Do I need to learn R?

Four good reasons to try the open source platform for data analysis

Catherine Dalzell

September 03, 2013

R is a flexible programming language designed to facilitate exploratory data analysis, classical statistical tests, and high-level graphics. With its rich and ever-expanding library of packages, R is on the leading edge of development in statistics, data analytics, and data mining. R has proven itself a useful tool within the growing field of big data and has been integrated into several commercial packages, such as IBM SPSS® and InfoSphere®, as well as Mathematica. This article offers a statistician's perspective on the value of R.

You have heard about R. Perhaps you read an article like Sam Siewert's "Big data in the cloud." You know that R is a programming language and that it has something to do with statistics, but is it right for you?

Why choose R?

R does statistics. You could view it as a competitor of analytic systems like SAS Analytics, not to mention simpler packages like StatSoft STATISTICA or Minitab. Many professional statisticians and methodologists in government, business, and the pharmaceutical industry spend their careers on IBM SPSS or SAS without writing one line of R code. So in part, the decision to learn and to use R is a matter of corporate culture and how you like to work. I use several tools in my statistical consulting practice, but most of what I do is done in R. These examples show why:

- **R is a powerful scripting language.** I was recently asked to analyze the results of a scoping study. The researchers had gone through 1,600 research papers and coded their contents on several criteria — a large number of criteria, in fact, with multiple options and forks. Their data, once flattened onto a Microsoft® Excel® spreadsheet, contained more than 8,000 columns, most of them void. The researchers wanted to roll up totals under different categories and headings. R is a powerful scripting language with access to Perl-like regular expressions for handling text. Messy data require the resources of a programming language, and although SAS and SPSS have scripting languages for tasks that go beyond the drop-down menu, R was written as a programming language and so is a better tool for that purpose.
- **R leads the way.** Many new developments in statistics appear first as R packages before making their way into commercial platforms. I recently obtained data from a medical study...
on patient recall. For each patient, we had the number of treatment items the physician had suggested, along with the number of items the patient actually remembered. The natural model is the beta-binomial distribution. This has been known since the 1950s, but estimation procedures relating the model to covariates of interest is recent. Data like these are usually handled by general estimating equations (GEE), but GEE methods are asymptotic and assume that the sample is large. I wanted a generalized linear model with beta-binomial R. A recent R package estimates this model: `betabinom` by Ben Bolker. SPSS does not.

- **Integration with document publishing.** R integrates smoothly with the LaTeX document publishing system, meaning that statistical output and graphics from R can be embedded in publication-quality documents. This isn't for everyone, but if you want to write a book about your data analytics or simply don't like copying your results into a word-processing document, the shortest and most elegant route lies through R and LaTeX.

- **No cost.** As the owner of a small business, I like that R is free. Even for a larger enterprise, it is nice to know that you can bring in someone on a temporary basis and immediately sit them down to a workstation with leading-edge analytic software. No need to worry about the budget.

**What is R, and what is it for?**

**The 140-character explanation**

R is an open source implementation of S, a programming environment for data analysis and graphics.

As a programming language, R is similar to many others. Anyone who has ever written code will find much in R that is familiar. The distinctiveness of R lies in the statistical philosophy that it supports.

**A statistical revolution: S and exploratory data analysis**

Computers have always been good at computing things — after you have written and debugged a program to carry out the algorithm you want. But in the 1960s and 1970s, they were weak in the display of information, especially graphics. These technical limitations, together with trends within statistical theory, meant that the practice of statistics and the training of statisticians focused on model building and hypothesis testing. One assumed a world in which researchers opined hypotheses (often agricultural), built carefully designed experiments (at an agricultural station), fit the model, and ran the test. A spreadsheet-based, menu-driven program like SPSS reflects this approach. In fact, the first versions of SPSS and SAS Analytics consisted of subroutines that could be invoked from a (Fortran or other) program to fit and test one out of a toolbox of models.

Into this formalized and theory-laden framework, John Tukey dropped the concept of exploratory data analysis (EDA) like a boulder through a glass roof. Today, it is difficult to imagine a time when the analysis of a data set could begin without a box plot to check for skewness and outliers or when the residuals of a linear model were not checked for normality against a quantile plot. These ideas originated with Tukey, and now, no introductory statistics course is given without them. It was not always so.
EDA is more an approach than a theory. Essential to that approach are the following rules of thumb:

- Where possible, use graphics to discern features of interest.
- Analysis is incremental. Try one model; based on the results, fit another model.
- Check model assumptions using graphics. Remark outliers, where present.
- Use robust methods to protect against departures from distributional assumptions.

Tukey's approach launched a wave of development of new graphical methods and robust estimators. It also inspired the development of a new software framework better suited to exploratory methods.

The S language was developed at the Bell Laboratories by John Chambers and colleagues as a platform for statistical analysis, especially of the Tukey sort. The first version, for internal Bell use, was developed in 1976, but it wasn't until 1988 that it reached something like its current form. By this time, the language was also available to users outside of Bell. Every aspect of the language fits the "new model" of data analysis:

- S is an interpreted language operating within a programming environment. The syntax of S is a lot like the syntax of C, but with the difficult bits left out. S takes care of memory management and variable declarations, for example, so the user does not have to write or debug such things. The lower programming overhead enables a number of analyses to be done quickly on the same data set.
- From the start, S allowed for the creation of high-level graphics, and you can add features to any open graphics window. You can readily highlight points of interest, query their values, add smoothers to scatter plots, etc.
- Object orientation was added to S by 1992. In a programming language, objects structure data and functions to meet the intuition of the user. Human thought is always object-oriented, and statistical reasoning especially so. The statistician works with frequency tables, time series, matrices, spreadsheets of diverse data types, models, etc. In every case, the raw numbers are vested with attributes and expectations: A time series consists of observations and time points, for instance. And for each data type, standard statistics and plots are expected. For a time series, I might do a time series plot and a correlogram; for a fitted model, I might plot fits and residuals. S enables the creation of objects for all of these concepts and you can create more object classes as needed. Objects make it easy to go from the conceptualization of a problem to its implementation in code.

A language with attitude: S, S-Plus, and hypothesis testing
The original S language took Tukey's EDA seriously, to the extent that it was awkward to do anything in S but EDA. This was a language with attitude. For example, although S came with several useful internal functions, it was lacking in some of the most obvious features you would expect statistical software to possess. There was no function to perform a two-sample *t* test or
indeed hypothesis testing of any kind. But Tukey notwithstanding, a hypothesis test is sometimes the right thing to do.

In 1988, Seattle-based Statistical Science licensed S and ported an enhanced version of the language, called S-Plus, to DOS and later Windows®. Realistically aware of what its customers wanted, Statistical Science added the functionality of classical statistics to S-Plus. Functions for the analysis of variance (ANOVA), the $t$ test, and other models were added. True to S's object orientation, the outcome of any such fitted model is itself an S object. Appropriate function calls deliver the fits, the residuals, and the $p$-value of a hypothesis test. A model object can even contain the intermediate computational steps of an analysis, like a QR decomposition (where Q is orthogonal and R is upper right triangular) of the design matrix.

**There's an R package for that! An open source community**

At about the same time that S-Plus was launched, Ross Ihaka and Robert Gentleman of the University of Auckland in New Zealand decided to try their hands at writing an interpreter. They chose the S language as their model. The project took shape and gained support. They named it R.

R is an implementation of S with the additional models developed by S-Plus. In some cases, the same people were involved. R is an open source project under the GNU licence. On that basis, R continues to grow, largely through the addition of packages. An R package is a collection of data sets, R functions, documentation, and dynamic load items in C or Fortran that can be installed as a group and accessed from an R session. R packages add new functionality to R, and through these packages, researchers can easily share computational methods among their peers. Some packages are limited in scope, others represent whole areas of statistics, and some contain leading-edge developments. In fact, many developments in statistics appear first as R packages before making it into commercial software.

At the time of this writing, 4,701 R packages appear on CRAN, the R download site. Of these, six were added on that day alone. R has a package for everything, or so it seems.

**What happens when I use R?**

*Note:* This article is not a tutorial for R. The following example attempts no more than to give you a sense of what an R session looks like.

R binaries are available for Windows, Mac OS X, and several Linux® distributions. Source code is also available for those who like to compile their own.

In Windows®, the installer adds R to the Start menu. To launch R in Linux, open a terminal window and type `r` at the prompt. You should see something like Figure 1.
Type a command at the prompt, and R responds.

At this point, in a real-world setting, you would probably read data to an R object from an external data file. R can read data from a variety of formats, but for this example, I use the `michelson` data set from the MASS package. This is the package that accompanies Venables and Ripley's landmark text, *Modern Applied Statistics with S-Plus* (see Related topics). `michelson` contains results from the famous Michelson and Morley experiments to measure the speed of light.

The commands provided in Listing 1 load the MASS package, get the `michelson` data and take a peek at it. Figure 2 shows the commands with responses from R. Each line contains an R function, with its arguments in square brackets ([]).

**Listing 1. Start an R session**

```r
2+2             # R can be a calculator. R responds, correctly, with 4.
library("MASS") # Loads into memory the functions and data sets from
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
                 # package MASS, that accompanies Modern Applied Statistics in S
```
Figure 2. Session start and R's responses

Now let's have a look at the data (see Listing 2). The output is shown in Figure 3.

Listing 2. A box plot in R

```r
# Basic boxplot
with(michelson, boxplot(Speed ~ Expt))

# I can add colour and labels. I can also save the results to an object.

michelson.bp = with(michelson, boxplot(Speed ~ Expt, xlab="Experiment", las=1,
                                ylab="Speed of Light - 299,000 m/s",
                                main="Michelson-Morley Experiments",
                                col="slateblue1")

# The current estimate of the speed of light, on this scale, is 734.5
# Add a horizontal line to highlight this value.
abline(h=734.5, lwd=2,col="purple")  #Add modern speed of light
```

It seems that Michelson and Morley systematically overestimated the speed of light. There also seems to be a some heterogeneity across experiments.

Figure 3. Plotting a box plot
When I am happy with my analysis, I can save all the commands to one R function. See Listing 3.

**Listing 3. A simple function in R**

```r
MyExample = function(){
  library(MASS)
  data(michelson)
  michelson.bw = with(michelson, boxplot(Speed ~ Expt, xlab="Experiment", las=1,
           ylab="Speed of Light - 299,000 m/s", main="Michelson-Morley Experiments",
           col="slateblue1"))
  abline(h=734.5, lwd=2,col="purple")
}
```

This simple example illustrates several important features of R:

**Does R need major hardware?**

I worked this example on an Acer netbook running Crunchbang Linux. R does not require a heavy machine to carry out small or medium-sized analyses. For 20 years, it has been said of R that it is slow because it is interpreted and that the size of data it can analyze is limited by computer memory. This is true but usually irrelevant on modern machines, unless the application is seriously huge (big data).

- **Saving results** — The `boxplot()` function returns a number of useful statistics along with the graph, and you can save these to an R object through an assignment statement like `michelson.bp = ...` and extract them as needed. The outcome of any assignment statement is available throughout the R session and could be the subject of further analysis. The `boxplot` function returns a matrix of statistics used to draw the box plot (medians, quartiles, etc.), the number of items in each box plot, and the values of the outliers (shown on the graph in Figure 3 as open circles). See Figure 4.

**Figure 4. Statistics from the `boxplot` function**

![Figure 4](image)

- **The formula language** — R (and S) has a compact language for expressing statistical models. The code `Speed ~ Expt` in the argument tells the function to do box plots of Speed for each level of Expt (the experiment number). Had I wished to do an ANOVA to test whether Speed varied significantly across experiments, I would have used the same formula: `lm(Speed ~ Expt)`. The formula language can express a wide variety of statistical models, including crossed and nested effects and fixed and random factors.
• **User-defined R functions**— It's a programming language.

### R carries on into the 21st century

Tukey's exploratory approach to data analysis has become the classroom norm. It's what we teach, and it's what statisticians do. R supports this approach, which may explain why it is still popular. Object orientation also helps R remain current, as new sources of data require new data structures for their analysis. InfoSphere® Streams now supports R analytics for data that are different from those envisaged by John Chambers.

#### R-project Toolkit in InfoSphere Streams

InfoSphere Streams is an advanced computing platform that allows user-developed applications to ingest, analyze, and correlate information quickly as it arrives from thousands of real-time sources, handling very high data throughput rates: up to millions of events or messages per second. It includes an R-project Toolkit. [Learn more](#) and [give it a try](#).

### R and InfoSphere Streams

InfoSphere Streams is a computing platform and integrated development environment for the analysis of high-velocity data arriving from thousands of sources. The content of these data streams is typically unstructured or semi-structured. The goal of the analyses is to detect changing patterns in the data and direct decision-making based on quickly changing events. SPL, the programming language for InfoSphere Streams, organizes data through a paradigm that reflects the dynamic nature of the data and the need for rapid analysis and response.

We are a long way from a spreadsheet and the usual flat files of classic statistical analysis, but R can adapt. As of Version 3.1, SPL applications can pass data to R and thus draw on R's extensive library of packages. InfoSphere Streams supports R analytics by creating appropriate R objects to receive the information contained in SPL **tuples**, the basic data structure in SPL. InfoSphere Streams data can thus be passed to R for further analysis and the results passed back to SPL.

### What R does not do well

In fairness, there are some things that R does not do well or at all. Nor is R equally well suited to every user:

- **R is not a data vault.** The easiest way to enter data in R is to enter it somewhere else, then import it to R. Efforts have been made to add a spreadsheet front end to R, but they have not caught on. Not only does the absence of a spreadsheet feature affect data entry but it is also difficult to visually inspect data in R, as you can do in SPSS or Excel.
- **R makes ordinary tasks difficult.** In medical research, for example, the first thing you do with the data is calculate summary statistics for all of the variables while listing the occurrence of nonresponse and missing data. This is a three-click process in SPSS, but R has no built-in function to calculate this fairly obvious information and display it in tabular form. You could write something easily enough, but sometimes you just want to point and click.
- **The learning curve for R is nontrivial.** A novice can open a menu-driven statistical platform and obtain results in minutes. Not everyone wants to become a programmer to be an analyst, and perhaps not everyone needs to.
• **R is open source.** The R community is large, mature, and active, and R is surely among the more successful open source projects. As I have shown, the implementation of R is more than 20 years old, and the S language has been around longer than that. This is a proven concept and a proven product. But with any open source product, reliability depends on transparency. We believe in the code because we can check it ourselves and because other people can check it and report errors. This is not the same as a corporate project that takes it upon itself to benchmark and validate its software. And in the case of lesser-used R packages, you have no reason to suppose that they actually produce correct results.

**Conclusion**

Do I need to learn R? Perhaps not; *need* is a strong word. But is R a valuable tool for data analytics? Certainly. The language was expressly designed to reflect the way that statisticians think and work. R reinforces good habits and sound analysis. To me, it’s the right tool for the job.
Related topics

- *The New S Language: A Programming Environment for Data Analysis and Graphics* (R.A. Becker, John M. Chambers, A.R. Wilks; Chapman & Hall, 1988): This foundational work is known in R and S circles as "The Blue Book." It lists all of the built-in functions that come with S and provides a complete description of the language.
- Check out *Exploratory Data Analysis*, by John Tukey (not to be confused with Paul Tukey). This book provided the conceptual inspiration that is implemented in S.
- *Modern Applied Statistics with S-Plus*, (Springer-Verlag, 1997) by W.N. Venables and B.D. Ripley, is a classic introduction to object orientation in S-Plus (and R). The data sets and a number of functions used in this book are found in the R package MASS.
- With Joris Meys and Andrie de Vries"s *R for Dummies* (2012), R hits the big time.
- Joseph Adler's *R in a Nutshell* (O'Reilly, 2009) is a solid introduction to R, intended for people doing standard statistical analyses on moderate data sets. It does not cover big data.
- Springer has a series of books with orange covers and titles like *Time Series Analysis in R* and *An Introduction to Applied Multivariate Analysis with R*. These are a good introduction for the R user with a particular application area in mind. Unlike general introductions, the books of this series focus on relevant packages for their subject area, with less to say about base R.
- Many R "books" are really papers in applied statistics that use R. Probably the hardest thing about using R is understanding the statistical methods that it implements. Along these lines, "*Data Analysis and Graphics Using R — An Example-Based Approach,*" by John Maindonald and John Braun (Cambridge UP, 2010), is one of my favorites. It covers a host of useful statistical techniques and shows you how to use these methods in R. It has a supporting R package with data and functions, as well.
- *The Art of R Programming*, by Norman Matloff (O'Reilly, 2011), is not a statistics book but rather one of the few books to teach R precisely as a programming language. It's essential if you plan to write much code in R rather than simply running packages.
- If you could buy only one R book, *Data Mining with R*, by Luis Torgo, should not be that book. But assuming you plan to own more than one book, this is a nice, intermediate-level read. It consists of three cases studies in data mining, all different, and walks you through each step of the way, including data cleaning and dealing with missing values.
- "*An introduction to InfoSphere Streams*" is an excellent introductory article to the Streams language.
- "*Overview of the R-project toolkit*" provides a description of the Streams toolkit for integrating R code into SPL applications.
- See the scope of products in the InfoSphere Platform for information-intensive projects.
- Click through the recent videos on big data that appeal to novices and experts alike.
- Try out InfoSphere Streams: download it for 90 days or try it in the cloud.
- Check out many IBM SPSS products for free:
  - SPSS Modeler, a data mining workbench that helps you build predictive models quickly and intuitively, without programming
  - SPSS Text Analytics for Surveys, which uses powerful natural language processing (NLP) technologies specifically designed for survey text.
Do I need to learn R?

• **SPSS Visualization Designer**, which lets you easily create and share compelling visualizations that better communicate your analytic results
• Download R and get documentation from [CRAN](https://cran.r-project.org).
• Learn more about big data in the [developerWorks Streams developer center](https://www.ibm.com/developerworks/streams). Find technical documentation, how-to articles, education, downloads, product information, and more.
• **Find resources to help you get started with InfoSphere BigInsights**, IBM's Hadoop-based offering that extends the value of open source Hadoop with features like Big SQL, text analytics, and BigSheets.
• **Download InfoSphere BigInsights Quick Start Edition**, available as a native software installation or as a VMware image.
• **Download InfoSphere Streams**, available as a native software installation or as a VMware image.

© Copyright IBM Corporation 2013

[Trademarks](http://www.ibm.com/developerworks/ibm/trademarks/)