Data Science with R: Problem Set - 1

Introduction to R

R is a statistical coding language made for and by statisticians. We will be using R predominantly in this course as well as throughout your degree. It is thus absolutely crucial that you invest in getting familiar with the language.

To setup, you need the following:

- R downloaded from https://cran.r-project.org/ compatible with your system
- Rstudio is **highly** recommended. RStudio is an Integrated Development Environment (IDE) for R. That means that Rstudio allows you to integrate the various features of R into one visual GUI environment.
- Latex installed on your system in order to use R Markdown (or now Quadro!). Quadro allows us to write pdf documents integrating mathematical equations, automatic code evaluations, plots!

Some Coding Practices

As we progress in this course, we want to make sure we develop good coding habits that will help us in the future

- Don't be lazy!
  - Codinglanguagesarelikeanylanguage. Every object, operation, etc is a word and you must use "space" to make it easier to read.
  - Comment your code! See `runme.R`‘ file for how I use comments in my code. That will help the reader understand you better.
- Be organized
  - Don't save files randomly your desktop. Make a folder for the course, and within that folder, make subfolders as appropriate.
  - Pretty soon you'll be writing long complicated codes. Think about which codes should be in which files.
  - Name files appropriately
  - Remember to push changes to GitHub when your work session is done.

- Naming objects
  - Don't use `a`, `b`, `c`, etc only. Use variable names based on their utility.
  - Assume that the next time you run your code, you have no idea what the code is supposed to do.
- Script vs Console
  - R has both an active console and a script.
  - The script is basically a text file that allows you to keep organize your working code. Make sure any script you write, can run in a different person's computer with a fresh R session.
  - The console is a place for you to run R code. Additionally, during the process of writing code, you can view your results as it comes along, debug your code, and keep a track of results.

Problems

Perlimiary Exercise

1. Write a function in R to find $n!$. Verify your answer with inbuilt `factorial()` function. You must try this problem in both way (Recursive & Iterative).
2. Euler's number, e, is accessed through exp(1) in R. Similarly e^3 is `exp(3)`. It is well known that
$$e = \lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n$$
Write a function in R that calculates the right hand side limit for a user-given value of n.
3. Recall the Sociology survey dataset use have encountered in Runme.R: Download the SOC171A Survey (Responses).csv file on your computer. Using `read.csv()` function, load the locally downloaded `seating.csv` into your R session.

```
data <- read.csv("SOC171A Survey (Responses).csv")
head(data)  # to see the top of the data
str(data)   # structure of the data
dim(data)   # dimensions
```

Now Run the following lines and observe what we are getting in Male_data object.

```
k = (data$Gender == "Male")
Male_data = data[k,]
```

Write R code to find the number of people who have category OBC and Marriage Preference as Love Marriage.

4. How do you install a package in R? The below command (in console) is used to install a package in R:
   install.packages("package_name")

Your Task is to install following packages in your System:

- tidyverse
- matlib
- pracma
- rvest

5. Go to "https://lgatto.github.io/2017_11_09_Rcourse_Jena/before-we-start.html" Read more from this page.
6. Define an R vector of the first 1000 prime numbers.
7. Define an R vector of the first 500 Fibonacci numbers.
8. An "all-rounder" in cricket is a player who performs well in both batting and bowling. The dataset below has the batting and bowling ODI averages of selected male players.

   ```
   cricket <- read.csv("https://dvats.github.io/assets/course/mth208/battingbowling.csv
   ```

   (A high batting average is good, a high bowling average is bad.) Let's say a decent batter is someone with a batting average higher than 25 and a decent bowler is someone with a bowling average below 40.

   1. Create a sub-dataset of all all-rounders.
   2. Which team has the most all-rounders?
   3. Which team has the least all-rounders.

Simulating Experiments

In this Section, we will learn about simulating random experiments in R. Let's start simple. Hope you have done your reading assignment(We are not suppose you to understand everything from your reading Assignment, Take a chill!!)

One of the simplest random experiment is tossing a fair coin. `R` can do this for us using command `rbinom()`. It is random dram from Binomial Distribution.

```r
# n =  number of coin tosses
# size = 1
# prob = probability of success
rbinom(n = 1, size = 1, prob = 0.5)
```

```
[1] 0
```

```r
# 1 = success (heads), 0 = failure (tails)
```

We can also do multiple coin tosses

```r
rbinom(n = 10, size = 1, prob = 0.5)
```

```
 [1] 0 1 1 0 1 0 0 0 0 1
```

Now that you've understood the commands, write `R` code for the following:

a. Simulate 1000 fair coin tosses and calculate the proportion of heads.

b. Simulate 1000 tosses of a coin that has probability of heads equal to 0.30. Calculat

2. We can simulate other experiments, like rolling a die:

```r
# Rolling a die
sample(x = 1:6, size = 1)
```

```
[1] 2
```

```r
# Rolling an unfair die
sample(x = 1:6, size = 1, prob = c(.1, .2, .1, .1, .3, .2))
```

```
[1] 5
```

```r
# drawing a random number between [a,b]
# n =  number of random numbers
# min = a
# max = b
runif(n = 1, min = 0, max = 1)
```

```
[1] 0.7100556
```

Now that you understand simulating the experiments above, try the following exercises:

a. In a bag, there are 7 balls of 3 different colors: 3 are red, 2 are green, 2 are blue. Write a code to randomly draw a ball from the bag.

b. Suppose I throw a dart anywhere at random on a thread of length 5 cm (assume I will always throw the dart on the thread and never miss the thread – I am very good at throwing darts). Write an `R` code to simulate where the dart lands on the thread.

c. Consider the following matrix A and let A_1, A_2, A_3 denote the columns of A

$$A = \begin{pmatrix} 3 & 4 & -1 \\ 1 & 5 & 2 \\ -2 & 3 & -2 \end{pmatrix}$$

```
A <- matrix(c(3, 1, -2, 4, 5, 3, -1, 2, -2), nrow = 3,ncol = 3)
A
```

```
     [,1] [,2] [,3]
[1,]    3    4   -1
[2,]    1    5    2
[3,]   -2    3   -2
```

Write an `R` code to choose column i with probability p_i:

$$p_i = \frac{\|A_i\|}{\sum_{j=1}^{3} \|A_j\|}.$$

Here, $\|\cdot\|$ denotes Euclidean norm and can be calculated using the function `norm()`.

3. We will try to run a simulation whose answer should be close to `exp(1)`. You will need to use a few new commands in this. Note that, to define a vector of length 1000, you can use command

```
new <- numeric(length = 1000)
```

Also, we have learned `for()` loops, which can be used to implement a loop when the number of loopings are known. However, when the number loops are unknown and based on some condition, we can use the `while(condition){---}` loop. This runs the loop as long as the `condition` within the `while` command is satisfied.

a. Write an `R` function to count the number of random [0,1] draws it takes for their sum to exceed 1. The function should have no inputs and should return just one numeric output.

b. Write an `R` program to call the above function 1000 times and store all 1000 outputs in a numeric vector.

c. Return the average of the 1000 outputs. This should be close to `exp(1)`.

4. It's your 25th birthday, and your friends bought you a cake with 25 candles on it. You make a wish and try to blow them out. Every time, you blow out a random number of candles between one and the number that remain.

   a. Write an `R` function that `age` as an input and returns the number of attempts it takes to blow out all the candles. You may need the `break` command to write this function or use the `while` loop.

   b. Write an `R` program to call the above function 1000 times and store all 1000 outputs in a numeric vector. You now have 1000 simulated candle blowing experiments.

   c. How many times, **on average**, do you need to blow at the cake until all the candles are extinguished?

   d. Repeat the above for you 30th birthday.