BSDS 100: Intro to Data Science with R
Assignment 2
Due 2/16 at 11:59pm

Directions: Write a single R markdown (.Rmd) file that answers each of these questions and produces a knitted .pdf which holds your responses. Make sure that all code can be successfully run on any computer. Put your name and date at the top of the document. For all questions that require written responses, write the answer (numbered appropriately) in markdown, not as a comment in the code. Turn in the .Rmd file and .pdf file on Canvas. Late assignments are not accepted.

Playlists: Here are a couple of playlists from me to help you get started on the assignment:

Disco Then/Now: https://open.spotify.com/playlist/1qTdsBJZprg9ab8I5VF8Xd?si=5b0bac7e73e446f3

Psych Bass Mix: https://open.spotify.com/playlist/4BfgHS6HINBNSpyPf0xD1v?si=25e5a49350214975

1. *Loading pre-stored data and viewing the data:* We first consider analyzing the *iris* dataset. This dataset is already available in R, so we only need to call the dataset from the console. Load and view, and read a description about the data using the below commands.

```
#Load the data
data(iris)

#Look at the data
iris

#Read a description of the data
help(iris)
```

When data is first loaded to your R console, it is stored as a **data frame** structure. Data frames are a tabular representation of data and can contain any mix of characters, numerical quantities, or factors. An important aspect of the data frame is that it contains both row and column names. Often, we want to extract a subset of the data to take a closer look. This can be done directly using the row or column name *or* by calling the number of the row/column you want to extract. Follow the example below for an illustration.

```
#Display the column and row names of the data
colnames(iris)
rownames(iris)

#Look at the 10th row
iris[10, ]

#Look at the 3rd column
```

```
iris[, 3]

#Alternatively, just look at the variable "Petal.Length"
iris$Petal.Length

#Store columns 1-2 and rows 10-20 for later use
subset_data = iris[10:20, 1:2]

#Store the species names for later use
species_names = iris$Species
```

Note: Remember that `R` is case sensitive!

Questions

(a) What are the different variables in this dataset? What are the measurements of each of these variables for the 10th sample?

(b) How many samples are in this dataset? What are the different species of flower that have been measured?

2. *Summary Statistics*: With any dataset, one of the first forms of exploratory analysis involves calculating summary statistics from the data like the five number summary: the minimum, median, third quartile, maximum and mean. We are also interested in the variation of a variable as measured by its standard deviation.

   - One can use the *summary()* function to calculate a five number summary. The *apply()* function can be used to find the standard deviation of each row or column of a data frame or matrix structure. We note that the *apply()* function is very general as it can apply any other `R` command across rows or columns of a dataset. Here, we apply the *sd()* function across the columns (and then rows) of the *iris* dataset. Calculate these summary statistics of the *iris* dataset using the code below.

     ```
     #calculate the 5-number summary of the {\it iris} dataset
     five_num = summary(iris)

     #calculate the standard deviation across the first four columns of the dataset
     sd_cols = apply(iris[,1:4], 2, sd)

     #calculate the standard deviation across the samples of the dataset
     sd_rows = apply(iris[, 1:4], 1, sd)
     ```

   - Now, we can vizualize the quantitative summary of the 4 variables by building a boxplot using the *boxplot()* command. Plot a boxplot using the following command:

     ```
     boxplot(iris[, 1:4], main = "Boxplot of Iris Variables")
     ```

Questions

(a) How many observations of each species are there in the dataset?

(b) Which two variables have a median that is smaller than their corresponding mean?

(c) What is the standard deviation of the sepal length measurements?

(d) Calculate the 5-number summary and standard deviations of the subset you extracted earlier (*subset_data*). What is the standard deviation of the sepal length measurements for this subset?

3. Create the vector

```
> atomic_vec = c(1, 4, 3, 2, NA, 3.22, -44, 2, NA, 0, 22, 34)
```

Now, create code that runs to answer each of the following questions.

(a) How many positive numbers (> 0) are there in this vector?

(b) How many negative numbers (< 0) are there in this vector?

(c) How many 0's are there in this vector?

(d) How many `NA`s are there is this vector?

(e) How many numbers in the vector are non-zero **and** not `NA`s?

(f) What is the sum of the positive numbers is this vector?

(g) What is the sum of the negative numbers is this vector?

4. Consider a vector of length 1000, where F_n is the nth number in the sequence. Then the Fibonacci sequence is the vector where the following recursion holds:

$$F_n = F_{n-1} + F_{n-2}$$

That is, the nth number in the sequence will be the sum of the previous two numbers.

(a) Create a vector, `fib_vec`, that has the first 1000 numbers in the Fibonacci sequence using the following code (which includes a `for` loop that we'll talk more about later in this course):

```
#initialize the vector
fib_vec = rep(0, 1000)

#store the first two entries to be 1
fib_vec[1] = 1
fib_vec[2] = 1

#iterate to get the remaining values
for(i in 3:1000){
fib_vec[i] = fib_vec[i-1] + fib_vec[i-2]
}
```

(b) What are the first 8 and last 8 entries of `fib_vec`?

(c) Using the Fibonacci numbers generated above, generate a vector (of length 999) with values (again, don't print these out)

$$\mathtt{z\_n} = \frac{F_{n+1}}{F_n}$$

For this sequence, one could use a `for` loop as used in part (a), or better yet note that dividing two vectors of the same length will return a vector whose entries contain the division of entries in each vector. This is very useful for such calculations and this vector operation is fairly unique to `R`!

(d) Plot the first 10 entries of the vector z_n using the command plot(z_n[1 : 10]). Then add a line to the plot using the following command:
abline(h = (1 + sqrt(5))/2)
This value is known as the golden ratio in mathematics.

(e) Comment on the plot that you obtain. What do you observe?

(f) What is wrong with typing the following code?

```
x = fib_vec(1:5)
```

5. Using the Fibonacci vector above, create the following data structures. Remember that using the ?foo will provide documentation on the function foo as needed. And be careful about the use of arguments here to get the data structure you want.

- A matrix of size 100 × 10 named fib_matrix1 whose columns, when stacked on top of one another will return the original vector.
- A matrix of size 100 × 10 named fib_matrix2 whose rows, when stacked side by side will return the original vector.
- An array of dimension 10 × 10 × 10 names fib_array where each 10 × 10 matrix in the array is such that when its columns are stacked on top of one another would generate a Fibbonaci vector of length 100.

Answer the following questions

(a) What is the mean of the 18th row of fib_matrix1?

(b) What is the standard deviation of the 8th column of fib_matrix2?

(c) What is the entry in the 5th row of the 2nd column of the 8th matrix in fib_array?