

Chapter 2

R Infrastructure

Chapter summary: In this chapter we discuss the practical realities in setting up an analytical environment based on R, including hardware, software, budgeting, and training needs. We will also walk through the basics of installing R, R's library of packages, updating R, and accessing the comprehensive user help.

Congratulations if you decided to install R! As choices go, this is the best one in open source statistical software that you could make for at least the next decade.

2.1 Choices in Setting up R for Business Analytics

Some options await you now before you set up your new analytical environment:

2.1.1 *Licensing Choices: Academic, Free, or Enterprise Version of R*

You can choose between two kinds of R installations. One is free and open source and is available at <http://r-project.org>; the other is commercial and offered by many vendors including Revolution Analytics. However, there are other commercial vendors too.

Commercial Vendors of R Language Products:

- Revolution Analytics: <http://www.revolutionanalytics.com/>
- XL Solutions: <http://www.experience-rplus.com/>
- Information Builder: <http://www.informationbuilders.com/products/webfocus/PredictiveModeling.html>
- Blue Reference (Inference for R): <http://inferenceforr.com/default.aspx>
- R for RExcel: <http://www.statconn.com/>

Enterprise R from Revolution Analytics has a complete R Development environment for Windows including the use of code snippets to make programming faster. Revolution is also expected to make a GUI available by 2012. Revolution Analytics claims several enhancements for its version of R including the use of optimized libraries for faster performance and the RevoScaleR package that uses the xdf format to handle large datasets.

2.1.2 Operating System Choices

Which operating system should the business analyst choose, Unix, Windows, or Mac OS? Often the choice is dictated by the information technology group in the business. However, we compare some of the advantages and disadvantages of each.

1. Microsoft Windows: This remains the most widely used operating system on the planet. If you are experienced in Windows-based computing and are active on analytical projects, it would not make sense for you to move to other operating systems unless there are significant cost savings and minimal business disruption as a result of the transition. In addition, compatibility issues are minimal for Microsoft Windows, and extensive help documentation is available. However, there may be some R packages that would not function well under Windows; in that case, a multiple operating system is your next option.
2. MacOS and iOS: The reasons for choosing MacOS remain its considerable appeal in esthetically designed software and performance in art or graphics related work, but MacOS is not a standard operating system for enterprise systems or statistical computing. However, open source R claims to be quite optimized and can be used for existing Mac users.
3. Linux: This is the operating system of choice for many R users due to the fact that it has the same open source credentials and so is a much better fit for all R packages. In addition, it is customizable for large-scale data analytics. The most popular versions of Linux are Ubuntu/Debian, Red Hat Enterprise Linux, OpenSUSE, CentOS, and Linux Mint.
 - (a) Ubuntu Linux is recommended for people making the transition to Linux for the first time. Ubuntu Linux had a marketing agreement with Revolution Analytics for an earlier version of Ubuntu, and many R packages can be installed in a straightforward way. Ubuntu/Debian packages are also available.
 - (b) Red Hat Enterprise Linux is officially supported by Revolution Analytics for its enterprise module.
4. Multiple operating systems

Virtualization versus dual boot: if you are using more than two operating systems on your PC. You can also choose between having VMware Player from VMware

(<http://www.vmware.com/products/player/>), if you want a virtual partition on your computer that is dedicated to R-based computing, and having a choice of operating system at startup. In addition, you can dual boot your computer with a USB installer from Ubuntu's Netbook remix (<http://www.ubuntu.com/desktop/get-ubuntu/windows-installer>).

A software program called wubi helps with the dual installation of Linux and Windows.

2.1.3 Operating System Subchoice: 32- or 64-bit

Given a choice between a 32-bit versus 64-bit version of an operating system like Linux Ubuntu, keep in mind that the 64-bit version would speed up processing by an approximate factor of 2. However, you need to check whether your current hardware can support 64-bit operating systems; if so, you may want to ask your information technology manager to upgrade at least some of the operating systems in your analytics work environment to 64-bit versions. Smaller hardware like netbooks do not support 64-bit Linux, whereas Windows Home Edition computers may have 32-bit version installed on it. There are cost differences due to both hardware and software. One more advantage for 64-bit computing is the support from Revolution Analytics for its version of R Enterprise.

2.1.4 Hardware Choices: Cost-Benefit Tradeoffs for Additional Hardware for R

At the time of writing of this book, the dominant computing paradigm is workstation computing followed by server-client computing. However, with the introduction of cloud computing, netbooks, and tablet PCs, hardware choices are much more flexible in 2011 than just a couple years ago.

Hardware costs represent a significant expense for an analytics environment and are also remarkably depreciated over a short period of time. Thus, it is advisable to examine your legacy hardware and your future analytical computing needs and decide accordingly regarding the various hardware options available for R.

Unlike other analytical software that can charge by the number of processors, or servers, which can be more expensive than workstations, or grid computing, which can be very costly as well if it is even available, R is well suited for all kinds of hardware environments with flexible costs.

Given the fact that R is memory intensive (it limits the size of data analyzed to the RAM size of the machine unless special formats or chunking is used), the speed at which R can process data depends on the size of the datasets used and the number of users analyzing a dataset concurrently. Thus the defining issue is not R but the

size of the data being analyzed and the frequency, repeatability, and level of detail of analysis required.

2.1.4.1 Choices Between Local, Cluster, and Cloud Computing

- **Local computing:** This denotes when the software is installed locally. For big data, the data to be analyzed are stored in the form of databases. The server version—Revolution Analytics has differential pricing for server–client versions (as is true for all analytical software pricing), but for the open source version it is free, as it is for server or workstation versions. The issue of number of servers versus workstations is best determined by the size of the data. R processes data in RAM, so it needs more RAM than other software of its class.

Cloud computing is defined as the delivery of data, processing, and systems via remote computers. It is similar to server–client computing, but the remote server (also called the cloud) has flexible computing in terms of number of processors, memory, and data storage. Cloud computing in the form of a public cloud enables people to do analytical tasks on massive datasets without investing in permanent hardware or software as most public clouds are priced on pay per usage. The biggest cloud computing provider is Amazon, and many other vendors provide services on top of it. Google also does data storage in the form of clouds (Google Storage) and uses machine learning in the form of an API (Google Prediction API).

1. *Amazon:* We will describe how to set up an R session on an Amazon EC2 machine.
2. *Google:* We will describe how to use Google Cloud Storage as well as Google Prediction API using packages.
3. *Cluster-grid computing/parallel processing:* To build a cluster, you need the RMpi and SNOW packages, plus other packages that help with parallel processing. This will be covered in general detail but detailed instructions for building a big cluster will not be provided as that is more suitable for a high-performance computing environment.

2.1.5 Interface Choices: Command Line Versus GUI. Which GUI Should You Choose as the Default Startup Option?

R can be used in various ways depending on the level of customization. The main GUIs suitable for business analyst audiences are as follows:

1. R Commander
2. Rattle
3. Deducer and JGR
4. GrapheR

5. RKWard
6. Red-R
7. Others including Sciviews-K

The interfaces to R will be covered in detail in Chap. 3, where a detailed description will also be given of how to access R from other mainstream analytical software applications like Oracle Data Miner, JMP, SAS/IML, KNIME, and Microsoft Excel. In addition to the standard desktop GUI, there are Web interfaces that use R and command line for default coding.

2.1.6 Software Component Choice

Which R packages should you install? There are almost 3,000 packages, some of them are complementary, others depend on each other, and almost all are free.

Throughout this book we will describe specialized packages that are best suited for creating the results of certain analytical tasks. In the R Programming language, multiple approaches, code, functions, and packages can be used to achieve the same result. The objective of this book is to focus on analysis rather than the language, and accordingly we will indicate the easiest approach to accomplishing the given business analysis task and mention other options and the advantages and disadvantages of using multiple options and approaches.

2.1.7 Additional Software Choices

What other applications do you need to achieve maximum accuracy, robustness, and speed of computing and how do you make use of existing legacy software and hardware to achieve the best complementary results with R?

Once we have covered the basics, we will describe, in Chap. 11, additional tips, tricks, and tweaks to help you optimize your R code. These include setting up benchmarks to measure and improve code efficiency and using syntax editors and integrated development environments.

2.2 Downloading and Installing R

To download and install the open source version of R, visit R's home page at <http://www.r-project.org/>.

You will be directed to the CRAN mirror closest to your location. CRAN, which stands for Comprehensive R Archive Network, is a set of online mirror servers that enable you to download R and its various packages. The global network thus ensures a fast, dedicated, reliable network for downloading and accessing software. In this manner, CRAN guarantees the highest likelihood of availability of R as it is very difficult to bring down servers of the entire CRAN, but an isolated server might get

overwhelmed due to traffic (especially at new product release times). It consists of 79 sites in 34 regions. R can be downloaded from <http://cran.r-project.org/mirrors.html>.

For Windows-R, installers exist in the form of downloadable binaries. Download the Windows.exe file and install the program. In addition, read the Frequently Asked Questions.

On Linux (Ubuntu): To install the complete R system, open a terminal window and use *sudo apt-get update sudo apt-get install r-base*.

Debian packages for R are a bit dated, but this is the easiest way to install. The other way is to modify your source file with a CRAN mirror before running apt-get. Documentation for this is on the Web site given previously.

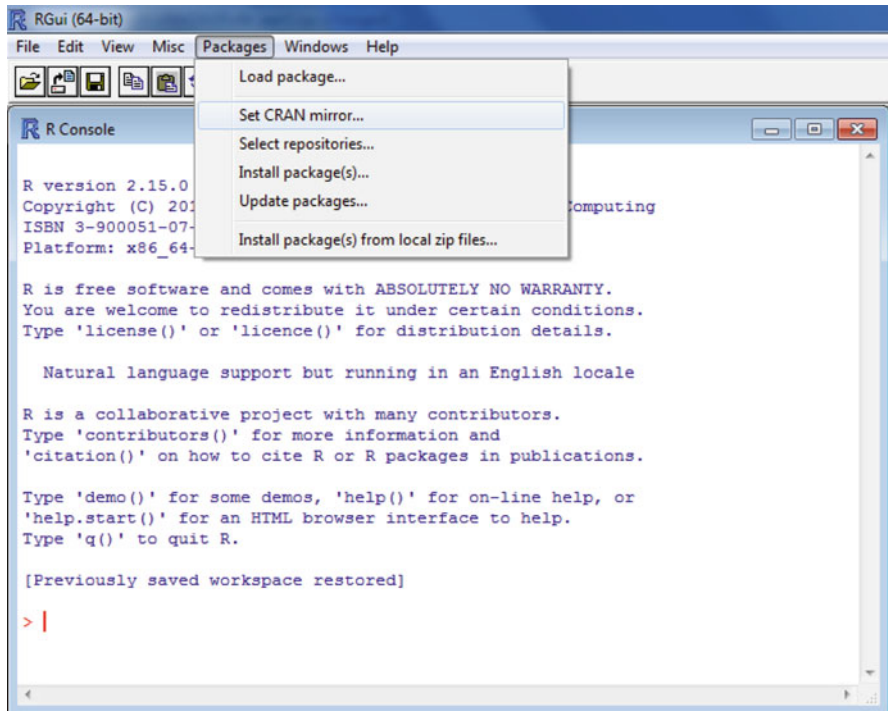
Mac OS has a separate, downloadable installer. You will need to refer to the main R Web site <http://www.r-project.org>.

The Australian CRAN Mirror can be accessed at <http://cran.ms.unimelb.edu.au/bin/windows/base/README.R-2.15.1> and FAQs at <http://cran.ms.unimelb.edu.au/bin/windows/base/rw-FAQ.html#Introduction>.

<CRAN MIRROR>/bin/windows/base/release.htm is the generic link for Windows releases. The latest version was 2.14.1 in January 2012, but this will change every 6 months.

2.3 Installing R Packages

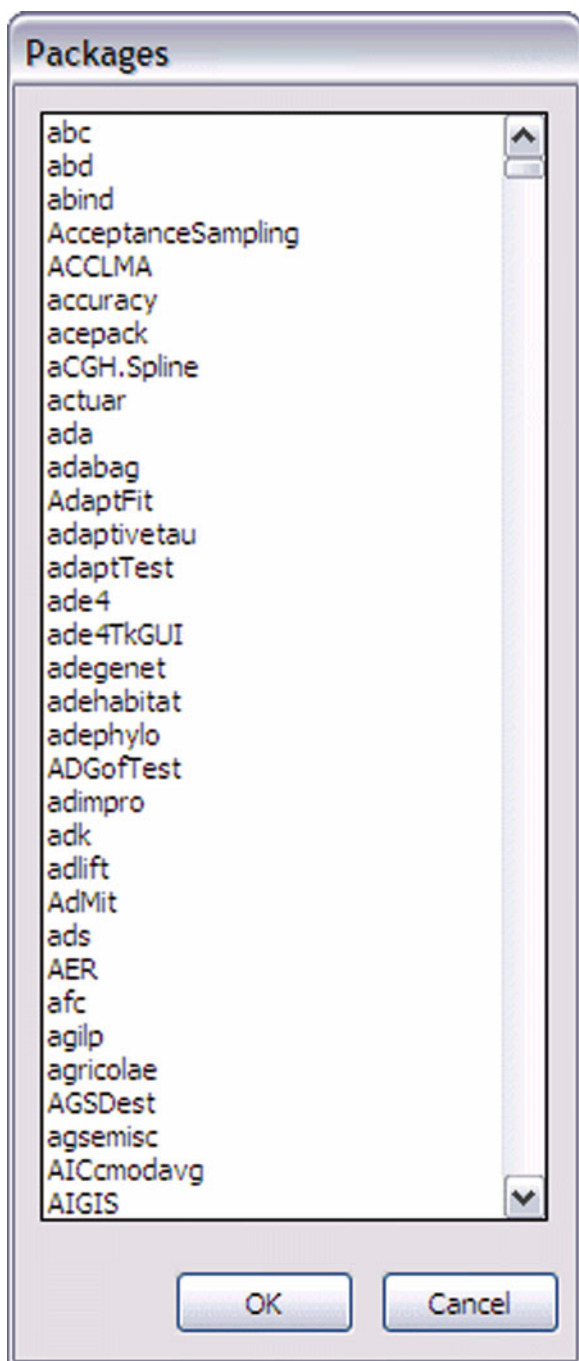
Unlike other traditional software applications that come in bundles, R comes in the form of one installer and a large number of small packages. There are an estimated 3,000 packages in R—so if you have a specific analytical need, chances are someone has created a package already for it. To launch R, simply click the icon that was created (for Windows users) or type R (at command terminal for Linux users).



Type `install.packages()` or select Install packages from the menu. You will then be asked to choose a CRAN mirror (or nearest location for download). Click on the nearest CRAN mirror.



Click on the package name and on OK to install that package as well as packages that are required for it to operate (dependencies).



Once a package is installed, type `library(package-name)` to check if it is working. In this example, we are trying to see if the GUI R Commander has installed.

```
>library(RCmdr)
```

- (a) Internet: To install a package from the Internet, you can use the following code as an example for modifying the installation to your needs:

```
install.packages("bigmemory", repos="http://R-Forge.R-project.org")
```

- (b) Local file: To set a local source code file (a tar.bz file in Linux or a .zip file in Windows) set `repos = null` in

```
install.packages(pkgs, lib, repos = getOption("repos"), contriburl = contrib.url(repos, type)).
```

Package Troubleshooting and Dependencies

- (a) Troubleshooting installations: For purposes of troubleshooting, read the package documentation as well as the official online R documentation at <http://cran.r-project.org/doc/manuals/R-admin.html#Installing-packages>.

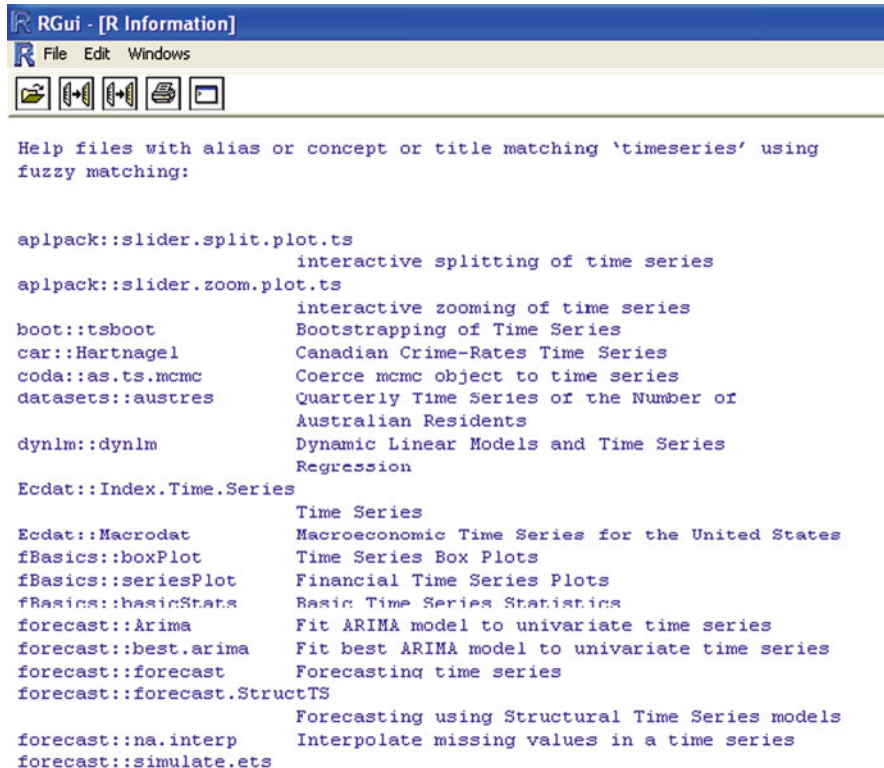
Entering `?` at the command prompt will give you help query results:

```
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> ?timeseries
No documentation for 'timeseries' in specified packages and libraries:
you could try '??timeseries'
> ??timeseries
> |
```

Entering a double question mark gives you more comprehensive help:



(b) Special notes:

1. GTK+ -GTK+ is a special requirement of some GUI packages like Rattle. To use GTK+ or the package RGTK, you first need to install the dependencies of GTK+ like, for example, Cairo, Pango, ATK, libglade. From the GTK Web site (<http://www.gtk.org/download-windows-64bit.html>) you will need the GLib, Cairo, Pango, ATK, gdk-pixbuf, and GTK+ developer packages to build software against GTK+. To run GTK+ programs, you will also need the gettext-runtime, fontconfig, freetype, expat, libpng, and zlib packages. It is ideal for the user to avoid troubleshooting problems by getting a Linux-based system for GTK-based packages; you can do this either by creating a dual-boot machine or using VMware Player to switch between operating systems.
2. ODBC: To connect to specific databases, you need to create an ODBC connection from the driver to the type of database. This will be covered in detail later.
3. In R we use the “ # ” symbol to comment out sentences. Commenting of code is done to make it more readable.

(c) Updating packages: You can use the following command to update all packages:
`>update.packages()`

2.4 Starting up Tutorial in R

- Get working directory
 - `getwd()`
- Set working directory
 - `setwd("C:/Users/KUs/Desktop")`
- List all objects
 - `ls()`
- List objects with certain words within name (i.e.fun)
 - `ls(pattern="fun")`
- Remove object “ajay”
 - `rm(ajay)`
- Remove all objects
 - `rm(list=ls())`
- Comment code: Use “#” in front of the part to be commented out.
 - `#comments`
- List what is in an object called ajay. Note: simply typing the name of the object is sufficient to print out the contents, so you need to be careful when dealing with huge amounts of data.
 - `> ajay`
 - `[1] 1 3 4 50`
 - List second value within an object
 - `> ajay[2]` (note bracket type)
 - `[1] 3 >`
 - List second to fourth values within an object (note colon “:”)
 - `> ajay[2:4]`
 - `[1] 3 4 50`
 - Look at the class of an object
 - `class(ajay)`
 - `[1] "numeric"` *#Classes can be numeric, logical, character, data frame, linear model (lm), time series (ts), etc.*
 - Use `as(object, value)` to coerce an object into a particular class
 - `ajay=as.data.frame(ajay)` *#turns the list into a data frame. This will change the dimensions and length of the object*
 - Find the dimensions of an object called “ajay”
 - `> dim(ajay)`
 - `[1] 4 1`

- Find the length of an object called “ajay”
- `> length(ajay)`
- `[1] 1`
- Suppose we change the object from data frame back into a list; we now have different dimensions
 - `> ajay=as.list(ajay)`
 - `> dim(ajay)`
 - `NULL`
 - `> length(ajay)`
 - `[1] 4`
- The scope of classes can be further investigated for building custom analytical solutions, but object-oriented programming is beyond the scope of this book as it is aimed at business analytics users. Interested readers may consult a brief tutorial in classes and methods at <http://www.biostat.jhsph.edu/~rpeng/biostat776/classes-methods.pdf> and the documentation at <http://cran.r-project.org/doc/manuals/R-lang.html#Objects>.

2.5 Types of Data in R

Unlike SAS and SPSS, which predominantly use the dataset/data frame structure for data, R has tremendous flexibility in reading data. Data can be read and stored as a list, matrix, or data.frame. This has great advantages for the power programmer experienced in object-oriented programming, but for the average business analyst the multiple ways of doing things in R can lead to some confusion and even more prolonged agony in the famous “learning curve” of R.

The various types of data in R are vectors, lists, arrays, matrixes, and data frames. For the purposes of this book, most data types will be data frames.

Data frames are rectangular formats of data with column names as variables and unique rows as records.

For other types of data please see <http://www.statmethods.net/input/datatypes.html> and <http://www.cyclismo.org/tutorial/R/types.html>.

2.6 Brief Interview with John Fox, Creator of Rcmdr GUI for R

What follows is a brief extract from a September 2009 interview with Prof. John Fox, creator of R Commander, one of R’s most commonly used GUIs.

Ajay: What prompted you to create R Commander? How would you describe R Commander as a tool, say, for a user of other languages and who want to learn R but are afraid of the syntax?

John: I originally programmed the R Commander so that I could use R to teach introductory statistics courses to sociology undergraduates. I previously taught this course with Minitab or SPSS, which were programs that I never used for my own work. I waited for someone to come up with a simple, portable, easily installed point-and-click interface to R, but nothing appeared on the horizon, and so I decided to give it a try myself.

I suppose that the R Commander can ease users into writing commands, inasmuch as the commands are displayed, but I suspect that most users don't look at them. I think that serious prospective users of R should be encouraged to use the command-line interface along with a script editor of some sort.

I wouldn't exaggerate the difficulty of learning R: I came to R—actually S then—after having programmed in perhaps a dozen other languages, most recently at that point Lisp, and found the S language particularly easy to pick up.

Ajay: I particularly like the R Cmdr plugins. Can anyone expand the R Commander's capabilities with a customized package plugin?

John: That's the basic idea, though the plugin author has to be able to program in R and must learn a little Tcl/Tk.

Ajay: What are the best ways to use the R Commander as a teaching tool (I noticed the help is a bit outdated).

John: Is the help outdated? My intention is that the R Commander should be largely self-explanatory. Most people know how to use point-and-click interfaces.

In the basic courses for which it is principally designed, my goals are to teach the essential ideas of statistical reasoning and some skills in data analysis. In this kind of course, statistical software should facilitate the basic goals of the course. As I said, for serious data analysis, I believe that it's a good idea to encourage use of the command-line interface.

Ajay: Do people on the R core team recognize the importance of GUIs? How does the rest of the R community feel? What kind of feedback have you gotten from users?

John: I feel that the R Commander GUI has been generally positively received, both by members of the R core team who have said something about it to me and by others in the R community. Of course, a nice feature of the R package system is that people can simply ignore packages in which they have no interest. I noticed recently that a paper I wrote several years ago for the *Journal of Statistical Software* on the Rcmdr package has been downloaded nearly 35,000 times. Because I wouldn't expect many students using the Rcmdr package in a course to read that paper, I expect that the package is being used fairly widely. [Update: As of February 2012 it has been downloaded 81,477 times.]

For more details on John's work see <http://socserv.mcmaster.ca/jfox/>

2.7 Summary of Commands Used in This Chapter

2.7.1 Packages

R Commander
Rattle
Deducer and JGR
GrapheR
RKWard
Red-R
Others including Sciviews-K
Snow
Rmpi
bigmemory

2.7.2 Functions

- `install.packages(FUN)`: Installs the package named FUN if available.
- `update.packages()`: Updates all packages on local system.
- `library(FUN)`: Loads the package FUN from local machine to R system.
- `?FUN`: Searches for help on keyword FUN.
- `??FUN`: Searches for comprehensive help on keyword FUN.
- `sudo apt-get install FUN`: Installs a software called FUN in Linux-based systems.

Citations and References

- Fox, J. The R Commander: A basic-statistics graphical user interface to R. J. Stat. Softw. **14**(9), 1–42 (2005). <http://www.jstatsoft.org/v14/i09/paper>
- R Blogger <http://www.r-bloggers.com/> aggregates blogs from almost 140 R blogs
- R email help lists: <http://www.r-project.org/mail.html>.
- Full interview with John Fox: <http://www.decisionstats.com/interview-professor-john-fox-creator-r-commander/>

R for Business Analytics

Ohri, A.

2013, XVIII, 312 p. 206 illus., 160 illus. in color.,

ISBN: 978-1-4614-4343-8