

**Submitted By:**

Name – Shreeya Chatterji

Roll number -102103447

Batch - 3COE16

**Submitted To:**

Dr. Rajanish Rai

July 2022 – December 2022

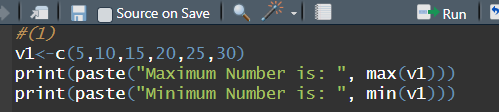
**Probability and Statistics (UCS410)**

**Experiment 1: Basics of R programming**

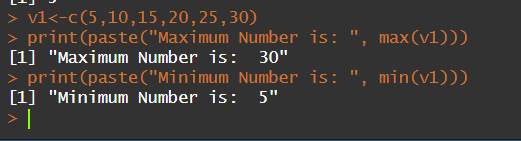
**(1) Create a vector c = [5,10,15,20,25,30] and write a program which returns the max-**

**imum and minimum of this vector.**

CODE:



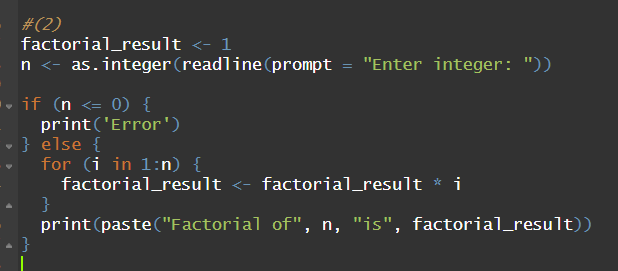
OUTPUT:



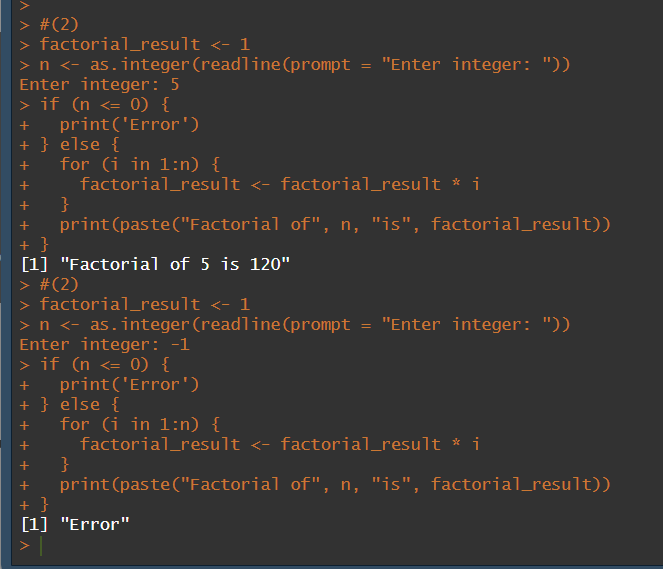
**(2) Write a program in R to find factorial of a number by taking input from user. Please**

**print error message if the input number is negative.**

CODE:



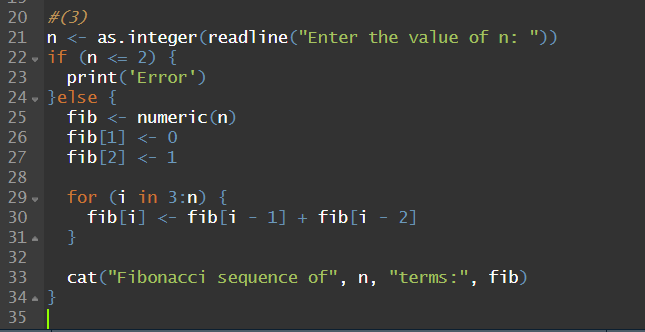
OUTPUT:



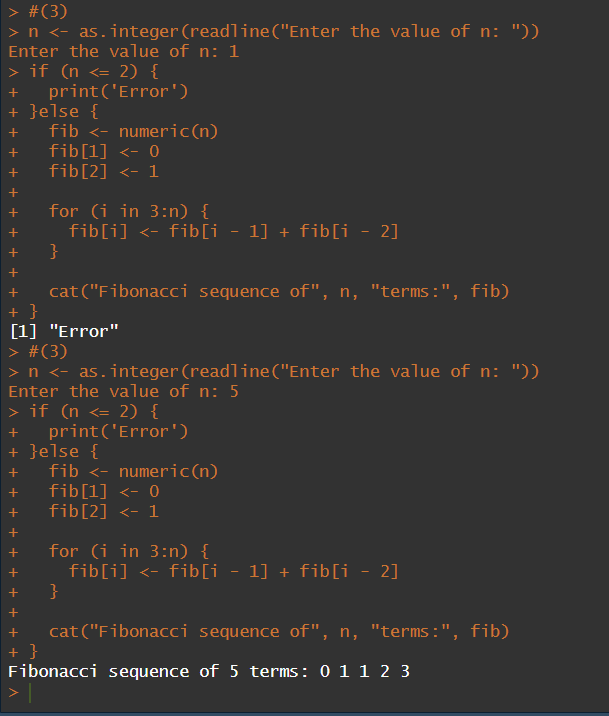
**(3) Write a program to write first n terms of a Fibonacci sequence. You may take n as an**

**input from the user.**

**CODE:**

****

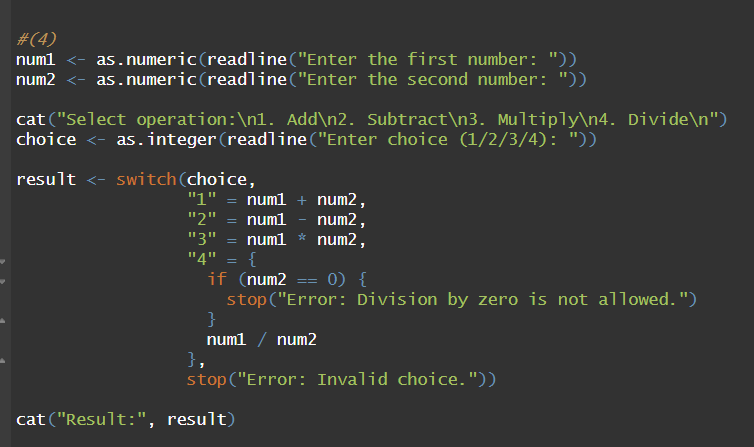
**OUTPUT:**

****

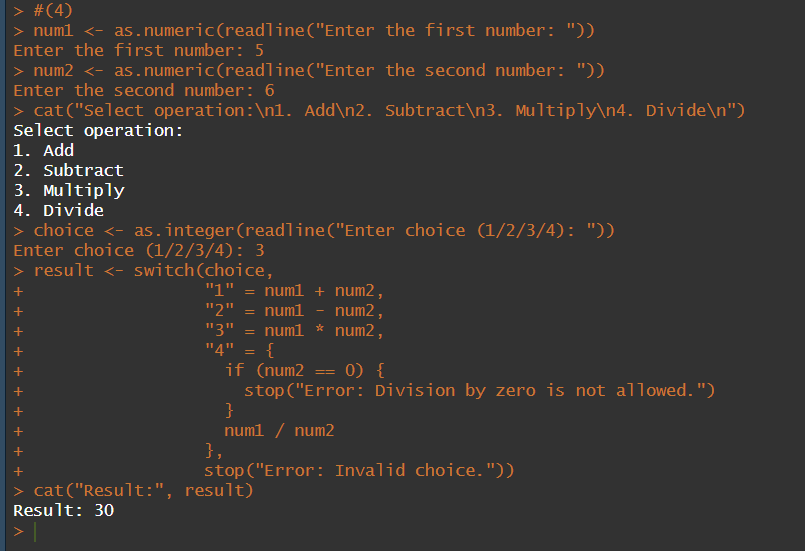
**(4) Write an R program to make a simple calculator which can add, subtract, multiply**

**and divide.**

**CODE:**

****

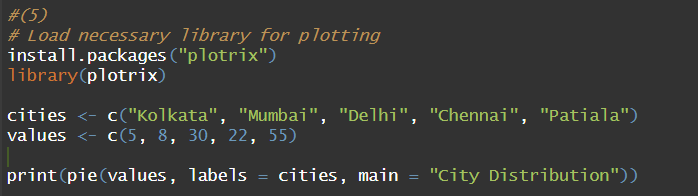
**OUTPUT:**

****

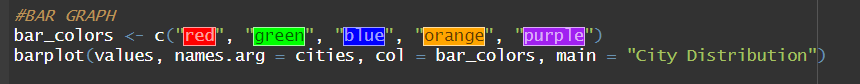
**(5) Explore plot, pie, barplot etc. (the plotting options) which are built-in functions in R.**

**CODE:**

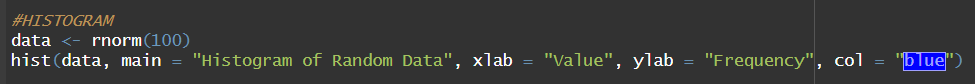
**PIE CHART:**

****

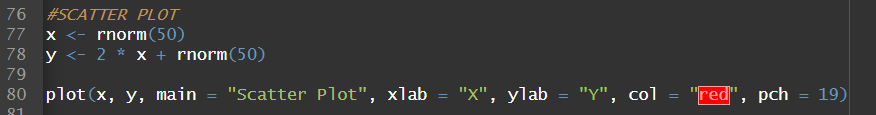
**BAR GRAPH:**

****

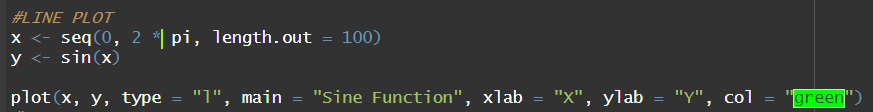
**HISTOGRAM:**

****

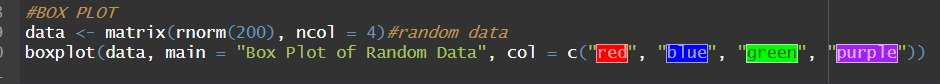
**SCATTER PLOT:**

****

**LINE PLOT:**

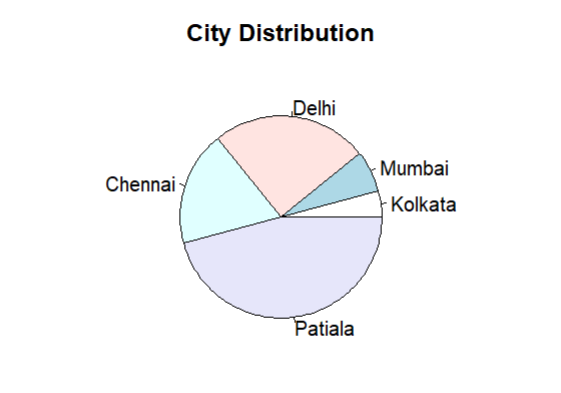
****

**BOX PLOT:**

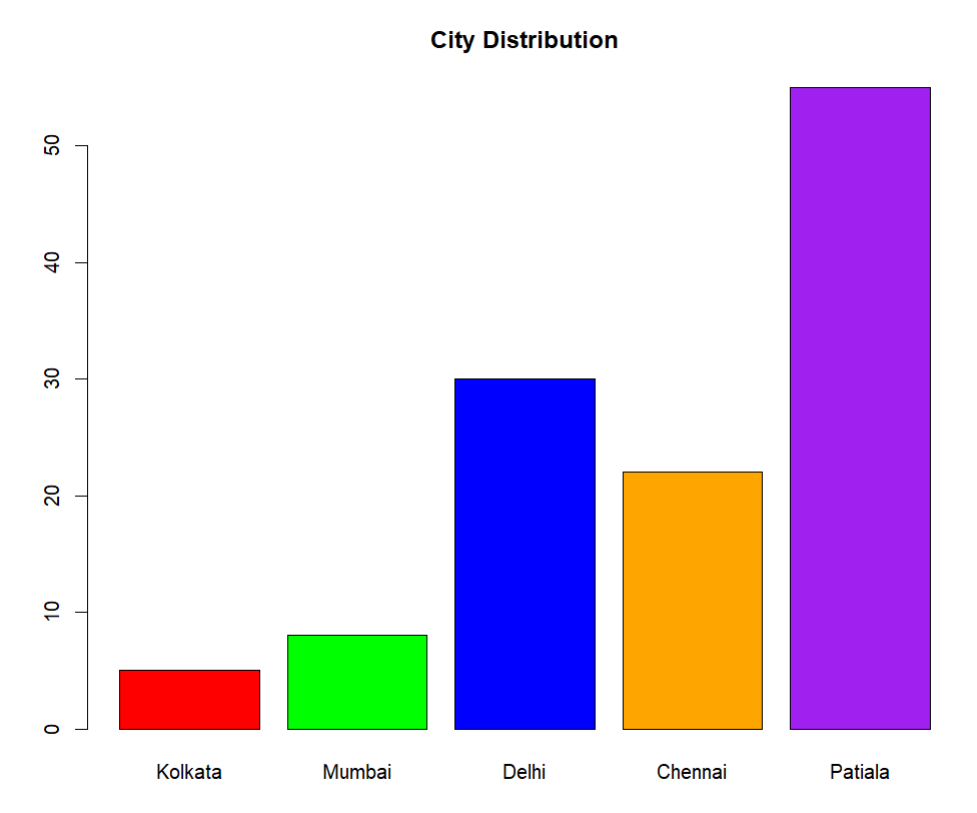
****

**OUTPUT:**

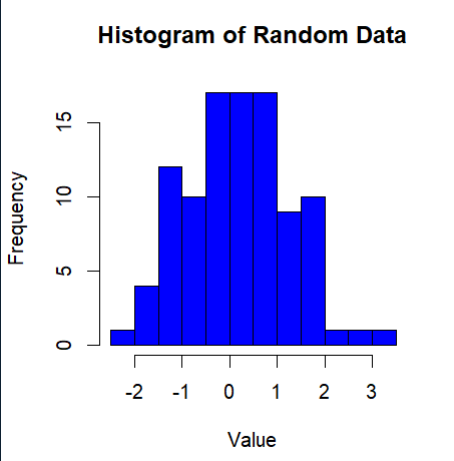
**PIE CHART:**

****

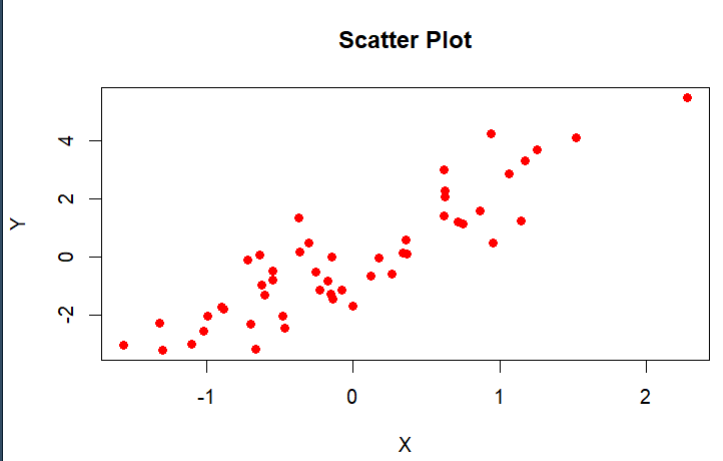
**BAR GRAPH:**

****

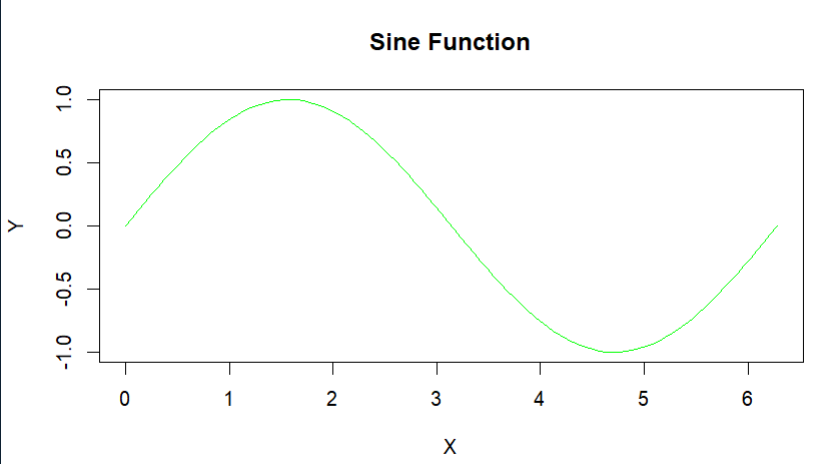
**HISTOGRAM:**

****

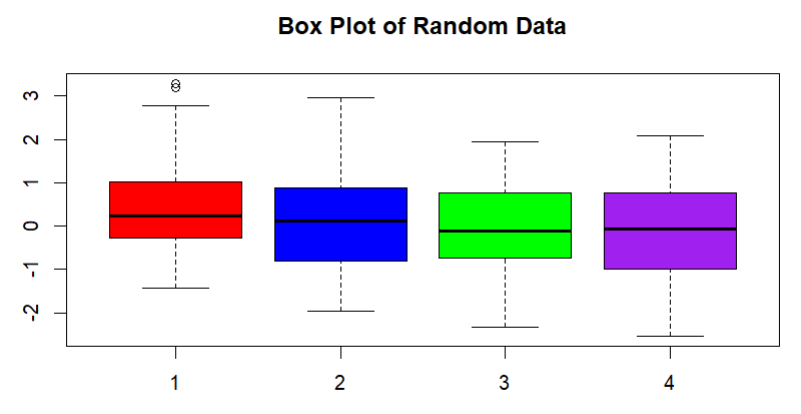
**SCATTER PLOT:**

****

**LINE PLOT:**

****

**BOX PLOT:**

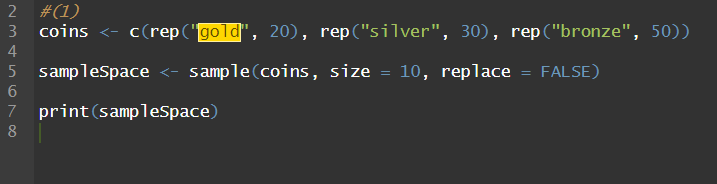
****

**Probability and Statistics (UCS410)**

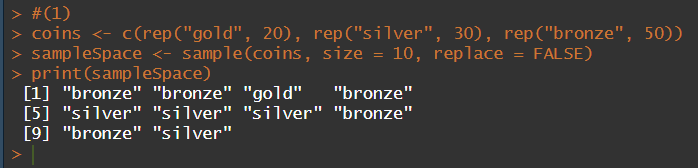
**Experiment 2: Descriptive statistics, Sample space, definition of Probability**

**(1) (a) Suppose there is a chest of coins with 20 gold, 30 silver and 50 bronze coins. You randomly draw 10 coins from this chest. Write an R code which will give us the sample space for this experiment. (use of sample(): an in-built function in R)**

**CODE:**

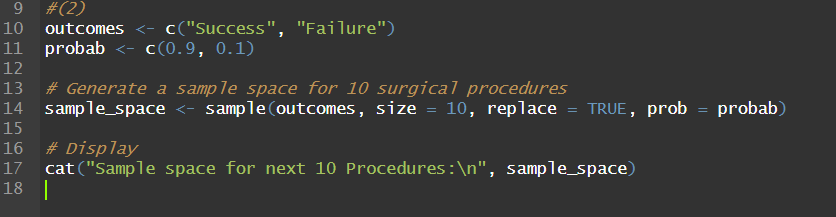
****

**OUTPUT:**

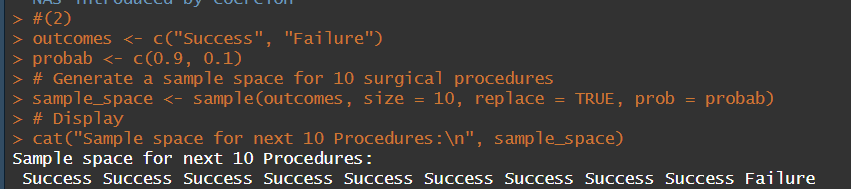
****

**(b) In a surgical procedure, the chances of success and failure are 90% and 10% respectively. Generate a sample space for the next 10 surgical procedures performed. (use of prob(): an in-built function in R)**

**CODE:**

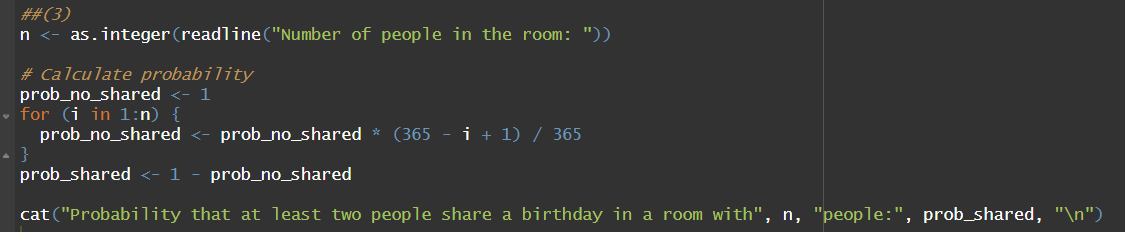
****

**OUTPUT:**

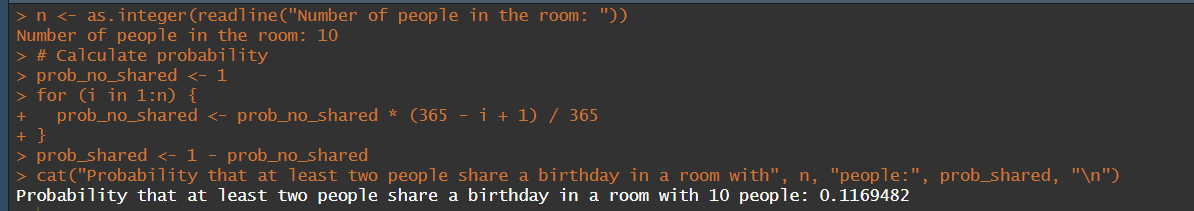
****

**(2) A room has n people, and each has an equal chance of being born on any of the 365 days of the year. (For simplicity, we’ll ignore leap years). What is the probability that two people in the room have the same birthday?**

**CODE:**

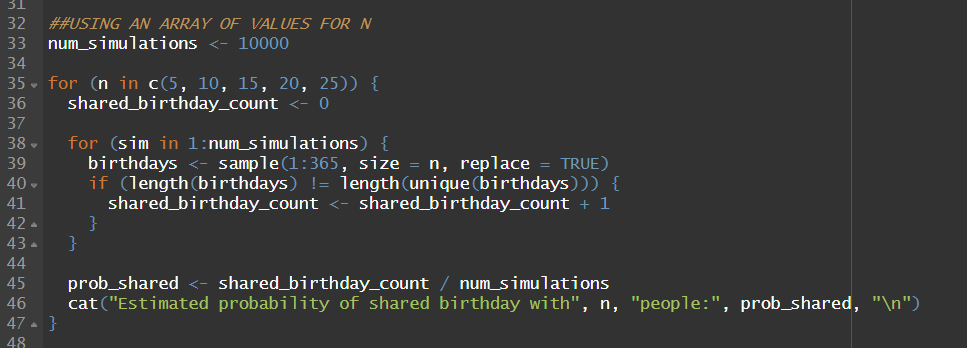
****

**OUTPUT:**

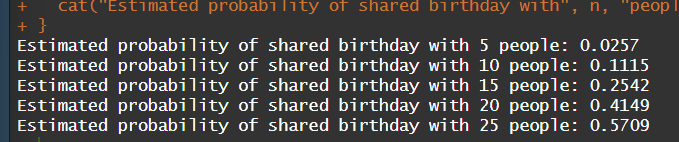
****

**(a) Use an R simulation to estimate this for various n.**

**CODE:**

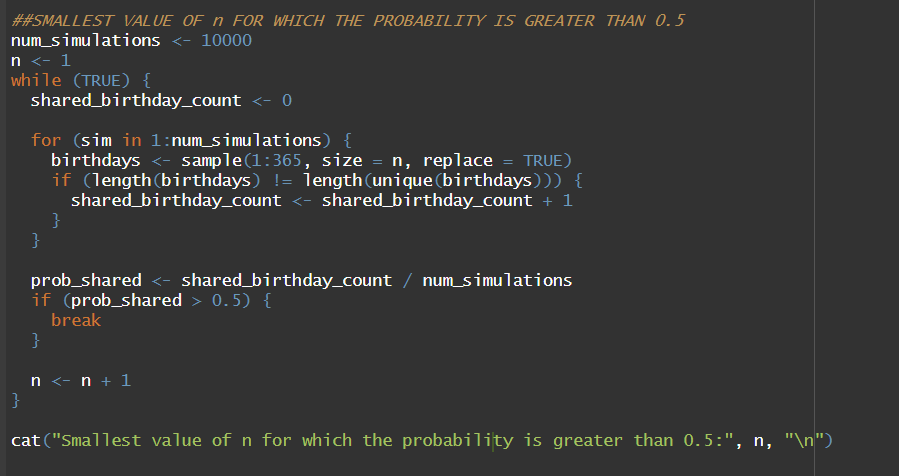
****

**OUTPUT:**

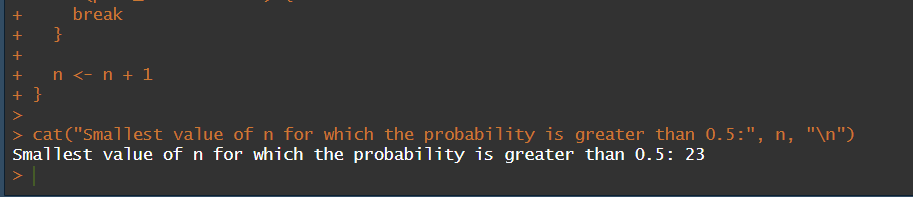
****

**(b) Find the smallest value of n for which the probability of a match is greater than .5.**

**CODE:**

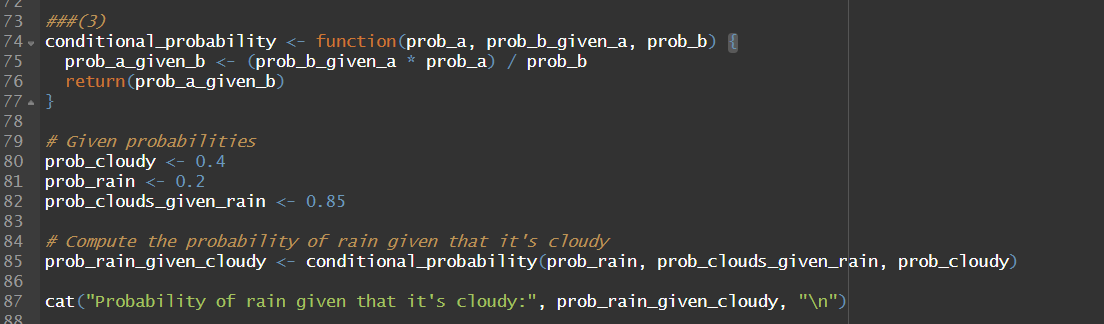
****

**OUTPUT:**

****

**(3) Write an R function for computing conditional probability. Call this function to do the following problem: Suppose the probability of the weather being cloudy is 40%. Also suppose the probability of rain on a given day is 20% and that the probability of clouds on a rainy day is 85%. If it’s cloudy outside on a given day, what is the probability that it will rain that day?**

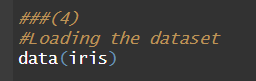
**CODE:**

****

**OUTPUT:**

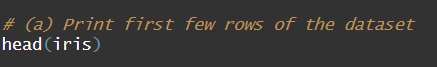
****

**(4) The iris dataset is a built-in dataset in R that contains measurements on 4 different attributes (in centimeters) for 150 flowers from 3 different species. Load this dataset and do the following:**

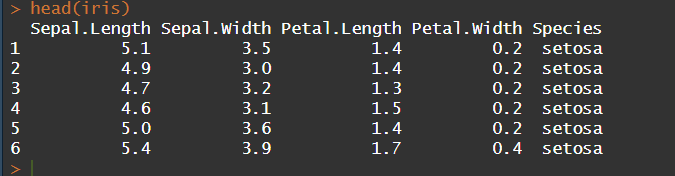
****

**(a) Print first few rows of this dataset.**

**CODE:**

****

**OUTPUT:**

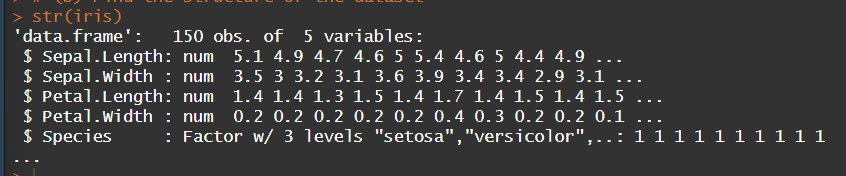


**(b) Find the structure of this dataset.**

**CODE:**

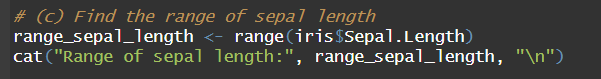
****

**OUTPUT:**

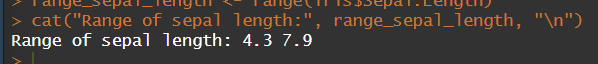
****

**(c) Find the range of the data regarding the sepal length of flowers.**

**CODE:**

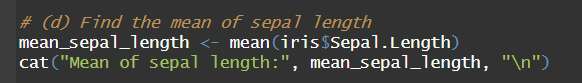
****

**OUTPUT:**

****

**(d) Find the mean of the sepal length.**

**CODE:**

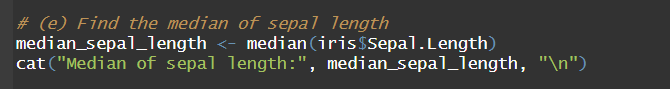
****

**OUTPUT:**

****

**(e) Find the median of the sepal length.**

**CODE:**

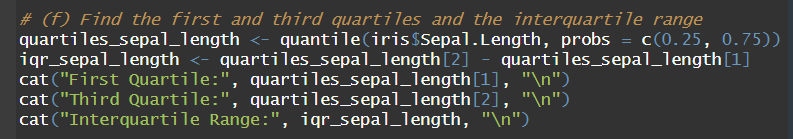
****

**OUTPUT:**

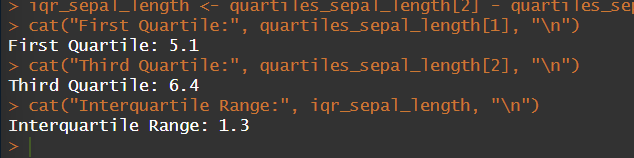
****

**(f) Find the first and the third quartiles and hence the interquartile range.**

**CODE:**

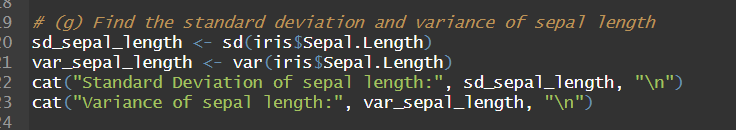
****

**OUTPUT:**

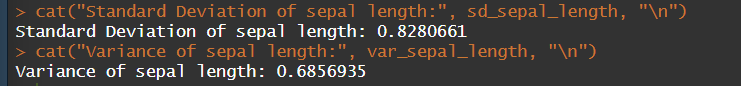
****

**(g) Find the standard deviation and variance.**

**CODE:**

****

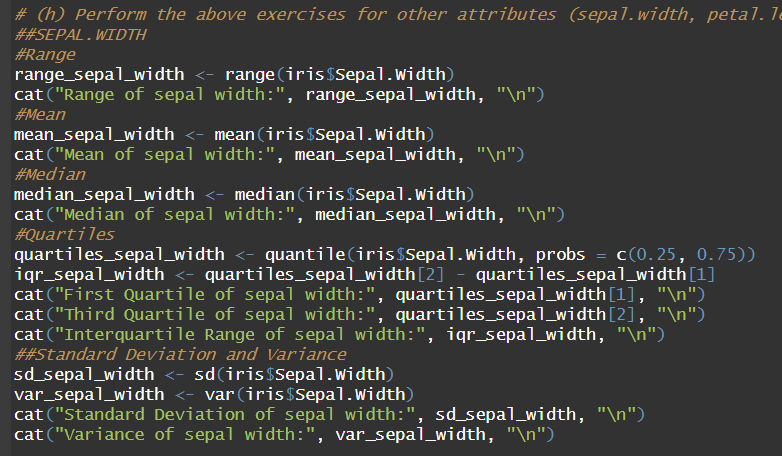
**OUTPUT:**

****

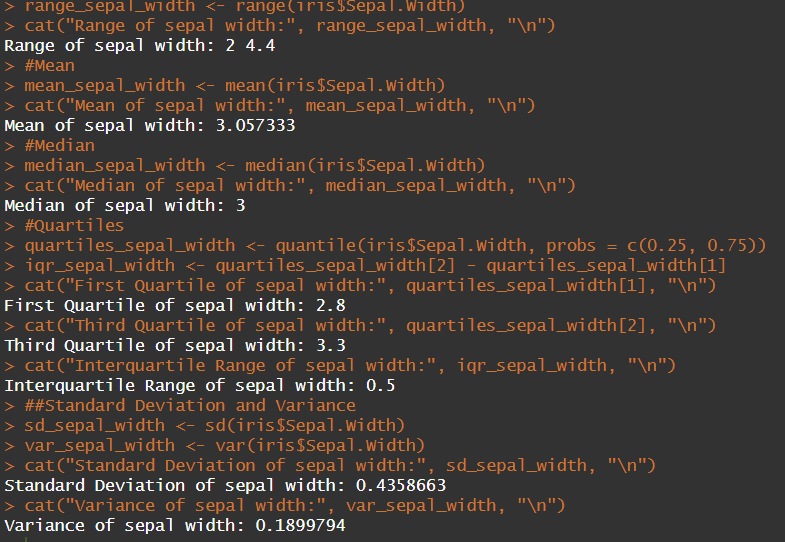
**(h) Try doing the above exercises for sepal.width, petal.length and petal.width.**

**SEPAL.WIDTH**

**CODE:**

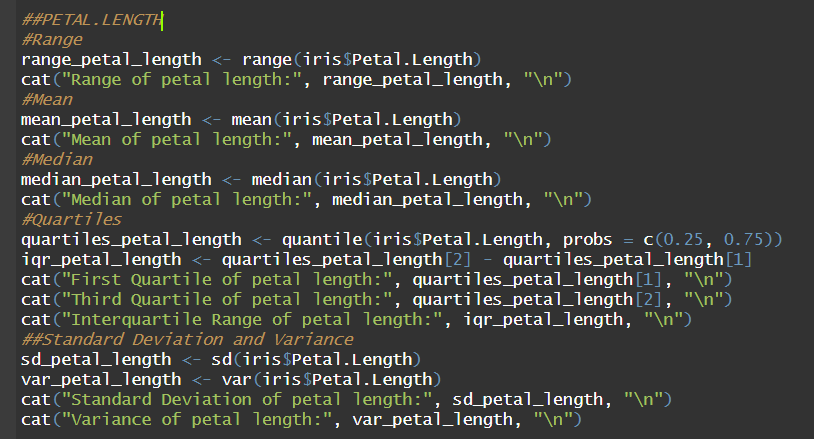
****

**OUTPUT:**

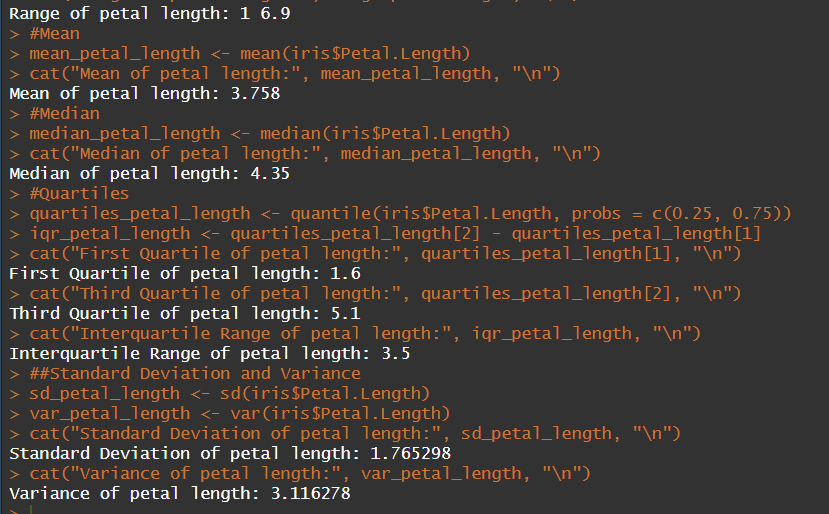
****

**PETAL.LENGTH**

**INPUT:**

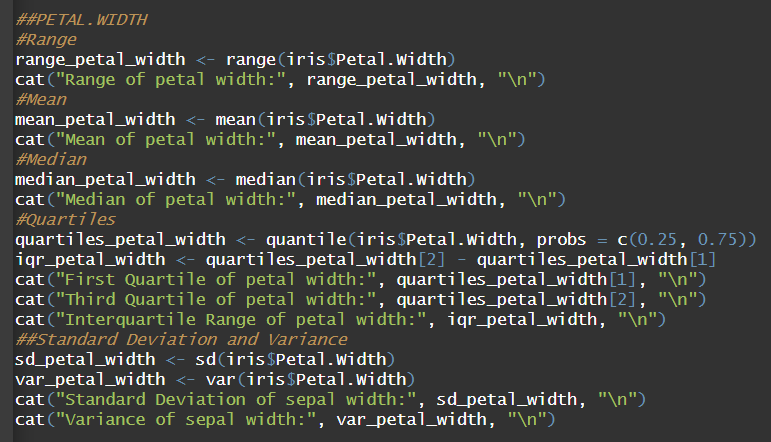
****

**OUTPUT:**

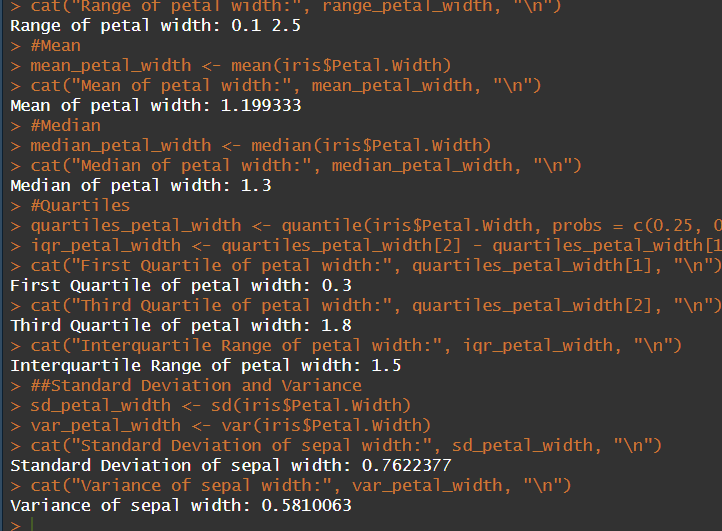
****

**PETAL.WIDTH**

**INPUT:**

****

**OUTPUT:**

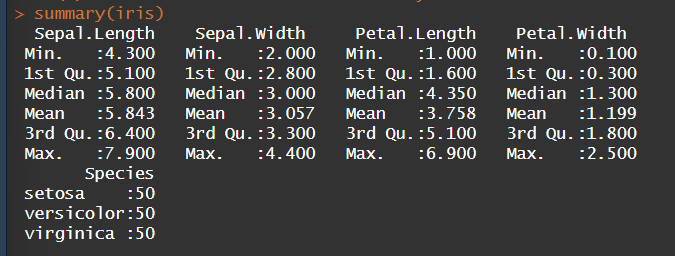
****

**(i) Use the built-in function summary on the dataset Iris.**

**CODE:**

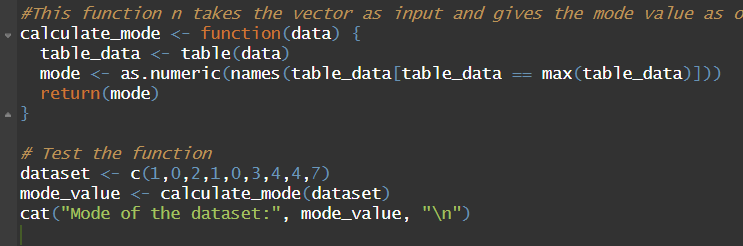
****

**OUTPUT:**

****

**(5) R does not have a standard in-built function to calculate mode. So, we create a user function to calculate mode of a data set in R. This function n takes the vector as input and gives the mode value as output.**

**CODE:**

****

**OUTPUT:**

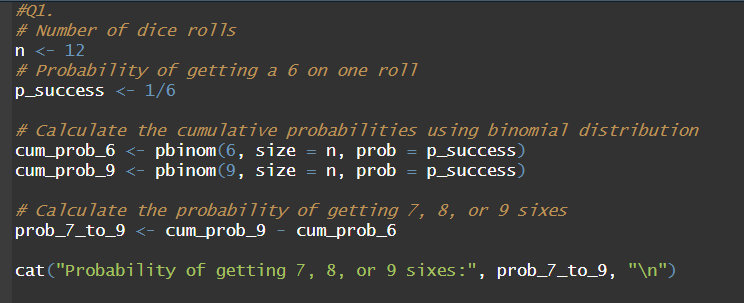
****

**Probability and Statistics (UCS410)**

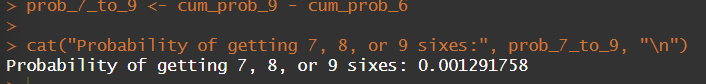
**Experiment 3: Probability distributions**

**(1) Roll 12 dice simultaneously, and let X denotes the number of 6’s that appear. Calculate the probability of getting 7, 8 or 9, 6’s using R. (Try using the function pbinom; If we set S = {get a 6 on one roll}, P(S) = 1/6 and the rolls constitute Bernoulli trials; thus X ∼ binom(size=12, prob=1/6) and we are looking for P(7 ≤ X ≤ 9).**

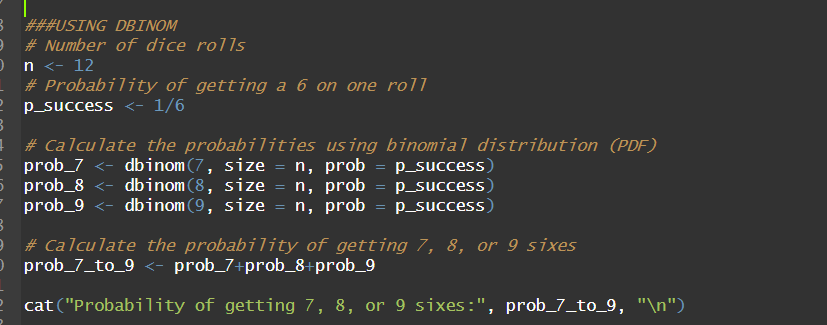
**CODE:**

****

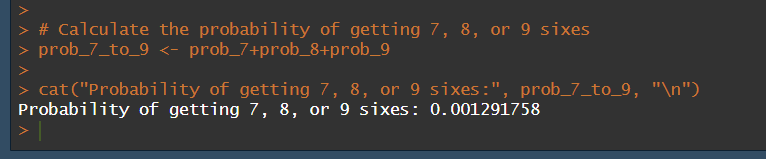
**OUTPUT:**

****

**CODE(USING dbinom):**

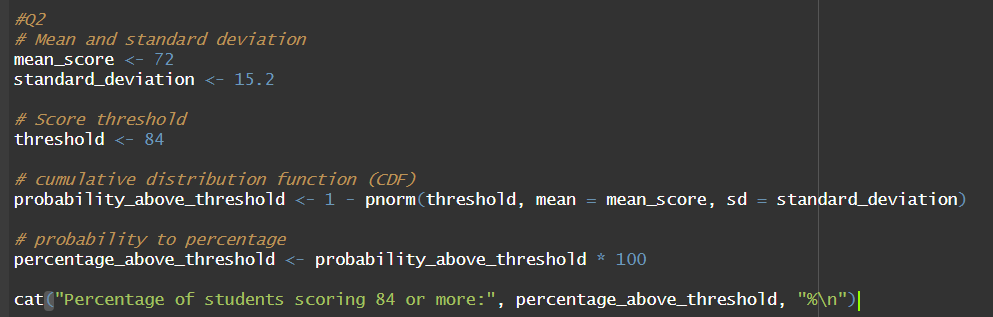
****

**OUTPUT:**

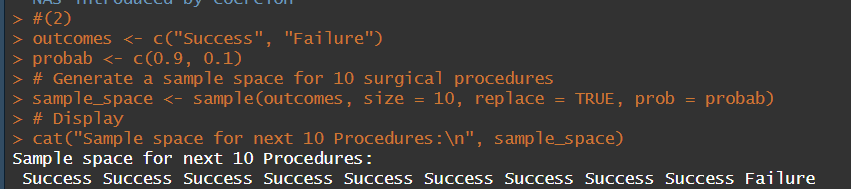
****

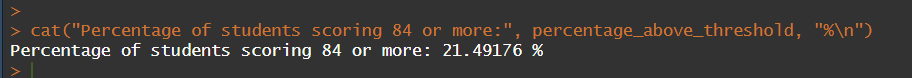
**(2) Assume that the test scores of a college entrance exam fits a normal distribution. Furthermore, the mean test score is 72, and the standard deviation is 15.2. What is the percentage of students scoring 84 or more in the exam?**

**CODE:**

****

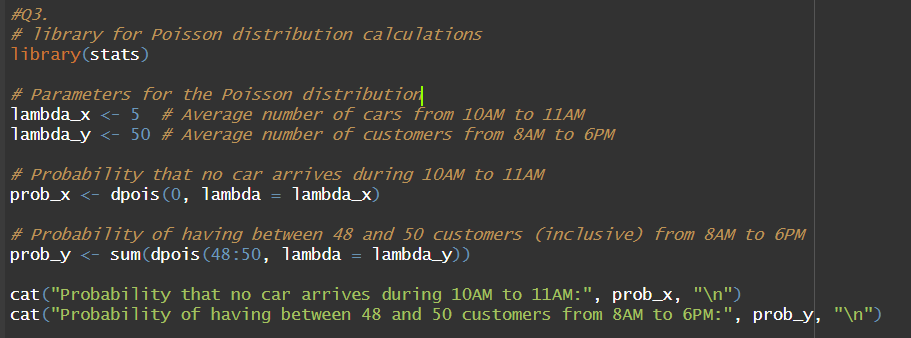
**OUTPUT:**

****

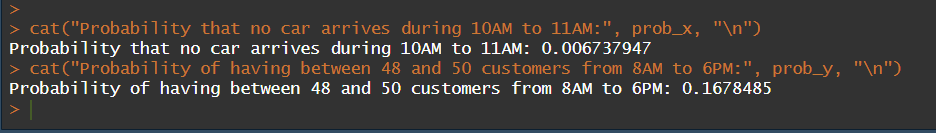
****

**(3) On the average, five cars arrive at a particular car wash every hour. Let X count the number of cars that arrive from 10AM to 11AM, then X ∼Poisson(λ = 5). What is probability that no car arrives during this time. Next, suppose the car wash above is in operation from 8AM to 6PM, and we let Y be the number of customers that appear in this period. Since this period covers a total of 10 hours, we get that Y ∼ Poisson(λ = 5×10 = 50). What is the probability that there are between 48 and 50 customers, inclusive?**

**CODE:**

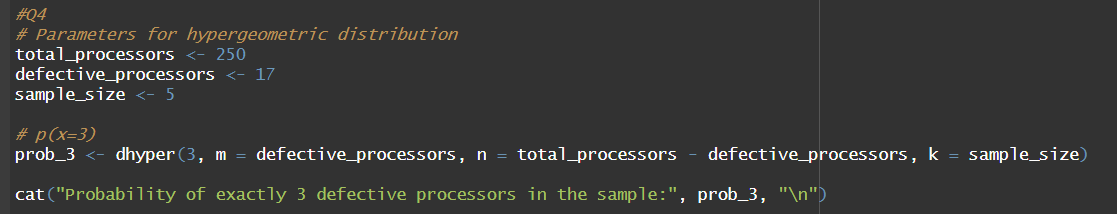
****

**OUTPUT:**

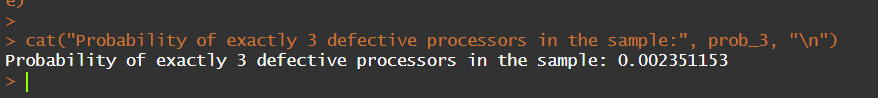
****

**(4) Suppose in a certain shipment of 250 Pentium processors there are 17 defective processors. A quality control consultant randomly collects 5 processors for inspection to determine whether or not they are defective. Let X denote the number of defectives in the sample. Find the probability of exactly 3 defectives in the sample, that is, find P(X = 3).**

**CODE:**

****

**OUTPUT:**

****

**(5) A recent national study showed that approximately 44.7% of college students have used Wikipedia as a source in at least one of their term papers. Let X equal the number of students in a random sample of size n = 31 who have used Wikipedia as a source.**

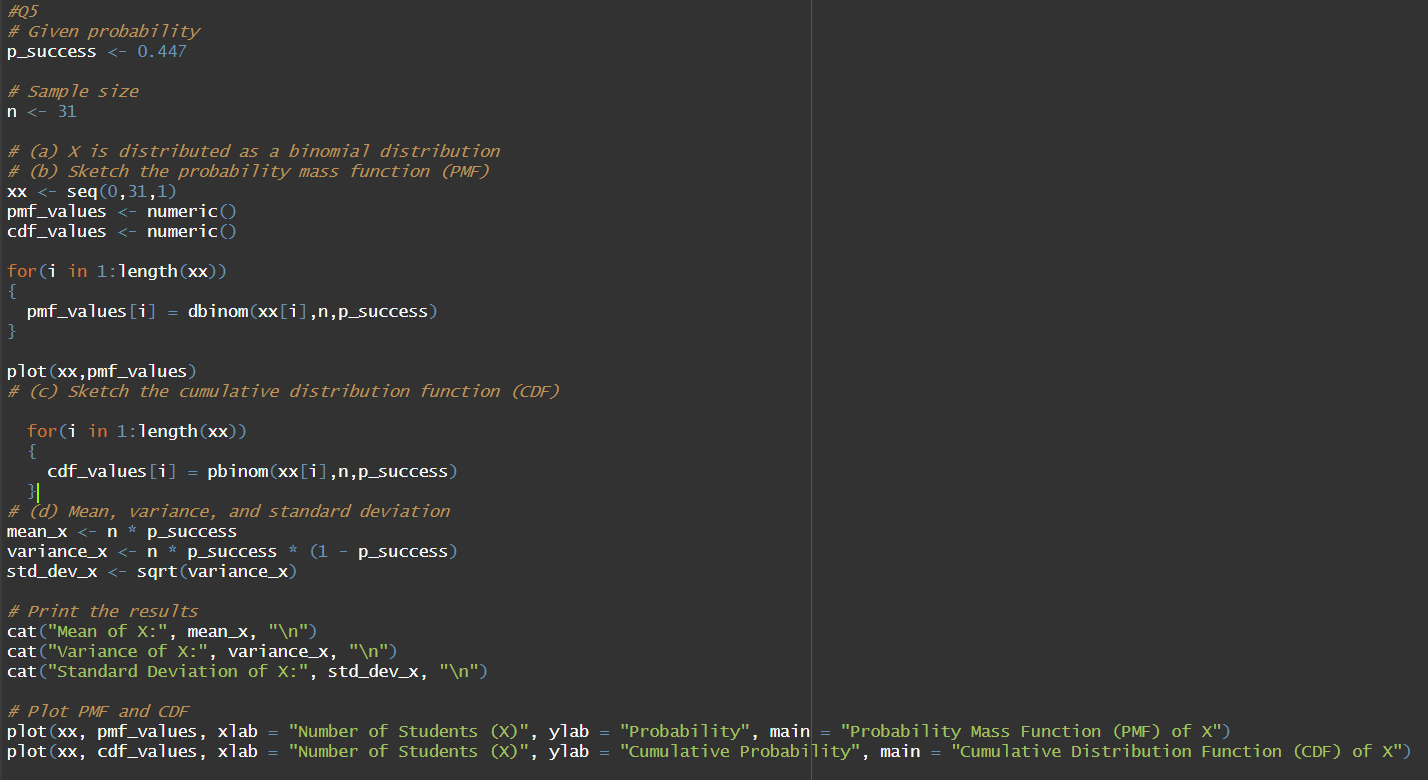
**(a) How is X distributed?**

**(b) Sketch the probability mass function.**

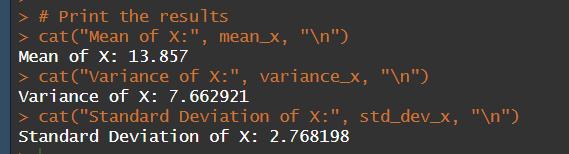
**(c) Sketch the cumulative distribution function.**

**(d) Find mean, variance and standard deviation of X.**

**CODE:**

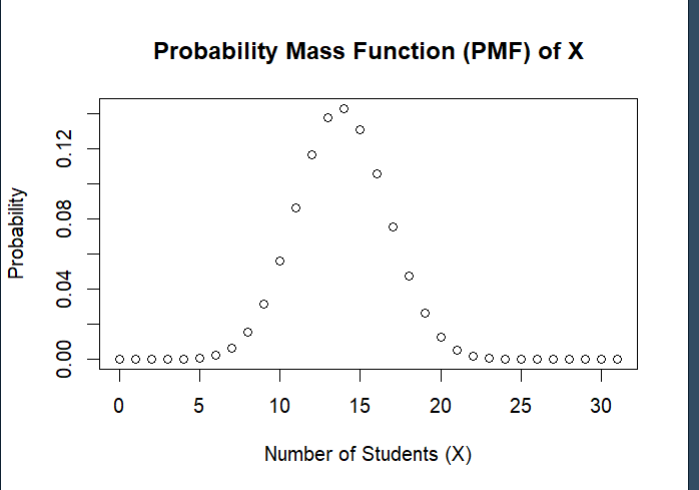
****

**OUTPUT:**

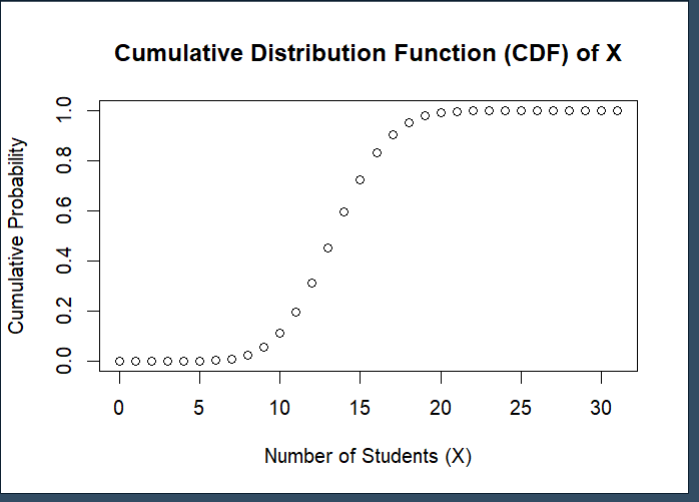
****

**PLOTS:**

*PDF: plot(xx, pmf\_values, xlab = "Number of Students (X)", ylab = "Probability", main = "Probability Mass Function (PMF) of X")*

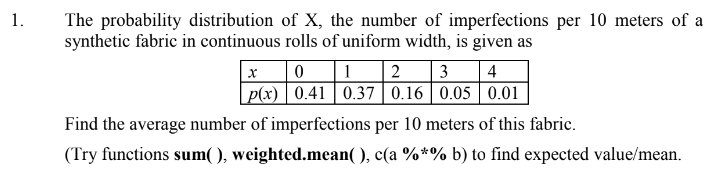


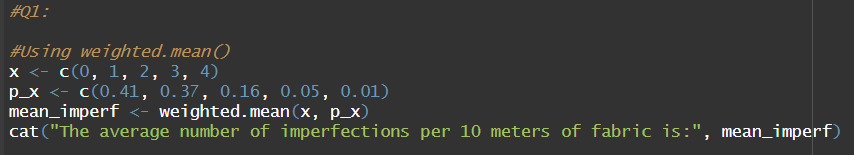
*CDF: plot(xx, cdf\_values, xlab = "Number of Students (X)", ylab = "Cumulative Probability", main = "Cumulative Distribution Function (CDF) of X")*



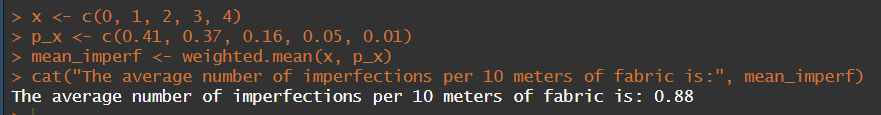
**Probability and Statistics (UCS410)**

**Experiment 4: Mathematical Expectation, Moments and Functions of Random Variables**

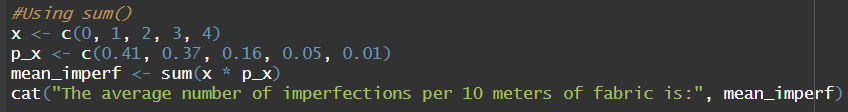
**CODE (using weighted.mean( )):**

****

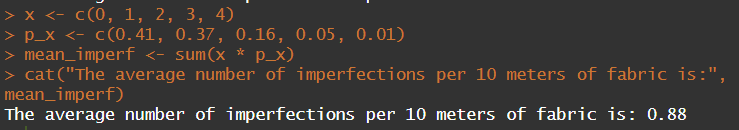
**OUTPUT:**

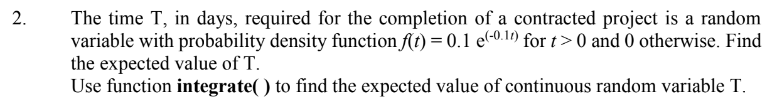
****

**CODE(sum()):**

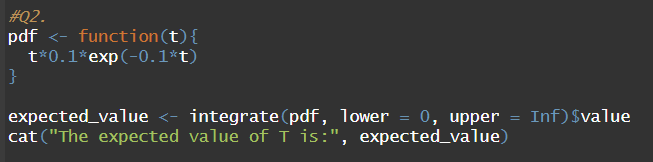
****

**OUTPUT:**

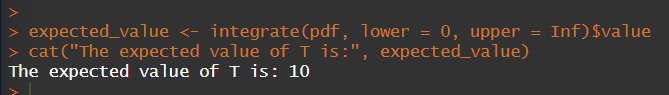
****

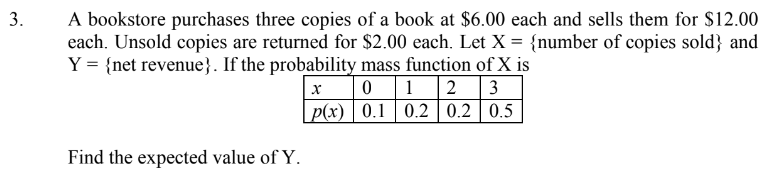
****

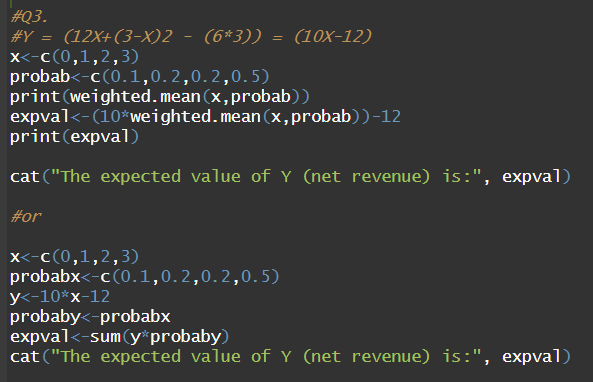
**CODE:**

****

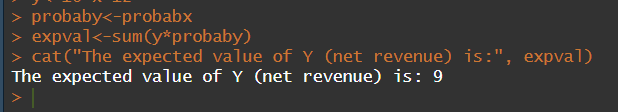
**OUTPUT:**

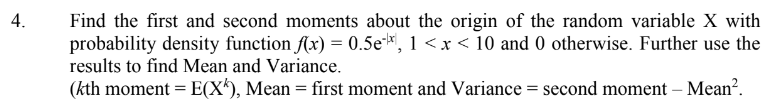
****

**CODE:**

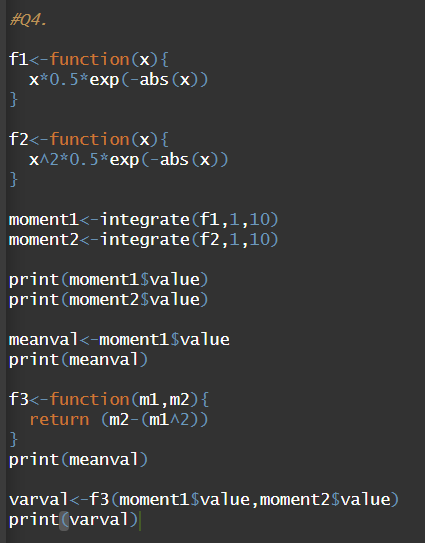
****

**OUTPUT:**

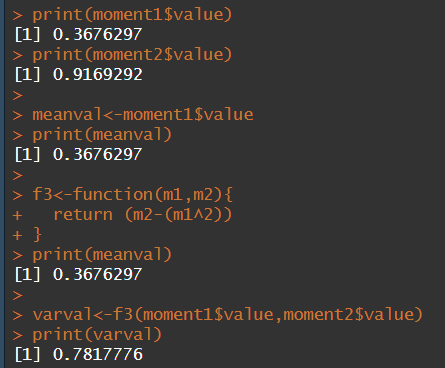
****

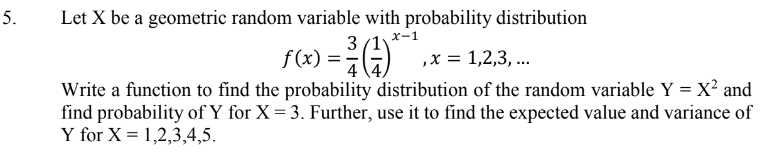
****

**CODE:**

****

**OUTPUT:**

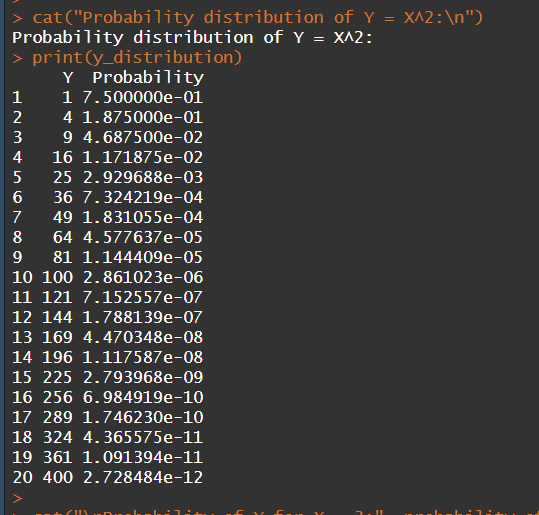
****

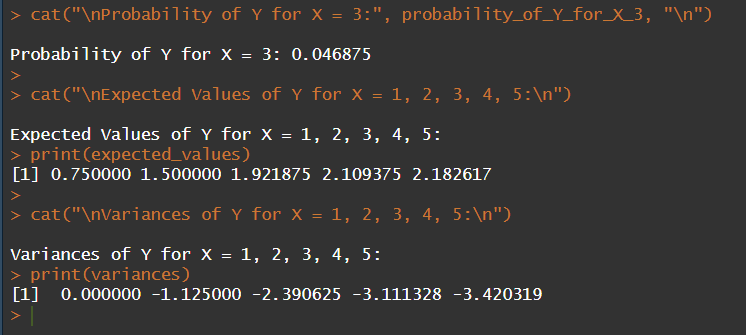
****

**CODE:**

****

**OUTPUT:**

****

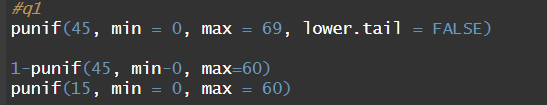


**Probability and Statistics (UCS410)**

**Experiment 5: Continuous Probability Distributions**

1. **Consider that X is the time (in minutes) that a person has to wait in order to take a flight. If each flight takes off each hour X ~ U(0, 60). Find the probability that**
2. **waiting time is more than 45 minutes, and**

Code:

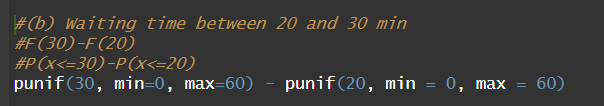


Output:

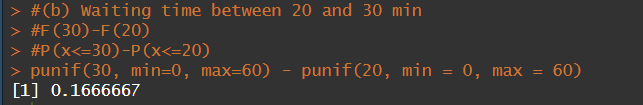


1. **waiting time lies between 20 and 30 minutes.**

Code:

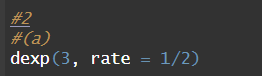


Output:

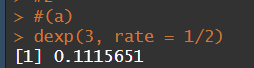


1. **The time (in hours) required to repair a machine is an exponential distributed random variable with parameter λ = 1/2.**
2. **Find the value of density function at x = 3.**

Code:

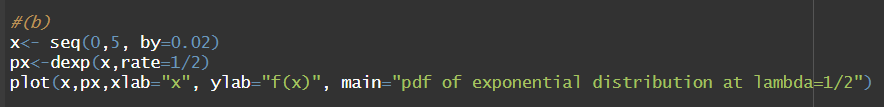


Output:

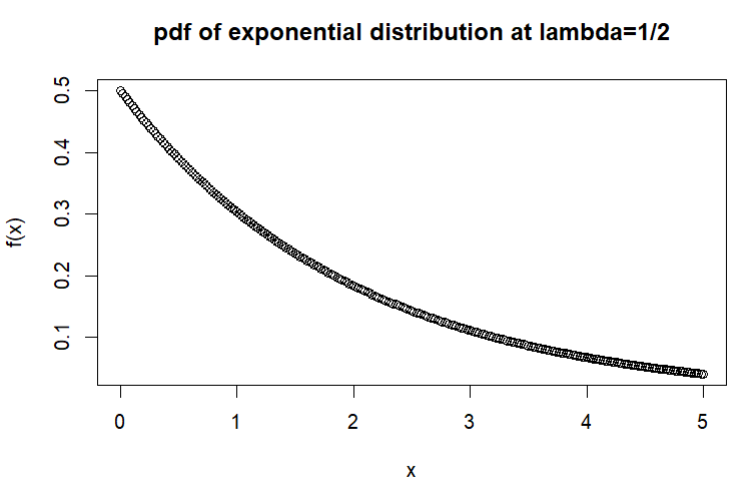


1. **Plot the graph of exponential probability distribution for 0 ≤ x ≤ 5.**

Code:

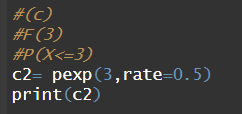


Output:

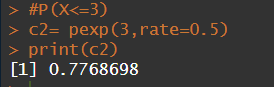


1. **Find the probability that a repair time takes at most 3 hours.**

Code:

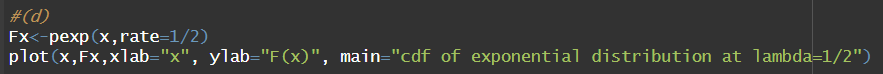


Output:

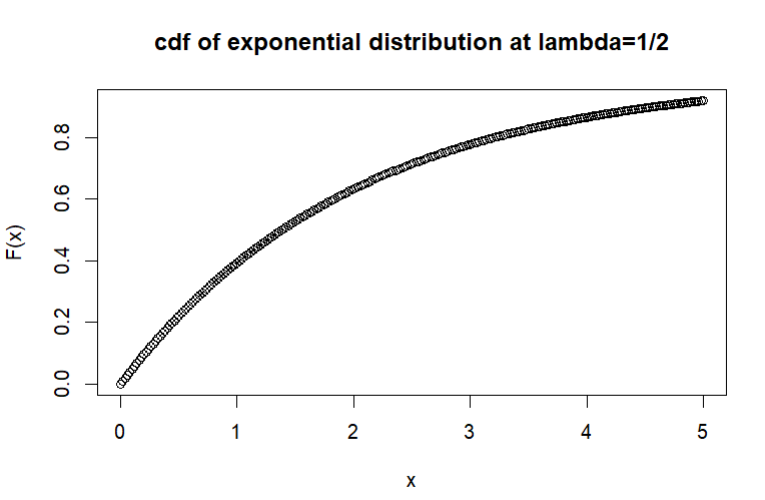


1. **Plot the graph of cumulative exponential probabilities for 0 ≤ x ≤ 5.**

Code:

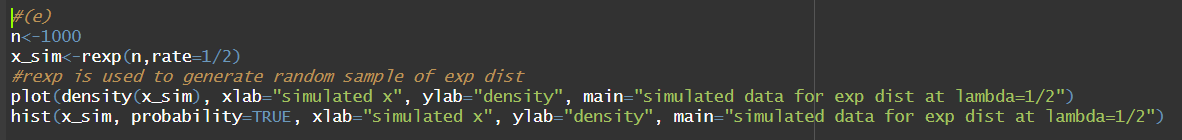


Output:

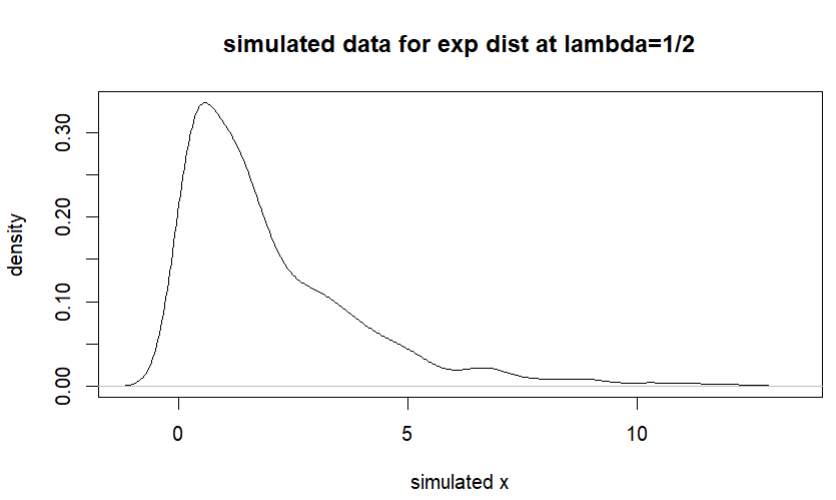


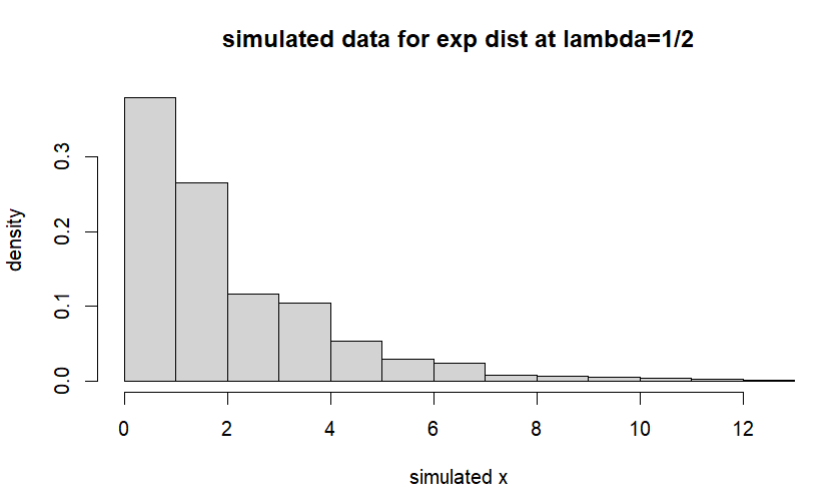
1. **Simulate 1000 exponential distributed random numbers with λ = 1⁄2 and plot the simulated data.**

Code:



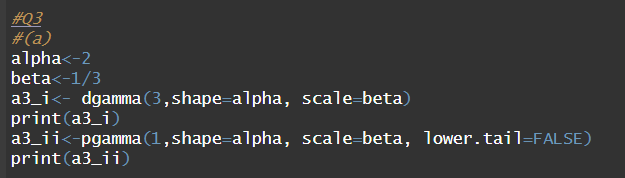
Output:



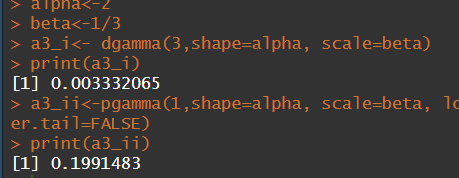


1. **The lifetime of certain equipment is described by a random variable X that follows Gamma distribution with parameters α = 2 and β = 1/3.**
2. **Find the probability that the lifetime of equipment is at least 1 unit of time.**

Code:

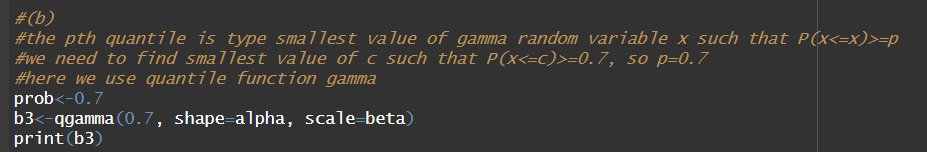
****

Output:

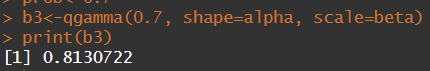


1. **What is the value of c, if P(X ≤ c) ≥ 0.70? (Hint: try quantile function qgamma())**

Code:

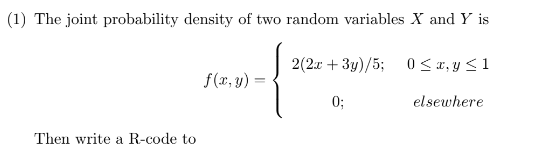


Output:

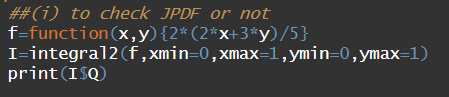


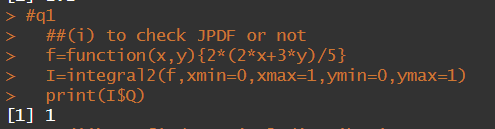
**Probability and Statistics (UCS410)**

**Experiment 6: Joint probability mass and density functions**

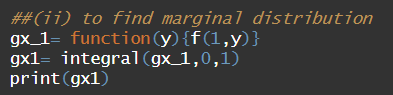


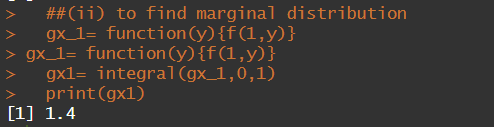
1. check that it is a joint density function or not? (Use integral2())



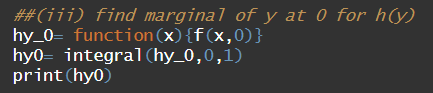


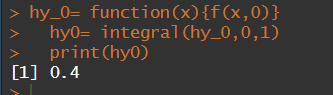
1. find marginal distribution g(x) at x = 1.



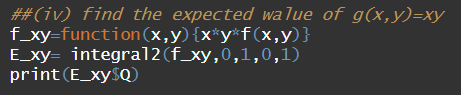


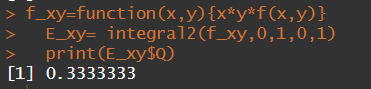
1. find the marginal distribution h(y) at y = 0.

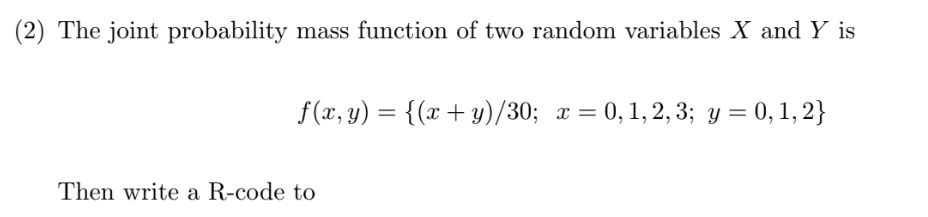




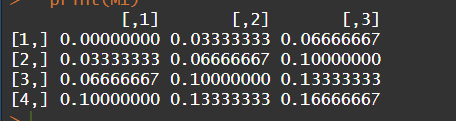
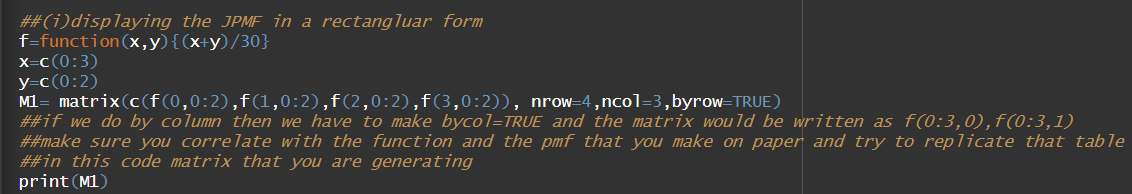
1. find the expected value of g(x, y) = xy.







1. display the joint mass function in rectangular (matrix) form.

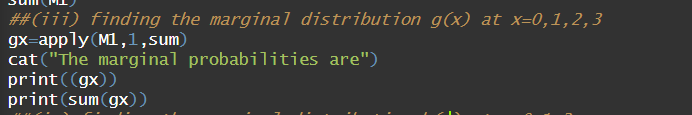


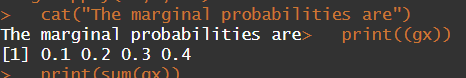
1. check that it is joint mass function or not? (use: Sum())



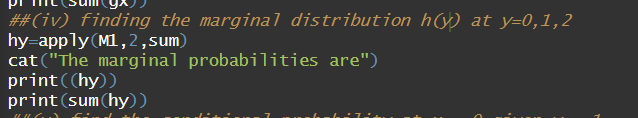


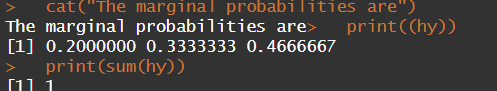
1. find the marginal distribution g(x) for x = 0, 1, 2, 3. (Use:apply())



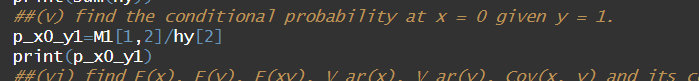


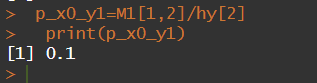
1. find the marginal distribution h(y) for y = 0, 1, 2. (Use:apply())



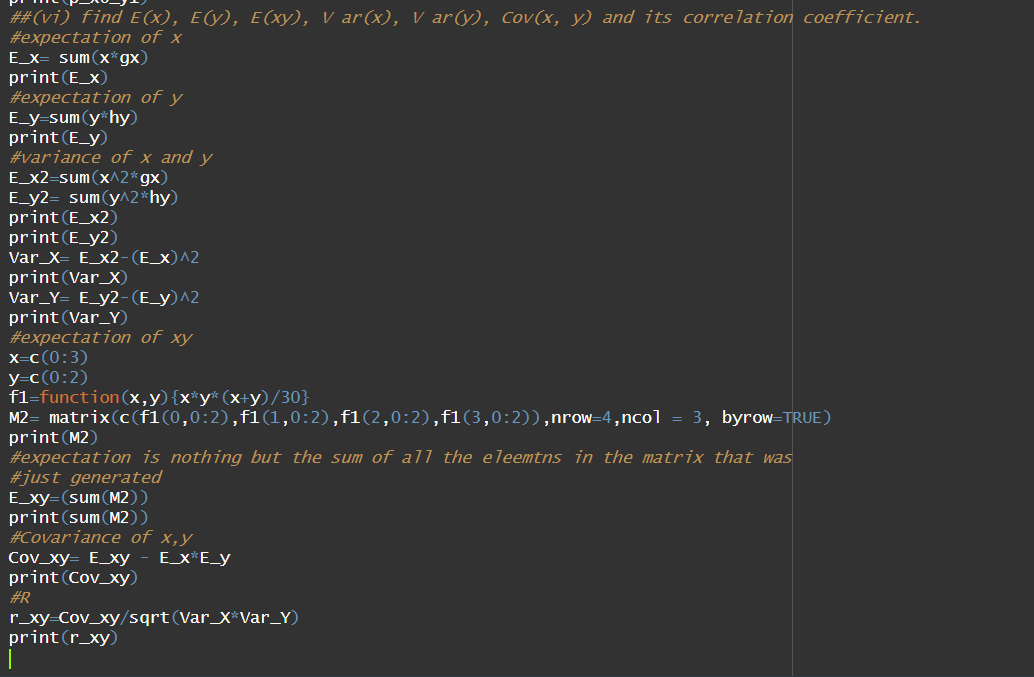


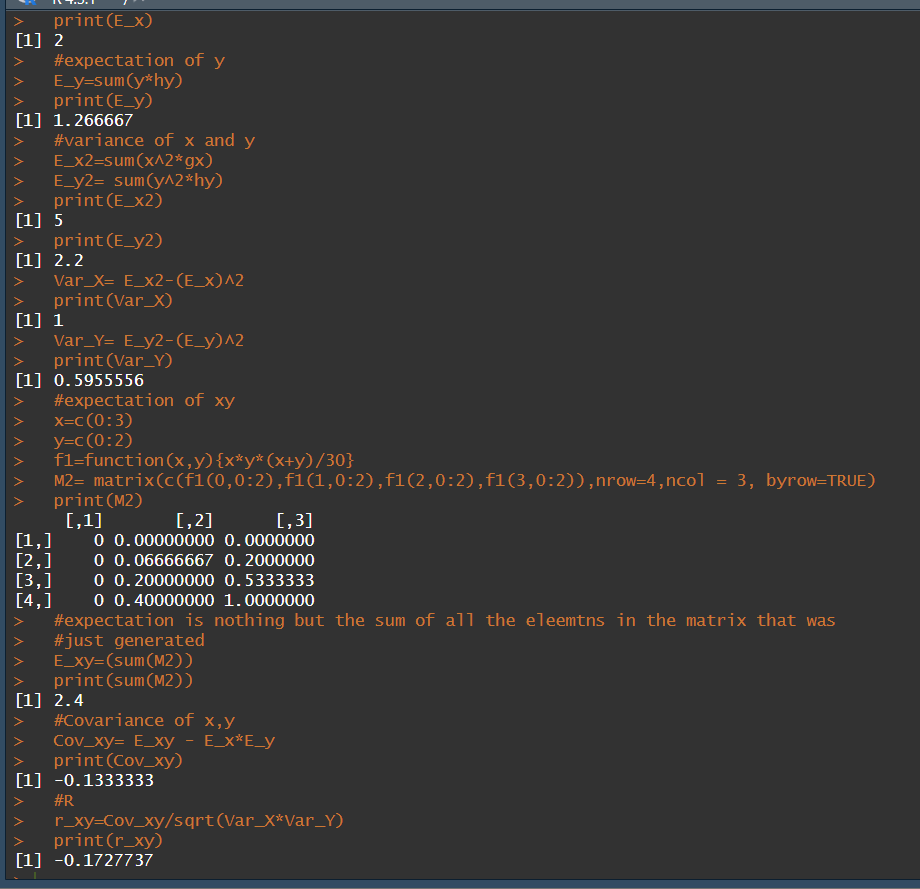
1. find the conditional probability at x = 0 given y = 1.





1. find E(x), E(y), E(xy), V ar(x), V ar(y), Cov(x, y) and its correlation coefficient.





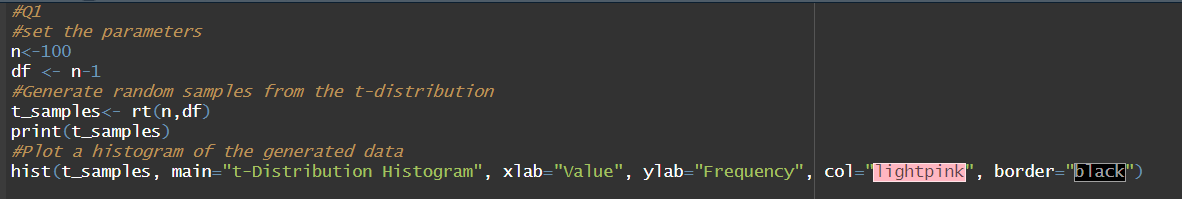
**Probability and Statistics (UCS410)**

**Experiment 7: Chi-square, t-distribution, F-distribution**

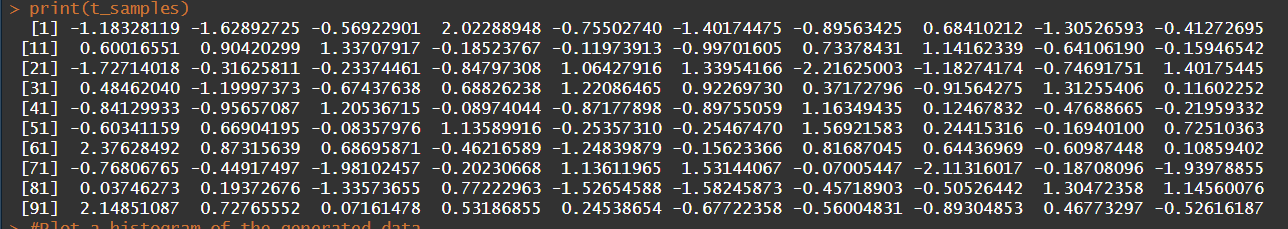
**1) Use the rt(n, df) function in r to investigate the t-distribution for n = 100 and df = n − 1 and plot**

**the histogram for the same.**

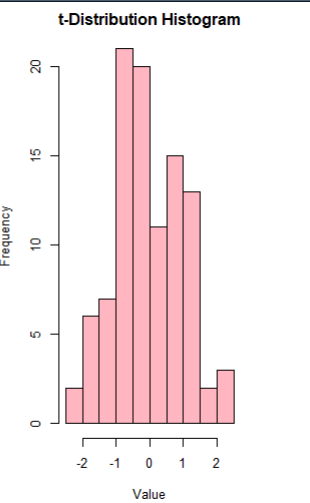
**CODE:**

****

**OUTPUT:**

****

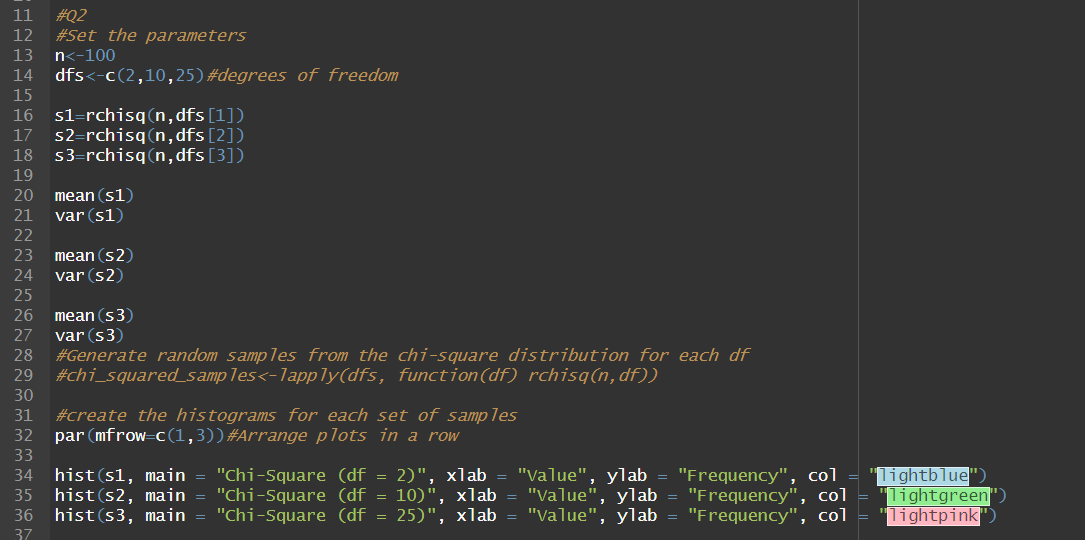
**PLOT:**

****

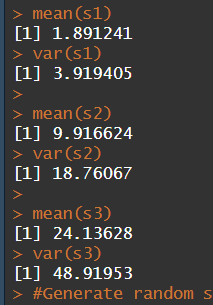
**(2) Use the rchisq(n, df) function in r to investigate the chi-square distribution with n = 100 and**

**df = 2, 10, 25.**

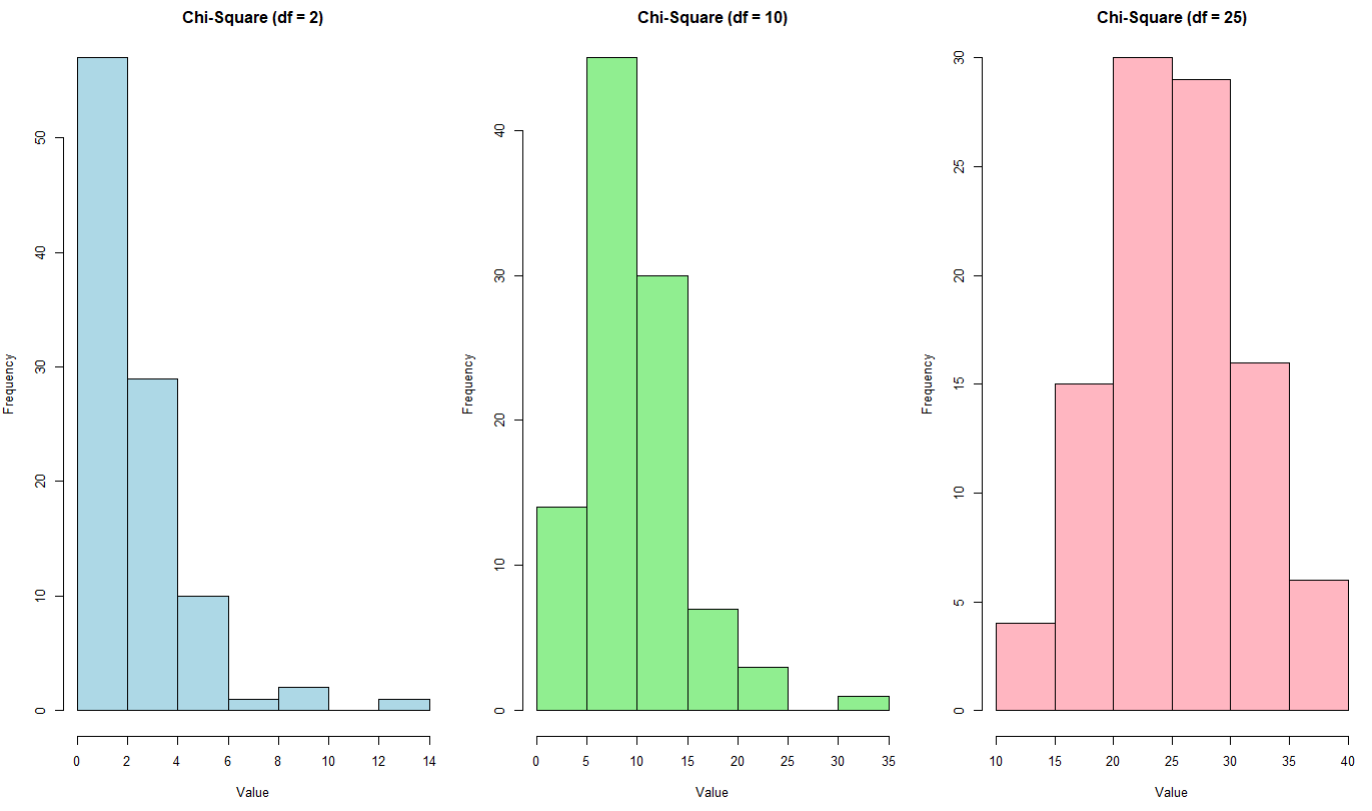
**CODE:**

****

**OUTPUT:**

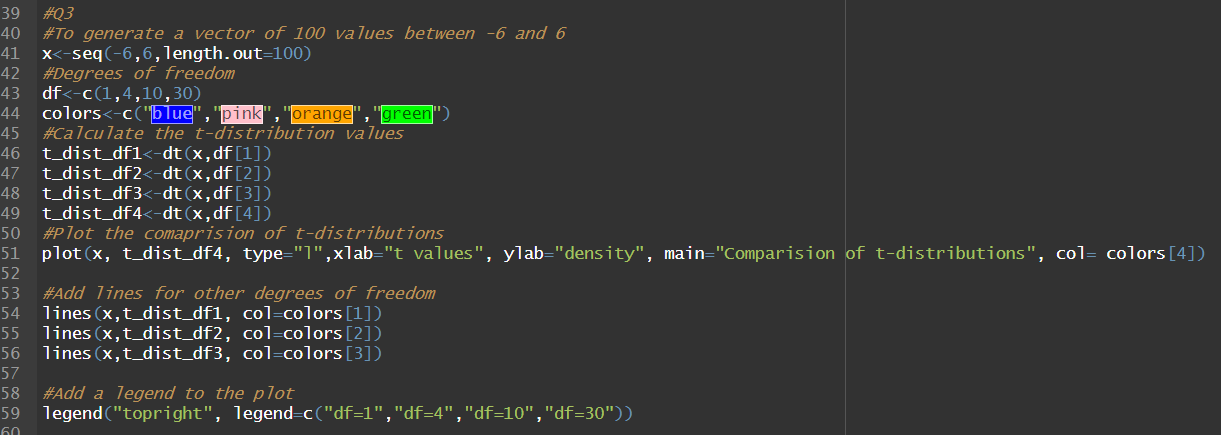
****

**PLOTS:**

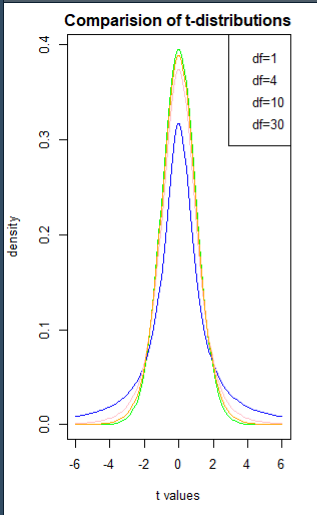
****

**(3) Generate a vector of 100 values between -6 and 6. Use the dt() function in r to find the values of a t-distribution given a random variable x and degrees of freedom 1,4,10,30. Using these values plot the density function for students t-distribution with degrees of freedom 30. Also shows a comparison of probability density functions having different degrees of freedom (1,4,10,30).**

**CODE:**

****

**OUTPUT/PLOT:**

****

**(4) Write a r-code**

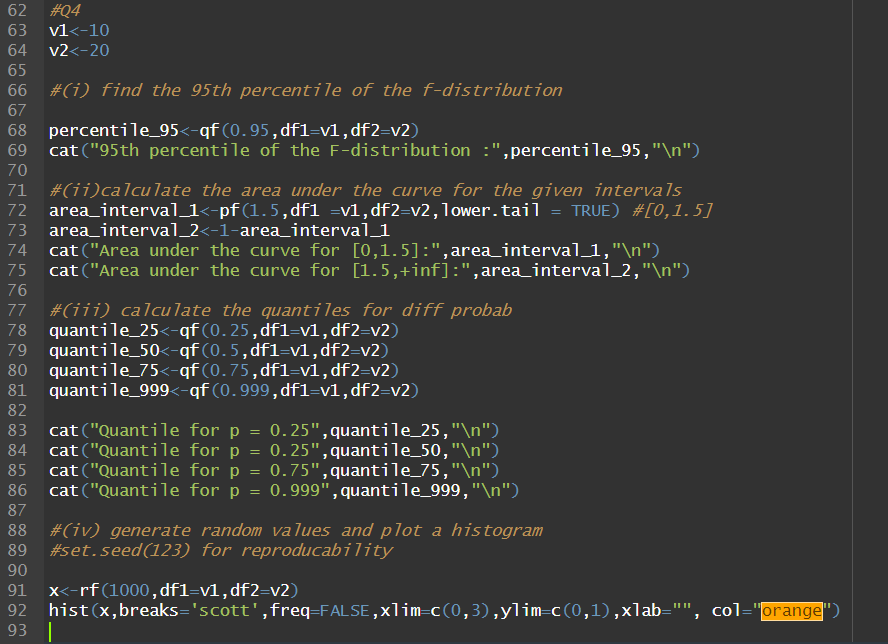
**(i) To find the 95th percentile of the F-distribution with (10, 20) degrees of freedom.**

**(ii) To calculate the area under the curve for the interval [0, 1.5] and the interval [1.5, +∞) of a F-curve with v1 = 10 and v2 = 20 (USE pf()).**

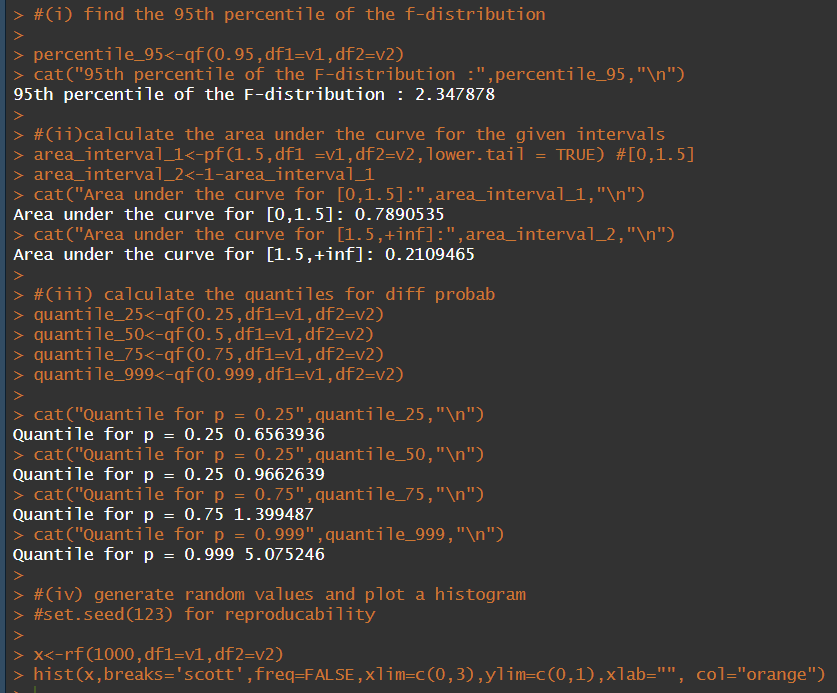
**(iii) To calculate the quantile for a given area (= probability) under the curve for a F-curve with v1 = 10 and v2 = 20 that corresponds to q = 0.25, 0.5, 0.75 and 0.999. (use the qf())**

**(iv) To generate 1000 random values from the F-distribution with v1 = 10 and v2 = 20 (use rf())and plot a histogram.**

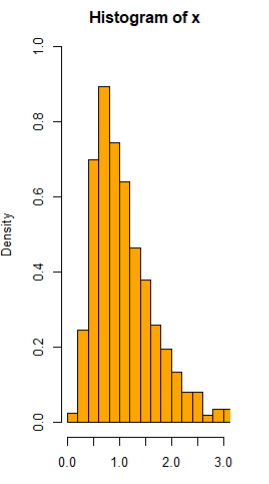
**CODE:**

****

**OUTPUT:**

****

**PLOT:**

****

**Probability and Statistics (UCS410)**

**Experiment 8**

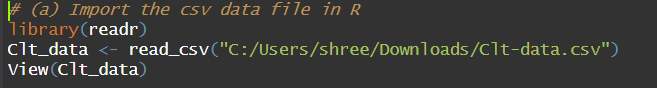
**A pipe manufacturing organization produces different kinds of pipes. We are given the monthly data of the wall thickness of certain types of pipes (data is available on LMS Clt-data.csv).**

**The organization has an analysis to perform and one of the basic assumptions of that analysis is that the data should be normally distributed.**

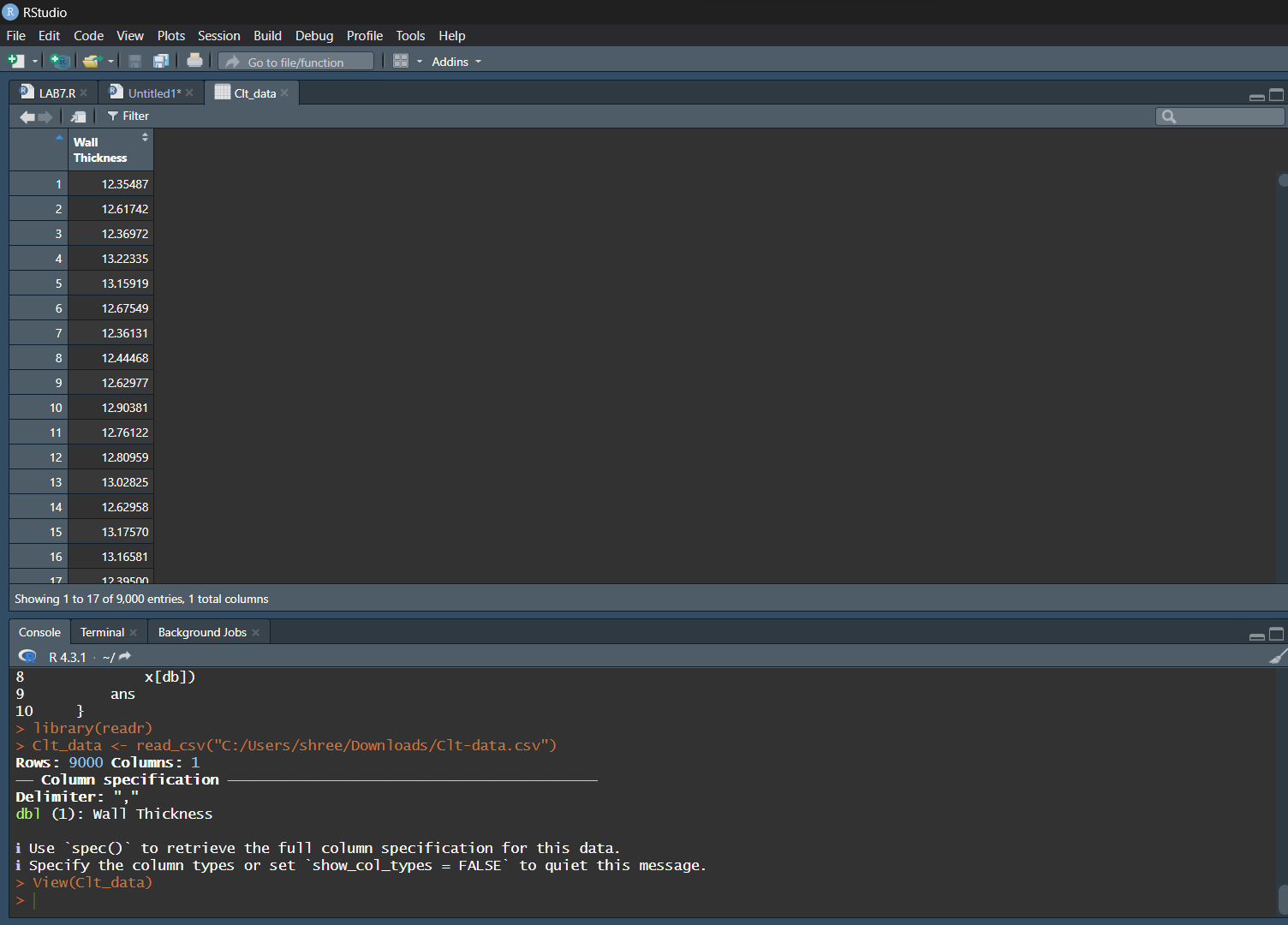
**You have the following tasks to do:**

1. **Import the csv data file in R.**

Code:

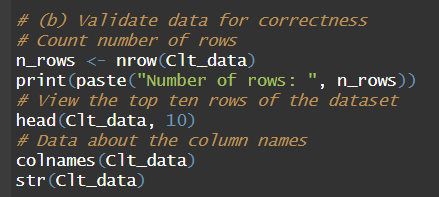


Output:

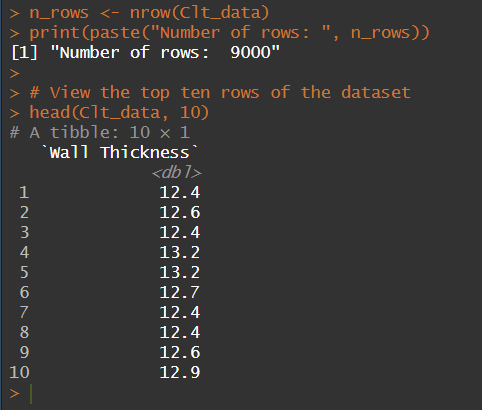
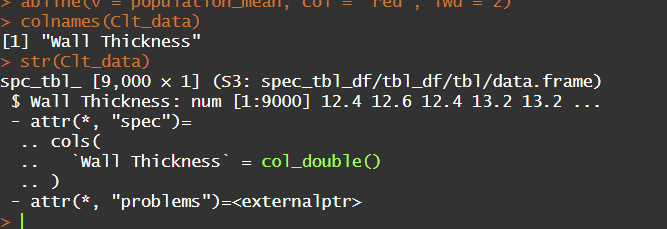


1. **Validate data for correctness by counting number of rows and viewing the top ten rows of the dataset.**

Code:

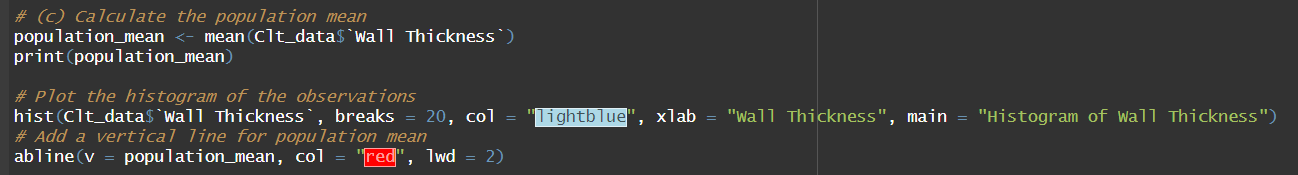


Output:

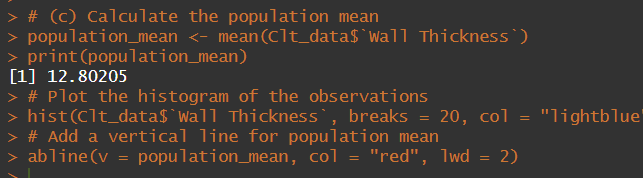
 

1. **Calculate the population mean and plot the observations by making a histogram.**

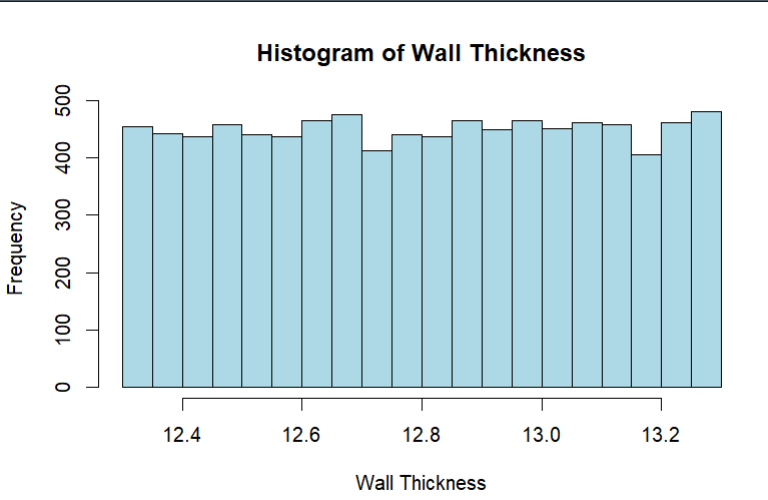
Code:



Output:

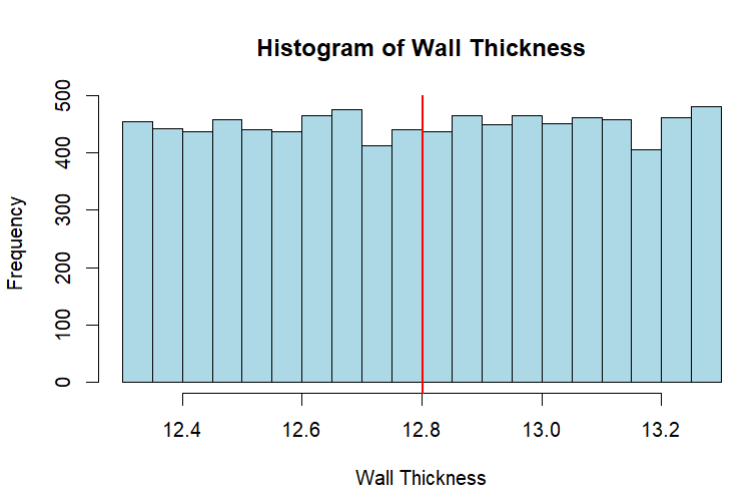


The Histogram:



1. **Mark the mean computed in last step by using the function abline.**

Histogram with abline:



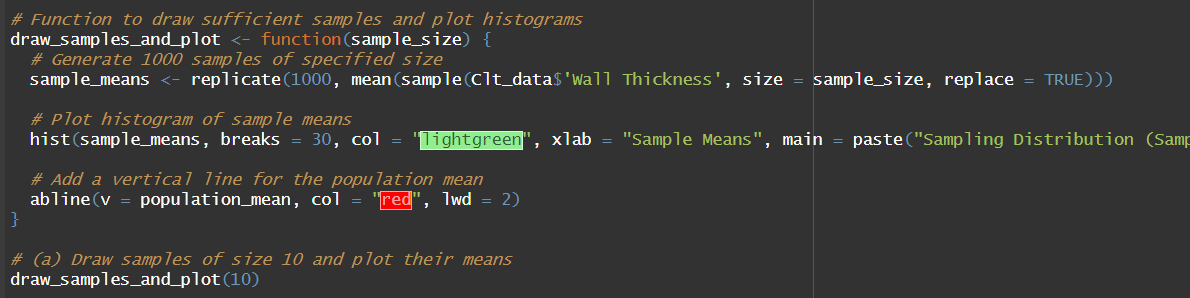
See the red vertical line in the histogram? That’s the population mean. Comment on whether the data is normally distributed or not?

Ans: Although the **abline** is right in the middle of the histogram still it does not confirm it is a normal distribution. After studying the histogram, we can clearly say that the histogram does NOT resemble a BELL-SHAPED CURVE so we can say that the data is **NOT NORMALLY DISTRIBUTED**.

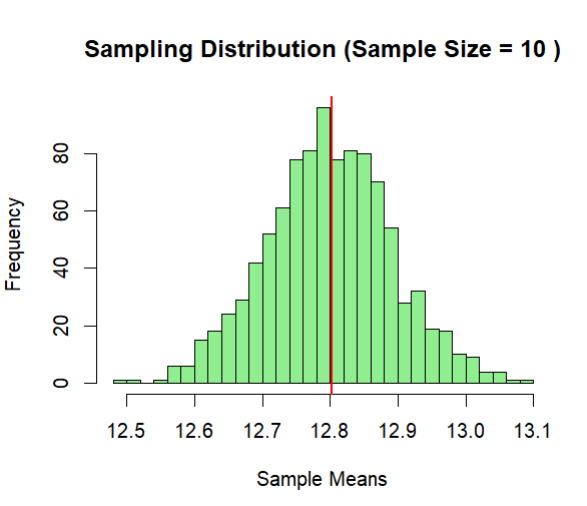
Now perform the following tasks:

1. **Draw sufficient samples of size 10, calculate their means, and plot them in R by making histogram. Do you get a normal distribution.**

Code:



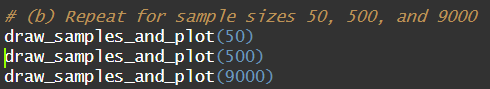
Histogram:



The mean is coming around 12.8. The histogram clearly represents a Bell-Shaped curve. So we can conclude that the sample is **NORMALLY DISTRIBUTED**.

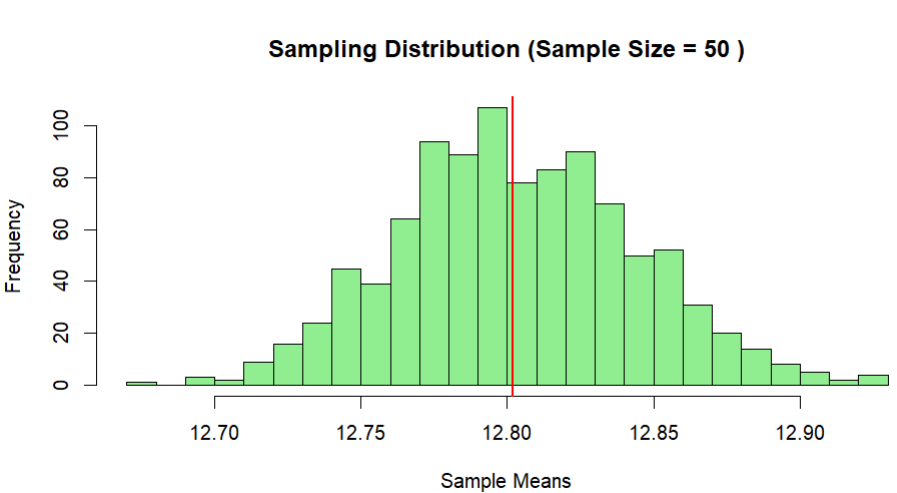
1. **Now repeat the same with sample size 50, 500 and 9000. Can you comment on what you observe.**

Code:



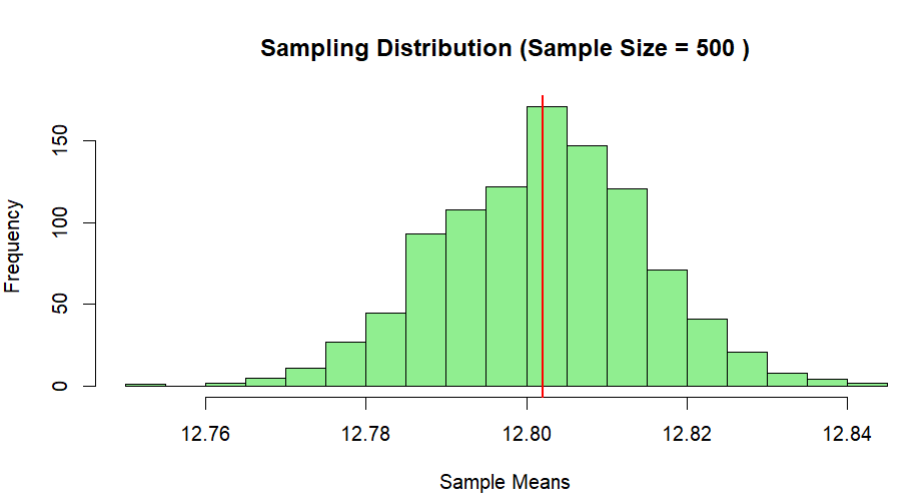
**50:**

Histogram:

****

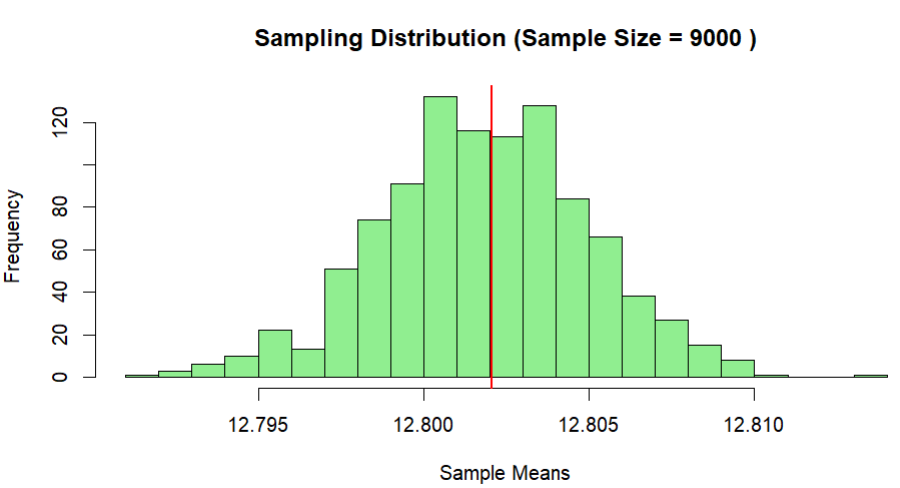
**500:**

Histogram:

****

**9000:**

Histogram:

****

Here, we get a good bell-shaped curve and the sampling distribution approaches normal distribution as the sample sizes increase. Therefore, we can recommend the organization to use sampling distributions of mean for further analysis.

**ALL CODES:**

**EXPERIMENT 1**

#(1)

v1<-c(5,10,15,20,25,30)

print(paste("Maximum Number is: ", max(v1)))

print(paste("Minimum Number is: ", min(v1)))

#(2)

factorial\_result <- 1

n <- as.integer(readline(prompt = "Enter integer: "))

if (n <= 0) {

print('Error')

} else {

for (i in 1:n) {

factorial\_result <- factorial\_result \* i

}

print(paste("Factorial of", n, "is", factorial\_result))

}

#(3)

n <- as.integer(readline("Enter the value of n: "))

if (n <= 2) {

print('Error')

}else {

fib <- numeric(n)

fib[1] <- 0

fib[2] <- 1

for (i in 3:n) {

fib[i] <- fib[i - 1] + fib[i - 2]

}

cat("Fibonacci sequence of", n, "terms:", fib)

}

#(4)

num1 <- as.numeric(readline("Enter the first number: "))

num2 <- as.numeric(readline("Enter the second number: "))

cat("Select operation:\n1. Add\n2. Subtract\n3. Multiply\n4. Divide\n")

choice <- as.integer(readline("Enter choice (1/2/3/4): "))

result <- switch(choice,

"1" = num1 + num2,

"2" = num1 - num2,

"3" = num1 \* num2,

"4" = {

if (num2 == 0) {

stop("Error: Division by zero is not allowed.")

}

num1 / num2

},

stop("Error: Invalid choice."))

cat("Result:", result)

#(5)

# Load necessary library for plotting

install.packages("plotrix")

library(plotrix)

cities <- c("Kolkata", "Mumbai", "Delhi", "Chennai", "Patiala")

values <- c(5, 8, 30, 22, 55)

print(pie(values, labels = cities, main = "City Distribution"))

#BAR GRAPH

bar\_colors <- c("red", "green", "blue", "orange", "purple")

barplot(values, names.arg = cities, main = "City Distribution")

#HISTOGRAM

data <- rnorm(100)

hist(data, main = "Histogram of Random Data", xlab = "Value", ylab = "Frequency", col = "blue")

#SCATTER PLOT

x <- rnorm(50)

y <- 2 \* x + rnorm(50)

plot(x, y, main = "Scatter Plot", xlab = "X", ylab = "Y", col = "red", pch = 19)

#LINE PLOT

x <- seq(0, 2 \* pi, length.out = 100)

y <- sin(x)

plot(x, y, type = "l", main = "Sine Function", xlab = "X", ylab = "Y", col = "green")

#BOX PLOT

data <- matrix(rnorm(200), ncol = 4)#random data

boxplot(data, main = "Box Plot of Random Data", col = c("red", "blue", "green", "purple"))

**EXPERIMENT 2**

##(1)

coins <- c(rep("gold", 20), rep("silver", 30), rep("bronze", 50))

sampleSpace <- sample(coins, size = 10, replace = FALSE)

print(sampleSpace)

##(2)

outcomes <- c("Success", "Failure")

probab <- c(0.9, 0.1)

# Generate a sample space for 10 surgical procedures

sample\_space <- sample(outcomes, size = 10, replace = TRUE, prob = probab)

# Display

cat("Sample space for next 10 Procedures:\n", sample\_space)

##(3)

###TAKING USER INPUT OF N

n <- as.integer(readline("Number of people in the room: "))

# Calculate probability

prob\_no\_shared <- 1

for (i in 1:n) {

prob\_no\_shared <- prob\_no\_shared \* (365 - i + 1) / 365

}

prob\_shared <- 1 - prob\_no\_shared

cat("Probability that at least two people share a birthday in a room with", n, "people:", prob\_shared, "\n")

##USING AN ARRAY OF VALUES FOR N

num\_simulations <- 10000

for (n in c(5, 10, 15, 20, 25)) {

shared\_birthday\_count <- 0

for (sim in 1:num\_simulations) {

birthdays <- sample(1:365, size = n, replace = TRUE)

if (length(birthdays) != length(unique(birthdays))) {

shared\_birthday\_count <- shared\_birthday\_count + 1

}

}

prob\_shared <- shared\_birthday\_count / num\_simulations

cat("Estimated probability of shared birthday with", n, "people:", prob\_shared, "\n")

}

##SMALLEST VALUE OF n FOR WHICH THE PROBABILITY IS GREATER THAN 0.5

num\_simulations <- 10000

n <- 1

while (TRUE) {

shared\_birthday\_count <- 0

for (sim in 1:num\_simulations) {

birthdays <- sample(1:365, size = n, replace = TRUE)

if (length(birthdays) != length(unique(birthdays))) {

shared\_birthday\_count <- shared\_birthday\_count + 1

}

}

prob\_shared <- shared\_birthday\_count / num\_simulations

if (prob\_shared > 0.5) {

break

}

n <- n + 1

}

cat("Smallest value of n for which the probability is greater than 0.5:", n, "\n")

###(3)

conditional\_probability <- function(prob\_a, prob\_b\_given\_a, prob\_b) {

prob\_a\_given\_b <- (prob\_b\_given\_a \* prob\_a) / prob\_b

return(prob\_a\_given\_b)

}

# Given probabilities

prob\_cloudy <- 0.4

prob\_rain <- 0.2

prob\_clouds\_given\_rain <- 0.85

# Compute the probability of rain given that it's cloudy

prob\_rain\_given\_cloudy <- conditional\_probability(prob\_rain, prob\_clouds\_given\_rain, prob\_cloudy)

cat("Probability of rain given that it's cloudy:", prob\_rain\_given\_cloudy, "\n")

###(4)

#Loading the dataset

data(iris)

# (a) Print first few rows of the dataset

head(iris)

# (b) Find the structure of the dataset

str(iris)

# (c) Find the range of sepal length

range\_sepal\_length <- range(iris$Sepal.Length)

cat("Range of sepal length:", range\_sepal\_length, "\n")

# (d) Find the mean of sepal length

mean\_sepal\_length <- mean(iris$Sepal.Length)

cat("Mean of sepal length:", mean\_sepal\_length, "\n")

# (e) Find the median of sepal length

median\_sepal\_length <- median(iris$Sepal.Length)

cat("Median of sepal length:", median\_sepal\_length, "\n")

# (f) Find the first and third quartiles and the interquartile range

quartiles\_sepal\_length <- quantile(iris$Sepal.Length, probs = c(0.25, 0.75))

iqr\_sepal\_length <- quartiles\_sepal\_length[2] - quartiles\_sepal\_length[1]

cat("First Quartile:", quartiles\_sepal\_length[1], "\n")

cat("Third Quartile:", quartiles\_sepal\_length[2], "\n")

cat("Interquartile Range:", iqr\_sepal\_length, "\n")

# (g) Find the standard deviation and variance of sepal length

sd\_sepal\_length <- sd(iris$Sepal.Length)

var\_sepal\_length <- var(iris$Sepal.Length)

cat("Standard Deviation of sepal length:", sd\_sepal\_length, "\n")

cat("Variance of sepal length:", var\_sepal\_length, "\n")

# (h) Perform the above exercises for other attributes (sepal.width, petal.length, petal.width)

##SEPAL.WIDTH

#Range

range\_sepal\_width <- range(iris$Sepal.Width)

cat("Range of sepal width:", range\_sepal\_width, "\n")

#Mean

mean\_sepal\_width <- mean(iris$Sepal.Width)

cat("Mean of sepal width:", mean\_sepal\_width, "\n")

#Median

median\_sepal\_width <- median(iris$Sepal.Width)

cat("Median of sepal width:", median\_sepal\_width, "\n")

#Quartiles

quartiles\_sepal\_width <- quantile(iris$Sepal.Width, probs = c(0.25, 0.75))

iqr\_sepal\_width <- quartiles\_sepal\_width[2] - quartiles\_sepal\_width[1]

cat("First Quartile of sepal width:", quartiles\_sepal\_width[1], "\n")

cat("Third Quartile of sepal width:", quartiles\_sepal\_width[2], "\n")

cat("Interquartile Range of sepal width:", iqr\_sepal\_width, "\n")

##Standard Deviation and Variance

sd\_sepal\_width <- sd(iris$Sepal.Width)

var\_sepal\_width <- var(iris$Sepal.Width)

cat("Standard Deviation of sepal width:", sd\_sepal\_width, "\n")

cat("Variance of sepal width:", var\_sepal\_width, "\n")

##PETAL.WIDTH

#Range

range\_petal\_width <- range(iris$Petal.Width)

cat("Range of petal width:", range\_petal\_width, "\n")

#Mean

mean\_petal\_width <- mean(iris$Petal.Width)

cat("Mean of petal width:", mean\_petal\_width, "\n")

#Median

median\_petal\_width <- median(iris$Petal.Width)

cat("Median of petal width:", median\_petal\_width, "\n")

#Quartiles

quartiles\_petal\_width <- quantile(iris$Petal.Width, probs = c(0.25, 0.75))

iqr\_petal\_width <- quartiles\_petal\_width[2] - quartiles\_petal\_width[1]

cat("First Quartile of petal width:", quartiles\_petal\_width[1], "\n")

cat("Third Quartile of petal width:", quartiles\_petal\_width[2], "\n")

cat("Interquartile Range of petal width:", iqr\_petal\_width, "\n")

##Standard Deviation and Variance

sd\_petal\_width <- sd(iris$Petal.Width)

var\_petal\_width <- var(iris$Petal.Width)

cat("Standard Deviation of sepal width:", sd\_petal\_width, "\n")

cat("Variance of sepal width:", var\_petal\_width, "\n")

##PETAL.LENGTH

#Range

range\_petal\_length <- range(iris$Petal.Length)

cat("Range of petal length:", range\_petal\_length, "\n")

#Mean

mean\_petal\_length <- mean(iris$Petal.Length)

cat("Mean of petal length:", mean\_petal\_length, "\n")

#Median

median\_petal\_length <- median(iris$Petal.Length)

cat("Median of petal length:", median\_petal\_length, "\n")

#Quartiles

quartiles\_petal\_length <- quantile(iris$Petal.Length, probs = c(0.25, 0.75))

iqr\_petal\_length <- quartiles\_petal\_length[2] - quartiles\_petal\_length[1]

cat("First Quartile of petal length:", quartiles\_petal\_length[1], "\n")

cat("Third Quartile of petal length:", quartiles\_petal\_length[2], "\n")

cat("Interquartile Range of petal length:", iqr\_petal\_length, "\n")

##Standard Deviation and Variance

sd\_petal\_length <- sd(iris$Petal.Length)

var\_petal\_length <- var(iris$Petal.Length)

cat("Standard Deviation of petal length:", sd\_petal\_length, "\n")

cat("Variance of petal length:", var\_petal\_length, "\n")

# (i) Use the built-in function summary on the dataset Iris

summary(iris)

#R does not have a standard in-built function to calculate mode.

#So we create a user function to calculate mode of a data set in R.

#This function n takes the vector as input and gives the mode value as output.

calculate\_mode <- function(data) {

table\_data <- table(data)

mode <- as.numeric(names(table\_data[table\_data == max(table\_data)]))

return(mode)

}

# Test the function

dataset <- c(1,0,2,1,0,3,4,4,7)

mode\_value <- calculate\_mode(dataset)

cat("Mode of the dataset:", mode\_value, "\n")

**EXPERIMENT 3**

#Q1.

# Number of dice rolls

n <- 12

# Probability of getting a 6 on one roll

p\_success <- 1/6

# Calculate the cumulative probabilities using binomial distribution

cum\_prob\_6 <- pbinom(6, size = n, prob = p\_success)

cum\_prob\_9 <- pbinom(9, size = n, prob = p\_success)

# Calculate the probability of getting 7, 8, or 9 sixes

prob\_7\_to\_9 <- cum\_prob\_9 - cum\_prob\_6

cat("Probability of getting 7, 8, or 9 sixes:", prob\_7\_to\_9, "\n")

###USING DBINOM

# Number of dice rolls

n <- 12

# Probability of getting a 6 on one roll

p\_success <- 1/6

# Calculate the probabilities using binomial distribution (PDF)

prob\_7 <- dbinom(7, size = n, prob = p\_success)

prob\_8 <- dbinom(8, size = n, prob = p\_success)

prob\_9 <- dbinom(9, size = n, prob = p\_success)

# Calculate the probability of getting 7, 8, or 9 sixes

prob\_7\_to\_9 <- prob\_7+prob\_8+prob\_9

cat("Probability of getting 7, 8, or 9 sixes:", prob\_7\_to\_9, "\n")

#Q2

# Mean and standard deviation

mean\_score <- 72

standard\_deviation <- 15.2

# Score threshold

threshold <- 84

# cumulative distribution function (CDF)

probability\_above\_threshold <- 1 - pnorm(threshold, mean = mean\_score, sd = standard\_deviation)

# probability to percentage

percentage\_above\_threshold <- probability\_above\_threshold \* 100

cat("Percentage of students scoring 84 or more:", percentage\_above\_threshold, "%\n")

#Q3.

# library for Poisson distribution calculations

library(stats)

# Parameters for the Poisson distribution

lambda\_x <- 5 # Average number of cars from 10AM to 11AM

lambda\_y <- 50 # Average number of customers from 8AM to 6PM

# Probability that no car arrives during 10AM to 11AM

prob\_x <- dpois(0, lambda = lambda\_x)

# Probability of having between 48 and 50 customers (inclusive) from 8AM to 6PM

prob\_y <- sum(dpois(48:50, lambda = lambda\_y))

cat("Probability that no car arrives during 10AM to 11AM:", prob\_x, "\n")

cat("Probability of having between 48 and 50 customers from 8AM to 6PM:", prob\_y, "\n")

#Q4

# Parameters for hypergeometric distribution

total\_processors <- 250

defective\_processors <- 17

sample\_size <- 5

# p(x=3)

prob\_3 <- dhyper(3, m = defective\_processors, n = total\_processors - defective\_processors, k = sample\_size)

cat("Probability of exactly 3 defective processors in the sample:", prob\_3, "\n")

#Q5

# Given probability

p\_success <- 0.447

# Sample size

n <- 31

# (a) X is distributed as a binomial distribution

# (b) Sketch the probability mass function (PMF)

xx <- seq(0,31,1)

pmf\_values <- numeric()

cdf\_values <- numeric()

for(i in 1:length(xx))

{

pmf\_values[i] = dbinom(xx[i],n,p\_success)

}

plot(xx,pmf\_values)

# (c) Sketch the cumulative distribution function (CDF)

for(i in 1:length(xx))

{

cdf\_values[i] = pbinom(xx[i],n,p\_success)

}

# (d) Mean, variance, and standard deviation

mean\_x <- n \* p\_success

variance\_x <- n \* p\_success \* (1 - p\_success)

std\_dev\_x <- sqrt(variance\_x)

# Print the results

cat("Mean of X:", mean\_x, "\n")

cat("Variance of X:", variance\_x, "\n")

cat("Standard Deviation of X:", std\_dev\_x, "\n")

# Plot PMF and CDF

plot(xx, pmf\_values, xlab = "Number of Students (X)", ylab = "Probability", main = "Probability Mass Function (PMF) of X")

plot(xx, cdf\_values, xlab = "Number of Students (X)", ylab = "Cumulative Probability", main = "Cumulative Distribution Function (CDF) of X")

**EXPERIMENT 4**

#Q1:

#Using weighted.mean()

x <- c(0, 1, 2, 3, 4)

p\_x <- c(0.41, 0.37, 0.16, 0.05, 0.01)

mean\_imperf <- weighted.mean(x, p\_x)

cat("The average number of imperfections per 10 meters of fabric is:", mean\_imperf)

#Using sum()

x <- c(0, 1, 2, 3, 4)

p\_x <- c(0.41, 0.37, 0.16, 0.05, 0.01)

mean\_imperf <- sum(x \* p\_x)

cat("The average number of imperfections per 10 meters of fabric is:", mean\_imperf)

#Q2.

pdf <- function(t){

t\*0.1\*exp(-0.1\*t)

}

expected\_value <- integrate(pdf, lower = 0, upper = Inf)$value

cat("The expected value of T is:", expected\_value)

#Q3.

#Y = (12X+(3-X)2 - (6\*3)) = (10X-12)

x<-c(0,1,2,3)

probab<-c(0.1,0.2,0.2,0.5)

print(weighted.mean(x,probab))

expval<-(10\*weighted.mean(x,probab))-12

print(expval)

cat("The expected value of Y (net revenue) is:", expval)

#or

x<-c(0,1,2,3)

probabx<-c(0.1,0.2,0.2,0.5)

y<-10\*x-12

probaby<-probabx

expval<-sum(y\*probaby)

cat("The expected value of Y (net revenue) is:", expval)

#Q4.

f1<-function(x){

x\*0.5\*exp(-abs(x))

}

f2<-function(x){

x^2\*0.5\*exp(-abs(x))

}

moment1<-integrate(f1,1,10)

moment2<-integrate(f2,1,10)

print(moment1$value)

print(moment2$value)

meanval<-moment1$value

print(meanval)

f3<-function(m1,m2){

return (m2-(m1^2))

}

print(meanval)

varval<-f3(moment1$value,moment2$value)

print(varval)

#Q5.

yf<-function(y){

(3/4)\*(1/4)^(sqrt(y)-1)

}

x<-as.integer(readline(prompt="Enter the value of x"))

y=x^2

proby<-yf(y)

print(proby)

x<-c(1,2,3,4,5)

y<-x^2

proby<-yf(y)

print(proby)

expval<-sum(y\*proby)

print(expval)

m<-expval

y1<-(y-m)^2

proby1<-yf(y1)

print(proby1)

varval<-sum(y1\*proby1)

print(varval)

**EXPERIMENT 5**

#q1

punif(45, min = 0, max = 69, lower.tail = FALSE)

1-punif(45, min-0, max=60)

punif(15, min = 0, max = 60)

#(b) Waiting time between 20 and 30 min

#F(30)-F(20)

#P(x<=30)-P(x<=20)

punif(30, min=0, max=60) - punif(20, min = 0, max = 60)

#2

#(a)

dexp(3, rate = 1/2)

#(b)

x<- seq(0,5, by=0.02)

px<-dexp(x,rate=1/2)

plot(x,px,xlab="x", ylab="f(x)", main="pdf of exponential distribution at lambda=1/2")

#(c)

#F(3)

#P(X<=3)

c2= pexp(3,rate=0.5)

print(c2)

#(d)

Fx<-pexp(x,rate=1/2)

plot(x,Fx,xlab="x", ylab="F(x)", main="cdf of exponential distribution at lambda=1/2")

#(e)

n<-1000

x\_sim<-rexp(n,rate=1/2)

#rexp is used to generate random sample of exp dist

plot(density(x\_sim), xlab="simulated x", ylab="density", main="simulated data for exp dist at lambda=1/2")

hist(x\_sim, probability=TRUE, xlab="simulated x", ylab="density", main="simulated data for exp dist at lambda=1/2")

#Q3

#(a)

alpha<-2

beta<-1/3

a3\_i<- dgamma(3,shape=alpha, scale=beta)

print(a3\_i)

a3\_ii<-pgamma(1,shape=alpha, scale=beta, lower.tail=FALSE)

print(a3\_ii)

#(b)

#the pth quantile is type smallest value of gamma random variable x such that P(x<=x)>=p

#we need to find smallest value of c such that P(x<=c)>=0.7, so p=0.7

#here we use quantile function gamma

prob<-0.7

b3<-qgamma(0.7, shape=alpha, scale=beta)

print(b3)

**EXPERIMENT 6**

library('pracma')

install.packages('pracma')

#q1

##(i) to check JPDF or not

f=function(x,y){2\*(2\*x+3\*y)/5}

I=integral2(f,xmin=0,xmax=1,ymin=0,ymax=1)

print(I$Q)

##(ii) to find marginal distribution

gx\_1= function(y){f(1,y)}

gx1= integral(gx\_1,0,1)

print(gx1)

##(iii) find marginal of y at 0 for h(y)

hy\_0= function(x){f(x,0)}

hy0= integral(hy\_0,0,1)

print(hy0)

##(iv) find the expected walue of g(x,y)=xy

f\_xy=function(x,y){x\*y\*f(x,y)}

E\_xy= integral2(f\_xy,0,1,0,1)

print(E\_xy$Q)

#Q2 JPMF is given

##(i)displaying the JPMF in a rectangluar form

f=function(x,y){(x+y)/30}

x=c(0:3)

y=c(0:2)

M1= matrix(c(f(0,0:2),f(1,0:2),f(2,0:2),f(3,0:2)), nrow=4,ncol=3,byrow=TRUE)

##if we do by column then we have to make bycol=TRUE and the matrix would be written as f(0:3,0),f(0:3,1)

##make sure you correlate with the function and the pmf that you make on paper and try to replicate that table

##in this code matrix that you are generating

print(M1)

##(ii) checking Joint Mass Function

sum(M1)

##(iii) finding the marginal distribution g(x) at x=0,1,2,3

gx=apply(M1,1,sum)

cat("The marginal probabilities are")

print((gx))

print(sum(gx))

##(iv) finding the marginal distribution h(y) at y=0,1,2

hy=apply(M1,2,sum)

cat("The marginal probabilities are")

print((hy))

print(sum(hy))

##(v) find the conditional probability at x = 0 given y = 1.

p\_x0\_y1=M1[1,2]/hy[2]

print(p\_x0\_y1)

##(vi) find E(x), E(y), E(xy), V ar(x), V ar(y), Cov(x, y) and its correlation coefficient.

#expectation of x

E\_x= sum(x\*gx)

print(E\_x)

#expectation of y

E\_y=sum(y\*hy)

print(E\_y)

#variance of x and y

E\_x2=sum(x^2\*gx)

E\_y2= sum(y^2\*hy)

print(E\_x2)

print(E\_y2)

Var\_X= E\_x2-(E\_x)^2

print(Var\_X)

Var\_Y= E\_y2-(E\_y)^2

print(Var\_Y)

#expectation of xy

x=c(0:3)

y=c(0:2)

f1=function(x,y){x\*y\*(x+y)/30}

M2= matrix(c(f1(0,0:2),f1(1,0:2),f1(2,0:2),f1(3,0:2)),nrow=4,ncol = 3, byrow=TRUE)

print(M2)

#expectation is nothing but the sum of all the eleemtns in the matrix that was

#just generated

E\_xy=(sum(M2))

print(sum(M2))

#Covariance of x,y

Cov\_xy= E\_xy - E\_x\*E\_y

print(Cov\_xy)

#R

r\_xy=Cov\_xy/sqrt(Var\_X\*Var\_Y)

print(r\_xy)

**EXPERIMENT 7**

#Q1

#set the parameters

n<-100

df <- n-1

#Generate random samples from the t-distribution

t\_samples<- rt(n,df)

print(t\_samples)

#Plot a histogram of the generated data

hist(t\_samples, main="t-Distribution Histogram", xlab="Value", ylab="Frequency", col="lightpink", border="black")

#Q2

#Set the parameters

n<-100

dfs<-c(2,10,25)#degrees of freedom

s1=rchisq(n,dfs[1])

s2=rchisq(n,dfs[2])

s3=rchisq(n,dfs[3])

mean(s1)

var(s1)

mean(s2)

var(s2)

mean(s3)

var(s3)

#Generate random samples from the chi-square distribution for each df

#chi\_squared\_samples<-lapply(dfs, function(df) rchisq(n,df))

#create the histograms for each set of samples

par(mfrow=c(1,3))#Arrange plots in a row

hist(s1, main = "Chi-Square (df = 2)", xlab = "Value", ylab = "Frequency", col = "lightblue")

hist(s2, main = "Chi-Square (df = 10)", xlab = "Value", ylab = "Frequency", col = "lightgreen")

hist(s3, main = "Chi-Square (df = 25)", xlab = "Value", ylab = "Frequency", col = "lightpink")

#Q3

#To generate a vector of 100 values between -6 and 6

x<-seq(-6,6,length.out=100)

#Degrees of freedom

df<-c(1,4,10,30)

colors<-c("blue","pink","orange","green")

#Calculate the t-distribution values

t\_dist\_df1<-dt(x,df[1])

t\_dist\_df2<-dt(x,df[2])

t\_dist\_df3<-dt(x,df[3])

t\_dist\_df4<-dt(x,df[4])

#Plot the comaprision of t-distributions

plot(x, t\_dist\_df4, type="l",xlab="t values", ylab="density", main="Comparision of t-distributions", col= colors[4])

#Add lines for other degrees of freedom

lines(x,t\_dist\_df1, col=colors[1])

lines(x,t\_dist\_df2, col=colors[2])

lines(x,t\_dist\_df3, col=colors[3])

#Add a legend to the plot

legend("topright", legend=c("df=1","df=4","df=10","df=30"))

#Q4

v1<-10

v2<-20

#(i) find the 95th percentile of the f-distribution

percentile\_95<-qf(0.95,df1=v1,df2=v2)

cat("95th percentile of the F-distribution :",percentile\_95,"\n")

#(ii)calculate the area under the curve for the given intervals

area\_interval\_1<-pf(1.5,df1 =v1,df2=v2,lower.tail = TRUE) #[0,1.5]

area\_interval\_2<-1-area\_interval\_1

cat("Area under the curve for [0,1.5]:",area\_interval\_1,"\n")

cat("Area under the curve for [1.5,+inf]:",area\_interval\_2,"\n")

#(iii) calculate the quantiles for diff probab

quantile\_25<-qf(0.25,df1=v1,df2=v2)

quantile\_50<-qf(0.5,df1=v1,df2=v2)

quantile\_75<-qf(0.75,df1=v1,df2=v2)

quantile\_999<-qf(0.999,df1=v1,df2=v2)

cat("Quantile for p = 0.25",quantile\_25,"\n")

cat("Quantile for p = 0.25",quantile\_50,"\n")

cat("Quantile for p = 0.75",quantile\_75,"\n")

cat("Quantile for p = 0.999",quantile\_999,"\n")

#(iv) generate random values and plot a histogram

#set.seed(123) for reproducability

x<-rf(1000,df1=v1,df2=v2)

hist(x,breaks='scott',freq=FALSE,xlim=c(0,3),ylim=c(0,1),xlab="", col="orange")

**EXPERIMENT 8**

# (a) Import the csv data file in R

library(readr)

Clt\_data <- read\_csv("C:/Users/shree/Downloads/Clt-data.csv")

View(Clt\_data)

# (b) Validate data for correctness

# Count number of rows

n\_rows <- nrow(Clt\_data)

print(paste("Number of rows: ", n\_rows))

# View the top ten rows of the dataset

head(Clt\_data, 10)

# Data about the column names

colnames(Clt\_data)

str(Clt\_data)

# (c) Calculate the population mean

population\_mean <- mean(Clt\_data$`Wall Thickness`)

print(population\_mean)

# Plot the histogram of the observations

hist(Clt\_data$`Wall Thickness`, breaks = 20, col = "lightblue", xlab = "Wall Thickness", main = "Histogram of Wall Thickness")

# Add a vertical line for population mean

abline(v = population\_mean, col = "red", lwd = 2)

# Function to draw sufficient samples and plot histograms

draw\_samples\_and\_plot <- function(sample\_size) {

# Generate 1000 samples of specified size

sample\_means <- replicate(1000, mean(sample(Clt\_data$'Wall Thickness', size = sample\_size, replace = TRUE)))

# Plot histogram of sample means

hist(sample\_means, breaks = 30, col = "lightgreen", xlab = "Sample Means", main = paste("Sampling Distribution (Sample Size =", sample\_size, ")"))

# Add a vertical line for the population mean

abline(v = population\_mean, col = "red", lwd = 2)

}

# (a) Draw samples of size 10 and plot their means

draw\_samples\_and\_plot(10)

# (b) Repeat for sample sizes 50, 500, and 9000

draw\_samples\_and\_plot(50)

draw\_samples\_and\_plot(500)

draw\_samples\_and\_plot(9000)