

Cambridge University Press
978-0-521-87265-2 - A First Course in Statistical Programming with R
W. John Braun and Duncan J. Murdoch
Frontmatter
[More information](#)

A First Course in Statistical Programming with R

This is the only introduction you'll need to start programming in R, the open-source language that is free to download, and lets you adapt the source code for your own requirements. Co-written by one of the R core development team, and by an established R author, this book comes with real R code that complies with the standards of the language.

Unlike other introductory books on the ground-breaking R system, this book emphasizes programming, including the principles that apply to most computing languages, and the techniques used to develop more complex projects. Learning the language is made easier by the frequent exercises within chapters which enable you to progress confidently through the book. More substantial exercises at the ends of chapters help to test your understanding.

Solutions, datasets, and any errata will be available from the book's website.

W. John Braun is an Associate Professor in the Department of Statistical and Actuarial Sciences at the University of Western Ontario. He is also a co-author, with John Maindonald, of *Data Analysis and Graphics Using R*.

Duncan J. Murdoch is an Associate Professor in the Department of Statistical and Actuarial Sciences at the University of Western Ontario. He was columnist and column editor of the statistical computing column of *Chance* during 1999–2000.

Cambridge University Press
978-0-521-87265-2 - A First Course in Statistical Programming with R
W. John Braun and Duncan J. Murdoch
Frontmatter
[More information](#)

Cambridge University Press
978-0-521-87265-2 - A First Course in Statistical Programming with R
W. John Braun and Duncan J. Murdoch
Frontmatter
[More information](#)

A First Course in Statistical Programming with R

W. John Braun and Duncan J. Murdoch



Cambridge University Press
978-0-521-87265-2 - A First Course in Statistical Programming with R
W. John Braun and Duncan J. Murdoch
Frontmatter
[More information](#)

CAMBRIDGE UNIVERSITY PRESS

University Printing House, Cambridge CB2 8BS, United Kingdom

Cambridge University Press is part of the University of Cambridge.

It furthers the University's mission by disseminating knowledge in the pursuit of education, learning and research at the highest international levels of excellence.

www.cambridge.org

Information on this title: www.cambridge.org/9780521872652

© W. John Braun and Duncan J. Murdoch 2007

This publication is in copyright. Subject to statutory exception and to the provisions of relevant collective licensing agreements, no reproduction of any part may take place without the written permission of Cambridge University Press.

First published 2007

9th printing 2014

Printed in the United Kingdom by Clays, St Ives plc.

A catalogue record for this publication is available from the British Library

ISBN 978-0-521-87265-2 Hardback

ISBN 978-0-521-69424-7 Paperback

Cambridge University Press has no responsibility for the persistence or accuracy of URLs for external or third-party internet websites referred to in this publication, and does not guarantee that any content on such websites is, or will remain, accurate or appropriate. Information regarding prices, travel timetables and other factual information given in this work are correct at the time of first printing but Cambridge University Press does not guarantee the accuracy of such information thereafter

Contents

	Preface	page ix
I	Getting started	1
1.1	What is statistical programming?	1
1.2	Outline of the book	2
1.3	The R package	3
1.4	Why use a command line?	3
1.5	Font conventions	4
1.6	Installation of R	4
2	Introduction to the R language	5
2.1	Starting and quitting R	5
2.1.1	Recording your work	6
2.2	Basic features of R	7
2.2.1	Calculating with R	7
2.2.2	Named storage	7
2.2.3	Functions	9
2.2.4	Exact or approximate?	9
2.2.5	R is case-sensitive	12
2.2.6	Listing the objects in the workspace	12
2.2.7	Vectors	12
2.2.8	Extracting elements from vectors	13
2.2.9	Vector arithmetic	14
2.2.10	Simple patterned vectors	15
2.2.11	Missing values and other special values	16
2.2.12	Character vectors	16
2.2.13	Factors	17
2.2.14	More on extracting elements from vectors	18
2.2.15	Matrices and arrays	18
2.2.16	Data frames	19
2.2.17	Dates and times	21
2.3	Built-in functions and online help	21
2.3.1	Built-in examples	22
2.3.2	Finding help when you don't know the function name	23
2.3.3	Built-in graphics functions	23
2.3.4	Additional elementary built-in functions	25
2.4	Logical vectors and relational operators	26
2.4.1	Boolean algebra	26
2.4.2	Logical operations in R	27
2.4.3	Relational operators	28
2.5	Data input and output	29
2.5.1	Changing directories	29

2.5.2	<code>dump()</code> and <code>source()</code>	29
2.5.3	Redirecting R output	30
2.5.4	Saving and retrieving image files	31
2.5.5	Data frames and the <code>read.table</code> function	31
2.5.6	Lists	31
	Chapter exercises	32
3	Programming statistical graphics	33
3.1	High-level plots	33
3.1.1	Bar charts and dot charts	34
3.1.2	Pie charts	35
3.1.3	Histograms	35
3.1.4	Box plots	36
3.1.5	Scatterplots	38
3.1.6	QQ plots	39
3.2	Choosing a high-level graphic	41
3.3	Low-level graphics functions	42
3.3.1	The plotting region and margins	42
3.3.2	Adding to plots	43
3.3.3	Setting graphical parameters	45
	Chapter exercises	46
4	Programming with R	47
4.1	Flow control	47
4.1.1	The <code>for()</code> loop	47
4.1.2	The <code>if()</code> statement	50
4.1.3	The <code>while()</code> loop	54
4.1.4	Newton's method for root finding	55
4.1.5	The <code>repeat</code> loop, and the <code>break</code> and <code>next</code> statements	57
4.2	Managing complexity through functions	59
4.2.1	What are functions?	59
4.2.2	Scope of variables	62
4.3	Miscellaneous programming tips	63
4.3.1	Using <code>fix()</code>	63
4.3.2	Documentation using <code>#</code>	64
4.4	Some general programming guidelines	65
4.4.1	Top-down design	67
4.5	Debugging and maintenance	72
4.5.1	Recognizing that a bug exists	72
4.5.2	Make the bug reproducible	73
4.5.3	Identify the cause of the bug	73
4.5.4	Fixing errors and testing	75
4.5.5	Look for similar errors elsewhere	75
4.5.6	The <code>browser()</code> and <code>debug()</code> functions	75
4.6	Efficient programming	77
4.6.1	Learn your tools	77
4.6.2	Use efficient algorithms	78
4.6.3	Measure the time your program takes	79

	4.6.4 Be willing to use different tools	80
	4.6.5 Optimize with care	80
	Chapter exercises	80
5	Simulation	82
	5.1 Monte Carlo simulation	82
	5.2 Generation of pseudorandom numbers	83
	5.3 Simulation of other random variables	88
	5.3.1 Bernoulli random variables	88
	5.3.2 Binomial random variables	89
	5.3.3 Poisson random variables	93
	5.3.4 Exponential random numbers	97
	5.3.5 Normal random variables	99
	5.4 Monte Carlo integration	101
	5.5 Advanced simulation methods	104
	5.5.1 Rejection sampling	104
	5.5.2 Importance sampling	107
	Chapter exercises	109
6	Computational linear algebra	112
	6.1 Vectors and matrices in R	113
	6.1.1 Constructing matrix objects	113
	6.1.2 Accessing matrix elements; row and column names	115
	6.1.3 Matrix properties	117
	6.1.4 Triangular matrices	118
	6.1.5 Matrix arithmetic	118
	6.2 Matrix multiplication and inversion	119
	6.2.1 Matrix inversion	120
	6.2.2 The <i>LU</i> decomposition	121
	6.2.3 Matrix inversion in R	122
	6.2.4 Solving linear systems	123
	6.3 Eigenvalues and eigenvectors	124
	6.4 Advanced topics	125
	6.4.1 The singular value decomposition of a matrix	125
	6.4.2 The Choleski decomposition of a positive definite matrix	126
	6.4.3 The QR decomposition of a matrix	127
	6.4.4 The condition number of a matrix	128
	6.4.5 Outer products	129
	6.4.6 Kronecker products	129
	6.4.7 <code>apply()</code>	129
	Chapter exercises	130
7	Numerical optimization	132
	7.1 The golden section search method	132
	7.2 Newton–Raphson	135
	7.3 The Nelder–Mead simplex method	138
	7.4 Built-in functions	142

7.5	Linear programming	142
7.5.1	Solving linear programming problems in R	145
7.5.2	Maximization and other kinds of constraints	145
7.5.3	Special situations	146
7.5.4	Unrestricted variables	149
7.5.5	Integer programming	150
7.5.6	Alternatives to <code>lp()</code>	151
7.5.7	Quadratic programming	151
	Chapter exercises	157
Appendix	Review of random variables and distributions	158
	Index	161

Preface

This text began as notes for a course in statistical computing for second year actuarial and statistical students at the University of Western Ontario. Both authors are interested in statistical computing, both as support for our other research and for its own sake. However, we have found that our students were not learning the right sort of programming basics before they took our classes. At every level from undergraduate through Ph.D., we found that students were not able to produce simple, reliable programs; that they didn't understand enough about numerical computation to understand how rounding error could influence their results; and that they didn't know how to begin a difficult computational project.

We looked into service courses from other departments, but we found that they emphasized languages and concepts that our students would not use again. Our students need to be comfortable with simple programming so that they can put together a simulation of a stochastic model; they also need to know enough about numerical analysis so that they can do numerical computations reliably. We were unable to find this mix in an existing course, so we designed our own.

We chose to base this text on R. R is an open-source computing package which has seen a huge growth in popularity in the last few years. Being open source, it is easily obtainable by students and economical to install in our computing lab. One of us (Murdoch) is a member of the R core development team, and the other (Braun) is a co-author of a book on data analysis using R. These facts made it easy for us to choose R, but we are both strong believers in the idea that there are certain universals of programming, and in this text we try to emphasize those: it is not a manual about programming in R, it is a course in statistical programming that uses R.

Students starting this course are not assumed to have any programming experience or advanced statistical knowledge. They should be familiar with university-level calculus, and should have had exposure to a course in introductory probability, though that could be taken concurrently: the probabilistic concepts start in Chapter 5. (We include a concise appendix reviewing the probabilistic material.) We include some advanced topics in

simulation, linear algebra, and optimization that an instructor may choose to skip in a one-semester course offering.

We have a lot of people to thank for their help in writing this book. The students in Statistical Sciences 259b have provided motivation and feedback, Lutong Zhou drafted several figures, and Diana Gillooly of Cambridge University Press, Professor Brian Ripley of Oxford University, and some anonymous reviewers all provided helpful suggestions. And of course, this book could not exist without R, and R would be far less valuable without the contributions of the worldwide R community.