Getting Started in R and R Studio outline

* Reminder of last week’s topics and what we covered
  + Mention error – maybe put up a meme from IT Crowd
* Reminder of website and finding resources there (show website)
* Outline learning goals for today (show website)
* Difference between R and R Studio
  + R is a programming language and a program to run R language
  + R Studio is a program that lets you run R and other many cool things from a more user-friendly interface
* Navigating R Studio
  + Talk about layout
    - Source pane: scripts and other files (define scripts)
      * To get a Source pane if you don’t have one, show options for how to get there. Shift+Cmd+N (Mac), Shift+Ctrl+N Windows)
    - Environment/History
      * Highlight environment/history, mention others
    - Console
      * Directly interact with R here
      * Also a couple of other things in here that can come in useful depending on what you’re doing, not covered in this workshop
    - Files/Plots/Help/Viewer
* Getting Set Up in R Studio
  + Make a new project in a new directory
    - Show the options for making new projects (from the cube icon, from the drop down menu)
    - R Projects are really useful when working in R Studio because they set your **working directory** automatically to be in that folder every time
    - You may come across scripts that use a function called “setwd()”, but these are challenging because they might describe a place or file path on a specific computer so you have to change them. Thus, they are not easily transferrable from computer to computer (“reproducible”)
  + Change the save data option in Tools -> Global Options
* Organizing a project “directory” or “folder”
  + Talk about the value of having a consistent folder and subfolder structure for organizing projects
  + Show the picture from the website of the diagram
  + Create these subfolders – show how you can create them either in your computer file explorer or else in the R Studio pane on the bottom right
* Working in R and RStudio
  + Console vs. script
    - Talk about how you can interface with R from either the console or the source pane
      * Benefit of source pane is that you create “scripts”. They’re kind of like “recipes” that you can save as .R files and run and re-run and change.
      * You can also make comments for yourself in a script by using a line starting with a #. Comments are useful for repeating your scripts and remembering what you did before
    - Console:
      * Where R code is run/excecuted
      * The prompt > is where you type things you want R to do
      * Pressing enter will run what you entered
      * Your history is saved in the history pane, but you can’t go back and repeat things you put directly into the console
    - Script:
      * A record of R commands saved in a .R file extentions
      * Show how to make a new script by either file -> new file -> r script or by Shift+Cmd+N or Shift+Ctrl+N.
      * If you type lines here – you can send them to the R console to be evaluated, but they don’t get evaluated right away
        + Cmd+Enter or Ctrl+Enter (or press run button)
        + If you highlight multiple lines, you can run them all in the same way
        + You can re-run scripts that you save over and over again (and change them)
        + You can leave comments with #
    - Example
      * Type 1+1 in console pane and press enter
      * Type 1+1 in the script, then show multiple ways to run that line of code
      * Show writing a comment in this script
      * Show writing multiple lines of code, highlighting, and running them both
* Understanding and getting help with R and R Studio
  + What are functions?
    - Functions are the way that many of the things in R are coded to make your life easier
    - Some of these are built-in, most are created by users. Behind the hood, they take things that are really complicated or could be really complicated and let you run them in one line of code.
      * Example: above, we typed 1+1 to get 2. We could also use a function to get the same answer
      * Type in sum(1,1) – explain functions this way
  + How to get help with functions?
    - What if I don’t understand a function?
      * Type in ?sum()
      * You’ll see that it gives you information from whoever created this function about that function in the Help pane. It also gives you information on “arguments”, or the obejcts you need to give the function to run. In this case, it gives a … and then na.rm = T. If we look at what the … means, it just says we need to give sum() a set of numbers and it will add those numbers together.
    - Sometimes these can be more helpful than others, so I sometimes just google “How to add numbers in R”
    - If you get errors in functions, or need help
      * Google
      * Type in the error you got
  + End: Install packages that we need (
  + Install.packages(c(‘tidyverse’, ‘lubridate’, ‘ratdat’))
  + Library(tidyverse)
  + Library(lubridate)
  + Library(ratdat)