## What Will Be Covered

We hope that students will be able to use the resources we provide (e.g. videos, practice problems, notes) to become comfortable using R for basic levels of data cleaning and analysis. The topics covered will include the layout of RStudio, the application that will be used throughout the demonstrations, how to upload data into RStudio, how to use numerous software packages within the “tidyverse” framework.

The specific skills we hope to teach through this resource include:

* Understanding different types of variable
* Understanding a dataset
* Removing and manipulating points in a dataset
* Understand logic tests and its uses
* Creating new variables in a dataset
* Writing code in a reproducible and coherent way
* Summarizing data into tables
* Performing power calculations
* Statistical analysis (Two sample T-Tests, Linear Regression)
* Graphing (scatter plot, box plot, bar graphs, etc.)

Layout of Lessons

**Introduction**

* We are CRASH
* Want this to be a way to learn the basics of using R for data cleaning, analysis, and presentation
* These are lessons about skills and not statistics itself. It is important to supplement this with the

**Lesson 0**: **Downloading Rstudios**

* R and Rstudios are free softwares that are compatible with several operating systems (OS).
* While the experience on Rstudios is nearly identical between systems, there are some minor differences in accessing files. This course will only be taught through MacOS, however, there are many informational resources on R throughout the internet.
* Steps to download:
  1. Go to <https://posit.co/download/rstudio-desktop/>
  2. Click “Download and Install R”
  3. Click “Download R” for your respective OS
  4. Click on the latest release of R (e.g. [R-4.3.0-arm64.pkg](https://cran.rstudio.com/bin/macosx/big-sur-arm64/base/R-4.3.0-arm64.pkg))
  5. Return to the link from step “a”
  6. Find what version of your respective OS you’re running on and click the appropriate Rstudio
  7. Add Rstudios into your application folder.

**Lesson 1a**: **What is Rstudios?**

* It is a software that helps you perform data organization and analysis in a precise and replicable way.
* Learning the language allows you to communicate what you want the software to do and its structure is similar between other languages.
* There are four panels in Rstudios with their own function:
  1. Script: Top left, this is where you will write and save code, almost like a microsoft word for code.
  2. Environment: Top right, this is where you can see the data you have loaded and the objects you created.
  3. Console: Bottom left, this is where any code you have in the script gets executed. You can write code directly into the console, however, it does not get saved once R is closed.
  4. File/Plots/Help/Viewer: This area allows you to get access to any file on your computer (File), visualize graphs and tables (Plot/Viewer), access informational help pages on specific functions (Help).
* R works within a working directory, which is essentially a folder with files that you want R to have access to. At the start of working on any project, you will need to create a project (a folder) that will set the working directory.
  1. Click file
  2. Click “New Directory”
  3. Click “New Project”
  4. Create a descriptive name for your project as it will become the name of your folder.
  5. Select where you want this folder to be located on your computer. It’s beneficial to make a folder for all your R projects so that you can easily find them.
* Clicking on the project file brings you to an Rstudio window with everything open to what you were working on.

**Lesson 1b**: **Basics of R: Objects and Functions?**

* Talk about how coding is giving the program something along with set of instructions to work on that thing and getting something out of it
  + The thing is called an input, which can be anything from a number to a data table.
  + The instruction is called a function, which can range from finding the arithmetic mean to making beautiful graphs
* You need to give more or less information depending on what you’re trying to do
* Explain what a storage object (i.e. object) is and how to assign it.
  + Give example with mean and vector

**Lesson 2a: Starting a Project**

* Once you create a project/working directory, you can begin writing code.
* In this course we will be doing work within a “R script” file, which amounts to a microsoft word file except for code.
* To create a new script file do as follows:
* Show them how to open a script file
* Talk about the “Run Button” and its shortcuts (command + enter)
* Now we’ll cover how to upload a file into the R workspace.
  1. Explain data frame as a R spreadsheet
  2. Download the tidyverse package (“library(tidyverse)”)
     + Explain packages and how they should always be on top of file
  3. Assign the data to an object as follows
     + Go through example highlighting the “read\_csv” function
     + Make sure to talk about how the file should be in the working directory
  4. Do again with an example of excel
     + Download readxl (“library(read\_xl)”)
     + Show how to choose sheet and range
* Talk a bit about objects in R, how they can be as simple as a single number or as complex as a spreadsheet
  1. Vectors
  2. Lists
  3. Data frames/tibbles

**Lesson 2b: RMarkdown**

* Explain how RMarkdown is a special type of file that allows you create a presentable html document that displays the output of your codes.
* Go through the big controls:
  + Header
  + Titles
  + Table of Content
  + How to insert code in Markdown that will be executed
* Explain knitting

**Lesson 3: Types of variables**

* Talk about how R assigns a type to each data point which determines how it will interact with it
* Talk about character vs numeric vs factor
* Talk about how to check how R views an object (e.g. is.numeric)
* Talk about how to change variable types (e.g. as.numeric)

**Lesson 4: Working within Dataset**

* Explain that dplyr is a package within tidyverse and how to use the following functions (don’t forget to emphasize the similar format amongst these functions):
  + Select
  + Filter (explain how use logic tests like case\_when)
  + Mutate (explain how use logic tests like case\_when)
  + Summarize (along with group\_by)
* Explain the pipe and how it can be used with the prior functions.

**Lesson 5: Working with Multiple Datasets**

* Explain how the “join()” function allows combining multiple datasets through the use of similar IDs.
* Talk about the join functions and their common format:
  + inner\_join()
  + left\_join()
  + right\_join()
  + full\_join()
* Talk about the differences in wide & long data and go over relevant functions:
  + pivot\_longer()
  + pivot\_wider()

**Lesson 6: Descriptive Analysis**

* Talk about how it’s useful to present the distribution and average values of a dataset
* Go over how to simply do this with summarize()
* Introduce gtsummary package and its relevant functions
  + tbl\_summary()
  + Save through gtsave

**Lesson 7: Regression**

* Talk about regression as a method of measuring association when accounting for several variables at once.
* Talk about linear and logistic regression and why you would use one over the other
* Show examples with the following functions
  + lm() for linear regression with glance(), tidy(), and tbl\_regression() functions
  + glm() for logistic regression with same functions and lmtest() for global likelihood ratio test (see if the model as a whole is significant)

**Lesson 8: Graphing**

* Explain how ggplot() is what will mostly be covered
* Explain the general format of it
* Explain the different geometries
* Explain mapping vs. setting
* Explain scales, themes, and labels
* Explain facets and pathwork