PROBABILITY AND STATISTICS PMA-303

[LAB ASSIGNMENT]

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

SUBMITTED BY: - HRITHIK GUPTA

SUBMITTED TO: - DR. RAJNISH KUMAR RAI

BRANCH: - MSC MATHS AND COMPUTING

ROLL NUMBER: - 302303005

Thapar Institute of Engineering & Technology, Patiala
Department of Mathematics
LAB Experiment 1: Basics of R programming

Q1. Write an R program to generate a sequence of numbers from 1 to 20.

CODE:
```
numbers<-1:20
print(numbers)
```

OUTPUT:
```
> # Ques-1: Write an R program to generate a sequence of numbers from 1 to 20.> numbers<-1:20> print(numbers) [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
```

Q2. Create a vector c = [10, 20, 30, 40, 50, 60] and write a program which returns the maximum and minimum of this vector.

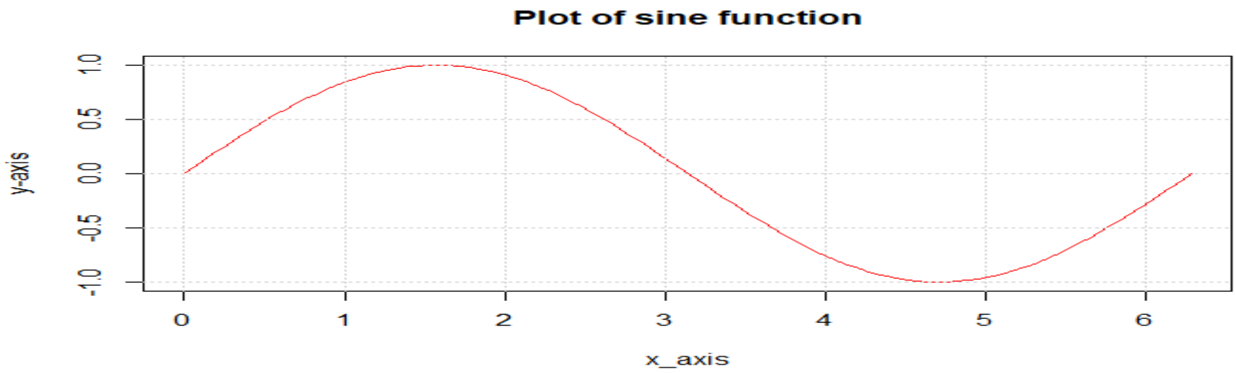
CODE:
```
c<-c(10,20,30,40,50,60)
c_maximum<-max(c)
c_mininmum<-min(c)
cat("Maximum value:",c_maximum," and Minimum value:",c_mininmum)
```

OUTPUT:
```
> # Ques-2: Create a vector c = [10, 20, 30, 40, 50, 60] and write a program which returns the > # maximum and minimum of this vector. > c<-c(10,20,30,40,50,60)> c_maximum<-max(c)> c_mininmum<-min(c)> cat("Maximum value:",c_maximum," and Minimum value:",c_mininmum)Maximum value: 60  and Minimum value: 10
```

Q3. Write an R program to plot a simple line graph of the sin function.

CODE:
```
x<-seq(0,2*pi,length.out=100)
y<-sin(x)
plot(x,y,type='l',col='red',lwd=1,
    main="Plot of sine function",
    xlab="x_axis",ylab="y-axis")
grid()
```

OUTPUT:



Q4. 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:

```
factorial<-function(n){
  if(n==0)
  {
    return(1)
  }else{
    return(n*factorial(n-1))
  }
}
input_number<-as.integer(readline(prompt = "Enter a number:"))
if (input_number<0)
{
  print("Give a valid number")
}else{
  factorial_result=factorial(input_number)
  cat("Factorial:",factorial_result)
}
```

OUTPUT:

```
> # Ques-4: 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.
> factorial<-function(n){
+   if(n==0)
+   {
+     return(1)
+   }else{
+     return(n*factorial(n-1))
+   }
+ }
> input_number<-as.integer(readline(prompt = "Enter a number:"))
Enter a number:8
> if (input_number<0)
+ {
+   print("Give a valid number")
+ }else{
+   factorial_result=factorial(input_number)
+   cat("Factorial:",factorial_result)
+ }
Factorial: 40320
> 
```

Q5. Write an R program to calculate the mean of a given numeric vector.

CODE:

```
vec<-c(10,20,30,40,50)
result<-mean(vec)
cat("Result:",result)
```

OUTPUT:

```
> # Ques-5: Write an R program to calculate the mean of a given numeric vector.> vec<-c
(10,20,30,40,50)>
result<-mean(vec)> cat("Result:",result)Result: 30
```

Q6. Write a program to write first n terms of a Fibonacci sequence. You may take n as an input from the user.

CODE:

```
fibonacci <- function(n) {
  if (n == 0) {
    return(0)  # Fibonacci(0) is 0
  } else if (n == 1) {
    return(1)  # Fibonacci(1) is 1
  } else {
    return(fibonacci(n - 1) + fibonacci(n - 2))  # Recursive case
  }
}

# Take input from user
number <- as.integer(readline(prompt = "Enter a number: "))

# Check for valid input
if (is.na(number) || number < 0) {
  print("Please enter a valid non-negative integer.")  # Error message for invalid input
} else {
  fibonacci_result <- fibonacci(number)  # Calculate Fibonacci
  cat("Fibonacci of", number, "is:", fibonacci_result, "\n")  # Print the result
}
```

OUTPUT:

```
> fibonacci <- function(n) {
+   if (n == 0) {
+     return(0)  # Fibonacci(0) is 0
+   } else if (n == 1) {
+     return(1)  # Fibonacci(1) is 1
+   } else {
+     return(fibonacci(n - 1) + fibonacci(n - 2))  # Recursive case
+   }
+ }
> # Take input from user
> number <- as.integer(readline(prompt = "Enter a number: "))
Enter a number: 5
> # Check for valid input
> if (is.na(number) || number < 0) {
+   print("Please enter a valid non-negative integer.")  # Error message for invalid input
+ } else {
+   fibonacci_result <- fibonacci(number)  # Calculate Fibonacci
+   cat("Fibonacci of", number, "is:", fibonacci_result, "\n")  # Print the result
+ }
Fibonacci of 5 is: 5
> # Take input from user
  number <- as.integer(readline(prompt = "Enter a number: "))
```

Q7. Write an R program to make a simple calculator which can add, subtract, multiply and divide.

CODE:

```
n1<-as.integer(readline(prompt="Enter the first number:"))
oper<-readline(prompt="Enter the operator(+,-,/,*)")
n2<-as.integer(readline(prompt="Enter the second number:"))
result <- switch(oper,
                 "+" = n1 + n2,
                 "-" = n1 - n2,
                 "*" = n1 * n2,
                 "/" = {
                   if (n2 == 0) {
                     return("Error: Division by zero is not allowed.")
                   } else {
                     n1 / n2
                   }
                 },
                 "Invalid operator"  # Default case for invalid operator
)
print(paste("Result:",n1,oper,n2,"=",result))
```

OUTPUT:

```
> # Ques-7: Write an R program to make a simple calculator which can add, subtract, multiply
> # and divide.
> n1<-as.integer(readline(prompt="Enter the first number:"))
Enter the first number:7
> oper<-readline(prompt="Enter the operator(+,-,/,*)")
Enter the operator(+,-,/,*)+
> n2<-as.integer(readline(prompt="Enter the second number:"))
Enter the second number:6
> result <- switch(oper,
+                  "+" = n1 + n2,
+                  "-" = n1 - n2,
+                  "*" = n1 * n2,
+                  "/" = {
+                    if (n2 == 0) {
+                      return("Error: Division by zero is not allowed.")
+                    } else {
+                      n1 / n2
+                    }
+                  },
+                  "Invalid operator"  # Default case for invalid operator
+ )
> print(paste("Result:",n1,oper,n2,"=",result))
[1] "Result: 7 + 6 = 13"
>
```

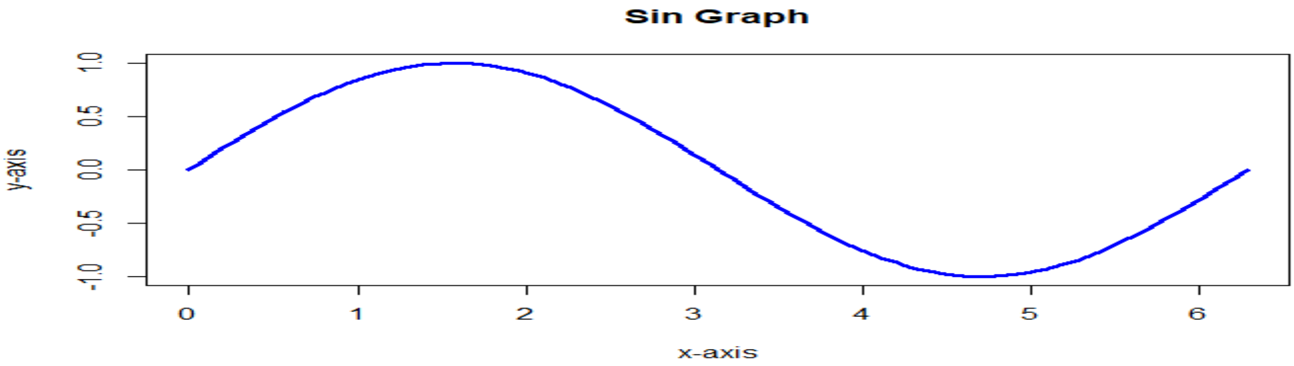
Q8. Explore plot, pie, barplot etc. (the plotting options) which are built-in functions in R.

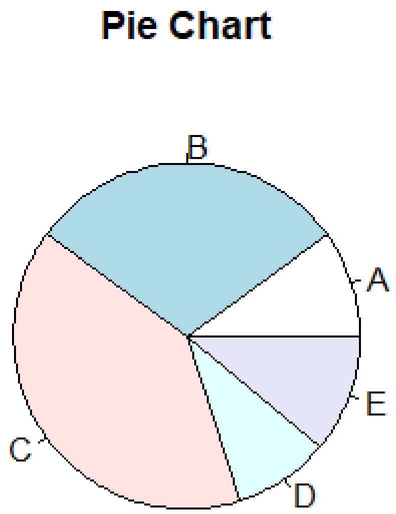
CODE:

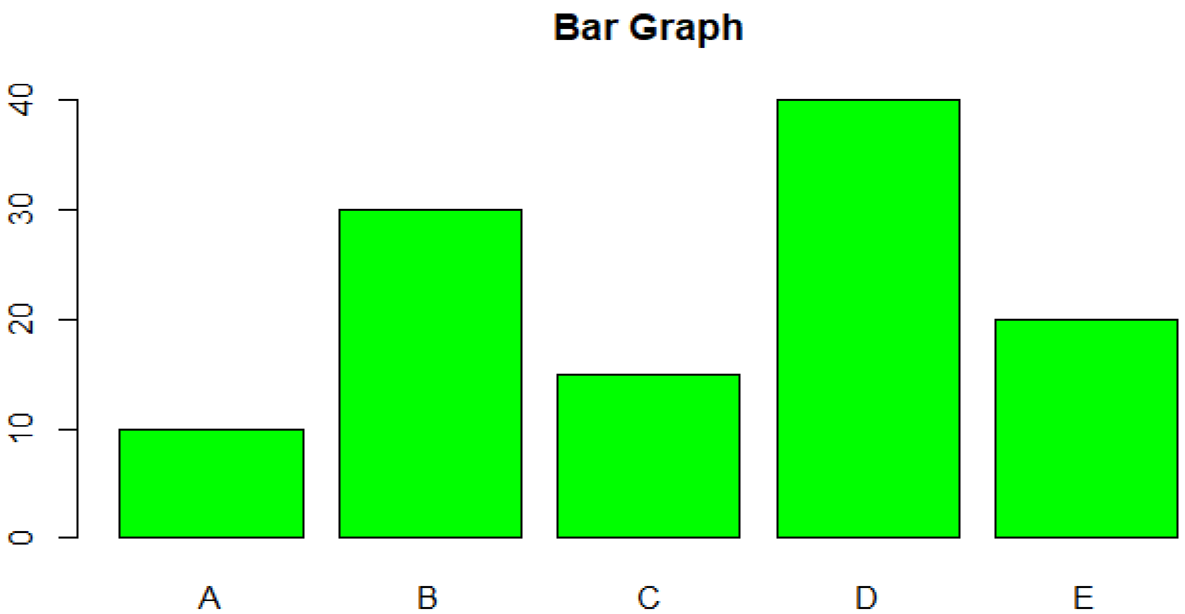
```
x<-seq(0,2*pi,length.out=100)
y<-sin(x)
plot(x,y,type="l",col="blue",lwd=3,main="Sin Graph",xlab="x-axis",ylab="y-axis")

data<-c(10,30,40,9,11)
label<-c("A","B","C","D","E")
```

```
pie(data,labels = label,main="Pie Chart")

value<-c(10,30,15,40,20)
type<-c("A","B","C","D","E")
barplot(value,names.arg=type,main="Bar Graph",col="green")
```

OUTPUT:







Thapar Institute of Engineering & Technology, Patiala
Department of Mathematics
LAB Experiment 2: Sample space, Total probability and Bayes theorem

Q1. The iris data-set is a built-in data-set in R that contains measurements on 4 different attributes (in centimeters) for 150 flowers from 3 different species. Load this data-set and do the following:
(a) Print first few rows of this data-set.
(b) Find the structure of this data-set.
(c) Find the range of the data regarding the sepal length of flowers.
(d) Find the mean of the sepal length.
(e) Find the median of the sepal length.
(f) Find the first and the third quartiles and hence the interquartile range.
(g) Find the standard deviation and variance.
(h) Try doing the above exercises for sepal.width, petal.length and petal.width.
(i) Use the built-in function summary on the data-set Iris.

CODE:
```
data(iris)
head(iris)
str(iris)
# Calculate the range of Sepal.Length
sepal_length_range <- range(iris$Sepal.Length)
# Display the range
print(sepal_length_range)
# Calculate the mean of Sepal.Length
mean_sepal_length <- mean(iris$Sepal.Length)
mean_sepal_length
# Calculate the median of Sepal.Length
median_sepal_length <- median(iris$Sepal.Length)
median_sepal_length
q1<-quantile(iris$Sepal.Length,0.25)
print(q1)
q3<-quantile(iris$Sepal.Length,0.75)
print(q3)
IQR(iris$Sepal.Length)
sd(iris$Sepal.Length)
var(iris$Sepal.Length)
summary(iris)
```

OUTPUT:

```
> head(iris)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> str(iris)
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
> # Calculate the range of Sepal.Length
> sepal_length_range <- range(iris$Sepal.Length)
> # Display the range
> print(sepal_length_range)
[1] 4.3 7.9
> # Calculate the mean of Sepal.Length
> mean_sepal_length <- mean(iris$Sepal.Length)
> mean_sepal_length
[1] 5.843333
> # Calculate the median of Sepal.Length
> median_sepal_length <- median(iris$Sepal.Length)
> median_sepal_length
[1] 5.8
> q1<-quantile(iris$Sepal.Length,0.25)
> print(q1)
25% 
5.1 
> q3<-quantile(iris$Sepal.Length,0.75)
> print(q3)
75% 
6.4 
> IQR(iris$Sepal.Length)
[1] 1.3
> sd(iris$Sepal.Length)
[1] 0.8280661
> var(iris$Sepal.Length)
[1] 0.6856935
> summary(iris)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species  
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100   setosa    :50  
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50  
 Median :5.800   Median :3.000   Median :4.350   Median :1.300   virginica :50  
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199                  
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800                  
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500                  
> 
```

Q2. 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 takes the vector as input and gives the mode value as output.

CODE:

```
calculate_mode <- function(x) {
  # Remove NA values
  x <- na.omit(x)
  # Get the unique values and their frequencies
  unique_values <- unique(x)
  frequencies <- table(x)#table uniquely count the value of vector
  # Find the maximum frequency
  max_freq <- max(frequencies)
  # Get the values that have the maximum frequency (mode)
  modes <- unique_values[frequencies == max_freq]
  return(modes)
}
# Example usage
```

```
data_vector <- c(1, 2, 2, 3, 4, 4, 4, 5, 5)
mode_value <- calculate_mode(data_vector)
# Display the mode
print(mode_value)
```

OUTPUT:

```
> #Q2. 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
> #takes the vector as input and gives the mode value as output.
> # Function to calculate the mode
> calculate_mode <- function(x) {
+   # Remove NA values
+   x <- na.omit(x)
+   # Get the unique values and their frequencies
+   unique_values <- unique(x)
+   frequencies <- table(x)#table uniquely count the value of vector
+   # Find the maximum frequency
+   max_freq <- max(frequencies)
+   # Get the values that have the maximum frequency (mode)
+   modes <- unique_values[frequencies == max_freq]
+   return(modes)
+ }
> # Example usage
> data_vector <- c(1, 2, 2, 3, 4, 4, 4, 5, 5)
> mode_value <- calculate_mode(data_vector)
> # Display the mode
> print(mode_value)
[1] 4
> |
```

Q3. A room has M people, and each has an equal chance of being born on any of the 365 days of the year. (For simplicity, we will ignore leap years). What is the probability that two people in the room have the same birthday ?
(a) Use an R simulation to estimate this for various M.
(b) Find the smallest value of M for which the probability of a match is greater than 0.5.

CODE:

```
M<-50  #out of 5000 we consider 50 persons
N<-5000 #number of simulation total no of persons exhaustive cases
fav<-0  #favourable cases
for(val in 1:N){
  a<-as.integer(any(duplicated(sample(365,M,replace=TRUE))))
  fav<-fav+a
}
prob<-fav/N
print(prob)
M <- 2
prob <- 0  # Initialize probability

while (TRUE) {
  prod <- 1
  for (i in 0:(M - 1)) {
    prod <- prod * (1 - i / 365)
  }

  prob <- 1 - prod  # Probability of at least one match

  if (prob > 0.5) {
```

```
    break  # Exit the loop if probability exceeds 0.5
  }

  M <- M + 1  # Increment M
}

print(M)  # Print the smallest M for which prob > 0.5
```

OUTPUT:

```
> #Q3. A room has M people, and each has an equal chance of being born on any of
> #the 365 days of the year. (For simplicity, we will ignore leap years). What is the
> #probability that two people in the room have the same birthday ?
> #(a) Use an R simulation to estimate this for various M.
> #(b) Find the smallest value of M for which the probability of a match is greater
> #than 0.5.
> M<-50  #out of 5000 we consider 50 persons
> N<-5000 #number of simulation total no of persons exhaustive cases
> fav<-0  #favourable cases
> for(val in 1:N){
+   a<-as.integer(any(duplicated(sample(365,M,replace=TRUE))))
+   fav<-fav+a
+ }
> prob<-fav/N
> print(prob)
[1] 0.9738
> M <- 2
> prob <- 0  # Initialize probability
> while (TRUE) {
+   prod <- 1
+   for (i in 0:(M - 1)) {
+     prod <- prod * (1 - i / 365)
+   }
+
+   prob <- 1 - prod  # Probability of at least one match
+
+   if (prob > 0.5) {
+     break  # Exit the loop if probability exceeds 0.5
+   }
+
+   M <- M + 1  # Increment M
+ }
> print(M)  # Print the smallest M for which prob > 0.5
[1] 23
> |
```

Q4. 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 prob- ability 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:

```
cond_prob<-function(pR,pCR,pC){ #pR=rain, pCR= rain under cloudy, pC=cloudy
  pRC<-(pR*pCR)/pC
  return(pRC)
}
#Define probability
pRain<-0.2
pCloudy<-0.4
pCloudyRain<-0.85
#Use function file
```

```
prob<-cond_prob(pRain,pCloudyRain,pCloudy)
print(prob)
```

OUTPUT:

```
> ##Q4. 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 prob- ability 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?
> cond_prob<-function(pR,pCR,pC){ #pR=rain, pCR= rain under cloudy, pC=cloudy
+   pRC<-(pR*pCR)/pC
+   return(pRC)
+ }
> #Define probability
> pRain<-0.2
> pCloudy<-0.4
> pCloudyRain<-0.85
> #Use function file
> prob<-cond_prob(pRain,pCloudyRain,pCloudy)
> print(prob)
[1] 0.425
```

Q5. Write an R function for computing conditional probability. Call this function to do the following problem: Three urns I, II and III contain 8 red, 4 white; 6 red, 6 white; and 5 red, 7 white balls, respectively. If a ball is drawn at random and found to be red, what is the probability that it is a drawn from (i) urn I, (ii) urn III?

CODE:

```
cond_prob<-function(PA, PE_given_A, PE) {
  PA_given_E <- (PA*PE_given_A)/PE
  return(PA_given_E)
}
# Problem setup
# Urn probabilities (assuming each urn is equally
#likely to be chosen)
P_urn_I<-1/3
P_urn_II<-1/3
P_urn_III<-1/3
# Probability of drawing a red ball given the urn
P_red_given_urn_I<-8/(8 + 4)
P_red_given_urn_II<-6/(6 + 6)
P_red_given_urn_III<-5/(5 + 7)
# Total probability of drawing a red ball
P_red <- (P_urn_I*P_red_given_urn_I) +
  (P_urn_II*P_red_given_urn_II) +
  (P_urn_III*P_red_given_urn_III)
# Calculate the conditional probabilities
P_urn_I_given_red <- cond_prob(P_urn_I, P_red_given_urn_I, P_red)
P_urn_III_given_red <- cond_prob(P_urn_III, P_red_given_urn_III, P_red)
# Print the results
cat("The probability that the red ball was drawn from urn I is:", P_urn_I_given_red,
    "\n")
cat("The probability that the red ball was drawn from urn III is:", P_urn_III_given_red,
    "\n")
```

OUTPUT:

```
> # Define the function for computing conditional probability
> cond_prob<-function(PA, PE_given_A, PE) {
+   PA_given_E <- (PA*PE_given_A)/PE
+   return(PA_given_E)
+ }
> # Problem setup
> # Urn probabilities (assuming each urn is equally
> #likely to be chosen)
> P_urn_I<-1/3
> P_urn_II<-1/3
> P_urn_III<-1/3
> # Probability of drawing a red ball given the urn
> P_red_given_urn_I<-8/(8 + 4)
> P_red_given_urn_II<-6/(6 + 6)
> P_red_given_urn_III<-5/(5 + 7)
> # Total probability of drawing a red ball
> P_red <- (P_urn_I*P_red_given_urn_I) +
+   (P_urn_II*P_red_given_urn_II) +
+   (P_urn_III*P_red_given_urn_III)
> # Calculate the conditional probabilities
> P_urn_I_given_red <- cond_prob(P_urn_I, P_red_given_urn_I, P_red)
> P_urn_III_given_red <- cond_prob(P_urn_III, P_red_given_urn_III, P_red)
> # Print the results
> cat("The probability that the red ball was drawn from urn I is:", P_urn_I_given_red,
+     "\n")
The probability that the red ball was drawn from urn I is: 0.4210526
> cat("The probability that the red ball was drawn from urn III is:", P_urn_III_given_red,
+     "\n")
The probability that the red ball was drawn from urn III is: 0.2631579
> |
```

Q6. Write an R function for computing conditional probability. Call this function to do the following problem: Companies A, B and C produces cars. The production capacity of company A is twice that of B while company B and C produces same number of cars in a given period. It is known that 2% of A, 3% of B and 4% of C are defective. All the cars produced are put into one showroom and then one car is chosen at random.
(a) Find the probability that the car is defective.
(b) Suppose a car chosen is defective, what is the probability that this is produced by company A?

CODE:

```
cond_prob<-function(P_A, PE_given_A, PE) {
  PA_given_E <- (P_A*PE_given_A)/PE
  return(PA_given_E)
}
P_A<-2/4
P_B<-1/4
P_C<-1/4
P_defective_given_A<-0.02
P_defective_given_B<-0.03
P_defective_given_C<-0.04
p_defective<-(P_A*P_defective_given_A)+
  (P_B*P_defective_given_B)+
  (P_C*P_defective_given_C)
cat("Probability that car is defective: ",p_defective,"\n")
P_A_given_defective<-cond_prob(P_A,P_defective_given_A,p_defective)
cat("Probability that car is defective given car is A : ",P_A_given_defective,"\n")
```

OUTPUT:

```
> #Q6. Write an R function for computing conditional probability. Call this function to
> #do the following problem: Companies A, B and C produces cars. The production
> #capacity of company A is twice that of B while company B and C produces same
> #number of cars in a given period. It is known that 2% of A, 3% of B and 4% of C
> #are defective. All the cars produced are put into one showroom and then one car
> #is chosen at random.
> #(a) Find the probability that the car is defective.
> #(b) Suppose a car chosen is defective, what is the probability that this is produced
> #by company A?
> cond_prob<-function(P_A, PE_given_A, PE) {
+   PA_given_E <- (P_A*PE_given_A)/PE
+   return(PA_given_E)
+ }
> P_A<-2/4
> P_B<-1/4
> P_C<-1/4
> P_defective_given_A<-0.02
> P_defective_given_B<-0.03
> P_defective_given_C<-0.04
> p_defective<-(P_A*P_defective_given_A)+
+   (P_B*P_defective_given_B)+
+   (P_C*P_defective_given_C)
> cat("Probability that car is defective: ",p_defective,"\n")
Probability that car is defective:  0.0275
> P_A_given_defective<-cond_prob(P_A,P_defective_given_A,p_defective)
> cat("Probability that car is defective given car is A : ",P_A_given_defective,"\n")
Probability that car is defective given car is A :  0.3636364
>
```

Q7. Write an R function for computing conditional probability. Call this function to do the following problem: The LED bulbs producing factories A, B and C supply LED bulbs to the market in the ratio 2:3:5. It is found that 1% of the items produced in factory A, 2% of the items produced in factory B and 3% of the items produced in factory C are defective. If a bulb is selected at random from the market what is the probability that it is a defective one? Also, if a randomly selected bulb is found to be defective then find the probability that it was produced by factory (i) A, (ii) B, (iii) C?

CODE:

```
cond_prob<-function(P_A, PE_given_A, PE) {
  PA_given_E <- (P_A*PE_given_A)/PE
  return(PA_given_E)
}
P_A<-2/10
P_B<-3/10
P_C<-5/10
P_defective_given_A<-0.01
P_defective_given_B<-0.02
P_defective_given_C<-0.03
p_defective<-(P_A*P_defective_given_A)+
  (P_B*P_defective_given_B)+
  (P_C*P_defective_given_C)
cat("Probability that car is defective: ",p_defective,"\n")
P_A_given_defective<-cond_prob(P_A,P_defective_given_A,p_defective)
```

```
cat("Probability that car is defective given car is A : ",P_A_given_defective,"\n")
P_B_given_defective<-cond_prob(P_B,P_defective_given_B,p_defective)
cat("Probability that car is defective given car is B : ",P_B_given_defective,"\n")
P_C_given_defective<-cond_prob(P_C,P_defective_given_C,p_defective)
cat("Probability that car is defective given car is C : ",P_C_given_defective,"\n")
```

OUTPUT:

```
> cond_prob<-function(P_A, PE_given_A, PE) {
+   PA_given_E <- (P_A*PE_given_A)/PE
+   return(PA_given_E)
+ }
> P_A<-2/10
> P_B<-3/10
> P_C<-5/10
> P_defective_given_A<-0.01
> P_defective_given_B<-0.02
> P_defective_given_C<-0.03
> p_defective<-(P_A*P_defective_given_A)+
+   (P_B*P_defective_given_B)+
+   (P_C*P_defective_given_C)
> cat("Probability that car is defective: ",p_defective,"\n")
Probability that car is defective:  0.023
> P_A_given_defective<-cond_prob(P_A,P_defective_given_A,p_defective)
> cat("Probability that car is defective given car is A : ",P_A_given_defective,"\n")
Probability that car is defective given car is A :  0.08695652
> P_B_given_defective<-cond_prob(P_B,P_defective_given_B,p_defective)
> cat("Probability that car is defective given car is B : ",P_B_given_defective,"\n")
Probability that car is defective given car is B :  0.2608696
> P_C_given_defective<-cond_prob(P_C,P_defective_given_C,p_defective)
> cat("Probability that car is defective given car is C : ",P_C_given_defective,"\n")
Probability that car is defective given car is C :  0.6521739
```

Thapar Institute of Engineering & Technology, Patiala
Department of Mathematics
LAB Experiment 3: Mathematical Expectation, Moments and Functions of Random Variables

Q1. The probability distribution of X, the number of imperfections per 10 meters of a synthetic fabric in continuous rolls of uniform width, is given as

X	0	1	2	3	4
P(X = x)	0.41	0.37	0.16	0.05	0.01

Use R-code to find the average number of imperfections per 10 meters of this fabric. (Try function sum(), weight.mean(), c(a %*% b) to find expected value/mean. Further use it to find E(X2), E(2X + 3), V ar(2X − 3).

CODE:
```
X<-c(0,1,2,3,4)
P_X<-c(0.41,0.37,0.16,0.05,0.01)
#Calculate the expected value of E(X)
E_X<-sum(X*P_X)
cat("E(X):",E_X,"/n")
#OR
Exp_Val<-weighted.mean(X,P_X)
print(Exp_Val)
#OR
Exp_Val<-c(X %*% P_X)  # %*% can be used for calculating sum of
# element wise product between two vectors
print(Exp_Val)

#Calculate E(X^2)
E_X2<-sum((X^2)*P_X)
cat("E(X^2): ",E_X2,"\n")

#Calculte E(2X + 3)
E_2X_plus_3<-2*E_X+3
cat("E(2X+3): ",E_2X_plus_3,"\n")

#Calculate the variance Var(X)
Var_X<-E_X2-E_X^2
cat("Var(X):",Var_X,"\n")

#Calculate the variance Var(2X - 3)
Var_2X_minus_3<-4*Var_X
cat("Var(2X - 3):",Var_2X_minus_3,"\n")
```

OUTPUT:

```
> #Ques-1
> #Define the values of X and their corresponding prob
> X<-c(0,1,2,3,4)
> P_X<-c(0.41,0.37,0.16,0.05,0.01)
> #Calculate the expected value of E(X)
> E_X<-sum(X*P_X)
> cat("E(X):",E_X,"/n")
E(X): 0.88 /n
> #OR
> Exp_Val<-weighted.mean(X,P_X)
> print(Exp_Val)
[1] 0.88
> #OR
> Exp_Val<-c(X %*% P_X)  # %*% can be used for calculating sum of
> # element wise product between two vectors
> print(Exp_Val)
[1] 0.88
> #Calculate E(X^2)
> E_X2<-sum((X^2)*P_X)
> cat("E(X^2): ",E_X2,"\n")
E(X^2):  1.62
> #Calculte E(2X + 3)
> E_2X_plus_3<-2*E_X+3
> cat("E(2X+3): ",E_2X_plus_3,"\n")
E(2X+3):  4.76
> #Calculate the variance Var(X)
> Var_X<-E_X2-E_X^2
> cat("Var(X):",Var_X,"\n")
Var(X): 0.8456
> #Calculate the variance Var(2X - 3)
> Var_2X_minus_3<-4*Var_X
> cat("Var(2X - 3):",Var_2X_minus_3,"\n")
Var(2X - 3): 3.3824
```

Q2. A bookstore purchases three copies of a book at $12 each and sells them for $24 each. Unsold copies are returned for $4 each. Let X = number of copies sold and Y = Net revenue. If the probability mass function of X is

X	**0**	**1**	**2**	**3**
P(X=x)	**0.1**	**0.2**	**0.2**	**0.5**

Use R-code to find the expected value and variance of Y

CODE:
```
#Revenue Y=h(x)= 24X+4(3-X)-36 = 20X-24
x<-c(0,1,2,3)
probx<-c(0.1,0.2,0.2,0.5)
y<-20*x-24
proby<-probx #E(phi(x))=sum phi(x)  P(X=x)=sum(10X-12) P(X=x)

E_Y<-sum(y*proby)
cat("E(Y): ",E_Y,"\n")

#Calculate the expected value of Y^2
E_Y2<-sum((y^2)*proby)
cat("E_Y^2: ",E_Y2,"\n")

#Calculate the variance of Y
Var_Y<-E_Y2-E_Y^2
cat("Var(Y): ",Var_Y,"\n")
```
OUTPUT:
```
> #Ques-2:
> #Revenue Y=h(x)= 24X+4(3-X)-36 = 20X-24
> x<-c(0,1,2,3)
> probx<-c(0.1,0.2,0.2,0.5)
> y<-20*x-24
> proby<-probx #E(phi(x))=sum phi(x)  P(X=x)=sum(10X-12) P(X=
> E_Y<-sum(y*proby)
> cat("E(Y): ",E_Y,"\n")
E(Y):  18
> #Calculate the expected value of Y^2
> E_Y2<-sum((y^2)*proby)
> cat("E_Y^2: ",E_Y2,"\n")
E_Y^2:  760
> #Calculate the variance of Y
> Var_Y<-E_Y2-E_Y^2
> cat("Var(Y): ",Var_Y,"\n")
Var(Y):  436
> |
```

Q3. Following is the cumulative probability distribution function of a discrete random
variable X:

X	-3	-1	0	1	2	3	5	8
F(x)	0.10	.30	.45	.5	.75	.90	.95	1

(a) Write R code to find the probability mass function of X.
(b) Write R code to find P(X = Even), P(1 ≤ X ≤ 8) and P(X ≥ 3|X > 0).

CODE:
```
#QUES-3:
```

```
x<-c(-3,-1,0,1,2,3,5,8)
cdf_x<-c(0.10,0.30,0.45,0.5,0.75,0.90,0.95,1.0)
i=0
while(i<8){
 pdf_x<-c(cdf_x[i+1]-cdf_x[i])
 i=i+1
 print(pdf_x)
}
for(j in 1:8){
if(x[j]%%2==0){
prob=cdf_x[j]-cdf_x[j-1]
cat("P( X =",x[j],")","=",prob,"\n")
}
}
```

OUTPUT:

```
> #QUES-3:
> x<-c(-3,-1,0,1,2,3,5,8)
> cdf_x<-c(0.10,0.30,0.45,0.5,0.75,0.90,0.95,1.0)
> i=0
> while(i<8){
+  pdf_x<-c(cdf_x[i+1]-cdf_x[i])
+  i=i+1
+  print(pdf_x)
+ }
numeric(0)
[1] 0.2
[1] 0.15
[1] 0.05
[1] 0.25
[1] 0.15
[1] 0.05
[1] 0.05
> for(j in 1:8){
+ if(x[j]%%2==0){
+ prob=cdf_x[j]-cdf_x[j-1]
+ cat("P( X =",x[j],")","=",prob,"\n")
+ }
+ }
P( X = 0 ) = 0.15
P( X = 2 ) = 0.25
P( X = 8 ) = 0.05
```

Q4. The time T, in days, required for the completion of a contracted project is a random
variable with probability density function

$$f(t) = \begin{cases} 0.2\, e^{-0.2t}, & \text{for } t > 0; \\ 0, & \text{otherwise.} \end{cases}$$

Use R-code to find the expected value and variance of T & 2T − 3. Use function integrate() to find the expected value and variance of continuous random variable T & 2T − 3.

CODE:

```
f1<-function(t){0.2*t*exp(-0.2*t)
  }
Exp_t<-integrate(f1,lower=0,upper=Inf)$value
print(Exp_t)
#OR:
f_T<-function(t){
  ifelse(t>0,0.2*t*exp(-0.2*t),0)
}
Exp_t_f_T<-integrate(function(t) t*f_T(t),lower=0,upper=Inf)$value
print(Exp_t_f_T)

#E(2t-3)
Exp_2t_3<-2*Exp_t-3
print(Exp_2t_3)

#E(t^2)
f2<-function(t)((0.2*t*t)*exp(-0.2*t))
Exp_t_2<-integrate(f2,lower=0,upper=Inf)$value
print(Exp_t_2)

#var(t)
Var_T<-Exp_t_2-Exp_t^2
print(Var_T)

#var(2t-3)
Var_2T_3<-4*(Exp_t_2-Exp_t^2)
print(Var_2T_3)
```

OUTPUT:
```
> #QUES-4:
> #E(t)
> f1<-function(t){0.2*t*exp(-0.2*t)
+   }
> Exp_t<-integrate(f1,lower=0,upper=Inf)$value
> print(Exp_t)
[1] 5
> #OR:
> f_T<-function(t){
+   ifelse(t>0,0.2*t*exp(-0.2*t),0)
+ }
> Exp_t_f_T<-integrate(function(t) t*f_T(t),lower=0,upper=Inf)$value
> print(Exp_t_f_T)
[1] 50
> #E(2t-3)
> Exp_2t_3<-2*Exp_t-3
> print(Exp_2t_3)
[1] 7
> #E(t^2)
> f2<-function(t)((0.2*t*t)*exp(-0.2*t))
> Exp_t_2<-integrate(f2,lower=0,upper=Inf)$value
> print(Exp_t_2)
[1] 50
> #var(t)
> Var_T<-Exp_t_2-Exp_t^2
> print(Var_T)
[1] 25
> #var(2t-3)
> Var_2T_3<-4*(Exp_t_2-Exp_t^2)
> print(Var_2T_3)
[1] 100
> 
```

Q5. Write R-code to find the first and second moments about the origin of the random variable X with probability density function

$$f(x) = \begin{cases} 3e^{-3x}, & x > 0; \\ 0, & \text{otherwise.} \end{cases}$$

Further use the results to find mean and variance of X and 2X + 3.

CODE:
```
f1<-function(x)(3*x*exp(-3*x))
Exp_t<-integrate(f1,0,Inf)$value
print(Exp_t)
#Mean(2x+3)
Exp_2x_plus_3<-2*Exp_t+3
print(Exp_2x_plus_3)
#Second moment of X
f2<-function(x)(3*x*x*exp(-3*x))
Exp_t_2<-integrate(f2,0,Inf)$value
print(Exp_t_2)
#VAR(X)
Var_x<-Exp_t_2-Exp_t^2
print(Var_x)
#VAR(2x+3)
```

```
Var_2x_plus_3<-4*Var_x
print(Var_x)
```

OUTPUT:

```
> #Ques-5:
> #Mean(x)
> f1<-function(x)(3*x*exp(-3*x))
> Exp_t<-integrate(f1,0,Inf)$value
> print(Exp_t)
[1] 0.3333333
> #Mean(2x+3)
> Exp_2x_plus_3<-2*Exp_t+3
> print(Exp_2x_plus_3)
[1] 3.666667
> #Second moment of X
> f2<-function(x)(3*x*x*exp(-3*x))
> Exp_t_2<-integrate(f2,0,Inf)$value
> print(Exp_t_2)
[1] 0.2222222
> #VAR(X)
> Var_x<-Exp_t_2-Exp_t^2
> print(Var_x)
[1] 0.1111111
> #VAR(2x+3)
> Var_2x_plus_3<-4*Var_x
> print(Var_x)
[1] 0.1111111
>
```

Q7. Two unbiased dice are thrown. Find the expected value and the variance of the sum of number of points on both.

CODE:

```
sums<-2:12
#prob for each sum
probs<-c(1,2,3,4,5,6,5,4,3,2,1)/36
#expected value
E_X<-sum(sums*probs)
print(E_X)
#E(x2)
E_X2<-sum(sums^2*probs)
print(E_X2)
#variance calculate
var<-E_X2-(E_X)^2
print(var)
```

OUTPUT:

```
> #ques 7
> #possible sum
> sums<-2:12
> #prob for each sum
> probs<-c(1,2,3,4,5,6,5,4,3,2,1)/36
> #expected value
> E_X<-sum(sums*probs)
> print(E_X)
[1] 7
> #E(x2)
> E_X2<-sum(sums^2*probs)
> print(E_X2)
[1] 54.83333
> #variance calculate
> var<-E_X2-(E_X)^2
> print(var)
[1] 5.833333
>
```

Q8. Let X be a geometric random variable with probability distribution

$$f(x) = \frac{3}{4}\left(\frac{1}{4}\right)^{x-1}, x = 1, 2, 3, \ldots$$

Write a function to find the probability distribution of the random variable Y = X2 and find probability of Y for X = 3. Further, use it to find the expected value and variance of Y for X = 1, 2, 3, 4, 5.

CODE:

```
func<-function(y){
  (3/4)*(1/4)^(sqrt(y)-1)
}
x<-as.integer(readline(prompt="Enter value of X : "))
y<-x^2
proby<-func(y)
print(proby)

#to find exp value and variance
x<-c(1,2,3,4,5)
y<-x^2
proby<-func(y)
print(proby)
#exp Y
E_Y<-sum(y*proby)
#exp Y2
E_Y2<-sum(y^2*proby)
var<-E_Y2-(E_Y)^2
print(var)
```

OUTPUT:

```
> #ques 8
> func<-function(y){
+   (3/4)*(1/4)^(sqrt(y)-1)
+ }
> x<-as.integer(readline(prompt="Enter value of X : "))
Enter value of X : y<-x^2
Warning message:
NAs introduced by coercion
> proby<-func(y)
Warning message:
In sqrt(y) : NaNs produced
> print(proby)
[1]          NaN          NaN 0.0117187500 0.0007324219
> #to find exp value and variance
> x<-c(1,2,3,4,5)
> y<-x^2
> proby<-func(y)
> print(proby)
[1] 0.750000000 0.187500000 0.046875000 0.011718750 0.002929688
> #exp Y
> E_Y<-sum(y*proby)
> #exp Y2
> E_Y2<-sum(y^2*proby)
> var<-E_Y2-(E_Y)^2
> print(var)
[1] 7.614112
```

Thapar Institute of Engineering & Technology, Patiala
Department of Mathematics
LAB Experiment 4: Discrete probability distribution

Q1. Roll 12 dice simultaneously, and let X denotes the number of 6’s that appear. (Try using the function pbinom & dbinom; 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). Write R-code to find probability of getting

(a) 8, 6’s i.e., $P(X = 8)$,

(b) at most 7, 6’s i.e., $P(X \leq 7)$,

(c) more than 6, 6’s i.e., $P(X > 6)$,

(d) getting 7, 8 or 9, 6’s, i.e., $P(7 \leq X \leq 9)$,

(e) getting 7 or 8, 6’s, i.e., $P(7 \leq X < 9)$.

CODE:

```
bino_pmf<-function(x,n,p){
        P_X<-choose(n,x)*p^(x)*(1-p)^(n-x)
        return(P_X)
}
#a=8
n<-12
p<-1/6
P_8<-bino_pmf(8,n,p)
print(P_8)
P_8<-dbinom(8,n,p)
print(P_8)
############## part b ######################
n<-12
p<-1/6
P_X_Lesser_than_7<-sum(bino_pmf(0:7,n,p))
print(P_X_Lesser_than_7)
#c part
p_x_greater_6<-1-sum(bino_pmf(0:6,n,p))
print(p_x_greater_6)
p_x_greater_6<-pbinom(6,size=12,prob=1/6,lower.tail=F)
print(p_x_greater_6)
####### part d ######
p<- pbinom(9,size=12,prob=1/6) - pbinom(6,size=12,prob=1/6)
print(p)
p<-sum(bino_pmf(0:9,12,1/6))-sum(bino_pmf(0:6,12,1/6))
print(p)
###### part e #####
p<- pbinom(8,size=12,prob=1/6) - pbinom(6,size=12,prob=1/6)
print(p)
```

OUTPUT:

```
> #que 1
> #X~B(n,p)=B(n=12,p=1/6)
> bino_pmf<-function(x,n,p){
+       P_X<-choose(n,x)*p^(x)*(1-p)^(n-x)
+       return(P_X)
+ }
> #a=8
> n<-12
> p<-1/6
> P_8<-bino_pmf(8,n,p)
> print(P_8)
[1] 0.0001421249
> P_8<-dbinom(8,n,p)
> print(P_8)
[1] 0.0001421249
> ############## part b #####################
> n<-12
> p<-1/6
> P_X_Lesser_than_7<-sum(bino_pmf(0:7,n,p))
> print(P_X_Lesser_than_7)
[1] 0.9998445
> #c part
> p_x_greater_6<-1-sum(bino_pmf(0:6,n,p))
> print(p_x_greater_6)
[1] 0.001292544
> p_x_greater_6<-pbinom(6,size=12,prob=1/6,lower.tail=F)
> print(p_x_greater_6)
[1] 0.001292544
> ####### part d ######
> p<- pbinom(9,size=12,prob=1/6) - pbinom(6,size=12,prob=1/6)
> print(p)
[1] 0.001291758
> p<-sum(bino_pmf(0:9,12,1/6))-sum(bino_pmf(0:6,12,1/6))
> print(p)
[1] 0.001291758
> ###### part e #####
> p<- pbinom(8,size=12,prob=1/6) - pbinom(6,size=12,prob=1/6)
> print(p)
[1] 0.001279124
> 
```

Q2. 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. Write R-code to find

(a) How is X distributed?

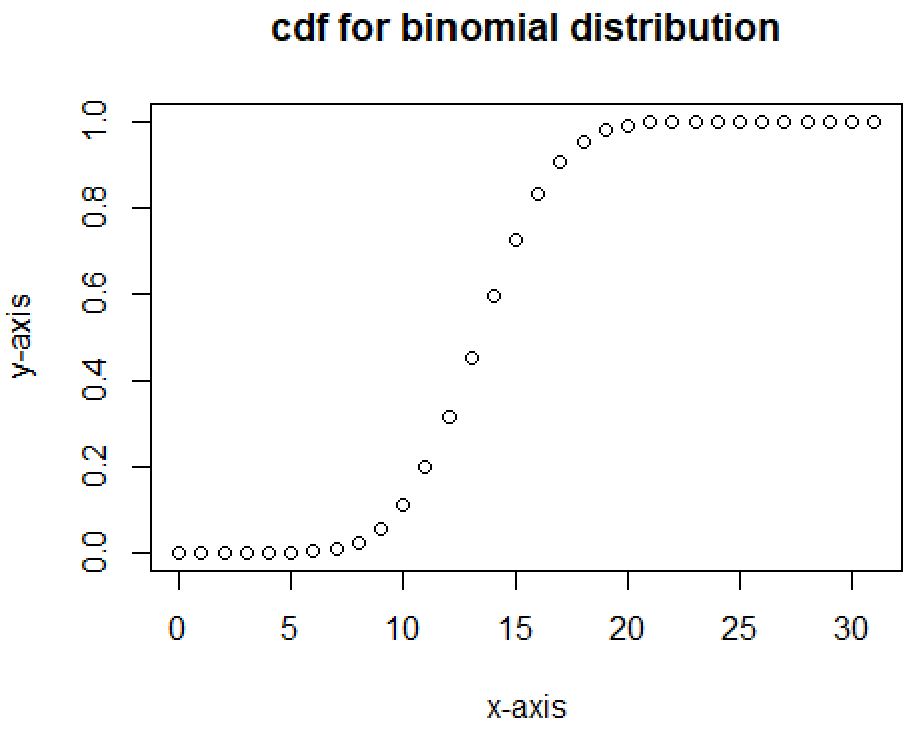
(b) Sketch the probability mass function.

(c) Sketch the cumulative distribution function.

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

```
CODE:
n<-31
p<-0.447
x<-0:n
pmf<-dbinom(x,n,p)
print(pmf)
plot(x,pmf,xlab='x-axis',ylab='y-axis',
    main='Pmf for binomial distribution')
cdf<-pbinom(x,n,p)
print(cdf)
plot(x,cdf,xlab='x-axis',ylab='y-axis',
    main='cdf for binomial distribution')
E_X<-sum(x*pmf)
print(E_X)
E_X_2<-sum(x*x*pmf)
print(E_X_2)
V_X<-E_X_2-(E_X)^2
print(V_X)
S_D<-sqrt(V_X)
print(S_D)
```

OUTPUT:



```
> #Ques-2:
> n<-31
> p<-0.447
> x<-0:n
> pmf<-dbinom(x,n,p)
> print(pmf)
 [1] 1.057984e-08 2.651082e-07 3.214377e-06 2.511632e-05 1.421138e-04 6.203153e-04 2.172786e-03 6.272510e-03
 [9] 1.521055e-02 3.142047e-02 5.587504e-02 8.622373e-02 1.161604e-01 1.372305e-01 1.426190e-01 1.306524e-01
[17] 1.056088e-01 7.532248e-02 4.735464e-02 2.618995e-02 1.270189e-02 5.378041e-03 1.975986e-03 6.250013e-04
[25] 1.684000e-04 3.811382e-05 7.109560e-06 1.064220e-06 1.228898e-07 1.027594e-08 5.537484e-10 1.443887e-11
> plot(x,pmf,xlab='x-axis',ylab='y-axis',
+      main='Pmf for binomial distribution')
> cdf<-pbinom(x,n,p)
> print(cdf)
 [1] 1.057984e-08 2.756880e-07 3.490065e-06 2.860638e-05 1.707202e-04 7.910356e-04 2.963822e-03 9.236332e-03
 [9] 2.444689e-02 5.586736e-02 1.117424e-01 1.979661e-01 3.141265e-01 4.513570e-01 5.939760e-01 7.246284e-01
[17] 8.302372e-01 9.055597e-01 9.529143e-01 9.791043e-01 9.918062e-01 9.971842e-01 9.991602e-01 9.997852e-01
[25] 9.999536e-01 9.999917e-01 9.999988e-01 9.999999e-01 1.000000e+00 1.000000e+00 1.000000e+00 1.000000e+00
> plot(x,cdf,xlab='x-axis',ylab='y-axis',
+      main='cdf for binomial distribution')
> E_X<-sum(x*pmf)
> print(E_X)
[1] 13.857
> E_X_2<-sum(x*x*pmf)
> print(E_X_2)
[1] 199.6794
> V_X<-E_X_2-(E_X)^2
> print(V_X)
[1] 7.662921
> S_D<-sqrt(V_X)
> print(S_D)
[1] 2.768198
>
```

Q3. If the probability that an individual suffer a bad reaction from an injection of a given serum is 0.001. Write R-code to determine the probability that out of 2000 individuals,
(a) exactly 3; and
(b) more than two individuals will suffer a bad reaction.

CODE:

```
n<-2000
p<-0.001
lambda<-n*p
x<-3
P_At_X_3<-dpois(x,lambda)
print(P_At_X_3)

#P(X>2)=1-P(X<=2)=1-F(2)
P_Atleast_2<-1-ppois(2,lambda)
print(P_Atleast_2)
```

OUTPUT:

```
> #QUES_3:
> n<-2000
> p<-0.001
> lambda<-n*p
> x<-3
> P_At_X_3<-dpois(x,lambda)
> print(P_At_X_3)
[1] 0.180447
```

Q4. A space craft has 100, 000 components. The probability of any one component being defective is 2×10−5.The mission will be in danger if five or more components become defective. Write R-code to find the probability of such an event.

CODE:

```
n<-100000
p<-0.00002
lambda<-n*p
#P(X>=5)=1-P(X<5)
P_Atleast_5<-1-ppois(4,lambda)
print(P_Atleast_5)
```

OUTPUT:

```
> #QUes-4:
> n<-100000
> p<-0.00002
> lambda<-n*p
> #P(X>=5)=1-P(X<5)
> P_Atleast_5<-1-ppois(4,lambda)
> print(P_Atleast_5)
[1] 0.05265302
```

Q5. 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 carwash 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). Write R-code to find the probability that there are between 48 and 50 customers, inclusive?

CODE:

```
lambda<-5
x<-0
P_X_0<-dpois(x,lambda)
print(P_X_0)

lambda<-50
#P(48<X<50)=F(49)-F(48)
P_48_X_50<-ppois(49,lambda)-ppois(48,lambda)
print(P_48_X_50)
```

OUTPUT:

```
> #Ques-5:
> lambda<-5
> x<-0
> P_X_0<-dpois(x,lambda)
> print(P_X_0)
[1] 0.006737947
> lambda<-50
> #P(48<X<50)=F(49)-F(48)
> P_48_X_50<-ppois(49,lambda)-ppois(48,lambda)
> print(P_48_X_50)
[1] 0.05632501
>
```

Q6. Suppose that a trainee solder shoots a target according to geometric distribution. If probability that a target hits in any shot is 0.6. Write R-code to find the probability that it takes an odd number of shots.

CODE:

```
p<-0.6
x<-seq(0,100,by=2)
P_X_0<-sum(dgeom(x,p))
print(P_X_0)
```

OUTPUT:

```
> #Ques-6:
> p<-0.6
> x<-seq(0,100,by=2)
> P_X_0<-sum(dgeom(x,p))
> print(P_X_0)
[1] 0.7142857
>
```

Q7. A boy is throwing stones at a target, Write R-code to find the probability that his 10^{th} throw is his 5^{th} hit, if the probability of hitting the target at any trial is 0.05.

CODE:

```
p<-0.05
x<-10
r<-5
P_X_5th_hit<-dnbinom(x,r,p)
print(P_X_5th_hit)
```

OUTPUT:

```
> #Ques-7:
> p<-0.05
> x<-10
> r<-5
> P_X_5th_hit<-dnbinom(x,r,p)
> print(P_X_5th_hit)
[1] 0.0001872924
```

Thapar Institute of Engineering & Technology, Patiala
Department of Mathematics
LAB Experiment 5: Continuous probability distribution

Q1. If X is uniformly distributed over the interval [−2, 2], write R-code to find:
(i) P(X < 0)
(ii) P|X − 1| ≥ ½
using cumulative distribution function (C.D.F.) approach.

CODE:
```
a<--2
b<-2
print(punif(0,min=a,max=b))
prob<-1-(punif(3/2,min=a,max=b)-punif(1/2,min=a,max=b))
print(prob)
```

OUTPUT:
```
> #Ques-1:
> a<--2
> b<-2
> print(punif(0,min=a,max=b))
[1] 0.5
> prob<-1-(punif(3/2,min=a,max=b)-punif(1/2,min=a,max=b))
> print(prob)
[1] 0.75
> |
```

Q2. 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). Write R-code to find the probability that
(i) waiting time is more than 35 minutes
(ii) waiting time lies between 10 and 25 minutes.

CODE:
```
#P(X>35)=1-P(X<=35)=1-F(35)
prob1<-1-punif(35,min=0,max=60)
print(prob1)
#P(10<X<25)=F(25)-F(10)
prob2<-punif
```

OUTPUT:
```
> #Ques-2:
> #P(X>35)=1-P(X<=35)=1-F(35)
> prob1<-1-punif(35,min=0,max=60)
> print(prob1)
[1] 0.4166667
> #P(10<X<25)=F(25)-F(10)
> prob2<-punif(25,min=0,max=60)-punif(10,min=0,max=60)
> print(prob2)
[1] 0.25
```

Q3. The time (in hours) required to repair a machine is an exponential distributed random variable with parameter λ = 1/2.
(i) Find the value of density function at x = 4.
(ii) Plot the graph of exponential probability distribution for 0 ≤ x ≤ 10.
(iii) Find the probability that a repair time takes at most 4 hours.
(iv) Plot the graph of cumulative exponential probabilities for 0 ≤ x ≤ 10.
(v) Simulate 1000 exponential distributed random numbers with λ = 1/2 and plot the simulated data.

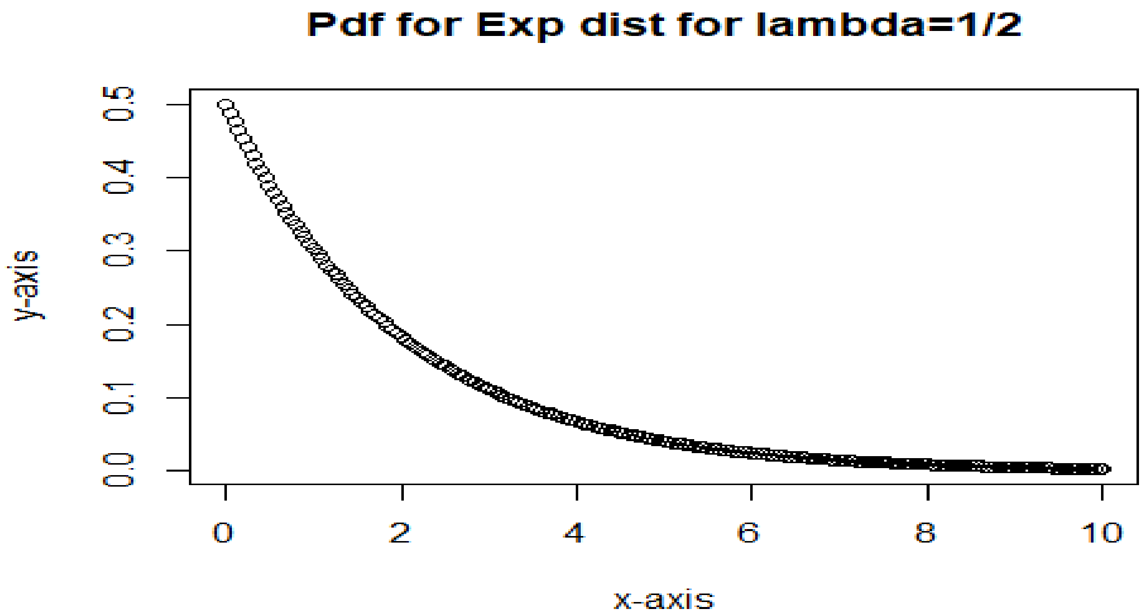
CODE:

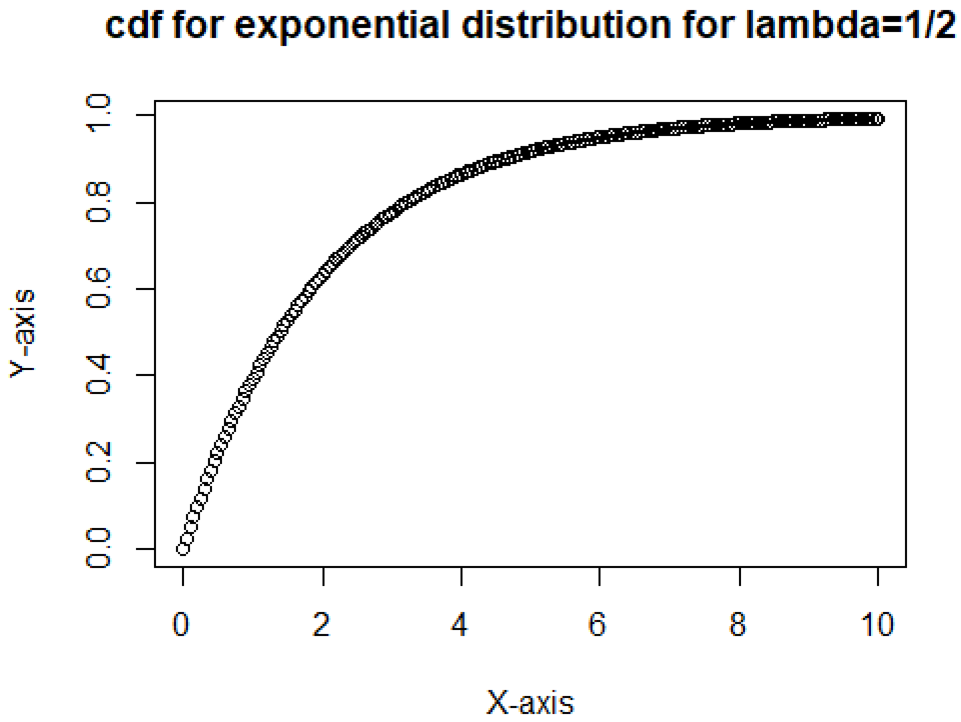
```
Exp_pdf<-function(x,lambda){
  f_x<-lambda*exp(-lambda*x)
  return(f_x)
}
a<-as.integer(readline(prompt='Enter the value of a:'))
p_x_4<-Exp_pdf(a,1/2)
print(p_x_4)

#OR

P_X_4<-dexp(4,rate=1/2)
print(P_X_4)
```

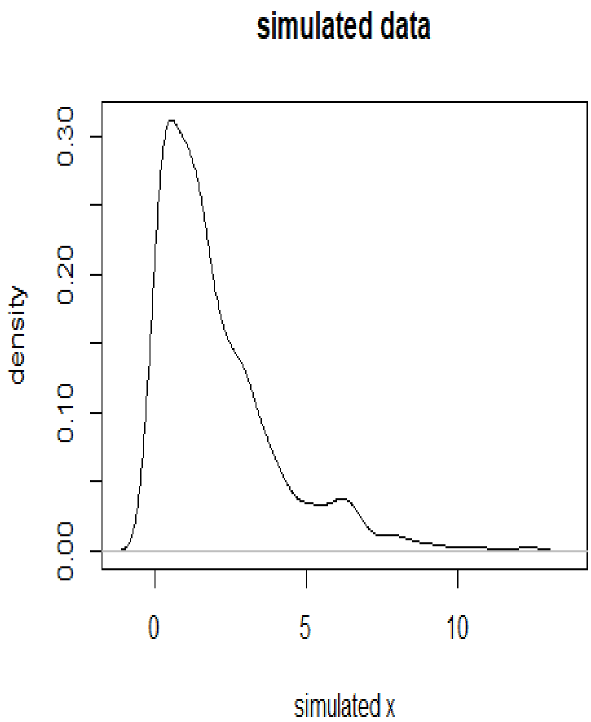
OUTPUT:

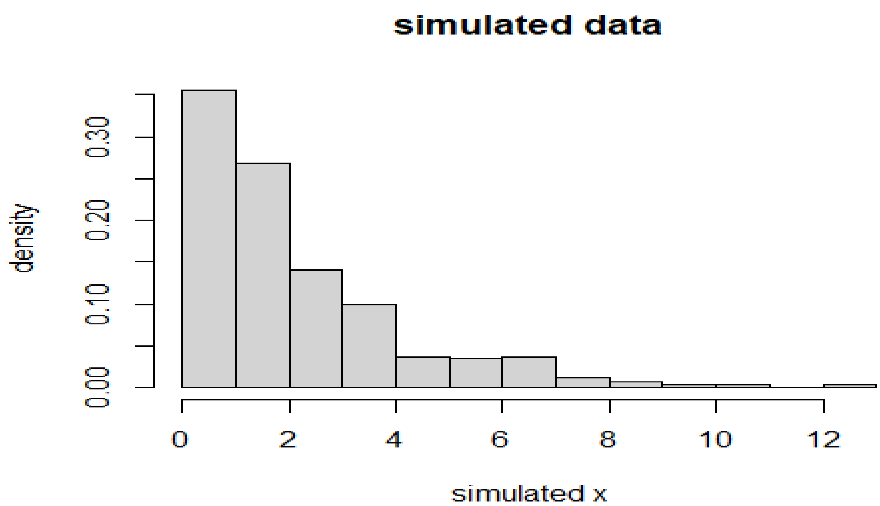
```
> #Ques-3 X~exp(lambda=1/2):
> Exp_pdf<-function(x,lambda){
+   f_x<-lambda*exp(-lambda*x)
+   return(f_x)
+ }
> a<-as.integer(readline(prompt='Enter the value of a:'))
Enter the value of a:p_x_4<-Exp_pdf(a,1/2)
Warning message:
NAs introduced by coercion
> print(p_x_4)
Error: object 'p_x_4' not found
> P_X_4<-dexp(4,rate=1/2)
> print(P_X_4)
[1] 0.06766764
```

```
> #Ques-3(b) plot pdf
> x<-seq(0,10,by=0.05) #by is step size
> f_x<-Exp_pdf(x,1/2)
> #print(f_x)
> plot(x,f_x,xlab='x-axis',ylab='y-axis',
+      main='Pdf for Exp dist for lambda=1/2')
> #OR Ques-3(b):
> x<-seq(0,10,by=0.05)
> f_x<-dexp(x,rate=1/2)
> plot(x,f_x,xlab="x-axis",ylab="y-axis"
+      ,main="Pdf for Exp dist for lambda=1/2")
> #Ques-3(c) P atmost 4:
> P_atmost_x_4<-pexp(4,rate=1/2)
> print(P_atmost_x_4)
[1] 0.8646647
> #Ques-3(d)
> x<-seq(0,10,by=0.05)
> F_X<-pexp(x,rate=1/2)
> plot(x,F_X,xlab="X-axis",ylab="Y-axis",
+      main="cdf for exponential distribution for lambda=1/2")
> #Ques-3(d)
> x<-seq(0,10,by=0.05)
> F_X<-pexp(x,rate=1/2)
> plot(x,F_X,xlab="X-axis",ylab="Y-axis",
+      main="cdf for exponential distribution for lambda=1/2")
> #Ques-3(e)
> n<-1000
> x_sim<-rexp(n,rate=1/2)#rexp for random numbers generate krne ke leye
> plot(density(x_sim),xlab="simulated x",ylab="density",
+      main="simulated data")
> hist(x_sim,probability=TRUE,xlab="simulated x",ylab="density",
+      main="simulated data")
```




cdf for exponential distribution for lambda=1/2
Y-axis
X-axis
0.0
0.2
0.4
0.6
0.8
1.0
0
2
4
6
8
10


simulated data
density
simulated x
0.00
0.10
0.20
0.30
0
5
10


simulated data
density
simulated x
0.00
0.10
0.20
0.30
0
2
4
6
8
10
12

Q4. The length of the shower on a tropical island during rainy season has an exponential distribution with parameter 2, time being measured in minutes. Write R-code to find:
(i) the probability that a shower will last more than 3 minutes?
(ii) If a shower has already lasted for 2 minutes, write R-code to find probability that it will last for at least one more minute?

CODE:
```
#P(X>3)=1-P(X<=3)=1-F(3)
prob1<-1-pexp(3,rate=2)
print(prob1)

prob2<-1-pexp(1,rate=2)
print(prob2)
```

OUTPUT:
```
> #Ques-4:
> #P(X>3)=1-P(X<=3)=1-F(3)
> prob1<-1-pexp(3,rate=2)
> print(prob1)
[1] 0.002478752
> prob2<-1-pexp(1,rate=2)
> print(prob2)
[1] 0.1353353
```

Q5. The marks obtained by a class of M.Sc. second year students in a mathematics course are found to be normally distributed with mean 64.5 and standard deviation 5. If class strength is 300, write R-code to find the number of students having marks
(i) less than 57,
(ii) between 57 to 72.

CODE:
```
prob1<-300*(pnorm(57,mean=64.5,sd=5))
print(prob1)

prob2<-300*(pnorm(72,mean=64.5,sd=5)-pnorm(57,mean=64.5,sd=5))
print(prob2)
```

OUTPUT:
```
> #Ques-5:X~N(mean=64.5, sd=5)
> prob1<-300*(pnorm(57,mean=64.5,sd=5))
> print(prob1)
[1] 20.04216
> prob2<-300*(pnorm(72,mean=64.5,sd=5)-pnorm(57,mean=64.5,sd=5))
> print(prob2)
[1] 259.9157
```

Q6. The lifetime of certain equipment is described by a random variable X that follows a Gamma distribution with parameters α = 2 and β = $1/3$
(i) Find the probability that the lifetime of equipment is at least 1 unit of time.
(ii) What is the value of c, if P(X ≤ c) ≥ 0.70?
(Hint: try quantile function qgamma())

CODE:
```
#P(X>=1)=1-P(X<1)=1-P(X<=1)=1-F(1)
P_i<-1-pgamma(1, shape=2,scale=1/3)
print(P_i)

#P(X<=c)>=0.70 find c
P_ii<-qgamma(0.70,shape=2,scale=1/3) #qgamma is used for quantile
print(P_ii)
```

OUTPUT:
```
> #Ques-6: X~gamma(alpha=2, beta=1/3)
> #P(X>=1)=1-P(X<1)=1-P(X<=1)=1-F(1)
> P_i<-1-pgamma(1, shape=2,scale=1/3)
> print(P_i)
[1] 0.1991483
> #P(X<=c)>=0.70 find c
> P_ii<-qgamma(0.70,shape=2,scale=1/3) #qgamma is used for quantile
> print(P_ii)
[1] 0.8130722
```

Q7. At a certain examination 10% of the students who appeared for the paper in Advanced Mathematics got less than 30 marks and 97% of the students got less than 62 marks. Assuming the distribution is normal, write R-code to find the mean and the standard deviation of the distribution.

CODE:
```
q1<-30 #10th percentile
p1<-0.10  #probability corresponding to q1
q2<-62  #97th percentile
p2<-0.97  #probability corresponding to q2
#use qnorm to find z value corresponding to p1 and p2
z1<-qnorm(p1)  #z value for 10th percentile
z2<-qnorm(p2)  #z value for 97th percentile
#z1<-q1-mu/sigma & z2<-q2-mu/sigma
#q1<=mu+Sigma * z1 and q2<=mu+sigma * z2
#After solving we get:
sigma<-(q2-q1)/(z2-z1)
mu<-q1-sigma*z1
cat("Mean (mu) ",mu,"\n")
cat("stadard deviation (sigma) ",sigma,"\n")
```

OUTPUT:

```
> #Ques-7:
> #given probabilities
> q1<-30 #10th percentile
> p1<-0.10  #probability corresponding to q1
> q2<-62  #97th percentile
> p2<-0.97  #probability corresponding to q2
> #use qnorm to find z value corresponding to p1 and p2
> z1<-qnorm(p1)  #z value for 10th percentile
> z2<-qnorm(p2)  #z value for 97th percentile
> #z1<-q1-mu/sigma & z2<-q2-mu/sigma
> #q1<=mu+Sigma * z1 and q2<=mu+sigma * z2
> #After solving we get:
> sigma<-(q2-q1)/(z2-z1)
> mu<-q1-sigma*z1
> cat("Mean (mu) ",mu,"\n")
Mean (mu)  42.96811
> cat("stadard deviation (sigma) ",sigma,"\n")
stadard deviation (sigma)  10.11907
```

Thapar Institute of Engineering & Technology, Patiala
Department of Mathematics
LAB Experiment 6: Joint probability mass and density functions

Q1. The joint probability mass function of two random variables X and Y is

$$p(x,y) = \frac{2x+y}{27};\ x = 0,1,2;\ y = 0,1,2.$$

Write a R-code to:

(i) Display the joint probability mass function in matrix form.

(ii) Check that it is a joint probability mass function or not ? (Use: `sum()`).

(iii) Find the marginal distribution $p_X(x)$ for $x = 0,1,2$ (Use: `apply()`).

(iv) Find the marginal distribution $p_Y(y)$ for $y = 0,1,2$ (Use: `apply()`).

(v) Find the conditional probability at $x = 0$, given $y = 1$.

(vi) Find $\mathbb{E}(x)$, $\mathbb{E}(y)$, $\mathbb{E}(xy)$, $\mathrm{Var}(x)$, $\mathrm{Var}(y)$, $\mathrm{Cov}(x,y)$, and the correlation coefficient between x and y.

CODE:

```
#f(x,y)=(2x+y)/27  x=0,1,2 and y=0,1,2
f<-function(x,y){
  f1<-(2*x+y)/27
  return(f1)
}

x<-c(0:2)
y<-c(0:2)
M1<-matrix(c(f(0,0:2),f(1,0:2),f(2,0:2)),nrow = 3, ncol = 3,byrow=TRUE)
print(M1)  #byrow used for row-wise means row-wise y hai and column-wise x hai

#check if it is jpmf or not
sum(M1) #sum se double summation work krta hai

#Marginal of x for x=0,1,2
px<-apply(M1,1,sum) #1 represent row sum of matrix M1
print(px)

#Marginal of y for x=0,1,2
py<-apply(M1,2,sum) #2 represent column sum of matrix M1
print(py)

#?apply - used to check what apply command does
# conditional prob for P(X=0|Y=1)
M1[1,2]
py[2]
P_x0_y1<-M1[1,2]/py[2]
print(P_x0_y1)
```

```
x<-c(0:2)
y<-c(0:2)
#E(x),E(y),var(x),var(y),cov(x,y)
E_X<-sum(x*px)
E_X
E_X2<-sum(x*x*px)
E_X2
Var_X<-E_X2-E_X^2
Var_X

E_Y<-sum(y*py)
E_Y
E_Y2<-sum(y*y*py)
E_Y2
Var_Y<-E_Y2-E_Y^2
Var_Y

x<-c(0:2)
y<-c(0:2)
f1<-function(x,y){x*y*(2*x+y)/27}
M2<-matrix(c(f1(0,0:2),f1(1,0:2),f1(2,0:2)),nrow=3,ncol=3,byrow=T)
M2
E_XY<-sum(M2)
E_XY
cov_XY<-E_XY-E_X*E_Y
cov_XY
rho<-cov_XY/sqrt(Var_X*Var_Y)
Rho
```

OUTPUT:

```
> #Ques-1:
> #jpmf matrix
> #f(x,y)=(2x+y)/27  x=0,1,2 and y=0,1,2
> f<-function(x,y){
+   f1<-(2*x+y)/27
+   return(f1)
+ }
> x<-c(0:2)
> y<-c(0:2)
> M1<-matrix(c(f(0,0:2),f(1,0:2),f(2,0:2)),nrow = 3, ncol = 3,byrow=TRUE)
> print(M1)  #byrow used for row-wise means row-wise y hai and column-wise x hai
           [,1]       [,2]       [,3]
[1,] 0.00000000 0.03703704 0.07407407
[2,] 0.07407407 0.11111111 0.14814815
[3,] 0.14814815 0.18518519 0.22222222
> #check if it is jpmf or not
> sum(M1) #sum se double summation work krta hai
[1] 1
> #Marginal of x for x=0,1,2
> px<-apply(M1,1,sum) #1 represent row sum of matrix M1
> print(px)
[1] 0.1111111 0.3333333 0.5555556
> #Marginal of y for x=0,1,2
> py<-apply(M1,2,sum) #2 represent column sum of matrix M1
> print(py)
[1] 0.2222222 0.3333333 0.4444444
> #?apply - used to check what apply command does
> # conditional prob for P(X=0|Y=1)
> M1[1,2]
[1] 0.03703704
> py[2]
[1] 0.3333333
> P_x0_y1<-M1[1,2]/py[2]
> print(P_x0_y1)
[1] 0.1111111
> x<-c(0:2)
> y<-c(0:2)
> #E(x),E(y),var(x),var(y),cov(x,y)
> E_X<-sum(x*px)
> E_X
```

```
[1] 1.444444
> E_X2<-sum(x*x*px)
> E_X2
[1] 2.555556
> Var_X<-E_X2-E_X^2
> Var_X
[1] 0.4691358
> E_Y<-sum(y*py)
> E_Y
[1] 1.222222
> E_Y2<-sum(y*y*py)
> E_Y2
[1] 2.111111
> Var_Y<-E_Y2-E_Y^2
> Var_Y
[1] 0.617284
> x<-c(0:2)
> y<-c(0:2)
> f1<-function(x,y){x*y*(2*x+y)/27}
> M2<-matrix(c(f1(0,0:2),f1(1,0:2),f1(2,0:2)),nrow=3,ncol=3,byrow=T)
> M2
     [,1]      [,2]      [,3]
[1,]    0 0.0000000 0.0000000
[2,]    0 0.1111111 0.2962963
[3,]    0 0.3703704 0.8888889
> E_XY<-sum(M2)
> E_XY
[1] 1.666667
> cov_XY<-E_XY-E_X*E_Y
> cov_XY
[1] -0.09876543
> rho<-cov_XY/sqrt(Var_X*Var_Y)
> rho
[1] -0.1835326
```

Q2. The joint probability density of two random variables X and Y is

$$f(x,y) = \begin{cases} \frac{2}{5}(x+4y), & 0 \le x, y \le 1; \\ 0, & \text{elsewhere.} \end{cases}$$

Write a R-code to:

(i) Check if it is a valid joint probability density function or not. (Use `integral2()`).

(ii) Find the marginal distribution $f_X(x)$ at $x = 1$.

(iii) Find the marginal distribution $f_Y(y)$ at $y = 0$.

(iv) Find the expected value of $g(x, y) = xy$.

CODE:

```
#(a) f(x,y)=2/5*(x+4y)  0<x,y<1
library(pracma)
f<-function(x,y){2*(x+4*y)/5}
I<-integral2(f, xmin = 0, xmax = 1, ymin = 0,ymax = 1)
print(I$Q)

#marginal distribution fx(X) at x=1
```

```
fx1<-function(y) f(1,y)
gx1<-integral(fx1,0,1)
print(gx1)

#marginal distribution fy(Y) at y=0
fy0<-function(x) f(x,0)
hy0<-integral(fy0,0,1)
print(hy0)


f_XY<-function(x,y){x*y*f(x,y)}
E_XY<-integral2(f_XY,0,1,0,1)
print(E_XY$Q)
```

OUTPUT:

```
> #Ques-2: $ is used to remove errors
> #(a) f(x,y)=2/5*(x+4y)  0<x,y<1
> library(pracma)
> f<-function(x,y){2*(x+4*y)/5}
> I<-integral2(f, xmin = 0, xmax = 1, ymin = 0,ymax = 1)
> print(I$Q)
[1] 1
> #marginal distribution fx(X) at x=1
> fx1<-function(y) f(1,y)
> gx1<-integral(fx1,0,1)
> print(gx1)
[1] 1.2
> #marginal distribution fy(Y) at y=0
> fy0<-function(x) f(x,0)
> hy0<-integral(fy0,0,1)
> print(hy0)
[1] 0.2
> f_XY<-function(x,y){x*y*f(x,y)}
> E_XY<-integral2(f_XY,0,1,0,1)
> print(E_XY$Q)
[1] 0.3333333
```

Thapar Institute of Engineering & Technology, Patiala
Department of Mathematics
LAB Experiment 7: Correlation and Regression

Q1. Write R-Code to calculate ρ between the following data:

1	2	3	4	5	6	7	8	9
9	8	10	12	11	13	14	16	15

Also, write R-code to obtain the two regression lines. Hence, calculate $\mathbb{E}(Y|X = 6.2)$.

```
CODE:
x<-c(1,2,3,4,5,6,7,8,9)
y<-c(9,8,10,12,11,13,14,16,15)
n<-9
E_x<-sum(x)/n
E_x
E_y<-sum(y)/n
E_y

E_x_2<-sum(x^2)/n
E_x_2
E_y_2<-sum(y^2)/n
E_y_2

E_xy<-sum(x*y)/n
cov_xy<-E_xy-E_x*E_y
var_x<-E_x_2-E_x^2
var_y<-E_y_2-E_y^2

rho<-cov_xy/sqrt(var_x*var_y)
rho


#OR using inbuilt function!
rho<-cor(x,y)
cat("Corr, coeff, rho b/w x and y= ",rho,"\n")

#ALso write Rcode to find lines of regression!
#Y on X i.e. E_YonX

reg_Y_on_X<-lm(y~x)
cat("Reg line Y on X: Y =",coef(reg_Y_on_X)[1],"+",coef(reg_Y_on_X)[2],
  "*x\n")

#Calculate at x=6.2
reg_Y_on_X<-lm(y~x)
cat("Reg line Y on X=6.2: Y =",coef(reg_Y_on_X)[1],"+",coef(reg_Y_on_X)[2],
  "*6.2\n")
```

reg_Y_on_X<-lm(y~x)
cat("Reg line Y on X=6.2: Y =",coef(reg_Y_on_X)[1]+coef(reg_Y_on_X)[2]*6.2,
"\n")

OUTPUT:

```
> #Ques-1: rho<-cov(X,Y)/sqrt(VarX*VarY)
> x<-c(1,2,3,4,5,6,7,8,9)
> y<-c(9,8,10,12,11,13,14,16,15)
> n<-9
> E_x<-sum(x)/n
> E_x
[1] 5
> E_y<-sum(y)/n
> E_y
[1] 12
> E_x_2<-sum(x^2)/n
> E_x_2
[1] 31.66667
> E_y_2<-sum(y^2)/n
> E_y_2
[1] 150.6667
> E_xy<-sum(x*y)/n
> cov_xy<-E_xy-E_x*E_y
> var_x<-E_x_2-E_x^2
> var_y<-E_y_2-E_y^2
> rho<-cov_xy/sqrt(var_x*var_y)
> rho
[1] 0.95
> #OR using inbuilt function!
> rho<-cor(x,y)
> cat("Corr, coeff, rho b/w x and y= ",rho,"\n")
Corr, coeff, rho b/w x and y=  0.95
> reg_Y_on_X<-lm(y~x)
> cat("Reg line Y on X: Y =",coef(reg_Y_on_X)[1],"+",coef(reg_Y_on_X)[2],
+     "*x\n")
Reg line Y on X: Y = 7.25 + 0.95 *x
> #Calculate at x=6.2
> reg_Y_on_X<-lm(y~x)
> cat("Reg line Y on X=6.2: Y =",coef(reg_Y_on_X)[1],"+",coef(reg_Y_on_X)[2],
+     "*6.2\n")
Reg line Y on X=6.2: Y = 7.25 + 0.95 *6.2
> reg_Y_on_X<-lm(y~x)
> cat("Reg line Y on X=6.2: Y =",coef(reg_Y_on_X)[1]+coef(reg_Y_on_X)[2]*6.2,
+     "\n")
Reg line Y on X=6.2: Y = 13.14
```

Q2. Let (X, Y) be a random vector with joint probability density function

$$f(x,y) = \begin{cases} 2, & 0 < x \leq y < 1, \\ 0, & \text{otherwise.} \end{cases}$$

Write R-code to determine the correlation coefficient between X and Y.

CODE:

```
reg_Y_on_x<-lm(y~x)
cat("Reg line Y on X: Y =",coef(reg_Y_on_X)[1],"+",coef(reg_Y_on_X)[2],
   "*x\n")
#f(x,y)=2 0<x<y<=1
n<-100000
#get sum for y from a uniform distribution in (0 1)
y<-runif(n,min=0,max=1)
#get sum for x from a unif dist in 0<x<=y
x<-runif(n,min=0,max=y)
#calculate corr coeff b/w x and y
corr_coeff<-cor(x,y)
cat("Approx corr coeff(rho): ", corr_coeff,"\n")

#ques-> regression line y on x
Reg_y_on_x<-lm(y~x)  #lm means linear model
cat("Reg line Y on X : Y =",coef(Reg_y_on_x)[1],"+",coef(Reg_y_on_x)[2],
   "*x\n")
```

OUTPUT:

```
> #Ques-2:
> reg_Y_on_x<-lm(y~x)
> cat("Reg line Y on X: Y =",coef(reg_Y_on_X)[1],"+",coef(reg_Y_on_X)[2],
+     "*x\n")
Reg line Y on X: Y = 7.25 + 0.95 *x
> #f(x,y)=2 0<x<y<=1
> n<-100000
> #get sum for y from a uniform distribution in (0 1)
> y<-runif(n,min=0,max=1)
> #get sum for x from a unif dist in 0<x<=y
> x<-runif(n,min=0,max=y)
> #calculate corr coeff b/w x and y
> corr_coeff<-cor(x,y)
> cat("Approx corr coeff(rho): ", corr_coeff,"\n")
Approx corr coeff(rho):  0.6525842
> #ques-> regression line y on x
> Reg_y_on_x<-lm(y~x)  #lm means linear model
> cat("Reg line Y on X : Y =",coef(Reg_y_on_x)[1],"+",coef(Reg_y_on_x)[2],
+     "*x\n")
Reg line Y on X : Y = 0.2867005 + 0.8540785 *x
```

Q3. Let (X, Y) be a random vector with joint probability density function

$$f(x,y) = \begin{cases} e^{-y}, & 0 < x < y < \infty; \\ 0, & \text{otherwise.} \end{cases}$$

Write R-code to find correlation coefficient ρ and find linear regression of Y on $X = x$.

CODE:

```
n<-100000

#Generate samples for Y using exponential distr (rate=1)
y<-rexp(n,rate = 1)

#generate x conditional y so that 0<x<y
x<-runif(n, min=0, max=y)

#calc the coerr coeff b/w x and y
corr_coeff<-cor(x,y)
cat("Approx corr coeff(rho): ", corr_coeff,"\n")
Reg_y_on_x<-lm(y~x)  #lm means linear model
cat("Reg line Y on X : Y =",coef(Reg_y_on_x)[1],"+",coef(Reg_y_on_x)[2],
   "*x\n")
```

OUTPUT:

```
> #Ques-3:
> n<-100000
> #Generate samples for Y using exponential distr (rate=1)
> y<-rexp(n,rate = 1)
> #generate x conditional y so that 0<x<y
> x<-runif(n, min=0, max=y)
> #calc the coerr coeff b/w x and y
> corr_coeff<-cor(x,y)
> cat("Approx corr coeff(rho): ", corr_coeff,"\n")
Approx corr coeff(rho):  0.7786864
> Reg_y_on_x<-lm(y~x)  #lm means linear model
> cat("Reg line Y on X : Y =",coef(Reg_y_on_x)[1],"+",coef(Reg_y_on_x)[2]
+     "*x\n")
Reg line Y on X : Y = 0.39644 + 1.20323 *x
```