

Quick R Tips

Heide Jackson
heidej@umd.edu

University of Maryland Population Research Center

August 2019

High Level Things to Know

- ▶ R is free and always will be.
- ▶ It's a flexible statistical software environment based on the S language.
- ▶ R is dynamic and updated more frequently than other statistical software packages like SAS or Stata.
- ▶ R is user driven and its functionality is greatly enhanced by user contributed packages.
- ▶ R Studio is a nice wrapper for accessing R

Where to Find R and R Studio

R can be downloaded at: <https://cran.r-project.org/>.

R Studio can be downloaded at:

<https://rstudio.com/products/rstudio/download/>.

The basics of doing anything in R

`object<-function(...)` #Frequent R users like this method

or equivalently

`object=function()`

- Functions may be built into R's basic system, loaded in from packages, or defined by you, the user.

Other Things To Know

- ▶ R is case sensitive. E.g. `x` and `X` will be stored in R as two different objects.
- ▶ The help function is your friend. If you are trying to figure out what a function does, type `help(functionname)` or equivalently `?functionname`.
- ▶ Objects can be embedded in other objects. If you call `data$x`, this can be different from calling `x`.
- ▶ `#` is used prior to a comment. Multi line comments aren't supported but there are system and program specific shortcuts for this if you want them.

Common Hazards and Missteps of Working in R

Redundancy

- ▶ Redundancy is major limitation and benefit to working in R.
 - ▶ Any task, no matter how simple or mundane can be accomplished at least five different ways using R or user supplied packages.
 - ▶ Users may be overwhelmed by choice.
 - ▶ The benefit to this is any task can be effectively dual coded. When in doubt about what a function is doing, look for another package that does the same thing.
 - ▶ When trying to decide how to code, it's sometimes worth doing a task in the easiest way and (if not too difficult) a method that uses just base R.

Common Hazards and Missteps of Working in R

Quality Control

- ▶ Great R programmers may not be great researchers and vice versa.
- ▶ How do you evaluate the quality of an R Package?
 - ▶ For common tasks, packages that have been more extensively used tend to be better debugged.
 - ▶ Right now the tidyverse collection of packages has become the R version of mainstream.
 - ▶ For complex methods, research the contributor. Where have they published? What documentation do they provide? What vignettes do they create? What data do they use?
 - ▶ When in doubt, check your work. Can estimation in R be reproduced in other statistical programs (such as Stata or SAS).

Common Hazards and Missteps of Working in R

Version Control

- ▶ R and its packages are frequently updated. R code may work one day and crash the next.
- ▶ Package control packages can help resolve this. A couple of options to consider are docker and packrat.
- ▶ As a global disclaimer, just because code is reproducible doesn't mean that it is correct.