

Übungsblatt 1

Einführung in R
Institut der deutschen Wirtschaft

Pekka Sagner

25. Oktober & 11. November 2022

Erste Schritte mit R und R-Studio

Aufgabe 1: Arbeiten mit Projekten

Starten Sie RStudio auf Ihrem Computer. Falls noch nicht geschehen, erstellen Sie ein neues RStudio-Projekt (wählen Sie in der Menüleiste **File** → **New Project...**). Erstellen Sie das Projekt in einem neuen Verzeichnis, indem Sie **New Directory** auswählen und dann **New Project** auswählen. Geben Sie dem Projekt im Feld **Directory name:** einen passenden Namen (evtl. *einfuhrung_in_r*) und wählen Sie aus, wo Sie dieses Projektverzeichnis erstellen möchten, indem Sie auf die Schaltfläche **Browse** klicken. Erstellen Sie abschließend das Projekt, indem Sie auf die Schaltfläche **Create Project** klicken.

Weitere Informationen zu RStudio-Projekten finden Sie beispielsweise hier:

- [RStudio Projects and Working Directories: A Beginner's Guide](#)
- [RStudio: Using Projects](#)

Aufgabe 2: Erstellen und Aufbau eines Skripts

- Erstellen Sie nun ein neues R-Skript in diesem Projekt, indem Sie im Hauptmenü **File** → **New File** → **R-Script** auswählen (oder den Shortcut verwenden). Bevor Sie mit dem Schreiben von Code beginnen, speichern Sie dieses Skript, indem Sie im Hauptmenü **File** → **Save** auswählen (oder den Shortcut verwenden). Nennen Sie dieses Skript *01_uebung* oder ähnlich. Klicken Sie auf die Registerkarte (Tab) **Files** im unteren rechten RStudio-Bereich, um zu sehen, ob Ihre Datei am richtigen Ort gespeichert wurde.
- Am Anfang fast jedes R-Skripts (es gibt nur sehr wenige Ausnahmen!) sollten Sie einige Informationen einfügen, damit Ihre Mitarbeiter (und Sie) in Zukunft wissen, wer das Skript geschrieben hat, wann es geschrieben wurde und was das Skript tut (unter anderem). Fügen Sie diese Informationen oben in Ihr R-Skript ein und stellen Sie sicher, dass Sie ein **#** am Anfang jeder Zeile platzieren, um R wissen zu lassen, dass es sich um einen Kommentar handelt.

Aufgabe 3: Die vier Hauptfenster in RStudio

Erkunden Sie RStudio und stellen Sie sicher, dass Sie die Funktionen jedes der vier Fenster verstehen. Nehmen Sie sich Zeit und sehen Sie sich die einzelnen Registerkarten in den Fenstern an. Die Funktion einiger dieser Registerkarten wird offensichtlich sein, während andere im Moment nicht nützlich sind. Im Allgemeinen schreiben Sie Ihren R-Code im Skripteditorfenster (normalerweise oberes linkes Fenster) und »schicken« Ihren Code dann in die R-Konsole (normalerweise unten links), indem Sie mit der Maus auf

eine beliebige Stelle in der entsprechenden Codezeile klicken und dann auf die Schaltfläche **Run** oben im Skripteditorfenster klicken. Wenn Sie nicht gerne auf Schaltflächen klicken (wie die meisten R-Nutzer), dann können Sie den Shortcut verwenden.

Aufgabe 4: Wichtige Grundeinstellungen

- a) Bevor es weitergeht, sollten einige der Standardeinstellungen in RStudio angepasst werden. Lesen Sie hierzu zunächst [R4DS, Kapitel 8.1 \(Wickham/Grolemund, 2021\)](#). Welche Einstellungsänderungen werden empfohlen?
- b) Die empfohlenen Einstellungen können über die Menüleiste via **Tools -> Global Options -> General** erreicht werden. Wenden Sie die empfohlenen Einstellungen an.
- c) Erkunden Sie außerdem die weiteren Einstellungsmöglichkeiten. Experimentieren Sie mit verschiedenen Editor-Themes. Das Layout der vier Fenster kann ebenfalls angepasst werden. Es ist empfohlen, dies erst später zu tun, um dem Kursverlauf zunächst besser folgen zu können.

Aufgabe 5: R als Taschenrechner

- a) Führen Sie folgende Rechenoperationen nacheinander in der Konsole aus:
 - i) $1 + 2$
 - ii) Multiplizieren Sie das Ergebnis aus i) mit 3.
 - iii) Ziehen Sie 6 vom Ergebnis auf ii) ab.
 - iv) Teilen Sie das Ergebnis aus iii) durch 4.
 - v) Berechnen Sie den natürlichen Logarithmus des Ergebnis aus iv).
- b) Öffnen Sie ein neues R-Skript und führen Sie die Rechenoperationen aus a) im Skript aus.
- c) R erlaubt uns das Speichern von Objekten. Alles in R ist ein Objekt. Objekte werden mit dem *assignment-operator* »gespeichert«. Definieren Sie einen Vektor `vektor_1` als $(2, 9, -4, 0)$. Betrachten Sie den Vektor `vektor_1` in der Konsole.
- d) Für die Beantwortung von Aufgabenteil c) nutzen wir die Funktion `c()`. Informieren Sie sich mittels der Hilfefunktion von RStudio, was diese Funktion bedeutet.
- e) Base-R bietet viele weitere nützliche Funktionen. Berechnen Sie (i) das arithmetische Mittel, (ii) das Minimum, (iii) das Maximum und (iv) den Median von `vektor_1`. Weisen Sie den Ergebnissen jeweils einen Namen zu.
- f) Definieren Sie nun einen Vektor `v_2` als $(2, 9, NA, -4, 0)$. Führen Sie die Berechnungen aus Aufgabenteil e) für Vektor `v_2` durch. Was fällt Ihnen auf? Wofür steht `NA`?

Aufgabe 6: Logische Operatoren

Mit logischen Operatoren führen wir logische Tests durch - anders ausgedrückt: wir prüfen ob eine Bedingung erfüllt ist.

- a) Erstellen Sie die folgenden Skalare und weisen Sie ihnen jeweils einen Objektnamen zu:

$$3, \pi, e, \log(10)$$

- b) Führen Sie mindestens 5 verschiedene logische Tests mit den vier Skalaren durch. Mindestens vier sollten das Ergebnis `TRUE` liefern, mindestens vier sollten das Ergebnis `FALSE` liefern. Benutzen Sie mindestens vier verschiedene logische Operatoren und alle Skalare mindestens ein Mal.

Aufgabe 7: Datentypen

Skalare und Vektoren bestehen aus verschiedenen Datentypen. Ein Grundverständnis dieser Datentypen ist für die Arbeit mit R unerlässlich.

- a) Verbinden Sie die Skalare aus Aufgabe 7 zu einem Vektor.
- b) Welchen Datentyp hat dieser Vektor?
- c) Erweitern Sie den Vektor um einen *character*-Skalar. Welchen Datentyp hat der neue Vektor jetzt?
- d) Erstellen Sie jeweils mindestens einen Vektor vom Typ **double**, **character**, **logical**. Die Vektoren sollten mindestens die Länge vier haben. (*Hinweis: Die Länge eines Vektors prüfen Sie mit der Funktion `length()`.*) Prüfen Sie den Datentyp der erstellten Vektoren, um sich zu vergewissern, ob Sie richtig liegen.
- e) Wandeln Sie einen Ihrer Vektoren in einen **factor**-Vektor um. Prüfen Sie den Datentyp und die Datenklasse.
- f) Erstellen Sie einen **factor**-Vektor (*diverse, female, male*). Ändern Sie die Reihenfolge der Faktorlevel, sodass **female**, vor **male** vor **diverse** steht (*Hinweis: nur bei den Levels, nicht in der Reihenfolge im Vektor.*).