Alonso\_Week 4 Homework Assignment

Martin Alonso

8/9/2018

IST687 Introduction to Data Science: Week 4 Homework.

Let’s continue our exploration of sampling

### Part 1: The Code

library(moments)  
library(stringr)  
  
# Step 1: Write a summarizing function to understand the distribution of a vector.  
# 1. The function should be called 'printVecInfo' and should take a vector as an input.  
# 2. The function should print the following info: mean, median, min & max, stddev, quantiles (0.05 & 0.95), and skewness  
# 3. Test the function with vector (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 50)  
  
printVecInfo <- function(v) {  
 mean\_v <- mean(v)  
 median\_v <- median(v)  
 min\_v <- min(v)  
 max\_v <- max(v)  
 sd\_v <- sd(v)  
 q05\_v <- quantile(v, 0.05)  
 q95\_v <- quantile(v, 0.95)  
 skew\_v <- skewness(v)  
   
 results <- cat(paste("mean:", mean\_v),   
 paste("\nmedian:", median\_v),   
 paste("\nmin:", min\_v, "max:", max\_v),  
 paste("\nsd:", sd\_v),  
 paste("\nquantile (0.05-0.95):", q05\_v, '-', q95\_v),  
 paste("\nskewness:", skew\_v), '\n')  
   
 return(invisible(results))  
}  
  
#This is the vector that will be used to test the function  
v <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 50)  
  
printVecInfo(v)  
  
#Step 2: Creating Samples in a Jar  
#4. Create a variable 'jar' that has 50 red and 50 blue marbles.  
#5. Confirm there are 50 red samples by summing.  
#6. Sample 10 marbles from 'jar'. How many are red? What was the percentage?  
  
#Create 'jar' using rep() to replicate 'red' and 'blue' strings 50 times each.  
jar <- c(rep('red', 50), rep('blue', 50))  
  
#Confirm there are 50 'red' strings  
sum(str\_detect(jar, 'red'))  
  
#Sampling 10 marbles from jar  
jar\_sample <- sample(jar, 10)  
print(jar\_sample)  
sum(str\_detect(jar\_sample, 'red')) #count number of 'red' marbles  
sum(str\_detect(jar\_sample, 'red'))/length(jar\_sample) #percentage of 'red' marbles  
  
  
#7. Replicate sample drawing 20 times. Then, use printVecInfo to check the distribution, and generate a histogram.   
sample20 <- replicate(20, mean(str\_detect(sample(jar, 10, replace = TRUE2), pattern = 'red')))  
  
printVecInfo(sample20)  
  
hist(sample20)  
  
#8. Repeat exercise 7 but drawing 20 samples 100 times.   
sample100 <- replicate(100, mean(str\_detect(sample(jar, 20, replace = TRUE), pattern = 'red')))  
  
printVecInfo(sample100)  
  
hist(sample100)  
  
#9. Repeat exercise 8 but drawing 100 samples 100 times.  
sample10000 <- replicate(100, mean(str\_detect(sample(jar, 100, replace = TRUE), pattern = 'red')))  
  
printVecInfo(sample10000)  
  
hist(sample10000)  
  
#Step 3: Explore the airquality dataset  
#10. Store the air quality dataset into a temporary variable.  
#11. Clean the dataset by removing NAs.  
#12. Explore the Ozone, Wind, and Temp columns using printVecInfo and create a histogram of each.  
  
#Store the dataset into variable 'aq'  
aq <- airquality  
  
#Remove NAs  
aq <- na.omit(aq)  
  
#Explore the dataset...   
printVecInfo(aq$Ozone)  
printVecInfo(aq$Wind)  
printVecInfo(aq$Temp)  
  
#...and create histograms  
hist(aq$Ozone)  
  
hist(aq$Wind)  
  
hist(aq$Temp)

### Part 2: Running the code

library(moments)  
library(stringr)  
  
# Step 1: Write a summarizing function to understand the distribution of a vector.  
# 1. The function should be called 'printVecInfo' and should take a vector as an input.  
# 2. The function should print the following info: mean, median, min & max, stddev, quantiles (0.05 & #0.95), and skewness  
# 3. Test the function with vector (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 50)  
  
printVecInfo <- function(v) {  
 mean\_v <- mean(v)  
 median\_v <- median(v)  
 min\_v <- min(v)  
 max\_v <- max(v)  
 sd\_v <- sd(v)  
 q05\_v <- quantile(v, 0.05)  
 q95\_v <- quantile(v, 0.95)  
 skew\_v <- skewness(v)  
   
 results <- cat(paste("mean:", mean\_v),   
 paste("\nmedian:", median\_v),   
 paste("\nmin:", min\_v, "max:", max\_v),  
 paste("\nsd:", sd\_v),  
 paste("\nquantile (0.05-0.95):", q05\_v, '-', q95\_v),  
 paste("\nskewness:", skew\_v), '\n')  
   
 return(invisible(results))  
}  
  
#This is the vector that will be used to test the function  
v <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 50)  
  
printVecInfo(v)

## mean: 9.54545454545454   
## median: 6   
## min: 1 max: 50   
## sd: 13.7212509368762   
## quantile (0.05-0.95): 1.5 - 30   
## skewness: 2.62039633563579

#Step 2: Creating Samples in a Jar  
#4. Create a variable 'jar' that has 50 red and 50 blue marbles.  
#5. Confirm there are 50 red samples by summing.  
#6. Sample 10 marbles from 'jar'. How many are red? What was the percentage?  
  
#Create 'jar' using rep() to replicate 'red' and 'blue' strings 50 times each.  
jar <- c(rep('red', 50), rep('blue', 50))  
  
#Confirm there are 50 'red' strings  
sum(str\_detect(jar, 'red'))

## [1] 50

#Sampling 10 marbles from jar  
jar\_sample <- sample(jar, 10)  
print(jar\_sample)

## [1] "red" "blue" "red" "red" "blue" "blue" "blue" "red" "red" "red"

sum(str\_detect(jar\_sample, 'red')) #count number of 'red' marbles

## [1] 6

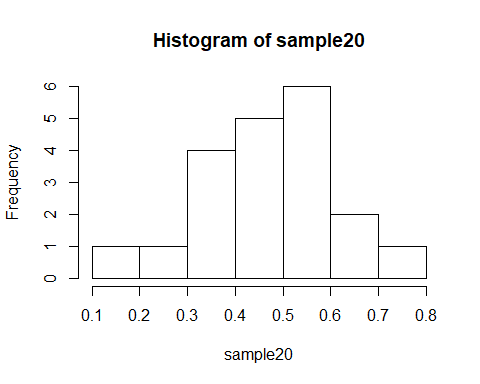
sum(str\_detect(jar\_sample, 'red'))/length(jar\_sample) #percentage of 'red' marbles

## [1] 0.6

#7. Replicate sample drawing 20 times. Then, use printVecInfo to check the distribution, and generate #a histogram.   
sample20 <- replicate(20, mean(str\_detect(sample(jar, 10, replace = TRUE), pattern = 'red')))  
  
printVecInfo(sample20)

## mean: 0.515   
## median: 0.5   
## min: 0.1 max: 0.8   
## sd: 0.156524758424985   
## quantile (0.05-0.95): 0.29 - 0.705   
## skewness: -0.676102283983123

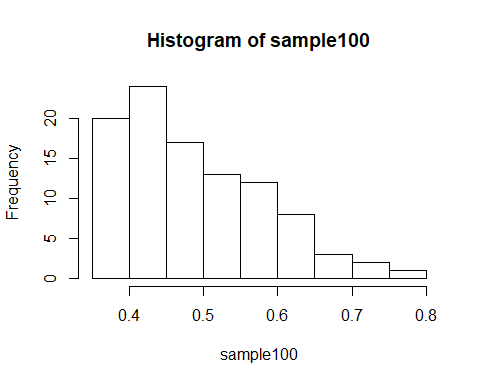
hist(sample20)



#8. Repeat exercise 7 but drawing 20 samples 100 times.   
sample100 <- replicate(100, mean(str\_detect(sample(jar, 20, replace = TRUE), pattern = 'red')))  
  
printVecInfo(sample100)

## mean: 0.508   
## median: 0.5   
## min: 0.35 max: 0.8   
## sd: 0.102178295348885   
## quantile (0.05-0.95): 0.35 - 0.7   
## skewness: 0.510480843572221

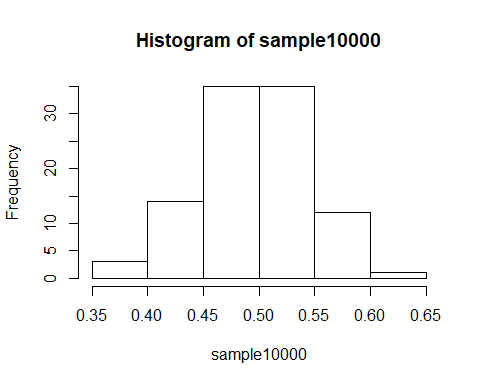
hist(sample100)



#9. Repeat exercise 8 but drawing 100 samples 100 times.  
sample10000 <- replicate(100, mean(str\_detect(sample(jar, 100, replace = TRUE), pattern = 'red')))  
  
printVecInfo(sample10000)

## mean: 0.5011   
## median: 0.5   
## min: 0.38 max: 0.61   
## sd: 0.0477344294778492   
## quantile (0.05-0.95): 0.4295 - 0.5805   
## skewness: -0.0544659970917247

hist(sample10000)



#Step 3: Explore the airquality dataset  
#10. Store the air quality dataset into a temporary variable.  
#11. Clean the dataset by removing NAs.  
#12. Explore the Ozone, Wind, and Temp columns using printVecInfo and create a histogram of each.  
  
#Store the dataset into variable 'aq'  
aq <- airquality  
  
#Remove NAs  
aq <- na.omit(aq)  
  
#Explore the dataset...   
printVecInfo(aq$Ozone)

## mean: 42.0990990990991   
## median: 31   
## min: 1 max: 168   
## sd: 33.2759686574274   
## quantile (0.05-0.95): 8.5 - 109   
## skewness: 1.24810370040404

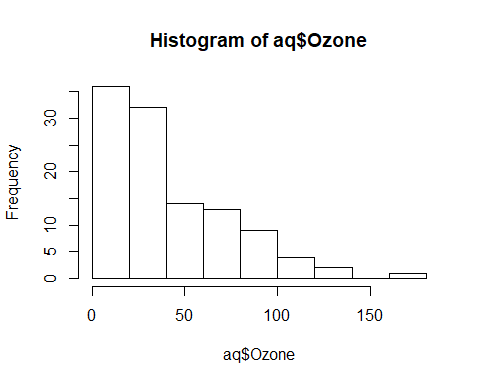
printVecInfo(aq$Wind)

## mean: 9.93963963963964   
## median: 9.7   
## min: 2.3 max: 20.7   
## sd: 3.55771324101922   
## quantile (0.05-0.95): 4.6 - 15.5   
## skewness: 0.455641432036776

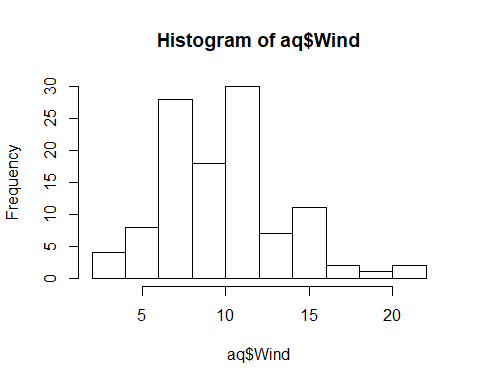
printVecInfo(aq$Temp)

## mean: 77.7927927927928   
## median: 79   
## min: 57 max: 97   
## sd: 9.52996910909533   
## quantile (0.05-0.95): 61 - 92.5   
## skewness: -0.225095889347339

#...and create histograms  
hist(aq$Ozone)



hist(aq$Wind)



hist(aq$Temp)

