ISE 3293/5013 Laboratory 5
Random sampling

This lab will investigate random sampling. We will start with the R function `sample().` This will introduce you to functions and how they can be very useful in simulation. The process of simulation utilizes random sampling and allows the statistician to trial theories in practice or to carry out analyses when there are no analytical results available. There are a number of discrete distributions that you can learn about in R. In this lab you will need to THINK through the code and understand the process.

Objectives

In this lab you will learn how to:

1. Simulate from first principles.
2. Use built in r-dist() functions.
3. Make appropriate plots.

Tasks

All output made please copy and paste into **this word file**. Save and place in the dropbox when completed. Anything you are asked to make should be recorded under the question in this document. There will be two files you need to upload:

- a pdf of this document (pdf) or the word file (docx)
- a text file of all the code you used to create answers (txt)

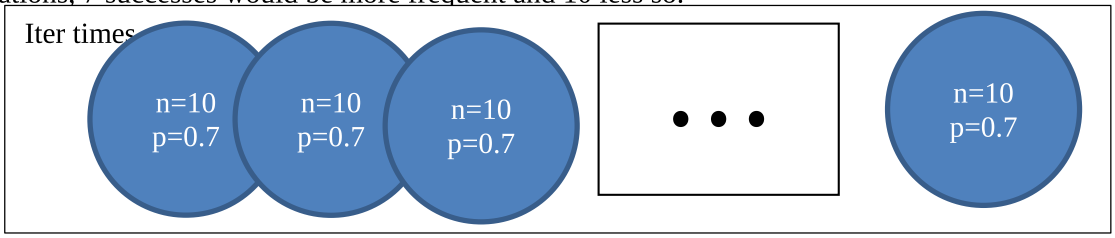
Note: All plots you are asked to make should be recorded in this document.

- Task 1
    - Make a folder LAB5
    - Download the file "lab5.r"
    - Place this file with the others in LAB5.
    - Start Rstudio
    - Open "lab5.r" from within Rstudio.
    - Go to the "session" menu within Rstudio and "set working directory" to where the source files are located.
    - Issue the function `getwd()` and copy the output here.

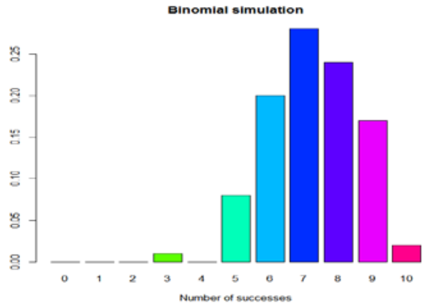
```
"F:/Google Drive - Saied/Courses/02 OU/11 Fundamentals of Engineering S
tatistical Analysis/02 Labs/05 Lab 5"
```

- Task 2
    - Make a new file for your code in RStudio editor, call it "mylab5.R" and place in it all the code you need to answer the tasks of this lab (copy and paste from lab5.R).
    - Use the hash # symbol and write your own comments in the code file explaining what the code does.

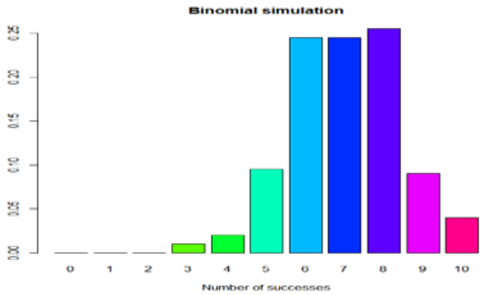
- Using code in lab5.R –`mybin()`, simulate a binomial experiment where n=10,p=0.7, and Y=number of successes. When running this particular function you will be carrying out iter experiments of making n=10 trials with the probability of a success p=0.7 and then recording the proportion of successes made over all the iterations. For example 1 success might happen on some iterations but it would have a low frequency over all the iterations, 7 successes would be more frequent and 10 less so.



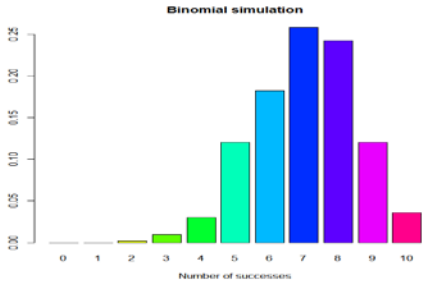
- Record the plots for each of the following (Use color)
  - 100 iterations



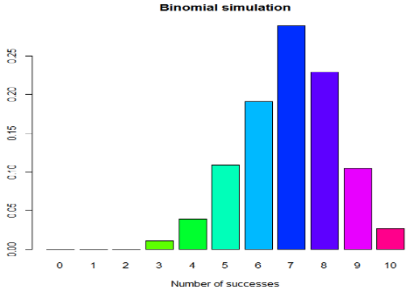
  - 200 iterations



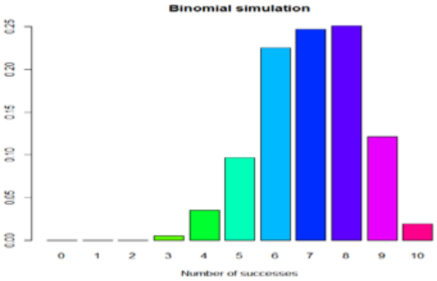
  - 500 iterations



        - 1000 iterations



        - 10,000 iterations



    - For the last simulation (10,000 iterations) copy and paste the table that is produced from the function.

```
0     1     2     3     4     5     6     7     8     9     10
0.000 0.000 0.001 0.014 0.035 0.098 0.198 0.270 0.229 0.126 0.029
```

    - Verify that each value in the table is approximately correct using `dbinom()` – you will need to look up the function using ?`dbinom`

```
0     1     2     3     4     5     6     7     8     9     10
0.000 0.000 0.001 0.009 0.037 0.103 0.200 0.267 0.233 0.121 0.028
```

- Task 3
    - Suppose that there is a bag of 20 marbles, 12 white ("1") and 8 black "0". Using the `sample()` function create a sample of size n=5 without replacement. Place the output here.

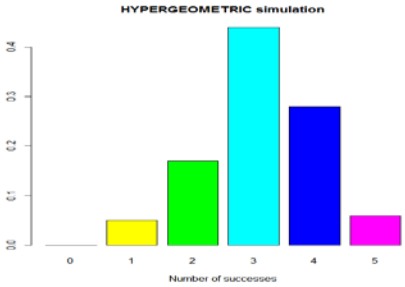
```
1 1 1 0 0
```

    - Now do the same with replace = TRUE.

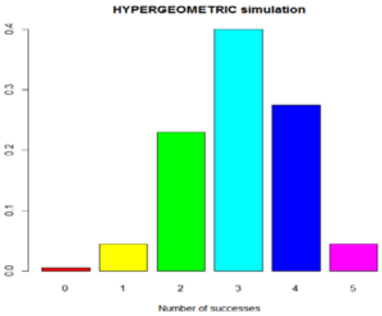
```
1 0 1 1 0
```

    - Use the function `myhyper()`, on D2L, assume Y = number of whites, remember sampling is without replacement, n=5. Use this code to make barplots of Y for the following number of iterations

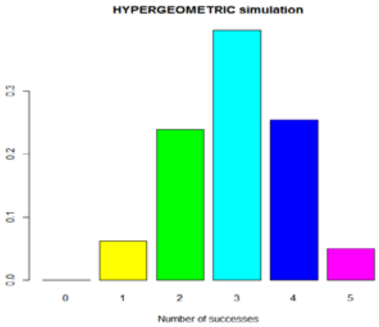
- 100



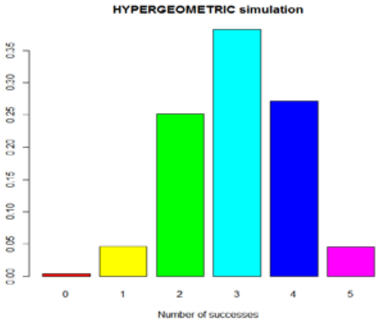
- 200



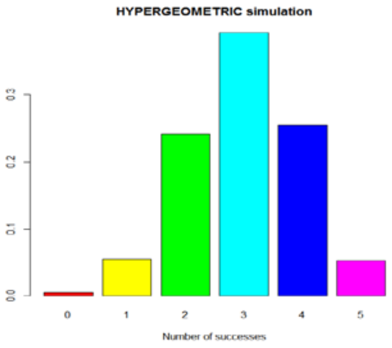
- 500



- 1000



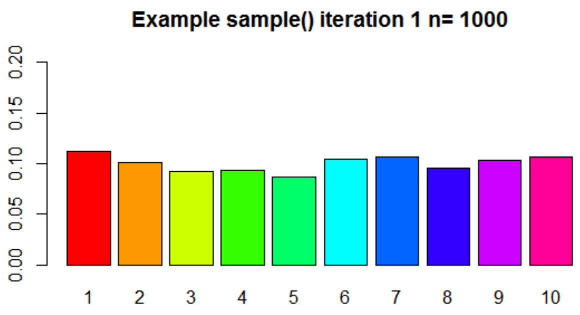
        - 10,000



    - For the last simulation (10,000 iterations) copy and paste the table that is produced from the function.

```
0      1      2      3      4      5
0.0179 0.1328 0.3519 0.3431 0.1357 0.0186
```

    - Verify that each value in the table is approximately correct using `dhyper()`
- Task 4
    - In lab5.r there is a function called `mysample().` Look at the code and describe what you think it should do – record your response here.
      For the first step, it make n samples from set of number between 1 and 10. Then make table of sample frequency. At the end, draw a bar plot of sample's relative frequency. The time parameter is used to pause for some seconds.
    - Run the function with the following arguments `mysample(n=1000, iter=30,time=1)`
    - Describe what you see here.
      The sampling occurs 30 times. Each sample includes n random numbers between 1 and 10. The bar chart show each of try for 1 second.
    - Record the last plot here.



- Task 5
    - Use R to calculate

        - $\binom{8}{4}$ – hint: Try choose()

          28
        - $P(Y > 4), Y \sim Pois(\lambda = 2)$

          0.9097765
        -
    - Some more calculations in R
        - $P(Y = 10), Y \sim NegBin(p = 0.4, r = 3)$ see D2L which has a full explanation of this.

          0.06449725
        -
        - $P(Y \leq 8), Y \sim Bin(n = 15, p = 0.4)$

          0.9049526
        -

################## LAB FINISHES HERE ##############################

- Task 6 – Extra for experts
    - Write some code that will simulate a neg binomial distribution. You could use the while() condition

```
mydnbinom = function(y = 10, r = 3, p = 0.4){

  choose(y-1,r-1) * p^r * (1-p)^(y-r)
}
```