R Einführung für W-Theorie

Professor Antony Unwin

Ziele

- Etwas über R lernen
- R Konzepte verstehen
- Finden in R was Sie brauchen
- Einfache Modelle in R berechnen

What is R? (1)

- R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes
  - an effective data handling and storage facility,
  - a suite of operators for calculations on arrays, in particular matrices,
  - a large, coherent, integrated collection of intermediate tools for data analysis,
  - graphical facilities for data analysis and display either on-screen or on hardcopy, and
  - A well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

What is R? (2)

- an *interface* to computational procedures of many kinds;
- *interactive*, hands-on in real time;
- *functional* in its model of programming;
- *object-oriented*, "everything is an object";
- *modular*, built from standardized pieces; and,
- *collaborative*, a world-wide, open-source effort.

John Chambers

R herunterladen

- R läuft unter
  - Windows
  - Unix
  - Macintosh
- http://cran.r-project.org/

R als Rechner*

```
> 72387*12545
[1] 908094915
```

Etwas komplizierter

```
> log(73)-20*pi
[1] -58.54139
```

Aber es wird im Output gerundet (nicht im Objekt)

```
> 723.87*125.45
[1] 90809.5
```

*Deshalb

```
> options(digits=15)
> 723.87*125.45
[1] 90809.4915
```

Einige Grundkonzepte in R

- (Fast) alles ist ein Objekt
- Vektororientiertes Rechnen
- Wie in einer Sprache, können Befehle kombiniert werden
- Funktion Voreinstellungen sind oft ausgezeichnet, manchmal trügerisch, gelegentlich schädlich

Datenobjekte

- *Lists* sammeln Objekte verschiedener Typen
- *Vectors* sind vom einem Typ (numeric, logical, string)
- *Data.frames* Listen von Vektoren, alle derselben Länge n
- *Matrices* sind 2-d Daten
- *Arrays* sind m-d Daten (eine Matrix ist eine 2-d Array)
- *Factors* sind Vektoren von kategoriellen Variablen

Verschiedene Objekte besitzen verschiedene Eigenschaften. Verschiedene Operationen können gelten.

Vektoren

- `x <- c(1,2,3,7)`
- `x <- c(2,4,c(1,2,3,7))`
- `x <- rep(0,10)`
- `x <- seq(1,12,0.5)`

Und Rechnen mit Vektoren

- `y <- x^2`
- `w <- y - 2*log10(x)`
- `x <- seq(0,47,1)`
- `sum(log10(x+1)*dbinom(x,47,0.3))`

Zugang zu Daten

- `w[7]`
- `w[2:4]`
- `w[c(4,8:20)]`
- `w[seq(2,10,2)]`
- `w[-c((1:6),9)]`
- `w1 <- w[w>100]`

Überprüfen mit

`summary, head, length, dim, plot,...`

Kombinatorik

- Fakultät
  - `factorial(n)`
- Kombinationen
  - `choose(n,k)`
- Listen von Permutationen und Kombinationen
  - (aus Paket *gtools*)
  - permutations(n, r, v=1:n)
  - combinations(n, r, v=1:n)
  - combinations(3, 2, letters[1:3], repeats=TRUE)

Verteilungen

- Viele Verteilungen werden in R angeboten:
  - Binomial, Poisson, Hypergeometrische, …
- Es gibt vier Befehle für alle
  - d Dichte oder $P(X = x)$
  - p Wahrscheinlichkeit oder $P(X \leq x)$
  - q Quantile für p oder $\min_q P(X \leq q) \geq p$
  - r Zufallszahlen d.h. eine zufällige Stichprobe aus der Verteilung

Verteilungen (Beispiele)

```
> dbinom(4,47,0.3)
> dbinom(0:10,47,0.3)
> pbinom(10,47,0.3)
> pbinom(10,47,0.3,lower=FALSE)
> qbinom(0.05,47,0.3)
> rbinom(20,47,0.3)
```

Statistische Funktionen

- `mean(), median(), sd()`
- `min(), max(), range()`
- `var(), cov(), cor()`
- `length(), dim()`
- `quantile(), rank(), cusum()`
- `scale()` ...

Einige nützliche Funktionen

- `names, summary, table, plot`
- `apply, tapply, by, lapply, aggregate`
- `order, subset, sample, reshape, cbind, rbind`
- `is.na, unique, duplicated, cut, levels, which`
- `attributes, str`
- `if, for, repeat, while` (besser, Vektoren zu verwenden)
- `q()`

Notation

`<-`	Assignment
`;`	command separator, `{}` code block
`[ ]`	array index, `[[ ]]` list index
`$`	(dataset)`$`variable
`&`	and (`&&` shortened and)
`|`	or (`||` shortened or)
`==`	exact equality
`!`	not
`" "`	string
`#`	comments

Graphiken in R

Graphiken in R

- Standard Graphiken
  - Histogramm, Boxplot
  - Streudiagramm
  - Säulendiagramm (und ~~Tortendiagramm~~)
- Plotfunktionen
- Gruppen von Plots
- Pakete (*vcd, grid, ggplot2, lattice, iplots*)

Histogramme

```
hist(w)

#als Dichte
hist(w,freq=FALSE)

#Welche Klassengrenzen?
h1<-hist(w,freq=FALSE)
h1$breaks

#Klassengrenzen bestimmen
hist(w,freq=FALSE,breaks=seq(-25,175,25))
```

Streudiagramme

```
#Mit dem Datensatz diamonds
plot(carat,price)

#Kleinere Punkte
plot(carat,price,pch=20)

#Andere Grenzen
…xlim=c(0.7,1.3),ylim=c(0,25000))

#Mit Titel
…,main="Price (in $) against carat")

#Mit horizontaler Linie
abline(h=0)
```

Säulendiagramme

```
> barplot(dpois(0:10,2),names.arg=0:10)

> dev.new()
> barplot(dbinom(0:10,365,2/365),names.arg=0:10)

> v <- rpois(365,2)
> dev.new()
> barplot(table(v),main="Stichprobe mit n=365")
```

Funktionen plotten

```
> curve(x^2-2*log10(x),1,10)

> dev.new()
> curve(x^2-2*log10(x),-1,5)

> dev.new()
> curve(x^2-2*log10(x),-1,5, n=500)
> lines(100*log(x)-x^3)

> dev.new()
> curve(100*log(x)-x^3,1,5, n=500)
> lines(x^2-2*log10(x))
```

Graphik Optionen

- `xlim, ylim` z.B. `xlim=c(0,500)`
- `xlab,ylab,main` z.B. `main="Picture",legend`
- `col`, z.B. `col="red"`
- `pch` (Plotsymbol) z.B. `pch=20`
- `lty` (Linientyp), `lwd` (Linienbreite)
- `abline` (Linien hinzufügen) z.B. `abline(a,b)` oder `abline(v=1)`
- `lines` (Funktionen hinzufügen) z.B. `lines(qchisq(x, 4),x)`

Mehrere plots

- Neues Graphikfenster, z.B. `dev.new(height=600, width=400)`
- Layout `par(mfrow=c(3,1))`
- Graphiken zeichnen, z.B.

```
> barplot(dbinom(0:10,10,0.5),names.arg=0:10)
> barplot(dbinom(0:10,10,0.5),names.arg=0:10,
xlim=c(0,20))
> barplot(dpois(0:10,5),names.arg=0:10,
xlim=c(0,20))
```

R Pakete

Contributed Packages

Available Packages

Currently, the CRAN package repository features 3398 available packages.

Table of available packages, sorted by date of publication

Table of available packages, sorted by name

Installation of Packages

Please type `help("INSTALL")` or `help("install.packages")` in R for information on how to install packages from this directory. The manual R Installation and Administration (also contained in the R base sources) explains the process in detail.

CRAN Task Views allow you to browse packages by topic and provide tools to automatically install all packages for special areas of interest. Currently, 28 views are available.

Benutzung von Paketen

- Ein Paket finden, das das macht, was man will (CRAN, Task Views, Search, …)
- Webseite des Pakets überfliegen
- Installieren und laden
- Beispiele laufen lassen, Syntax untersuchen
- Vignetten lesen (falls es welche gibt)
- Ergebnisse prüfen (Sind sie sinnvoll? Plotten)

Paket Beispiele

- `chron` für Zeitrechnung

```
> julian (7,24,1988)
> day.of.week (12,25,2009)
> unlist(month.day.year(-3000))
```

- `gtools` für Kombinationen
- `maps` für Karten
- `tm` für Text Mining (z.B. Konkordanz berechnen)
- `actuar` für Versicherungsmathematik

Pakete können gefährlich sein

- wegen
  - Fehler, numerischer Genauigkeit, Zuverläßigkeit
  - Inkonsistenz bei Befehlen und Optionen
  - Überschreiben von Befehlen
  - Behandlung von Spezialfällen
  - Codequalität, Dokumentation, Hilfe

Pakete können großartig sein

- weil sie liefern
  - bessere Funktionen (z.B. *amap* für Clustering)
  - Zugang zu modernen Methoden (z.B. *mda*, *lars*, *glasso*: Hastie, Tibshirani, Friedman)
  - mathematische Funktionen (z.B. *partitions*, *orthopolynom*)
  - Wahrscheinlichkeitsverteilungen (z.B. *actuar*, *evd*, *mvtnorm*)
  - spezielle Modelle (z.B. *geoR*, *spatstat*, *TSA*)
  - Funktionen für Anwendungen (z.B. *tm*, *fOptions*, *genetics*, *tuneR*)

Help (?) Beispiele

- hist (first example)

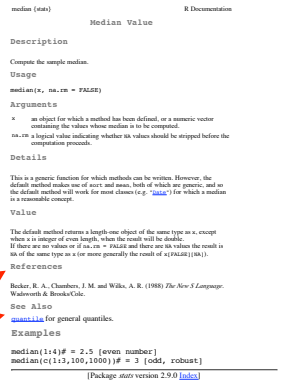
```
op <- par(mfrow=c(2, 2))
hist(islands)
utils::str(hist(islands, col="gray", labels = TRUE))
```

- fda command in mda (Hastie & Tibshirani) with port to R by Leisch, Hornik, Ripley.

  "This software is not well-tested, we would like to hear of any bugs."

Vewrwendung von Help

- ? (topic)
  - Description
  - Usage
  - Arguments
  - Details
  - Value
  - References, Author(s)
  - See Also
  - Examples



median {stats} R Documentation

Median Value

Description

Compute the sample median.

Usage

median(x, na.rm = FALSE)

Arguments

x an object for which a method has been defined, or a numeric vector containing the values whose median is to be computed.

na.rm a logical value indicating whether NA values should be stripped before the computation proceeds.

Details

This is a generic function for which methods can be written. However, the default method makes use of sort and mean, both of which are generic, and so the default method will work for most classes (e.g. "Date") for which a median is a reasonable concept.

Value

The default method returns a length-one object of the same type as x, except when x is integer of even length, when the result will be double.
If there are no values or if na.rm = FALSE and there are NA values the result is NA of the same type as x (or more generally the result of x[FALSE][NA]).

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

See Also

quantile for general quantiles.

Examples

```
median(1:4)# = 2.5 [even number]
median(c(1:3,100,1000))# = 3 [odd, robust]
```

[Package *stats* version 2.9.0 Index]

Unterstützung

- ? Help
- R's homepage: www.R-project.org
- Cran (cran.r-project.org) und BioConductor
  - Manuals, Task Views, FAQs, Vignettes, Mailing lists, R News (now R Journal), Google
- R Wiki, Rseek
- (N.B. update.packages())

Benutzeroberflächen / Drucken

- Benutzeroberflächen
  - Command-line
  - GUIs (z.B., JGR, Rcmdr, ESS)
- Drucken
  - R Output ist meistens nicht schön und muss bearbeitet werden
  - Graphiken können (generell) als pdf gedruckt werden (wie, hängt von Ihrem Betriebssystem ab)

Gute Praxis in R

- Geben Sie neuen Objekten Namen
- Befehle und Optionen schrittweise testen
- Defaults (Voreinstellungen) kontrollieren
- Ergebnisse prüfen! (Graphiken, Werte anschauen)
- Funktionen kombinieren wie in Sprachen
- Sitzungen annotieren und speichern

Zusammenfassung

- R ist eine Sprache.
- "This is R. There is no if. Only how."
- Experimentieren.
- Fragen, wenn Sie Probleme haben.