CIND 123 - Data Analytics: Basic Methods

Stephanie Boissonneault

Assignment 2 (10%)

[Stephanie Boissonneault]

[D30 & 500616408]

## Instructions

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. Review this website for more details on using R Markdown <http://rmarkdown.rstudio.com>.

Use RStudio for this assignment. Complete the assignment by inserting your R code wherever you see the string “#INSERT YOUR ANSWER HERE”.

When you click the **Knit** button, a document (PDF, Word, or HTML format) will be generated that includes both the assignment content as well as the output of any embedded R code chunks.

Submit **both** the rmd and generated output files. Failing to submit both files will be subject to mark deduction.

## Sample Question and Solution

Use seq() to create the vector .

seq(100, 3, -3)

## [1] 100 97 94 91 88 85 82 79 76 73 70 67 64 61 58 55 52 49 46  
## [20] 43 40 37 34 31 28 25 22 19 16 13 10 7 4

## Question 1 (40 points)

The Titanic Passenger Survival Data Set provides information on the fate of passengers on the fatal maiden voyage of the ocean liner “Titanic.” The dataset is available from the Department of Biostatistics at the Vanderbilt University School of Medicine (<https://biostat.app.vumc.org/wiki/pub/Main/DataSets/titanic3.csv>) in several formats. Store the Titanic Data Set titanic\_train using the following commands.

#install.packages("titanic")  
library(titanic)

## Warning: package 'titanic' was built under R version 4.3.2

titanicDataset <- read.csv(file = "https://biostat.app.vumc.org/wiki/pub/Main/DataSets/titanic3.csv", stringsAsFactors = F)  
str(titanicDataset)

## 'data.frame': 1309 obs. of 14 variables:  
## $ pclass : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ survived : int 1 1 0 0 0 1 1 0 1 0 ...  
## $ name : chr "Allen, Miss. Elisabeth Walton" "Allison, Master. Hudson Trevor" "Allison, Miss. Helen Loraine" "Allison, Mr. Hudson Joshua Creighton" ...  
## $ sex : chr "female" "male" "female" "male" ...  
## $ age : num 29 0.92 2 30 25 48 63 39 53 71 ...  
## $ sibsp : int 0 1 1 1 1 0 1 0 2 0 ...  
## $ parch : int 0 2 2 2 2 0 0 0 0 0 ...  
## $ ticket : chr "24160" "113781" "113781" "113781" ...  
## $ fare : num 211 152 152 152 152 ...  
## $ cabin : chr "B5" "C22 C26" "C22 C26" "C22 C26" ...  
## $ embarked : chr "S" "S" "S" "S" ...  
## $ boat : chr "2" "11" "" "" ...  
## $ body : int NA NA NA 135 NA NA NA NA NA 22 ...  
## $ home.dest: chr "St Louis, MO" "Montreal, PQ / Chesterville, ON" "Montreal, PQ / Chesterville, ON" "Montreal, PQ / Chesterville, ON" ...

1. Extract and show the columns cabin, age, embarked and pclass into a new data frame of the name ‘titanicSubset’. (5 points)

titanicSubset <- data.frame(titanicDataset$cabin, titanicDataset$age, titanicDataset$embarked, titanicDataset$pclass)  
  
print(str(titanicSubset))

## 'data.frame': 1309 obs. of 4 variables:  
## $ titanicDataset.cabin : chr "B5" "C22 C26" "C22 C26" "C22 C26" ...  
## $ titanicDataset.age : num 29 0.92 2 30 25 48 63 39 53 71 ...  
## $ titanicDataset.embarked: chr "S" "S" "S" "S" ...  
## $ titanicDataset.pclass : int 1 1 1 1 1 1 1 1 1 1 ...  
## NULL

1. Numerical data: Use the count() function from the dplyr package to display the total number of passengers that survived or not. (5 points) HINT: To count the occurrences of survived or not in the titanicDataset data frame using the dplyr package, you can use the pipe operator (%>%) to chain operations.

#install.packages('dplyr')  
library(dplyr)

## Warning: package 'dplyr' was built under R version 4.3.2

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

print(titanicDataset %>% count(survived))

## survived n  
## 1 0 809  
## 2 1 500

1. Categorical data: Use count() and group\_by() functions from the dplyr package to calculate the number of passengers by embarked. (5 points) HINT: Use group\_by() first then pipe the result to count() to calculate the number of passengers.

n\_embarked <- group\_by(titanicDataset %>% count(embarked))  
n\_embarked

## # A tibble: 4 × 2  
## embarked n  
## <chr> <int>  
## 1 "" 2  
## 2 "C" 270  
## 3 "Q" 123  
## 4 "S" 914

1. Find the passengers in data frame whose embarked information is an empty character (““), and fill them by the most frequent embarked value. (3 points)

#Most frequent embarked value is "S"  
for (i in 1:length(titanicDataset$embarked)) {  
 if (titanicDataset$embarked[i] == "")   
 {titanicDataset$embarked[i] <- "S"}  
}  
  
print(titanicDataset$embarked)

## [1] "S" "S" "S" "S" "S" "S" "S" "S" "S" "C" "C" "C" "C" "S" "S" "S" "C" "C"  
## [19] "C" "C" "S" "S" "C" "C" "S" "C" "C" "C" "S" "S" "S" "C" "S" "S" "S" "C"  
## [37] "S" "S" "S" "C" "C" "C" "S" "C" "C" "S" "S" "S" "C" "C" "C" "S" "S" "S"  
## [55] "S" "S" "S" "S" "S" "C" "S" "S" "S" "S" "S" "S" "C" "S" "C" "S" "S" "C"  
## [73] "C" "S" "S" "S" "C" "C" "C" "S" "S" "S" "S" "S" "C" "C" "S" "S" "S" "S"  
## [91] "S" "S" "S" "S" "S" "S" "C" "C" "C" "C" "C" "C" "C" "C" "C" "C" "S" "S"  
## [109] "C" "S" "C" "S" "S" "S" "S" "S" "S" "C" "S" "S" "C" "S" "C" "C" "C" "S"  
## [127] "S" "S" "S" "C" "C" "C" "C" "C" "C" "C" "C" "S" "S" "S" "C" "C" "C" "C"  
## [145] "C" "C" "C" "S" "S" "S" "S" "C" "S" "C" "S" "S" "S" "S" "S" "C" "C" "S"  
## [163] "S" "S" "C" "S" "C" "S" "S" "C" "S" "S" "S" "C" "C" "S" "S" "S" "S" "S"  
## [181] "S" "S" "C" "C" "C" "S" "C" "S" "S" "S" "S" "S" "C" "S" "S" "S" "C" "S"  
## [199] "S" "C" "C" "S" "S" "C" "C" "S" "Q" "Q" "Q" "C" "S" "S" "C" "C" "C" "C"  
## [217] "S" "S" "C" "C" "C" "C" "C" "S" "S" "S" "S" "S" "C" "C" "S" "S" "S" "C"  
## [235] "S" "S" "C" "C" "S" "S" "S" "S" "C" "C" "C" "S" "C" "C" "S" "C" "C" "C"  
## [253] "C" "C" "S" "C" "S" "C" "C" "S" "S" "S" "S" "S" "C" "S" "S" "C" "S" "S"  
## [271] "S" "S" "S" "C" "C" "C" "C" "C" "C" "S" "C" "C" "C" "C" "S" "S" "S" "S"  
## [289] "S" "S" "S" "S" "S" "S" "C" "C" "C" "C" "C" "C" "S" "S" "C" "C" "C" "S"  
## [307] "S" "S" "C" "S" "S" "S" "C" "C" "C" "S" "C" "C" "S" "C" "S" "S" "C" "C"  
## [325] "C" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S"  
## [343] "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S"  
## [361] "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S"  
## [379] "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "C" "C" "S" "S" "S"  
## [397] "S" "S" "S" "S" "S" "C" "C" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S"  
## [415] "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S"  
## [433] "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S"  
## [451] "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "C" "S" "S"  
## [469] "S" "Q" "Q" "S" "Q" "S" "S" "S" "S" "Q" "C" "C" "C" "C" "C" "S" "S" "C"  
## [487] "S" "Q" "S" "S" "S" "C" "C" "C" "C" "C" "S" "S" "S" "S" "S" "S" "S" "S"  
## [505] "S" "S" "S" "S" "S" "S" "S" "Q" "C" "C" "S" "S" "S" "S" "S" "S" "C" "S"  
## [523] "S" "S" "C" "S" "C" "S" "S" "S" "S" "C" "S" "S" "S" "S" "S" "S" "C" "C"  
## [541] "S" "S" "S" "S" "S" "S" "S" "C" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S"  
## [559] "S" "S" "S" "S" "Q" "S" "S" "S" "C" "S" "S" "S" "S" "S" "S" "S" "S" "S"  
## [577] "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S"  
## [595] "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "C" "S" "S" "S" "S"  
## [613] "S" "C" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S"  
## [631] "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "C"  
## [649] "C" "S" "C" "C" "S" "C" "C" "S" "S" "C" "C" "C" "C" "S" "C" "S" "C" "C"  
## [667] "C" "Q" "S" "S" "S" "S" "C" "C" "S" "S" "S" "S" "C" "C" "C" "C" "Q" "Q"  
## [685] "Q" "S" "Q" "S" "S" "S" "S" "S" "Q" "Q" "Q" "Q" "S" "S" "S" "S" "S" "S"  
## [703] "Q" "Q" "S" "C" "C" "S" "S" "Q" "Q" "S" "S" "Q" "S" "S" "C" "C" "S" "S"  
## [721] "Q" "S" "S" "Q" "Q" "Q" "Q" "Q" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S"  
## [739] "S" "S" "S" "C" "S" "S" "S" "Q" "Q" "S" "S" "S" "S" "S" "S" "S" "S" "S"  
## [757] "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "Q" "S" "S"  
## [775] "S" "C" "Q" "S" "S" "Q" "S" "C" "Q" "S" "S" "S" "S" "S" "S" "C" "C" "C"  
## [793] "C" "S" "S" "C" "S" "Q" "S" "S" "Q" "Q" "Q" "Q" "Q" "S" "S" "S" "S" "S"  
## [811] "S" "S" "Q" "S" "Q" "S" "C" "S" "Q" "Q" "S" "S" "S" "S" "S" "S" "S" "S"  
## [829] "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "Q" "S" "S" "S" "S"  
## [847] "S" "C" "S" "S" "S" "S" "S" "S" "Q" "C" "Q" "S" "S" "Q" "S" "S" "S" "S"  
## [865] "S" "Q" "S" "S" "S" "S" "S" "Q" "S" "S" "S" "C" "S" "S" "S" "S" "C" "S"  
## [883] "S" "S" "S" "S" "Q" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S"  
## [901] "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "C" "C"  
## [919] "C" "C" "Q" "S" "Q" "Q" "Q" "S" "Q" "C" "C" "Q" "Q" "Q" "S" "S" "S" "S"  
## [937] "S" "S" "S" "S" "C" "C" "C" "S" "S" "S" "S" "S" "Q" "S" "S" "S" "S" "C"  
## [955] "S" "S" "S" "S" "S" "S" "C" "Q" "Q" "S" "S" "S" "S" "S" "S" "S" "S" "Q"  
## [973] "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "Q" "S" "S" "Q" "Q" "S"  
## [991] "S" "C" "Q" "Q" "C" "C" "S" "C" "C" "Q" "Q" "Q" "Q" "Q" "Q" "Q" "Q" "Q"  
## [1009] "Q" "Q" "S" "S" "Q" "S" "S" "S" "Q" "S" "S" "S" "S" "S" "S" "Q" "S" "S"  
## [1027] "S" "S" "Q" "Q" "Q" "S" "Q" "S" "C" "C" "C" "C" "S" "Q" "Q" "S" "Q" "Q"  
## [1045] "Q" "S" "S" "C" "C" "C" "C" "S" "S" "C" "Q" "S" "C" "C" "S" "S" "S" "S"  
## [1063] "S" "S" "S" "S" "C" "S" "S" "Q" "Q" "Q" "Q" "Q" "Q" "S" "Q" "Q" "Q" "S"  
## [1081] "Q" "Q" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "Q" "S" "S"  
## [1099] "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "C" "S" "S" "S" "S" "S" "S"  
## [1117] "S" "S" "S" "S" "S" "C" "C" "C" "Q" "S" "S" "S" "S" "S" "S" "S" "S" "S"  
## [1135] "S" "S" "S" "C" "S" "S" "S" "Q" "Q" "Q" "Q" "Q" "Q" "S" "S" "Q" "S" "S"  
## [1153] "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "Q" "Q" "C" "C" "C" "Q" "S" "S"  
## [1171] "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "C" "C" "C" "S"  
## [1189] "S" "S" "S" "S" "S" "Q" "S" "Q" "S" "S" "Q" "S" "S" "C" "S" "S" "S" "S"  
## [1207] "S" "S" "S" "S" "S" "S" "S" "S" "S" "Q" "Q" "S" "S" "S" "S" "S" "C" "S"  
## [1225] "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "C" "C"  
## [1243] "C" "C" "C" "S" "S" "S" "S" "Q" "S" "S" "S" "C" "S" "C" "C" "C" "C" "S"  
## [1261] "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "S" "C"  
## [1279] "S" "S" "S" "S" "S" "S" "S" "S" "C" "S" "S" "S" "S" "S" "S" "S" "S" "S"  
## [1297] "S" "S" "S" "C" "C" "C" "C" "C" "C" "C" "C" "C" "S"

1. Use the aggregate() function to calculate the ‘survivalCount’ of each embarked and calculate the survival rate of each embarked. Then draw the conclusion on which embarked has the higher survival rate. (5 points)

survivalCount <- aggregate(survived ~ embarked, data=titanicDataset, FUN = sum)  
survivalCount

## embarked survived  
## 1 C 150  
## 2 Q 44  
## 3 S 306

#Calculate the survival rate of each embarked  
survivalLength <- aggregate(survived ~ embarked, data=titanicDataset, FUN = length)  
  
survivalRate <- survivalCount$survived/survivalLength$survived  
  
#Survival rates are:  
cbind(survivalCount$embarked, survivalRate)

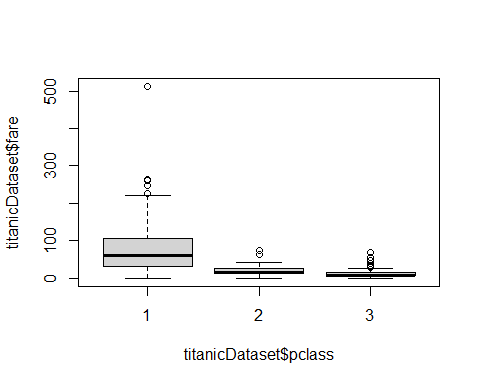
## survivalRate   
## [1,] "C" "0.555555555555556"  
## [2,] "Q" "0.357723577235772"  
## [3,] "S" "0.334061135371179"

print("Embarked C has the highest survival rates")

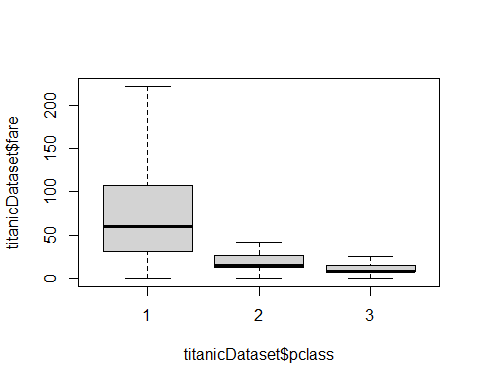
## [1] "Embarked C has the highest survival rates"

1. Use boxplot to display the distribution of fare for each pclass and infer which passenger class is more expensive. (5 points)

#Boxplot wit outliers  
boxplot(titanicDataset$fare ~ titanicDataset$pclass)



#Boxplot without outliers  
boxplot(titanicDataset$fare ~ titanicDataset$pclass, outline = FALSE)



print("The most expensive pclass is class 1")

## [1] "The most expensive pclass is class 1"

1. Calculate the average fare for three pclass and describe if the calculation agrees with the box plot. (5 points)

index\_class1 <- which(titanicDataset$pclass == 1)  
index\_class2 <- which(titanicDataset$pclass == 2)  
index\_class3 <- which(titanicDataset$pclass == 3)  
  
avg\_fare\_class1 <- mean(titanicDataset$fare[index\_class1], na.rm = TRUE)  
avg\_fare\_class2 <- mean(titanicDataset$fare[index\_class2], na.rm = TRUE)  
avg\_fare\_class3 <- mean(titanicDataset$fare[index\_class3], na.rm = TRUE)  
  
avg\_fare\_class1

## [1] 87.50899

avg\_fare\_class2

## [1] 21.1792

avg\_fare\_class3

## [1] 13.30289

print("The calculated average fare for reach class agrees with findings from the boxplot. The average fare of class 1 is the highest, followed by class 2, and lastly class three.")

## [1] "The calculated average fare for reach class agrees with findings from the boxplot. The average fare of class 1 is the highest, followed by class 2, and lastly class three."

1. Use the for loop and if control statements to list the mens’s non-empty home destinations, age 54 or less that embarked from Q (Queenstown), on the Titanic. (7 points)

#Cleaning data by converting empty strings to NA  
cleanData <- titanicDataset  
cleanData[cleanData == ""] <- NA  
  
homeDest <- list()  
for (i in 1:length(cleanData$sex)) {  
 if (cleanData$sex[i] == "male" &   
 !is.na(cleanData$home.dest[i]) &   
 !is.na(cleanData$age[i]) &   
 cleanData$age[i] <= 54 &  
 cleanData$embarked[i] == "Q"){  
 homeDest<- append(homeDest,cleanData$home.dest[i])}  
}   
  
print(homeDest)

## [[1]]  
## [1] "Fond du Lac, WI"  
##   
## [[2]]  
## [1] "Ireland Chicago, IL"  
##   
## [[3]]  
## [1] "Kingwilliamstown, Co Cork, Ireland New York, NY"  
##   
## [[4]]  
## [1] "Co Cork, Ireland Charlestown, MA"  
##   
## [[5]]  
## [1] "Ireland Philadelphia, PA"  
##   
## [[6]]  
## [1] "Ireland New York, NY"  
##   
## [[7]]  
## [1] "Co Limerick, Ireland Sherbrooke, PQ"  
##   
## [[8]]  
## [1] "Philadelphia, PA"  
##   
## [[9]]  
## [1] "Ireland Brooklyn, NY"  
##   
## [[10]]  
## [1] "Co Athlone, Ireland New York, NY"  
##   
## [[11]]  
## [1] "Ireland New York, NY"  
##   
## [[12]]  
## [1] "Aughnacliff, Co Longford, Ireland New York, NY"  
##   
## [[13]]  
## [1] "Ireland Chicago, IL"  
##   
## [[14]]  
## [1] "New York, NY"

## Question 2 (15 points)

80 engines work together in a sequence. The historical data shows that the probability of each engine’s failure is 0.05. We know that if one engine fails, the whole system stops.

1. What is the probability that the system operates without failure? (5 points)

# If the probability of each engine's failure is 0.05, the probability of each engine working (not failing) is 0.95.  
  
#For the system to work, all engines must work, therefore looking for intersection of all 80 engines working  
  
#When A and B are independent A∩B = (A)(B), therefore multiply the probability of success of each of the 80 engines working to determine success rate of whole system:  
  
prob\_all\_engines\_work<- 0.95^80  
prob\_all\_engines\_work

## [1] 0.01651537

1. Use the Binomial approximation to calculate the probability that at least 3 engines are defective? (5 points)

1 - pbinom( q= 2, size = 80, prob = 0.05)

## [1] 0.7693795

1. What is the probability that the second engine (B) is defective given the first engine (A) is not defective, i.e., P(B is defective|A is not defective), while we know that the first and second engines are independent. (5 points)

#P(B is defective|A is not defective)  
#When A and B are independent, P(B|A) = P(B)   
#Therefore P(B is defective) = 0.05  
  
print("P(B is defective|A is not defective) = 0.05")

## [1] "P(B is defective|A is not defective) = 0.05"

## Question 3 (25 points)

On average, John visits his parents 4 times a month

1. Find the probabilities that John visits his parents 1 to 6 times in a month? (5 points)

#probabilities John visits P(1), P(2), P(3), P(4), P(5) and P(6) times in a month   
dpois(1:6, 4)

## [1] 0.07326256 0.14652511 0.19536681 0.19536681 0.15629345 0.10419563

1. Find the probability that John visits his parents 3 times or more in a month? (5 points)

ppois(2, 4, lower = FALSE)

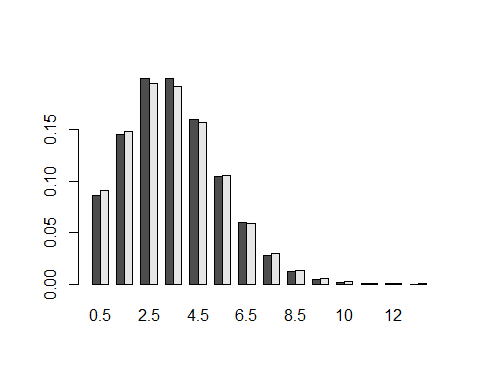
## [1] 0.7618967

1. Compare the similarity between Binomial and Poisson distribution. (15 points @ 5 point each)
2. Create 100,000 samples for a Binomial random variable using parameters described in Question 2
3. Create 100,000 samples for a Poisson random variable using parameters described in Question 3
4. then illustrate on how well the Poisson probability distribution approximates the Binomial probability distribution. HINT: use multhist() from the ‘plotrix’ package

#1) Parameters in Question 2 (Size = 80, Prob of failure = 0.05)  
  
binomSample <- rbinom(100000, 80, 0.05)  
#binomSample  
  
  
#2) Parameters in Question 3 (lambda = 4)  
  
poisSample <- rpois(100000, 4)  
#poisSample  
  
#3) then illustrate on how well the Poisson probability distribution approximates the Binomial probability distribution. HINT: use multhist() from the 'plotrix' package  
  
#install.packages("plotrix")  
library("plotrix")

## Warning: package 'plotrix' was built under R version 4.3.2

binom\_and\_pois\_hist <-multhist(list(binomSample, poisSample), freq = F)



binom\_and\_pois\_hist

## [[1]]  
## $breaks  
## [1] 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14  
##   
## $counts  
## [1] 17676 29363 39491 39153 31702 21014 11847 5756 2508 976 374 104  
## [13] 29 7  
##   
## $density  
## [1] 0.088380 0.146815 0.197455 0.195765 0.158510 0.105070 0.059235 0.028780  
## [9] 0.012540 0.004880 0.001870 0.000520 0.000145 0.000035  
##   
## $mids  
## [1] 0.5 1.5 2.5 3.5 4.5 5.5 6.5 7.5 8.5 9.5 10.5 11.5 12.5 13.5  
##   
## $xname  
## [1] "unlist(x)"  
##   
## $equidist  
## [1] TRUE  
##   
## attr(,"class")  
## [1] "histogram"  
##   
## [[2]]  
## [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]  
## [1,] 0.08561 0.14495 0.19963 0.19937 0.15953 0.10442 0.06008 0.02812 0.01186  
## [2,] 0.09115 0.14868 0.19528 0.19216 0.15749 0.10572 0.05839 0.02944 0.01322  
## [,10] [,11] [,12] [,13] [,14]  
## [1,] 0.00442 0.00148 0.00046 0.00006 1e-05  
## [2,] 0.00534 0.00226 0.00058 0.00023 6e-05

print("As illustrated in the histrogram, the Poisson probability distribution approximates the Binomial probability distribution.")

## [1] "As illustrated in the histrogram, the Poisson probability distribution approximates the Binomial probability distribution."

## Question 4 (20 points)

Write a script in R to compute the following probabilities of a normal random variable with mean 9 and variance 25

1. The probability that it lies between 8.2 and 17.3 (inclusive) (5 points)

#mean = 9  
#variance = 25  
#standard deviation = 5  
my\_sd = sqrt(25)  
my\_sd

## [1] 5

prob\_q4a <- pnorm(17.3, mean = 9, sd = 5) - pnorm(8.2, mean = 9, sd = 5)  
prob\_q4a

## [1] 0.5151022

1. The probability that it is greater than 15.02 (5 points)

prob\_q4b <- pnorm(15.02, mean = 9, sd = 5, lower.tail = FALSE)  
prob\_q4b

## [1] 0.1142948

1. The probability that it is less than or equal to 11.8 (5 points)

prob\_q4c <- pnorm(11.8, mean = 9, sd = 5, lower.tail = TRUE)  
prob\_q4c

## [1] 0.7122603

1. The probability that it is less than 10 or greater than 13 (5 points)

prob\_q4d <- pnorm(10, mean = 9, sd = 5, lower.tail = TRUE) + pnorm(13, mean = 9, sd = 5, lower.tail = FALSE)  
prob\_q4d

## [1] 0.7911151

END of Assignment #2.