

RAMNIRANJAN JHUNJHUNWALA COLLEGE GHATKOPAR (W), MUMBAI - 400 086

DEPARTMENT OF INFORMATION TECHNOLOGY

2022 - 2023

S.Y. B. Sc.( I.T.) SEM IV

Paper RJSUITP403 – Computer Oriented Statistical techniques

Name :- Rajbhar Sudesh Dinesh Sushiladevi

Roll No. 3067

Hindi Vidya Prachar Samiti’s

RAMNIRANJAN JHUNJHUNWALA COLLEGE

Ghatkopar (W), Mumbai-400 086

*Certificate*



This is to certify that Mr./Ms. Rajbhar Sudesh Dinesh Sushiladevi Roll No 3067 of S.Y.B.Sc.(I.T.) class has completed the required number of experiments in the subject of Computer Oriented Statistical Techniques in the Department of Information Technology during the academic year 2022 - 2023 .

Professor In-Charge Co-ordinator of IT Department

Prof. Bharati Bhole

Prof. Archana Bhide

College Seal & Date Examiner

Computer Oriented Statistical Technique Practicals Journal

### INDEX

| Sr. No | Practical | Date |
| --- | --- | --- |
| 1. | Using R execute the basic commands, array, list and frames. | 15/12/2022 |
| 2. | Create a Matrix using R and Perform the operations addition, inverse, transpose and multiplication operations. | 21/01/2023 |
| 3. | Using R Execute the statistical functions: mean, median, mode, quartiles, range, inter quartile range histogram | 28/01/2023 |
| 4. | Using R import the data from Excel / .CSV file and perform the functions.  mean, median, mode, quartiles, range, inter quartile range histogram | 04/02/2023 |
| 5. | Calculate the standard deviation, variance, co-variance. | 12/01/2023 |
| 6. | Moment,skewness,kurtosis. | 19/01/2023 |
| 7. | Perform the hypothetical testing. | 14/02/2023 |
| 8. | Perform the Chi-squared Test. | 21/02/2023 |
| 9. | Binomial and Normal distribution | 02/02/2023 |
| 10. | Perform the Linear Regression using R. | 09/02/2023 |

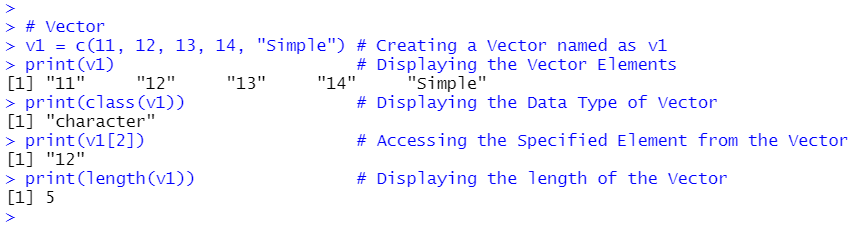
### Practical 1

#### Using R execute the basic commands, array, list and frames.

##### Vector:

| # Vector  v1 = c(11, 12, 13, 14, "Simple") # Creating a Vector named as v1  print(v1) # Displaying the Vector Elements  print(class(v1)) # Displaying the Data Type of Vector  print(v1[2]) # Accessing the Specified Element from the Vector  print(length(v1)) # Displaying the length of the Vector |
| --- |

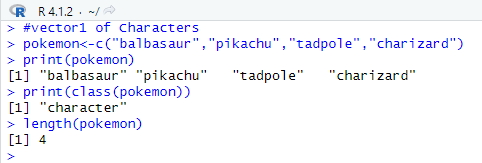
Output



Vector program2:length

| #vector1 of Characters  pokemon<-c("bulbasaur","pikachu","tadpole","charizard")  print(pokemon)  print(class(pokemon))  length(pokemon) |
| --- |

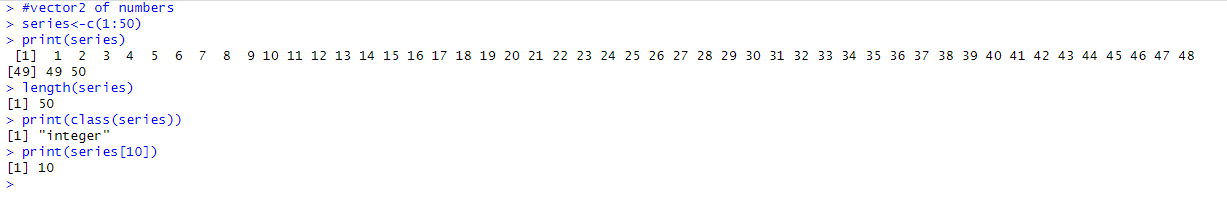
Output



Vector program 3:use of c.

| #vector2 of numbers  series<-c(1:50)  print(series)  length(series)  print(class(series))  print(series[10]) |
| --- |

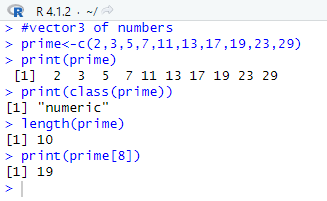
Output



Vector program 4:class

| #vector3 of numbers  prime<-c(2,3,5,7,11,13,17,19,23,29)  print(prime)  print(class(prime))  length(prime)  print(prime[8]) |
| --- |

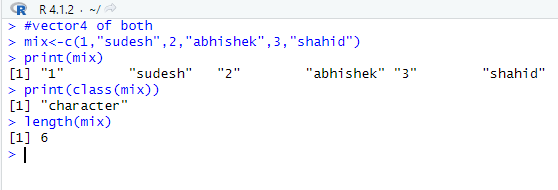
Output



Vector PrograM 5:use of mix

| #vector4 of both  mix<-c(1,"sudesh",2,"abhishek",3,"shahid")  print(mix)  print(class(mix))  length(mix) |
| --- |

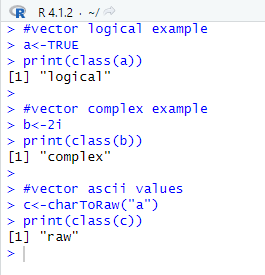
Output



Vector Program 6: use of Char to Raw

| #vector logical example  a<-TRUE  print(class(a))  #vector complex example  b<-2i  print(class(b))  #vector ascii values  c<-charToRaw("a")  print(class(c)) |
| --- |

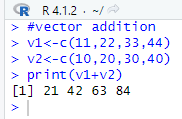
Output



Vector Program 7:addition

| #vector addition  v1<-c(11,22,33,44)  v2<-c(10,20,30,40)  print(v1+v2) |
| --- |

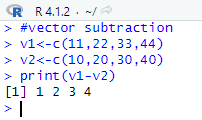
Output



Vector Program 8: subtraction

| #vector subtraction  v1<-c(11,22,33,44)  v2<-c(10,20,30,40)  print(v1-v2) |
| --- |

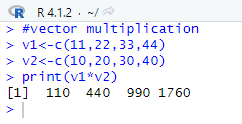
Output



Vector Program 9:multiplication

| #vector multiplication  v1<-c(11,22,33,44)  v2<-c(10,20,30,40)  print(v1\*v2) |
| --- |

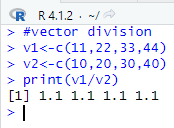
Output



Vector Program 10: division

| #vector division  v1<-c(11,22,33,44)  v2<-c(10,20,30,40)  print(v1/v2) |
| --- |

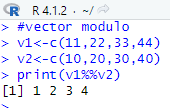
Output



Vector Program 11:modulo

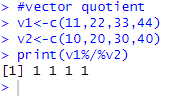
| #vector modulo  v1<-c(11,22,33,44)  v2<-c(10,20,30,40)  print(v1%%v2) |
| --- |

Output



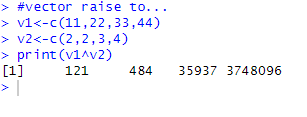
Vector Program 12: quotient

| #vector quotient  v1<-c(11,22,33,44)  v2<-c(10,20,30,40)  print(v1%/%v2) |
| --- |



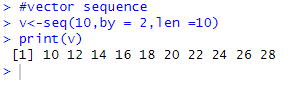
Vector Program 13: exponent

| #vector raise to...  v1<-c(11,22,33,44)  v2<-c(2,2,3,4)  print(v1^v2) |
| --- |



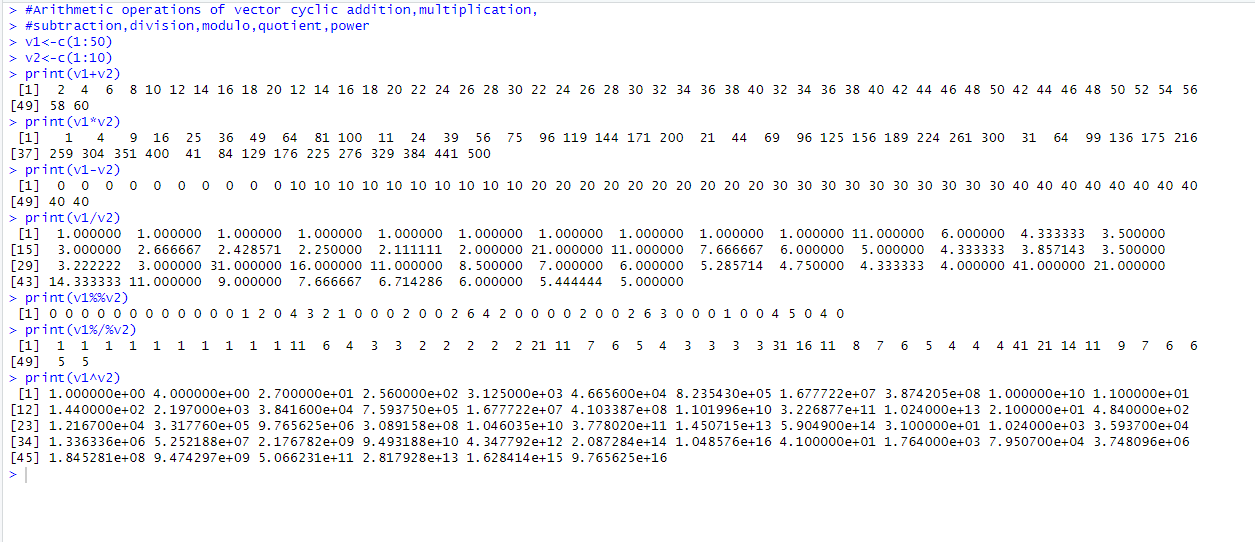
Vector Program 14: use of Sequence

| #vector sequence  v<-seq(10,by = 2,len =10)  print(v) |
| --- |



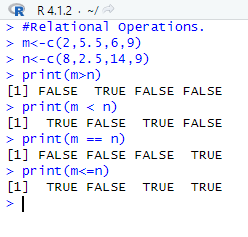
Vector Program 15

| #Arithmetic operations of vector cyclic addition,multiplication,  #subtraction,division,modulo,quotient,power  v1<-c(1:50)  v2<-c(1:10)  print(v1+v2)  print(v1\*v2)  print(v1-v2)  print(v1/v2)  print(v1%%v2)  print(v1%/%v2)  print(v1^v2) |
| --- |



Vector Program 16

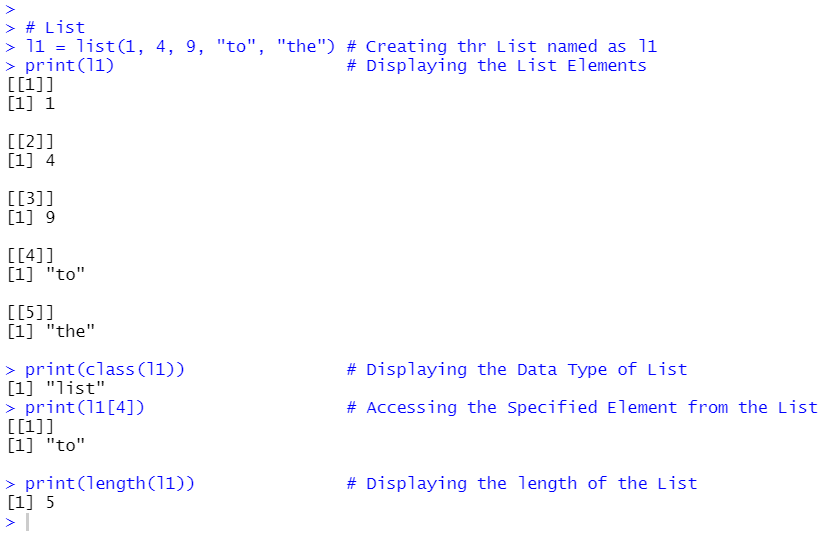
| #Relational Operations.  m<-c(2,5.5,6,9)  n<-c(8,2.5,14,9)  print(m>n)  print(m < n)  print(m == n)  print(m<=n) |
| --- |



##### List

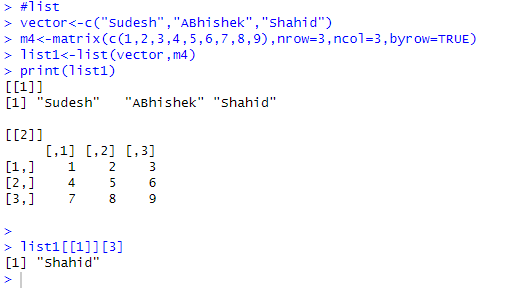
| # List  l1 = list(1, 4, 9, "to", "the") # Creating the List named as l1  print(l1) # Displaying the List Elements  print(class(l1)) # Displaying the Data Type of List  print(l1[4]) # Accessing the Specified Element from the List  print(length(l1)) # Displaying the length of the List |
| --- |

Output



| #list  vector<-c("Sudesh","ABhishek","Shahid")  m4<-matrix(c(1,2,3,4,5,6,7,8,9),nrow=3,ncol=3,byrow=TRUE)  list1<-list(vector,m4)  print(list1)  list1[[1]][3] |
| --- |

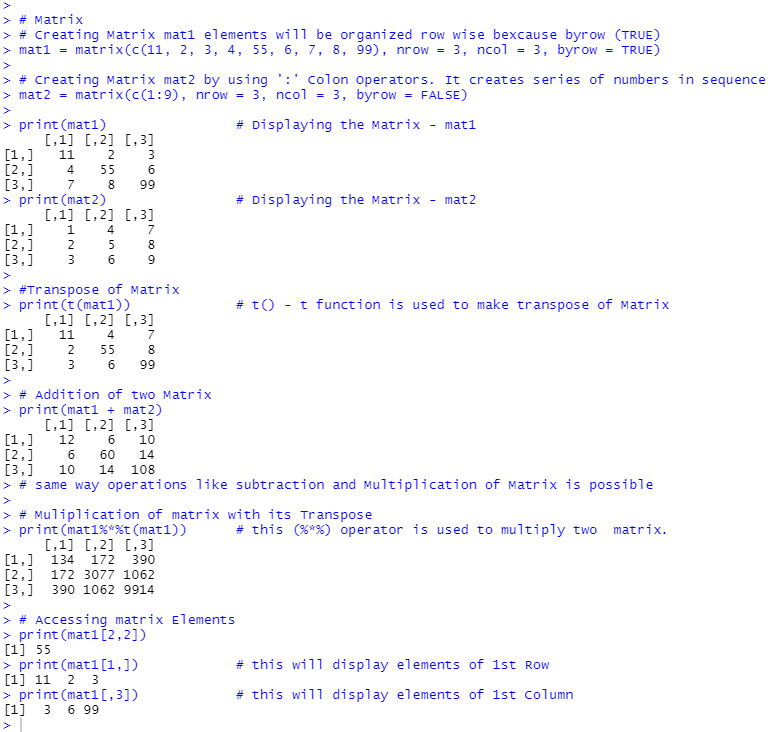
Output



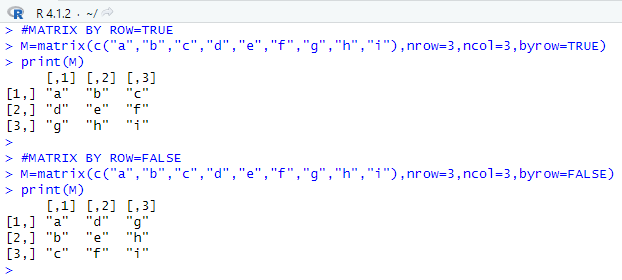
##### Matrix

| # Matrix  # Creating Matrix mat1 elements will be organized row wise because byrow (TRUE)  mat1 = matrix(c(11, 2, 3, 4, 55, 6, 7, 8, 99), nrow = 3, ncol = 3, byrow = TRUE)  # Creating Matrix mat2 by using ':' Colon Operators. It creates series of numbers in sequence  mat2 = matrix(c(1:9), nrow = 3, ncol = 3, byrow = FALSE)  print(mat1) # Displaying the Matrix - mat1  print(mat2) # Displaying the Matrix - mat2  #Transpose of Matrix  print(t(mat1)) # t() - t function is used to make transpose of Matrix  # Addition of two Matrix  print(mat1 + mat2)  # same way operations like subtraction and Multiplication of Matrix is possible  # Multiplication of matrix with its Transpose  print(mat1%\*%t(mat1)) # this (%\*%) operator is used to multiply two Matrices..  # Accessing matrix Elements  print(mat1[2,2])  print(mat1[1,]) # this will display elements of 1st Row  print(mat1[,3]) # this will display elements of 1st Column |
| --- |

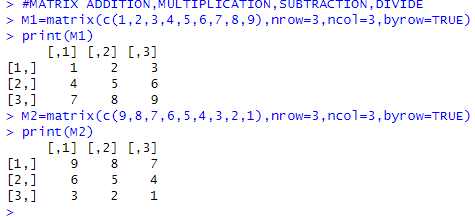
Output

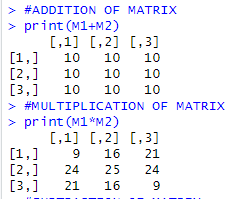


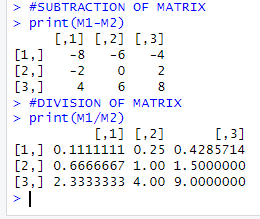
| #MATRIX BY ROW=TRUE  M=matrix(c("a","b","c","d","e","f","g","h","i"),nrow=3,ncol=3,byrow=TRUE)  print(M)  #MATRIX BY ROW=FALSE  M=matrix(c("a","b","c","d","e","f","g","h","i"),nrow=3,ncol=3,byrow=FALSE)  print(M) |
| --- |



| #MATRIX ADDITION,MULTIPLICATION,SUBTRACTION,DIVIDE  M1=matrix(c(1,2,3,4,5,6,7,8,9),nrow=3,ncol=3,byrow=TRUE)  print(M1)  M2=matrix(c(9,8,7,6,5,4,3,2,1),nrow=3,ncol=3,byrow=TRUE)  print(M2)  #ADDITION OF MATRIX  print(M1+M2)  #MULTIPLICATION OF MATRIX  print(M1\*M2)  #SUBTRACTION OF MATRIX  print(M1-M2)  #DIVISION OF MATRIX  print(M1/M2) |
| --- |



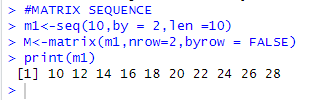




Matrix seq

| #MATRIX SEQUENCE  m1<-seq(10,by = 2,len =10)  M<-matrix(m1,nrow=2,byrow = FALSE)  print(m1) |
| --- |

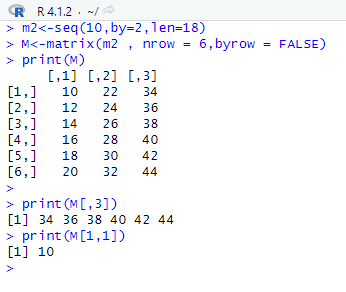
Output:



MATRIX PROG

| m2<-seq(10,by=2,len=18)  M<-matrix(m2 , nrow = 6,byrow = FALSE)  print(M)  print(M[,3])  print(M[1,1]) |
| --- |

Output



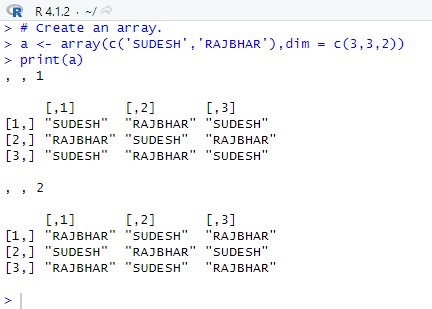
##### Array

| # Array  arr1 = array(c(1:5)) # 1D Array  arr2 = array(c(1:9), dim=c(3, 3)) # 2D Array  arr3 = array(c(1:9), dim=c(3, 3, 2)) # 2D Array but 2 equal Arrays 3rd parameter in dim is No. of Arrays  # printing Arrays  print(arr1) # 1D  print(arr2) # 2D  print(arr3) # 2 equal arrays  # Accessing Array Element  print(arr2[2,2])  # Checking whether Element is Present in data Structure  print(5%in%arr2)  print(10%in%arr2) |
| --- |

Output



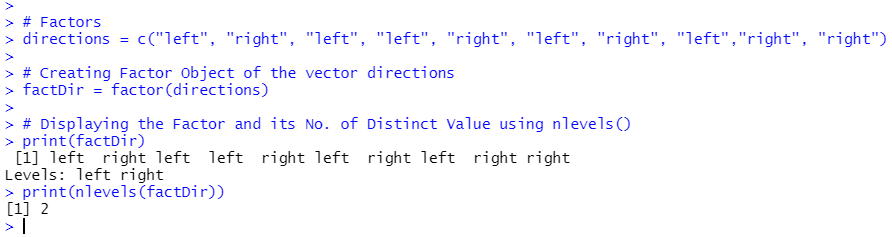
| # Create an array.  a <- array(c('SUDESH','RAJBHAR'),dim = c(3,3,2))  print(a) |
| --- |



##### Factor

| # Factors  directions = c("left", "right", "left", "left", "right", "left", "right", "left","right", "right")  # Creating Factor Object of the vector directions  factDir = factor(directions)  # Displaying the Factor and its No. of Distinct Value using nLevels()  print(factDir)  print(nlevels(factDir)) |
| --- |

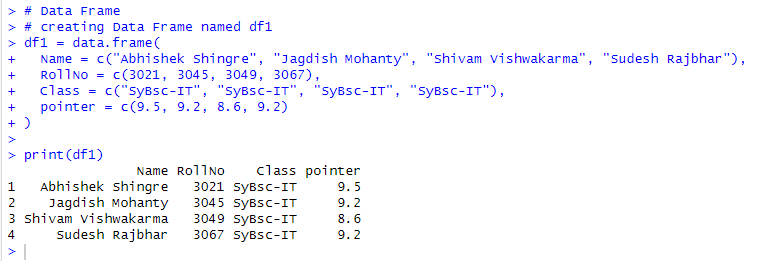
Output



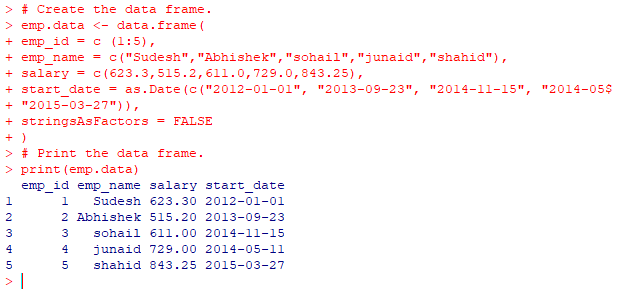
##### Data Frames

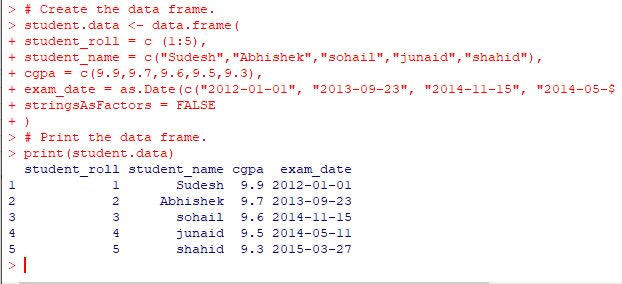
| # Data Frame  # creating Data Frame named df1  df1 = data.frame(  Name = c("Abhishek Shingre", "Jagdish Mohanty", "Shivam Vishwakarma", "Sudesh Rajbhar"),  RollNo = c(3021, 3045, 3049, 3067),  Class = c("SyBsc-IT", "SyBsc-IT", "SyBsc-IT", "SyBsc-IT"),  pointer = c(9.5, 9.2, 8.6, 9.2)  )  print(df1) |
| --- |

Output

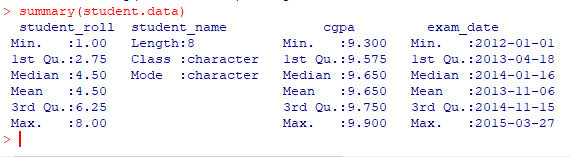


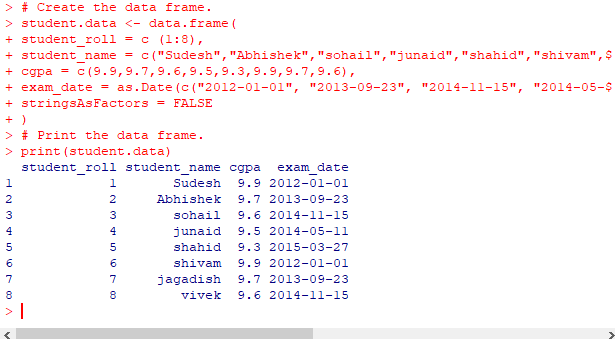
Student example:

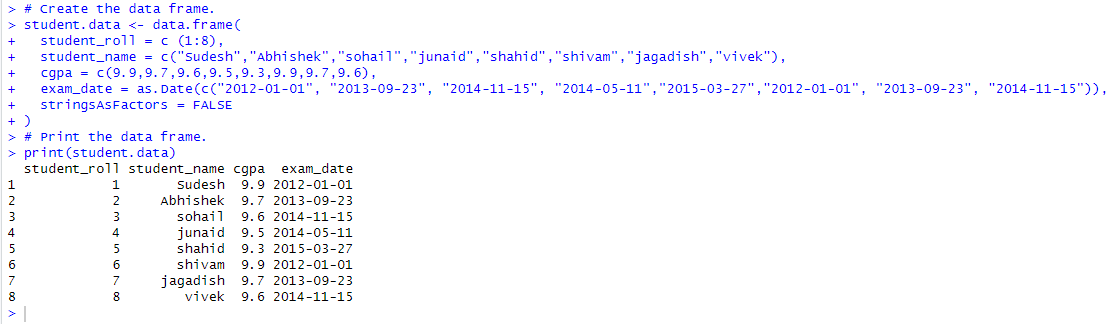




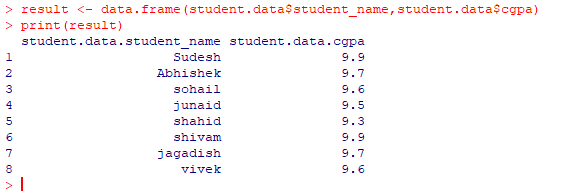
summary()







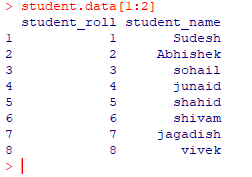
Specific column:



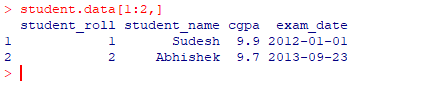
class:



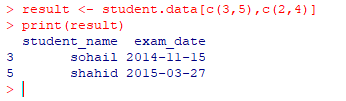
selection:



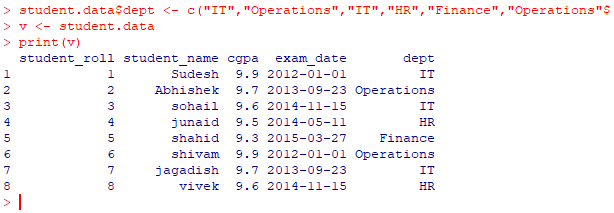
selection:



selection:



Add\_column:



### Practical 2

#### Create a Matrix using R and Perform the operations addition, inverse, transpose and multiplication operations.

### 

Matrix Arithmetic Operations

> a=matrix(c(2,4,5,6),nrow=2,ncol=2)

> b=matrix(c(1,6,2,3),nrow=2,ncol=2)

> a

[,1] [,2]

[1,] 2 5

[2,] 4 6

> b

[,1] [,2]

[1,] 1 2

[2,] 6 3

#Addition

> a+b

[,1] [,2]

[1,] 3 7

[2,] 10 9

#Subtraction

> a-b

[,1] [,2]

[1,] 1 3

[2,] -2 3

#Multiplication

> a\*b

[,1] [,2]

[1,] 2 10

[2,] 24 18

#Division

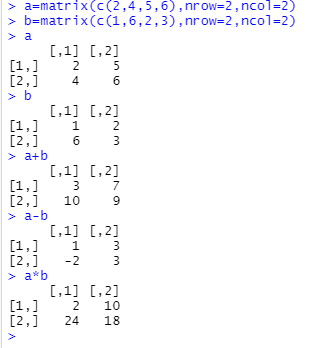
> a/b

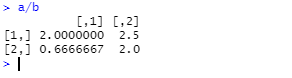
[,1] [,2]

[1,] 2.0000000 2.5

[2,] 0.6666667 2.0

Output:-





Length

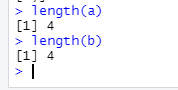
> length(a)

[1] 4

> length(b)

[1] 4

Output:-



Is Something a Matrix

is.matrix(a)

Output:-



is.vector(a)

Output:-

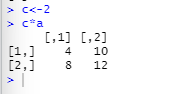


Multiplication by a Scalar

c<-2

c\*a

Output:-



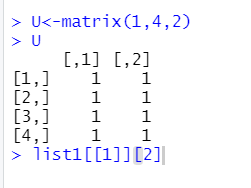
Unit Matrix

variable<-matrix(1,row,column)

U<-matrix(1,4,2)

U

Output:-

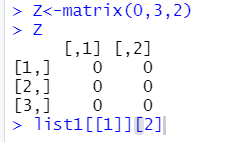


Zero Matrix

variable(0,row,col)

Z<-matrix(0,3,2)

Output:-

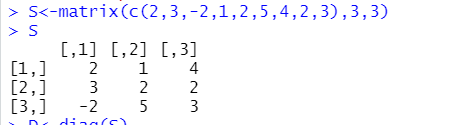


Diagonal Matrix

S<-matrix(c(2,3,-2,1,2,5,4,2,3),3,3)

S

Output:-



D<-diag(S)

D

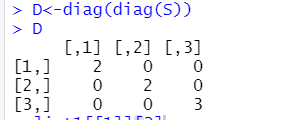
Output:-



D<-diag(diag(S))

D

Output:-

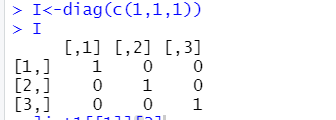


Identity Matrix

I<-diag(c(1,1,1))

I

Output:-

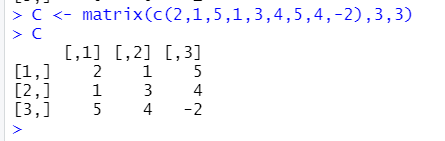


Symmetric Matrix

C <- matrix(c(2,1,5,1,3,4,5,4,-2),3,3)

C

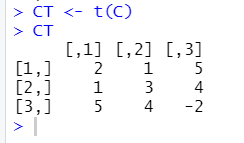
Output:-



CT <- t(C)

CT

Output:-

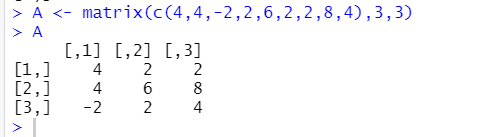


##### Inverse of a Matrix

A <- matrix(c(4,4,-2,2,6,2,2,8,4),3,3)

A

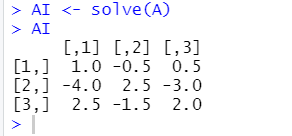
Output:-



AI <- solve(A)

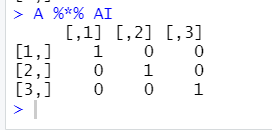
AI

Output:-



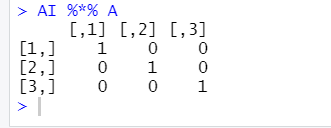
A %\*% AI

Output:-



AI %\*% A

Output:-

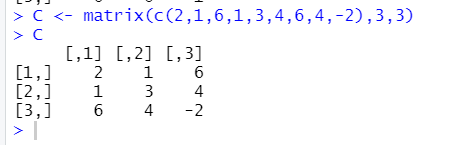


##### Inverse & Determinant of a Matrix

C <- matrix(c(2,1,6,1,3,4,6,4,-2),3,3)

C

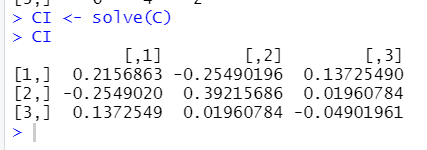
Output:-



CI <- solve(C)

CI

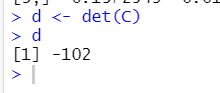
Output:-



d <- det(C)

D

Output:-



##### Transpose

> a=matrix(c(1,4,6,8,9,3),nrow=2,ncol=3,byrow=TRUE)

> a

[,1] [,2] [,3]

[1,] 1 4 6

[2,] 8 9 3

> print(t(a))

[,1] [,2]

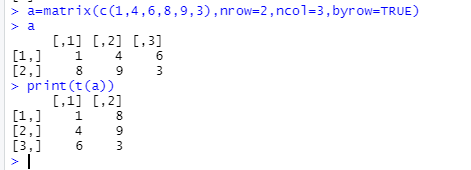
[1,] 1 8

[2,] 4 9

[3,] 6 3

>

Output:-

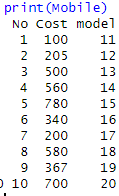


### Practical 3

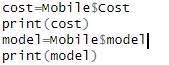
#### Using R Execute the statistical functions: mean, median, mode, quartiles, range, inter quartile range histogram

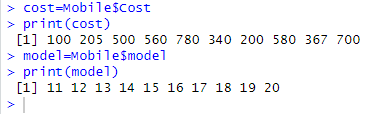
Creating Data frame for above Calculations:

| # Data Frame  # creating Data Frame for Performing Practical 3  Mobile = data.frame(  No = c(1:10),  Cost = c(100,205,500,560,780,340,200,580,367,700),  model = c(11:20)  )  print(Mobile) |
| --- |



Creating separate variables for Cost and model:

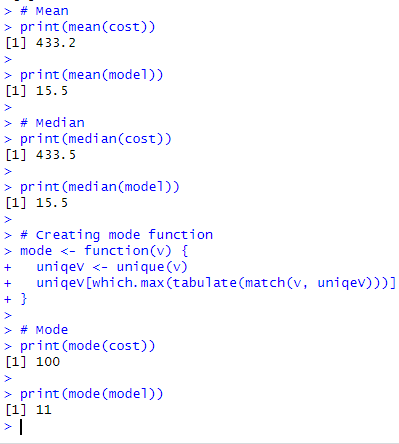




##### Mean Median and Mode:

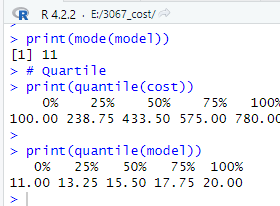
| # Mean  print(mean(cost))  print(mean(model))  # Median  print(median(cost))  print(median(model))  # Creating mode function  mode <- function(v) {  uniqeV <- unique(v)  uniqeV[which.max(tabulate(match(v, uniqeV)))]  }  # Mode  print(mode(cost))  print(mode(model)) |
| --- |

Output:



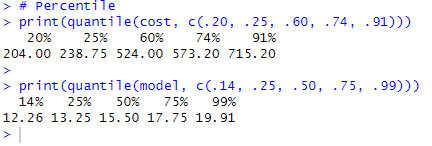
##### Quartile:

| # Quartile  print(quantile(cost))  print(quantile(model)) |
| --- |



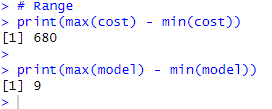
##### Percentile:

| # Percentile  print(quantile(cost, c(.20, .25, .60, .74, .91)))  print(quantile(model, c(.14, .25, .50, .75, .99))) |
| --- |



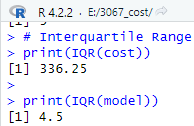
##### Range:

| # Range  print(max(cost) - min(cost))  print(max(model) - min(model)) |
| --- |



##### Interquartile Range:

| # Interquartile Range  print(IQR(cost))  print(IQR(model)) |
| --- |



##### Histogram:

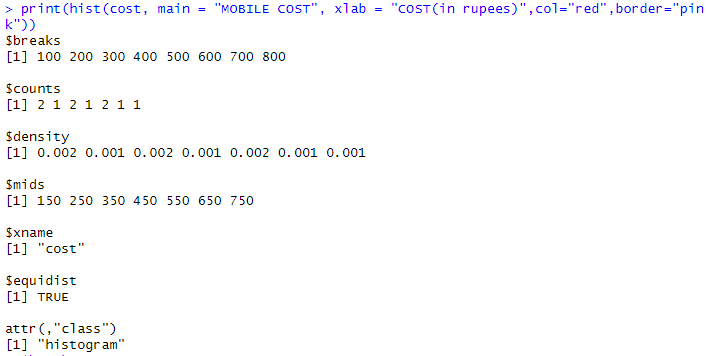
hist()

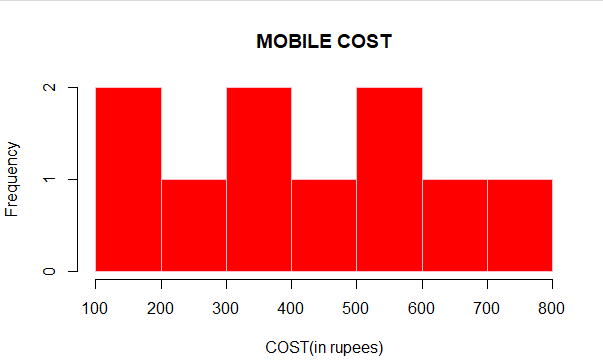
hist(v,main,xlab,ylab,xlim,ylim,breaks,col,border)

* v is a vector containing numeric values used in histogram.
* main indicates title of the chart.
* col is used to set color of the bars.
* border is used to set border color of each bar.
* xlab is used to give description of x-axis.
* xlim is used to specify the range of values on the x-axis.
* ylim is used to specify the range of values on the y-axis.
* breaks is used to mention the width of each bar.

Cost Histogram:

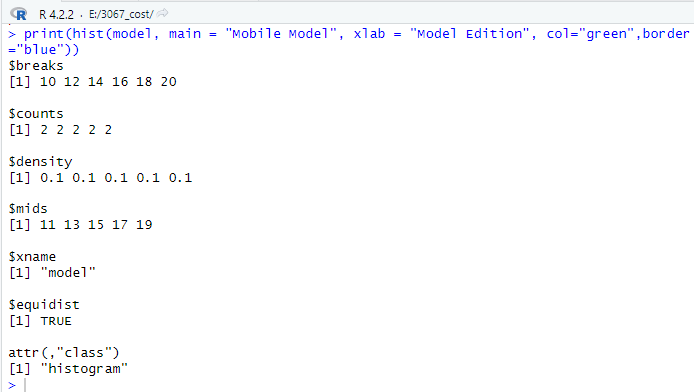
| print(hist(cost, main = "MOBILE COST", xlab = "COST(in rupees)",col="red",border="pink")) |
| --- |

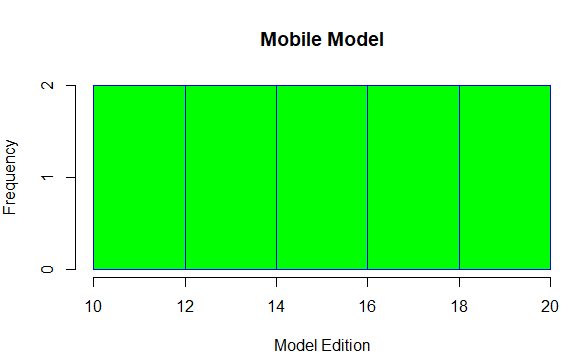




Model Histogram:

| print(hist(model, main = "Mobile Model", xlab = "Model Edition", col="green",border="blue")) |
| --- |





### 

### Practical 4

#### Using R import the data from Excel / .CSV file and perform the above functions.

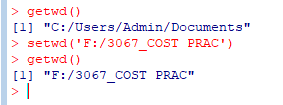
##### Pre Required steps for Excel / .CSV file import

Getting and Setting the Working Directory

| # Get and print the current working directory.  print(getwd())  # Set current working directory.  setwd("F:/3067\_COST PRAC")  # Get and print the current working directory.  print(getwd()) |
| --- |



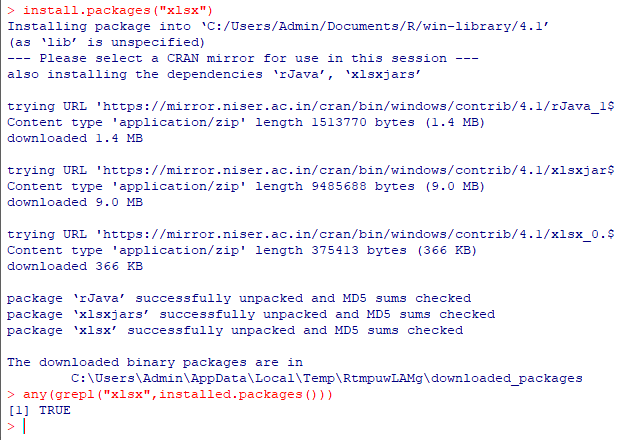






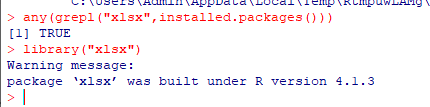
Importing and checking all required packages for xlsx

| install.packages("xlsx")  library("xlsx")  any(grepl("xlsx", installed.packages())) |
| --- |



Importing and checking all required packages for csv

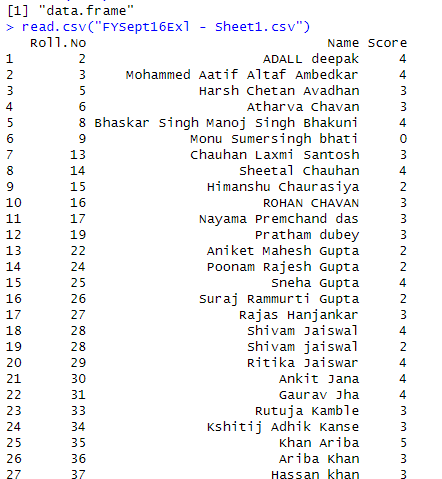
| install.packages("csv")  library("csv")  any(grepl("csv", installed.packages())) |
| --- |



Example of Reading a CSV File:

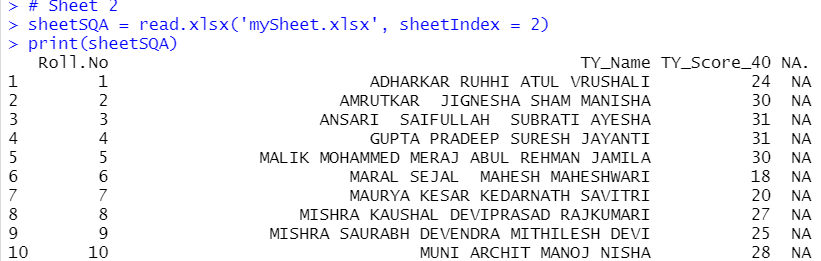
| read.csv("FYSept16Exl - Sheet1.csv") |
| --- |





Reading first file:

| # Sheet 2  sheetSQA = read.xlsx('mySheet.xlsx', sheetIndex = 2)  print(sheetSQA)  print(class(sheetSQA)) |
| --- |

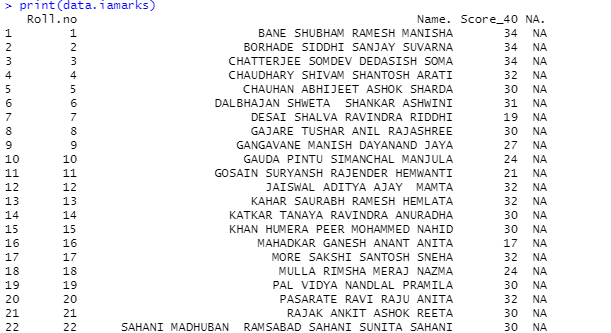


Reading 2nd file

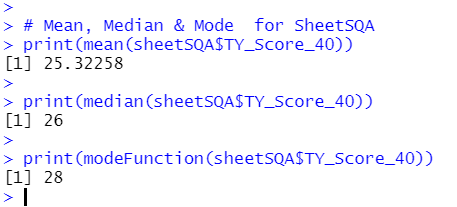
data.iamarks=read.xlsx('IA\_Practical\_COST\_SQA\_2022.xlsx',sheetIndex=1,header=TRUE)

print(data.iamarks)

Output:-

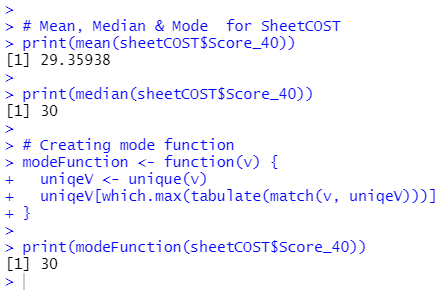


| # Mean, Median & Mode for SheetSQA  print(mean(sheetSQA$TY\_Score\_40))  print(median(sheetSQA$TY\_Score\_40))  print(modeFunction(sheetSQA$TY\_Score\_40)) # using modeFunction Created in previous Block |
| --- |



Mean Median & Mode

| # Mean, Median & Mode for SheetCOST  print(mean(sheetCOST$Score\_40))  print(median(sheetCOST$Score\_40))  # Creating mode function  modeFunction <- function(v) {  uniqeV <- unique(v)  uniqeV[which.max(tabulate(match(v, uniqeV)))]  }  print(modeFunction(sheetCOST$Score\_40)) |
| --- |



Mean-

>print(mean(data.iamarks$Score\_40))

Output:-



Median-

>print(median(data.iamarks$Score\_40))

Output:-



Mode -

getmode = function(v){

uni\_v = unique(v)

uni\_v[which.max(tabulate(match(v,uni\_v)))]

}

print(getmode(data.iamarks$Score\_40))

Output -



Mean-

>print(mean(data.iamarks$TY\_Score\_40))

Output:-



Median-

> print(median(data.iamarks$TY\_Score\_40))

Output:-



Mode -

getmode = function(v){

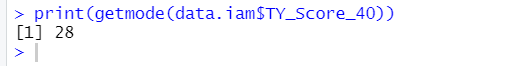
uni\_v = unique(v)

uni\_v[which.max(tabulate(match(v,uni\_v)))]

}

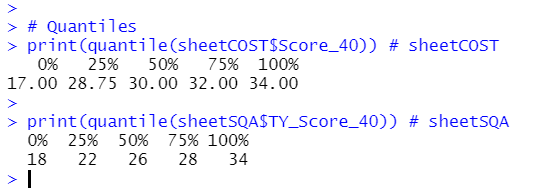
print(getmode(data.iam$TY\_Score\_40))

Output-



Quantiles

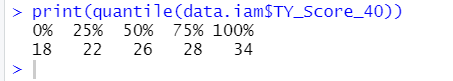
| # Quantiles  print(quantile(sheetCOST$Score\_40)) # sheetCOST  print(quantile(sheetSQA$TY\_Score\_40)) # sheetSQA |
| --- |



Quartile-

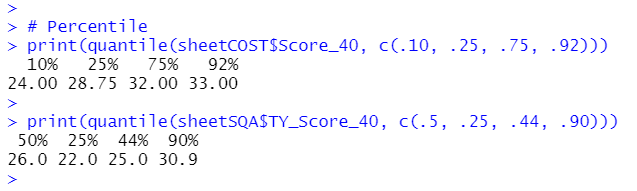
>print(quantile(data.iam$TY\_Score\_40))

Output -



Percentile

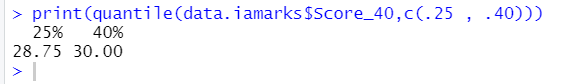
| # Percentile  print(quantile(sheetCOST$Score\_40, c(.10, .25, .75, .92)))  print(quantile(sheetSQA$TY\_Score\_40, c(.5, .25, .44, .90))) |
| --- |



Percentile –

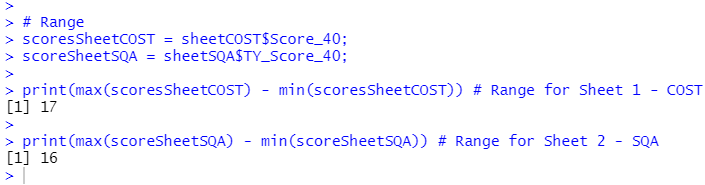
print(quantile(data.iamarks$Score\_40,c(.25 , .40)))

Output -



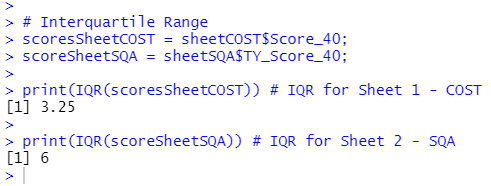
Range

| # Range  scoresSheetCOST = sheetCOST$Score\_40;  scoreSheetSQA = sheetSQA$TY\_Score\_40;  print(max(scoresSheetCOST) - min(scoresSheetCOST)) # Range for Sheet 1 - COST  print(max(scoreSheetSQA) - min(scoreSheetSQA)) # Range for Sheet 2 - SQA |
| --- |



Interquartile Range

| # Interquartile Range  scoresSheetCOST = sheetCOST$Score\_40;  scoreSheetSQA = sheetSQA$TY\_Score\_40;  print(IQR(scoresSheetCOST)) # IQR for Sheet 1 - COST  print(IQR(scoreSheetSQA)) # IQR for Sheet 2 - SQA |
| --- |



Interquartile –

>print(IQR(data.iamarks$Score\_40))

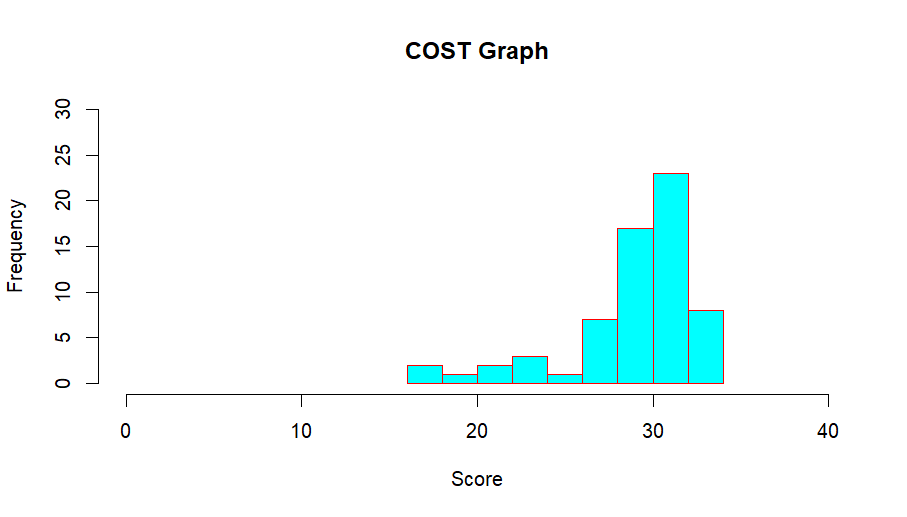
Output –

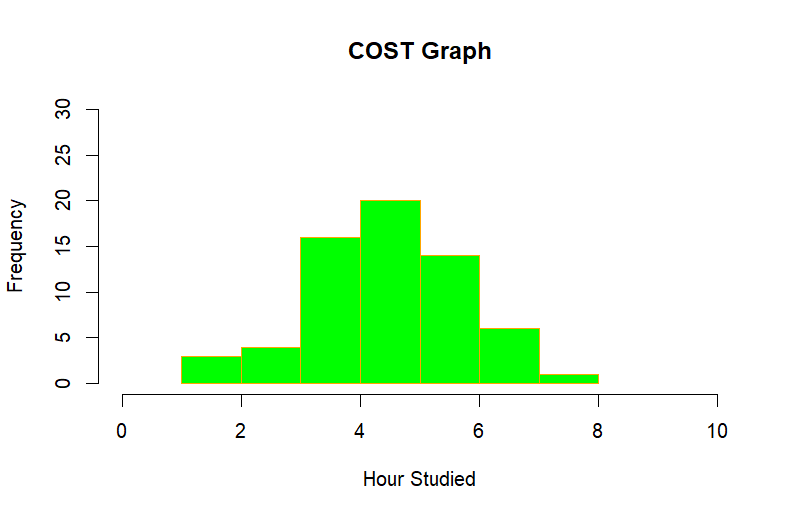


Histogram

Sheet 1 - sheetCOST

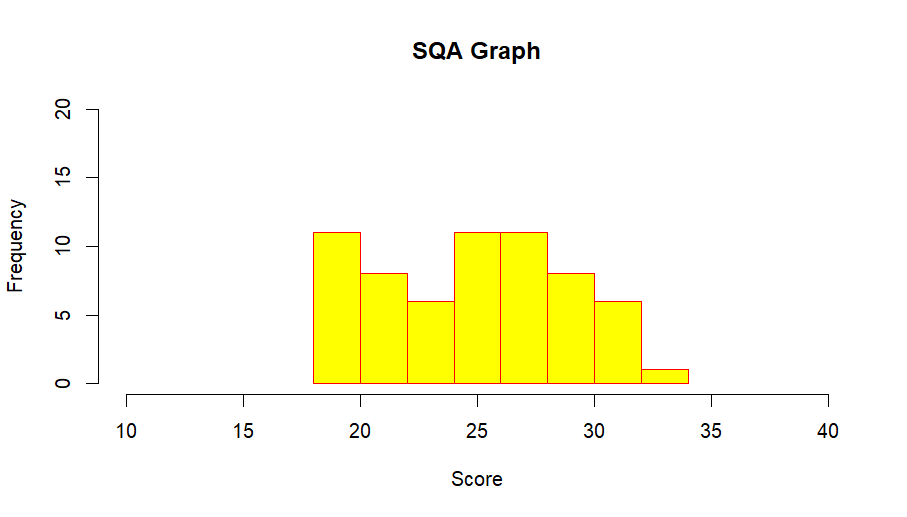
| print(hist(scoreCOST, main = "COST Graph", xlab = "Score", xlim = c(0, 40), ylim = c(0, 30), col = "cyan", border = "red")) # score in COST  print(hist(hrStudyCOST, main = "COST Graph", xlab = "Hour Studied", xlim = c(0, 10), ylim = c(0, 30), col = "green", border = "orange")) # Hour Studied for COST |
| --- |

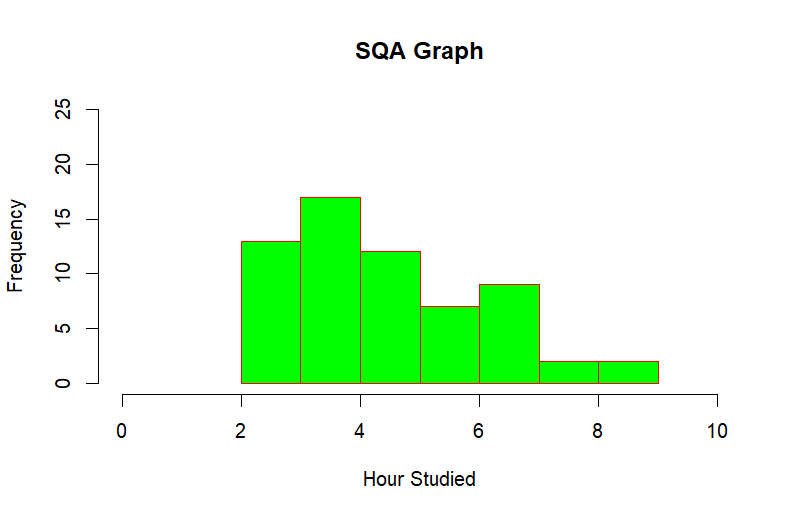




Sheet 2 - sheetSQA

| print(hist(scoreSQA, main = "SQA Graph", xlab = "Score", xlim = c(10, 40), ylim = c(0, 20), col = "yellow", border = "red")) # Score in SQA  print(hist(hrStudySQA, main = "SQA Graph", xlab = "Hour Studied", xlim = c(0, 10), ylim = c(0, 25), col = "green", border = "red")) # Hour Studied for SQA |
| --- |

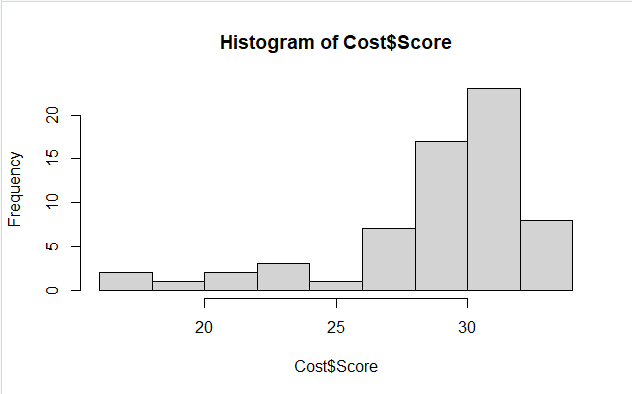




CODE:



HISTOGRAM:



| | > hist(Cost$Score,col = "cyan", border = "red") | | --- | |
| --- | --- |
| > hist(Cost$Score,main='score\_histogram',col = "cyan", border = "red",xlab = 'score',ylab='marks') |
| hist(Cost$Score,main='score\_histogram',col = "cyan", border = "red",xlab = 'score',ylab='marks',xlim = c(10, 40), ylim = c(0, 20)) |

### Practical 5

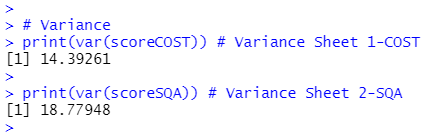
#### Using R import the data from Excel / .CSV file and calculate the standard deviation, variance, co-variance.

Source code

| setwd("F:/3067\_COST PRAC")  print(getwd())  install.packages("xlsx")  library("xlsx")  any(grepl("xlsx", installed.packages()))  # Sheet 1  sheetCOST = read.xlsx('mySheet.xlsx', sheetIndex = 1)  # Sheet 2  sheetSQA = read.xlsx('mySheet.xlsx', sheetIndex = 2)  # Variables used  scoreCOST = sheetCOST$Score\_40  scoreSQA = sheetSQA$TY\_Score\_40  hrStudyCOST = sheetCOST$hour\_studied  hrStudySQA = sheetSQA$hour\_studied |
| --- |

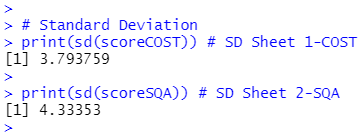
##### variance

| # Variance  print(var(scoreCOST)) # Variance for Sheet 1-COST  print(var(scoreSQA)) # Variance for Sheet 2-SQA |
| --- |



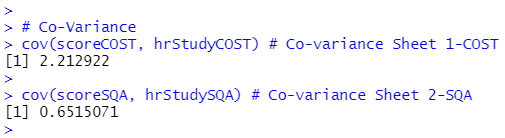
##### Standard Deviation

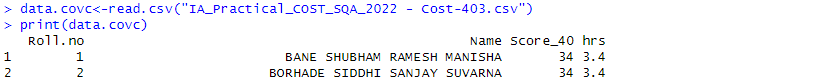
| # Standard Deviation  print(sd(scoreCOST)) # SD for Sheet 1-COST  print(sd(scoreSQA)) # SD for Sheet 2-SQA |
| --- |

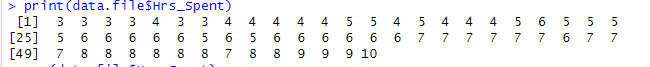


##### Covariance

| # Co-Variance  cov(scoreCOST, hrStudyCOST) # Co-variance for Sheet 1-COST  cov(scoreSQA, hrStudySQA) # Co-variance for Sheet 2-SQA |
| --- |



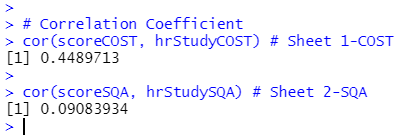






##### Correlation Coefficient

| # Correlation Coefficient  cor(scoreCOST, hrStudyCOST) # Sheet 1-COST  cor(scoreSQA, hrStudySQA) # Sheet 2-SQA |
| --- |



| data.iamarks=read.xlsx('IA\_Practical\_COST\_SQA\_2022.xlsx',sheetIndex=1,header=TRUE)  print(data.iamarks)  Variance -  >print(var(data.iamarks$Score\_40))  Output -    Standard Deviation –  print(sd(data.iamarks$Score\_40))  Output –    Range –  print(max(data.iamarks$Score\_40) - min(data.iamarks$Score\_40) )  Output - |
| --- |

### Practical 6

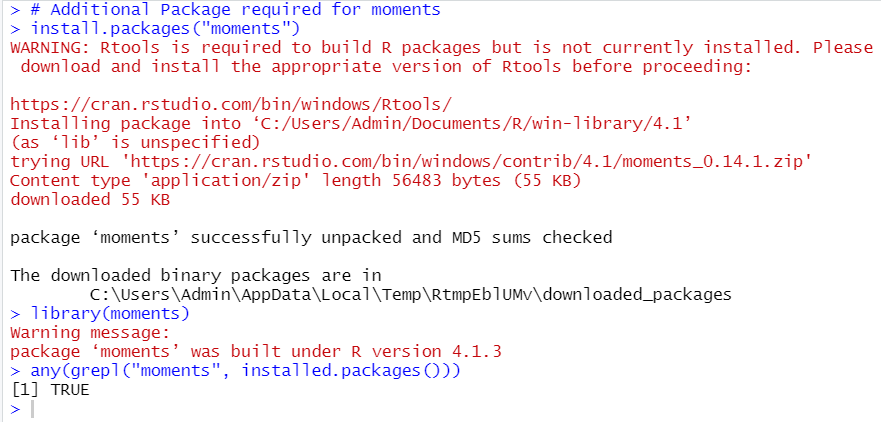
#### Using R import the data from Excel / .CSV file and find Moment, skewness & Koutsis.

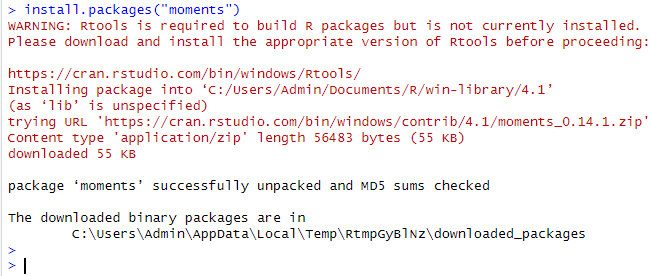
Source code

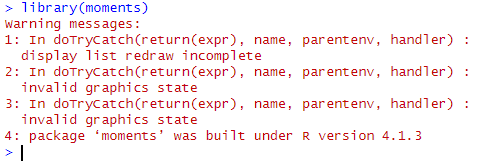
| setwd("F:/3067\_COST PRAC")  getwd()  install.packages("xlsx")  library("xlsx")  any(grepl("xlsx", installed.packages()))  # Sheet 1  sheetCOST = read.xlsx('mySheet.xlsx', sheetIndex = 1)  # Sheet 2  sheetSQA = read.xlsx('mySheet.xlsx', sheetIndex = 2)  # Variables used  scoreCOST = sheetCOST$Score\_40  scoreSQA = sheetSQA$TY\_Score\_40  hrStudyCOST = sheetCOST$hour\_studied  hrStudySQA = sheetSQA$hour\_studied |
| --- |

Additional Package required

| # Additional Package required for moments  install.packages("moments")  library(moments)  any(grepl("moments", installed.packages())) |
| --- |



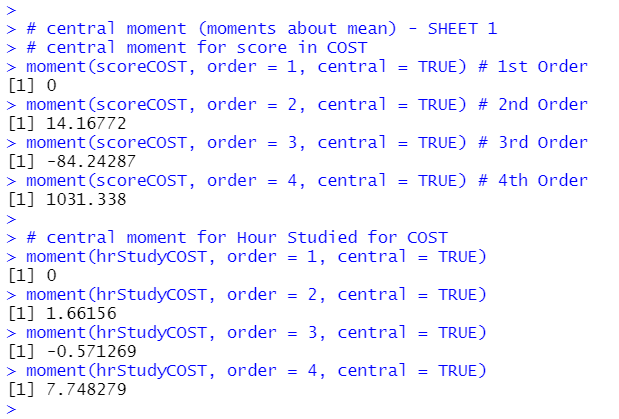






##### Moment

| # central moment (moments about mean) - SHEET 1  # central moment for score in COST  moment(scoreCOST, order = 1, central = TRUE) # 1st Order  moment(scoreCOST, order = 2, central = TRUE) # 2nd Order  moment(scoreCOST, order = 3, central = TRUE) # 3rd Order  moment(scoreCOST, order = 4, central = TRUE) # 4th Order  # central moment for Hour Studied for COST  moment(hrStudyCOST, order = 1, central = TRUE)  moment(hrStudyCOST, order = 2, central = TRUE)  moment(hrStudyCOST, order = 3, central = TRUE)  moment(hrStudyCOST, order = 4, central = TRUE) |
| --- |



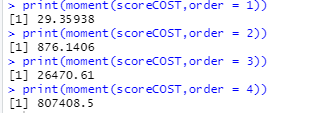
print(moment(scoreCOST,order = 1))

print(moment(scoreCOST,order = 2))

print(moment(scoreCOST,order = 3))

print(moment(scoreCOST,order = 4))

Output :-



Problem

Find the third central moment of eruption duration in the data set faithful.

To execute the function we need to first install packages of moments

>library(moments)

> duration = faithful$eruptions

> moment(duration, order=3)

Output:-



Problem

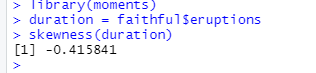
Find the skewness of eruption duration in the data set faithful.

> library(moments)

> duration = faithful$eruptions

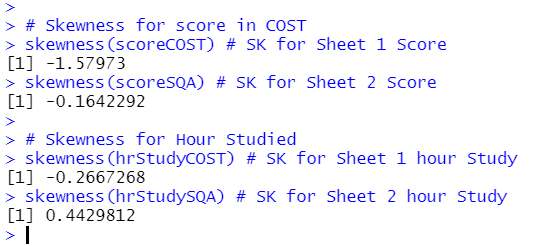
> skewness(duration)

Output:-



##### Skewness

| # Skewness for score in COST  skewness(scoreCOST) # SK for Sheet 1 Score  skewness(scoreSQA) # SK for Sheet 2 Score  # Skewness for Hour Studied  skewness(hrStudyCOST) # SK for Sheet 1 hour Study  skewness(hrStudySQA) # SK for Sheet 2 hour Study |
| --- |



Source Code: -

#Skewness

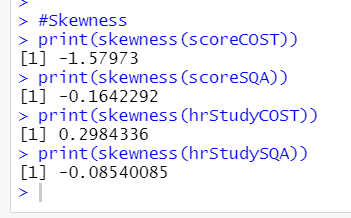
print(skewness(scoreCOST))

print(skewness(scoreSQA))

print(skewness(hrStudyCOST))

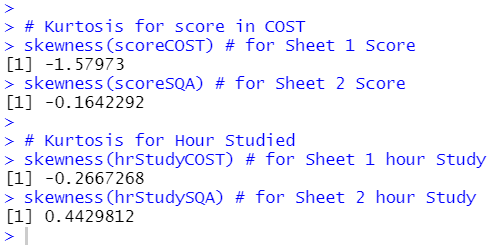
print(skewness(hrStudySQA))

Output: -



##### Kurtosis

| # Kurtosis for score in COST  skewness(scoreCOST) # for Sheet 1 Score  skewness(scoreSQA) # for Sheet 2 Score  # Kurtosis for Hour Studied  skewness(hrStudyCOST) # for Sheet 1 hour Study  skewness(hrStudySQA) # for Sheet 2 hour Study |
| --- |



Source Code :-

#Kurtosis

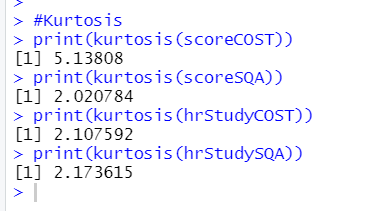
print(kurtosis(scoreCOST))

print(kurtosis(scoreSQA))

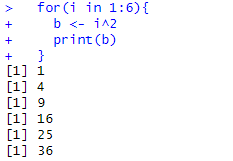
print(kurtosis(hrStudyCOST))

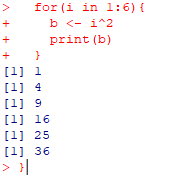
print(kurtosis(hrStudySQA))

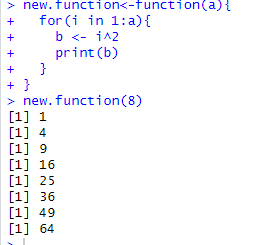
Output :-

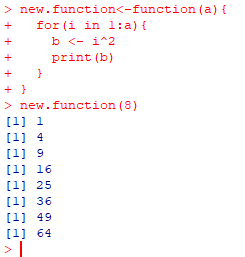


## User defined functions:









### Practical 7

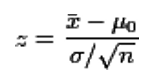
Lower Tail Test of Population Mean with Known Variance

The null hypothesis of the lower tail test of the population mean can be expressed as follows:



where μ0 is a hypothesized lower bound of the true population mean μ.

Let us define the test statistic z in terms of the sample mean, the sample size and the population standard deviation σ :



Then the null hypothesis of the lower tail test is to be rejected if z ≤−zα , where zα is the 100(1 − α) percentile of the standard normal distribution.

Z more than 30 then z distribution

Less than 30 then tdistribution

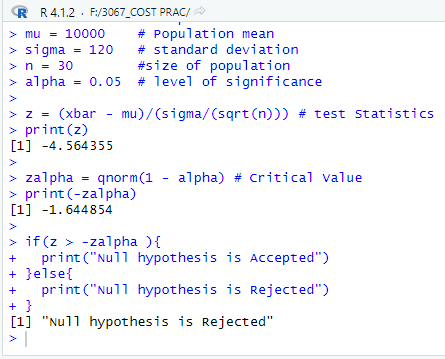
#### Z-Statistics:

1. 1.Write a Script for hypothetical testing.

#### 1.Suppose the manufacturer claims that the mean lifetime of a light bulb is more than 10,000 hours. In a sample of 30 light bulbs, it was found that they only last 9,900 hours on average. Assume the population standard deviation is 120 hours. At .05 significance level, can we reject the claim by the manufacturer?

Code:

| #3067\_Sudesh Rajbhar  # Null hypothesis H0 ; mu >= 10000 Alternate hypothesis is H1 : mu < 10000  xbar = 9900 # sample mean  mu = 10000 # Population mean  sigma = 120 # standard deviation  n = 30 #size of population  alpha = 0.05 # level of significance  z = (xbar - mu)/(sigma/(sqrt(n))) # test Statistics  print(z)  zalpha = qnorm(1 - alpha) # Critical Value  print(-zalpha)  if(z > -zalpha ){  print("Null hypothesis is Accepted")  }else{  print("Null hypothesis is Rejected")  } |
| --- |





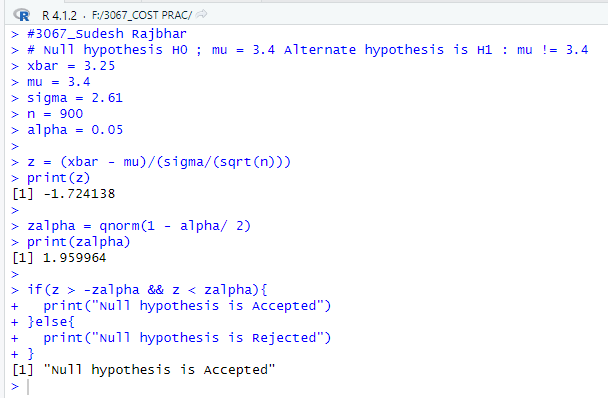
The test statistic -4.5644 is less than the critical value of -1.6449. Hence, at the .05 significance level, we reject the claim that the mean lifetime of a light bulb is above 10,000 hours.

1. 2.Write a Script for hypothetical testing.

#### 2.A sample of 900 members has a mean 3.4 cms and s.d 2.61 cms. Can the sample be regarded as one drawn from the population with mean 3.25 cms? Using the level of significance as 0.05, is the claim acceptable? (Two tailed test)

Code:

| #3067\_Sudesh Rajbhar  # Null hypothesis H0 ; mu = 3.4 Alternate hypothesis is H1 : mu != 3.4  xbar = 3.25  mu = 3.4  sigma = 2.61  n = 900  alpha = 0.05  z = (xbar - mu)/(sigma/(sqrt(n)))  print(z)  zalpha = qnorm(1 - alpha/ 2)  print(zalpha)  if(z > -zalpha && z < zalpha){  print("Null hypothesis is Accepted")  }else{  print("Null hypothesis is Rejected")  } |
| --- |



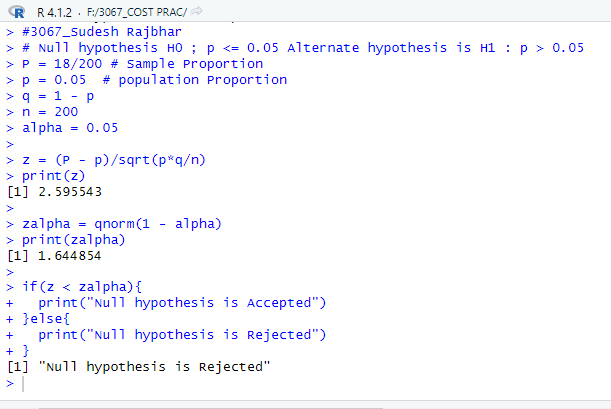
The null hypothesis is selected and the sample population has a mean of 3.4.

3.Write a Script for hypothetical testing.

#### 3.A manufacturer claimed that at least 95% of the equipment which he supplied to a factory conformed to specification. An examination of a sample of 200 pieces of equipment revealed that 18 were faulty. Test this claim at a significance level of 0.05.

Code:

| #3067\_Sudesh Rajbhar  # Null hypothesis H0 ; p <= 0.05 Alternate hypothesis is H1 : p > 0.05  P = 18/200 # Sample Proportion  p = 0.05 # population Proportion  q = 1 - p  n = 200  alpha = 0.05  z = (P - p)/sqrt(p\*q/n)  print(z)  zalpha = qnorm(1 - alpha)  print(zalpha)  if(z < zalpha){  print("Null hypothesis is Accepted")  }else{  print("Null hypothesis is Rejected")  } |
| --- |



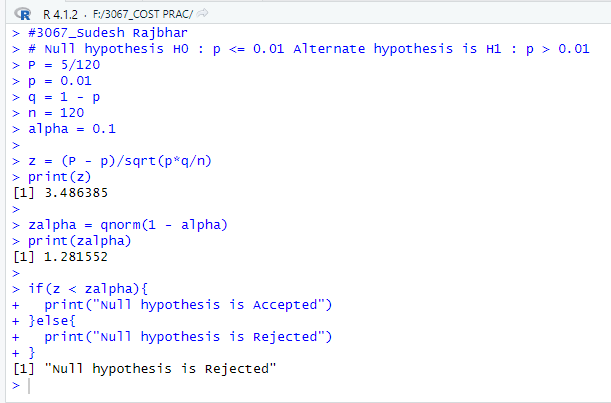
The alternate hypothesis is accepted and population proportion of faulty equipments is more than 0.05

#### 4.A bottle manufacturing process is ‘under control’ if no more than 1% of the bottles are defective. A random sample of 120 bottles showed 5 to be defective. DO this data indicative that the process is out of control at 10% significance level. (Sample proportion P=5/120 =0.042, Population Proportion p=.01 )

Code:

| #3067\_Sudesh Rajbhar  # Null hypothesis H0 : p <= 0.01 Alternate hypothesis is H1 : p > 0.01  P = 5/120  p = 0.01  q = 1 - p  n = 120  alpha = 0.1  z = (P - p)/sqrt(p\*q/n)  print(z)  zalpha = qnorm(1 - alpha)  print(zalpha)  if(z < zalpha){  print("Null hypothesis is Accepted")  }else{  print("Null hypothesis is Rejected")  } |
| --- |

Output:

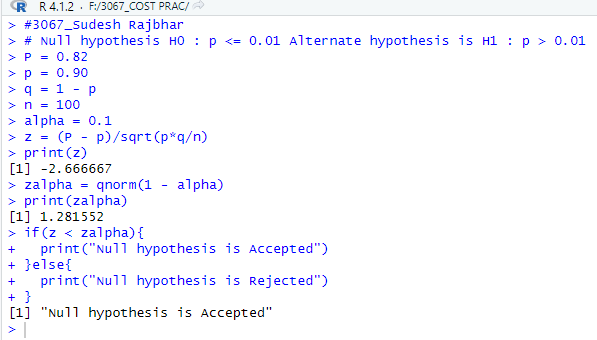


#### 5.A survey claims that 9 out of 10 doctors recommend aspirin for their patients with headaches. To test this claim, a random sample of 100 doctors is obtained. Of these 100 doctors, 82 indicate that they recommend aspirin. Is this claim accurate? Use alpha = 0.05.

Code:

| #3067\_Sudesh Rajbhar  # Null hypothesis H0 : p <= 0.01 Alternate hypothesis is H1 : p > 0.01  P = 0.82  p = 0.90  q = 1 - p  n = 100  alpha = 0.1  z = (P - p)/sqrt(p\*q/n)  print(z)  zalpha = qnorm(1 - alpha)  print(zalpha)  if(z < zalpha){  print("Null hypothesis is Accepted")  }else{  print("Null hypothesis is Rejected")  } |
| --- |

Output:



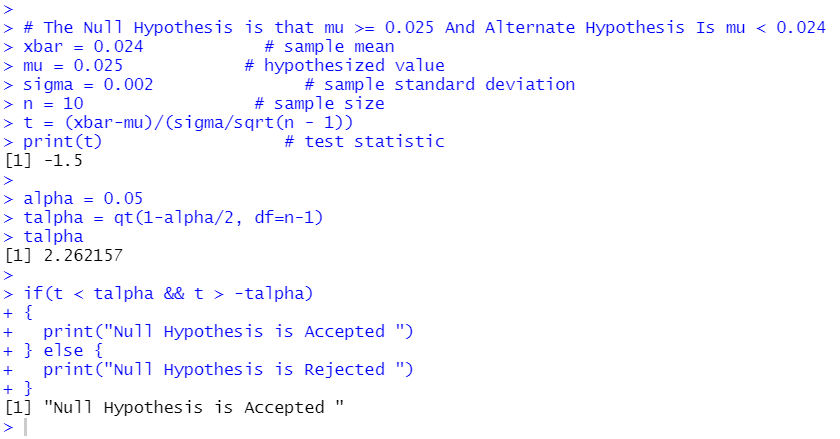
#### T-Statistics:

#### 1.A machine is designed to produce insulating washers for electrical devices of average thickness of 0.025 cm. A random sample of 10 washers was found to have an average thickness of 0.024 cm with a standard deviation of 0.002 cm. Test the significance of the deviation. Level of significance is 0.05. (Two tailed test) Use qt() function (25< then qt distribution or else z distribution)

Code:

| # The Null Hypothesis is that mu >= 0.025 And Alternate Hypothesis Is mu < 0.024  xbar = 0.024 # sample mean  mu = 0.025 # hypothesized value  sigma = 0.002 # sample standard deviation  n = 10 # sample size  t = (xbar-mu)/(sigma/sqrt(n - 1))  print(t) # test statistic  alpha = 0.05  talpha = qt(1-alpha/2, df=n-1)  talpha  if( t < talpha && t > -talpha)  {  print("Null Hypothesis is Accepted ")  } else {  print("Null Hypothesis is Rejected ")  } |
| --- |

Output:

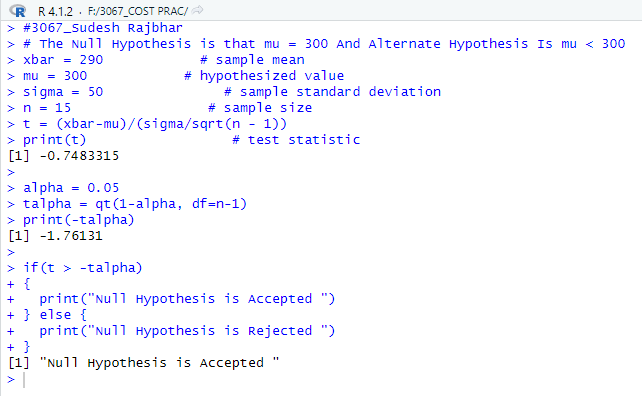


#### 2.A CEO of light bulb manufacturing comp claims that an average light bulb lasts 300 days. A researcher randomly selects 15 bulbs for testing. The sample bulbs last an average of 290 days with a S.D of 50 days. If the CEO claim were true, what is the probability that 15 randomly selected bulbs would have an average life of no more than 290 days. (Lower tail test)

Code:

| #3067\_Sudesh Rajbhar  # The Null Hypothesis is that mu = 300 And Alternate Hypothesis Is mu < 300  xbar = 290 # sample mean  mu = 300 # hypothesized value  sigma = 50 # sample standard deviation  n = 15 # sample size  t = (xbar-mu)/(sigma/sqrt(n - 1))  print(t) # test statistic  alpha = 0.05  talpha = qt(1-alpha, df=n-1)  print(-talpha)  if(t > -talpha)  {  print("Null Hypothesis is Accepted ")  } else {  print("Null Hypothesis is Rejected ")  } |
| --- |

Output:

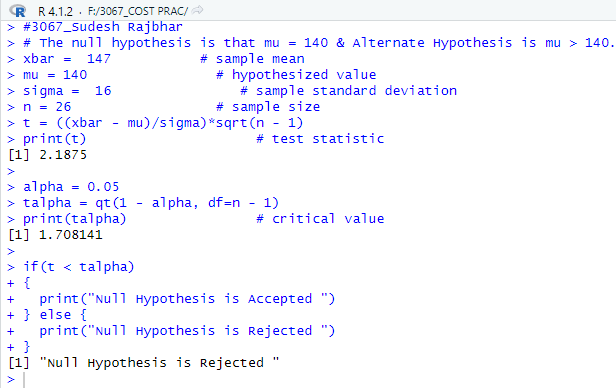


#### 3.A soap manufacturer company was distributing a particular brand of soap through a large number of retail shops. Before a heavy advertisement campaign, the mean sales per shop was 140 dozen. After the campaign a sample of 26 shops was taken and the mean sales figure was found to be 147 dozen with standard deviation of 16. Can you consider the advertisement effective?

Code:

| #3067\_Sudesh Rajbhar  # The null hypothesis is that mu = 140 & Alternate Hypothesis is mu > 140.  xbar = 147 # sample mean  mu = 140 # hypothesized value  sigma = 16 # sample standard deviation  n = 26 # sample size  t = ((xbar - mu)/sigma)\*sqrt(n - 1)  print(t) # test statistic  alpha = 0.05  talpha = qt(1 - alpha, df=n - 1)  print(talpha) # critical value  if(t < talpha)  {  print("Null Hypothesis is Accepted ")  } else {  print("Null Hypothesis is Rejected ")  } |
| --- |

Output:



### Practical 8

#### Perform the Chi-squared Test.

**Chi squared Test**

Chi-Square test is a statistical method to determine if two categorical variables have a

significant correlation between them.

**Properties of Chi Square Distribution-**

1. The mean of a Chi Square distribution is equal to the degrees of freedom.

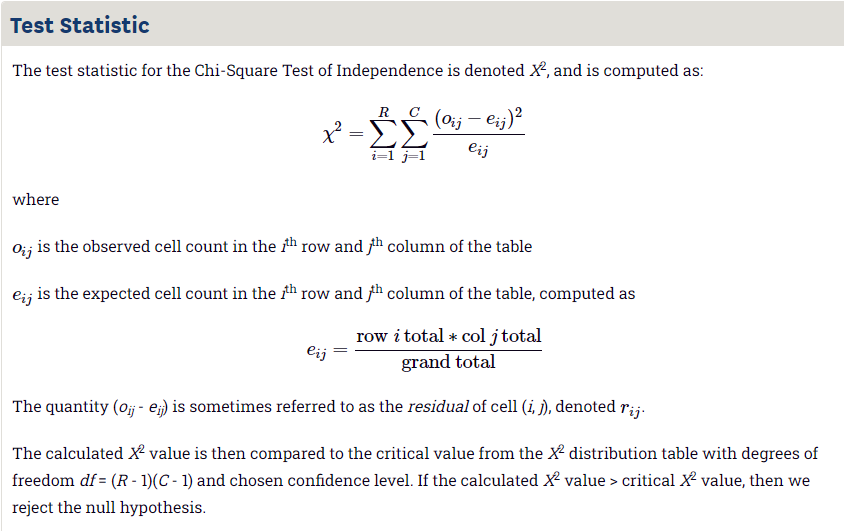
2. Chi Square curve is always positively skewed

3. Standard deviation of Chi Square distribution is , where v is the degree of freedom.

4. Chi Square values increase with the increase in degree of freedom.

5. With different degrees of freedom, the shape of the curve will be different.

| **Chi-Square Test of Independence** The Chi-Square Test of Independence determines whether there is an association between categorical variables (i.e., whether the variables are independent or related). It is a nonparametric test.  This test is also known as:   * **Chi-Square Test of Association.**   This test utilizes a contingency table to analyze the data. A contingency table (also known as a cross-tabulation, crosstab, or two-way table) is an arrangement in which data is classified according to two categorical variables. The categories for one variable appear in the rows, and the categories for the other variable appear in columns. Each variable must have two or more categories. Each cell reflects the total count of cases for a specific pair of categories. |
| --- |

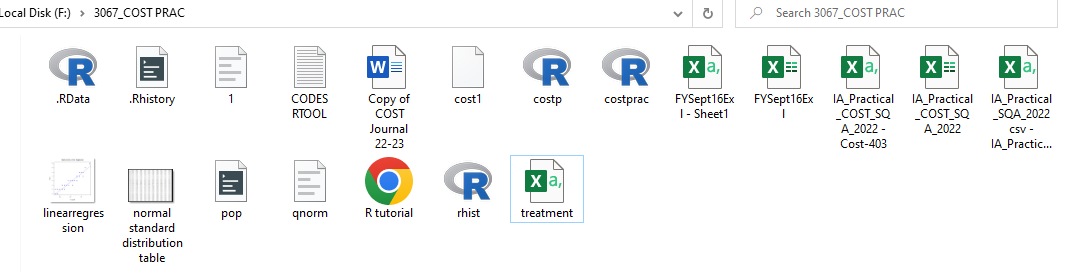


**Syntax:**

**The function used for performing chi-Square test is chisq.test()**

#### 

#### 1.Suppose we have 105 patients under study and 50 of them were treated with the drug. Moreover, the remaining 55 patients were kept under control samples. Thus, the health condition of all patients was checked after a week. With the following table, we can assess if their condition has improved or not. By observing this table, one can tell if the drug had a positive effect on the patient.

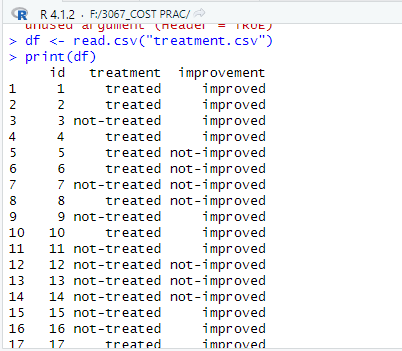


Treatment file:

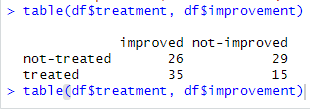
| <https://docs.google.com/spreadsheets/d/1HATM2BM6yzwzQMxd8w5TNjN-TMdyU4Jy/edit?usp=sharing&ouid=101847589461082648003&rtpof=true&sd=true> |
| --- |

Code:

| df <- read.csv("treatment.csv")  table(df$treatment, df$improvement)  print(df) |
| --- |



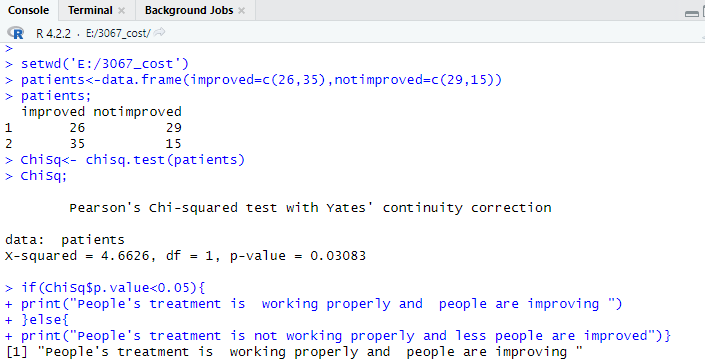
**Table:**



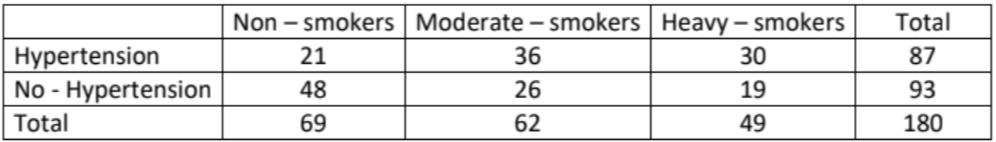
Code for Calculation:

| # Chi-sq test  chisq.test(df$treatment, df$improvement, correct=FALSE) |
| --- |

| patients<-data.frame(improved=c(26,35),notimproved=c(29,15))  patients;  ChiSq<- chisq.test(patients)  ChiSq;  if(ChiSq$p.value<0.05){  print("People's treatment is working properly and people are improving ")  }else{  print("People's treatment is not working properly and less people are improved")} |
| --- |



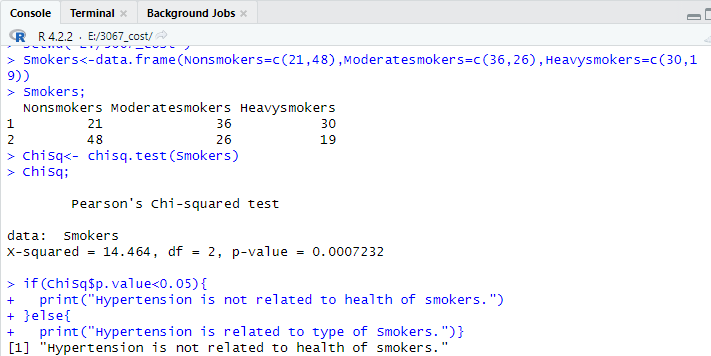
#### 2.In an experiment to study the independence of hypertension on smoking habits, the following data are taken from 180 individuals:



Code:

| Smokers<-data.frame(Nonsmokers=c(21,48),Moderatesmokers=c(36,26),Heavysmokers=c(30,19))  Smokers;  ChiSq<- chisq.test(Smokers)  ChiSq;  if(ChiSq$p.value<0.05){  print("Hypertension is not related to health of smokers.")  }else{  print("Hypertension is related to type of Smokers.")} |
| --- |

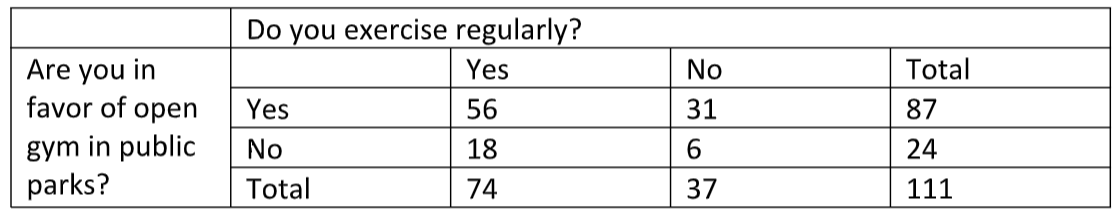
Output:



#### 3.In a sample survey of public opinion, answer to the questions

a. Do you exercise regularly?

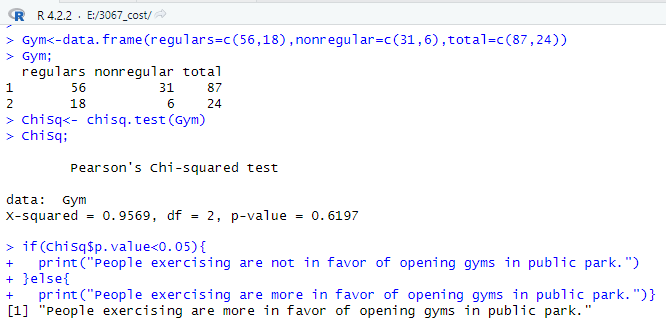
b. Are you in favor of open gyms in public parks?



Code:

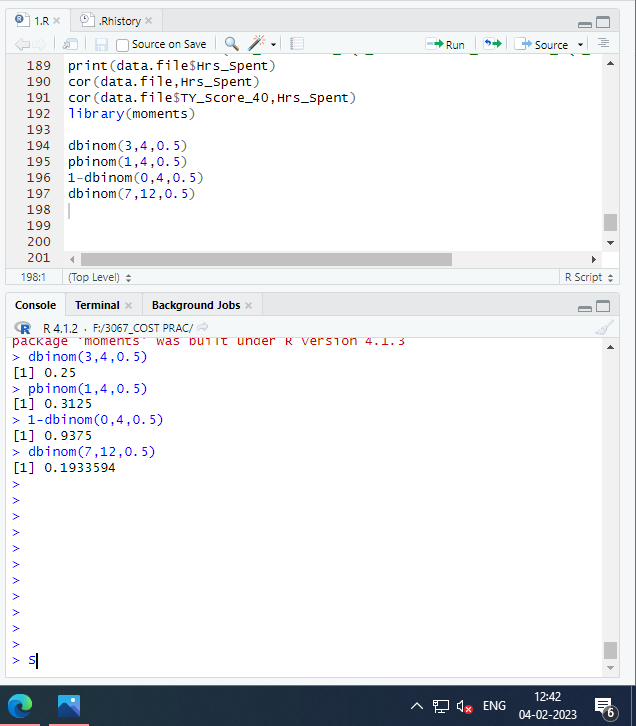
| Gym<-data.frame(regulars=c(56,18),nonregular=c(31,6),total=c(87,24))  Gym;  ChiSq<- chisq.test(Gym)  ChiSq;  if(ChiSq$p.value<0.05){  print("People exercising are not in favor of opening gyms in public park.")  }else{  print("People exercising are more in favor of opening gyms in public park.")} |
| --- |

Output:



### Practical 9

1. Binomial and Normal distribution.



## dbinom()-dbinom(x,n,p)

This function gives the probability density distribution at each point

#### 

#### q1)Here, we are using the example to the binomial distribution to find the probability where a person makes 70% of his throw attempts, and if he shoots 20 throws, so what will be the probability that the person makes exactly 12 of them attempt, so here we are simply using the dbinom function and passing the given statements properties as its parameters and further getting the result in the R. function.

| dbinom(12,20,0.7) |
| --- |



#### q2)In this example, we are simply calculating the probability to get the heads exactly 17 times if the coin is tossed 50 times fairly using the dbinom function. Since the coin is tossed fairly the prob parameter to the function is passed to be 0.5.

| dbinom(17,50,0.5) |
| --- |



## 

## 

## 

## 

## pbinom()

This function gives the cumulative probability of an event. It is a single value representing the probability.

#### 

#### q3)Under this example, we are calculating the probability to get a head more than 3 times if the coin is flipped fairly 10 times using the pbinom() function. Since the coin is tossed fairly the prob parameter to the function is passed to be 0.5.

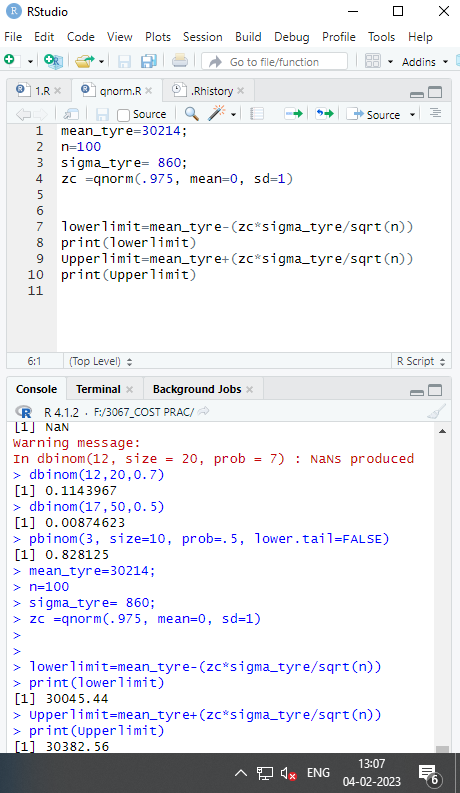
| > pbinom(3, size=10, prob=.5, lower.tail=FALSE) |
| --- |



## qnorm: quantile function of the normal distribution

#### q4)The quality control manager of a tyre company has sampled 100 tyres and has found that the mean lifetime of a tyre to be 30,214 Kms. The population is 860. Construct a 95% confidence interval for the mean lifetime for this particular brand of tyres.

| mean\_tyre=30214;  > n=100  > sigma\_tyre= 860;  > zc =qnorm(.975, mean=0, sd=1)  >  >  > lower limit=mean\_tyre-(zc\*sigma\_tyre/sqrt(n))  > print(lower limit) |
| --- |



## pnorm: pnorm(x,mean, sd)- The pnorm function gives the Cumulative Distribution Function (CDF) of the Normal distribution in R, which is the probability that the variable X. X takes a value lower or equal to x.By default lower tail =TRUE

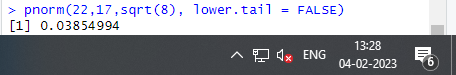
#### q5)The weekly distances, D km, a leopard walks in search of food are thought to be Normally distributed with a mean of 62 and a standard deviation of 10. Find the probability that in a given week a leopard will walk less than 80 km.

| > pnorm(80,62,10) |
| --- |



#### q6) The time, T minutes, Jason takes to cycle to school is Normally distributed with a mean of 17 and a variance of 8 . Find the probability that on a given day Jason will take over 22 minutes to cycle to school.

| > pnorm(22,17,sqrt(8), lower.tail = FALSE) |
| --- |



#### q7)calculate the percentage of students at this school who are taller than 75 inches height of males at a certain school is normally distributed with a mean of μ=70 inches and a standard deviation of σ =3 inches using the pnorm() function in the R.

| > pnorm(75,70,3) |
| --- |

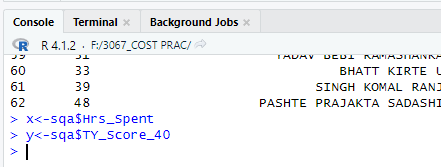


### Practical 10

#### Linear Regression

Connect the File to perform :

| setwd("F:/3067\_COST PRAC")  getwd()  sqa<-read.csv('IA\_Practical\_SQA\_2022csv - IA\_Practical\_SQA\_2022csv.csv')  class(sqa)  sqa  x<-sqa$Hrs\_Spent  y<-sqa$TY\_Score\_40 |
| --- |

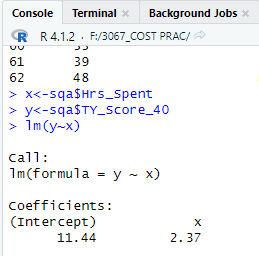


lm() Function

This function creates the relationship model between the predictor and the response variable.

| x<-sqa$Hrs\_Spent  y<-sqa$TY\_Score\_40 |
| --- |

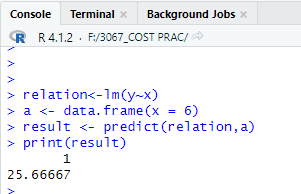
Output:



predict() Function

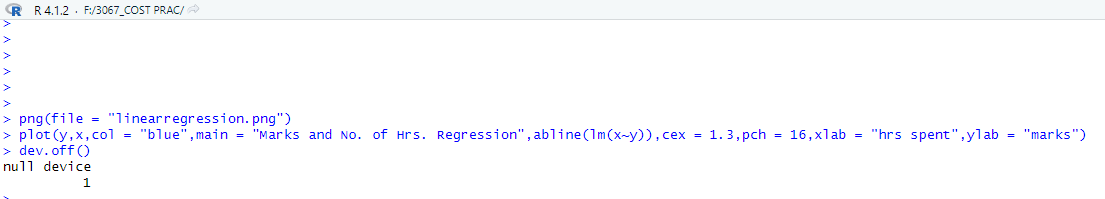
The basic syntax for predict() in linear regression is −

| #find marks by studying 6 hrs  x<-sqa$Hrs\_Spent  y<-sqa$TY\_Score\_40  relation<-lm(y~x)  a <- data.frame(x = 6)  result <- predict(relation,a)  print(result) |
| --- |

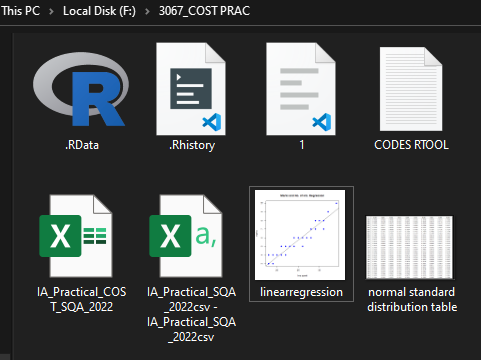


Visualize the Regression Graphically

| #plot the function  # Give the chart file a name.  png(file = "linearregression.png")  # Plot the chart.  plot(y,x,col = "blue",main = "Marks and No. of Hrs. Regression",  abline(lm(x~y)),cex = 1.3,pch = 16,xlab = "hrs spent",ylab = "marks")  # Save the file.  dev.off() |
| --- |



File saved In Folder:



File of png:

