An Introduction to R

36-290 – Introduction to Statistical Research Methodology

Week 1 – Fall 2021

What is R?

From the R FAQ:

2.1 What is R?

R is a system for statistical computation and graphics. It consists of a language plus a run-time environment with graphics, a debugger, access to certain system functions, and the ability to run programs stored in script files.

The design of R has been heavily influenced by two existing languages: Becker, Chambers & Wilks' S (see [What is S?](https://cran.r-project.org/doc/FAQ/R-FAQ.html#What-is-S_003f)) and Sussman's Scheme. Whereas the resulting language is very similar in appearance to S, the underlying implementation and semantics are derived from Scheme...

The core of R is an interpreted computer language which allows branching and looping as well as modular programming using functions. Most of the user-visible functions in R are written in R. It is possible for the user to interface to procedures written in the C, C++, or FORTRAN languages for efficiency. The R distribution contains functionality for a large number of statistical procedures. Among these are: linear and generalized linear models, nonlinear regression models, time series analysis, classical parametric and nonparametric tests, clustering and smoothing. There is also a large set of functions which provide a flexible graphical environment for creating various kinds of data presentations...

...

R has a [home page](https://www.R-project.org/). It is free software distributed under a GNU-style copyleft, and an official part of the GNU project ("GNU S").

The R Language is an Interpreted Language

A key phrase on the previous slide is "[t]he core of R is an interpreted computer language." This means that instructions are executed directly when typed into the console of `RStudio`. For instance:

```
> 666
[1] 666
```

In this particular case, `>R` doesn't know what to do with the input, so it outputs it to the screen (as the first element of an output vector...hence the `[1]`). If you input something of unknown type, like an uninitialized variable, you'd get an error instead:

```
> x
Error: object 'x' not found
```

But for now, let's defer learning more about `R` programming...

Installing Necessary Software

To download `R`, go to a "mirror site" of the Comprehensive R Archive Network, like this one at CMU. Download the version appropriate for your operating system and install it.

Once `R` is installed, download `RStudio` (the Desktop version, with a free open source license) by going to this web page. Again, download the version appropriate for your operating system and install it.

`R` comes bundled with a small subset of the 10,000+ packages that have been written for it. A particular package bundle that we would like to use, collectively dubbed the `tidyverse`, is not part of that subset. To install the `tidyverse`, you would begin by opening `RStudio` and noting how there are different panes. One includes tabs that say "Files," "Plots," and "Packages," etc. Click on "Packages," then click on "Install," then in the window that pops up, type in "tidyverse".

Once the `tidyverse` is successfully installed, we should be ready to go!

But Wait, I Still Don't Know R

We will begin our work with `R` by reading in a clean dataset and using the `tidyverse` to manipulate and visualize it. This basic `tidyverse` usage will be covered in another set of notes.

In the meantime, here is a resource for exploring `R` on your own (as optional homework):

- `RStudio` Primers