

```
species <- read.table("species.txt", header = T, colClasses = list(
head(species)
  pH Biomass Species
high 0.4692972 30
high 1.7308704 39
high 2.0897785 44
high 3.9257871 35
high 4.3667927 25
high 5.4819747 29
species_mod2 <- glm(Species ~ Biomass * pH, poisson, data = species)
summary(species_mod2)

all:
lm(formula = Species ~ Biomass * pH, family = poisson, data = species)

Deviance Residuals:
  Min       1Q   Median       3Q      Max
-2.4978  -0.7485  -0.0402   0.5575   2.2297

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    3.76812    0.06153   61.240 < 2e-16 ***
Biomass       -0.10713    0.01249  -8.577 < 2e-16 ***
pHlow         -0.81557    0.10284  -7.931 2.18e-15 ***
pHmid         -0.33146    0.09217  -3.596 0.00027 ***
Biomass:pHlow -0.15503    0.04003  -3.873 0.00008 ***
Biomass:pHmid -0.03189    0.02308  -1.382 0.166954

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

Dispersion parameter for poisson family (using method ML):
on deviance: 452.346 on 89 degrees of freedom
on deviance: 83.201 on 84 degrees of freedom

Number of Fisher Scoring iterations: 4
```

Third Edition

The

R

Book

Elinor Jones • Simon Harden • Michael J. Crawley

WILEY

The R Book

The R Book

Third Edition

Elinor Jones

University College London, UK

Simon Harden

University College London, UK

Michael J. Crawley

Imperial College London, UK

WILEY

This third edition first published 2023
© 2023 John Wiley & Sons Ltd

Edition History: John Wiley & Sons Ltd (1e, 2007; 2e, 2013)

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except as permitted by law. Advice on how to obtain permission to reuse material from this title is available at <http://www.wiley.com/go/permissions>.

The right of Elinor Jones, Simon Harden and Michael J. Crawley to be identified as the authors of this work has been asserted in accordance with law.

Registered Offices

John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, USA
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, UK

Editorial Office

9600 Garsington Road, Oxford, OX4 2DQ, UK

For details of our global editorial offices, customer services, and more information about Wiley products visit us at www.wiley.com.

Wiley also publishes its books in a variety of electronic formats and by print-on-demand. Some content that appears in standard print versions of this book may not be available in other formats.

Limit of Liability/Disclaimer of Warranty

In view of ongoing research, equipment modifications, changes in governmental regulations, and the constant flow of information relating to the use of experimental reagents, equipment, and devices, the reader is urged to review and evaluate the information provided in the package insert or instructions for each chemical, piece of equipment, reagent, or device for, among other things, any changes in the instructions or indication of usage and for added warnings and precautions. While the publisher and authors have used their best efforts in preparing this work, they make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including without limitation any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives, written sales materials or promotional statements for this work. The fact that an organization, website, or product is referred to in this work as a citation and/or potential source of further information does not mean that the publisher and authors endorse the information or services the organization, website, or product may provide or recommendations it may make. This work is sold with the understanding that the publisher is not engaged in rendering professional services. The advice and strategies contained herein may not be suitable for your situation. You should consult with a specialist where appropriate. Further, readers should be aware that websites listed in this work may have changed or disappeared between when this work was written and when it is read. Neither the publisher nor authors shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

Library of Congress Cataloging-in-Publication Data

Names: Jones, Elinor (Associate Professor), author. | Harden, Simon, author. | Crawley, Michael J., author.
Title: The R book / Elinor Jones, Simon Harden, and Michael J. Crawley.
Description: Third edition. | Hoboken, NJ : Wiley, 2022. | Includes bibliographical references and index.
Identifiers: LCCN 2022008352 (print) | LCCN 2022008353 (ebook) | ISBN 9781119634324 (cloth) | ISBN 9781119634409 (adobe pdf) | ISBN 9781119634430 (epub)
Subjects: LCSH: R (Computer program language) | Mathematical statistics--Data processing.
Classification: LCC QA276.45.R3 J662 2022 (print) | LCC QA276.45.R3 (ebook) | DDC 005.5/5--dc23/eng20220528
LC record available at <https://lcn.loc.gov/2022008352>
LC ebook record available at <https://lcn.loc.gov/2022008353>

Cover design: Wiley

Cover image: Courtesy of Simon Harden; © enjoynz/Getty Images

Set in 10/12pt HelveticaLTStd by Straive, Chennai, India

Contents

<i>List of Tables</i>	xxi
<i>Preface</i>	xxiii
<i>Acknowledgments</i>	xxv
<i>About the Companion Website</i>	xxvii
1 Getting Started	1
2 Technical Background	17
3 Essentials of the <i>R</i> Language	55
4 Data Input and Dataframes	207
5 Graphics	249
6 Graphics in More Detail	297
7 Tables	359
8 Probability Distributions in <i>R</i>	373
9 Testing	405
10 Regression	439
11 Generalised Linear Models	499
12 Generalised Additive Models	579
13 Mixed-Effect Models	601
14 Non-linear Regression	627
15 Survival Analysis	649
16 Designed Experiments	667
17 Meta-Analysis	699
18 Time Series	715
19 Multivariate Statistics	741

20	Classification and Regression Trees	761
21	Spatial Statistics	779
22	Bayesian Statistics	799
23	Simulation Models	823
	<i>Index</i>	839

Detailed Contents

<i>List of Tables</i>	xxi
<i>Preface</i>	xxiii
<i>Acknowledgments</i>	xxv
<i>About the Companion Website</i>	xxvii
1 Getting Started	1
1.1 Navigating the book	1
1.1.1 How to use this book	1
1.2 <i>R</i> vs. RStudio	3
1.3 Installing <i>R</i> and RStudio	3
1.4 Using RStudio	4
1.4.1 Using <i>R</i> directly via the console	5
1.4.2 Using text editors	5
1.5 The Comprehensive <i>R</i> Archive Network	7
1.5.1 Manuals	7
1.5.2 Frequently asked questions	8
1.5.3 Contributed documentation	8
1.6 Packages in <i>R</i>	8
1.6.1 Contents of packages	9
1.6.2 Finding packages	9
1.6.3 Installing packages	9
1.7 Getting help in <i>R</i>	11
1.7.1 Worked examples of functions	12
1.7.2 Demonstrations of <i>R</i> functions	13
1.8 Good housekeeping	13
1.8.1 Variable types	13
1.8.2 What's loaded or defined in the current session	14
1.8.3 Attaching and detaching objects	14
1.8.4 Projects	15
1.9 Linking to other computer languages	15
References	15

2	Technical Background	17
2.1	Mathematical functions	17
2.1.1	Logarithms and exponentials	18
2.1.2	Trigonometric functions	19
2.1.3	Power laws	20
2.1.4	Polynomial functions	22
2.1.5	Gamma function	24
2.1.6	Asymptotic functions	25
2.1.7	Sigmoid (S-shaped) functions	27
2.1.8	Biexponential function	28
2.1.9	Transformations of model variables	29
2.2	Matrices	30
2.2.1	Matrix multiplication	31
2.2.2	Diagonals of matrices	32
2.2.3	Determinants	33
2.2.4	Inverse of a matrix	35
2.2.5	Eigenvalues and eigenvectors	36
2.2.6	Solving systems of linear equations using matrices	39
2.3	Calculus	40
2.3.1	Differentiation	40
2.3.2	Integration	41
2.3.3	Differential equations	42
2.4	Probability	45
2.4.1	The central limit theorem	45
2.4.2	Conditional probability	49
2.5	Statistics	50
2.5.1	Least squares	51
2.5.2	Maximum likelihood	51
	Reference	53
3	Essentials of the R Language	55
3.1	Calculations	56
3.1.1	Complex numbers	57
3.1.2	Rounding	58
3.1.3	Arithmetic	59
3.1.4	Modular arithmetic	61
3.1.5	Operators	62
3.1.6	Integers	63
3.2	Naming objects	64
3.3	Factors	64
3.4	Logical operations	67
3.4.1	TRUE, T, FALSE, F	68
3.4.2	Testing for equality of real numbers	69
3.4.3	Testing for equality of non-numeric objects	70
3.4.4	Evaluation of combinations of TRUE and FALSE	72
3.4.5	Logical arithmetic	73
3.5	Generating sequences	74
3.5.1	Generating repeats	76
3.5.2	Generating factor levels	77

3.6	Class membership	78
3.7	Missing values, infinity, and things that are not numbers	82
3.7.1	Missing values: NA	83
3.8	Vectors and subscripts	86
3.8.1	Extracting elements of a vector using subscripts	87
3.8.2	Classes of vector	89
3.8.3	Naming elements within vectors	90
3.9	Working with logical subscripts	91
3.10	Vector functions	93
3.10.1	Obtaining tables using <code>tapply ()</code>	95
3.10.2	Applying functions to vectors using <code>sapply ()</code>	97
3.10.3	The <code>aggregate ()</code> function for grouped summary statistics	99
3.10.4	Parallel minima and maxima: <code>pmin</code> and <code>pmax</code>	100
3.10.5	Finding closest values	101
3.10.6	Sorting, ranking, and ordering	102
3.10.7	Understanding the difference between <code>unique ()</code> and <code>duplicated ()</code>	104
3.10.8	Looking for runs of numbers within vectors	106
3.10.9	Sets: <code>union ()</code> , <code>intersect ()</code> , and <code>setdiff ()</code>	108
3.11	Matrices and arrays	109
3.11.1	Matrices	111
3.11.2	Naming the rows and columns of matrices	112
3.11.3	Calculations on rows or columns of matrices	113
3.11.4	Adding rows and columns to matrices	115
3.11.5	The <code>sweep ()</code> function	117
3.11.6	Applying functions to matrices	119
3.11.7	Scaling a matrix	120
3.11.8	Using the <code>max.col ()</code> function	121
3.11.9	Restructuring a multi-dimensional array using <code>aperm ()</code>	123
3.12	Random numbers, sampling, and shuffling	126
3.12.1	The <code>sample ()</code> function	127
3.13	Loops and repeats	128
3.13.1	More complicated <code>while ()</code> loops	131
3.13.2	Loop avoidance	133
3.13.3	The slowness of loops	134
3.13.4	Do not 'grow' data sets by concatenation or recursive function calls	135
3.13.5	Loops for producing time series	136
3.14	Lists	138
3.14.1	Summarising lists and <code>lapply ()</code>	140
3.14.2	Manipulating and saving lists	142
3.15	Text, character strings, and pattern matching	147
3.15.1	Pasting character strings together	149
3.15.2	Extracting parts of strings	150
3.15.3	Counting things within strings	151
3.15.4	Upper and lower case text	153
3.15.5	The <code>match ()</code> function and relational databases	153
3.15.6	Pattern matching	155
3.15.7	Substituting text within character strings	159
3.15.8	Locations of a pattern within a vector	160

3.15.9	Comparing vectors using <code>%in%</code> and <code>which ()</code>	162
3.15.10	Stripping patterned text out of complex strings	163
3.16	Dates and times in <i>R</i>	164
3.16.1	Reading time data from files	165
3.16.2	Calculations with dates and times	168
3.16.3	Generating sequences of dates	170
3.16.4	Calculating time differences between the rows of a dataframe	173
3.16.5	Regression using dates and times	175
3.17	Environments	177
3.17.1	Using <code>attach ()</code> or not!	178
3.17.2	Using <code>attach ()</code> in this book	180
3.18	Writing <i>R</i> functions	181
3.18.1	Arithmetic mean of a single sample	181
3.18.2	Median of a single sample	182
3.18.3	Geometric mean	183
3.18.4	Harmonic mean	184
3.18.5	Variance	186
3.18.6	Variance ratio test	187
3.18.7	Using the variance	189
3.18.8	Plots and deparsing in functions	191
3.18.9	The <code>switch ()</code> function	192
3.18.10	Arguments in our function	193
3.18.11	Errors in our functions	195
3.18.12	Outputs from our function	196
3.19	Structure of <i>R</i> objects	200
3.20	Writing from <i>R</i> to a file	203
3.20.1	Saving data objects	203
3.20.2	Saving command history	204
3.20.3	Saving graphics or plots	204
3.20.4	Saving data for a spreadsheet	204
3.20.5	Saving output from functions to a file	205
3.21	Tips for writing <i>R</i> code	206
	References	206
4	Data Input and Dataframes	207
4.1	Working directory	207
4.2	Data input from files	208
4.2.1	Data input using <code>read.table ()</code> and <code>read.csv ()</code>	208
4.2.2	Input from files using <code>scan ()</code>	210
4.2.3	Reading data from a file using <code>readLines ()</code>	213
4.3	Data input directly from the web	215
4.4	Built-in data files	215
4.5	Dataframes	216
4.5.1	Subscripts and indices	220
4.5.2	Selecting rows from the dataframe at random	222
4.5.3	Sorting dataframes	223
4.5.4	Using logical conditions to select rows from the dataframe	229

4.5.5	Omitting rows containing missing values, <code>NA</code>	232
4.5.6	A dataframe with row names instead of row numbers	235
4.5.7	Creating a dataframe from another kind of object	236
4.5.8	Eliminating duplicate rows from a dataframe	239
4.5.9	Dates in dataframes	239
4.6	Using the <code>match()</code> function in dataframes	241
4.6.1	Merging two dataframes	243
4.7	Adding margins to a dataframe	245
4.7.1	Summarising the contents of dataframes	247
5	Graphics	249
5.1	Plotting principles	249
5.1.1	Axes labels and titles	251
5.1.2	Plotting symbols and colours	251
5.1.3	Saving graphics	254
5.2	Plots for single variables	255
5.2.1	Histograms vs. bar charts	255
5.2.2	Histograms	256
5.2.3	Density plots	260
5.2.4	Boxplots	261
5.2.5	Dotplots	262
5.2.6	Bar charts	263
5.2.7	Pie charts	264
5.3	Plots for showing two numeric variables	265
5.3.1	Scatterplot	265
5.3.2	Plots with many identical values	270
5.4	Plots for numeric variables by group	272
5.4.1	Boxplots by group	272
5.4.2	Dotplots by group	274
5.4.3	An inferior (but popular) option	275
5.5	Plots showing two categorical variables	277
5.5.1	Grouped bar charts	277
5.5.2	Mosaic plots	277
5.6	Plots for three (or more) variables	279
5.6.1	Plots of all pairs of variables	279
5.6.2	Incorporating a third variable on a scatterplot	280
5.6.3	Basic 3D plots	281
5.7	Trellis graphics	283
5.7.1	Panel boxplots	285
5.7.2	Panel scatterplots	286
5.7.3	Panel barplots	289
5.7.4	Panels for conditioning plots	290
5.7.5	Panel histograms	291
5.7.6	More panel functions	292
5.8	Plotting functions	293
5.8.1	Two-dimensional plots	293
5.8.2	Three-dimensional plots	295
	References	295

6	Graphics in More Detail	297
6.1	More on colour	297
6.1.1	Colour palettes with categorical data	297
6.1.2	The <code>RColorBrewer</code> package	299
6.1.3	Foreground colours	302
6.1.4	Background colours	302
6.1.5	Background colour for legends	303
6.1.6	Different colours for different parts of the graph	304
6.1.7	Full control of colours in plots	305
6.1.8	Cross-hatching and grey scale	307
6.2	Changing the look of graphics	308
6.2.1	Shape and size of plot	308
6.2.2	Multiple plots on one screen	309
6.2.3	Tickmarks and associated labels	309
6.2.4	Font of text	311
6.3	Adding items to plots	311
6.3.1	Adding text	311
6.3.2	Adding smooth parametric curves to a scatterplot	313
6.3.3	Fitting non-parametric curves through a scatterplot	314
6.3.4	Connecting observations	316
6.3.5	Adding shapes	321
6.3.6	Adding mathematical and other symbols	322
6.4	The grammar of graphics and <code>ggplot2</code>	326
6.4.1	Basic structure	327
6.4.2	Examples	327
6.5	Graphics cheat sheet	330
6.5.1	Text justification, <code>adj</code>	332
6.5.2	Annotation of graphs, <code>ann</code>	332
6.5.3	Delay moving on to the next in a series of plots, <code>ask</code>	332
6.5.4	Control over the axes, <code>axis</code>	332
6.5.5	Background colour for plots, <code>bg</code>	333
6.5.6	Boxes around plots, <code>bty</code>	334
6.5.7	Size of plotting symbols using the character expansion function, <code>cex</code>	334
6.5.8	Changing the shape of the plotting region, <code>plt</code>	335
6.5.9	Locating multiple graphs in non-standard layouts using <code>fig</code>	336
6.5.10	Two graphs with a common <i>X</i> scale but different <i>Y</i> scales using <code>fig</code>	336
6.5.11	The <code>layout</code> function	338
6.5.12	Creating and controlling multiple screens on a single device	340
6.5.13	Orientation of numbers on the tick marks, <code>las</code>	341
6.5.14	Shapes for the ends and joins of lines, <code>lend</code> and <code>ljoin</code>	342
6.5.15	Line types, <code>lty</code>	343
6.5.16	Line widths, <code>lwd</code>	343
6.5.17	Several graphs on the same page, <code>mfrow</code> and <code>mfcop</code>	344
6.5.18	Margins around the plotting area, <code>mar</code>	345
6.5.19	Plotting more than one graph on the same axes, <code>new</code>	346
6.5.20	Outer margins, <code>oma</code>	347
6.5.21	Packing graphs closer together	348
6.5.22	Square plotting region, <code>pty</code>	350

6.5.23	Character rotation, <code>srt</code>	350
6.5.24	Rotating the axis labels	351
6.5.25	Tick marks on the axes	351
6.5.26	Axis styles	353
6.5.27	Summary	353
	References	357
7	Tables	359
7.1	Tabulating categorical or discrete data	359
7.1.1	Tables of counts	359
7.1.2	Tables of proportions	360
7.2	Tabulating summaries of numeric data	362
7.2.1	General summaries by group	362
7.2.2	Bespoke summaries by group	364
7.3	Converting between tables and dataframes	367
7.3.1	From a table to a dataframe	367
7.3.2	From a dataframe to a table	370
	Reference	371
8	Probability Distributions in <i>R</i>	373
8.1	Probability distributions: the basics	374
8.1.1	Discrete and continuous probability distributions	374
8.1.2	Describing probability distributions mathematically	374
8.1.3	Independence	375
8.2	Probability distributions in <i>R</i>	376
8.3	Continuous probability distributions	377
8.3.1	The Normal (or Gaussian) distribution	377
8.3.2	The Uniform distribution	380
8.3.3	The Chi-squared distribution	381
8.3.4	The F distribution	382
8.3.5	Student's <i>t</i> distribution	383
8.3.6	The Gamma distribution	385
8.3.7	The Exponential distribution	386
8.3.8	The Beta distribution	387
8.3.9	The Lognormal distribution	388
8.3.10	The Logistic distribution	389
8.3.11	The Weibull distribution	390
8.3.12	Multivariate Normal distribution	390
8.4	Discrete probability distributions	392
8.4.1	The Bernoulli distribution	392
8.4.2	The Binomial distribution	392
8.4.3	The Geometric distribution	395
8.4.4	The Hypergeometric distribution	397
8.4.5	The Multinomial distribution	398
8.4.6	The Poisson distribution	399
8.4.7	The Negative Binomial distribution	400
8.5	The central limit theorem	402
	References	404

9	Testing	405
9.1	Principles	406
9.1.1	Defining the question to be tested	406
9.1.2	Assumptions	408
9.1.3	Interpreting results	408
9.2	Continuous data	410
9.2.1	Single population average	410
9.2.2	Two population averages	412
9.2.3	Multiple population averages	414
9.2.4	Population distribution	415
9.2.5	Checking and testing for normality	417
9.2.6	Comparing variances	419
9.3	Discrete and categorical data	421
9.3.1	Sign test	421
9.3.2	Test to compare proportions	423
9.3.3	Contingency tables	427
9.3.4	Testing contingency tables	429
9.4	Bootstrapping	431
9.5	Multiple tests	433
9.6	Power and sample size calculations	434
9.7	A table of tests	436
	References	437
10	Regression	439
10.1	The simple linear regression model	440
10.1.1	Model format and assumptions	440
10.1.2	Building a simple linear regression model	443
10.2	The multiple linear regression model	446
10.2.1	Model format and assumptions	446
10.2.2	Building a multiple linear regression model	447
10.2.3	Categorical covariates	449
10.2.4	Interactions between covariates	454
10.3	Understanding the output	458
10.3.1	Residuals	458
10.3.2	Estimates of coefficients	459
10.3.3	Testing individual coefficients	459
10.3.4	Residual standard error	460
10.3.5	R^2 and its variants	460
10.3.6	The regression F -test	460
10.3.7	ANOVA: Same model, different output	461
10.3.8	Extracting model information	464
10.4	Fitting models	465
10.4.1	The principle of parsimony	465
10.4.2	First plot the data	467
10.4.3	Comparing nested models	468
10.4.4	Comparing non-nested models	470
10.4.5	Dealing with large numbers of covariates	471
10.5	Checking model assumptions	473
10.5.1	Residuals and standardised residuals	473
10.5.2	Checking for linearity	474

10.5.3	Checking for homoscedasticity of errors	476
10.5.4	Checking for normality of errors	476
10.5.5	Checking for independence of errors	478
10.5.6	Checking for influential observations	479
10.5.7	Checking for collinearity	481
10.5.8	Improving fit	483
10.6	Using the model	491
10.6.1	Interpretation of model	491
10.6.2	Making predictions	495
10.7	Further types of regression modelling	497
	References	498
11	Generalised Linear Models	499
11.1	How GLMs work	499
11.1.1	Error structure	499
11.1.2	Linear predictor	500
11.1.3	Link function	501
11.1.4	Model checking	502
11.1.5	Interpretation and prediction	506
11.2	Count data and GLMs	507
11.2.1	A straightforward example	508
11.2.2	Dispersion	511
11.2.3	An alternative to Poisson counts	516
11.3	Count table data and GLMs	522
11.3.1	Log-linear models	522
11.3.2	All covariates might be useful	522
11.3.3	Spine plot	534
11.4	Proportion data and GLMs	537
11.4.1	Theoretical background	538
11.4.2	Logistic regression with binomial errors	541
11.4.3	Predicting x from y	544
11.4.4	Proportion data with categorical explanatory variables	545
11.4.5	Binomial GLM with ordered categorical covariates	550
11.4.6	Binomial GLM with categorical and continuous covariates	556
11.4.7	Revisiting lizards	559
11.5	Binary Response Variables and GLMs	560
11.5.1	A straightforward example	562
11.5.2	Graphical tests of the fit of the logistic curve to data	564
11.5.3	Mixed covariate types with a binary response	567
11.5.4	Spine plot and logistic regression	570
11.6	Bootstrapping a GLM	574
	References	577
12	Generalised Additive Models	579
12.1	Smoothing example	580
12.2	Straightforward examples of GAMs	583
12.3	Background to using GAMs	588
12.3.1	Smoothing	588
12.3.2	Suggestions for using <code>gam</code> ()	588

12.4	More complex GAM examples	589
12.4.1	Back to <code>Ozone</code>	590
12.4.2	An example with strongly humped data	592
12.4.3	GAMs with binary data	596
12.4.4	Three-dimensional graphic output from <code>gam</code>	598
	References	599
13	Mixed-Effect Models	601
13.1	Regression with categorical covariates	601
13.2	An alternative method: random effects	602
13.3	Common data structures where random effects are useful	603
13.3.1	Nested (hierarchical) structures	604
13.3.2	Non-nested structures	604
13.3.3	Longitudinal structures	605
13.4	<i>R</i> packages to deal with mixed effects models	605
13.4.1	The <code>nlme</code> package	605
13.4.2	The <code>lme4</code> package	606
13.4.3	Methods for fitting mixed models	606
13.5	Examples of implementing random effect models	607
13.5.1	Multilevel data (two levels)	607
13.5.2	Multilevel data (three levels)	611
13.5.3	Designed experiment: split-plot	614
13.5.4	Longitudinal data	617
13.6	Generalised linear mixed models	622
13.6.1	Logistic mixed model	622
13.7	Alternatives to mixed models	625
	References	625
14	Non-linear Regression	627
14.1	Example: modelling deer jaw bone length	628
14.1.1	An exponential model for the deer data	629
14.1.2	A Michaelis–Menten model for the deer data	632
14.1.3	Comparison of the exponential and the Michaelis–Menten model	634
14.2	Example: grouped data	634
14.3	Self-starting functions	638
14.3.1	Self-starting Michaelis–Menten model	638
14.3.2	Self-starting asymptotic exponential model	640
14.3.3	Self-starting logistic	642
14.3.4	Self-starting four-parameter logistic	643
14.4	Further considerations	645
14.4.1	Model checking	645
14.4.2	Confidence intervals	647
	References	648
15	Survival Analysis	649
15.1	Handling survival data	649
15.1.1	Structure of a survival dataset	649
15.1.2	Survival data in <i>R</i>	652

15.2	The survival and hazard functions	652
15.2.1	Non-parametric estimation of the survival function	653
15.2.2	Parametric estimation of the survival function	654
15.3	Modelling survival data	655
15.3.1	The data	657
15.3.2	The Cox proportional hazard model	658
15.3.3	Accelerated failure time models	660
15.3.4	Cox proportional hazard or a parametric model?	665
	References	665
16	Designed Experiments	667
16.1	Factorial experiments	667
16.1.1	Expanding data	672
16.2	Pseudo-replication	673
16.2.1	Split-plot effects	673
16.2.2	Removing pseudo-replication	675
16.2.3	Derived variable analysis	676
16.3	Contrasts	677
16.3.1	Contrast coefficients	678
16.3.2	An example of contrasts using R	679
16.3.3	Model simplification for contrasts	684
16.3.4	Helmert contrasts	688
16.3.5	Sum contrasts	689
16.3.6	Polynomial contrasts	691
16.3.7	Contrasts with multiple covariates	694
	References	698
17	Meta-Analysis	699
17.1	Elements of a meta-analysis	699
17.1.1	Choosing studies for a meta-analysis	700
17.1.2	Effects and effect size	700
17.1.3	Weights	701
17.1.4	Fixed vs. random effect models	701
17.2	Meta-analysis in R	703
17.2.1	Formatting information from studies	703
17.2.2	Computing the inputs of a meta-analysis	703
17.2.3	Conducting the meta-analysis	706
17.3	Examples	707
17.3.1	Meta-analysis Of scaled differences	707
17.4	Meta-analysis of categorical data	711
	References	714
18	Time Series	715
18.1	Moving average	715
18.2	Blowflies	717
18.3	Seasonal data	723
18.3.1	Point of view	724
18.3.2	Built in <code>ts</code> () functions	724
18.3.3	Cycles	726
18.3.4	Testing for a time series trend	728

18.4	Multiple time series	729
18.5	Some theoretical background	730
18.5.1	Autocorrelation	731
18.5.2	Autoregressive models	732
18.5.3	Partial autocorrelation	732
18.5.4	Moving average models	732
18.5.5	More general models: ARMA and ARIMA	733
18.6	ARIMA example	733
18.7	Simulation of time series	735
	Reference	739
19	Multivariate Statistics	741
19.1	Visualising data	742
19.2	Multivariate analysis of variance	743
19.3	Principal component analysis	745
19.4	Factor analysis	748
19.5	Cluster analysis	751
19.5.1	<i>k</i> -means	751
19.6	Hierarchical cluster analysis	754
19.7	Discriminant analysis	756
19.8	Neural networks	758
	References	760
20	Classification and Regression Trees	761
20.1	How CARTs work	763
20.2	Regression trees	764
20.2.1	The <code>tree</code> package	764
20.2.2	The <code>rpart</code> package	765
20.2.3	Comparison with linear regression	767
20.2.4	Model simplification	769
20.3	Classification trees	771
20.3.1	Classification trees with categorical explanatory variables	771
20.3.2	Classification trees for replicated data	773
20.4	Looking for patterns	775
	References	777
21	Spatial Statistics	779
21.1	Spatial point processes	779
21.1.1	How can we check for randomness?	781
21.1.2	Models	785
21.1.3	Marks	790
21.2	Geospatial statistics	793
21.2.1	Models	794
	References	798
22	Bayesian Statistics	799
22.1	Components of a Bayesian Analysis	800
22.1.1	The likelihood (the model and data)	800
22.1.2	Priors	801
22.1.3	The Posterior	802

22.1.4	Markov chain Monte Carlo (MCMC)	803
22.1.5	Considerations for MCMC	803
22.1.6	Inference	805
22.1.7	The Pros and Cons of going Bayesian	806
22.2	Bayesian analysis in <i>R</i>	806
22.2.1	Installing JAGS	807
22.2.2	Running JAGS in <i>R</i>	807
22.2.3	Writing BUGS models	808
22.3	Examples	810
22.3.1	MCMC for a simple linear regression	810
22.3.2	MCMC for longitudinal data	814
22.4	MCMC for a model with binomial errors	818
	References	821
23	Simulation Models	823
23.1	Temporal dynamics	823
23.1.1	Chaotic dynamics in population size	823
23.1.2	Investigating the route to chaos	825
23.2	Spatial simulation models	826
23.2.1	Meta-population dynamics	826
23.2.2	Coexistence resulting from spatially explicit (local) density dependence	829
23.2.3	Pattern generation resulting from dynamic interactions	834
23.3	Temporal and spatial dynamics: random walk	837
	References	838
	<i>Index</i>	839

List of Tables

Table 1.1	Libraries used in this book that come supplied as part of the base package of <i>R</i>	8
Table 1.2	Task Views on CRAN	10
Table 3.1	Mathematical functions	61
Table 3.2	Common operators	62
Table 3.3	Logical and relational operations	67
Table 3.4	Data types	80
Table 3.5	Vector functions	94
Table 3.6	Format codes for dates and times	167
Table 3.7	Escape sequences for use with <code>cat ()</code>	199
Table 4.1	Correctly set out dataset for importing into a dataframe	216
Table 4.2	Dataset that will not form a dataframe correctly	217
Table 4.3	Dataset that will form a dataframe correctly	217
Table 4.4	Selecting parts of a dataframe called <code>df_dummy</code>	223
Table 5.1	Plotting single variables	255
Table 6.1	Orientation and sizes of labels	310
Table 6.2	Drawing mathematical expressions in text	323
Table 6.3	Graphical parameters and their default values	354
Table 8.1	Some commonly used probability distributions supported by <i>R</i>	376
Table 9.1	Tests used in Chapter 9	436
Table 10.1	Functions for various regression models	497
Table 10.2	Frequently used functions to extract information about regression models	498
Table 11.1	Common members of the exponential family	501

Table 14.1	Useful non-linear functions	628
Table 14.2	Useful non-linear self-starting functions	639
Table 15.1	Common parametric forms of the survival and hazard functions	654
Table 17.1	Data from Study A	711

Preface

R is the most powerful tool in the known universe for carrying out statistical analysis, and it's free! This book is aimed at those who wish to carry out such work – exploring, plotting, and modelling data – but who do not have much experience in *R* and/or statistics. *R* is described from scratch with instructions for loading and getting going with the software in Chapter 1 and a description of its essential elements in Chapter 3. Later chapters discuss statistical methods and are written so that they can be used either as a beginner's guide or as a reference manual on particular procedures in *R*. The theory behind the analyses is covered in enough depth, we hope, to make it comprehensible but without overburdening the reader with too much mathematics. The datasets used to illustrate various analyses are available at <https://www.wiley.com/go/jones/therbook3e>.

Using *R* has become far simpler with the introduction of RStudio, which is also free (other editors are available). RStudio provides a friendly front end and easy access to tools, all of which seem a long way from *R*'s original rather forbidding command prompt. This book assumes the use of RStudio rather than using *R* directly, but the code presented will work using the latter setup too.

While there is still the usual hurdle of getting to know powerful software, the benefits, particularly in graphics and modelling, far outweigh the effort. Academic papers in many disciplines routinely use and report results using *R*. In addition, the open-source nature of the software means that users have added extra functionality by writing packages to broaden *R*'s capabilities. There are currently over 18,000 packages that, together with useful links and information, can be found at the official *R* distribution site, CRAN: <https://cran.r-project.org/>.

This book is contingent upon the existence of *R*. Those involved are too numerous to mention, but we are hugely grateful to all involved in its creation and continuing evolution. When you use *R*, *R* packages (e.g. *spatstat*), and RStudio, please cite them. Up-to-date citation details for each of these can be found by typing the following in *R*, respectively:

```
citation ()  
citation ("spatstat")  
RStudio.Version ()
```

Elinor Jones
Simon Harden
Michael J. Crawley
August 2022

Acknowledgments

This book would not exist without its previous editions so thanks, firstly, to the originating author, Michael J. Crawley.

It has been a pleasure to revise The R Book to create this third edition. We are very grateful to Professor Crawley for allowing us to use materials from previous versions, including his fantastic array of datasets that make a welcome return in this edition.

Finally, we would like to thank the Department of Statistical Science at University College London for giving us time and space to complete the book during a difficult period for everybody.

Elinor Jones
Simon Harden
August 2022

About the Companion Website

This book is accompanied by a companion website.

www.wiley.com/go/jones/therbook3e



This website include: Datasets



Getting Started

1.1 Navigating the book

The material covered in this book has been arranged by topic. The first few chapters cover the essentials, including basic technical knowledge (Chapter 2), the fundamentals of *R* (Chapter 3), and data handling in *R* (Chapter 4). Subsequent chapters deal with statistical procedures, including graphics (Chapters 5 and 6), statistical testing (Chapter 9), and common statistical models (from Chapter 10).

To make navigating the book easier, the following conventions will be used:

- New terms are highlighted in **bold** when first used;
- *R* functions and function arguments written in-line are highlighted in red, for example the `plot ()` function and the `pch` argument (note the use of the round brackets when referring to functions);
- Stand-alone *R* code is written in red, with output in blue, for example:

```
1+3  
[1] 4
```

- Datasets, variable names, model names, and so on, are written in `typewriter` font;
- *R* packages (see Section 1.6) are highlighted in blue, for example `MASS`.

1.1.1 How to use this book

This book is intended to serve a wide audience from complete beginners through to those in need of an *R* reference manual. Below, we offer advice on how to use the book depending on level of experience in statistics and computing.

Beginner in both computing and statistics

The book is structured principally with such a reader in mind. There are six key things to learn: how to arrange data, how to read data into *R*, how to check data once within *R*, how to select an appropriate analysis, how to interpret the output, and how to present the analysis for publication. A thorough understanding of the basics is essential before trying to do the more complicated things, so we recommend studying Chapters 3 to 4 carefully to begin with. Do all of the exercises that are illustrated in the text on your own computer.

Now comes the hard part, which is selecting the right statistics to use. Model choice is extremely important and is the thing that will develop most with experience. Don't be afraid to ask for expert help with this. Never do an analysis that is more complicated than it needs to be, so start by reading about graphical representations of data (Chapters 5 and 6). Sometimes this is all that's needed.

Student needing help with project work

A good understanding of variable types is key (broadly, variables are either numeric or categorical, see Section 1.8.1). An analysis of a dataset will depend – at least in part – on the type of variables in the dataset and the research question of interest. Does the research question point to a particular 'response' variable, and if so, what type of variable is this?

From here, the first port of call is to plot or tabulate the data, depending on the nature of the variables (see Chapters 5–7). That might be enough in itself, or further statistical analyses might be needed. For example, if the response variable (if any) is a count, consider using hypothesis tests (Chapter 9), tables (Chapter 7), or possibly a model (Chapter 11). If the response variable is a continuous measure (e.g. a weight), then consider using hypothesis tests (Chapter 9), or a regression model (Chapter 10).

Done some R and some statistics, but keen to learn more of both

The best plan is to skim quickly through the introductory material in case there is anything new to be learned. It is a good idea to read Chapter 3 on the fundamentals of the *R* language and Chapters 5 and 6 on graphics. Much of the rest of the book is organised by analysis type making it easy to jump to the relevant chapter.

Done regression, but want to learn more advanced statistical modelling

For readers who have experience of regression in another language, the best plan is to go directly to Chapters 10 and 11 to see how the output from linear models is handled by *R*. Familiarity with data input and dataframes is essential (Chapter 4), then the chapters on more advanced modelling should be accessible.

Experienced in statistics, but a beginner in R

The first thing is to get a thorough understanding of dataframes and data input to *R*, so start with Chapter 4. Then, chapters on statistical modelling should be accessible. It is a good idea to browse, for example Chapters 9 (Testing) and 10 (Regression) to understand the output from *R*. Working through Chapters 5 and 6 will provide the foundations of graphics in *R*.

Experienced in computing, but a beginner in R

Well-written *R* code is highly intuitive and very readable. The most unfamiliar parts of *R* are likely to be the way it handles functions and the way it deals with environments. It is impossible to anticipate the order in which more advanced users are likely to encounter material and hence want to learn about specific features of the language, but vectorised calculations, subscripts on dataframes, function-writing and suchlike are bound to crop up early (Chapter 3). When faced with an unfamiliar name in some code, just type the name immediately after a question mark; for example to find out more about the `rnbinom ()` function, type:

```
?rnbinom
```

Recognizing mathematical functions is quite straightforward because of their names and the fact that their arguments are enclosed in round brackets `()`. Subscripts on objects have square brackets `[]`. Multi-line blocks of *R* code are enclosed within curly brackets `{ }`. The idea of lists might be new, or applying functions to lists; elements within lists have double square brackets `[[]]`.

Look at the sections at the start of Chapter 3 as a starting point. The index is probably the most sensible entry point for queries about specifics.

Familiar with statistics and computing, but need a friendly reference manual

For information about a *topic*, use the chapter list and the Detailed Contents to find the most appropriate section. For aspects of the *R language*, look at the sections mentioned at the start of Chapter 3. Spending time browsing the contents of general material such as Chapters 5 and 6 on graphics is a good idea.

Get used to *R*'s help pages. Each *R* function has a help page which can be accessed by typing a question mark followed directly by the function name. To find out what all the graphics parameters mean, for instance, just type:

```
?par
```

1.2 R vs. RStudio

R is a powerful open-source software for statistical computing (R Core Team, 2021). It can be used directly, or for a more pleasing user experience, can be used via the RStudio interface which is freely available (RStudio Team, 2020). We strongly recommend using RStudio rather than *R* directly as it makes managing workspace easy and avoids some of *R*'s rather cumbersome features. The rest of the book will assume the use of RStudio, but all code presented will work – and provide identical results – if used in ‘native’ *R* instead.

We will generally write ‘*R*’ instead of ‘RStudio’ throughout this book.

1.3 Installing R and RStudio

You will need to install both *R* and RStudio. Both will run under Windows, a number of flavours of Linux (more for *R* than for RStudio so check the links below) and even Apple’s Mac OS X.

First download and install *R*. Note that this needs to be done first before attempting to download RStudio.

- Go to the CRAN project webpage <https://cran.r-project.org/mirrors.html> and choose the closest CRAN site to you (e.g. Imperial College London). It doesn't matter too much which of these is chosen if several are close by;
- Select the link for *Download R for ...*, for your operating system;
- Follow the instructions, noting that the default set-up is perfectly adequate.

Now download RStudio.

- Go to the RStudio webpage <https://rstudio.com/products/rstudio/download/#download> and select the 'Download' for RStudio Desktop. The free version is generally adequate and is certainly so for this book.
- Follow the instructions. At some point you'll be asked to locate where *R* has been installed. Remember that RStudio is just an *R* interface.

Rather than downloading *R* and RStudio, there is the option of accessing the software online via RStudio Cloud (<https://www.rstudio.com/products/cloud/>). There are some advantages to using RStudio online, for example when working on a group project. However, for individual use, it is likely that downloading *R* and RStudio is the best way forward.

1.4 Using RStudio

Once installed, open RStudio. The screen is divided into three parts as in Figure 1.1.

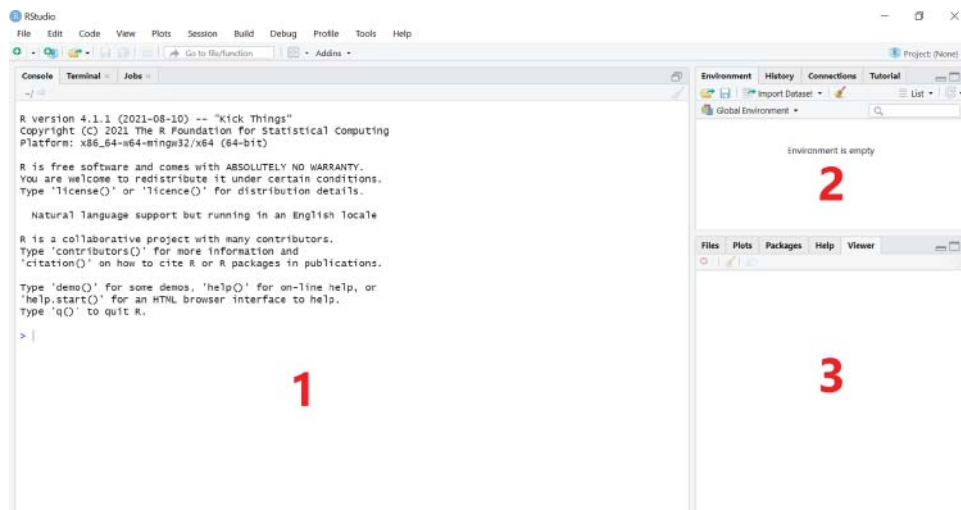


Figure 1.1 RStudio windows. RStudio, PBC

On the left is the **console**, numbered 1 in Figure 1.1. This is *R*. All code will be passed to the console where it will be executed, and numerical output will also be displayed here. The console displays the version number of *R*, its date, and version name (always comedic).

In the top right-hand corner is the **workspace**, numbered 2 in Figure 1.1. This is the control centre and gives an at-a-glance overview of what has been done so far in the session.

The bottom right corner, numbered 3 in Figure 1.1, hosts a number of things. Switch between them by clicking on the relevant tabs:

- **Files:** Shows the accessible file directories (more on this in Section 1.8.4);
- **Plots:** This is where plots and other graphics will be displayed;
- **Packages:** Lists packages that have been installed and provides functionality for installing others (more on this in Section 1.6);
- **Help:** As the name might suggest, help with various functions or procedures can be found here (more on this in Section 1.7);
- **Viewer:** Used for viewing local web content.

1.4.1 Using *R* directly via the console

Before exploring further, we'll return to the console. Below the header – which contains useful information about version number, citation, and a health warning – is a blank line with a `>` symbol in the left-hand margin. This is called the **prompt** and is *R*'s way of saying 'What now?'. Commands can be typed in directly here, though we suggest a more efficient way of working via **text editors** (see Section 1.4.2).

To begin with, we can use the console as a calculator, for example typing in the following command then pressing enter on the keyboard to execute the command (for neatness, we don't present the `>` at the start of each line of code in this book):

```
2 + 3
```

```
[1] 5
```

When working, a `+` is sometimes displayed at the left-hand side of the screen instead of `>`. This means that the last command typed is incomplete. The most common cause of this is forgetting one or more brackets. If what's missing is clear (e.g. a final right-hand bracket), then just type the missing character and press enter, at which point the command will execute. If a mistake has been made, then press the Esc key and the command line prompt `>` will reappear. Use the Up arrow key to scroll through previous commands, then use the Left and Right arrow keys to navigate to the mistake and correct it.

1.4.2 Using text editors

Writing commands in the console directly is rarely a good idea. It is good practice to keep a record of the code we use, which makes correcting mistakes, updating analyses, or just running the command(s) again very easy. RStudio has a built-in text editor to store and execute code.

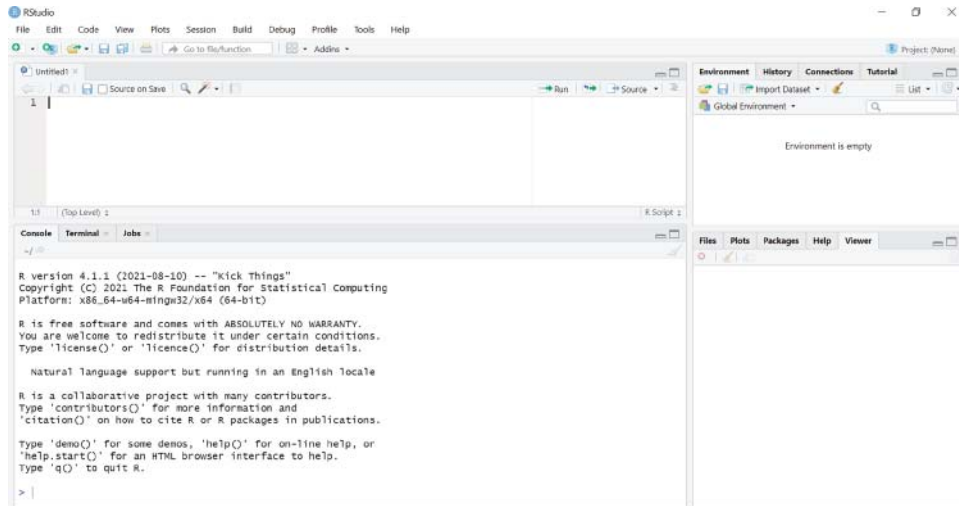


Figure 1.2 RStudio windows with text editor. RStudio, PBC

A failsafe way of opening a blank text editor, which doesn't depend on the operating system of your machine, is to go to *File*, then *New File*, then *R Script*. It will appear in the top left hand corner as in Figure 1.2.

The text editor is where we write commands, using a new line for each one. Click the text editor to activate it, before writing the following:

```
2 + 3
3 * 6
exp (2)
```

To run commands, highlight the relevant lines and click 'Run' (top right corner of the text editor) or press Ctrl and Enter, simultaneously. Output will be displayed in the console (bottom left). For readability, output will be shown directly beneath each of the relevant command throughout this book like this:

```
2 + 3
[1] 5

3 * 6
[1] 18

exp (2)
[1] 7.389056
```

The commands so far request calculations to be performed, but we can also define **objects** (see Chapter 3 for details), for example `a` is assigned (`<-`) the value 5 while we define `b` to be $\ln(10)$:

```
a <- 5
b <- log (10)
```

When we run this, we notice two things:

- there is no output in the console (because all we've done is define two objects);
- the **Environment** tab in the workspace has been populated with the definitions of `a` and `b`.

It is helpful to understand that *R* is an **object-orientated** programming (OOP) language: it is based on applying *actions* (commands) on *objects*. For example, a dataset which we load into *R* (see Chapter 4) will be considered by *R* as an object. Any action applied (e.g. finding the mean for each variable in the dataset) will be an action on a particular object (the dataset in this case). An object doesn't have to be a dataset, however, as we saw above.

The best way to learn *R* and RStudio is to play with them. The introduction here gives a very brief overview but is in no way complete. A good place to start is with RStudio cheatsheets <https://www.rstudio.com/resources/cheatsheets/>.

1.5 The Comprehensive *R* Archive Network

CRAN <https://cran.r-project.org/> is the first port of call for everything to do with *R*. It is from here that you download and install *R* (see Section 1.3), find contributed packages to solve particular problems (see Section 1.6), find the answers to frequently asked questions, read about the latest developments, get programming tips, and much more besides.

It is well worth browsing through *The R Journal*, accessible via the CRAN webpage. This is the refereed journal of the *R* project for statistical computing. It features short- to medium-length articles covering topics that might be of interest to users or to developers of *R*, including

- **Add-on packages:** Short introductions to or reviews of *R* extension packages.
- **Changes in *R*:** Details of recent changes to *R*.
- **Applications:** Demonstrating how a new or existing technique can be applied in an area of current interest using *R*, providing a fresh view of such analyses in *R* that is of benefit beyond the specific application.

1.5.1 Manuals

There are several manuals available on CRAN, for example (descriptions are taken from the webpage):

- *An Introduction to R* gives an introduction to the language and how to use *R* for doing statistical analysis and graphics.
- A draft of the *R Language Definition*, which documents the language *per se* – that is, the objects that it works on, and the details of the expression evaluation process, which are useful to know when programming *R* functions. This is perhaps the most important of all the manuals.

- *Writing R Extensions* covers how to create your own packages, write *R* help files, and use the foreign language (C, C++, Fortran, ...) interfaces.
- *Data Import/Export* describes the import and export facilities available either in *R* itself or via packages which are available from CRAN.
- *R Installation and Administration*, which is self-explanatory.

These manuals are also available in RStudio by going to the Help tab in the bottom right-hand corner and clicking the 'home' icon.

The most useful part of the site, however, is the Search facility. This is a good starting point for investigating the contents of most of the *R* documents, functions, and searchable mail archives.

1.5.2 Frequently asked questions

R has three collections of answers to FAQs:

- the *R* FAQ, which contains useful information for users on all platforms (Linux, Mac, Unix, Windows);
- the *R* Mac OS X FAQ for all users of Apple operating systems;
- the *R* Windows FAQ for all users of Microsoft operating systems.

Read the first of these, plus the appropriate one for your platform.

1.5.3 Contributed documentation

This contains a wide range of longer (more than 100 pages) and shorter manuals, tutorials, and exercises provided by users of *R*. You should browse these to find the ones most relevant to your needs.

1.6 Packages in *R*

A lot can be done with *R* or RStudio 'straight out of the box', also known as **base-*R***. Table 1.1 lists some of the packages that come supplied as part of the base-*R* installation.

Table 1.1 Libraries used in this book that come supplied as part of the base package of *R*.

Package name	Functionality
<code>lattice</code>	graphics for panel plots or trellis graphs
<code>MASS</code>	package associated with Venables and Ripley's book entitled <i>Modern Applied Statistics using S-PLUS</i>
<code>mgcv</code>	generalised additive models
<code>nlme</code>	mixed-effects models (both linear and nonlinear)
<code>nnet</code>	feed-forward neural networks and multinomial log-linear models
<code>spatial</code>	functions for kriging and point pattern analysis
<code>survival</code>	survival analysis, including penalised likelihood

However, there is a huge community of *R* users who contribute to its functionality via **packages**. A package contains additional functionality for *R* that can be loaded during a session. Navigating contributed packages can be tricky simply because there are so many of them, and the name of the package is not always as indicative of its function as one might hope.

Viewing existing packages can be done in RStudio by clicking on the Packages tab. Clicking the box next to a package loads it. A far better way of loading a package is to do so via the `library ()` function, which also means we have it as part of our code. For example, to load the `MASS` package (Venables and Ripley, 2002), which has a wide range of useful functions and datasets, type:

```
library (MASS)
```

See Section 1.6.3 for information on installing new packages.

1.6.1 Contents of packages

It is easy to use the `help` function to discover the contents of library packages. Supposing that we wanted to find out about the contents of the `spatial` package, we'd type:

```
library (help = spatial)
```

This brings up general information about the package in a new tab of the text editor in RStudio, including a list of all the functions and data sets.

To find out how to use, say, Ripley's K (`kfn ()`) from `spatial`, we load the package and then use `?` to query the function:

```
library (spatial)  
?Kfn
```

1.6.2 Finding packages

There is no comprehensive cross-referenced index of packages, but there is a very helpful feature called 'Task Views' on the CRAN website, which explains the packages available under a limited number of usefully descriptive headings. Click on Task Views to see bundles of packages assembled by topic. Currently, there are 40 Task Views on CRAN as listed in Table 1.2.

Click on the Task View to get an annotated list of the packages available under any particular heading. If Base-*R* doesn't cover your needs, it is highly likely that a package exists that does.

1.6.3 Installing packages

The base package does not contain some of the libraries referred to in this book, but installing these is very simple.

It is best to install packages using the `install.packages ()` function, as shown below, rather than doing so via RStudio's Packages tab (therein, click on install, then search for the package needed). The packages used in this book are

```
install.packages ("akima")  
install.packages ("boot")
```