BIFS-613
Assigned Exercise (AE) #1
100 pts

Due: Sunday 09/15/19 by 11:59 PM EST (end of Week 1)

Purpose: Preparing for the course by installing the tools and learning some base R coding syntax.

Submit <u>three</u> files:

1. Parts I-III as "LastName_FirstName_AE1.**pdf**" (example: Norris_Alexis_AE1.pdf)
    a. Insert your answers into **AE1_template.docx** (in Week 1 Course Content) and then save/export as pdf
2. Part III *code* as separate file named: "LastName_FirstName_AE1.**Rmd**"
3. Part III *report* as separate file named: "LastName_FirstName_AE1.**html**"

Note about styling I use to help with readability:

- Code is written like `this`
- The specific output/components you should include in your AE1 submission start with "➤"

Pre-requirement: R and RStudio installed

RStudio is a popular IDE for using R. I recommend using swirlstats instructions. If you already have R and RStudio installed, I recommend updating to the newest versions.

If you are having issues with the installations:

1. Watch the videos under R Resources in Course Content:
    a. Mac - R, RStudio
    b. Windows - R
2. Post as new thread in R issues discussion, including:
    a. Your operating system and version (macOS 10.14, Windows 10, Ubuntu 18.04, etc.)
    b. Details, such as the error message(s) you are getting

PART I: Creating accounts (10 pts total)

These accounts are useful and free. You can use whatever email address you prefer to register. If you already have the account(s) setup, you do not need to create new one(s) - just provide screenshot(s) of your existing.

A. GitHub (5 pts)

GitHub is a popular platform for sharing tools and code. It can serve as your "code portfolio" and thus be useful in your career advancement. Later this course, in the Group Project, there is an opportunity for extra credit if you upload your R code files to your GitHub account.

Create a GitHub account - FREE version (not "Pro"!) here.

➤ Insert url for your github account.

For example, mine is: https://github.com/anorris8

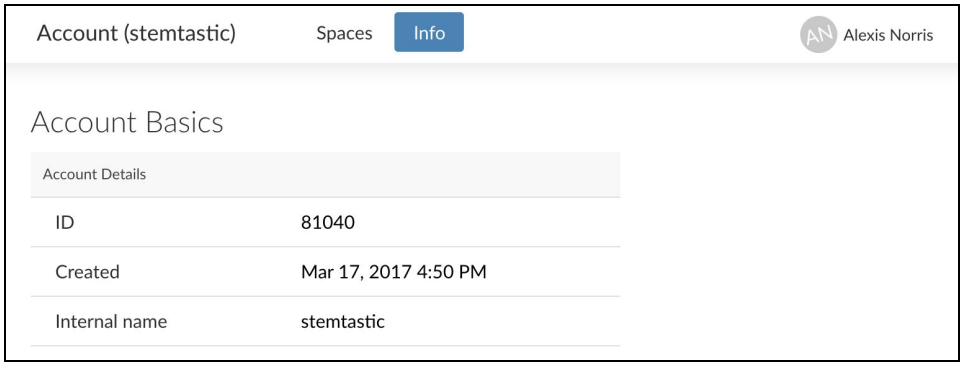
B. rstudio.cloud (5 pts)

RStudio Cloud is a web application version of RStudio desktop. It will serve as a backup to the desktop version of RStudio (very useful if you have issues with the desktop version).

Create an RStudio Cloud account here. You can use your GitHub account (optional).

➤ Insert a screenshot of your Account Info screen. To get to the screen: click on your name (top right) → click on Account (in right menu that appears) → click on Info (at top).

For example, mine is:



PART II: Installing R packages (15 pts total)

Most R packages are in the CRAN, Bioconductor, and GitHub repositories. Here you will check that you are able to install packages from each. First, open RStudio.

A. CRAN (5 pts)

Install devtools package. Hint: scroll down to "Materials" and click on "README" link. Then look under the "Installation" section in the webpage that opens.

Notice that when you install the package, a message appears in your console telling you what additional packages are being installed because devtools requires them:

```
also installing the dependencies 'ini', 'clisymbols', 'desc', 'gh', 'xopen', 'brew', 'commonmark',
 'usethis', 'git2r', 'memoise', 'pkgbuild', 'pkgload', 'rcmdcheck', 'remotes', 'roxygen2'
```

devtools has many useful functions. One is installing packages from other repositories (which we'll do in **IIC** below). Another is the `loaded_packages()` function. When you type `loaded_packages()`, it tells you both what packages you already have loaded and where on your computer they were installed ("path").

➤ What is the path of devtools on your computer? Hint: you'll first need to load the package using library function: `library(devtools)`

B. Bioconductor (5 pts)

Install bioconductor core packages using the code (in grey box) under "Install Bioconductor Packages" here. You may be prompted to update packages; unless you have a reason to not update your packages (such as if you are currently working on an analysis that uses the package and you

want to avoid changing package versions until you complete the analysis), it's usually appropriate to update all.

For example, I received the following message and typed **`a`** *to update all:*

```
Bioconductor version 3.9 (BiocManager 1.30.4), R 3.6.1 (2019-07-05)
Update old packages: 'devtools', 'ggforce', 'ggpubr', 'ggraph', 'IRanges', 'pbapply',
  'seriation'
Update all/some/none? [a/s/n]:
```

One useful bioconductor package is Biostrings. Install and then load the Biostrings package. We can lookup information about a function in a package using **`?`** before the function. Type **`?reverseComplement`** to find out how to use it. The documentation will appear in the "Help" tab (default location is lower right corner in RStudio).

➤ What is the purpose of the Biostrings' **`reverseComplement()`** function? Hint: look under "Description."

C. GitHub (5 pts)

Install tidyverse package's GitHub version, which includes newer features not yet added to its CRAN version. We use devtools (installed above in **IIA**) to install GitHub packages.

Hint: scroll down to the "Installation" section under "README.md" file, use the instructions under "`# Or the development version from GitHub`"

You may receive a prompt that asks you if you would like to update other packages that the tidyverse package depends on. Similar to already mentioned above in IIA, you may receive a prompt to update your existing packages.

For example, I received the following message and typed **`1`** *to update all:*

```
Downloading GitHub repo hadley/tidyverse@master
These packages have more recent versions available.
Which would you like to update?

1: All
2: CRAN packages only
3: None
4: hms       (0.5.0 -> 0.5.1) [CRAN]
5: rmarkdown (1.14  -> 1.15 ) [CRAN]
6: whisker   (0.3-2 -> 0.4  ) [CRAN]
7: sys       (3.2   -> 3.3  ) [CRAN]
8: xfun      (0.8   -> 0.9  ) [CRAN]

Enter one or more numbers, or an empty line to skip updates:
```

The tidyverse package is actually a collection of packages. Load the tidyverse package and you will see the names and versions of the packages included in tidyverse. One of the packages we will use a lot is ggplot2. To learn more about a package (see its documentation), we can type a **`??`** before the package name and look in the "Help" tab (like we did above in **IIB**). Try looking up the ggplot2 documentation.

➤ Insert a screenshot (partial is fine) of the ggplot2 package's documentation.

PART III: RMarkdown (75 pts total)

First, complete swirlstats: swirlstats is a way to learn R interactively and *in* R. You'll learn some basics in "A_(very)_short_introduction_to_R" lesson's Modules 1 & 2. There is an accompanying pdf guide that might be helpful if you are new to R. Instructions:

1. Open RStudio
2. Install swirlstats by typing: `install.packages("swirl")`
3. Load swirlstats by typing*: `library(swirl)`
4. Install the lesson by typing: `install_course("A_(very)_short_introduction_to_R")`
5. Start the lesson by typing: `swirl()`
6. Complete modules 1 & 2 (plan *at least* one hour if you are new to R)

*If this doesn't work, you haven't successfully installed the lesson. Try repeating Step #2 and seeing if any error messages appear in the console.

If you run into issues, you can post to R Problems Discussion (as new thread; please include details about your issue!) or type `skip()` to move to the next question (try to avoid this).

If you have time, you can start Module 3 (optional).

Then, start the RMarkdown assignment:

RMarkdowns are versatile documents that contain text and code. They will serve as our "lab notebooks" for this course. We'll use the rnorm function as an example (like we did in the swirlstats lesson). It creates a random series of numbers from a normal distribution. Answer the questions below. For help: use the swirl lesson guide for `?rnorm`, R4DS Chapter 27 for RMarkdown formatting, and for anything and everything - it's ok to google! Googling is often the best way to troubleshoot your code.

Create RMarkdown file (35 pts total)

Download the **BIFS613_AE1.Rmd** file (in Week 1 Course Content). Then open the Rmd file in RStudio. Note that you will first need to install the

1. Install the `sessioninfo` package from CRAN. **(5 pts)**

2. In the YAML, enter your name for `author:` and today's date for the `date:` **(5 pts)**

3. In code chunk #2: Create a variable `x1` that has 500 random numbers from a normal distribution with a mean = 50 and standard deviation (SD) = 5 using the `rnorm()` function using the `rnorm()` function. Use the `summary()` function to calculate the mean and median of `x1`. **(10 pts)**

4. In code chunk #3: Using the `hist()` function, plot the distribution of `x1` values. This visually shows us that the rnorm is generating numbers in a normal distribution. We'll talk more about types of distributions; the important lesson here is the value of plotting your data to see/check it. **(5 pts)**

5. In code chunk #4: Create another two variables `x2` and `x3` like you did above for `x1` (500 numbers, mean = 50, and SD = 5 using `rnorm()`) - so the same `code`. But now add `set.seed(1324)` before each, like below. **(5 pts)**

```
set.seed(1324)
x2 <- rnorm(insert your code here)
set.seed(1324)
x3 <- rnorm(insert your code here)
```

6. In code chunk #5: Using the `plot()` function, create the following two plots. **(5 pts)**

Plot #1: `x2` (x-axis) vs. `x1` (y-axis)

Plot #2: `x2` (x-axis) vs. `x3` (y-axis)

➤ Submit your final Rmd code as "LastName_FirstName_AE1.**Rmd**"

Generate report from RMarkdown file (15 pts total)

Click "knit" button at the top of Rmd file, and select "knit to html." Make sure the code runs correctly and you generate the html report. You will receive full credit if you have made the edits to the Rmd from above in **A** and the Rmd successfully generates your html report (even if incorrect code/plots). Note: The html will be in the directory (folder) where you have the Rmd saved.

➤ Submit the html report from your final Rmd code as "LastName_FirstName_AE1.**html**"

Interpretation of results in report (25 pts total)

➤ Answer the following questions. Remember, it's OK to google!

A. What happens if you add the code chunk option `eval=FALSE` to code chunk #1 and re-knit? Why does this happen? Hint: See R4DS Chapter 27 for help. **(5 pts)**

B. What happens if you remove the `###` (above code chunks) and re-knit? **(5 pts)**

C. Looking at Plot #1 and Plot #2, what did adding the `set.seed()` before running `rnorm()` do? **(5 pts)**

D. What is the value of using `set.seed()` before using a function that generates random values? **(5 pts)**

E. Why do we end the RMarkdown with `sessioninfo()` (in code chunk #6)? **(5 pts)**