Introduction to RMarkdown

Math 141

Week 1

R Markdown

- Why are we using R Markdown (instead of Word, a Tex editor, etc)?
  - It allows for a fully reproducible workflow. This makes it easier to share your work with others and to keep track of changes.
  - It is fairly easy to learn.
  - You can insert LaTeX code: μ.
  - You put everything (code, output, write-up) all in one document and then when you compile (i.e. knit) the document, you get a pretty output file.

R code and output

- To include R code and output, we need to create an **R chunk**. Here's an example R chunk:

```
# Compute the mean
mean(1:10)
```

```
## [1] 5.5
```

```
# Store the number 5 in an object called "x"
x <- 5
x
```

```
## [1] 5
```

- Notice that the three back ticks with `r` in curly brackets start the R chunk and three more back ticks close the R chunk.

- Compile the document and notice what changes about the chunk.

- For most assignments, I will give you partially completed R chunks. To ensure the document still knits, I will then tell R not to evaluate certain R chunks using the following option:

```
x <-
x
```

- **As you work through assignments, you will fill-in the missing code and then will change `eval=FALSE` to `eval=TRUE` once a chunk is complete.**

- In the upper right-hand corner of the R chunks, there are three items. The first allows you to set more options. The second runs all the R chunks above the one you are working in. The third runs the current chunk. Other options for running code:

  - put the cursor on the same line as the code and hit Command+Enter (for a Mac) and Control+Enter (for a PC).
  - put the cursor on the same line as the code and hit the Run button.

- You can add your own chunk by clicking on the green box with a "C". Try that now.

Knitting

- Find the “Knit” button and click it. You may get a message to install some Markdown packages. Allow it. You may also get a message about allowing popups. Allow them.
  - Examine the output file.
  - Notice that a **new** file appeared in your Files tab. This is the output document.
- Click on the arrow next to **Knit** and notice that you can compile an Rmd file into a pdf, html, or Word document.

Workflow

- Modify the Rmd file.
- If the modifications are code, run the code in the console to debug.
- Knit.
- Look over the output document.
- Repeat.

Packages

- Base R contains LOTS of useful commands to analyze, describe, and manipulate data. However, other people have developed new commands which are not packaged in the standard version of the program. Therefore, people create packages which contain their new commands. We will use the `dplyr` package quite often in this class so it is time to learn how to load a package.

- The following chunk loads the `dplyr` package into your R Markdown document. You also need to run the code if you want `dplyr` to be loaded in your RStudio session.

```r
library(dplyr)
```

- Go to the **Packages** Tab. Notice that `dplyr` is listed and the box next to `dplyr` is checked if it is loaded.

- You may want to use a package that is NOT listed in the Packages Tab. In that case, you will need to installed it using the “Install” button. Once a package is installed, it will show up in your packages list.

Loading Data

- For most labs a first step will be to load in a dataset for analysis. Below is an example of loading a dataset of all recorded volcanic eruptions:

```r
library(readr)
eruptions <- read_csv("/home/courses/math141f19/Data/GVP_Eruption_Results.csv")
```

- Run the code to load the data into the RStudio Session.

Stored Objects

- You should now have two objects in the *Environment* tab: **eruptions** and **x**. Notice that the number of observations and number of variables are listed for datasets. Click on **eruptions** to view the dataset.

Avoiding Problems

- Knit often! Knitting saves your work and will tell you if there is a bug.
- If the document won't knit…
  - Your Rmd file must be self-sufficient. Therefore, if you read in a file in the console but didn't put the `read_csv(...)` command into your Rmd, then it won't knit. If you defined an object in the console (e.g, `a <- 1`) and not the document, it won't knit. Each time a document knits, it only uses the functions and objects provided in the document.
  - Make sure you didn't put "interactive" functions such as *View(eruptions)* and *?mean* in the Rmd file.
  - Check that you changed `eval=FALSE` to `eval=TRUE` in R chunks where you filled in the code. It may be that an object created in the unevaluated chunk is needed in a later R chunk.