MATH 4753 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 work is to be done on RMarkdown.

Submit all knitted documents including Rmd and txt Rscript file.

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

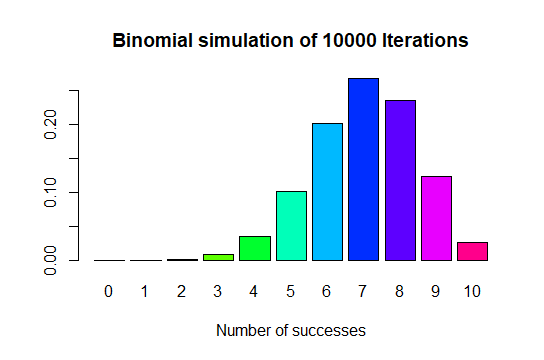
* Task 1
  + Make a folder LAB5
  + Download the file “lab5.r”
  + Place this file with the others in LAB5.
  + Start Rstudio
  + Open “lab5.r” from within Rstudio.
  + Go to the “session” menu within Rstudio and “set working directory” to where the source files are located.
  + Issue the function getwd()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

"C:/Users/cglen/Documents/Stat Methods/Labs/LAB5"

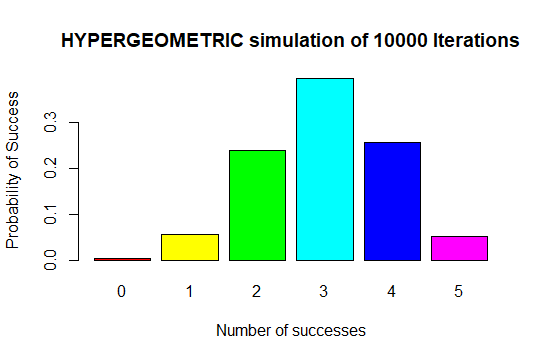
* Task 2
  + Make a new file for your code in RStudio editor, call it “mylab5.R” and place in it all the code you need to answer the tasks of this lab (copy and paste from lab5.R).
  + Use the hash # symbol and write your own comments in the code file explaining what the code does.
  + Using code in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

Iter times

* + 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.



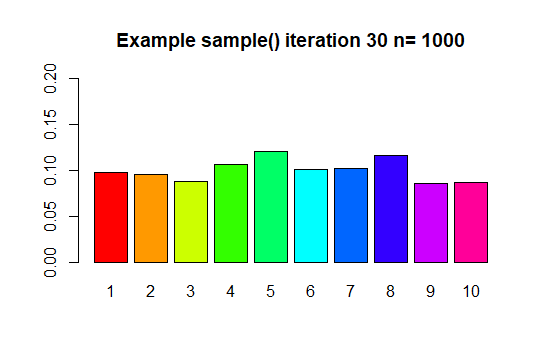
* + Verify that each value in the table is approximately correct using dbinom() – you will need to look up the function using ?dbinom
    - 0.0000059049 : 0.0001377810 : 0.0014467005 : 0.0090016920 : 0.0367569090 : 0.1029193452 : 0.2001209490 : 0.2668279320 : 0.2334744405 : 0.1210608210 : 0.0282475249
    - Yes, the plot corresponds to the binomial function that 7 is the peak, and rest of graph tends down.
* 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.
    - 0 1 0 1 0
  + Now do the same with replace=TRUE.
    - 1 0 1 0 0
  + Use the function myhyper(), on CANVAS, 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.



* + Verify that each value in the table is approximately correct using dhyper()

0.003611971 : 0.054179567 : 0.238390093 : 0.397316821 : 0.255417957 : 0.051083591

* Yes, the Hypergeometric plot follows the path of the Hypergeometric function for 0:5
* 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.
    - I think the sample code will show a 30 sample plots in 30 seconds
  + Run the function with the following arguments mysample(n=1000, iter=30,time=1)
  + Describe what you see here.
    - The code shows a sample plot for every n iteration of the sample
  + Record the last plot here.



* Task 5
  + Use R to calculate
    - – hint: Try choose()
      * 70
      * 0.947347
  + Some more calculations in R
    - see CANVAS which has a full explanation of this.
      * 0.02554091
      * 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