Technische Universität Dortmund
Fakultät Statistik
Lehrstuhl Computergestützte Statistik

Uwe Ligges
Marieke Stolte
Luca Sauer
Rudi Zulauf

Übung zur Vorlesung

Computergestützte Statistik

Wintersemester 2022/2023

Übungsblatt Nr. 4

Abgabe ist Dienstag, der 08.11.2022 bis 08:00 Uhr im Moodle

Ankündigung: Ab der Korrektur dieses Übungszettels werden wir Verstöße gegen unere Programmierhinweise mit bis zu 0.5 Punkten Abzug bestrafen. Weiterhin werden wir Abgaben, die nicht fehlerfrei sourcebar sind, ebenfalls mit 0.5 Punkten Abzug versehen. Jede Funktion, die Kern einer (Teil-) Aufgabe ist, ist zu dokumentieren und zu testen. Ab sofort wird es für das Fehlen dieser Punktabzüge geben, auch wenn die Aufgabenstellung nicht explizit danach fragt.

Kurz: Ab sofort ziehen wir Punkte für Missachtung der Programmierhinweise ab.

Aufgabe 1 (4 Punkte)

Um Ihnen die Programmierhinweise noch mal im Detail in Erinnerung zu rufen, wenden Sie diese auf den R-Code in der Datei `schlechter_R_Code.R` an. Finden Sie heraus, was die beiden Funktionen in der Datei berechnen. Pro Funktion können 2 Punkte erreicht werden. Gehen Sie dazu wie folgt vor:

a) Kopieren Sie die Funktionen in Ihre Abgabe.

b) Formatieren Sie den Quellcode entsprechend den bekannten Regeln neu. Achten Sie insbesondere auf sinnvolle Variablennamen.

c) Versuchen Sie anhand der auf die Eingabeparameter angewendeten Funktionen / Operationen zu erraten, welchen Typ und welche Struktur diese haben. Wenn auf den Parameter `a` der Funktion beispielsweise die Funktion `ncol` angewendet wird, dann ist `a` mit hoher Wahrscheinlichkeit eine Matrix oder ein Dataframe. Dokumentieren Sie Ihre Vermutung, indem Sie die Funktion mit einem Dokumentationskopf versehen.

d) Beheben Sie mögliche Effizienz-Probleme der Funktion. (Die Funktionen sind teilweise sehr *dumm* und ineffizient implementiert, dies soll behoben werden.)

e) Versuchen Sie den implementierten Algorithmus zu erkennen. Führen Sie hierzu die Funktion ggf. mit verschiedenen Eingaben aus und beobachten Sie die Rückgabewerte.

Aufgabe 2 (4 Punkte)

Betrachten Sie quadratische Funktionen der Art $g(x, a) = a_2x^2 + a_1x + a_0$. Seien weiterhin x^* und a_i^* die entsprechenden Repräsentation der Zahlen im Computer mit relativen Fehlern δ_x bzw. δ_{a_i}. Schätzen Sie den relativen Fehler ab, der bei der Auswertung der Funktion gemacht wird. Für welche Werte von a, x können große Fehler auftreten? Geben Sie ein passendes Beispiel in `R` an.

Aufgabe 3 **(4 Punkte)**

a) (2 Punkte) Finden Sie Gleitkommazahlen u und v ($b = 10,\ p = 4$), so dass gilt:

$$(u \otimes u) \ominus (v \otimes v) \neq (u \ominus v) \otimes (u \oplus v).$$

Versuchen Sie dabei insbesondere Zahlen zu finden, so dass die Differenz zwischen den beiden Ausdrücken möglichst groß ist. Begründen Sie ihr Vorgehen.

b) Auch einfachste Aufgaben wie das Summieren von Zahlen können auf dem Rechner schief gehen, wenn die Zahlen nur *schlecht* genug sind. Kann man das Summieren von n Zahlen auch so intelligent machen, dass nichts schief geht?

i) (0.5 Punkte) Implementieren Sie eine Funktion `naiveSum`, die die Summe von n Zahlen bestimmt, indem Sie diese händisch in einer Schleife aufsummieren.

ii) (0.5 Punkte) Testen Sie Ihre Funktion auf Ihre numerische Genauigkeit. Erzeugen Sie sich dazu Vektoren beliebiger Länge mit beliebig großen Zahlen, deren exakte Summe Sie kennen. **Tipp**: Analytisch gilt: `sum(sample(c(x, - x , c))) = c` für beliebige Vektoren x und beliebige Konstanten c. Ist ihre Funktion *exakt*? Vergleichen Sie Ihre Ergebnisse mit denen der `sum`-Funktion.

iii) (1 Punkt) Überlegen Sie sich eine Möglichkeit, ihre Funktion zu verbessern. Implementieren Sie diese `smartSum` Funktion und vergleichen Sie sie mit dem naiven Ansatz und der Funktion `sum`. Welche Funktion würden Sie in der Praxis benutzen und warum?