

## Aufgabenblatt 1 – Lösungen

### Aufgabe 1 (Einführung in R)

- a) Die Dokumentation *simpleR* von John Verzani liefert eine gute Einführung in R. Downloaden Sie das Dokument via <http://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf> und lesen Sie *Section 2: Data*, um zu lernen, wie Datensätze mit der `c()` Funktion eingegeben werden können. Erzeugen Sie anschließend folgende zwei Vektoren in R:

$$x^T = [10 \ 8 \ 13 \ 9 \ 11 \ 14 \ 6 \ 4 \ 12 \ 7 \ 5]$$
$$y^T = [8.1 \ 6.9 \ 7.5 \ 8.8 \ 8.3 \ 9.9 \ 7.2 \ 4.2 \ 10.8 \ 4.8 \ 5.6]$$

```
x <- c(10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5)
y <- c(8.1, 6.9, 7.5, 8.8, 8.3, 9.9, 7.2, 4.2, 10.8, 4.8, 5.6)
```

- b) Welche Operation wird durch `x + y` ausgeführt? Wenden Sie auch `-`, `*`, `/`, `%*%` an.

`x + y` ist die elementweise Addition. Analog sind `-`, `*`, `/` die elementweise Subtraktion, Multiplikation und Division. `x %*% y` ist das Matrikprodukt  $x^T y$ .

- c) Die Einträge der beiden Vektoren können wir als Wertepaare  $(x_1, y_1), \dots, (x_{11}, y_{11})$  betrachten. Erzeugen Sie mit `plot()` ein Streudiagramm dieser Daten.

```
plot(x, y)
```

- d) Lesen Sie *Section 5: Multivariate Data* und lernen Sie den Datentyp `data.frame` kennen. Erstellen Sie einen `data.frame` aus `x` und `y`.

```
data <- data.frame(x = x, y = y)
```

- e) Schätzen Sie die Koeffizienten  $\beta_0$  und  $\beta_1$  des linearen Modells  $y = \beta_0 + \beta_1 x + u$  mithilfe der Funktion `lm()` (steht für *lineares Modell*). Erklärungen finden Sie in *Section 13: Regression Analysis*. Zeichnen Sie dann mithilfe von `abline()` die angepasste Regressionsgerade in Ihr Streudiagramm ein.

```
model <- lm(formula = y ~ x, data = data)
abline(model)
```

- f) Erklären Sie den Output von `summary()`, angewendet auf Ihr geschätztes Modell.

```
summary(model)
```

Eine Erklärung des Outputs liefert <https://www.youtube.com/watch?v=NEfjirpOj7s>.

## Aufgabe 2 (Einführung in R Markdown)

R Markdown ist eine Kombination aus R, unserer Programmiersprache für Datenanalyse, und Markdown, einer einfachen Auszeichnungssprache für die Erstellung von Berichten. Durch die Kombination können wir R Code direkt in Textdokumente integrieren und mit minimalem Aufwand dynamische Analyseberichte generieren. Entwickelt wurde R Markdown 2014 von Yihui Xie, der einen einfachen Weg gesucht hat, seine R Hausaufgaben aufzuschreiben.

- a) Das Buch *R for Data Science* von Hadley Wickham bietet unter anderem eine gute Einführung in R Markdown. Das Buch ist kostenlos online unter <https://r4ds.had.co.nz/> verfügbar. Lesen Sie *Chapter 27: R Markdown*, um zu lernen, wie ein R Markdown Dokument erstellt werden kann. Verwenden Sie die Vorlage aus Abschnitt 27.2 und erstellen Sie selbst ein R Markdown Dokument im .html-Format.

Dazu in R Studio links oben **File > New File > R Markdown** wählen. Es öffnet sich ein Fenster, dort links unten auf **Create Empty Document** klicken. Es öffnet sich eine leere Datei, dort die Vorlage einfügen. Anschließend auf **Knit** klicken.

- b) Fügen Sie in das R Markdown Dokument Ihre Lösungen aus Aufgabe 1) ein.

```
---
title: "Lösung zu Aufgabe 1"
date: "17.04.2023"
author: "Lennart"
output: html_document
---

### a)
```{r}
x <- c(10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5)
y <- c(8.1, 6.9, 7.5, 8.8, 8.3, 9.9, 7.2, 4.2, 10.8, 4.8, 5.6)
```
```

- c) Wie können Sie ein .pdf-Dokument erstellen?

Indem `output: html_document` durch `output: pdf_document` ersetzt wird.