STA 570

lab 2

Open a new RMarkdown file and save it as lab2\_lastname

# Project Goals

In this lab assignment, you will be asked to:

* Simulate data to calculate probabilities

# Simulation Background

Calculating probabilities is sometimes painfully difficult, but *simulations* provide use with a very practical alternative to calculations based on formal rules. A **simulation** of a procedure is a process that behaves the same way as the procedure so that similar results are produced. Instead of calculating the probability of getting exactly 5 boys in 10 births, you could repeatedly toss 10 coins and count the number of times that exactly 5 heads (or simulated “boys”) occur. Better yet, you could do the simulation with a random number generator on a computer or calculator to randomly generate 1s (or simulate “boys”) and 0s (or simulated “girls”).

# Simulation Exercise

Let’s consider the following problem:

**Find the probability that among 23 randomly selected people, at least 2 have the same birthday.**

For the above problem, a simulation begins by representing birthdays by integers from 1 through 365 (for the number of days in a year), where 1 represents a birthday of January 1, and 2 represents January 2, and so on. We can simulate 23 birthdays by using a calculator or computer to generate 23 random numbers (with repetition allowed) between 1 and 365. Those numbers can then be sorted, so it becomes easy to examine the list to determine whether any 3 of the simulated birthdates are the same. (After sorting, equal numbers are adjacent.) We can repeat the process as many times as we wish, until we are satisfied that we have a good estimate of probability. Use the following code to simulate 1000 different groups of 23 birthdays. Use the results to estimate the probability that among 50 randomly selected people, at least 2 have the same birthday.

1. Create a new RMarkdown file and entitle it lab2\_lastname. Be sure to watch the lab 2 video in the assignment.
2. Run the following code. This part is just to see what one simulated sample of 23 people from 1000 looks like with respect to the number of people with the same birthday. Note that the ‘#’ is used to designate comments made within the R chunk.
3. How many days have two people with the same birthday? How many days have three or more people with the same birthday? **Note: Each time you run this code you will get a different result. You can add the line set.seed(1) before the first line of code in the R chunk to prevent different results.**

#to generate 23 birthdays  
dates <- sample(1:365, 23, replace = TRUE)  
  
#sort  
 sorted <-sort(dates)  
 sorted

#check for duplicates so we can see them  
repeats <- rle((sort(dates)))$length  
repeats

#this checks to see if there are duplicates  
length(dates) != length(unique(dates)) #TRUE if there are duplicates

The above code checks whether the one sample contains a repeat.

Now, you can run a for loop of 1000 iterations to get an estimate of the probability that among 23 selected people, at least 2 individuals have the same birthday.

1. The following code is the same as the above except now we are taking 1000 samples of 23 people, instead of just one sample. With the code below (you can run it all in just one R chunk), provide an estimate of the probability that at least two people in 23 have the same birthday. Since there are 1000 samples of 50, the estimate is found from the “print(dupe\_count/runs)” line of code.
2. Does the value that R calculated in problem 2 surprise you? Why or why not?
3. Google the “birthday problem”. You are looking for the problem that explains the probability of two people having the same birthday. How many people do you need to have the probability that two people have the same birthday is at least 1/2? Does this seem surprising to you? Why or why not?

#repeat the above sample 10,000 times  
#set a counter to count the number of repeat birthdays  
dupe\_count = 0  
runs =10000  
for (i in 1:runs) {  
 dates <- sample(1:365, 23, replace = TRUE)  
 if (length(dates) != length(unique(dates))) {  
 dupe\_count = dupe\_count + 1  
 }  
}  
#printing the percentage of duplicate birthdays to est. probability  
print(dupe\_count/runs)