

Tutorium Session 3

[Code ▾](#)

10-12-2020

Übung 3

Aufgabe 1: - Erstellen Sie einen tibble mit folgenden Spalten in R

[Hide](#)

```
## Erstmal die Vektoren erstellen

VPNumber<- c(1:100)

sex<- sample(c("m", "f"), size = 100, replace = TRUE) ## sample(c(Vektor mit den Werten, die zufällig sein
sollen)

                                ## size = wie viele Elemente?, replace = mit zurückle
gen (T = Ja)

age<- sample(c(18:30), size=100, replace = T)

verbal_score<- sample(c(1:9), size=100, T)

logic_score<- sample(c(1:9), 100, T)

## erstmal tidyverse laden!
library(tidyverse)

## dann tibble erstellen

IQ_data<- tibble(VPNumber, sex, age, verbal_score, logic_score)
```

Das Ganze geht natürlich auch in einem Befehl:

[Hide](#)

```
Iq_data_in_Kurz<- tibble(VPNumber_short = (c(1:100)),
                        sex_short = sample(c("f", "m"), 100, T),
                        age_short = sample(c(18:30), size = 100, replace = T),
                        verbal_score_short = sample(c(1:9), size = 100, T),
                        logic_score_short = sample(c(1:9),100,T))
```

Nochmal kurz Überblick verschaffen:

[Hide](#)

```
summary(Iq_data_in_Kurz)
```

Aufgabe 2: Fügen Sie folgende zusätzlichen Vektoren hinzu

[Hide](#)

```
## Erstmal Mittelwert von beiden
means <- (IQ_data$verbal_score+IQ_data$logic_score)/2

## Zum Datensatz hinzufügen
IQ_data$stanine_IQ<- means

## oder gleich
IQ_data$stanine_IQ<- (IQ_data$verbal_score+IQ_data$logic_score)/2

##### zweite Variante #####

IQ_data <- IQ_data %>%
  mutate(stanine_IQ = (verbal_score+ logic_score)/2)
```

Von stanine_IQ in IQ

[Hide](#)

```

z_score<- scale(IQ_data$stanine_IQ)

IQ<- z_score * 15 + 100

IQ_data$IQ<- IQ

##### oder tidyverse-Variante ###

IQ_data_tidy <- Iq_data_in_Kurz %>%
  mutate(stanine_IQ = (verbal_score_short + logic_score_short)/2,
         IQ = (stanine_IQ - 5)/1.96,
         IQ = round(IQ * 15 + 100))

```

Aufgabe 3:

Logische Indizierung

Hide

```

IQ_data_extended<-IQ_data %>%
  mutate(lower_IQ = IQ <= 85,
         high_IQ = IQ >116 & IQ <130,
         mid_IQ = !lower_IQ & !high_IQ)

## Ifelse-Funktion

IQ_data_extended<-IQ_data_extended %>%
  mutate(verbal = ifelse(lower_IQ, 'low', 'mid'),
         verbal = ifelse(high_IQ, 'high', verbal)) %>%
  select(-contains('iq_'))

## cut-funktion

IQ_data_tidy$Category <- cut(IQ_data_tidy$IQ, breaks = c(-Inf, 84,115, 130),
                             labels =c("unterdurchschnittlich", "durchschnittlich", "überdurchschnittlich"), right = TRUE)

Schlaubi_Schlumpf <-IQ_data_tidy %>%
  select(1,2,3) %>%
  filter(Category == "überdurchschnittlich")

print(Schlaubi_Schlumpf)

```

Aufgabe 4

gesamten Datensatz in absteigender Reihenfolge des IQs ausgeben

Hide

```

IQ_data_sortiert<-IQ_data_tidy %>% arrange(desc(IQ))

```

Kovarianz- und Korrelationsmatrix erstellen

Hide

```

Kovarianzmatrix<- IQ_data_sortiert %>%
  select(4,5,6,7) %>% ## Welche Spalten werden benötigt
  cov()

Korrelationsmatrix<- IQ_data_sortiert %>%
  select(4,5,6,7) %>%
  cor()

```