

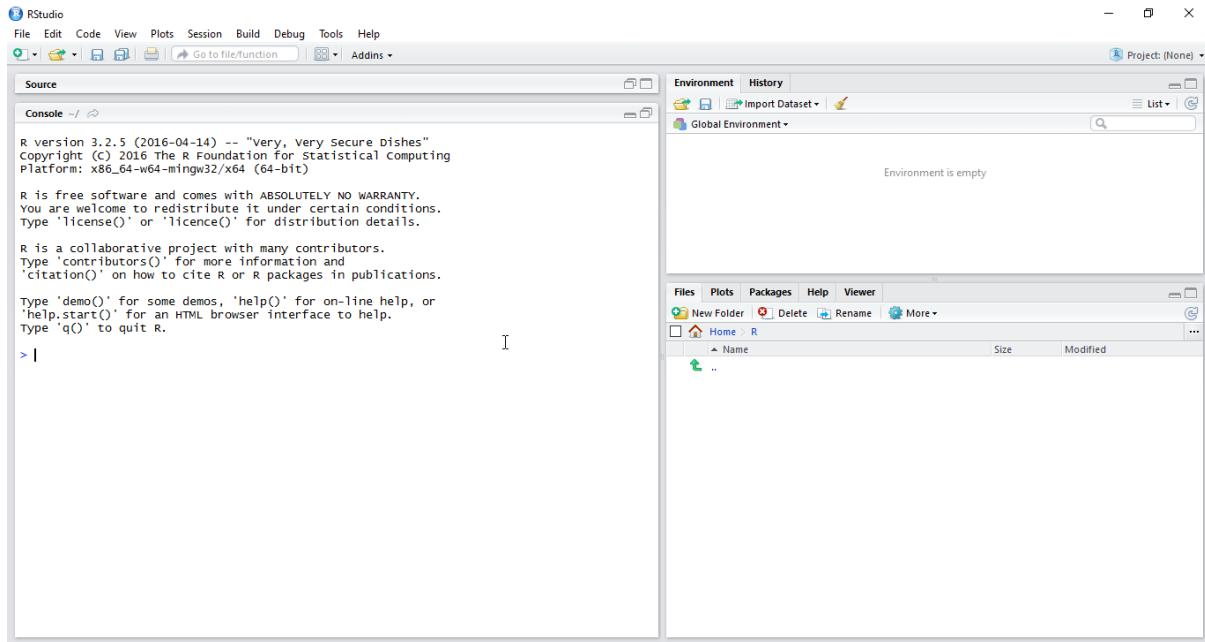
Erste Schritte in R

Karsten Lübke

Hinweise

R ist der Name der Programmiersprache für Statistik und Datenanalyse, **R Studio** ist eine komfortable Entwicklungsumgebung für R.

Nach dem Start von R Studio erscheint folgender Bildschirm



Links, in der *Console* werden die Befehle eingegeben, Rechts oben können Sie z. B. die Daten, aber auch andere Objekte mit denen Sie arbeiten, betrachten, auch die Historie der Befehle wird dort angezeigt. Rechts unten können Sie u. a. Dateien und Abbildungen auswählen, aber auch Hilfeseiten und Tipps betrachten.

Wir werden zunächst in der Konsole arbeiten.

Ein paar Anmerkungen vorweg:

- R unterscheidet zwischen Groß- und Kleinbuchstaben
- R verwendet den Punkt `.` als Dezimaltrennzeichen
- Fehlende Werte werden in R durch `NA` kodiert
- Kommentare werden mit dem Rautezeichen `#` eingeleitet
- R wendet Befehle direkt an
- R ist objektorientiert, d. h. dieselbe Funktion hat evtl. je nach Funktionsargument unterschiedliche Rückgabewerte
- Hilfe zu einem Befehl erhält man über ein vorgestelltes Fragezeichen `?`
- Zusätzliche Funktionalität kann über Zusatzpakete hinzugeladen werden. Diese müssen ggf. zunächst vorher installiert werden
- Mit der Pfeiltaste nach oben können Sie einen vorherigen Befehl wieder aufrufen

R als Taschenrechner

Auch wenn Statistik nicht Mathe ist, so kann man mit R auch rechnen. Geben Sie zum Üben die Befehle in der R Konsole hinter der Eingabeaufforderung `>` ein und beenden Sie die Eingabe mit **Return** bzw. **Enter**.

```
4+2
```

```
## [1] 6
```

Das Ergebnis wird direkt angezeigt. Bei

```
x <- 4+2
```

erscheint zunächst kein Ergebnis. Über `<-` wird der Variable `x` der Wert `4+2` zugewiesen. Wenn Sie jetzt

eingeben wird Ergebnis

```
## [1] 6
```

angezeigt. Sie können jetzt auch mit `x` weiterrechnen.

```
x/4
```

```
## [1] 1.5
```

Vielleicht fragen Sie sich was die `[1]` vor dem Ergebnis bedeutet. R arbeitet vektororientiert, und die `[1]` zeigt an, dass es sich um das erste (und hier auch letzte) Element des Vektors handelt.

R zur Datenanalyse

Wir wollen R aber als Tool zur Datenanalyse verwenden. Daher müssen wir zunächst Daten einlesen. Zunächst laden wir die Daten als `csv` herunter

```
download.file("https://goo.gl/whKjnl", destfile = "tips.csv")
```

R hat die Datei jetzt im lokalen Verzeichnis unter den Namen `tips.csv` gespeichert.

Hier können Sie mehr über die Daten erfahren.

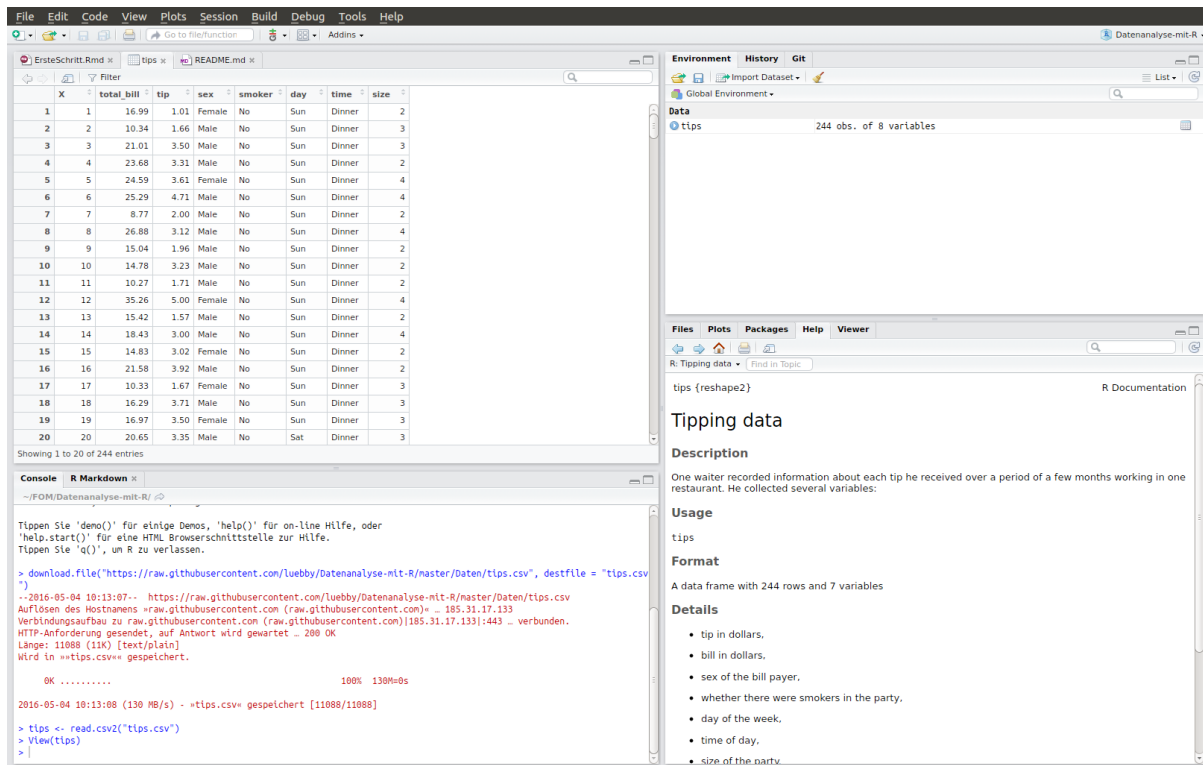
Wo das lokale Verzeichnis ist, können Sie über

```
getwd()
```

erfahren. Das Einlesen von `csv` Dateien erfolgt über

```
tips <- read.csv2("tips.csv")
```

Der Datensatz `tips` taucht jetzt im **Environment** Fenster Rechts oben in R Studio auf. Durch Klicken auf den Namen können Sie diese betrachten.



Erste Analyse des tips Datensatzes

Dieser Datensatz aus

Bryant, P. G. and Smith, M (1995) Practical Data Analysis: Case Studies in Business Statistics. Homewood, IL: Richard D. Irwin Publishing

enthält Trinkgelddaten. Diese sind in tabellarischer Form dargestellt, d. h. üblicherweise, dass die Beobachtungen zeilenweise untereinander stehen, die einzelnen Variablen spaltenweise nebeneinander. In R heißen solche Daten *data frame*. Um einen ersten Überblick über die verschiedenen Variablen zu erhalten geben wir den Befehl `str()` ein:

```
str(tips)
```

```
## 'data.frame':    244 obs. of  7 variables:
## $ total_bill: num  17 10.3 21 23.7 24.6 ...
## $ tip       : num  1.01 1.66 3.5 3.31 3.61 4.71 2 3.12 1.96 3.23 ...
## $ sex       : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 2 2 2 2 2 ...
## $ smoker    : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
## $ day       : Factor w/ 4 levels "Fri","Sat","Sun",...: 3 3 3 3 3 3 3 3 3 3 ...
## $ time      : Factor w/ 2 levels "Dinner","Lunch": 1 1 1 1 1 1 1 1 1 1 ...
## $ size      : int   2 3 3 2 4 4 2 4 2 2 ...
```

Dieser enthält also 244 Zeilen (Beobachtungen) und 7 Spalten (Variablen). Alternativ kann man diese Information auch über

```
dim(tips)
```

erhalten.

Metrische Variablen sind in R in der Regel vom Typ `numeric`, kategoriale Variablen vom Typ `factor`. `str` und `dim` sind erste Befehle, d. h. Funktionen in R, denen in der Klammer das jeweilige Funktionsargument übergeben wird.

```
head(tips) # Obere Zeilen
tail(tips) # Untere Zeilen
```

Ermöglichen ebenfalls einen Einblick über die Daten. Der Befehl

```
names(tips)
```

gibt die Variablennamen zurück. Mit Hilfe des \$ Operators kann auf einzelne Variablen zugegriffen werden:

```
tips$sex
```

erhalten Sie bspw. das Geschlecht des Rechnungszahlers.

Übung: Lassen Sie sich die Rechnungshöhe `total_bill` anzeigen.

mosaic

`mosaic` ist ein Zusatzpaket, welches die Analyse mit R erleichtert. Sofern noch nicht geschehen muss es *einmalig* über

```
install.packages("mosaic")
```

installiert werden.

Um es verwenden zu können muss es für *jede* neue R Sitzung über geladen werden.

Der Grundgedanke von `mosaic` ist die *Modellierung*. In R und insbesondere in `mosaic` wird dafür die Tilde `~` verwendet. `y~x` kann dabei gelesen werden wie “y ist eine Funktion von x”. Beispielsweise um eine Abbildung (Scatterplot) des Trinkgeldes `tip` und Rechnungshöhe `total_bill` zu erhalten muss in R der Befehl eingegeben werden:

```
xyplot(tip ~ total_bill, data=tips)
```

Das Argument `data=tips` stellt klar, aus welchen Datensatz die Variablen kommen. Die Abbildung ist jetzt Rechts unten im *Plots* Fenster zu sehen.

Übung: Wie würden Sie den Trend beschreiben?

Wie oben erwähnt können wir R auch gut als Taschenrechner benutzen, sollten aber bedenken, dass R vektorweise arbeitet. D. h.

```
tips$tip/tips$total_bill
```

gibt für jede Beobachtung die relative Trinkgeldhöhe bezogen auf die Rechnungshöhe an. Über

```
(tips$tip/tips$total_bill)<0.10
```

erhalten wir einen Vektor vom Typ `logical`. Dieser nimmt nur zwei Werte an, nämlich `TRUE` und `FALSE`. Neben `<` und `>` bzw. `<=` und `>=` gibt es ja auch noch das `=`. Hierfür werden in R gleich zwei Gleichheitszeichen verwendet, also `==`.

Übung: Was gibt folgender der Befehl zurück?

```
tips$sex=="Female"
```

Logische Vektoren können mit `und` & oder `oder` | verknüpft werden:

```
tips$sex=="Female" & tips$smoker=="Yes"
```

gibt die Tischgesellschaften wieder, in denen die Rechnung von von Frauen beglichen wurde *und* geraucht wurde,

```
tips$sex=="Female" | tips$smoker=="Yes"
```

gibt die Tischgesellschaften wieder, in denen die Rechnung von von Frauen beglichen wurde *oder* geraucht wurde.

Intern wird TRUE mit der Zahl 1 hinterlegt, FALSE mit 0. Der Befehl `sum()` summiert einen Vektor, also erfahren wir über

```
sum(tips$sex=="Female" & tips$smoker=="Yes")
```

dass bei 33 Tischgesellschaften bei denen geraucht wurde eine Frau die Rechnung bezahlte. Im Verhältnis zu allen Tischgesellschaften, bei denen eine Frau zahlte liegt der Raucheranteil also bei 0.3793103:

```
sum(tips$sex=="Female" & tips$smoker=="Yes") / sum(tips$sex=="Female")
```

Übung: Wurde bei den Tischgesellschaften, bei denen ein Mann zahlte häufiger geraucht?

Übung: Teaching Rating

Dieser Datensatz analysiert u. a. den Zusammenhang zwischen Schönheit und Evaluierungsergebnis:

Hamermesh, D.S., and Parker, A. (2005). Beauty in the Classroom: Instructors' Pulchritude and Putative Pedagogical Productivity. Economics of Education Review, 24, 369–376.

Sie können ihn von <https://goo.gl/6Y3KoK> herunterladen. Hier gibt es eine Beschreibung.

1. Lesen Sie den Datensatz in R ein.
2. Wie viele Zeilen, wie viele Spalten liegen vor?
3. Wie heißen die Variablen?
4. Betrachten Sie visuell den Zusammenhang von dem Evaluierungsergebnis `eval` und Schönheit `beauty`. Was können Sie erkennen?
5. Sind relativ mehr Frauen oder mehr Männer (`gender`) auf einen Tenure Track (`tenure`)?

Daten importieren

Der Datenimport in R ist in vielen unterschiedlichen Dateiformaten möglich. Das `csv` Format eignet sich besonders zum Übertragen von Datendateien. Im deutschsprachigen Raum wird dabei als Dezimaltrennzeichen das Komma `,` und als Datentrennzeichen das Semikolon `;` verwendet. In der ersten Zeile müssen die Variablenamen stehen. Das Einlesen in einen R Data-Frame (hier `meineDaten`) kann dann über

```
meineDaten <- read.csv2(file.choose()) # Datei auswählen
```

erfolgen.

Der Befehl `file.choose()` öffnet dabei den Dateiordner. Bei “internationalen” `csv` Dateien ist das Datentrennzeichen i. d. R. ein Komma `,`, das Dezimaltrennzeichen ein Punkt `.`. Hier funktioniert der Import in R dann über den Befehl `read.csv`.

In R Studio gibt es im Reiter **Environment** im Fenster Rechts oben einen Menüpunkt **Import Dataset** der mehr Einstellmöglichkeiten bietet.

Hier finden Sie eine Linksammlung zu verschiedenen Datenquellen.

Diese Übung basiert teilweise auf Übungen zum Buch OpenIntro von Andrew Bray und Mine Çetinkaya-Rundel unter der Lizenz Creative Commons Attribution-ShareAlike 3.0 Unported.

Versionshinweise:

- Datum erstellt: 2016-05-10
- R Version: 3.3.0
- `mosaic` Version: 0.13.0