Lab 1: Introduction to R

Introduction

R is a free software environment for statistics.

In this lab, you will install R and practice loading a basic dataset.

Setup

Installing R

The latest version of R is available for download on Linux, OS X and Windows on the Comprehensive R Archive Network (CRAN): https://cran.r-project.org/.

Installing RStudio (optional)

RStudio is a (free) intergrated developement environment for R. You can think of R Studio as a companion program for basic R that helps you by keeping track of your things and organizing your various windows. Some people prefer to use R within in RStudio, but you do **not** need RStudio for this course.

Install R before installing RStudio (if install RStudio).

RStudio is available for download on www.rstudio.com. You do **not** need to pay money for RStudio. You can use RStudio Desktop under the Open Source License.

R Packages

Installing R Packages

In addition to hosting the actual R software, CRAN also hosts a variety of packages of R code. These packages are neatly bundled pieces of R code that add functionality to your basic R environment.

One package that is very handy for data science is Tidyverse (actually a collection of packages). Install Tidyverse by running the following command:

```
install.packages("tidyverse")
```

Loading R Packages

Once installed, packages need to be loaded into your environment before you can use them. You can load the core packages of Tidyverse by running the following command.

```
library("tidyverse")
```

Help! How do I use this?

To quickly open the documentation for package or function using the `help()` function. This can be shortened to just a question mark. In RStudio, this will open a help window on your chosen function.

```
# An example
help(tidyverse)
?tidyverse #this does the same thing
```

You can find more information on getting help with R on the R Project website.

Reading Data into R

R comes with a variety of functions built-in to help you load data. When we load Tidyverse, a package called `readr` is also loaded which contains additional functions to help you load data.

There are two basic ways to load comma-separated value files into R, now that we've loaded the Tidyverse package. We can use the `read.csv()` function from base R or the `read_csv()` function from the `readr` package which got loaded as part of the Tidyverse core. One notable difference between `read.csv()` and `read_csv()` is that `read.csv()` will return a standard R dataframe whereas `read_csv()` will return a tibble, which like a standard R dataframe but with certain changes to improve their behavior.

Look up the details on how to use one of these two functions using the help documentation. What do you need to feed into the function to make it work properly? You might also find the function `file.choose()` to be handy. You can place this function anywhere a path to a file is expected, and R will open up an interactive window to let you select a file.

Don't forget that you need to use the assignment operator `<-` to actually save the file you open to R's memory. The assignment operator saves whatever is on its right side to the name specified on its left side. For example `abc <- 2 + 2`, saves the value "4" to the name `abc`. You can now access the value inside `abc` in R at any time using its name.

```
?read.csv() # opens the help page for read.csv() function.
loaded_data <- read.csv(file.choose()) # selects a .csv file
loaded_data # a preview of what's inside.
```

In addition to `read.csv()` or `read_csv()`, both base R and the `readr` package contain functions to read many other formats. Explore these on your own.

If they don't contain code that will read in your preferred format, there may be another package in existence that does. For example, if you want to load in data directly from a Microsoft Excel spreadsheet, you could do this by using the readxl package.

Your Turn: Loading a Dataset

You can choose between a publicly available health dataset or you can use your own data if you have any! Here are some example publicly available datasets, that you can download if you want (use the Export > CSV button near the top right part of the page). If you load these, you'll need to use `read_csv()` from Tidyverse rather than the built-in `read.csv()` because `read.csv()` from **base R can only read files less than 5 MB in size.**

- Inpatient Prospective Payment System (IPPS) Provider Summary for the Top 100 Diagnosis-Related Groups (DRG) - FY2011, 26.8 MB filesize
- U.S. Chronic Disease Indicators (CDI), 159.2 MB filesize

Start a new R script file to save your work (showing each of the following steps) Load your preferred dataset into R. Print (show) the first 5 rows and 5 columns. If your dataset was loaded and assigned properly, you will be able to do this by using the square brackets "[]" to show a subset of the dataset. If your data was assigned to the name `my_data_set`, you would enter 'my_data_set[1:5,1:5] to display rows 1-5 and columns 1-5. Submit your work as an R script (somefilename.R) that demonstrates all of the requested actions (i.e. if we were to run your script, it should load data into R and then display the first 5 rows and 5 columns).

The report

For this first lab, save your code as an R script file and submit it through the course 2GW site. Clean up your code first if necessary, but ensure that it works as submitted.

Due date

Day 7, Week 2