Data Management in R

Stephen Paolillo

1 Introduction

This guide will walk through best practices for data management and cleaning in R. We will run through what it looks like to code in R for statistical data management and analysis, and how best to write reproducible and clean code.

2 Packages

R has a robust library of user-made packages that allow for easier data management, cleaning, analysis, and visualization. R packages number in the tens of thousands, so any list will almost certainly be incomplete, but below are some packages that I have found useful.

- tidyverse: essential library of packages, which includes the following:
  - ggplot2: my main tool for visualization
  - dplyr: my main tool for manipulating data frames
  - lubridate: some useful tools for working with dates
  - stringr: some useful tools for working with strings
  - tidyr: some more tools beyond dplyr for dataframes
- ggmap: for making maps in R
- haven: reading in Stata/SAS/SPSS files
- readxl: reading in XLSX files
- stargazer: quick and easy package for regression tables
- kableExtra: more extensive tables package
- modelsummary: complicated tables package, but with a lot of flexibility.

3 Folder System

For any project, in any language, folders should be set up for each part of the project. For a data analysis project, your file system should be something similar to the following:

- Data
- Code
- Literature
- Output

You can also include a working data or a raw data directory if needed.

4 R Environment

It is possible to run and save R code using base R, but most users will use RStudio, a free wrapper available online that is cleaner than base R and has a number of useful functions. For example, it is possible in RStudio to see, at the same time, what your code looks like, a command's help file, the data itself, and what you just ran in the console.

5 Saving Data

R can hold many different data frames in memory at once, so it is not as necessary to save after a particular step as it is in Stata or other languages. However, R is less powerful than some other languages, and may take longer to run a particularly large data step. This means that it may help to save the data after a particularly lengthy step so that you don't have to keep running that bit of code and waiting.

6 DOCUMENT YOUR CODE

Comment on your code throughout so that someone else who looks at your code can work with it more easily. This will often not even be another person, but you in the future looking at your code and saying "Oh what the hell problem was this supposed to fix?!". Commenting in R uses the # symbol and you should generally comment at least before each large block of code if not more often.

It is also important that you write code so that you could run the code from beginning to end and achieve the correct results. Code should not have sections that one is "not supposed to run" or lines that you have to enter yourself in the console. Additionally, hard-coding (i.e., having a number or string that will only work with how the data is currently set up) a particular line should be used very sparingly if at all.