Vorbereitungsseminar für Abschlussarbeiten

Einführung in R

Thomas Kneib

SoSe 2012

1 Die statistische Software R

- Open Source Statistik-Software.
- Kompilierte Versionen für alle gängigen Betriebssysteme und Source-Code erhältlich unter

  `http://www.r-project.org`

- Kombiniert
  - umfangreiche statistische Methoden,
  - flexible Grafikerstellung,
  - vollständige Programmierumgebung.
- Benutzer können Ihre eigenen Erweiterungs-Pakete (Packages) schreiben und diese im Comprehensive R Archive Network (CRAN) zugänglich machen.
- Lingua Franca der Statistik und (zumindest im akademischen Bereich) dominierend.

- Bedienung (fast) ausschließlich über die Kommandozeile.
- R ist case-sensitive.
- Hilfe:
  - Für spezielle Funktionen: `help("`*function*`")`, `?`*function*.
  - Volltextsuche: `help.search("`*text*`")`, `??"`*text*".
  - Hilfe-Startseite: `help.start()`.
  - Suche auf `http://www.r-project.org`
  - "`See Also`"-Abschnitte ähnlicher Hilfeseiten.

- Editoren:
  - Tinn-R für Windows (`http://www.sciviews.org/Tinn-R/`).
  - RStudio für Windows, Mac, Linux (`http://www.rstudio.org/`).
  - Notepad++ für Windows (`http://notepad-plus-plus.org/`)
  - JGR für Windows, Mac, Linux (`http://rforge.net/JGR/`).
  - Emacs Speaks Statistics (EES) für Emacs, Xemacs, etc.
  - Allgemeine Übersicht: `http://www.sciviews.org/_rgui/`.

2 Grundlagen

- Kommentare beginnen mit #
- R ist auch ein großer Taschenrechner.
  - Operatoren: `+`, `-`, `*`, `/`, `^`
  - `help("+")`
  - R hält die üblichen Regeln zur Punkt- vor Strichrechnung ein.
  - Klammerung von Ausdrücken über `(...)`.
  - Ganzzahlige Division: %/%
  - Rest der ganzzahligen Division: %%

- Neben den Grundrechenarten gibt es zahlreiche mathematische Funktionen, z.B.
  - `exp`, `log`
  - `sqrt`
  - `sin`, `cos`, `tan`, `atan`, `...`
  - `factorial`, `choose`, `...`
- Vergleich von Werten:
  - `==`, `!=`, `>`, `>=`, `<`, `<=`
  - `all.equal`

- Die Ergebnisse von Berechnungen lassen sich in Variablen abspeichern:
  - `x <- 7`
  - `x = 7`
  - `7 -> x`
  - `print(x)`

- Einfache Datentypen:
  - Logisch: `TRUE`, `FALSE`, `T`, `F`
  - Integer: `1`, `12^3`, `42`
  - Numerisch: `1.0`, `3.1415297`, `NaN`, `Inf`, `-Inf`
  - Zeichenketten: `"test"`, `"Hallo Welt"`
  - Fehlende Werte: `NA`
  - Konstanten: `pi`

- Logische Operatoren: `&`, `|`
- Bestimmung von Datentypen:
  - `mode(x)`
  - `is.numeric(x)`
  - `is.character(x)`

- Arbeiten mit Zeichenketten:
  - `paste`
  - `cat`
  - `substr`
  - `strsplit`

3 Vektoren

- Vektoren kombinieren Elemente des gleichen Typs.
- Erstellung z.B. über die Funktion c (combine):
  - `x <- c(5,2,8,99)`
  - `x <- c(x,3,4)`
  - `y <- c("A","B")`
  - `z <- c(x,y)`
- Eigenschaften eines Vektors:
  - Länge: `length(x)`
  - Typ: `mode(x)`

- Sequenzen: `seq(from=, to=, by=)`
  - `seq(from=0, to=10, by=2)`
  - `seq(from=0, to=10)`
  - `seq(to=10)`
  - `1:10`

- Wiederholungen: `rep(times=, length=, each=)`
  - `rep(0, times=5)`
  - `rep(1:3, times=5)`
  - `rep(1:3, length=5)`
  - `rep(1:3, each=5)`

- Zugriff auf Elemente:
  - Angabe der interessierenden Elemente: `x[c(1,5,8)]`, `x[1:5]`
  - Ausschluss von nicht interessierenden Elementen: `x[-1]`, `x[-c(2,4,5)]`
  - Angabe einer Bedingung: `x[x>4]`, `x[x<5 & x%%2==0]`
  - Komplexe Verschachtelungen möglich
- Auffinden spezieller Elemente:
  - `which(x<3 & x%%2==0)`

- Rechnen mit Vektoren:
  - R rechnet elementweise mit Vektoren: `x+1`, `x*y`,
  - Ordnen und sortieren: `order(x)`, `sort(x)`

4 Matrizen

- Erstellen von Matrizen:
  - `m <- matrix(1:12, ncol=4, nrow=3)`
  - `m <- matrix(1:12, ncol=4, nrow=3, byrow=T)`
- Zugriff auf Elemente:
  - Einzelne Elemente: `m[1,2]`
  - Spalten: `m[,1]`
  - Zeilen: `m[2,]`
  - Kombinationen: `m[2:3,-4]`

- Eigenschaften einer Matrix:
  - Dimensonen: `dim(m)`
  - Typ: `mode(m)`
- Erstellung über die Kombination von Vektoren:
  - Verbinden als Spalten: `m <- cbind(x,y)`
  - Verbinden als Zeilen: `m <- rbind(x,y)`

- Rechnen mit Matrizen:
  - Transponieren: `t(m)`
  - Matrixmultiplikation (bei passenden Dimensionen): `m1%*%m2`
  - Inversion: `solve(m)`
  - Lösen von Gleichungssystemen: `solve(m, rhs)`
  - Determinante: `det(m)`
  - Rang: `qr(m)$rank`
  - Definition von Diagonalmatrizen mit den Werten aus dem Vektor `x` auf der Diagonalen: `diag(x)`
  - Extraktion der Diagonalen: `diag(m)`
- Erweiterung von Matrizen auf höhere Dimensionen: `array`

5 Listen

- Erzeugen von Listen:
  - `l <- list(vector=1:5, char="ABC", mat=matrix(1:4,ncol=2,nrow=2))`
  - `names(l)`
- Zugriff auf die Elemente der Liste:
  - Per Name: `l$char`
  - Per Index: `l[[2]]`

6 Data Frames

- Data frames sind die typische Möglichkeit um Datensätze abzuspeichern.
- Zwei Sichtweisen auf Data Frames:
  - Matrix mit zusätzlichen Attributen und Eigenschaften.
  - Liste in der alle Elemente die gleiche Länge haben müssen.
- Erzeugen von Data Frames:
  - `d <- data.frame(v1=c(1,2,3), v2=rep(1,3))`

- Einlesen von Datensätzen:
  - `d <- read.table(file, header = FALSE, sep = "", dec = ".")`
  - `d <- read.csv(file, header = TRUE, sep = ",", dec=".")`
  - `foreign`-Paket: `read.dta`, `read.spss`,
- Speichern von Datensätzen:
  - `write.table(x, file, quote = TRUE, sep = " ", eol = "\n", row.names = TRUE, col.names = TRUE)`

- Anschauen von Datensätzen:
  - `fix(d)`
  - `head(d, n=10)`
  - `names(d)`
- Zugriff auf Variablen:
  - Per Namen: `d$spaltenname`
  - Per Index (Liste): `d[[2]]`
  - Per Index (Matrix): `d[,2]`

- Faktorvariablen sollten auch als Faktoren gespeichert werden:
  - Umwandeln in einen Faktor: `factor`, `as.factor`
  - Umkodieren eines Faktors: `recode` (aus dem Paket `car`)

7 Grafiken

- Darstellung von diskreten Häufigkeitsverteilungen:
  - Balkendiagramm: `barplot`
  - Kuchendiagramm: `pie`
- Darstellung stetiger Häufigkeitsverteilungen:
  - Histogramm: `hist`
  - Kerndichteschätzung: `density`
  - Boxplot: `boxplot`

- Generische Plot-Funktion: `plot`
  - `type`: Typ der Grafik (Punkte, Liniensegmente, Stufenfunktion, . . .)
  - `pch`: Plot-Charakter
  - `main`: Überschrift
  - `xlab`, `ylab`: Beschriftung der x- und y-Achse
  - `xlim`, `ylim`: Wertebereich der x- und y-Achse
  - `cex`, `cex.axis`, `cex.lab`, `cex.main`, . . . : Faktor zur Vergrößerung / Verkleinerung der Schriftgröße relativ zum Default.
  - `col`, `col.axis`, `col.lab`, `col.main`, . . . : Farbe der Beschriftung.
  - `lty`, `lwd`: Typ und Breite der Linien

- Ergänzungen zu einer bestehenden Grafik:
  - `lines`
  - `segments`
  - `points`
  - `text`
  - `arrows`
  - `abline`
  - `rect`
  - `polygon`
  - `axis`
  - `box`
- Option `xpd=TRUE` erlaubt das ergänzen außerhalb des plot-Bereichs.

- `par`: Mit Hilfe von `par` werden global Einstellungen für Grafiken definiert.
  - `mfrow`: Aufteilung des Grafik-Fensters in verschiedene Teilgrafiken.
  - `las`: Ausrichtung der Label-Beschriftungen an den Achsen.
  - Alternative zu `mfrow` mit mehr Optionen: `layout`.
- Eine `plot`-Funktion existiert für viele verschiedene R-Objekte (z.B. Ergebnisse eines linearen Modells).

- Darstellung von Oberflächen:
  - `image`
  - `contour`
  - `persp`

- Abspeichern von Grafiken:
  - Über das Menü (`File -> Save As`)
  - Per Befehl: `pdf`, `postscript`, `jpeg`, `bmp`, ...

8 Praktische Hinweise zum Schluss

- Die Installation in ein Verzeichnis ohne Leerzeichen kann bestimmte Probleme vermeiden.
- Code am Besten in einem Editor entwickeln und abspeichern, um spätere Reproduzierbarkeit zu sichern.
- Code dokumentieren!
- Mit Hilfe von Sweave ist eine direkte Verbindung von R und LaTeXmöglich (aber nicht Inhalt dieses Kurses).