

Statistische Software (R) – Hausarbeit 2

Sommersemester 2021, 14.06.2021 - 21.06.2021

Name: _____

Immatrikulationsnummer: _____

Studiengang: _____

Hiermit bestätige ich, dass ich die Anweisungen auf diesem Blatt gelesen und verstanden habe. Ich bestätige, dass die abgegebene Lösung vollständig und alleinig von mir bearbeitet und erstellt worden ist, ohne Hilfe von anderen in Anspruch zu nehmen. Ich bestätige, dass ich über die Vorlesungsmaterialien hinausgehende Quellen wie Bücher oder Internetseiten im Code angegeben und falls zutreffend verlinkt sind.

Unterschrift: _____

Prüfungshinweise:

1. Überprüfen sie ob die heruntergeladene Angabe vollständig ist. Sie sollte 3 Aufgaben enthalten. Einzelne Aufgaben können aus mehreren Teilaufgaben bestehen.
2. Insgesamt können (ohne Bonuspunkte) 20 Punkte erreicht werden. Die Aufteilung der Punkte auf die einzelnen Aufgaben kann der Angabe entnommen werden.
3. Die Lösung soll in Form einer `.Rmd` Datei abgegeben werden. Nutzen Sie Markdown um Begin und Ende einzelner Aufgaben und Teilaufgaben zu kennzeichnen. Ist die Zugehörigkeit von Code zu einer der (Teil-)Aufgaben nicht eindeutig deklariert, kann es passieren, dass Sie dafür keine Punkte bekommen.
4. Sollten Sie technische oder andere Schwierigkeiten haben, kontaktieren Sie Bitte die Kursleiter. E-mail: andreas.bender@stat.uni-muenchen.de, philipp.kopper@stat.uni-muenchen.de. (Bitte die Emails an alle gelisteten Personen schicken!)
5. Die Aufgaben müssen alle eigenständig bearbeitet werden. Insbesondere sind keine Arbeitsgruppen erlaubt und sonstige Diskussion der Aufgaben und Lösungen mit anderen Personen (egal ob diese Statistik studieren oder nicht) nicht zulässig.
6. Das Internet kann passiv genutzt werden. D.h. es dürfen Internetseiten oder Foren aufgerufen und gelesen werden, das aktive Stellen von Fragen, die relevant zur Lösung der Aufgaben sind, ist allerdings nicht zulässig. Ebenso dürfen keine Aufgaben oder Lösungsvorschläge und anderen Hinweise im Internet gepostet oder per Chat, Email und anderen Kommunikationswegen diskutiert oder verteilt werden.
7. Sollte der Verdacht auf Plagiat, Betrug oder anderweitig unzulässiges Verhalten bestehen, können zusätzliche (mündliche) Prüfungen einberufen werden um die eigenständige Bearbeitung der Aufgaben zu prüfen.
8. Zweifel an der eigenständigen Bearbeitung ihrer Abgabe führen zum nicht-bestehen der Prüfung und dem Einschalten des Prüfungsausschusses.
9. Die Abgabe erfolgt bis Mitternacht (23:59 Uhr) am 21.06.2021.

Aufgabe 1

4 Punkte

Sie können bei diesem Assignment folgende Bonuspunkte sammeln:

- (a) Abgabe via Github Classroom (BONUS: 2P)
- (b) Rmarkdown kompiliert ohne Fehler (BONUS: 1P)
- (c) R Code folgt dem Advanced R Style guide (<http://adv-r.had.co.nz/Style.html>) (BONUS: 1P)

In den folgenden beiden Aufgaben geht es in erster Linie um Loops/Iterationen in R. Als Aufgabenkontext lehnen wir uns an Verteilungen aus Statistik 2 an. Achten Sie darauf den Style-Guide einzuhalten, Funktionen ordentlich zu dokumentieren und wenn nötig Inputs zu überprüfen.

Aufgabe 2

0.25 Punkte

Beschreiben Sie kurz, was am untenstehenden Code problematisch ist.

```
i <- 1
while (i > 0) {
  print(i)
  i <- i + 1
}
```

Aufgabe 3

9.75 Punkte

Ein*e Freund*in wettet mit Ihnen mit einer Münze vier mal hintereinander Kopf zu werfen. Es wird behauptet, dass die Münze fair ist (d.h. bei einem Wurf der Münze kommt Kopf oder Zahl mit je 50% Wahrscheinlichkeit). Kommt 4 mal hintereinander Kopf, müssen Sie 50 Euro bezahlen, ansonsten kriegen Sie einen Euro. Leider haben Sie aufgrund einer partiellen Amnesie vergessen, wie man ausrechnet, ob Sie in Erwartung Gewinn machen. Dafür können Sie sich aber noch sehr gut an die Programmiervorlesung erinnern und Sie haben einen Laptop mit R bei sich. Sie erinnern sich insbesondere, dass man mit `sample` zufällig Elemente aus einer Menge ziehen und somit Zufallsexperimente durchführen kann. Der untenstehende code simuliert z.B. das viermalige Ziehen einer Münze.

```
set.seed(20210614)
outcomes <- c("KOPF", "ZAHL")
sample(outcomes, 1, prob = c(.5, .5))

## [1] "KOPF"
```

```

sample(outcomes, 1, prob = c(.5, .5))

## [1] "ZAHL"

sample(outcomes, 1, prob = c(.5, .5))

## [1] "ZAHL"

sample(outcomes, 1, prob = c(.5, .5))

## [1] "ZAHL"

```

Sie wollen nun herausfinden mit welcher Wahrscheinlichkeit man 4 mal hintereinander Kopf wirft. Stellen Sie dabei sicher, dass ihre Ergebnisse reproduzierbar sind.

Hinweis: Man kann auch Funktionen ohne Inputs definieren, falls keine Inputs notwendig sind.

- (a) Schreiben Sie eine Funktion die das obige Experiment (4 mal Münze werfen) durchführt und zurückgibt in wie vielen (von den 4) Versuchen Kopf geworfen wurde. Benutzen Sie innerhalb der Funktion ein `for` loop.
- (b) Unter Verwendung der Funktion aus der vorherigen Teilaufgabe, schreiben Sie eine Funktion, die `n` (Input) solcher Experimente durchführt (also `n` mal "4 mal Münze werfen"). Output soll ein Vektor der Länge `n` sein, der pro Eintrag die Anzahl der Versuche mit Ausgang "Kopf" angibt. Verwenden Sie die Funktion mit `n = 10000` und speichern Sie das Ergebnis im Objekt `outcome`. Berechnen Sie mit Hilfe von `outcome` die Wahrscheinlichkeit für "4 mal Kopf bei 4 Münzwürfen".
- (c) Bonus: Welchen Gewinn/Verlust erwarten Sie, wenn Sie auf die Wette eingehen?
- (d) Sie wollen nun die Verteilung schätzen, die Auskunft gibt wie oft man das Experiment (4 mal werfen) wiederholen muss, bis 4 mal Kopf geworfen wird. Schreiben Sie eine Funktion, die solange das Experiment aus Teilaufgabe (a) wiederholt, bis das erste mal 4 mal Kopf geworfen wird. Der Output soll die benötigte Anzahl Wiederholungen sein.
- (e) Lassen Sie die Funktion aus der vorherigen Teilaufgabe 100 mal laufen und speichern Sie sich jeweils die Anzahl benötigter Versuche bis 4 mal Kopf geworfen wird. Speichern Sie das Ergebnis im Vektor `x`. Wieviele Versuche werden im Durchschnitt gebraucht, bis 4 mal Kopf kommt.
- (f) Verwenden Sie `plot(density(x))` um die Dichte der Verteilung graphisch darzustellen.

Aufgabe 4

10 Punkte

In dieser Aufgabe geht es darum Ihre Visualisierungskünste von einfachen deskriptiven Statistiken mit aktuellem Bezug abzu prüfen. Sie sollen mit Ihren bisherigen `ggplot2` Kenntnissen einige Teilgraphiken des Impfdashboards des RKI nachbauen. Wir beziehen

unsere Daten vom RKI (<https://impfdashboard.de/daten>, 07.06.2021). Laden Sie hierfür den Datensatz **vaccs** von der Moodle Seite herunter.

Verwenden Sie für die Visualisierung das Paket **ggplot2**. Teilweise ist pre-processing der Daten notwendig, danach sollen die Graphiken mit einem zusammenhängenden **ggplot2** Befehl erstellt werden. Achten Sie bei allen Teilaufgaben darauf ordentliche und sauber beschriftete Plots zu erstellen.

- (a) Plotten Sie die gesamte Zeitreihe für das Bundesland Bayern (**region == "DE-BY"**) als Liniendiagramm. Jeder einzelne Impfstoff sollte als eigene Linie mit einer individuellen Farbe dargestellt sein.
- (b) Erzeugen Sie einen **gemeinsamen** Plot, in dem das in a) erzeugte Diagramm für die Bundesländer Saarland, Bremen, Hessen und Berlin ("**DE-SL**", "**DE-HB**", "**DE-HE**", "**DE-BE**") dargestellt wird. Die 4 Diagramme sollen in 2 Zeilen und 2 Spalten dargestellt werden.
- (c) Die Darstellung mit absoluten Zahlen aus der vorherigen Teilaufgabe ist nur bedingt sinnvoll, wenn man den Impffortschritt in verschiedenen Bundesländern vergleichen möchte. Wiederholen Sie die vorherige Teilaufgabe mit der relativen Anzahl geimpfter (= Anzahl geimpfter/Bevölkerung Bundesland). Die Bevölkerungszahlen können hier entnommen werden: https://de.wikipedia.org/wiki/Liste_der_deutschen_Bundesl%C3%A4nder_nach_Bev%C3%B6lkerung. Die *y*-Achse soll von 0 bis 1 gehen.