

Programmieren mit statistischer Software

Eva Endres, M.Sc.

Institut für Statistik

Ludwig-Maximilians-Universität München

Grundbegriffe



Danksagung

Herzlichen Dank an Manuel Eugster, Cornelia Oberhauser und Moritz Berger für das Überlassen ihrer Materialien!

R ist ein Kommandozeileninterpreter!

- Kommandozeile wartet auf Input
- Symbol für Eingabeaufforderung ist `>`
- Shortcut zum Ausführen eines Befehls in R-Studio:
Strg+Enter
- R evaluiert die Expression und zeigt den Output sofort

```
> 1+1
```

```
[1] 2
```

- Falls Expression nicht vollständig ist, wartet R auf mehr Input und zeigt dies mit `+` an.
- Änderung der globalen Einstellung mit `?options` oder in R-Studio über Tools → Global Options

Objekte I

- Speichern eines Objekts unter einem Namen
- Groß- und Kleinschreibung wird unterschieden
- Namen dürfen nicht mit einer Zahl beginnen
- Alles in R ist ein Objekt

```
> a <- 1+1
```

```
> a
```

```
[1] 2
```

```
> b<-"Hello World"
```

```
> b
```

```
[1] "Hello World"
```

- Anzeigen aller gespeicherten Objekte

```
> ls()
```

```
[1] "a" "b"
```

```
> objects()
```

```
[1] "a" "b"
```

- Entfernen eines Objekts

```
> rm(a)
```

```
> a
```

```
Error in eval(expr, envir, enclos): Objekt 'a' nicht gefunden
```

```
> ls()
```

```
[1] "b"
```

- Entfernen aller Objekte (Alternative zu R schließen und neu starten)

```
> rm(list=ls())
```

Workspace I

- Working Directory

```
> setwd("Laufwerk:/Pfad")  
> setwd("S:/ProgStat")  
> setwd("S:\\ProgStat")  
> # setwd("S:\\ProgStat") funktioniert nicht
```

- Anzeigen des aktuellen Arbeitsverzeichnisses

```
> getwd()
```

- Speichern und Laden des gesamten Workspace

```
> save.image()  
> load(".RData")
```

- Speichern einzelner Objekte

```
> save()
```

- Speichern und Laden im Arbeitsverzeichnis oder Unterordnern

```
> x <- 1:10  
> save(x, file="./Ergebnisse/jargon.RData")  
> load("./Ergebnisse/jargon.RData")
```

- Speichern und Laden in übergeordnetem Ordner (in Bezug auf das Arbeitsverzeichnis)
 - > `setwd("../Ergebnisse")`
 - > `save(x, file="../jargon.RData")`
 - > `load("../jargon.RData")`

- Zwei Möglichkeiten, die Hilfe aufzurufen: `help()` oder `?`
 - > `help(ls)`
 - > `?ls`
- Hilfe für Keywords (z.B.: `if`, `while`)
 - Keywords müssen in Anführungszeichen gesetzt werden
 - > `"if"`
 - > `# ?if funktioniert nicht`
 - > `"while"`
- Beispiele auf der Hilfeseite ausführen
 - > `example("ls")`
- Suche nach bestimmten Schlagworten, falls die explizite Funktion nicht bekannt
 - > `help.search("linear models")`
 - > `help.search("nonparametric")`
 - > `??"subset"`
- Weiteres hilfreiches Paket: `sos`

- Alles in R sind Objekte (Groß- und Kleinschreibung relevant)

```
> x <-1
```

```
> x
```

```
[1] 1
```

```
> X <- "a"
```

```
> X
```

```
[1] "a"
```

1 und 'a' sind die Werte der Objekte

- Funktionen

- Funktion besteht aus Funktionsname und Argumenten

```
f(argument,...)
```

- Jede Funktion hat einen Rückgabewert

```
> identity(1:10)
```

```
[1] 1 2 3 4 5 6 7 8 9 10
```

- Ausgabe des Quellcodes ohne Klammern

```
> identity
```

```
> table
```

- Mathematische Operatoren

- `> 1 + 1`

- `[1] 2`

- `> -1`

- `[1] -1`

- `> 2*4`

- `[1] 8`

- `> 12/4`

- `[1] 3`

- `> '+'(1,1)`

- `[1] 2`

- Übersicht über Operatoren: ?Syntax

- Vektoren

- ```
> x <- 4.2
> is.vector(x)
[1] TRUE
> length(x)
[1] 1
```

- Ermöglichen vektorisierte Berechnungen

```
> x <- c(4.2, 13, 5)
> cos(x)
[1] -0.4902608 0.9074468 0.2836622
> sin(x)
[1] -0.8715758 0.4201670 -0.9589243
> x+1
[1] 5.2 14.0 6.0
```

- Matrizen

- Initialisierung von Matrizen

```
> matrix(c(1,0,0,1), nrow=2, ncol=2)
```

```
 [,1] [,2]
[1,] 1 0
[2,] 0 1
```

```
> # analog
```

```
> diag(2)
```

```
 [,1] [,2]
[1,] 1 0
[2,] 0 1
```

- Mathematische Operationen mit Matrizen

```
> diag(2) + diag(2)
```

```
 [,1] [,2]
[1,] 2 0
[2,] 0 2
```

- Elementweise multiplizieren

```
> 2*diag(2)*diag(2)
```

```
 [,1] [,2]
[1,] 2 0
[2,] 0 2
```

- Matrizenprodukt

```
> (diag(2)*2) %*% (diag(2)*4)
```

```
 [,1] [,2]
[1,] 8 0
[2,] 0 8
```

- Kommentare

```
> 1+1
[1] 2
> # 2+2
```

- In R-Studio können über Code -> Comment/Uncomment Lines ganze Abschnitte auskommentiert werden

# Packages I

---

- Alle Funktionen und Datensätze sind in Paketen gespeichert
- Nur wenn das Paket geladen ist, sind die Funktionen verfügbar
- Welche Pakete sind installiert?  
> `library()`
- Welche Pakete sind geladen?  
> `search()`
- Laden zusätzlicher Pakete  
> `library("tools")`  
> `require("sos")`
- Geladene Pakete entfernen  
> `detach("package:tools")`
- Pakete installieren  
> `install.packages("archetypes")`

# Packages II

---

- Update aller bereits installierten Pakete  
> `update.packages()`
- Herunterladen von Pakete ohne Installation  
> `download.packages()`
- Welches Repository wird verwendet? (Plattform auf der Pakete zur Verfügung gestellt werden, z.B. CRAN, Bioconductor)  
> `getOption("repos")`
- Aktueller Pfad, wohin die Pakete installiert werden (falls mehrere Pfade vorhanden sind, wird der erste verwendet)  
> `.libPaths()`
- Überblick über Inhalte der Pakete  
> `library(help = "graphics")`  
> `library(help = "archetypes")`

# Packages III

- Übersicht und Laden von Daten eines Pakets

```
> data(package = "archetypes")
> data("body", package = "archetypes")
> body
```

- Übersicht und Ausführen von Demos eines Pakets

```
> demo(package = "archetypes")
> demo("robust-ozone", package = "archetypes")
```

- Quellcode der Demo

```
> edit(file = system.file("demo", "robust-ozone.R",
+ package = "archetypes"))
```

- Übersicht und Laden von Vignetten eines Pakets

```
> vignette(package = "archetypes")
> vignette("archetypes", package = "archetypes")
> # Folgendes funktioniert in R, aber nicht in RStudio
> edit(vignette("archetypes", package = "archetypes"))
> # so funktioniert es in RStudio
> edit(vignette("archetypes", package = "archetypes"), editor="internal")
```

- Folgende Befehle müssen in der Programm-Datei 'basic-1.R' gespeichert werden ...

```
> a <- 2:8
> b <- pi
> c <- a * b
> save(c, file = "res.RData")
> load("res.RData")
```
- ... und können dann aus einer anderen Programm-Datei heraus ausgeführt werden

```
> source("basic-1.R")
```