Statistics and Computing

John M. Chambers

Software for Data Analysis

Springer

Statistics and Computing

Series Editors:
J. Chambers
D. Hand
W. Härdle

Statistics and Computing

Brusco/Stahl: Branch and Bound Applications in Combinatorial Data Analysis
Chambers: Software for Data Analysis: Programming with R
Dalgaard: Introductory Statistics with R
Gentle: Elements of Computational Statistics
Gentle: Numerical Linear Algebra for Applications in Statistics
Gentle: Random Number Generation and Monte Carlo Methods, 2nd ed.
Härdle/Klinke/Turlach: XploRe: An Interactive Statistical Computing Environment
Hörmann/Leydold/Derflinger: Automatic Nonuniform Random Variate Generation
Krause/Olson: The Basics of S-PLUS, 4th ed.
Lange: Numerical Analysis for Statisticians
Lemmon/Schafer: Developing Statistical Software in Fortran 95
Loader: Local Regression and Likelihood
Ó Ruanaidh/Fitzgerald: Numerical Bayesian Methods Applied to Signal Processing
Pannatier: VARIOWIN: Software for Spatial Data Analysis in 2D
Pinheiro/Bates: Mixed-Effects Models in S and S-PLUS
Unwin/Theus/Hofmann: Graphics of Large Datasets: Visualizing a Million
Venables/Ripley: Modern Applied Statistics with S, 4th ed.
Venables/Ripley: S Programming
Wilkinson: The Grammar of Graphics, 2nd ed.

John M. Chambers

Software for Data Analysis

Programming with R

John Chambers
Department of Statistics–Sequoia Hall
390 Serra Mall
Stanford University
Stanford, CA 94305-4065
USA
jmc@r-project.org

ISBN: 978-0-387-75935-7 e-ISBN: 978-0-387-75936-4
DOI: 10.1007/978-0-387-75936-4

Library of Congress Control Number: 2008922937

Printed on acid-free paper.

9 8 7 6 5 4 3 2 1

springer.com

Preface

This is a book about *Software for Data Analysis*: using computer software to extract information from some source of data by organizing, visualizing, modeling, or performing any other relevant computation on the data. We all seem to be swimming in oceans of data in the modern world, and tasks ranging from scientific research to managing a business require us to extract meaningful information from the data using computer software.

This book is aimed at those who need to select, modify, and create software to explore data. In a word, programming. Our programming will center on the R system. R is an open-source software project widely used for computing with data and giving users a huge base of techniques. Hence, *Programming with R*.

R provides a general language for interactive computations, supported by techniques for data organization, graphics, numerical computations, model-fitting, simulation, and many other tasks. The core system itself is greatly supplemented and enriched by a huge and rapidly growing collection of software packages built on R and, like R, largely implemented as open-source software. Furthermore, R is designed to encourage learning and developing, with easy starting mechanisms for programming and also techniques to help you move on to more serious applications. The complete picture—the R system, the language, the available packages, and the programming environment—constitutes an unmatched resource for computing with data.

At the same time, the "with" word in *Programming with R* is important. No software system is sufficient for exploring data, and we emphasize interfaces between systems to take advantage of their respective strengths.

Is it worth taking time to develop or extend your skills in such programming? Yes, because the investment can pay off both in the ability to ask questions and in the trust you can have in the answers. Exploring data with the right questions and providing trustworthy answers to them are the key to analyzing data, and the twin principles that will guide us.

What's in the book?

A sequence of chapters in the book takes the reader on successive steps from user to programmer to contributor, in the gradual progress that R encourages. Specifically: using R; simple programming; packages; classes and methods; inter-system interfaces (Chapters 2; 3; 4; 9 and 10; 11 and 12). The order reflects a natural progression, but the chapters are largely independent, with many cross references to encourage browsing.

Other chapters explore computational techniques needed at all stages: basic computations; graphics; computing with text (Chapters 6; 7; 8). Lastly, a chapter (13) discusses how R works and the appendix covers some topics in the history of the language.

Woven throughout are a number of reasonably serious examples, ranging from a few paragraphs to several pages, some of them continued elsewhere as they illustrate different techniques. See "Examples" in the index. I encourage you to explore these as leisurely as time permits, thinking about how the computations evolve, and how you would approach these or similar examples.

The book has a companion R package, `SoDA`, obtainable from the main CRAN repository, as described in Chapter 4. A number of the functions and classes developed in the book are included in the package. The package also contains code for most of the examples; see the documentation for `"Examples"` in the package.

Even at five hundred pages, the book can only cover a fraction of the relevant topics, and some of those receive a pretty condensed treatment. Spending time alternately on reading, thinking, and interactive computation will help clarify much of the discussion, I hope. Also, the final word is with the online documentation and especially with the software; a substantial benefit of open-source software is the ability to drill down and see what's really happening.

Who should read this book?

I've written this book with three overlapping groups of readers generally in mind.

First, "data analysts"; that is, anyone with an interest in exploring data, especially in serious scientific studies. This includes statisticians, certainly, but increasingly others in a wide range of disciplines where data-rich studies now require such exploration. Helping to enable exploration is our mission

here. I hope and expect that you will find that working with R and related software enhances your ability to learn from the data relevant to your interests.

If you have not used R or S-Plus® before, you should precede this book (or at least supplement it) with a more basic presentation. There are a number of books and an even larger number of Web sites. Try searching with a combination of "introduction" or "introductory" along with "R". Books by W. John Braun and Duncan J. Murdoch [2], Michael Crawley [11], Peter Dalgaard [12], and John Verzani [24], among others, are general introductions (both to R and to statistics). Other books and Web sites are beginning to appear that introduce R or S-Plus with a particular area of application in mind; again, some Web searching with suitable terms may find a presentation attuned to your interests.

A second group of intended readers are people involved in research or teaching related to statistical techniques and theory. R and other modern software systems have become essential in the research itself and in communicating its results to the community at large. Most graduate-level programs in statistics now provide some introduction to R. This book is intended to guide you on the followup, in which your software becomes more important to your research, and often a way to share results and techniques with the community. I encourage you to push forward and organize your software to be reusable and extendible, including the prospect of creating an R package to communicate your work to others. Many of the R packages now available derive from such efforts..

The third target group are those more directly interested in software and programming, particularly software for data analysis. The efforts of the R community have made it an excellent medium for "packaging" software and providing it to a large community of users. R is maintained on all the widely used operating systems for computing with data and is easy for users to install. Its package mechanism is similarly well maintained, both in the central CRAN repository and in other repositories. Chapter 4 covers both using packages and creating your own. R can also incorporate work done in other systems, through a wide range of inter-system interfaces (discussed in Chapters 11 and 12).

Many potential readers in the first and second groups will have some experience with R or other software for statistics, but will view their involvement as doing only what's absolutely necessary to "get the answers". This book will encourage moving on to think of the interaction with the software as an important and valuable part of your activity. You may feel inhibited by not having done much programming before. Don't be. Programming with

R can be approached gradually, moving from easy and informal to more ambitious projects. As you use R, one of its strengths is its flexibility. By making simple changes to the commands you are using, you can customize interactive graphics or analysis to suit your needs. This is the takeoff point for programming: As Chapters 3 and 4 show, you can move from this first personalizing of your computations through increasingly ambitious steps to create your own software. The end result may well be your own contribution to the world of R-based software.

How should you read this book?

Any way that you find helpful or enjoyable, of course. But an author often imagines a conversation with a reader, and it may be useful to share my version of that. In many of the discussions, I imagine a reader pausing to decide how to proceed, whether with a specific technical point or to choose a direction for a new stage in a growing involvement with software for data analysis. Various chapters chart such stages in a voyage that many R users have taken from initial, casual computing to a full role as a contributor to the community. Most topics will also be clearer if you can combine reading with hands-on interaction with R and other software, in particular using the `Examples` in the `SoDA` package.

This pausing for reflection and computing admittedly takes a little time. Often, you will just want a "recipe" for a specific task—what is often called the "cookbook" approach. By "cookbook" in software we usually imply that one looks a topic up in the index and finds a corresponding explicit recipe. That should work sometimes with this book, but we concentrate more on general techniques and extended examples, with the hope that these will equip readers to deal with a wider range of tasks. For the reader in a hurry, I try to insert pointers to online documentation and other resources.

As an enthusiastic cook, though, I would point out that the great cookbooks offer a range of approaches, similar to the distinction here. Some, such as the essential *Joy of Cooking* do indeed emphasize brief, explicit recipes. The best of these books are among the cook's most valuable resources. Other books, such as Jacques Pépin's masterful *La Technique*, teach you just that: techniques to be applied. Still others, such as the classic *Mastering the Art of French Cooking* by Julia Child and friends, are about learning and about underlying concepts as much as about specific techniques. It's the latter two approaches that most resemble the goals of the present book. The book presents a number of explicit recipes, but the deeper emphasis is in on concepts and techniques. And behind those in turn, there will be two general principles of good software for data analyis.

Acknowledgments

The ideas discussed in the book, as well as the software itself, are the results of projects involving many people and stretching back more than thirty years (see the appendix for a little history).

Such a scope of participants and time makes identifying all the individuals a hopeless task, so I will take refuge in identifying groups, for the most part. The most recent group, and the largest, consists of the "contributors to R", not easy to delimit but certainly comprising hundreds of people at the least. Centrally, my colleagues in R-core, responsible for the survival, dissemination, and evolution of R itself. These are supplemented by other volunteers providing additional essential support for package management and distribution, both generally and specifically for repositories such as CRAN, BioConductor, omegahat, RForge and others, as well as the maintainers of essential information resources—archives of mailing lists, search engines, and many tutorial documents. Then the authors of the thousands of packages and other software forming an unprecedented base of techniques; finally, the interested users who question and prod through the mailing lists and other communication channels, seeking improvements. This community as a whole is responsible for realizing something we could only hazily articulate thirty-plus years ago, and in a form and at a scale far beyond our imaginings.

More narrowly from the viewpoint of this book, discussions within R-core have been invaluable in teaching me about R, and about the many techniques and facilities described throughout the book. I am only too aware of the many remaining gaps in my knowledge, and of course am responsible for all inaccuracies in the descriptions herein.

Looking back to the earlier evolution of the S language and software, time has brought an increasing appreciation of the contribution of colleagues and management in Bell Labs research in that era, providing a nourishing environment for our efforts, perhaps indeed a unique environment. Rick Becker, Allan Wilks, Trevor Hastie, Daryl Pregibon, Diane Lambert, and W. S. Cleveland, along with many others, made essential contributions.

Since retiring from Bell Labs in 2005, I have had the opportunity to interact with a number of groups, including students and faculty at several universities. Teaching and discussions at Stanford over the last two academic years have been very helpful, as were previous interactions at UCLA and at Auckland University. My thanks to all involved, with special thanks to Trevor Hastie, Mark Hansen, Ross Ihaka and Chris Wild.

A number of the ideas and opinions in the book benefited from collab-