Aufgabenblatt 1: Einführung in R und RStudio (Lösungsvorschlag)

TU Dortmund, Lehrstuhl für Mikroökonomie, Wintersemester 2026/27

Kausalanalyse: Economics meets Data Science

Marc Justin Schmidt

22. Oktober 2026

Aufgabe 1: Grundlegende Funktionalitäten in R (Base R)

Aufgabe 1.1: Vektoren

a. Erstellen Sie einen Vektor `x`, der die Zahlen von 1 bis 10 enthält.

```
x <- 1:10
x
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

b. Addieren Sie 2 zu jedem Element und multipliziere das Ergebnis mit 3.

```
x_transformed <- (x + 2) * 3
x_transformed
```

```
##  [1]  9 12 15 18 21 24 27 30 33 36
```

c. Berechnen Sie den Mittelwert und die Standardabweichung des Vektors `x`.

```
mean(x)
```

```
## [1] 5.5
```

```
sd(x)
```

```
## [1] 3.02765
```

Aufgabe 1.2: Matrizen

a. Erstellen Sie eine 4x4-Matrix `A` mit den Zahlen von 1 bis 16.

```
A <- matrix(1:16, nrow = 4, ncol = 4)
A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    5    9   13
## [2,]    2    6   10   14
## [3,]    3    7   11   15
## [4,]    4    8   12   16
```

b. Greifen Sie auf das Element in der 2. Zeile und 3. Spalte zu.

```r
A[2, 3]
```

```
## [1] 10
```

c. Berechnen Sie die Spaltensummen der Matrix `A`.

```r
colSums(A)
```

```
## [1] 10 26 42 58
```

Aufgabe 1.3: Data Frames

a. Erstellen Sie einen Data Frame mit 5 Personen (Name, Alter, Studiengang).

```r
personen <- data.frame(
  Name = c("Anna", "Ben", "Clara", "David", "Eva"),
  Alter = c(21, 24, 23, 20, 25),
  Studiengang = c("WiWi", "Informatik", "VWL", "Psychologie", "WiWi")
)
personen
```

```
##    Name Alter Studiengang
## 1  Anna    21        WiWi
## 2   Ben    24  Informatik
## 3 Clara    23         VWL
## 4 David    20 Psychologie
## 5   Eva    25        WiWi
```

b. Filteren Sie alle Personen, die älter als 22 Jahre sind.

```r
personen %>%
  filter(Alter > 22)
```

```
##    Name Alter Studiengang
## 1   Ben    24  Informatik
## 2 Clara    23         VWL
## 3   Eva    25        WiWi
```

c. Berechnen Sie das Durchschnittsalter der Personen im Data Frame.

```r
mean(personen$Alter)
```

```
## [1] 22.6
```

d. Erstellen Sie den folgenden Datensatz:

```
##           song     kuenstler laenge_min genre streams_mio
## 1 Shape of You    Ed Sheeran        4.2   Pop        3200
## 2         Numb   Linkin Park        3.0  Rock        1800
## 3         Halo       Beyoncé        4.3   RnB        1500
## 4    Despacito    Luis Fonsi        4.0 Latin        2800
## 5      Bad Guy Billie Eilish        3.2   Pop        2200
```

```r
playlist <- data.frame(
  song = c("Shape of You", "Numb", "Halo", "Despacito", "Bad Guy"),
  kuenstler = c("Ed Sheeran", "Linkin Park", "Beyoncé", "Luis Fonsi",
                "Billie Eilish"),
  laenge_min = c(4.2, 3.0, 4.3, 4.0, 3.2),
  genre = c("Pop", "Rock", "RnB", "Latin", "Pop"),
  streams_mio = c(3200, 1800, 1500, 2800, 2200)
)

playlist
```

e. Filteren Sie nach Künstlern, die mit "B" beginnen ("RegEx").

```r
playlist %>%
  filter(grepl("^B", kuenstler))
```

```
##      song     kuenstler laenge_min genre streams_mio
## 1    Halo       Beyoncé        4.3   RnB        1500
## 2 Bad Guy Billie Eilish        3.2   Pop        2200
```

f. Erstellen Sie einen Teildatensatz, der nur den Künstler und das Genre enthält.

```r
playlist %>%
  select(kuenstler, genre)
```

```
##       kuenstler genre
## 1    Ed Sheeran   Pop
## 2   Linkin Park  Rock
## 3       Beyoncé   RnB
## 4    Luis Fonsi Latin
## 5 Billie Eilish   Pop
```

Aufgabe 2: Datenbearbeitung mit dem `tidyverse`

a. Installieren und laden Sie das Paket `tidyverse`. Laden Sie den Datensatz `mtcars` (in R vorinstalliert) und speichern Sie ihn als `cars_data`.

```
library(tidyverse)

cars_data <- mtcars
```

b. Wähle die Spalten `mpg`, `cyl`, und `hp` aus.

```
cars_data %>%
  select(mpg, cyl, hp) %>%
  head()
```

```
##                    mpg cyl  hp
## Mazda RX4         21.0   6 110
## Mazda RX4 Wag     21.0   6 110
## Datsun 710        22.8   4  93
## Hornet 4 Drive    21.4   6 110
## Hornet Sportabout 18.7   8 175
## Valiant           18.1   6 105
```

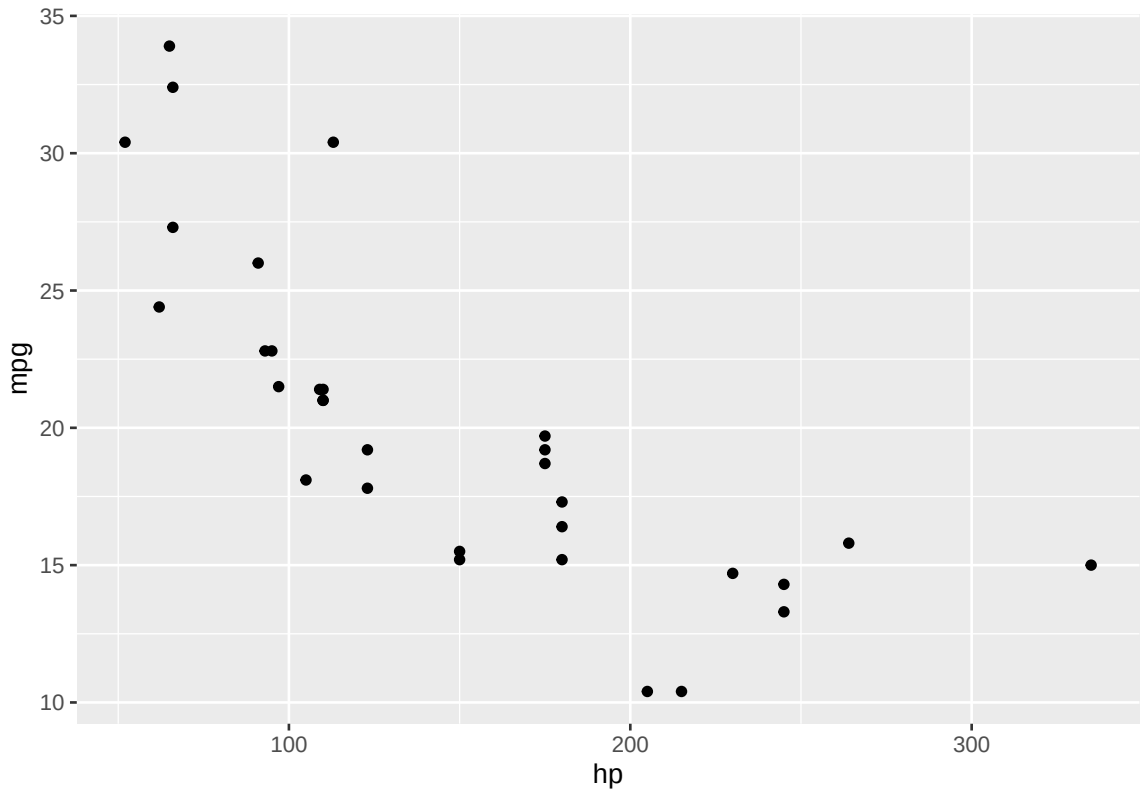
c. Filtern Sie die Autos mit mehr als 100 PS (`hp > 100`).

```
cars_data %>%
  select(mpg, cyl, hp) %>%
  filter(hp > 100) %>%
  head()
```

```
##                    mpg cyl  hp
## Mazda RX4         21.0   6 110
## Mazda RX4 Wag     21.0   6 110
## Hornet 4 Drive    21.4   6 110
## Hornet Sportabout 18.7   8 175
## Valiant           18.1   6 105
## Duster 360        14.3   8 245
```

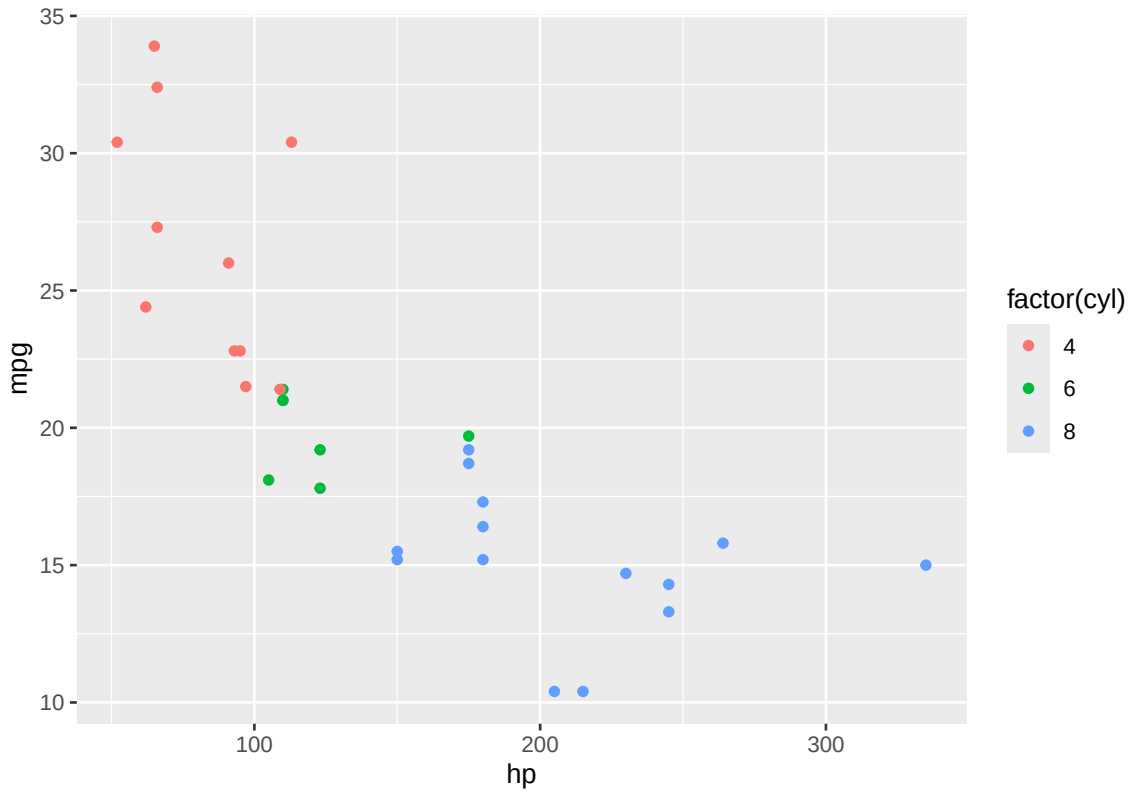
d. Erstellen Sie ein Scatterplot von `hp` gegen `mpg`. (*Hinweis: Benutzen Sie die Funktion `ggplot()` aus dem `ggplot2` Paket.*)

```
ggplot(cars_data, aes(x = hp, y = mpg)) +
  geom_point()
```



e. Koloriere die Punkte nach der Anzahl der Zylinder (`cyl`).

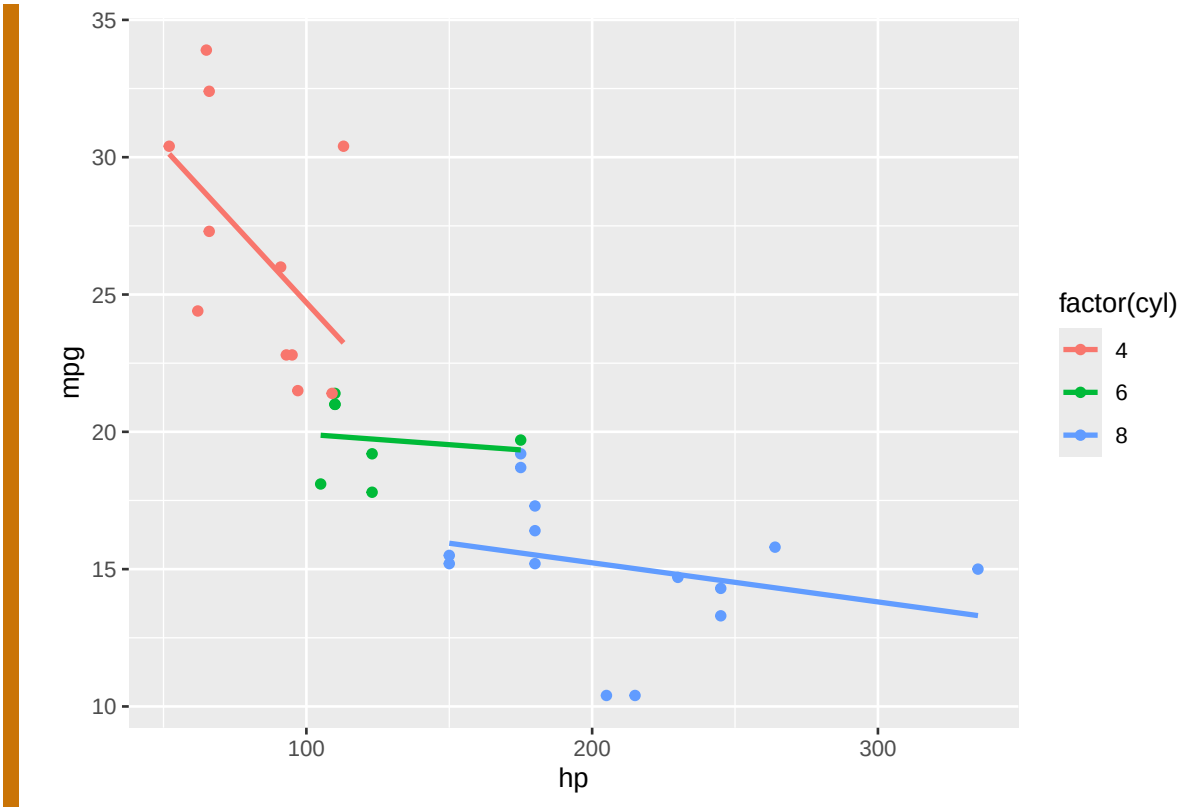
```
ggplot(cars_data, aes(x = hp, y = mpg, color = factor(cyl))) +
  geom_point()
```



f. Fügen Sie eine Regressionslinie hinzu, die den Zusammenhang zwischen `hp` und `mpg` zeigt.

```
ggplot(cars_data, aes(x = hp, y = mpg, color = factor(cyl))) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```


35
30
25
20
15
10
mpg
100
200
300
hp
factor(cyl)
4
6
8

Aufgabe 3: Eigenes "Programmieren" in R

Aufgabe 3.1: Bedingungen

a. Schreiben Sie ein `if`-Statement, das überprüft, ob eine Zahl `x` positiv, negativ oder null ist, und eine entsprechende Nachricht ausgibt.

```
x <- 2
if (x > 0) {
  print("Positiv")
  } else if (x < 0) {
    print("Negativ")
    } else {
      print("Null")
      }
```

```
## [1] "Positiv"
```

b. Schreiben Sie ein `if`/`while`-Statement, das solange eine Zahl `x` halbiert, bis sie kleiner als 1 ist.

```
x <- 24
while (x >= 1) {
  x <- x / 2
  print(x)
}
```

```
## [1] 12
## [1] 6
## [1] 3
## [1] 1.5
## [1] 0.75
```

```
halbier_bis_eins <- function(x) {
  print(x)
  if (x >= 1) {
    halbier_bis_eins(x / 2)
  }
}
halbier_bis_eins(24)
```

```
## [1] 24
## [1] 12
## [1] 6
## [1] 3
## [1] 1.5
## [1] 0.75
```

Aufgabe 3.2: Schleifen

a. Schreiben Sie eine Schleife, die die Quadratzahlen von 1 bis 10 berechnet und ausgibt.

```r
for (i in 1:10) {
  print(i^2)
}
```

```
## [1] 1
## [1] 4
## [1] 9
## [1] 16
## [1] 25
## [1] 36
## [1] 49
## [1] 64
## [1] 81
## [1] 100
```

b. Schreiben Sie eine Schleife, die durch die Matrix `A` (aus Aufgabe 1) iteriert und die Summe jedes Elements ausgibt.

```r
sum_A <- 0
for (i in A) {
  sum_A <- sum_A + i
}
sum_A
```

```
## [1] 136
```

Aufgabe 3.3: Eigene Funktionen

a. Erstellen Sie eine Funktion, die Fahrenheit in Celsius umwandelt. (*Hinweis: Celsius* = $(Fahrenheit - 32) \times 5/9$)

```r
fahr_to_cel <- function(f) {
  (f - 32) * 5/9
}
fahr_to_cel(100)
```

```
## [1] 37.77778
```

b. Erstellen Sie eine Funktion, die die Anzahl von Wörtern in einem gegebenen Text zählt.

```r
word_count <- function(text){
  length(strsplit(text, " ")[[1]])
}
word_count("R ist eine coole Sprache für Datenanalyse.")
```

```
## [1] 7
```

c. Erstellen Sie eine Funktion, die n mal einen Würfel würfelt und die Ergebnisse als Vektor zurückgibt.

```r
roll_dice <- function(n) {
  sample(1:6, size = n, replace = TRUE)
}
roll_dice(2)
```

```
## [1] 4 6
```