

# Preface

This book aims to provide a broad introduction to the R statistical computing environment (R Development Core Team, 2009a) in the context of applied regression analysis, which is typically studied by social scientists and others in a second course in applied statistics. We assume that the reader is learning or is otherwise familiar with the statistical methods that we describe; thus, this book is a *companion* to a text or course on modern applied regression, such as, but not necessarily, our own *Applied Regression Analysis and Generalized Linear Models*, second edition (Fox, 2008) and *Applied Linear Regression*, third edition (Weisberg, 2005). Of course, different texts and courses have somewhat different content, and if you encounter a topic that is unfamiliar or that is not of interest, feel free to skip it or to pass over it lightly. With a caveat concerning the continuity of examples within chapters, the book is designed to let you skip around and study only the sections you need.

The availability of cheap, powerful, and convenient computing has revolutionized the practice of statistical data analysis, as it has revolutionized other aspects of our society. Once upon a time, but well within living memory, data analysis was typically performed by statistical packages running on mainframe computers. The primary input medium was the punchcard, large data sets were stored on magnetic tapes, and printed output was produced by line printers; data were in rectangular case-by-variable format. The job of the software was to combine instructions for data analysis with a data set to produce a printed report. Computing jobs were submitted in batchmode, rather than interactively, and a substantial amount of time—hours, or even days—elapsed between the submission of a job and its completion.

Eventually, batch-oriented computers were superseded by interactive, time-shared, terminal-based computing systems and then successively by personal computers and workstations, networks of computers, and the Internet—and perhaps in a few years by cloud computing. But some statistical software still in use traces its heritage to the days of the card reader and line printer. Statistical packages, such as SAS and IBM SPSS,<sup>1</sup> have acquired a variety of accoutrements, including programming capabilities, but they are still principally oriented toward combining instructions with rectangular data sets to produce printed output.<sup>2</sup>

<sup>1</sup>SPSS was acquired by IBM in October 2009.

<sup>2</sup>With SAS, in particular, the situation is not so clear-cut, because there are several facilities for programming: The SAS `DATA` step is a simple programming language for manipulating data sets, the `IML` (interactive matrix language) procedure provides a programming language for matrix computations, and the macro facility allows the user to build applications that incorporate `DATA` steps and calls to SAS procedures. Nevertheless, programming in SAS is considerably less consistent and convenient than programming in a true statistical programming environment, and it remains fair to say that SAS principally is oriented toward processing rectangular data sets to produce printed output. Interestingly, both SAS and SPSS recently introduced facilities to link to R code.

The package model of statistical computing can work well in the application of standard methods of analysis, but we believe that it has several serious drawbacks, both for students and for practitioners of statistics. In particular, we think that the package approach of submitting code and getting output to decipher is not a desirable pedagogical model for learning new statistical ideas, and students can have difficulty separating the ideas of data analysis that are under study from the generally rigid implementation of those ideas in a statistical package. We prefer that students learn to request specific output, examine the result, and then modify the analysis or seek additional output. This feedback loop of intermediate examination forces students to think about what is being computed and why it is useful. Once statistical techniques are mastered, the data-analytic process of inserting the intelligence of the user in the middle of an analysis becomes second nature—in our view, a very desirable outcome.

The origins of R are in the S programming language, which was developed at Bell Labs by experts in statistical computing, including John Chambers, Richard Becker, and Allan Wilks (see, e.g., Becker et al., 1988, Preface). Like most good software, S has evolved considerably since its origins in the mid-1970s. Although Bell Labs originally distributed S directly, it is now available only as the commercial product S-PLUS. R is an independent, open-source, and free implementation of the S language, developed by an international team of statisticians, including John Chambers. As described in Ihaka and Gentleman (1996), what evolved into the R Project for Statistical Computing was originated by Ross Ihaka and Robert Gentleman at the University of Auckland, New Zealand. A key advantage of the R system is that it is free—simply download and install it, as we will describe shortly, and then use it.

R is a *statistical computing environment* and includes an *interpreter*, with which the user-programmer can interact in a conversational manner.<sup>3</sup> R is one of several statistical programming environments; others include Gauss, Stata, and Lisp-Stat (which are described, e.g., in Stine and Fox, 1996).

If you can master the art of typing commands, a good statistical programming environment allows you to have your cake and eat it too. Routine data analysis is convenient, but so are programming and the incorporation of new statistical methods. We believe that R balances these factors especially well:

- R is very capable out of the box, including a wide range of standard statistical applications. Contributed packages, which are easy to obtain and add to the basic R software, extend the range of routine data analysis both to new general techniques and to specialized methods, which are of interest only to users in particular areas of application.

---

<sup>3</sup>A *compiler* translates a program written in a programming language into an independently executable program in machine code. In contrast, an *interpreter* translates and executes a program under the control of the interpreter. Although it is in theory possible to write a compiler for a high-level, interactive language such as R, it is difficult to do so. Compiled programs usually execute more efficiently than interpreted programs. In advanced use, R has facilities for incorporating compiled programs written in Fortran and C.

- Once you get used to it, the R programming language is reasonably easy to use—as easy a programming language as we have encountered—and is finely tuned to the development of statistical applications.
- The S programming language and its descendant R are also carefully designed from the point of view of computer science as well as statistics. John Chambers, the principal architect of S, won the 1998 Software System Award of the Association for Computing Machinery (ACM) for the S System. Similarly, in 2010, Robert Gentleman and Ross Ihaka were awarded a prize for R by the Statistical Computing and Statistical Graphics sections of the American Statistical Association.
- The implementation of R is very solid under the hood—incorporating, for example, sound numerical algorithms for statistical computations—and it is regularly updated, currently at least twice per year.

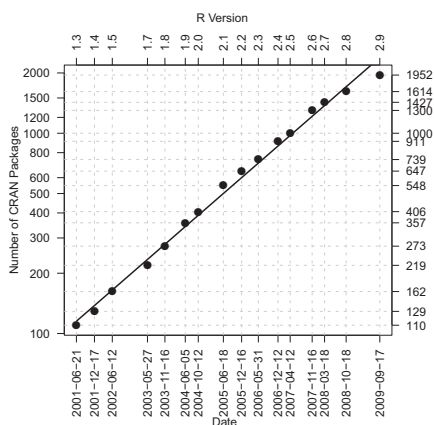
One of the great strengths of R is that it allows users and experts in particular areas of statistics to add new capabilities to the software. Not only is it possible to write new programs in R, but it is also convenient to combine related sets of programs, data, and documentation in *R packages*. The previous edition of this book, published in 2002, touted the then “more than 100 contributed packages available on the R website, many of them prepared by experts in various areas of applied statistics, such as resampling methods, mixed models, and survival analysis” (p. xii). The Comprehensive R Archive Network (abbreviated CRAN and variously pronounced *see-ran* or *kran*) now holds more than 2,500 packages (see Figure 1, drawn, of course, with R); other R package archives—most notably the archive of the Bioconductor project, which develops software for bioinformatics—add several hundred more packages to the total. In the statistical literature, new methods are often accompanied by implementations in R; indeed, R has become a kind of *lingua franca* of statistical computing—at least among statisticians—although interest in R is also robust in other areas, including the social and behavioral sciences.

## New in the Second Edition

---

Readers familiar with the first edition of this book will immediately notice two key changes. First, and most significant, there are now two authors, the first edition having been written by John Fox alone. Second, “S-PLUS” is missing from the title of the book (originally *An R and S-PLUS Companion to Applied Regression*), which now describes only R. In the decade since the first edition of the book was written, the open-source, free R has completely eclipsed its commercial cousin, S-PLUS. Moreover, where R and S-PLUS differ, we believe that the advantage generally goes to R. Although most of the contents of this second edition are applicable to S-PLUS as well as to R, we see little reason to discuss S-PLUS explicitly.

We have added a variety of new material—for example, with respect to transformations and effects plots—and in addition, virtually all the text has



**Figure 1** The number of packages on CRAN grew roughly exponentially since reliable data first became available in 2001 through 2009. *Source: Fox (2009).*

been rewritten. We have taken pains to make the book as self-contained as possible, providing the information that a new user needs to get started. Many topics, such as R graphics (in Chapter 7) and R programming (in Chapter 8), have been considerably expanded in the second edition.

The book has a companion R package called **car**, and we have substantially added to, extended, and revised the functions in the **car** package to make them more consistent, easier to use, and, we hope, more useful. The new **car** package includes several functions inherited from the **alr3** package designed to accompany Weisberg (2005). The **alr3** package still exists, but it now contains mostly data.

## Obtaining and Installing R

We assume that you're working on a single-user computer on which R has not yet been installed and for which you have administrator privileges to install software. To state the obvious, before you can start using R, you have to get it and install it. The good news is that R is free and runs under all commonly available computer operating systems—Windows, Mac OS X, and Linux and Unix—and that precompiled binary distributions of R are available for these systems. There is no bad news—at least not yet. It is our expectation that most readers of the book will use either the Windows or the Mac OS X implementations of R, and the presentation in the text reflects that assumption. Virtually everything in the text, however, applies equally to Linux and Unix systems, although the details of installing R vary across specific Linux distributions and Unix systems.

The best way to obtain R is by downloading it over the Internet from CRAN, at <http://cran.r-project.org/>. It is faster, and better netiquette, to

download R from one of the many CRAN mirror sites than from the main CRAN site: Click on the “Mirrors” link near the top left of the CRAN home page and select a mirror near you.

*Warning:* The following instructions are current as of version 2.11.0 of R. Some of the details may change, so check for updates on the website for this book, and also consult the instructions on the CRAN site.

## INSTALLING R ON A WINDOWS SYSTEM

Click on the “Windows” link in the “Download and Install R” section near the top of the CRAN home page. Then click on the “base” link on the “R for Windows” page. We recommend that you install the latest “patched build” of the current version of R; the patched release incorporates fixes to known bugs, usually small. Click on the link in “Patches to this release are incorporated in the r-patched snapshot build” and then on “Download R-x.y.z Patched build for Windows” to download the R Windows installer. “R-x.y.z” is the current version of R—for example, R-2.11.0.

R installs as a standard Windows application. We suggest that you take all the defaults in the installation, with one exception: We recommend that you select the *single-document interface (SDI)* in preference to the default *multiple-document interface (MDI)*. In the former, various R windows float freely on the desktop, while in the latter they are contained within a master window.

Once R is installed, you can start it as you would any Windows application, for example, by double-clicking on its desktop icon.

Whenever you start R, a number of files are automatically read and their contents executed. The start-up process provides the user with the ability to customize the program to meet particular needs or tastes, and as you gain experience with R, you may wish to customize the program in this way. On a single-user Windows system, probably the easiest route to customization is to edit the `Rprofile.site` file located in R’s `etc` subdirectory. The possibilities for customization are nearly endless, but here are two useful steps, both of which assume that you have an active Internet connection:

- Permanently select a CRAN mirror site, so that you don’t have to specify the mirror in each session that you install or update packages; just uncomment the following lines (with the exception of the first) in the `Rprofile.site` file by removing the pound signs (`#`):

```
# set a CRAN mirror
# local({r <- getOption("repos")
#       r["CRAN"] <- "http://my.local.cran"
#       options(repos=r)})
```

You must then replace the dummy site `http://my.local.cran` with a link to a real mirror site, such as `http://probability.ca/cran` for the CRAN mirror at the University of Toronto. This is, of course, just an example: You should pick a mirror site near you.

- Whenever you start R, automatically update any installed packages for which new versions are available on CRAN; just insert the following line into `Rprofile.site`:

```
utils::update.packages(ask=FALSE)
```

A disadvantage of the last change is that starting up R will take a bit longer. If you find the wait annoying, you can always remove this line from your `Rprofile.site` file.

Edit the `Rprofile.site` file with a plain-text (ASCII) editor, such as Windows Notepad; if you use a word-processing program, such as Word, make sure to save the file as plain text.

You can also customize certain aspects of the R graphical user interface via the *Edit* → *GUI preferences* menu.

## INSTALLING R ON A MAC OS X SYSTEM

Click on the “Mac OS X” link in the “Download and Install R” section near the top of the CRAN home page. Click on the “R-x.y.z.pkg (latest version)” link on the “R for Mac OS X” page to download the R Mac OS X installer. “R-x.y.z,” as mentioned earlier, is the current version of R—for example, R-2.11.0.

R installs as a standard Mac OS X application. Just double-click on the downloaded installer package, and follow the on-screen directions. Once R is installed, you can treat it as you would any Mac OS X application. For example, you can put the `R.app` program (or, on a 64-bit system, the `R64.app` program) on the Mac OS X Dock, from which it can conveniently be launched.

R is highly configurable under Mac OS X, but some of the details differ from the Windows details. The possibilities for customization are nearly endless. Here are the same two customizations that we suggested for Windows users:

- Permanently select a CRAN mirror site, so that you don’t have to specify the mirror in each session that you install or update a package. From the menus in the *R Console*, select *R* → *Preferences*, and then select the *Startup* tab. Pick the URL of a mirror site near you.
- Whenever you start R, automatically update any installed packages for which new versions are available. Using a text editor capable of saving plain-text (ASCII) files (we recommend the free Text Wrangler, which can also be configured as a programming editor for R), create a file named `.Rprofile` in your home directory, being careful not to omit the initial period (`.`), and insert the following line in the file:

```
utils::update.packages(ask=FALSE)
```

## INSTALLING AND USING THE CAR PACKAGE

Most of the examples in this book require the **car** package, which is not part of the standard R installation. The **car** package is available on CRAN. It “depends” on some other packages and “suggests” still others; the packages on which it depends will automatically be installed along with the **car** package.

Although both the Windows and the Mac OS X versions of R have menus for installing packages, the following command entered at the R command prompt will install the **car** package and all the other packages that it requires (i.e., both depends on and suggests):

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

Installing a package does not make it available for use in a particular R session. When R starts up, it automatically loads a set of standard packages that are part of the R distribution. To access the programs and data in another package, you must first load the package using the `library` command:<sup>4</sup>

```
> library(car)
```

This command also loads all the packages on which the **car** package depends. If you want to use still other packages, you need to enter a separate `library` command for each. The process of loading packages as you need them will come naturally as you grow more familiar with R. You can also arrange to load packages automatically at the beginning of every R session by adding a pair of commands such as the following to your R profile:

```
pkgs <- getOption("defaultPackages")
options(defaultPackages = c(pkgs, "car", "alr3"))
```

## The Website for the *R Companion*

There is a website for this book at <http://socserv.socsci.mcmaster.ca/jfox/Books/Companion/>.<sup>5</sup> If you are currently using R and are connected to the Internet, the `carWeb` command will open the website for the book in your browser:

```
> library(car)
> carWeb()
```

---

<sup>4</sup>The name of the `library` command is an endless source of confusion among new users of R. The command loads a *package*, such as **car**, which in turn resides in a *library* of packages. If you want to be among the R cognoscenti, never call a package a “library”!

<sup>5</sup>If you have difficulty accessing this website, please check the Sage Publications website at [www.sagepub.com](http://www.sagepub.com) for up-to-date information. Search for “John Fox,” and follow the links to the website for the book.

The website for the book includes the following materials:

- An appendix, referred to as the “online appendix” in the text, containing brief information on using R for various extensions of regression analysis not considered in the main body of the book: nonlinear regression; robust and resistant regression; nonparametric regression; time-series regression; Cox regression for survival data; multivariate linear models, including repeated-measures analysis of variance; mixed-effects models; structural-equation models; multiple imputation of missing data; and bootstrapping. We have relegated this material to a downloadable appendix in an effort to keep the text to a reasonable length. We plan to update the appendix from time to time as new developments warrant.
- Downloadable scripts for all the examples in the text.
- Exercises for the material on R in Chapters 1, 2, 7, and 8. As will be clear from the chapter synopses below, the remaining chapters deal with statistical material, for which a text on regression analysis should provide exercises.
- A few data files discussed in this *Companion* but not included in the **car** package.
- Errata and updated information about R.

All these can be accessed using the `carWeb` function; after loading the **car** package in R, type `help(carWeb)` for details.

## Using This Book

---

This book is intended primarily as a companion for use with another textbook that covers linear and generalized linear models. For details on the statistical methods, particularly in Chapters 3 to 6, you will need to consult the regression textbook that you are using. To help you with this task, we provide sections of complementary readings, including references to Fox (2008) and Weisberg (2005).

While the *R Companion* is not intended as a comprehensive users’ manual for R, we anticipate that most students learning regression methods and researchers already familiar with regression but interested in learning to use R will find this book sufficiently thorough for their needs.<sup>6</sup> Various features of R are introduced as they are needed, primarily in the context of detailed, worked-through examples. If you want to locate information about a particular feature, however, consult the index of functions and operators, or the

---

<sup>6</sup>A set of manuals in PDF and HTML format is distributed with R and can be accessed with Windows or Mac OS X through the *Help* menu. The manuals are also available on the R website. R has a substantial user community, which contributes to active and helpful email lists. See the previously mentioned website for details. And please remember to observe proper netiquette: Look for answers in the documentation and frequently-asked-questions (FAQ) lists before posting a question to an email discussion list; the people who answer your question are volunteering their time. Also, check the posting guide, at [www.r-project.org/posting-guide.html](http://www.r-project.org/posting-guide.html), before posting a question to one of the R email lists.

subject index, at the end of the book; there is also an index of the data sets used in the text.

Occasionally, more demanding material (e.g., requiring a knowledge of matrix algebra or calculus) is marked with an asterisk; this material may be skipped without loss of continuity, as may the footnotes.<sup>7</sup>

Most readers will want to try out the examples in the text. You should therefore install R and the **car** package associated with this book before you start to work through the book. As you duplicate the examples in the text, feel free to innovate, experimenting with R commands that do not appear in the examples. Examples are often reused within a chapter, and so later examples in a chapter can depend on earlier ones in the same chapter; packages used in a chapter are loaded only once. The examples in *different* chapters are independent of each other, however: Think of the R code in each chapter as pertaining to a separate R session.

Here are brief chapter synopses:

- Chapter 1 explains how to interact with the R interpreter, introduces basic concepts, and provides a variety of examples, including an extended illustration of the use of R in data analysis. The chapter includes a brief presentation of R functions for basic statistical methods and concludes with a description of the **Rcmdr** (R Commander) package, which provides a basic point-and-click interface to R.
- Chapter 2 shows you how to read data into R from several sources and how to work with data sets. There are also discussions of basic data structures, such as vectors, matrices, arrays, and lists; on handling character data; and on dealing with large data sets in R.
- Chapter 3 discusses the exploratory examination and transformation of data, with an emphasis on graphical displays.
- Chapter 4 describes the use of R functions for fitting, testing, manipulating, and displaying linear models, including simple- and multiple-regression models and linear models with categorical predictors (factors).
- Chapter 5 focuses on generalized linear models (GLMs) in R. Particular attention is paid to GLMs for categorical data and to Poisson and related GLMs for counts.
- Chapter 6 describes methods—often called “regression diagnostics”—for determining whether linear models and GLMs adequately

---

<sup>7</sup>Footnotes include references to supplementary material (e.g., cross-references to other parts of the text), elaboration of points in the text, and indications of portions of the text that represent (we hope) innocent distortion for the purpose of simplification. The object is to present more complete and correct information without interrupting the flow of the text and without making the main text overly difficult.

describe the data to which they are fit. Many of these methods are implemented in the **car** package associated with this book.

Chapter 7 contains material on plotting in R, describing a step-by-step approach to constructing complex R graphs and introducing trellis displays constructed with the **lattice** package.

Chapter 8 is a general introduction to programming in R, including discussions of function definition, operators and functions for handling matrices, control structures, debugging and improving R programs, object-oriented programming, writing statistical-modeling functions, and the scoping rules of the R programming language.

With the possible exception of starred material, Chapters 1 and 2 contain general information that should be of interest to all readers. Chapters 3 to 6 cover material that will be contained in most regression courses. The material in Chapters 7 and 8 has less to do with regression specifically and more to do with using R in real-world applications, where the facilities provided either in the basic packages or in the **car** package need to be modified to meet a particular goal. Readers with an interest in programming may prefer to read the last two chapters before Chapters 3 to 6.

We employ a few simple typographical conventions:

- Input and output are printed in slanted and upright monospaced (typewriter) fonts, respectively—for example,

```
> mean(1:10) # an input line
[1] 5.5
```

The `>` prompt at the beginning of the input and the `+` prompt (not illustrated in this example), which begins continuation lines, are provided by R, not typed by the user.

- R input and output are printed as they appear on the computer screen, although we sometimes edit output for brevity or clarity; elided material in computer output is indicated by three widely spaced periods (`. . .`).
- Data set names, variable names, the names of R functions and operators, and R expressions that appear in the body of the text are in a monospaced (typewriter) font: `Duncan`, `income`, `mean`, `+`, `lm

```
(prestige~income + education, data=Prestige)
````.
- The names of R packages are in boldface: **car**.
- Occasionally, generic specifications (to be replaced by particular information, such as a variable name) are given in typewriter italics: `mean(variable-name)`.
- Menus, menu items, and the names of windows are set in an italic sans-serif font: *File*, *Exit*, *R Console*.

- We use a sans-serif font for other names, such as names of operating systems, programming languages, software packages, and directories: Windows, R, SAS, c:\Program Files\R\R-2.11.0\etc.

Graphical output from R is shown in many figures scattered through the text; in normal use, graphs appear on the computer screen in graphics device windows that can be moved, resized, copied into other programs, saved, or printed (as described in Section 7.4).

There is, of course, much to R beyond the material in this book. The S language is documented in several books by John Chambers and his colleagues: *The New S Language: A Programming Environment for Data Analysis and Graphics* (Becker et al., 1988) and an edited volume, *Statistical Models in S* (Chambers and Hastie, 1992), describe what came to be known as S3, including the S3 object-oriented programming system, and facilities for specifying and fitting statistical models. Similarly, *Programming With Data* (Chambers, 1998) describes the S4 language and object system. The R dialect of S incorporates both S3 and S4, and so these books remain valuable sources.

Beyond these basic sources, there are now so many books that describe the application of R to various areas of statistics that it is impractical to compile a list here, a list that would inevitably be out-of-date by the time this book goes to press. We include complementary readings at the end of many chapters, however. There is nevertheless one book that is especially worthy of mention here: The fourth edition of *Modern Applied Statistics With S* (Venables and Ripley, 2002), though somewhat dated, demonstrates the use of R for a wide range of statistical applications. The book is associated with several R packages, including the **MASS** package, to which we make occasional reference. Venables and Ripley's text is generally more advanced and has a broader focus than our book. There are also some differences in emphasis: For example, the *R Companion* has more material on diagnostic methods.

## Acknowledgments

---

We are grateful to a number of individuals who provided valuable assistance in writing this book and its predecessor:

- Several people have made contributions to the **car** package that accompanies the book; they are acknowledged in the package itself—see `help(package=car)`.
- Michael Friendly and three unusually diligent (and at the time anonymous) reviewers, Jeff Gill, J. Scott Long, and Bill Jacoby (who also commented on a draft of the second edition), made many excellent suggestions for revising the first edition of the book, as did eight anonymous reviewers of the second edition.
- C. Deborah Laughton, the editor at Sage responsible for the first edition, and Vicki Knight, the Sage editor responsible for the second edition, were both helpful and supportive.

- A draft of this book was used in Sociology 740 in the winter semester of 2010. Several students, most notably Arthur McLuhan, pointed out typographical and other errors in the text.
- The book was written in  $\text{\LaTeX}$  using live R code compiled with the wonderful Sweave document preparation system. We are grateful to Fritz Leisch (Leisch, 2002) for Sweave.
- Finally, we wish to express our gratitude to the developers of R and to those who have contributed to R software for the wonderful resource that they have created with their collaborative and, in many instances, selfless efforts.