

GNU R/ Druckversion

Wikibooks.org

3. Dezember 2012

On the 28th of April 2012 the contents of the English as well as German Wikibooks and Wikipedia projects were licensed under Creative Commons Attribution-ShareAlike 3.0 Unported license. An URI to this license is given in the list of figures on page 251. If this document is a derived work from the contents of one of these projects and the content was still licensed by the project under this license at the time of derivation this document has to be licensed under the same, a similar or a compatible license, as stated in section 4b of the license. The list of contributors is included in chapter Contributors on page 249. The licenses GPL, LGPL and GFDL are included in chapter Licenses on page 255, since this book and/or parts of it may or may not be licensed under one or more of these licenses, and thus require inclusion of these licenses. The licenses of the figures are given in the list of figures on page 251. This PDF was generated by the \LaTeX typesetting software. The \LaTeX source code is included as an attachment (`source.7z.txt`) in this PDF file. To extract the source from the PDF file, we recommend the use of <http://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/utility> or clicking the paper clip attachment symbol on the lower left of your PDF Viewer, selecting `Save Attachment`. After extracting it from the PDF file you have to rename it to `source.7z`. To uncompress the resulting archive we recommend the use of <http://www.7-zip.org/>. The \LaTeX source itself was generated by a program written by Dirk Häjnniger, which is freely available under an open source license from http://de.wikibooks.org/wiki/Benutzer:Dirk_Huenniger/wb2pdf. This distribution also contains a configured version of the `pdflatex` compiler with all necessary packages and fonts needed to compile the \LaTeX source included in this PDF file.

Inhaltsverzeichnis

I.	Einleitung	3
1.	Vorwort	5
1.1.	Vorwort	5
2.	Was ist R?	7
2.1.	Was ist R?	7
2.2.	Warum R?	7
2.3.	Was spricht gegen R?	8
2.4.	Literatur	8
3.	Installation	9
3.1.	Unix / Linux	9
3.2.	Windows	9
3.3.	MacOS	10
3.4.	Weblinks	10
3.5.	Quellen	10
4.	R starten	11
4.1.	R beenden	11
5.	Hilfe	13
5.1.	apropos()	13
5.2.	help()	13
5.3.	help.start()	14
5.4.	RSiteSearch()	14
6.	Die Basispakete	15
6.1.	R Standardpakete	15
7.	Zusatzpakete	17
7.1.	Installation per install.packages	17
7.2.	Fehlermeldung bei Mac OS X	17
7.3.	Installation von Zusatzpaketen über die Shell	18
7.4.	Zusatzpakete aktivieren und deaktivieren	18
7.5.	Zusatzpakete aktualisieren	18
7.6.	Übersicht über installierte Zusatzpakete	19
7.7.	siehe auch	19
II.	R benutzen	21
8.	Rechnen mit R	23
8.1.	Beispiele	23
8.2.	Wurzel	24
8.3.	Kommastellen	24
8.4.	Runden	25
8.5.	Integrieren	26
9.	Daten in R	27
9.1.	Variablen	27

9.2.	Einfache Datentypen (<i>modes</i>)	28
9.3.	Komplexe Datentypen: Klassen	29
9.4.	Zeichenketten	30
9.5.	Vektoren	30
10.	Manuelle Dateneingabe	33
10.1.	Eingabe von logischen Werten:	33
11.	Umgang mit Datensätzen (Erstellen, Auswählen und Filtern)	35
12.	Datenimport und -export	37
12.1.	Import	37
12.2.	Export	43
13.	Daten laden und speichern	45
14.	Daten laden	47
15.	Daten speichern	49
15.1.	Arbeitsspeicher speichern	49
15.2.	einzelne Objekte speichern	49
16.	Arbeitsverzeichnis	51
17.	Daten selektieren	53
18.	Daten auswählen	55
19.	Matrix- und Listenoperationen	57
19.1.	Matrix- und Listenoperationen	57
20.	Benutzeroberflächen und Erweiterungen	59
21.	Graphische Benutzeroberflächen	61
21.1.	JGR (Jaguar)	61
21.2.	R Commander	61
21.3.	RKward	61
21.4.	Sciviews-R	62
21.5.	RStudio	62
22.	Sonstige Schnittstellen	63
22.1.	R in Emacs: ESS	63
23.	Automation und dynamische Dokumenterstellung	65
24.	Batchmode	67
25.	Dynamische Dokumenterstellung mit dem Paket (S-/odf-)Weave	69
25.1.	Grundprinzipien dynamischer Dokumenterstellung	69
25.2.	R und Open Office	70
25.3.	R und Latex	71
25.4.	Dynamische Webseiten mit R	71
25.5.	Ausblick	71
25.6.	Weblinks	72
III.	Statistik mit R	73
26.	Diagramme und Grafiken erzeugen	75
26.1.	Funktionen zum Erstellen von Diagrammen	75
26.2.	Funktionen zum Erstellen zusätzlicher grafischer Elemente	76
26.3.	Grafiken speichern	76
26.4.	siehe auch	76
26.5.	Literatur	76
26.6.	Weblinks	76
27.	Deskriptive Statistik	79
27.1.	Gruppenspezifische Auswertung von Variablen	82

28.	Regressionsanalyse	85
28.1.	Einleitung	85
28.2.	Lineare Regression	85
29.	Clusteranalyse	89
30.	Signifikanztests	91
30.1.	Testauswahl	91
30.2.	Testdurchführung in R	93
30.3.	Weblinks	93
31.	Rasch-Modelle	95
32.	Package eRm	97
32.1.	Installation	97
32.2.	Das dichotome logistische Rasch-Modell	97
32.3.	Das linear-logistische Test-Modell	113
32.4.	Das Ratingskalen-Modell	113
32.5.	Das partial-credit Modell	120
32.6.	Das lineare Ratingskalen-Modell	120
32.7.	Das lineare partial-credit Modell	121
32.8.	unterstützte Funktionen im eRm-Package	121
33.	Package ltm	125
34.	Literatur	127
35.	Weblinks	129
36.	Datum- und Zeitfunktionen	131
37.	Einleitung	133
38.	Grundlegende Datumfunktionen	135
39.	Analyse von Zeitreihen	139
40.	Block-Randomisierung	141
40.1.	Einleitung	141
40.2.	blockrand	141
40.3.	siehe auch	142
IV.	Programmieren mit R	143
41.	Programmieren mit R	145
41.1.	Programmieren mit R	145
41.2.	siehe auch	148
42.	Eigene Funktionen programmieren	149
42.1.	R-Sprache	149
42.2.	siehe auch	151
42.3.	Literatur	151
42.4.	Weblinks	151
43.	Dateioperationen	153
43.1.	Dateioperationen	153
44.	Eigene Zusatzpakete erstellen	155
44.1.	Paket erstellen	155
44.2.	Dokumentation schreiben	156
44.3.	Paket überprüfen und fertig stellen	158
44.4.	Paket zum CRAN hochladen	159
44.5.	Weblinks	159
45.	Andere Programmiersprachen einbinden	161
45.1.	C++	161

45.2.	Java	161
46.	Finden von Programmfehlern und Debugging	163
46.1.	Manuelles Einfügen von Ausgabewerten	163
46.2.	Gängige Fehler	163
46.3.	Error-Recovery mit dem Browser	164
46.4.	Verfolgen von Funktionen mit trace	166
V.	Anhang	167
47.	Anwendungsbeispiele	169
48.	Datenbeispiele	171
48.1.	Beispiel 1	171
48.2.	Beispiel 2	171
48.3.	Beispiel 3	172
48.4.	Beispiel 4	172
48.5.	Beispiel 5	173
48.6.	Beispiel 6	173
48.7.	Beispiel 7	178
48.8.	Beispiel 8	179
48.9.	Beispiel 9	181
48.10.	Beispiel 10	184
49.	Graphikbeispiele	187
49.1.	Graphikbeispiel 1	187
49.2.	Graphikbeispiel 2	189
49.3.	Graphikbeispiel 3	190
49.4.	Graphikbeispiel 4	191
49.5.	Graphikbeispiel 5	193
50.	siehe auch	195
51.	Befehle-Index	197
51.1.	A	197
51.2.	B	198
51.3.	C	199
51.4.	D	201
51.5.	E	202
51.6.	F	203
51.7.	G	204
51.8.	H	205
51.9.	I	206
51.10.	J	208
51.11.	K	209
51.12.	L	210
51.13.	M	211
51.14.	N	213
51.15.	O	214
51.16.	P	215
51.17.	Q	217
51.18.	R	218
51.19.	S	219
51.20.	T	222
51.21.	U	223

51.22. V	224
51.23. W	225
51.24. X	226
51.25. Y	227
51.26. Z	228
52. Lösung der Übungsaufgaben	229
53. Umgang mit Datensätzen (Erstellen, Auswählen und Filtern)	231
54. Programmierbeispiele	233
54.1. Beispiel 1: Abschlussnote	233
54.2. Beispiel 2: Cut-Off-Points	233
54.3. Beispiel 3: Entfernen von Umlauten	235
54.4. Beispiel 4: Zeit Sampler	236
54.5. siehe auch	236
55. Beispielskripte	237
55.1. Importieren von SPSS-Datensätzen	237
56. Übersicht programmspezifischer Voreinstellungen	239
57. Das <code>wikibooks</code> -Zusatzpaket	243
57.1. Installation	243
57.2. Paket aktivieren	243
57.3. Paketinhalt	244
57.4. Weblinks	246
57.5. siehe auch	247
58. Autoren	249
Abbildungsverzeichnis	251
59. Licenses	255
59.1. GNU GENERAL PUBLIC LICENSE	255
59.2. GNU Free Documentation License	256
59.3. GNU Lesser General Public License	256

Teil I.

Einleitung

1. Vorwort

1.1. Vorwort

Statistiksoftware gehört heute wie Datenbanksysteme oder Office-Programme zu den entscheidenden Softwarekomponenten in Unternehmen wie auch in der Forschung. Dabei kann die Statistiksoftware versteckt als eine Komponente z.B. im Data Mining vorkommen oder als eigenständige Software, wie R, genutzt werden. Später werden wir noch sehen, dass man R auch in andere Software integrieren kann. Statistiksoftware, wie R, wird heute nicht primär im Kerngebiet der Mathematik angewandt. Vielmehr wird R für die Verarbeitung wissenschaftlicher und ökonomischer Daten verwandt. So ist R heute eine gefragte Software bei der Auswertung genetischer Experimente. Ohne die moderne Computertechnik und entsprechender Software, wäre die moderne Genetik kaum denkbar.

Dieses Buch soll den Leser in einer einfachen Art und Weise und anhand kleiner Beispiele in die Praxis der Software R einführen. Nach dem Lesen und Durcharbeiten des Buchs wird der Leser in der Lage sein, R sicher zu benutzen. Weiterhin werden alle Aspekte von R in diesem Buch angesprochen und erläutert. Dadurch soll es dem erfahren R-Benutzer als stetiges Nachschlagewerk dienen. Für das Verständnis dieses Buches werden **keine** Programmierkenntnisse vorausgesetzt. Weil die statistische Auswertung von Daten die Kernaufgabe von R ist, ist es für den Leser hilfreich, wenn er Erfahrungen auf dem Gebiet der Statistik bzw. deren Anwendung hat.

Tip

Eine Einführung in Statistik bietet das Wikibook Mathematik: _Statistik ¹ .
--

Zur Zeit ist dieses Buch nicht vollständig und auch nicht konsistent. Wenn beim Lesen der Text beim Leser Fragen aufgeworfen werden, wenn Passagen nicht ausreichend verständlich sind oder wenn Dinge fehlen, bitte nutzen Sie die Diskussionsfunktion um uns, die Autoren dieses Buches, darauf hinzuweisen. Anregungen sind gerne willkommen.

¹ http://de.wikibooks.org/wiki/Mathematik%3A_Statistik

2. Was ist R?

2.1. Was ist R?

Dieses Buch führt in das Programm R ein. R steht dabei für *The R Project for Statistical Computing*.

- R ist eine Software zur statistischen Datenverarbeitung, sowie deren grafischer Darstellungen (Visualisierung).
- R ist eine Umsetzung der statistischen Programmiersprache S.
- R läuft unter diversen UNIX-, Linux- und Unix-ähnlichen Betriebssystemen, sowie auf Windows und Mac OS X. Ältere R-Versionen sind auch noch für das klassische Mac OS verfügbar. Für viele Betriebssysteme gibt es bereits kompilierte Pakete.
- Die Sprache kann leicht durch neue Funktionen erweitert werden.
- Eine große Anzahl von existierenden Zusatzpaketen ergänzen die R-Funktionalität um Methoden aus Spezial- und Anwendungsbereichen der Statistik.
- R kann mit anderen Programmiersprachen wie GRASS, Perl, Python, C oder Java verbunden werden.
- R kann sowohl interaktiv, also im Einzelbefehlsmodus, als auch als Skriptsprache und im Batch-Modus verwendet werden.
- Der R-Quelltext wird unter der GNU General Public License (GPL) der Free Software Foundation veröffentlicht.

Die Homepage von R lautet <http://www.r-project.org>

<http://>

Zusätzliche Skripte zur Einführung in R können im Internet kostenfrei abgerufen werden (z.B. Handl 2006¹). Des Weiteren stehen diverse (meist englischsprachige) Internetseiten (z.B. Das R Wiki²) sowie Publikationen (siehe Literatur) zum Thema R zur Verfügung.

2.2. Warum R?

- Neben gängigen Programmen zur statistischen Auswertung, wie beispielsweise „SPSS“ oder „STATA“, bietet R den Vorteil, dass es auf der ganzen Welt kostenlos (unter der freien GNU-Lizenz) zur Verfügung steht.
- R kann die meisten gängigen Formate importieren, gewährleistet volle Kontrolle über die Daten und bietet ein verlässliches, quelloffenes Format für erstellte Datensätze.
- Darüber hinaus stellt R z.T. mächtigere und mehr Auswertungsverfahren zur Verfügung als andere Programme.

1 <http://www2.wiwi.uni-bielefeld.de/~frohn/Mitarbeiter/Handl/stagrund.html>

2 <http://rwiki.sciviews.org/doku.php>

- *R* ist eine Programmierumgebung. Funktionen können bequem den eigenen Bedürfnissen angepasst werden. Komplexe Probleme lassen sich auch dann lösen, wenn die Entwickler diese (noch) nicht implementiert haben.
- *R* wird von der Scientific Community kontinuierlich weiterentwickelt und erweitert. Neue statistische Verfahren werden in der Regel auch in *R* integriert. Ein standardisiertes Pakete-System erleichtert die Nachinstallation ebenso wie die Veröffentlichung eigener Pakete.
- *R* erstellt professionelle Graphiken in einer Vielzahl an Formaten.
- *R* ist auch eine funktionierende Benutzer- und Entwickler-Gemeinschaft, die für Fragen offen ist und so den Einstieg erleichtert.
- *R* kann systemübergreifend auf verschiedenen Plattformen genutzt werden.
- *R* verfügt über äußerst flexible Schnittstellen der Daten-Ein- und -Ausgabe und kann mit verschiedenen anderen Applikationen zusammenarbeiten.
- *R* kann in einem Batch-Mode zur automatisierten Auswertung und Dokumenterstellung verwendet werden und lässt sich auch in Webserver / dynamische Websites integrieren.
- *R* ist trotz graphischer Benutzeroberflächen textbasiert. Im Zentrum steht nicht Design, sondern Funktion.

2.3. Was spricht gegen *R*?

- Für den Anfänger ist die Funktionsweise und Bedienung von *R* zweifellos gewöhnungsbedürftig.
- Bei der Programmierung in *R* fällt im Vergleich mit modernen Sprachen auf, dass Einiges in unerwarteter Weise funktioniert.
- Bestimmte Grundverfahren (z. B. Häufigkeitsgewichte) sind zur Zeit nur umständlich oder gar nicht implementiert.

2.4. Literatur

Mit folgender Literatur haben wir einen guten Einstieg in *R* gefunden:

- Günter Faes (2007): "Einführung in *R*, Ein Kochbuch zur statistischen Datenanalyse mit *R*", BoD, ISBN 9783833491849
- Dubravko Dolic (2003): „Statistik mit *R*. Einführung für Wirtschafts- und Sozialwissenschaftler“, Oldenbourg, ISBN 3486275372

3. Installation

Da R zur freien Verfügung steht, kann eine Installationsdatei aus dem Internet herunter geladen werden. Alle Downloads zu R stehen unter dem „Comprehensive R Archive Network“ (CRAN) zur Verfügung.

Rufen Sie zur Installation die Webseite <http://cran.r-project.org/mirrors.html> auf, und wählen Sie einen Server, von dem aus Sie R herunterladen möchten. Eine gute Downloadverbindung bieten normalerweise solche Server, deren Standort nicht all zu weit von Ihrem Wohnsitz entfernt ist. Nachdem Sie ein Installationspaket für Ihr System (Windows, Apple, Linux) heruntergeladen haben, installieren Sie R. Dies erfolgt in der Regel durch einen Doppelklick auf die Installationsdatei.

3.1. Unix / Linux

- Vorkompiliertes RPM - Paket für die entsprechende Distribution herunterladen
- Paket installieren
- in der Shell mit *R* starten

Für Debian ist auch eine Installation mit *apt-get* möglich. Optional kann die neueste R-Version über einen Backport installiert werden.¹

Für Ubuntu heißt das entsprechende Paket *r-recommended*.

Für Ubuntu bietet das R-Project Repositories an. Diese sind an die jeweiligen Releases angepaßt und bieten bisweilen aktuellere Versionen als Ubuntu².

Für Solaris (OpenSolaris) wird ein Packet auf sunfreeware.com angeboten

3.2. Windows

1. Das Setup-Programm im Paket *base* herunterladen (ca. 23 MB).
2. Dieses Programm ausführen.
3. Die R-Oberfläche unter Windows startet ein Textfenster, das der R-Shell unter Unix entspricht.

¹ <http://cran.r-project.org/bin/linux/debian/> Debian GNU R Repository

² <http://cran.r-project.org/bin/linux/ubuntu/>

3.3. MacOS

- Installer herunterladen und ausführen

3.4. Weblinks

- Anleitung zur Einbindung der Ubunturepositories des R-Projektes³

3.5. Quellen

en:R Programming/Settings⁴ pt:R (linguagem de programação)/Instalação⁵

3 <http://cran.r-project.org/bin/linux/ubuntu/>

4 <http://en.wikibooks.org/wiki/R%20Programming%2FSettings>

5 <http://pt.wikibooks.org/wiki/R%20%28linguagem%20de%20programa%29%2FInstala%29>

4. R starten

Nachdem Sie R installiert haben, starten Sie R mit einem Doppelklick auf das R-Symbol in Ihrem Programm-Verzeichnis.

Unter Linux wird R mit dem Befehl `R` in der Kommandozeile gestartet.

```
R
```

Nachdem Sie R gestartet haben, erscheint ein Begrüßungstext:

```
R : Copyright 2006, The R Foundation for Statistical Computing
Version 2.3.1 (2006-06-01)
ISBN 3-900051-07-0

R ist freie Software und kommt OHNE JEGLICHE GARANTIE.
Sie sind eingeladen, es unter bestimmten Bedingungen weiter zu
verbreiten.
Tippen Sie 'license()' or 'licence()' für Details dazu.

R ist ein Gemeinschaftsprojekt mit vielen Beitragenden.
Tippen Sie 'contributors()' für mehr Information und 'citation()',
um zu erfahren, wie R oder R packages in Publikationen zitiert werden
können.

Tippen Sie 'demo()' für einige Demos, 'help()' für on-line Hilfe,
oder 'help.start()' für eine HTML Browserschnittstelle zur Hilfe.
Tippen Sie 'q()', um R zu verlassen.
>
```

Das Symbol `>` zeigt an, dass der R-Prozessor bereit ist, und Sie nun Ihre Befehle eingeben können. Beispielsweise können Sie direkt den Anmerkungen des Begrüßungstexts folgen und die Lizenzbedingungen von R aufrufen mit:

```
> licence()
```

4.1. R beenden

Zum Beenden gibt man `q()`¹ ein. Es erscheint die Frage:

```
Save workspace image? [y/n/c]:
```

¹ http://de.wikibooks.org/wiki/GNU_R%3A_q

Die Eingabe von *y* (yes) speichert die aktuelle Sitzungskonfiguration (z.B Objekte, Liste der geladenen Bibliotheken). *n* (no) verwirft die aktuellen Sitzungsänderungen, *c* (cancel) bricht den Beendungsprozess ab.

5. Hilfe

Zu jeder R-Funktionsweise gibt es eine R-interne Hilfeseite. Diese kann aufgerufen werden, wenn Sie vor die gefragte Funktion ein `?` schreiben, z.B.:

```
> ?1licence
```

Es öffnet sich nun ein Hilfefenster, in welchem die gefragte Funktion erläutert wird (leider nur auf englisch). In der ersten Zeile der Hilfedatei wird der Name der Funktion angezeigt, gefolgt von dem Paket, in welchem diese vorhanden ist. Des Weiteren werden alle Optionen angezeigt, die mit der Funktion angegeben werden können.

5.1. `apropos()`

`apropos()`² sucht einen Ausdruck in der Liste der Funktionen und Vektorvariablen.

```
3apropos4("plot") </code>
```

liefert eine Liste der Funktionsnamen und Vektoren, welche den Ausdruck **'plot'** enthalten.

5.2. `help()`

Die Funktion **help()** zeigt die selben R-interne Dokumentation zu einem Befehl an, wie das oben beschriebene Fragezeichen (`?`). Beispielsweise liefert

```
help5(plot)
```

die Dokumentation zum Befehl `plot()`⁶. Der Aufruf lässt sich aber wie oben beschrieben abkürzen als

```
7?8plot
```

2 <http://de.wikibooks.org/wiki/GNU%20R%3A%20apropos>

5 <http://de.wikibooks.org/wiki/GNU%20R%3A%20help>

6 <http://de.wikibooks.org/wiki/GNU%20R%3A%20plot>

7 <http://de.wikibooks.org/wiki/GNU%20R%3A%20%3F>

8 <http://de.wikibooks.org/wiki/%3F>

5.3. `help.start()`

Mit `help.start()`⁹ werden im Standardbrowser die Links zu den englischsprachigen R-Anleitungen angezeigt. Sie sind im Installationsverzeichnis von R unter `doc/html/` abgelegt. Enthalten sind unter anderem eine Einführung in R, Datenim- und export, R-Installation und Administration, Referenzen zu den Paketen, sowie eine Suchmaschine mit einem Verzeichnis von Schlagworten.

5.4. `RSiteSearch()`

Die Funktion `RSiteSearch('Suchbegriff')`¹⁰ ist ein einfacher Weg über das Internet mehrere Hilfeseiten auf einmal zu durchsuchen. Im Internetbrowser werden die Treffer aus der Suche in R-Dokumenten, den Hilfeseiten der Funktionen und der R-help-Mailingliste aufgelistet.

⁹ <http://de.wikibooks.org/wiki/GNU%20R%3A%20help.start>

¹⁰ <http://de.wikibooks.org/wiki/GNU%20R%3A%20RSiteSearch%28%29>

6. Die Basispakete

6.1. R Standardpakete

- `base`¹ -- Grundliegende Funktionen
- `graphics`² -- Die wichtigsten Funktionen zum Erstellen von Grafiken
- `stats`³ -- Wichtige Statistikfunktionen
- `utils`⁴ -- Pakete installieren, Hilfsfunktionen, Dateneingabe

1 http://de.wikibooks.org/wiki/GNU_R%3A%20base
2 http://de.wikibooks.org/wiki/GNU_R%3A%20graphics
3 http://de.wikibooks.org/wiki/GNU_R%3A%20stats
4 http://de.wikibooks.org/wiki/GNU_R%3A%20utils

7. Zusatzpakete

Da R ein offenes Projekt ist, an dem jeder mitarbeiten kann, stehen zahlreiche Zusatzpakete mit zusätzlichen Funktionen zur Verfügung.

Sofern Sie über einen Internetanschluss verfügen, können Sie diese Pakete direkt von der R-Kommandozeile nachinstallieren.

7.1. Installation per `install.packages`

Der Befehl hierzu lautet: `install.packages()`¹

Eine Liste aller zur Verfügung stehenden Zusatzpakete finden Sie unter: http://cran.r-project.org/web/packages/available_packages_by_name.html

Wenn Sie beispielsweise das „foreign“-Paket installieren möchten, geben Sie folgenden Befehl ein:

```
> install.packages("foreign")
```

Manche der Zusatzpakete benötigen weitere (abhängige) Zusatzpakete. Diese werden automatisch mitinstalliert, wenn Sie den Befehl mit einer Option erweitert eingeben:

```
> install.packages("foreign", dependencies=TRUE)
```

Sobald Sie den Befehl eingegeben haben, werden Sie aufgefordert, einen Server auszuwählen, von dem aus Sie die Zusatzpakete installieren möchten. Wählen Sie einen Server aus Ihrer Nähe und bestätigen Sie mit „OK“. Sie können den Server mit dem Befehl `chooseCRANmirror()`² wechseln. Dies ist z.B. notwendig, wenn das gewünschte Zusatzpaket auf dem ausgewählten Server nicht verfügbar ist.

Weitere Pakete sind zu finden unter: <http://cran.r-project.org/>

7.2. Fehlermeldung bei Mac OS X

Wenn Sie R von einem Mac aus betreiben und ein Zusatzpaket installieren möchten, bekommen Sie unter Umständen folgende Fehlermeldung:

1 <http://de.wikibooks.org/wiki/GNU%20R%3A%20install.packages>
2 <http://de.wikibooks.org/wiki/GNU%20R%3A%20chooseCRANmirror>

```
"cannot create HTML package index in: make.packages.html() "
```

Dies ist keine echte Fehlermeldung. Das Problem liegt darin, dass die Hilfedateien und -verzeichnisse erst nach dem ersten Aufruf von „Hilfe“ auf Ihrem Rechner angelegt werden. Wählen Sie also aus dem Menü „Hilfe => R-Hilfe“ aus. Es erscheint die R-Hilfe-Seite. Von nun an ist das erforderliche Verzeichnis angelegt, und die Fehlermeldung sollte nicht mehr erscheinen.

7.3. Installation von Zusatzpaketen über die Shell

Falls Sie ein Paket manuell über die CRAN-Webseiten heruntergeladen haben, muss dieses über die Kommandozeile installiert werden. Wechseln Sie hierzu auf der Shell in das Verzeichnis, in welches Sie die tar-Datei heruntergeladen haben, und geben ein:

```
R CMD INSTALL paket
```

wobei Sie "paket" durch den Datei- bzw. Paketnamen (**ohne** .tar.gz) ersetzen.

7.4. Zusatzpakete aktivieren und deaktivieren

Bitte beachten Sie, dass die Funktionen der Zusatzpakete zunächst nicht aufgerufen werden können! Hierfür müssen Sie das entsprechende Paket zuerst mit `library(PAKETNAME)` aktivieren. Um beispielsweise das „foreign“-Paket zu aktivieren, geben Sie also ein:

```
> library3(foreign)
```

Wenn die Funktionen der Zusatzpakete nicht mehr benötigt werden, können diese mit `detach("package:PAKETNAME")` deaktiviert werden, in unserem Fall also per:

```
> detach4("package:foreign")
```

7.5. Zusatzpakete aktualisieren

Die R-Basis- und Zusatzpakete lassen sich mit `update.packages()`⁵ aktualisieren.

⁵ <http://de.wikibooks.org/wiki/GNU%20R%3A%20update.packages>

7.6. Übersicht über installierte Zusatzpakete

Mit `installed.packages()`⁶ erhalten Sie ein Übersicht über alle installierten Zusatzpakete sowie deren Versionen.

7.7. siehe auch

- Das `wikibooks`-Zusatzpaket⁷
- Eigene Zusatzpakete erstellen⁸

⁶ <http://de.wikibooks.org/wiki/GNU%20R%3A%20installed.packages>

⁷ Kapitel 57 auf Seite 243

⁸ Kapitel 44 auf Seite 155

Teil II.

R benutzen

8. Rechnen mit R

R ist unter anderem ein mächtiger Taschenrechner. Folgende Rechenfunktionen können mit R ausgeführt werden:

Addition	+
Subtraktion	-
Multiplikation	*
Division	/
Exponenten	^ oder **
Modulo	%%
Wurzel	$\text{sqrt}^1(x)$

8.1. Beispiele

```
> 2+4  
[1] 6
```

```
> 4-1  
[1] 3
```

```
> 2*2  
[1] 4
```

```
> 4/2  
[1] 2
```

```
> 2^3  
[1] 8
```

¹ <http://de.wikibooks.org/wiki/GNU%20R%3A%20sqrt>

```
> 0.1**2  
[1] 0.01
```

```
> 10%%3  
[1] 1
```

```
> (-1)%%4  
[1] 3
```

```
> sqrt2(100)  
[1] 10
```

8.2. Wurzel

Das Ziehen einer Wurzel erfolgt über die Funktion `sqrt`³. Damit Funktionen arbeiten können, benötigen sie ein „Argument“. Das Argument ist in unserem Fall die Zahl 100, da wir die Wurzel aus 100 ziehen wollen. Argumente werden der Funktion in runden Klammern angehängt. So erklärt sich der Befehl `sqrt(100)` (siehe oben).

8.3. Kommastellen

In R ist das Dezimaltrennzeichen ein Punkt (und nicht, wie im deutschsprachigen Raum üblich, ein Komma).

```
> 2.01 + 4.2  
[1] 6.21
```

```
> 2.6 * 3.72  
[1] 9.672
```

8.3.1. Nachkommastellen

R zeigt standardmäßig 6 Nachkommastellen an. Intern rechnet R mit "doppelter Genauigkeit"⁴. Damit sind Nummern mit ca. 17 Stellen (1 Ziffer + 16 Nachkommastellen oder eben eine 17 stellige Zahl ohne Nachkommastellen) repräsentierbar. Möchte man die Nachkommastellen entsprechend angezeigt bekommen, so kann man dies mit der `options`⁵-Funktion einstellen:

```
options(digits=17)
```

Nach dieser Eingabe liefert R Ergebnisse mit bis zu 16 Nachkommastellen.

3 <http://de.wikibooks.org/wiki/GNU%20R%3A%20sqrt>

4 http://de.wikipedia.org/wiki/Double_precision

5 Kapitel 56 auf Seite 239

8.4. Runden

R zeigt standardmäßig 6 Nachkommastellen an. Ziehen wir die Wurzel aus 3, erhalten wir folgendes Ergebnis:

```
> sqrt6(3)
[1] 1.732051
```

Dieses Ergebnis kann mit der Funktion `round` gerundet werden:

```
> round7(sqrt8(3))
[1] 2
```

Es besteht des Weiteren die Möglichkeit, die Anzahl der Nachkommastellen in der `round`-Funktion mitanzugeben. Die `round`-Funktion kann zwei Argumente verarbeiten. Das erste Argument heißt `x` und ist die Zahl, die gerundet werden soll (in unserem Beispiel „Wurzel aus 3“). Das zweite Argument heißt „`digits`“ und gibt an, auf wieviele Nachkommastellen das Argument `x` gerundet werden soll. Wollen wir das Ergebnis mit zwei Nachkommastellen angegeben bekommen, so lautet der Befehl:

```
> round9(x=sqrt(3), digits=2)
[1] 1.73
```

Wenn man den Namen des Argumentes mit in die Funktion schreibt, spielt die Reihenfolge der Argumente keine Rolle. So könnten wir die Berechnung in unserem Falle auch erzeugen durch:

```
> round10(digits=2, x=sqrt(3))
[1] 1.73
```

Eine Übersicht über die Argumente einer Funktion (und ihrer Reihenfolge) liefert die Help-Seite, die wir in unserem Falle aufrufen können per:

```
> ?11round
```

Wenn man die Reihenfolge der Argumente einer Funktion kennt, so muss man die Argumente nicht unbedingt als solche benennen. In unserem Beispiel funktioniert auch die Eingabe:

```
> round12(sqrt13(3), 2)
[1] 1.73
```

Dies liegt daran, dass das erste Argument `x` (in unserem Falle `sqrt(3)`) und das zweite Argument `digits` (in unserem Falle 2 Nachkommastellen) heißt. Wichtig ist hierbei allerdings, dass man die Argumentwerte in der tatsächlich geforderten Reihenfolge schreibt, welche durch die Help-Seite (`?round`) eingesehen werden kann.

8.5. Integrieren

Zum Integrieren kann die Funktion `integrate`¹⁴ verwendet werden. In folgendem Beispiel wird die Dichtefunktion der Standardnormalverteilung von 0 bis +Unendlich integriert (und damit die Wahrscheinlichkeit bestimmt, dass eine Standardnormalverteilung einen positiven Wert annimmt):

```
> integrate(dnorm, 0, Inf)
0.5 with absolute error < 4.7e-05
```

Möchte man der zu integrierenden Funktion weitere Parameter übergeben, kann man das im Anschluss an die drei ersten Parameter der Funktion `integrate()` tun. In folgendem Beispiel integrieren wir die Dichte einer $N(2,5^2)$ verteilten Zufallsgröße von 0 bis +Unendlich:

```
> integrate(dnorm, 0, Inf, mean=1, sd=5)
0.5792597 with absolute error < 1.4e-05
```

¹⁴ <http://de.wikibooks.org/wiki/GNU%20R%3A%20integrate>

9. Daten in R

Daten in **R** können in Form verschiedener Objekte auftreten und auf vielfältige Weise manipuliert werden. Sie können direkt eingegeben oder über eine Textdatei oder Datenbank eingelesen und ausgegeben werden¹.

9.1. Variablen

Variablen dienen in R, wie in anderen Programmiersprachen, der Speicherung von Daten. Dies können sowohl einfache Datentypen sein oder auch komplexe Datentypen wie Vektoren. Variablen bestehen aus einem Namen und einem Wert. Der Name darf nicht mit einer Zahl beginnen. Variablennamen wie *1r* sind in R nicht erlaubt. Der Wert einer Variable kann in R auf zwei Weisen zugewiesen werden:

```
> x=10
> y<-20
> x
[1] 10
> y
[1] 20
>
```

Der ersten Variable mit dem Namen *x* wurde der Wert *10* zugewiesen, mittels = Zeichen, der Wert von *y* wurde mittels <- zugewiesen. Beide Weisen bedeuten das gleiche. Anschließend wurden die Werte ausgegeben. Dazu muss man einfach den Variablennamen eintippen. Im Gegensatz zu vielen anderen Programmiersprachen wird der Typ einer Variable nicht explizit definiert.

R unterscheidet zwischen Groß- und Kleinschreibung bei den Variablennamen.

```
> X
Fehler: Objekt 'X' nicht gefunden
> x
[1] 10
```

Die Variable X wurde nicht gefunden, die Variable x schon.

9.1.1. Den Wert von Variablen ausgeben

Durch die einfache Eingabe eines Variablennamens in die R-Konsole wird der Inhalt der Variable ausgegeben. Dies gilt auch für Vektoren oder Matrizen.

¹ Kapitel 12 auf Seite 37

```
z
[1] 5.0 3.0 4.0 8.0 6.0 2.5 0.0 1.3
```

9.1.2. Variablen anzeigen und Variablen löschen

Der Befehl `ls()`² listet alle Variablen auf, die sich in der aktuellen Worksession befinden. Genauso einfach kann man auch Variablen löschen. Dies geschieht durch den Befehl `rm()`³ und die Angabe des Variablennamen in der Klammer.

```
> ls()
[1] "bigtable" "bt"      "fl"      "names"   "t100"    "tt"
> rm(t100)
> ls()
[1] "bigtable" "bt"      "fl"      "names"   "tt"
>
```

In diesem Beispiel wurden zuerst alle Variablennamen aufgelistet und anschließend wurde die Variable `t100` mittels `rm` gelöscht.

9.1.3. Variablen vergleichen

Das Vergleichen von zwei Variablen ist vor allem in der Programmierung sehr wichtig.

```
> x==y
[1] FALSE
> y=10
> x==y
[1] TRUE
```

Beim Vergleich der beiden Variablen `x` und `y` werden deren Inhalte verglichen. Mittels `==` wird verglichen, ob der Wert beider Variablen gleich ist. Dieser Fakt klingt trivial, aber in einigen objekt-orientierten Programmiersprachen wird dies anders gehandhabt.

9.2. Einfache Datentypen (*modes*)

Bei der Einführung der Variablen wurde nur kurz erwähnt, dass Variablen verschiedene Dinge beinhalten können. Welche Datentypen dies sein können, wird in diesem und den folgenden Abschnitten erläutert. Prinzipiell gibt es in R drei Datentypen, einfache Datentypen, komplexe Datentypen und speziellere Datentypen. Wir beginnen mit den einfachen Datentypen, die in den bisherigen Beispielen auch schon eingesetzt haben. Die wesentlichen einfachen Datentypen sind:

- `numeric` für Zahlen (wobei zwischen `integer` und `double` unterschieden werden kann)
- `complex` für komplexe Zahlen
- `logical` für logische Werte

² <http://de.wikibooks.org/wiki/GNU%20R%3A%20ls>

³ <http://de.wikibooks.org/wiki/GNU%20R%3A%20rm>

- `character` für Zeichen
- `raw` für Bytes

```
> 5
> 3i+2
> TRUE
> "abcde"
> as.raw(65)
```

Zusätzlich gibt es einige etwas speziellere Datentypen, die zunächst nicht weiter behandelt werden sollen

- `function` für Funktionen
- `call` für Funktionsaufrufe
- `expression` für Ausdrücke

Mit der Funktion `mode()` lässt sich der Datentyp eines Objekts bestimmen. Der *mode* ist eine grundlegende Eigenschaft eines jeden Objekts. Eine weitere grundlegende Eigenschaft ist `length()`. Zusätzliche Eigenschaften können mit `attributes()` ermittelt werden.

9.3. Komplexe Datentypen: Klassen

Aus einfachen Daten lassen sich verschiedene Arten von Datenstrukturen zusammensetzen. Dies sind

- `vector` für Listen von Objekten gleichen Typs
- `list` für beliebige Listen von Objekten (auch rekursiv)
- `factor` für Vektoren zur Gruppierung von Objekten
- `matrix`
- `array`
- `data.frame`
- ...

Welche Datenstruktur ein Objekt besitzt, kann mit `class()` ermittelt werden. Klassen sind ein allgemeines Konzept von erweiterten Datentypen. Jedes Objekt besitzt eine Klasse, die mit einem einfachen Datentyp übereinstimmt oder zusätzlich festgelegt werden kann.

Klassen sind wichtig für so genannte *generische* Funktionen, die je abhängig von der Klasse eines übergebenen Objekts arbeiten. Beispielsweise gibt es die Klasse `Date`, mit der ein numerischer Wert als Datum interpretiert werden kann. Die Klasse eines Objekts lässt sich mit `class()` feststellen.

```
d <- as.Date4("2005-01-01") # Umwandlung einer Zeichenkette in ein
Date-Objekt
mode5(d)
[1] "numeric"
class6(d)
[1] "Date"
```

9.4. Zeichenketten

Zeichen oder Zeichenketten werden mit einfachen oder doppelten Anführungszeichen geschrieben. Dabei werden die Daten als n-dimensionale Vektoren gespeichert. Mehrere dieser Vektoren können zu einer Matrix zusammengefügt werden. Die Vektoren können drei verschiedene Datentypen enthalten, numerische und logische Werte sowie Text. Des weiteren existieren Funktionen, mit welchen Datenreihen erzeugt werden können.

```
h <- "Hallo"
w <- "Welt!"
paste(h,w, sep=", ")
[1] "Hallo, Welt!"
```

In diesem Beispiel werden zwei Zeichenketten miteinander verknüpft und ein Separator zwischen den Zeichenketten eingefügt. Es werden also eigentlich drei Zeichenketten miteinander verknüpft. Für diese Zeichenkettenverknüpfung wird der Befehl `paste`⁷ benutzt.

Die Länge von Zeichenketten lässt sich mit dem Befehl `nchar`⁸ ermitteln.

```
> x=" ABC"
> nchar(x)
[1] 4
```

Die Zeichenkette `x` hat die Länge 4, das Leerzeichen vor dem ersten A wird mitgezählt.

Zeichenketten lassen sich mit Hilfe verschiedener Befehle wie `split`, `grep` und `substr` bearbeiten.

```
> substr(x,2,3)
[1] "AB"
```

Hier wird ein Substring aus `x` erzeugt. Der erste Parameter gibt den Start und der letzte das Ende des Substrings in der ursprünglichen Zeichenkette an. In diesem Fall werden die Buchstaben AB ausgewählt. Wenn der Start- und der Endwert gleich sind, wird nur ein Buchstabe ausgewählt.

```
> substr(x,2,2)
[1] "A"
```

9.5. Vektoren

Ein Vektor ist ein Datentyp in R, mit dem man eine Reihe anderer Datentypen z.B. numeric oder character enthält. Ein Vektor ist ein eindimensionales Array. Hier ein Beispiel aus der Genetik, ein Vektor mit SNPs von 150 Tieren.

```
> s1
[1] GG GG GG GG GG AC GG AA GG AC GG AG AG AC AG AC
GG GG AC GG AA GG GG AG CC AA GG AC AA GG AG AC GG
```

⁷ <http://de.wikibooks.org/wiki/GNU%20R%3A%20paste>

⁸ <http://de.wikibooks.org/wiki/GNU%20R%3A%20nchar>

```

GG AG GG
[37] GG GG AG GG AG GG AG AC AG AG GG AG GG AA GG AG
     AG GG AA AC AG GG AG GG AA AC GG GG AA GG GG GG AG
GG AG AG
[73] AA AC GG AG AG AA AG AG AG AG AA AA AG AC GG GG
     CC AA GG GG AC CC GG AC AG AC AC GG AG GG AC AG AA
AG GG CC
[109] AG GG AG AG AG AC AG AA AC AA AC AA AG AG GG GG
      GG AG GG GG AG GG AG GG AG AA GG GG AA AC AG AG GG
      AG AG AA
[145] AA GG AG AC AA GG

```

Auf die Elemente eines Vektors kann man wie folgt zugreifen:

```

> s1[1]
[1] GG
Levels: AA AC AG CC GG

```

Hier wurde ein einzelnes Element, das erste Element des Vektors, ausgewählt. Im Gegensatz zu den meisten anderen Programmiersprachen wird das erste Element eines Vektors in R nicht mit 0 sondern mit 1 angesprochen. Zusätzlich werden immer die Level bei Vektoren ausgegeben, die Faktoren enthalten.

```

> s1[3:15]
[1] GG GG GG AC GG AA GG AC GG AG AG AC AG
Levels: AA AC AG CC GG

```

Durch die Angabe eines Bereichs können auch mehrere Elemente ausgewählt werden. In diesem Beispiel wurden die Elemente 3 bis 15 ausgewählt.

```

> class(s1)
[1] "factor"
> dim(s1)
NULL
> length(s1)
[1] 150

```

Mit *class* findet man heraus, um was für einen Vektor es sich handelt, hier ist es ein Vektor der Faktoren enthält. Der Befehl *dim* liefert überraschender Weise bei Vektoren *null*, die Länge eines Vektors lässt sich aber mittels *length* bestimmen, hier enthält der Vektor 150 Elemente.

```

> levels(s1)
[1] " AA" " AC" " AG" " CC" " GG"

```

Wenn man einen Vektor mit Faktoren hat, kann man mittels *levels* die unterschiedlichen Elemente extrahieren und in Kombination mit *length* die Anzahl der Elemente ermitteln.

```

length(levels(s1))
[1] 5

```

Wenn man herausfinden möchte wie oft die einzelnen Faktoren im Vektor vorkommen, kann man den Befehl *table* verwenden.

```
> table(s1)
s1
  AA  AC  AG  CC  GG
  22  22  46   4  56
```

10. Manuelle Dateneingabe

Eine einfache Möglichkeit manueller Dateneingabe ist die Funktion `c1` :

```
c2(5,3,4,8,6,2.5,0,1.3)
```

liest die Zahlenfolge "5, 3, 4, 8, 6, 2.5, 0, 1.3" ein. Wichtig ist hierbei, dass als **Dezimaltrennzeichen** ein Punkt stehen muss.

Der eben genannte Aufruf erzeugt nur die Ausgabe

```
[1] 5.0 3.0 4.0 8.0 6.0 2.5 0.0 1.3
```

10.1. Eingabe von logischen Werten:

```
abc <- c3(TRUE, FALSE, TRUE, FALSE, FALSE, T, F, T, T)
```

speichert die logischen Werte in *abc*. TRUE kann immer mit *T* und FALSE mit *F* abgekürzt werden.

¹ <http://de.wikibooks.org/wiki/GNU%20R%3A%20c>

11. Umgang mit Datensätzen (Erstellen, Auswählen und Filtern)

12. Datenimport und -export

Der Datenaustausch zwischen verschiedenen Anwendungen ist oft eine mühsame Angelegenheit mit vielen Fallstricken. Er erfordert daher einiges an Aufmerksamkeit und Sorgfalt. R kann zahlreiche Formate einlesen. Welches Format im Einzelfall am geeignetsten ist, muss oft ausprobiert werden, besonders wenn das native Format eines Programmes nicht unterstützt wird. So wird z.B. das Dateiformat der Statistiksoftware Statistica von R nicht unterstützt. Allerdings bietet Statistica die Möglichkeit, die Daten unter anderem als SPSS-Datei oder als CSV-Datei¹ (via Text File) zu exportieren, die von R eingelesen werden können.

12.1. Import

12.1.1. Import aus Textdateien

Mit der `read.csv`², `read.table`³ und verwandten Funktionen können Daten aus Textdateien (txt) importiert werden. `read.table`⁴ ist die Basisfunktion zum Import von Textdateien. `read.csv`⁵ und `read.csv2`⁶ sowie einige Funktionen mehr sind Anpassungen an häufig auftretenden Fälle. CSV⁷ ist z.B. das Standardexportformat für viele Tabellenkalkulationen und andere Anwendungen, unterscheidet sich aber, je nachdem ob als Dezimaltrennzeichen der Punkt (englischer Sprachraum) oder das Komma (bei uns) verwendet wird.

Legen Sie eine txt-Datei mit folgendem Inhalt an, indem Sie die folgenden Zeilen markieren, kopieren und in einen Texteditor einfügen:

```
Geschlecht Alter Gewicht Groesse
m 28 80 170
w 18 55 174
w 25 74 183
m 29 101 190
m 21 84 185
w 19 74 178
w 27 65 169
w 26 56 163
m 31 88 189
m 22 78 184
```

-
- 1 <http://de.wikipedia.org/wiki/CSV-Datei>
 - 2 <http://de.wikibooks.org/wiki/GNU%20R%3A%20read.csv>
 - 3 <http://de.wikibooks.org/wiki/GNU%20R%3A%20read.table>
 - 4 <http://de.wikibooks.org/wiki/GNU%20R%3A%20read.table>
 - 5 <http://de.wikibooks.org/wiki/GNU%20R%3A%20read.csv>
 - 6 <http://de.wikibooks.org/wiki/GNU%20R%3A%20read.csv>
 - 7 <http://de.wikipedia.org/wiki/CSV-Datei>

Speichern Sie diese Textdatei unter dem Namen „Tabellentest.txt“ auf Ihre Festplatte. Nun können Sie die Tabelle mit der Funktion `read.table()`⁸ in R einlesen:

```
> meinetabelle <- read.table9("/Pfad/zur/Tabelle.txt", header=TRUE)
```

Das Argument `header=TRUE` besagt, dass die erste Zeile der Datei keine Daten enthält, sondern die Bezeichnungen (Variablen) der Tabellenspalten angibt.

Die Tabelle ist nun in dem Objekt `meinetabelle` vorhanden. Sie können sie sich ansehen, indem Sie eingeben:

```
> meinetabelle
```

Geschlecht	Alter	Gewicht	Groesse	
1	m	28	80	170
2	w	18	55	174
3	w	25	74	183
4	m	29	101	190
5	m	21	84	185
6	w	19	74	178
7	w	27	65	169
8	w	26	56	163
9	m	31	88	189
10	m	22	78	184

In unserem Beispiel waren die Daten jeweils mit einem Leerzeichen getrennt. Haben Sie eine Textdatei, in welcher die Daten durch einen Tab getrennt sind, müssen sie die `read.table`-Funktion um das Argument `sep="\t"` erweitern:

```
> meinetabelle <- read.table10("/Pfad/zur/Tabelle.txt", sep="\t", header=TRUE)
```

Wurden die Daten von einer Tabellenkalkulation, wie z.B. Excel, aber auch aus Statistikprogrammen, als CSV-Datei (comma separated values) exportiert, dann muss noch unterschieden werden, welche Schreibkonvention verwendet wurde. Bei englischsprachiger Software wird der Punkt als Dezimaltrennzeichen verwendet und das Komma, um die verschiedenen Werte voneinander zu trennen. In diesem Fall ist `read.csv`¹¹ zu verwenden. Sofern das Komma als Dezimaltrennzeichen dient und der Strichpunkt (;) die einzelnen Werte voneinander trennt, dann kommt `read.csv2`¹² zum Einsatz.

Tip

8 <http://de.wikibooks.org/wiki/GNU%20R%3A%20read.table>

11 <http://de.wikibooks.org/wiki/GNU%20R%3A%20read.csv>

12 <http://de.wikibooks.org/wiki/GNU%20R%3A%20read.csv>

Bei CSV-Dateien kommt es leicht zu Problemen, wenn Textvariablen, z.B. mit Kommentaren, vorhanden sind. Diese können das jeweilige Trennzeichen enthalten. Beim Import kommt es dann dazu, dass nachfolgende Felder mit falschen Inhalten gefüllt werden. Nicht jedes importierende Programm gibt eine Fehlermeldung aus, obwohl mehr Felder als erwartet auftreten. Das gilt auch für R.

Eine Abhilfe besteht darin, ein alternatives Trennzeichen zu verwenden, das garantiert nicht in den Textvariablen auftritt, z.B. @ (wenn es sich nicht zufällig um Emailadressen handelt). Allerdings ist das nicht mit allen Programmen möglich. Eine weitere Möglichkeit besteht darin, Stringvariablen in Anführungszeichen zu setzen. Auch das ist nicht immer möglich. Im ungünstigsten Fall müssen die Textvariablen nachbearbeitet und das jeweilige Trennzeichen ersetzt werden. Auf jeden Fall ist es erforderlich die Daten nach dem Import auf Fehler zu kontrollieren.

Bei R führt das zusätzliche Datenfeld dazu, dass ein zusätzlicher Fall angelegt wird, der durchgängig mit fehlenden Werten (NA) belegt ist. Mit `dim(x)`¹³ kann einfach überprüft werden, ob mehr Fälle als erwartet vorliegen.

12.1.2. Import aus dem Zwischenspeicher / Clipboard

Analog dem Import aus Textdateien kann statt einer File bei `read.table` auch `stdin()` als Quelle angegeben werden. In diesem Fall wartet R auf Texteingaben. Wenn sich im Zwischenspeicher eine Tabelle befindet, kann diese per Copy and Paste übernommen werden.

```
> a <- read.table(file=stdin(),header=T)

0: Name Geschlecht Lieblingsfarbe           # R wartet auf
Eingabe

1: Hans m gruen

2: Waldemar w blau

3: Tanja m rosa

4:                                           # Return in
leerer Zeile schließt Eingabe ab

> a

      Name Geschlecht Lieblingsfarbe
1    Hans           m           gruen
2 Waldemar           w           blau
```

¹³ <http://de.wikibooks.org/wiki/GNU%20R%3A%20dim>

3	Tanja	m	rosa
---	-------	---	------

Dies ist oft die einfachste Variante des Imports, wenn aus Browsern oder PDF-Files Tabellen übernommen werden sollen, die bereits hinreichend formatiert sind.

Tip

Wenn die Daten aus einer HTML-Seite übernommen werden, ist die Verwendung des Webbrowser Firefox¹⁴ in Verbindung mit dem Add-On Table2Clipboard¹⁵ hilfreich. Wenn die Standardeinstellung mit Tabulator-Trennung Probleme macht, funktioniert zum Beispiel die Parameterkombination row-seperator: "\n", column-seperator: "@" in Table2Clipboard und `read.table(file=stdin(), header=T, dec=".", sep="@")` in R.

12.1.3. Import aus MySQL

Um aus R auf eine MySQL-Datenbank zugreifen zu können, muss das Paket RMySQL¹⁶ installiert sein. Es benötigt zusätzlich das Paket DBI¹⁷ (und unter Windows die Datei `libmysql.dll`, die in MySQL enthalten ist).

```
library18(RMySQL)      # package RMySQL laden
drv = dbDriver("MySQL") # MySQL verwenden

# Verbinde mit einer Datenbank
con = dbConnect(drv, "user/password@dbname")
# Verbinde mit einer Datenbank und verwende Benutzer/Passwort aus
der Datei .my.cnf
con = dbConnect(drv, "dbname")
# Andere Möglichkeit
con <- mysqlNewConnection19(drv, "dbname", "user", "password")

# listet alle Tabellen auf
dbListTables20(con)

# Anfrage
rs <- dbGetQuery21(con, "SELECT COUNT(*) FROM mytable")
```

Hilfreich sind auch die Funktionen `mysqlReadTable`²² und `mysqlWriteTable`²³.

12.1.4. Import aus SPSS

Sie können Ihre vorliegenden SPSS-Daten in R importieren. Hierzu müssen Sie allerdings zunächst ein Zusatzpaket in R installieren. Funktionen zum Datenimport aus SPSS bieten beispielsweise die Pakete `foreign` und `Hmisc`. Sie installieren also zunächst eines der Zusatzpakete per:

-
- 14 <http://www.mozilla.com/>
 - 15 <http://dafizilla.sourceforge.net/table2clip/>
 - 16 <http://cran.r-project.org/web/packages/RMySQL/index.html>
 - 17 <http://cran.r-project.org/src/contrib/Descriptions/DBI.html>
 - 22 <http://de.wikibooks.org/wiki/GNU%20R%3A%20mysqlReadTable>
 - 23 <http://de.wikibooks.org/wiki/GNU%20R%3A%20mysqlWriteTable>

```
install.packages24("foreign", dependencies = TRUE)
```

oder

```
install.packages25("Hmisc", dependencies = TRUE)
```

Das Argument `dependencies=TRUE` besagt, dass alle weiteren relevanten Zusatzpakete, die eventuell für `foreign` und `Hmisc` benötigt werden, direkt mitinstalliert werden. Bevor sie die Funktionen der Zusatzpakete nutzen können, müssen Sie diese aktivieren per:

```
library26(foreign)
oder
library27(Hmisc)
```

Der Import kann nun über die folgenden Funktionen erfolgen:

- `read.spss()`²⁸ (aus dem `foreign`-Paket)
- `spss.get()`²⁹ (aus dem `Hmisc`-Paket)

Dies geschieht mit der `read.spss`-Funktion beispielsweise so:

```
meinspss <- read.spss30("/der/pfad/zur/SPSSdatei.sav")
```

Beachten Sie, dass Sie den kompletten Pfad zur SPSS-Datei angeben müssen. Auf Windows-Systemen werden Pfadangaben normalerweise mit einem Backslash (\) dargestellt. Dies funktioniert in R nicht, da der Backslash in R ein Steuerungszeichen ist. Die übliche Schreibweise in R mit dem Schrägstrich (/) funktioniert aber auch unter Windows einwandfrei. Falls unter Windows der Backslash beibehalten werden soll, muss jeder Backslash durch zwei Backslashes ersetzt werden (z.B. `C:\\Pfad\\zur\\SPSSdatei.sav`).

Sind in der SPSS-Datei nominale Daten vorhanden, deren numerische Ausprägungen mit Labels hinterlegt sind (z.B. bei einer Variable „Geschlecht“ sind die Ausprägungslabes 0=weiblich, 1=männlich gesetzt), so werden die ausgeschriebenen Labels als Variablenwert in R übernommen

Wenn Sie möchten, dass nicht die Variablenlabels, sondern die numerischen Ausprägungen (z.B. `Geschlecht = 0;1`) übernommen werden, muss der Befehl um das Argument `use.value.labels=FALSE` erweitert werden:

```
meinspss <- read.spss31("/pfad/zur/SPSSdatei.sav", use.value.labels=FALSE)
```

Jetzt sind in dem Objekt `meinspss` die Daten der SPSS-Datei vorhanden. Sie können sie sich ansehen, indem Sie eingeben:

²⁸ <http://de.wikibooks.org/wiki/GNU%20R%3A%20read.spss>

²⁹ <http://de.wikibooks.org/wiki/GNU%20R%3A%20spss.get>

```
meinspss
```

Tip

Für Variablennamen mit Umlauten können beim Import aus SPSS-Dateien Probleme auftreten. Deshalb ist es möglicherweise notwendig vor dem Import die Sprachoptionen richtig zu setzen (z.B. `Sys.setlocale(locale="de_DE.ISO8859-15")`).
`read.spss` übernimmt auch die in SPSS vergebenen Variablenlabels, allerdings als spezielles Attribut. Diese können über `attr(meinspss, "variable.labels")` als Vektor ausgegeben und entsprechend als Titel für Tabellen und Graphiken weiter verwendet werden. (Als Beispiel, wie sich SPSS-Labels in R importiert lassen, dient dieses Skript³².)

12.1.5. Import aus XML

12.1.6. Import aus Excel-Dateien und CSV-Dateien

Excel Dateien können auf verschiedenen Wegen in R importiert werden. Welcher Weg für den Benutzer der Richtige ist hängt von folgenden Bedingungen ab:

- Excel Version,
- Umfang der Daten,
- Strukturierung der Daten.

Excel ist sehr weit verbreitetes Programm zur Verarbeitung von Tabellen. Es gibt diverse Versionen dieser Software und durch die kontinuierliche Weiterentwicklung von Excel gibt es auch verschieden Versionen des Excel Dateiformats. Für die verschiedenen Excel Formate gibt es Importmöglichkeiten in R z.B. via eines Datenbanktreibers.

Neben diesen Importmöglichkeiten gibt es den oft praktikableren Weg, die Datei im Excel Format in eine CSV (Comma-Separated Values) Datei umzuwandeln. Dazu kann man Excel selbst aber auch andere Programme wie Open Office verwenden. Der Nachteil dieses Weges ist, dass man nur eine Tabelle umwandeln kann. Oft bestehen Excel Dateien aus mehreren Tabellen. Nach der Umwandlung kann man die CSV Datei in R wie folgt einlesen:

```
tt = read.csv("t.csv")
```

Hier wird die Datei `t.csv` aus dem aktuellen Arbeitsverzeichnis in die Variable `tt` vom Typ "list" eingelesen. Alternativ kannman auch folgende Anweisung verwenden

```
tt = read.table("t.csv", sep=",")
```

Die Option "sep" legt fest, welches Zeichen R als Zelltrenner in der Datei interpretieren soll. Weitere Optionen sind unter GNU R: `read.table`³³ zu finden.

³² Kapitel 55.1 auf Seite 237

³³ <http://de.wikibooks.org/wiki/GNU%20R%3A%20read.table>

12.2. Export

12.2.1. HDF5

Siehe von <http://cran.r-project.org/web/packages/hdf5/hdf5.pdf>

```
# Import von HDF-lib
install.packages("hdf5")
library("hdf5")
(m <- cbind(A = 1, diag(4)))
l1 <- list(a=1:10, b=letters[1:8]);
l2 <- list(C="c", l=l1); PP <- pi
# Speichern der Objekte als HDF Datei
hdf5save("ex1.hdf", "m","PP","l1","l2")
# Entfernen der Objekte aus dem R-Workspace
rm(m,PP,l1,l2) # and reload them:
# Laden der Objekte aus der HDF Datei
hdf5load("ex1.hdf",verbosity=3)
m          # read from "ex1.hdf"; buglet: dimnames dropped
str(l1)
str(l2)
```


13. Daten laden und speichern

14. Daten laden

Daten können mit `load1("Dateiname")` geladen werden.

Wenn die Datei im Internet liegt, muss sie mit Hilfe der `url()`²-Funktion geladen werden

```
> load(url("http://Pfad/zu/Datei.RData")) # Beispiel-URL
```

1 <http://de.wikibooks.org/wiki/GNU%20R%3A%20load>
2 http://de.wikibooks.org/wiki/GNU_R%3A_url

15. Daten speichern

Mit der Funktion `ls()`¹ listet R alle vorliegenden Variablen auf.

15.1. Arbeitsspeicher speichern

Mit `save.image("dateiname.RData")` wird das vorliegende Datenframe (alle Variablen) als sichtbare Datei im `working directory`³ (`getwd()`⁴) gespeichert (Wenn man nur `save.image()`⁵ ausführt, wird das Datenframe als ".RData" gespeichert und ist dann im Dateibrowser unsichtbar).

Wenn man dann R wieder startet, wird diese Datei automatisch geladen. Unter Linux scheint es notwendig zu sein, den Arbeitsspeicher per Hand zu laden: Zunächst muss man in das Verzeichnis wechseln, in dem die Datei liegt. Diese wird dann mittels `load("Dateiname.RData")` geladen.

15.2. einzelne Objekte speichern

Wenn nur einzelne Objekte gespeichert werden sollen, wird die `save`-Funktion angewandt. Mit dem `list`-Parameter werden die gewünschten Objekte angegeben, welche in die Datei `file` gespeichert werden sollen.

```
save(file="/Users/produnis/working/R/Einzelobjekt.RData",
list="Objekt") # Beispiel
```

1 <http://de.wikibooks.org/wiki/GNU%20R%3A%20ls>
2 <http://de.wikibooks.org/wiki/GNU%20R%3A%20save.image>
3 http://de.wikibooks.org/wiki/GNU_R%3A_Arbeitsverzeichnis
4 <http://de.wikibooks.org/wiki/GNU%20R%3A%20getwd>
5 <http://de.wikibooks.org/wiki/GNU%20R%3A%20save.image>

16. Arbeitsverzeichnis

Das aktuelle Arbeitsverzeichnis kann per `getwd()`¹ angezeigt werden. Mit dem Befehl `dir()`² werden die Dateien des aktuellen Arbeitsverzeichnisses angezeigt.

Möchte man das Arbeitsverzeichnis wechseln, so benutzt man den Befehl `setwd()`³.

```
setwd4("/Users/produnis/R/") # Beispiel
```

Auf Dauer kann es einfacher sein, ein eigenes Arbeitsverzeichnis beim Startvorgang einzustellen. Für diese Fälle gibt es die Datei `Rprofil.site`. Dieser muss nur eine letzte Zeile (gefolgt von einem Zeilenumbruch) mit dem Befehl "setwd" hinzugefügt werden.

1 <http://de.wikibooks.org/wiki/GNU%20R%3A%20getwd>

2 <http://de.wikibooks.org/wiki/GNU%20R%3A%20dir>

3 <http://de.wikibooks.org/wiki/GNU%20R%3A%20setwd>

17. Daten selektieren

18. Daten auswählen

Sehr oft möchte man nicht alle Daten aus einer eingelesenen Datei verarbeiten, sondern nur Teile davon. Hier bietet R elegante Lösungen basierend auf Rs sehr flexiblen Parameterverarbeitung.

Problem: Ich habe ein Tabelle und möchte nur diese Tabelle reduzieren. Es sollen nur die Spalten ausgewählt werden, die mit den Buchstaben XX anfangen.

Lösung: Hier ein Beispiel.

```
> dim(bt)
[1] 150 63983
```

Dies ist meine Originaltabelle.

```
> r = colnames(bt, prefix="Row")
```

Die Variable r enthält alle Spaltennamen aus der Originaltabelle.

```
> rr = grep("Row*", r)
```

Die Variable r enthält alle *Indizes* der Namen die mit *Row* beginnen. Nicht die Namen selbst.

```
> snp = bt[, rr]
> dim(snp)
[1] 150 52727
> snp[1:5, 1:5]
  Row_14496 Row_14497 Row_14498 Row_14499 Row_14500
1         AG         AC         AG         AG         AG
2         AA         AC         GG         AG         AG
3         AG         AC         GG         AA         AA
4         AA         AC         AG         AA         AG
5         AA         AA         GG         AG         AG
```

Zuletzt wird die Tabelle reduziert mit Hilfe der mittels *grep* gefundenen Indizes.

19. Matrix- und Listenoperationen

19.1. Matrix- und Listenoperationen

19.1.1. Matrizen

Die wichtigsten Befehle für Matrizen in Kurzform:

```
t(matrix) .....transponierte matrix
eigen(matrix)$val ... Eigenwerte
eigen(matrix)$vec ... Eigenvektoren
cov(swiss) ..... Kovarianz
cor(swiss) ..... Korrelation
diag(Eigenwerte) .... Diagonale der Eigenwerte
%*% ..... Matrixmultiplikation
```

19.1.2. Listen

Für Listen stehen die Funktionen `lapply` bzw. `sapply` zur Verfügung. Dabei wird auf jedes Element eines geordneten Objektes eine definierbare Funktion angewandt.

Beispiel: In dem `data.frame cars` sind die beiden metrischen Variablen `speed` (Geschwindigkeit) und `dist` (Distanz bis zum Stillstand) enthalten. Wir wollen nun zu jeder Variable das arithmetische Mittel:

```
R> lapply ( cars, function(x){ mean(x) } )
$speed
[1] 15.4

$dist
[1] 42.98
```

Generell gibt `lapply` eine Liste mit den resultierenden Objekten aus:

```
R> class ( lapply ( cars, function(x){ mean(x) } ) )
[1] "list"
```

Die Funktion `sapply` fasst die daraus resultierenden Listenelemente zusätzlich zu einem einzelnen Objekt zusammen:

```
R> sapply ( cars, function(x){ mean(x) } )
speed  dist
15.40 42.98
R> class ( sapply ( cars, function(x){ mean(x) } ) )
[1] "numeric"
```

Bei komplexeren Objekten (und insbesondere bei Datumsobjekten der Klassen `POSIXt`) tritt zuweilen das Problem auf, dass bei der Verwendung von `lapply` die Klassendefinition verloren geht. Dieses Problem kann durch folgender Aufruf gelöst werden:

```
do.call( c , lapply ( liste , function (x) { x$datum } ) )
```

20. Benutzeroberflächen und Erweiterungen

R kann von anderen Programmen/Prozessen Befehle empfangen und Ergebnisse zurückgeben. Dadurch ist es möglich graphische Benutzeroberflächen für R zu schreiben oder in andere Anwendungen zu integrieren.

21. Graphische Benutzeroberflächen

Die Entwicklung grafischer Benutzeroberflächen für GNU R steckt noch in den Anfängen. Eine komplette Liste der verschiedenen Projekte findet sich unter <http://www.r-project.org/GUI>. Die unten gelisteten Projekte verfolgen das Ziel, eine mehr oder weniger vollständige Benutzeroberfläche zu entwickeln (im Gegensatz z.B. zu reinen Skript-Editoren):

21.1. JGR (Jaguar)

JGR¹ wird von der Uni Augsburg entwickelt. *JGR* ist eine einfache grafische Benutzeroberfläche für R, geschrieben in Java. Nützlich sind hier vor allem der Paketmanager zum Laden von Zusatzpaketen und der Objektbrowser, welcher die vorhandenen Daten übersichtlich ordnet und auch bearbeiten lässt. Dem Programmmenü können eigene Funktionen hinzugefügt werden. Eine Anleitung hierzu befindet sich in der Paketbeschreibung.

JGR benötigt Sun 1.5 SDK und muss für Linux/Unix selbst kompiliert werden.

21.2. R Commander

R Commander ist eine Menu-Oberfläche von John Fox für Anfänger, Umsteiger und Mausbenutzer, die sich schwer damit tun, die Kommandos von R zu lernen.

<http://socserv.mcmaster.ca/jfox/Misc/Rcmdr/>

R Commander kann als Bibliothek unter R geladen werden. Man benötigt also kein Java und man muss nichts kompilieren.

21.3. RKWard

RKWard [//de.wikipedia.org/wiki/RKWard](http://de.wikipedia.org/wiki/RKWard)² ist ein GUI für R unter KDE 4. Es ist derzeit für Linux/Unix und Windows verfügbar. Enthalten sind unter anderem ein Paketmanager, Skript-Editor, Daten-Editor, Objektbrowser und Hilfebrowser. Neue Dialoge zu Statistikfunktionen lassen sich über Plugins hinzufügen.

1 <http://stats.math.uni-augsburg.de/JGR/>

2 <http://de.wikibooks.org/wiki/>

21.4. Sciviews-R

Sciviews ist eine Verbindung von mehreren Anwendungen, die das Arbeiten mit R erleichtern sollen. Sciviews ist derzeit nur unter Windows lauffähig.

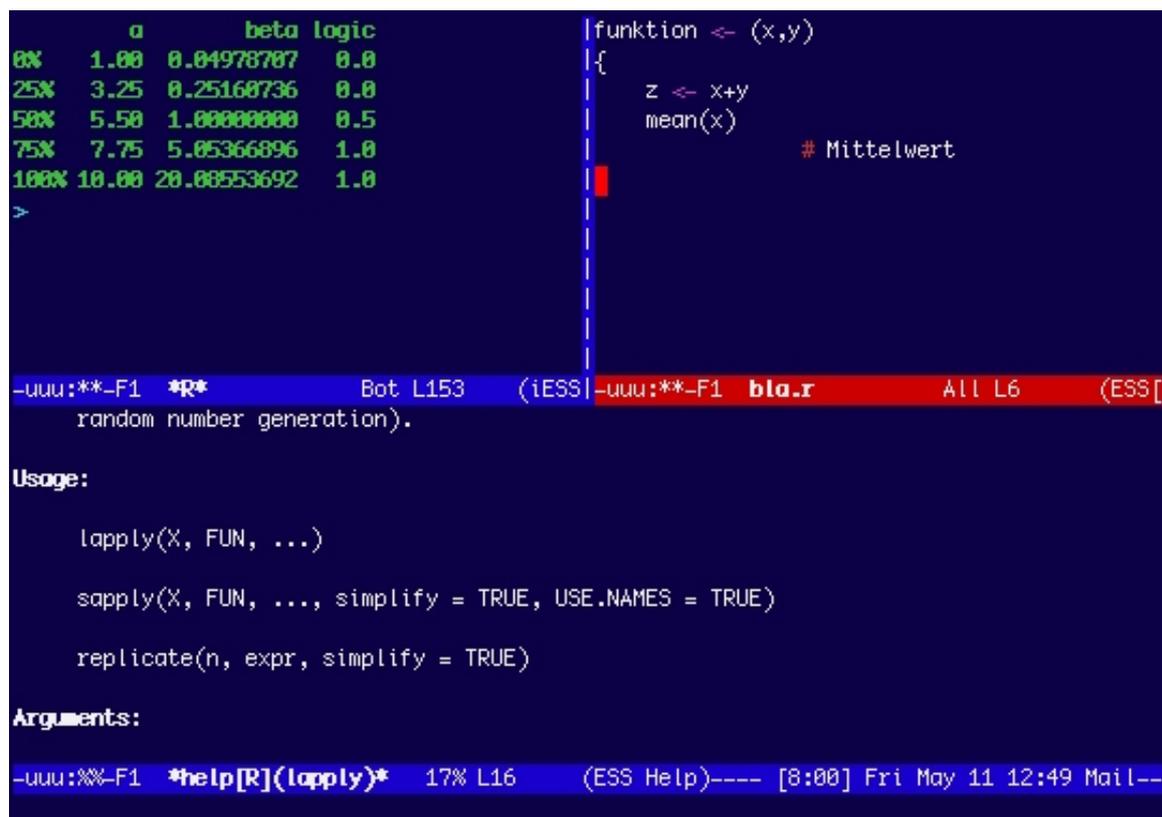
21.5. RStudio

RStudio hält unter einer GUI Console und mehrere Skriptfenster (Source) zum Einen sowie Workspaceinhalte, History und Grafikfenster, Workingdirectory, aktive/passive Pakete und ein Hilfefenster zum Anderen gemeinsam parat. Das Grafikfenster hält eine Historyfunktion und einfache Exportmöglichkeiten vor. Aus der History kann sowohl in die Console als auch das Skriptfenster kopiert werden. RStudio ist für Windows, Mac und Linux verfügbar.

22. Sonstige Schnittstellen

22.1. R in Emacs: ESS

Für den Texteditor Emacs //de.wikipedia.org/wiki/Emacs¹ gibt es mit ESS (Emacs Speaks Statistics) bereits eine fertige All-In-One-Lösung.



The screenshot shows the Emacs ESS interface. The top window displays R console output for a function named 'logic' with parameters 'a', 'beta', and 'logic'. The output shows values for 'a' (1.00, 3.25, 5.50, 7.75, 10.00), 'beta' (0.04978787, 0.25168736, 1.00000000, 5.05366896, 20.08553692), and 'logic' (0.0, 0.0, 0.5, 1.0, 1.0). The bottom window shows R code defining a function 'funktion' that takes parameters 'x' and 'y', calculates 'z <- x+y', and returns 'mean(x)'. A comment '# Mittelwert' is present. The status bar at the bottom shows the current window is 'ESS Help' and the file is 'bla.r'.

```
      a      beta logic
0%    1.00  0.04978787  0.0
25%   3.25  0.25168736  0.0
50%   5.50  1.00000000  0.5
75%   7.75  5.05366896  1.0
100% 10.00 20.08553692  1.0
>

funktion <- (x,y)
{
  z <- x+y
  mean(x)
  # Mittelwert
}

-uuu:**-F1 *R* Bot L153 (iESS) -uuu:**-F1 bla.r All L6 (ESS[
random number generation).

Usage:

lapply(X, FUN, ...)

sapply(X, FUN, ..., simplify = TRUE, USE.NAMES = TRUE)

replicate(n, expr, simplify = TRUE)

Arguments:

-uuu:%%-F1 *help[R](lapply)* 17% L16 (ESS Help)---- [8:00] Fri May 11 12:49 Mail--
```

Abb. 1

Neben Standardfunktionen (automatische Codeformatierung, kontextabhängige Hilfe) besteht der Hauptvorteil darin, dass in einem Fensterbereich programmiert werden kann und sich dann flexibel und schnell einzelne Befehle, Funktionen, Teile oder ganze Programme an R senden lassen. Ergebnisse von R werden in diesem Fall in einem zweiten Fensterbereich in Emacs oder der ausgewählten Grafikumgebungen ausgeben.

Auf diese Weise ist es z.B. auch möglich R-Prozesse auf entfernten Rechnern zu steuern.

¹ <http://de.wikibooks.org/wiki/>

Näheres zu ESS findet sich unter: <http://ess.r-project.org/>

Tip

Wenn Sie ESS unter Mac OS X mit dem für die GUI-Version standardmässig verwendeten Quartz-Window-Manager verwenden wollen, dann empfiehlt sich die Installation und Einbindung des Pakets CarbonEL und setzen von `options(device="quartz")`.

23. Automation und dynamische Dokumenterstellung

24. Batchmode

Mit R kann der gesamte Funktionsumfang auch automatisiert werden. Dazu wird lediglich eine Source-Datei mit Programmcode benötigt, die dann z.B. folgendermaßen unter LINUX zur Ausführung gebracht werden kann:

```
bash> R --vanilla --slave < /verzeichnis/meine_sourcefile.r
```

Auf diese Weise lassen sich ohne komplizierte Navigation durch Menüs usw. aktuelle Graphiken erstellen oder Berechnungen durchführen und (z.B. via cron) automatisieren.

25. Dynamische Dokumenterstellung mit dem Paket (S-/odf-)Weave

Statistiken sind kein Selbstzweck. Zumeist steht am Ende statistischer Auswertungen ein Bericht oder eine wie auch immer geartete Veröffentlichung der Ergebnisse. Mit dem Paket Sweave ist es möglich den Prozess der Statistischen Auswertung und der Dokumentation dieser Arbeit in einem Workflow zu behandeln. Die Idee ist, dass R-Befehle in ein Dokument eingefügt werden und durch statistische Auswertungen ersetzt werden. Dadurch lassen sich leicht aktualisierbare Berichte erstellen oder Forschungsarbeiten schon während der statistischen Auswertung erstellen. Sweave ist auch ein großer Schritt in Richtung replizierbarer statistischer Auswertungen, da sich Daten, Auswertungsprozeduren und Auswertungsergebnisse in einem Dokument ablegen lassen.

Derzeit ist dies für drei Dokumentformate möglich: .odf , .tex und html.

Open document Format (.odt) ist das Standardformat der freien Textverarbeitung **Open Office** bzw. **LibreOffice**

Tex ist ein relativ altes und etabliertes Textsatzsystem, das vom gewohnten What-you-see-is-what-you-get zu Gunsten des What-you-see-is-what-you-mean-Paradigmas abweicht. Text wird hier von Anweisungen und Tags umgeben, die dem System sagen, wie ein bestimmter Textteil zu setzen ist. Dadurch können während des Schreibens Layout-Fragen im Hintergrund bleiben.

odfWeave ist ein eigenes Paket, das (noch) nicht Teil der Standardinstallation ist und muss manuell nachinstalliert werden:

```
> install.packages1("odfWeave", dependencies=TRUE)
> library2(utils) # Laden der neuinstallierten Pakete
> library(odfWeave)
```

Sweave ist Teil des `utils`-Paketes und muss ggf. ebenfalls installiert werden.

25.1. Grundprinzipien dynamischer Dokumenterstellung

Die Idee hinter (S)Weave ist folgende: Wir erstellen ein Dokument. An bestimmten Stelle geben wir eine Anweisung, dass eine Statistik oder Graphik eingefügt werden soll. Bei Bedarf lassen wir R das Dokument berechnen und erhalten ein Dokument mit Tabellen, Diagrammen usw.

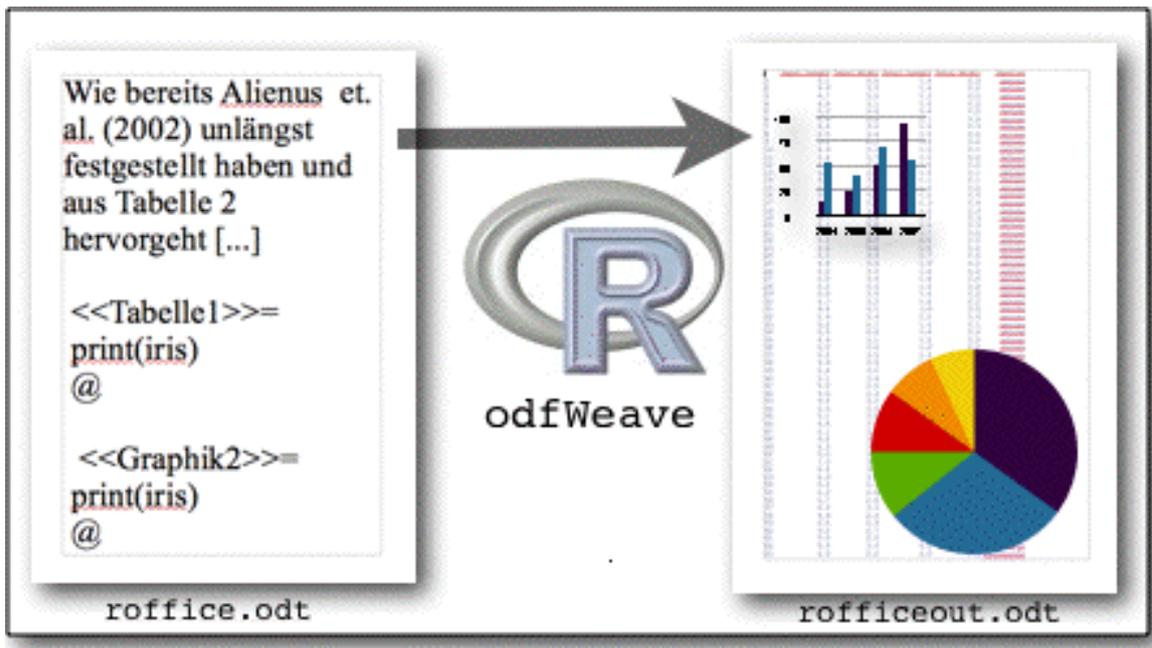


Abb. 2

Die Anweisungen an R werden bei Verwendung des Sweave-Pakets typischerweise in einer charakteristischen Form gegeben, die der Noweb-Syntax folgen. Dabei handelt es sich um Blöcke (genannt Chunks) mit folgender Syntax:

<code><<Blockname,Aufrufparameter,... >>=</code>	#1. Startteil
<code>#R Befehle</code> <code>print(iris)</code>	#2. Anweisungen
<code>@</code>	#3. Schlussteil

Zu beachten ist die dreiteilige Struktur: Der Startteil hat die Form `<<... >>`, der Anweisungsteil besteht aus gewöhnlichen R Befehlen und zum Schluss jedes Chunks steht als charakteristisches Erkennungszeichen ein `@`

25.2. R und Open Office

Öffnen Sie ein Open-Office-Dokument, fügen Sie an einer beliebigen Stelle das folgende Chunk ein:

<code><<Unser_erster_Testoutput,echo=FALSE>>=</code> <code>print(iris) #Ausgabe des Datensatz iris</code> <code>@</code>
--

Speichern Sie das neu erstellte Dokument zum Beispiel unter dem Namen "roffice.odt".

In R können Sie jetzt mit dem Befehl

```
> odfWeave3("/pfad/zur/datei/roffice.odt", "/pfad/zur/datei/rofficeout.odt")
```

alle Chunks in dem Office Dokument `roffice.odt` durch die entsprechenden Berechnungen ersetzen lassen. In unserem Fall sollte das neu erstellte Dokument `rofficeout.odt` nach dem Öffnen in Open Office den Ausdruck des Iris-Datensatzes enthalten. Damit dürfte das wesentliche Prinzip deutlich geworden sein. Da Open Office auch problemlos in das derzeit noch verbreitete `doc`-Format exportieren kann, bietet `odfWeave` die entscheidende Schnittstelle für alle R-Nutzerinnen, die auf den Austausch mit Windows/MS Office-Produkten angewiesen sind.

Bleibt der Vollständigkeit halber noch der Parameter `echo` zu erklären. `echo=TRUE` hätte zur Folge gehabt, dass die Anweisung `print(iris)` mit in das Dokument `rofficeout.odt` geschrieben worden wäre.

25.3. R und Latex

Zur Erzeugung von (La)Tex-Dokumenten geht man identisch vor wie bei `odf`-Dateien. In eine `tex`-Datei werden chunks integriert und in einem zweiten Schritt mit R durch die Ergebnisse ersetzt. Konventionell enden Tex-Sweave-Dateien allerdings mit der Endung `.Rnw` (für R - Noweb) und statt `odfWeave` kommt Sweave zum Einsatz:

```
> Sweave4("texfile.Rnw")
```

generiert die Datei `"texfile.tex"` im selben Verzeichnis. Diese kann dann mit Latex in verschiedenste Formate wie Postscript oder PDF gebracht werden.

Unter LINUX lässt sich der gesamte Arbeitsschritt folgendermaßen automatisieren:

```
bash> echo "library(\"utils\"); Sweave(\"/Pfad/Dateiname.Rnw\")" | R
--no-save --no-restore ;
bash> latex /Pfad/Dateiname.tex
```

25.4. Dynamische Webseiten mit R

Das Paket `R2HTML` implementiert den Sweave-Workflow für Html-Seiten und ermöglicht damit weltweit abrufbare und automatisch aktualisierte statistische Auswertungen.

25.5. Ausblick

So weit die grundlegende Funktionsweise von `odfWeave` und `Sweave`. Der eigentliche Clou ist jedoch sicher, dass auf diese Weise automatisiert Graphiken und formatierte Tabellen eingebunden werden können, was in Kürze in einem der nächsten Kapitel ausgeführt werden wird.

Bis dahin sind die entsprechenden Hilfeseiten zu empfehlen:

```
> ?Sweave
> ?odfWeave
```

(Und wer sich schon einarbeitet, kann natürlich auch diesen Artikel ergänzen. (-;)

25.6. Weblinks

- [odfWeave-Manual \(PDF, englisch\)](#)⁵
- [Wikipedia-Artikel: Open Document Format \(odf / odt\)](#)⁶
- [Wikipedia-Artikel: Open Office](#)⁷
- [Wikipedia-Artikel: TeX](#)⁸

5 <http://cran.at.r-project.org/web/packages/odfWeave/odfWeave.pdf>

6 <http://de.wikipedia.org/wiki/OpenDocument>

7 <http://de.wikipedia.org/wiki/Open%20Office>

8 <http://de.wikipedia.org/wiki/TeX>

Teil III.

Statistik mit R

26. Diagramme und Grafiken erzeugen

Mit R können die verschiedensten Diagramme und Grafiken erstellt werden.

26.1. Funktionen zum Erstellen von Diagrammen

- `barplot()`¹ -- Erstellen von Säulendiagrammen
- `boxplot()`² -- Erstellen von Boxplots (beinhaltet: Maximalwert, Minimalwert, Median, Quartile, Ausreißer)
- `contour()`³ -- Erstellen von Konturdiagrammen, plotten von Isolinien
- `dotchart()`⁴ -- Erstellen von Punktediagrammen
- `forestplot`⁵ - (aus dem Zusatzpaket "rmeta") erzeugt ein so genanntes "Forest Plot" zusammen mit einer Texttabelle.
- `hist()`⁶ -- Erstellen von Histogrammen
- `map`⁷ - (aus dem Paket "maps" und "mapdata") erstellt Karten von Ländern, Kontinenten und der Welt
- `metaplot`⁸ - (aus dem Zusatzpaket "rmeta") erzeugt ein so genanntes "Forest Plot" (Meta-Analyse-Plot), welches im Rahmen von Metaanalysen gängig ist.
- `par()`⁹ -- Setzen von grafischen Parametern
- `persp()`¹⁰ -- Erstellen von Dreidimensionalen Abbildungen
- `pie()`¹¹ -- Erstellen von Kreisdiagrammen
- `plot()`¹² -- Standardfunktion zum Erstellen von Diagrammen
- `title()`¹³ -- Beschriftung von Diagrammen

1 http://de.wikibooks.org/wiki/GNU_R%3A%20barplot
2 http://de.wikibooks.org/wiki/GNU_R%3A%20boxplot
3 http://de.wikibooks.org/wiki/GNU_R%3A%20contour
4 http://de.wikibooks.org/wiki/GNU_R%3A%20dotchart
5 http://de.wikibooks.org/wiki/GNU_R%3A%20forestplot
6 http://de.wikibooks.org/wiki/GNU_R%3A%20hist
7 <http://de.wikibooks.org/wiki/GNU%20R%3A%20map>
8 http://de.wikibooks.org/wiki/GNU_R%3A%20metaplot
9 http://de.wikibooks.org/wiki/GNU_R%3A%20par
10 http://de.wikibooks.org/wiki/GNU_R%3A%20persp
11 http://de.wikibooks.org/wiki/GNU_R%3A%20pie
12 http://de.wikibooks.org/wiki/GNU_R%3A%20plot
13 http://de.wikibooks.org/wiki/GNU_R%3A%20title

26.2. Funktionen zum Erstellen zusätzlicher grafischer Elemente

- `colors()`¹⁴ zeigt eine Übersicht aller Standardfarben, genauer gesagt ihrer Farbwörter (blue, red, green, ...) an.
- `lines(x,y)`¹⁵ - Zum Zeichnen von beliebigen Linienzügen und Funktionen.
- `polygon(x,y)`¹⁶ - erzeugt ein (beliebiges) Vieleck, das schraffiert oder mit Farbe gefüllt werden kann.
- `plotmath`¹⁷ - fügt einem Plot mathematische Symbol hinzu

26.3. Grafiken speichern

- `jpeg()`¹⁸ - speichert die Grafik als jpeg-Datei ab.
- `png()`¹⁹ - speichert die Grafik als png-Datei ab.
- `pdf()`²⁰ -- speichert eine Grafik als PDF-Datei ab.
- `postscript()`²¹ -- Speichert die Grafikausgabe in eine Postscript-Datei.
- `savePlot()`²² -- Speichert die aktuelle Grafik in eine Datei
- `devSVG()`²³ -- speichert eine Grafik als SVG-Datei ab. Dazu ist das Zusatzpaket `RSvgDevice` erforderlich.

26.4. siehe auch

- Grafikbeispiele²⁴

26.5. Literatur

- Paul Murrell (2005): "R Graphics", Chapman & Hall, 301 Seiten, ISBN 158488486X

26.6. Weblinks

- Beispiele von Benutzer Thire in der Wikipedia²⁵

14 <http://de.wikibooks.org/wiki/GNU%20R%3A%20colors>

15 http://de.wikibooks.org/wiki/GNU_R%3A_lines

16 http://de.wikibooks.org/wiki/GNU_R%3A_polygon

17 <http://de.wikibooks.org/wiki/GNU%20R%3A%20plotmath>

18 <http://de.wikibooks.org/wiki/GNU%20R%3A%20jpeg>

19 <http://de.wikibooks.org/wiki/GNU%20R%3A%20png>

20 http://de.wikibooks.org/wiki/GNU_R%3A%20pdf

21 http://de.wikibooks.org/wiki/GNU_R%3A%20postscript

22 http://de.wikibooks.org/wiki/GNU_R%3A%20savePlot

23 http://de.wikibooks.org/wiki/GNU_R%3A%20devSVG

24 Kapitel 49 auf Seite 187

25 http://de.wikipedia.org/wiki/Benutzer%3AThire%2FBilder%23Mathematische_Grafiken

- [Category:Created_with_R²⁶](#) in Wikimedia Commons
- [R Graphics by Paul Murrell²⁷](#)

²⁶ http://commons.wikimedia.org/wiki/Category:Created_with_R

²⁷ <http://www.stat.auckland.ac.nz/~paul/RGraphics/rgraphics.html>

27. Deskriptive Statistik

Dieses Kapitel gibt einen kurzen Überblick über gängige Verfahren der deskriptiven Statistik. Deskriptive Statistik versucht im wesentlichen die Eigenschaften einer grossen Anzahl von Fällen in möglichst charakteristische Kennwerte zusammenzufassen.

Als Beispiel verwenden wir den Datensatz `Bundesliga` im Paket `wikibooks`¹:

```
library(wikibooks)
data(Bundesliga)
```

27.0.1. Überblick über den Datensatz

Wie `class(Bundesliga)` zeigt, ist `Bundesliga` ein Objekt vom Typ `data.frame`². Ein Überblick über die enthaltenen Variablen liefert:

```
R> names(Bundesliga)
 [1] "Saison"           "Spieltag"         "Datum"
 [4] "Anpfiff"          "Heim"             "Gast"
 [7] "Tore.Heim"        "Tore.Gast"        "Tore.Heim.Halbzeit"
[10] "Tore.Gast.Halbzeit"
```

Einen ersten Eindruck über den ganzen `data.frame` liefert `str`³(`Bundesliga`) hinsichtlich der Variablentypen und `summary`⁴(`Bundesliga`) hinsichtlich deren typischen Ausprägungen.

27.0.2. Mittelwert, Extremwerte und Streuung

Mittelwerte

Angenommen uns interessieren zunächst die Tore pro Spiel (also der Tore der Heim- und Gast-Mannschaft zusammen):

```
R> tore <- Bundesliga$Tore.Heim + Bundesliga$Tore.Gast
```

1 Kapitel 57 auf Seite 243

2 <http://de.wikibooks.org/wiki/GNU%20R%3A%20data.frame>

3 <http://de.wikibooks.org/wiki/GNU%20R%3A%20str>

4 <http://de.wikibooks.org/wiki/GNU%20R%3A%20summary>

Das **arithmetische Mittel** $\bar{x} = \frac{\sum_{i=1}^n x_i}{n}$ (durchschnittliche Anzahl an Toren pro Begegnung) errechnet sich dann als

```
R> mean5(tore)
[1] 3.0991
```

Den **Zentralwert (Median)** als ebenfalls gebräuchlicher Mittelwert kann auch für ordinale Daten berechnet werden und liefert den Wert, der die 50% kleineren von den 50% grösseren Variablenwerte trennt:

```
R> median6(tore)
[1] 3
```

Die Methode des Median lässt sich durch **Quantile** beliebig verallgemeinern:

```
R> quantile(tore,c(0,0.5,1))
0% 50% 100%
0 3 12
R> quantile(tore,seq(0.1,by=0.1))
10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
1 2 2 2 3 3 4 5 6 12
```

Was war die geringste (**Minimum**) und höchste Zahl (**Maximum**) an Toren in einem Spiel?

```
R> cat7( min8(tore), max9(tore) )
0 12
```

Minimum und Maximum liefert auch die Funktion `range`.

Wie im Kapitel Umgang mit Datensätzen¹⁰ beschrieben können wir uns die entsprechenden Fälle jederzeit anzeigen lassen:

```
Bundesliga[tore==max(tore),]
```

Streuung

Die **Spannweite** zwischen höchstem und niedrigstem Wert $\max(x) - \min(x)$ erhält man durch:

```
diff11( range12(tore) )
```

Das gebräuchlichste Maß für die Streuung einer Variablen ist die **Varianz**:

$$\text{var}(x) = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}$$

¹⁰ http://de.wikibooks.org/wiki/GNU_R%3A_Umgang_mit_Datens%2528Erstellen%252C_Ausw%2528hlen_und_Filtern%2529

In Bezug auf unser Beispiel errechnet sich die Varianz entsprechend als:

```
R> (sum( (tore-mean(tore) ) ^2 ) / (length(tore) )
[1] 3.4091
```

Natürlich verfügt *R* über diese Standardfunktion. Allerdings berechnet die Funktion `var`¹³ die inferenzstatistische, davon abweichend definierte Variante der Varianz:

$$var_{inf}(x) = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1} = var(x) \cdot \frac{n}{n-1}$$

Somit ermitteln wir die deskriptive Varianz am einfachsten mit:

```
R> var(tore) * (length(tore)-1) / length(tore)
[1] 3.4091
```

Das selbe Problem ergibt sich hinsichtlich der **Standardabweichung**, die als Quadratwurzel der eben berechneten Varianz definiert ist. Die Funktion `sd`¹⁴ liefert nur die inferenzstatistische Variante.

Daher empfiehlt sich bei häufigem Gebrauch die Definition folgender Funktionen¹⁵:

```
varianz <- function(x) {n=length(x) ; var(x) * (n-1) / n}
stdabw <- function(x) {n=length(x) ; sqrt(var(x) * (n-1) / n)}
```

Damit kann die (deskriptive) Standardabweichung leichter berechnet werden:

```
stdabw(tore)
[1] 1.8464
```

27.0.3. Häufigkeitstabellen und Prozenttabellen

Die Funktion `table` fasst verschiedene Kategorien zusammen und erstellt daraus eine **Häufigkeitstabelle**:

```
R> table(tore)
tore
 0   1   2   3   4   5   6   7   8   9  10  11  12
880 1627 3057 2670 2438 1319 828 358 146 44 24 10 5
```

Die entsprechende **Prozentwert-Tabelle** erhält man, wenn man die Absolutwerte durch die Gesamtzahl der Begegnungen n (=Länge des Vektors `Tore`) teilt und mit 100 multipliziert. Der Übersichtlichkeit halber runden wir mit `round`¹⁶ noch auf 2 Nachkommastelle:

¹³ <http://de.wikibooks.org/wiki/GNU%20R%3A%20var>

¹⁴ <http://de.wikibooks.org/wiki/GNU%20R%3A%20sd>

¹⁵ Kapitel 42 auf Seite 149

¹⁶ <http://de.wikibooks.org/wiki/GNU%20R%3A%20round>

```
R> round( table(tore) / length(tore) *100 ,2)
tore
  0    1    2    3    4    5    6    7    8    9   10
11   12
6.56 12.14 22.80 19.92 18.19  9.84  6.18  2.67  1.09  0.33  0.18
0.07  0.04
```

Eine schöne Formatierungsalternative stellt auch die Funktion `describe`¹⁷ zur Verfügung, wenn das Paket `Hmisc` installiert wurde:

```
R> describe18(tore)
tore
  n missing  unique   Mean   .05   .10   .25   .50
.75  .90   .95
13406    0    13   3.099    0    1    2    3
 4    6    6
Frequency  0    1    2    3    4    5    6    7    8    9 10 11 12
%          7   12   23   20   18   10    6    3    1  0  0  0  0:
```

27.0.4. Mehrdimensionale Häufigkeitsverteilungen

Die Funktion `table`¹⁹(Zeilenvariable, Spaltenvariable, weitere Dimensionen) erstellt auch **konditionale Häufigkeitstabellen**. Damit lassen sich zum Beispiel die Tore in Abhängigkeit von der Saison darstellen. Eine andere Möglichkeit ist `ftable`, die auch die Formelnotation

(abhängige Var. ~ 1. unabhängig. + 2. unabhängig. + ...)

akzeptiert:

```
ftable(tore~Saison, data=Bundesliga)
      tore  0  1  2  3  4  5  6  7  8  9 10 11 12
Saison
1963/1964   13 19 45 46 49 30 20 10  4  1  1  1  1
1964/1965   12 26 46 56 42 29 18  4  3  3  1  0  0
1965/1966   21 39 65 59 43 41 19 12  1  3  2  1  0
[...]
```

27.1. Gruppenspezifische Auswertung von Variablen

Angenommen uns interessiert für Datensatz `bsp4` der mittlere Wert des Einkommens abhängig vom Geschlecht. Mit der Funktion `tapply(VARIABLE, GRUPPENVARIABLE, FUNKTION)`²⁰ ist dies besonders einfach zu berechnen.

17 <http://de.wikibooks.org/wiki/GNU%20R%3A%20describe>

19 <http://de.wikibooks.org/wiki/GNU%20R%3A%20table>

20 <http://de.wikibooks.org/wiki/GNU%20R%3A%20tapply>

```
> tapply(bsp4$Einkommen, bsp4$Geschlecht, mean)
```

Dabei ist das erste Argument die betrachtete Variable und das zweite Argument eine Variable, die die Gruppenzugehörigkeit bezeichnet (in unserem Fall das Geschlecht). Als 3. Argument kann eine beliebige Funktion bestimmt werden. Das ist im Falle der Funktion `mean` ein benannter Vektor:

intersexuell	maennlich	weiblich
2400.00	1166.50	1899.75

28. Regressionsanalyse

28.1. Einleitung

Mit Regressionen wird versucht eine abhängige, metrische Variable in Abhängigkeit von einer oder mehreren unabhängigen Variablen zu beschreiben. Die abhängige Variable soll dadurch üblicherweise kausal auf die Effekte andere Variablen zurückgeführt werden. (Bspw. Regression der persönlichen Laune abhängig vom Wetter). Es gibt zum Teil recht unterschiedliche Regressionsverfahren und R stellt eine Vielzahl an Methoden bereit. Die einfachste Variante eines Regressionsmodells ist die lineare Regression.

28.2. Lineare Regression

28.2.1. Ein erstes Beispiel: Lebensalter und Gewicht

Als erstes Beispiel verwenden wir den Datensatz aus Beispiel 5¹, welcher das Gewicht eines Babys an verschiedenen Lebenstagen enthält. Zunächst stellen wir die Daten mit `plot()`² graphisch dar:

```
plot3(bsp5, type="o", main="Gewicht des Babys", ylab="Gewicht in Gramm",  
xlab="Lebenstag", col="blue", lwd=2)
```

Nun fügen wir die Regressionsgeraden hinzu, indem wir die Funktion `lm(Y~X)`⁴ mit dem Befehl `abline()`⁵ in die Graphik integrieren.

- Y ist in diesem Falle die Spalte des Gewichts (also hier: `bsp5[,2]`)
- X ist in diesem Falle die Spalte der Lebensstage (also hier: `bsp5[,1]`)

Der Befehl lautet demzufolge:

```
abline6(lm7(bsp5[,2]~bsp5[,1]))
```

1 Kapitel 48.5 auf Seite 173

2 <http://de.wikibooks.org/wiki/GNU%20R%3A%20plot>

4 <http://de.wikibooks.org/wiki/GNU%20R%3A%20lm>

5 <http://de.wikibooks.org/wiki/GNU%20R%3A%20abline>

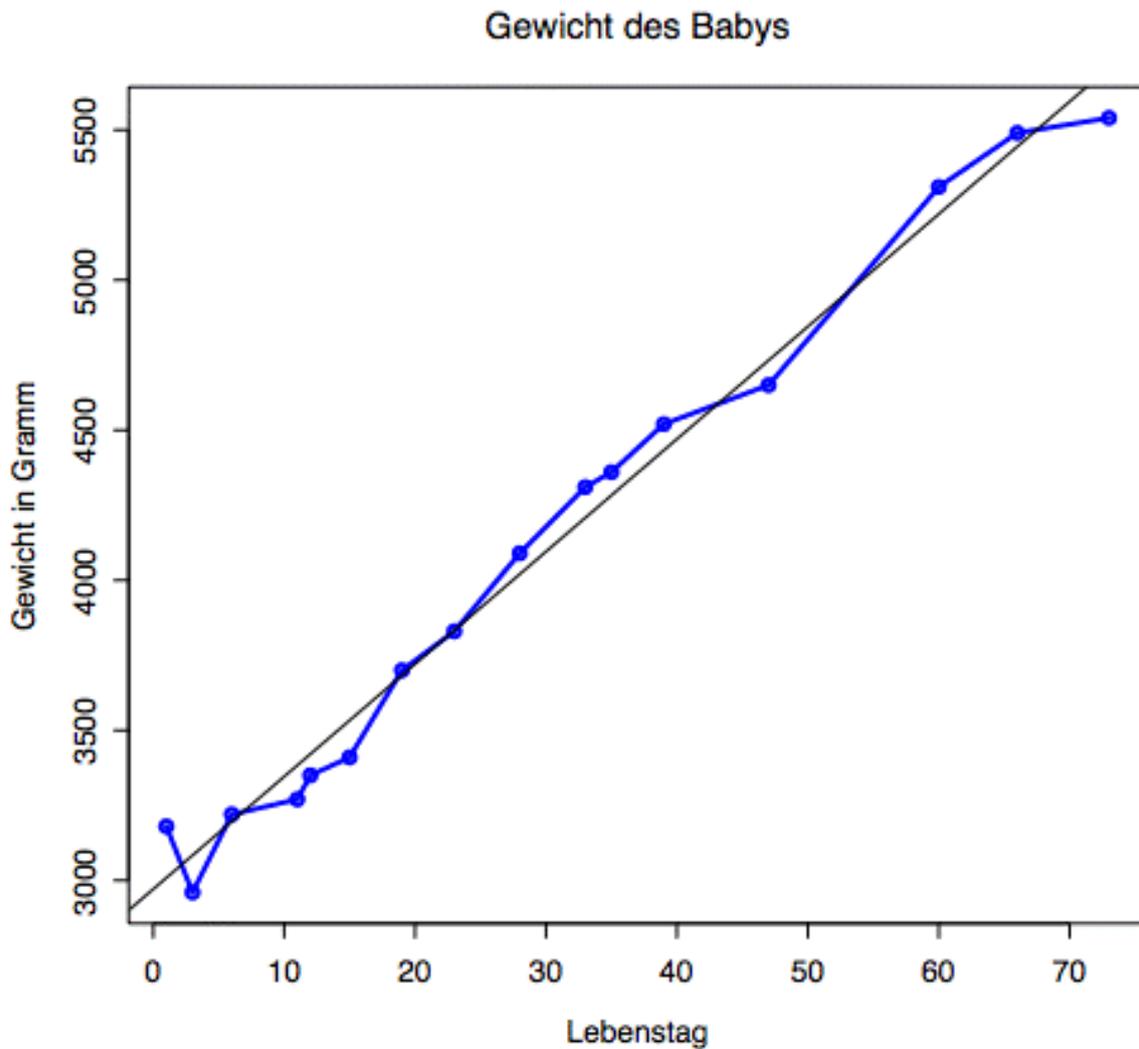


Abb. 3

Alternative mit xyplot

Um eine Regressionsgerade durch eine Punktwolke zu ziehen, kann auch der Befehl `xyplot` aus dem Zusatzpaket `lattice` verwendet werden. Falls `lattice` noch nicht installiert ist, muss dies natürlich nun nachgeholt werden:

```
install.packages("lattice", dependencies = T)
```

Anschließend aktivieren wir das `lattice`-Paket:

```
library(lattice)
```

Eine Regressionsgerade erhält man durch:

```
xypplot10(y~x, type=c11("p","r"))
```

Angewendet auf Beispiel 5¹² lautet der Befehl:

```
xypplot13(bsp5[,2]~bsp5[,1], type=c14("p","r"))
```

Wir erhalten nun eine Punktwolke (type="p") der Daten, durch welche die Regressionsgerade (type="r") verläuft.

28.2.2. Lineare Modelle mit R

Bei linearer Modellierung wird angenommen, dass die Effekte verschiedener unabhängiger Variablen folgendermassen aufaddiert werden können: $\hat{Y} = b_0 + b_1 \cdot X_1 + b_2 \cdot X_2 + \dots + b_n \cdot X_n$

In R können zur Berechnung linearer Modelle zum Beispiel die Funktionen lm¹⁵ (für "linear model") oder glm¹⁶ (für "generalized linear model") genutzt werden. Diese verwenden für den obigen Ausdruck die folgende **Formelnotation**:

$Y \sim \text{Variable_X1} + \text{Variable_X2} + \dots$

Die Funktion lm berechnet eine Reihe von Statistiken für dieses Modell, die allerdings in der für diese Funktion voreingestellten Ausgabemethode¹⁷ print.lm unterdrückt werden. Ausgegeben werden standardmässig nur die Aufrufbedingungen ("Call:") und die Koeffizienten:

```
R> lm(bsp5$Gewicht~bsp5$Lebenstag)

Call:
lm(formula = bsp5$Gewicht ~ bsp5$Lebenstag)

Coefficients:
(Intercept)  bsp5$Lebenstag
 2971.1         37.5
```

Eine sehr viel detailliertere Auswertung bekommt man zum Beispiel durch Verwendung der Funktion summary¹⁸:

```
R> summary19( lm20( bsp5$Gewicht ~ bsp5$Lebenstag ) )

Call:
lm(formula = bsp5$Gewicht ~ bsp5$Lebenstag)
```

12 Kapitel 48.5 auf Seite 173

15 <http://de.wikibooks.org/wiki/GNU%20R%3A%20lm>

16 <http://de.wikibooks.org/wiki/GNU%20R%3A%20glm>

17 <http://de.wikibooks.org/wiki/GNU%20R%3A%20print>

18 <http://de.wikibooks.org/wiki/GNU%20R%3A%20summary>

Residuals:				
Min	1Q	Median	3Q	Max
-167.0	-90.3	20.4	79.7	171.4

Coefficients:				
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2971.13	43.69	68.0	< 2e-16 ***
bsp5\$Lebenstag	37.48	1.19	31.5	2.1e-14 ***

 Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 104 on 14 degrees of freedom
 Multiple R-Squared: 0.986, Adjusted R-squared: 0.985
 F-statistic: 991 on 1 and 14 DF, p-value: 2.15e-14

Residuals

- Hier sind die Quartile der Residuen angegeben.

Coefficients

- In der ersten Spalte werden die Konstante (Intercept) und die Regressionskoeffizienten des Modells ausgegeben. (Vorsicht beim Vergleich: Dabei handelt es sich nicht um Beta-Werte)

In den folgenden Spalte werden (für die Beispieldaten selbstverständlich sinnlose) inferenzstatistische Masse (Masse für die Verallgemeinerbarkeit der Stichprobe auf die Grundgesamtheit) angezeigt:

- Für jeden der i Koeffizienten wird neben der Berechnung Standardfehler (Std. Error) ein t-Test mit der Null-Hypothese $H_0 : b_i = 0$ durchgeführt, für den der entsprechende t-Wert und die Signifikanz $Pr(>|t|)$ angegeben wird.

Größen, die das Gesamtmodell betreffen:

- Der Standardfehler der Gesamtschätzung (Residual standard error).
- Das Bestimmtheitsmass R^2 (Multiple R-Square) und das um die Anzahl der Modellvariablen Korrigierte Bestimmtheitsmass $R^2_{kor.}$ (Adjusted R-squared) geben an wieviel Prozent der Varianz der Residuen von den realen Werten durch das Modell erklärt wird (in welchem Umfang also die Schätzung von \hat{Y} von den realen Werten abweicht) . Bei einem Wert von 1 liegen alle Punkte auf der Gerade / Fläche. Je näher an 0 sich der Wert annähert, umso mehr streuen die Messwerte um den angenommenen Verlauf.
- Die Ergebnisse des F-Test in der letzten Zeile prüfen die Verallgemeinerbarkeit des Gesamtmodells (F-Wert, Freiheitsgrade und Signifikanz).

29. Clusteranalyse

Clusteranalyse-Algorithmen finden sich in verschiedenen Paketen von R. Eine Übersicht findet sich in der Taskview `Cluster`¹.

- k-Means Algorithmen (Standardpaket `stats`):

```
kmeans(x, centers, iter.max = 10, nstart = 1, algorithm =  
c("Hartigan-Wong", "Lloyd", "Forgy", "MacQueen"))
```

- Fuzzy C-Means Algorithmen (Paket `e1071`):

```
cmeans(x, centers, iter.max = 100, verbose = FALSE, dist =  
"euclidean", method = "cmeans", m = 2, rate.par = NULL, weights = 1,  
control = list())
```

- Hierarchische Clusteranalyse (Standardpaket `stats`):

```
hclust(d, method = "complete", members=NULL)
```

- Zur Berechnung der Distanzen `d` (Standardpaket `stats`):

```
dist(x, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)
```

¹ <http://cran.r-project.org/web/views/Cluster.html>

30. Signifikanztests

Mit \mathbf{R} lassen sich diverse Signifikanztests durchführen.

30.1. Testauswahl

Bei der Auswahl des geeigneten Verfahrens ist von entscheidender Bedeutung:

1. das Datenniveau (nominal-ordinal-metrisch)
2. ob die Daten normalverteilt sind
3. die Stichprobe (verbunden/unverbunden)

NOMINAL		ORDINAL		METRISCH	
unabhängig	abhängig	unabhängig	abhängig	normalverteilt	abhängig
χ^2 für: k x 1	χ^2 McNemar-Test für: 2 x 2	Mann-Whitney	Wilcoxon	F-Test (Varianzquotiententest) entscheidet über: Varianzhomogenität	t-Test für verbundene Stichproben
-Felder	2 x 2 Felder			Welch-Test	
nichtparametrische Testverfahren					
parametrische Testverfahren					

30.2. Testdurchführung in R

Test	Durchführung in R
Chi-Quadrat-Test (χ^2)	<code>chisq.test</code> ¹
Fisher-Exact-Test	<code>fisher.test</code> ²
Kolmogoroff-Smirnov-Test	<code>ks.test</code> ³
Mann-Whitney-Test	<code>wilcox.test</code> ⁴ , mit spezieller Optionseinstellung <code>wilcox_test</code> ⁵ aus dem <code>coin</code> -Package
Shapiro-Wilk-Test	<code>shapiro.test</code> ⁶
t-Test	<code>t.test</code> ⁷
Varianzquotienten-Test (<i>alias</i> F-Test)	<code>var.test</code> ⁸
Welch-Test	<code>t.test</code> ⁹ , mit spezieller Optionseinstellung
Wilcoxon-Test	<code>wilcox.test</code> ¹⁰

30.3. Weblinks

- Signifikanztests bei Wikipedia¹¹

1 <http://de.wikibooks.org/wiki/GNU%20R%3A%20chisq.test>
2 http://de.wikibooks.org/wiki/GNU_R%3A_fisher.test
3 http://de.wikibooks.org/wiki/GNU_R%3A_ks.test
4 <http://de.wikibooks.org/wiki/GNU%20R%3A%20wilcox.test%23Mann-Whitney-Test>
5 http://de.wikibooks.org/wiki/GNU%20R%3A%20wilcox_test
6 http://de.wikibooks.org/wiki/GNU_R%3A_shapiro.test
7 http://de.wikibooks.org/wiki/GNU_R%3A_t.test
8 <http://de.wikibooks.org/wiki/GNU%20R%3A%20var.test>
9 http://de.wikibooks.org/wiki/GNU_R%3A_t.test%23Zweistichprobe
10 <http://de.wikibooks.org/wiki/GNU%20R%3A%20wilcox.test>
11 <http://de.wikipedia.org/wiki/Signifikanztest>

31. Rasch-Modelle

Für **R** liegen Zusatzpakete bereit, mit deren Hilfe diverse Rasch-Modelle¹ analysiert werden können.

¹ <http://de.wikipedia.org/wiki/Rasch-Modell>

32. Package eRm

Mit Hilfe des Package `eRm` (extended Rasch modeling) lassen sich folgende Analysen durchführen:

- das dichotome logistische Rasch-Modell
- das linear-logistische Test-Modell
- das Ratingskalen-Modell
- das partial-credit Modell
- das lineare Ratingskalen-Modell
- das lineare partial-credit Modell

32.1. Installation

Das Package wird installiert mit dem Befehl (Unter Linux ist `gcc-fortran` als Voraussetzung zu installieren):

```
install.packages1("eRm", dependencies=TRUE)
```

Mit Eingabe des Befehls werden die benötigten Zusatzpakete `gtools`, `splines`, `ROCR`, `gplots`, `gdata` mitinstalliert (sofern sie bislang noch nicht installiert wurden).

Vor der Nutzung des Packages muss es erst aktiviert werden per:

```
library2(eRm)
```

32.2. Das dichotome logistische Rasch-Modell

Der Befehl zur Analyse nach dem dichotomen logistischen Rasch-Modell lautet `RM()`.

Als Untersuchungsobjekt für `RM()` wird eine Matrix benötigt, welche dichotome Daten enthält (0 und 1). Die Reihen der Matrix entsprechen den Untersuchungsfällen (n) und die Spalten entsprechen den Ergebnissen der Items.

32.2.1. `RM()`

An den `RM()`-Befehl können folgende Parameter übergeben werden:

RM(X, W, se=TRUE, sum0=TRUE, etaStart)

- X = die zu analysierende Datenmatrix
- W = optionale Designmatrix zur Normierung auf ein bestimmtes Items
- se = Berechnung der Standardfehler (TRUE=ja / FALSE=nein)
- sum0 = TRUE=Summennormierung / FALSE=Itemnormierung
- etaStart = optionale Übergabe von vermuteten Parameterschätzwerten

32.2.2. Beispieldatensatz

Als Beispieldatensatz sollen die Ergebnisse des Minigolfwettbewerbs aus Beispiel 8³ genommen werden.

- 0 bedeutet, dass das entsprechende Loch nicht mit einem Schlag getroffen wurde
- 1 bedeutet, dass das entsprechende Loch mit einem Schlag getroffen wurde

Die Ergebnisse des Wettbewerbs lauten wie folgt.

	Loch1	Loch2	Loch3	Loch4	Loch5	Loch6	Loch7	Loch8	Loch9
Loch10									
Hans 1	0	1	1	0	1	1	0	1	1
Karola 0	1	0	1	1	1	0	1	1	1
Matthias 0	1	1	1	0	1	1	1	1	0
Stefan 1	0	0	1	1	0	1	1	0	0
Sabine 0	1	1	1	1	1	1	1	0	0
Irma 0	1	1	0	1	1	1	0	1	0
Heike 1	1	0	1	0	1	1	0	0	0
Ralf 0	1	1	1	0	1	1	0	0	0
Rainer 1	1	1	0	1	1	0	0	1	0
Simon 0	1	0	1	1	1	0	1	0	1
Andreas 0	1	1	1	0	1	0	0	0	0
Elke 0	1	1	0	1	0	1	0	0	1
Gabi 1	0	1	1	1	0	0	1	0	0
David 0	1	1	0	1	0	0	0	0	0
Jonas 0	1	1	0	1	1	0	1	1	0
Nicklas 0	1	1	1	1	0	1	0	0	1
Sandra 1	0	1	0	0	1	0	1	1	0
Mario 0	1	0	1	0	1	1	0	0	0

3 Kapitel 48.8 auf Seite 179

Guido	1	1	1	0	0	0	0	0	1
0									
Lisa	0	1	1	0	0	0	1	0	0
0									
Peter	1	0	1	0	1	0	0	0	1
0									
Justus	1	1	1	0	0	0	1	0	0
1									
Josef	1	0	1	0	0	0	0	0	0
0									
Brigitte	1	1	1	0	0	0	1	0	1
0									
Kevin	0	1	0	0	1	0	0	1	0
1									
Marcel	1	1	0	0	0	0	0	0	0
0									
Nadine	1	0	0	1	0	1	0	0	1
0									
Alex	1	0	0	0	0	0	0	0	0
0									
Katharina	0	1	0	0	0	1	1	0	1
1									
Daniel	1	1	0	0	0	0	0	0	0
0									
Jens	1	1	0	1	0	0	1	0	1
0									
Dieter	1	0	0	0	0	0	1	1	0
0									
Sebastian	1	0	1	0	1	0	0	0	1
1									
Anne	0	0	0	0	1	0	1	0	0
1									

Diese Tabelle wird wie folgt erzeugt:

```
minigolf <- structure(list(Loch1 = c(0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,
1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0,
1, 1, 1, 0, 1, 1, 1, 1, 0), Loch2 = c(1, 0, 1, 0, 1, 1, 0, 1, 1, 0,
1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1,
1, 0, 0, 1, 1, 1, 0, 0, 0), Loch3 = c(1, 1, 1, 1, 0, 1, 1, 0, 1,
1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0,
0, 0, 0, 0, 0, 0, 0, 1, 0), Loch4 = c(0, 1, 0, 1, 1, 1, 0, 0, 1, 1,
0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1, 0, 0, 0, 1, 0, 0, 0), Loch5 = c(1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1,
0, 0, 0, 0, 0, 0, 1, 1), Loch6 = c(1, 0, 1, 1, 1, 1, 1, 1, 0, 0,
0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0,
0, 1, 0, 1, 0, 0, 0, 0, 0), Loch7 = c(0, 1, 1, 1, 1, 0, 0, 0, 0, 1,
0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0,
0, 0, 0, 1, 0, 1, 1, 0, 1), Loch8 = c(1, 1, 1, 0, 0, 1, 0, 0, 1,
0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1,
0, 0, 0, 0, 0, 0, 1, 0, 0), Loch9 = c(1, 1, 0, 0, 0, 0, 0, 0, 0, 1,
0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0,
1, 0, 1, 0, 1, 0, 1, 0), Loch10 = c(1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0,
0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0,
0, 1, 0, 0, 0, 1, 1)), .Names = c("Loch1", "Loch2", "Loch3",
"Loch4", "Loch5", "Loch6", "Loch7", "Loch8", "Loch9",
"Loch10"), class = "data.frame", row.names = c("Hans", "Karola",
"Matthias", "Stefan", "Sabine", "Irma", "Heike",
"Ralf", "Rainer", "Simon", "Andreas", "Elke", "Gabi", "David",
"Jonas", "Nicklas", "Sandra", "Mario", "Guido",
"Lisa", "Peter", "Justus", "Josef", "Brigitte", "Kevin", "Marcel",
"Nadine", "Alex", "Katharina", "Daniel", "Jens",
"Dieter", "Sebastian", "Anne"))
```

32.2.3. Parameterschätzung

Die Analyse und Parameterschätzung erfolgt zunächst durch Aufruf der Funktion `RM()`

```
RM(minigolf)
```

Dies liefert folgende Ausgabe:

```
Results of RM estimation:

Call: RM(X = minigolf)

Conditional log-likelihood: -155.7848
Number of iterations: 17
Number of parameters: 9

Basic Parameters eta:
      eta 1      eta 2      eta 3      eta 4      eta 5
eta 6      eta 7      eta 8      eta 9
Estimate 0.7802107 0.4036248 -0.3177005 0.1641012 -0.4439830
-0.07408871 -0.8568075 -0.4439830 -0.5748605
Std.Err  0.3448683 0.3320590 0.3359948 0.3291390 0.3407310
0.33035472 0.3653347 0.3407311 0.3469895
```

Diese Ausgabe zeigt "lediglich" die eta-Parameter mit dazugehöriger Standardabweichung an. Da zu weiteren Analysen der Output des `RM(minigolf)`-Befehls benötigt wird, speichern wir diesen in die Variable `result`

```
result <- RM(minigolf)
```

Einen besseren Überblick über die soeben durchgeführte Analyse bietet die `summary()`⁴-Funktion. Mit ihrem Aufruf werden auch die wichtigen beta-Parameter ausgegeben:

```
summary5(result)
```

Dies liefert folgende Ausgabe:

```
Results of RM estimation:

Call: RM(X = minigolf)

Conditional log-likelihood: -155.7848
Number of iterations: 17
Number of parameters: 9

Basic Parameters (eta) with 0.95 CI:
      Estimate Std. Error lower CI upper CI
eta 1    0.780    0.345    0.104    1.456
eta 2    0.404    0.332   -0.247    1.054
```

⁴ <http://de.wikibooks.org/wiki/GNU%20R%3A%20summary>

eta 3	-0.318	0.336	-0.976	0.341
eta 4	0.164	0.329	-0.481	0.809
eta 5	-0.444	0.341	-1.112	0.224
eta 6	-0.074	0.330	-0.722	0.573
eta 7	-0.857	0.365	-1.573	-0.141
eta 8	-0.444	0.341	-1.112	0.224
eta 9	-0.575	0.347	-1.255	0.105

Item Easiness Parameters (beta) with 0.95 CI:

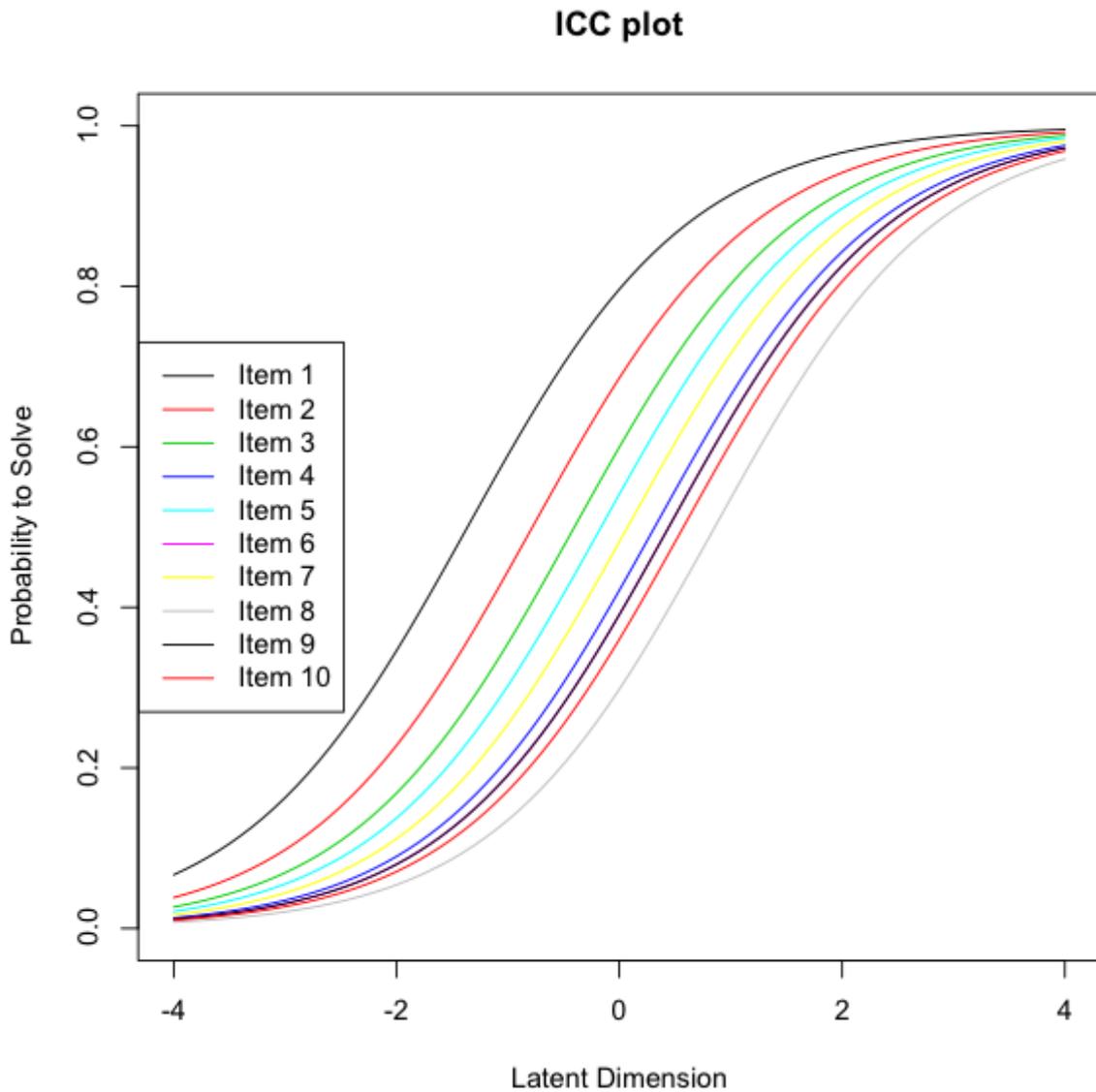
	Estimate	Std. Error	lower CI	upper CI
beta Loch1	1.363	0.385	0.609	2.118
beta Loch2	0.780	0.345	0.104	1.456
beta Loch3	0.404	0.332	-0.247	1.054
beta Loch4	-0.318	0.336	-0.976	0.341
beta Loch5	0.164	0.329	-0.481	0.809
beta Loch6	-0.444	0.341	-1.112	0.224
beta Loch7	-0.074	0.330	-0.722	0.573
beta Loch8	-0.857	0.365	-1.573	-0.141
beta Loch9	-0.444	0.341	-1.112	0.224
beta Loch10	-0.575	0.347	-1.255	0.105

Zur graphischen Darstellung stehen die Funktionen `plotICC()` und `plotjointICC()` zur Verfügung. `plotjointICC()` benötigt das `RM()`-Output und zeigt die Item-Characteristic-Curves aller Variablen an:

```
plotjointICC(RM(minigolf))
```

oder in unserem Fall

```
plotjointICC(result)
```

**Abb. 4**

Mit `plotICC()` werden die Item-Characteristic-Curve für jedes Item einzeln graphisch dargestellt. Hierbei wird durch drücken der Returnntaste zur nächsten Item-Graphik gewechselt.

```
plotICC(result)
```

Im Gegensatz zu `plotjointICC()` kann `plotICC()` auch die Outputs von `PCM()`⁶ und `RSM()`⁷ verarbeiten.

⁶ Kapitel 32.5 auf Seite 120

⁷ Kapitel 32.4 auf Seite 113

Schätzung der Personenparameter

Die Schätzung der Personenparameter erfolgt mit dem Befehl `person.parameter()`. Als Input benötigt der Befehl den Output der `RM()`-Analyse.

```
person.parameter(RM(minigolf))
```

oder in unserem Fall auch

```
person.parameter(result)
```

Dies liefert die Ausgabe:

```
Person Parameters:
  Raw Score   Estimate Std.Error
    0      -3.30890751      NA
    1      -2.37460583  1.0814193
    2      -1.50799727  0.8233509
    3      -0.91866630  0.7238722
    4      -0.43133018  0.6781861
    5       0.01539553  0.6628318
    6       0.45847521  0.6727929
    7       0.93481125  0.7136595
    8       1.41970665      NA
    9       1.90460205      NA
   10       2.38949746      NA
```

Auch hier ist es sinnvoll, den Output in eine Variable zu speichern

```
pp <- person.parameter(result)
```

Die Ausgabe lässt sich mit der `plot()`⁸-Funktion grafisch darstellen:

```
plot(pp)
```

⁸ <http://de.wikibooks.org/wiki/GNU%20R%3A%20plot>

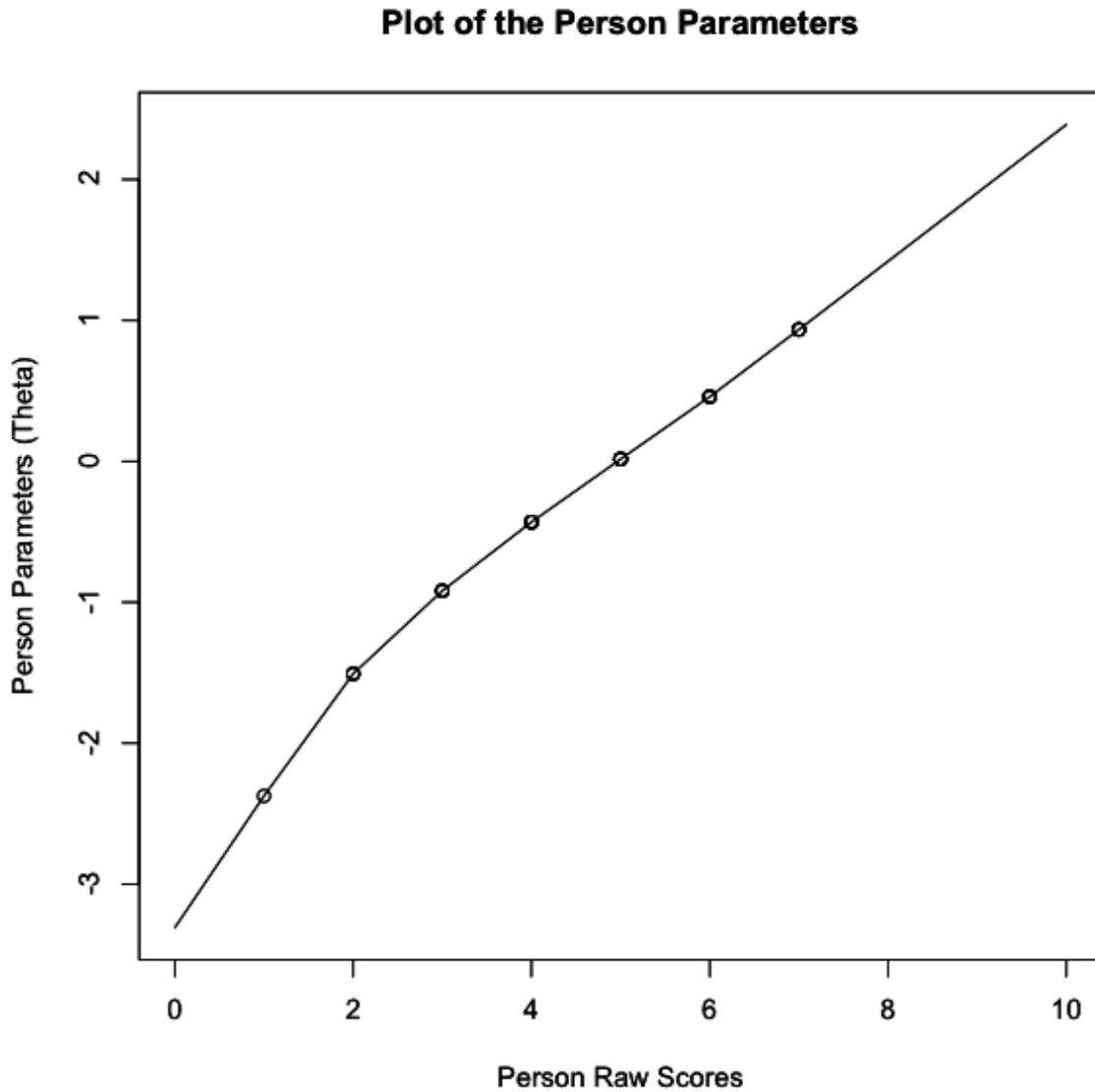


Abb. 5

Ruft man den `summary()`⁹-Befehl auf, erhält man die Personenparameter für jeden einzelnen Probanden:

```
summary(pp)
```

```
Estimation of Ability Parameters  
Collapsed log-likelihood: -37.38117  
Number of iterations: 9  
Number of parameters: 7
```

⁹ <http://de.wikibooks.org/wiki/GNU%20R%3A%20summary>

ML estimated ability parameters (without spline interpolated values):

	Estimate	Std. Err.	2.5 %	97.5 %
theta Hans	0.93481125	0.7136595	-0.4639356	2.3335581
theta Karola	0.93481125	0.7136595	-0.4639356	2.3335581
theta Matthias	0.93481125	0.7136595	-0.4639356	2.3335581
theta Stefan	0.01539553	0.6628318	-1.2837309	1.3145219
theta Sabine	0.93481125	0.7136595	-0.4639356	2.3335581
theta Irma	0.45847521	0.6727929	-0.8601747	1.7771251
theta Heike	0.01539553	0.6628318	-1.2837309	1.3145219
theta Ralf	0.01539553	0.6628318	-1.2837309	1.3145219
theta Rainer	0.45847521	0.6727929	-0.8601747	1.7771251
theta Simon	0.45847521	0.6727929	-0.8601747	1.7771251
theta Andreas	-0.43133018	0.6781861	-1.7605505	0.8978901
theta Elke	0.01539553	0.6628318	-1.2837309	1.3145219
theta Gabi	0.01539553	0.6628318	-1.2837309	1.3145219
theta David	-0.91866630	0.7238722	-2.3374298	0.5000972
theta Jonas	0.45847521	0.6727929	-0.8601747	1.7771251
theta Nicklas	0.45847521	0.6727929	-0.8601747	1.7771251
theta Sandra	0.01539553	0.6628318	-1.2837309	1.3145219
theta Mario	-0.43133018	0.6781861	-1.7605505	0.8978901
theta Guido	-0.43133018	0.6781861	-1.7605505	0.8978901
theta Lisa	-0.91866630	0.7238722	-2.3374298	0.5000972
theta Peter	-0.43133018	0.6781861	-1.7605505	0.8978901
theta Justus	0.01539553	0.6628318	-1.2837309	1.3145219
theta Josef	-1.50799727	0.8233509	-3.1217353	0.1057408
theta Brigitte	0.01539553	0.6628318	-1.2837309	1.3145219
theta Kevin	-0.43133018	0.6781861	-1.7605505	0.8978901
theta Marcel	-1.50799727	0.8233509	-3.1217353	0.1057408
theta Nadine	-0.43133018	0.6781861	-1.7605505	0.8978901
theta Alex	-2.37460583	1.0814193	-4.4941487	-0.2550629
theta Katharina	0.01539553	0.6628318	-1.2837309	1.3145219
theta Daniel	-1.50799727	0.8233509	-3.1217353	0.1057408
theta Jens	0.01539553	0.6628318	-1.2837309	1.3145219
theta Dieter	-0.91866630	0.7238722	-2.3374298	0.5000972
theta Sebastian	0.01539553	0.6628318	-1.2837309	1.3145219
theta Anne	-0.91866630	0.7238722	-2.3374298	0.5000972

32.2.4. Gültigkeitskontrollen

Das aufgestellte Rasch-Modell lässt sich auf verschiedene Weisen hinsichtlich seiner Gültigkeit überprüfen.

- bedingter Likelihood-Quotiententest

bedingter Likelihood-Quotiententest

Der bedingte Likelihood-Quotiententest erfolgt durch Aufruf der Funktion `LRtest`.

```
LRtest(object, splitcr = "median", se = FALSE)
```

Hierbei gelten die Parameter:

- `object` = Output von `RM()`, `PCM()` oder `RSM()`
- `splitcr` = Teilungskriterium "median", "mean" oder in Form eines Vektors
- `se` = Berechnung der Standardfehler (TRUE=ja / FALSE=nein)

Es gelten folgende Hypothesen:

- H_0 = "Das Rasch-Modell **gilt**" (keine Signifikanz)
- H_1 = "Das Rasch-Modell gilt **nicht**" (Signifikanz)

Wir rufen nun die Funktion für unseren Minigolfwettbewerb auf:

```
LRtest(result)
```

und erhalten die Ausgabe:

```
Warning message: Persons with median raw scores are assigned to the
lower raw score group!

Andersen LR-test:
LR-value: 12.58
Chi-square df: 9
p-value: 0.183
```

Hierbei ist LR-value der Chi²-Wert und df die Anzahl der Freiheitsgrade. Da der p-Wert nicht signifikant ist, wird H_0 = "Das Rasch-Modell gilt" beibehalten. Der *kritische* Chi²-Wert kann noch manuell per `qchisq()`¹⁰ errechnet werden. Hierbei müssen Testwahrscheinlichkeit und Anzahl der Freiheitsgrade übergeben werden:

```
qchisq11(0.95, df=21)
```

Dies liefert das Ergebnis

```
[1] 32.67057
```

Da der Chi²-Wert des Tests kleiner ist als der kritische Wert, bestätigt sich erneut die Annahme, dass H_0 = "Das Rasch-Modell gilt" beibehalten werden kann.

Wald-Test

Der Wald-Test erfolgt durch Aufruf der Funktion `waldtest`.

```
waldtest(object, splitcr = "median")
```

Hierbei gelten die Parameter:

- `object` = Output von `RM()`
- `splitcr` = Teilungskriterium "median", "mean" oder in Form eines Vektors

¹⁰ <http://de.wikibooks.org/wiki/GNU%20R%3A%20qchisq>

Erreicht ein Item Signifikanz, kann es nicht weiterverwendet werden. Rufen wir die Funktion auf

```
Waldtest(result)
```

erhalten wir die Ausgabe:

```
Wald test on item level (z-values):
```

	z-statistic	p-value
beta Loch1	0.127	0.899
beta Loch2	0.462	0.644
beta Loch3	0.764	0.445
beta Loch4	-1.317	0.188
beta Loch5	-1.299	0.194
beta Loch6	0.009	0.993
beta Loch7	0.721	0.471
beta Loch8	-1.661	0.097
beta Loch9	0.795	0.427
beta Loch10	2.041	0.041

Wir sehen, dass für "Loch10" ein signifikanter p-Wert errechnet wurde. Somit kann dieses Item nicht weiter verwendet werden.

Graphischer Test

Zur graphischen Überprüfung steht die Funktion `plotGOF()` zur Verfügung. Sie benötigt den Output der Funktion `LRtest()`.

```
plotGOF(LRtest(result))
```

Wir erhalten die folgende Graphik:

Graphical Model Check

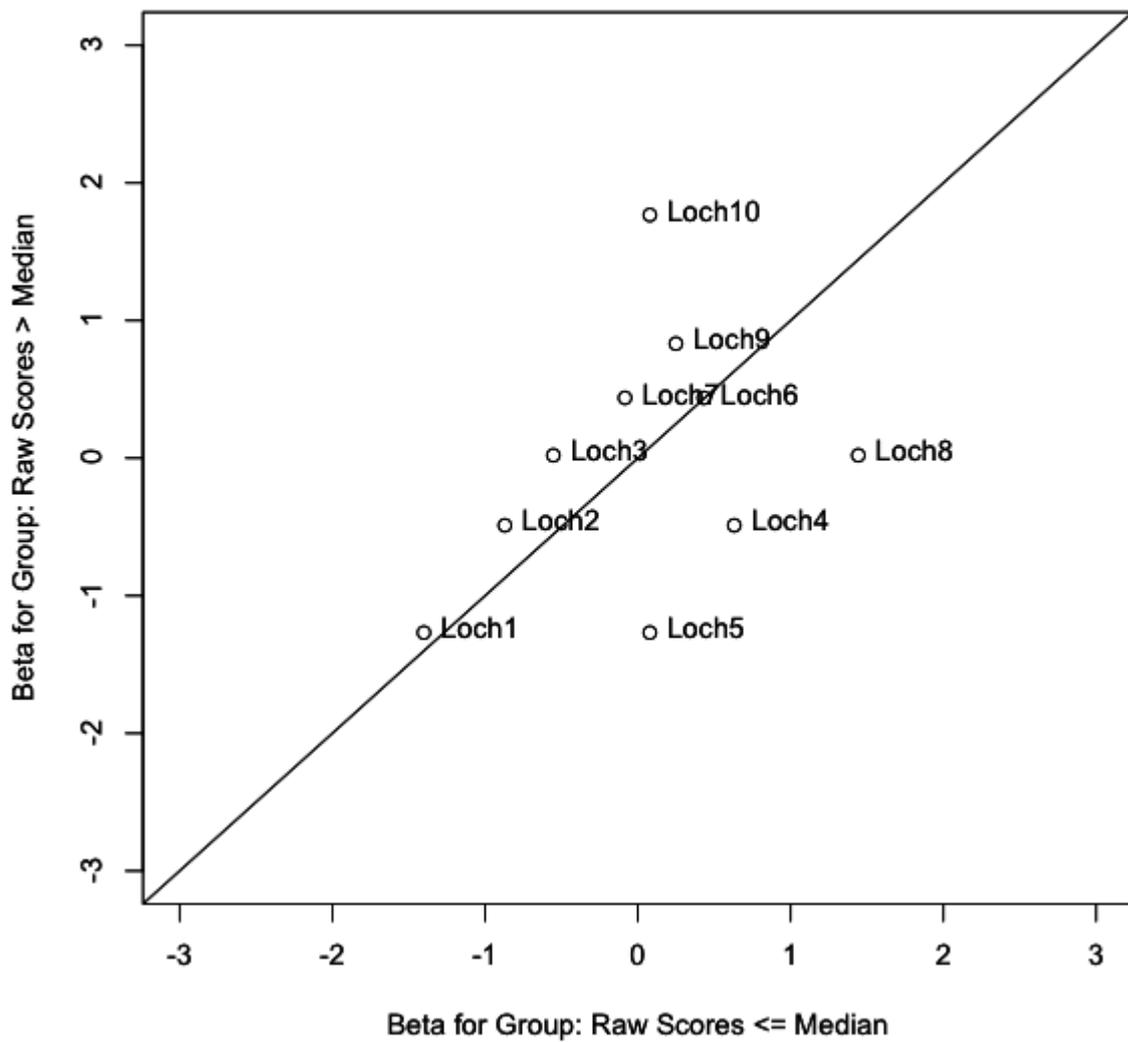


Abb. 6

Itemfit, Personfit und theoretische Wahrscheinlichkeiten

eRm stellt weitere Funktionen zur Überprüfung des Modells zur Verfügung. Mit `itemfit()` lassen sich Item-, mit `personfit()` Personenparameter überprüfen.

```
itemfit(pp)
```

liefert die Ausgabe:

```

Itemfit Statistics:
      Chisq df p-value Outfit MSQ Infit MSQ
Loch1 43.838 34 0.120    1.289  1.230
Loch2 36.199 34 0.366    1.065  1.055
Loch3 33.027 34 0.515    0.971  0.997
Loch4 29.060 34 0.708    0.855  0.901
Loch5 28.248 34 0.745    0.831  0.880
Loch6 28.045 34 0.754    0.825  0.902
Loch7 33.386 34 0.498    0.982  1.010
Loch8 28.035 34 0.754    0.825  0.860
Loch9 31.049 34 0.613    0.913  0.994
Loch10 34.331 34 0.452    1.010  1.082

```

und

```
personfit (pp)
```

liefert die Ausgabe:

```

Personfit Statistics:
      Chisq df p-value Outfit MSQ Infit MSQ
Hans    17.798 10 0.058    1.780  1.303
Karola   11.819 10 0.297    1.182  1.095
Matthias  7.756 10 0.653    0.776  0.893
Stefan   14.821 10 0.139    1.482  1.397
Sabine   6.598 10 0.763    0.660  0.761
Irma    10.073 10 0.434    1.007  1.057
Heike   10.010 10 0.440    1.001  1.017
Ralf     7.068 10 0.719    0.707  0.731
Rainer  10.336 10 0.412    1.034  1.087
Simon    9.683 10 0.469    0.968  0.948
Andreas  5.985 10 0.817    0.598  0.628
Elke     9.789 10 0.459    0.979  1.019
Gabi    12.173 10 0.274    1.217  1.138
David    7.577 10 0.670    0.758  0.782
Jonas    9.314 10 0.503    0.931  0.975
Nicklas  8.608 10 0.570    0.861  0.918
Sandra  13.950 10 0.175    1.395  1.299
Mario    8.679 10 0.563    0.868  0.897
Guido    7.427 10 0.685    0.743  0.756
Lisa     8.748 10 0.556    0.875  0.981
Peter    8.679 10 0.563    0.868  0.897
Justus   7.841 10 0.644    0.784  0.810
Josef    5.786 10 0.833    0.579  0.745
Brigitte 7.546 10 0.673    0.755  0.783
Kevin   13.795 10 0.183    1.380  1.296
Marcel   4.687 10 0.911    0.469  0.641
Nadine   11.708 10 0.305    1.171  1.183
Alex     3.566 10 0.965    0.357  0.715
Katharina 14.189 10 0.165    1.419  1.349
Daniel   4.687 10 0.911    0.469  0.641
Jens     9.023 10 0.530    0.902  0.940
Dieter  12.208 10 0.271    1.221  1.070
Sebastian 10.010 10 0.440    1.001  1.017
Anne    13.278 10 0.209    1.328  1.333

```

Die Funktion `pmat ()` liefert die Item-Personen-Matrix zurück, welche die theoretisch zu erwartenden Wahrscheinlichkeiten enthält.

pmat (pp)					
	Loch1	Loch2	Loch3	Loch4	Loch5
Loch6	Loch7	Loch8	Loch9	Loch10	
Hans	0.9087360	0.8474865	0.7922326	0.64956115	0.75005628
0.6203015	0.70281159	0.51949106	0.6203015	0.58902851	
Karola	0.9087360	0.8474865	0.7922326	0.64956115	0.75005628
0.6203015	0.70281159	0.51949106	0.6203015	0.58902851	
Matthias	0.9087360	0.8474865	0.7922326	0.64956115	0.75005628
0.6203015	0.70281159	0.51949106	0.6203015	0.58902851	
Stefan	0.7988114	0.6890338	0.6032488	0.42499412	0.54475410
0.3944637	0.48533092	0.30123749	0.3944637	0.36367126	
Sabine	0.9087360	0.8474865	0.7922326	0.64956115	0.75005628
0.6203015	0.70281159	0.51949106	0.6203015	0.58902851	
Irma	0.8608014	0.7753352	0.7030992	0.53513568	0.65080429
0.5036230	0.59493064	0.40171309	0.5036230	0.47093647	
Heike	0.7988114	0.6890338	0.6032488	0.42499412	0.54475410
0.3944637	0.48533092	0.30123749	0.3944637	0.36367126	
Ralf	0.7988114	0.6890338	0.6032488	0.42499412	0.54475410
0.3944637	0.48533092	0.30123749	0.3944637	0.36367126	
Rainer	0.8608014	0.7753352	0.7030992	0.53513568	0.65080429
0.5036230	0.59493064	0.40171309	0.5036230	0.47093647	
Simon	0.8608014	0.7753352	0.7030992	0.53513568	0.65080429
0.5036230	0.59493064	0.40171309	0.5036230	0.47093647	
Andreas	0.7175126	0.5863461	0.4930741	0.32103255	0.43358751
0.2941499	0.37626806	0.21616820	0.2941499	0.26772600	
Elke	0.7988114	0.6890338	0.6032488	0.42499412	0.54475410
0.3944637	0.48533092	0.30123749	0.3944637	0.36367126	
Gabi	0.7988114	0.6890338	0.6032488	0.42499412	0.54475410
0.3944637	0.48533092	0.30123749	0.3944637	0.36367126	
David	0.6094070	0.4654413	0.3740124	0.22506903	0.31982741
0.2038101	0.27036825	0.14486293	0.2038101	0.18339296	
Jonas	0.8608014	0.7753352	0.7030992	0.53513568	0.65080429
0.5036230	0.59493064	0.40171309	0.5036230	0.47093647	
Nicklas	0.8608014	0.7753352	0.7030992	0.53513568	0.65080429
0.5036230	0.59493064	0.40171309	0.5036230	0.47093647	
Sandra	0.7988114	0.6890338	0.6032488	0.42499412	0.54475410
0.3944637	0.48533092	0.30123749	0.3944637	0.36367126	
Mario	0.7175126	0.5863461	0.4930741	0.32103255	0.43358751
0.2941499	0.37626806	0.21616820	0.2941499	0.26772600	
Guido	0.7175126	0.5863461	0.4930741	0.32103255	0.43358751
0.2941499	0.37626806	0.21616820	0.2941499	0.26772600	
Lisa	0.6094070	0.4654413	0.3740124	0.22506903	0.31982741
0.2038101	0.27036825	0.14486293	0.2038101	0.18339296	
Peter	0.7175126	0.5863461	0.4930741	0.32103255	0.43358751
0.2941499	0.37626806	0.21616820	0.2941499	0.26772600	
Justus	0.7988114	0.6890338	0.6032488	0.42499412	0.54475410
0.3944637	0.48533092	0.30123749	0.3944637	0.36367126	
Josef	0.4639351	0.3256806	0.2489215	0.13875159	0.20687009
0.1243376	0.17050026	0.08589618	0.1243376	0.11077415	
Brigitte	0.7988114	0.6890338	0.6032488	0.42499412	0.54475410
0.3944637	0.48533092	0.30123749	0.3944637	0.36367126	
Kevin	0.7175126	0.5863461	0.4930741	0.32103255	0.43358751
0.2941499	0.37626806	0.21616820	0.2941499	0.26772600	
Marcel	0.4639351	0.3256806	0.2489215	0.13875159	0.20687009
0.1243376	0.17050026	0.08589618	0.1243376	0.11077415	
Nadine	0.7175126	0.5863461	0.4930741	0.32103255	0.43358751
0.2941499	0.37626806	0.21616820	0.2941499	0.26772600	
Alex	0.2667609	0.1687664	0.1222835	0.06342887	0.09881113
0.0563279	0.07953407	0.03800055	0.0563279	0.04976174	
Katharina	0.7988114	0.6890338	0.6032488	0.42499412	0.54475410
0.3944637	0.48533092	0.30123749	0.3944637	0.36367126	
Daniel	0.4639351	0.3256806	0.2489215	0.13875159	0.20687009
0.1243376	0.17050026	0.08589618	0.1243376	0.11077415	

Jens	0.7988114	0.6890338	0.6032488	0.42499412	0.54475410
0.3944637	0.48533092	0.30123749	0.3944637	0.36367126	
Dieter	0.6094070	0.4654413	0.3740124	0.22506903	0.31982741
0.2038101	0.27036825	0.14486293	0.2038101	0.18339296	
Sebastian	0.7988114	0.6890338	0.6032488	0.42499412	0.54475410
0.3944637	0.48533092	0.30123749	0.3944637	0.36367126	
Anne	0.6094070	0.4654413	0.3740124	0.22506903	0.31982741
0.2038101	0.27036825	0.14486293	0.2038101	0.18339296	

Die Funktion `residuals()`¹² gibt schließlich die Residuen aus.

	Loch1	Loch2	Loch3	Loch4	Loch5
Loch6	Loch7	Loch8	Loch9	Loch10	
Hans	-3.1555061	0.4242166	0.5121089	-1.3614569	0.5772636
0.7823805	-1.5378130	0.9617489	0.7823805	0.8352908	
Karola	0.3169064	-2.3572861	0.5121089	0.7345073	0.5772636
-1.2781505	0.6502741	0.9617489	0.7823805	-1.1971879	
Matthias	0.3169064	0.4242166	0.5121089	-1.3614569	0.5772636
0.7823805	0.6502741	0.9617489	-1.2781505	-1.1971879	
Stefan	-1.9926014	-1.4885510	0.8109814	1.1631740	-1.0938990
1.2389866	1.0297814	-0.6565831	-0.8071113	1.3227759	
Sabine	0.3169064	0.4242166	0.5121089	0.7345073	0.5772636
0.7823805	0.6502741	-1.0397724	-1.2781505	-1.1971879	
Irma	0.4021296	0.5382980	-1.5388725	0.9320327	0.7325027
0.9927801	-1.2119047	1.2203847	-1.0072724	-0.9434682	
Heike	0.5018565	-1.4885510	0.8109814	-0.8597166	0.9141612
1.2389866	-0.9710798	-0.6565831	-0.8071113	1.3227759	
Ralf	0.5018565	0.6717943	0.8109814	-0.8597166	0.9141612
1.2389866	-0.9710798	-0.6565831	-0.8071113	-0.7559859	
Rainer	0.4021296	0.5382980	-1.5388725	0.9320327	0.7325027
-1.0072724	-1.2119047	1.2203847	-1.0072724	1.0599192	
Simon	0.4021296	-1.8577071	0.6498264	0.9320327	0.7325027
-1.0072724	0.8251474	-0.8194137	0.9927801	-0.9434682	
Andreas	0.6274582	0.8399270	1.0139491	-0.6876225	1.1429521
-0.6455474	-0.7766935	-0.5251513	-0.6455474	-0.6046561	
Elke	0.5018565	0.6717943	-1.2330739	1.1631740	-1.0938990
1.2389866	-0.9710798	-0.6565831	1.2389866	-0.7559859	
Gabi	-1.9926014	0.6717943	0.8109814	1.1631740	-1.0938990
-0.8071113	1.0297814	-0.6565831	-0.8071113	1.3227759	
David	0.8005870	1.0716803	-0.7729656	1.8555542	-0.6857223
-0.5059463	-0.6087318	-0.4115862	-0.5059463	-0.4738979	
Jonas	0.4021296	0.5382980	-1.5388725	0.9320327	0.7325027
-1.0072724	0.8251474	1.2203847	-1.0072724	-0.9434682	
Nicklas	0.4021296	0.5382980	0.6498264	0.9320327	-1.3651826
0.9927801	-1.2119047	-0.8194137	0.9927801	-0.9434682	
Sandra	-1.9926014	0.6717943	-1.2330739	-0.8597166	0.9141612
-0.8071113	1.0297814	1.5230364	-0.8071113	1.3227759	
Mario	0.6274582	-1.1905796	1.0139491	-0.6876225	1.1429521
1.5490729	-0.7766935	-0.5251513	-0.6455474	-0.6046561	
Guido	0.6274582	0.8399270	1.0139491	-0.6876225	-0.8749273
-0.6455474	-0.7766935	-0.5251513	1.5490729	-0.6046561	
Lisa	-1.2490835	1.0716803	1.2937187	-0.5389226	-0.6857223
-0.5059463	1.6427596	-0.4115862	-0.5059463	-0.4738979	
Peter	0.6274582	-1.1905796	1.0139491	-0.6876225	1.1429521
-0.6455474	-0.7766935	-0.5251513	1.5490729	-0.6046561	
Justus	0.5018565	0.6717943	0.8109814	-0.8597166	-1.0938990
-0.8071113	1.0297814	-0.6565831	-0.8071113	1.3227759	
Josef	1.0749298	-0.6949654	1.7370465	-0.4013791	-0.5107127
-0.3768191	-0.4533717	-0.3065414	-0.3768191	-0.3529500	
Brigitte	0.5018565	0.6717943	0.8109814	-0.8597166	-1.0938990

¹² <http://de.wikibooks.org/wiki/GNU%20R%3A%20residuals>

```

-0.8071113  1.0297814 -0.6565831  1.2389866 -0.7559859
Kevin      -1.5937316  0.8399270 -0.9862428 -0.6876225  1.1429521
-0.6455474 -0.7766935  1.9042131 -0.6455474  1.6538325
Marcel     1.0749298  1.4389206 -0.5756898 -0.4013791 -0.5107127
-0.3768191 -0.4533717 -0.3065414 -0.3768191 -0.3529500
Nadine     0.6274582 -1.1905796 -0.9862428  1.4542864 -0.8749273
1.5490729 -0.7766935 -0.5251513  1.5490729 -0.6046561
Alex       1.6579131 -0.4505900 -0.3732561 -0.2602394 -0.3311273
-0.2443156 -0.2939495 -0.1987502 -0.2443156 -0.2288398
Katharina -1.9926014  0.6717943 -1.2330739 -0.8597166 -1.0938990
1.2389866  1.0297814 -0.6565831  1.2389866  1.3227759
Daniel     1.0749298  1.4389206 -0.5756898 -0.4013791 -0.5107127
-0.3768191 -0.4533717 -0.3065414 -0.3768191 -0.3529500
Jens       0.5018565  0.6717943 -1.2330739  1.1631740 -1.0938990
-0.8071113  1.0297814 -0.6565831  1.2389866 -0.7559859
Dieter     0.8005870 -0.9331141 -0.7729656 -0.5389226 -0.6857223
-0.5059463  1.6427596  2.4296249 -0.5059463 -0.4738979
Sebastian  0.5018565 -1.4885510  0.8109814 -0.8597166  0.9141612
-0.8071113 -0.9710798 -0.6565831  1.2389866  1.3227759
Anne       -1.2490835 -0.9331141 -0.7729656 -0.5389226  1.4583163
-0.5059463  1.6427596 -0.4115862 -0.5059463  2.1101592

```

Informationsindices

Die Informationsindices "Akaike Information Criterion" (AIC), "Bayes Information Criterion" (BIC) und "consistent Akaike Information Criterion" (CAIC) überprüfen, welches Modell am besten auf die Daten passt. Hierbei gilt: je kleiner der Informationsindex, desto besser passt das Modell. Die dazugehörige Funktion lautet `IC()` und benötigt den Output der `person.parameter()`-Funktion. In unserem Beispiel rufen wir sie also auf per:

```
IC(person.parameter(RM(minigolf)))
```

oder

```
IC(person.parameter(result))
```

oder

```
IC(pp)
```

Die Funktion liefert die folgende Ausgabe:

```

Information Criteria:
      value npar    AIC    BIC    cAIC
joint log-lik    -199.8864    16 431.7728 456.1946 472.1946
marginal log-lik  -216.1203     9 450.2406 463.9779 472.9779
conditional log-lik -155.7848     9 329.5697 343.3069 352.3069

```

32.3. Das linear-logistische Test-Modell

Der Befehl zur Analyse nach dem linear-logistischen Test-Modell lautet `LLTM()`

Als Untersuchungsobjekt für `LLTM()` wird eine Matrix benötigt, welche dichotome Daten enthält (0 und 1). Die Reihen der Matrix entsprechen den Untersuchungsfällen (n) und die Spalten entsprechen den Ergebnissen der Items.

32.3.1. LLTM()

An den `LLTM()`-Befehl können folgende Parameter übergeben werden:

```
LLTM(X, W, mpoints = 1, groupvec = 1, se = TRUE, sum0 = TRUE,
etaStart)
```

- `X` = die zu analysierende Datenmatrix
- `W` = optionale Designmatrix zur Normierung auf ein bestimmtes Items
- `mpoints` = Anzahl der Messpunkte
- `groupvec` = Vector der Länge N, welche die Gruppenzugehörigkeiten enthält. Wird `groupvec=1` gesetzt, werden keine Gruppen verwendet
- `se` = Berechnung der Standardfehler (TRUE=ja / FALSE=nein)
- `sum0` = TRUE=Normierung der Parameter auf sum-0 / FALSE=Begrenzung des ersten Parameters auf 0
- `etaStart` = optionale Übergabe von vermuteten Parameterschätzwerten

32.4. Das Ratingskalen-Modell

Der Befehl zur Analyse nach dem Ratingskalen-Modell lautet `RSM()`

Als Untersuchungsobjekt für `RSM()` wird eine Matrix oder Datenframe benötigt. Die Reihen der Matrix entsprechen den Untersuchungsfällen (n) und die Spalten entsprechen den Ergebnissen der Items.

32.4.1. RSM()

Die folgenden Parameter können an `RSM()` übergeben werden:

```
RSM(X, W, se = TRUE, sum0 = TRUE, etaStart)
```

- `X` = die zu analysierende Datenmatrix
- `W` = optionale Designmatrix zur Normierung auf ein bestimmtes Items
- `se` = Berechnung der Standardfehler (TRUE=ja / FALSE=nein)
- `sum0` = TRUE=Normierung der Parameter auf sum-0 / FALSE=Begrenzung des ersten Parameters auf 0

- etaStart = optionale Übergabe von vermuteten Parameterschätzwerten

32.4.2. Beispieldatensatz

Als Beispieldatensatz nehmen wir die Fragebogenerhebung aus Beispiel 9¹³. Dieser Fragebogen besteht aus 27 Fragen, welche die Antworten "nie / selten / manchmal / oft / immer" zulassen. Die Antworten werden wie folgt codiert:

- 1 = nie
- 2 = selten
- 3 = manchmal
- 4 = oft
- 5 = immer

Der Fragebogen wurde von 122 Personen ausgefüllt. Der Befehl zur Erzeugung dieses Datensatzes lautet:

```
bsp9 <- structure(list(Frage1 = c(4, 4, 5, 5, 5, 1, 3, 3, 3, 3, 5, 5,
  5, 3, 3, 4, 5, 4, 3, 2, 2, 2, 3, 4, 3, 4, 4, 2,
  5, 5, 5, 5, 5, 5, 1, 3, 4, 3, 3, 4, 3, 3, 3, 3, 5, 3, 3, 3, 2, 5, 4,
  5, 2, 4, 3, 2, 5, 5, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
  3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1, 1,
  1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3,
  3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5), Frage2 = c(3, 5, 4,
  5, 4, 5, 3, 3, 5, 3, 5, 4, 4, 4, 2, 5, 5, 3, 2,
  2, 2, 2, 3, 2, 3, 5, 4, 4, 3, 3, 5, 3, 5, 2, 3, 3, 5, 4, 3, 4, 1, 4,
  4, 4, 5, 2, 2, 3, 2, 3, 5, 5, 3, 5, 3, 3, 5, 4,
  3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5,
  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5,
  5, 5), Frage3 = c(4, 4, 4, 4, 3, 3, 2, 2, 4, 2, 5,
  3, 1, 2, 3, 4, 4, 3, 3, 3, 4, 5, 4, 5, 4, 4, 3, 3, 5, 5, 2, 4, 4,
  3, 2, 3, 4, 3, 3, 3, 1, 3, 3, 2, 4, 1, 1, 3, 3, 3, 5,
  5, 4, 2, 5, 4, 4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4,
  4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5,
  5, 5, 5, 5, 5, 5, 5), Frage4 = c(4, 3, 1, 3, 3,
  4, 4, 3, 3, 3, 5, 5, 1, 3, 2, 2, 4, 4, 2, 2, 4, 5, 4, 5, 3, 4, 5, 5,
  5, 3, 5, 2, 5, 3, 2, 3, 3, 4, 3, 4, 1, 2, 3, 4, 5,
  2, 3, 2, 3, 2, 4, 5, 4, 1, 4, 2, 4, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3,
  3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 1, 1, 1, 1,
  1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3,
  3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5),
  Frage5 = c(5, 1, 2, 5, 2, 4, 4, 1, 1, 4, 5, 4, 3, 5, 3, 5, 1, 1, 4,
  5, 5, 4, 2, 4, 3, 1, 2, 1, 3, 4, 2, 1, 1, 4, 1, 1,
  2, 4, 2, 3, 3, 4, 4, 4, 4, 5, 2, 2, 2, 3, 3, 4, 2, 3, 3, 4, 3, 5, 2, 3,
  3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4,
  4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2,
  2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5,
  5, 5, 5, 5, 5, 5, 5), Frage6 = c(5, 3, 1, 4, 5, 5, 4, 3, 3, 4, 5, 5,
  4, 4, 3, 2, 4, 5, 3, 3, 3, 3, 3, 4, 3, 4, 5, 4, 2,
  5, 5, 4, 5, 4, 2, 2, 5, 3, 3, 3, 1, 4, 4, 4, 5, 3, 2, 3, 1, 2, 5, 4,
  3, 5, 4, 3, 4, 5, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
  4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3,
  3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5), Frage7 = c(3, 3, 2, 4, 4, 3,
  4, 3, 5, 3, 4, 5, 3, 4, 2, 3, 4, 3, 4, 3, 2, 2, 2,
  1, 4, 4, 4, 4, 3, 3, 5, 5, 4, 4, 4, 2, 5, 3, 3, 4, 2, 5, 3, 3, 5, 4,
```

¹³ Kapitel 48.9 auf Seite 181

3, 5, 3, 3, 4, 5, 4, 3, 5, 4, 4, 5, 3, 3, 3, 3, 3,
 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3,
 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5), Frage8 =
 c(4, 5, 5, 5, 4, 5, 4, 2, 5, 4, 5, 5, 4, 4, 4, 4, 5,
 3, 3, 1, 3, 4, 4, 3, 3, 5, 5, 5, 5, 4, 5, 5, 5, 3, 3, 4, 5, 3, 3, 4,
 2, 5, 3, 4, 5, 3, 3, 4, 3, 4, 5, 4, 4, 4, 4, 5, 4, 5,
 4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2,
 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5, 5,
 5, 5), Frage9 = c(5, 5, 5, 5, 5, 5, 5, 2, 5, 3, 5, 4,
 4, 4, 2, 4, 5, 4, 2, 2, 3, 4, 4, 4, 3, 5, 5, 5, 5, 3, 4, 4, 5, 4, 5,
 4, 5, 4, 4, 4, 2, 5, 4, 4, 5, 2, 3, 2, 4, 2, 4, 4, 4, 4,
 5, 2, 3, 5, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4,
 4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2,
 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5,
 5, 5, 5, 5, 5), Frage10 = c(2, 1, 5, 4, 3, 4, 2, 4,
 5, 1, 3, 2, 1, 1, 1, 5, 4, 2, 2, 5, 3, 5, 2, 2, 3, 1, 1, 2, 5, 5, 2,
 3, 3, 4, 2, 5, 4, 5, 5, 3, 2, 5, 4, 3, 4, 2, 4, 1, 4,
 3, 5, 3, 3, 4, 4, 5, 3, 5, 4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4,
 4, 4, 4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5,
 5, 5, 5, 5, 5, 5, 5, 5), Frage11 = c(3, 1, 5, 4,
 4, 3, 3, 2, 1, 3, 4, 4, 1, 4, 4, 4, 4, 1, 3, 5, 5, 5, 3, 5, 5, 2, 5,
 2, 1, 2, 5, 3, 5, 5, 1, 1, 2, 5, 4, 4, 3, 3, 4, 4, 5,
 3, 3, 3, 3, 3, 5, 5, 5, 5, 3, 3, 5, 5, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3,
 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5), Frage12 = c(1,
 5, 3, 2, 5, 3, 2, 1, 1, 2, 1, 4, 3, 1, 3, 1, 1, 3, 3, 1, 3, 1, 2, 2,
 4, 4, 5, 5, 3, 1, 3, 1, 4, 2, 1, 5, 4, 3, 3, 2, 4,
 3, 2, 5, 5, 1, 1, 3, 3, 4, 5, 4, 4, 4, 2, 5, 1, 3, 3, 3, 3, 3, 3, 3,
 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3,
 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5), Frage13 = c(2, 4, 2, 3, 3, 3, 3, 3, 2, 2, 4, 4, 1, 2, 1, 2, 4, 4, 2,
 3, 2, 3, 3, 1, 2, 3, 5, 4, 4, 2, 1, 3, 5, 3, 4, 1, 5,
 3, 3, 3, 1, 2, 3, 1, 4, 5, 3, 1, 4, 4, 2, 4, 2, 2, 5, 2, 3, 5, 3, 3,
 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
 4, 4, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2,
 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5,
 5, 5, 5), Frage14 = c(4, 3, 2, 4, 4, 5, 4, 3, 2, 4, 5, 4, 5, 3, 1,
 4, 5, 5, 1, 3, 2, 3, 3, 5, 3, 3, 5, 4, 4, 3, 5, 3, 5, 3,
 4, 3, 4, 4, 4, 2, 1, 4, 3, 3, 4, 2, 2, 5, 4, 4, 1, 2, 2, 3, 4, 3, 3,
 5, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4,
 4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2,
 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5,
 5, 5, 5, 5, 5, 5), Frage15 = c(4, 3, 2, 5, 4, 5, 3, 2, 1, 4, 5, 4,
 5, 3, 3, 3, 5, 5, 2, 3, 4, 4, 1, 2, 3, 4, 5, 5, 3, 5, 1,
 2, 3, 3, 1, 4, 5, 3, 5, 2, 2, 5, 4, 4, 4, 5, 5, 4, 3, 3, 1, 5, 1, 1,
 5, 2, 3, 5, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4,
 4, 4, 4, 4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2,
 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5,
 5, 5, 5, 5, 5, 5), Frage16 = c(3, 5, 1, 5, 4, 1, 3, 2, 3,
 2, 5, 5, 5, 4, 4, 3, 4, 4, 4, 4, 4, 3, 4, 4, 4, 5, 3,
 1, 2, 5, 4, 3, 3, 5, 4, 4, 4, 5, 4, 4, 3, 4, 3, 4, 5, 5, 5, 4, 4, 5,
 5, 1, 5, 5, 4, 5, 5, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3,
 3, 3, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5), Frage17 = c(4, 4, 4, 2, 3, 2,
 2, 4, 5, 3, 1, 4, 5, 3, 2, 4, 4, 4, 1, 2, 4, 5, 3, 4, 1,
 4, 5, 3, 1, 3, 3, 4, 1, 5, 2, 4, 3, 5, 5, 4, 1, 5, 3, 4, 4, 5, 4,
 5, 4, 4, 5, 5, 3, 4, 5, 4, 5, 5, 3, 3, 3, 3, 3, 3, 3, 3,
 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3,
 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5), Frage18 = c(4, 2, 1,
 5, 3, 5, 2, 4, 3, 4, 2, 3, 1, 3, 2, 2, 2, 1, 1, 4, 1, 5,
 2, 5, 2, 1, 2, 5, 3, 2, 5, 4, 4, 1, 3, 1, 2, 5, 4, 5, 3, 5, 3, 5,


```
"63", "64", "65", "66", "67", "68", "69", "70", "71", "72",
"73", "74", "75", "76", "77", "78", "79", "80", "81", "82", "158",
"159", "160", "161", "162", "163", "164", "165", "166",
"167", "168", "169", "170", "171", "172", "173", "174", "175",
"176", "177", "178", "179", "180", "181", "182", "183", "184",
"185", "186", "187", "188", "189", "190", "191", "192", "193",
"194", "195", "196", "197", "198", "199", "200", "201",
"202", "203", "204", "205", "206", "207", "208", "209", "210",
"211", "212", "213", "214", "215", "216", "217", "218", "219",
"220", "221"), class = "omit"))
```

Der Datensatz ist nun in der Variablen `bsp9` gespeichert. Die Daten können eingesehen werden, indem die Variable aufgerufen wird:

```
bsp9
```

32.4.3. Parameterschätzung

Die Analyse erfolgt durch Aufruf der `RSM()`-Funktion

```
RSM(bsp9)
```

Da der Output noch für weitere Analysen benötigt wird, speichern wir ihn in die Variable `rsm.result`:

```
rsm.result <- RSM(bsp9)
```

Somit wird durch Aufruf der Variablen der Output wieder ausgegeben:

```
rsm.result
```

```
Results of RSM estimation:

Call:  RSM(X = bsp9)

Conditional log-likelihood: -3045.451
Number of iterations: 20
Number of parameters: 29

Basic Parameters eta:
      eta 1      eta 2      eta 3      eta 4      eta 5
eta 6      eta 7      eta 8
Estimate 0.03942847 -0.1626761 -0.1761274 -0.4188407 0.0664840
0.03942847 0.3960766 0.3263914
Std.Err  0.11411907 0.1139070 0.1139206 0.1147358 0.1142076
0.11411903 0.1164774 0.1158098

      eta 9      eta 10      eta 11      eta 12      eta 13
eta 14      eta 15      eta 16
Estimate -0.2434006 -0.01457923 -0.5822437 -0.5003087 -0.0819464
-0.08194639 0.2436041 0.03942844
Std.Err  0.1140388 0.11398516 0.1158746 0.1152457 0.1138975
```

```

0.11389774 0.1151493 0.11411904

          eta 17      eta 18      eta 19      eta 20      eta 21      eta
22      eta 23      eta 24      eta 25
Estimate -0.3917694 -0.1357728 0.08002696 0.3263914 0.3681175
0.2711103 0.01240908 0.0664840 0.3541815
Std.Err  0.1145923 0.1138901 0.11425720 0.1158098 0.1161972
0.1153532 0.11404490 0.1142077 0.1160637

          eta 26      eta 27      eta 28      eta 29
Estimate 0.08002695 -1.2993059 -4.0572947 -7.5319724
Std.Err  0.11425714 0.1444744 0.2686848 0.4036785

```

Wird die Variable `rsm.result` an die `summary()`¹⁴-Funktion übergeben, werden neben den eta-Werten auch die beta-Werte angezeigt:

```
summary(RSM(bsp9))
```

oder in unserem Fall

```
summary(rsm.result)
```

liefern die (hier in Wikibooks aus Platzgründen gekürzte) Ausgabe:

```

Results of RSM estimation:

Call:  RSM(X = bsp9)

Conditional log-likelihood: -3045.451
Number of iterations: 20
Number of parameters: 29

Basic Parameters (eta) with 0.95 CI:
      Estimate Std. Error lower CI upper CI
eta 1      0.039      0.114   -0.184    0.263
eta 2     -0.163      0.114   -0.386    0.061
eta 3     -0.176      0.114   -0.399    0.047
eta 4     -0.419      0.115   -0.644   -0.194

( . . . )

eta 27    -1.299      0.144   -1.582   -1.016
eta 28    -4.057      0.269   -4.584   -3.531
eta 29    -7.532      0.404   -8.323   -6.741

Item Easiness Parameters (beta) with 0.95 CI:
      Estimate Std. Error lower CI upper CI
beta Frage1.c1  0.080      0.114   -0.144    0.304
beta Frage1.c2 -1.139      0.270   -1.667   -0.611
beta Frage1.c3 -3.817      0.433   -4.666   -2.968
beta Frage1.c4 -7.212      0.606   -8.399   -6.024
beta Frage2.c1  0.039      0.114   -0.184    0.263
beta Frage2.c2 -1.220      0.270   -1.749   -0.692

```

¹⁴ <http://de.wikibooks.org/wiki/GNU%20R%3A%20summary>

```

beta Frage2.c3 -3.939      0.434 -4.789 -3.089
beta Frage2.c4 -7.374      0.607 -8.564 -6.185
beta Frage3.c1 -0.163      0.114 -0.386  0.061
beta Frage3.c2 -1.625      0.271 -2.155 -1.094
beta Frage3.c3 -4.545      0.438 -5.403 -3.687
beta Frage3.c4 -8.183      0.615 -9.388 -6.978

```

(. . .)

```

beta Frage26.c1  0.354      0.116  0.127  0.582
beta Frage26.c2 -0.591      0.271 -1.122 -0.060
beta Frage26.c3 -2.995      0.433 -3.843 -2.147
beta Frage26.c4 -6.115      0.601 -7.294 -4.937
beta Frage27.c1  0.080      0.114 -0.144  0.304
beta Frage27.c2 -1.139      0.270 -1.667 -0.611
beta Frage27.c3 -3.817      0.433 -4.666 -2.968
beta Frage27.c4 -7.212      0.606 -8.399 -6.024

```

Schwellenwerte

Die Schwellenwerte (Thresholds) der einzelnen Items werden mit der Funktion `thresholds()` angezeigt.

```
thresholds(rsm.result)
```

liefert die Ausgabe:

```

Design Matrix Block 1:
      Location Threshold 1 Threshold 2 Threshold 3 Threshold 4
Frage1  1.80297   -0.08002    1.21928    2.67797    3.39466
Frage2  1.84356   -0.03943    1.25988    2.71856    3.43525
Frage3  2.04567    0.16268    1.46198    2.92066    3.63735
Frage4  2.05912    0.17613    1.47543    2.93412    3.65081
Frage5  2.30183    0.41884    1.71815    3.17683    3.89352
Frage6  1.81651   -0.06648    1.23282    2.69150    3.40819
Frage7  1.84356   -0.03943    1.25988    2.71856    3.43525
Frage8  1.48692   -0.39608    0.90323    2.36191    3.07860
Frage9  1.55660   -0.32639    0.97291    2.43160    3.14829
Frage10 2.12639    0.24340    1.54271    3.00139    3.71808
Frage11 1.89757    0.01458    1.31389    2.77257    3.48926
Frage12 2.46524    0.58224    1.88155    3.34023    4.05692
Frage13 2.38330    0.50031    1.79961    3.25830    3.97499
Frage14 1.96494    0.08195    1.38125    2.83994    3.55662
Frage15 1.96494    0.08195    1.38125    2.83994    3.55662
Frage16 1.63939   -0.24360    1.05570    2.51438    3.23107
Frage17 1.84356   -0.03943    1.25988    2.71856    3.43525
Frage18 2.27476    0.39177    1.69108    3.14976    3.86645
Frage19 2.01877    0.13577    1.43508    2.89376    3.61045
Frage20 1.80297   -0.08003    1.21928    2.67796    3.39465
Frage21 1.55660   -0.32639    0.97291    2.43160    3.14829
Frage22 1.51488   -0.36812    0.93119    2.38987    3.10656
Frage23 1.61188   -0.27111    1.02820    2.48688    3.20357
Frage24 1.87058   -0.01241    1.28690    2.74558    3.46227
Frage25 1.81651   -0.06648    1.23282    2.69150    3.40819
Frage26 1.52881   -0.35418    0.94512    2.40381    3.12050
Frage27 1.80297   -0.08003    1.21928    2.67796    3.39465

```

32.5. Das partial-credit Modell

Der Befehl zur Analyse nach dem partial-credit Modell lautet `PCM()`

Als Untersuchungsobjekt für `PCM()` wird eine Matrix oder Datenframe benötigt. Die Reihen der Matrix entsprechen den Untersuchungsfällen (n) und die Spalten entsprechen den Ergebnissen der Items.

32.5.1. PCM()

Die folgenden Parameter können an `PCM()` übergeben werden:

```
PCM(X, W, se = TRUE, sum0 = TRUE, etaStart)
```

- `X` = die zu analysierende Datenmatrix
- `W` = optionale Designmatrix zur Normierung auf ein bestimmtes Items
- `se` = Berechnung der Standardfehler (TRUE=ja / FALSE=nein)
- `sum0 = TRUE`=Normierung der Parameter auf sum-0 / `FALSE`=Begrenzung des ersten Parameters auf 0
- `etaStart` = optionale Übergabe von vermuteten Parameterschätzwerten

32.6. Das lineare Ratingskalen-Modell

Der Befehl zur Analyse nach dem linearen Ratingskalen-Modell lautet `LRSM()`

Als Untersuchungsobjekt für `LRSM()` wird eine Matrix oder Datenframe benötigt. Die Reihen der Matrix entsprechen den Untersuchungsfällen (n) und die Spalten entsprechen den Ergebnissen der Items.

32.6.1. LRSM()

Die folgenden Parameter können an `LRSM()` übergeben werden:

```
LRSM(X, W, mpoints = 1, groupvec = 1, se = TRUE, sum0 = TRUE,  
etaStart)
```

- `X` = die zu analysierende Datenmatrix
- `W` = optionale Designmatrix zur Normierung auf ein bestimmtes Items
- `mpoints` = Anzahl der Messpunkte
- `groupvec` = Vector der Länge N, welche die Gruppenzugehörigkeiten enthält. Wird `groupvec=1` gesetzt, werden keine Gruppen verwendet
- `se` = Berechnung der Standardfehler (TRUE=ja / FALSE=nein)
- `sum0 = TRUE`=Normierung der Parameter auf sum-0 / `FALSE`=Begrenzung des ersten Parameters auf 0

- `etaStart` = optionale Übergabe von vermuteten Parameterschätzwerten

32.7. Das lineare partial-credit Modell

Der Befehl zur Analyse nach dem linearen partial-credit Modell lautet `LPCM()`

Als Untersuchungsobjekt für `LPCM()` wird eine Matrix oder Datenframe benötigt. Die Reihen der Matrix entsprechen den Untersuchungsfällen (n) und die Spalten entsprechen den Ergebnissen der Items.

32.7.1. LPCM()

Die folgenden Parameter können an `LPCM()` übergeben werden:

```
LPCM(X, W, mpoints = 1, groupvec = 1, se = TRUE, sum0 = TRUE,
etaStart)
```

- `X` = die zu analysierende Datenmatrix
- `W` = optionale Designmatrix zur Normierung auf ein bestimmtes Items
- `mpoints` = Anzahl der Messpunkte
- `groupvec` = Vector der Länge N, welche die Gruppenzugehörigkeiten enthält. Wird `groupvec=1` gesetzt, werden keine Gruppen verwendet
- `se` = Berechnung der Standardfehler (TRUE=ja / FALSE=nein)
- `sum0` = TRUE=Normierung der Parameter auf sum-0 / FALSE=Begrenzung des ersten Parameters auf 0
- `etaStart` = optionale Übergabe von vermuteten Parameterschätzwerten

32.8. unterstützte Funktionen im eRm-Package

Das eRm-Package enthält/unterstützt weitere Funktionen, denen Objekte aus der Rasch-Analyse übergeben werden müssen.

Funktion	benötigt Output von
• <code>plotjointICC()</code>	• <code>RM()</code> ¹⁵

- LRtest ()
 - plotICC ()
 - Waldtest ()

 - person.parameter ()
 - andere Packages ----
 - coef¹⁹ ()
 - confint²⁰ ()
 - logLik²¹ ()
 - model.matrix²² ()
 - summary²³ ()
 - vcov²⁴ ()

 - thresholds ()

 - plotGOF ()
- RM ()¹⁶, PCM ()¹⁷, RSM ()¹⁸

 - RM ()²⁵, PCM ()²⁶, RSM ()²⁷,
LLTM ()²⁸, LRSM ()²⁹, LPCM ()³⁰

 - PCM ()³¹, RSM ()³², LRSM ()³³,
LPCM ()³⁴

 - LRtest ()

16 Kapitel 32.2 auf Seite 97

17 Kapitel 32.5 auf Seite 120

18 Kapitel 32.4 auf Seite 113

19 <http://de.wikibooks.org/wiki/GNU%20R%3A%20coef>

20 <http://de.wikibooks.org/wiki/GNU%20R%3A%20confint>

21 <http://de.wikibooks.org/wiki/GNU%20R%3A%20logLik>

22 <http://de.wikibooks.org/wiki/GNU%20R%3A%20model.matrix>

23 <http://de.wikibooks.org/wiki/GNU%20R%3A%20summary>

24 <http://de.wikibooks.org/wiki/GNU%20R%3A%20vcov>

25 Kapitel 32.2 auf Seite 97

26 Kapitel 32.5 auf Seite 120

27 Kapitel 32.4 auf Seite 113

28 Kapitel 32.3 auf Seite 113

29 Kapitel 32.6 auf Seite 120

30 Kapitel 32.7 auf Seite 121

31 Kapitel 32.5 auf Seite 120

32 Kapitel 32.4 auf Seite 113

33 Kapitel 32.6 auf Seite 120

34 Kapitel 32.7 auf Seite 121

- `IC()`
- `itemfit()`
- `personfit()`
- `pmat()`
- andere Packages ----
- `plot35()`
- `residuals36()`
- `person.parameter()`

³⁵ <http://de.wikibooks.org/wiki/GNU%20R%3A%20plot>

³⁶ <http://de.wikibooks.org/wiki/GNU%20R%3A%20residuals>

33. Package Itm

34. Literatur

- Poinstingl, Mair & Hatzinger (2007): "Manual zum Softwarepackage eRm (extended Rasch modeling)", Pabst, ISBN 9783899674385
- Mair & Hatzinger (2007): "Extended Rasch modeling: The eRm package for the application of IRT Models in R", Journal of Statistical Software 20(9) (PDF)¹
- Rizopoulos (2006): "Irm: An R Package for Latent Variable Modeling and Item Response Analysis." Journal of Statistical Software 17(5) (PDF)²

1 <http://www.jstatsoft.org/v20/i09>
2 <http://www.jstatsoft.org/v17/i05/>

35. Weblinks

- Rasch-Modell bei Wikipedia¹

¹ <http://de.wikipedia.org/wiki/Rasch-Modell>

36. Datum- und Zeitfunktionen

37. Einleitung

Oft sind wir am Zeitpunkt der Datenerhebung interessiert. Typischerweise wird dann zusätzlich zu den relevanten Variablen eine oder mehrere Variable mit Datum und/oder Uhrzeit erhoben. Dies gilt ganz besonders für technische Anwendungen wie z.B. die Analyse von Messdaten oder die automatisierte Auswertung von Log-Dateien.

R verfügt über eine Vielzahl an Funktionen um Zeitpunkte zu repräsentieren, Zeitintervalle zu berechnen oder Zeitreihen zu analysieren, die in diesem Kapitel vorgestellt werden.

38. Grundlegende Datumfunktionen

38.0.1. Umwandeln von Character-Vektoren in Datum-Variablen

Typischerweise liegen Datum-Variablen nach einem Daten-Import nur als `character` vor:

```
R> datensatz$datum
[1] "1. Nov 2007 12:00"
[2] "3. Nov 2007 11:23"
[3] "3. Nov 2007 14:12"
[... ]
R> class(datensatz$datum)
[1] "character"
```

Um diese Variable als Datum zu behandeln, müssen wir definieren, in welcher Form das Datum vorliegt. Hierzu bietet sich die Funktion `strptime()`¹ an. Datumbestandteile und Trennzeichen lassen sich flexibel über den Parameter `format` angeben. Datumsbestandteile beginnen immer mit einem `%`-Zeichen. Die Bedeutung dieser Zeichen lässt sich über `?strptime` nachlesen, für unser Beispiel funktioniert die folgende Syntax:

```
R> datum <- strptime2(datensatz$datum, format="%d. %b %Y %R")
R> datum
[1] "2007-11-01 12:00:00"
[2] "2007-11-03 11:23:00"
[3] "2007-11-03 14:12:00"
[... ]

R> class(datum)

[1] "POSIXt" "POSIXlt"
```

Die Variable ist nun als Datum definiert.

38.0.2. Einfache Definition von Datum-Variablen

Die Funktion `ISOdate(Jahr, Monat, Tag, Stunde, Minute, Sekunde)`³ kann zur einfachen Definition eines Datum-Objekts verwendet werden:

¹ <http://de.wikibooks.org/wiki/GNU%20R%3A%20strptime>

³ <http://de.wikibooks.org/wiki/GNU%20R%3A%20ISOdate>

```
tag_der_befreiung<- ISOdate(1945,05,08,23,01)
tag_der_befreiung
[1] "1945-05-08 23:01:00 GMT"
```

38.0.3. Aktuelles Datum: Sys.time()

Die Funktion `Sys.time()`⁴ liefert das aktuelle Datum.

```
jetzt <- Sys.time()
```

38.0.4. Berechnen von Zeitintervallen

Zeitintervalle lassen sich mittels der Funktion `difftime()`⁵ oder noch einfacher folgendermassen berechnen:

```
jetzt - tag_der_befreiung
Time difference of 22851 days
intervall <- (jetzt-tag_der_befreiung)
class(intervall)
[1] difftime
```

Angenommen wir interessieren uns eigentlich für die *Stunden* seit der offiziellen Kapitulation des Nazi-Regimes. Dann können wir das Objekt `intervall` vom Typ `difftime` folgendermassen umrechnen:

```
as.numeric(intervall,units='hours')
[1] 548427
```

38.0.5. Plotten zeitabhängiger Daten

Die Funktion `plot()`⁶ verwendet eine spezielle Methode, wenn als *unabhängige* Variable Zeitdaten spezifiziert werden:

```
zeitpunkte <- sort(ISOdate(2007,1:12,1:12,0:23,0:23))
# erzeugt 24 Beispiel-Zeitpunkte und sortiert sie zeitlich

werte <- 1:24 - 4*sin(1:24)
# erzeugt 24 Beispielwerte
plot(werte~zeitpunkte,type='l',xlab='Jahr')
```

4 <http://de.wikibooks.org/wiki/GNU%20R%3A%20Sys.time>

5 <http://de.wikibooks.org/wiki/GNU%20R%3A%20difftime>

6 <http://de.wikibooks.org/wiki/GNU%20R%3A%20plot>

```
2007',ylab='Beispieldaten') # Plottet die Zeitreihe als  
Liniendiagramm
```

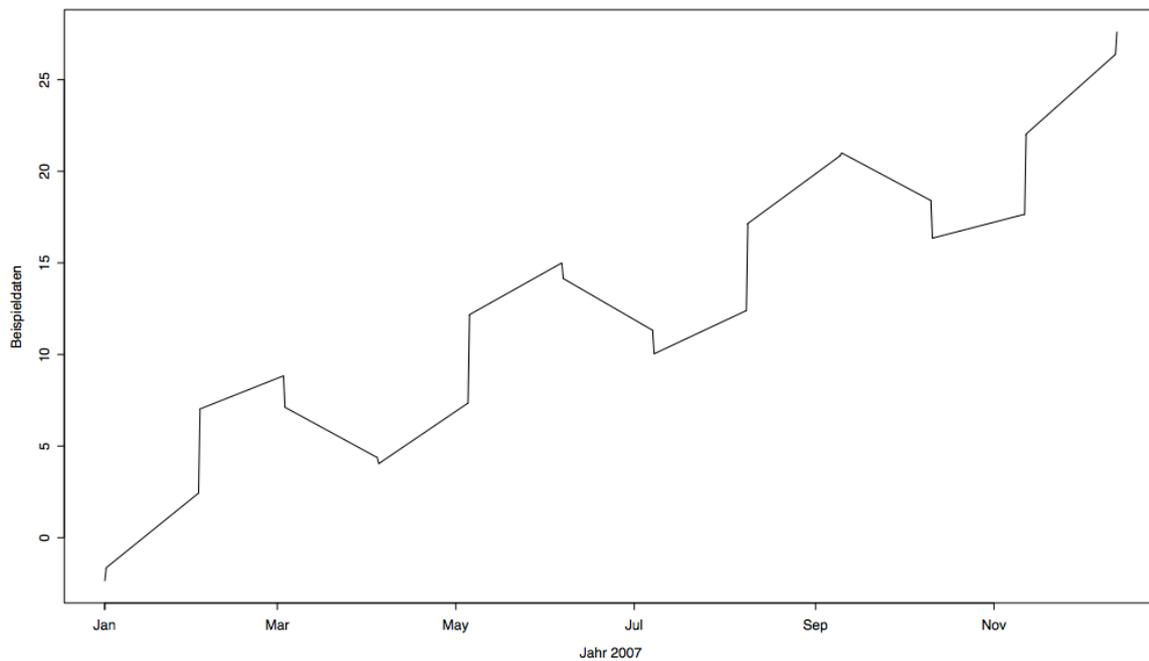


Abb. 7

Die Skalierung und Beschriftung der x-Achse passt sich entsprechend der Zeitspanne der Daten an und wird automatisch entweder in Stunden, Tagen, Monaten oder Jahren ausgegeben.

39. Analyse von Zeitreihen

Als einfachste Funktion zur Darstellung von Zeitreihen bietet R den `ts` Befehl an. Dieser ordnet mit der Struktur `ts(Datenvektor, start=Jahreszahl, frequency=Datenpunkte pro Jahr)` einer univariaten Reihe von Zahlen einen jeweiligen Zeitpunkt, beginnend mit dem Startjahr und in der genannten Frequenz pro Jahr, zu.

40. Block-Randomisierung

40.1. Einleitung

Innerhalb von klinischen Studien (Clinical Trials) kann es notwendig sein, eine Block-Randomisierung durchzuführen.

In R steht hierzu z.B. das Zusatzpaket¹ "blockrand" zur Verfügung.

40.2. blockrand

40.2.1. Installation

Das Paket wird mit folgendem Befehl über das Internet installiert:

```
install.packages("blockrand", dependencies=TRUE)
```

40.2.2. Verwendung

Bevor das Paket genutzt werden kann, muss es aktiviert werden per:

```
library(blockrand)
```

Jetzt kann eine einfache Randomisierungsliste erzeugt werden per:

```
blockrand(150,2)
```

Hierbei werden 150 Probanden zufällig in 2 Gruppen eingeteilt. Standardmäßig heißen diese Gruppen "A" und "B". Um ihnen eigene Labels zuzuweisen nutzt man die Option "levels":

```
blockrand(150,2, levels=c("Intervention", "Placebo"))
```

Auch die Blockgröße kann angegeben werden:

¹ Kapitel 7 auf Seite 17

```
blockrand(150,2, block.sizes=2, levels=c("Intervention", "Placebo"))
```

Hierbei gilt, dass der Wert von `block.sizes` von R mit 2 multipliziert wird, um dann Blockgrößen aller geraden Zahlen innerhalb des Werts zu generieren. Wird `block.sizes = 2` gewählt, generiert R zufällig Blöcke der Größe 2 und 4. Wird `block.sizes = 3` gewählt, generiert R zufällig Blöcke der Größe 2, 4 und 6.

40.3. siehe auch

Teil IV.

Programmieren mit R

41. Programmieren mit R

Ein weiterer Vorteil von R besteht darin, dass man neue Funktionsweisen nach seinen Bedürfnissen programmieren kann. Im Anhang Programmierbeispiele¹ werden die hier verwendeten Programmierbeispiele gesammelt.

41.1. Programmieren mit R

In R ist es möglich eigene Programme zu schreiben und eigene Zusatzpakete zu entwickeln. Elementar für die Programmierung ist die Möglichkeit eigene Funktionen zu definieren und die Möglichkeit Kontrollstrukturen zu benutzen. R erlaubt auch die rekursive Ausführung von Funktionen.

Für Programmierer die mit anderen Programmiersprachen arbeiten und sich mit der R-Programmierung vertraut machen wollen, sind folgende Unterschiede augenfällig:

- Die Handhabung von Parametern bei Funktionsaufrufen ist in R sehr variable. Die erscheinenden Spielarten des Funktionsaufrufe werden in einem späteren Abschnitt erläutert.
- Alle Aufrufe wie z.B. "getwd()" sind Methoden, auch wenn sie sie im interaktiven Modus wie Shell-Kommandos benutzt werden. Es sind immer Klammern zu verwenden. Der Vorteil ist, dass es keine Kollision zwischen Variablennamen und Methodenaufrufen geben kann.
- Die Namen von Methoden lehnen sich an bekannten Kommandonamen an, sind aber wie alle Namen in R schwer vorherzusehen. So liefert "ls()" nicht die Auflistung eines Verzeichnisses sondern eine Auflistung aller Variablen. Auch die in R verwandte Punktsyntax unterscheidet sich z.B. von jener in Java. So würde man in Java das Erzeugen einer Datentabelle eher in der Form `table.read("data.csv")` umsetzen. In R wird das Verb vor das Objekt gestellt `read.table("daten.csv")`.

41.1.1. Iterationen

Iteration, Schleifen, lassen sich in R auf drei Arten definieren, durch

- die **for**-Schleife
- die **while**-Schleife
- die **repeat**-Schleife.

Schleifen bestehen aus zwei Teilen der Schleifenkontrolle und dem Inhalt der Schleife. Der Schleifeninhalt ist ein eigener Programmblock, der mit geschweiften Klammern umschlossen wird. Wenn man mehr als einen Befehl in der Schleife ausführen möchte, das ist die Regel, dann müssen die einzelnen Befehle durch Semikolons getrennt werden. Jetzt konkret. Der Syntax der for-Schleife lautet *for(Variable in Wert){R Befehle in der Schleife}*. Zur Illustration dieser sehr abstrakten Beschreibung hier ein Beispiel:

¹ Kapitel 54 auf Seite 233

```
> x = 0
> for(i in 1:10){x=x+1; print(x);}
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
```

In diesem Beispiel wird innerhalb der Schleife, die von 1 bis 10 läuft die Variable **x** um eins erhöht und mit **print** ausgegeben. Aufeinander folgende Befehle müssen durch das Semikolon getrennt werden. Dieselbe Schleife kann mittels while-Anweisung wie folgt formuliert werden:

```
> x = 0
> while(x<10){x=x+1; print(x);}
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
```

Jetzt fehlt nur noch die repeat-Schleife. Eigentlich handelt es sich bei der repeat-Schleife um ein umgedrehte while-Schleife. In R ist es ein wenig komplizierter.

```
> x = 0
> repeat{x=x+1; print(x); if(x>=10) break}
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
[1] 6
[1] 7
[1] 8
[1] 9
[1] 10
```

Die repeat-Schleife muss explizit per break-Befehl beendet werden. Aus diesem Grund kommt hier die if-Anweisung zum Einsatz. Die if-Anweisung dient der bedingten Ausführung von Befehlen. Diese werden im nächsten Abschnitt beschrieben.

41.1.2. Bedingte Ausführung

Innerhalb der Funktion können Variablenbedingungen mit dem `if()`²-Befehl abgefragt werden. Der Aufruf erfolgt etwa so:

```
if3(VARIABLENBEDINGUNG) {FUNKTIONSWEISE}
```

Innerhalb der Klammern des `if()`⁴-Befehls werden die Variablenbedingungen gesetzt. Falls diese Bedingungen erfüllt sind, wird der Code innerhalb der geschweiften Klammern ausgeführt. Folgende Bedingungen können abgefragt werden:

Bedingungen

<code>==</code>	gleich
<code>!=</code>	ungleich
<code><</code>	kleiner
<code><=</code>	kleiner-gleich
<code>></code>	größer
<code>>=</code>	größer-gleich

Verknüpfungen

<code>&&</code>	und
<code> </code>	oder

Mit diesem Wissen könnten wir nun unsere Funktion etwas erweitern:

```
myfunc <- function5(x=3,y=7) {
  z <- x+y # Die Summe von x und y wird
  in z gespeichert
  if6(z>20) { # Abfrage, ob die Summe
  größer als 20 ist
    z <- "wow, bist du gross" # wenn ja, dann schreibe
    einen Text in das Objekt z
  }
  return7(z) # z wird zurückgemeldet
}
```

Mehrere Bedingungen können verknüpft werden, z.B. so:

```
myfunc <- function8(x=3,y=7) {
  if9(x<0 & y<0) { # Abfrage, ob x und y negativ sind
    x <- x*(-1) # wenn ja, dann mache beide
    positiv
    y <- y*(-1)
  }
  z <- x+y # Die Summe von x und y wird in z
  gespeichert
  if10(z==0 | z>50) { # Abfrage, ob z gleich 0 oder größer
  50 ist
    z <- "Summe ist 0 oder größer 50" # wenn ja, schreibe
```

² <http://de.wikibooks.org/wiki/GNU%20R%3A%20if>

⁴ <http://de.wikibooks.org/wiki/GNU%20R%3A%20if>

```
einen Text...  
    }  
    return11(z)           # z wird zurückgemeldet  
}
```

In der R-Sprache:

- eigene Funktionen programmieren¹²

41.2. siehe auch

- Programmierbeispiele¹³
- `package.skeleton()`¹⁴

¹² Kapitel 42 auf Seite 149

¹³ Kapitel 54 auf Seite 233

¹⁴ <http://de.wikibooks.org/wiki/GNU%20R%3A%20package.skeleton>

42. Eigene Funktionen programmieren

In R lassen sich eigene Funktionen problemlos integrieren. Die wohl einfachste Methode hierbei ist, die Funktionen in der R-Sprache selber zu schreiben. Dies wird in den folgenden Abschnitten beschrieben. Weiterhin besteht auch die Möglichkeit, neue Funktionen in Programmiersprachen wie z.B. C++ zu programmieren.

42.1. R-Sprache

42.1.1. Funktionsaufbau

Funktionen in R unterscheiden sich von Funktionen in anderen Programmiersprachen wie z.B. Java. Auffällige Unterschiede sind, dass Funktionen in ihrer Signatur keine Rückgabewerte beschreiben. Datenkapselung und Zugriffsarten wie sie aus Objektorientierten Programmiersprachen bekannt sind, gibt es nicht. Man kann Funktionen nicht als *private* oder *public* definieren. Auch kann man Variablen oder Parameter von Funktionen nicht als *final* deklarieren. Auch eine Fehlermanagement mit z.B. *Exceptions* ist in R nicht vorhanden.

R-Funktionen werden in der Regel in eigenen Dateien gespeichert. Diese R-Skriptdateien kann man mittels `source()`¹ laden. Wenn man diese Datei geladen hat, kann die darin enthaltenen Funktionen aufrufen. Die R-Skriptdateien haben im Vergleich zu anderen Programmiersprachen keine weitere Bedeutung. Oft muss man Dinge in R wiederholen, dann kann man diese Wiederkehrenden Prozessschritte in R-Scriptdateien auslagern. Dies spart Tipparbeit und macht die Analysen und Resultate nachvollziehbar.

Zum Erstellen einer eigenen Funktion steht der Befehl `function()`² zur Verfügung. Dieser ist wie folgt aufzurufen:

```
function(Parameter1, Parameter2, ...) {  
    Anweisungen;  
}
```

In den Klammern direkt nach `function`³ können Parameter benannt werden, welche dann beim Funktionsaufruf der Funktion zur Verfügung stehen. Innerhalb der geschweiften Klammern wird die eigentliche Funktionsweise geschrieben. Zu beachten dabei ist, dass mehrere Anweisungen innerhalb der FUNKTIONSWEISE mit einem Semikolon terminiert werden müssen.

1 <http://de.wikibooks.org/wiki/GNU%20R%3A%20source>

2 <http://de.wikibooks.org/wiki/GNU%20R%3A%20function>

3 <http://de.wikibooks.org/wiki/GNU%20R%3A%20function>

Um eine neue Funktion in R aufrufen zu können, wird die Funktionsweise einem neuen Objekt (z.B. `myfunc`) zugeordnet.

```
myfunc <- function(x,y){ # Neue Funktion mit den Parametern x und y
  z <- x+y;             # Die Summe von x und y wird in z
  gespeichert
  return4(z)           # z wird zurückgemeldet
}
```

Der obigen Funktion "myfunc" werden im Funktionsaufruf die Parameter `x` und `y` übergeben. Innerhalb der Funktion wird die Summe aus `x` und `y` gebildet, und per `return()`⁵ an die R-Konsole zurückgemeldet. Rufen wir diese Funktion beispielsweise auf mit:

```
myfunc(4,6)
```

erhalten wir als Rückmeldung:

```
10
```

Ein Vorteil von R ist die Möglichkeit, Standardwerte für die Parameter vorzugeben, welche verwendet werden, falls der Funktionsaufruf ohne diese Parameter erfolgt.

```
myfunc <- function(x=3,y=7){ # Neue Funktion mit Standardwerten für
  die Parameter x und y
  z <- x+y                 # Die Summe von x und y wird in z
  gespeichert
  return6(z)             # z wird zurückgemeldet
}
```

Rufen wir die Funktion **ohne** Parameter auf:

```
myfunc()
```

so werden die Standardwerte genutzt, und wir erhalten das Ergebnis

```
10
```

Rufen wir die Funktion **mit** Parametern auf, z.B.

```
myfunc(12,4)
```

so erhalten wir das Ergebnis

⁵ <http://de.wikibooks.org/wiki/GNU%20R%3A%20return>

42.1.2. Schleifen

Mit Schleifen können wir gewisse Textstellen in unserer Funktion beliebig bis unendlich oft wiederholen und uns somit Unmengen von Arbeit sparen:

```
Strecke <- function(x=10,y=x) {  
  while(a=0,a<=,a++) {  
    z=z+x          # z wird zu z+x  
    z=z+y          # z wird zu z+y  
  }  
  return(z)  
}                  # z wird zurückgemeldet
```

42.2. siehe auch

42.3. Literatur

42.4. Weblinks

43. Dateioperationen

Eine der wichtigsten Eigenschaften von Programmen ist Kooperation mit anderen Programmen. Also welche Schnittstellen (engl. Interfaces) bietet mein Programm. Wie kann ich Daten importieren, exportieren und wie kann ich mein Programm steuern. In dem ganzen Buch sind schon einzelne Dinge wie der Import von Daten beschrieben worden. R kann Daten aus verschiedenen Quellen importieren, die einfachste Form ist das Einlesen von CSV-Dateien. R ermöglicht eine Datei-basierte Eingabe, Ausgabe und Steuerung von R. Nachfolgend werden alle relevanten R-Funktionen beschrieben die diese Datei-basierte Ein-, Ausgabe und Steuerung ermöglichen.

43.1. Dateioperationen

Basis für alle R-Dateioperationen ist das Arbeitsverzeichnis (eng.: working directory). Alle Dateioperationen beziehen sich auf dieses Verzeichnis, es sei denn bei dem Aufruf einer Dateioperation wird explizit etwas anderes angegeben.

44. Eigene Zusatzpakete erstellen

R wird von einer großen Anzahl von Freiwilligen weiter auf- und ausgebaut.

Es besteht die Möglichkeit, sich mit selbsterstellten Funktionen und/oder Datensätzen am Aufbau zu beteiligen. Die gängigste Methode ist, seine Daten in einem eigenen Zusatzpaket¹ zu bündeln, und dieses dann über das CRAN der Allgemeinheit zugänglich zu machen.

44.1. Paket erstellen

Um die Daten und Funktionen zu einem Paket zu bündeln steht die Funktion `package.skeleton()`² zur Verfügung. Sie erstellt die erforderlichen Unterverzeichnisse des Pakets, legt die Daten und Funktionen in eben diesen ab und generiert Dummy-Dateien zur Dokumentation.

44.1.1. Die Paketverzeichnisse

Ein Paket besteht aus einem Hauptverzeichnis (welches den Namen des Pakets haben sollte) und mehreren Unterverzeichnissen. Die benötigten Paket-Unterverzeichnisse werden von `package.skeleton()`³ automatisch angelegt.

Die Inhalte der Verzeichnisse werden im Folgenden dargestellt:

1 Kapitel 7 auf Seite 17

2 <http://de.wikibooks.org/wiki/GNU%20R%3A%20package.skeleton>

3 <http://de.wikibooks.org/wiki/GNU%20R%3A%20package.skeleton>

Hauptverzeichnis:

- **DESCRIPTION:** Diese Datei gibt eine Beschreibung des Pakets. Das Format ihres Inhalts ist vorgegeben. Da mit `package.skeleton()` eine Dummy-DESCRIPTION-Datei angelegt wird, müssen die Inhalte nur noch entsprechend ausgefüllt bzw. ergänzt werden.
Package: (Name des Pakets)
Type: Package
Version: (Paketversion, z.B. 1.3)
Date: (Datum des Erstellens, z.B. 2007-06-02)
Author: (Liste der Autoren, mit Komma getrennt)
Maintainer: (Name des Hauptautors und dessen E-Mail-Adresse)
Depends: R (>= 2.4.0),
PKGNAME (R-Version und andere benötigte Zusatzpakete)
Description: (Eine kurze Beschreibung des Pakets, Sinn und Zweck)
License: GPL version 2 or newer
URL: (einen oder mehrere Links)

Unterverzeichnisse:

- **data:** In diesem Verzeichnis befinden sich die Datensätze, die in dem Paket enthalten sein sollen.
- **man:** Hier befinden sich alle Dokumentationen im "R Dokumentationformat (Rd)". Dokumentationen sollten für alle Datensätze und Funktionen des Pakets erstellt werden.
- **R:** In diesem Verzeichnis befinden sich die Funktionen des Pakets.

44.2. Dokumentation schreiben

Dokumentationen werden im "R Documentation"-Format geschrieben. Dies ist eine ASCII-Datei, welche die Dateiendung ".Rd" besitzt und ins man-Verzeichnis gelegt wird. Wie bereits oben beschreiben legt die Funktion `package.skeleton()` Dummy-Dokumentations-Dateien zu allen

Objekten, Daten und Funktionen an, welche mit dem Paket gebündelt werden sollen. Sie müssen "lediglich" noch mit den passenden Infos erweitert werden.

44.2.1. Dokumentation zu datasets

Jeder Datensatz muss eine Dokumentationsdatei besitzen. Diese ist wie folgt aufzubauen:

```

\name{DATENSATZNAME}      # hier wird der Name des Datensatzes
eingetragen
\docType{data}           # hier muss bei datasets immer data eingetragen
werden.
\title{KURZBESCHREIBUNG} # hier kommt eine kurze Überschrift der
Daten hin
\description{BESCHREIBUNG} # hier kommt eine kurze Beschreibung über
die Daten hin.
\usage{DATENAUFTRUF}     # hier wird beschrieben, wie die Daten
aufzurufen sind
                          (dies ist meist der Name des
Datensatzes).
\format{FORMAT}          # hier wird das Datenformat eingetragen,
also z.B. vector, matrix, data frame
                          bei Matritzen und Data frames sollte
eine Beschreibung der Zeilen und
                          Spalten gegeben werden.
\references{QUELLEN}     # hiermit können Literaturangaben
angegeben werden.
\source{QUELLE}          # hier wird eingetragen, woher die Daten
stammen (Originalquelle)
\keyword{datasets}       # hier muss bei datasets immer datasets eingetragen
werden.

```

44.2.2. Dokumentation zu Funktionen

Jede Funktion sollte im man-Verzeichnis eine eigene Dokumentationsdatei erhalten. Diese ist wie folgt aufzubauen:

```

\name{FUNKTIONSNAMEN}   # hier wird der Name der Funktion
eingetragen
\alias{ALIASNAME}       # Falls mehrere Funktionen auf die selbe
Help-Seite verweisen
                          werden diese hier mit je einer
alias-Zeile aufgelistet
\title{KURZBESCHREIBUNG} # hier kommt eine kurze Überschrift der
Funktion hin
\description{BESCHREIBUNG} # hier kommt eine kurze Beschreibung über
die Funktionsweise
                          der Funktion hin.
\usage{FUNKTIONSAUFTRUF} # hier wird beschrieben, wie die Funktion
aufzurufen ist.
                          z.B.: function(argument1, argument2,
...)
\arguments{              # hier werden die Argumente, die
übergeben werden können,
  \item{ARGUMENT1}       # durch die Aufzählung "item"
  \item{ARGUMENT2}
}

```

```
\details{DETAILS}          # hier kommt eine exakte Beschreibung
hin, welche den "description"
                          -Part ergänzt und erweitert.
\references{QUELLEN}       # hiermit können Quellenangaben angegeben
werden.
\note{NOTIZ}              # hier können Notizen eingetragen werden.
\author{AUTOR}            # Informationen über die Autoren. URLs
und E-Mail können innerhalb
                          per \url{http://www.foo.bar} und
\email{e@mail.de}         angegeben werden
\examples{BEISPIELE}      # innerhalb der Klammern können
Beispielaufrufe formuliert werden.
\keyword{keyword1}        # Hier muss mindestens ein R-Keyword
gesetzt werden. Für eine Übersicht
\keyword{keyword2}        # an Keywords tippen Sie ein: file.show(file.path(R.home("doc"),
"KEYWORDS"))
```

44.2.3. Dokumentation zum Paket

Im Gegensatz zu Datensätzen und Funktionen muss die Dummy-Dokumentations-Datei zur Paketdokumentation mit dem Befehl `promptPackage()`⁴ angelegt werden. Die einzig geforderte Angabe innerhalb dieser Datei ist:

```
\docType{package}        # das "package" muss so stehen bleiben!
```

44.3. Paket überprüfen und fertig stellen

Wenn man alle oben aufgeführten Schritte durchgeführt hat, ist es an der Zeit, das Paket fertigzustellen. Bevor dies geschieht, muss das Paket allerdings noch auf Fehler überprüft werden. Dies geschieht in der Shell. Man geht hierzu in das Verzeichnis, in welchem das Hauptverzeichnis des Pakets (als Unterverzeichnis) liegt, und tippt in die Shell den folgenden Befehl ein:

```
R CMD check PAKETNAME
```

Da das Paketverzeichnis den Namen des Pakets tragen sollte (s.o.), weiss das Script, wo es nach Daten suchen muss. Nun wird das Paket auf mögliche Fehler hin überprüft. Es ist wichtig, dass man erst weiter arbeitet, wenn diese Routine fehlerfrei durchläuft!!!

Ist die Überprüfung fehlerfrei verlaufen, kann das Paket zu einer `tar.gz`-Datei geschnürt werden. Hierzu gibt man in die Shell ein:

```
R CMD build PAKETNAME
```

⁴ <http://de.wikibooks.org/wiki/GNU%20R%3A%20promptPackage>

Das Script holt sich die passenden Versions-Informationen aus der DESCRIPTION-Datei, so dass die tar-Datei (entsprechend der Version) automatisch als z.B. PKGNAME_1.12.tar.gz erstellt wird.

44.4. Paket zum CRAN hochladen

Das fertige Paket kann nun per FTP zum CRAN hochgeladen werden. Die URL lautet:

- <ftp://cran.R-project.org/incoming/>
- Username: anonymous
- Passwort: (Ihre E-Mail-Adresse)

Mit der als Passwort angegebenen Adresse muss dann eine Info-Mail an cran@r-project.org⁵ gesendet werden. Das Paket wird zunächst auf mögliche Fehler überprüft und anschließend dem CRAN hinzugefügt.

44.5. Weblinks

- [Writing R Extensions \(PDF\)](#)⁶
- [Das wikibooks-Paket](#)⁷ **BROKEN LINK**
- [Pakete erstellen](#)⁸

5 <mailto:cran@r-project.org>

6 <http://cran.r-project.org/doc/manuals/R-exts.pdf>

7 <http://cran.at.r-project.org/src/contrib/Descriptions/wikibooks.html>

8 http://www.r-statistik.de/R_erweitern/Pakete/pakete.html

45. Andere Programmiersprachen einbinden

Auch andere Programmiersprachen können eingebunden werden wie z.B.:

45.1. C++

45.2. Java

46. Finden von Programmfehlern und Debugging

Ursache von Programmfehlern sind zumeist Fehlannahmen über die von (Unter-)Funktionen zurückgegebenen Inhalte oder Datentypen der übergebenen Objekte. Oft ist es deshalb wünschenswert, den Zustand beim Programmabbruch genauer zu analysieren. R stellt hierfür verschiedene Methoden und Optionen zur Verfügung.

R unterstützt das Auffinden von Fehlern mit mehreren Hilfsfunktionen und Optionen. Welche bei der Fehlersuche eingesetzt werden, hängt letztlich von Vorlieben der ProgrammiererInnen ab. Dieser Abschnitt soll einen Überblick über die Möglichkeiten geben.

46.1. Manuelles Einfügen von Ausgabewerten

R gibt zumeist relativ präzise Fehlermeldungen, warum ein Programmabbruch erfolgt ist, aber leider keine Zeilenangaben. Dies erschwert es manchmal, die genaue Stelle zu finden, an der der Fehler auftrat. Daher ist es oft sinnvoll, mehrere `print` oder `cat`-Funktionen mit unterscheidbarer Ausgabe an verschiedenen Stellen zu setzen. Falls die entsprechende Ausgabe erfolgt, ist der Fehler an dieser Stelle noch nicht aufgetreten.

Wenn eingegrenzt werden konnte, wo der Fehler auftritt, macht es Sinn, sich die Zustände der der Funktion übergebenen Objekte an charakteristischen Stellen mit `print` ausgeben zu lassen. Je nach Fehlermeldung sind auch die Funktionen `length` und `class` sehr nützlich bei der Fehlersuche.

46.2. Gängige Fehler

Wie bereits erwähnt treten Fehler häufig deshalb auf, weil Funktionen andere Objekte übergeben werden als diese erwarten. Wer verstanden hat, wie Objektklassen funktionieren, dürfte auch die meisten der Fehlermeldungen verstehen. Dennoch kann die Interpretation von Fehlermeldungen und die Fehlersuche in R nicht nur Anfängern immer wieder Schwierigkeiten bereiten.

Hier sollen ein paar häufige Fehler und Lösungen beschrieben werden:

- Manchmal ist es nicht ganz einfach zu verstehen, welche Klasse eine Funktion erwartet. Auskunft hierüber gibt immer die Hilfe-Seite einer Funktion. Hier wird meist genau unter `arguments` beschrieben was die Funktion an Eingabeobjekten erwartet. Unter `value` sind die Objekte und deren Klasse beschrieben, die die Funktion zurückgibt.
- Problematisch ist z.B. die Klasse `factor`. Sie verhält sich aus Nutzersicht wie ein `character`-Vektor, besteht aber in Wahrheit aus Elementen vom Typ `numeric`. Dies ist insbesondere beim Datenimport aus Dateien (z.B. mit `read.csv`) problematisch, wenn Spalten mit Nummern

(z.B. wegen zusätzlicher Buchstaben) als Strings importiert und auf Grund der Einstellungen automatisch ein `factor` erzeugt werden. Eine Konvertierung in `as.numeric` gibt dann unter Umständen nicht die erwarteten Werte zurück, sondern die interne Repräsentation des `factor`. Eine korrekte Behandlung erfordert in diesem Fall kontraintuitives Vorgehen. `as.numeric (as.character (variable))`.

- *Listenreferenzierungen* sorgen bei Anfängern ebenfalls oft für Verwirrung.

46.3. Error-Recovery mit dem Browser

Eine der nützlichsten Einstellungen betrifft die Frage, was passiert, wenn ein Fehler aufgetreten ist. Dies kann über `options(error)` festgelegt werden: Standardmässig wird eine Fehlermeldung dargestellt und die Programmausführung abgebrochen: `options(error=print)` Stattdessen kann aber mit `options(error=recover)` im Fehlerfall auch automatisch ein so genannter *Browser* gestartet werden. Dies soll in einem Beispiel verdeutlicht werden. Einem `data.frame` werden statt Vektoren oder Listen der Code der Funktion `sum` übergeben:

```
> options(error=print)
> data.frame(1:10,sum)

Fehler in as.data.frame.default(x[[i]] , optional = TRUE) :

kann Klasse '"function"' nicht in data.frame umwandeln
```

Wenn `options(error=recover)` gesetzt wird, springt R in den browser:

```
> options(error=recover)
> data.frame(1:10,sum)
Fehler in as.data.frame.default(x[[i]], optional = TRUE) :

kann Klasse '"function"' nicht in data.frame umwandeln

Enter a frame number, or 0 to exit

1: data.frame(1:10, sum)

2: as.data.frame(x[[i]], optional = TRUE)

3: as.data.frame.default(x[[i]], optional = TRUE)

Auswahl: _
```

Der Browser dient dazu durch die Hierarchien des Funktionsaufrufs (sogenannte frames) zum Zeitpunkt des Fehlers zu navigieren. In diesem Fall gibt es drei frames: Den Funktionsaufruf (1:) und zwei von der Funktion `data.frame` aufgerufene Unterfunktionen (2: und 3:).

Die Browser hat zwei Zustände:

- `frame-Auswahl`: In der `frame-Auswahl` kann durch Eingabe der entsprechenden Nummern in den entsprechenden frame gesprungen werden.
- `frame-Editor`: Im `frame-Editor` können alle Funktionen und Kommandos wie gewöhnlich benutzt werden. Insbesondere werden Objekte durch einfache Eingabe ihres Namens ausgegeben. Durch eine Leereingabe `RETURN` springt der Browser zurück in die `frame-Auswahl`.

Wählen wir beispielsweise `frame 1`, unseren ursprünglichen Funktionsaufruf, in dem wir 1 eingeben:

```
Auswahl: 1

Called from: data.frame(1:10, sum)
```

Wir befinden uns nun in `frame 1` und können die Objekte dort betrachten. Um alle Objekte des `frame` aufzulisten, benutzen wir die Funktion `ls()`:

```
ls()
 [1] "check.names"      "check.rows"      "data.row.names"  "i"
 [5] "mrn"              "n"               "namesi"
"ncols"
 [9] "no.vn"           "nrows"          "object"
"row.names"
[13] "stringsAsFactors" "tmpname"         "vlist"
"vnames"

[17] "x"                "xi"
```

Dies sind alle Variablen, die die Funktion `data.frame()` intern bei der Erstellung neuer Objekte generiert. Uns interessieren hier beispielhaft die der Funktion `data.frame` übergebenen Werte, die in `x` gespeichert sind:

```
[[1]]

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

[[2]]

function (... , na.rm = FALSE) .Primitive("sum")
```

Erkennbar ist, dass eine Liste mit zwei Werten übergeben wurde. Das Listenelement `[[1]]` enthält die Werte 1:10, das Listenelement `[[2]]` die Funktion `sum`.

Wir verlassen den Browser durch eine Leereingabe (RETURN) (zum Wechseln in die frame-Auswahl) und die Eingabe einer 0 (zum Beenden des Browsers)

Derart lassen sich alle Werte zum Zeitpunkt des Programmabbruchs untersuchen. Wenn ein Fehler auftritt und nicht klar ist, welche Funktionen zum Fehler führten und welche Werte vorher übergeben wurden, sollte generell `options(error=recover)` gesetzt werden. Die error-recovery kann nach der Korrektur der Fehler mit `options(error=print)` wieder deaktiviert werden.

46.4. Verfolgen von Funktionen mit trace

Teil V.
Anhang

47. Anwendungsbeispiele

In diesem Anhang finden sich Anwendungsbeispiele und -datensätze, die zum Testen der beschriebenen R-Funktionen dienen

48. Datenbeispiele

48.1. Beispiel 1

Geschlecht	Alter	Gewicht	Groesse
m	28	80	170
w	18	55	174
w	25	74	183
m	29	101	190
m	21	84	185
w	19	74	178
w	27	65	169
w	26	56	163
m	31	88	189
m	22	78	184

R-Befehl zum Erzeugen dieser Tabelle:

```
w <- factor1(c2("m", "w", "w", "m", "m", "w", "w", "w", "m", "m"))
x <- c3(28, 18, 25, 29, 21, 19, 27, 26, 31, 22)
y <- c4(80, 55, 74, 101, 84, 74, 65, 56, 88, 78)
z <- c5(170, 174, 183, 190, 185, 178, 169, 163, 189, 184)
bspl <- data.frame6(w, x, y, z)
colnames7(bspl) <- c("Geschlecht", "Alter", "Gewicht", "Groesse")
rm8(w, x, y, z)
bspl
```

48.2. Beispiel 2

Geschlecht	Note
m	2
w	1
m	5
m	3
w	4
m	3
w	2
w	2
w	1
m	3
m	1
w	4
m	2
w	1
m	4
m	3
w	5

```
m      2
w      1
w      2
```

R-Befehl zum Erzeugen dieser Tabelle:

```
x <- factor9(c10("m", "w", "m", "m", "w", "m", "w", "w", "w", "w", "m", "m", "w",
"m", "w", "m", "m", "w", "m", "w", "w"))
y <- c11(2, 1, 5, 3, 4, 3, 2, 2, 1, 3, 1, 4, 2, 1, 4, 3, 5, 2, 1, 2)
bsp2 <- data.frame12(x,y)
colnames13(bsp2) <- c("Geschlecht", "Note")
rm14(x,y)
bsp2
```

48.3. Beispiel 3

	Erfolg	Abschlussnote
1	0	5
2	1	3
3	1	2
4	0	4
5	1	1
6	0	6
7	1	3
8	1	2
9	0	4
10	1	3
11	0	6
12	0	5
13	0	4
14	1	3

R-Befehl zum Erzeugen dieser Tabelle:

```
a <- factor15(c16(0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1))
b <- c17(5, 3, 2, 4, 1, 6, 3, 2, 4, 3, 6, 5, 4, 3)
bsp3 <- data.frame18(a,b)
colnames19(bsp3) <- c("Erfolg", "Abschlussnote")
rm20(a,b)
bsp3
```

48.4. Beispiel 4

	Name	Geschlecht	Lieblingsfarbe	Einkommen
1	Hans	maennlich	gruen	1233
2	Caro	weiblich	blau	800
3	Lars	intersexuell	gelb	2400
4	Ines	weiblich	schwarz	4000
5	Samira	weiblich	gelb	899
6	Peter	maennlich	gruen	1100
7	Sarah	weiblich	blau	1900

R-Befehl zum Erzeugen dieser Tabelle:

```
w <- c21("Hans","Caro","Lars","Ines","Samira","Peter","Sarah")
x <- factor22(c23("maennlich","weiblich","intersexuell","weiblich","maennlich","weiblich"))
y <- factor24(c25("gruen","blau","gelb","schwarz","gelb","gruen","blau"))
z <- c26(1233,800,2400,4000,899,1100,1900)
bsp4 <- data.frame27(w,x,y,z)
colnames28(bsp4) <- c("Name", "Geschlecht", "Lieblingsfarbe", "Einkommen")
rm29(w,x,y,z)
bsp4
```

48.5. Beispiel 5

	Lebenstag	Gewicht
1	1	3180
2	3	2960
3	6	3220
4	11	3270
5	12	3350
6	15	3410
7	19	3700
8	23	3830
9	28	4090
10	33	4310
11	35	4360
12	39	4520
13	47	4650
14	60	5310
15	66	5490
16	73	5540

R-Befehl zum Erzeugen dieser Tabelle:

```
x <- c30(1, 3, 6, 11, 12, 15, 19, 23, 28, 33, 35, 39, 47, 60, 66, 73)
y <- c31(3180, 2960, 3220, 3270, 3350, 3410, 3700, 3830, 4090, 4310,
4360, 4520, 4650, 5310, 5490, 5540)
bsp5 <- data.frame32(x,y)
colnames33(bsp5) <- c("Lebenstag", "Gewicht")
rm34(x,y)
```

48.6. Beispiel 6

	Person	Date	Age	Sex	DimBody	DimEmotion	DimSelf	DimFamily
1	190053	2008-08-20	15	f	75.00	85.00	65.00	85.00
	70.00	90.00	78.33					
2	190050	2008-01-24	16	m	80.00	80.00	75.00	65.00
	70.00	60.00	71.67					
3	190052	2008-02-05	11	m	70.00	65.00	35.00	80.00
	45.00	95.00	65.00					
4	190002	2006-12-07	11	f	85.00	90.00	85.00	70.00
	80.00	55.00	77.50					
5	190002	2008-01-31	13	f	75.00	90.00	75.00	55.00

Datenbeispiele

	85.00	70.00	75.00				
6	190005	2007-02-22	12 m	65.00	90.00	70.00	95.00
	75.00	65.00	76.67				
7	190035	2007-05-22	14 f	60.00	85.00	65.00	75.00
	80.00	55.00	70.00				
8	190021	2006-08-30	16 f	55.00	50.00	50.00	30.00
	55.00	55.00	49.17				
9	190014	2006-12-07	12 m	75.00	90.00	40.00	85.00
	75.00	60.00	70.83				
10	190029	2006-11-21	13 f	55.00	70.00	60.00	65.00
	85.00	55.00	65.00				
11	190025	2006-12-07	12 f	100.00	95.00	95.00	100.00
	90.00	85.00	94.17				
12	190025	2007-02-20	13 f	85.00	95.00	85.00	85.00
	95.00	75.00	86.67				
13	190036	2007-06-05	12 m	55.00	75.00	80.00	85.00
	90.00	75.00	76.67				
14	190031	2007-02-13	13 f	60.00	80.00	60.00	80.00
	90.00	55.00	70.83				
15	190034	2007-05-19	16 f	50.00	55.00	45.00	80.00
	70.00	30.00	55.00				
16	190044	2007-10-23	13 m	75.00	65.00	60.00	85.00
	60.00	75.00	70.00				
17	190041	2007-08-16	16 f	90.00	90.00	90.00	100.00
	70.00	60.00	83.33				
18	190046	2007-11-15	14 f	70.00	75.00	90.00	70.00
	80.00	75.00	76.67				
19	190034	2007-07-19	16 f	50.00	60.00	45.00	90.00
	75.00	55.00	62.50				
20	191028	2006-10-26	7 f	66.67	100.00	100.00	66.67
	50.00	83.33	77.78				
21	191035	2007-05-22	6 f	66.67	83.33	66.67	100.00
	100.00	100.00	86.11				
22	191037	2007-05-31	6 m	66.67	83.33	100.00	83.33
	66.67	83.33	80.56				
23	191036	2007-06-19	6 f	100.00	83.33	100.00	83.33
	83.33	66.67	86.11				
24	192052	2008-02-05	8 f	80.00	60.00	60.00	80.00
	75.00	65.00	70.00				
25	190028	2006-10-26	9 f	65.00	65.00	60.00	80.00
	75.00	75.00	70.00				
26	190028	2008-02-15	11 f	85.00	90.00	70.00	95.00
	90.00	90.00	86.67				
27	191028	2008-02-16	9 f	80.00	95.00	100.00	95.00
	80.00	85.00	89.17				
28	190039	2007-05-29	9 f	70.00	90.00	80.00	95.00
	90.00	95.00	86.67				
29	190037	2007-05-31	8 m	90.00	75.00	60.00	100.00
	90.00	80.00	82.50				
30	191030	2007-02-22	9 f	80.00	75.00	60.00	85.00
	85.00	60.00	74.17				
31	192030	2007-02-22	8 m	85.00	95.00	60.00	85.00
	60.00	80.00	77.50				
32	190030	2007-02-22	11 f	70.00	90.00	60.00	65.00
	80.00	55.00	70.00				
33	190022	2006-05-07	10 m	95.00	95.00	80.00	85.00
	50.00	65.00	78.33				
34	190048	2008-01-08	11 f	65.00	75.00	60.00	85.00
	70.00	60.00	69.17				
35	190017	2007-12-18	8 f	40.00	80.00	70.00	60.00
	55.00	75.00	63.33				
36	190015	2006-11-28	12 f	60.00	60.00	60.00	75.00
	90.00	80.00	70.83				
37	191002	2008-01-31	9 f	80.00	100.00	90.00	70.00
	95.00	65.00	83.33				
38	271050	2008-01-31	13 m	85.00	65.00	70.00	70.00
	70.00	70.00	71.67				
39	270050	2008-01-31	16 m	55.00	60.00	60.00	55.00

	70.00	50.00	58.33				
40	270052	2008-02-05	11 m	40.00	45.00	55.00	65.00
	40.00	65.00	51.67				
41	271052	2008-02-05	10 m	85.00	45.00	40.00	70.00
	50.00	55.00	57.50				
42	272052	2008-02-05	8 f	60.00	55.00	70.00	70.00
	80.00	75.00	68.33				
43	271002	2007-02-22	8 f	80.00	65.00	65.00	75.00
	40.00	60.00	64.17				
44	271002	2007-01-26	8 f	85.00	80.00	80.00	75.00
	80.00	90.00	81.67				
45	270002	2007-01-26	12 f	75.00	85.00	90.00	70.00
	95.00	75.00	81.67				
46	271002	2008-01-31	9 f	60.00	60.00	65.00	60.00
	70.00	75.00	65.00				
47	270002	2008-01-31	13 f	80.00	70.00	90.00	70.00
	85.00	80.00	79.17				
48	270005	2007-01-23	12 m	65.00	45.00	45.00	45.00
	50.00	60.00	51.67				
49	270005	2007-01-23	12 m	65.00	55.00	45.00	35.00
	55.00	55.00	51.67				
50	270010	2007-05-22	9 f	70.00	65.00	75.00	90.00
	50.00	95.00	74.17				
51	270014	2007-04-18	13 m	75.00	65.00	50.00	65.00
	30.00	65.00	58.33				
52	270022	2006-12-07	10 m	85.00	90.00	75.00	100.00
	55.00	70.00	79.17				
53	270025	2006-11-30	12 f	100.00	80.00	80.00	85.00
	95.00	95.00	89.17				
54	270025	2007-02-20	13 f	90.00	95.00	80.00	80.00
	95.00	95.00	89.17				
55	270028	2006-10-26	9 f	55.00	65.00	60.00	45.00
	50.00	35.00	51.67				
56	270028	2008-02-16	11 f	55.00	45.00	50.00	50.00
	50.00	55.00	50.83				
57	270029	2006-11-21	13 f	70.00	50.00	70.00	50.00
	85.00	80.00	67.50				
58	271030	2006-12-14	9 f	35.00	65.00	60.00	55.00
	75.00	45.00	55.83				
59	270030	2006-12-14	11 f	60.00	55.00	35.00	60.00
	60.00	55.00	54.17				
60	270035	2007-05-22	14 f	60.00	60.00	55.00	65.00
	90.00	55.00	64.17				
61	270036	2007-06-05	12 m	80.00	60.00	65.00	80.00
	70.00	85.00	73.33				
62	270036	2007-06-19	12 m	60.00	55.00	75.00	75.00
	65.00	85.00	69.17				
63	270037	2007-05-31	8 m	80.00	90.00	75.00	85.00
	95.00	85.00	85.00				
64	270039	2007-05-29	9 f	45.00	80.00	70.00	90.00
	95.00	85.00	77.50				
65	270041	2007-08-16	16 f	85.00	95.00	75.00	75.00
	90.00	70.00	81.67				
66	270044	2007-10-23	13 m	60.00	65.00	55.00	75.00
	55.00	50.00	60.00				
67	270046	2007-11-15	14 f	60.00	70.00	65.00	80.00
	60.00	80.00	69.17				
68	270048	2008-01-08	11 f	60.00	60.00	55.00	80.00
	75.00	50.00	63.33				
69	270002	2007-02-22	12 f	70.00	75.00	65.00	55.00
	75.00	60.00	66.67				
70	270004	2006-11-25	14 f	90.00	100.00	100.00	95.00
	85.00	100.00	95.00				
71	270017	2007-12-13	8 f	60.00	80.00	65.00	85.00
	50.00	80.00	70.00				
72	271028	2008-02-16	9 f	95.00	75.00	75.00	70.00
	80.00	75.00	78.33				
73	270015	2006-11-28	12 f	85.00	90.00	85.00	75.00

Datenbeispiele

85.00	95.00	85.83						
74	271036	2007-06-19	6	f	85.00	85.00	80.00	60.00
95.00	80.00	80.83						
75	270045	2007-10-23	7	m	75.00	80.00	55.00	70.00
60.00	75.00	69.17						
76	270040	2007-08-07	5	f	90.00	70.00	75.00	75.00
65.00	70.00	74.17						
77	271037	2007-05-31	6	m	80.00	90.00	80.00	95.00
80.00	55.00	80.00						
78	271035	2007-05-22	7	f	80.00	100.00	85.00	80.00
75.00	75.00	82.50						
79	271010	2007-05-22	6	m	100.00	65.00	70.00	90.00
75.00	80.00	80.00						
80	270033	2007-03-22	4	m	80.00	80.00	90.00	75.00
75.00	65.00	77.50						
81	271036	2007-06-05	7	f	60.00	80.00	70.00	85.00
85.00	65.00	74.17						
82	271028	2006-10-26	7	f	95.00	85.00	80.00	90.00
70.00	65.00	80.83						

R-Befehl zur Erzeugung des Datensatzes:

```
bsp6 <- structure(list(Person = c("190053", "190050", "190052",  
"190002", "190002", "190005",  
"190035", "190021", "190014", "190029", "190025", "190025",  
"190036", "190031", "190034",  
"190044", "190041", "190046", "190034", "191028", "191035",  
"191037", "191036", "192052",  
"190028", "190028", "191028", "190039", "190037", "191030",  
"192030", "190030", "190022",  
"190048", "190017", "190015", "191002", "271050", "270050",  
"270052", "271052", "272052",  
"271002", "271002", "270002", "271002", "270002", "270005",  
"270005", "270010", "270014",  
"270022", "270025", "270025", "270028", "270028", "270029",  
"271030", "270030", "270035",  
"270036", "270036", "270037", "270039", "270041", "270044",  
"270046", "270048", "270002",  
"270004", "270017", "271028", "270015", "271036", "270045",  
"270040", "271037", "271035",  
"271010", "270033", "271036", "271028"), Date =  
structure(c(1219183200, 1201129200,  
1202166000, 1165446000, 1201734000, 1172098800, 1179784800,  
1156888800, 1165446000,  
1164063600, 1165446000, 1171926000, 1180994400, 1171321200,  
1179525600, 1193090400,  
1187215200, 1195081200, 1184796000, 1161813600, 1179784800,  
1180562400, 1182204000,  
1202166000, 1161813600, 1203030000, 1203116400, 1180389600,  
1180562400, 1172098800,  
1172098800, 1172098800, 1146952800, 1199746800, 1197932400,  
1164668400, 1201734000,  
1201734000, 1201734000, 1202166000, 1202166000, 1202166000,  
1172098800, 1169766000,  
1169766000, 1201734000, 1201734000, 1169506800, 1169506800,  
1179784800, 1176847200,  
1165446000, 1164841200, 1171926000, 1161813600, 1203116400,  
1164063600, 1166050800,  
1166050800, 1179784800, 1180994400, 1182204000, 1180562400,  
1180389600, 1187215200,  
1193090400, 1195081200, 1199746800, 1172098800, 1164409200,  
1197500400, 1203116400,  
1164668400, 1182204000, 1193090400, 1186437600, 1180562400,  
1179784800, 1179784800,
```

```

1174518000, 1180994400, 1161813600), class = c("POSIXt",
"POSIXct" ), tzzone = ""),
Age = c(15, 16, 11, 11, 13, 12, 14, 16, 12, 13, 12, 13,
16, 13, 16, 14, 16, 7, 6,
6, 6, 8, 9, 11, 9, 9, 8, 9, 8, 11, 10, 11, 8, 12, 9, 13, 16,
11, 10, 8, 8, 8, 12, 9, 13,
12, 12, 9, 13, 10, 12, 13, 9, 11, 13, 9, 11, 14, 12, 12, 8, 9,
16, 13, 14, 11, 12, 14, 8,
9, 12, 6, 7, 5, 6, 7, 6, 4, 7, 7), Sex = structure(c(1L, 2L,
2L, 1L, 1L, 2L, 1L, 1L, 2L,
1L, 1L, 1L, 2L, 1L, 1L, 1L, 1L, 1L, 2L, 1L, 1L, 1L,
1L, 1L, 1L, 2L, 1L, 2L,
1L, 2L, 1L, 1L, 1L, 1L, 2L, 2L, 2L, 2L, 1L, 1L, 1L, 1L, 1L, 1L,
2L, 2L, 1L, 2L, 2L, 1L, 1L,
1L, 1L, 1L, 1L, 1L, 1L, 1L, 2L, 2L, 2L, 1L, 1L, 2L, 1L, 1L, 1L, 1L,
1L, 1L, 1L, 1L, 2L, 1L, 2L,
1L, 2L, 2L, 1L, 1L), .Label = c("f", "m"), class = "factor"),
DimBody = c(75, 80, 70, 85,
75, 65, 60, 55, 75, 55, 100, 85, 55, 60, 50, 75, 90, 70, 50,
66.67, 66.67, 66.67, 100, 80,
65, 85, 80, 70, 90, 80, 85, 70, 95, 65, 40, 60, 80, 85, 55, 40,
85, 60, 80, 85, 75, 60, 80,
65, 65, 70, 75, 85, 100, 90, 55, 55, 70, 35, 60, 60, 80, 60,
80, 45, 85, 60, 60, 60, 70,
90, 60, 95, 85, 85, 75, 90, 80, 80, 100, 80, 60, 95),
DimEmotion = c(85, 80, 65, 90, 90,
90, 85, 50, 90, 70, 95, 95, 75, 80, 55, 65, 90, 75, 60, 100,
83.33, 83.33, 83.33, 60, 65,
90, 95, 90, 75, 75, 95, 90, 95, 75, 80, 60, 100, 65, 60, 45,
45, 55, 65, 80, 85, 60, 70,
45, 55, 65, 65, 90, 80, 95, 65, 45, 50, 65, 55, 60, 60, 55, 90,
80, 95, 65, 70, 60, 75,
100, 80, 75, 90, 85, 80, 70, 90, 100, 65, 80, 80, 85), DimSelf =
c(65, 75, 35, 85, 75, 70,
65, 50, 40, 60, 95, 85, 80, 60, 45, 60, 90, 90, 45, 100, 66.67,
100, 100, 60, 60, 70, 100,
80, 60, 60, 60, 60, 80, 60, 70, 60, 90, 70, 60, 55, 40, 70, 65,
80, 90, 65, 90, 45, 45,
75, 50, 75, 80, 80, 60, 50, 70, 60, 35, 55, 65, 75, 75, 70, 75,
55, 65, 55, 65, 100, 65,
75, 85, 80, 55, 75, 80, 85, 70, 90, 70, 80), DimFamilie = c(85,
65, 80, 70, 55, 95, 75,
30, 85, 65, 100, 85, 85, 80, 80, 85, 100, 70, 90, 66.67, 100,
83.33, 83.33, 80, 80, 95, 95,
95, 100, 85, 85, 65, 85, 85, 60, 75, 70, 70, 55, 65, 70, 70,
75, 75, 70, 60, 70, 45, 35,
90, 65, 100, 85, 80, 45, 50, 50, 55, 60, 65, 80, 75, 85, 90,
75, 75, 80, 80, 55, 95, 85,
70, 75, 60, 70, 75, 95, 80, 90, 75, 85, 90), DimFriends = c(70,
70, 45, 80, 85, 75, 80, 55,
75, 85, 90, 95, 90, 90, 70, 60, 70, 80, 75, 50, 100, 66.67,
83.33, 75, 75, 90, 80, 90, 90,
85, 60, 80, 50, 70, 55, 90, 95, 70, 70, 40, 50, 80, 40, 80, 95,
70, 85, 50, 55, 50, 30, 55,
95, 95, 50, 50, 85, 75, 60, 90, 70, 65, 95, 95, 90, 55, 60, 75,
75, 85, 50, 80, 85, 95,
60, 65, 80, 75, 75, 75, 85, 70), DimSchool = c(90, 60, 95, 55,
70, 65, 55, 55, 60, 55, 85,
75, 75, 55, 30, 75, 60, 75, 55, 83.33, 100, 83.33, 66.67, 65,
75, 90, 85, 95, 80, 60, 80,
55, 65, 60, 75, 80, 65, 70, 50, 65, 55, 75, 60, 90, 75, 75, 80,
60, 55, 95, 65, 70, 95, 95,
35, 55, 80, 45, 55, 55, 85, 85, 85, 85, 70, 50, 80, 50, 60,
100, 80, 75, 95, 80, 75, 70,
55, 75, 80, 65, 65, 65), TotalScore = c(78.33, 71.67, 65, 77.5,
75, 76.67, 70, 49.17,

```

```

70.83, 65, 94.17, 86.67, 76.67, 70.83, 55, 70, 83.33, 76.67,
62.5, 77.78, 86.11, 80.56,
86.11, 70, 70, 86.67, 89.17, 86.67, 82.5, 74.17, 77.5, 70,
78.33, 69.17, 63.33, 70.83,
83.33, 71.67, 58.33, 51.67, 57.5, 68.33, 64.17, 81.67, 81.67,
65, 79.17, 51.67, 51.67,
74.17, 58.33, 79.17, 89.17, 89.17, 51.67, 50.83, 67.5, 55.83,
54.17, 64.17, 73.33, 69.17,
85, 77.5, 81.67, 60, 69.17, 63.33, 66.67, 95, 70, 78.33, 85.83,
80.83, 69.17, 74.17, 80,
82.5, 80, 77.5, 74.17, 80.83)), .Names = c("Person", "Date",
"Age", "Sex", "DimBody",
"DimEmotion", "DimSelf", "DimFamily", "DimFriends",
"DimSchool", "TotalScore"), row.names
= c(NA, -82L), class = "data.frame")

```

48.7. Beispiel 7

Die Ergebnisse eines Hochsprungwettbewerbs lauten wie folgt.

- 0 bedeutet, dass die entsprechende Höhe nicht übersprungen wurde
- 1 bedeutet, dass die entsprechende Höhe übersprungen wurde

	1.00	1.10	1.20	1.30	1.50	1.60	1.70	1.90	2.00	2.20
Hans	1	1	1	1	1	1	1	1	1	1
Karola	1	1	1	1	1	1	1	1	1	0
Matthias	1	1	1	1	1	1	1	1	0	0
Stefan	1	1	1	1	1	1	1	0	0	0
Sabine	1	1	1	1	1	1	1	0	0	0
Irma	1	1	1	1	1	1	0	0	0	0
Heike	1	1	1	1	1	1	0	0	0	0
Ralf	1	1	1	1	1	1	0	0	0	0
Rainer	1	1	1	1	1	0	0	0	0	0
Simon	1	1	1	1	1	0	0	0	0	0
Andreas	1	1	1	1	1	0	0	0	0	0
Elke	1	1	1	1	0	0	0	0	0	0
Gabi	1	1	1	1	0	0	0	0	0	0
David	1	1	1	1	0	0	0	0	0	0
Jonas	1	1	1	1	0	0	0	0	0	0
Nicklas	1	1	1	1	0	0	0	0	0	0
Sandra	1	1	1	0	0	0	0	0	0	0
Mario	1	1	1	0	0	0	0	0	0	0
Guido	1	1	1	0	0	0	0	0	0	0
Lisa	1	1	1	0	0	0	0	0	0	0
Peter	1	1	1	0	0	0	0	0	0	0
Justus	1	1	1	0	0	0	0	0	0	0
Josef	1	1	1	0	0	0	0	0	0	0
Brigitte	1	1	1	0	0	0	0	0	0	0
Kevin	1	1	0	0	0	0	0	0	0	0
Marcel	1	1	0	0	0	0	0	0	0	0
Nadine	1	1	0	0	0	0	0	0	0	0
Alex	1	1	0	0	0	0	0	0	0	0
Katharina	1	1	0	0	0	0	0	0	0	0
Daniel	1	1	0	0	0	0	0	0	0	0
Jens	1	1	0	0	0	0	0	0	0	0
Dieter	1	0	0	0	0	0	0	0	0	0
Sebastian	1	0	0	0	0	0	0	0	0	0
Anne	1	0	0	0	0	0	0	0	0	0


```

0, 0, 0, 1, 0, 1, 1, 0, 1), Loch8 = c(1, 1, 1, 0, 0, 1, 0, 0, 1,
0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1,
0, 0, 0, 0, 0, 0, 1, 0, 0), Loch9 = c(1, 1, 0, 0, 0, 0, 0, 0, 0, 1,
0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0,
1, 0, 1, 0, 1, 0, 1, 0 ), Loch10 = c(1, 0, 0, 1, 0, 0, 1, 0, 1, 0,
0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0,
0, 1, 0, 0, 0, 1, 1)), .Names = c("Loch1", "Loch2", "Loch3",
"Loch4", "Loch5", "Loch6", "Loch7", "Loch8", "Loch9",
"Loch10" ), class = "data.frame", row.names = c("Hans", "Karola",
"Matthias", "Stefan", "Sabine", "Irma", "Heike",
"Ralf", "Rainer", "Simon", "Andreas", "Elke", "Gabi", "David",
"Jonas", "Nicklas", "Sandra", "Mario", "Guido",
"Lisa", "Peter", "Justus", "Josef", "Brigitte", "Kevin", "Marcel",
"Nadine", "Alex", "Katharina", "Daniel", "Jens",
"Dieter", "Sebastian", "Anne"))

```

48.9. Beispiel 9

Ein Fragebogen besteht aus 27 Fragen, welche die Antworten "nie / selten / manchmal / oft / immer" zulassen. Die Antworten werden wie folgt codiert:

- 1 = nie
- 2 = selten
- 3 = manchmal
- 4 = oft
- 5 = immer

Der Fragebogen wird nun von 122 Personen ausgefüllt. Da dieser Datensatz relativ groß ist, wird auf eine Darstellung der Tabelle verzichtet. Der Code zum Erzeugen des Datensatzes lautet:

```

bsp9 <- structure(list(Frage1 = c(4, 4, 5, 5, 5, 1, 3, 3, 3, 3, 5, 5,
5, 3, 3, 4, 5, 4, 3, 2, 2, 3, 4, 3, 4, 4, 2,
5, 5, 5, 5, 5, 5, 1, 3, 4, 3, 3, 4, 3, 3, 3, 5, 3, 3, 3, 2, 5, 4,
5, 2, 4, 3, 2, 5, 5, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3,
3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5), Frage2 = c(3, 5, 4,
5, 4, 5, 3, 3, 5, 3, 5, 4, 4, 4, 2, 5, 5, 3, 2,
2, 2, 2, 3, 2, 3, 5, 4, 4, 3, 3, 5, 3, 5, 2, 3, 3, 5, 4, 3, 4, 1, 4,
4, 4, 5, 2, 2, 3, 2, 3, 5, 5, 3, 5, 3, 3, 5, 4,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 5,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5,
5, 5), Frage3 = c(4, 4, 4, 4, 3, 3, 2, 2, 4, 2, 5,
3, 1, 2, 3, 4, 4, 3, 3, 3, 4, 5, 4, 5, 4, 4, 3, 3, 5, 5, 2, 4, 4,
3, 2, 3, 4, 3, 3, 3, 1, 3, 3, 2, 4, 1, 1, 3, 3, 3, 5,
5, 4, 2, 5, 4, 4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5,
5, 5, 5, 5, 5, 5, 5), Frage4 = c(4, 3, 1, 3, 3,
4, 4, 3, 3, 3, 5, 5, 1, 3, 2, 2, 4, 4, 2, 2, 4, 5, 3, 4, 5, 5,
5, 3, 5, 2, 5, 3, 2, 3, 3, 4, 3, 4, 1, 2, 3, 4, 5,
2, 3, 2, 3, 2, 4, 5, 4, 1, 4, 2, 4, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5),
Frage5 = c(5, 1, 2, 5, 2, 4, 4, 1, 1, 4, 5, 4, 3, 5, 3, 5, 1, 1, 4,
5, 5, 4, 2, 4, 3, 1, 2, 1, 3, 4, 2, 1, 1, 4, 1, 1,

```

2, 4, 2, 3, 3, 4, 4, 4, 5, 2, 2, 2, 3, 3, 4, 2, 3, 3, 4, 3, 5, 2, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5,
5, 5, 5, 5, 5, 5, 5), Frage6 = c(5, 3, 1, 4, 5, 5, 4, 3, 3, 4, 5, 5,
4, 4, 3, 2, 4, 5, 3, 3, 3, 3, 3, 4, 3, 4, 5, 4, 2,
5, 5, 4, 5, 4, 2, 2, 5, 3, 3, 3, 1, 4, 4, 4, 5, 3, 2, 3, 1, 2, 5, 4,
3, 5, 4, 3, 4, 5, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5), Frage7 = c(3, 3, 2, 4, 4, 3,
4, 3, 5, 3, 4, 5, 3, 4, 2, 3, 4, 3, 4, 3, 2, 2, 2,
1, 4, 4, 4, 4, 3, 3, 5, 5, 4, 4, 4, 2, 5, 3, 3, 4, 2, 5, 3, 3, 5, 4,
3, 5, 3, 3, 4, 5, 4, 3, 5, 4, 4, 5, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5), Frage8 =
c(4, 5, 5, 5, 4, 5, 4, 2, 5, 4, 5, 5, 4, 4, 4, 4, 5,
3, 3, 1, 3, 4, 4, 3, 3, 5, 5, 5, 5, 4, 5, 5, 5, 3, 3, 4, 5, 3, 3, 4,
2, 5, 3, 4, 5, 3, 3, 4, 3, 4, 5, 4, 4, 4, 5, 4, 5,
4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5,
5, 5), Frage9 = c(5, 5, 5, 5, 5, 5, 5, 2, 5, 3, 5, 4,
4, 4, 2, 4, 5, 4, 2, 2, 3, 4, 4, 4, 3, 5, 5, 5, 5, 3, 4, 4, 5, 4, 5,
4, 5, 4, 4, 4, 2, 5, 4, 4, 5, 2, 3, 2, 4, 2, 4, 4, 4,
5, 2, 3, 5, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5,
5, 5, 5, 5), Frage10 = c(2, 1, 5, 4, 3, 4, 2, 4,
5, 1, 3, 2, 1, 1, 1, 5, 4, 2, 2, 5, 3, 5, 2, 2, 3, 1, 1, 2, 5, 5, 2,
3, 3, 4, 2, 5, 4, 5, 5, 3, 2, 5, 4, 3, 4, 2, 4, 1, 4,
3, 5, 3, 3, 4, 5, 3, 5, 4, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4,
4, 4, 4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5,
5, 5, 5, 5, 5, 5, 5, 5), Frage11 = c(3, 1, 5, 4,
4, 3, 3, 2, 1, 3, 4, 4, 1, 4, 4, 4, 4, 1, 3, 5, 5, 5, 3, 5, 5, 2, 5,
2, 1, 2, 5, 3, 5, 5, 1, 1, 2, 5, 4, 4, 3, 3, 4, 4, 5,
3, 3, 3, 3, 3, 5, 5, 5, 5, 3, 3, 5, 5, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5), Frage12 = c(1,
5, 3, 2, 5, 3, 2, 1, 1, 2, 1, 4, 3, 1, 3, 1, 1, 3, 3, 1, 3, 1, 2, 2,
4, 4, 5, 5, 3, 1, 3, 1, 1, 4, 2, 1, 5, 4, 3, 3, 2, 4,
3, 2, 5, 5, 1, 1, 3, 3, 4, 5, 4, 4, 4, 2, 5, 1, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3,
3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 5), Frage13 = c(2, 4, 2, 3, 3, 3, 3, 2, 2, 4, 4, 1, 2, 1, 2, 4, 4, 2,
3, 2, 3, 3, 1, 2, 3, 5, 4, 4, 2, 1, 3, 5, 3, 4, 1, 5,
3, 3, 3, 1, 2, 3, 1, 4, 5, 3, 1, 4, 4, 2, 4, 2, 2, 5, 2, 3, 5, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5, 5, 5,
5, 5, 5), Frage14 = c(4, 3, 2, 4, 4, 5, 4, 3, 2, 4, 5, 4, 5, 3, 1,
4, 5, 5, 1, 3, 2, 3, 3, 5, 3, 3, 5, 4, 4, 3, 5, 3, 5, 3,
4, 3, 4, 4, 4, 2, 1, 4, 3, 3, 4, 2, 2, 5, 4, 4, 1, 2, 2, 3, 4, 3, 3,
5, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5,
5, 5, 5, 5, 5, 5), Frage15 = c(4, 3, 2, 5, 4, 5, 3, 2, 1, 4, 5, 4,
5, 3, 3, 3, 5, 5, 2, 3, 4, 4, 1, 2, 3, 4, 5, 5, 3, 5, 1,
2, 3, 3, 1, 4, 5, 3, 5, 2, 2, 5, 4, 4, 4, 5, 5, 4, 3, 3, 1, 5, 1, 1,
5, 2, 3, 5, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4,
4, 4, 4, 4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5,
5, 5, 5, 5, 5, 5, 5, 5, 5), Frage16 = c(3, 5, 1, 5, 4, 1, 3, 2, 3,
2, 5, 5, 5, 4, 4, 3, 4, 4, 4, 4, 4, 3, 4, 4, 5, 3,


```

2, 4, 5, 4, 4, 3, 5, 4, 3, 4, 5, 5, 3, 5, 3, 4, 5, 4, 3, 5, 4, 4, 5,
5, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 5, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2,
2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 5, 5, 5, 5,
5, 5, 5, 5, 5, 5)), .Names = c("Frage1", "Frage2", "Frage3",
"Frage4", "Frage5", "Frage6", "Frage7", "Frage8", "Frage9",
"Frage10", "Frage11", "Frage12", "Frage13", "Frage14", "Frage15",
"Frage16", "Frage17", "Frage18", "Frage19", "Frage20",
"Frage21", "Frage22", "Frage23", "Frage24", "Frage25", "Frage26",
"Frage27"), row.names = c(NA, 122L), class = "data.frame",
na.action = structure(123:231, .Names = c("38", "39", "40", "41",
"42", "43", "44", "45", "46", "47", "48", "49", "50", "51",
"52", "53", "54", "55", "56", "57", "58", "59", "60", "61", "62",
"63", "64", "65", "66", "67", "68", "69", "70", "71", "72",
"73", "74", "75", "76", "77", "78", "79", "80", "81", "82", "158",
"159", "160", "161", "162", "163", "164", "165", "166",
"167", "168", "169", "170", "171", "172", "173", "174", "175",
"176", "177", "178", "179", "180", "181", "182", "183", "184",
"185", "186", "187", "188", "189", "190", "191", "192", "193",
"194", "195", "196", "197", "198", "199", "200", "201",
"202", "203", "204", "205", "206", "207", "208", "209", "210",
"211", "212", "213", "214", "215", "216", "217", "218", "219",
"220", "221"), class = "omit"))

```

48.10. Beispiel 10

In einer Untersuchung wurde bei Kinder (Self) und deren Eltern (Proxy) bestimmte Werte über die Kinder abgefragt. Die Kinder schätzten sich also zunächst selber ein, anschließend haben die Eltern ihre Kinder eingeschätzt. Zusätzlich existieren zu jedem Item Normwerte (Norm), die den deutschen Durchschnitt widerspiegeln.

```

bbsp10 <- structure(list(Case = c("1", "1", "1", "1", "1", "1", "2",
"2", "2", "2", "2", "2", "3", "3", "3", "3", "3", "3", "3", "3", "6", "6",
"6", "6", "6", "6", "8", "8", "8", "8", "8", "8", "8", "87", "87", "87",
"87", "87", "87", "97", "97", "97", "97", "97", "97", "97", "3", "3", "3",
"3", "3", "3", "3", "3", "3", "3", "3", "3", "3", "4", "4", "4", "4",
"4", "4", "4", "5", "5", "5", "5", "5", "5", "5", "5", "6", "6", "6", "6", "6",
"6", "6", "8", "8", "8", "8", "8", "8", "8", "8", "87", "87", "87", "87", "87",
"87", "97", "97", "97", "97", "97", "97", "97", "Norm", "Norm", "Norm",
"Norm", "Norm", "Norm", "Norm", "Norm", "Norm", "Norm", "Norm", "Norm",
"Norm"), Item = structure(c(2L, 3L, 4L, 5L, 6L, 7L, 2L, 3L, 4L, 5L,
6L, 7L, 2L, 3L, 4L, 5L, 6L, 7L, 2L, 3L, 4L, 5L, 6L, 7L, 2L, 3L, 4L,
5L, 6L, 7L, 2L, 3L, 4L, 5L, 6L, 7L, 2L, 3L, 4L, 5L, 6L, 7L, 2L, 3L,
4L, 5L, 6L, 7L, 2L, 3L, 4L, 5L, 6L, 7L, 2L, 3L, 4L, 5L, 6L, 7L, 2L,
3L, 4L, 5L, 6L, 7L, 2L, 3L, 4L, 5L, 6L, 7L, 2L, 3L, 4L, 5L, 6L,
7L, 2L, 3L, 4L, 5L, 6L, 7L), .Label = c("Item", "Body", "Emotion",
"Family", "Friends", "School", "Index"), class = "factor"),
Score = c(73.201612015845, 59.5103286592648, 51.211152156733,
57.8338409160427, 62.8406196084782, 59.8493862271105,
64.2956371038732, 43.2100676781528, 45.2472845564316,
53.2323487558682, 48.0853674487552, 59.8493862271105,
47.0820037411972, 50.611904867224, 38.3277278491481,
33.7624512920086, 42.9406488823006, 44.4784623272101,
49.6265680017606, 37.86511552469, 46.5337891235228,
57.8338409160427, 34.8124918251294, 42.2729427242036,
55.6028554137324, 44.7951224546969, 53.2511808274062,
49.7945287460269, 51.0750344123124, 48.2897549745108,
64.2956371038732, 46.5276241871987, 49.4651816728235,
46.928570478635, 51.0750344123124, 48.2897549745108,
64.2956371038732, 43.2100676781528, 47.9305655106504,

```


49. Graphikbeispiele

Die hier verwendeten Grafiken stammen von Benutzer Thire¹ und sollen die Themen des Kapitels GNU R: Diagramme² veranschaulichen.

49.1. Graphikbeispiel 1

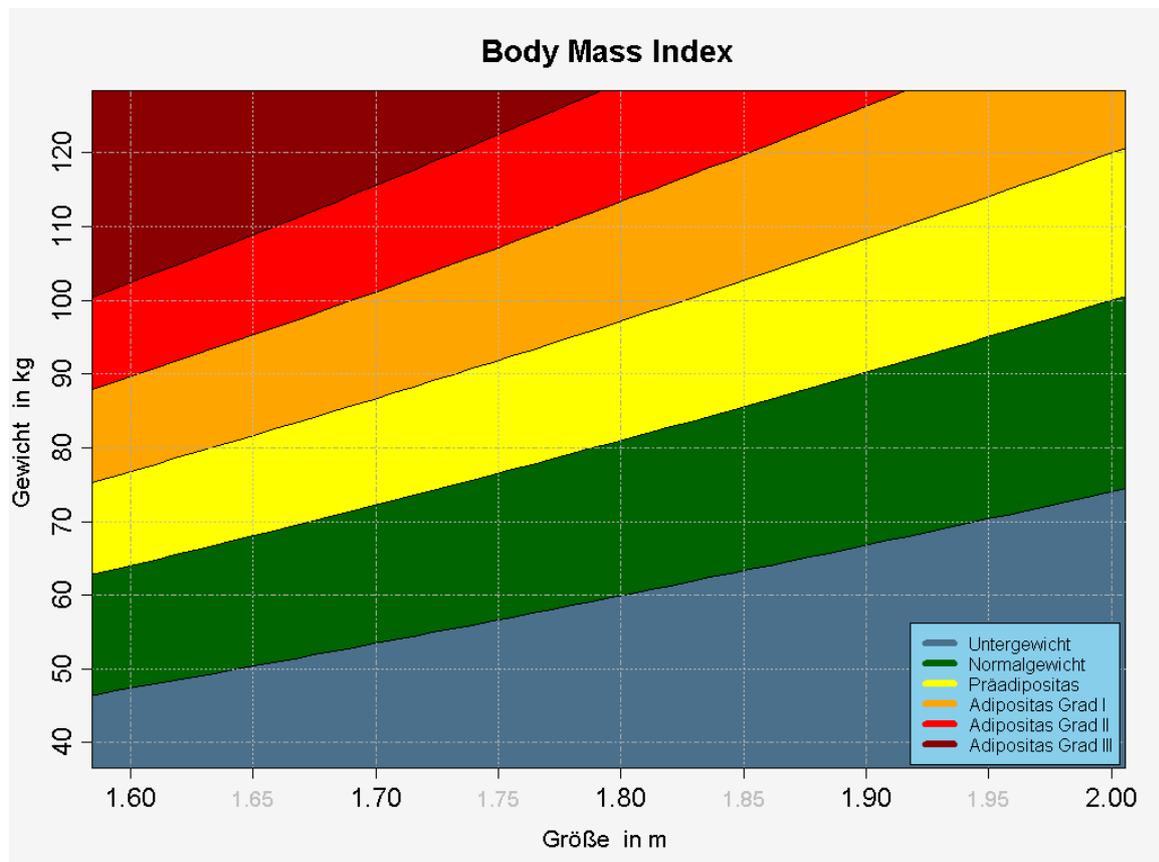


Abb. 8

Quelltext:

- http://de.wikipedia.org/wiki/Benutzer%3ATHire%2FBilder%23Mathematische_Grafiken
- Kapitel 26 auf Seite 75

```

class=c3(0,18.5,25,30,35,40,100)
col
s=c4("skyblue4","darkgreen","yellow","orange","red","darkred","black")
gr=seq5(1.4,2.2,length=100)

bmi.k<-function6(groesse,konstant) {
  return(groesse2*konstant)
}

png7(filename="BodyMassIndex.png", width=1024, height=768,
  pointsize=12)
par8(bg="whitesmoke")
plot9(gr,bmi.k(gr,18),type="n",xlim=c(1.60,1.99),ylim=c(40,125)
),xaxt="n",yaxt="n",cex.axis=1.4,cex.lab=1.3,cex.main=1.7,xlab="Größe
[in m]",ylab="Gewicht [in kg]", main="Body Mass Index")
polygon10
(c11(gr,rev12(gr)),c(bmi.k(gr,class[1]),rev13(bmi.k(gr,class[2]))),col=cols[1])
polygon
(c(gr,rev14(gr)),c(bmi.k(gr,class[2]),rev15(bmi.k(gr,class[3]))),col=cols[2])
polygon(c(g
r,rev(gr)),c(bmi.k(gr,class[3]),rev(bmi.k(gr,class[4]))),col=cols[3])
polygon(c(g
r,rev(gr)),c(bmi.k(gr,class[4]),rev(bmi.k(gr,class[5]))),col=cols[4])
polygon(c(g
r,rev(gr)),c(bmi.k(gr,class[5]),rev(bmi.k(gr,class[6]))),col=cols[5])
polygon(c(g
r,rev(gr)),c(bmi.k(gr,class[6]),rev(bmi.k(gr,class[7]))),col=cols[6])
polygon(c(g
r,rev(gr)),c(bmi.k(gr,class[7]),rev(bmi.k(gr,class[8]))),col=cols[7])

box16()
grid17(lty="dotdash",col="darkgrey")
abline18
(v=seq(1.65,1.95,by=0.1),h=seq(50,110,by=20),lty="dotted",col="grey")

legend19(x="bottomright",inset=0.00
5,legend=c("Untergewicht","Normalgewicht","Präadipositas","Adipositas
Grad I","Adipositas Grad II","Adipositas Grad
III"),col=cols,lwd=6,bg="skyblue")
axis20(1,at=format(seq(1.60,2,by=0.
1),nsmall=2),labels=format(seq(1.60,2,by=0.1),nsmall=2),cex.axis=1.5)
axis(1,at=seq(1.65,2,by=0.1),cex.axis=1.2,col.axis="grey")
axis(2,at=seq(40,120,by=10),cex.axis=1.5)

dev.off21()

```

49.2. Graphikbeispiel 2

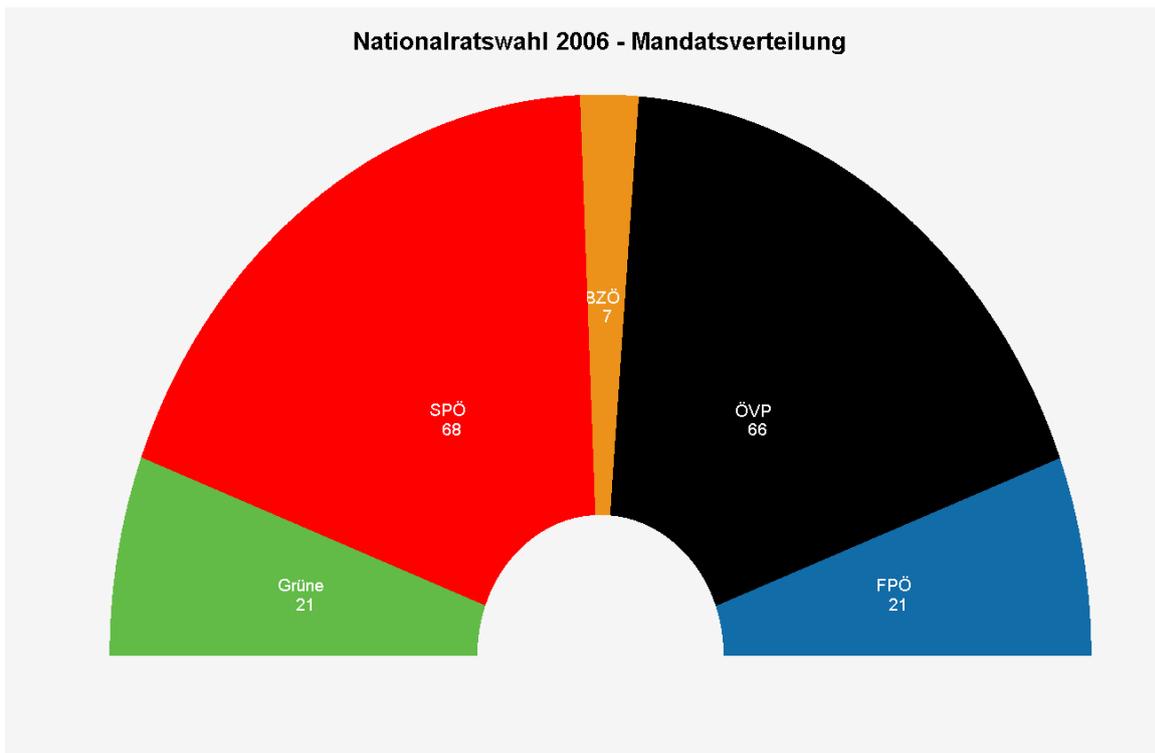


Abb. 9

Quelltext:

```
parliament<-function22(x, labels=names(x), edges=5000,
col=rainbow(length(x)), rad.in=0.25, ...) {
  plot23(c24(-1,1), c(0,1), type="n", axes=FALSE, xlab="", ylab="", ...)
  y=c(0,pi*cumsum(x)/sum(x))
  ang=seq25(0,pi,length=edges)
  for (aktiv in 1:(length(y)-1)) {
    poly=ang[ y[aktiv]<=ang & ang<=y[aktiv+1] ]
    polygon26(c(cos(poly),0),
c(sin(poly),0), col=col[aktiv],border=col[aktiv])
    text27( (cos28(y[ak
tiv])+cos(y[aktiv+1])+rad.in*cos(y[aktiv])+rad.in*cos(y[aktiv+1]))/4,
(sin(y[ak
tiv])+sin(y[aktiv+1])+rad.in*sin(y[aktiv])+rad.in*sin(y[aktiv+1]))/4,
col="white", labels=paste(labels[aktiv], "\n" ,x[aktiv]), cex=1.2,
adj=c(0.5,0.5))
  }
  polygon29(rad.in*cos(ang), rad.in*sin(ang), col=par("bg"),border=par("bg"))
}

x<-c(21,66,7,68,21)
names30(x)<-c("FPÖ", "ÖVP", "BZÖ", "SPÖ", "Grüne")

png31(filename="Mandate2006.png", width=1300, height=850, pointsize=12)
par32(bg="whitesmoke")
parliament(x, col=c("#126ca8", "black", "#ec921a", "red",
"#61bb46"), main="Nationalratswahl 2006 - Mandatsverteilung",
```

```
cex.main=1.7)
dev.off33()
```

49.3. Graphikbeispiel 3

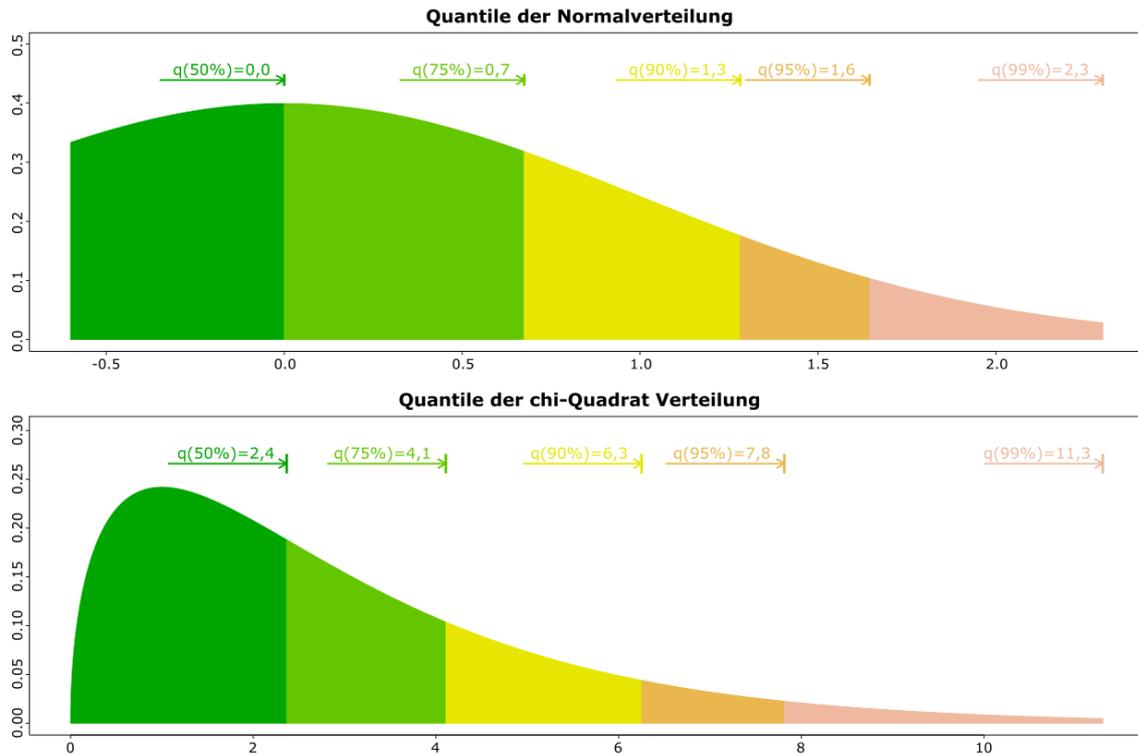


Abb. 10

Quelltext:

```
qs=c(0,0.5,0.75,0.9,0.95,0.99)
cs=heat.colors(length(qs))

qplot<-function34(qs,x,pdf,cdf,pfl) {
  plot35(x,pdf,type="n",ylim=range(pdf,0)*1.25,xlab="",ylab="")
  for (q in 1:(length36(qs)-1)) {
    xd= x[(cdf>=qs[q])&(cdf<=qs[q+1])]
    yd= pdf[(cdf>=qs[q])&(cdf<=qs[q+1])]
    polygon37(c(xd,xd[length(xd)]),c(yd,0),col=cs[q],border=cs[q])
    lines38(c(xd[length(xd)]),xd
[length(xd)],c(max(pdf)*1.1-0.01,max(pdf)*1.1+0.01),col=cs[q],lwd=3)
    arrows39(xd[length
(xd)]-pfl,max(pdf)*1.1,xd[length(xd)],max(pdf)*1.1,lwd=2,length=0.15)
    text40(xd[length(xd)],max(pdf)*1.15,paste("q(",format(qs[q+1]*100),"%
)=",format(xd[length(xd)],digi
ts=1,nsmall=1,decimal.mark=","),sep=""),col=cs[q],cex=1.5,adj=c(1,0))
  }
}
```

```

png41(filename = "Quantile_graph.png", width=1500, height=1000,
pointsize=12)
par42(mfrow=c(2,1),bg="whitesmoke")

## normalverteilung
x=seq(-0.6,2.3,by=0.0001)
pdf=dnorm43(x=x)
cdf=pnorm44(q=x)

qplot(qs,x,pdf,cdf,0.35)
title45(main="Quantile der Normalverteilung")

## chi-quadrat verteilung
df=3
x=seq46(0,11.3,length=10000)
pdf=dchisq47(x=x,df=df)
cdf=pchisq48(q=x,df=df)

qplot(qs,x,pdf,cdf,1.3)
title49(main="Quantile der chi-Quadrat Verteilung")
dev.off50()

```

49.4. Graphikbeispiel 4

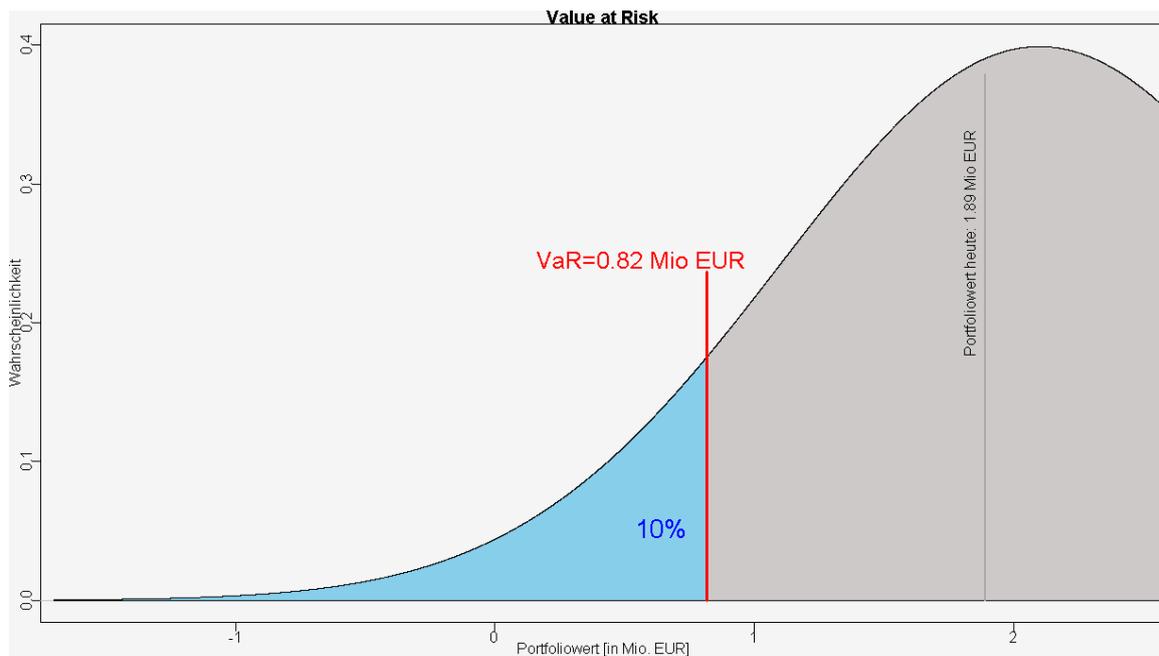


Abb. 11

Quelltext:

```

#value at risk graph
quant=0.10

```

```

min=-1.7
max=2.6
mean=2.1
s=seq51(min,max,length=10000)
d=dnorm52(s,mean=mean)
q=qnorm53(quant,mean=mean)
sq=c54(s[s<q],q,min)
dq=c(d[s<q],0,0)

png55(filename = "VaR_graph.png", width=1300, height=800, pointsize=12)
par56(bg="whitesmoke")
plot57(s,d,type="n",xlab="Portfoliowert [in Mio.
EUR]",ylab="Wahrscheinlichkeit",main="Value at
Risk",xlim=range(s)*0.935)
abline58(h=0,col="grey")
polygon59(x=c(s,max,min),y=c(d,0,0),col="snow3")

polygon60(x=sq,y=dq,col="skyblue")

text61(x=(q-min)*0.93+min,y=dnorm(q,mean=me
an)*0.3,label=paste(format(100*quant),"%",sep=""),col="blue",cex=1.7)

lines62(x=c(q,q),y=c(0,dnorm(q,mean=mean)*1.35),col="red",lwd=3)
text63(x=
(q-min)*0.9+min,y=dnorm(q,mean=mean)*1.4,label=paste("VaR=",format(q,
digits=2)," Mio EUR",sep=""),col="red",cex=1.6)

lines64(x=c(m
ean,mean)*0.9,y=c(0,dnorm(mean,mean=mean)*0.95),col="darkgrey",lwd=2)
text65(x=
mean*0.87,y=dnorm(mean*0.9,mean=mean)*0.66,label=paste("Portfoliowert
heute: ",mean*0.9," Mio EUR",sep=""),col="black",srt=90)

dev.off66()

```

49.5. Graphikbeispiel 5

Each kid: first measure self and proxy

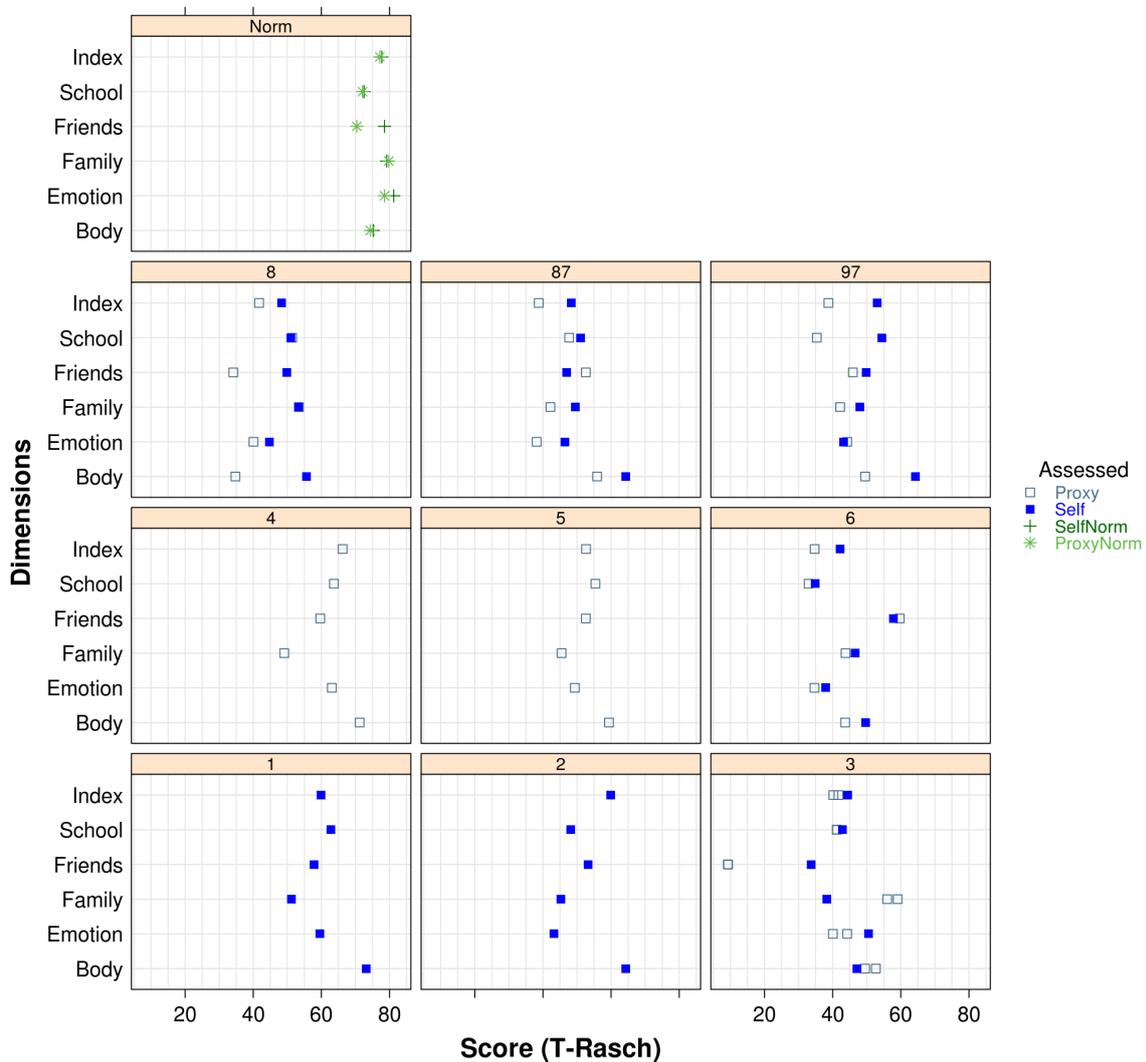


Abb. 12

Verwendet werden die Daten aus Beispiel 10⁶⁷.

Der Befehl zum Erzeugen der Grafik lautet:

```
library68(lattice) # Lädt das lattice-Grafikpaket
```

```
colors <- c("skyblue4", "blue", "darkgreen", "#61bb46") # Farben
für Werte und Labels
```

⁶⁷ Kapitel 48.10 auf Seite 184

```

key.list <- list(space = "right",      # Legende Ausrichtung
               title="Assessed", cex.title=1.2, # Legendenüberschrift und
               Größe
               points=list(pch=c(0,15,3,8), col=colors, cex=1), #
               Legendensymbole
               text=list(levels(bsp10$Proxy), cex=1, font=1),      # Legende
               mit Schriftgröße und -art
               col=colors)      # LegendenSchriftfarbe
grafik <- dotplot69(Item ~ Score | Case , groups=Proxy, data=bsp10,
               layout = c(3,4), # Diagramme werden in 3 Spalten und 4 Reihen
               aufgeteilt
               key = key.list, # Verwende die Daten aus dem Objekt
               "key.list"
               xlab = list(label="Score (T-Rasch)", cex=1.5, font=2), #
               Label für X-Achse, Schriftgröße, fett
               ylab = list(label="Dimensions", cex=1.5, font=2),      #
               Label für Y-Achse, Schriftgröße, fett,
               main= list(label="Each kid: first measure self and proxy",
               cex=2,5), # Überschrift
               #sub = list(label="Kinder: alle Erst- und Zweiterhebungen",
               cex=1, font=3),
               scales = list(cex=1.2, ces=2), # Achsenbeschriftung
               (Schriftgröße)
               between = list(x=0.5, y=0.5), # Abstand zwischen den
               Diagrammen
               cex = 1,      # Größe der Symbole
               pch=c(0,15,3,8), col=colors) # Symbolart der Werte und
               deren Farbe
update70(grafik,
               panel = function(...){panel.grid(h=-1,v=-20) # Gitter
               hinzufügen
               panel.dotplot(...) })

```

```
rm71(colors,key.list,grafik)
```

50. siehe auch

- Programmierbeispiele¹

¹ Kapitel 54 auf Seite 233

51. Befehle-Index

Auf dieser Seite sollen alle R-Befehle alphabetisch gesammelt werden.

51.1. A

A¹ - B² - C³ - D⁴ - E⁵ - F⁶ - G⁷ - H⁸ - I⁹ - J¹⁰ - K¹¹ - L¹² - M¹³ - N¹⁴ - O¹⁵ - P¹⁶ - Q¹⁷ - R¹⁸ - S¹⁹ - T²⁰ - U²¹ - V²² - W²³ - X²⁴ - Y²⁵ - Z²⁶

Befehl

`acf(x)`²⁷

`all(x)`²⁸

`any(x)`²⁹

Kurzbeschreibung

`acf(x)` errechnet die Autokorrelation der Daten mit verschiedenen Intervallen und erstellt eine entsprechende Grafik.

`all(x)` fragt ab, ob das Objekt `x` nur den Wahrheitswert TRUE enthält.

`any(x)` fragt ab, ob das Objekt `x` mindestens einmal den Wahrheitswert TRUE enthält.

-
- 1 Kapitel 51.1 auf Seite 197
 - 2 Kapitel 51.2 auf Seite 198
 - 3 Kapitel 51.3 auf Seite 199
 - 4 Kapitel 51.4 auf Seite 201
 - 5 Kapitel 51.5 auf Seite 202
 - 6 Kapitel 51.6 auf Seite 203
 - 7 Kapitel 51.7 auf Seite 204
 - 8 Kapitel 51.8 auf Seite 205
 - 9 Kapitel 51.9 auf Seite 206
 - 10 Kapitel 51.10 auf Seite 208
 - 11 Kapitel 51.11 auf Seite 209
 - 12 Kapitel 51.12 auf Seite 210
 - 13 Kapitel 51.13 auf Seite 211
 - 14 Kapitel 51.14 auf Seite 213
 - 15 Kapitel 51.15 auf Seite 214
 - 16 Kapitel 51.16 auf Seite 215
 - 17 Kapitel 51.17 auf Seite 217
 - 18 Kapitel 51.18 auf Seite 218
 - 19 Kapitel 51.19 auf Seite 219
 - 20 Kapitel 51.20 auf Seite 222
 - 21 Kapitel 51.21 auf Seite 223
 - 22 Kapitel 51.22 auf Seite 224
 - 23 Kapitel 51.23 auf Seite 225
 - 24 Kapitel 51.24 auf Seite 226
 - 25 Kapitel 51.25 auf Seite 227
 - 26 Kapitel 51.26 auf Seite 228
 - 27 <http://de.wikibooks.org/wiki/GNU%20R%3A%20acf>
 - 28 <http://de.wikibooks.org/wiki/GNU%20R%3A%20all>
 - 29 <http://de.wikibooks.org/wiki/GNU%20R%3A%20any>

<code>apply(x)</code> ³⁰	<code>apply(x, col, func)</code> wendet auf die Spalten <code>col</code> eines Arrays <code>x</code> die Funktion <code>func</code> an.
<code>apropos()</code> ³¹	<code>apropos()</code> sucht einen Ausdruck in der Liste der Funktionen und Vektorvariablen.
<code>as.factor(x)</code> ³²	<code>as.factor(x)</code> wandelt das Objekt <code>x</code> in einen Faktor um.
<code>as.matrix(x)</code> ³³	<code>as.matrix(x)</code> wandelt das Objekt <code>x</code> in eine Matrix um. Parametrische Schätzungen benötigen zur weiteren Berechnung eine Matrix.
<code>attach(x)</code> ³⁴	<code>attach(x)</code> nimmt das Objekt <code>x</code> in den Suchpfad auf.
<code>available.packages()</code> ³⁵	<code>available.packages()</code> zeigt alle verfügbaren Zusatzpakete (eines Servers) an.

51.2. B

A³⁶ - B³⁷ - C³⁸ - D³⁹ - E⁴⁰ - F⁴¹ - G⁴² - H⁴³ - I⁴⁴ - J⁴⁵ - K⁴⁶ - L⁴⁷ - M⁴⁸ - N⁴⁹ - O⁵⁰ - P⁵¹ - Q⁵² - R⁵³ - S⁵⁴ - T⁵⁵ - U⁵⁶ - V⁵⁷ - W⁵⁸ - X⁵⁹ - Y⁶⁰ - Z⁶¹

Befehl	Kurzbeschreibung
30	http://de.wikibooks.org/wiki/GNU%20R%3A%20apply
31	http://de.wikibooks.org/wiki/GNU%20R%3A%20apropos
32	http://de.wikibooks.org/wiki/GNU%20R%3A%20as.factor
33	http://de.wikibooks.org/wiki/GNU%20R%3A%20as.matrix
34	http://de.wikibooks.org/wiki/GNU%20R%3A%20attach
35	http://de.wikibooks.org/wiki/GNU%20R%3A%20available.packages
36	Kapitel 51.1 auf Seite 197
37	Kapitel 51.2 auf Seite 198
38	Kapitel 51.3 auf Seite 199
39	Kapitel 51.4 auf Seite 201
40	Kapitel 51.5 auf Seite 202
41	Kapitel 51.6 auf Seite 203
42	Kapitel 51.7 auf Seite 204
43	Kapitel 51.8 auf Seite 205
44	Kapitel 51.9 auf Seite 206
45	Kapitel 51.10 auf Seite 208
46	Kapitel 51.11 auf Seite 209
47	Kapitel 51.12 auf Seite 210
48	Kapitel 51.13 auf Seite 211
49	Kapitel 51.14 auf Seite 213
50	Kapitel 51.15 auf Seite 214
51	Kapitel 51.16 auf Seite 215
52	Kapitel 51.17 auf Seite 217
53	Kapitel 51.18 auf Seite 218
54	Kapitel 51.19 auf Seite 219
55	Kapitel 51.20 auf Seite 222
56	Kapitel 51.21 auf Seite 223
57	Kapitel 51.22 auf Seite 224
58	Kapitel 51.23 auf Seite 225
59	Kapitel 51.24 auf Seite 226
60	Kapitel 51.25 auf Seite 227
61	Kapitel 51.26 auf Seite 228

<code>barplot()</code> ⁶²	<code>barplot()</code> erstellt Säulendiagramme.
<code>blockrand()</code> ⁶³	<code>blockrand()</code> (aus dem gleichnamigen Zusatzpaket) erzeugt Block-Randomisierungen für Clinical Trials.
<code>boxplot()</code> ⁶⁴	<code>boxplot()</code> erstellt ein Boxplot der Daten.

51.3. C

A⁶⁵ - B⁶⁶ - C⁶⁷ - D⁶⁸ - E⁶⁹ - F⁷⁰ - G⁷¹ - H⁷² - I⁷³ - J⁷⁴ - K⁷⁵ - L⁷⁶ - M⁷⁷ - N⁷⁸ - O⁷⁹ - P⁸⁰ - Q⁸¹ - R⁸² - S⁸³ - T⁸⁴ - U⁸⁵ - V⁸⁶ - W⁸⁷ - X⁸⁸ - Y⁸⁹ - Z⁹⁰

Befehl

`c`⁹¹

`cat()`⁹²

Kurzbeschreibung

`concatenate` (abgekürzt mit `c`) bedeutet "aneinanderhängen". So lassen sich Zahlen- oder Buchstabenreihen einem Vektor zuordnen, z.B.:

```
mein.vector <- c(2, 4, 5, 6)
```

mit `cat(a, b, c)` werden die Elemente `a`, `b` und `c` zusammengeführt und in Zeichenketten konvertiert.

62 <http://de.wikibooks.org/wiki/GNU%20R%3A%20barplot>

63 <http://de.wikibooks.org/wiki/GNU%20R%3A%20blockrand>

64 <http://de.wikibooks.org/wiki/GNU%20R%3A%20boxplot>

65 Kapitel 51.1 auf Seite 197

66 Kapitel 51.2 auf Seite 198

67 Kapitel 51.3 auf Seite 199

68 Kapitel 51.4 auf Seite 201

69 Kapitel 51.5 auf Seite 202

70 Kapitel 51.6 auf Seite 203

71 Kapitel 51.7 auf Seite 204

72 Kapitel 51.8 auf Seite 205

73 Kapitel 51.9 auf Seite 206

74 Kapitel 51.10 auf Seite 208

75 Kapitel 51.11 auf Seite 209

76 Kapitel 51.12 auf Seite 210

77 Kapitel 51.13 auf Seite 211

78 Kapitel 51.14 auf Seite 213

79 Kapitel 51.15 auf Seite 214

80 Kapitel 51.16 auf Seite 215

81 Kapitel 51.17 auf Seite 217

82 Kapitel 51.18 auf Seite 218

83 Kapitel 51.19 auf Seite 219

84 Kapitel 51.20 auf Seite 222

85 Kapitel 51.21 auf Seite 223

86 Kapitel 51.22 auf Seite 224

87 Kapitel 51.23 auf Seite 225

88 Kapitel 51.24 auf Seite 226

89 Kapitel 51.25 auf Seite 227

90 Kapitel 51.26 auf Seite 228

91 <http://de.wikibooks.org/wiki/GNU%20R%3A%20c>

92 <http://de.wikibooks.org/wiki/GNU%20R%3A%20cat>

<code>cbind()</code> ⁹³	mit <code>cbind(a, b, c)</code> lassen sich die Vektoren <code>a</code> , <code>b</code> und <code>c</code> spaltenweise zu einer Matrix zusammenführen (siehe auch <code>rbind()</code> ⁹⁴)
<code>ceiling(x)</code> ⁹⁵	<code>ceiling(x)</code> rundet alle Werte im Objekt <code>x</code> auf ganze Zahlen auf (siehe auch <code>floor()</code> ⁹⁶ zum abrunden).
<code>chisq.test(x)</code> ⁹⁷	<code>chisq.test(x)</code> führt den Chi-Quadrat-Test auf das Objekt <code>x</code> aus.
<code>chooseCRANmirror()</code> ⁹⁸	Mit <code>chooseCRANmirror()</code> wird ein CRAN-Mirror für die bestehende Session ausgewählt.
<code>class()</code> ⁹⁹	<code>class()</code>
<code>colnames()</code> ¹⁰⁰	<code>colnames()</code> weist den Spalten einer Matrix einen "Namen" (Label) zu. Siehe <code>rownames()</code> ¹⁰¹ für Reihen
<code>colors()</code> ¹⁰²	<code>colors()</code> zeigt eine Übersicht aller Farben an, die derzeit verfügbar sind.
<code>comment()</code> ¹⁰³	<code>comment()</code> weist einem Objekt einen Kommentar (Label) zu
<code>cor(a, b)</code> ¹⁰⁴	<code>cor(a, b)</code> errechnet Korrelationskoeffizienten zwischen <code>a</code> und <code>b</code> .
<code>cor.test(a, b)</code> ¹⁰⁵	<code>cor.test(a, b)</code> errechnet Korrelationskoeffizienten samt Signifikanzniveau und Konfidenzintervallen.

93 <http://de.wikibooks.org/wiki/GNU%20R%3A%20cbind>

94 Kapitel 51.18 auf Seite 218

95 <http://de.wikibooks.org/wiki/GNU%20R%3A%20ceiling>

96 <http://de.wikibooks.org/wiki/GNU%20R%3A%20floor>

97 <http://de.wikibooks.org/wiki/GNU%20R%3A%20chisq.test>

98 <http://de.wikibooks.org/wiki/GNU%20R%3A%20chooseCRANmirror>

99 <http://de.wikibooks.org/wiki/GNU%20R%3A%20class>

100 <http://de.wikibooks.org/wiki/GNU%20R%3A%20colnames>

101 <http://de.wikibooks.org/wiki/GNU%20R%3A%20rownames>

102 <http://de.wikibooks.org/wiki/GNU%20R%3A%20colors>

103 <http://de.wikibooks.org/wiki/GNU%20R%3A%20comment>

104 <http://de.wikibooks.org/wiki/GNU%20R%3A%20cor>

105 <http://de.wikibooks.org/wiki/GNU%20R%3A%20cor.test>

51.4. D

A¹⁰⁶ - B¹⁰⁷ - C¹⁰⁸ - D¹⁰⁹ - E¹¹⁰ - F¹¹¹ - G¹¹² - H¹¹³ - I¹¹⁴ - J¹¹⁵ - K¹¹⁶ - L¹¹⁷ - M¹¹⁸ - N¹¹⁹ - O¹²⁰ - P¹²¹ - Q¹²² - R¹²³ - S¹²⁴ - T¹²⁵ - U¹²⁶ - V¹²⁷ - W¹²⁸ - X¹²⁹ - Y¹³⁰ - Z¹³¹

Befehl

`data()`¹³²

`data.entry(x)`¹³³

`data.frame(a, b, c)`¹³⁴

`deparse(x)`¹³⁵

`detach(x)`¹³⁶

`devSVG()`¹³⁷

Kurzbeschreibung

`data()` aktiviert einen Datensatz (z.B. aus einem Zusatzpaket) bzw. listet verfügbare Datensätze auf.

`data.entry(x)` öffnet den Daten-Editor und ermöglicht so die Bearbeitung der Daten (hinzufügen/löschen/editieren) des Objekts `x`.

`data.frame(a, b, c)` führt die Objekte `a`, `b`, `c` zu einem Datenframe zusammen.

`deparse(x)` gibt einen dem Ausdruck oder Objekt `x` entsprechenden String zurück. Damit lassen sich z.B. die internen Repräsentationen von Datenobjekten ausgeben und bearbeiten/weiterverwenden.

`detach(x)` löscht das Datenframe `x` aus dem Suchpfad bzw. deaktiviert eingebundene Zusatzpakete.

`devSVG()` speichert eine Grafik als SVG-Datei ab. Dazu ist jedoch das Zusatzpaket `RSvgDevice` erforderlich.

106 Kapitel 51.1 auf Seite 197

107 Kapitel 51.2 auf Seite 198

108 Kapitel 51.3 auf Seite 199

109 Kapitel 51.4 auf Seite 201

110 Kapitel 51.5 auf Seite 202

111 Kapitel 51.6 auf Seite 203

112 Kapitel 51.7 auf Seite 204

113 Kapitel 51.8 auf Seite 205

114 Kapitel 51.9 auf Seite 206

115 Kapitel 51.10 auf Seite 208

116 Kapitel 51.11 auf Seite 209

117 Kapitel 51.12 auf Seite 210

118 Kapitel 51.13 auf Seite 211

119 Kapitel 51.14 auf Seite 213

120 Kapitel 51.15 auf Seite 214

121 Kapitel 51.16 auf Seite 215

122 Kapitel 51.17 auf Seite 217

123 Kapitel 51.18 auf Seite 218

124 Kapitel 51.19 auf Seite 219

125 Kapitel 51.20 auf Seite 222

126 Kapitel 51.21 auf Seite 223

127 Kapitel 51.22 auf Seite 224

128 Kapitel 51.23 auf Seite 225

129 Kapitel 51.24 auf Seite 226

130 Kapitel 51.25 auf Seite 227

131 Kapitel 51.26 auf Seite 228

132 <http://de.wikibooks.org/wiki/GNU%20R%3A%20data>

133 <http://de.wikibooks.org/wiki/GNU%20R%3A%20data.entry>

134 <http://de.wikibooks.org/wiki/GNU%20R%3A%20data.frame>

135 <http://de.wikibooks.org/wiki/GNU%20R%3A%20deparse>

136 <http://de.wikibooks.org/wiki/GNU%20R%3A%20detach>

137 http://de.wikibooks.org/wiki/GNU_R%3A%20devSVG

<code>dim(x)</code> ¹³⁸	Mit <code>dim(x)</code> wird die Anzahl der Dimensionen (Anzahl der Spalten und Anzahl der Zeilen) der Matrix <code>x</code> ermittelt.
<code>dimnames(x)</code> ¹³⁹	Mit <code>dimnames(x)</code> werden den Dimensionen (Spalten/Zeilen) der Matrix <code>x</code> "Namen" (Labels) zugeordnet.
<code>dir()</code> ¹⁴⁰	<code>dir()</code> ist ein Alias zu <code>list.files()</code> und zeigt die Dateien des aktuellen Arbeitsverzeichnisses an.
<code>dotchart(x)</code> ¹⁴¹	<code>dotchart(x)</code> erstellt Punktediagramme.
<code>duplicated(data)</code> ¹⁴²	<code>duplicated(data)</code> überprüft den Datensatz <code>data</code> auf doppelte bzw. mehrfach eingetragene Fälle (Duplikate) und gibt TRUE/FALSE-Werte für jeden Eintrag zurück (siehe auch <code>unique()</code> ¹⁴³).

51.5. E

A¹⁴⁴ - B¹⁴⁵ - C¹⁴⁶ - D¹⁴⁷ - E¹⁴⁸ - F¹⁴⁹ - G¹⁵⁰ - H¹⁵¹ - I¹⁵² - J¹⁵³ - K¹⁵⁴ - L¹⁵⁵ - M¹⁵⁶ - N¹⁵⁷ - O¹⁵⁸ - P¹⁵⁹ - Q¹⁶⁰ - R¹⁶¹ - S¹⁶² - T¹⁶³ - U¹⁶⁴ - V¹⁶⁵ - W¹⁶⁶ - X¹⁶⁷ - Y¹⁶⁸ - Z¹⁶⁹

Befehl	Kurzbeschreibung
--------	------------------

138	http://de.wikibooks.org/wiki/GNU%20R%3A%20dim
139	http://de.wikibooks.org/wiki/GNU%20R%3A%20dimnames
140	http://de.wikibooks.org/wiki/GNU%20R%3A%20dir
141	http://de.wikibooks.org/wiki/GNU%20R%3A%20dotchart
142	http://de.wikibooks.org/wiki/GNU%20R%3A%20duplicated
143	http://de.wikibooks.org/wiki/GNU%20R%3A%20unique
144	Kapitel 51.1 auf Seite 197
145	Kapitel 51.2 auf Seite 198
146	Kapitel 51.3 auf Seite 199
147	Kapitel 51.4 auf Seite 201
148	Kapitel 51.5 auf Seite 202
149	Kapitel 51.6 auf Seite 203
150	Kapitel 51.7 auf Seite 204
151	Kapitel 51.8 auf Seite 205
152	Kapitel 51.9 auf Seite 206
153	Kapitel 51.10 auf Seite 208
154	Kapitel 51.11 auf Seite 209
155	Kapitel 51.12 auf Seite 210
156	Kapitel 51.13 auf Seite 211
157	Kapitel 51.14 auf Seite 213
158	Kapitel 51.15 auf Seite 214
159	Kapitel 51.16 auf Seite 215
160	Kapitel 51.17 auf Seite 217
161	Kapitel 51.18 auf Seite 218
162	Kapitel 51.19 auf Seite 219
163	Kapitel 51.20 auf Seite 222
164	Kapitel 51.21 auf Seite 223
165	Kapitel 51.22 auf Seite 224
166	Kapitel 51.23 auf Seite 225
167	Kapitel 51.24 auf Seite 226
168	Kapitel 51.25 auf Seite 227
169	Kapitel 51.26 auf Seite 228

<code>edit(x)</code> ¹⁷⁰	<code>edit(x)</code> ruft das Objekt <code>x</code> in einem externen Editor zur Bearbeitung auf (siehe auch <code>fix(x)</code> ¹⁷¹).
<code>exp(x)</code> ¹⁷²	<code>exp(x)</code> gibt die Exponentialfunktion für die in <code>x</code> enthaltenen Werte.

51.6. F

A¹⁷³ - B¹⁷⁴ - C¹⁷⁵ - D¹⁷⁶ - E¹⁷⁷ - F¹⁷⁸ - G¹⁷⁹ - H¹⁸⁰ - I¹⁸¹ - J¹⁸² - K¹⁸³ - L¹⁸⁴ - M¹⁸⁵ - N¹⁸⁶ - O¹⁸⁷ - P¹⁸⁸ - Q¹⁸⁹ - R¹⁹⁰ - S¹⁹¹ - T¹⁹² - U¹⁹³ - V¹⁹⁴ - W¹⁹⁵ - X¹⁹⁶ - Y¹⁹⁷ - Z¹⁹⁸

Befehl	Kurzbeschreibung
<code>factanal(x)</code> ¹⁹⁹	<code>factanal(x)</code> führt eine Haupt- und Maximum-Likelihood-Faktorenanalyse auf das Objekt <code>x</code> durch (siehe auch <code>princomp(x)</code> ²⁰⁰).
<code>factor()</code> ²⁰¹	<code>factor()</code> erzeugt ein Faktor-Objekt.
<code>fisher.test(x)</code> ²⁰²	<code>fisher.test(x)</code> führt den exakten Test nach Fisher für das Objekt <code>x</code> durch.

¹⁷⁰ <http://de.wikibooks.org/wiki/GNU%20R%3A%20edit>

¹⁷¹ <http://de.wikibooks.org/wiki/GNU%20R%3A%20fix>

¹⁷² <http://de.wikibooks.org/wiki/GNU%20R%3A%20exp>

¹⁷³ Kapitel 51.1 auf Seite 197

¹⁷⁴ Kapitel 51.2 auf Seite 198

¹⁷⁵ Kapitel 51.3 auf Seite 199

¹⁷⁶ Kapitel 51.4 auf Seite 201

¹⁷⁷ Kapitel 51.5 auf Seite 202

¹⁷⁸ Kapitel 51.6 auf Seite 203

¹⁷⁹ Kapitel 51.7 auf Seite 204

¹⁸⁰ Kapitel 51.8 auf Seite 205

¹⁸¹ Kapitel 51.9 auf Seite 206

¹⁸² Kapitel 51.10 auf Seite 208

¹⁸³ Kapitel 51.11 auf Seite 209

¹⁸⁴ Kapitel 51.12 auf Seite 210

¹⁸⁵ Kapitel 51.13 auf Seite 211

¹⁸⁶ Kapitel 51.14 auf Seite 213

¹⁸⁷ Kapitel 51.15 auf Seite 214

¹⁸⁸ Kapitel 51.16 auf Seite 215

¹⁸⁹ Kapitel 51.17 auf Seite 217

¹⁹⁰ Kapitel 51.18 auf Seite 218

¹⁹¹ Kapitel 51.19 auf Seite 219

¹⁹² Kapitel 51.20 auf Seite 222

¹⁹³ Kapitel 51.21 auf Seite 223

¹⁹⁴ Kapitel 51.22 auf Seite 224

¹⁹⁵ Kapitel 51.23 auf Seite 225

¹⁹⁶ Kapitel 51.24 auf Seite 226

¹⁹⁷ Kapitel 51.25 auf Seite 227

¹⁹⁸ Kapitel 51.26 auf Seite 228

¹⁹⁹ <http://de.wikibooks.org/wiki/GNU%20R%3A%20factanal>

²⁰⁰ <http://de.wikibooks.org/wiki/GNU%20R%3A%20princomp>

²⁰¹ <http://de.wikibooks.org/wiki/GNU%20R%3A%20factor>

²⁰² <http://de.wikibooks.org/wiki/GNU%20R%3A%20fisher.test>

<code>fix(x)</code> ²⁰³	<code>fix(x)</code> ruft das Objekt <code>x</code> in einem graphischen Editor auf. So lassen sich Daten korrigieren, editieren und erweitern (siehe auch <code>edit(x)</code> ²⁰⁴ für externen Editor).
<code>floor(x)</code> ²⁰⁵	<code>floor(x)</code> rundet alle Werte im Objekt <code>x</code> auf ganze Zahlen ab (siehe auch <code>ceiling(x)</code> ²⁰⁶ zum aufrunden).
<code>forestplot</code> ²⁰⁷	<code>forestplot</code> (aus dem Zusatzpaket "rmeta") erzeugt ein so genanntes "Forest Plot" zusammen mit einer Texttabelle. (siehe auch <code>metaplot</code> ²⁰⁸)

51.7. G

A²⁰⁹ - B²¹⁰ - C²¹¹ - D²¹² - E²¹³ - F²¹⁴ - G²¹⁵ - H²¹⁶ - I²¹⁷ - J²¹⁸ - K²¹⁹ - L²²⁰ - M²²¹ - N²²² - O²²³ - P²²⁴ - Q²²⁵ - R²²⁶ - S²²⁷ - T²²⁸ - U²²⁹ - V²³⁰ - W²³¹ - X²³² - Y²³³ - Z²³⁴

Befehl	Kurzbeschreibung
<code>gcd(x, y)</code> ²³⁵	<code>gcd(x, y)</code> (aus dem <code>schoolmath</code> -Paket) berechnet den größten gemeinsamen Teiler der Zahlen <code>x</code> und <code>y</code> .

-
- 203 <http://de.wikibooks.org/wiki/GNU%20R%3A%20fix>
 - 204 <http://de.wikibooks.org/wiki/GNU%20R%3A%20edit>
 - 205 <http://de.wikibooks.org/wiki/GNU%20R%3A%20floor>
 - 206 <http://de.wikibooks.org/wiki/GNU%20R%3A%20ceiling>
 - 207 http://de.wikibooks.org/wiki/GNU_R%3A%20forestplot
 - 208 http://de.wikibooks.org/wiki/GNU_R%3A%20metaplot
 - 209 Kapitel 51.1 auf Seite 197
 - 210 Kapitel 51.2 auf Seite 198
 - 211 Kapitel 51.3 auf Seite 199
 - 212 Kapitel 51.4 auf Seite 201
 - 213 Kapitel 51.5 auf Seite 202
 - 214 Kapitel 51.6 auf Seite 203
 - 215 Kapitel 51.7 auf Seite 204
 - 216 Kapitel 51.8 auf Seite 205
 - 217 Kapitel 51.9 auf Seite 206
 - 218 Kapitel 51.10 auf Seite 208
 - 219 Kapitel 51.11 auf Seite 209
 - 220 Kapitel 51.12 auf Seite 210
 - 221 Kapitel 51.13 auf Seite 211
 - 222 Kapitel 51.14 auf Seite 213
 - 223 Kapitel 51.15 auf Seite 214
 - 224 Kapitel 51.16 auf Seite 215
 - 225 Kapitel 51.17 auf Seite 217
 - 226 Kapitel 51.18 auf Seite 218
 - 227 Kapitel 51.19 auf Seite 219
 - 228 Kapitel 51.20 auf Seite 222
 - 229 Kapitel 51.21 auf Seite 223
 - 230 Kapitel 51.22 auf Seite 224
 - 231 Kapitel 51.23 auf Seite 225
 - 232 Kapitel 51.24 auf Seite 226
 - 233 Kapitel 51.25 auf Seite 227
 - 234 Kapitel 51.26 auf Seite 228
 - 235 <http://de.wikibooks.org/wiki/GNU%20R%3A%20gcd>

<code>getwd()</code> ²³⁶	<code>getwd()</code> gibt das aktuelle Arbeitsverzeichnis aus.
<code>ggplot()</code> ²³⁷	<code>ggplot()</code> aus dem Paket ggplot2 stellt eine Alternative zur <code>Standard-plot()</code> -Funktion dar.
<code>glm()</code> ²³⁸	<code>glm()</code> ist die Hauptfunktion für <i>generalized linear models</i> (inkl. logistische Regressionsanalyse).
<code>graphics.off()</code> ²³⁹	Mit <code>graphics.off()</code> werden alle Graphikfenster geschlossen.
<code>grep(TEXT, x)</code> ²⁴⁰	<code>grep(TEXT, x)</code> untersucht, ob der Text <code>TEXT</code> im Objekt <code>x</code> enthalten ist.
<code>gsub</code> ²⁴¹	Mit <code>gsub("Muster", "Ersetzung", Objekt)</code> lassen sich Zeichenketten (mehrfach) in einem Objekt ersetzen.

51.8. H

A²⁴² - B²⁴³ - C²⁴⁴ - D²⁴⁵ - E²⁴⁶ - F²⁴⁷ - G²⁴⁸ - H²⁴⁹ - I²⁵⁰ - J²⁵¹ - K²⁵² - L²⁵³ - M²⁵⁴ - N²⁵⁵ - O²⁵⁶ - P²⁵⁷ - Q²⁵⁸ - R²⁵⁹ - S²⁶⁰ - T²⁶¹ - U²⁶² - V²⁶³ - W²⁶⁴ - X²⁶⁵ - Y²⁶⁶ - Z²⁶⁷

Befehl	Kurzbeschreibung
--------	------------------

236	http://de.wikibooks.org/wiki/GNU%20R%3A%20getwd
237	http://de.wikibooks.org/wiki/GNU%20R%3A%20ggplot
238	http://de.wikibooks.org/wiki/GNU%20R%3A%20glm
239	http://de.wikibooks.org/wiki/GNU%20R%3A%20graphics.off
240	http://de.wikibooks.org/wiki/GNU%20R%3A%20grep
241	http://de.wikibooks.org/wiki/GNU%20R%3A%20sub
242	Kapitel 51.1 auf Seite 197
243	Kapitel 51.2 auf Seite 198
244	Kapitel 51.3 auf Seite 199
245	Kapitel 51.4 auf Seite 201
246	Kapitel 51.5 auf Seite 202
247	Kapitel 51.6 auf Seite 203
248	Kapitel 51.7 auf Seite 204
249	Kapitel 51.8 auf Seite 205
250	Kapitel 51.9 auf Seite 206
251	Kapitel 51.10 auf Seite 208
252	Kapitel 51.11 auf Seite 209
253	Kapitel 51.12 auf Seite 210
254	Kapitel 51.13 auf Seite 211
255	Kapitel 51.14 auf Seite 213
256	Kapitel 51.15 auf Seite 214
257	Kapitel 51.16 auf Seite 215
258	Kapitel 51.17 auf Seite 217
259	Kapitel 51.18 auf Seite 218
260	Kapitel 51.19 auf Seite 219
261	Kapitel 51.20 auf Seite 222
262	Kapitel 51.21 auf Seite 223
263	Kapitel 51.22 auf Seite 224
264	Kapitel 51.23 auf Seite 225
265	Kapitel 51.24 auf Seite 226
266	Kapitel 51.25 auf Seite 227
267	Kapitel 51.26 auf Seite 228

<code>head()</code> ²⁶⁸	Mit dem Befehl <code>head(DATA, n)</code> lassen sich die ersten <code>n</code> Datenreihen des Datensatzes <code>DATA</code> anzeigen. Der Befehl <code>tail()</code> ²⁶⁹ liefert hingegen die letzten <code>n</code> Datenreihen.
<code>help()</code> ²⁷⁰ / <code>?²⁷¹</code>	<code>help()</code> bzw. <code>?</code> zeigt die R-interne Dokumentation zu einem Befehl an.
<code>help.start()</code> ²⁷²	<code>help.start()</code> öffnet den Standardbrowser mit den Links zu den R-Manualen.
<code>hist()</code> ²⁷³	<code>hist()</code> erstellt ein Histogramm der Daten.
<code>history()</code> ²⁷⁴	<code>history()</code> gibt die letzten Befehle in ein separates Fenster aus.

51.9. I

A²⁷⁵ - B²⁷⁶ - C²⁷⁷ - D²⁷⁸ - E²⁷⁹ - F²⁸⁰ - G²⁸¹ - H²⁸² - I²⁸³ - J²⁸⁴ - K²⁸⁵ - L²⁸⁶ - M²⁸⁷ - N²⁸⁸ - O²⁸⁹ - P²⁹⁰ - Q²⁹¹ - R²⁹² - S²⁹³ - T²⁹⁴ - U²⁹⁵ - V²⁹⁶ - W²⁹⁷ - X²⁹⁸ - Y²⁹⁹ - Z³⁰⁰

Befehl	Kurzbeschreibung
--------	------------------

268	http://de.wikibooks.org/wiki/GNU%20R%3A%20head
269	http://de.wikibooks.org/wiki/GNU%20R%3A%20tail
270	http://de.wikibooks.org/wiki/GNU%20R%3A%20help
271	http://de.wikibooks.org/wiki/GNU%20R%3A%20help
272	http://de.wikibooks.org/wiki/GNU%20R%3A%20help.start
273	http://de.wikibooks.org/wiki/GNU%20R%3A%20hist
274	http://de.wikibooks.org/wiki/GNU%20R%3A%20history
275	Kapitel 51.1 auf Seite 197
276	Kapitel 51.2 auf Seite 198
277	Kapitel 51.3 auf Seite 199
278	Kapitel 51.4 auf Seite 201
279	Kapitel 51.5 auf Seite 202
280	Kapitel 51.6 auf Seite 203
281	Kapitel 51.7 auf Seite 204
282	Kapitel 51.8 auf Seite 205
283	Kapitel 51.9 auf Seite 206
284	Kapitel 51.10 auf Seite 208
285	Kapitel 51.11 auf Seite 209
286	Kapitel 51.12 auf Seite 210
287	Kapitel 51.13 auf Seite 211
288	Kapitel 51.14 auf Seite 213
289	Kapitel 51.15 auf Seite 214
290	Kapitel 51.16 auf Seite 215
291	Kapitel 51.17 auf Seite 217
292	Kapitel 51.18 auf Seite 218
293	Kapitel 51.19 auf Seite 219
294	Kapitel 51.20 auf Seite 222
295	Kapitel 51.21 auf Seite 223
296	Kapitel 51.22 auf Seite 224
297	Kapitel 51.23 auf Seite 225
298	Kapitel 51.24 auf Seite 226
299	Kapitel 51.25 auf Seite 227
300	Kapitel 51.26 auf Seite 228

<code>identical(x, y)</code> ³⁰¹	<code>identical(x, y)</code> überprüft, ob die Objekte <code>x</code> und <code>y</code> identisch sind. Als Antwort wird TRUE/FALSE ausgegeben.
<code>install.packages(mypkg)</code> ³⁰²	<code>install.packages(mypkg)</code> installiert das Zusatzpaket <code>mypkg</code>
<code>intersect(x, y)</code> ³⁰³	<code>intersect(x, y)</code> gibt die gemeinsamen Elemente (Schnittmenge) beider Vektoren <code>x</code> und <code>y</code> als neuen Vektor aus.
<code>is.character(x)</code> ³⁰⁴	<code>is.character(x)</code> überprüft, ob das Objekt <code>x</code> vom Typ <code>character</code> ist. Als Antwort wird TRUE/FALSE ausgegeben.
<code>is.element(x, y)</code> ³⁰⁵	<code>is.element(x, y)</code> überprüft, ob das Objekt <code>x</code> im Objekt <code>y</code> enthalten ist.
<code>is.factor(x)</code> ³⁰⁶	<code>is.factor(x)</code> überprüft, ob das Objekt <code>x</code> vom Typ <code>factor</code> ist. Als Antwort wird TRUE/FALSE ausgegeben.
<code>is.logical(x)</code> ³⁰⁷	<code>is.logical(x)</code> fragt die <code>logical</code> -Eigenschaft des Objekts <code>x</code> ab. Als Antwort wird TRUE/FALSE ausgegeben.
<code>is.matrix(x)</code> ³⁰⁸	<code>is.matrix(x)</code> überprüft, ob das Objekt <code>x</code> eine Matrix ist. Als Antwort wird TRUE/FALSE ausgegeben.
<code>is.na(x)</code> ³⁰⁹	<code>is.na(x)</code> überprüft, ob ein Objekt <code>x</code> ein Element NA ist. Als Antwort wird TRUE/FALSE ausgegeben.
<code>is.numeric(x)</code> ³¹⁰	<code>is.numeric(x)</code> überprüft, ob das Objekt <code>x</code> numerisch ist. Als Antwort wird TRUE/FALSE ausgegeben.
<code>is.prim(x)</code> ³¹¹	<code>is.prim(x)</code> (aus dem <code>schoolmath</code> -Paket) überprüft, ob die Zahl <code>x</code> eine Primzahl ist.
<code>is.vector(x)</code> ³¹²	<code>is.vector(x)</code> überprüft, ob das Objekt <code>x</code> ein Vektor ist. Als Antwort wird TRUE/FALSE ausgegeben.
<code>ISOdate()</code> ³¹³	<code>ISOdate(Jahr, Monat, Tag, Stunde, Minute, Sekunde)</code> kann zur einfachen Definition eines Datum-Objekts verwendet werden.

301 <http://de.wikibooks.org/wiki/GNU%20R%3A%20identical>

302 <http://de.wikibooks.org/wiki/GNU%20R%3A%20install.packages>

303 <http://de.wikibooks.org/wiki/GNU%20R%3A%20intersect>

304 <http://de.wikibooks.org/wiki/GNU%20R%3A%20is.character>

305 <http://de.wikibooks.org/wiki/GNU%20R%3A%20is.element>

306 <http://de.wikibooks.org/wiki/GNU%20R%3A%20is.factor>

307 <http://de.wikibooks.org/wiki/GNU%20R%3A%20is.logical>

308 <http://de.wikibooks.org/wiki/GNU%20R%3A%20is.matrix>

309 <http://de.wikibooks.org/wiki/GNU%20R%3A%20is.na>

310 <http://de.wikibooks.org/wiki/GNU%20R%3A%20is.numeric>

311 <http://de.wikibooks.org/wiki/GNU%20R%3A%20is.prim>

312 <http://de.wikibooks.org/wiki/GNU%20R%3A%20is.vector>

313 <http://de.wikibooks.org/wiki/GNU%20R%3A%20ISOdate>

51.10. J

A³¹⁴ - B³¹⁵ - C³¹⁶ - D³¹⁷ - E³¹⁸ - F³¹⁹ - G³²⁰ - H³²¹ - I³²² - J³²³ - K³²⁴ - L³²⁵ - M³²⁶ - N³²⁷ - O³²⁸ -
P³²⁹ - Q³³⁰ - R³³¹ - S³³² - T³³³ - U³³⁴ - V³³⁵ - W³³⁶ - X³³⁷ - Y³³⁸ - Z³³⁹

Befehl	Kurzbeschreibung
<code>jpeg()</code> ³⁴⁰	Mit Hilfe von <code>jpeg()</code> können Graphiken in eine .jpeg-Datei gespeichert werden. Für das png-Format steht der Befehl <code>png()</code> ³⁴¹ zur Verfügung.

-
- 314 Kapitel 51.1 auf Seite 197
 - 315 Kapitel 51.2 auf Seite 198
 - 316 Kapitel 51.3 auf Seite 199
 - 317 Kapitel 51.4 auf Seite 201
 - 318 Kapitel 51.5 auf Seite 202
 - 319 Kapitel 51.6 auf Seite 203
 - 320 Kapitel 51.7 auf Seite 204
 - 321 Kapitel 51.8 auf Seite 205
 - 322 Kapitel 51.9 auf Seite 206
 - 323 Kapitel 51.10 auf Seite 208
 - 324 Kapitel 51.11 auf Seite 209
 - 325 Kapitel 51.12 auf Seite 210
 - 326 Kapitel 51.13 auf Seite 211
 - 327 Kapitel 51.14 auf Seite 213
 - 328 Kapitel 51.15 auf Seite 214
 - 329 Kapitel 51.16 auf Seite 215
 - 330 Kapitel 51.17 auf Seite 217
 - 331 Kapitel 51.18 auf Seite 218
 - 332 Kapitel 51.19 auf Seite 219
 - 333 Kapitel 51.20 auf Seite 222
 - 334 Kapitel 51.21 auf Seite 223
 - 335 Kapitel 51.22 auf Seite 224
 - 336 Kapitel 51.23 auf Seite 225
 - 337 Kapitel 51.24 auf Seite 226
 - 338 Kapitel 51.25 auf Seite 227
 - 339 Kapitel 51.26 auf Seite 228
 - 340 <http://de.wikibooks.org/wiki/GNU%20R%3A%20jpeg>
 - 341 <http://de.wikibooks.org/wiki/GNU%20R%3A%20png>

51.11. K

A³⁴² - B³⁴³ - C³⁴⁴ - D³⁴⁵ - E³⁴⁶ - F³⁴⁷ - G³⁴⁸ - H³⁴⁹ - I³⁵⁰ - J³⁵¹ - K³⁵² - L³⁵³ - M³⁵⁴ - N³⁵⁵ - O³⁵⁶ - P³⁵⁷ - Q³⁵⁸ - R³⁵⁹ - S³⁶⁰ - T³⁶¹ - U³⁶² - V³⁶³ - W³⁶⁴ - X³⁶⁵ - Y³⁶⁶ - Z³⁶⁷

Befehl

`ks.test()`³⁶⁸

Kurzbeschreibung

`ks.test()` führt den Kolmogorov-Smirnov-Test durch.

342 Kapitel 51.1 auf Seite 197
 343 Kapitel 51.2 auf Seite 198
 344 Kapitel 51.3 auf Seite 199
 345 Kapitel 51.4 auf Seite 201
 346 Kapitel 51.5 auf Seite 202
 347 Kapitel 51.6 auf Seite 203
 348 Kapitel 51.7 auf Seite 204
 349 Kapitel 51.8 auf Seite 205
 350 Kapitel 51.9 auf Seite 206
 351 Kapitel 51.10 auf Seite 208
 352 Kapitel 51.11 auf Seite 209
 353 Kapitel 51.12 auf Seite 210
 354 Kapitel 51.13 auf Seite 211
 355 Kapitel 51.14 auf Seite 213
 356 Kapitel 51.15 auf Seite 214
 357 Kapitel 51.16 auf Seite 215
 358 Kapitel 51.17 auf Seite 217
 359 Kapitel 51.18 auf Seite 218
 360 Kapitel 51.19 auf Seite 219
 361 Kapitel 51.20 auf Seite 222
 362 Kapitel 51.21 auf Seite 223
 363 Kapitel 51.22 auf Seite 224
 364 Kapitel 51.23 auf Seite 225
 365 Kapitel 51.24 auf Seite 226
 366 Kapitel 51.25 auf Seite 227
 367 Kapitel 51.26 auf Seite 228
 368 <http://de.wikibooks.org/wiki/GNU%20R%3A%20ks.test>

51.12. L

A³⁶⁹ - B³⁷⁰ - C³⁷¹ - D³⁷² - E³⁷³ - F³⁷⁴ - G³⁷⁵ - H³⁷⁶ - I³⁷⁷ - J³⁷⁸ - K³⁷⁹ - L³⁸⁰ - M³⁸¹ - N³⁸² - O³⁸³ - P³⁸⁴ - Q³⁸⁵ - R³⁸⁶ - S³⁸⁷ - T³⁸⁸ - U³⁸⁹ - V³⁹⁰ - W³⁹¹ - X³⁹² - Y³⁹³ - Z³⁹⁴

Befehl

`lapply(x, func)`³⁹⁵

`length(x)`³⁹⁶

`level(x)`³⁹⁷

`library()`³⁹⁸

`lines(x, y)`³⁹⁹

Kurzbeschreibung

`lapply` ermöglicht es die selbe Funktion `func` auf eine Liste `x` von Objekten anzuwenden. `lapply` ist damit eine (i.d.R. bessere) Alternative zur Benutzung von Schleifen/Loop-Konstruktionen.

Mit `length(x)` wird die Anzahl der Elemente von `x` ermittelt.

Mit `level(x)` werden die Level `x` zurückgegeben. Dabei kann `x` ein Vector oder eine z.B. eine Spalte eines `data.frame` sein.

`library()` gibt eine Liste aller installierten Pakete (`packages`) aus. Mit `library(package)` wird ein Bibliothek eingebunden.

`lines(x, y)` verbindet mit den Vektoren `x` und `y` übergebene Punktkoordinaten mit Linien.

369 Kapitel 51.1 auf Seite 197

370 Kapitel 51.2 auf Seite 198

371 Kapitel 51.3 auf Seite 199

372 Kapitel 51.4 auf Seite 201

373 Kapitel 51.5 auf Seite 202

374 Kapitel 51.6 auf Seite 203

375 Kapitel 51.7 auf Seite 204

376 Kapitel 51.8 auf Seite 205

377 Kapitel 51.9 auf Seite 206

378 Kapitel 51.10 auf Seite 208

379 Kapitel 51.11 auf Seite 209

380 Kapitel 51.12 auf Seite 210

381 Kapitel 51.13 auf Seite 211

382 Kapitel 51.14 auf Seite 213

383 Kapitel 51.15 auf Seite 214

384 Kapitel 51.16 auf Seite 215

385 Kapitel 51.17 auf Seite 217

386 Kapitel 51.18 auf Seite 218

387 Kapitel 51.19 auf Seite 219

388 Kapitel 51.20 auf Seite 222

389 Kapitel 51.21 auf Seite 223

390 Kapitel 51.22 auf Seite 224

391 Kapitel 51.23 auf Seite 225

392 Kapitel 51.24 auf Seite 226

393 Kapitel 51.25 auf Seite 227

394 Kapitel 51.26 auf Seite 228

395 <http://de.wikibooks.org/wiki/GNU%20R%3A%20lapply>

396 <http://de.wikibooks.org/wiki/GNU%20R%3A%20length>

397 <http://de.wikibooks.org/wiki/GNU%20R%3A%20level>

398 <http://de.wikibooks.org/wiki/GNU%20R%3A%20library>

399 <http://de.wikibooks.org/wiki/GNU%20R%3A%20lines>

<code>list.files()</code> ⁴⁰⁰	<code>list.files()</code> zeigt die Dateien des aktuellen Arbeitsverzeichnisses an. Alternativ kann auch <code>dir()</code> verwendet werden.
<code>load(file)</code> ⁴⁰¹	<code>load(file)</code> lädt eine gespeicherte R-Sitzung in den Arbeitsspeicher.
<code>loadedNamespaces()</code> ⁴⁰²	<code>loadedNamespaces()</code> liefert eine Liste der aktuellen Namensräume.
<code>ls()</code> ⁴⁰³	<code>ls()</code> nennt alle Variablen in der aktuellen R-Sitzung (genau so wie <code>objects()</code> ⁴⁰⁴).
<code>ls.str()</code> ⁴⁰⁵	<code>ls.str()</code> listet alle Objekte der aktuellen R-Sitzung mit ihrer Struktur. Es handelt sich also um eine Kombination von <code>ls()</code> ⁴⁰⁶ und <code>str()</code> ⁴⁰⁷ .

51.13. M

A⁴⁰⁸ - B⁴⁰⁹ - C⁴¹⁰ - D⁴¹¹ - E⁴¹² - F⁴¹³ - G⁴¹⁴ - H⁴¹⁵ - I⁴¹⁶ - J⁴¹⁷ - K⁴¹⁸ - L⁴¹⁹ - M⁴²⁰ - N⁴²¹ - O⁴²² - P⁴²³ - Q⁴²⁴ - R⁴²⁵ - S⁴²⁶ - T⁴²⁷ - U⁴²⁸ - V⁴²⁹ - W⁴³⁰ - X⁴³¹ - Y⁴³² - Z⁴³³

400 <http://de.wikibooks.org/wiki/GNU%20R%3A%20list.files>

401 <http://de.wikibooks.org/wiki/GNU%20R%3A%20load>

402 <http://de.wikibooks.org/wiki/GNU%20R%3A%20loadedNamespaces>

403 <http://de.wikibooks.org/wiki/GNU%20R%3A%20ls>

404 <http://de.wikibooks.org/wiki/GNU%20R%3A%20objects>

405 <http://de.wikibooks.org/wiki/GNU%20R%3A%20ls.str>

406 <http://de.wikibooks.org/wiki/GNU%20R%3A%20ls>

407 <http://de.wikibooks.org/wiki/GNU%20R%3A%20str>

408 Kapitel 51.1 auf Seite 197

409 Kapitel 51.2 auf Seite 198

410 Kapitel 51.3 auf Seite 199

411 Kapitel 51.4 auf Seite 201

412 Kapitel 51.5 auf Seite 202

413 Kapitel 51.6 auf Seite 203

414 Kapitel 51.7 auf Seite 204

415 Kapitel 51.8 auf Seite 205

416 Kapitel 51.9 auf Seite 206

417 Kapitel 51.10 auf Seite 208

418 Kapitel 51.11 auf Seite 209

419 Kapitel 51.12 auf Seite 210

420 Kapitel 51.13 auf Seite 211

421 Kapitel 51.14 auf Seite 213

422 Kapitel 51.15 auf Seite 214

423 Kapitel 51.16 auf Seite 215

424 Kapitel 51.17 auf Seite 217

425 Kapitel 51.18 auf Seite 218

426 Kapitel 51.19 auf Seite 219

427 Kapitel 51.20 auf Seite 222

428 Kapitel 51.21 auf Seite 223

429 Kapitel 51.22 auf Seite 224

430 Kapitel 51.23 auf Seite 225

431 Kapitel 51.24 auf Seite 226

432 Kapitel 51.25 auf Seite 227

433 Kapitel 51.26 auf Seite 228

Befehl	Kurzbeschreibung
<code>map()</code> ⁴³⁴	<code>map()</code> (aus den Paketen "maps" und "mapdata") erstellt Karten von Ländern, Kontinenten und der Welt.
<code>match()</code> ⁴³⁵	<code>match(v, s)</code> durchsucht einen Vektor B, nach Übereinstimmungen in einem Suchvektor s.
<code>matrix(a, b, c)</code> ⁴³⁶	<code>matrix(a, b, c)</code> führt die Objekte a, b, c zu einer Matrix zusammen.
<code>mean(x)</code> ⁴³⁷	<code>mean(x)</code> bestimmt das arithmetische Mittel eines Zahlenvektors x.
<code>median(x)</code> ⁴³⁸	<code>median(x)</code> bestimmt den Median eines Zahlenvektors x.
<code>metaplot</code> ⁴³⁹	<code>metaplot</code> erzeugt ein so genanntes "Forest Plot" (Meta-Analyse-Plot), welches im Rahmen von Metaanalysen gängig ist. (siehe auch <code>forestplot</code> ⁴⁴⁰)
<code>mode(x)</code> ⁴⁴¹	<code>mode(x)</code> überprüft, von welchem Typ das Objekt x ist. Als Antwort wird entsprechend <code>character</code> , <code>numeric</code> ausgegeben.

434 <http://de.wikibooks.org/wiki/GNU%20R%3A%20map>

435 <http://de.wikibooks.org/wiki/GNU%20R%3A%20match>

436 <http://de.wikibooks.org/wiki/GNU%20R%3A%20matrix>

437 <http://de.wikibooks.org/wiki/GNU%20R%3A%20mean>

438 <http://de.wikibooks.org/wiki/GNU%20R%3A%20median>

439 <http://de.wikibooks.org/wiki/GNU%20R%3A%20metaplot>

440 http://de.wikibooks.org/wiki/GNU_R%3A%20forestplot

441 <http://de.wikibooks.org/wiki/GNU%20R%3A%20mode>

51.14. N

A⁴⁴² - B⁴⁴³ - C⁴⁴⁴ - D⁴⁴⁵ - E⁴⁴⁶ - F⁴⁴⁷ - G⁴⁴⁸ - H⁴⁴⁹ - I⁴⁵⁰ - J⁴⁵¹ - K⁴⁵² - L⁴⁵³ - M⁴⁵⁴ - N⁴⁵⁵ - O⁴⁵⁶ - P⁴⁵⁷ - Q⁴⁵⁸ - R⁴⁵⁹ - S⁴⁶⁰ - T⁴⁶¹ - U⁴⁶² - V⁴⁶³ - W⁴⁶⁴ - X⁴⁶⁵ - Y⁴⁶⁶ - Z⁴⁶⁷

Befehl	Kurzbeschreibung
<code>na.omit(x)</code> ⁴⁶⁸	<code>na.omit(x)</code> filtert (für Berechnungen) die fehlenden Werte (missing data, NA) aus dem Objekt <code>x</code> heraus.
<code>names(x)</code> ⁴⁶⁹	<code>names(x)</code> gibt die Variablennamen des Data-Frames <code>x</code> an.
<code>nchar(x)</code> ⁴⁷⁰	<code>nchar(x)</code> gibt Anzahl der Buchstaben in <code>x</code> an.
<code>ncol(x)</code> ⁴⁷¹	<code>ncol(x)</code> gibt Anzahl der Spalten in <code>x</code> an.
<code>nrow(x)</code> ⁴⁷²	<code>nrow(x)</code> gibt Anzahl der Zeilen in <code>x</code> an.

442	Kapitel 51.1 auf Seite 197
443	Kapitel 51.2 auf Seite 198
444	Kapitel 51.3 auf Seite 199
445	Kapitel 51.4 auf Seite 201
446	Kapitel 51.5 auf Seite 202
447	Kapitel 51.6 auf Seite 203
448	Kapitel 51.7 auf Seite 204
449	Kapitel 51.8 auf Seite 205
450	Kapitel 51.9 auf Seite 206
451	Kapitel 51.10 auf Seite 208
452	Kapitel 51.11 auf Seite 209
453	Kapitel 51.12 auf Seite 210
454	Kapitel 51.13 auf Seite 211
455	Kapitel 51.14 auf Seite 213
456	Kapitel 51.15 auf Seite 214
457	Kapitel 51.16 auf Seite 215
458	Kapitel 51.17 auf Seite 217
459	Kapitel 51.18 auf Seite 218
460	Kapitel 51.19 auf Seite 219
461	Kapitel 51.20 auf Seite 222
462	Kapitel 51.21 auf Seite 223
463	Kapitel 51.22 auf Seite 224
464	Kapitel 51.23 auf Seite 225
465	Kapitel 51.24 auf Seite 226
466	Kapitel 51.25 auf Seite 227
467	Kapitel 51.26 auf Seite 228
468	http://de.wikibooks.org/wiki/GNU%20R%3A%20na.omit
469	http://de.wikibooks.org/wiki/GNU%20R%3A%20names
470	http://de.wikibooks.org/wiki/GNU%20R%3A%20nchar
471	http://de.wikibooks.org/wiki/GNU%20R%3A%20ncol
472	http://de.wikibooks.org/wiki/GNU%20R%3A%20nrow

51.15. O

A⁴⁷³ - B⁴⁷⁴ - C⁴⁷⁵ - D⁴⁷⁶ - E⁴⁷⁷ - F⁴⁷⁸ - G⁴⁷⁹ - H⁴⁸⁰ - I⁴⁸¹ - J⁴⁸² - K⁴⁸³ - L⁴⁸⁴ - M⁴⁸⁵ - N⁴⁸⁶ - O⁴⁸⁷ - P⁴⁸⁸ - Q⁴⁸⁹ - R⁴⁹⁰ - S⁴⁹¹ - T⁴⁹² - U⁴⁹³ - V⁴⁹⁴ - W⁴⁹⁵ - X⁴⁹⁶ - Y⁴⁹⁷ - Z⁴⁹⁸

Befehl	Kurzbeschreibung
<code>objects()</code> ⁴⁹⁹	<code>objects()</code> nennt alle Variablen in der aktuellen R-Sitzung (genau so wie <code>ls()</code> ⁵⁰⁰).
<code>odfWeave()</code> ⁵⁰¹	<code>odfWeave()</code> ersetzt die in Open Office Dokumenten integrierten R-Anweisungen durch die resultierenden Textausgaben, Tabellen und Graphiken.
<code>options()</code> ⁵⁰²	<code>options()</code> ermöglicht Abfrage und Setzen globaler Voreinstellungen wie Textausgabeeigenschaften, <code>GraphicDevice</code> , etc.
<code>order()</code> ⁵⁰³	Der Befehl <code>order(OBJEKT\$x)</code> ordnet den Datensatz <code>OBJEKT</code> anhand des dort enthaltenen metrischen Vektor <code>x</code> in aufsteigender Reihenfolge.

473 Kapitel 51.1 auf Seite 197

474 Kapitel 51.2 auf Seite 198

475 Kapitel 51.3 auf Seite 199

476 Kapitel 51.4 auf Seite 201

477 Kapitel 51.5 auf Seite 202

478 Kapitel 51.6 auf Seite 203

479 Kapitel 51.7 auf Seite 204

480 Kapitel 51.8 auf Seite 205

481 Kapitel 51.9 auf Seite 206

482 Kapitel 51.10 auf Seite 208

483 Kapitel 51.11 auf Seite 209

484 Kapitel 51.12 auf Seite 210

485 Kapitel 51.13 auf Seite 211

486 Kapitel 51.14 auf Seite 213

487 Kapitel 51.15 auf Seite 214

488 Kapitel 51.16 auf Seite 215

489 Kapitel 51.17 auf Seite 217

490 Kapitel 51.18 auf Seite 218

491 Kapitel 51.19 auf Seite 219

492 Kapitel 51.20 auf Seite 222

493 Kapitel 51.21 auf Seite 223

494 Kapitel 51.22 auf Seite 224

495 Kapitel 51.23 auf Seite 225

496 Kapitel 51.24 auf Seite 226

497 Kapitel 51.25 auf Seite 227

498 Kapitel 51.26 auf Seite 228

499 <http://de.wikibooks.org/wiki/GNU%20R%3A%20objects>

500 <http://de.wikibooks.org/wiki/GNU%20R%3A%20ls>

501 <http://de.wikibooks.org/wiki/GNU%20R%3A%20odfWeave>

502 Kapitel 56 auf Seite 239

503 <http://de.wikibooks.org/wiki/GNU%20R%3A%20order>

51.16. P

A⁵⁰⁴ - B⁵⁰⁵ - C⁵⁰⁶ - D⁵⁰⁷ - E⁵⁰⁸ - F⁵⁰⁹ - G⁵¹⁰ - H⁵¹¹ - I⁵¹² - J⁵¹³ - K⁵¹⁴ - L⁵¹⁵ - M⁵¹⁶ - N⁵¹⁷ - O⁵¹⁸ - P⁵¹⁹ - Q⁵²⁰ - R⁵²¹ - S⁵²² - T⁵²³ - U⁵²⁴ - V⁵²⁵ - W⁵²⁶ - X⁵²⁷ - Y⁵²⁸ - Z⁵²⁹

Befehl

`pairs()`⁵³⁰

`par()`⁵³¹

`package.skeleton()`⁵³²

`paste(a, b, c)`⁵³³

`pdf()`⁵³⁴

`persp()`⁵³⁵

`pie()`⁵³⁶

Kurzbeschreibung

`pair()` ist eine Funktion zum Zeichnen einer Scatterplot-Matrix

`par()` ist eine Funktion zum Setzen grafischer Parameter
`package.skeleton()` erstellt automatisch die Grundstruktur für neue (selbstprogrammierte) Zusatzpakete.

Mit `paste(a, b, c)` können beliebig viele Argumente (a, b, c, "beliebig") ausgeprintet werden.

Mit Hilfe von `pdf()` können Graphiken im weit verbreiteten .pdf-Format gespeichert werden.

Mit `persp()` können dreidimensionalen Abbildungen erstellt werden.

Mit `pie()` können Kreisdiagramme erstellt werden.

504 Kapitel 51.1 auf Seite 197

505 Kapitel 51.2 auf Seite 198

506 Kapitel 51.3 auf Seite 199

507 Kapitel 51.4 auf Seite 201

508 Kapitel 51.5 auf Seite 202

509 Kapitel 51.6 auf Seite 203

510 Kapitel 51.7 auf Seite 204

511 Kapitel 51.8 auf Seite 205

512 Kapitel 51.9 auf Seite 206

513 Kapitel 51.10 auf Seite 208

514 Kapitel 51.11 auf Seite 209

515 Kapitel 51.12 auf Seite 210

516 Kapitel 51.13 auf Seite 211

517 Kapitel 51.14 auf Seite 213

518 Kapitel 51.15 auf Seite 214

519 Kapitel 51.16 auf Seite 215

520 Kapitel 51.17 auf Seite 217

521 Kapitel 51.18 auf Seite 218

522 Kapitel 51.19 auf Seite 219

523 Kapitel 51.20 auf Seite 222

524 Kapitel 51.21 auf Seite 223

525 Kapitel 51.22 auf Seite 224

526 Kapitel 51.23 auf Seite 225

527 Kapitel 51.24 auf Seite 226

528 Kapitel 51.25 auf Seite 227

529 Kapitel 51.26 auf Seite 228

530 <http://de.wikibooks.org/wiki/GNU%20R%3A%20pairs>

531 <http://de.wikibooks.org/wiki/GNU%20R%3A%20par>

532 <http://de.wikibooks.org/wiki/GNU%20R%3A%20package.skeleton>

533 <http://de.wikibooks.org/wiki/GNU%20R%3A%20paste>

534 <http://de.wikibooks.org/wiki/GNU%20R%3A%20pdf>

535 <http://de.wikibooks.org/wiki/GNU%20R%3A%20persp>

536 <http://de.wikibooks.org/wiki/GNU%20R%3A%20pie>

<code>plot(x, y)</code> ⁵³⁷	<code>plot(x, y)</code> ist die universelle Funktion zur Erzeugung von Streudiagrammen und Linienzügen aus den Vektoren <code>x</code> und <code>y</code> .
<code>png()</code> ⁵³⁸	Mit Hilfe von <code>png()</code> können Graphiken in eine <code>.png</code> -Datei gespeichert werden. Für das <code>jpeg</code> -Format steht der Befehl <code>jpeg()</code> ⁵³⁹ zur Verfügung.
<code>polygon(x, y)</code> ⁵⁴⁰	Mit der Grafikfunktion <code>polygon(x, y)</code> können Grafiken mit beliebigen geschlossenen Linienzügen erweitert werden.
<code>postscript()</code> ⁵⁴¹	<code>postscript()</code> speichert die Grafikausgabe in eine Postscript-Datei.
<code>prime.factor(x)</code> ⁵⁴²	<code>prime.factor(x)</code> (aus dem <code>schoolmath</code> -Paket) zerlegt die Zahl <code>x</code> in ihre Primfaktoren.
<code>primes(x, y)</code> ⁵⁴³	<code>primes(x, y)</code> (aus dem <code>schoolmath</code> -Paket) zeigt die Primzahlen an, welche sich zwischen <code>x</code> und <code>y</code> befinden.
<code>print</code> ⁵⁴⁴	<code>print(x)</code> gibt das Objekt <code>x</code> entsprechend seiner definierten Methode auf dem Bildschirm/das Standardgerät aus.
<code>princomp(x)</code> ⁵⁴⁵	<code>princomp(x)</code> führt eine Hauptkomponentenanalyse auf das Objekt <code>x</code> durch (siehe auch <code>factanal(x)</code> ⁵⁴⁶).

537 <http://de.wikibooks.org/wiki/GNU%20R%3A%20plot>

538 <http://de.wikibooks.org/wiki/GNU%20R%3A%20png>

539 <http://de.wikibooks.org/wiki/GNU%20R%3A%20jpeg>

540 <http://de.wikibooks.org/wiki/GNU%20R%3A%20polygon>

541 <http://de.wikibooks.org/wiki/GNU%20R%3A%20postscript>

542 <http://de.wikibooks.org/wiki/GNU%20R%3A%20prime.factor>

543 <http://de.wikibooks.org/wiki/GNU%20R%3A%20primes>

544 <http://de.wikibooks.org/wiki/GNU%20R%3A%20print>

545 <http://de.wikibooks.org/wiki/GNU%20R%3A%20princomp>

546 <http://de.wikibooks.org/wiki/GNU%20R%3A%20factanal>

51.17. Q

A⁵⁴⁷ - B⁵⁴⁸ - C⁵⁴⁹ - D⁵⁵⁰ - E⁵⁵¹ - F⁵⁵² - G⁵⁵³ - H⁵⁵⁴ - I⁵⁵⁵ - J⁵⁵⁶ - K⁵⁵⁷ - L⁵⁵⁸ - M⁵⁵⁹ - N⁵⁶⁰ - O⁵⁶¹ - P⁵⁶² - Q⁵⁶³ - R⁵⁶⁴ - S⁵⁶⁵ - T⁵⁶⁶ - U⁵⁶⁷ - V⁵⁶⁸ - W⁵⁶⁹ - X⁵⁷⁰ - Y⁵⁷¹ - Z⁵⁷²

Befehl

`quit ()`⁵⁷³ / `q ()`⁵⁷⁴

Kurzbeschreibung

`quit ()` bzw. `q ()` beendet R.

547 Kapitel 51.1 auf Seite 197
 548 Kapitel 51.2 auf Seite 198
 549 Kapitel 51.3 auf Seite 199
 550 Kapitel 51.4 auf Seite 201
 551 Kapitel 51.5 auf Seite 202
 552 Kapitel 51.6 auf Seite 203
 553 Kapitel 51.7 auf Seite 204
 554 Kapitel 51.8 auf Seite 205
 555 Kapitel 51.9 auf Seite 206
 556 Kapitel 51.10 auf Seite 208
 557 Kapitel 51.11 auf Seite 209
 558 Kapitel 51.12 auf Seite 210
 559 Kapitel 51.13 auf Seite 211
 560 Kapitel 51.14 auf Seite 213
 561 Kapitel 51.15 auf Seite 214
 562 Kapitel 51.16 auf Seite 215
 563 Kapitel 51.17 auf Seite 217
 564 Kapitel 51.18 auf Seite 218
 565 Kapitel 51.19 auf Seite 219
 566 Kapitel 51.20 auf Seite 222
 567 Kapitel 51.21 auf Seite 223
 568 Kapitel 51.22 auf Seite 224
 569 Kapitel 51.23 auf Seite 225
 570 Kapitel 51.24 auf Seite 226
 571 Kapitel 51.25 auf Seite 227
 572 Kapitel 51.26 auf Seite 228
 573 <http://de.wikibooks.org/wiki/GNU%20R%3A%20quit>
 574 <http://de.wikibooks.org/wiki/GNU%20R%3A%20quit>

51.18. R

A⁵⁷⁵ - B⁵⁷⁶ - C⁵⁷⁷ - D⁵⁷⁸ - E⁵⁷⁹ - F⁵⁸⁰ - G⁵⁸¹ - H⁵⁸² - I⁵⁸³ - J⁵⁸⁴ - K⁵⁸⁵ - L⁵⁸⁶ - M⁵⁸⁷ - N⁵⁸⁸ - O⁵⁸⁹ - P⁵⁹⁰ - Q⁵⁹¹ - R⁵⁹² - S⁵⁹³ - T⁵⁹⁴ - U⁵⁹⁵ - V⁵⁹⁶ - W⁵⁹⁷ - X⁵⁹⁸ - Y⁵⁹⁹ - Z⁶⁰⁰

Befehl

`rbind()`⁶⁰¹

`read.csv(file)`⁶⁰³

`read.csv2(file)`⁶⁰⁴

`read.table(file)`⁶⁰⁵

`rep(x, n)`⁶⁰⁶

`rev(x)`⁶⁰⁷

`rm(x)`⁶⁰⁸

Kurzbeschreibung

mit `rbind(a, b, c)` lassen sich die Vektoren `a`, `b` und `c` reihenweise zu einer Matrix zusammenführen (siehe auch `cbind()`⁶⁰²)

`read.csv(file)` liest eine ASCII-Datei `file`, welche im amerikanischen CSV-Format vorliegt, ein.

`read.csv2(file)` liest eine ASCII-Datei `file`, welche im deutschen CSV-Format vorliegt, ein.

`read.table(file)` liest eine Tabelle, welche in der ASCII-Datei `file` vorliegt, ein.

`rep(x, n)` gibt `n` Wiederholungen des Elements `x` zurück.

`rev(x)` dreht die Reihenfolge der Zahlenreihe `x` um.

`rm(x)` löscht das Objekt `x` aus dem Arbeitsspeicher.

575 Kapitel 51.1 auf Seite 197

576 Kapitel 51.2 auf Seite 198

577 Kapitel 51.3 auf Seite 199

578 Kapitel 51.4 auf Seite 201

579 Kapitel 51.5 auf Seite 202

580 Kapitel 51.6 auf Seite 203

581 Kapitel 51.7 auf Seite 204

582 Kapitel 51.8 auf Seite 205

583 Kapitel 51.9 auf Seite 206

584 Kapitel 51.10 auf Seite 208

585 Kapitel 51.11 auf Seite 209

586 Kapitel 51.12 auf Seite 210

587 Kapitel 51.13 auf Seite 211

588 Kapitel 51.14 auf Seite 213

589 Kapitel 51.15 auf Seite 214

590 Kapitel 51.16 auf Seite 215

591 Kapitel 51.17 auf Seite 217

592 Kapitel 51.18 auf Seite 218

593 Kapitel 51.19 auf Seite 219

594 Kapitel 51.20 auf Seite 222

595 Kapitel 51.21 auf Seite 223

596 Kapitel 51.22 auf Seite 224

597 Kapitel 51.23 auf Seite 225

598 Kapitel 51.24 auf Seite 226

599 Kapitel 51.25 auf Seite 227

600 Kapitel 51.26 auf Seite 228

601 <http://de.wikibooks.org/wiki/GNU%20R%3A%20rbind>

602 Kapitel 51.3 auf Seite 199

603 <http://de.wikibooks.org/wiki/GNU%20R%3A%20read.csv>

604 <http://de.wikibooks.org/wiki/GNU%20R%3A%20read.csv2>

605 <http://de.wikibooks.org/wiki/GNU%20R%3A%20read.table>

606 <http://de.wikibooks.org/wiki/GNU%20R%3A%20rep>

607 <http://de.wikibooks.org/wiki/GNU%20R%3A%20rev>

608 <http://de.wikibooks.org/wiki/GNU%20R%3A%20rm>

<code>rnorm(n)</code> ⁶⁰⁹	<code>rnorm(n)</code> erzeugt n normalverteilte Zufallszahlen.
<code>round(x)</code> ⁶¹⁰	<code>round(x)</code> rundet die Zahl x auf oder ab.
<code>rownames()</code> ⁶¹¹	<code>rownames()</code> weist den Reihen einer Matrix einen "Namen" (Label) zu. Siehe <code>colnames()</code> ⁶¹² für Spalten)

51.19. S

A⁶¹³ - B⁶¹⁴ - C⁶¹⁵ - D⁶¹⁶ - E⁶¹⁷ - F⁶¹⁸ - G⁶¹⁹ - H⁶²⁰ - I⁶²¹ - J⁶²² - K⁶²³ - L⁶²⁴ - M⁶²⁵ - N⁶²⁶ - O⁶²⁷ - P⁶²⁸ - Q⁶²⁹ - R⁶³⁰ - S⁶³¹ - T⁶³² - U⁶³³ - V⁶³⁴ - W⁶³⁵ - X⁶³⁶ - Y⁶³⁷ - Z⁶³⁸

Befehl

`sample(x)`⁶³⁹

`sapply(x, func)`⁶⁴⁰

`save()`⁶⁴¹

Kurzbeschreibung

Mit `sample(x)` kann ein Vektor umgruppiert oder Teilstichproben daraus gezogen werden.

`sapply` ermöglicht es, die selbe Funktion `func` auf eine Liste x von Objekten anzuwenden, und gibt das Ergebnis als Vektor aus.

`save()` speichert ausgewählte Objekte in eine Datei.

609 <http://de.wikibooks.org/wiki/GNU%20R%3A%20rnorm>

610 <http://de.wikibooks.org/wiki/GNU%20R%3A%20round>

611 <http://de.wikibooks.org/wiki/GNU%20R%3A%20rownames>

612 <http://de.wikibooks.org/wiki/GNU%20R%3A%20colnames>

613 Kapitel 51.1 auf Seite 197

614 Kapitel 51.2 auf Seite 198

615 Kapitel 51.3 auf Seite 199

616 Kapitel 51.4 auf Seite 201

617 Kapitel 51.5 auf Seite 202

618 Kapitel 51.6 auf Seite 203

619 Kapitel 51.7 auf Seite 204

620 Kapitel 51.8 auf Seite 205

621 Kapitel 51.9 auf Seite 206

622 Kapitel 51.10 auf Seite 208

623 Kapitel 51.11 auf Seite 209

624 Kapitel 51.12 auf Seite 210

625 Kapitel 51.13 auf Seite 211

626 Kapitel 51.14 auf Seite 213

627 Kapitel 51.15 auf Seite 214

628 Kapitel 51.16 auf Seite 215

629 Kapitel 51.17 auf Seite 217

630 Kapitel 51.18 auf Seite 218

631 Kapitel 51.19 auf Seite 219

632 Kapitel 51.20 auf Seite 222

633 Kapitel 51.21 auf Seite 223

634 Kapitel 51.22 auf Seite 224

635 Kapitel 51.23 auf Seite 225

636 Kapitel 51.24 auf Seite 226

637 Kapitel 51.25 auf Seite 227

638 Kapitel 51.26 auf Seite 228

639 <http://de.wikibooks.org/wiki/GNU%20R%3A%20sample>

640 <http://de.wikibooks.org/wiki/GNU%20R%3A%20sapply>

641 <http://de.wikibooks.org/wiki/GNU%20R%3A%20save>

<code>save.image("file.RData")</code> ⁶⁴²	<code>save.image("file.RData")</code> speichert den aktuellen Arbeitsspeicher.
<code>savePlot()</code> ⁶⁴³	<code>savePlot()</code> speichert die aktuelle Grafik in eine Datei.
<code>scan()</code> ⁶⁴⁴	<code>scan()</code> ermöglicht eine manuelle Dateneingabe für einen Vektor.
<code>scm(x, y)</code> ⁶⁴⁵	<code>scm(x, y)</code> (aus dem <code>schoolmath</code> -Paket) berechnet das kleinste gemeinsame Vielfache der Zahlen <code>x</code> und <code>y</code> .
<code>search()</code> ⁶⁴⁶	<code>search()</code> zeigt die Suchliste an, unter anderem auch geladene Bibliotheken.
<code>searchpaths()</code> ⁶⁴⁷	<code>searchpaths()</code> zeigt die Suchliste mit Pfadangaben der geladenen Bibliotheken an.
<code>seq()</code> ⁶⁴⁸	Mit Hilfe von <code>seq()</code> kann eine Datensequenz erstellt werden.
<code>setdiff(x, y)</code> ⁶⁴⁹	<code>setdiff(x, y)</code> gibt die Elemente von <code>x</code> , die nicht in <code>y</code> enthalten sind, als neuen Vektor aus. Funktioniert auch umgekehrt mit <code>setdiff(y, x)</code> .
<code>setequal(x, y)</code> ⁶⁵⁰	<code>setequal(x, y)</code> untersucht beide Vektoren auf Gleichheit und gibt <code>TRUE/FALSE</code> aus.
<code>setwd(dir)</code> ⁶⁵¹	Mit <code>setwd(dir)</code> kann das Arbeitsverzeichnis geändert werden.
<code>shapiro.test(x)</code> ⁶⁵²	<code>shapiro.test(x)</code> führt einen Shapiro-Wilk-Test auf die Zahlenreihe <code>x</code> durch. Hierdurch wird bestimmt, ob die Zahlenreihe <code>x</code> normalverteilt ist.
<code>sink()</code> ⁶⁵³	<code>sink()</code> leitet den Konsolen-Output in eine Datei um.
<code>sort()</code> ⁶⁵⁴	Der Befehl <code>sort(x)</code> ordnet den metrischen Vektor <code>x</code> in aufsteigender Reihenfolge (siehe <code>order()</code> ⁶⁵⁵ zum Sortieren eines Datensatzes).
<code>source()</code> ⁶⁵⁶	Mit dem Befehl <code>source()</code> werden R-Skripte geladen und sofort ausgeführt.
<code>sqrt(x)</code> ⁶⁵⁷	<code>sqrt(x)</code> zieht die Quadratwurzel aus der Zahl <code>x</code> .

642 <http://de.wikibooks.org/wiki/GNU%20R%3A%20save.image>

643 <http://de.wikibooks.org/wiki/GNU%20R%3A%20savePlot>

644 <http://de.wikibooks.org/wiki/GNU%20R%3A%20scan>

645 <http://de.wikibooks.org/wiki/GNU%20R%3A%20scm>

646 <http://de.wikibooks.org/wiki/GNU%20R%3A%20search>

647 <http://de.wikibooks.org/wiki/GNU%20R%3A%20searchpaths>

648 <http://de.wikibooks.org/wiki/GNU%20R%3A%20seq>

649 <http://de.wikibooks.org/wiki/GNU%20R%3A%20setdiff>

650 <http://de.wikibooks.org/wiki/GNU%20R%3A%20setequal>

651 <http://de.wikibooks.org/wiki/GNU%20R%3A%20setwd>

652 <http://de.wikibooks.org/wiki/GNU%20R%3A%20shapiro.test>

653 <http://de.wikibooks.org/wiki/GNU%20R%3A%20sink>

654 <http://de.wikibooks.org/wiki/GNU%20R%3A%20sort>

655 <http://de.wikibooks.org/wiki/GNU%20R%3A%20order>

656 <http://de.wikibooks.org/wiki/GNU%20R%3A%20source>

657 <http://de.wikibooks.org/wiki/GNU%20R%3A%20sqrt>

<code>str()</code> ⁶⁵⁸	<code>str()</code> gibt Auskunft über die Struktur von R-Objekten.
<code>strptime()</code> ⁶⁵⁹	<code>strptime()</code> wandelt Datum- und Zeitangaben in das Format <code>POSIXlt</code> bzw. <code>POSIXct</code> um.
<code>strsplit()</code> ⁶⁶⁰	<code>strsplit()</code> spaltet eine Zeichenkette abhängig vom angegebenen Muster.
<code>sub()</code> ⁶⁶¹	Mit <code>sub("Muster", "Ersetzung", Objekt)</code> lassen sich Zeichenketten in einem Objekt ersetzen.
<code>subset()</code> ⁶⁶²	Mit <code>subset()</code> lässt sich eine Teilgruppe von Daten aus einem <code>data.frame</code> ⁶⁶³ bilden.
<code>sum()</code> ⁶⁶⁴	Mit <code>sum()</code> wird die Summe der übergebenen Werte bestimmt.
<code>summary()</code> ⁶⁶⁵	Mit <code>summary()</code> lassen sich das Minimum, das Maximum, das 1. und 3. Quantil sowie Mittelwert und der Median von Verteilungen ausgeben.
<code>Sweave()</code> ⁶⁶⁶	<code>Sweave()</code> ersetzt die in (La)Tex-Dokumenten integrierten R-Anweisungen durch die resultierenden Textausgaben, Tabellen oder generiert Graphiken und bindet diese automatisch in die Tex-Datei ein.
<code>Sys.time()</code> ⁶⁶⁷	<code>Sys.time()</code> liefert das aktuelle Datum und die aktuelle Uhrzeit im Format <code>POSIXlt</code> bzw. <code>POSIXct</code>

658 <http://de.wikibooks.org/wiki/GNU%20R%3A%20str>

659 <http://de.wikibooks.org/wiki/GNU%20R%3A%20strptime>

660 <http://de.wikibooks.org/wiki/GNU%20R%3A%20strsplit>

661 <http://de.wikibooks.org/wiki/GNU%20R%3A%20sub>

662 <http://de.wikibooks.org/wiki/GNU%20R%3A%20subset>

663 <http://de.wikibooks.org/wiki/GNU%20R%3A%20data.frame>

664 <http://de.wikibooks.org/wiki/GNU%20R%3A%20sum>

665 <http://de.wikibooks.org/wiki/GNU%20R%3A%20summary>

666 <http://de.wikibooks.org/wiki/GNU%20R%3A%20Sweave>

667 <http://de.wikibooks.org/wiki/GNU%20R%3A%20Sys.time>

51.20. T

A⁶⁶⁸ - B⁶⁶⁹ - C⁶⁷⁰ - D⁶⁷¹ - E⁶⁷² - F⁶⁷³ - G⁶⁷⁴ - H⁶⁷⁵ - I⁶⁷⁶ - J⁶⁷⁷ - K⁶⁷⁸ - L⁶⁷⁹ - M⁶⁸⁰ - N⁶⁸¹ - O⁶⁸² - P⁶⁸³ - Q⁶⁸⁴ - R⁶⁸⁵ - S⁶⁸⁶ - T⁶⁸⁷ - U⁶⁸⁸ - V⁶⁸⁹ - W⁶⁹⁰ - X⁶⁹¹ - Y⁶⁹² - Z⁶⁹³

Befehl	Kurzbeschreibung
<code>t.test()</code> ⁶⁹⁴	<code>t.test()</code> führt einen t-Test (und bei Bedarf einen Welch-Test) durch.
<code>table()</code> ⁶⁹⁵	Mit <code>table()</code> werden Häufigkeitstabellen erstellt.
<code>tail()</code> ⁶⁹⁶	Mit dem Befehl <code>tail(DATA, n)</code> lassen sich die letzten <code>n</code> Datenreihen des Datensatzes <code>DATA</code> anzeigen. Der Befehl <code>head()</code> ⁶⁹⁷ liefert hingegen die ersten <code>n</code> Datenreihen.
<code>tan(x)</code> ⁶⁹⁸	<code>tan(x)</code> berechnet den Tangens von <code>x</code> .
<code>title()</code> ⁶⁹⁹	<code>title()</code> ermöglicht das (nachträgliche) Beschriften von Grafiken.
<code>tolower(x)</code> ⁷⁰⁰	<code>tolower(x)</code> wandelt den String <code>x</code> in Kleinbuchstaben um.
<code>toupper(x)</code> ⁷⁰¹	<code>toupper(x)</code> wandelt den String <code>x</code> in Großbuchstaben um.

668 Kapitel 51.1 auf Seite 197

669 Kapitel 51.2 auf Seite 198

670 Kapitel 51.3 auf Seite 199

671 Kapitel 51.4 auf Seite 201

672 Kapitel 51.5 auf Seite 202

673 Kapitel 51.6 auf Seite 203

674 Kapitel 51.7 auf Seite 204

675 Kapitel 51.8 auf Seite 205

676 Kapitel 51.9 auf Seite 206

677 Kapitel 51.10 auf Seite 208

678 Kapitel 51.11 auf Seite 209

679 Kapitel 51.12 auf Seite 210

680 Kapitel 51.13 auf Seite 211

681 Kapitel 51.14 auf Seite 213

682 Kapitel 51.15 auf Seite 214

683 Kapitel 51.16 auf Seite 215

684 Kapitel 51.17 auf Seite 217

685 Kapitel 51.18 auf Seite 218

686 Kapitel 51.19 auf Seite 219

687 Kapitel 51.20 auf Seite 222

688 Kapitel 51.21 auf Seite 223

689 Kapitel 51.22 auf Seite 224

690 Kapitel 51.23 auf Seite 225

691 Kapitel 51.24 auf Seite 226

692 Kapitel 51.25 auf Seite 227

693 Kapitel 51.26 auf Seite 228

694 <http://de.wikibooks.org/wiki/GNU%20R%3A%20t.test>

695 <http://de.wikibooks.org/wiki/GNU%20R%3A%20table>

696 <http://de.wikibooks.org/wiki/GNU%20R%3A%20tail>

697 <http://de.wikibooks.org/wiki/GNU%20R%3A%20head>

698 <http://de.wikibooks.org/wiki/GNU%20R%3A%20tan>

699 <http://de.wikibooks.org/wiki/GNU%20R%3A%20title>

700 <http://de.wikibooks.org/wiki/GNU%20R%3A%20tolower>

701 <http://de.wikibooks.org/wiki/GNU%20R%3A%20toupper>

51.21. U

A⁷⁰² - B⁷⁰³ - C⁷⁰⁴ - D⁷⁰⁵ - E⁷⁰⁶ - F⁷⁰⁷ - G⁷⁰⁸ - H⁷⁰⁹ - I⁷¹⁰ - J⁷¹¹ - K⁷¹² - L⁷¹³ - M⁷¹⁴ - N⁷¹⁵ - O⁷¹⁶ - P⁷¹⁷ - Q⁷¹⁸ - R⁷¹⁹ - S⁷²⁰ - T⁷²¹ - U⁷²² - V⁷²³ - W⁷²⁴ - X⁷²⁵ - Y⁷²⁶ - Z⁷²⁷

Befehl

`union(x, y)`⁷²⁸

`unique()`⁷²⁹

`unlink()`⁷³¹

`update(x)`⁷³²

`update.packages()`⁷³³

Kurzbeschreibung

`union(x, y)` vereint die Werte des Vektors `x` mit den Werten des Vektors `y`, die nicht bereits in `x` enthalten sind.

Umgekehrt funktioniert auch `union(y, x)`.

`unique(data)` überprüft den Datensatz `data` auf doppelte bzw. mehrfache Einträge und liefert den Datensatz ohne Duplikate zurück (siehe auch `duplicated()`⁷³⁰).

`unlink()` löscht eine Datei oder ein Verzeichnis

`update(x)` erweitert das bestehende Modell `x`.

`update.packages()` aktualisiert die geladenen R-Basis- und Zusatzpakete.

702 Kapitel 51.1 auf Seite 197

703 Kapitel 51.2 auf Seite 198

704 Kapitel 51.3 auf Seite 199

705 Kapitel 51.4 auf Seite 201

706 Kapitel 51.5 auf Seite 202

707 Kapitel 51.6 auf Seite 203

708 Kapitel 51.7 auf Seite 204

709 Kapitel 51.8 auf Seite 205

710 Kapitel 51.9 auf Seite 206

711 Kapitel 51.10 auf Seite 208

712 Kapitel 51.11 auf Seite 209

713 Kapitel 51.12 auf Seite 210

714 Kapitel 51.13 auf Seite 211

715 Kapitel 51.14 auf Seite 213

716 Kapitel 51.15 auf Seite 214

717 Kapitel 51.16 auf Seite 215

718 Kapitel 51.17 auf Seite 217

719 Kapitel 51.18 auf Seite 218

720 Kapitel 51.19 auf Seite 219

721 Kapitel 51.20 auf Seite 222

722 Kapitel 51.21 auf Seite 223

723 Kapitel 51.22 auf Seite 224

724 Kapitel 51.23 auf Seite 225

725 Kapitel 51.24 auf Seite 226

726 Kapitel 51.25 auf Seite 227

727 Kapitel 51.26 auf Seite 228

728 <http://de.wikibooks.org/wiki/GNU%20R%3A%20union>

729 <http://de.wikibooks.org/wiki/GNU%20R%3A%20unique>

730 <http://de.wikibooks.org/wiki/GNU%20R%3A%20duplicated>

731 <http://de.wikibooks.org/wiki/GNU%20R%3A%20unlink>

732 <http://de.wikibooks.org/wiki/GNU%20R%3A%20update>

733 <http://de.wikibooks.org/wiki/GNU%20R%3A%20update.packages>

51.22. V

A⁷³⁴ - B⁷³⁵ - C⁷³⁶ - D⁷³⁷ - E⁷³⁸ - F⁷³⁹ - G⁷⁴⁰ - H⁷⁴¹ - I⁷⁴² - J⁷⁴³ - K⁷⁴⁴ - L⁷⁴⁵ - M⁷⁴⁶ - N⁷⁴⁷ - O⁷⁴⁸ - P⁷⁴⁹ - Q⁷⁵⁰ - R⁷⁵¹ - S⁷⁵² - T⁷⁵³ - U⁷⁵⁴ - V⁷⁵⁵ - W⁷⁵⁶ - X⁷⁵⁷ - Y⁷⁵⁸ - Z⁷⁵⁹

Befehl

`var(x, y)`⁷⁶⁰

`var.test(x, y)`⁷⁶¹

Kurzbeschreibung

`var(x, y)` berechnet die Varianz oder Kovarianz der Variable x bzw. der Variablen x und y .

`var.test(x, y)` führt einen Varianzquotienten-Test (F-Test) durch.

734 Kapitel 51.1 auf Seite 197

735 Kapitel 51.2 auf Seite 198

736 Kapitel 51.3 auf Seite 199

737 Kapitel 51.4 auf Seite 201

738 Kapitel 51.5 auf Seite 202

739 Kapitel 51.6 auf Seite 203

740 Kapitel 51.7 auf Seite 204

741 Kapitel 51.8 auf Seite 205

742 Kapitel 51.9 auf Seite 206

743 Kapitel 51.10 auf Seite 208

744 Kapitel 51.11 auf Seite 209

745 Kapitel 51.12 auf Seite 210

746 Kapitel 51.13 auf Seite 211

747 Kapitel 51.14 auf Seite 213

748 Kapitel 51.15 auf Seite 214

749 Kapitel 51.16 auf Seite 215

750 Kapitel 51.17 auf Seite 217

751 Kapitel 51.18 auf Seite 218

752 Kapitel 51.19 auf Seite 219

753 Kapitel 51.20 auf Seite 222

754 Kapitel 51.21 auf Seite 223

755 Kapitel 51.22 auf Seite 224

756 Kapitel 51.23 auf Seite 225

757 Kapitel 51.24 auf Seite 226

758 Kapitel 51.25 auf Seite 227

759 Kapitel 51.26 auf Seite 228

760 <http://de.wikibooks.org/wiki/GNU%20R%3A%20var>

761 <http://de.wikibooks.org/wiki/GNU%20R%3A%20var.test>

51.23. W

A⁷⁶² - B⁷⁶³ - C⁷⁶⁴ - D⁷⁶⁵ - E⁷⁶⁶ - F⁷⁶⁷ - G⁷⁶⁸ - H⁷⁶⁹ - I⁷⁷⁰ - J⁷⁷¹ - K⁷⁷² - L⁷⁷³ - M⁷⁷⁴ - N⁷⁷⁵ - O⁷⁷⁶ - P⁷⁷⁷ - Q⁷⁷⁸ - R⁷⁷⁹ - S⁷⁸⁰ - T⁷⁸¹ - U⁷⁸² - V⁷⁸³ - W⁷⁸⁴ - X⁷⁸⁵ - Y⁷⁸⁶ - Z⁷⁸⁷

Befehl

`which()`⁷⁸⁸

`wilcox.test(x, y)`⁷⁸⁹

`wilcox_
test(x, y)`⁷⁹⁰

`write()`⁷⁹¹

`write.table()`⁷⁹²

Kurzbeschreibung

Die Funktion `which()` gibt an, welche Indizes eines Objektes den Wert "TRUE" besitzen.

`wilcox.test(x, y)` führt je nach Optionsgabe den Wilcoxon-Test oder den Mann-Whitney-Test für x und y durch.

`wilcox_test(x, y)` (aus dem `coin`-Package) führt den Mann-Whitney-Test durch.

`write()` schreibt Daten in eine Datei.

`write()` schreibt einen Data Table in eine Datei.

762 Kapitel 51.1 auf Seite 197

763 Kapitel 51.2 auf Seite 198

764 Kapitel 51.3 auf Seite 199

765 Kapitel 51.4 auf Seite 201

766 Kapitel 51.5 auf Seite 202

767 Kapitel 51.6 auf Seite 203

768 Kapitel 51.7 auf Seite 204

769 Kapitel 51.8 auf Seite 205

770 Kapitel 51.9 auf Seite 206

771 Kapitel 51.10 auf Seite 208

772 Kapitel 51.11 auf Seite 209

773 Kapitel 51.12 auf Seite 210

774 Kapitel 51.13 auf Seite 211

775 Kapitel 51.14 auf Seite 213

776 Kapitel 51.15 auf Seite 214

777 Kapitel 51.16 auf Seite 215

778 Kapitel 51.17 auf Seite 217

779 Kapitel 51.18 auf Seite 218

780 Kapitel 51.19 auf Seite 219

781 Kapitel 51.20 auf Seite 222

782 Kapitel 51.21 auf Seite 223

783 Kapitel 51.22 auf Seite 224

784 Kapitel 51.23 auf Seite 225

785 Kapitel 51.24 auf Seite 226

786 Kapitel 51.25 auf Seite 227

787 Kapitel 51.26 auf Seite 228

788 <http://de.wikibooks.org/wiki/GNU%20R%3A%20which>

789 <http://de.wikibooks.org/wiki/GNU%20R%3A%20wilcox.test>

790 http://de.wikibooks.org/wiki/GNU%20R%3A%20wilcox_test

791 <http://de.wikibooks.org/wiki/GNU%20R%3A%20write>

792 <http://de.wikibooks.org/wiki/GNU%20R%3A%20write.table>

51.24. X

A⁷⁹³ - B⁷⁹⁴ - C⁷⁹⁵ - D⁷⁹⁶ - E⁷⁹⁷ - F⁷⁹⁸ - G⁷⁹⁹ - H⁸⁰⁰ - I⁸⁰¹ - J⁸⁰² - K⁸⁰³ - L⁸⁰⁴ - M⁸⁰⁵ - N⁸⁰⁶ - O⁸⁰⁷ - P⁸⁰⁸ - Q⁸⁰⁹ - R⁸¹⁰ - S⁸¹¹ - T⁸¹² - U⁸¹³ - V⁸¹⁴ - W⁸¹⁵ - X⁸¹⁶ - Y⁸¹⁷ - Z⁸¹⁸

Befehl	Kurzbeschreibung
x11 () ⁸¹⁹	x11 () öffnet ein Grafikfenster per X11.
xtabs () ⁸²⁰	xtabs () dient dazu Kreuztabellen zu erstellen.

793	Kapitel 51.1 auf Seite 197
794	Kapitel 51.2 auf Seite 198
795	Kapitel 51.3 auf Seite 199
796	Kapitel 51.4 auf Seite 201
797	Kapitel 51.5 auf Seite 202
798	Kapitel 51.6 auf Seite 203
799	Kapitel 51.7 auf Seite 204
800	Kapitel 51.8 auf Seite 205
801	Kapitel 51.9 auf Seite 206
802	Kapitel 51.10 auf Seite 208
803	Kapitel 51.11 auf Seite 209
804	Kapitel 51.12 auf Seite 210
805	Kapitel 51.13 auf Seite 211
806	Kapitel 51.14 auf Seite 213
807	Kapitel 51.15 auf Seite 214
808	Kapitel 51.16 auf Seite 215
809	Kapitel 51.17 auf Seite 217
810	Kapitel 51.18 auf Seite 218
811	Kapitel 51.19 auf Seite 219
812	Kapitel 51.20 auf Seite 222
813	Kapitel 51.21 auf Seite 223
814	Kapitel 51.22 auf Seite 224
815	Kapitel 51.23 auf Seite 225
816	Kapitel 51.24 auf Seite 226
817	Kapitel 51.25 auf Seite 227
818	Kapitel 51.26 auf Seite 228
819	http://de.wikibooks.org/wiki/GNU%20R%3A%20x11
820	http://de.wikibooks.org/wiki/GNU%20R%3A%20xtabs

51.25. Y

A⁸²¹ - B⁸²² - C⁸²³ - D⁸²⁴ - E⁸²⁵ - F⁸²⁶ - G⁸²⁷ - H⁸²⁸ - I⁸²⁹ - J⁸³⁰ - K⁸³¹ - L⁸³² - M⁸³³ - N⁸³⁴ - O⁸³⁵ -
 P⁸³⁶ - Q⁸³⁷ - R⁸³⁸ - S⁸³⁹ - T⁸⁴⁰ - U⁸⁴¹ - V⁸⁴² - W⁸⁴³ - X⁸⁴⁴ - Y⁸⁴⁵ - Z⁸⁴⁶

Befehl

Kurzbeschreibung

821 Kapitel 51.1 auf Seite 197
 822 Kapitel 51.2 auf Seite 198
 823 Kapitel 51.3 auf Seite 199
 824 Kapitel 51.4 auf Seite 201
 825 Kapitel 51.5 auf Seite 202
 826 Kapitel 51.6 auf Seite 203
 827 Kapitel 51.7 auf Seite 204
 828 Kapitel 51.8 auf Seite 205
 829 Kapitel 51.9 auf Seite 206
 830 Kapitel 51.10 auf Seite 208
 831 Kapitel 51.11 auf Seite 209
 832 Kapitel 51.12 auf Seite 210
 833 Kapitel 51.13 auf Seite 211
 834 Kapitel 51.14 auf Seite 213
 835 Kapitel 51.15 auf Seite 214
 836 Kapitel 51.16 auf Seite 215
 837 Kapitel 51.17 auf Seite 217
 838 Kapitel 51.18 auf Seite 218
 839 Kapitel 51.19 auf Seite 219
 840 Kapitel 51.20 auf Seite 222
 841 Kapitel 51.21 auf Seite 223
 842 Kapitel 51.22 auf Seite 224
 843 Kapitel 51.23 auf Seite 225
 844 Kapitel 51.24 auf Seite 226
 845 Kapitel 51.25 auf Seite 227
 846 Kapitel 51.26 auf Seite 228

51.26. Z

A⁸⁴⁷ - B⁸⁴⁸ - C⁸⁴⁹ - D⁸⁵⁰ - E⁸⁵¹ - F⁸⁵² - G⁸⁵³ - H⁸⁵⁴ - I⁸⁵⁵ - J⁸⁵⁶ - K⁸⁵⁷ - L⁸⁵⁸ - M⁸⁵⁹ - N⁸⁶⁰ - O⁸⁶¹ -
P⁸⁶² - Q⁸⁶³ - R⁸⁶⁴ - S⁸⁶⁵ - T⁸⁶⁶ - U⁸⁶⁷ - V⁸⁶⁸ - W⁸⁶⁹ - X⁸⁷⁰ - Y⁸⁷¹ - Z⁸⁷²

Befehl

Kurzbeschreibung

847	Kapitel 51.1 auf Seite 197
848	Kapitel 51.2 auf Seite 198
849	Kapitel 51.3 auf Seite 199
850	Kapitel 51.4 auf Seite 201
851	Kapitel 51.5 auf Seite 202
852	Kapitel 51.6 auf Seite 203
853	Kapitel 51.7 auf Seite 204
854	Kapitel 51.8 auf Seite 205
855	Kapitel 51.9 auf Seite 206
856	Kapitel 51.10 auf Seite 208
857	Kapitel 51.11 auf Seite 209
858	Kapitel 51.12 auf Seite 210
859	Kapitel 51.13 auf Seite 211
860	Kapitel 51.14 auf Seite 213
861	Kapitel 51.15 auf Seite 214
862	Kapitel 51.16 auf Seite 215
863	Kapitel 51.17 auf Seite 217
864	Kapitel 51.18 auf Seite 218
865	Kapitel 51.19 auf Seite 219
866	Kapitel 51.20 auf Seite 222
867	Kapitel 51.21 auf Seite 223
868	Kapitel 51.22 auf Seite 224
869	Kapitel 51.23 auf Seite 225
870	Kapitel 51.24 auf Seite 226
871	Kapitel 51.25 auf Seite 227
872	Kapitel 51.26 auf Seite 228

52. Lösung der Übungsaufgaben

53. Umgang mit Datensätzen (Erstellen, Auswählen und Filtern)

1. Lassen Sie sich die zweite und fünfte Zeile von `meinedaten` ausgeben.

```
meinedaten[c(2,5),]
  Name Geschlecht Lieblingsfarbe Einkommen
2  Caro weiblich          blau          800
5 Samira weiblich          gelb          899
```

2. Lassen Sie nur die Namen der Personen ausgeben, deren Lieblingsfarbe "gelb" ist.

Gesucht sind die Fälle, deren Lieblingsfarbe "gelb" ist. Die Namen stehen in der 1. Spalte:

```
> minedaten[meinedaten$Lieblingsfarbe=="gelb",1]
[1] "Lars" "Samira"
```

zum selben Ergebnis führt:

```
> minedaten[meinedaten$Lieblingsfarbe=="gelb", "Name"]
[1] "Lars" "Samira"
```

3. Welchen Namen und welches Geschlecht hat die Person mit dem niedrigsten Einkommen? (Benutzen sie hierzu die Funktion `rank(variable)`, die für eine gegebene Variable den Rang in einer aufsteigend geordneten Liste liefert.)

Gesucht ist also der Fall, dessen Einkommen in einer geordneten Liste den ersten Rang hat. Hier interessieren uns nur die Spalten/Variablen "Name" und "Geschlecht":

```
meinedaten[rank(meinedaten$Einkommen)==1, c("Name", "Geschlecht")]
  Name Geschlecht
2  Caro weiblich
```


54. Programmierbeispiele

Diese Seite sammelt die Programmierbeispiele für das Kapitel Programmieren mit R¹.

54.1. Beispiel 1: Abschlussnote

Eine (gedachte) Abschlussnote ergibt sich aus 3 Teilnoten. Hierbei fließen die ersten 2 Noten zu 30% - und die dritte Note zu 40% in die Abschlussnote ein. Wir programmieren uns also eine nette kleine Funktion, die uns die Abschlussnote aus den Teilnoten errechnet.

```
Abschlussnote <- function(x,y,z){  
  x.note <- (x/100)*30  
  y.note <- (y/100)*30  
  z.note <- (z/100)*40  
  abschluss <- x.note + y.note + z.note  
  cat3 ("Abschlussnote:", abschluss, "\n")  
}
```

Wir können die Funktion nun aufrufen per: `Abschlussnote(x, y, z)`, wobei `x`, `y`, `z` durch die jeweiligen Teilnoten ersetzt werden, z.B. so:

```
Abschlussnote(1.1, 1.7, 1.5)
```

Wir erhalten:

```
Abschlussnote: 1.44
```

54.2. Beispiel 2: Cut-Off-Points

Bestimmung des Cut-Off-Points eines Assessmentinstruments anhand von Sensitivität und Spezifität.

Übergeben werden muss der Funktion:

- ein Vektor `x`, welcher die einzelnen Summenwerte (des Assessmentinstruments) enthält
- ein Vektor `y`, welcher für den entsprechenden Summenwert angibt, ob ein Risiko vorliegt (bzw. Ereignis eintraf) oder nicht (z.B. "0 und 1" oder "j und n").

¹ Kapitel 41 auf Seite 145

- der Parameter `risk`, welcher angibt, wodurch die positive Gruppe im Vektor `y` repräsentiert wird (s.o., z.B. "0 oder 1" bzw. "j oder n")
- der Parameter `dir`, welcher anzeigt
 - ob ein höherer Summenwert (`x`) die Chance zur positiven Gruppenzugehörigkeit erhöht (`dir="GREATER"`)
 - ob ein niedrigerer Summenwert (`x`) die Chance zur positiven Gruppenzugehörigkeit erhöht (`dir="LESS"`)
- der Parameter `plot`, welcher per `TRUE` / `FALSE` angibt, ob eine Graphik ausgegeben werden soll, oder nicht.

```

sens.spec <- function4(x,y, risk=1, dir="LESS", plot=F) {

  frame <- data.frame5(x,y)
  var.min <- min6(na.omit7(x)) #
  welches ist der niedrigste Wert?
  var.max <- max8(na.omit9(x)) #
  welches ist der höchste Wert?
  dummy <- var.min

  cat10("\\r")
  cat("Minimum of value: ", var.min, "\\r")
  cat("Maximum of value: ", var.max, "\\r", "\\r")
  cat("Risk is coded with: ", risk, "\\r")

  if11 (tolower12(dir) %in% c("greater", "g")) {
    cat13("greater value means higher risk", "\\r", "\\r")
  }

  if14 (tolower15(dir) %in% c("less", "l")) {
    cat16("lesser value means higher risk", "\\r", "\\r")
  }

  sesp.table <- cbind17(999, 999, 999, 999, 999, 999, 999) #
  dient der Indizierung, wird später gelöscht (s.u.)
  while18(dummy <= var.max) {
    ### true/false positive/negative
    if19 (tolower20(dir) %in% c("less", "l")) {
      tp <- length21(frame$x[frame$x<=dummy &
frame$y==risk]) # true positive
      fp <- length(frame$x[frame$x<=dummy &
frame$y!=risk]) # false positive
      tn <- length(frame$x[frame$x>dummy &
frame$y!=risk]) # true negative
      fn <- length(frame$x[frame$x>dummy &
frame$y==risk]) # false negative
    }

    if22 (tolower23(dir) %in% c("greater", "g")) {
      tp <- length24(frame$x[frame$x>=dummy &
frame$y==risk]) # true positive
      fp <- length(frame$x[frame$x>=dummy &
frame$y!=risk]) # false positive
      tn <- length(frame$x[frame$x<dummy &
frame$y!=risk]) # true negative
      fn <- length(frame$x[frame$x<dummy &
frame$y==risk]) # false negative
    }

    sensi <- round25((tp / (tp+fn)), digits=3)
    # Sensitivität
    speci <- round((tn / (tn+fp)), digits=3)
  }
}

```

```

# Spezifität
sesp.table <- rbind26(sesp.table, c27(dummy, sensi, speci,
tp,fp,tn,fn))
dummy <- (dummy+1)
}

colnames28(sesp.table) <- c("Value", "Sensitivity", "Specificity", "tp",
"fp", "tn", "fn")
sesp.table <- seSP.table[-1,] # hier werden die "999"
gelöscht

if (plot==T) {
plot.table <- cbind29(sesp.table[,2], seSP.table[,3])
plot30(plot.table)
}

if (plot==F) {
print31(sesp.table)
cat("\r")
cat("Cut-Off-Points include positive cases", "\r")
cat("\r")
}
}
sens.spec(x, y) # Aufruf der Funktion

```

54.3. Beispiel 3: Entfernen von Umlauten

Diese Funktion entfernt störende Umlaute

```

noumlaute <- function32(variable)
{
## -
-----

## Funktion entfernt stoerende Umlaute, unten stehende Liste ggf.
erweitern

## -
-----

variable <- gsub33("ä", "ae", variable)
variable <- gsub("ü", "ue", variable)
variable <- gsub("ö", "oe", variable)
variable <- gsub("Û", "Ue", variable)
variable <- gsub("Ä", "Ae", variable)
variable <- gsub("Ö", "Oe", variable)
variable <- gsub("ß", "ss", variable)
return(variable)
}

```

54.4. Beispiel 4: Zeit Sampler

Diese Funktion erzeugt eine randomisierte Liste von je einem aller Wochentagen im Monat September.

```
randay<-function34(name1, name2)#name1 und name2 sind nur labels fuer den
Output
{
  start.date <- strptime35("2008/09/01","%Y/%m/%d")      # erzeugt das
  Startdaum, welches dem ersten Montag im Montag entspricht
  end.date <- strptime("2008/09/30","%Y/%m/%d")      # erzeugt das
  Enddatum, welches hier immer gleich ist
  MON <- seq(start.date, end.date, by="7 days") # erzeugt die
  Sequenz "vom Startdatum bis zum Enddatum, alle 7 Tage"
  start.date <- strptime("2008/09/02","%Y/%m/%d")      # und legt das
  Ergebnis in einem Object ab.
  end.date <- strptime("2008/09/30","%Y/%m/%d")
  TUE <- seq(start.date, end.date, by="7 days")
  start.date <- strptime("2008/09/03","%Y/%m/%d")
  end.date <- strptime("2008/09/30","%Y/%m/%d")
  WED <- seq(start.date, end.date, by="7 days")
  start.date <- strptime("2008/09/04","%Y/%m/%d")
  end.date <- strptime("2008/09/30","%Y/%m/%d")
  THU <- seq(start.date, end.date, by="7 days")
  start.date <- strptime("2008/09/05","%Y/%m/%d")
  end.date <- strptime("2008/09/30","%Y/%m/%d")
  FRI <- seq(start.date, end.date, by="7 days")
  start.date <- strptime("2008/09/06","%Y/%m/%d")
  end.date <- strptime("2008/09/30","%Y/%m/%d")
  SAT <- seq(start.date, end.date, by="7 days")
  start.date <- strptime("2008/09/07","%Y/%m/%d")
  end.date <- strptime("2008/09/30","%Y/%m/%d")
  SUN <- seq(start.date, end.date, by="7 days")
  a <- c(sample(MON,1), sample(TUE,1), s
  ample(WED,1), sample(THU,1), sample(FRI,1), sample(SAT,1), sample(SUN,1))
  # Sampling
  a <- sort(a)
  cat("Name 1:", name1, "Name 2:", name2, "\n", format(a, "%a
  %m/%d/%y"), "\n") }
```

54.5. siehe auch

- Programmieren mit R³⁶
- Anwendungsbeispiele³⁷

³⁶ Kapitel 41 auf Seite 145

³⁷ Kapitel 47 auf Seite 169

55. Beispielskripte

55.1. Importieren von SPSS-Datensätzen

Das folgende Skript verdeutlicht den Import von Variable Labels in SPSS, der sich mit read.spss etwas umständlich gestaltet.

```
## Zweck: Das Skript lädt eine SPSS-Datei incl. Kommentaren

## -
-----

## Verwendung: Pfad der SPSS-Datei und auszugebenden R-Datei in

## spssfile und rfile definieren und ausführen. Die exportierte
R-Datei

## ist dann im angegebenen Verzeichnis verfügbar und kann mit

## load ("/Pfad/zur/Datei/rdatei.r") in SPSS eingebunden werden

## Ggf. kann der Datensatzname "a" angepasst werden.

## -
-----

## Author: euro, Date: 30 May 2007, 21:43

# SPSS-Datei

spssfile <- "/Pfad/zur/Datei/spssdatei.sav"
# R-datei

rfile <- "/Pfad/zur/Datei/rdatei.r"
#####
#####

noumlaute <- function(variable)
{
  ## -
-----
```

```
## Funktion entfernt stoerende Umlaute, unten stehende Liste
ggf. erweitern

## -
-----

variable <- gsub("ä", "ae", variable)
variable <- gsub("ü", "ue", variable)
variable <- gsub("ö", "oe", variable)
variable <- gsub("ÿ", "Ue", variable)
variable <- gsub("Ä", "Ae", variable)
variable <- gsub("Ö", "Oe", variable)
variable <- gsub("ß", "ss", variable)
return(variable)
}
library(foreign)

## Sprachoptionen

## ggf. durch andere zu ersetzen

Sys.setlocale(locale="de_DE.ISO8859-15")

## eigentlicher Import

a.spss <- (read.spss(spssfile))
a <- as.data.frame(a.spss)

## Übernahme der Kommentare in jede Variable des Datensatz "a"

cat("importing names.")
for(i in 1:length(attr(a.spss, "variable.labels")))
{
  comment(a[,i]) <-
noumlaute(attr(a.spss, "variable.labels")[i])
cat(".")
}
cat(" ready/n")

## Speichern der Datei

save(a, file=rfile)
```

56. Übersicht programmspezifischer Voreinstellungen

Die Anweisung

```
unlist ( options () )
```

ermöglicht die übersichtliche Darstellung der programmspezifischen Voreinstellungen, z.B. so:

```

                                HTTPUserAgent
                                OutDec
"R (2.5.0 powerpc-apple-darwin8.9.1 powerpc darwin8.9.1) " "."
                                STERM
                                X11colortype
                                "iESS" "true"
                                X11fonts1
                                X11fonts2
"-adobe-helvetica-%s-%s-***%d-***-***-***-***" "-adobe-symbol-medium-r-***%d-***-***-***-***"
                                add.smooth
                                browser
                                "TRUE" "/usr/bin/open"
                                .
                                .
                                .

```

Mit der Anweisung

```
options ( VOREINSTELLUNGSPARAMETER = "NEUER WERT" )
```

können Voreinstellungen neu gesetzt werden.

Wenn z.B. das Zeichen "," statt dem standardmässigen "." für die Trennung von Kommawerten verwendet werden soll, kann dies mit folgender Anweisung verändert werden:

```
options ( OutDec = "," )
```

Folgende Optionen gibt es:

Parameter	Funktion	Voreinstellung
OutDec	Ausgabezeichen für Trennung von Kommazahlen.	."
STERM		
X11colortype		
X11fonts		
add.smooth		

browser	Kommando mit dem Html-Seiten und die Html-basierte Hilfe dargestellt wird.	
check.bounds		
continue		
contrasts		
device	Standard -> Graphik.	"X11"
digits		
dvipscmd		
echo		
editor		
encoding	Zeichenkodierung der Texteingabe.	
example.ask		
expressions		
help.try.all.packages		
internet.info		
keep.source		
keep.source.pkgs		
latexcmd		
locatorBell		
mailer	Standardprogramm für den Mailversand.	
max.print		
menu.graphics		
na.action		"na.omit"
pager		
papersize	Format für die Erstellung von druckfähigen Dokumenten. (bspw. "A4")	
par.ask.default		
pdfviewer	Standardprogramm für die Darstellung von PDF-Dokumenten.	
pkgType		
printcmd		
prompt	Zeichen zur Markierung der Eingabebereitschaft.	">"
repos		
show.coef.Pvalues		
show.error.messages		
show.signif.stars		
str		
stringsAsFactors		
timeout		
ts.S.compat		
ts.eps		
unzip	Anweisung zum automatischen Entpacken von ZIP-Dateien.	"/usr/bin/unzip" (Linux)
verbose		
warn		

warnings.length
width

Ausgabebreite in Zeichen / Zeile.

"1000"

57. Das wikibooks-Zusatzpaket

Für dieses Wikibook wurde ein eigenes Zusatzpaket¹ erstellt. Es beinhaltet die in diesem Buch verwendeten Funktionen und Datensätze.

Der Name des Pakets lautet `wikibooks`

57.1. Installation

Zur Installation wird folgender Befehl verwendet:

```
install.packages2("wikibooks")
```

Derzeit ist das Paket nur für die R-Version 2.5 vorkompiliert. Falls Sie eine ältere R-Version verwenden, müssen Sie das Paket manuell installieren:

1. Laden Sie das Paket [hier](#)³ herunter
2. Öffnen Sie eine Shell
3. Gehen Sie in das Verzeichnis, in welches Sie das Paket heruntergeladen haben und geben Sie ein:

```
R CMD INSTALL wikibooks
```

57.2. Paket aktivieren

Nachdem das Paket installiert ist, müssen Sie es per `library`⁴ ("`wikibooks`") aktivieren. Erst jetzt stehen die Funktionen und Datensätze zur Verfügung.

Um einen Datensatz aus dem Paket verwenden zu können, müssen Sie auch diesen erst aktivieren. Um beispielsweise auf den Bundesliga-Datensatz zugreifen zu können, geben Sie in die R-Kommandozeile ein:

```
data5(Bundesliga)
```

1 Kapitel 7 auf Seite 17

3 <http://www.cran.r-project.org/web/packages/wikibooks/index.html>

4 <http://de.wikibooks.org/wiki/GNU%20R%3A%20library>

57.3. Paketinhalt

57.3.1. Funktionen

Abschlussnote

Die Funktion `Abschlussnote` errechnet eine gedachte Abschlussnote, welche aus drei Prüfungsergebnissen zusammengesetzt wird (vgl. Programmierbeispiel 1⁶)

```
Abschlussnote(1.7, 2.4, 3.0)
```

Bundesliga.Mannschaft

Diese Funktion benötigt den `Bundesliga`⁷-Datensatz und sucht alle Spielpaarungen und Ergebnisse einer Mannschaft für eine spezifische bzw. für alle verfügbaren Saisons heraus. Die Standardeinstellungen lauten

```
Bundesliga.Mannschaft(Mannschaft, Saison = "all")
```

Um sich beispielsweise die Paarungen des FC Schalke 04 für alle Saisons anzeigen zu lassen, wird die Funktion aufgerufen per

```
Bundesliga.Mannschaft("FC Schalke 04")
```

Interessiert man sich nur für die Paarungen der Saison 1999/2000 lautet der Aufruf

```
Bundesliga.Mannschaft("FC Schalke 04", "1999/2000")
```

Bundesliga.Tabelle

Diese Funktion benötigt den `Bundesliga`⁸-Datensatz und zeigt u.a. die Tabellenstände einer bestimmten Saison zu einem bestimmten Spieltag an. Des weiteren lässt sich mit der Funktion die Platzierungen der Mannschaften im Saisonverlauf anzeigen. Der Standardaufruf lautet:

```
Bundesliga.Tabelle(Saison, Spieltag = 1, output = "Tabelle")
```

Hierbei gelten die Parameter:

6 Kapitel 54.5 auf Seite 236

7 Kapitel 57.3.2 auf Seite 246

8 Kapitel 57.3.2 auf Seite 246

- Saison = die gewünschte Saison, z.B. "1976/1977"
- Spieltag = der gewünschte Spieltag, z.B. 14
- output
 - "Tabelle" = die Tabelle am gewünschten Spieltag der gewünschten Saison
 - "Platzierung" = die Mannschaftsplatzierungen vom 1. bis zum gewünschten Spieltag

Wenn man die Tabelle des 5. Spieltags der Saison 2003/2004 angezeigt bekommen möchte, so lautet der Aufruf:

```
Bundesliga.Tabelle("2003/2004", 5)
```

Möchte man sehen, welche Platzierungen die teilnehmenden Mannschaften bis zum 8. Spieltag erzielten, so lautet der Aufruf:

```
Bundesliga.Tabelle("2006/2007", 8, "Platzierung")
```

Bundesliga.XML

Diese Funktion benötigt den `Bundesliga`⁹-Datensatz und erstellt eine XML-Datei der Spielpaarungen einer bestimmten bzw. aller verfügbaren Saisons. Die XML-Datei wird standardmäßig per `sink()`¹⁰ in die Datei `Bundesliga.xml` im Arbeitsverzeichnis geschrieben.

Der Standardaufruf lautet:

```
Bundesliga.XML(Datei = "Bundesliga.xml", Saison = "all")
```

Mit dieser Eingabe werden alle verfügbaren Paarungen in die Datei `Bundesliga.xml` im Arbeitsverzeichnis geschrieben. Möchte man lediglich die Saison 1968/1969 in der Datei "Saison1968-69.xml" abspeichern, so lautet der Aufruf:

```
Bundesliga.XML("Saison1968-69.xml", "1968/1969")
```

Eine mit der Funktion erzeugte XML-Datei aller Paarungen kann [hier](#)¹¹ gedownloadet werden.

sens.spec

Die Funktion `sens.spec` errechnet Sensitivitäts- und Spezifitätswerte eines Assessmentinstruments für alle möglichen Cut-Off-Punkte (vgl. Programmierbeispiel 2¹²).

9 Kapitel 57.3.2 auf Seite 246

10 <http://de.wikibooks.org/wiki/GNU%20R%3A%20sink>

11 <http://www.produnis.de/fTip/Bundesliga.xml.zip>

12 Kapitel 54.5 auf Seite 236

Für diese Funktion steht der Datensatz `cms`¹³ zur Verfügung.

```
data14(cms)
sens.spec(cms$ascore, cms$arisk, risk=1)
```

57.3.2. Datensätze

Bundesliga

Der Datensatz `Bundesliga` enthält alle Paarungen und Ergebnisse der Fußball-Bundesliga von 1963-2007. Um auf den Datensatz zugreifen zu können, muss folgender Befehl eingegeben werden:

```
library15("wikibooks")
data16(Bundesliga)
```

Für diesen Datensatz beinhaltet das `wikibooks`-Paket die Funktionen `Bundesliga.Tabelle`¹⁷, `Bundesliga.Mannschaft`¹⁸ und `Bundesliga.XML`¹⁹.

`cms`

Der Datensatz `cms` enthält 620 Datenfälle für die Funktion `sens.spec`²⁰. Der Datensatz besteht aus 2 Spalten:

`ascore` - Summenergebnisse eines Assessmentinstruments

`arisk` - Risikogruppenzuteilung: 0 = kein Risiko, 1 = Risiko

Die Aktivierung des Datensatzes erfolgt per:

```
data21(cms)
```

57.4. Weblinks

- CRAN-Seite des Pakets²²

13 Kapitel 57.3.2 auf Seite 246

17 Kapitel 57.3.1 auf Seite 244

18 Kapitel 57.3.1 auf Seite 244

19 Kapitel 57.3.1 auf Seite 245

20 Kapitel 57.3.1 auf Seite 245

22 <http://www.cran.r-project.org/web/packages/wikibooks/index.html>

57.5. siehe auch

- Eigene Zusatzpakete erstellen²³

²³ Kapitel 44 auf Seite 155

58. Autoren

Edits	User
1	Aksum ⁷ ¹
7	Albmont ²
1	AndréWilke ³
2	Ciciban ⁴
3	Conny ⁵
2	Daniel B ⁶
44	Dirk Huenniger ⁷
2	Don michele kurac ⁸
1	Duschenhocker ⁹
10	E^(nix) ¹⁰
202	Europol ¹¹
6	Faduci ¹²
1	Faron ¹³
1	Fhkade ¹⁴
1	Flowerfairy ¹⁵
2	Geekux ¹⁶
1	Gronau ¹⁷
43	Hagezussa ¹⁸
1	HansImGlueck ¹⁹
1	Hardy42 ²⁰
3	Jstein ²¹

1 http://de.wikibooks.org/w/index.php?title=Benutzer:Aksum_7
2 <http://de.wikibooks.org/w/index.php?title=Benutzer:Albmont>
3 <http://de.wikibooks.org/w/index.php?title=Benutzer:Andr%C3%A9Wilke>
4 <http://de.wikibooks.org/w/index.php?title=Benutzer:Ciciban>
5 <http://de.wikibooks.org/w/index.php?title=Benutzer:Conny>
6 http://de.wikibooks.org/w/index.php?title=Benutzer:Daniel_B
7 http://de.wikibooks.org/w/index.php?title=Benutzer:Dirk_Huenniger
8 http://de.wikibooks.org/w/index.php?title=Benutzer:Don_michele_kurac
9 <http://de.wikibooks.org/w/index.php?title=Benutzer:Duschenhocker>
10 <http://de.wikibooks.org/w/index.php?title=Benutzer:E%5E%28nix%29>
11 <http://de.wikibooks.org/w/index.php?title=Benutzer:Europol>
12 <http://de.wikibooks.org/w/index.php?title=Benutzer:Faduci>
13 <http://de.wikibooks.org/w/index.php?title=Benutzer:Faron>
14 <http://de.wikibooks.org/w/index.php?title=Benutzer:Fhkade>
15 <http://de.wikibooks.org/w/index.php?title=Benutzer:Flowerfairy>
16 <http://de.wikibooks.org/w/index.php?title=Benutzer:Geekux>
17 <http://de.wikibooks.org/w/index.php?title=Benutzer:Gronau>
18 <http://de.wikibooks.org/w/index.php?title=Benutzer:Hagezussa>
19 <http://de.wikibooks.org/w/index.php?title=Benutzer:HansImGlueck>
20 <http://de.wikibooks.org/w/index.php?title=Benutzer:Hardy42>
21 <http://de.wikibooks.org/w/index.php?title=Benutzer:Jstein>

- 2 Juetho²²
- 1 MM-Stat²³
- 5 MichaelFrey²⁴
- 26 MichaelFreyTool²⁵
- 5 Mickel²⁶
- 9 Nichtich²⁷
- 6 Nikocrow²⁸
- 19 Noreponse²⁹
- 1 Octanitrocuban³⁰
- 3 Philipendula³¹
- 400 Produnis³²
- 2 Qaswed³³
- 1 René Schwarz³⁴
- 1 Salatgurke³⁵
- 2 Sigbert³⁶
- 4 Skee³⁷
- 2 Student³⁸
- 6 Sutruban³⁹
- 45 Tfb785⁴⁰
- 1 Tomukas⁴¹
- 1 WissensDürster⁴²

-
- 22 <http://de.wikibooks.org/w/index.php?title=Benutzer:Juetho>
 - 23 <http://de.wikibooks.org/w/index.php?title=Benutzer:MM-Stat>
 - 24 <http://de.wikibooks.org/w/index.php?title=Benutzer:MichaelFrey>
 - 25 <http://de.wikibooks.org/w/index.php?title=Benutzer:MichaelFreyTool>
 - 26 <http://de.wikibooks.org/w/index.php?title=Benutzer:Mickel>
 - 27 <http://de.wikibooks.org/w/index.php?title=Benutzer:Nichtich>
 - 28 <http://de.wikibooks.org/w/index.php?title=Benutzer:Nikocrow>
 - 29 <http://de.wikibooks.org/w/index.php?title=Benutzer:Noreponse>
 - 30 <http://de.wikibooks.org/w/index.php?title=Benutzer:Octanitrocuban>
 - 31 <http://de.wikibooks.org/w/index.php?title=Benutzer:Philipendula>
 - 32 <http://de.wikibooks.org/w/index.php?title=Benutzer:Produnis>
 - 33 <http://de.wikibooks.org/w/index.php?title=Benutzer:Qaswed>
 - 34 http://de.wikibooks.org/w/index.php?title=Benutzer:Ren%C3%A9_Schwarz
 - 35 <http://de.wikibooks.org/w/index.php?title=Benutzer:Salatgurke>
 - 36 <http://de.wikibooks.org/w/index.php?title=Benutzer:Sigbert>
 - 37 <http://de.wikibooks.org/w/index.php?title=Benutzer:Skee>
 - 38 <http://de.wikibooks.org/w/index.php?title=Benutzer:Student>
 - 39 <http://de.wikibooks.org/w/index.php?title=Benutzer:Sutruban>
 - 40 <http://de.wikibooks.org/w/index.php?title=Benutzer:Tfb785>
 - 41 <http://de.wikibooks.org/w/index.php?title=Benutzer:Tomukas>
 - 42 <http://de.wikibooks.org/w/index.php?title=Benutzer:WissensD%C3%BCrster>

Abbildungsverzeichnis

- GFDL: Gnu Free Documentation License. <http://www.gnu.org/licenses/fdl.html>
- cc-by-sa-3.0: Creative Commons Attribution ShareAlike 3.0 License. <http://creativecommons.org/licenses/by-sa/3.0/>
- cc-by-sa-2.5: Creative Commons Attribution ShareAlike 2.5 License. <http://creativecommons.org/licenses/by-sa/2.5/>
- cc-by-sa-2.0: Creative Commons Attribution ShareAlike 2.0 License. <http://creativecommons.org/licenses/by-sa/2.0/>
- cc-by-sa-1.0: Creative Commons Attribution ShareAlike 1.0 License. <http://creativecommons.org/licenses/by-sa/1.0/>
- cc-by-2.0: Creative Commons Attribution 2.0 License. <http://creativecommons.org/licenses/by/2.0/>
- cc-by-2.0: Creative Commons Attribution 2.0 License. <http://creativecommons.org/licenses/by/2.0/deed.en>
- cc-by-2.5: Creative Commons Attribution 2.5 License. <http://creativecommons.org/licenses/by/2.5/deed.en>
- cc-by-3.0: Creative Commons Attribution 3.0 License. <http://creativecommons.org/licenses/by/3.0/deed.en>
- GPL: GNU General Public License. <http://www.gnu.org/licenses/gpl-2.0.txt>
- LGPL: GNU Lesser General Public License. <http://www.gnu.org/licenses/lgpl.html>
- PD: This image is in the public domain.
- ATTR: The copyright holder of this file allows anyone to use it for any purpose, provided that the copyright holder is properly attributed. Redistribution, derivative work, commercial use, and all other use is permitted.
- EURO: This is the common (reverse) face of a euro coin. The copyright on the design of the common face of the euro coins belongs to the European Commission. Authorised is reproduction in a format without relief (drawings, paintings, films) provided they are not detrimental to the image of the euro.
- LFK: Lizenz Freie Kunst. <http://artlibre.org/licence/lal/de>
- CFR: Copyright free use.

- EPL: Eclipse Public License. <http://www.eclipse.org/org/documents/epl-v10.php>

Copies of the GPL, the LGPL as well as a GFDL are included in chapter Licenses⁴³. Please note that images in the public domain do not require attribution. You may click on the image numbers in the following table to open the webpage of the images in your webbrowser.

43 Kapitel 59 auf Seite 255

1		GFDL
2	Europol ⁴⁴	GFDL
3		GFDL
4	MyName (Produnis ⁴⁵ (talk ⁴⁶))	PD
5	MyName (Produnis ⁴⁷ (talk ⁴⁸))	PD
6	MyName (Produnis ⁴⁹ (talk ⁵⁰))	PD
7		GFDL
8	Thomas Steiner ⁵¹	cc-by-sa-2.5
9	Thomas Steiner ⁵²	cc-by-sa-2.5
10	René Schwarz ⁵³	GFDL
11	Thomas Steiner ⁵⁴	GFDL
12	Produnis ⁵⁵	GFDL

44 <http://de.wikibooks.org/wiki/User%3AEuropol>

45 <http://de.wikibooks.org/wiki/User%3AProdunis>

46 <http://de.wikibooks.org/wiki/User%20talk%3AProdunis>

47 <http://de.wikibooks.org/wiki/User%3AProdunis>

48 <http://de.wikibooks.org/wiki/User%20talk%3AProdunis>

49 <http://de.wikibooks.org/wiki/User%3AProdunis>

50 <http://de.wikibooks.org/wiki/User%20talk%3AProdunis>

51 <http://de.wikibooks.org/wiki/%3Ade%3ABenutzer%3AThire>

52 <http://de.wikibooks.org/wiki/%3Ade%3ABenutzer%3AThire>

53 <http://de.wikibooks.org/wiki/User%3ARen%E9%20Schwarz>

54 <http://de.wikibooks.org/wiki/%3Ade%3ABenutzer%3AThire>

55 <http://de.wikibooks.org/wiki/User%3AProdunis>

59. Licenses

59.1. GNU GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program—to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps: (1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents can not be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow. TERMS AND CONDITIONS 0. Definitions.

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each license is addressed as "you", "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion. 1. Source Code.

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run

the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work. 2. Basic Permissions.

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable and exclusive; the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary. 3. Protecting Users' Legal Rights From Anti-Circumvention Law.

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures. 4. Conveying Verbatim Copies.

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee. 5. Conveying Modified Source Versions.

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

* a) The work must carry prominent notices stating that you modified it, and giving a relevant date. * b) The work must carry prominent notices stating that it is released under this License and any conditions added under section 7. This requirement modifies the requirement in section 4 to "keep intact all notices". * c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it. * d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume or a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate. 6. Conveying Non-Source Forms.

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

* a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange. * b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge. * c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you convey the object code with such an offer, in accord with subsection 6b. * d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source. Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements. * e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects to use, or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified versions of a covered work that is User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying. 7. Additional Terms.

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

* a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or * b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or * c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or * d) Limiting the use of that material for publicity purposes of names of licensors or authors of the material; or * e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or * f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way. 8. Termination.

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.9. Acceptance Not Required for Having Copies.

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants

you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so. 10. Automatic Licensing of Downstream Recipients.

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it. 11. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by or as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deposit yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law. 12. No Surrender of Others' Freedom.

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program. 13. Use with the GNU Affero General Public License.

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such. 14. Revised Versions of this License.

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.

If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of

acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version. 15. Disclaimer of Warranty.

THESE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. 16. Limitation of Liability.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PRO-

59.2. GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed. 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference. 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains in its source the copyright holder's saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A Secondary Section is named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The Invariant Sections are certain Secondary Sections whose titles are designated in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, presented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text nearest the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section Entitled XYZ means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that

GRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. 17. Interpretation of Sections 15 and 16.

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

translates XYZ into another language. (Here XYZ stands for a specific section name mentioned below, such as Acknowledgements, "Dedications", Endorsements, or "History"). To "Preserve the Title" of a section XYZ when you modify the Document means that it remains a section Entitled XYZ according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties; any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License. 2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies. 3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document. 4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

* A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission. * B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement. * C. State on the Title Page the name of the publisher of the Modified Version, as the publisher. * D. Preserve all the copyright notices of the Document. * E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices. * F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below. * G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice. * H. Include an unaltered copy of this License. * I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous section. * J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions if they were based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission. * K. For any section Entitled Acknowledgements or "Dedications", Preserve the Title of the section, and preserve in the section

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file but most effectively state the extension of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

<one line to give the program's name and a brief idea of what it does.> Copyright (C) <year> <name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <<http://www.gnu.org/licenses/>>.

Also add information on how to contact you by electronic and paper mail.

all the substance and tone of each of the contributor acknowledgements and/or dedications given therein. * L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles. * M. Delete any section Entitled Endorsements. Such a section may not be included in the Modified Version. * N. Do not retile any existing section to be Entitled Endorsements to conflict in title with any Invariant Section. * O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled Endorsements, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. You may add a passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version. 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled Acknowledgements, and any sections Entitled "Dedications". You must delete all sections Entitled Endorsements. 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document. 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an aggregate if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate. 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

If the program does terminal interaction, make it output a short notice like this when it starts in an interactive mode:

<program> Copyright (C) <year> <name of author> This program comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, your program's commands might be different; for a GUI interface, you would use an "about box".

You should also get your employer (if you work as a programmer) or school, if any, to sign a "copyright disclaimer" for the program, if necessary. For more information on this, and how to apply and follow the GNU GPL, see <<http://www.gnu.org/licenses/>>.

The GNU General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Lesser General Public License instead of this License. But first, please read <<http://www.gnu.org/philosophy/why-not-lgpl.html>>.

If a section in the Document is Entitled Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title. 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it. 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License or any later version applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document. 11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is eligible for relicensing if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing. ADDENDUM: How to use this License for your documents

To use this license in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (C) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation, with Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with ... Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

59.3. GNU Lesser General Public License

GNU LESSER GENERAL PUBLIC LICENSE

Version 3, 29 June 2007

Copyright © 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below. 0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The “Corresponding Application Code” for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work. 1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL. 2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

* a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or * b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

The object code form of an Application may incorporate material from a header file that is part of the Library. You may convey such object code under

terms of your choice, provided that, if the incorporated material is not limited to numerical parameters, data structure layouts and accessors, or small macros, inline functions and templates (ten or fewer lines in length), you do both of the following:

* a) Give prominent notice with each copy of the object code that the Library is used in it and that the Library and its use are covered by this License. * b) Accompany the object code with a copy of the GNU GPL and this license document.

4. Combined Works.

You may convey a Combined Work under terms of your choice that, taken together, effectively do not restrict modification of the portions of the Library contained in the Combined Work, and reverse engineering for debugging such modifications, if you also do each of the following:

* a) Give prominent notice with each copy of the Combined Work that the Library is used in it and that the Library and its use are covered by this License. * b) Accompany the Combined Work with a copy of the GNU GPL and this license document. * c) For a Combined Work that displays copyright notices during execution, include the copyright notice for the Library among these notices, as well as a reference directing the user to the copies of the GNU GPL and this license document. * d) Do one of the following: o 0) Convey the Minimal Corresponding Source under the terms of this License, and the Corresponding Application Code in a form suitable for, and

under terms that permit, the user to recombine or relink the Application with a modified version of the Linked Version to produce a modified Combined Work, in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source. o 1) Use a suitable shared library mechanism for linking with the Library. A suitable mechanism is one that (a) uses at run time a copy of the Library already present on the user's computer system, and (b) will operate properly with a modified version of the Library that is interface-compatible with the Linked Version. * e) Provide Installation Information, but only if you would otherwise be required to provide such information under section 6 of the GNU GPL, and only to the extent that such information is necessary to install and execute a modified version of the Combined Work produced by recombining or relinking the Application with a modified version of the Linked Version. (If you use option 4d0, the Installation Information must accompany the Minimal Corresponding Source and Corresponding Application Code. If you use option 4d1, you must provide the Installation Information in the manner specified by section 6 of the GNU GPL for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

* a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License. * b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License “or any later version” applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.