

## Statistische Software (R) – Hausarbeit 3

Wintersemester 2022

Name: Mingyi Zhou

Immatrikulationsnummer: 12056620

Studiengang: Statistik und Data Science +VWL

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 1 Aufgabe mit mehreren Teilaufgaben beinhalten.
2. Insgesamt können 10 Punkte (+2 Bonuspunkte) erreicht werden.
3. Die Lösung soll in Form von einer einzelnen .Rmd Datei (Rmarkdown) abgegeben werden. Benennen Sie diese Datei **AS3.Rmd**. Ihre Lösung muss Ihren vollständigen Namen und Ihre Immatrikulationsnummer beinhalten. Idealerweise oben in den Metainformationen des Dokuments unter "author".
4. Setzen Sie die **code chunk options** so, dass der Code sowie der R-Output im output file zu sehen sind.
5. Achten Sie darauf, dass Ihre .Rmd Datei kompilierbar ist, es dürfen keine Fehler im Code sein, die das kompilieren unmöglich machen.
6. Laden Sie die unterschriebene Angabe zusätzlich als pdf, jpg oder png Datei hoch. Wenn die Datei zum Zeitpunkt der Abgabe fehlt oder nicht unterzeichnet ist, werden sofort 0 Punkte eingetragen und keine Ausnahmen gemacht.
7. Die Abgabe erfolgt über Github Classrooms oder über Moodle. Für eine Abgabe mit Github Classrooms gibt es 1 Bonuspunkt. Bei einer Abgabe mit Github Classrooms zählt immer das aktuellste commit innerhalb der deadline.
8. Machen Sie Beginn und Ende einzelner Probleme, Aufgaben und Teilaufgaben kenntlich. Ist die Zugehörigkeit von Code zu einer der (Teil-)Aufgaben nicht eindeutig deklariert, kann es passieren, dass Sie dafür keine Punkte bekommen.
9. Achten Sie darauf, dass alle Funktionen nach der Vorgabe in den Übungen dokumentiert sind.
10. Sollten Sie Verständnisfragen haben, nutzen Sie hierzu bitte das Forum, welches auf der Moodle Seite unter Forum zu finden ist.
11. Sollten Sie technische oder andere Schwierigkeiten haben, kontaktieren Sie bitte die Kursleiter.  
E-mail: andreas.bender@stat.uni-muenchen.de, julia.niebisch@stat.uni-muenchen.de. (Bitte die Emails an alle gelisteten Personen schicken!)

12. 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.
13. 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 andere Hinweise im Internet gepostet oder per Chat, Email und anderen Kommunikationswegen diskutiert oder verteilt werden.
14. 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. Wir nutzen Neuronale Netze um Unterschleif (teil-)automatisiert zu prüfen.
15. Zweifel an der eigenständigen Bearbeitung ihrer Abgabe führen zum Nicht-bestehen der Prüfung und dem Einschalten des Prüfungsausschusses.
16. Verwenden und laden Sie keine zusätzlichen R Pakete.
17. Die Abgabe erfolgt bis zum 09.01.2023 um 17 Uhr.

Click on the GitHub Classroom invitation URL on moodle and accept the assignment. A repository under your GitHub name will automatically be created. For example, if your GitHub name is `janedoe`, your repository will be named `assignment3-janedoe`. Clone the repository to your local machine. Create a `rmarkdown` document in your repository folder with the title `AS3` and output format pdf. Save your file with the name `AS3.Rmd` in your local GitHub repository. **Sign this pdf on page 1 and put it in your repository in pdf, png or jpg format.** When you are done with this assignment, push your solution file `AS3.Rmd` and the signed pdf to the remote GitHub repository within the assignment deadline. Check the remote repository to see if it worked. If the solution file and the pdf show up in your remote GitHub repository, you are done with this assignment. It is also advisable to push your solution file after each subtask or after each day you work on it. That way, you get used to the GitHub workflow and you can make sure everything works. (1 Bonuspoint)

- (a) You are on a hill with your friends hiking. You want to reach the closest valley with a pub as soon as possible as it is getting dark. The fastest way to get down is to examine the ground and walk in the direction of the steepest descent. You brought a fancy new robot on your hike that is able to examine the slope to find the steepest descent and you want to try it out. The robot is able to evaluate the derivative of  $f(x)$ , here denoted by  $df(x)$  when prompted, but is not able to do anything else. The hill you are on is almost exactly approximated by this function

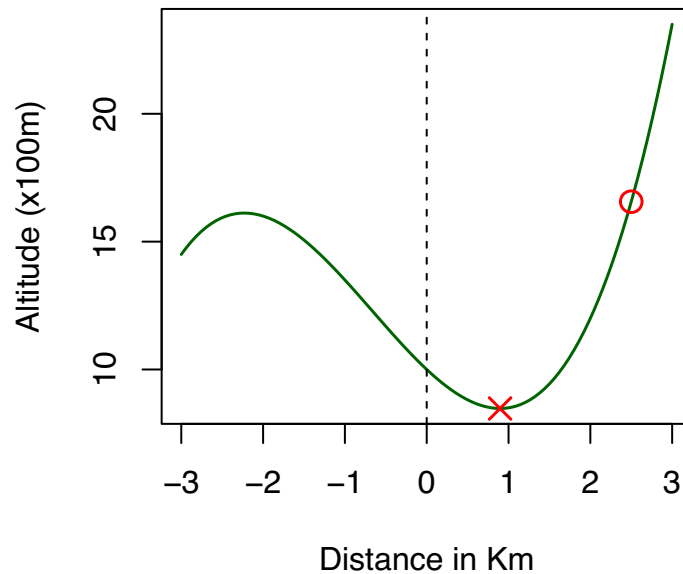
$$f(x) = 0.5x^3 + x^2 - 3x + 10 \quad (1)$$

Add a code chunk to your `rmarkdown` file and define the function  $f(x)$  in R. Calculate the analytical derivative  $df(x)$  using pen and paper and define it in R as well.

- (b) Now that you have defined the above functions  $f(x)$  and  $df(x)$  using R Code, include them as an equation in your rendered `Rmd` file as well. They should look like equation 1 for  $f(x)$  in subtask (a) (*Hint: See for example here how to include math in Rmarkdown files*).
- (c) Replicate the plot shown below of the function from subtask (a). Note: Here the plot is created using base R plot functions, but you can use `ggplot2` if you want.

The red dot is the location of your friends, the robot and you in  $(2.5, f(2.5))$ , the red cross marks the closest pub at the lowest value of  $f(x)$ . Replicate the plot as exactly as possible. It does not matter if you do not have the exact color of green or if the font size of the axis labels is not the same, however it should replicate the main features of the plot.

## Hiking with friends



- (d) You use your robot to find the direction and magnitude of the steepest descent and go down 10 steps in that direction starting from the location  $(2.5, f(2.5))$ . The step size is 1 meter and it is multiplied by the value of  $df(x)$  in each step. Keep in mind that the scale of the plot is in kilometers. Using your functions from subtask (a), write a **for** loop that calculates your 10 steps and **report your location after those steps**. Recreate your plot from subtask (c) and add the current location to the plot. If your implementation is correct, you should be in location  $(2.4662, f(2.4662) = 16.183)$  after 3 steps. Use this information to sanity-check your code.
- (e) You decide to increase the step size to 250 meters. That means you use your robot again to find the direction of the steepest descent and walk in that direction for 250 meters multiplied by the value of  $df(x)$  without checking the surroundings and without consulting the robot. What is your location after 1 step (250 meters) starting from your original starting position in  $(2.5, f(2.5))$ ? What happened and why is it a problem for you and your friends?
- (f) Using all the information above, write a function **downhill** that repeatedly takes steps in the direction of the closest valley starting in **x\_start** using a step size of 100 meters. Your function stops when the slope at that point is close to 0 with a tolerance of 0.01. Use the function signature below and - as always - write a documentation of your function. Your function should return a list with the following information: the final location  $x$ , the value of  $f$  in the final location  $f(x)$ , the value of  $df$  in the final location  $df(x)$  and the number of steps  $i$  that were needed to arrive there. Test your function with **x\_start** = 2.5 and **x\_start** = -1. How many steps does it need to arrive at the pub?

```
downhill <- function(f, df, x_start, stepsize, tol = 0.01){  
  # to do  
}
```

- (g) BONUSPOINT QUESTION (1 Point): What happens when you insert `x.start = -3` into your function from subtask (f) and why? What would be your suggestion for fixing the problem in your function? **It is important to not include the code** for this in your `rmarkdown` document. If you know the answer, you know why. **Answer this Bonuspoint question with text only.**