Data Tools Handout: R and RStudio

Daina Bouquin, Harvard University & Zachary Painter, UMass Dartmouth

New England e-Science Data Tools Forum, 2015 November 20

Introduction & Additional Resources

This document is a modified version of an introduction to R given before by Zac designed to provide you background information before our presentation on 20 November, with supplemental details from Daina. It is principally influenced by four scholars; Dr. William Jacoby of Michigan State, Dr. Michael Hawthorne of UNC-Pembroke, Dr. Hadley Wickham of RStudio and Rice, and Dr. Roger Peng, Dr. Jeff Leek, and Dr. Brian Caffo of Johns Hopkins.

http://rseek.org/ - A helpful search engine for all things R.
http://statmethods.net/ - Quick R is a great online resource for understanding the basics of R.
http://www.cookbook-r.com/Graphs - One of R's great strengths is graphics. Here's a resource for that.
R for Everyone, by Jared P. Lander - Daina's print reference text.
A Survivor's Guide to R, by Kurt Taylor Gaubatz - Zac's print reference text.
http://swirlstats.com/students.html - Swirl is an R tutorial. Good to play with if you have never used R.

Getting Started: Installing Software

First you will need to install R. In order to use RStudio you will first need to download R. R is a language and environment for statistical computing and graphics.

https://cran.rstudio.com/index.html

Next, install RStudio. You don’t need RStudio to use R, but it is probably an easier introduction to the language as it has more menus and a cleaner environment to think about what you are doing.

https://www.rstudio.com/products/RStudio/

Getting Started: Installing Packages

Packages are probably the biggest reason why R is popular, as they extend R’s base functionality for other operations. In the bottom right panel you will see a tab that says “Packages”. From there, there is a button to Install additional packages; type in each package name and then execute the command.

Swirl – An Introduction to R, within R. Do the first 2 modules for an understanding of our talk.
dplyr, tidyr – Two of the most popular Wrangling packages, which will show package interplay.
ggplot2– Possibly the most popular non-default graphics package, we will use ggplot2 for Analysis.

Data Wrangling R (Cheat Sheet Here)

Data Wrangling is the process of converting data into something useable. Datasets are sometimes unrefined or difficult to make sense of, and wrangling is an important step in this process. In this talk we will demonstrate how to wrangle or munge data using dplyr and tidyr. Before we do that we will explain basic data structures and types in R so that we can actually start using R.

Our dataset needs to be restructured, because it would be difficult to analyze it. So we will use a gather function to get our data, and then use tidyr to rearrange the set to make it easier to work with.

Then we want to extract records where the ad revenue was more than 1. We can then use dplyr to extract (or subset) this data to view the smaller portions.

Data Analysis in R (Cheat Sheet Here)

Data Analysis is the process of figuring out what your data actually means. It is great to actually have a dataset, but if you don't know what it means then you can't use it to its full potential. In this talk we will demonstrate how to do visual analysis using R's base graphics capabilities, and then use ggplot2 for advanced graphs.

First we will create a Dot Chart using the base graphical capability of R. This will show the distribution of data points on a simple scale. It is easy to look at and use and we can do some basic analysis here, but we need more advanced tools to go further into analysis.

Our first plot is a Box and Whiskers Plot, which is useful for measuring centrality and showing the spread of data points. It is a nice graph and we can do some good analysis, but Box and Whisker Plots can't show distribution well, so we can change the code just slightly in ggplot2 to create a graph that will.

This next example is a Violin Plot, which is a modified Box and Whiskers plot. We didn't create a new dataset or modify the code a lot, but we now see a different way to analyze the data. Now we can see in more detail how our data is structured and perhaps gain new insight into what we are working with.

If we change the code just a little more to Show Points, we can now see each data point, which in turn allows us to get a glimpse of things like outliers which can escape the naked eye.