IST 687: Lab 4

Brandon, TS, Michelle

May 9, 2018

Step 1: Annotate the Code

set.seed(2) #set the seed of R's random number generator =22, which the random objects can be reproduced  
  
sampleSize <- 30 #making a random array of 20000 numbers with a mean of 20 and standard deviation of 3  
  
studentPop <- rnorm(20000,mean=20,sd=3) #Taking a sample size of 30 of the studentPop variable we just created  
  
undergrads <- sample(studentPop,size=sampleSize,replace=TRUE) #Taking a sample size of 30 of the studentPop variable we just created  
  
grads <- rnorm(sampleSize,mean=25,sd=3) #making another random array of 30 numbers with a mean of 25 and standard deviation of 3  
  
if (runif(1)>0.5) { testSample <- grads } else { testSample <- undergrads }   
mean(testSample) #randomly choose grads or undergrads for test sample

## [1] 25.54158

# 2 The next line of code should generate a list of sample means from the population called “studentPop.” Very similar code to accomplish this appears right in the middle of Chapter 7. How many sample means should you generate? Really you can create any number that you want - hundreds, thousands, whatever - but I suggest for ease of inspection that you generate just 100 means. That is a pretty small number, but it makes it easy to think about percentiles and ranks.

studentPopSamples <- replicate(100,mean(sample(studentPop,size=sampleSize,replace=TRUE)))  
mean(studentPopSamples)

## [1] 20.04577

# 3 Once you have your list of sample means generated from studentPop, the trick is to

compare mean(testSample) to that list of sample means and see where it falls. Is it in the middle of the pack? Far out toward one end? Here is one hint that will help you: In chapter 7, the quantile() command is used to generate percentiles based on thresholds of 2.5% and 97.5%. Those are the thresholds we want, and the quantile() command will help you create them.

quantile (testSample, c(0.025, 0.975))

## 2.5% 97.5%   
## 20.51289 29.68256

quantile (studentPopSamples, c(0.025, 0.975))

## 2.5% 97.5%   
## 19.14567 21.10799

# 4 Your code should end with a print() statement that could say either, “Sample mean is extreme,” or, “Sample mean is not extreme.”

sampleExtreme <- function(testSample,studentPopSamples) {  
 testSampleMean <- mean(testSample)   
 lowerQuantile <- quantile(testSample, probs = .025)  
 upperQuantile <- quantile(testSample, probs = .975)  
 meanStudentPopSample <- mean(studentPopSamples)  
 if (meanStudentPopSample > upperQuantile) {  
 print("Sample mean is extreme")  
 } else if (meanStudentPopSample < lowerQuantile) {  
 print("Sample mean is extreme")  
 } else {  
 print ("Sample mean is not extreme")  
 }  
}

sampleExtreme(testSample, studentPopSamples)

## [1] "Sample mean is extreme"

# Since the sample mean is extreme, we can assume that the sample mean of the test sample is undergrad. -MIC DROP