

Contents

1

Foreword	vii	2
List of Figures	xxvii	3
List of Tables	xxxii	4
Mathematical Notations	xxxiii	5
Part I Preliminaries		6
1 Introducing R	3	7
1.1 Presentation of the Software	3	8
1.1.1 Origins	3	9
1.1.2 Why Use R?	3	10
1.2 R and Statistics	5	11
1.3 R and Plots	5	12
1.4 The R Graphical User Interface	7	13
1.5 First Steps in R	7	14
1.5.1 Using RCommander	7	15
1.5.1.1 Launching RCommander	8	16
1.5.1.2 Handling Data with RCommander	8	17
1.5.1.3 A Few Statistical Tasks with RCommander ...	13	18
1.5.1.4 Adding Functionalities to the RCommander Interface	18	20
1.5.2 Using R with the Console	19	21
1.5.2.1 The Strength of R Shown on an Example ...	19	22
1.5.2.2 A Brief Introduction of R Syntax Through Some Instructions to Type	23	23
2 A Few Data Sets and Research Questions	29	25
2.1 Body Mass Index of Children	29	26
2.2 Weight at Birth	30	27

2.3	Intima–Media Thickness	31	28
2.4	Diet of Elderly People	32	29
2.5	Study Case of Myocardial Infarction	33	30
2.6	Summary Table of Use of Data Sets	33	31
Part II The Bases of R			32
3	Basic Concepts and Data Organisation	37	33
3.1	Your First Session	37	34
3.1.1	R Is a Calculator	38	35
3.1.2	Displaying Results and Variable Redirecting	39	36
3.1.3	Work Strategy	40	37
3.1.4	Using Functions	43	38
3.2	Data in R	46	39
3.2.1	Data Nature (or Type, or Mode)	46	40
3.2.1.1	Numeric Type (<code>numeric</code>)	46	41
3.2.1.2	† Complex Type (<code>complex</code>)	47	42
3.2.1.3	Boolean or Logical Type (<code>logical</code>)	48	43
3.2.1.4	Missing Data (<code>NA</code>)	48	44
3.2.1.5	Character String Type (<code>character</code>)	49	45
3.2.1.6	† Raw Data (<code>raw</code>)	50	46
	Summary	50	47
3.2.2	Data Structures	50	48
3.2.2.1	Vectors (<code>vector</code>)	51	49
3.2.2.2	Matrices (<code>matrix</code>) and Arrays (<code>array</code>)	52	50
3.2.2.3	Lists (<code>list</code>)	53	51
3.2.2.4	The Individual×Variable Table (<code>data.frame</code>)	55	53
3.2.2.5	Factors (<code>factor</code>), Ordinal Variables (<code>ordered</code>)	56	55
3.2.2.6	Dates	57	56
3.2.2.7	Time Series	57	57
	Summary	58	58
	Memorandum	59	59
	Exercises	59	60
	Worksheet	60	61
4	Importing, Exporting and Producing Data	63	62
4.1	Importing Data	63	63
4.1.1	Importing Data from an ASCII Text File	63	64
4.1.1.1	Reading Data with <code>read.table()</code>	64	65
4.1.1.2	Reading Data with <code>read.ftable()</code>	67	66
4.1.1.3	Reading Data with the Function <code>scan()</code>	68	67

4.1.2	Importing Data from Excel or the Open Office Spreadsheet	68	69
4.1.2.1	Copy-Pasting	69	70
4.1.2.2	Using an Intermediary ASCII File	70	71
4.1.2.3	Using Specialized Packages	70	72
4.1.3	Importing Data from SPSS, Minitab, SAS or Matlab	70	73
4.1.4	Large Data Files	71	74
4.2	Exporting Data	72	75
4.2.1	Exporting Data to an ASCII Text File	72	76
4.2.2	Exporting Data to Excel or OpenOffice Calc	72	77
4.3	Creating Data	73	78
4.3.1	Entering Toy Data	73	79
4.3.2	Generating Pseudo-Random Numbers	74	80
4.3.3	Entering Data from a Hard Copy	74	81
4.4	† Reading/Writing in Databases	76	82
4.4.1	Creating a Database and a Table	76	83
4.4.2	Creating a Data Source Compatible with MySQL	76	84
4.4.3	Writing in a Table	78	85
4.4.4	Reading a Table	79	86
	Memorandum	80	87
	Exercises	80	88
	Worksheet	81	89
5	Data Manipulation, Functions	85	90
5.1	Operations on Vectors, Matrices and Lists	85	91
5.1.1	Vector Arithmetic	85	92
5.1.2	Recycling	86	93
5.1.3	Basic Functions	87	94
5.1.4	Operations on Matrices and Data.Frames	88	95
5.1.4.1	Information on Architecture	88	96
5.1.4.2	Merging Tables	89	97
5.1.4.3	The Function <code>apply()</code>	93	98
5.1.4.4	The Function <code>sweep()</code>	94	99
5.1.4.5	The Function <code>stack()</code>	94	100
5.1.4.6	The Function <code>aggregate()</code>	95	101
5.1.4.7	The Function <code>transform()</code>	95	102
5.1.5	Operations on Lists	96	103
5.2	Logical and Relational Operations	97	104
5.3	Operations on Sets	98	105
5.4	Extracting and Inserting Elements	99	106
5.4.1	Extracting from/Inserting into Vectors	100	107
5.4.2	Extracting from/Inserting into Matrices	102	108
5.4.3	Extracting from/Inserting into Arrays	106	109
5.4.4	Extracting from/Inserting into Lists	106	110
5.5	Manipulating Character Strings	108	111

5.6	Manipulating Dates and Time Units	111	112
5.6.1	Displaying the Current Date	111	113
5.6.2	Extracting Dates	112	114
5.6.3	Operations on Dates	113	115
5.7	Control Flow	115	116
5.7.1	Conditional Instructions	116	117
5.7.2	Loop Instructions	118	118
5.8	Creating Functions	120	119
5.9	† Fixed-Point and Floating Point Number Representation	127	120
5.9.1	Representing a Number in a Base	127	121
5.9.2	Floating Point Representations	128	122
5.9.2.1	Definitions	128	123
5.9.2.2	Limitations of This Representation due to the Significand	129	124 125
5.9.2.3	Avoiding Some Numerical Pitfalls	130	126
5.9.2.4	Limitations of This Representation due to the Exponent	132	127 128
	Memorandum	134	129
	Exercises	134	130
	Worksheet	136	131
6	R and Its Documentation	141	132
6.1	Integrated Help	141	133
6.1.1	The Command <code>help()</code>	141	134
6.1.2	Some Complementary Commands	143	135
6.2	† Help on the Web	145	136
6.2.1	Search Engines	145	137
6.2.2	Message Boards	146	138
6.2.3	Mailing Lists	146	139
6.2.4	Internet Relay Chat (IRC)	147	140
6.2.5	<i>Wiki</i>	147	141
6.3	† Literature About R	147	142
6.3.1	Online	147	143
6.3.2	Printed Material	148	144
	Memorandum	149	145
	Exercises	149	146
	Worksheet	149	147
7	Drawing Curves and Plots	151	148
7.1	Graphics Windows	151	149
7.1.1	Basic Graphics Windows, Manipulation and Saving ...	151	150
7.1.2	Splitting the Graphics Window: <code>layout()</code>	153	151
7.2	Low-Level Drawing Functions	156	152
7.2.1	The Functions <code>plot()</code> and <code>points()</code>	156	153
7.2.2	The Functions <code>segments()</code> , <code>lines()</code> and <code>abline()</code>	158	154 155

7.2.3	The Function <code>arrows()</code>	160	156
7.2.4	The Function <code>polygon()</code>	161	157
7.2.5	The Function <code>curve()</code>	162	158
7.2.6	The Function <code>box()</code>	162	159
7.3	Managing Colours	163	160
7.3.1	The Function <code>colors()</code>	163	161
7.3.2	Hexadecimal Colour Coding	164	162
7.3.3	The Function <code>image()</code>	166	163
7.4	Adding Text	169	164
7.4.1	The Function <code>text()</code>	169	165
7.4.2	The Function <code>mtext()</code>	170	166
7.5	Titles, Axes and Captions	171	167
7.5.1	The Function <code>title()</code>	171	168
7.5.2	The Function <code>axis()</code>	172	169
7.5.3	The Function <code>legend()</code>	173	170
7.6	Interacting with the Plot	175	171
7.6.1	The Function <code>locator()</code>	175	172
7.6.2	The Function <code>identify()</code>	175	173
7.7	† Fine-Tuning Graphical Parameters: <code>par()</code>	176	174
7.8	† Advanced Plots: <code>rgl</code> , <code>lattice</code> and <code>ggplot2</code>	187	175
	Memorandum	188	176
	Exercises	188	177
	Worksheet	189	178
8	Programming in R	193	179
8.1	Preamble	193	180
8.2	Developing Functions	194	181
8.2.1	Quick Start: Declaring, Creating and Calling Functions	194	182
8.2.2	Basic Concepts on Functions	195	184
8.2.2.1	Body of a Function	195	185
8.2.2.2	List of Formal and Effective Arguments	195	186
8.2.2.3	Object Returned by a Function	198	187
8.2.2.4	Variable Scope in the Body of a Function	200	188
8.2.3	Application to the Practical Problem	202	189
8.2.4	Operators	202	190
8.2.5	R Seen as a Functional Language	204	191
8.3	† Object-Oriented Programming	204	192
8.3.1	How the Internal Object-Oriented Mechanism Works ..	205	193
8.3.1.1	Class of an Object and Declaring an Object ..	205	194
8.3.1.2	Declaring Objects and Using Methods	206	195
8.3.2	Back to the Practical Problem	209	196
8.3.3	Information About Methods	211	197
8.3.4	Inheriting Classes	213	198

8.4	† Going Further in R Programming	216	199
8.4.1	R Attributes	216	200
8.4.1.1	Attribute class	218	201
8.4.1.2	Attribute dim	218	202
8.4.1.3	Attributes names and dimnames	221	203
8.4.2	Other R Objects	224	204
8.4.2.1	R Expressions	224	205
8.4.2.2	R Formulae	226	206
8.4.2.3	The R Environment	228	207
8.5	† Interfacing R and C/C++ or Fortran	230	208
8.5.1	Creating and Running a C/C++ or Fortran Function ..	231	209
8.5.2	Calling C/C++ (or Fortran) from R	237	210
8.5.3	Calling External C/C++ or Fortran Libraries	242	211
8.5.3.1	The R API	243	212
8.5.3.2	The newmat Library	246	213
8.5.3.3	The BLAS and LAPACK Packages	248	214
8.5.3.4	Mixing C/C++ and Fortran Packages	250	215
8.5.4	Calling R Code from a C/C++ Program Called by R ...	252	216
8.5.5	Calling R Code from Fortran	255	217
8.5.6	Some Useful Functions	255	218
8.6	† Debugging Functions	255	219
8.6.1	Debugging Functions in Pure R	255	220
8.6.2	Error in R Code	257	221
8.6.3	Error in the C/C++ or Fortran Code	258	222
8.6.4	Debugging with GDB	259	223
8.6.4.1	Debugging with Emacs	262	224
8.6.4.2	Debugging with DDD	264	225
8.6.4.3	Debugging with Insight	265	226
8.6.4.4	Detecting Memory Leaks	270	227
8.7	Parallel Computing and Computation on Graphical Cards	273	228
8.7.1	Parallel Computing	273	229
8.7.2	Computation on Graphical Cards	274	230
	Memorandum	276	231
	Exercises	276	232
	Worksheet	278	233
9	Managing Sessions	283	234
9.1	R Commands, Objects and Storage	283	235
9.2	Workspace: .RData Files	285	236
9.3	Command History: .Rhistory Files	287	237
9.4	Saving Plots	288	238
9.5	Managing Packages	290	239
9.6	Managing Access Paths to R Objects	290	240
9.7	† Other Useful Commands	292	241

9.8	† Problems in Memory Management	293	242
9.8.1	Organization of RAM	293	243
9.8.2	Accessing the Memory	294	244
	9.8.2.1 Problems Caused by Memory Management of Integers		245
	9.8.2.2 Successive Allocation of Memory	295	246
9.8.3	Object Size in R	296	247
9.8.4	Total Memory used by R	298	248
9.8.5	A Few Recommendations	299	249
9.9	† Using R in BATCH Mode	301	250
9.10	† Creating a Simple R Package	302	251
	Memorandum	303	252
	Exercises	306	253
	Worksheet	306	254
		307	255

Part III Elementary Mathematics and Statistics 256

10	Basic Mathematics: Matrix Operations, Integration, Optimization		257
10.1	Basic Mathematical Functions	313	258
10.2	Matrix Operations	313	259
	10.2.1 Basic Matrix Operations	315	260
	10.2.2 Outer Product	316	261
	10.2.3 Kronecker Product	318	262
	10.2.4 Triangular Matrices	319	263
	10.2.5 Operators vec and Half vec	319	264
	10.2.6 Determinant, Trace and Condition Number	320	265
	10.2.7 Scaling and Centring Data	320	266
	10.2.8 Eigenvalues and Eigenvectors	321	267
	10.2.9 Square Root of a Hermitian Positive-Definite Matrix ...	321	268
	10.2.10 Singular Value Decomposition	322	269
	10.2.11 Cholesky Decomposition	323	270
	10.2.12 QR Decomposition	323	271
10.3	Numerical Integration	324	272
10.4	Differentiation	325	273
	10.4.1 Symbolic Differentiation	326	274
	10.4.2 Numerical Differentiation	326	275
	10.5 Optimization	327	276
	10.5.1 Optimization Functions	327	277
	10.5.2 Roots of a Function	327	278
	Memorandum	331	279
	Exercises	333	280
	Worksheet	333	281
		334	282

11 Descriptive Statistics	339	283
11.1 Introduction	339	284
11.2 Structuring Variables According to Type	340	285
11.2.1 Structuring Qualitative Variables	341	286
11.2.2 Structuring Ordinal Variables	342	287
11.2.3 Structuring Discrete Quantitative Data	342	288
11.2.4 Structuring Continuous Quantitative Variables	343	289
11.3 Data Tables	343	290
11.3.1 Individual Data Tables	343	291
11.3.2 Tables of Counts and Frequency Tables	343	292
11.3.3 Tables of Grouped Data	344	293
11.3.4 Cross Tabulation	344	294
11.3.4.1 Contingency Tables	344	295
11.3.4.2 Joint Distribution	345	296
11.3.4.3 Marginal Distributions	346	297
11.3.4.4 Conditional Distributions	346	298
11.4 Numerical Summaries	347	299
11.4.1 Summaries of the Location of a Distribution	348	300
11.4.1.1 Modes	348	301
11.4.1.2 Median	348	302
11.4.1.3 Mean	350	303
11.4.1.4 Quantiles	350	304
11.4.2 Summaries of the Dispersion of a Distribution	350	305
11.4.3 Summaries of the Shape of a Distribution	351	306
11.5 Measures of Association	352	307
11.5.1 Measures of Association Between Two Qualitative Variables	352	308
11.5.1.1 Pearson's χ^2 Statistic	352	310
11.5.1.2 ϕ^2 , Cramér's V and Pearson's Contingency Coefficient	353	311
11.5.2 Measures of Association Between Ordinal Variables (or Ranks)	354	314
11.5.2.1 Kendall's τ and τ_b	354	315
11.5.2.2 Spearman's Rank Correlation Coefficient ρ ...	355	316
11.5.3 Measures of Association Between Two Quantitative Variables	355	317
11.5.3.1 Covariance and Pearson's Correlation Coefficient	355	319
11.5.4 Measures of Association Between a Quantitative Variable and a Qualitative Variable	356	321
11.5.4.1 Correlation Ratio $\eta^2_{Y X}$	356	322
11.6 Graphical Representations	357	324
11.6.1 Plotting Qualitative Variables	357	325
11.6.1.1 Cross Chart	357	326
11.6.1.2 Bar Charts	359	327

11.6.1.3	Pareto Chart	360	328
11.6.1.4	Stacked Bar Chart	361	329
11.6.1.5	Pie Chart	361	330
11.6.2	Plotting Ordinal Variables	362	331
11.6.2.1	Bar Chart with Cumulative Frequencies Line	362	332 333
11.6.3	Plotting Discrete Quantitative Variables	363	334
11.6.3.1	Cross Chart	363	335
11.6.3.2	Bar Chart	363	336
11.6.3.3	Plotting the Empirical Distribution Function	363	337
11.6.3.4	Stemplot	365	338
11.6.3.5	Boxplot	365	339
11.6.4	Plotting Continuous Quantitative Variables	367	340
11.6.4.1	Empirical Distribution Function	367	341
11.6.4.2	Stemplot	367	342
11.6.4.3	Boxplots	368	343
11.6.4.4	Density Histogram with Identical or Different Class Ranges	368	344 345
11.6.4.5	Frequency Polygon	369	346
11.6.4.6	Cumulative Frequency Polygon	370	347
11.6.5	Graphical Representations in a Bivariate Setting	371	348
11.6.5.1	Two-Way Plots for Two Qualitative Variables	371	349
11.6.5.2	Two-way Plots for Two Quantitative Variables	374	350 351
11.6.5.3	Two-Way Plots for One Qualitative and One Quantitative Variable	375	352 353
	Memorandum	377	354
	Exercises	377	355
	Worksheet	378	356
12	A Better Understanding of Random Variables, Distributions and Simulations Using R Specificities	381	358
12.1	Notions on Random Number Generation	381	359
12.2	The Notion of Random Variables	383	360
12.2.1	Realizations of a Random Variable and Functioning Law	383	361 362
12.2.2	I.i.d. Random Variables	384	363
12.2.3	Characterizing the Distribution of a Random Variable	385	364
12.2.3.1	Density Function, Distribution Function and Quantile Function	387	365 366
12.2.4	Parameters of the Distribution of a Random Distribution	390	367
12.3	Law of Large Numbers and Central Limit Theorem	392	368
12.3.1	Law of Large Numbers	392	369
12.3.2	Central Limit Theorem	393	370

12.4 Inferential Statistics	394	371
12.4.1 Point Estimate of Parameters	394	372
12.4.2 Empirical Cumulative Distribution Function	396	373
12.4.3 Maximum Likelihood Estimation	397	374
12.4.4 Sampling Variation and Properties of an Estimator	399	375
12.5 A Few Techniques to Draw from a Distribution	401	376
12.5.1 Simulating from Another Distribution	401	377
12.5.2 Inverse Transform Method	402	378
12.5.3 Rejection Sampling	402	379
12.5.4 Simulation of Discrete Random Variables	403	380
12.6 Bootstrap	404	381
12.7 Standard and Less Standard Distributions	405	382
12.7.1 Standard Distributions	405	383
12.7.2 † Less Standard Distributions	408	384
12.8 Modelling a Phenomenon	410	385
Memorandum	413	386
Exercises	413	387
Worksheet	413	388
13 Confidence Intervals and Hypothesis Testing	417	389
13.1 Notations	417	390
13.2 Confidence Intervals	418	391
13.2.1 Confidence Intervals for the Mean	418	392
13.2.2 Confidence Intervals for a Proportion p	419	393
13.2.3 Confidence Intervals for a Variance	421	394
13.2.4 Confidence Intervals for a Median	422	395
13.2.5 Confidence Intervals for a Correlation Coefficient	423	396
13.2.6 Summary Table for Confidence Intervals	424	397
13.3 Standard Hypothesis Testing	424	398
13.3.1 Parametric Tests	426	399
13.3.1.1 Tests of the Mean	426	400
13.3.1.2 Tests of Variance	429	401
13.3.1.3 Tests of Proportion	431	402
13.3.1.4 Tests of Correlation	433	403
13.3.2 Independence Tests	435	404
13.3.2.1 χ^2 Test for Independence	435	405
13.3.2.2 Yates' χ^2 Test	437	406
13.3.2.3 Fisher's Exact Test	438	407
13.3.3 Non-parametric Tests	439	408
13.3.3.1 Goodness-of-Fit Tests	439	409
13.3.3.2 Tests of Position	442	410
13.3.4 Memorandum of Standard Tests	447	411
13.4 Other Tests	447	412
Memorandum	449	413
Exercises	449	414
Worksheet	449	415

14	Simple and Multiple Linear Regression	455	416
14.1	Introduction	455	417
14.2	Simple Linear Regression	456	418
14.2.1	Aim and Model	456	419
14.2.2	Fitting Data	457	420
14.2.3	Confidence and Prediction Intervals for a New Value ..	461	421
14.2.4	Analysis of Residuals	463	422
14.2.5	Student's Tests for Means and Linear Model	466	423
14.2.6	Summary	468	424
14.3	Multiple Linear Regression	468	425
14.3.1	Aim and Model	468	426
14.3.2	Fitting Data	469	427
14.3.3	Confidence and Prediction Intervals for a New Value ..	473	428
14.3.4	Testing a Linear Sub-hypothesis: Partial Fisher Test ...	473	429
14.3.5	Qualitative Variables with More Than Two Modalities	474	430 431
14.3.6	Interaction Between Variables	478	432
14.3.7	Issues with Collinearity	481	433
14.3.8	Variable Selection	482	434
14.3.9	Analysis of Residuals	490	435
14.3.10	Polynomial Regression	496	436
14.3.11	Summary	496	437
	Memorandum	497	438
	Exercises	497	439
	Worksheet	498	440
15	Elementary Analysis of Variance	503	441
15.1	Analysis of Variance with One Factor	503	442
15.1.1	Aims, Data and Model	503	443
15.1.2	Example and Graphical Inspection	504	444
15.1.3	ANOVA Table and Parameter Estimation	505	445
15.1.4	Validation of Assumptions	509	446
15.1.5	Multiple Comparisons and Contrasts	510	447
15.1.6	Summary	512	448
15.2	Analysis of Variance with Two Factors	513	449
15.2.1	Aims, Data and Model	513	450
15.2.2	Example and Graphical Inspection	514	451
15.2.3	ANOVA Table, Tests and Parameter Estimation	516	452
15.2.4	Validating Assumptions	519	453
15.2.5	Contrasts	519	454
15.2.6	Summary	521	455
15.3	Repeated Measures Analysis of Variance	521	456
15.3.1	One-Way Repeated Measures ANOVA	522	457
15.3.2	Two-Factor Model with Repeated Measures for Both Factors	523	458 459

15.3.3 Two-Factor Model with Repeated Measures for One Factor	460 525 461
Memorandum	527 462
Exercises	527 463
Worksheet	527 464
Appendix: Installing R and R Packages	531 465
A.1 Installing R Under Microsoft Windows	531 466
A.2 Installing Additional Packages	532 467
A.2.1 Installing from a File on Your Disk	532 468
A.2.2 Installing Directly from the Internet	533 469
A.2.3 Installing from the Command Line	535 470
A.2.4 Installing Packages Under Linux	535 471
A.3 Loading Installed Packages	536 472
References	539 473
General Index	541 474
Index of R Commands and Symbols	549 475
Index of Authors	563 476
List of R Packages Mentioned in the Book	565 477
Solutions to Exercises	567 478
Solutions to Worksheet	579 479

List of Figures

1.1	A few of the graphical possibilities offered by R	6	2
1.2	The RCommander graphical interface	9	3
1.3	Entering data with the RCommander graphical interface	10	4
1.4	Basic statistics with RCommander	11	5
1.5	Manipulating a data set with RCommander	12	6
1.6	Mean comparison test with RCommander	15	7
1.7	Independence test with RCommander	16	8
1.8	Least squares plane	18	9
3.1	The script window and the command console	41	10
3.2	Characteristics of a complex number	47	11
3.3	Illustration of an array	53	12
7.1	Effect of argument <code>m</code> of function <code>par()</code> . Numbers have been added to gain a better understanding of where future plots will be drawn	13	14
		154	15
7.2	Potential of the function <code>layout()</code>	155	16
7.3	The function <code>layout()</code> and its arguments <code>widths</code> and <code>heights</code>	156	17
7.4	The function <code>plot()</code>	157	18
7.5	The function <code>points()</code>	158	19
7.6	The functions <code>segments()</code> and <code>lines()</code>	159	20
7.7	The function <code>abline()</code>	159	21
7.8	The function <code>arrows()</code>	161	22
7.9	The function <code>curve()</code>	162	23
7.10	The function <code>box()</code>	163	24
7.11	The argument <code>col</code> of function <code>plot()</code>	164	25
7.12	The argument <code>alpha</code> of function <code>rgb()</code>	165	26
7.13	An example using function <code>rainbow()</code>	166	27
7.14	The function <code>display.brewer.all()</code>	167	28
7.15	The function <code>image()</code>	168	29
7.16	The function <code>image()</code> with a coherent display of the data	168	30

7.17	The function <code>text()</code>	169	31
7.18	The function <code>mtext()</code>	170	32
7.19	The function <code>title()</code>	171	33
7.20	Plot title on several lines	172	34
7.21	The function <code>axis()</code>	173	35
7.22	The function <code>legend()</code> with squares	174	36
7.23	The function <code>legend()</code> with line segments	174	37
7.24	Figure illustrating the fine management of graphical parameters ...	178	38
7.25	Managing the colours of a plot	179	39
7.26	Example of use of the arguments <code>adj</code> and <code>srt</code>	181	40
7.27	Using different fonts on a plot	182	41
7.28	Labels on a plot	184	42
7.29	The arguments <code>lend</code> and <code>ljoin</code>	185	43
7.30	The argument <code>pch</code>	186	44
7.31	The arguments <code>lty</code> and <code>lwd</code>	186	45
8.1	Result of the call of the function <code>mydisplay.reg1()</code>	211	46
8.2	Emacs and GDB	263	47
8.3	DDD and GDB	265	48
9.1	Illustration of storage of values in memory. Each little box contains a binary number (0 or 1). Each green number gives the decimal representation of the number in binary form in the block above. Each red number gives the address (expressed here in decimal notation) of the 8-box block above. Note that the same memory addresses could have been written in hexadecimal notation ($b = 16$), giving 3C, 3D, 3E and 3F	294	49 50 51 52 53 54 55
9.2	Illustration of R storage in memory of a (signed) integer. Each little box contains a binary digit (0 or 1). The green number gives the decimal notation of the integer expressed in binary notation in the four blocks above. The red number gives the address (expressed here in decimal base) of the first 8-box memory block above. Note that here, a number is stored over 32 boxes and not over 8 as in Fig. 9.1. Furthermore, the first box is used to specify the sign of the number, negative here	295	56 57 58 59 60 61 62 63
10.1	Modified sinc function	329	64
11.1	Algorithm to determine the type of a variable	340	65
11.2	Cross chart for a qualitative variable	358	66
11.3	Dot chart for a qualitative variable	358	67
11.4	Bar chart for a qualitative variable	359	68
11.5	Pareto chart for a qualitative variable	360	69
11.6	Stacked bar chart for a qualitative variable	361	70
11.7	Bar chart with cumulative frequencies line for an ordinal variable	363	71 72

11.8	Bar chart for a discrete quantitative variable	364	73
11.9	Empirical distribution function for a discrete quantitative variable..	364	74
11.10	Boxplot and explanations	366	75
11.11	Plot of the empirical distribution function for a continuous quantitative variable.	367	76 77
11.12	Density histogram with identical or different class ranges	369	78
11.13	Frequency polygon	370	79
11.14	Cumulative frequency polygon	371	80
11.15	Bar plot for two qualitative variables	372	81
11.16	Mosaic plot for two qualitative variables	372	82
11.17	Cohen–Friendly association plot for two qualitative variables	373	83
11.18	table.cont plot for two qualitative variables	374	84
11.19	Plot of two quantitative variables	375	85
11.20	Box plots of a quantitative variable as a function of the levels of a qualitative variable	376	86 87
11.21	stripchart plot for a quantitative and a qualitative variable.	376	88
12.1	Plot approximating the density of X	388	89
12.2	Convergence in distribution in action on an example with simulated data.	394	90 91
14.1	Scatter plot of child weight against mother weight.	457	92
14.2	Least squares regression line on a scatter plot.	458	93
14.3	Visualization of confidence and prediction intervals	463	94
14.4	Graphical inspection of normality of residuals	464	95
14.5	Plot of residuals as a function of predicted values	465	96
14.6	Scatter plot of all pairs of variables	470	97
14.7	Effect of age on BWT in a model without interaction	479	98
14.8	Effect of age on BWT in a model with interaction	480	99
14.9	Selecting variables with the BIC	484	100
14.10	Checking the assumptions of homoscedasticity (<i>left</i>) and normality (<i>right</i>)	490	101 102
14.11	Residuals as a function of explanatory variables	491	103
14.12	Visualizing outliers: studentized residuals against fitted values	492	104
14.13	Visualizing influential observations: Cook’s distance	494	105
15.1	Box plot of scarring times for each treatment.	506	106
15.2	Analysing the residuals in single-factor ANOVA.	509	107
15.3	Exploration of interaction in two-way ANOVA	516	108
15.4	Residual analysis in two-way ANOVA	520	109

UNCORRECTED PROOF

List of Tables

3.1	The various data types in R	50	2
3.2	The various data structures in R	58	3
4.1	Data importation functions	64	4
4.2	Main arguments to <code>read.table()</code>	64	5
4.3	Packages and R importation functions from common software.	71	6
5.1	Operators and functions which take logical values as input or output	98	7 8
5.2	Operations on sets	99	9
5.3	Codes for the function <code>strptime()</code>	113	10
5.4	Correspondence between BMI and weight categories	124	11
7.1	Parameters to manage the graphics window.	177	12
7.2	Colour management parameters.	179	13
7.3	Managing text displayed on a plot.	180	14
7.4	Parameters to manage axes.	183	15
7.5	Parameters for lines and symbols	185	16
8.1	Conventions on argument types.	238	17
10.1	Table of basic mathematical functions	314	18
12.1	Standard discrete distributions.	405	19
12.2	Standard continuous distributions.	406	20
12.3	Less standard distributions I.	408	21
12.4	Less standard distributions II.	409	22
13.1	Some notation for standard parameter estimation	417	23
13.2	Notation of various quantiles of order p	418	24
13.3	Summary for confidence intervals	424	25
13.4	Standard tests	447	26

14.1	Main R functions for simple linear regression	468	27
14.2	Main R functions for multiple linear regression	496	28
15.1	Main functions for single-factor ANOVA	513	29
15.2	Main functions for two-way ANOVA	521	30

UNCORRECTED PROOF

Mathematical Notations

$:=$	Symbol indicating different notations for a single object	2
\cup	Table fusion	3
$a \in A$	a belongs to set A	4
$A \subset B$	A is included in B	5
$A \supset B$	A includes B	6
$A \cap B$	Intersection of sets A and B	7
$A \cup B$	Union of sets A and B	8
$A \setminus B$	Complement of set B in set A	9
$(A \cup B) \setminus (A \cap B)$	Symmetric difference of sets A and B	10
f_i	Frequency of a modality	11
$ x $	Absolute value of number x	12
$x!$	Factorial of number x	13
$\binom{n}{p}$	Binomial coefficient: number of ways of picking p elements out of n	14
$\Gamma(\cdot)$	Gamma function	15
γ	Euler's constant	16
$\psi(\cdot)$	Digamma function	17
π	Number π	18
λ	Scalar number	19
$\mathcal{A}, \mathcal{B}, \mathcal{C} \dots$	Matrices	20
\mathcal{I}	Identity matrix	21
$n \times p$	Indicates the size of a matrix	22
\mathcal{A}^\top	Transpose of matrix \mathcal{A}	23
\mathcal{B}^{-1}	Inverse of matrix \mathcal{B}	24
$\overline{\mathcal{C}}$	Conjugate of complex matrix \mathcal{C}	25
$\mathbf{x} = (x_1, \dots, x_n)^\top$	Column vector	26
\mathbf{x}^\top	Transpose of vector \mathbf{x}	27
$\mathcal{A} \otimes \mathcal{B}$	Kronecker product of matrix \mathcal{A} with matrix \mathcal{B}	28
$vec(\mathcal{A})$	Vector resulting from the stacking of columns of matrix \mathcal{A}	29
$vech(\mathcal{A})$	Vector resulting from the stacking of columns of matrix \mathcal{A} , but excluding elements above the diagonal	30

\mathcal{A}^*	Adjunct matrix (conjugate transpose) of matrix \mathcal{A}	31
$\mathcal{A}^{1/2}$	Square root of matrix \mathcal{A}	32
$\mathbb{1}_{[A]}(x)$	Equals 1 if $x \in A$ and 0 otherwise	33
$[a, b]$	Interval of values between a and b	34
$\det(\mathcal{A})$	Determinant of matrix \mathcal{A}	35
$\Phi(\cdot)$	Cumulative distribution function of a standard normal random variable $\mathcal{N}(0, 1)$	36
$\hat{\mathcal{X}}$	Matrix given by centring the columns of matrix \mathcal{X}	37
$\mathbb{1}_n$	Vector $(1, \dots, 1)^\top$ of length n	38
X, Y	Non-random variables (descriptive statistics)	39
N	Population size	40
n	Sample size	41
$m_e := q_{1/2}$	Median	42
$PFC_X(\cdot)$	Value of the polygon of cumulative frequencies of X	43
μ_X	Expected value of random variable X , or population mean in descriptive statistics	44
q_p or x_p	Fractile (quantile) of order p of a variable	45
$q_{1/4}, q_{3/4}$	First and third quartiles (also noted q_1 and q_3)	46
$\sigma_{Pop}^2(\mathbf{x})$	Population variance (descriptive statistics)	47
$\sigma_{Pop}(\mathbf{x})$	Population standard error (descriptive statistics)	48
c_v	Population coefficient of variation (descriptive statistics)	49
γ_1	Skewness	50
β_2	Kurtosis	51
μ_3	Centred moment of order 3	52
μ_4	Centred moment of order 4	53
χ^2	Pearson's χ^2 statistic	54
Φ^2, V^2	Cramér's Φ^2 and V^2	55
τ, τ_b	Kendall's τ and τ_b	56
ρ	Theoretical Pearson coefficient of correlation	57
$\eta_{Y X}^2$	Correlation ratio	58
X, Y, ϵ	Random variables	59
x_i, y_i, ϵ_i	Realizations of random variables X, Y, ϵ	60
$\mathbf{X}, \mathbf{Y}, \epsilon$	Random vectors	61
\mathbf{X}_n	Sample (random)	62
\mathbf{x}_n	Sample (observed)	63
\mathbf{X}	Random matrix	64
\mathcal{L}	Generic distribution of a random variable	65
$\mathcal{N}(0, 1)$	Standard normal distribution	66
$\mathcal{N}(\mu, \sigma^2)$	Normal distribution with mean μ and variance σ^2	67
$\mathcal{U}(a, b)$	Uniform distribution over the interval $[a, b]$	68
$\text{Bin}(n, p)$	Binomial distribution with parameters n and p	69
$\mathcal{E}(\lambda)$	Exponential distribution with parameter λ	70

$\mathcal{P}(\lambda)$	Poisson distribution with parameter λ	71
$\mathcal{T}(n)$	Student distribution with n degrees of freedom	72
$\chi^2(n)$ or χ_n^2	χ^2 distribution with n degrees of freedom	73
$\mathcal{F}(n, m)$	Fisher distribution with n and m degrees of freedom	74
$f_X(\cdot)$	Probability density function of random variable X	75
$F_X(\cdot)$	Cumulative distribution function of random variable X	76
$F_X^{-1}(\cdot)$	Inverse cumulative distribution function of random variable X	77
μ	Expected value of a random variable	78
σ^2	Variance of a random variable	79
$\mathbb{E}(Y)$	Theoretical expectation of random variable Y	80
$\text{Var}(Y)$	Theoretical variance of random variable Y	81
\bar{X}_n	Empirical mean $\frac{1}{n} \sum_{i=1}^n X_i$ of sample $\mathbf{X}_n = (X_1, \dots, X_n)^\top$, estimator of μ_X	82
\bar{x}_n	Realization of the empirical mean $\frac{1}{n} \sum_{i=1}^n X_i$ of sample $\mathbf{X}_n = (X_1, \dots, X_n)^\top$, estimate of μ_X	83
\xrightarrow{P}	Convergence in probability	84
$\hat{F}_n(\cdot) := \hat{F}_{X_n}(\cdot)$	Empirical cumulative distribution function of sample \mathbf{X}_n	85
θ	Unknown parameter (the true unknown value of the parameter will sometimes be noted θ^*)	86
$\hat{\theta}(X_1, \dots, X_n)$ or $\hat{\theta}$	Estimator of unknown parameter θ based on the sample $\mathbf{X}_n = (X_1, \dots, X_n)^\top$	87
$\hat{\theta}(x_1, \dots, x_n)$ or $\hat{\theta}$	Estimate of unknown parameter θ based on the observed sample $\mathbf{x}_n = (x_1, \dots, x_n)^\top$	88
$\mathbb{B}(\hat{\theta}(X_1, \dots, X_n); \theta)$	Bias of estimator $\hat{\theta}(X_1, \dots, X_n)$ to estimate unknown parameter θ	89
$P[A]$	Probability of set A	90
$\mathcal{V}(\theta; X_1, \dots, X_n)$	Likelihood function of sample \mathbf{X}_n evaluated at θ	91
$\mathbf{x}^* = (x_1^*, \dots, x_n^*)^\top$	Bootstrap sample generated from the observed sample $\mathbf{x}_n = (x_1, \dots, x_n)^\top$	92
$\hat{\sigma}$	Estimator of σ	93
$\hat{\sigma}$	Estimate of σ	94
p	Proportion	95
\hat{p}	Estimator of a proportion (or of a probability)	96
\hat{p}	Estimate of a proportion (or of a probability)	97
\widehat{m}_e	Estimator of a median	98
\widehat{m}_e	Estimate of a median	99
M	Number of iterations (of generated samples) in a Monte Carlo simulation	100
B	Number of generated bootstrap samples	101
$B(\cdot, \cdot), \Gamma(\cdot)$	Beta function, gamma function	102
$I'_x(\cdot, \cdot)$	Derivative of incomplete beta function	103

$I(\cdot)$	Modified Bessel function	104
$I_\alpha(\cdot)$	Modified Bessel functions	105
u_p	Quantile of order p of a $\mathcal{N}(0, 1)$	106
t_p^n	Quantile of order p of a $\mathcal{T}(n)$	107
q_p^n	Quantile of order p of a $\chi^2(n)$	108
$f_p^{n,m}$	Quantile of order p of a $\mathcal{F}(n, m)$	109
$CI_{1-\alpha}(\theta)$	Random confidence interval at confidence level $1 - \alpha$ for θ	110
$ci_{1-\alpha}(\theta)$	Realized confidence interval at confidence level $1 - \alpha$ for θ	111
$1 - \alpha$	Level of a confidence interval	112
$(x_{(1)}, \dots, x_{(n)})$	Observed sample, sorted from smallest to largest value	113
\mathcal{H}_1	Assertion of interest in hypothesis testing	114
\mathcal{H}_0	“Null” hypothesis, opposite of \mathcal{H}_1	115
α	Significance level or risk of the first kind in hypothesis testing	116
R	Random Pearson empirical coefficient of correlation	117
r	Realized Pearson empirical coefficient of correlation	118
β_0, β_1	Unknown coefficients of a simple linear regression model	119
$\hat{\beta}_0, \hat{\beta}_1$	Estimates of unknown coefficients of a simple linear regression model	120
$\hat{\epsilon}_i$	Observed residuals in a simple linear regression model	121
\hat{y}_i	Adjusted observed values in a simple linear regression model	122
R^2	Random coefficient of determination in regression	123
r^2	Realized coefficient of determination in regression	124
R_a^2	Random adjusted coefficient of determination in regression	125
r_a^2	Realized adjusted coefficient of determination in regression	126
\hat{Y}^p	Predictor of random variable Y for a new value of the explanatory variable X in regression	127
$PI_{1-\alpha}(Y_0, x_0)$	Prediction interval at level $1 - \alpha$ for random variable Y_0 associated with a new value x_0 of the explanatory variable	128
$\beta = (\beta_0, \dots, \beta_p)^\top$	Vector of the $p + 1$ unknown coefficients in a multiple linear regression model with p explanatory variables	129
$\hat{\beta} = (\mathcal{X}^\top \mathcal{X})^{-1} \mathcal{X}^\top \mathbf{Y}$	Estimator of the vector of unknown parameters β for the matrix \mathcal{X} of observed explanatory variables and for the observed vector of explained values in a multiple linear regression model	130
$\hat{\beta}$	Estimate of β	131
VIF	Variance inflation factor	132
AIC	An Information Criterion	133
BIC	Bayesian information criterion	134
h_{ii}	Leverage of i th observation in regression	135
t_i	Standardized residuals	136
t_i^*	Studentized residuals	137
$\hat{\sigma}_{(-i)}$	Estimate of σ excluding i th observation	138

C_i	Cook's distances	139
$\hat{y}_j^{(-i)}$	Prediction of y_j , not using the i th observation	140
$\hat{\beta}_j^{(-i)}$	Estimate of β_j , not using the i th observation	141
I, J	Number of levels of a factor in ANOVA	142
$\mu_{\bullet\bullet}$	Mean general effect in ANOVA	143
$\mu_{i\bullet}$	Effect of level i of a factor in ANOVA	144
$\mu_{\bullet j}$	Effect of level j of a factor in ANOVA	145

UNCORRECTED PROOF

UNCORRECTED PROOF

Part I 146

Preliminaries 147

UNCORRECTED PROOF

UNCORRECTED PROOF

Chapter 1

Introducing R

1
2

Prerequisites and goals of this chapter

- You may find it useful to read the chapter on installing R in the Appendix first.
- This chapter presents the origins, objectives and specificities of R.

3
4

SECTION 1.1

Presentation of the Software

5

1.1.1 Origins

6

R is a piece of statistical software created by Ross Ihaka and Robert Gentleman [21]. R is both a programming language and a work environment. Commands are executed using descriptive code. Results are displayed as text and the plots are visualized directly in their own window. R is clone of the statistical software S-plus. S-plus is an object-oriented programming language S developed by AT&T Bell Laboratories in 1988 [3]. S-plus is used to manipulate data, draw plots and perform statistical analyses of data.

7
8
9
10
11
12
13

1.1.2 Why Use R?

14

First of all, R is **free** and **open-source**. It works under UNIX (and Linux), Microsoft Windows and Macintosh Mac OS: it is **cross-platform**. It is being developed in the

free software movement by a large and growing community of eager volunteers. 16
 Anyone can contribute to and improve R by integrating more functionalities or 17
 analysis methods. It is thus a quickly and constantly evolving piece of software. 18

R is a very powerful statistical tool. The learning curve in R is steeper than other 20
 statistical software on the market such as SPSS, SAS or Minitab. R is not the kind of 21
 statistical package, which you can use with a few clicks of the mouse in the menus. 22
 In order to use it, you need to understand the statistical method that you are trying to 23
 implement, so R is a didactic program. R is also very efficient and easy to implement 24
 once you have mastered it. You will be able to create your own tools and you will 25
 be able to handle and work on very sophisticated data analyses. 26

Warning



R is harder to comprehend than other software on the market. You need to 27
 spend time learning the syntax and commands. 28

R is especially powerful for data manipulation, calculations and plots. Its features 27
 include: 28

- an integrated and very well-conceived documentation system (in English) 29
- Efficient procedures for data treatment and storage; 30
- a suite of operators for calculations on tables, especially matrices; 31
- a vast and coherent collection of statistical procedures for data analysis; 32
- advanced graphical capabilities; 33
- a simple and efficient programming language, including conditioning, loops, 34
 recursion, and input-output possibilities. 35

Note

For the readers already used to SAS, SPSS or Stata, we advise to read the 27
 books [32, 33] and also to consult the two following Internet websites: 28

- <http://rforsasandspssusers.com>
- <http://www.statmethods.net>



Note also that it is possible to call R functions directly from Matlab using 29
 the R.matlab package and from Excel using the RExcelInstaller pack- 30
 age. Reading of [20] might be useful in this context. Finally, a similar tool 31
 for OpenOffice, called R00o, exists; see the Internet website [http://rcom.](http://rcom.univie.ac.at) 32
[univie.ac.at](http://rcom.univie.ac.at). 33

SECTION 1.2

R and Statistics

36

Many classical and modern statistical techniques are implemented in R. The most common methods for statistical analysis, such as

- descriptive statistics; 39
- hypothesis testing; 40
- analysis of variance; 41
- linear regression methods (simple and multiple) 42

are directly included at the core of the system. It should be noted that most advanced statistical methods are also available through external packages. These are easy to install, directly from a menu. They are all grouped and can be browsed on the website of the *comprehensive R archive network* (CRAN) (<http://cran.r-project.org>). This website also includes, for some large domains of interest, a commented list of packages associated with a theme (called Task View). This facilitates the search for a package on a specific statistical method. Furthermore, detailed documentation for each package is available on the CRAN.

It should also be noted that recent statistical methods are added on a regular basis by the statistics community itself.

See also

Section A.2, p. 532, gives details on the procedure to install a new package.



SECTION 1.3

R and Plots

55

One of the main strengths of R is its capacity (much greater than that of other software on the market) to combine a programming language with the ability to draw high-quality plots. Usual plots are easily drawn using predefined functions. These functions also include many parameters, for example to add titles, captions and colours. But it is also possible to create more sophisticated plots to represent complex data such as contour lines, volumes with a 3D effect, density curves, and many other things. It is also possible to add mathematical formulae. You can arrange or overlay several plots in the same window and use many colour palettes.

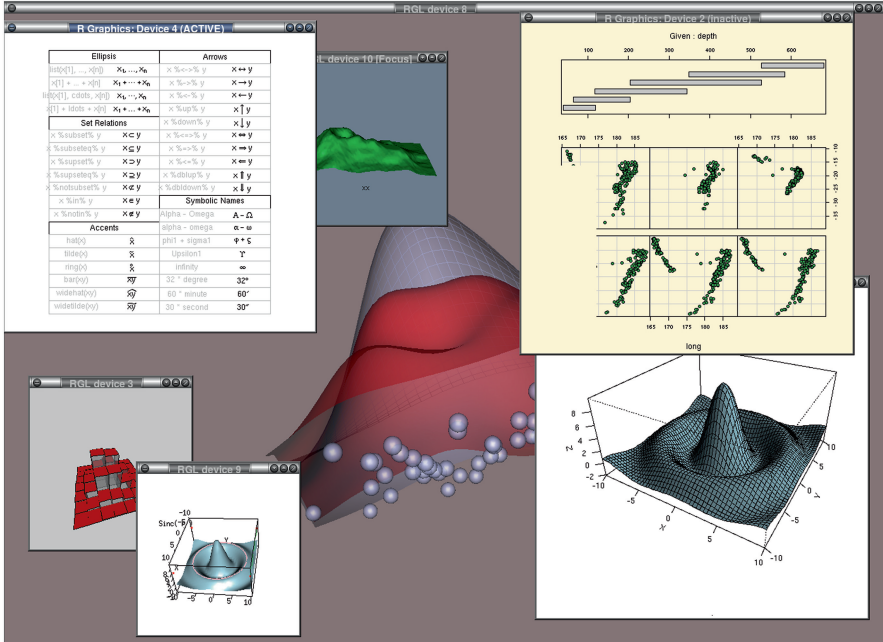


Fig. 1.1: A few of the graphical possibilities offered by R

You can get a demonstration of the graphical possibilities in R by typing in the following instructions:

```
demo(image)
example(contour)
demo(graphics)
demo(persp)
demo(plotmath)
demo(Hershey)
require("lattice") # Load the package, which you must have
                  # previously installed by using the menu
                  # Packages/Install packages.

demo(lattice)
example(wireframe)
require("rgl")    # Same remark as above.
demo(rgl)         # You can interact by using your mouse.
example(persp3d)
```

Figure 1.1 above shows a few of these plots.

66

67

SECTION 1.4

The R Graphical User Interface

68

The R graphical user interface (GUI) (i.e. its set of menus) is very limited, and completely nonexistent on some operating systems, when compared to other standard software. This minimality can set back some new users. However, this drawback is limited since:

- it has the didactic advantage that it incites users to know well the statistical procedures they wish to use;
- there are additional tools which extend the GUI.

In the next section, we present the package `Rcmdr`, which can be installed from the menu `Packages` and which allows standard graphical and statistical analyses with a more user-friendly interface, which includes drop-down menus. Furthermore, the R instructions for the analysis chosen from the `RCommander` menus are displayed in dedicated panel. This can be useful if you do not know (or remember) the R instruction for a specific task.

Tip

Note that after you have learnt R thoroughly, you will be able to develop yourself tools similar to `Rcmdr`, made for a final users who do not desire to learn R but only to use, in the most user-friendly way, a procedure created by you. To this end, you can use the package `tcltk`.



Warning

Note that by using `RCommander`, we are distancing ourselves from what makes the strength and flexibility of R. We therefore advise against using such a tool if you wish to become an advanced user.



SECTION 1.5

First Steps in R

82

1.5.1 Using `RCommander`

83

In this section, we offer a brief introduction to the package `Rcmdr`. We then present some functionalities given by this interface for statistical manipulations. We conclude by explaining how to add functionalities to the `RCommander` interface.

1.5.1.1 Launching RCommander 87

Follow these steps to start RCommander. 88

- ▶ Double-click on the R icon on your Desktop. 89
- ▶ In the console, type `install.packages("Rcmdr")`. Choose a nearby mirror. 90
- ▶ In the console, type `require("Rcmdr")`. Answer Yes to all the questions you may be asked. The RCommander graphical interface then opens. Another option is to click on the menu Packages, then Load package . . . , then Rcmdr. 91
- ▶ In the Messages panel, you should see `WARNING: the Windows version of R Commander works better under RGui with the single document interface (SDI)`. 92
- ▶ To remedy this issue, close RCommander. 93
- ▶ In RGui, go to Edit, then Preferences. Check SDI then click on Save . . . and on Save. You can take this opportunity to customize R. 94
- ▶ Close R and save an image of the session. 95
- ▶ Restart R, then RCommander by typing `require("Rcmdr")` in the R console. 96
- ▶ Restart R, then RCommander by typing `require("Rcmdr")` in the R console. 97
- ▶ Restart R, then RCommander by typing `require("Rcmdr")` in the R console. 98
- ▶ Restart R, then RCommander by typing `require("Rcmdr")` in the R console. 99
- ▶ Restart R, then RCommander by typing `require("Rcmdr")` in the R console. 100
- ▶ Restart R, then RCommander by typing `require("Rcmdr")` in the R console. 101

See also



We refer the reader to Sect. A.2 which details how to install the package Rcmdr.

Mac



Macintosh users may find useful the instructions at <http://socserv.mcmaster.ca/jfoxx/Misc/Rcmdr/installation-notes.html>, after installing package tcltk which is available on the CRAN.

The graphical interface of RCommander includes four parts as shown on Fig. 1.2: 102

- (a) Drop-down menus to perform specific tasks 103
- (b) A Script window which presents the code executed thanks to click on a drop-down menu 104
- (c) An Output window which gives the output of the executed code 105
- (d) A Messages window giving a message on the last task 106

108

1.5.1.2 Handling Data with RCommander 109

To perform statistical analyses, you need data. 110

• Entering data by hand 111

Follow these steps to enter data by hand. 112

113

114



Fig. 1.2: The RCommander graphical interface

- ▶ In the menu `Data`, choose `New data set...` 115
- ▶ In the window `New data table`, choose a name for your data set, for example `Data1`. 116
- ▶ A data editor appears. Click on `var1` and replace it with `Name`. Enter a few names for this variable: Peter, Jack, Ben (see Fig. 1.3). 117
- ▶ Create a variable `Height` of type `numeric` with the following values: 182, 184, 190. 120
- ▶ Click on the cross (X) at the top-right corner of the active window to close the data editor. 121
- ▶ You can visualize your data set by clicking on `View`. 122

We can now calculate some basic statistics. 125

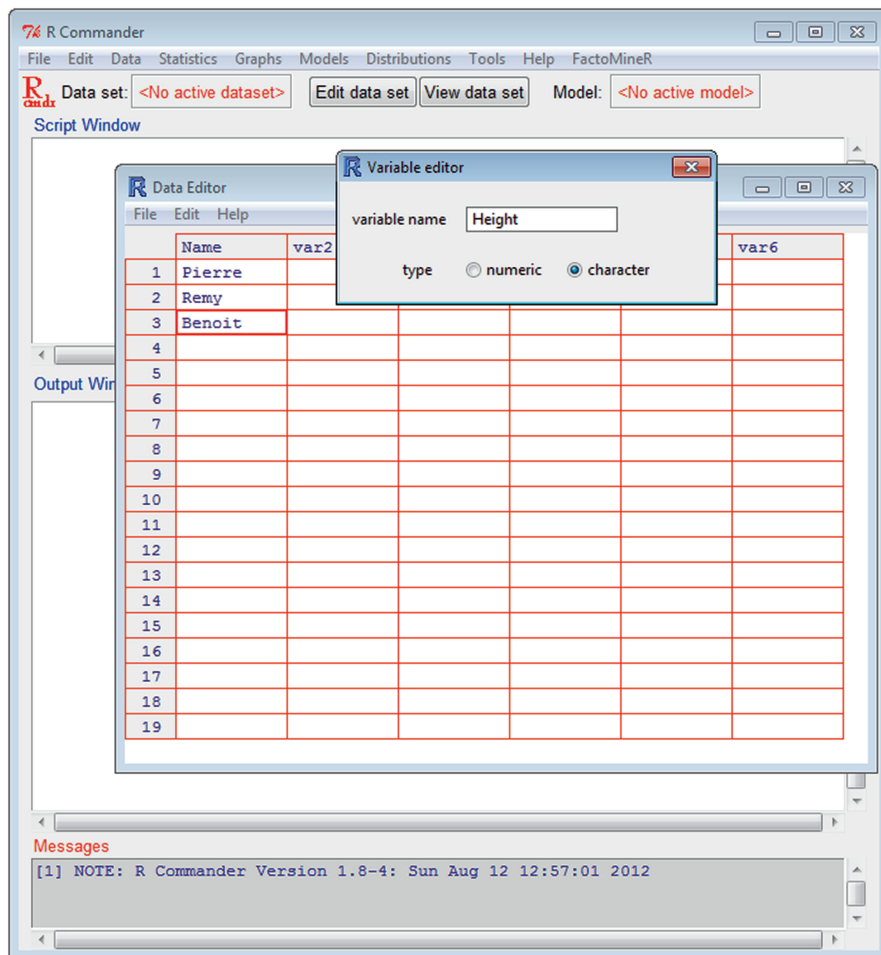


Fig. 1.3: Entering data with the RCommander graphical interface

• Basic statistics

Follow these steps to get some basic statistics on your data set:

- ▶ In the menu Statistics, choose Summary, then Descriptive statistics
- ▶ A window called General statistics opens up; the only numeric variable in our data set is the variable Height.
- ▶ Choose the statistics Mean, Standard deviation and Quantiles and click on OK.
- ▶ The result is displayed in the Output window. Note that you can check the R instruction which was used for this task in the Script window (see Fig. 1.4).

126

127

128

129

130

131

132

133

134

135

136

137

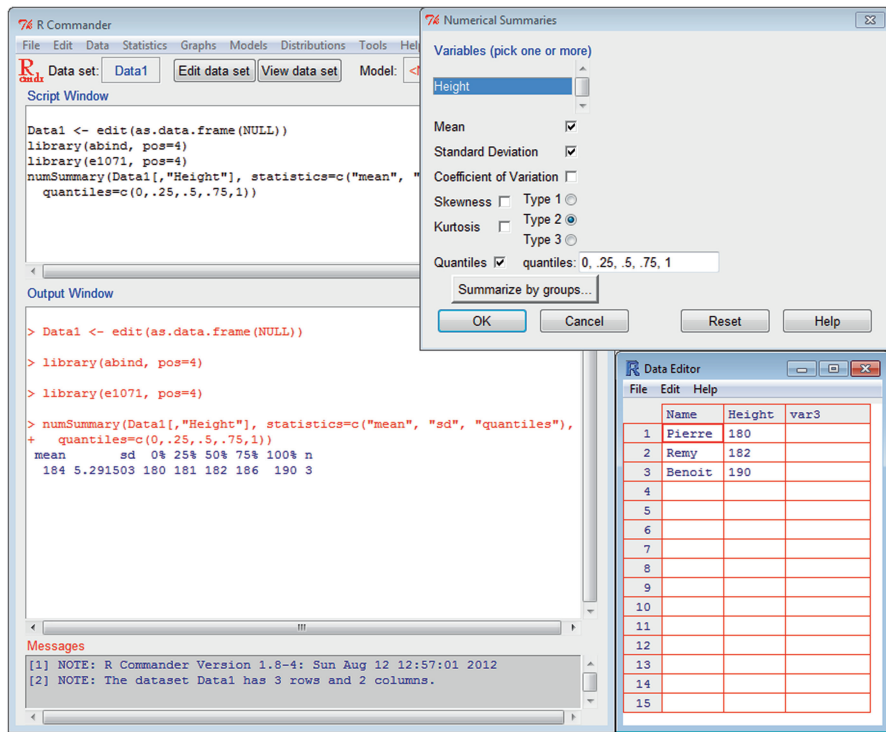


Fig. 1.4: Basic statistics with RCommander

Note that it is also possible to type an instruction directly in the Script window without using a menu. Here is an example. 138
139

► Type in the Script window: 140

```
numSummary(Data1[, "Height"], statistics=c("mean", "sd"))
```

► Click on that line so that the cursor is displayed there, then click on Submit. 141

► You have just computed the mean and standard deviation of variable Height which contains 3 observations. The result appears in the Output window: 142
143

```
mean      sd % n
184 5.291503 0 3
```

144

• Manipulating the data set 145

146

In our toy example, suppose that we also have the weight and wish to compute the body mass index: $BMI = Weight/Height^2$ (height in metres). 147
148

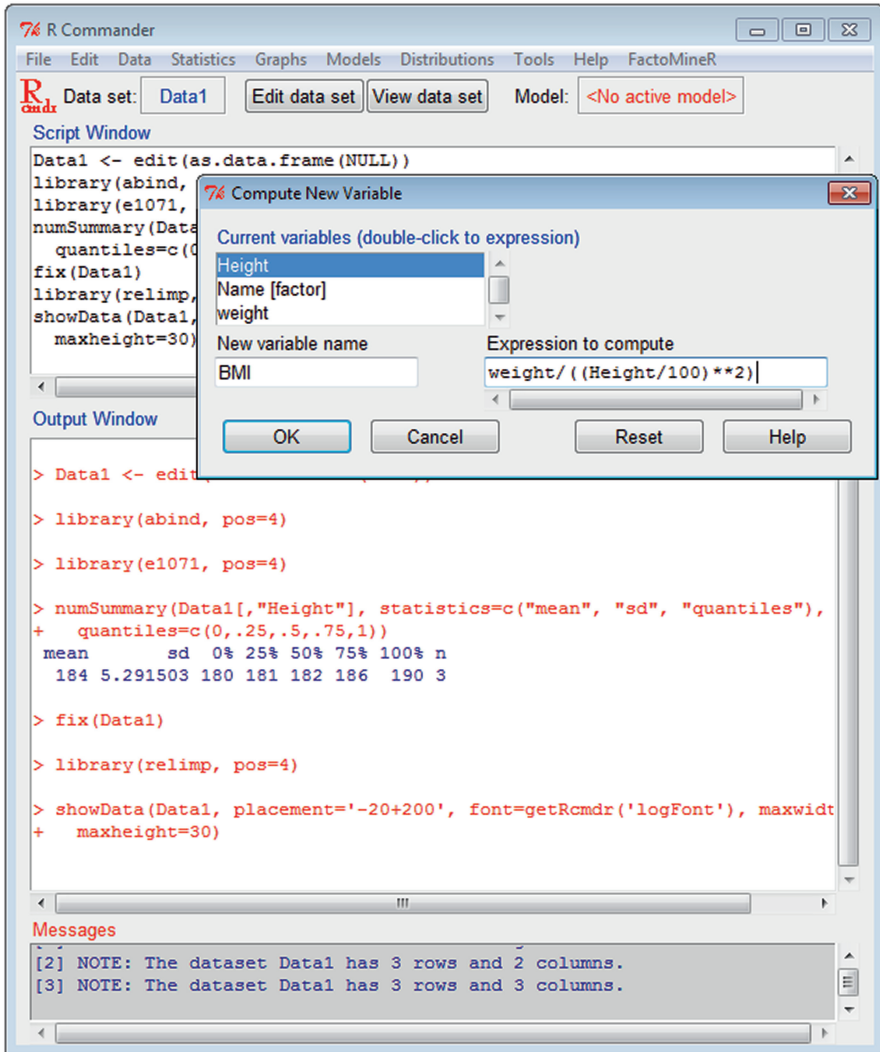


Fig. 1.5: Manipulating a data set with RCommander

- ▶ Click on Edit (below the RCommander menus). 149
- ▶ The data editor opens up and you can add the numeric variable Weight, with the 150
following values: 70, 72 and 75. Now close the data editor. 151
- ▶ In the Data menu, choose Manage variables in the active data set, 152
then Calculate a new variable... A window opens. 153
- ▶ For Name of new variable, type BMI and for Expression to calculate: 154
Weight/((Height/100)**2) (see Fig. 1.5). Click on OK to complete the calcu- 155
lation. 156
- ▶ Click on View to see the result for your data set. 157

You are starting to feel tired and need a coffee break! But before you take one, follow these steps to save your data set.

- ▶ In the Data menu, choose **Active dataset**, then **Save active dataset...**
- ▶ A window opens. You can choose a location to save your data set. We shall call it BMI and by default it has the `.RData` extension.
- ▶ Close RCommander and answer OK to the question **Do you wish to quit?**, No to **Save script file?** and No to **Save output file?**.
- ▶ You can now close R and answer No to the question **Save session image?**.

After a well-deserved break, you wish to add new data to your file `BMI.RData`.

- ▶ Open an R session. Type `library("Rcmdr")`.
- ▶ In the Data menu, choose **Load data set...**
- ▶ A window opens. Navigate to and open the file `BMI.RData`.
- ▶ Click on **View** to display your data set.
- ▶ Add the information for a new person ("**Julia**", `Height=150`, `Weight=52`) by clicking on **Edit**.
- ▶ After closing the editor, you can check the changes by clicking on **View**. You then see the value `NA` (*not available*) for Julia's BMI.
- ▶ To get Julia's BMI, you need to go through the steps of section *manipulating the data set* again. We shall see later on how to create a function which calculates the BMI in a more user-friendly fashion.

You now wish to send your data set to a colleague who does not use R yet.

- ▶ In the Data menu, choose **Active dataset**, then **Export active data set ...**
- ▶ A first window opens. Uncheck the box **Write names of individuals (rows)** since we have not defined these. Choose **Spaces** for the field separator.
- ▶ Click on **OK**. A second window opens. You can choose a place to save your data set. We shall call it BMI and it has the default extension `.txt`.
- ▶ You can now send your data set `BMI.txt` to your colleague and use this opportunity to mention the wonderfulness of R, which has a rather user-friendly interface for data manipulation.

1.5.1.3 A Few Statistical Tasks with RCommander

In this section, we present a brief overview of how to use RCommander for statistical tasks. We start with a mean comparison test and a chi-square test of independence. We then show how to use RCommander to visualize the standard distributions (binomial, poisson, normal, gamma, etc.). We conclude with a linear model fit.

• Mean comparison test

We propose to use data already available in R. Follow these steps to load a data set:

- ▶ In the Data menu, choose Data in packages, then Read data from an attached package. . . .
- ▶ A window opens. Double-click on datasets in the Package section, then on sleep in the right column.
- ▶ sleep appears in the box Enter a dataset name (see Fig. 1.6).
- ▶ You can now click on Help on the selected dataset to have some information about it.
- ▶ Click on OK to close the previous window, then visualize the data set by clicking on View.

These data are used to compare the effect on sleep of a soporific drug, compared to a control group. We shall first visualize the distribution of sleep gain in both groups, then do a mean comparison test to see whether there is any statistical significant difference between the drug and the control.

- ▶ In the Graphs menu, choose Box plot. . . .
- ▶ A window opens. Click on Plot by group. . . ., then on the variable group, then on OK twice.
- ▶ You can now see two box plots representing the sleep time gain in both groups.
- ▶ You can save this plot by clicking on File, then Save as. Several formats are possible.

You can also enhance this plot, for example, by adding colours. In the script window, type

```
boxplot(extra~group,ylab="extra",xlab="group",data=sleep,
        col=c("red","blue"))
```

then click on Submit.

See also



Chapter 7 is dedicated to plots in R.

We now perform a mean comparison test.

- ▶ In the Statistics menu, choose Means, then independent t-test. . . .
- ▶ Click on group in section Groups (one). You now see specified the difference 1-2 (group 1 vs. group 2).
- ▶ Click on OK to see the result in the Output window (see also Fig. 1.6).

The p -value of this test (greater than 5%) does not allow us to conclude that there is a significant difference between the sleep gains given by the drug and the control.

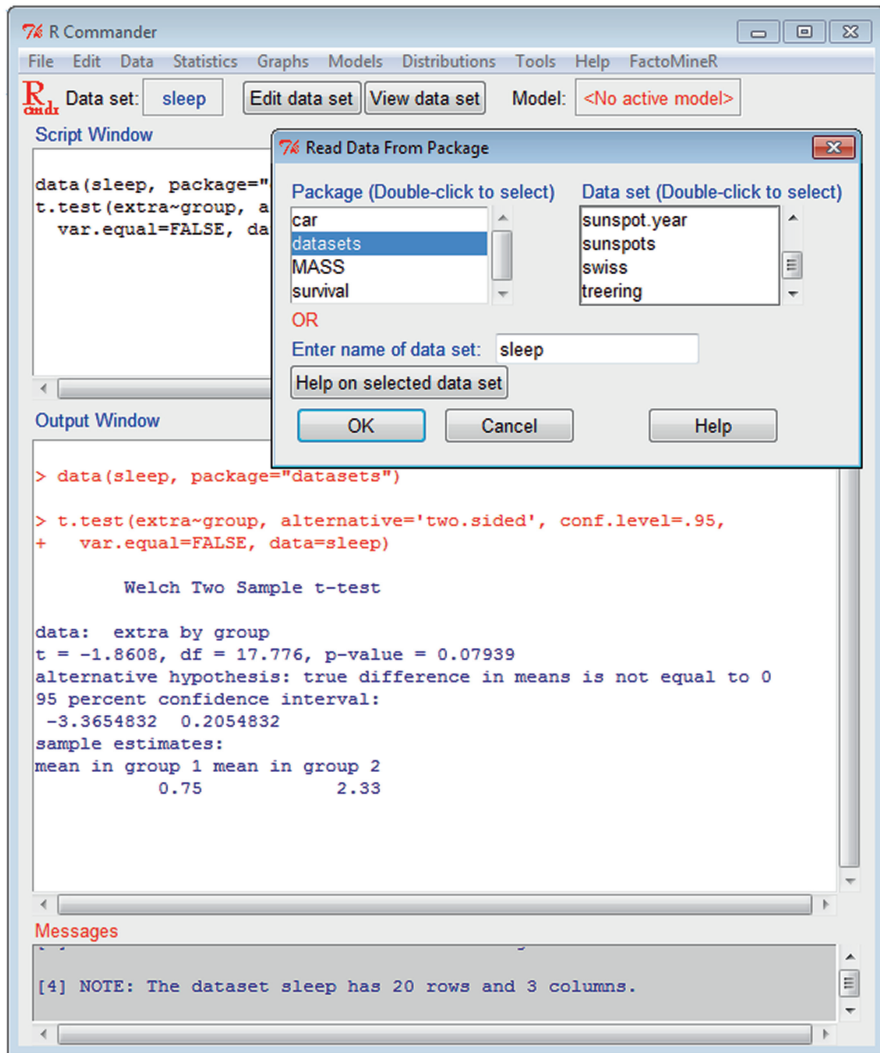


Fig. 1.6: Mean comparison test with RCommander

• Test on a double entry table

229
230
231
232
233
234
235
236

In a therapeutic test, the underlying question is whether a treatment on HIV-positive mothers has an effect on the HIV status of the child. If it does not, then the HIV status of the child is independent of the treatment taken by the mother. In this test, out of 391 children, 100 are HIV negative, 193 have mothers under treatment and 41 are HIV positive and have mothers under treatment. To know whether the treatment has an effect, follow these steps:

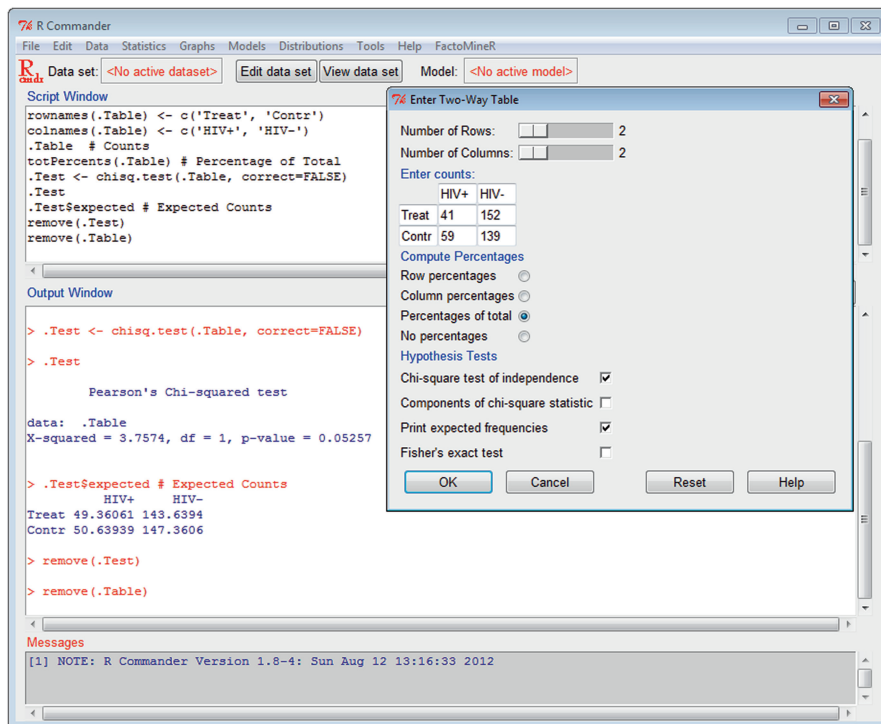


Fig. 1.7: Independence test with RCommander

- ▶ In the Statistics menu, choose Contingency tables, then Fill and analyse a double entry table... 237
- ▶ A window opens. Fill the table as indicated in Fig.1.7. Choose Total percentages and Print expected frequencies. 238
- ▶ Click on OK to see the result in the Output window. 239

At the 5% risk level, we cannot conclude that the treatment has an effect on the child's HIV status. 240

• Exploring distributions 241

RCommander can be used to visualize standard distributions. 242

- ▶ In the Distributions menu, choose Continuous distributions, the Normal distribution, then Plot of normal distribution... 243
- ▶ A window opens. Specify a mean of 4 and a standard deviation of 2. Click on OK. 244
- ▶ The curve of the density of a normal distribution centred at 4 and with standard deviation 2 appears in a graphical window. 245

You can follow the same steps for other probability distributions. 246

• Fitting a linear model

RCommander can be used to easily fit standard regression models. We illustrate this with the linear model. We shall first download a data set from an Internet address (URL). It contains the measures, for 80 patients with a disabling illness, of the variables GENDER (1 = Male, 2 = Female), WEIGHT (in kg), HEIGHT (in cm), PAIN (ordinal variable: a=least pain), DISTANCE (number of metres walked), MOBILITY (self-evaluation of mobility; 1=most mobile) and STAIRS (number of steps climbed).

- ▶ In the Data menu, choose Import data, then from a text file, the clipboard or a URL...
- ▶ A window opens. Call the data table Illness. Check the box Internet link (URL) in Data file and the box Tabulations for Field separator; click on OK.
- ▶ In the field Internet link (URL), type `http://biostatisticien.eu/springer/illness.txt`.
- ▶ Click on OK and you should see the following in the Messages window: The illness data set contains 80 rows and 8 columns.

We shall fit a multiple regression model. Follow these steps.

- ▶ In the Statistics menu, choose Model fitting, then Linear regression ...
- ▶ Choose for example Model.1 as your model name in the field Enter a name for the model.
- ▶ Choose variable DISTANCE as the response variable, and variables WEIGHT and HEIGHT as the explanatory variables (keep the CTRL key pressed).
- ▶ Click on OK. The result of your linear model adjustment appears in the Output window. This result corresponds to the instructions

```
Model.1 <- lm(DISTANCE~WEIGHT+HEIGHT, data=Illness)
summary(Model.1)
```

which are shown in the Script window.

See also

Chapter 14 presents the linear model in further detail.



We now visualize the least squares plane corresponding to the fitted model.

- ▶ In the Plot menu, choose 3D plot, then 3D scatterplot...
- ▶ Choose variable DISTANCE as the response variable and the variables WEIGHT and HEIGHT as explanatory variables (use the CTRL key).
- ▶ Choose Ordinary least squares as the surface to fit. Click on OK.

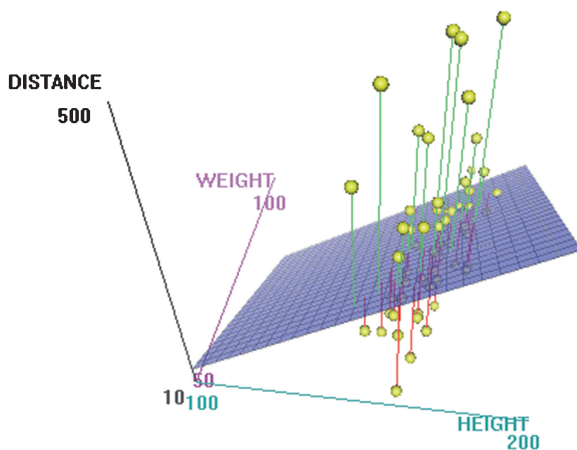


Fig. 1.8: Least squares plane

You can now see the 3D scatterplot (shown in Fig. 1.8) and the least squares 288
plane. You can move the image with your mouse. 289

1.5.1.4 Adding Functionalities to the RCommander Interface 290

Some packages available on the official R website can also be integrated to 291
the RCommander menus. They are easy to identify: their names start with 292
RcmdrPlugin. We now illustrate how to use such a package. 293

See also



You can read the article [17] which explains how to build a package for
RCommander integration.

• The TeachingDemos package 294

The RcmdrPlugin.TeachingDemos package can be used to illustrate some sta- 295
tistical concepts. 296
297

- ▶ Type `install.packages("RcmdrPlugin.TeachingDemos")` in the Script 298
window. Click on Submit and choose a nearby mirror. Once the installation is 299
complete, close and reopen RCommander using the instruction `Commander()`. 300
- ▶ In the Tools menu, choose Load Rcmdr plug-ins..., click on OK and answer 301
Yes to the question Restart now?. 302
- ▶ There is a new menu called Demos. In this menu, you can choose for example the 303
submenu Simple Correlation and explore the notion of correlation. 304

This plug-in also adds submenus to pre-existing menus. For example, in the Distributions menu, you can now choose Visualize distributions, then t distributions. By checking Show Normal Distribution, and by playing with the d.f. (*degree of freedom*) cursor, you can visualize the closeness of the Student distribution and the normal distribution.

• The sos package

The RcmdrPlugin.sos package can be used to ease the search for help on a given concept or function. Follow the same steps as before to install this plug-in. A new submenu called Search R Help ... (sos) appears in the Help menu. Explore this new RCommander functionality, for example, by typing linear model.

See also

Chapter 6 describes how to search for information about R.



1.5.2 Using R with the Console

In the previous subsection, we saw how to use R through menus. In fact, this way of proceeding is far from optimal, since it imposes many limitations on the possibilities offered by R. Many analyses, either deeper or more recent and innovative, are not available in the RCommander menus. It is thus very useful to escape from the “button clicking” approach and master the R programming language. You will then be able to perform simulations and to code repetitive tasks. We have already encountered a few R instructions when using RCommander, which is itself a tool written in the R language. We now propose a brief introduction to a few elements of the R syntax, first through an analysis of complex data arising from a functional magnetic resonance imaging (MRI) experiment, then by letting the reader type a few R commands and think about the output.

1.5.2.1 The Strength of R Shown on an Example

Some neuroscientists work on finding which part of the brain deals with visual information on colour. To this end, a visual stimulus, consisting in an alternance of coloured and non-coloured moving patterns, is shown to a subject. During this time, volumic images of the subject’s brain are acquired at time $t = 1, \dots, T$ with an MRI scanner. Each 3D image is in fact a large (Rubik’s!) cube made of many voxels, the 3D equivalents of 2D pixels. At time $t = 1, \dots, T$, each voxel contains an electromagnetic measurement value $x(t)$. We can thus consider that in each voxel, we have observed a time series $\{x(t); t = 1, \dots, T\}$ representing

electromagnetic variations. The acquired data (given in file `Mond4D.nii`, produced during a Mondrian experiment performed by M. Dojat and J. Huppé) thus consist in a 4-dimensional array, the concatenation of several volumic brain images measured through time.

We used R to find, in each brain slice, which voxel had temporal variations most correlated with the stimulus signal. The code below can be downloaded from <http://biostatisticien.eu/springerR/brain-code.R> and opened, thanks to the submenu `Open script...` of the File menu in R. The key combination `CTRL+R` then executes one by one the instructions of this script. You can try to execute these instructions to visualize the results. This will help you familiarize yourself with some of the possibilities offered by R.

We first download the data files we need (the files `Mondanat.img` and `Mondanat.hdr` contain an anatomical image of the subject's brain).

```
> getfile <- function(myfile)
+   download.file(paste("http://biostatisticien.eu/springer/",
+   myfile, sep=""), paste(getwd(), "/", myfile, sep=""), mode="wb")
> getfile("Mond4D.nii")
> getfile("Mondanat.hdr")
> getfile("Mondanat.img")
```

We then install the package to read the data.

```
> install.packages("AnalyzefMRI") # Choose a mirror.

> # File names.
> file.func <- paste(getwd(), "/", "Mond4D.nii", sep="")
> file.anat <- paste(getwd(), "/", "Mondanat.img", sep="")

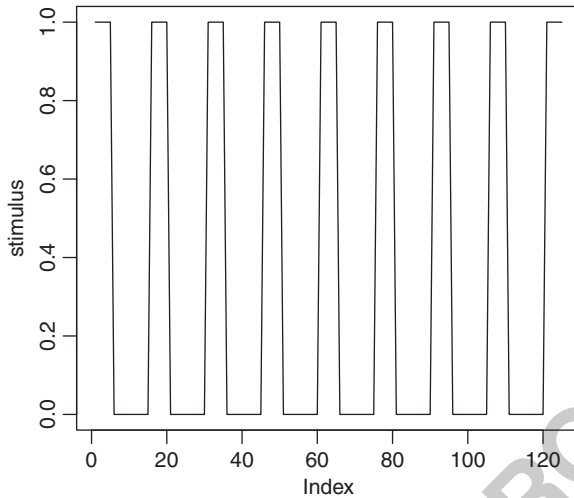
> # Brain slice number.
> slice <- 10
```

The next instructions read the data.

```
> anat.slice <- f.read.nifti.slice(file.anat, slice, 1)
> class(anat.slice)
[1] "matrix"
> dim(anat.slice)
[1] 128 128
> func.slice <- f.read.nifti.slice.at.all.timepoints(file.func,
  slice)
> class(func.slice)
[1] "array"
> dim(func.slice)
[1] 128 128 125
```

We now create the coding of the visual stimulus signal (1=colour, 0=no colour).

```
> stimulus <- c(rep(c(1,1,1,1,1,0,0,0,0,0,0,0,0,0,0), 8), 1,1,1,1,1)
> plot(stimulus, type="l")
```

355

We compute correlations between the observed time series in each voxel and the stimulus series. 356

```
357
> corMat <- matrix(NA,nrow=128,ncol=128)
> for (i in 1:128) {
+   for (j in 1:128) {
+     corMat[i,j] <- cor(func.slice[i,j,],stimulus)
+   }
+ }
```

We can now compute the coordinates of the voxel most strongly correlated with the stimulus 358

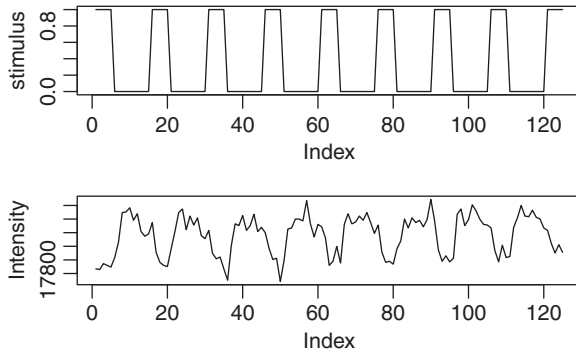
```
359
> which(abs(corMat)==max(abs(corMat),na.rm=TRUE),arr.ind=TRUE)
      row col
[1,]  67 117
```

and the correlation value of this voxel 360

```
> corMat[67,117]
[1] -0.6675017
```

We can then plot the time series observed in this voxel. 361

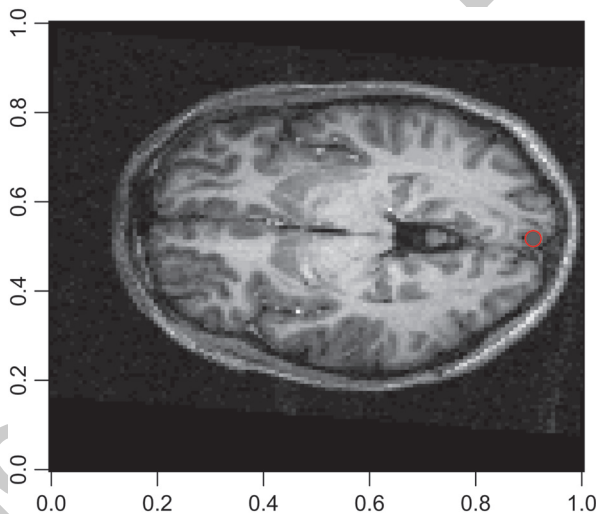
```
> par(mfrow=c(2,1))
> plot(stimulus,type="l")
> plot(func.slice[67,117,],type="l",ylab="Intensity")
```



362

We are now able to identify on the anatomical image of the brain the most active voxel for the visual stimulus. 363
364

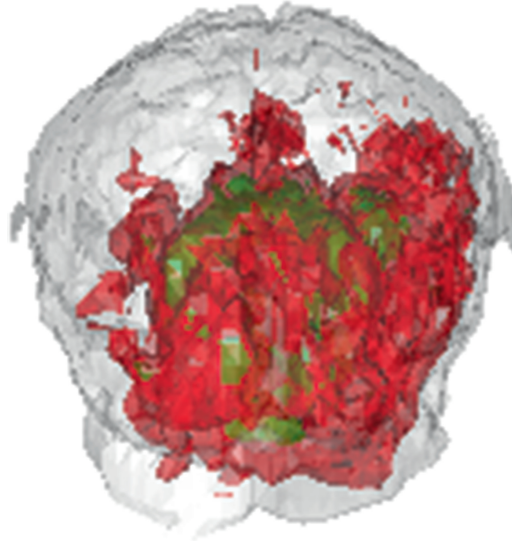
```
> image(as.matrix(rev(as.data.frame(t(anat.slice))))),
        col=gray((0:32)/32))
> points(117/128,67/128,col="red",cex=2,pch=19)
```



365

Note that you can also visualize these data in 3D. The following instructions, 366
taken from the help file for the function `contour3d()` from package `misc3d`, give 367
an interactive 3D view of the brain. 368

```
> install.packages("misc3d")
> require("misc3d")
> a <- f.read.analyze.volume(system.file("example.img",
+                                       package="AnalyzeFMRI"))
> a <- a[,,,1]
> contour3d(a,1:64,1:64,1.5*(1:21),lev=c(3000, 8000, 10000),
+          alpha=c(0.2,0.5,1),color=c("white","red","green"))
```



369

You can try to move the image with your mouse.

370

1.5.2.2 A Brief Introduction of R Syntax Through Some Instructions to Type

371

• Basic operations

372

We advise the reader to play with these commands and try to understand how they work.

373

374

375

```

> 1*2*3*4
[1] 24
> factorial(4)
[1] 24
> cos(pi)
[1] -1
> x <- 1:10
> x
[1] 1 2 3 4 5 6 7 8 9 10
> exp(x)
[1] 2.718282 7.389056 20.085537 54.598150
[5] 148.413159 403.428793 1096.633158 2980.957987
[9] 8103.083928 22026.465795
> x^2
[1] 1 4 9 16 25 36 49 64 81 100
> chain <- "R is great!"
> chain
[1] "R is great!"
> nchar(chain)
[1] 11
> ?nchar

```

```

> M <- matrix(x,ncol=5,nrow=2)
> M
      [,1] [,2] [,3] [,4] [,5]
[1,]   1   3   5   7   9
[2,]   2   4   6   8  10
> M[2,3]
[1] 6
> L <- list(matrix=M,vector=x,chain=chain)
> L[[3]]
[1] "R is great!"
> while(TRUE) {
+   toguess <- sample(1:2,1)
+   {cat("Guess a number among 1, 2, 3: "); value <- readline()}
+   if (value == toguess) {print("Well done!"); break()}
+   else print("Try again.")
+ }
> ls()
[1] "chain" "L"      "M"      "x"
> rm(chain)

```

The following commands perform matrix operations:

376

```

> A <- matrix(runif(9),nrow=3)
> 1/A
      [,1] [,2] [,3]
[1,] 2.270797 1.546875 1.422103
[2,] 1.268152 1.957924 1.057803
[3,] 1.642736 5.273120 2.174020
> A * (1/A)
      [,1] [,2] [,3]
[1,]   1   1   1
[2,]   1   1   1
[3,]   1   1   1
> B <- matrix(1:12,nrow=3)
> A * B
Error in A * B : non-conformable arrays
> A %*% B
      [,1] [,2] [,3] [,4]
[1,] 3.842855 9.212923 14.582990 19.95306
[2,] 4.646105 11.380053 18.114001 24.84795
[3,] 2.367954 6.143031 9.918107 13.69318
> (invA <- solve(A))
      [,1] [,2] [,3]
[1,] -1.145642 -3.376148 5.187347
[2,] 4.379786 -4.641906 2.844607
[3,] -3.321872 6.381822 -5.863772
> A %*% invA
      [,1] [,2] [,3]
[1,] 1.000000e+00 0.000000e+00 0
[2,] 0.000000e+00 1.000000e+00 0
[3,] -2.220446e-16 4.440892e-16 1

```

```

> det(A)
[1] 0.04857799
> eigen(A)
$values
[1] 1.6960690+0.000000i -0.1424863+0.091319i
[3] -0.1424863-0.091319i
$vectors
      [,1]          [,2]          [,3]
[1,] 0.5859852+0i 0.6140784-0.1816841i 0.6140784+0.1816841i
[2,] 0.7064296+0i 0.2234155+0.2505528i 0.2234155-0.2505528i
[3,] 0.3969616+0i -0.6908020+0.0000000i -0.6908020+0.0000000i

```

• Statistics

377

Here are a few statistical calculations.

378

379

```

> weight <- c(70,75,74)
> mean(weight)
[1] 73
> height <- c(182,190,184)
> mat <- cbind(weight,height)
> mat
      weight height
[1,]     70    182
[2,]     75    190
[3,]     74    184
> apply(mat,MARGIN=2,FUN=mean)
      weight height
73.0000 185.3333
> ?apply
> colMeans(mat)
      weight height
73.0000 185.3333
> names <- c("Peter","Ben","John")
> data <- data.frame(Names=names,height,height,weight)
> summary(data)
      Names      height      weight
Ben :1   Min.   :182.0   Min.   :70.0
John :1  1st Qu.:183.0   1st Qu.:72.0
Peter:1  Median :184.0   Median :74.0
      Mean   :185.3   Mean   :73.0
      3rd Qu.:187.0   3rd Qu.:74.5
      Max.   :190.0   Max.   :75.0

```

• Some plots

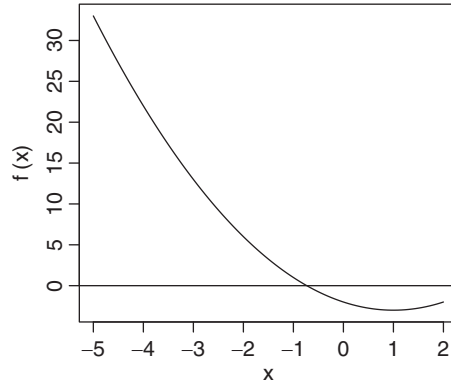
380

381

```

> f <- function(x) x^2-2*x-2
> curve(f,xlim=c(-5,2));abline(h=0)
> locator(1) # Click on the intersection of the two curves.

```



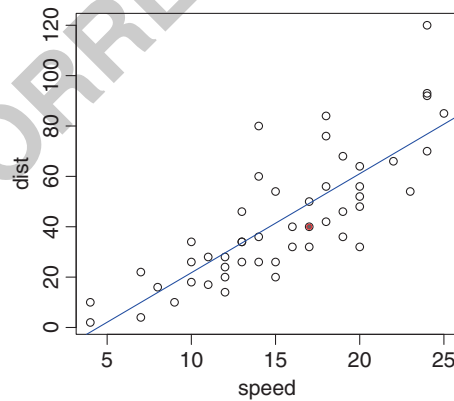
382

```

> uniroot(f,c(-5,2))
$root
[1] -0.7320503
$f.root
[1] -1.874450e-06
$iter
[1] 8
$estim.prec
[1] 6.103516e-05

> plot(cars)
> abline(lm(dist~speed,data=cars),col="blue")
> points(cars[30,],col="red",pch=20)

```

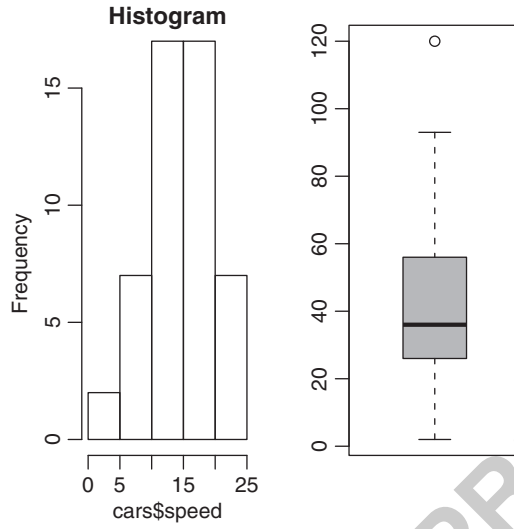


383

```

> par(mfrow=c(1,2))
> hist(cars$speed,main="Histogram")
> boxplot(cars$dist,col="orange")

```



See also

This link points to a reference card of the most useful R functions <http://cran.r-project.org/doc/contrib/Short-refcard.pdf>



UNCORRECTED PROOF