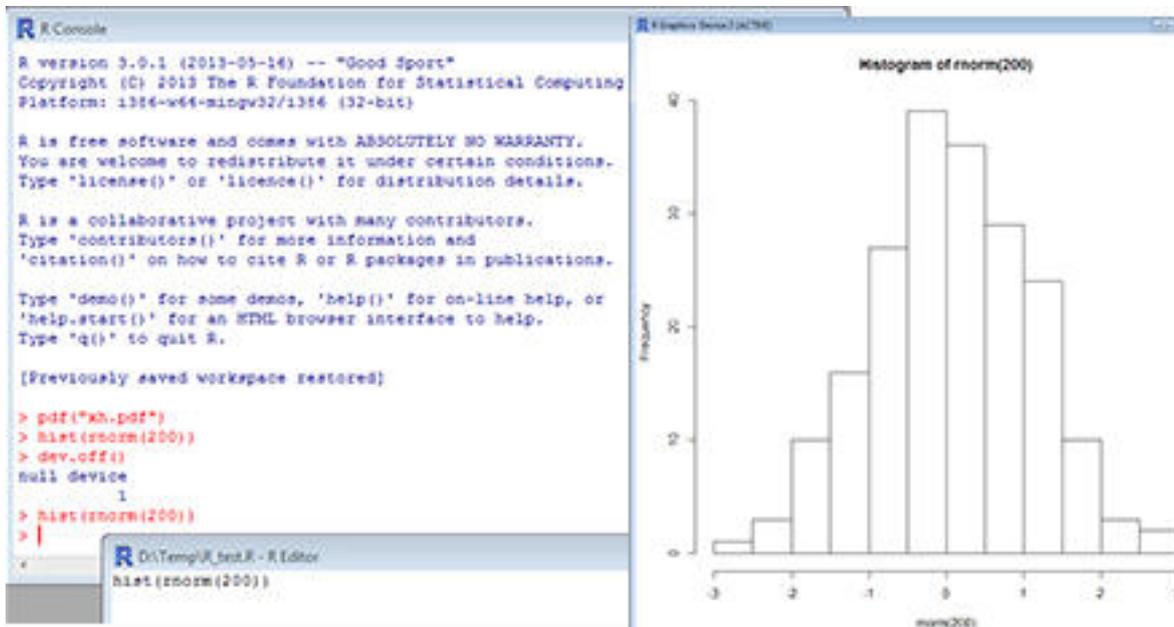


Programming in Arrr....

John R. Joyce, Ph.D.



This latest post, inspired by *International Talk Like a Pirate Day*,^{1,2} will take a brief look at the Arrr..., I mean **R**, programming language.

R is an open source functional scripting language designed for statistical analysis and data visualization. It is based on the S programming language developed by Rick Becker, John Chambers and Allan Wilks at AT&T.³ While it borrows many features and concepts from other languages, it also incorporates many deviations that can easily trip up anyone who makes assumptions regarding its behavior. Despite that, even statisticians admit that it is easy to learn.⁴

Multiple implementations of this language exist, and a number of them can be freely downloaded from the R home page⁵ over the Comprehensive R Archive Network (CRAN). After clicking on CRAN, simply select the precompiled binary appropriate for your operating system. CRAN also serves as a major repository of thousands of different projects (sometimes called libraries) that users have written for R.

Like many scripting languages, R can be used both interactively and in batch mode. Classically, R is a command oriented language, meaning that text commands are entered by typing them into a terminal window. However, for those who prefer to use a graphic user interface (GUI) to develop their code, a number of graphic oriented integrated development environments (IDE) are available. While other commercial and Open Source IDEs are available, several free ones include:

- RStudio (<http://www.rstudio.org> [1])

- StatET (<http://www.walware.de/goto/statet> [2])
- ESS (<http://ess.r-project.org> [3])

When working in a command line system, such as Linux, R is normally launched by typing the command 'R' in a terminal window, followed by the Enter key. In GUI systems, such as Microsoft Windows, it is usually launched by double clicking on the R icon, though the exact labeling of the icon might vary with the R distribution that you are using. If you are not using one of the GUI IDEs, once it is running, R will display a greeting containing version and other information followed by the R prompt, which is a greater than symbol (>). You can interactively type commands at this prompt for immediate execution.

In many cases, the hardest part of an R program is accessing your target data. Once that is done, the actual processing is frequently very straightforward. Depending on what you are attempting to do, a program in R might consist of just a few lines of code, such as that shown below.

```
pdf("output.pdf")  
  
hist(rnorm(200))  
  
dev.off()
```

This perfectly valid three-line program defines a PDF format file for the programs output, generates 200 random normal values and draws a histogram of these values in the output file, and finally closes the PDF output file.

While R as a programming language incorporates powerful features, its forte is statistical and graphical data analysis. While it can be used for many other things, its inherent limitations may well mean that it is not the optimum language to use for many of these projects. It is for that reason that R has been designed to be able to call routines written in other languages. Conversely, when programming in another language, such as C, FORTRAN or Python, you can call R routines to take advantage of the languages' data manipulation capabilities. Its associated libraries are what give R its real power. The binary distributions of R come with a number of these libraries, and others can be downloaded from sites such as Bioconductor⁶ and The Omega Project for Statistical Computing.⁷ However, the major repository for R libraries is R-Forge,⁸ which currently has over 1,600 projects registered.

I've found that a very useful book for a programmer interested in experimenting with R is Norman Matloff's ***The Art of R Programming: A Tour of Statistical Software Design***⁹ [No Starch Press, San Francisco, ISBN: 978-1-59327-384-2, ©2011, 373 pp, \$39.95]. It not only describes the features of the language and how it is used, but puts particular emphasis on R's differences from other languages and where these differences are likely to trip you up. Matloff frequently ends a chapter with a practical in-depth example of how the different features are used. Many of these examples are designed so that you can use them as tools in your data analysis. If you are attempting the more challenging task of learning both R and

Programming in Arrr....

Published on Scientific Computing (<http://www.scientificcomputing.com>)

statistics at the same time, it might be helpful to read through Benjamin Yakir's ***Introduction to Statistical Thinking (With R, Without Calculus)***.¹⁰ In-depth examination of R can be found in the ***R Journal***,¹¹ an open access, refereed journal exploring all aspects of the R language.

How useful knowing R would be to you obviously depends on the type of data with which you are dealing and what it is used for. However, even if it is not your primary development tool, if you have to deal with statistical analysis, build numeric models of systems, or visualize your experimental data, R is definitely a tool deserving of consideration.

References

1. Baur, J. & Summers, M. Welcome to the Official site for Talk Like A Pirate Day - September 19. *Int. Talk Pirate Day* (2013). <http://www.talklikeapirate.com/piratehome.html> [4]
2. International Talk Like a Pirate Day - Wikipedia. *Wikipedia Free Encycl.* (2013). http://en.wikipedia.org/wiki/International_Talk_Like_a_Pirate_Day [5]
3. AT&T Labs -- Research: Statistics Research, Our History. *ATT Labs - Res.* <http://stats.research.att.com/research/history.php> [6]
4. Wass, J. The R Language: Fun with statistics and programming. *Sci. Comput.* (2007). <http://www.scientificcomputing.com/articles/2007/08/r-language-fun-stati...> [7]
5. The R Project for Statistical Computing. *R Proj. Stat. Comput.* <http://www.r-project.org> [8]
6. Bioconductor - Home. *Bioconductor - Open Source Softw. Bioinforma.* <http://www.bioconductor.org> [9]
7. The Omega Project for Statistical Computing. *Omega Proj. Stat. Comput.* <http://www.omegahat.org> [10]
8. R-Forge: Welcome. *R-Forge* <https://r-forge.r-project.org> [11]
9. Matloff, N. *The Art of R Programming: A Tour of Statistical Software Design.* (No Starch Press, 2011). <http://nostarch.com/artofr.htm> [12]
10. Yakir, B. *Introduction to Statistical Thinking (With R, Without Calculus).* *Introd. Stat. Think. R Calc.* (2011). <http://pluto.huji.ac.il/~msby/StatThink/index.html> [13]
11. Welcome. *The R Journal.* *R J.* <http://journal.r-project.org> [14]

John Joyce is a laboratory informatics specialist based in Richmond, VA. He may be reached at editor@ScientificComputing.com [15].

Source URL (retrieved on 12/11/2013 - 9:24pm):

<http://www.scientificcomputing.com/blogs/2013/10/programming-arrr>

Links:

[1] <http://www.rstudio.org>

Programming in Arrr....

Published on Scientific Computing (<http://www.scientificcomputing.com>)

[2] <http://www.walware.de/goto/statet>

[3] <http://ess.r-project.org>

[4] <http://www.talklikeapirate.com/piratehome.html>

[5] http://en.wikipedia.org/wiki/International_Talk_Like_a_Pirate_Day

[6] <http://stats.research.att.com/research/history.php>

[7] <http://www.scientificcomputing.com/articles/2007/08/r-language-fun-statistics-and-programming>

[8] <http://www.r-project.org>

[9] <http://www.bioconductor.org>

[10] <http://www.omegahat.org>

[11] <https://r-forge.r-project.org>

[12] <http://nostarch.com/artofr.htm>

[13] <http://pluto.huji.ac.il/~msby/StatThink/index.html>

[14] <http://journal.r-project.org>

[15] <mailto:editor@ScientificComputing.com>