

Höhere Mathematik IV - Stochastik für Ingenieure

Übungsblatt 1

Aufgabe 1.1 (Erste Schritte in R und Rstudio)

- a) Installieren Sie R von der Website <https://cran.r-project.org>.
- b) Installieren Sie Rstudio von der Website <https://www.rstudio.com>.
- c) Öffnen Sie Rstudio, tippen Sie `1+2` in die Kommandozeile und drücken Sie Enter.
- d) Tippen Sie die folgenden Ausdrücke in die Konsole in Rstudio ein und erklären Sie die Ergebnisse.

```
> (5+7)*2
> 5+7*2
> 1.5*2
> 1,5*2
> 2^3^4
> (2^3)^4
> exp(1)
> ?exp
> sqrt(9)
> 9^(1/2)
> 9^0.5
> Sqrt(9)
```

Aufgabe 1.2 (Werte zuweisen)

Zahlen und andere Objekte können in R mittels Variablen gespeichert werden. Dies erreichen wir mit dem Zuweisungsoperator `=` bzw. `<-`.

- a) Weisen Sie den Wert `3` der Variable `x` zu. Anschließend weisen Sie den Wert

$$\sqrt{x^3 + \log(e^{3x})}$$

der Variable `y` zu.

b) Geben Sie die folgenden Zeilen in Rstudio ein und erklären Sie die Resultate.

```
> x
> y
> X
> x+y
> x.prod <- x*y
> x.prod
> x.prod <- x.prod*x
> ls()
> rm(x)
> x
> ls()
```

Um Zeichenketten zuzuweisen, benutzen wir (einfache oder doppelte) Anführungszeichen.

c) Was sind die Ergebnisse der folgenden Zeilen?

```
> avenger.1 <- "Captain America"
> avenger.2 <- "Thor"
> number <- 10
> no.number <- "10"
> number + 1
> no.number + 1
```

Aufgabe 1.3 (Vektoren)

Um Vektoren zu erzeugen, benutzen wir die Funktion `c()`.

a) Definieren Sie einen Vektor `x` mit Werten 1, 10, 100, 1000, einen Vektor `city` mit Werten Frankfurt, Berlin, New York, Kassel, und einen Vektor `z` mit Werten 0,1,2,“3”.

Um Vektoren mit vielen Einträgen einfacher zu definieren, können die folgenden Operatoren hilfreich sein:

- Der Doppelpunkt-Operator `a:b` erzeugt einen Vektor von ganzen Zahlen, beginnend bei `a` und endend bei `b`.
- Der Folgen-Operator `seq(from = a, to = b, by = step)` erzeugt einen Vektor von (nicht zwangsweise ganzen) Zahlen, beginnend bei `a` und endend bei `b`, mit der Differenz `step` zwischen jeder Zahl.
- Der Wiederholungs-Operator `rep(vec, n)` erzeugt einen Vektor, der aus `n` Kopien des Vektors `vec` besteht.

b) Definieren Sie die folgenden Vektoren in R.

```
> 7 8 9 10 11 12
> 12 11 10 9 8 7 6 5 4 3 2 1
> -1.0 -0.5 0.0 0.5 1.0
> 500 1000 1500 2000 2500
> -2 -1 0 1 2 -2 -1 0 1 2
> 1.0 2.0 3.0 4.0 5.0 5.1 5.2 5.3 5.4 5.5
```

Berechnungen mit Vektoren werden komponentenweise durchgeführt. Außerdem erhalten wir mit den Befehlen `length()` und `sum()` die Länge bzw. die Summe aller Elemente eines Vektors.

c) Definieren Sie die Vektoren

```
> x <- 10:8
> y <- seq(from = 0.0, to = 1.0, by = 0.2)
```

Erklären Sie die Ergebnisse der folgenden Zeilen.

```
> x + y
> x * y
> x^2 + 2
> n <- length(x+y)
> sum(x+y)/n
```

Um auf bestimmte Elemente eines Vektors direkt zuzugreifen, benutzen wir eckige Klammern `[]`. Innerhalb dieser Klammern sind auch Vergleichs- oder Logikoperatoren wie `<` (kleiner), `==` (gleich), `&` (UND), `|` (ODER) erlaubt.

d) Definieren Sie die Vektoren

```
> x <- seq(from = 0, to = 50, by = 2)
> y <- 50:1
```

Was sind die Resultate der folgenden Zeilen?

```
> x[10]
> y[c(1,3,10)]
> x[1:4]
> x[x>45]
> x[x>10 & x<= 20]
> y[y == 5 | y > 45 | y < 3]
```

Aufgabe 1.4 (Dataframes)

In Aufgabe 1.3 haben wir gesehen, dass Vektoren Mengen von *einem* Merkmal sind. Möchten wir verschiedene Merkmale in einer Struktur zusammenfassen, so verwenden wir die Funktion `data.frame()`. In diesem Fall repräsentiert jede Spalte eine unterschiedliche Komponente des Dataframes.

Erzeugen Sie das folgende Beispiel in Rstudio:

```
> names <- factor(c("Michaela", "Alexander", "Nadja", "Alina", "Ricarda", "Omar"))
> age <- c(20, 20, 19, 21, 21, 20)
> numbers <- c(123451, 123452, 123453, 123454, 123455, 123456)
> grade <- ordered(c("A", "C", "C", "B-", "F", "A-"), levels = c("F", "C", "B-", "A-", "A"))
> passed <- c(grade > "F")
> Students <- data.frame(Name = names, Alter = age, Matr.nr = numbers,
  Note = grade, Pruefung.bestanden = passed)
> Students
```

Um auf Elemente des Dataframes zuzugreifen, können wir ähnlich vorgehen wie im Falle von Vektoren. In diesem Fall haben wir allerdings Zeilen und Spalten, die durch ein Komma getrennt werden [,]. Wollen wir mit einer speziellen Komponente des Dataframes arbeiten, so können wir den Operator \$ benutzen.

- a) Erklären Sie die folgenden Zeilen in R.

```
> Students[1,3]
> Students[2,]
> Students[1:3, c(1,4)]
> Students[,4]
> Students$Note
> Students[grade > "C",]
```

- b) Benutzen Sie die folgende Tabelle, um ein Dataframe zu erzeugen.

Alter	Geschlecht	Fernsehzeit in h/Woche	besitzt ein Smartphone
6	männlich	5	nein
8	männlich	10	ja
7	weiblich	3	nein
10	männlich	15	ja
6	weiblich	8	ja
7	weiblich	9	nein
7	weiblich	13	ja
9	männlich	8	ja

- c) Erzeugen Sie ein Dataframe, das nur das Alter und die Fernsehzeiten von allen Kindern jünger als 8 enthält.
- d) Berechnen Sie die durchschnittlichen Fernsehzeiten von allen Kindern, die ein Smartphone besitzen.

Aufgabe 1.5 (Erstellen eines R Skripts bzw. einer R Markdown Datei)

Ein R Skript ist eine Text-Datei, die alle Befehle enthält, die man normalerweise in der Kommandozeile in R eingeben kann. Ein Vorteil von R Skripten ist zum einen, dass sie als eine Art Dokumentation dienen. Zum anderen kann die Datei jederzeit editiert werden (z.B. mit Kommentaren) und später erneut verwendet werden.

- a) Erzeugen Sie eine .R Datei mit dem Namen `stochingR1.R` (in R studio via **File** → **New** → **R Script**), und fügen Sie den folgenden Kommentar (mittels #) in die ersten Zeilen des Dokuments ein:

```
#' ---
#' title: "Lösungen zu Übungsblatt 1"
#' author: "Ihr Name"
#' date: "30. Oktober 2019"
#' ---
```

- b) Fügen Sie nun alle Lösungen der vorherigen Aufgaben in diese Datei ein. Fügen Sie vor jeder Aufgabe einen Kommentar hinzu, um zu verdeutlichen, um welche Aufgabe es sich handelt.
- c) Ergänzen Sie die Datei mit ausreichend Kommentaren, um Ihr Vorgehen zu erläutern.
- d) Nun können Sie die Datei Zeile für Zeile (oder durch Markieren auch mehrere Zeilen) ausführen, indem Sie **Ctrl + Enter** drücken.

Eine pdf-Datei der Ein- und Ausgabe können Sie beispielsweise mit dem Knit-Befehl von Rstudio (**Ctrl + Shift + K**) erzeugen. Bitte beachten Sie aber hierbei, dass insbesondere Grafiken unschön dargestellt werden könnten. Eine bessere Alternative wäre eine R Markdown Datei (mit Endung `.rmd`) zu erstellen. In ihr können ohne großes Mehrwissen Berichte geschrieben werden, die Texte, R-Code, und den Output beinhalten.

- e) Erzeugen Sie eine `.rmd` Datei mit dem Namen `stochingR1.rmd` (in R studio via **File** → **New** → **R Markdown**), und fügen Sie die folgenden Zeilen am Beginn des Dokuments ein:

```
---
title: "Lösungen zu Übungsblatt 1"
author: "Ihr Name"
date: "30. Oktober 2019"
output: pdf_document oder html_document oder word_document
---

```{r setup, include=FALSE, chache = F}
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_chunk$set(error = TRUE)
```
```

- f) Kopieren Sie nun den R-Code der Lösungen in das Dokument. Dabei sollten Sie beachten, dass R-Code immer zwischen den folgenden Zeilen eingefügt werden muss:

```
```{r}
hier R-Code einfügen!
```
```

Statt diese Zeilen zu tippen, kann in Rstudio auch einfach auf **Insert** → **R** geklickt werden. Auch hier kann der Code nun mit **Ctrl + Enter** ausgeführt werden. Damit bildet R Markdown keine Einschränkungen gegenüber einem R Skript.

- g) Erstellen Sie anschließend eine pdf- (oder html- oder Word-) Datei, indem Sie den Knit-Befehl von Rstudio benutzen.
