R-Kurs: Tag 2

Lukas Mödl, Matthias Becher, Erin Sprünken

Institut für Biometrie und Klinische Epidemiologie

Charité - Universitätsmedizin Berlin, Berlin

erin-dirk.spruenken@charite.de

July 20, 2022

CHARITÉ UNIVERSITÄTSMEDIZIN BERLIN

iBikE

Laden

- `load()`
- `read.table()`
- `read.csv()`

Optionen bei read.csv()

Wenn man CSVs in R lädt kann man verschiedene Parameter einstellen, um der Funktion zu sagen wie die CSV formatiert ist. Die wichtigsten werden hier vorgestellt:

- `header(TRUE/FALSE)`: Zeigt an, ob in der CSV Spaltennamen in der ersten Reihe stehen
- `sep`: Welches Zeichen wird verwendet um Spalten zu trennen. Default ist ",". Es werden aber auch häufig ";" oder "\t" verwendet
- `dec`: Welches Zeichen wird bei Dezimalzahlen verwendet "." oder ","
- Beispiel: `read.csv("data.csv", header=TRUE, sep=";", dec=",")`

Indizierung

Häufig möchte man nur bestimmte Elemente eines Vektors, einer Liste oder eines Data Frames auswählen. Um das zu tun gibt es mehrere Möglichkeiten. Die direkteste ist es, die Indizes zu verwenden. Angenommen wir haben den Vektor `x <- c(1, 2, 3, 4, 5)`

- Einen bestimmten Wert auswählen $\Leftrightarrow$ `x[1]`
- Mehrere Werte auswählen $\Leftrightarrow$ `x[c(1, 3, 5)]`
- Eine Reihe von Werten auswählen $\Leftrightarrow$ `x[1:3]`
- Einen bestimmten Wert weglassen $\Leftrightarrow$ `x[-1]`

Indizierung von Listen und Data Frames

Liste

- `x[1]`
- `x[[1]]`

Data Frame

- `x[1,]`
- `x[,1]`
- `x[,"Spalte1" ]`
- `x$Spalte1`

Summary()

```
    Numeric       Factor  Character            Logical            Date
Min.   :  1.00   a:25    Length:100         Mode :logical   Min.   :2022-01-01
1st Qu.: 25.75   b:25    Class :character   FALSE:50        1st Qu.:2022-01-25
Median : 50.50   c:25    Mode  :character   TRUE :50        Median :2022-02-19
Mean   : 50.50   d:25                                       Mean   :2022-02-19
3rd Qu.: 75.25                                              3rd Qu.:2022-03-16
Max.   :100.00                                              Max.   :2022-04-10
```

Funktionen für die Deskriptive Statistik

- Mean = `mean()`
- Median = `median()`
- Minimum = `min()`
- Maximum = `max()`
- Standard Deviation = `sd()`
- Variance = `var()`
- Quantile = `quantile()`
- Correlation = `cor()`
- Covariance = `cov()`
- Crosstable = `table()`

Anmerkung: sd() und var() teilen bei der Berechnung durch `n-1`

Umgang mit NAs

Bei der Berechnung der verschiedenen Statistiken kann es zu Problemen kommen, wenn NAs in den Daten vorhanden sind.

- Beispiel: `mean(c(1,2,3,4,5,NA))` gibt als Ausgaben `NA`

Das lässt sich leicht lösen, indem man der Funktion den Parameter `rm.na = TRUE` mitgibt.

- Beispiel: `mean(c(1,2,3,4,5,NA), na.rm = TRUE)` gibt als Ausgabe `3`

Mit der Funktion `is.na()` kann man testen ob sich in einem Vektor, data frame etc. NAs befinden

Konvertieren von Daten

- Numeric $\Leftrightarrow$ `as.numeric()`
- Character $\Leftrightarrow$ `as.character()`
- Factor $\Leftrightarrow$ `as.factor()`
- Date $\Leftrightarrow$ `as.Date()`
- Logical $\Leftrightarrow$ `as.logical()`

Filtern

Häufig kommt es vor, dass wir unsere Daten filtern möchten um bespielsweise nur die Männer bzw. Frauen zu untersuchen oder nur Patienten ab einem bestimmten Alter zu betrachten. In R gibt es verschiedene Befehle mit denen man das erreichen kann.

- `==`
  - `data[data$Sex == "W",]`
- `%in%`
  - `data[data$Color %in% c("blue", "red"),]`
- `subset()`
  - `subset(data, Age < 50)`

Logische Operatoren

Manchmal möchte man nach mehreren Spalten gleichzeitig filtern. Anstatt das nacheinaner zu tun, kann man auch mehrere Filter mit logischen Operatoren verbinden, die hier einmal vorgestellt werden:

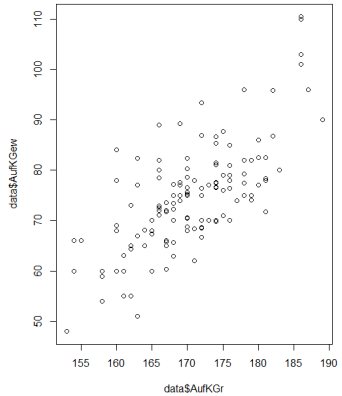
- Und = `&`
- Oder = `|`
- Nicht = `!`

Beispiel:

- `data[data$Sex != "M" & data$Age > 50,]`

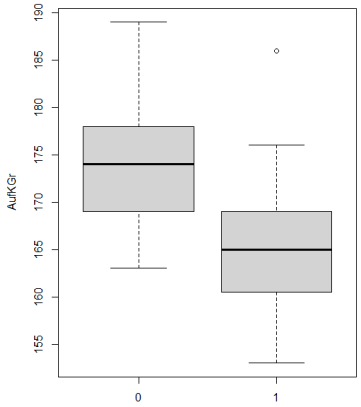
Scatterplot

- `plot(data$Height, data$Weight)`



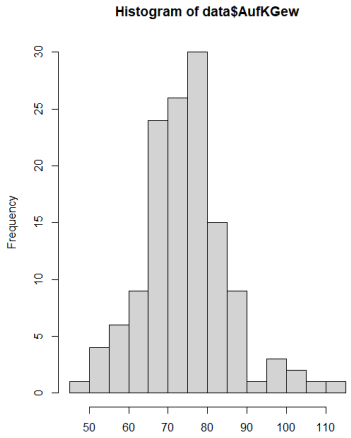
Boxplot

- `boxplot(ata$Sex == 0,]$Height, data[data$Sex == 1,]$Heigh)`



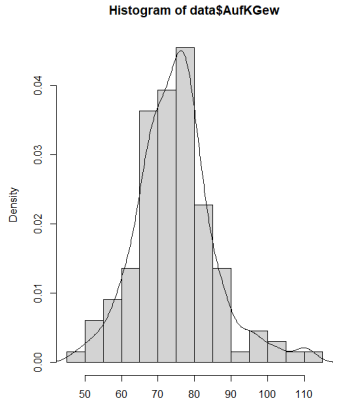
Histogram

- `hist(data$Weight)`



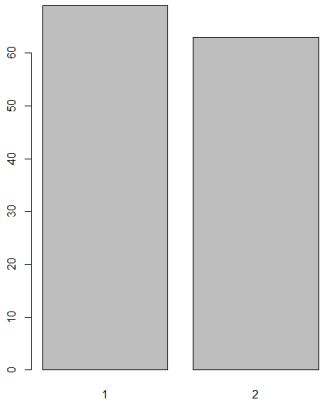
Histogram mit Density

- `hist(data$Weight, probability = T)`
  `lines(density(data$Weight))`



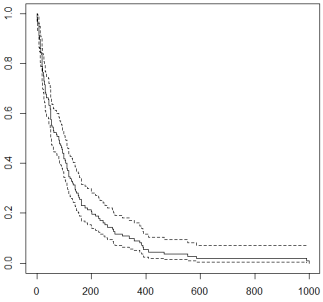
Barplot

- `barplot(summary(as.factor(data$Klinik)))`



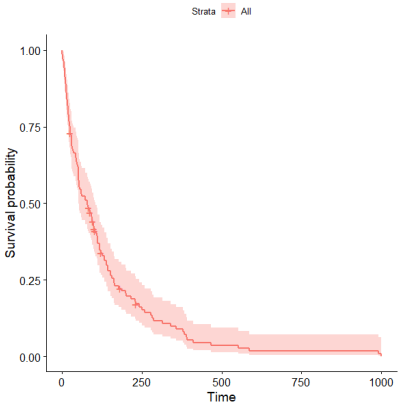
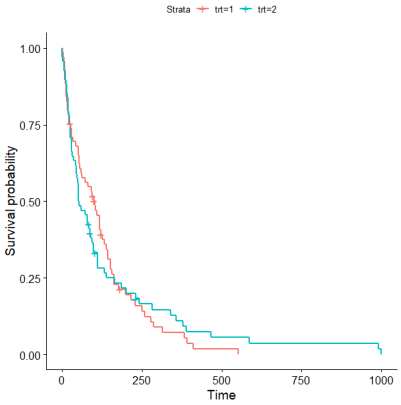
Kaplan-Meier Plot

```
library(survival)
data_vet <- veteran
km_fit <- survfit(Surv(time, status) ~ 1, data=data_vet)
plot(km_fit)
```



Kaplan-Meier Plot

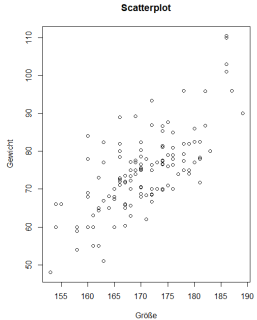
- ```
  library(survminer)
  ggsurvplot(km_fit)
  ```


  ```

Kaplan-Meier Plot

- ```
  km_fit <- survfit(Surv(time, status) ~ trt, data=data_vet)
  ggsurvplot(km_fit)
  ```


  ```

Beschriftungen für Plots

Es gibt verschiedene Möglichkeiten Plots anzupassen. Unter anderem kann man Achsenbeschriftungen oder einen Titel einfügen.

- Beispiel:
```
plot(data$Height, data$Weight, xlab="Größe,"
     ylab="Gewicht", main="Scatterplot")
```



Speichern von Plots