Übungsblatt 4: Datenanalysen in R

Wissenschaftliches Schreiben und Arbeiten, WiSe 2023/'24

Aufgabe A: Datentypen

1. Erstelle ein neues R-Skript `aufgaben_ABC_<Vorname>_<Nachname>.R`, wobei du die Platzhalter `<Vorname>` und `<Nachname>` durch deinen Vor- und Nachnamen ersetzen solltest.
   (Alle Teilaufgaben von Aufgaben A, B und C müssen in diesem R-Skript erledigt werden.)
2. Definiere eine Variable `a` und weise ihr den Zahlenwert der Division von 5 durch 2 zu.
3. Definiere eine Variable `b` und weise ihr 12 als strikt ganzzahligen Wert zu.
4. Überprüfe die Datentypen von `a` und `b` mithilfe des `class(...)`-Befehls.
   (Stelle sicher, dass `a` und `b` unterschiedliche Datentypen haben.)
5. Definiere eine Variable `c` und weise ihr den Textwert `"-105.5"` zu.
6. Konvertiere den Wert von `c` von einem Text- in einen Zahlenwert, indem du einen Befehl der Form `as.<Ziel-Datentyp>(...)` verwendest. Weise das Ergebnis der Konversion wieder der Variablen `c` zu.
7. Definiere eine Variable `d` und weise ihr das Produkt von `a`, `b` und `c` zu.
8. Überprüfe mit `print(...)` den Wert von `d` und mit `class(...)` den Datentyp von `d`.
   (Hinweis: Es sollten `-3165` und `"numeric"`, d. h. Zahlenwert, herauskommen.)
9. Definiere eine Variable `e` und weise ihr den Wahrheitswert `FALSE` zu. Überprüfe anschließend den Datentyp von `e` mithilfe des `class(...)`-Befehls.
10. Konvertiere den Wert von `e` von einem Wahrheits- in einen Textwert, indem du einen Befehl der Form `as.<Ziel-Datentyp>(...)` verwendest. Weise das Ergebnis der Konversion wieder der Variablen `e` zu. Überprüfe anschließend den Datentyp von `e` mithilfe des `class(...)`-Befehls.

Aufgabe B: Homogene Datenstrukturen

1. Definiere eine Variable `vec` und weise ihr einen numerischen Vektor mit den acht Elementen `12`, `-3`, `4`, `0.5`, `100`, `-1`, `6` und `-3.5` zu.
2. Du kannst übrigens mathematische Operationen anstatt nur auf eine einzelne Zahl auch auf Vektoren von Zahlen anwenden!
   Probiere das aus, indem du der Variable `vec` nun den Ausdruck `2 * vec - 4` zuweist.
   Gib den Inhalt des Vektors `vec` nun mit dem `print(...)`-Befehl aus und vergewissere dich, dass du nachvollziehen kannst, wie die neuen darin enthaltenen Werte zustande kommen.
3. Definiere eine Variable `mat` und weise ihr eine Matrix mit vier Reihen und zwei Spalten zu, wobei die erste Spalte die ersten vier Elemente des Vektors `vec` enthält und die zweite Spalte die letzten vier Elemente des Vektors `vec` enthält. Prüfe anschließend die Struktur der erstellten Matrix, indem du den `print(...)`-Befehl benutzt.
4. Definiere eine Variable `arr` und weise ihr ein Array der Dimensionen $2 \times 2 \times 2$ zu, wobei das Array als Elemente die acht Werte aus dem Vektor `vec` enthalten soll, und gib sie dann mit `print(...)` aus.
5. Berechne die Summe des 5. Elements von Vektor `vec` und desjenigen Elements der Matrix `mat`, das sich in ihrer 3. Reihe sowie ihrer 1. Spalte befindet. Weise den resultierenden Zahlenwert einer neu definierten Variablen mit dem Variablennamen `zweihundert` zu.

Aufgabe C: Heterogene Datenstrukturen

1. Erstelle eine simple Liste (ohne Beschriftungen der Elemente), die die Textwerte `"Kartoffeln"`, `"Zwiebeln"`, `"Öl"`, `"Essig"`, `"Senf"`, `"Petersilie"` sowie die Zahlenwerte `17`, `-5`, `8` enthält, und speichere diese in einer neu definierten Variablen namens `zutaten_und_zahlen`.
2. Definiere eine Variable `potsdam` und weise ihr eine beschriftete Liste zu, die Folgendes enthält:
   - mit Label `Stadtname` den Textwert `"Potsdam"`,
   - mit Label `Einwohnerzahl` den Zahlenwert `185750`,
   - mit Label `Quadratkilometer` den Zahlenwert `188.24`,
   - mit Label `Bundesland` den Textwert `"Brandenburg"`,
   - mit Label `Landeshauptstadt` den Wahrheitswert `TRUE`,
   - mit Label `Bundeshauptstadt` den Wahrheitswert `FALSE`,
   - schließlich mit Label `Bezirke` einen Vektor, der die Textwerte `"Potsdam Nord"`, `"Nördliche Vorstädte"`, `"Westliche Vorstädte"`, `"Innenstadt"`, `"Babelsberg"`, `"Potsdam Süd"`, `"Potsdam Südost"` und `"Nördliche Ortsteile"` umfasst.
3. Wähle das 6. Element der Liste `zutaten_und_zahlen` aus und lass es mithilfe des `print(...)`-Befehls ausgeben.
   (Hinweis: Achte darauf, dass beim Auswählen von Elementen aus Listen oder Datenrahmen doppelte eckige Klammern `[[` bzw. `]]` um den Positionsindex gesetzt werden müssen.)
4. Wähle das 3. Element der Liste `potsdam` aus und lass es mithilfe des `print(...)`-Befehls ausgeben.
5. Wähle das Element der Liste `potsdam` aus, das das Label `Quadratkilometer` trägt, und lass es per `print(...)` ausgeben.
6. Wähle das Element der Liste `potsdam` aus, das das Label `Bezirke` trägt (es ist ein Vektor) und wähle aus diesem Vektor das 5. Element aus. Lass dir das Ergebnis per `print(...)` ausgeben.
7. Erstelle einen Datenrahmen `potsdamer_bezirke`, der zwei Spalten hat:
   - die erste Spalte mit Label `Name` soll den Inhalt des Vektors `potsdam$Bezirke` enthalten;
   - die zweite Spalte mit Label `AnzahlOrtsteile` soll den Inhalt eines Vektors enthalten, der die Zahlenwerte `7`, `3`, `2`, `3`, `3`, `5`, `3` und `6` umfasst.
8. Berechne den Durchschnittswert der Spalte `AnzahlOrtsteile` im Datenrahmen `potsdamer_bezirke` und speichere das Ergebnis in einer neu definierten Variablen `Durchschnitt_AnzahlOrtsteile`.
9. Erstelle einen Datenrahmen `potsdamer_bezirke_mehr_als_3_ortsteile`, der genau so strukturiert ist wie `potsdamer_bezirke`, aber lediglich diejenigen Zeilen enthält, für die in Spalte `AnzahlOrtsteil` ein Zahlenwert vorliegt, der größer als 3 ist.
   (Hinweis: Du kannst hierfür zunächst das `dplyr`-Paket laden und anschließend einen Ausdruck der Form `<ursprünglicherDatenrahmen> %>% filter(<Wahrheitsbedingung>)` verwenden.)

Aufgabe D: Datenanalyse und Bericht

1. Lade dir die Tabellendatei `L1vsL2SprecherLesezeiten.csv` aus dem Kurs-Moodle herunter.
   (Sie enthält die Ergebnisse eines **fiktiven** Lesezeit-Experiments mit 300 Teilnehmenden.)
2. Erstelle ein neues R-Skript `aufgabe_D_<Vorname>_<Nachname>.R`, wobei du die Platzhalter `<Vorname>` und `<Nachname>` durch deinen Vor- und Nachnamen ersetzen solltest.
   (Alle folgenden ausgeführten R-Befehle für Aufgabe D dann bitte in diesem R-Skript aufschreiben.)
3. Importiere den Inhalt von `L1vsL2SprecherLesezeiten.csv` in R, indem du den `read.csv(...)`-Befehl benutzt und das Ergebnis in einem neuen Datenrahmen `lesezeiten` speicherst.
   (Hinweis: Achte wie immer darauf, dass sich die Datei im aktuellen Arbeitsverzeichnis von R befindet.)
4. Verschaffe dir mithilfe des `str(...)`-Befehls einen Überblick zur Struktur des Datenrahmens `lesezeiten`.

5. Lass dir mit dem `table(...)`-Befehl eine Frequenztabelle für die Spalte `Muttersprachler` ausgeben.
6. Berechne Minimum, Maximum, Median, Durchschnitt und Varianz der Spalte `AlterJahre`.
7. Berechne Minimum, Maximum, Median, Durchschnitt und Varianz der Spalte `SatzLesezeitMs`.
8. Erstelle unter Verwendung des Pakets `ggplot2` einen Box-Plot, dessen x-Achse die zwei Kategorien der Spalte `Muttersprachler` unterscheidet (d. h. `"ja"` vs. `"nein"`) und dessen y-Achse die Zahlenwerte aus der Spalte `SatzLesezeitMs` repräsentiert. Zusätzlich soll die Kategorieunterscheidung `Muttersprachler "ja"/"nein"` auch durch unterschiedliche Füllfarben hervorgehoben werden.
   Der Plot soll den Titel „Satz-Lesezeiten von Mutter- vs. Nichtmuttersprachlern“ tragen; außerdem soll die y-Achse die Beschriftung „Satz-Lesezeit in ms“ haben.
   (Hinweis: Du kannst einen solchen Box-Plot ganz nach Blaupause des **Codes für den auf Folie 78 der letzten Vorlesung gezeigten Box-Plot** erstellen.)
9. Speichere den erstellten Plot als PDF-Grafik, indem du in RStudio rechts unten im Tab Plots auf „Export“ klickst, dann „Save as PDF. . . “, dann bei *PDF Size:* die Größe $\underline{5.20} \times \underline{4.00}$ *inches* auswählst.
10. Schreibe ein kleines LaTeX-Dokument, in dem du in einfachem Text (entweder stichpunktartig oder in kurzen Sätzen) die Ergebnisse der Auswertungen aus den oberen Unteraufgaben 6 und 7 berichtest, d. h. Minimum/Maximum/Median/Durchschnitt/Varianz von `AlterJahre` und `SatzLesezeitMs`.
    Außerdem soll in diesem LaTeX-Dokument der in Unteraufgabe 9 exportierte Box-Plot als eingebettete Abbildung erscheinen.
    (Hinweis: Mit dem LateX-Befehl `\includegraphics[width=...]{<Dateiname>}`) lassen sich in PDF-Format vorliegende Grafiken genauso gut einbetten wie etwa auch PNG-Bilddateien.)
11. Kompiliere dieses LaTeX-Dokument und speichere es als `bericht1_<Vorname>_<Nachname>.pdf`.
12. Ändere jetzt den Code im R-Skript `aufgabe_D_<Vorname>_<Nachname>.R` so, dass unmittelbar nach dem Importieren des Datensatzes `lesezeiten` aus der Datei `L1vsL2SprecherLesezeiten.csv` (Unteraufgabe 1) ein Befehl hinzugefügt wird, der alle Beobachtungen aus dem Datenrahmen `lesezeiten` entfernt, für die der Wert von `AlterJahre` kleiner als 18 oder größer als 27 ist.
    Der so reduzierte Datenrahmen soll wieder der Variablen `lesezeiten` zugewiesen werden.
13. Führe dann alle folgenden Code-Zeilen (Unteraufgaben 4–8) erneut aus und schaue dir an, wie die Ergebnisse diesmal aussehen. Exportiere auch den so neu generierten Box-Plot, wieder wie in Unteraufgabe 9 beschrieben, als PDF-Grafik.
14. Erstelle nun ein neues LaTeX-Dokument, ganz nach dem Schema des alten LaTeX-Dokuments aus Unteraufgabe 10, das aber jetzt die Ergebnisse der neuen Datenanalyse auf der reduzierten Version des Datensatzes `lesezeiten` präsentieren soll.
15. Kompiliere das neue LaTeX-Dokument und speichere es als `bericht2_<Vorname>_<Nachname>.pdf`.

Fertig! Wie du vielleicht gemerkt hast, war der Workflow der Unteraufgaben 13–15 etwas mühsam: Du musstest nach bloß einer kleinen Änderung im R-Code alle relevanten Auswertungen erneut generieren lassen und dann händisch wieder Stück für Stück in ein (neues) LaTeX-Dokument übertragen . . .
Gäbe es da nicht eine bessere Lösung? Könnte man den Workflow der Übertragung erneuerter Analyse-Ergebnisse in ein Dokument nicht irgendwie automatisieren, um keine Zeit zu verschwenden?
Ja, gibt es! Stichwort: dynamische Dokumente. Und noch ein Stichwort: R Markdown.
Was genau es damit auf sich hat, das schauen wir uns ab nächster Woche einmal genauer an :)

Abgabe:

Lade bis zum 14.12.23 um 23:59 Uhr folgende Dateien in der Abgabemaske im Kurs-Moodle hoch:

- `aufgaben_ABC_<Vorname>_<Nachname>.R` (in deinem letzten Bearbeitungsstand)
- `aufgabe_D_<Vorname>_<Nachname>.R` (in deinem letzten Bearbeitungsstand)
- `bericht1_<Vorname>_<Nachname>.pdf`
- `bericht2_<Vorname>_<Nachname>.pdf`