Quantitative Things

Fall 2018
Lead: Dave Braun

Welcome to *Quantitative Things*. We're glad you're here.

This is an informal workshop that is intended to give you some of the basics for using the R programming language for data analysis. R is an increasingly-popular, open source language that's largely used for statistical data analysis.

What are our goals?

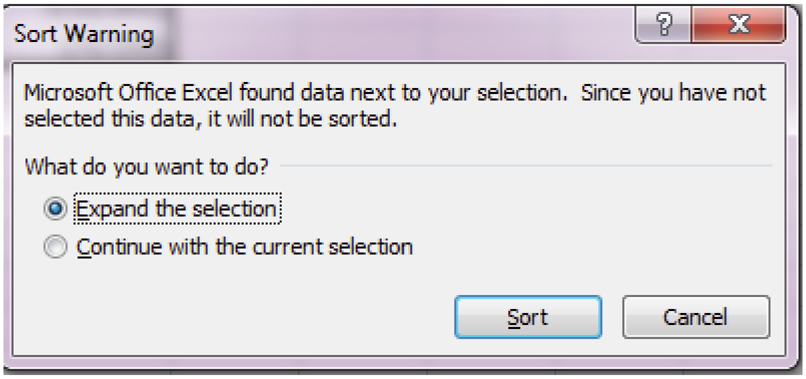
My goal for you leaving this workshop is to have a basic understanding of how to conduct analysis of psychological data in R. I want you to have familiarity with some fundamental concepts of data structures, how to manipulate data, and how to run and access the output of analyses. You won't know *everything* you need to know in order to fully analyze your data—that's okay. I still learn stuff every time I sit down in R. But the point of this workshop is to help you **get over that initial learning curve.** Once you have the basic building blocks, it will be much easier for you to learn and explore on your own.

Why should you care?

I could go on for awhile about the advantages of being proficient in R. One compelling reason involves the increasingly-prominent open science movement in psychology—**analytic transparency** is quickly becoming the standard in our field, and using scripts (either in R or SPSS) helps increase reproducibility. Conducting your analyses in R makes it significantly easier to share with someone *exactly* what operations you performed on some data. There's nowhere to hide! R is also **open source** (ie, **FREE**), so you don't need to rely on having an expensive SPSS license.

Also, I am a firm believer that programming is just one of the best (non-mandatory) skill sets that one can acquire in grad school. Programming is a hard skill that is highly valued both within and outside of academia. It also allows you so much more flexibility in achieving some data-driven goal. There are generally countless ways of manipulating data or performing some analysis. There is a large and helpful community behind R, and finding a way to achieve some goal in R is usually just a matter of Googling and patience. Once you generate code that you like, you can easily implement that code again (ie, you don't have to point and click every time to do the same thing). This really **streamlines** your workflow. R's analytical flexibility really allows for a type of creative expression that is absent in GUI-based software (eg, Excel, SPSS). You are truly the architect of how you solve your own goals. The machine *wants* to be your friend—all you need to do is learn how to speak the language.

The last thing I'll say is that programming greatly reduces the **error proneness** in your workflow. If you make an error, it'll generally be a systematic error that will affect your whole dataset, and you might be more likely to spot it as a result. It's also much easier to diagnose and correct errors when you have the programmatic operations directly in front of you. It's easier to debug code than to try and recall what incorrect button you might have pressed in Excel. Let's take a vivid textbook example. Say you want to sort your data by a particular variable; you want to organize rows in your dataset by increasing age of your sample, for example. In Excel, you select the "Age" variable, and click sort. Excel returns the following dialog box:



Sort Warning

Microsoft Office Excel found data next to your selection. Since you have not selected this data, it will not be sorted.

What do you want to do?

● Expand the selection

○ Continue with the current selection

You can either (a) do the operation that you intended to do, or (b) completely randomize your data. Why even offer the option of complete catastrophe? Making such an error will be *immensely* difficult to spot after the fact and will completely ruin an analysis, potentially costing you months or years of frustration and tears.

Communication / Resources

The majority of the communication for this workshop will occur on the **#quant** Slack channel. I cannot stress highly enough how important it is that you get in that channel. As I mention below, we often push around the dates of our meetings depending on people's availability in any given week (we all know this workshop isn't a high priority). If you're not in the Slack channel, you might miss out on these announcements, and this is especially crucial since we are so spread out this year and word of mouth isn't as effective as it once was.

As far as resources go, I'll just make a general reference to **stackexchange.com.** If you're serious about learning R (or any programming language for that matter), you'll be spending a lot of time on this site. The nice thing about just starting out is that you'll often encounter many mundane issues that other beginners have struggled with before. Part of the learning process is getting better at Googling your issues. As you go along, you'll get a better understanding of both the terminology and the conceptual framework of what it is that you're doing, which will allow you to Google more precise questions and get better answers. The senior grads are always here too if you need help.

Schedule (LOCATION AND SPECIFIC TIMES TBA)

The general way we've done this in the past is to try to meet every other week to cover a new topic. These sessions will be a bit more lecture style, where I'll distribute notes and guide everyone through them during the session. In the weeks between these sessions, we'll have what we call "wrangle sessions," where I encourage everyone to bring in their own data and work through some of the stuff we covered last week and we'll all be available to help each other and ask questions and we'll act like a big happy community.

The dates are flexible. If people are jammed up with work one week we tend to just push back as needed. Therefore I'm adding content through October and sort of anticipating that actually covering all this stuff will take longer than the dates listed.

Soft Outline of Topics

Topic	**Week**
Introduction to Rstudio interface. Simple operations. The logic of vectors.	9/3
Functions. For loops. Scripting in R. All general tools that will be helpful for weathering the storm known as Dr. Christopher Burke's PSYC 421 assignments. (generally intended for first years only)	9/10 Note: this breaks the every-other-week cycle. But this is really for first years only and I wanted to get this information across to first years early in the term.
Rmarkdown. Packages. Importing data. Restructuring data. Working with data frames. Dplyr.	9/24
Plotting. Everything about using ggplot2 to make pretty graphs.	10/8
Correlation, Regression.	10/22
ANOVAs.	10/29