

AB03: Mengenlogik

Julius Henning

Lernziele

- Anwendung von Indexierung zur Auswahl und Manipulation von Daten
- Verstehen und Anwenden mengenlogischer Aussagen auf Daten

Vorbereitung

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)
```

Aufgabe 1:

1.1 Lese die folgenden Codefragmente und versuche zu verstehen was passiert. Formuliere deine Annahme kurz und präzise und überprüfe sie anschließend indem du den Code ausführst.

Code 1 - Was passiert in den mit **##** gekennzeichneten Zeilen?

```
set.1 <- LETTERS[1:17]  
set.2 <- LETTERS[12:25]  
  
union(set.1, set.2) ##  
intersect(set.1, set.2) ##  
setdiff(set.1, set.2) ##
```

Code 2 - Worin besteht der Unterschied der mit **##** gekennzeichneten Zeilen?

```
set <- sample(1:100, 1000, replace = T)

3 %in% set ##
3 == set ##
```

Code 3 - Was passiert in den mit ## gekennzeichneten Zeilen?

```
set <- 1:100

cond.leq40 <- set <= 40
cond.greater80 <- set > 80
cond.even <- set %% 2 == 0

set[cond.leq40] ##
set[cond.greater80 | cond.leq40] ##
set[cond.greater80 & cond.even] ##
set[!(cond.greater80 | cond.leq40)] ##
```

1.2 Mit dem Wissen aus Teilaufgabe 1.1 löse für

```
set <- sample(1:1000, 100, replace = T)
```

die folgenden Aufgaben.

- Bilde aus `set` eine Teilmenge `set.a` die alle ungeraden Zahlen enthält.
- Bilde aus `set` eine Teilmenge `set.b` die alle Zahlen enthält die größer oder gleich 500 sind.
- Gib alle Zahlen aus, die sich **nicht** durch 5 teilen lassen.

Aufgabe 2: Mengenlogische Aussagen und Filter

2.1 Gegeben ist der Vektor `zahlen <- c(3, 5, 7, 9, 10, 12, 14, 15)`. Schreibe eine Bedingung, die die geraden Elemente des Vektors `zahlen` auswählt.

2.2 Prüfe mit einer logischen Aussage, ob der Wert 7 in dem Vektor `zahlen` enthalten ist.

2.3 Filtere aus dem Vektor `zahlen` die Elemente, die größer als 10 **und** durch 2 teilbar sind.

2.4 Erstelle die Mengen `a <- c(1, 2, 3, 5)` und `b <- c(3, 4, 5, 6)`. Führe dann die folgende Operationen durch:

- Vereinigung der Mengen `a` und `b`.
- Schnittmenge der Mengen `a` und `b`.
- Differenz der Menge `a` zu `b` (d.h. die Elemente, die in `a` sind, aber nicht in `b`).

Aufgabe 3: Praktische Anwendung

Wie muss der Code lauten, um die folgenden Informationen zu erhalten:

- Welche Studierenden haben eine 3 oder besser erhalten?
- Gibt es mehr Studierende mit einer 2 als mit den Noten 5 und 6?
- Verbinde die Vektoren zu einem Dataframe `data.passed` mit den Spalten `namen` und `noten` isoliere alle Personen die bestanden haben

i Lösungshinweise:

- `sample(x, n, replace = TRUE)`: generiert eine Stichprobe der Größe n aus dem Vektor x . Das Argument `replace = TRUE` erlaubt Wiederholungen in der Stichprobe.
- *Summieren*: Der Befehl `sum()` summiert die Elemente eines numerischen Vektors auf, auch für Wahrheitswerte kann man ihn verwenden. Dabei entspricht dann `TRUE` der 1 und `FALSE` dem Wert 0.
- *Dataframes*: Die am meisten verwendete Datenstruktur in diesem Kurs sind die Dataframes. Mit dem Befehl `data.frame()` können Vektoren gleicher Länge einfach zu einem dataframe verbunden werden
- *Subdatensätze*: Mit dem Befehl `subset` können aus einem Dataframe sehr einfach Daten mit gewünschten Eigenschaften isoliert werden.