

AB02: Funktionen in R

Julius Henning

Lernziele

- Wiederholung der Übungsinhalte
- Anwendung von Indexierung zur Auswahl und Manipulation von Daten
- Verstehen und Anwenden von Funktionen in R
- Lesen von Codefragmenten

Aufgabe 1

Kopiere den folgenden Code und führe ihn in R aus.

```
vec.namen <- c("Mia", "Noah", "Emma", "Elias", "Sophia", "Leon", "Clara",  
              "Finn", "Hannah", "Luca", "Emilia", "Paul", "Lena", "Jonas",  
              "Amelie", "Max", "Marie", "Anton", "Lisa", "Tim")  
vec.noten <- c(2, 1, 6, 4, 4, 2, 2, 1, 1, 3, 4, 3, 5, 3, 1, 1, 3, 2, 2, 2)
```

1.1 Erstelle aus den Vektoren `vec.namen` und `vec.noten` ein `Data.frame`, das die Spalten Namen und Noten enthält.

1.2 Mit welchen Funktionen lassen sich die beste und schlechteste Note bestimmen? Wie erhält man den Mittelwert und Median?

1.3 Erstelle dir eine Übersicht in der du aufführst, welche Datentypen oder Strukturen in die Funktionen aus Teilaufgaben a und b eingegeben (`inputs`) und welche Datentypen/-Strukturen jeweils ausgegeben (`output`) werden.

Aufgabe 2

Mache dich vertraut mit den Funktionen `setw()` und `getwd()`

- Versuche in eigenen Worten wiederzugeben, was passiert wenn diese Funktionen ausgeführt werden.
- Welche Relevanz könnten die Funktionen im Bezug auf unsere Arbeit mit R haben?
- Lassen sich die Funktionen im Rahmen der Begriffe `in-` und `output` beschreiben? Welche Schwierigkeiten gibt es möglicherweise?

Aufgabe 3

In dieser Übung beschäftigen wir uns viel damit Anweisungen in die Programmiersprache R zu übersetzen. Um Fehler im Code besser nachvollziehen zu können ist es jedoch auch wichtig diese Anweisungen zurück in die normale Sprache übersetzen zu können.

3.1 Übersetze die nachfolgenden Codefragmente in eigene Worte und versuche dabei so präzise wie möglich zu sein.

- `vec.quant <- 1:10*.1`
- `rep(c(TRUE, FALSE), times = 5)`
- Für einen Vektor `x` von Länge 10: `y <- x[1+2*0:8]`

3.2 Was macht der folgende Code?

- Für eine gerade Zahl `n`

```
Vec.Int <- 1:(n/2)
x <- 2*Vec.Int
y <- x-1
mean(x-y)
```

b.

```
Vec.Int <- sample(1:26,  
                size = 8)  
LETTERS[Vec.Int]  
  
vec.cond <- rep(c(TRUE, FALSE),  
               times = 13)  
  
identical(LETTERS[vec.cond],  
          LETTERS[2*0:12+1])  
  
identical(LETTERS[1:26 <= 13],  
          LETTERS[13:26])
```

i Lösungshinweise:

- *LETTERS*: Vektor (*character*) der alle Großbuchstaben enthält
- Um mehr Informationen über Befehle in R zu bekommen kannst du die Hilfe nutzen oder Seiten wie [geeksforgeeks](#) nutzen
- **TRUE** und **FALSE** können als logische Werte auch zur Indexierung von Vektoren genutzt werden