Better Data Analysis with R

Eric Koziol

Installing R and RStudio

Eric Koziol

What is R?

R is a very powerful open source statistical programming language that allows for robust and reproducible data analysis. R can handle files much larger than Excel (I have loaded over 1 GB of data into R for analysis) and allows your data analysis to be reproducible. That is, anyone can extract the same output from your analysis without having to have any errors caused by manual steps such as filtering out a given column. I think you will find the use of R extremely valuable in performing higher quality data analysis.

Installing R

To install R, visit this web address:

http://cran.rstudio.com/

and select the appropriate operating system that you are using. Then following the instructions. This will give you the shell of R which can run any myriad of R code.

Installing RStudio

While it nice to run code just with R, it is much better to develop code and perform data anlysis in an Integrated Development Environment (IDE). The best IDE for R is called RStudio and can be downloaded here:

http://www.rstudio.com/products/rstudio/#Desk

RStudio has four main windows:

1. The top left window is where you can view and edit code files. You can also run code file from here by clicking the run button in the upper right. Alternatively you can bring variables into your environment by pressing the source button.
2. The bottom left window is an R console where you can run R commands, install packages and test code.
3. The top right window displays the current variables in your environment. Clicking on a variable will preview you it in the first window. There is also a history tab in this window so you can see what commands have been entered in the R console (bottom left window).
4. The bottom right window contains a directory navigation system. Using this system you can change your working directory and open files into the top left editor window. Any plots that are not saved, will also be previewed in this window.

Installing packages for R

From what we have downloaded so far, we have a somewhat limited shell of R commands available to us. In order to greatly increase the usefulness and power of R, we need to install what are called packages. Packages are libraries of software code that contain groups of commands that we can use. Anyone can build and deploy R packages including you! The entire list of R packages that have been deployed to CRAN (the makers of RStudio) can be found here:

http://cran.rstudio.com/

However, we will install only a few that will be very helpful for performing better data analysis.

We can load in the necessary packages as such (you can copy and paste this code into the R console of RStudio). Note that in order to install a package you must quotes around the name.

```
install.packages(c("devtools", "lattice", "ggplot2", "reshape2",
                   "plyr", "slidify", "knitr", "swirl"))
```

The devtools packages contains a lot of dependencies that are used in many R packages. The lattice and ggplot2 packages are very powerful graphing packages that will be used in tutorials to follow. The reshape2 and plyr packages are very useful for transforming data. Slidify and knitr are packages used for presenting information. Slidify allows you to make a slide deck in R, which can allow you to update the data in a presentation seamlessly (since the R code is embedded). Knitr is similar to Slidify except instead of slides it creates markdown and html documents, which can be exported to PDFs. This document you are using was created with knitr. Swirl is an optional package that teaches you how to program in R interactively.

Even though we have installed all of the packages, they still needed to be loaded into the system in order to be used. Unlike the install.packages command, quotes are not needed when loading a library. Also, each package must be loaded separately. Let's load them now:

```
library(devtools)
library(lattice)
library(ggplot2)
library(reshape2)
library(plyr)
library(slidify)
library(knitr)

# Swirl is optional if you would like to learn more about R programming.
# Going to the course repository will tell you how to install swirl courses.
# These courses are interactive and performed within RStudio.
library(swirl)
```

A quick primer on R

For users familiar with other programming languages such as Matlab or Python, there are two main differences to R than to most other languages. Remembering these will prevent a lot of early headaches.

1. When assigning values to variables most languages use '=', such as x = 2. However in R, one must use '<-' as in x <- 2. This may take some getting used to. Alternatively you can also assign values in reverse such as 2 -> x.
2. In order to create arrays instead of using '[]' as in [1,2,3] one must use 'c()' as in c(1,2,3).

So if we wanted to store an array for future use we could type

```
x <- c(1,2,3)
```

If we wanted to add 1 to each element we would type

```
x + 1
```

```
## [1] 2 3 4
```

or if wanted to keep using x + 1 we would type

```
y <- x + 1
y
```

```
## [1] 2 3 4
```

or to overwrite x

```
x <- x + 1
x
```

```
## [1] 2 3 4
```

```
x+y
```

```
## [1] 4 6 8
```