SC212: Lab 1

New `R` Commands:

- `read.csv()`: reads a .csv file and creates a data frame
- `names()`: shows names of columns of a dataset
- `library()`: loads package
- `data()`: loads a dataset from a package
- `mean()`: calculate the mean of a variable

Additional Information:

- Use `$` to access variables within a dataset. That is, `dat$var1` will output the variable named `var1` from the dataset named `dat`.
- Use `?` to access help files for functions. The command `?mean` will access the help file for the `mean()` function, for example.
- Use `<-` to assign values to a variable name. `x <- 3` assigns the value 3 to the variable `x`.

Getting Started with R and RStudio

`R` is a open-source and freely available programming language used for statistical computing. RStudio is a user interface which makes `R` more user-friendly. In this course, we will make sure of both. Colby has a server which runs `R` and RStudio for SC212. You can access it by opening a web browser and typing:

https://rstudio.colby.edu

Alternatively, you can download and use `R` and Rstudio on your own machine. First, navigate to https://www.r-project.org and download `R`. Next, go to https://www.rstudio.com and download RStudio.

When you first open RStudio, your screen should show 4 distinct panels — top-left, top-right, bottom-left, and bottom-right. If you only see 3, click the icon in the top left corner (the blank sheet of paper with a

plus sign) and select "R Script" from the menu. The top left window is the *script window*. This is where you'll mostly write and save code for homework and labs. Commands put here aren't executed until you run them. The bottom left panel is the *console*. The console is where R code is executed. Move your cursor to the console, type `12*11` and hit Enter.

The top right panel is the *environment/history* section of RStudio. The environments tells you what variables/values you have available to you in `R`. You might not have anything in there yet, but once you start saving datasets and variables, they will appear there. The history window just saves everything you've executed in the console, so it should say `12*11`, or whatever you just typed and executed.

The bottom right panel has spots for *files*, *plots*, *packages*, and *help*. The files pane just shows what files you have available on your machine. It's useful for finding where you've saved .csv files, for example. The plots pane previews any plots you produce. The packages pane allows you to see what packages are available on the machine. Packages contains functions and datasets that help you perform statistical work. Some are created by the the `R` developers and some by the statistics community at large. Lastly, the help panel displays information on `R` functions and datasets. Go to your console and type `?mean` to access the help file for the `mean()` function.

Using the R Console

The console is where you type commands to run them immediately. Here, `R` can be used like a calculator. Go to the console and on the line with the bottom-most `>`, type:

```
111*111
```

What did you get? We can use typical mathematical operations like: `+, -, *, /, sqrt(x), sin(x), x^y`, etc.

Using the Script Window

While you will certainly use the console frequently, ultimately your code needs to go in the script window (top-left). We do this to make our code reproducible. If you make an error in `R` and need to fix it, you have a record of everything you did and can make the necessary changes. If you had completed your homework strictly in the console, you would need to rewrite and redo everything all over again. Furthermore, writing `R` code in the script window makes it shareable with professors and with other students — anyone that you're collaborating with.

One of the most common things we'll be doing in the script window is using the assignment operator `<-`

which saves a value (number, vector, matrix, string, dataset, etc.) to variable that you appropriately name. Go back to R-studio and open a new script window by clicking File > New File > R Script. Once you've got a new script window open, type:

```
my.num <- 2
my.num^3
```

The first command saves the number 2 to the variable named `my.num`. The second command raises `my.num` to the power of 3. You can run these commands by clicking the Run button near the top right of the script window, or by using a keyboard shortcut (differs by operating system, usually `Ctrl + Enter` on Windows and `Command + Enter` on Mac). You can also highlight multiple lines and run them all at once.

Once you are done, make sure you save your scripts as needed!

Loading a Dataset

We will be loading datasets in a number of ways. For this class, we'll mostly be using either (1) .csv files or (2) pre-loaded datasets from an `R` package.

Loading from a .csv file

Follow these steps to load a .csv file into `R`.

1. Download the .csv file to your computer and save it to a logical location.
2. If you are *not* using Colby's RStudio server, skip to step (3). Otherwise, navigate to the Files pane in the bottom right and click upload. Upload the .csv file to an appropriate place in your Colby file explorer.
3. Lastly, load your data with the following code:

```
my.dat <- read.csv(file = file.choose())
```

This will open a new window and you simply select the .csv file. Once you click OK, the dataset will be available to you with the name `my.dat`. Note that the name `my.dat` is not a very informative name. In Example 1 below, we will practice betting naming habits by giving our dataset a more descriptive name.

Example 1

Download the `surveydata.csv` file from the Moodle page. The variables in this dataset are described in the table below.

Variable	Description
`classyr`	year and semester course was taken
`sex`	1=female
`tv`	number of hours of TV watched per week
`salary`	expected annual salary 10 years after graduating from Colby
`interest`	initial interest in course (1=no interest, 10=very interested)
`politics`	political views (1=very liberal, 10=very conservative)
`class`	number of years at Colby (1 = First Year, 4 = Senior)
`dining`	favorite dining hall (1=dana, 2=foss, 3=bobs)
`biology`	1 for bio majors, 0 for others
`ne`	1 for from New England, 0 for from someplace else
`number`	random number between 1 and 100
`mother`	mother's height in inches
`father`	father's height in inches
`you`	respondent's height in inches
`money`	how much money respondent was carrying
`hometown`	how far away respondent's hometown is from Colby

Open a new script window and within the script window, try loading these data using the method described above. Assign the dataset to the variable name `surveydata`.

Once you've loaded the data, view the first $n = 6$ observations with the command `head(surveydata)`. You can also view the whole dataset in spreadsheet format with the command `View(surveydata)` [Note the capital "V"].

To see the column (variable) names, use the command `names(surveydata)`. You can now reference a specific variable within the dataset using `surveydata$biology` or `surveydata$salary` [referencing whatever variable you want after the $].

Loading from a package

To load data from a pre-existing `R` package, first load the package using the `library()` command. Then load the data with the `data()` command. Try:

```
library(datasets)
```

```
data(faithful)
```

This loads a dataset on the Old Faithful Geyser which is available to you under the name `faithful`. If you've done this successfully, it should appear in your Environment pane (top-right of RStudio).

Example 2

Now let's try loading a dataset from the textbook. We will need to load the package `Lock5Data` and we will access the dataset called `BaseballHits`. Use the `library()` and `data()` commands to load these data into your `R` environment.

Once loaded, look at your dataset using `head()` or `View()` to make sure it looks OK.

Now, let's do some simple operations on this dataset. Let's find the average number of home runs hit during the season. The function we use to take the average is called `mean()`. Find the variable name for home runs using `names()` — the variable is called `HomeRuns`. Now find the average number of home runs using the command:

```
mean(BaseballHits$HomeRuns)
```

Work through RStudio Tutorial

Now that you know some of the basics, you can learn some tricks in RStudio that will make life easier for you (and also review some of the topics discussed above). Download the IntroToRStudio r-script from our Moodle page (under today's heading). After you download, upload it into your local `R` directory by clicking on the 'Files' tab in the lower right window and navigating to the file location (probably in the downloads folder). The file is actually an R-script, so you can open it as an r-script from the File menu: File > Open file. Select the IntroToRStudio file and it should appear in the script window and the text should appear green. If that's not what you're seeing, ask for help.

Once you've loaded the script into R-Studio, take your time to read it, follow the instructions, and work through it.

Help reminder You should expect to struggle a bit anytime you learn a new type of software. But you shouldn't be in a constant state of frustration! Google is a superb tool when trying to figure out how to do something in `R`. Use it! You should also refer back to the lab handouts, tutorials, and the R Users Guide that comes with your book (posted on Moodle). It's got a chapter-by-chapter breakdown of the commands that you'll need. Normally, I don't post all the book stuff, but this guide is actually really good. Use it! And, of course, you can always stop in to see me during office hours or send me an email!