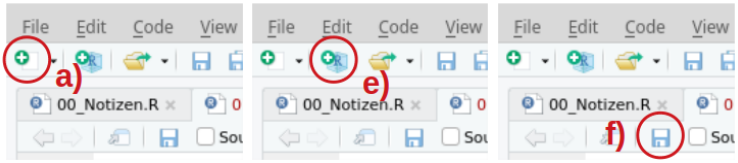
Base R Arbeitsblatt

Statistik und Programmieren mit R - Campus 2019

13.08.2019

1 Lege in RStudio ein Projekt an und speichere darin ein Skript

a) Klicke hierfür in RStudio auf das blaue Symbol oben links
b) Wähle "New Directory"
c) Wähle "New Project"
d) Gib in das erste Feld ein, wie das Projekt heißen soll, und in das zweite Feld, wo es auf deinem Computer angelegt werden soll. Klicke auf "Create Project". RStudio wechselt jetzt zu deinem neuen Verzeichnis.
e) Um ein neues Skript anzulegen, klicke auf das weiße Symbol oben links. Wähle "R Script" aus.
f) Um das Skript im Projektverzeichnis zu speichern, klicke auf das Diskettensymbol im Skriptbereich. Löse die Aufgaben in diesem Skript, sodass du den Code speichern kannst.



2 Vektoren

Löse diese Aufgaben mit Hilfe des Kastens "Vectors" auf dem Base R Cheatsheet. Oft gibt es mehrere mögliche Lösungswege.

a) Erzeugt folgende Vektoren. Gib ihnen die Namen a-e (für die späteren Teilaufgaben).

- a: 1 2 3 4 5 6 7
- b: 2 5 3 9 12
- c: 1 1 1 2 2 2 3 3 3
- d: 5 6 7 5 6 7
- e: 2 4 6 8 10

b) Sortiere den Vektor b.
c) Erzeuge einen Vektor, der die Werte des Vektors c jeweils nur einmal enthält.
d) Wähle das 2. und 3. Element des Vektors e aus.
e) Wähle alle Elemente außer dem dritten und fünften des Vektors d aus.
f) Wähle alle Elemente des Vektors a aus, die größer als 4 sind.

3 Rechnen mit Vektoren

Benutze für diese Aufgaben den Kasten "Math Functions" auf dem Base R Cheatsheet.

a) Welches ist das größte Element in Vektor e?
b) Was ist der Median des Vektors a?
c) Berechne die Summe der natürlichen Logarithmen aller Elemente des Vektors b
d) Berechne den Logarithmus des zweiten Elements von c und runde den Wert auf zwei Nachkommastellen

4 Datentypen

Bisher hast du mit einem bestimmten Datentyp gearbeitet: Numerische Daten ("numeric"). Numerische Daten sind reelle Zahlen, man kann mit ihnen rechnen. Unterkategorien von numerischen Daten sind integer (nur ganze Zahlen) und double (auch Zahlen mit Nachkommastellen). Außerdem gibt es noch einige andere Datentypen, die wichtigsten sind character und logical.

Datentyp	Beispiele	Beschreibung
character	"a", "23", "Hallo Welt"	Zeichenfolgen. Immer durch Anführungszeichen gekennzeichnet.
logical	TRUE, FALSE	
numeric	2, -4, 0.34	Entweder integer (ganze Zahlen) oder double (mit Nachkommastellen).

Innerhalb eines Vektors müssen alle Elemente den gleichen Datentyp haben. R "entscheidet" selber, als welchen Datentyp es den Vektor interpretiert ("coercion").

a) Erzeuge folgende Vektoren:

- f: TRUE, FALSE, TRUE
- g: "Apfel", "Birne", "Mango"
- h: 5, 6, "Obstsalat"
- i: 3, 5, FALSE, TRUE
- j: "A", FALSE, TRUE

b) Mit den Funktionen typeof() und is.character(), is.logical() und is.numeric() kann man den Datentyp abfragen. Probiert die Funktionen an den eben erzeugten Vektoren aus. Nach welchen Regeln werden Datentypen angepasst, um einen einheitlichen Vektor zu erzeugen?

c) Datentypen können mit den Funktionen as.numeric(), as.logical() und as.character() ineinander umgewandelt werden. Mache...

- Vektor f zu einem numerischen und zu einem character Vektor
- Vektor i zu einem character und zu einem logischen Vektor
- Vektor h zu einem numerischen und einem logischen Vektor

d) Fehlende Werte werden in R durch NA ("not available") angegeben. Probiere folgende Code selber aus:

```
x <- NA
# Mit is.na() kann abgefragt werden, ob ein Wert NA ist
is.na(x)
# Innerhalb eines Vektors können einzelne Werte (oder alle Werte) NA sein
y <- c(5, 6, NA, 7, NA)
is.na(y)
# Wenn man versucht, mit NAs zu rechnen, ist das Ergebis ebenfalls NA
y + 5
mean(y)
# Eine Möglichkeit, das Problem zu lösen: vorher die fehlenden Werte herausfiltern
y[!is.na(y)] + 5 # Das Ausrufezeichen bedeutet "not"
# Bei vielen Funktionen, z.B. mean(), gibt es das Argument na.rm = (kurz für na remove),
# mit dem nur nicht fehlende Werte für die Berechnung verwendet werden.
mean(y, na.rm = TRUE)
```

5 Data Frames

Daten werden R meistens in Form von data frames gespeichert, analysiert und modifiziert. Data frames sind zweidimensionale "Tabellen", wobei jede Spalte eine Variable und jede Zeile eine Beobachtung ist. Sie setzen sich aus Vektoren zusammen: Jede Spalte/Variable ist ein Vektor. Die Namen der Vektoren werden zu Namen der Spalten. In einem data frame können die Spalten verschiedene Datentypen beinhalten, innerhalb einer Spalte muss der Datentyp jedoch der selbe sein. Die Spalten müssen gleich lang sein.

a) Probiere folgenden Code aus:

```r
# Siehe auch Base R Cheatsheet - Data Frames
# Data frames kann mit data.frame() erstellen
df1 <- data.frame(farbe = c("gelb", "rot", "blau"), anzahl = 1:3)
df1
# Man kann auch bereits existierende Vektoren zusammefügen
df2 <- data.frame(f, g)
df2
# Aber die Vektoren müssen gleich lang sein
df3 <- data.frame(e, f)
# Mit cbind können data frames zusammengefügt werden
cbind(df1, df2)
# Auswahl von einzelnen Elementen eines data frames:
# Zeile
df[2, ]
# Spalte
df[ ,2]
# Beobachtung
df[2,2]
# Spalte
df$farbe
```

b) Lade den in R vorhandenen Datensatz "iris" mit dem Befehl `data("iris")`[1]. Du kannst jetzt unter dem Namen `iris` auf den Datensatz zugreifen. Benutze das Base R Cheatsheet, um die folgenden Aufgaben zu lösen.

c) Wie viele Zeilen und Spalten hat der Datensatz?

d) Wie lautet die Beobachtung in der dritten Spalte und dreißigsten Zeile?

e) Was ist der Mittelwert der Variable Sepal.Length?

f) Was ist der Datentyp der Variable Sepal.Length?

6 Einlesen von Daten

Meistens erstellt man seine Datensätze nicht in R, sondern in Excel oder einem ähnlichen Programm, und importiert sie dann in R für die weitere Bearbeitung. Das ideale Format dafür ist .csv. Bei deutschen Versionen von Excel muss man eventuell die Spracheinstellungen ändern, damit das Komma als Feldseparator und der Punkt als Dezimalseparator verwendet wird. In meinem Github-Verzeichnis findest du den Datensatz "test.csv". Speichere ihn in deinem Projektordner und lies ihn mit der Funktion read.csv() (siehe Cheatsheet) ein.

7 Exportieren von Daten

Erstelle einen eigenen kleinen Datensatz in R zu einem Thema deiner Wahl und exportiere ihn mit der Funktion write.csv() (siehe Cheatsheet).

[1] Der Datensatz enthält Daten zu drei Arten der Pflanzengattung *Iris* (*setosa*, *versicolor* und *virginica*). Für jede Art wurde an 50 Individuen jeweils die Länge und die Breite der Kelchblätter (sepals) und der Kronblätter (petals) gemessen.

8 Plotting

Erstelle mit dem iris-Datensatz folgende Plots (siehe auch Base R Cheatsheet):

a) Mit der Funktion plot(): einen Scatterplot mit Petal.Length auf der x-Achse und Petal.Width auf der y-Achse
b) Mit der Funktion hist(): Ein Histogramm der Variable Sepal.Length
c) Versuche, folgenden Boxplot nachzubauen. Die benötigte Funktion steht nicht auf dem Cheatsheet, benutze eine Suchmaschine.