Homework 1 - File Management, R, RStudio, and LaTeX

#NAME: Andres Castano #NETID: ac986 #**DUE DATE: August 29, 2016, 1:00pm**

#Instructions

1) Read through this document and perform all indicated actions and answer all questions.
2) When you are finished, knit this document and upload the following **three** files to blackboard:
    a. This modified .Rmd document
    b. The pdf that results after knitting this .Rmd file (and, if Word or html were used, converting to a pdf).
    c. The screenshot requested on Homework1.html.

Code Chunks

RMarkdown documents (.Rmd) are designed for reproducible research. They are composed of a combination of text and evaluated R code. Using R Markdown, you can describe your study and perform the statistical analysis in R in one document. This allows you to recreate this analysis at any later point in time.

To run statistical analysis through R Markdown, you need to specify in your .Rmd document where R can find the commands you would like to run. Any R code you would like to run in your document must be in a *code chunk*. A code chunk looks like this:

To see the output of any code you put into a code chunk, you must knit your document by selecting *Knit (HTML or PDF or Word)* from the menu for this document. If you knit your document right now, nothing will show up for the above code chunk since there is no code to run in it.

You will see several other code chunks throughout this document that contain R code, but the code has not been evaluated (these are highlighted in light gray). In this .Rmd document, the first line of these code chunks are all set to `eval=FALSE`. For these code chunks to be evaluated in your pdf or Word document, you need to set `eval=TRUE`. For now, keep these code chunks set to `eval=FALSE`.

1. Add a code "Chunk" to this document here.

To do this, first click on the place in this document where you want the code chunk to be added, then click on the arrow next to "Chunks" on the menu for this document and choose "Insert Chunk". **Important: Code chunks must be left justified to run the R code within properly.**

In the next sections, we will explore some of the commands that can be run in a code chunk.

Setting Your Working Directory

The first thing you want to do when using R or Rstudio is to set your "working directory". R will assume that all of the files it needs to access are in this directory. All .Rmd documents and the Console below have *separate* working directories. The following two R functions associated with working directories can be used either in a code chunk (for a .Rmd document) or in the console below.

1) `getwd()` - This command indicates your current working directory.

2) `setwd('/filepath/')` - You can set the working directory for a .Rmd document or the console by specifying a filepath to the folder you would like to use as your working directory.

For the console only your working directory can also be set by selecting the following from the RStudio menu at the top of your screen, *Session -> Set Working Directory -> Choose Directory.*

1. First we will determine and then set the working directory of the console.

    a. In the console below, type in: `getwd()` (without the tickmarks). This will show you the current working directory for the console.

    b. Change this working directory to the folder for homework 1 by either using the `setwd()` command or by using the menu options indicated above.

2. Now we will confirm that the working directory for this .Rmd document is the folder for homework 1 by using the `getwd()` command in a code chunk here. **Reminder: To see the output of any code you put into a code chunk, you must knit your .Rmd document.** Tell R to evaluate the following code in your knitted pdf/html/Word document by changing the first line in the code chunk given below to `eval=TRUE` (alternatively you can delete `, eval = FALSE`). Knit this document now to see the filepath of the working directory for your .Rmd document.

```
getwd()
```

```
## [1] "/Users/andres/Dropbox/3 CORNELL/1 Fall 2016/BTRY 6010 Statistical Methods I/Homework/HW1"
```

#Some Basic R Commands

1. At the most basic level, R can be used as a calculator. The following chunk of code gives some basic R commands for calculation.

```
5-3
```

```
## [1] 2
```

```
5+3
```

```
## [1] 8
```

```
5*3
```

```
## [1] 15
```

```
5/3
```

```
## [1] 1.666667
```

```
5^3
```

```
## [1] 125
```

Now in the first line of this code chunk, set `eval=TRUE`. Knit your pdf to see these mathematical expressions evaluated in your pdf file.

For any code in your R Markdown document, you can also run it directly in your R console by highlighting the code you wish to run and choosing the "Run" option from the menu at the top of the .Rmd file. Do this now with the code above. You will see your output in the console below. This allows you to see how your code is working without having to knit a Word document or pdf.

2. You can use variables to represent various kinds of objects (numbers, vectors, matrices, function output, and data frames, which we will explain later) in R.

The assignment operator in R is "<-" or "=". It assigns whatever is on the left hand side to have the value of whatever is on the right hand side. In the following data chunk, we will assign the variable x to represent the number 5 and the variable y to represent the number 3. We will now run the previous calculations using the variables x and y.

```
x=5
y=3
x-y
```

```
## [1] 2
```

```
x+y
```

```
## [1] 8
```

```
x*y
```

```
## [1] 15
```

```
x/y
```

```
## [1] 1.666667
```

```
x^y
```

```
## [1] 125
```

Change `eval=TRUE` so that this code will run in your pdf or word file.

You don't need to use just one letter as can be seen by running the code below.

```
TOM=x
JERRY=y
TOM+JERRY
```

```
## [1] 8
```

However, R is case sensitive, so you will see an error when R tries to run the code `Tom+Jerry` below. Set `eval=TRUE`. Two things happen when you try to knit your pdf: 1) you will not get an updated pdf and 2) an error message will pop up in the R Markdown tab of RStudio below. The important information from this error message is that the "object 'Tom' is not found." Now, set `eval=FALSE` so that you can again knit this .Rmd document.

```
Tom+Jerry
```

Note, unless you clear your workspace, R "remembers" any previously defined variables.

3. All of your R Markdown documents and the R Console below have separate "workspaces." Your workspace contains all of the variables you have defined in your R Markdown document or your console with their values.

Any variables that we defined in this R Markdown document that were not run through the R Console below are in the workspace for this .Rmd document, but not the workspace for the R Console. The workspace of your R Console can be found under the "Environment" tab in the upper right window of RStudio. Currently, this workspace should be empty since all code where we have assigned values to variables was run only by knitting a pdf from the .Rmd document and not directly through the R Console.

R code can be run directly through the R Console in three ways:

1) You can copy code from another document and paste it at the command prompt in your R Console.

2) As mentioned above, you can highlight the code in your R Markdown document and choose "Run" from the menu for that document.

3) You can type code in directly at the command prompt.

Variables defined using any of these three methods will then be in the workspace for the R Console.

Type `BUGSBUNNY=2` into the console below. `BUGSBUNNY` is now in the workspace for the R Console. Check to make sure you now see `BUGSBUNNY` defined under the "Environment" tab.

Notice `x` and `y` are not in the workspace for the R Console, they are only in the workspace of this R Markdown document. Type `x` in the console below. You should get the error message, "Error: object 'x' not found." You can put `x` and `y` in your console's workspace by either copying the first chunk of code in part (2) and pasting it at the command prompt in your console below OR by highlighting this code and choosing "Run" from the menu above. Please do this now. Note that `x` and `y` are now in your workspace for the R console.

Now try and knit your pdf with the following code chunk set to `eval=TRUE`.

```
BUGSBUNNY=2
BUGSBUNNY+1
```

```
## [1] 3
```

Again, you should have generated an error message in the R Markdown tab below, and you should not have an updated pdf or Word document. Although, `BUGSBUNNY` is defined in your R Console workspace, it is not defined in this R Markdown document.

Edit the code above by including the line `BUGSBUNNY=2` above `BUGSBUNNY+1`. Try to knit your document again.

Since you have now defined `BUGSBUNNY` in the .Rmd document, this code should be evaluated in your pdf or Word document without error.

Every time you start a new project, you will want to clear your workspace in the R Console. To clear your workspace at any time, from the RStudio menu choose *Session -> Clear Workspace.*

4. For any function in R for which you currently have the library for that function installed and loaded, all documentation for that function can be found by typing a question mark in front of the name of the function and running it in the R console. For instance, to get documentation on the `getwd()` function, run the following R code by highlighting it and selecting "Run" from the menu above.

```
?getwd
```

The documentation for `getwd` will now appear in the lower right window of Rstudio under the tab "Help."

Some General Comments on Coding and Debugging

Every programming language has a proper syntax that it understands. When you do not use this syntax properly, one of the following will happen:

1) Your code will not run, and you will receive an error message.
2) Your code will run, but not as you intended it to.

When you use a R Markdown document, you are working with two different computer languages: one that tells R Markdown how to knit the document and the other to evaluate all code chunks in the document (R). Keep the following in mind when you are editing R Markdown documents:

1) Before you edit your R Markdown document for a given homework, knit it. This will give you a starting point from which you know that your document could knit correctly.
2) You do not need to knit your document every time you edit the R Markdown file, but you should knit often enough that it narrows down the area in which you have a coding error.
3) How things are spaced is important in R Markdown. If your file is not knitting *nicely*, you may have an issue with spacing.
4) R Markdown will not knit *copies.* So, if your R markdown file has a (1) or (2) before .Rmd (e.g. BienJ-HW1(1).Rmd), you need to re-name it with the (1) or (2) removed and then knit.

5) Don't panic if you have a coding error either in R or R Markdown.

   a) Your first step should be to look at the error message generated when you tried to run your code or knit your document. Often (but not always) the error message will tell you exactly what you need to fix. If you don't understand the error message, copy and paste the error message into google and see if something useful comes up. Google results from stackoverflow are particularly helpful for coding errors.

   b) If (a) doesn't help:

      i. Re-save your file under a different file name.

      ii. Find the original un-edited .Rmd file in your "Downloads" folder. Move it to the folder you created for the current homework and re-name it as specified for the current homework (e.g. BienJ-HW1.Rmd).

      iii. Copy and paste your code/answers from your file from (1) one by one into the new file. Knit every couple of edits to confirm your document is still knitting.

      iv. If your document doesn't knit after a specific edit, your error is in this piece of code.

6) **How to ask for coding help from TA's**: If you have a bug in your code and are not able to fix it, first try to have the following information

   a) Have a good idea of where the error in your code is e.g. by going through the steps in 5b.

   b) Take a screenshot of the error that pops up while trying to knit your code. If you can get a screenshot of an error while running the code in the console, this could potentially be more informative.

   If you know there is a specific line of your code that is causing the error, **and the code does not give away a homework solution** you can post your code + error on Piazza. Otherwise you can send an email to the course account btry6010@cornell.edu. Please include your .rmd file, a short description of where you believe the error is occurring, and a screenshot of the error you are receiving.