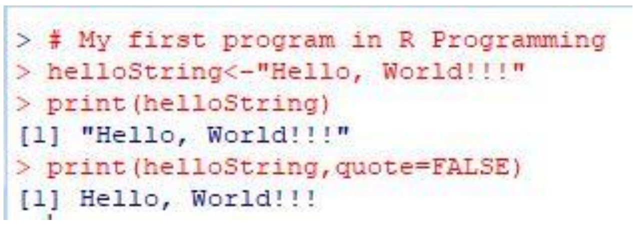
PRACTICAL NO. 1

BASICS OF R

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

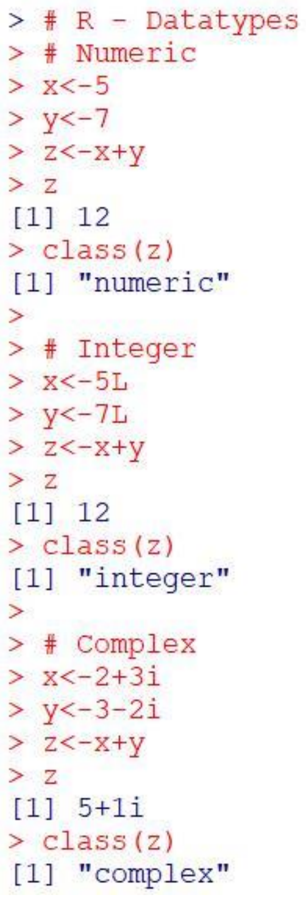
SOURCE CODE & OUTPUT:

1. **Hello World Program:**

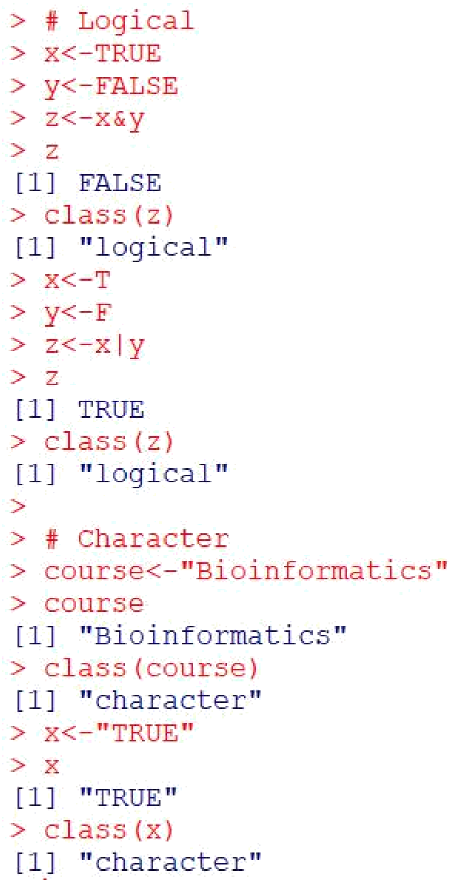


```
> # My first program in R Programming
> helloString<-"Hello, World!!!"
> print(helloString)
[1] "Hello, World!!!"
> print(helloString,quote=FALSE)
[1] Hello, World!!!
```

2. **R – Datatypes:**
   i. **Numeric**
   ii. **Integer**
   iii. **Complex**
   iv. **Logical**
   v. **Character**



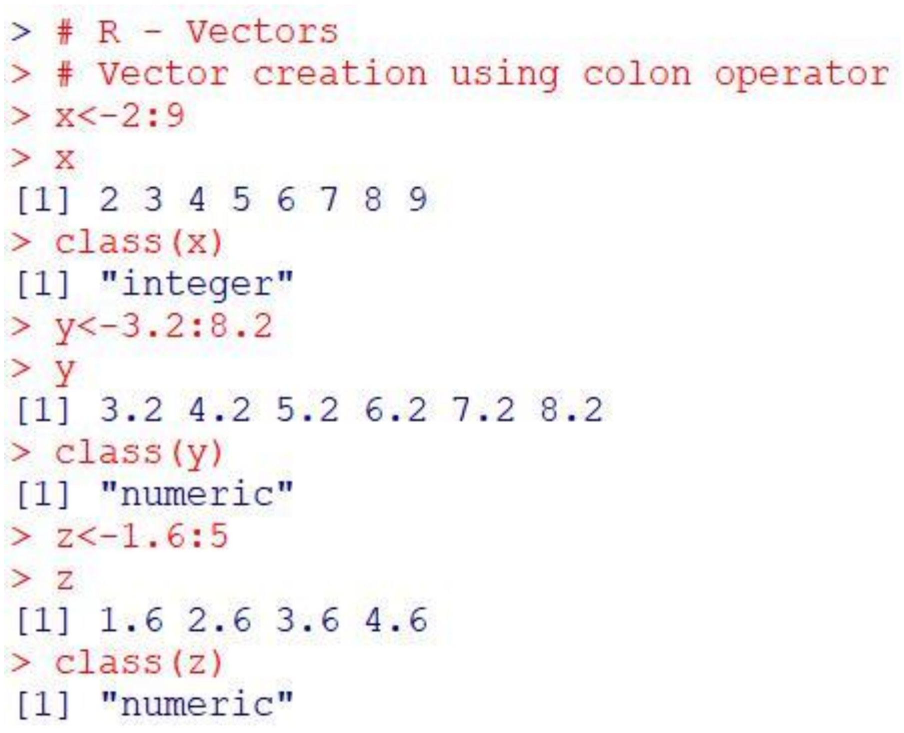
```
> # R - Datatypes
> # Numeric
> x<-5
> y<-7
> z<-x+y
> z
[1] 12
> class(z)
[1] "numeric"
>
> # Integer
> x<-5L
> y<-7L
> z<-x+y
> z
[1] 12
> class(z)
[1] "integer"
>
> # Complex
> x<-2+3i
> y<-3-2i
> z<-x+y
> z
[1] 5+1i
> class(z)
[1] "complex"
```



```
> # Logical
> x<-TRUE
> y<-FALSE
> z<-x&y
> z
[1] FALSE
> class(z)
[1] "logical"
> x<-T
> y<-F
> z<-x|y
> z
[1] TRUE
> class(z)
[1] "logical"
>
> # Character
> course<-"Bioinformatics"
> course
[1] "Bioinformatics"
> class(course)
[1] "character"
> x<-"TRUE"
> x
[1] "TRUE"
> class(x)
[1] "character"
```

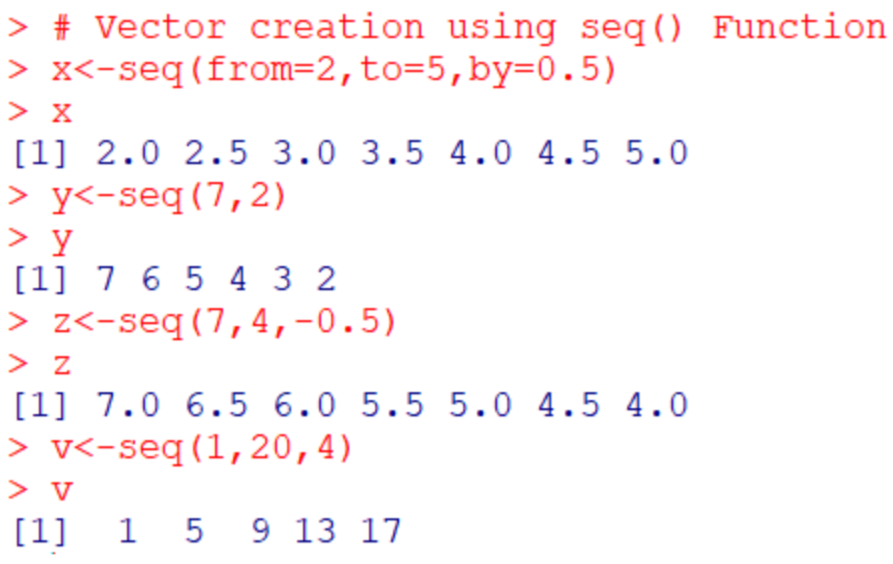
3. R – Vectors:

i. Vector Creation Using Colon Operator:

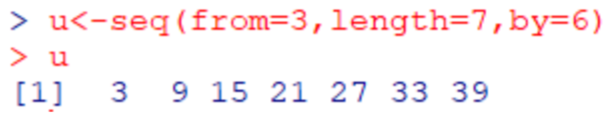


```
> # R - Vectors
> # Vector creation using colon operator
> x<-2:9
> x
[1] 2 3 4 5 6 7 8 9
> class(x)
[1] "integer"
> y<-3.2:8.2
> y
[1] 3.2 4.2 5.2 6.2 7.2 8.2
> class(y)
[1] "numeric"
> z<-1.6:5
> z
[1] 1.6 2.6 3.6 4.6
> class(z)
[1] "numeric"
```

ii. **Vector Creation Using seq() Function:**

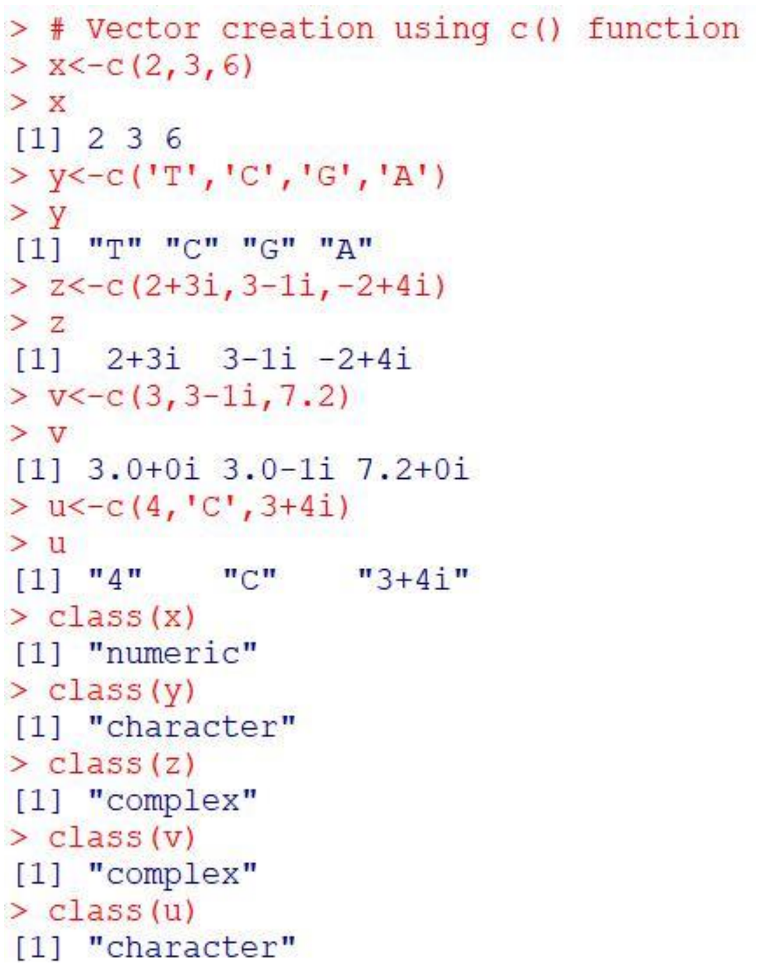


```
> # Vector creation using seq() Function
> x<-seq(from=2,to=5,by=0.5)
> x
[1] 2.0 2.5 3.0 3.5 4.0 4.5 5.0
> y<-seq(7,2)
> y
[1] 7 6 5 4 3 2
> z<-seq(7,4,-0.5)
> z
[1] 7.0 6.5 6.0 5.5 5.0 4.5 4.0
> v<-seq(1,20,4)
> v
[1]  1  5  9 13 17
```



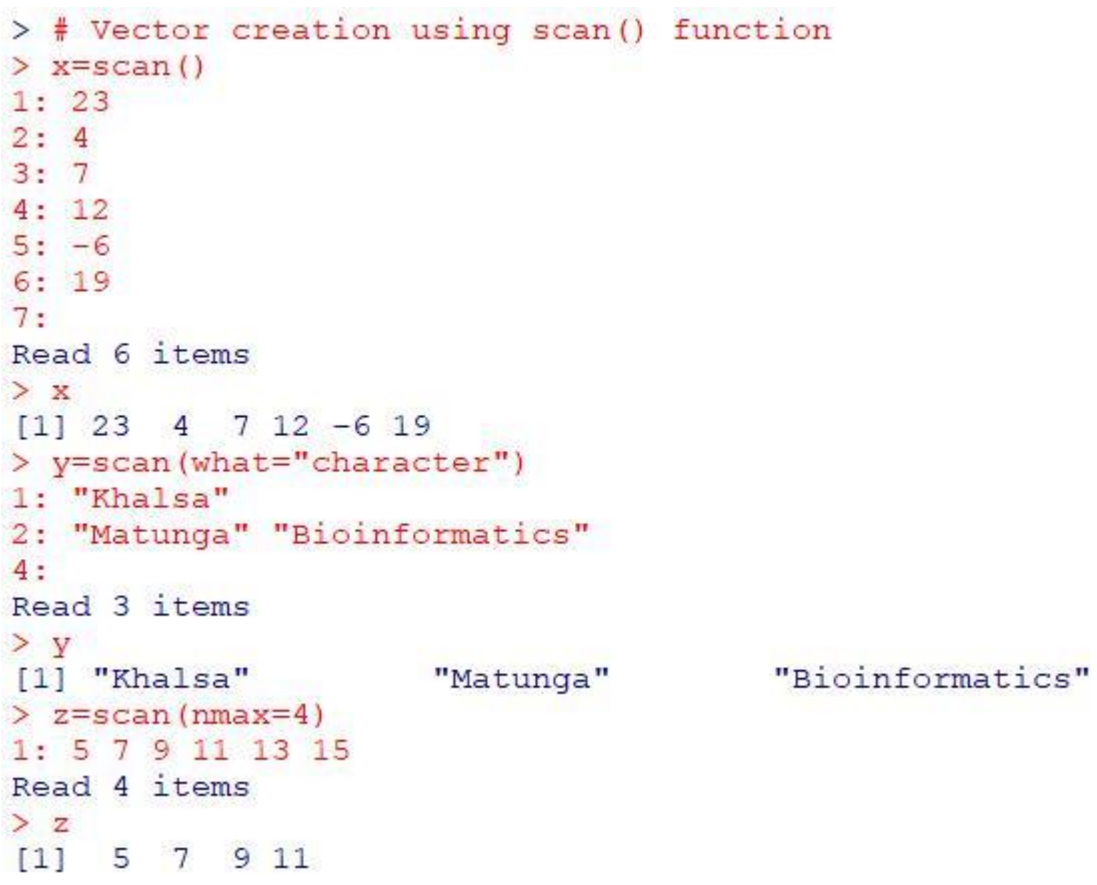
```
> u<-seq(from=3,length=7,by=6)
> u
[1]  3  9 15 21 27 33 39
```

iii. **Vector Creation Using c() Function:**



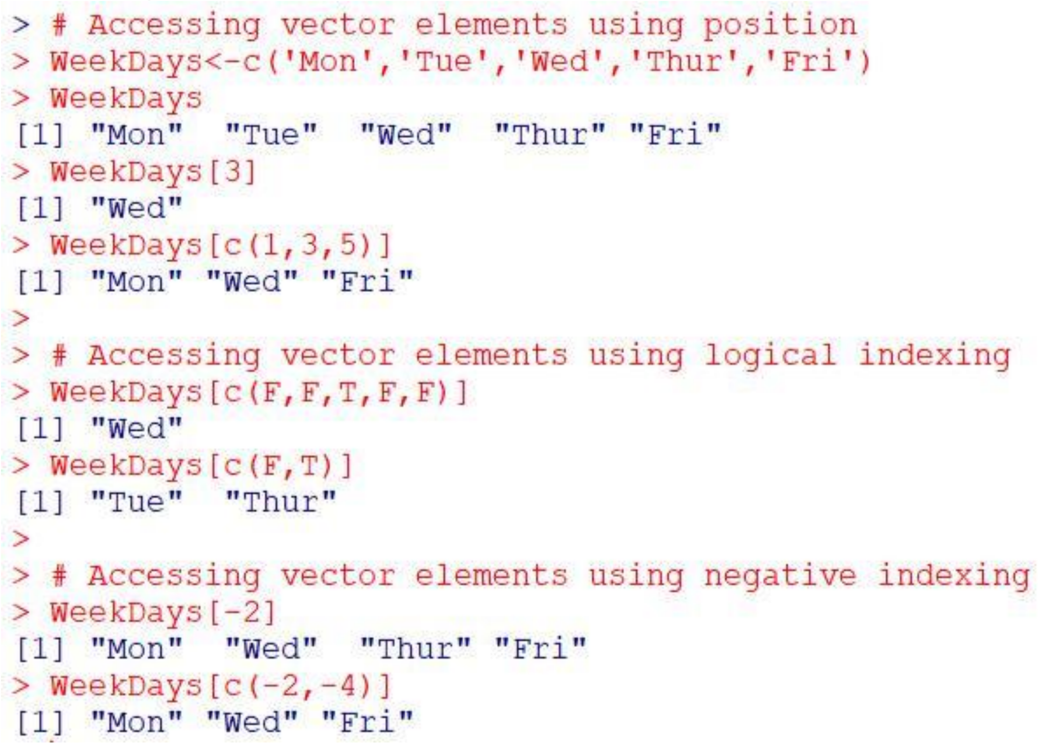
```
> # Vector creation using c() function
> x<-c(2,3,6)
> x
[1] 2 3 6
> y<-c('T','C','G','A')
> y
[1] "T" "C" "G" "A"
> z<-c(2+3i,3-1i,-2+4i)
> z
[1]  2+3i  3-1i -2+4i
> v<-c(3,3-1i,7.2)
> v
[1] 3.0+0i 3.0-1i 7.2+0i
> u<-c(4,'C',3+4i)
> u
[1] "4"    "C"    "3+4i"
> class(x)
[1] "numeric"
> class(y)
[1] "character"
> class(z)
[1] "complex"
> class(v)
[1] "complex"
> class(u)
[1] "character"
```

iv. **Vector Creation Using scan() Function:**



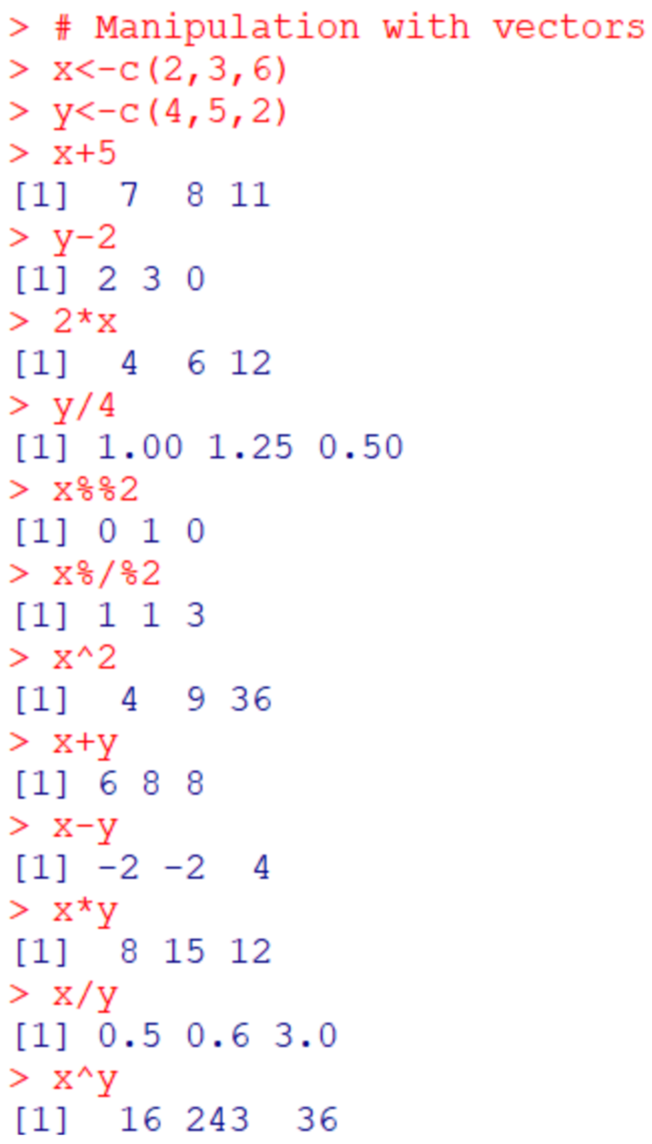
```
> # Vector creation using scan() function
> x=scan()
1: 23
2: 4
3: 7
4: 12
5: -6
6: 19
7:
Read 6 items
> x
[1] 23  4  7 12 -6 19
> y=scan(what="character")
1: "Khalsa"
2: "Matunga" "Bioinformatics"
4:
Read 3 items
> y
[1] "Khalsa"         "Matunga"         "Bioinformatics"
> z=scan(nmax=4)
1: 5 7 9 11 13 15
Read 4 items
> z
[1]  5  7  9 11
```

- **Accessing Vector Elements:**

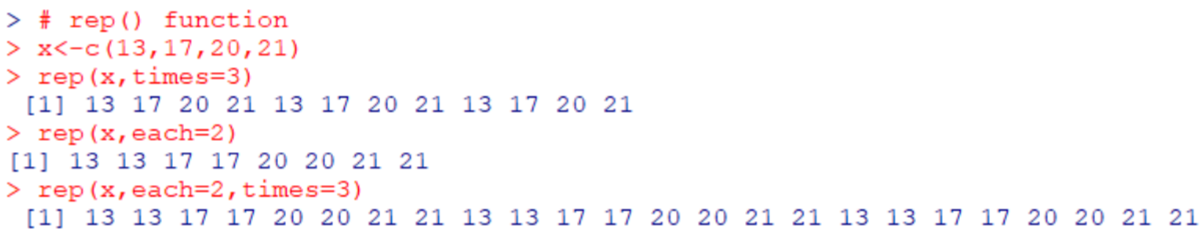


```
> # Accessing vector elements using position
> WeekDays<-c('Mon','Tue','Wed','Thur','Fri')
> WeekDays
[1] "Mon"  "Tue"  "Wed"  "Thur" "Fri"
> WeekDays[3]
[1] "Wed"
> WeekDays[c(1,3,5)]
[1] "Mon" "Wed" "Fri"
>
> # Accessing vector elements using logical indexing
> WeekDays[c(F,F,T,F,F)]
[1] "Wed"
> WeekDays[c(F,T)]
[1] "Tue"  "Thur"
>
> # Accessing vector elements using negative indexing
> WeekDays[-2]
[1] "Mon"  "Wed"  "Thur" "Fri"
> WeekDays[c(-2,-4)]
[1] "Mon" "Wed" "Fri"
```

- **Manipulation with Vectors:**

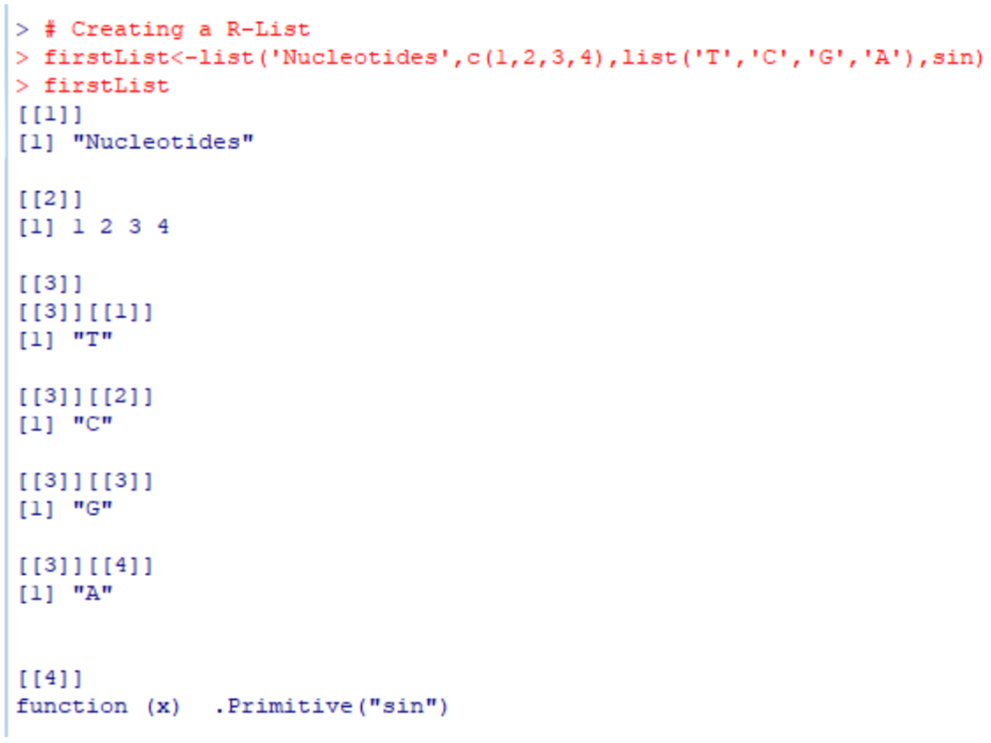


```
> # Manipulation with vectors
> x<-c(2,3,6)
> y<-c(4,5,2)
> x+5
[1]  7  8 11
> y-2
[1] 2 3 0
> 2*x
[1]  4  6 12
> y/4
[1] 1.00 1.25 0.50
> x%%2
[1] 0 1 0
> x%/%2
[1] 1 1 3
> x^2
[1]  4  9 36
> x+y
[1] 6 8 8
> x-y
[1] -2 -2  4
> x*y
[1]  8 15 12
> x/y
[1] 0.5 0.6 3.0
> x^y
[1]  16 243  36
```

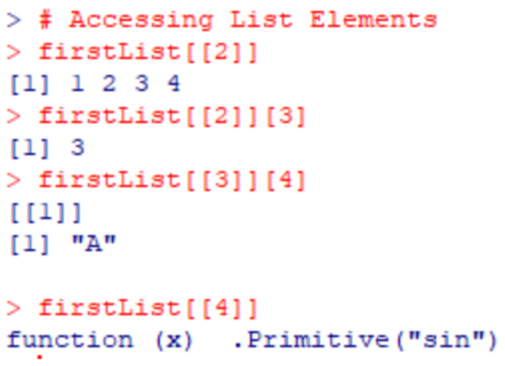


```
> # rep() function
> x<-c(13,17,20,21)
> rep(x,times=3)
 [1] 13 17 20 21 13 17 20 21 13 17 20 21
> rep(x,each=2)
[1] 13 13 17 17 20 20 21 21
> rep(x,each=2,times=3)
 [1] 13 13 17 17 20 20 21 21 13 13 17 17 20 20 21 21 13 13 17 17 20 20 21 21
```

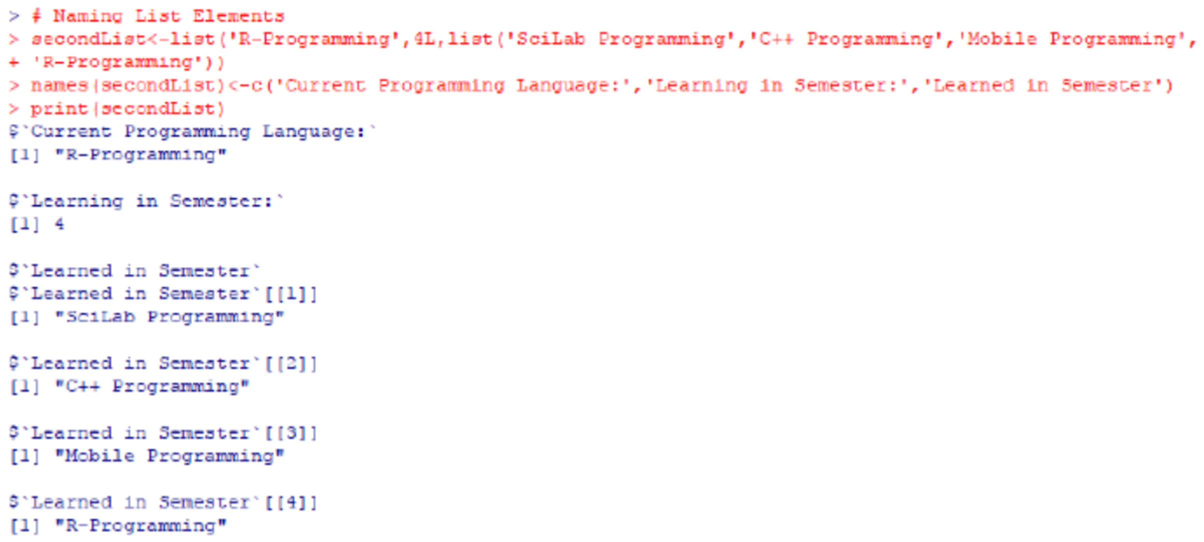
4. R – Lists:



```
> # Creating a R-List
> firstList<-list('Nucleotides',c(1,2,3,4),list('T','C','G','A'),sin)
> firstList
[[1]]
[1] "Nucleotides"

[[2]]
[1] 1 2 3 4

[[3]]
[[3]][[1]]
[1] "T"

[[3]][[2]]
[1] "C"

[[3]][[3]]
[1] "G"

[[3]][[4]]
[1] "A"


[[4]]
function (x)  .Primitive("sin")
```



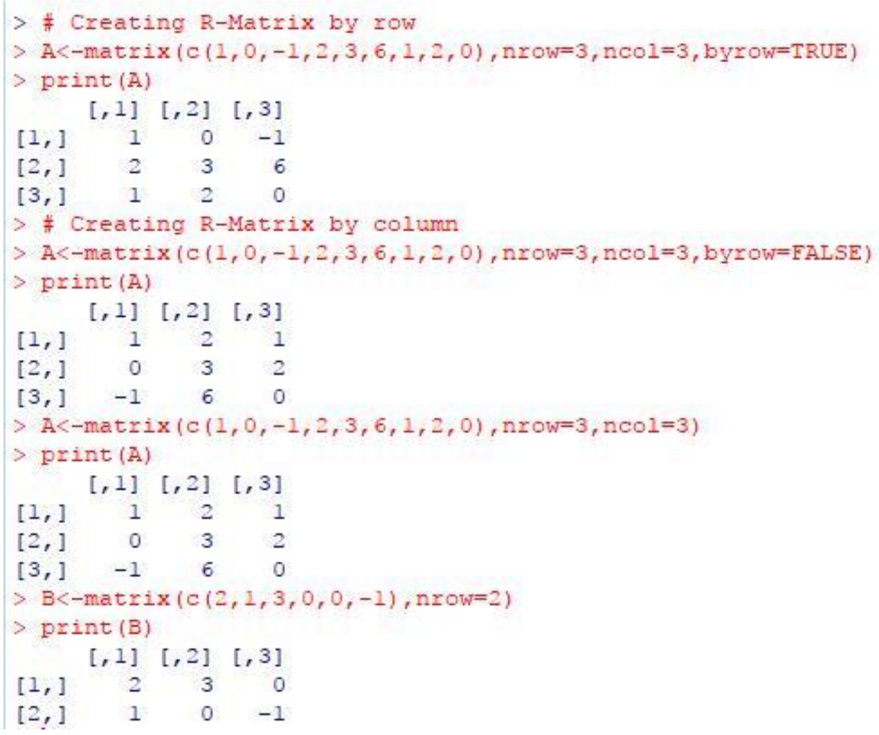
```
> # Accessing List Elements
> firstList[[2]]
[1] 1 2 3 4
> firstList[[2]][3]
[1] 3
> firstList[[3]][4]
[[1]]
[1] "A"

> firstList[[4]]
function (x)  .Primitive("sin")
```



```
> # Naming List Elements
> secondList<-list('R-Programming',4L,list('SciLab Programming','C++ Programming','Mobile Programming',
+ 'R-Programming'))
> names(secondList)<-c('Current Programming Language:','Learning in Semester:','Learned in Semester')
> print(secondList)
$`Current Programming Language:`
[1] "R-Programming"

$`Learning in Semester:`
[1] 4

$`Learned in Semester`
$`Learned in Semester`[[1]]
[1] "SciLab Programming"

$`Learned in Semester`[[2]]
[1] "C++ Programming"

$`Learned in Semester`[[3]]
[1] "Mobile Programming"

$`Learned in Semester`[[4]]
[1] "R-Programming"
```

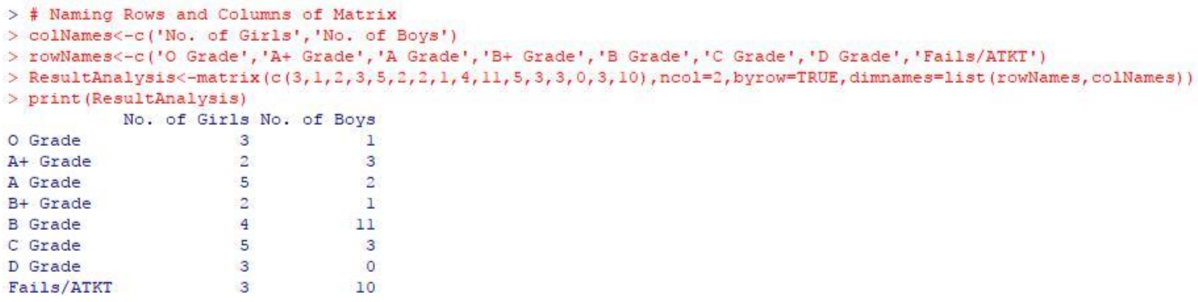
5. **R – Matrices:**

- **Creating R – Matrix:**



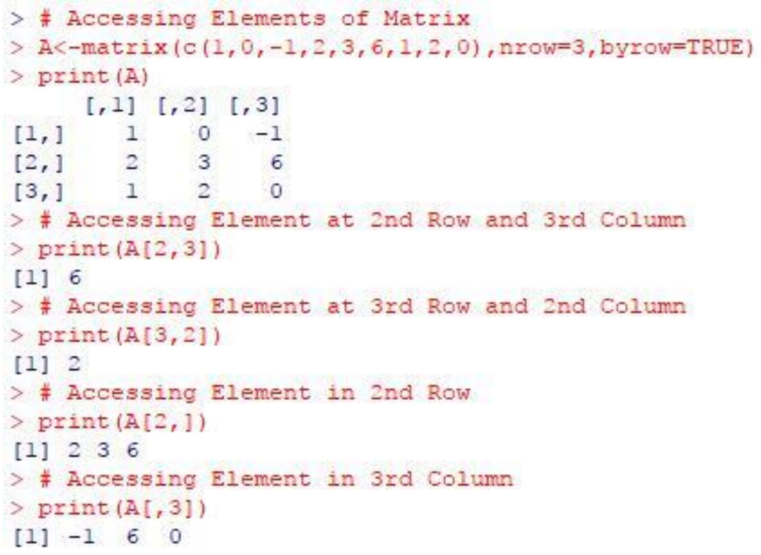
```
> # Creating R-Matrix by row
> A<-matrix(c(1,0,-1,2,3,6,1,2,0),nrow=3,ncol=3,byrow=TRUE)
> print(A)
     [,1] [,2] [,3]
[1,]    1    0   -1
[2,]    2    3    6
[3,]    1    2    0
> # Creating R-Matrix by column
> A<-matrix(c(1,0,-1,2,3,6,1,2,0),nrow=3,ncol=3,byrow=FALSE)
> print(A)
     [,1] [,2] [,3]
[1,]    1    2    1
[2,]    0    3    2
[3,]   -1    6    0
> A<-matrix(c(1,0,-1,2,3,6,1,2,0),nrow=3,ncol=3)
> print(A)
     [,1] [,2] [,3]
[1,]    1    2    1
[2,]    0    3    2
[3,]   -1    6    0
> B<-matrix(c(2,1,3,0,0,-1),nrow=2)
> print(B)
     [,1] [,2] [,3]
[1,]    2    3    0
[2,]    1    0   -1
```

- **Naming Rows and Columns of Matrix:**



```
> # Naming Rows and Columns of Matrix
> colNames<-c('No. of Girls','No. of Boys')
> rowNames<-c('O Grade','A+ Grade','A Grade','B+ Grade','B Grade','C Grade','D Grade','Fails/ATKT')
> ResultAnalysis<-matrix(c(3,1,2,3,5,2,2,1,4,11,5,3,3,0,3,10),ncol=2,byrow=TRUE,dimnames=list(rowNames,colNames))
> print(ResultAnalysis)
            No. of Girls No. of Boys
O Grade                3           1
A+ Grade               2           3
A Grade                5           2
B+ Grade               2           1
B Grade                4          11
C Grade                5           3
D Grade                3           0
Fails/ATKT             3          10
```

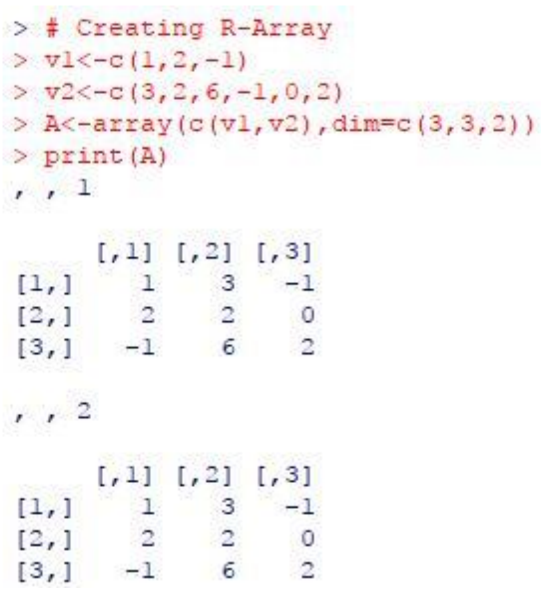
- **Accessing Elements of Matrix:**



```
> # Accessing Elements of Matrix
> A<-matrix(c(1,0,-1,2,3,6,1,2,0),nrow=3,byrow=TRUE)
> print(A)
     [,1] [,2] [,3]
[1,]    1    0   -1
[2,]    2    3    6
[3,]    1    2    0
> # Accessing Element at 2nd Row and 3rd Column
> print(A[2,3])
[1] 6
> # Accessing Element at 3rd Row and 2nd Column
> print(A[3,2])
[1] 2
> # Accessing Element in 2nd Row
> print(A[2,])
[1] 2 3 6
> # Accessing Element in 3rd Column
> print(A[,3])
[1] -1  6  0
```

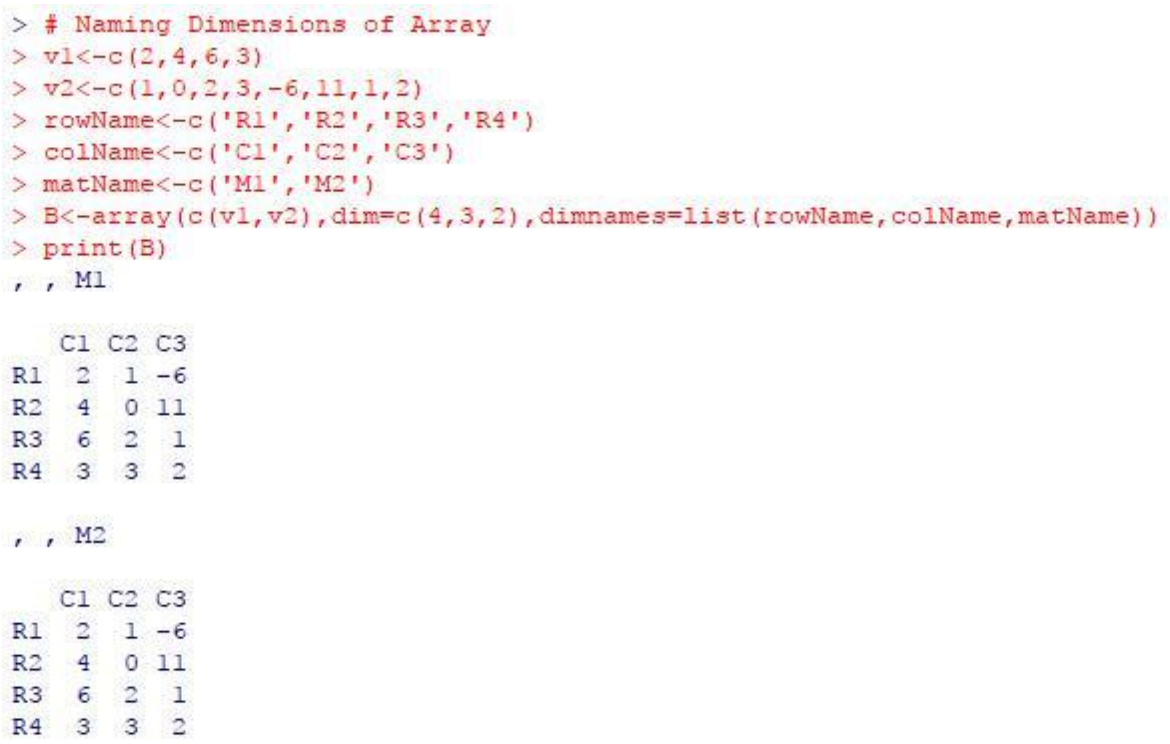
6. R – Arrays:

- **Creating R – Array:**



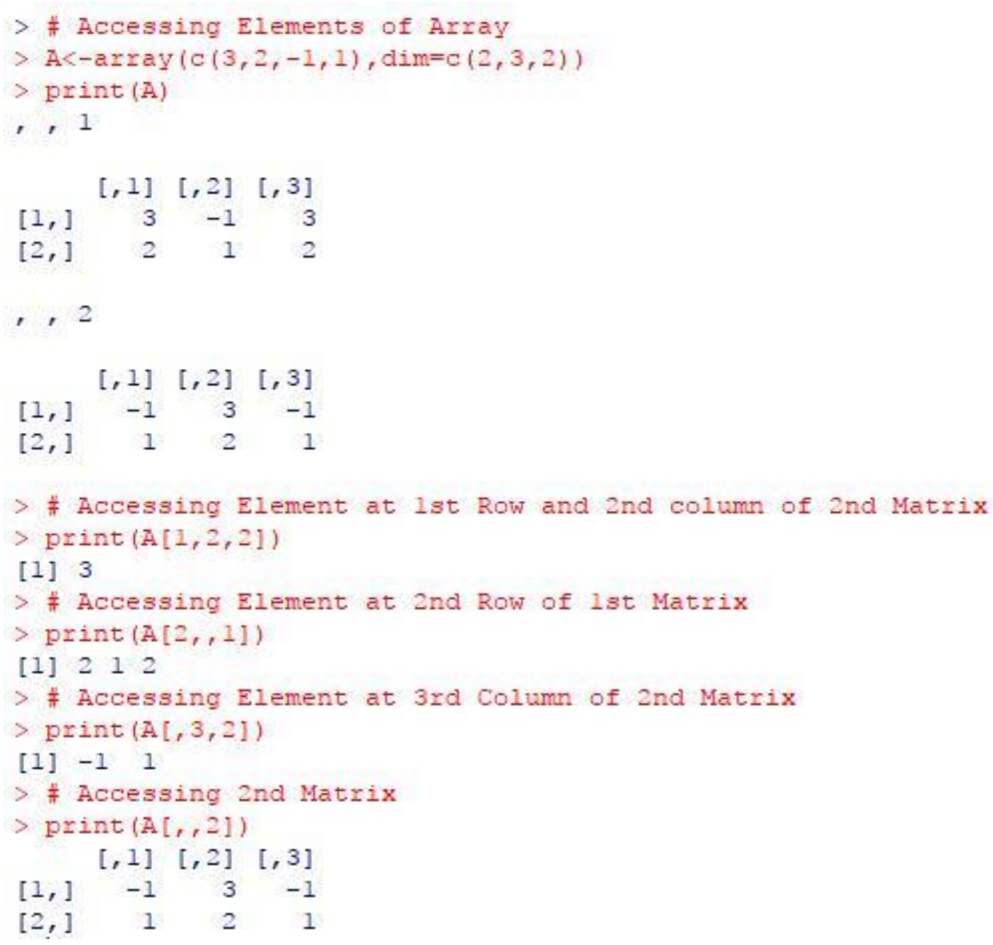
```
> # Creating R-Array
> v1<-c(1,2,-1)
> v2<-c(3,2,6,-1,0,2)
> A<-array(c(v1,v2),dim=c(3,3,2))
> print(A)
, , 1

     [,1] [,2] [,3]
[1,]    1    3   -1
[2,]    2    2    0
[3,]   -1    6    2

, , 2

     [,1] [,2] [,3]
[1,]    1    3   -1
[2,]    2    2    0
[3,]   -1    6    2
```

- **Naming Dimensions of Array**



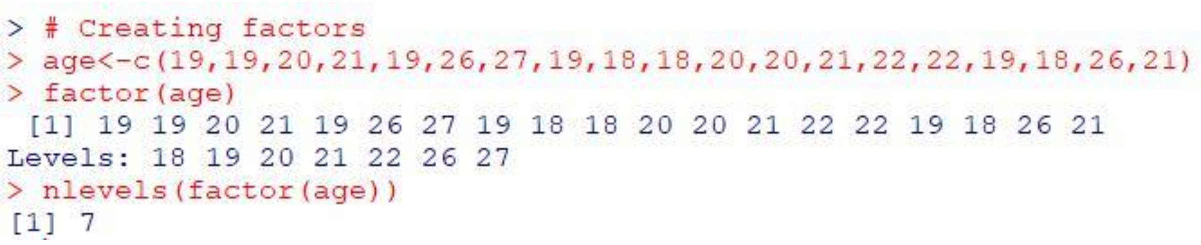
```
> # Naming Dimensions of Array
> v1<-c(2,4,6,3)
> v2<-c(1,0,2,3,-6,11,1,2)
> rowName<-c('R1','R2','R3','R4')
> colName<-c('C1','C2','C3')
> matName<-c('M1','M2')
> B<-array(c(v1,v2),dim=c(4,3,2),dimnames=list(rowName,colName,matName))
> print(B)
, , M1

   C1 C2 C3
R1  2  1 -6
R2  4  0 11
R3  6  2  1
R4  3  3  2

, , M2

   C1 C2 C3
R1  2  1 -6
R2  4  0 11
R3  6  2  1
R4  3  3  2
```

- **Accessing Elements of Array:**



```
> # Accessing Elements of Array
> A<-array(c(3,2,-1,1),dim=c(2,3,2))
> print(A)
, , 1

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

, , 2

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

> # Accessing Element at 1st Row and 2nd column of 2nd Matrix
> print(A[1,2,2])
[1] 3
> # Accessing Element at 2nd Row of 1st Matrix
> print(A[2,,1])
[1] 2 1 2
> # Accessing Element at 3rd Column of 2nd Matrix
> print(A[,3,2])
[1] -1  1
> # Accessing 2nd Matrix
> print(A[,,2])
     [,1] [,2] [,3]
[1,]   -1    3   -1
[2,]    1    2    1
```

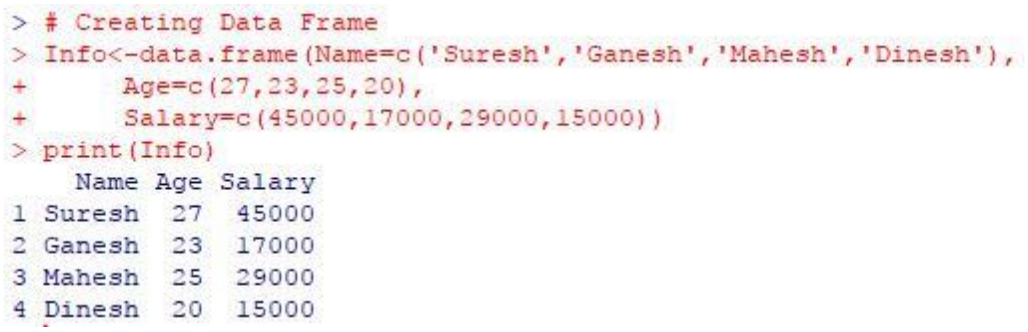
7. R – Factors



```
> # Creating factors
> age<-c(19,19,20,21,19,26,27,19,18,18,20,20,21,22,22,19,18,26,21)
> factor(age)
 [1] 19 19 20 21 19 26 27 19 18 18 20 20 21 22 22 19 18 26 21
Levels: 18 19 20 21 22 26 27
> nlevels(factor(age))
[1] 7
```

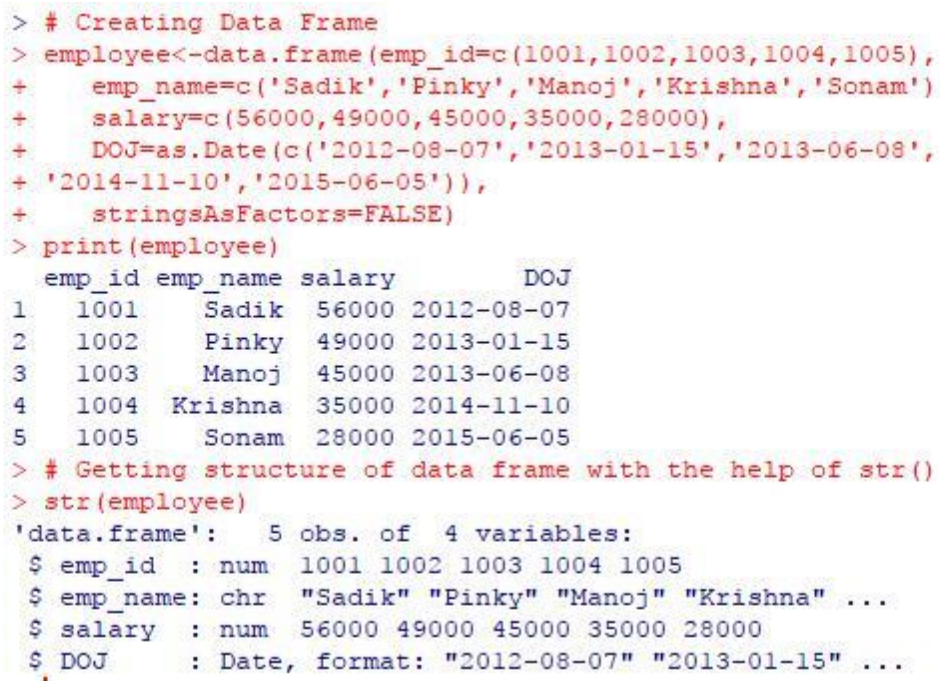
8. R – Data Frames:

- **Creating R – Data Frames:**



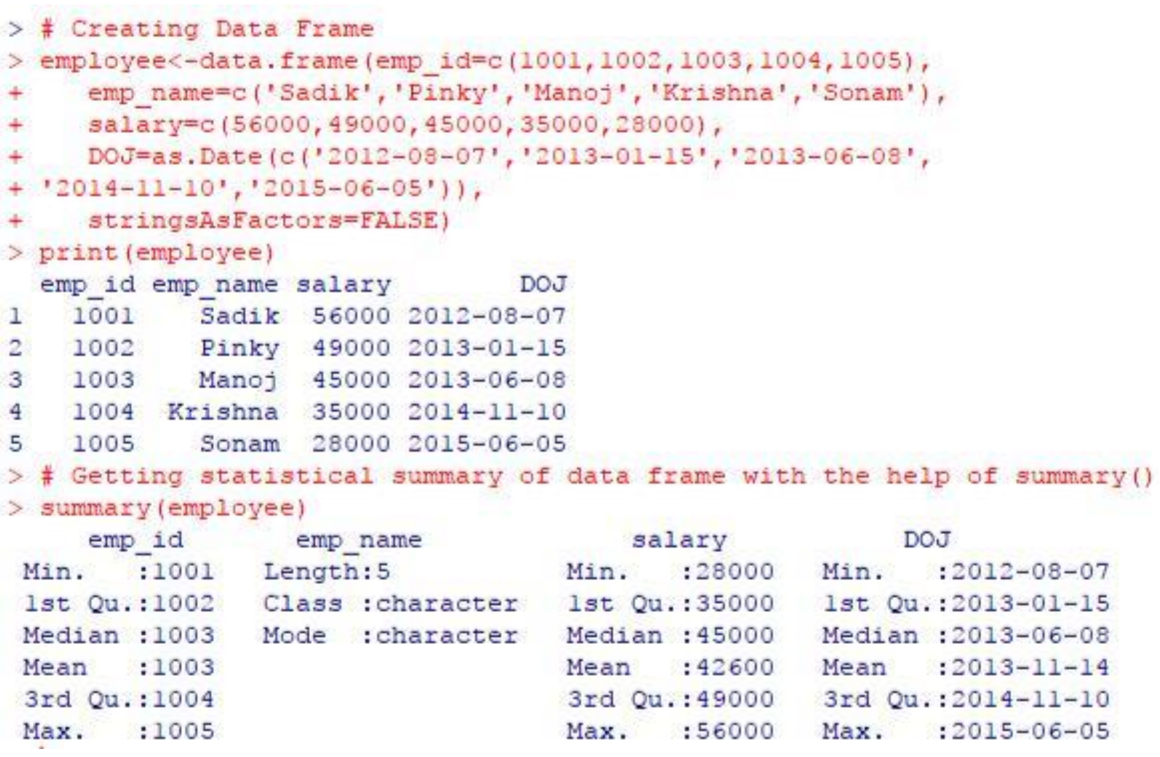
```
> # Creating Data Frame
> Info<-data.frame(Name=c('Suresh','Ganesh','Mahesh','Dinesh'),
+      Age=c(27,23,25,20),
+      Salary=c(45000,17000,29000,15000))
> print(Info)
    Name Age Salary
1 Suresh  27  45000
2 Ganesh  23  17000
3 Mahesh  25  29000
4 Dinesh  20  15000
```

- **Getting Structure of Data Frame**



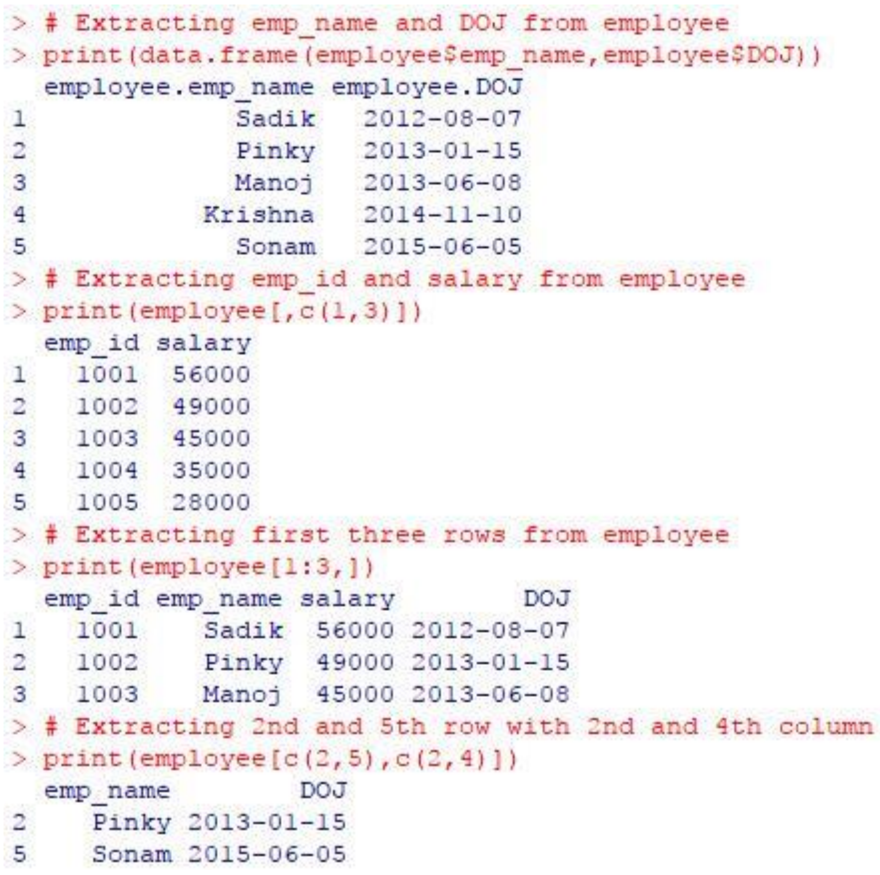
```
> # Creating Data Frame
> employee<-data.frame(emp_id=c(1001,1002,1003,1004,1005),
+    emp_name=c('Sadik','Pinky','Manoj','Krishna','Sonam'),
+    salary=c(56000,49000,45000,35000,28000),
+    DOJ=as.Date(c('2012-08-07','2013-01-15','2013-06-08',
+ '2014-11-10','2015-06-05')),
+    stringsAsFactors=FALSE)
> print(employee)
  emp_id emp_name salary        DOJ
1   1001    Sadik  56000 2012-08-07
2   1002    Pinky  49000 2013-01-15
3   1003    Manoj  45000 2013-06-08
4   1004  Krishna  35000 2014-11-10
5   1005    Sonam  28000 2015-06-05
> # Getting structure of data frame with the help of str()
> str(employee)
'data.frame':   5 obs. of  4 variables:
 $ emp_id  : num  1001 1002 1003 1004 1005
 $ emp_name: chr  "Sadik" "Pinky" "Manoj" "Krishna" ...
 $ salary  : num  56000 49000 45000 35000 28000
 $ DOJ     : Date, format: "2012-08-07" "2013-01-15" ...
```

- **Getting Statistical Summary**



```
> # Creating Data Frame
> employee<-data.frame(emp_id=c(1001,1002,1003,1004,1005),
+    emp_name=c('Sadik','Pinky','Manoj','Krishna','Sonam'),
+    salary=c(56000,49000,45000,35000,28000),
+    DOJ=as.Date(c('2012-08-07','2013-01-15','2013-06-08',
+ '2014-11-10','2015-06-05')),
+    stringsAsFactors=FALSE)
> print(employee)
  emp_id emp_name salary        DOJ
1   1001    Sadik  56000 2012-08-07
2   1002    Pinky  49000 2013-01-15
3   1003    Manoj  45000 2013-06-08
4   1004  Krishna  35000 2014-11-10
5   1005    Sonam  28000 2015-06-05
> # Getting statistical summary of data frame with the help of summary()
> summary(employee)
     emp_id       emp_name              salary            DOJ
 Min.   :1001   Length:5           Min.   :28000   Min.   :2012-08-07
 1st Qu.:1002   Class :character   1st Qu.:35000   1st Qu.:2013-01-15
 Median :1003   Mode  :character   Median :45000   Median :2013-06-08
 Mean   :1003                      Mean   :42600   Mean   :2013-11-14
 3rd Qu.:1004                      3rd Qu.:49000   3rd Qu.:2014-11-10
 Max.   :1005                      Max.   :56000   Max.   :2015-06-05
```

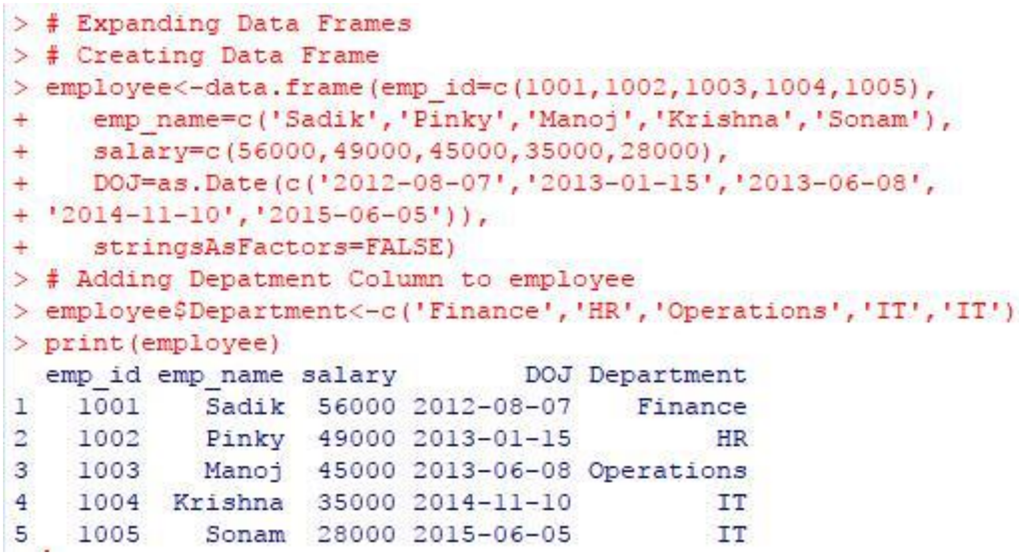
- **Extracting Data from Data Frame:**



```
> # Extracting emp_name and DOJ from employee
> print(data.frame(employee$emp_name,employee$DOJ))
  employee.emp_name employee.DOJ
1             Sadik   2012-08-07
2             Pinky   2013-01-15
3             Manoj   2013-06-08
4           Krishna   2014-11-10
5             Sonam   2015-06-05
> # Extracting emp_id and salary from employee
> print(employee[,c(1,3)])
  emp_id salary
1   1001  56000
2   1002  49000
3   1003  45000
4   1004  35000
5   1005  28000
> # Extracting first three rows from employee
> print(employee[1:3,])
  emp_id emp_name salary        DOJ
1   1001    Sadik  56000 2012-08-07
2   1002    Pinky  49000 2013-01-15
3   1003    Manoj  45000 2013-06-08
> # Extracting 2nd and 5th row with 2nd and 4th column
> print(employee[c(2,5),c(2,4)])
  emp_name        DOJ
2    Pinky 2013-01-15
5    Sonam 2015-06-05
```

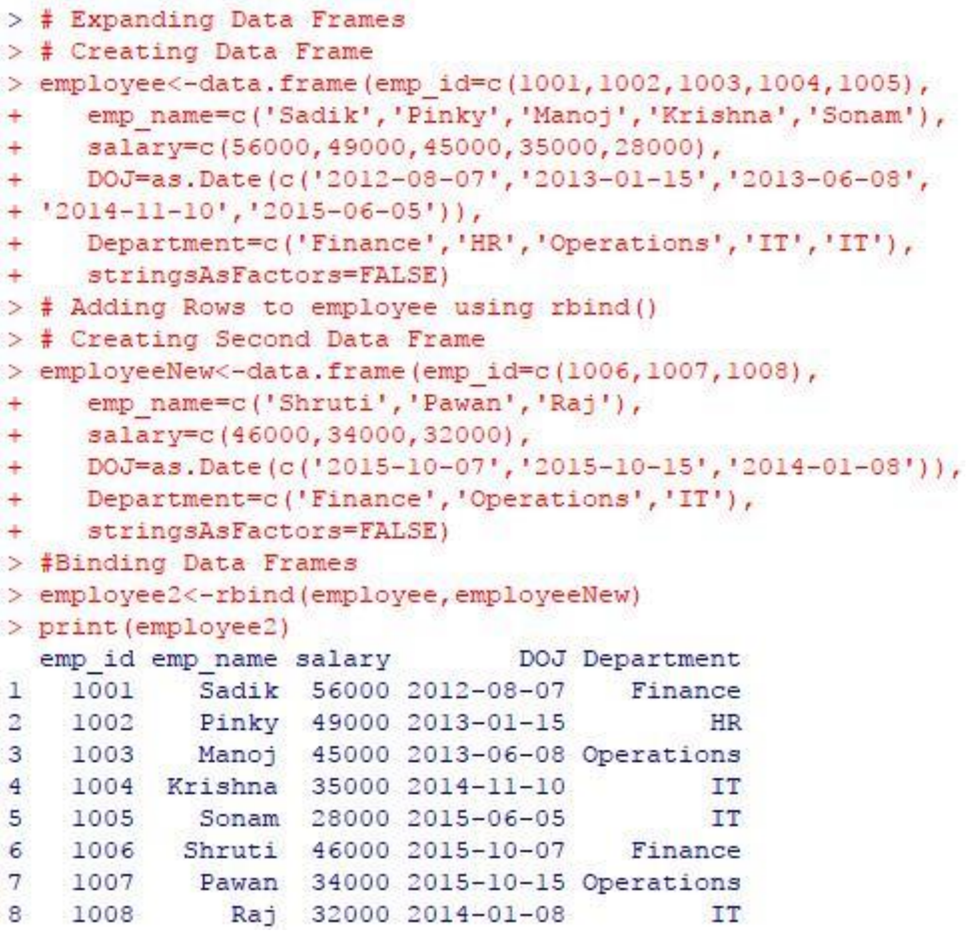
- **Expanding Data Frame**

  i. **Adding Column:**



```
> # Expanding Data Frames
> # Creating Data Frame
> employee<-data.frame(emp_id=c(1001,1002,1003,1004,1005),
+    emp_name=c('Sadik','Pinky','Manoj','Krishna','Sonam'),
+    salary=c(56000,49000,45000,35000,28000),
+    DOJ=as.Date(c('2012-08-07','2013-01-15','2013-06-08',
+ '2014-11-10','2015-06-05')),
+    stringsAsFactors=FALSE)
> # Adding Depatment Column to employee
> employee$Department<-c('Finance','HR','Operations','IT','IT')
> print(employee)
  emp_id emp_name salary        DOJ Department
1   1001    Sadik  56000 2012-08-07    Finance
2   1002    Pinky  49000 2013-01-15         HR
3   1003    Manoj  45000 2013-06-08 Operations
4   1004  Krishna  35000 2014-11-10         IT
5   1005    Sonam  28000 2015-06-05         IT
```

  ii. **Adding Rows:**



```
> # Expanding Data Frames
> # Creating Data Frame
> employee<-data.frame(emp_id=c(1001,1002,1003,1004,1005),
+    emp_name=c('Sadik','Pinky','Manoj','Krishna','Sonam'),
+    salary=c(56000,49000,45000,35000,28000),
+    DOJ=as.Date(c('2012-08-07','2013-01-15','2013-06-08',
+ '2014-11-10','2015-06-05')),
+    Department=c('Finance','HR','Operations','IT','IT'),
+    stringsAsFactors=FALSE)
> # Adding Rows to employee using rbind()
> # Creating Second Data Frame
> employeeNew<-data.frame(emp_id=c(1006,1007,1008),
+    emp_name=c('Shruti','Pawan','Raj'),
+    salary=c(46000,34000,32000),
+    DOJ=as.Date(c('2015-10-07','2015-10-15','2014-01-08')),
+    Department=c('Finance','Operations','IT'),
+    stringsAsFactors=FALSE)
> #Binding Data Frames
> employee2<-rbind(employee,employeeNew)
> print(employee2)
  emp_id emp_name salary        DOJ Department
1   1001    Sadik  56000 2012-08-07    Finance
2   1002    Pinky  49000 2013-01-15         HR
3   1003    Manoj  45000 2013-06-08 Operations
4   1004  Krishna  35000 2014-11-10         IT
5   1005    Sonam  28000 2015-06-05         IT
6   1006   Shruti  46000 2015-10-07    Finance
7   1007    Pawan  34000 2015-10-15 Operations
8   1008      Raj  32000 2014-01-08         IT
```

```
> # Creating Data Frame
> Name<-c('Manish','Danish','David','Sifa')
> Age<-c(32,21,28,25)
> Salary<-c(41000,20000,32000,28000)
> Info<-data.frame(Name,Age,Salary)
> Info
    Name Age Salary
1 Manish  32  41000
2 Danish  21  20000
3  David  28  32000
4   Sifa  25  28000
> Info$Name
[1] Manish Danish David  Sifa
Levels: Danish David Manish Sifa
> class(Info$Name)
[1] "factor"
> Info$Name<-as.character(Name)
> class(Info$Name)
[1] "character"
> Info$Salary
[1] 41000 20000 32000 28000
> Info$Name[3]
[1] "David"
> Info$Salary[3]
[1] 32000
```

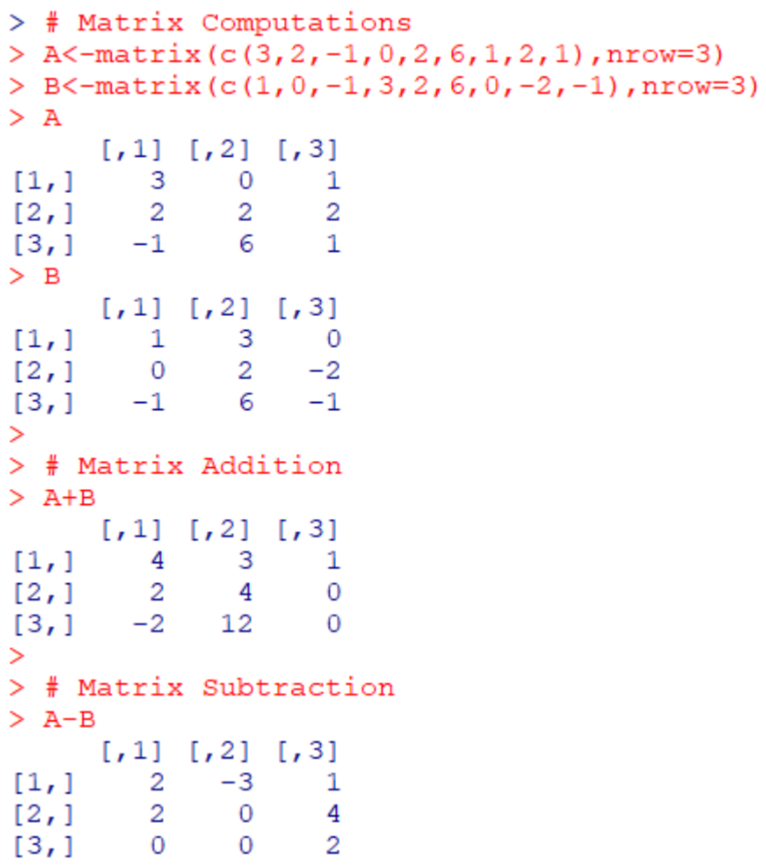
```
> # Adding a column to existing data frame
> Emp_ID<-c(1001,1002,1003,1004)
> InfoNew<-cbind(Emp_ID,Info)
> InfoNew
  Emp_ID   Name Age Salary
1   1001 Manish  32  41000
2   1002 Danish  21  20000
3   1003  David  28  32000
4   1004   Sifa  25  28000
> # Adding a row to existing data frame
> NewRow<-c(1005,'Ashok',20,19000)
> InfoNew2<-rbind(InfoNew,NewRow)
> InfoNew2
  Emp_ID   Name Age Salary
1   1001 Manish  32  41000
2   1002 Danish  21  20000
3   1003  David  28  32000
4   1004   Sifa  25  28000
5   1005  Ashok  20  19000
```

PRACTICAL No. 2

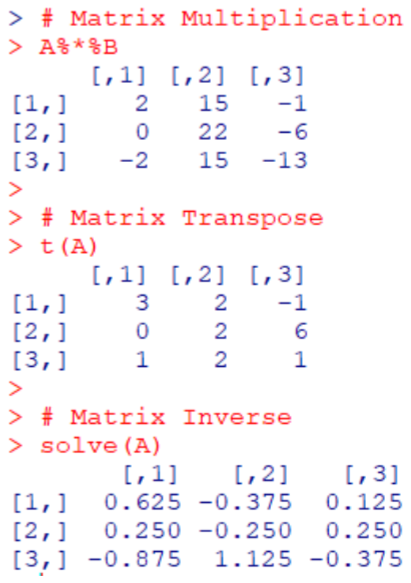
MATRIX COMPUTATIONS

AIM: Create a Matrix using R and Perform the operations addition, subtraction, multiplication, transpose, inverse etc.

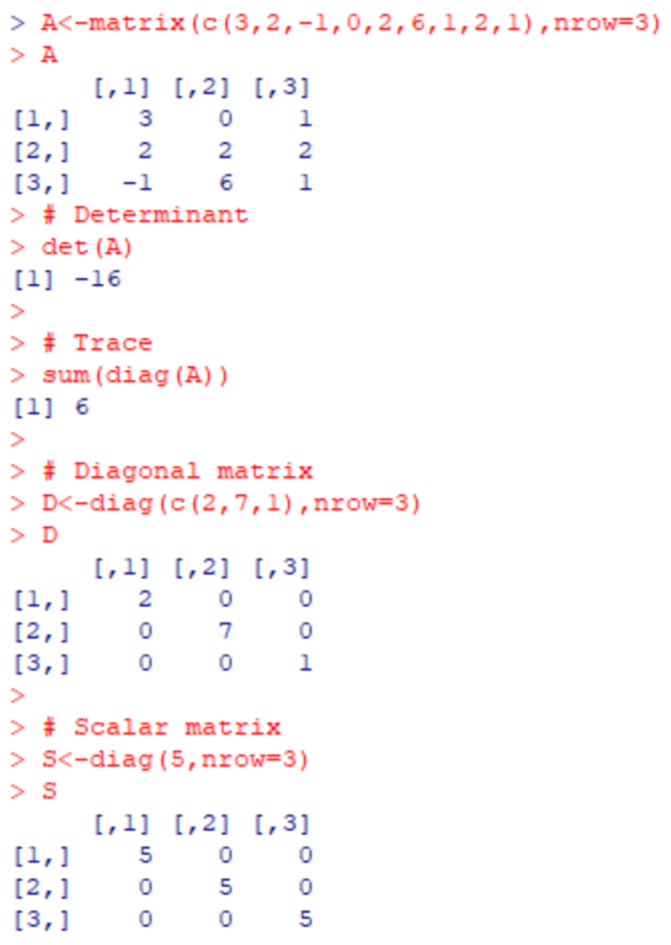
SOURCE CODE & OUTPUT:



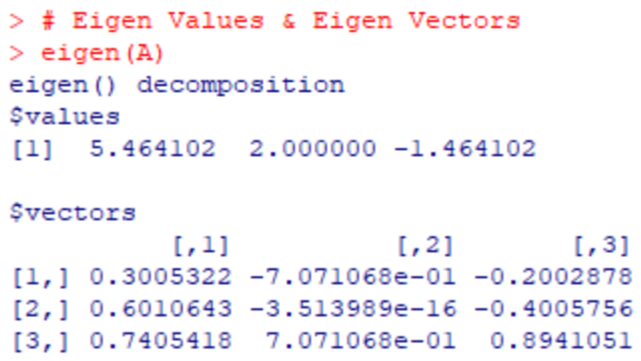
```
> # Matrix Computations
> A<-matrix(c(3,2,-1,0,2,6,1,2,1),nrow=3)
> B<-matrix(c(1,0,-1,3,2,6,0,-2,-1),nrow=3)
> A
     [,1] [,2] [,3]
[1,]    3    0    1
[2,]    2    2    2
[3,]   -1    6    1
> B
     [,1] [,2] [,3]
[1,]    1    3    0
[2,]    0    2   -2
[3,]   -1    6   -1
>
> # Matrix Addition
> A+B
     [,1] [,2] [,3]
[1,]    4    3    1
[2,]    2    4    0
[3,]   -2   12    0
>
> # Matrix Subtraction
> A-B
     [,1] [,2] [,3]
[1,]    2   -3    1
[2,]    2    0    4
[3,]    0    0    2
```



```
> # Matrix Multiplication
> A%*%B
     [,1] [,2] [,3]
[1,]    2   15   -1
[2,]    0   22   -6
[3,]   -2   15  -13
>
> # Matrix Transpose
> t(A)
     [,1] [,2] [,3]
[1,]    3    2   -1
[2,]    0    2    6
[3,]    1    2    1
>
> # Matrix Inverse
> solve(A)
       [,1]   [,2]   [,3]
[1,]  0.625 -0.375  0.125
[2,]  0.250 -0.250  0.250
[3,] -0.875  1.125 -0.375
```



```
> A<-matrix(c(3,2,-1,0,2,6,1,2,1),nrow=3)
> A
     [,1] [,2] [,3]
[1,]    3    0    1
[2,]    2    2    2
[3,]   -1    6    1
> # Determinant
> det(A)
[1] -16
>
> # Trace
> sum(diag(A))
[1] 6
>
> # Diagonal matrix
> D<-diag(c(2,7,1),nrow=3)
> D
     [,1] [,2] [,3]
[1,]    2    0    0
[2,]    0    7    0
[3,]    0    0    1
>
> # Scalar matrix
> S<-diag(5,nrow=3)
> S
     [,1] [,2] [,3]
[1,]    5    0    0
[2,]    0    5    0
[3,]    0    0    5
```



```
> # Eigen Values & Eigen Vectors
> eigen(A)
eigen() decomposition
$values
[1]  5.464102  2.000000 -1.464102

$vectors
          [,1]          [,2]       [,3]
[1,] 0.3005322 -7.071068e-01 -0.2002878
[2,] 0.6010643 -3.513989e-16 -0.4005756
[3,] 0.7405418  7.071068e-01  0.8941051
```

PRACTICAL No. 3

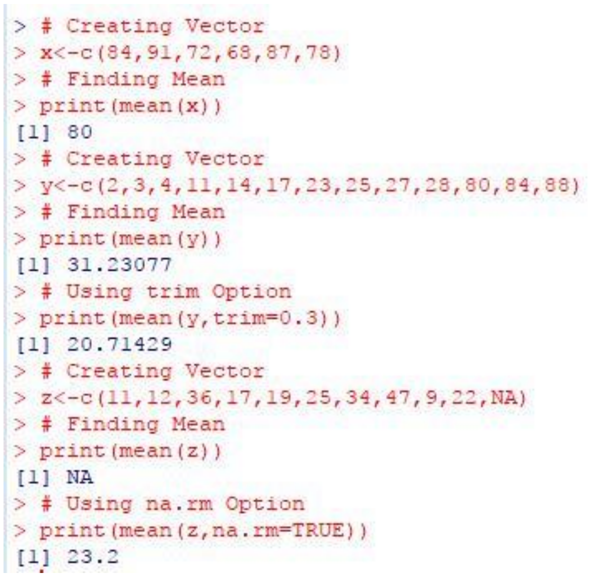
STATISTICAL FUNCTIONS

Mean, Median, Mode, Quartiles, Range, Inter-Quartile Range & Histogram

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

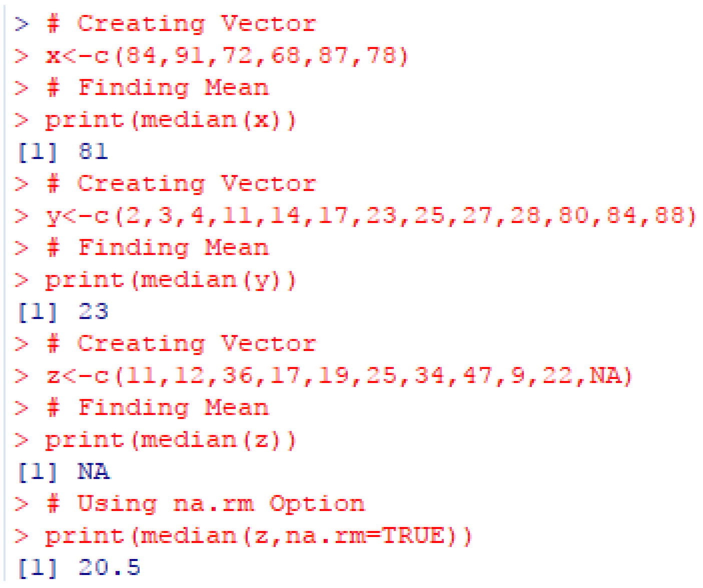
SOURCE CODE & OUTPUT:

1. Mean:



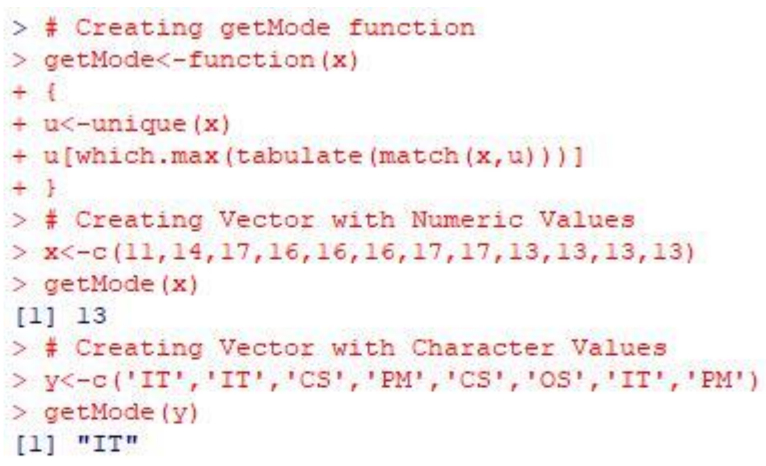
```
> # Creating Vector
> x<-c(84,91,72,68,87,78)
> # Finding Mean
> print(mean(x))
[1] 80
> # Creating Vector
> y<-c(2,3,4,11,14,17,23,25,27,28,80,84,88)
> # Finding Mean
> print(mean(y))
[1] 31.23077
> # Using trim Option
> print(mean(y,trim=0.3))
[1] 20.71429
> # Creating Vector
> z<-c(11,12,36,17,19,25,34,47,9,22,NA)
> # Finding Mean
> print(mean(z))
[1] NA
> # Using na.rm Option
> print(mean(z,na.rm=TRUE))
[1] 23.2
```

2. Median:



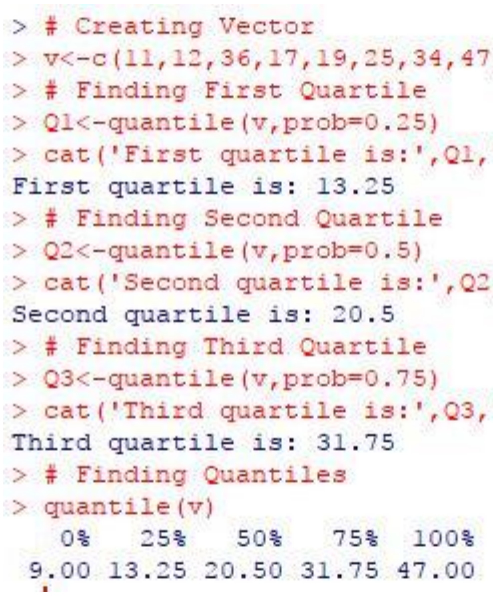
```
> # Creating Vector
> x<-c(84,91,72,68,87,78)
> # Finding Mean
> print(median(x))
[1] 81
> # Creating Vector
> y<-c(2,3,4,11,14,17,23,25,27,28,80,84,88)
> # Finding Mean
> print(median(y))
[1] 23
> # Creating Vector
> z<-c(11,12,36,17,19,25,34,47,9,22,NA)
> # Finding Mean
> print(median(z))
[1] NA
> # Using na.rm Option
> print(median(z,na.rm=TRUE))
[1] 20.5
```

3. Mode:



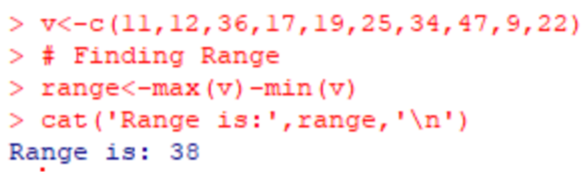
```
> # Creating getMode function
> getMode<-function(x)
+ {
+ u<-unique(x)
+ u[which.max(tabulate(match(x,u)))]
+ }
> # Creating Vector with Numeric Values
> x<-c(11,14,17,16,16,16,17,17,13,13,13,13)
> getMode(x)
[1] 13
> # Creating Vector with Character Values
> y<-c('IT','IT','CS','PM','CS','OS','IT','PM')
> getMode(y)
[1] "IT"
```

4. Quartiles:



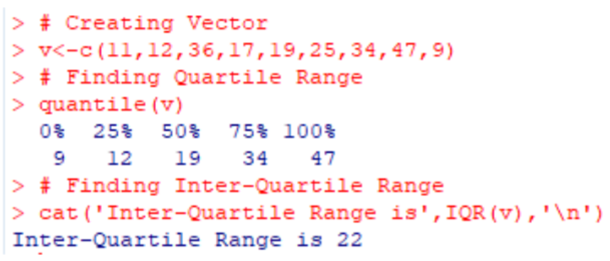
```
> # Creating Vector
> v<-c(11,12,36,17,19,25,34,47,9,22)
> # Finding First Quartile
> Q1<-quantile(v,prob=0.25)
> cat('First quartile is:',Q1,'\n')
First quartile is: 13.25
> # Finding Second Quartile
> Q2<-quantile(v,prob=0.5)
> cat('Second quartile is:',Q2,'\n')
Second quartile is: 20.5
> # Finding Third Quartile
> Q3<-quantile(v,prob=0.75)
> cat('Third quartile is:',Q3,'\n')
Third quartile is: 31.75
> # Finding Quantiles
> quantile(v)
   0%   25%   50%   75%  100% 
 9.00 13.25 20.50 31.75 47.00 
```

5. Range:



```
> v<-c(11,12,36,17,19,25,34,47,9,22)
> # Finding Range
> range<-max(v)-min(v)
> cat('Range is:',range,'\n')
Range is: 38
```

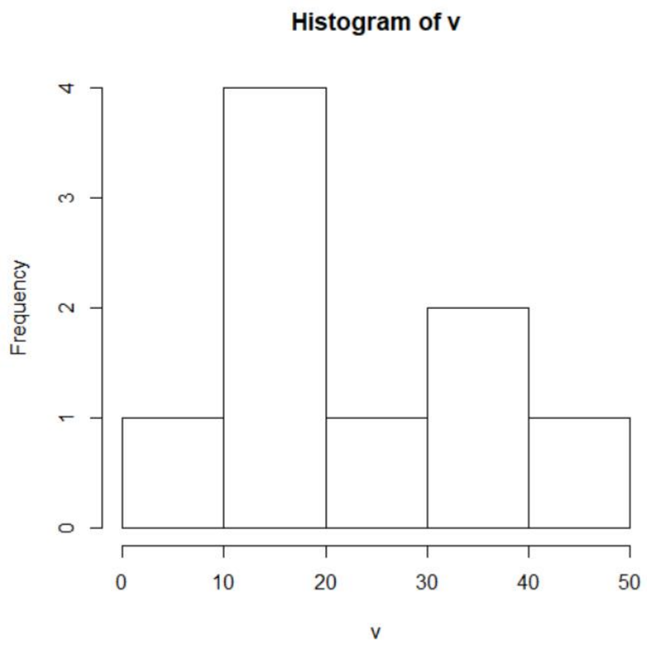
6. Inter-Quartile Range:



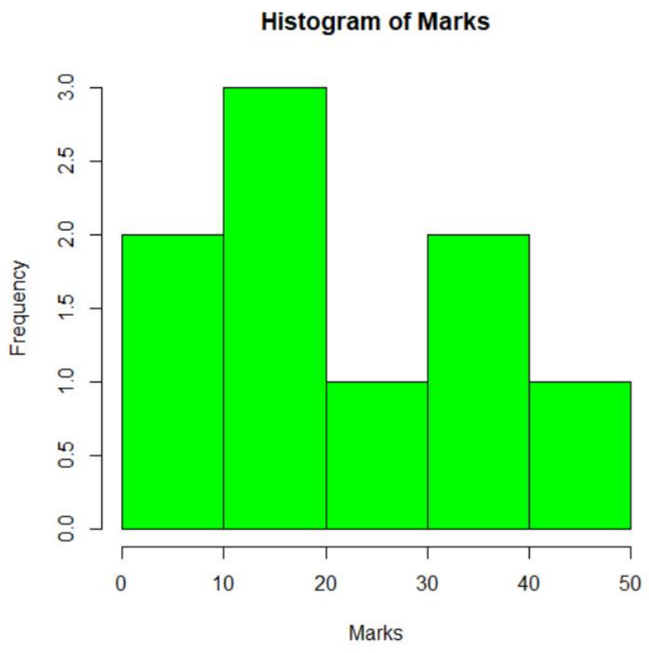
```
> # Creating Vector
> v<-c(11,12,36,17,19,25,34,47,9)
> # Finding Quartile Range
> quantile(v)
  0%  25%  50%  75% 100% 
   9   12   19   34   47 
> # Finding Inter-Quartile Range
> cat('Inter-Quartile Range is',IQR(v),'\n')
Inter-Quartile Range is 22
```

7. Histogram:

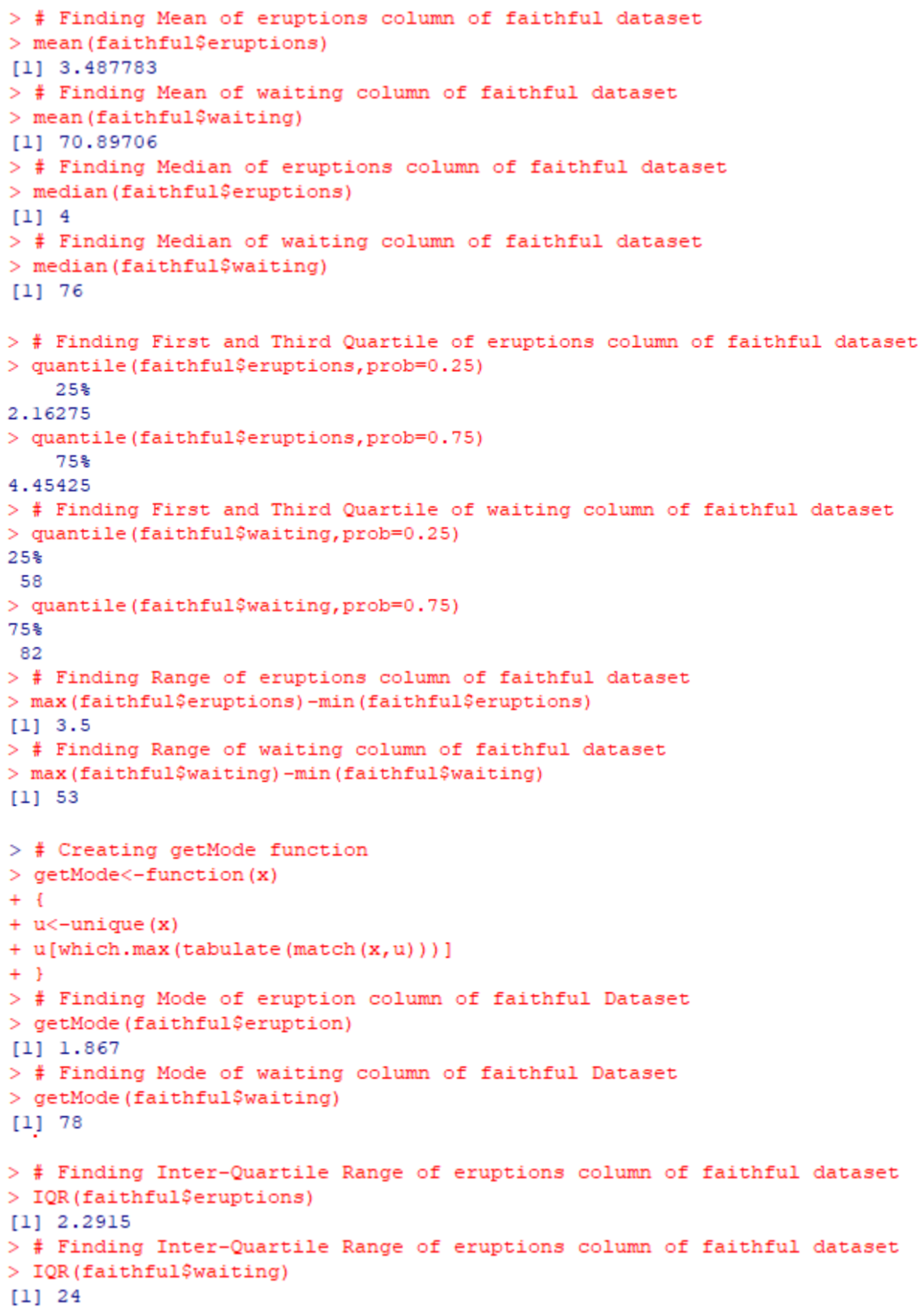
```
> # Creating Vector
> v<-c(11,12,36,17,19,25,34,47,9)
> # Creating Histogram
> hist(v)
```



```
> # Creating Vector
> v<-c(8,12,36,17,19,25,34,47,9)
> # Creating Histogram with Various Available Options
> hist(v,main='Histogram of Marks',xlab='Marks',col='green')
```

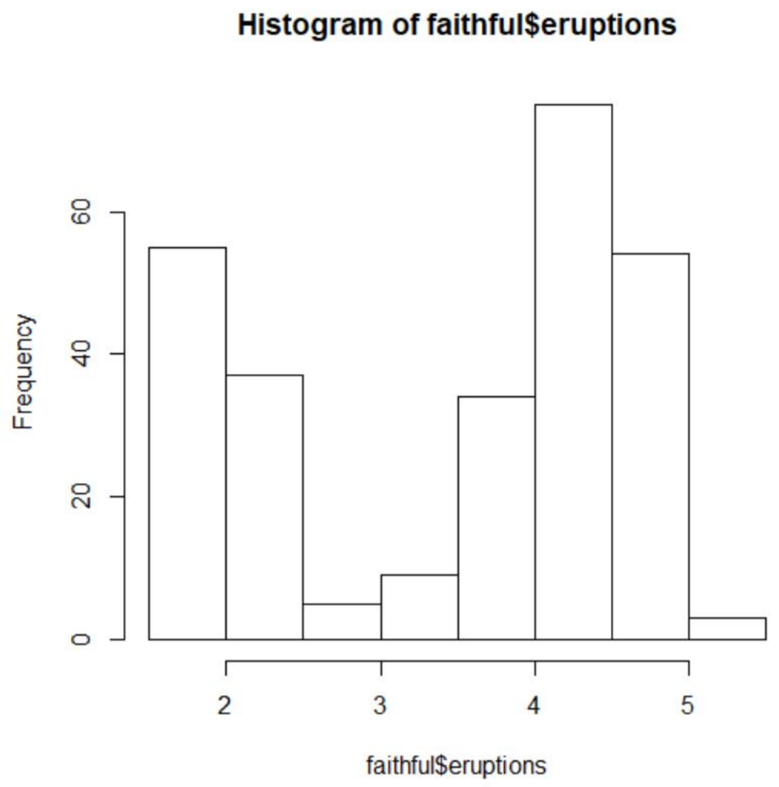


8. Performing Above Statistical Functions on ‘faithful’ Dataset:

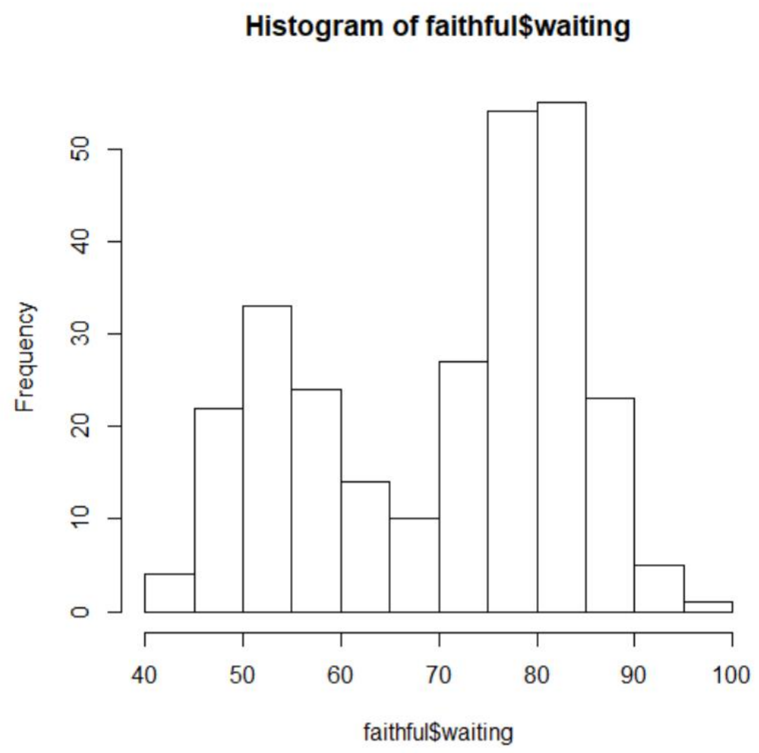


```
> # Finding Mean of eruptions column of faithful dataset
> mean(faithful$eruptions)
[1] 3.487783
> # Finding Mean of waiting column of faithful dataset
> mean(faithful$waiting)
[1] 70.89706
> # Finding Median of eruptions column of faithful dataset
> median(faithful$eruptions)
[1] 4
> # Finding Median of waiting column of faithful dataset
> median(faithful$waiting)
[1] 76

> # Finding First and Third Quartile of eruptions column of faithful dataset
> quantile(faithful$eruptions,prob=0.25)
    25%
2.16275
> quantile(faithful$eruptions,prob=0.75)
    75%
4.45425
> # Finding First and Third Quartile of waiting column of faithful dataset
> quantile(faithful$waiting,prob=0.25)
25%
 58
> quantile(faithful$waiting,prob=0.75)
75%
 82
> # Finding Range of eruptions column of faithful dataset
> max(faithful$eruptions)-min(faithful$eruptions)
[1] 3.5
> # Finding Range of waiting column of faithful dataset
> max(faithful$waiting)-min(faithful$waiting)
[1] 53

> # Creating getMode function
> getMode<-function(x)
+ {
+ u<-unique(x)
+ u[which.max(tabulate(match(x,u)))]
+ }
> # Finding Mode of eruption column of faithful Dataset
> getMode(faithful$eruption)
[1] 1.867
> # Finding Mode of waiting column of faithful Dataset
> getMode(faithful$waiting)
[1] 78

> # Finding Inter-Quartile Range of eruptions column of faithful dataset
> IQR(faithful$eruptions)
[1] 2.2915
> # Finding Inter-Quartile Range of eruptions column of faithful dataset
> IQR(faithful$waiting)
[1] 24
```

```
> hist(faithful$eruptions)
```



```
> hist(faithful$waiting)
```



PRACTICAL No. 4

FINDING MEAN, MEDIAN, MODE, QUARTILES, RANGE, INTER-QUARTILE RANGE, & HISTOGRAM

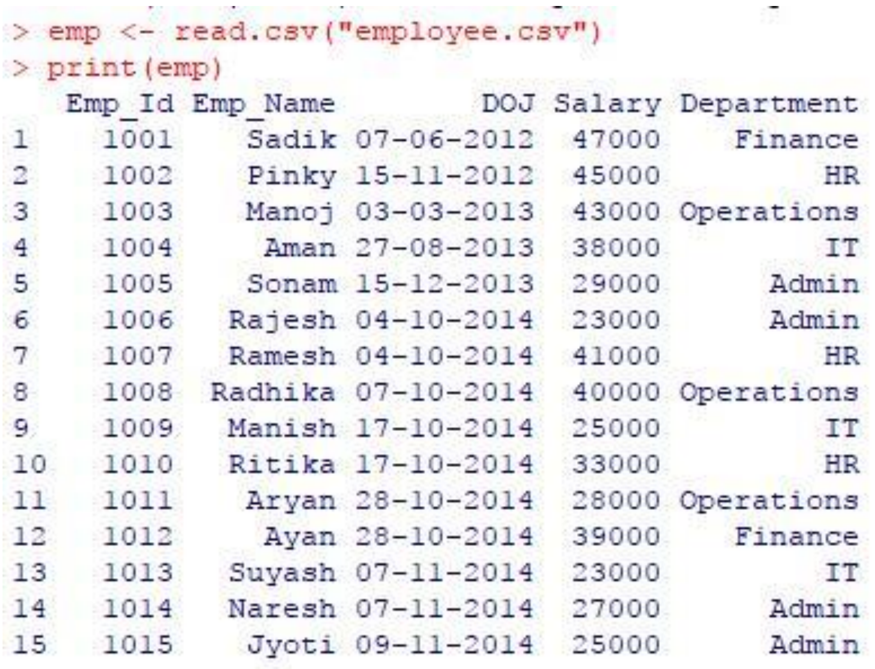
OF EXCEL/.CSV DATA

AIM: Using R import the data from Excel/.CSV file and find mean, median, mode, quartiles, range, inter quartile range, histogram.

SOURCE CODE & OUTPUT:

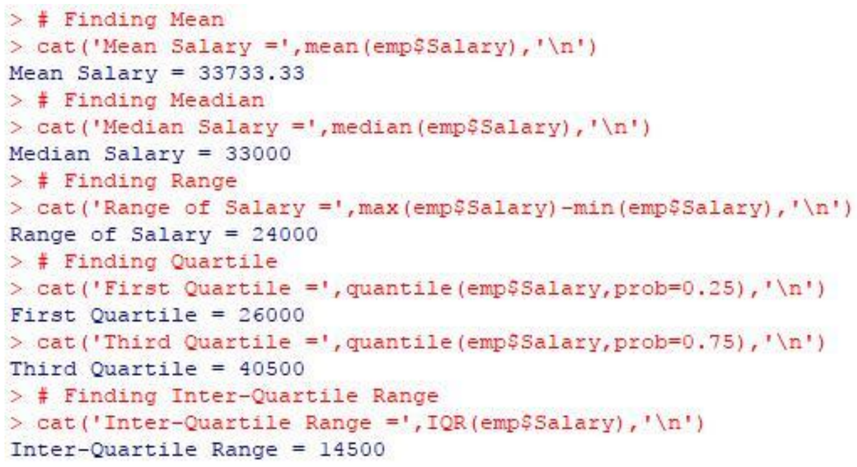
- **Working with CSV File:**

1. **Copy and paste .csv file in working directory.**
2. **Importing Data from .CSV File:**



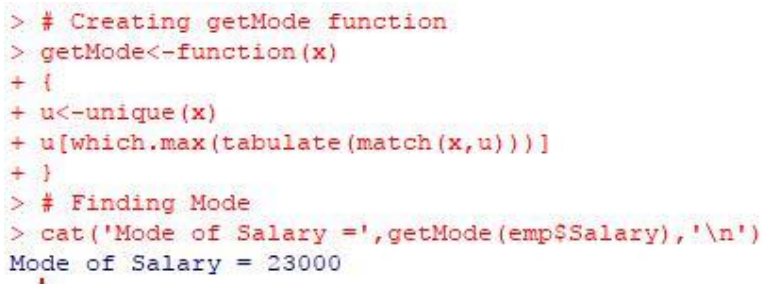
```
> emp <- read.csv("employee.csv")
> print(emp)
   Emp_Id Emp_Name        DOJ Salary Department
1    1001    Sadik 07-06-2012  47000    Finance
2    1002    Pinky 15-11-2012  45000         HR
3    1003    Manoj 03-03-2013  43000 Operations
4    1004     Aman 27-08-2013  38000         IT
5    1005    Sonam 15-12-2013  29000      Admin
6    1006   Rajesh 04-10-2014  23000      Admin
7    1007   Ramesh 04-10-2014  41000         HR
8    1008  Radhika 07-10-2014  40000 Operations
9    1009   Manish 17-10-2014  25000         IT
10   1010   Ritika 17-10-2014  33000         HR
11   1011    Aryan 28-10-2014  28000 Operations
12   1012     Ayan 28-10-2014  39000    Finance
13   1013   Suyash 07-11-2014  23000         IT
14   1014   Naresh 07-11-2014  27000      Admin
15   1015    Jyoti 09-11-2014  25000      Admin
```

3. Finding Mean, Median, Range, Quartiles, Inter-Quartile Range:



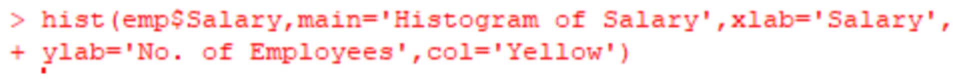
```
> # Finding Mean
> cat('Mean Salary =',mean(emp$Salary),'\n')
Mean Salary = 33733.33
> # Finding Meadian
> cat('Median Salary =',median(emp$Salary),'\n')
Median Salary = 33000
> # Finding Range
> cat('Range of Salary =',max(emp$Salary)-min(emp$Salary),'\n')
Range of Salary = 24000
> # Finding Quartile
> cat('First Quartile =',quantile(emp$Salary,prob=0.25),'\n')
First Quartile = 26000
> cat('Third Quartile =',quantile(emp$Salary,prob=0.75),'\n')
Third Quartile = 40500
> # Finding Inter-Quartile Range
> cat('Inter-Quartile Range =',IQR(emp$Salary),'\n')
Inter-Quartile Range = 14500
```

4. Finding Mode:

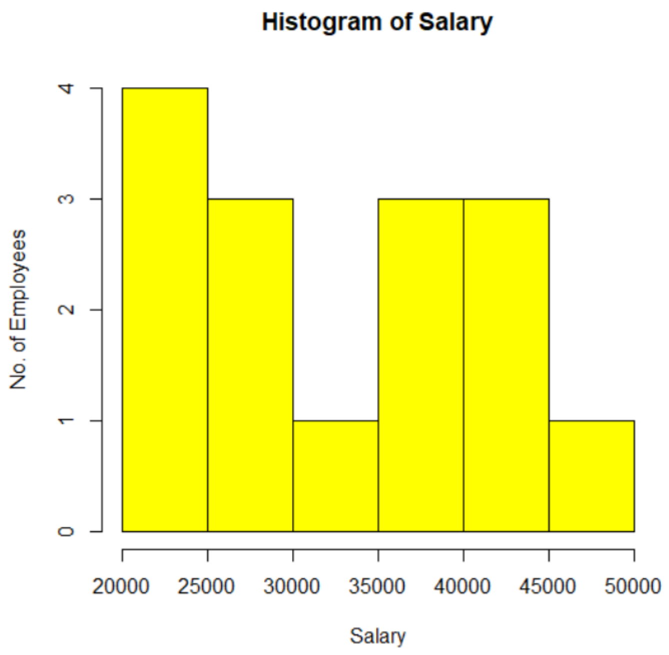


```
> # Creating getMode function
> getMode<-function(x)
+ {
+ u<-unique(x)
+ u[which.max(tabulate(match(x,u)))]
+ }
> # Finding Mode
> cat('Mode of Salary =',getMode(emp$Salary),'\n')
Mode of Salary = 23000
```

5. Histogram:



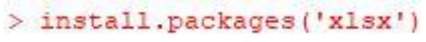
```
> hist(emp$Salary,main='Histogram of Salary',xlab='Salary',
+ ylab='No. of Employees',col='Yellow')
```



- **Working with Excel File:**

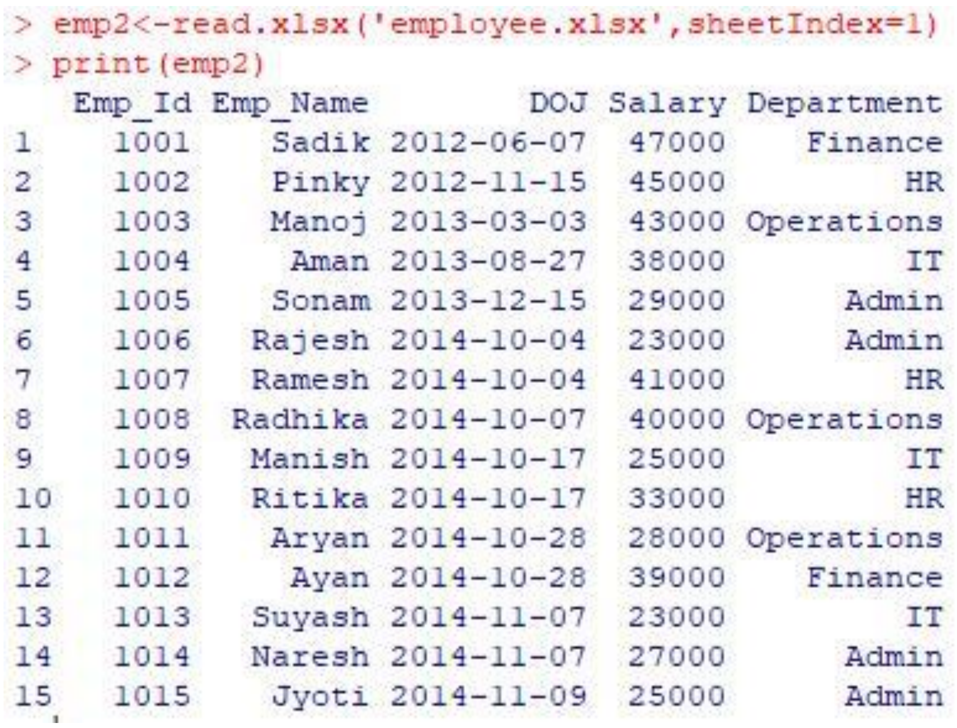
1. **Copy and paste .xlsx file in working directory.**

2. **Installing xlsx Package:**



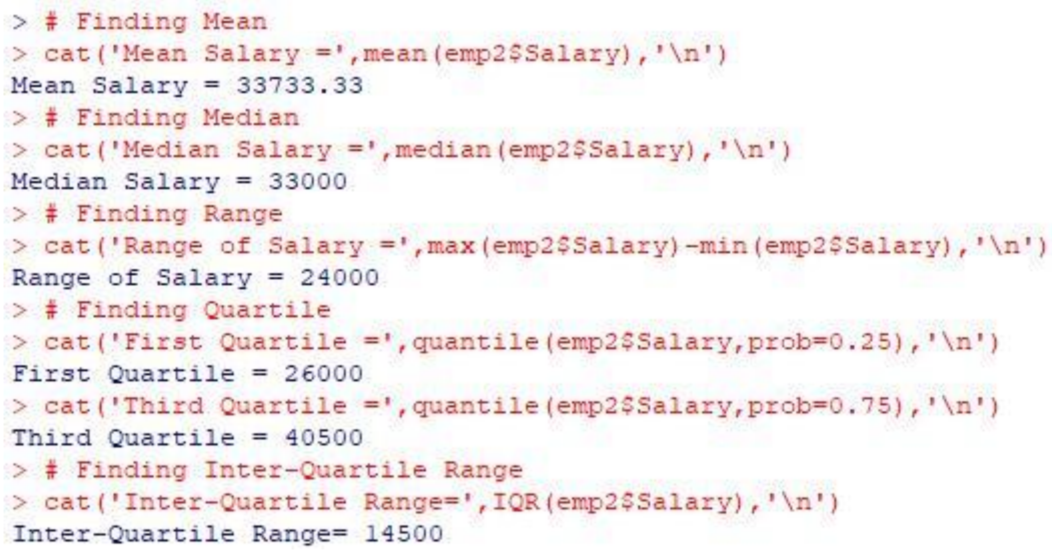
```
> install.packages('xlsx')
```

3. **Importing Data from .xlsx File:**



```
> emp2<-read.xlsx('employee.xlsx',sheetIndex=1)
> print(emp2)
   Emp_Id Emp_Name        DOJ Salary Department
1    1001    Sadik 2012-06-07  47000    Finance
2    1002    Pinky 2012-11-15  45000         HR
3    1003    Manoj 2013-03-03  43000 Operations
4    1004     Aman 2013-08-27  38000         IT
5    1005    Sonam 2013-12-15  29000      Admin
6    1006   Rajesh 2014-10-04  23000      Admin
7    1007   Ramesh 2014-10-04  41000         HR
8    1008  Radhika 2014-10-07  40000 Operations
9    1009   Manish 2014-10-17  25000         IT
10   1010   Ritika 2014-10-17  33000         HR
11   1011    Aryan 2014-10-28  28000 Operations
12   1012     Ayan 2014-10-28  39000    Finance
13   1013   Suyash 2014-11-07  23000         IT
14   1014   Naresh 2014-11-07  27000      Admin
15   1015    Jyoti 2014-11-09  25000      Admin
```

4. **Finding Mean, Median, Range, Quartiles, Inter-Quartile Range:**



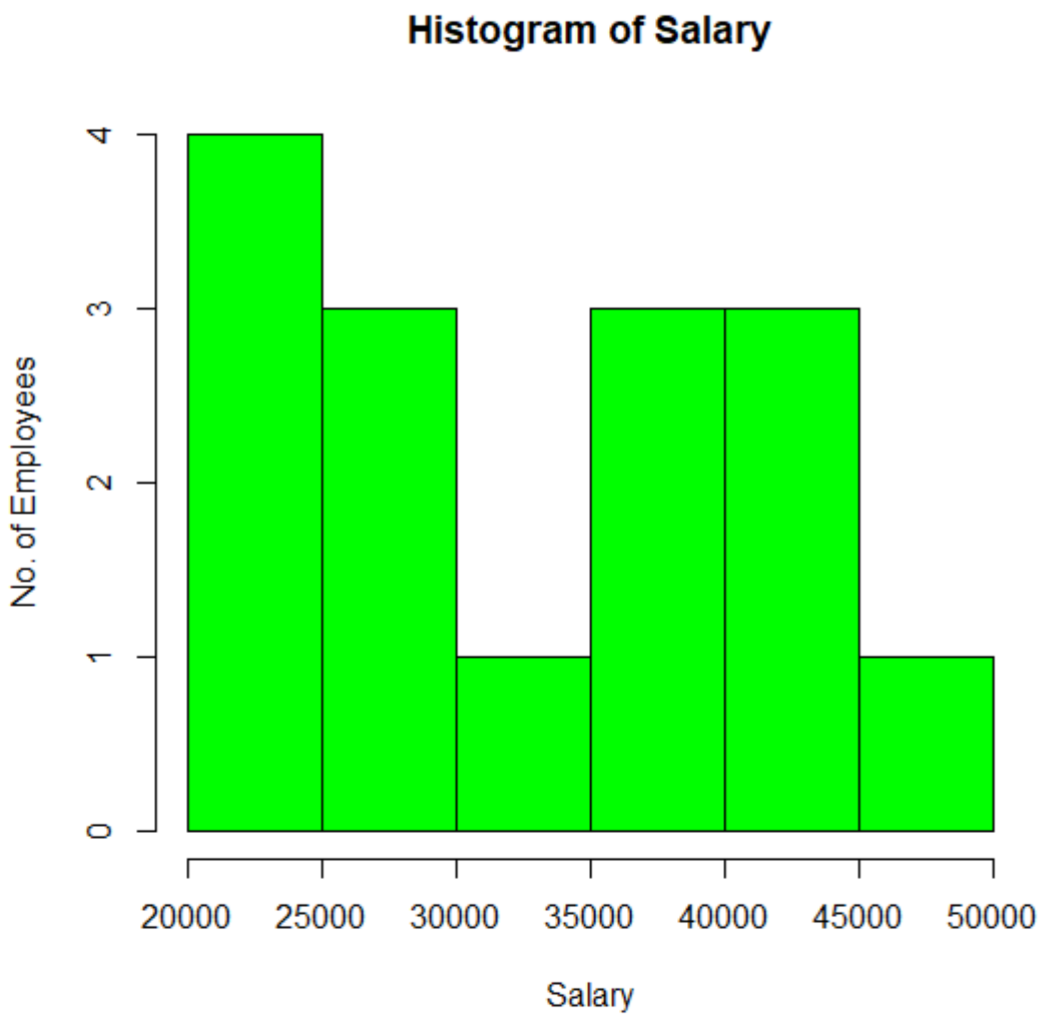
```
> # Finding Mean
> cat('Mean Salary =',mean(emp2$Salary),'\n')
Mean Salary = 33733.33
> # Finding Median
> cat('Median Salary =',median(emp2$Salary),'\n')
Median Salary = 33000
> # Finding Range
> cat('Range of Salary =',max(emp2$Salary)-min(emp2$Salary),'\n')
Range of Salary = 24000
> # Finding Quartile
> cat('First Quartile =',quantile(emp2$Salary,prob=0.25),'\n')
First Quartile = 26000
> cat('Third Quartile =',quantile(emp2$Salary,prob=0.75),'\n')
Third Quartile = 40500
> # Finding Inter-Quartile Range
> cat('Inter-Quartile Range=',IQR(emp2$Salary),'\n')
Inter-Quartile Range= 14500
```

5. Finding Mode:

```
> # Creating getMode function
> getMode<-function(x)
+ {
+ u<-unique(x)
+ u[which.max(tabulate(match(x,u)))]
+ }
> # Finding Mode
> cat('Mode of Salary =',getMode(emp2$Salary),'\n')
Mode of Salary = 23000
```

6. Histogram:

```
> hist(emp2$Salary,main='Histogram of Salary',xlab='Salary',ylab='No. of Employees',col='green')
```



PRACTICAL No. 5

FINDING STANDARD DEVIATION,

VARIANCE & CO-VARIANCE

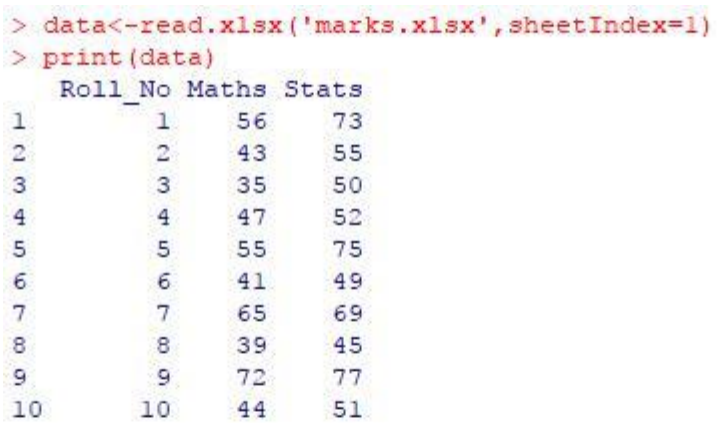
OF EXCEL/.CSV DATA

AIM: Using R import the data from Excel/.CSV file and find standard deviation, variance and co-variance.

SOURCE CODE & OUTPUT:

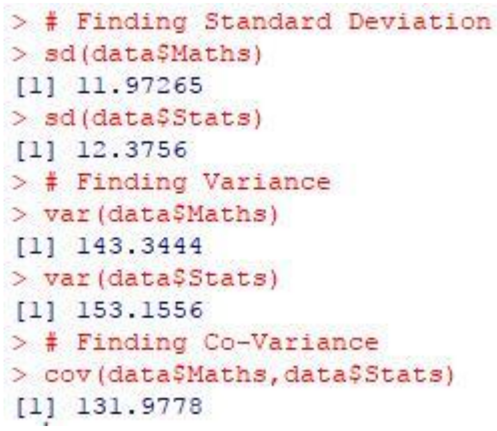
- **Working with Excel File:**

1. **Copy and paste .xlsx file in working directory.**
2. **Importing Data from .xlsx File:**



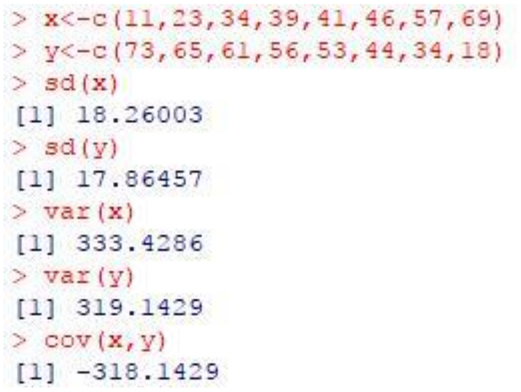
```
> data<-read.xlsx('marks.xlsx',sheetIndex=1)
> print(data)
   Roll_No Maths Stats
1        1    56    73
2        2    43    55
3        3    35    50
4        4    47    52
5        5    55    75
6        6    41    49
7        7    65    69
8        8    39    45
9        9    72    77
10      10    44    51
```

3. **Finding Standard Deviation, Variance & Co-Variance:**



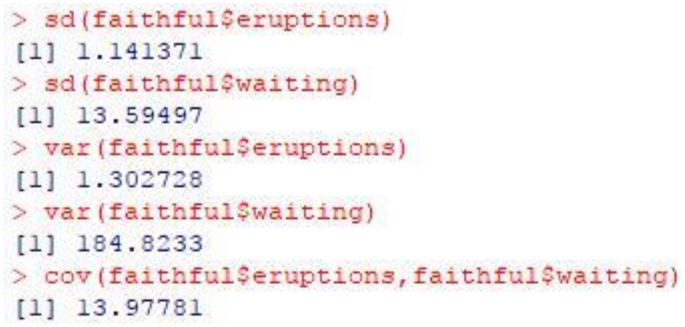
```
> # Finding Standard Deviation
> sd(data$Maths)
[1] 11.97265
> sd(data$Stats)
[1] 12.3756
> # Finding Variance
> var(data$Maths)
[1] 143.3444
> var(data$Stats)
[1] 153.1556
> # Finding Co-Variance
> cov(data$Maths,data$Stats)
[1] 131.9778
```

4. **Finding Standard Deviation, Variance & Co-Variance for given x & y:**



```
> x<-c(11,23,34,39,41,46,57,69)
> y<-c(73,65,61,56,53,44,34,18)
> sd(x)
[1] 18.26003
> sd(y)
[1] 17.86457
> var(x)
[1] 333.4286
> var(y)
[1] 319.1429
> cov(x,y)
[1] -318.1429
```

5. **Finding Standard Deviation, Variance & Co-Variance from faithful Dataset:**



```
> sd(faithful$eruptions)
[1] 1.141371
> sd(faithful$waiting)
[1] 13.59497
> var(faithful$eruptions)
[1] 1.302728
> var(faithful$waiting)
[1] 184.8233
> cov(faithful$eruptions,faithful$waiting)
[1] 13.97781
```

PRACTICAL No. 6

FINDING SKEWNESS & KURTOSIS

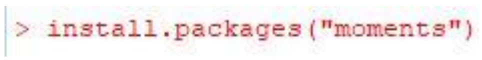
OF EXCEL/.CSV DATA

AIM: Using R import the data from Excel/.CSV file and find skewness and kutosis.

SOURCE CODE & OUTPUT:

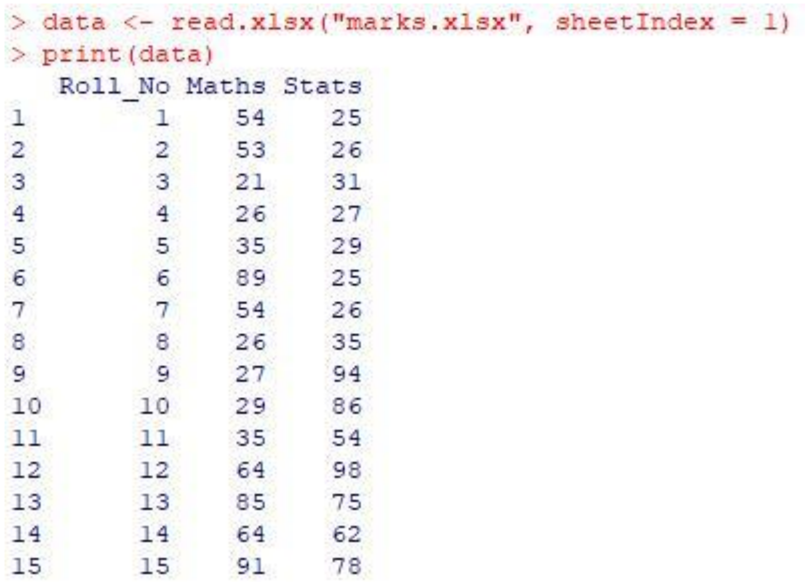
- **Working with Excel File:**

1. **Copy and paste .xlsx file in working directory.**
2. **Installing moments package:**



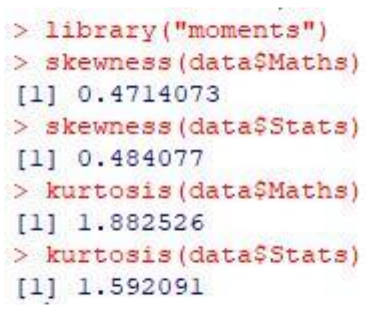
```
> install.packages("moments")
```

3. **Importing Data from .xlsx File:**



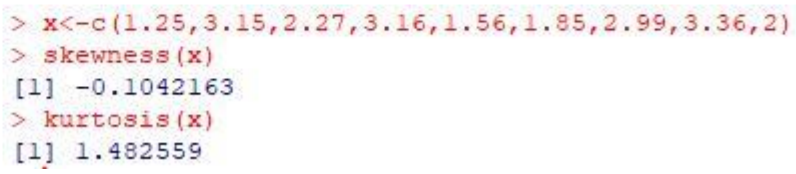
```
> data <- read.xlsx("marks.xlsx", sheetIndex = 1)
> print(data)
   Roll_No Maths Stats
1        1    54    25
2        2    53    26
3        3    21    31
4        4    26    27
5        5    35    29
6        6    89    25
7        7    54    26
8        8    26    35
9        9    27    94
10      10    29    86
11      11    35    54
12      12    64    98
13      13    85    75
14      14    64    62
15      15    91    78
```

4. **Finding Skewness and Kurtosis:**



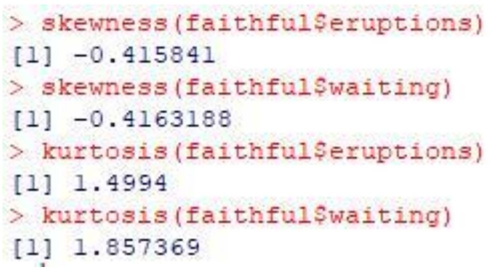
```
> library("moments")
> skewness(data$Maths)
[1] 0.4714073
> skewness(data$Stats)
[1] 0.484077
> kurtosis(data$Maths)
[1] 1.882526
> kurtosis(data$Stats)
[1] 1.592091
```

5. **Finding Skewness and Kurtosis for given x:**



```
> x<-c(1.25,3.15,2.27,3.16,1.56,1.85,2.99,3.36,2)
> skewness(x)
[1] -0.1042163
> kurtosis(x)
[1] 1.482559
```

6. **Finding Skewness and Kurtosis from faithful Dataset:**



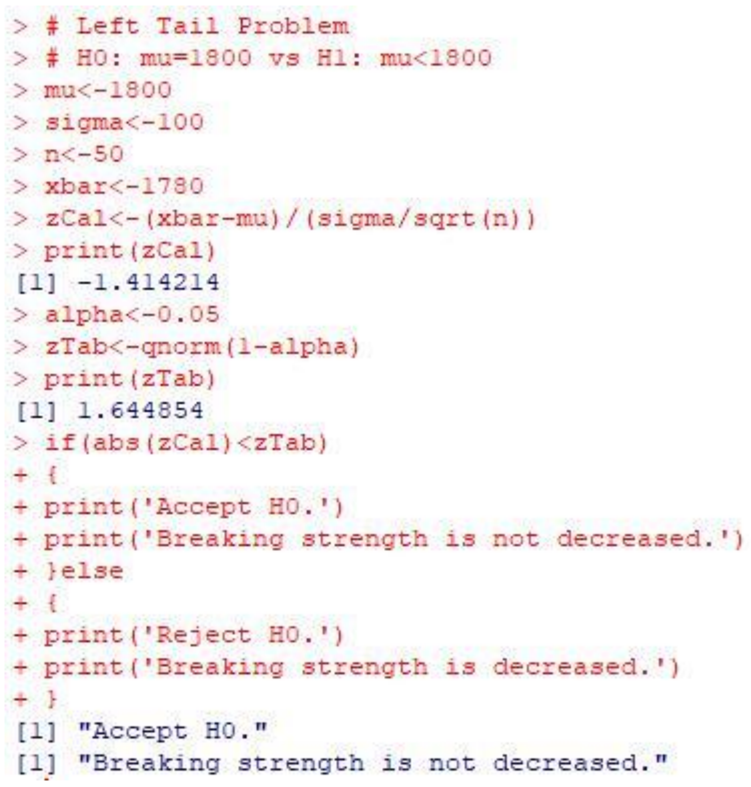
```
> skewness(faithful$eruptions)
[1] -0.415841
> skewness(faithful$waiting)
[1] -0.4163188
> kurtosis(faithful$eruptions)
[1] 1.4994
> kurtosis(faithful$waiting)
[1] 1.857369
```

PRACTICAL No. 7

HYPOTHESIS TESTING

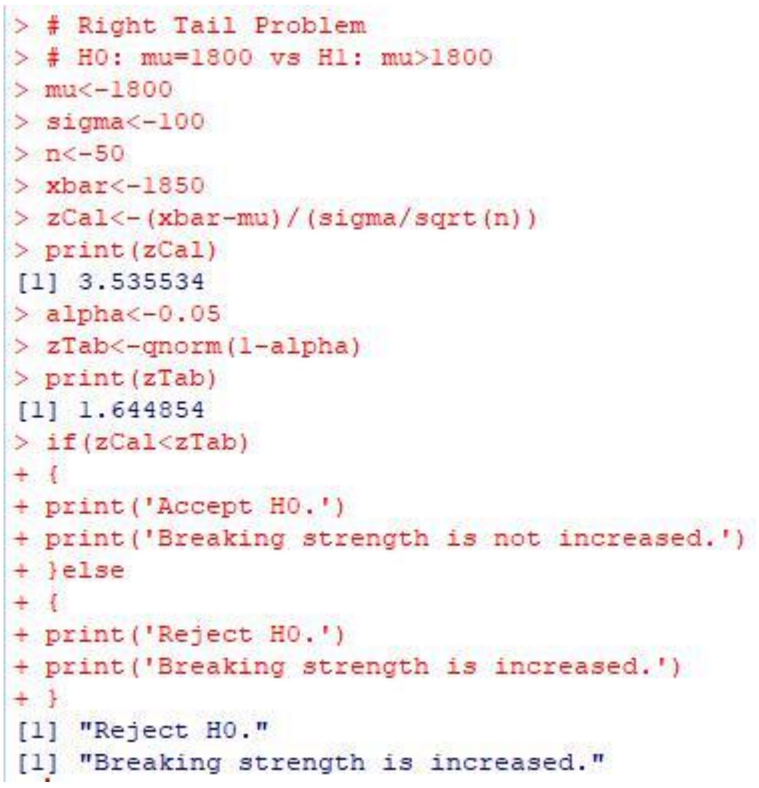
AIM: Perform hypothesis testing for the following:

Q.1 The mean breaking strength of cables produced by a manufacturer have a mean of 1800 lb and a standard deviation of 100 lb. A sample of 50 cables is tested and it is found that the mean breaking strength is 1780 lb. Test the hypothesis that the mean breaking strength of the cables has decreased at 0.05 significance level.



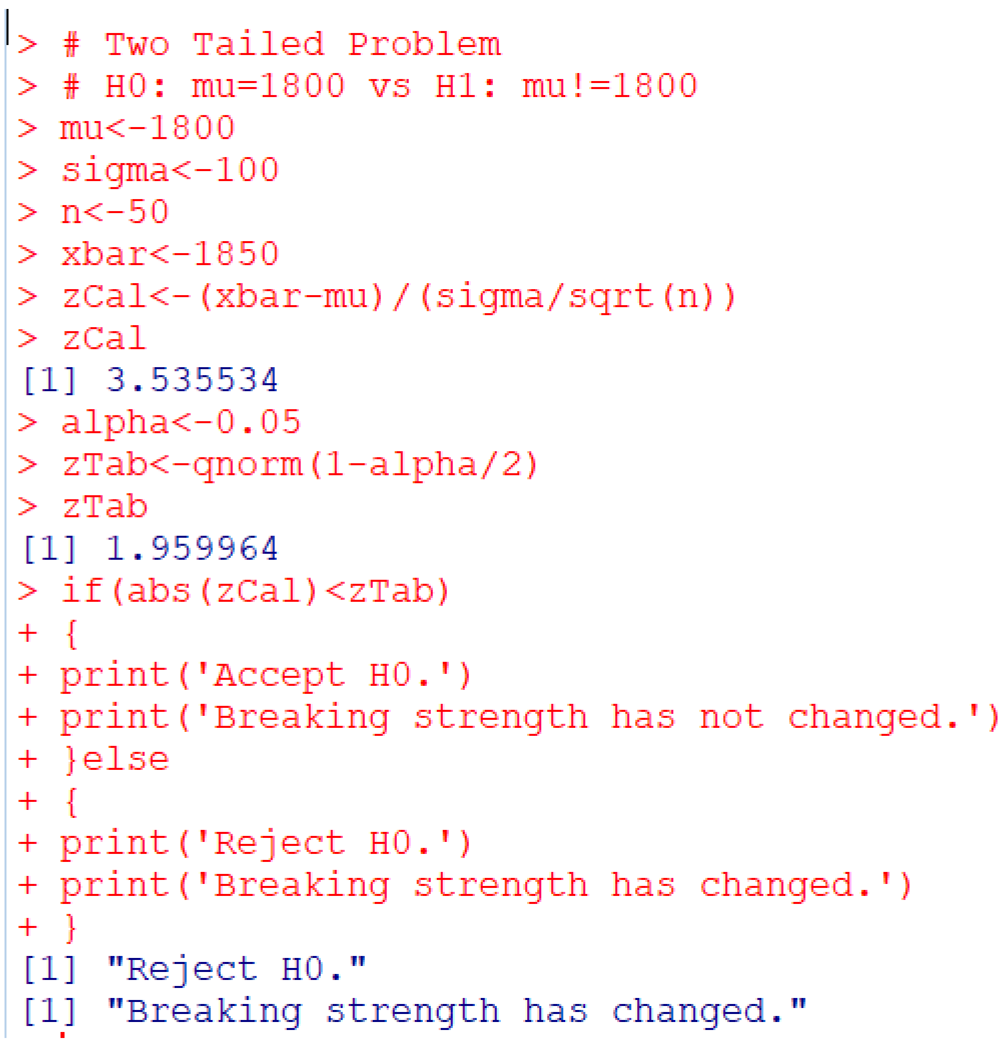
```
> # Left Tail Problem
> # H0: mu=1800 vs H1: mu<1800
> mu<-1800
> sigma<-100
> n<-50
> xbar<-1780
> zCal<-(xbar-mu)/(sigma/sqrt(n))
> print(zCal)
[1] -1.414214
> alpha<-0.05
> zTab<-qnorm(1-alpha)
> print(zTab)
[1] 1.644854
> if(abs(zCal)<zTab)
+ {
+ print('Accept H0.')
+ print('Breaking strength is not decreased.')
+ }else
+ {
+ print('Reject H0.')
+ print('Breaking strength is decreased.')
+ }
[1] "Accept H0."
[1] "Breaking strength is not decreased."
```

Q.2 The mean breaking strength of cables produced by a manufacturer have a mean of 1800 lb and a standard deviation of 100 lb. A sample of 50 cables is tested and it is found that the mean breaking strength is 1850 lb. Test the hypothesis that the mean breaking strength of the cables has increased at 0.05 significance level.



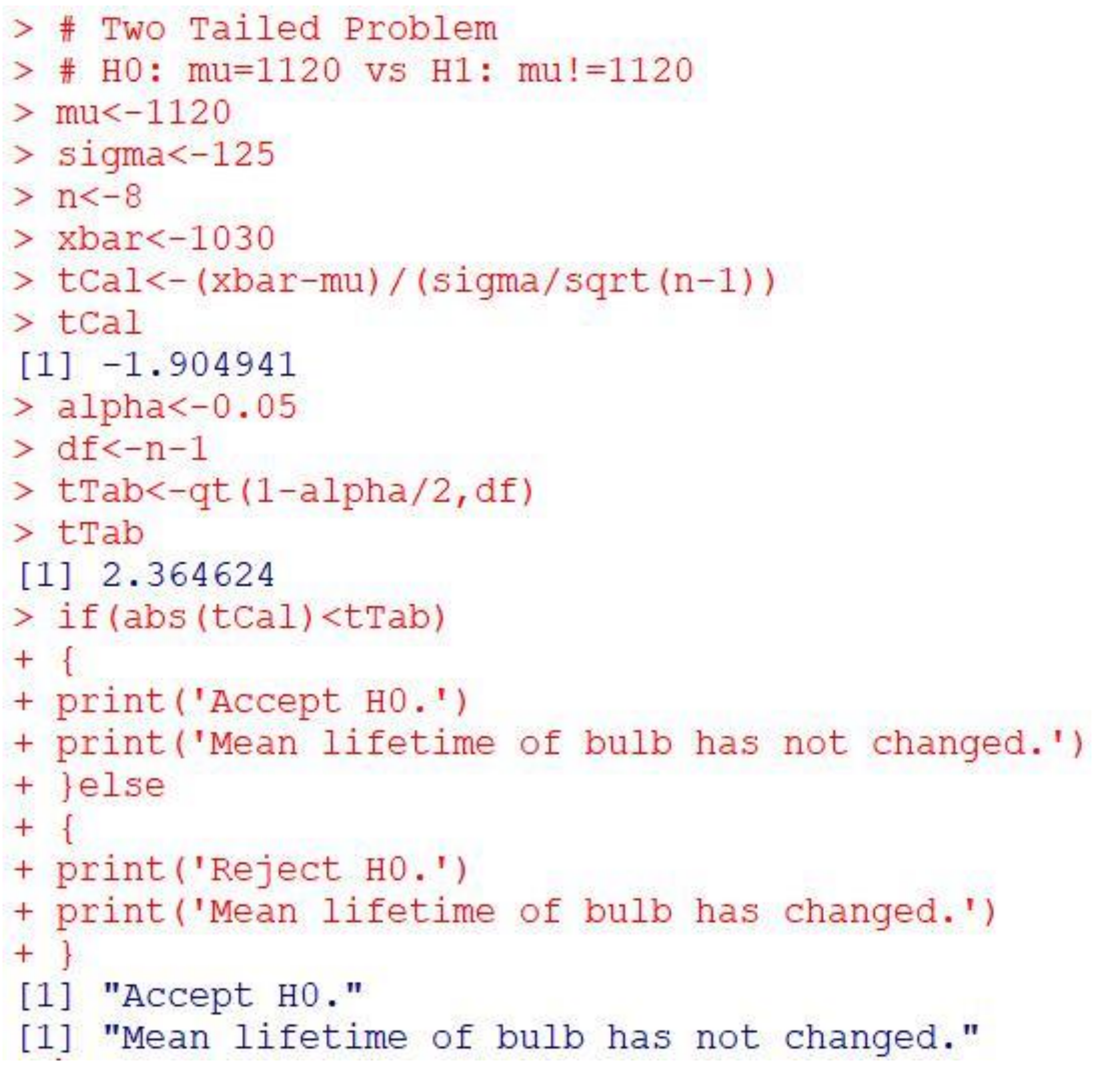
```
> # Right Tail Problem
> # H0: mu=1800 vs H1: mu>1800
> mu<-1800
> sigma<-100
> n<-50
> xbar<-1850
> zCal<-(xbar-mu)/(sigma/sqrt(n))
> print(zCal)
[1] 3.535534
> alpha<-0.05
> zTab<-qnorm(1-alpha)
> print(zTab)
[1] 1.644854
> if(zCal<zTab)
+ {
+ print('Accept H0.')
+ print('Breaking strength is not increased.')
+ }else
+ {
+ print('Reject H0.')
+ print('Breaking strength is increased.')
+ }
[1] "Reject H0."
[1] "Breaking strength is increased."
```

Q.3 The mean breaking strength of cables produced by a manufacturer have a mean of 1800 lb and a standard deviation of 100 lb. A sample of 50 cables is tested and it is found that the mean breaking strength is 1850 lb. Test the hypothesis that the mean breaking strength of the cables has changed at 0.05 significance level.



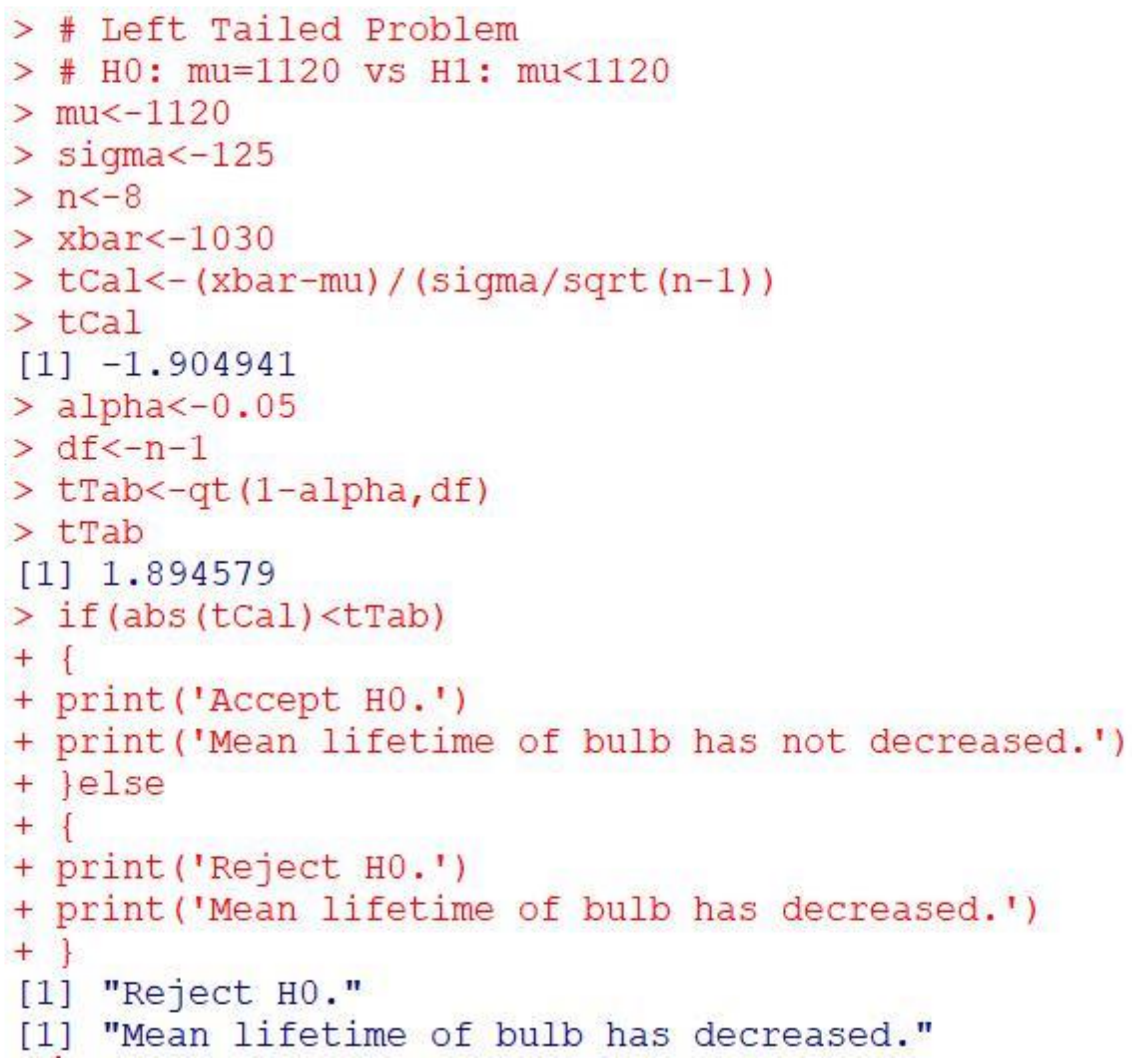
```
> # Two Tailed Problem
> # H0: mu=1800 vs H1: mu!=1800
> mu<-1800
> sigma<-100
> n<-50
> xbar<-1850
> zCal<-(xbar-mu)/(sigma/sqrt(n))
> zCal
[1] 3.535534
> alpha<-0.05
> zTab<-qnorm(1-alpha/2)
> zTab
[1] 1.959964
> if(abs(zCal)<zTab)
+ {
+ print('Accept H0.')
+ print('Breaking strength has not changed.')
+ }else
+ {
+ print('Reject H0.')
+ print('Breaking strength has changed.')
+ }
[1] "Reject H0."
[1] "Breaking strength has changed."
```

Q.4 The mean lifetime of electric light bulbs produced by a company has in the past been 1120h with a standard deviation of 125h. A sample of 8 electric bulbs recently chosen from supply of newly produced bulb showed a mean lifetime of 1030h. Test the hypothesis that the mean lifetime of the bulb has not changed at 0.05 significance level.



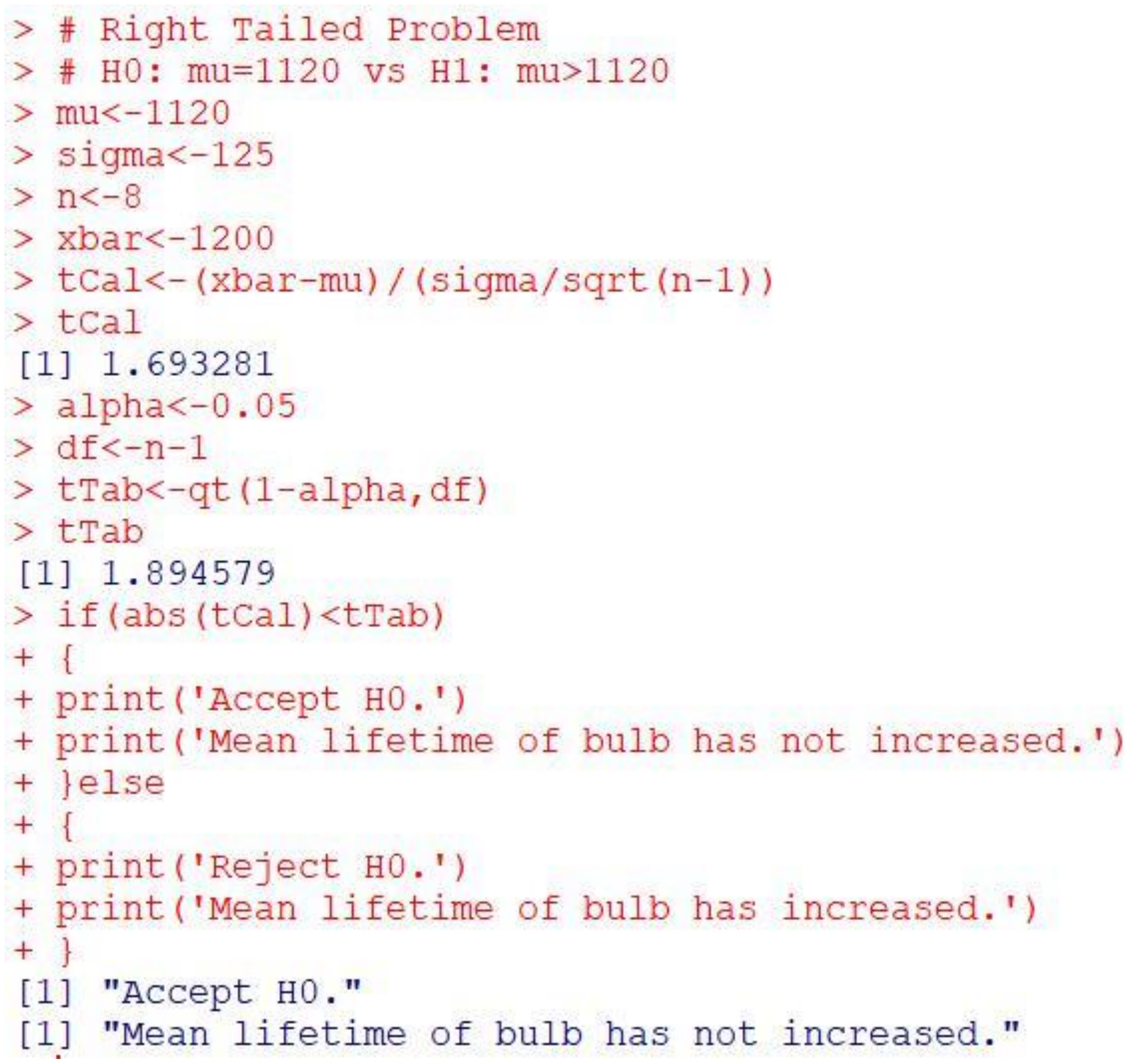
```
> # Two Tailed Problem
> # H0: mu=1120 vs H1: mu!=1120
> mu<-1120
> sigma<-125
> n<-8
> xbar<-1030
> tCal<-(xbar-mu)/(sigma/sqrt(n-1))
> tCal
[1] -1.904941
> alpha<-0.05
> df<-n-1
> tTab<-qt(1-alpha/2,df)
> tTab
[1] 2.364624
> if(abs(tCal)<tTab)
+ {
+ print('Accept H0.')
+ print('Mean lifetime of bulb has not changed.')
+ }else
+ {
+ print('Reject H0.')
+ print('Mean lifetime of bulb has changed.')
+ }
[1] "Accept H0."
[1] "Mean lifetime of bulb has not changed."
```

Q.5 The mean lifetime of electric light bulbs produced by a company has in the past been 1120h with a standard deviation of 125h. A sample of 8 electric bulbs recently chosen from supply of newly produced bulb showed a mean lifetime of 1030h. Test the hypothesis that the mean lifetime of the bulb has decreased at 0.05 significance level.



```
> # Left Tailed Problem
> # H0: mu=1120 vs H1: mu<1120
> mu<-1120
> sigma<-125
> n<-8
> xbar<-1030
> tCal<-(xbar-mu)/(sigma/sqrt(n-1))
> tCal
[1] -1.904941
> alpha<-0.05
> df<-n-1
> tTab<-qt(1-alpha,df)
> tTab
[1] 1.894579
> if(abs(tCal)<tTab)
+ {
+ print('Accept H0.')
+ print('Mean lifetime of bulb has not decreased.')
+ }else
+ {
+ print('Reject H0.')
+ print('Mean lifetime of bulb has decreased.')
+ }
[1] "Reject H0."
[1] "Mean lifetime of bulb has decreased."
```

Q.6 The mean lifetime of electric light bulbs produced by a company has in the past been 1120h with a standard deviation of 125h. A sample of 8 electric bulbs recently chosen from supply of newly produced bulb showed a mean lifetime of 1200h. Test the hypothesis that the mean lifetime of the bulb has increased at 0.05 significance level.



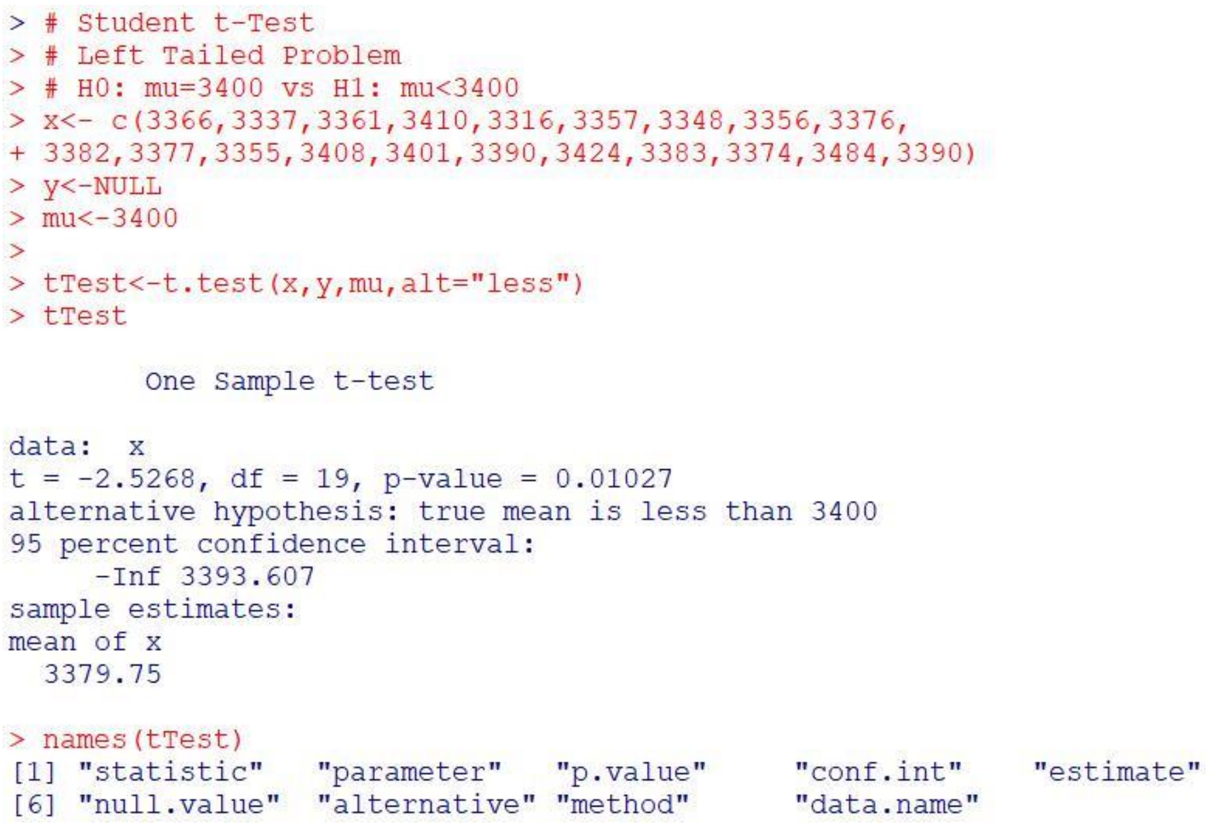
```
> # Right Tailed Problem
> # H0: mu=1120 vs H1: mu>1120
> mu<-1120
> sigma<-125
> n<-8
> xbar<-1200
> tCal<-(xbar-mu)/(sigma/sqrt(n-1))
> tCal
[1] 1.693281
> alpha<-0.05
> df<-n-1
> tTab<-qt(1-alpha,df)
> tTab
[1] 1.894579
> if(abs(tCal)<tTab)
+ {
+ print('Accept H0.')
+ print('Mean lifetime of bulb has not increased.')
+ }else
+ {
+ print('Reject H0.')
+ print('Mean lifetime of bulb has increased.')
+ }
[1] "Accept H0."
[1] "Mean lifetime of bulb has not increased."
```

Q.7

Q.7 Test the hypothesis $H_0: \mu = 3400\ vs\ H_1: \mu < 3400$ for the following data at 5% LOS.
3366, 3337, 3361, 3410, 3316, 3357, 3348, 3356, 3376, 3382, 3377, 3355, 3408, 3401, 3390, 3424, 3383, 3374, 3384, 3390.

t.test(x,y,mu,alt,conf.level,paired)

where,	**x:**	**vector of observations**
	y:	**vector of observations**
		set NULL if not applicable
	mu:	**specified value of true mean**
	alt:	**alternative hypothesis (can take value 'less', 'greater', 'two.sided')**
	conf.level:	**confidence level (1-alpha)**
		default value is 0.95
	paired:	**can take value TRUE or FALSE**
		default value is FALSE



```
> # Student t-Test
> # Left Tailed Problem
> # H0: mu=3400 vs H1: mu<3400
> x<- c(3366,3337,3361,3410,3316,3357,3348,3356,3376,
+ 3382,3377,3355,3408,3401,3390,3424,3383,3374,3484,3390)
> y<-NULL
> mu<-3400
>
> tTest<-t.test(x,y,mu,alt="less")
> tTest

        One Sample t-test

data:  x
t = -2.5268, df = 19, p-value = 0.01027
alternative hypothesis: true mean is less than 3400
95 percent confidence interval:
     -Inf 3393.607
sample estimates:
mean of x
  3379.75

> names(tTest)
[1] "statistic"   "parameter"   "p.value"     "conf.int"    "estimate"
[6] "null.value"  "alternative" "method"      "data.name"
```

```
> tTest$statistic
        t
-2.526799
> tTest$parameter
df
19
> tTest$p.value
[1] 0.01027214
> tTest$conf.int
[1]      -Inf 3393.607
attr(,"conf.level")
[1] 0.95
> tTest$estimate
mean of x
  3379.75
> tTest$null.value
mean
3400
> tTest$alternative
[1] "less"
> tTest$method
[1] "One Sample t-test"
> tTest$data.name
[1] "x"

> if(tTest$p.value<0.05)
+ {
+ print('Reject H0 i.e. population mean is less than 3400.')
+ }else
+ {
+ print('Accept H0 i.e. population mean is 3400.')
+ }
[1] "Reject H0 i.e. population mean is less than 3400."
```

Q.8 The following data refer to the amount of coffee (in ounces) filled by a machine in six randomly picked jars: 15.7, 15.9, 16.3, 16.2, 15.7 and 15.9. Is the true mean amount of coffee in a jar is 16 ounces? Use LOS 5%.

```
> # Student t-Test
> # Two Tailed Problem
> # H0: mu=16 vs H1: mu!=16
> x<-c(15.7,15.9,16.3,16.2,15.7,15.9)
> y<-NULL
> mu<-16
>
> tTest<-t.test(x,y,mu,alt="two.sided")
> tTest

        One Sample t-test

data:  x
t = -0.48795, df = 5, p-value = 0.6462
alternative hypothesis: true mean is not equal to 16
95 percent confidence interval:
 15.68659 16.21341
sample estimates:
mean of x
    15.95

> if(tTest$p.value<0.05)
+ {
+ print('Reject H0 i.e. mean amount of coffee in a jar is not 16.')
+ }else
+ {
+ print('Accept H0 i.e. mean amount of coffee in a jar is 16.')
+ }
[1] "Accept H0 i.e. mean amount of coffee in a jar is 16."
```

Q.9 Below are given the gain in weights (in lbs) of pigs fed on two diets A and B.
Diet A: 25,32,30,43,24,14,32,24,31,31,35,25
Diet B: 44,34,22,10,47,31,40,30,32,35,18,21,35,29,22
Test, if the two diets differ significantly as regards their effect on increase in weight. Use LOS 5%.

```
> # Student t-Test for double mean
> # Two Tailed Problem
> # H0: No significant difference between means of x and y vs
> # H1: Significant difference between means of x and y
> x<-c(25,32,30,34,24,14,32,24,30,31,35,25)
> y<-c(44,34,22,10,47,31,40,30,32,35,18,21,35,29,22)
>
> tTest<-t.test(x,y,var.equal=T)
> tTest

        Two Sample t-test

data:  x and y
t = -0.61028, df = 25, p-value = 0.5472
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -8.749507  4.749507
sample estimates:
mean of x mean of y
       28        30
> if(tTest$p.value<0.05)
+ {
+ print('Reject H0 i.e. there is significant difference b/n means.')
+ }else
+ {
+ print('Accept H0 i.e. there is no significant difference b/n means.')
+ }
[1] "Accept H0 i.e. there is no significant difference b/n means."
```

Q.10 Eleven school boys were given a test in mathematics. They were given a month's tuition and a second test was held at the end of it. Do the marks give evidence that the student's have benefited by the extra coaching? Use LOS 5%.
Marks in test-1: 23, 20, 19, 21, 18, 20, 18, 17, 23, 16, 19
Marks in test-2: 24, 19, 22, 18, 20, 22, 20, 20, 23, 20, 17

```
> # Paired t-Test
> # Two Tailed Problem
> # H0: No significant difference between x and y vs
> # H1: Significant difference between x and y
> x<-c(23,20,19,21,18,20,18,17,23,16,19)
> y<-c(24,19,22,18,20,22,20,20,23,20,17)
>
> tTest<-t.test(x,y,paired=T)
> tTest

        Paired t-test

data:  x and y
t = -1.4832, df = 10, p-value = 0.1688
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.5022109  0.5022109
sample estimates:
mean of the differences
                     -1
> if(tTest$p.value<0.05)
+ {
+ print('Reject H0 i.e. there is significant difference b/n x & y.')
+ }else
+ {
+ print('Accept H0 i.e. there is no significant difference b/n x & y.')
+ }
[1] "Accept H0 i.e. there is no significant difference b/n x & y."
```

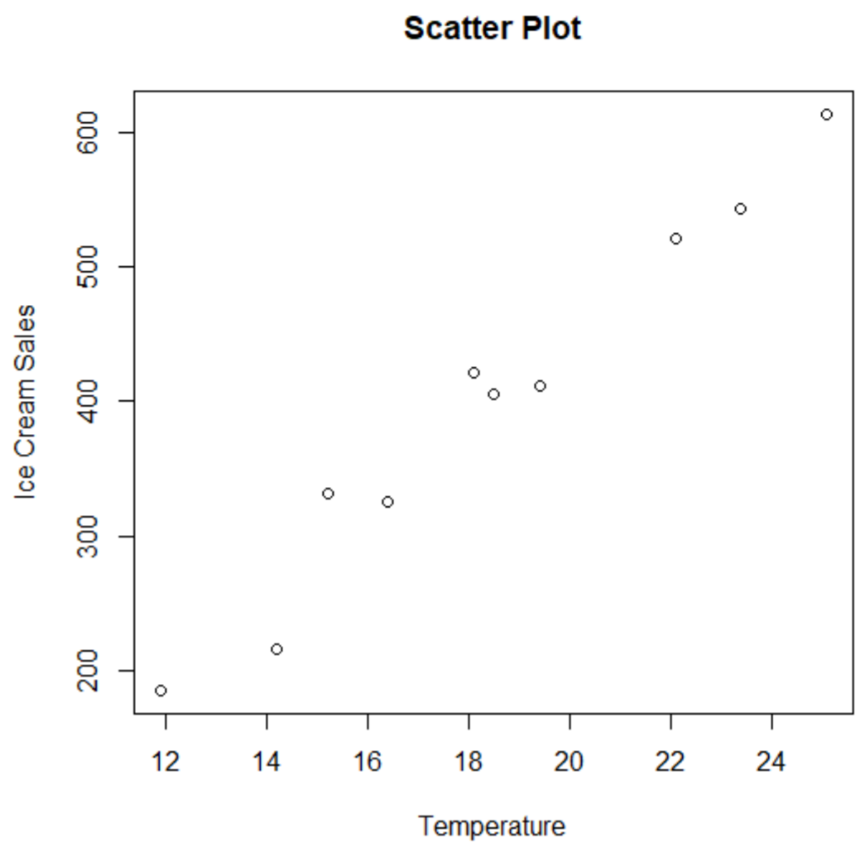
Practical No. 8

CORRELATION

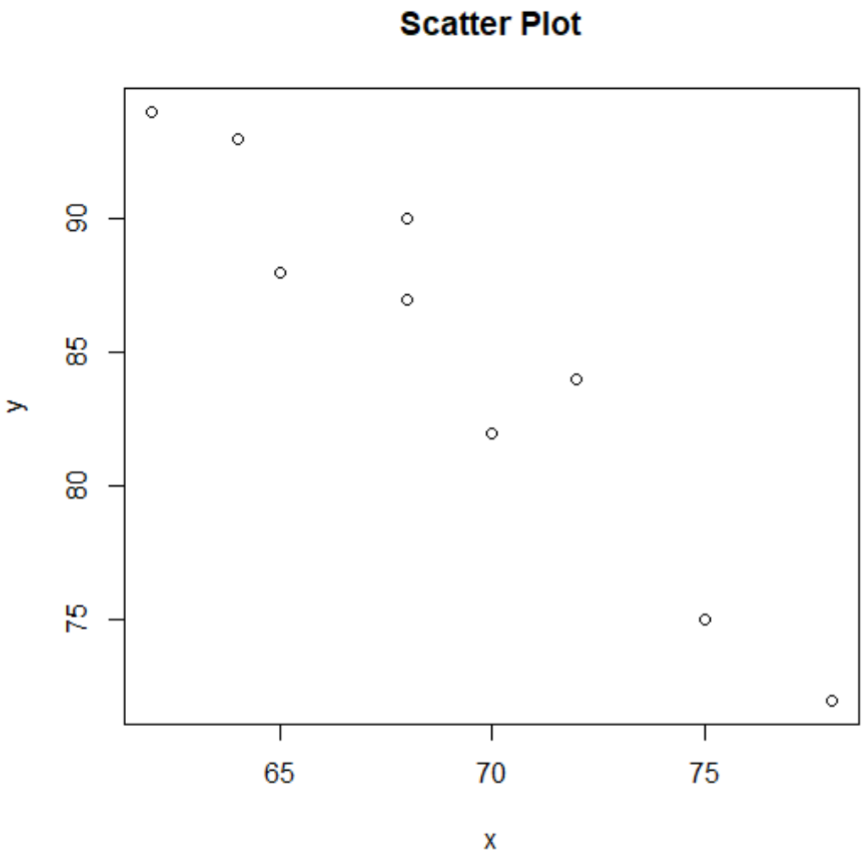
AIM: Plot the scatter diagram and find the correlation coefficient using R.

SOURCE CODE & OUTPUT:

```
> # Positive Correlation Example
> temp<-c(14.2,16.4,11.9,15.2,18.5,22.1,19.4,25.1,23.4,18.1)
> sales<-c(215,325,185,332,406,522,412,614,544,421)
> r<-cor(temp,sales,method="pearson")
> cat('Coefficient of Correlation between temperature
+ and ice cream sales is ',r,'.\n')
Coefficient of Correlation between temperature
and ice cream sales is  0.9842961 .
>
> # Scatter diagram
> plot(temp,sales,xlab="Temperature",ylab="Ice Cream Sales",
+ main="Scatter Plot")
```



```
> # Negative Correlation Example
> x<-c(68,72,65,70,62,75,78,64,68)
> y<-c(90,84,88,82,94,75,72,93,87)
> r<-cor(x,y,method="pearson")
> cat('Coefficient of Correlation between x
+ and y is ',r,'.\n')
Coefficient of Correlation between x
and y is  -0.9591069 .
>
> # Scatter diagram
> plot(x,y,xlab="x",ylab="y",main="Scatter Plot")
```



```
> # Spearman's Rank Correlation
> R1<-c(1,2,3,4,5,6,7,8,9,10)
> R2<-c(4,8,1,3,2,5,10,7,6,9)
> R<-cor(R1,R2,method="spearman")
> R
[1] 0.5151515
> cat('Spearman rank correlation coefficient is ',R,'.\n')
Spearman rank correlation coefficient is  0.5151515 .
```

Practical No. 9

REGRESSION

AIM: Perform the linear regression using R.

SOURCE CODE & OUTPUT:

```
> # Regression
> temp<-c(14.2,16.4,11.9,15.2,18.5,22.1,19.4,25.1,23.4,18.1)
> sales<-c(215,325,185,332,406,522,412,614,544,421)
> reg<-lm(sales~temp)
> reg

Call:
lm(formula = sales ~ temp)

Coefficients:
(Intercept)         temp
    -200.60        32.46

> reg$coefficients[1]
(Intercept)
   -200.596
> reg$coefficients[2]
    temp
32.45773
> cat('Regression equation is
+ y=',reg$coefficients[1],'+(',reg$coefficients[2],')x.\n')
Regression equation is
y= -200.596 +( 32.45773 )x.

> fitted(reg)
       1        2        3        4        5        6        7        8
260.3038 331.7108 185.6510 292.7615 399.8720 516.7199 429.0840 614.0931
       9       10
558.9149 386.8889
```

```
> # Regression
> IT<-c(25,28,35,32,31,36,29,38,34,32)
> STATS<-c(43,46,49,41,36,32,31,30,33,39)
> reg<-lm(STATS~IT)
> reg

Call:
lm(formula = STATS ~ IT)

Coefficients:
(Intercept)           IT
    59.2571      -0.6643

> reg$coefficients[1]
(Intercept)
   59.25714
> reg$coefficients[2]
        IT
-0.6642857
> cat('Regression equation is
+ y=',reg$coefficients[1],'+(',reg$coefficients[2],')x.\n')
Regression equation is
y= 59.25714 +( -0.6642857 )x.

> fitted(reg)
       1        2        3        4        5        6        7        8
42.65000 40.65714 36.00714 38.00000 38.66429 35.34286 39.99286 34.01429
       9       10
36.67143 38.00000


> # Multiple Linear Regression
> x1<-c(3.33,3.96,4.58,5.33,3.13,3.67,4.58,3.00,4.50,3.50)
> x2<-c(3.92,2.58,3.92,3.08,3.54,4.17,4.17,3.67,4.67,4.25)
> y<-c(3.38,3.61,3.83,3.92,3.92,3.96,4.00,4.00,4.04,4.04)
> reg<-lm(y~x1+x2)
> reg

Call:
lm(formula = y ~ x1 + x2)

Coefficients:
(Intercept)           x1           x2
    3.07253      0.05609      0.15156

> cat('Regression equation is
+ y=',reg$coefficients[1],'+(',reg$coefficients[2],')x1','+(',reg$coefficients[3],')x2.\n')
Regression equation is
y= 3.072531 +( 0.05609011 )x1 +( 0.1515576 )x2.
> fitted(reg)
       1        2        3        4        5        6        7        8        9       10
3.853417 3.685667 3.923530 3.838289 3.784607 3.910377 3.961419 3.797018 4.032711 3.912966
```