), u. a. zur Bestimmung von p-Werten.

```
statistik(y ~ shuffle(x), data = Daten)
```

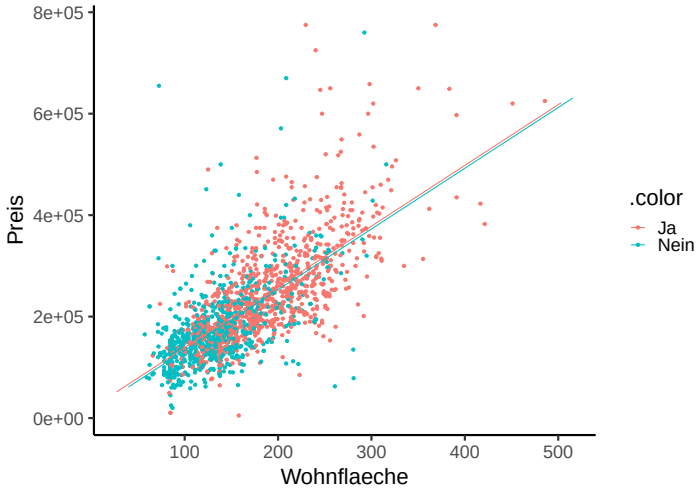
- **Bootstrap**, hier: simuliere zufälliges Ziehen einer Stichprobe[11]. Schätze Verteilung einer Statistik der Stichprobe, u. a. zur Bestimmung von Konfidenzintervallen oder Standardfehlern.

```
statistik(y ~ x, data = resample(Daten))
```

[10] d. h. ohne Zurücklegen
[11] d. h. mit Zurücklegen

Lineares Modell (I/III)

```
modnum <- lm(Preis ~ Wohnflaeche + Kamin, data = Houses)
plotModel(modnum)
```



Lineares Modell (II/III)

```
summary(modnum)
```

```
##
## Call:
## lm(formula = Preis ~ Wohnflaeche + Kamin, data = Houses)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -271421  -39935   -7887   28215  554651
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19166.54    6286.94   3.049  0.00233 **
## Wohnflaeche  1197.15      31.94  37.476  < 2e-16 ***
## KaminNein   -5567.38    3716.95  -1.498  0.13436
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 69080 on 1725 degrees of freedom
## Multiple R-squared:  0.5081, Adjusted R-squared:  0.5076
## F-statistic:   891 on 2 and 1725 DF,  p-value: < 2.2e-16
```

Lineares Modell (III/III)

```
anova(modnum)
```

```
## Analysis of Variance Table
##
## Response: Preis
##                Df     Sum Sq    Mean Sq   F value Pr(>F)
## Wohnflaeche     1 8.4934e+12 8.4934e+12 1779.8488 <2e-16 ***
## Kamin           1 1.0706e+10 1.0706e+10    2.2435 0.1344
## Residuals    1725 8.2317e+12 4.7720e+09
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Übung 4: Kovariablen

Gegeben die Wohnfläche: Ist dann der (marginale) Effekt des Kamins in einem linearen Modell auf den Preis *signifikant* ($\alpha = 5\%$)?

- Ja.
- Nein.

Übung 4: Kovariablen

Gegeben die Wohnfläche: Ist dann der (marginale) Effekt des Kamins in einem linearen Modell auf den Preis *signifikant* ($\alpha = 5\%$)?

- Ja.
- Nein.

Nein: Der p-Wert (`Pr(>|t|)`) bzw. (`Pr(>F)`) für $H_0 : \beta_{\text{Kamin}} = 0$ ist größer als $\alpha = 5\%$.

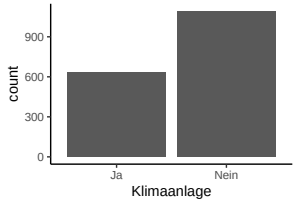
Schlussmöglichkeiten

	zufällige Zuordnung	**keine zufällige Zuordnung**
zufällige Stichprobe	Kausalschluss, generalisierbar für die Population	kein Kausalschluss, Aussage generalisierbar für die Population
keine zufällige Stichprobe	Kausalschluss, nur für die Stichprobe	kein Kausalschluss, Aussage nur für die Stichprobe

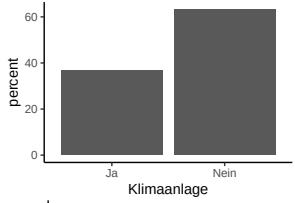
2 Tipps für die Darstellung mit `ggformula`

Säulendiagramme

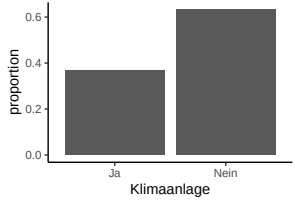
```
# Säulendiagramm
gf_bar( ~ Klimaanlage,
        data = Houses)
```




```
# Säulendiagramm mit Prozentangabe
gf_percents( ~ Klimaanlage,
             data = Houses)
```

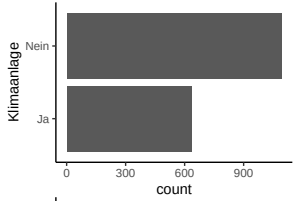



```
# Säulendiagramm mit Anteilswerten
gf_props( ~ Klimaanlage,
          data = Houses)
```

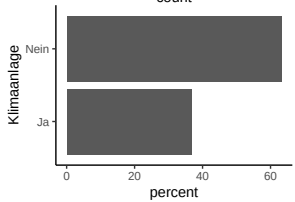


Balkendiagramme

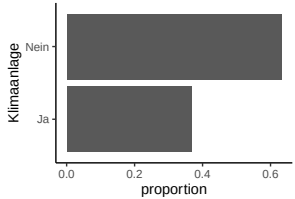
```
# Balkendiagramm
gf_countsh( ~ Klimaanlage,
            data = Houses)
```




```
# Balkendiagramm mit Prozentangabe
gf_percentsh( ~ Klimaanlage,
              data = Houses)
```

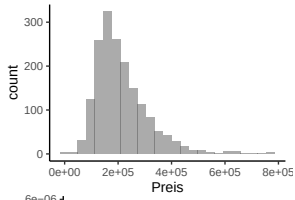



```
# Balkendiagramm mit Anteilswerten
gf_propsh( ~ Klimaanlage,
           data = Houses)
```

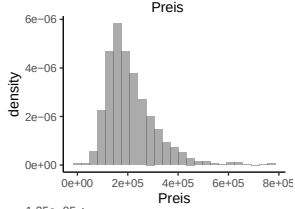


Histogramme

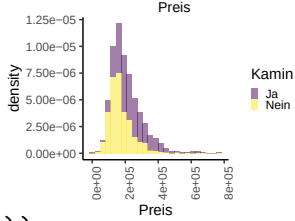
```
# Histogramm (Anzahl)
gf_histogram( ~ Preis,
              data = Houses)
```




```
# Histogramm (Dichte)
gf_dhistogram( ~ Preis,
               data = Houses)
```

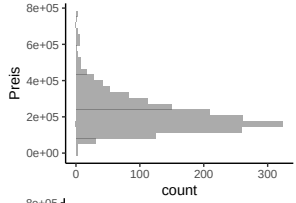



```
# Histogramm (Dichte), gruppiert
gf_dhistogram( ~ Preis,
               fill = ~ Kamin,
               data = Houses) +
    scale_fill_viridis_d() +
    theme(axis.text.x = element_text(angle = 90))
```

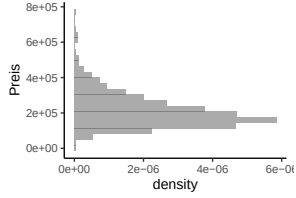


Horizontale Histogramme

```
# Histogramm (Anzahl)
gf_histogramh( ~ Preis,
           data = Houses)
```




```
# Histogramm (Dichte)
gf_dhistogramh( ~ Preis,
                data = Houses)
```

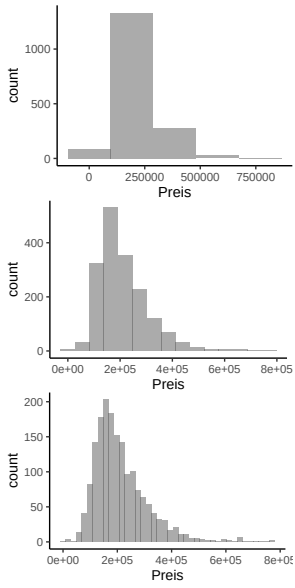


Histogramme mit der Option bins=

```
# Histogramm mit 5 Rechtecken
gf_histogram( ~ Preis,
              bins = 5,
              data = Houses)
```

```
# Histogramm mit 15 Rechtecken
gf_histogram( ~ Preis,
              bins = 15,
              data = Houses)
```

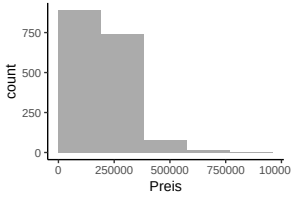
```
# Histogramm mit 40 Rechtecken
gf_histogram( ~ Preis,
              bins = 40,
              data = Houses)
```



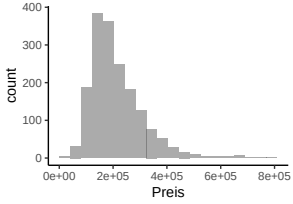
Histogramme mit der Option binwidth= und center=

```
h <- diff(range( ~ Preis, data=Houses)) # Gesamtbreite bestimmen
```

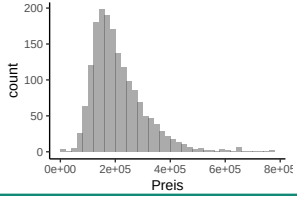
```
# Histogramm mit 1/4 der Gesamtbreite
# pro Korb
gf_histogram( ~ Preis,
              binwidth = h/4,
              center = h/8,
              data = Houses)
```




```
# Histogramm mit 1/19 der Gesamtbreite
# pro Korb
gf_histogram( ~ Preis,
              binwidth = h/19,
              center=h/38,
              data = Houses)
```

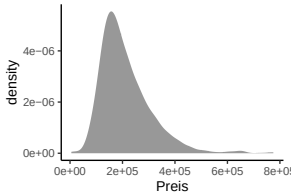



```
# Histogramm mit Korbbreite 20000
gf_histogram( ~ Preis,
              binwidth = 20000,
              center = 10000,
              data = Houses)
```

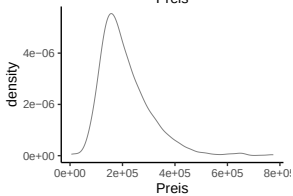


Dichte(schätzer)

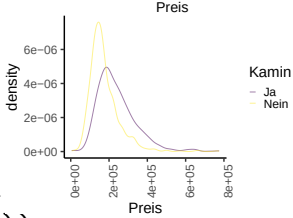
```
# Kerndichteschätzung
gf_density( ~ Preis,
            data = Houses)
```




```
# Kerndichteschätzung
gf_dens( ~ Preis,
         data = Houses)
```



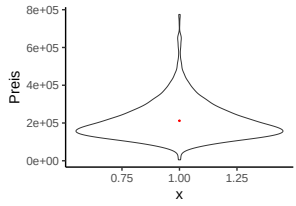

```
# Kerndichteschätzung, gruppiert
# mit Farbe
gf_dens( ~ Preis,
         color = ~ Kamin,
         data = Houses) + scale_color_viridis_d
    theme(axis.text.x = element_text(angle = 90))
```

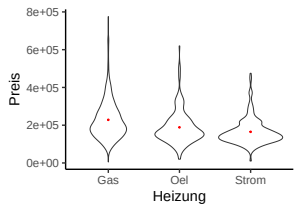


Violinen-Plots

```
# Violine mit Mittelwert
gf_violin( Preis ~ 1,
           data = Houses) %>%
    gf_point(Preis ~ 1, stat="summary",
             fun.y="mean", color="red",
             data= Houses)
```

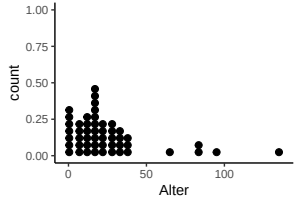
```
# Violinen mit Mittelwerten
gf_violin( Preis ~ Heizung,
           data = Houses) %>%
    gf_point(Preis ~ Heizung,
             stat="summary",
             fun.y="mean", color="red",
             data= Houses)
```



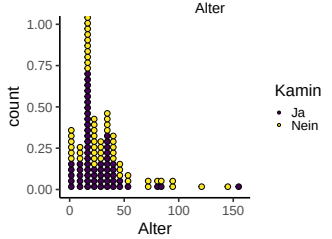


Dot-Plots

```
# Dotplot
gf_dotplot( Alter ~ 1,
            data = sample(Houses, 50))
```

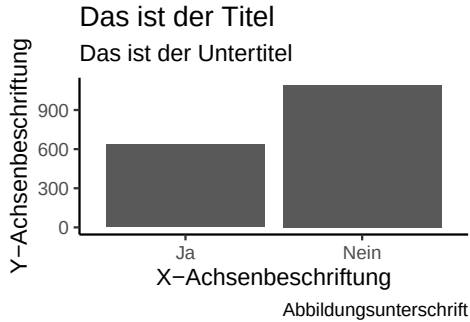



```
library(viridis)
# Dotplot zweier Gruppen
gf_dotplot(  ~ Alter,
            fill = ~ Kamin,
            stackgroups = TRUE,
            binpositions="all",
            data = sample(Houses, 120)
        ) + scale_fill_viridis_d()
```



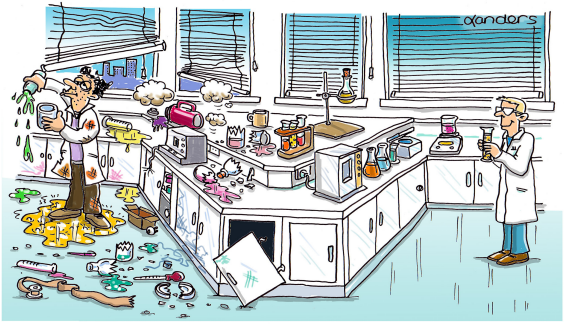
Überschrift, Untertitel, Y- & X-Achsenbezeichner und Abbildungsunterschrift

```
# Säulendiagramm
gf_bar( ~ Klimaanlage,
        title = "Das ist der Titel",
        subtitle = "Das ist der Untertitel",
        xlab = "X-Achsenbeschriftung",
        ylab = "Y-Achsenbeschriftung",
        caption = "Abbildungsunterschrift",
        data = Houses)
```



3 Datenhandling

Cartoon: Datenbereinigung



„Vergiss nicht, Deine schmutzigen Daten aufzuräumen.“[12]

[12] https://www.CAUSEweb.org/ © J. B. Landers, Überschrift J. A. Morrow

Datenvorverarbeitung

Häufig müssen Daten vor der eigentlichen Analyse vorverarbeitet werden, z. B.:

- Variablen auswählen: `select()`
- Beobachtungen auswählen: `filter()`
- Variablen verändern, neu erzeugen: `mutate()`
- Beobachtungen zusammenfassen: `summarise()`
- ...

Das Paket `dplyr`[13] bietet dazu viele Möglichkeiten.

Umfangreiche Dokumentation: http://dplyr.tidyverse.org/index.html

[13] Wird mit `mosaic` installiert und geladen

Vorbereitung

Dick De Veaux: How much is a Fireplace Worth?[14]

- `Preis`: Preis in $.
- `Wohnflaeche`: Wohnfläche in m^2.
- `Alter`: Alter der Immobilie in Jahren.
- `Klimaanlage`: Inwieweit eine (zentrale) Klimaanlage vorhanden ist.
- `Kamin`: Inwieweit ein Kamin vorhanden ist.
- `Heizung`: Heizsystem: Gas, Strom oder Öl.

```
# Paket laden
library(mosaic)

# URL
daten_url <- "http://statistix.org/Data/SaratogaHouses.csv"

# Daten einlesen
Houses <- read.csv2(daten_url)
```

[14] Siehe auch: `?mosaicData::SaratogaHouses`

Variablen auswählen: `select()`

```
Houses %>%
  select(Preis, Heizung) %>%
  inspect()
```

```
##
## categorical variables:
##      name     class levels    n missing
## 1 Heizung character      3 1728       0
##                                   distribution
## 1 Gas (69.3%), Strom (18.2%) ...
##
## quantitative variables:
##       name   class  min     Q1 median     Q3    max     mean       sd
## ...1 Preis integer 5000 145000 189900 259000 775000 211966.7 98441.39
##         n missing
## ...1 1728       0
```

Vergleichen

- gleich, ($=$): ==
- ungleich ($\neq$): !=
- kleiner, kleiner gleich ($<, \leqslant$): <, <=
- größer, größer gleich ($>, \geqslant$): >, >=

```
4 == 5
```

```
## [1] FALSE
```

```
4 != 5
```

```
## [1] TRUE
```

```
4 <= 5
```

```
## [1] TRUE
```

```
4 > 5
```

```
## [1] FALSE
```

Beobachtungen auswählen: `filter()`

Häuser mit Gas-Heizung, aber nicht zu Teuer (Preis < 100.000)

```
Houses %>%
  filter(Heizung=="Gas" & Preis < 100000) %>%
  inspect()
```

```
##
## categorical variables:
##          name     class levels  n missing
## 1 Klimaanlage character      2 50       0
## 2       Kamin character      2 50       0
## 3     Heizung character      1 50       0
##                                  distribution
## 1 Nein (82%), Ja (18%)
## 2 Nein (78%), Ja (22%)
## 3 Gas (100%)
##
## quantitative variables:
##             name   class        min          Q1      median         Q3
## ...1       Preis integer 5000.00000 75125.00000 86092.50000 90075.0000
## ...2 Wohnflaeche numeric   66.88963    85.00557    98.05834   130.0864
## ...3       Alter integer    0.00000    14.25000    43.50000    83.0000
##              max       mean          sd  n missing
## ...1 98500.0000 81592.7800 16286.91144 50       0
## ...2   223.0583   108.2943    28.98784 50       0
## ...3   225.0000    55.6800    53.06309 50       0
```

Offene Übung 5: Datensatz auswählen

Erzeugen Sie einen Datensatz `HousesOelBig`, der nur die Variable `Preis` enthält, und zwar für die Häuser, die eine Oelheizungen haben und deren Wohnfläche größer als 100qm ist.

Offene Übung 5: Datensatz auswählen

Offene Übung 5: Datensatz auswählen

Erzeugen Sie einen Datensatz `HousesOelBig`, der nur die Variable `Preis` enthält, und zwar für die Häuser, die eine Oelheizungen haben und deren Wohnfläche größer als 100qm ist.

```
Houses %>% filter(Heizung == „Oel“ & Wohnflaeche > 100) %>% select(Preis) ->
HousesOelBig
```

Variablen verändern/erzeugen: `mutate()`

```
Houses %>%
  mutate(qmPreis = Preis / Wohnflaeche) %>%
  select(qmPreis) %>%
  inspect()
```

```
##
## quantitative variables:
##         name   class      min       Q1   median       Q3  max     mean
## ...1 qmPreis numeric 31.65882 1047.757 1259.297 1510.444 9039 1318.974
##            sd    n missing
## ...1 467.2638 1728       0
```

Übung 6: Billiger Wohnraum

Wie viele Beobachtungen haben einen Quadratmeter Preis unter 1000 ?

A. 1728
B. 350
C. 1378

Übung 6: Billiger Wohnraum

Wie viele Beobachtungen haben einen Quadratmeter Preis unter 1000 ?

A. 1728
B. 350
C. 1378

B:

```
Houses %>% mutate(qmPreis = Preis / Wohnflaeche) %>% filter(qmPreis < 1000) %>% nrow()
```

Variablen verändern: `case_when()`

```
Houses %>%
  mutate(Groesse = case_when(Wohnflaeche <= 90 ~ "klein",
                             Wohnflaeche <= 150 ~ "mittel",
                             Wohnflaeche > 150 ~ "groß")) %>%
  select(Groesse) %>%
  table()
```

```
## .
##   groß  klein mittel
##    885    150    693
```

Hinweis: Anstelle der letzten Abfrage (`Wohnflaeche > 150`) hätte auch einfach `TRUE` verwendet werden können.

Übung 7: Variablen erzeugen

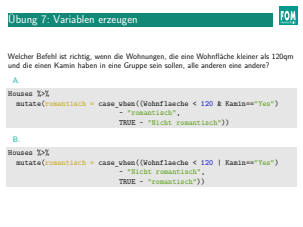
Welcher Befehl ist richtig, wenn die Wohnungen, die eine Wohnfläche kleiner als 120qm und die einen Kamin haben in eine Gruppe sein sollen, alle anderen eine andere?

A.

```
Houses %>%
  mutate(romantisch = case_when((Wohnflaeche < 120 & Kamin=="Yes")
                                ~ "romantisch",
                                TRUE ~ "Nicht romantisch"))
```

B.

```
Houses %>%
  mutate(romantisch = case_when((Wohnflaeche < 120 | Kamin=="Yes")
                                ~ "Nicht romantisch",
                                TRUE ~ "romantisch"))
```



`Wohnfläche < 120 & Kamin=="Yes""` sind Wohnungen kleiner 120qm UND mit Kamin, also ***A***.

Daten zusammenfassen: summarise()

```
Houses %>%
  summarise(Durchschnittspreis=mean(Preis), n=n())
```

```
##   Durchschnittspreis    n
## 1           211966.7 1728
```

Nach Gruppen zusammenfassen: group_by()

```
Houses %>%
  group_by(Heizung, Kamin) %>%
  summarise(Durchschnittspreis=mean(Preis), n=n())
```

```
## # A tibble: 6 x 4
## # Groups:   Heizung [3]
##   Heizung Kamin Durchschnittspreis     n
##   <chr>   <chr>              <dbl> <int>
## 1 Gas     Ja               251298.   767
## 2 Gas     Nein             187933.   430
## 3 Oel     Ja               215461.    92
## 4 Oel     Nein             168905.   124
## 5 Strom   Ja               189668.   129
## 6 Strom   Nein             147786.   186
```

Übung 8: Datenvorverarbeitung

Mit welchem Befehl können Beobachtungen mit bestimmten Eigenschaften ausgewählt werden?

A. `select()`
B. `filter()`
C. `mutate()`
D. `summarise()`

Mit welchem Befehl können Beobachtungen mit bestimmten Eigenschaften ausgewählt werden?

A. `select()`
B. `filter()`
C. `mutate()`
D. `summarise()`

`select` wählt Variablen aus, `mutate` verändert sie, `summarise` fasst sie zusammen. Beobachtungen werden daher mit `filter` (***B***) zusammengefasst.

Die obersten *n* Beobachtungen: `top_n()`

```
Houses %>%
  group_by(Kamin) %>%
  top_n(n=3, Preis) %>%
  arrange(-Preis)
```

```
## # A tibble: 6 x 6
## # Groups:   Kamin [2]
##    Preis Wohnflaeche Alter Klimaanlage Kamin Heizung
##    <int>       <dbl> <int> <chr>       <chr> <chr>
## 1 775000       230.      5 Ja          Ja    Gas
## 2 775000       369.     31 Ja          Ja    Gas
## 3 760000       292.      2 Ja          Nein  Gas
## 4 725000       240.      3 Ja          Ja    Gas
## 5 670000       209.    121 Nein        Nein  Gas
## 6 655000        72.5    55 Nein        Nein  Gas
```

Hinweis: Auf diese Art und Weise können auch Datensätze balanciert werden.[15]

[15] Vgl. geschichtete Stichprobe: `group_by() %>% sample_n()`

Spalten sortieren: `arrange()`

```
Houses %>%
  group_by(Kamin) %>%
  top_n(n=3, Preis) %>%
  arrange(Preis)
```

```
## # A tibble: 6 x 6
## # Groups:   Kamin [2]
##    Preis Wohnflaeche Alter Klimaanlage Kamin Heizung
##    <int>       <dbl> <int> <chr>       <chr> <chr>
## 1 655000        72.5    55 Nein        Nein  Gas
## 2 670000       209.    121 Nein        Nein  Gas
## 3 725000       240.      3 Ja          Ja    Gas
## 4 760000       292.      2 Ja          Nein  Gas
## 5 775000       230.      5 Ja          Ja    Gas
## 6 775000       369.     31 Ja          Ja    Gas
```

Das Problem mit „falsch“ formatierten Tabellen

```
Tabelle_A <- read.csv2("../datasets/TabelleA.csv") %>% select(-X)
Tabelle_B <- read.csv2("../datasets/TabelleB.csv") %>% select(-X)
names(Tabelle_A) <- c("Land", "1999", "2000")
names(Tabelle_B) <- c("Land", "1999", "2000")
```

Tabelle_A: (Anzahl an Vorfällen im Land im jeweiligen Jahr)

```
Tabelle_A
```

```
##          Land   1999   2000
## 1 Afghanistan    745   2666
## 2      Brazil  37737  80488
## 3       China 212258 213766
```

Tabelle_B: (Einwohner im Land im jeweiligen Jahr)

```
Tabelle_B
```

```
##          Land       1999       2000
## 1 Afghanistan   19987071   20595360
## 2      Brazil  172006362  174504898
## 3       China 1272915272 1280428583
```

Diese beiden Tabellen sind nicht in „Normalform“, also jedes Zeile eine Beobachtung, jede Spalte eine Merkmal.

Umformen mit `pivot_longer()`

```
library(tidyr)
Tabelle_A <- Tabelle_A %>%
  pivot_longer(-Land, names_to="Jahr", values_to="Vorfälle")
Tabelle_B <- Tabelle_B %>%
  pivot_longer(-Land, names_to="Jahr", values_to="Einwohner")
```

Tabelle_A: (Anzahl an Vorfällen im Land im jeweiligen Jahr)

```
Tabelle_A
```

```
## # A tibble: 6 x 3
##   Land        Jahr  Vorfälle
##   <chr>       <chr>    <int>
## 1 Afghanistan 1999       745
## 2 Afghanistan 2000      2666
## 3 Brazil      1999     37737
## 4 Brazil      2000     80488
## 5 China       1999    212258
## 6 China       2000    213766
```

Tabelle_B: (Einwohner im Land im jeweiligen Jahr)

```
Tabelle_B
```

```
## # A tibble: 6 x 3
##   Land        Jahr    Einwohner
##   <chr>       <chr>       <int>
## 1 Afghanistan 1999     19987071
## 2 Afghanistan 2000     20595360
## 3 Brazil      1999    172006362
## 4 Brazil      2000    174504898
## 5 China       1999   1272915272
## 6 China       2000   1280428583
```

Zusammenfassen in eine Tabelle (inner_join())

```
Tabelle <- inner_join(Tabelle_A, Tabelle_B, by= c("Land", "Jahr"))
Tabelle
```

```
## # A tibble: 6 x 4
##   Land        Jahr  Vorfälle  Einwohner
##   <chr>       <chr>    <int>      <int>
## 1 Afghanistan 1999       745   19987071
## 2 Afghanistan 2000      2666   20595360
## 3 Brazil      1999     37737  172006362
## 4 Brazil      2000     80488  174504898
## 5 China       1999    212258 1272915272
## 6 China       2000    213766 1280428583
```

pivot_wider(), für die die es brauchen

```
Tabelle %>%
  pivot_wider(names_from="Jahr", values_from=c(Einwohner, Vorfälle))
```

```
## # A tibble: 3 x 5
##   Land        Einwohner_1999 Einwohner_2000 Vorfälle_1999 Vorfälle_2000
##   <chr>                <int>          <int>         <int>         <int>
## 1 Afghanistan       19987071       20595360           745          2666
## 2 Brazil           172006362      174504898         37737         80488
## 3 China           1272915272     1280428583        212258        213766
```

Anhang

Lizenz / Version

- Datum erstellt: 2021-12-06
- R Version: 4.1.2
- `mosaic` Version: 1.8.3

Bitte melden Sie Fehler und Verbesserungsvorschläge: nmarkgraf@hotmail.com

- Autor*innen: Oliver Gansser, Matthias Gehrke, Tanja Kistler, Bianca Krol, Karsten Lübke, Norman Markgraf, Sebastian Sauer, Tabea Griesenbeck
- Hinweise u. a. von Thomas Christiaans, Jörg Horst, Ute Twisselmann, Martin Vogt, Daniel Ziggel. **Vielen Dank!**