Semester	B.E. Semester VIII – INFT
Subject	R programming lab
Laboratory Teacher:	Shruti Agrawal
Laboratory	-

Student Name	Soham Sahare	
Roll Number	18101B0010	
Grade and Subject Teacher's Signature		

Experiment Number	5	
Experiment Title	To understand Matrix in R	
Problem Statement	Write a R program to accept matrix element from user.	
Resources / Apparatus Required	Hardware: Desktop/Laptop	Software: R studio
Code:	**Matrix:** Matrices are two-dimensional data structures in R and are arranged in a rectangular layout. Matrices can contain only one data type. A data structure is a particular way of organizing data in a computer so that it can be used effectively. The idea is to reduce the space and time complexities of different tasks. It's a m*n array with similar data type. It is created using a vector input. A matrix in which the	

number of rows is equal to the number of columns is said to be a square matrix. You can perform many arithmetic operations on R matrix like – addition, subtraction, multiplication, and divisions.

matrix() creates a matrix from the given set of values.
Syntax: **matrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE, dimnames = NULL)**

Arguments
data
an optional data vector (including a list or expression vector). Non-atomic classed R objects are coerced by as.vector and all attributes discarded.

nrow
the desired number of rows.

ncol
the desired number of columns.

byrow
logical. If FALSE (the default) the matrix is filled by columns, otherwise the matrix is filled by rows.

dimnames
A dimnames attribute for the matrix: NULL or a list of length 2 giving the row and column names respectively. An empty list is treated as NULL, and a list of length one as row names. The list can be named, and the list names will be used as names for the dimensions.

as.matrix: The default method for as.matrix calls as.vector(x), and hence e.g. coerces factors to character vectors.

When coercing a vector, it produces a one-column matrix, and promotes the names (if any) of the vector to the rownames of the matrix.

Code:

```
r = as.integer(readline(prompt = "Enter number of rows: "))
c = as.integer(readline(prompt = "Enter number of columns: "))
values = readline(prompt = "Enter numbers for matrix: ")
```

With as.matrix

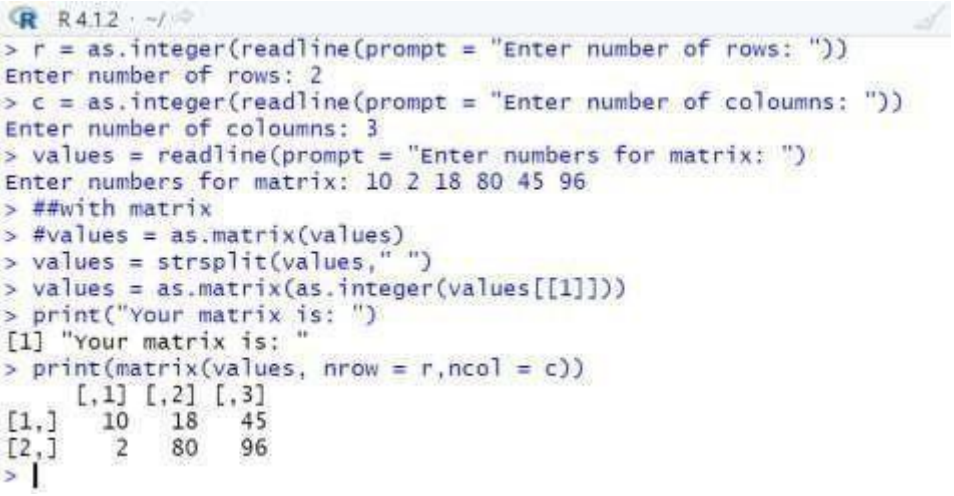
```
#values = as.matrix(values)
values = strsplit(values," ")
values = as.matrix(as.integer(values[[1]]))
print("Your matrix is: ")
print(matrix(values, nrow = r,ncol = c))
```

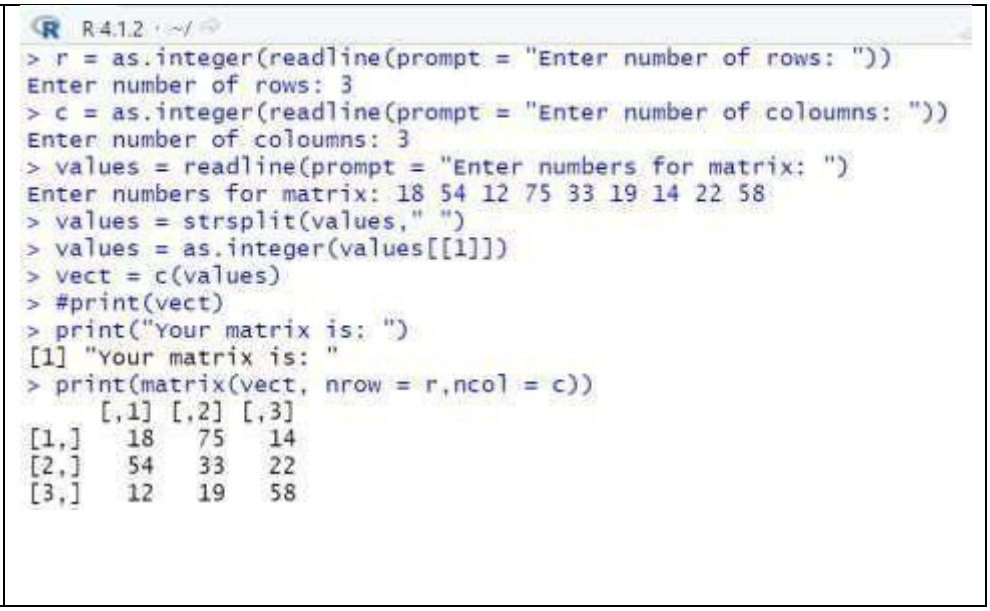
OR

With vector and matrix

```
r = as.integer(readline(prompt = "Enter number of rows: "))
c = as.integer(readline(prompt = "Enter number of coloumns: "))
values = readline(prompt = "Enter numbers for matrix: ")
values = strsplit(values," ")
values = as.integer(values[[1]])
vect = c(values)
#print(vect)
print("Your matrix is: ")
print(matrix(vect, nrow = r,ncol = c))
```

Output:



```
R 4.1.2 · ~/
> r = as.integer(readline(prompt = "Enter number of rows: "))
Enter number of rows: 3
> c = as.integer(readline(prompt = "Enter number of coloumns: "))
Enter number of coloumns: 3
> values = readline(prompt = "Enter numbers for matrix: ")
Enter numbers for matrix: 18 54 12 75 33 19 14 22 58
> values = strsplit(values," ")
> values = as.integer(values[[1]])
> vect = c(values)
> #print(vect)
> print("Your matrix is: ")
[1] "Your matrix is: "
> print(matrix(vect, nrow = r,ncol = c))
     [,1] [,2] [,3]
[1,]   18   75   14
[2,]   54   33   22
[3,]   12   19   58
```

VIT Vidyalankar Institute of Technology ACCREDITED A+ BY NAAC	**Department of Information Technology**

Semester	B.E. Semester VIII – INFT
Subject	R programming lab
Laboratory Teacher:	Prof. Shruti Agrawal
Laboratory	-

Student Name	Soham Sahare	
Roll Number	18101B0010	
Grade and Subject Teacher's Signature		

Experiment Number	6	
Experiment Title	To understand Matrix in R	
Problem Statement	Write a menu driven R program on matrix operations.	
Resources / Apparatus Required	Hardware: Desktop/Laptop	Software: R studio
Code:	**Matrix:** Matrices are two-dimensional data structures in R and are arranged in a rectangular layout. Matrices can contain only one data type. A data structure is a particular way of organizing data in a computer so that it can be used effectively. The idea is to reduce the space and time complexities of different tasks. It's a m*n array with similar data type. It is created using a vector input. A matrix in which the	

number of rows is equal to the number of columns is said to be a square matrix. You can perform many arithmetic operations on R matrix like – addition, subtraction, multiplication, and divisions.

matrix() creates a matrix from the given set of values.
Syntax: **matrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE, dimnames = NULL)**

Arguments
data
an optional data vector (including a list or expression vector). Non-atomic classed R objects are coerced by as.vector and all attributes discarded.

nrow
the desired number of rows.

ncol
the desired number of columns.

byrow
logical. If FALSE (the default) the matrix is filled by columns, otherwise the matrix is filled by rows.

dimnames
A dimnames attribute for the matrix: NULL or a list of length 2 giving the row and column names respectively. An empty list is treated as NULL, and a list of length one as row names. The list can be named, and the list names will be used as names for the dimensions.

Code:

```
r1 = as.integer(readline(prompt = "For matrix 1 \n Enter rows: "))
c1 = as.integer(readline(prompt = "Enter coloumns: "))
valuesm1 = readline(prompt = "Enter numbers for matrix 1: ")
valuesm1 = strsplit(valuesm1," ")
valuesm1 = as.integer(valuesm1[[1]])
m1 = matrix(c(valuesm1),r1,c1)
print(m1)

r2 = as.integer(readline(prompt = "For matrix 2 \n Enter rows: "))
c2 = as.integer(readline(prompt = "Enter coloumns: "))
```

	valuesm2 = readline(prompt = "Enter numbers for matrix 2: ")
	valuesm2 = strsplit(valuesm2," ")
	valuesm2 = as.integer(valuesm2[[1]])
	m2 = matrix(c(valuesm2),r2,c2)
	print(m2)
	
	addsub_matrix = function(a,b){
	  if(ch==1){
	  if ((r1 == r2) && (c1==c2)){
	  return(a+b)
	  }else{
	  return("Enter identical matrices")
	  }
	  }
	  else if(ch==2){
	  if ((r1 == r2) && (c1==c2)){
	  return(a-b)
	  }else{
	  return("Enter identical matrices")
	  }
	  }
	}
	while (TRUE) {
	  print("1. addition \n 2. subtraction 3. multiplication \n 4.division
	  \n 5. remainder \n 6. Exit")
	
	  ch=as.integer(readline(prompt = "Enter choice: "))
	
	  switch (ch,
	  1: print(addsub_matrix(m1,m2)),
	  2: print(addsub_matrix(m1,m2)),
	  3: print(m1 %*% m2),
	  4: print(m1 / m2),
	  5: print(m1 %% m2),
	  6: break,
	
	  print("Invalid choice")
	  )

Output:	

```
R 4.1.2 · ~/
>
For matrix 1
 Enter rows: 2
Enter coloumns: 2
Enter numbers for matrix 1: 1 7 5 3
     [,1] [,2]
[1,]    1    5
[2,]    7    3
For matrix 2
 Enter rows: 2
Enter coloumns: 2
Enter numbers for matrix 2: 12 0 -1 9
     [,1] [,2]
[1,]   12   -1
[2,]    0    9
[1] "1. addition \n 2. subtraction 3. multiplication 4.division
 \n\n      5. remainder \n 6. Exit"
Enter choice: 1
     [,1] [,2]
[1,]   13    4
[2,]    7   12
[1] "1. addition \n 2. subtraction 3. multiplication 4.division
 \n\n      5. remainder \n 6. Exit"
Enter choice: 2
     [,1] [,2]
[1,]  -11    6
[2,]    7   -6
```

```
Enter choice: 3
     [,1] [,2]
[1,]   12   44
[2,]   84   20
[1] "1. addition \n 2. subtraction 3. multiplication 4.division
 \n\n      5. remainder \n 6. Exit"
Enter choice: 4
           [,1]       [,2]
[1,] 0.08333333 -5.0000000
[2,]        Inf  0.3333333
[1] "1. addition \n 2. subtraction 3. multiplication 4.division
 \n\n      5. remainder \n 6. Exit"
Enter choice: 5
     [,1] [,2]
[1,]    1    0
[2,]   NA    3
[1] "1. addition \n 2. subtraction 3. multiplication 4.division
 \n\n      5. remainder \n 6. Exit"
```

```
>
For matrix 1
 Enter rows: 2
Enter coloumns: 3
Enter numbers for matrix 1: 1 4 2 5 3 6
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
For matrix 2
 Enter rows: 3
Enter coloumns: 2
Enter numbers for matrix 2: 10 20 30 11 21 31
     [,1] [,2]
[1,]   10   11
[2,]   20   21
[3,]   30   31
[1] "1. addition \n 2. subtraction 3. multiplication 4.div
ision \n\n      5. remainder "
Enter choice: 3
     [,1] [,2]
[1,]  140  146
[2,]  320  335
```

```
>
For matrix 1
 Enter rows: 2
Enter coloumns: 2
Enter numbers for matrix 1: 3 1 2 4
     [,1] [,2]
[1,]    3    2
[2,]    1    4
For matrix 2
 Enter rows: 3
Enter coloumns: 2
Enter numbers for matrix 2: 3 1 5 2 4 3
     [,1] [,2]
[1,]    3    2
[2,]    1    4
[3,]    5    3
[1] "1. addition \n 2. subtraction 3. multiplication 4.division
 \n\n      5. remainder \n 6. Exit"
Enter choice: 1
[1] "Enter identical matrices"
```

VIT Vidyalankar Institute of Technology ACCREDITED A+ BY NAAC	**Department of Information Technology**

Semester	B.E. Semester VIII – INFT
Subject	R programming lab
Laboratory Teacher:	Prof. Shruti Agrawal
Laboratory	-

Student Name	Soham Sahare	
Roll Number	18101B0010	
Grade and Subject Teacher's Signature		

Experiment Number	7	
Experiment Title	To understand plots in R	
Problem Statement	Write a R program to two vectors containing 10 students name and their percentage marks. Plot Piechart, Barplot, Boxplot, Histogram, Line Graph, Scatter Plot.	
Resources / Apparatus Required	Hardware: Desktop/Laptop	Software: R studio
Code:	**R language** is mostly used for statistics and data analytics purposes to represent the data graphically in the software. To represent those data graphically, charts and graphs are used in R. **Types of R – Charts**	

	• Bar Plot or Bar Chart • Pie Diagram or Pie Chart • Histogram • Scatter Plot • Box Plot • Line Graph **Pie chart:** Pie chart is a circular chart divided into different segments according to the ratio of data provided. The total value of the pie is 100 and the segments tell the fraction of the whole pie. It is another method to represent statistical data in graphical form and **pie()** function is used to perform the same. **Syntax:** pie(x, labels, col, main, radius) **where,** • **x** is data vector • **labels** shows names given to slices • **col** fills the color in the slices as given parameter • **main** shows title name of the pie chart • **radius** indicates radius of the pie chart. It can be between -1 to +1 **Bar chart:** Bar plot or Bar Chart in R is used to represent the values in data vector as height of the bars. The data vector passed to the function is represented over y-axis of the graph. Bar chart can behave like histogram by using **table()** function instead of data vector. **Syntax:** barplot(data, xlab, ylab) **where:** • **data** is the data vector to be represented on y-axis • **xlab** is the label given to x-axis • **ylab** is the label given to y-axis **Box plot:** Box plot shows how the data is distributed in the data vector. It represents five values in the graph i.e., minimum, first quartile,

	second quartile(median), third quartile, the maximum value of the data vector. **Syntax:** boxplot(x, xlab, ylab, notch) **where,** • **x** specifies the data vector • **xlab** specifies the label for x-axis • **ylab** specifies the label for y-axis • **notch,** if TRUE then creates notch on both the sides of the box **Histogram:** Histogram is a graphical representation used to create a graph with bars representing the frequency of grouped data in vector. Histogram is same as bar chart but only difference between them is histogram represents frequency of grouped data rather than data itself. **Syntax:** hist(x, col, border, main, xlab, ylab) **where:** • **x** is data vector • **col** specifies the color of the bars to be filled • **border** specifies the color of border of bars • **main** specifies the title name of histogram • **xlab** specifies the x-axis label • **ylab** specifies the y-axis label **Line graph:** **The plot() function in R** is used to create the line graph. **Syntax:** plot(v, type, col, xlab, ylab) **Parameters:** • **v:** This parameter is a contains only the numeric values • **type:** This parameter has the following value: 1. **"p" :** This value is used to draw only the point s. 2. **"l" :** This value is used to draw only the lines. 3. **"o":** This value is used to draw both points an d lines • **xlab:** This parameter is the label for x axis in the chart. • **ylab:** This parameter is the label for y axis in the chart. • **main:** This parameter main is the title of the chart.

- **col:** This parameter is used to give colors to both the po ints and lines.

Scatter plot:

A Scatter plot is another type of graphical representation used to plot the points to show relationship between two data vectors. One of the data vectors is represented on x-axis and another on y-axis.

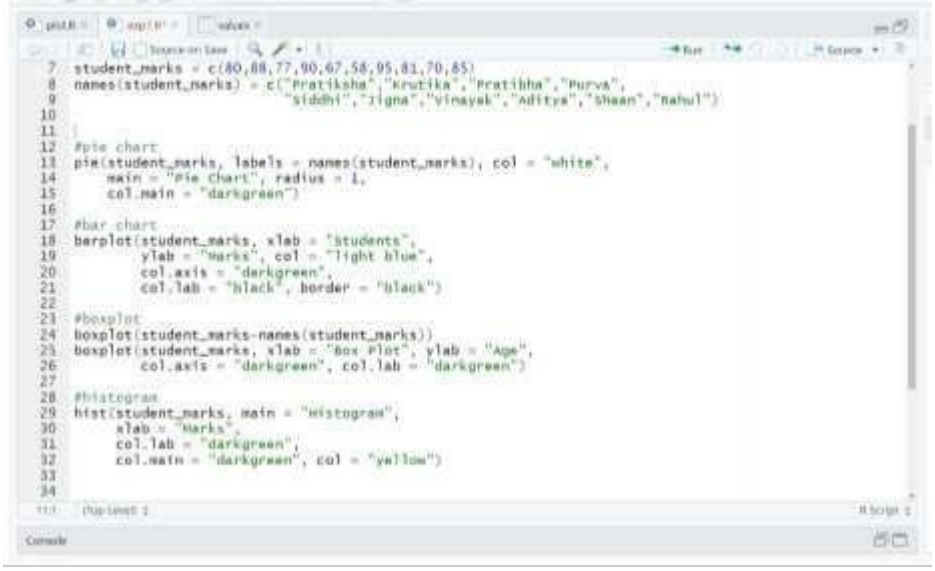
Syntax: plot(x, y, type, xlab, ylab, main)
Where,

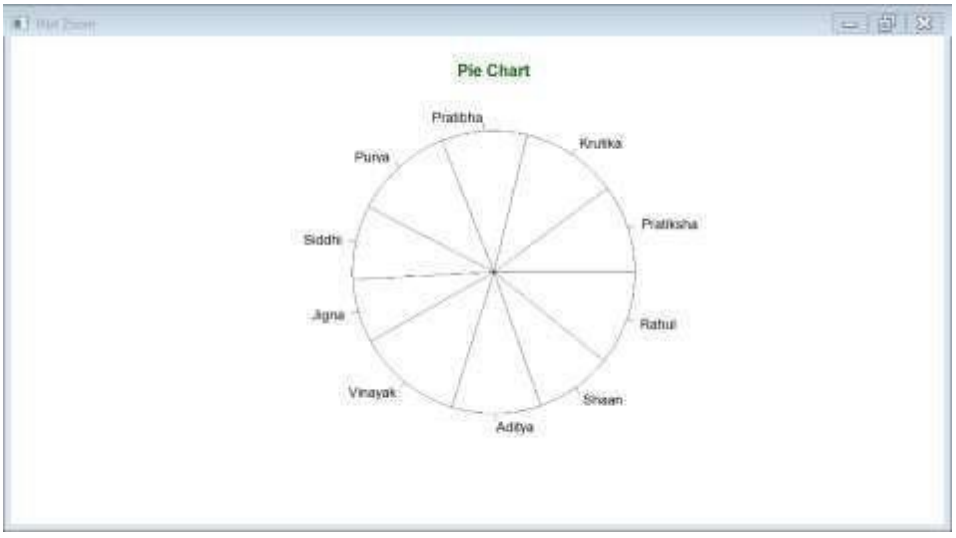
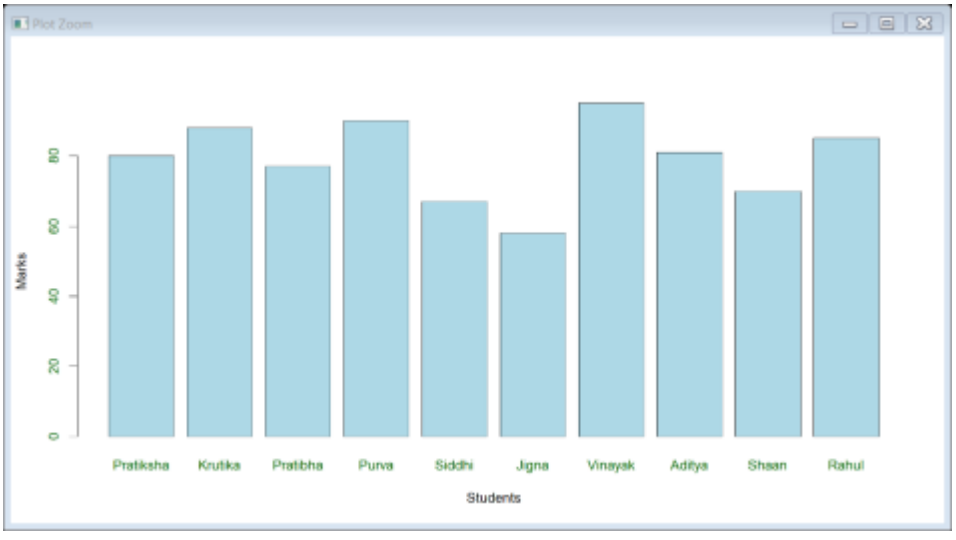
- **x** is the data vector represented on x-axis
- **y** is the data vector represented on y-axis
- **type** specifies the type of plot to be drawn. For example , "l" for lines, "p" for points, "s" for stair steps, etc.
- **xlab** specifies the label for x-axis
- **ylab** specifies the label for y-axis
- **main** specifies the title name of the graph

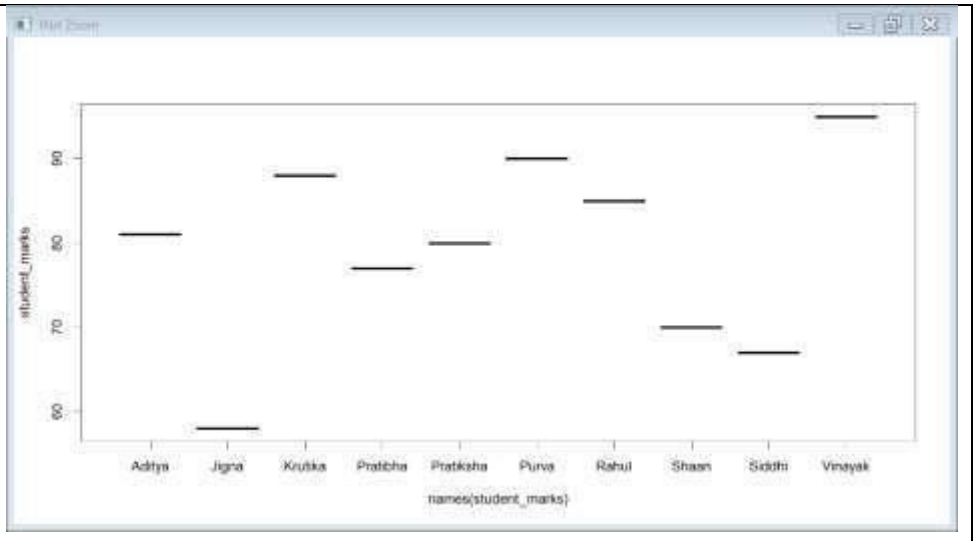
Code:

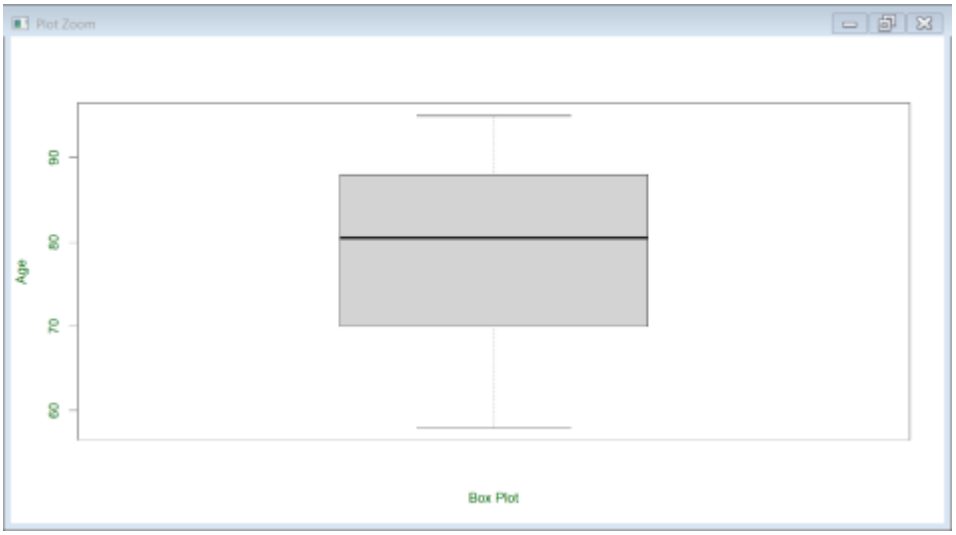
```
student_marks = c(80,88,77,90,67,58,95,81,70,85)
names(student_marks) = c("Pratiksha","Krutika","Pratibha","Purva",
                         "Siddhi","Jigna","Vinayak","Aditya","Shaan","Rahul")

#pie chart
pie(student_marks, labels = names(student_marks), col = "white",
   main = "Pie Chart", radius = 1,
   col.main = "darkgreen")

#bar chart
barplot(student_marks, xlab = "Students",
       ylab = "Marks", col = "light blue",
       col.axis = "darkgreen",
       col.lab = "black", border = "black")

#boxplot
boxplot(student_marks~names(student_marks))
boxplot(student_marks, xlab = "Box Plot", ylab = "Age",
       col.axis = "darkgreen", col.lab = "darkgreen")
```

```
#histogram
hist(student_marks, main = "Histogram",
     xlab = "Marks",
     col.lab = "darkgreen",
     col.main = "darkgreen", col = "yellow")


#line graph
plot(student_marks, type = "o", col = "green", xlab = "Students",
ylab = "Marks",
     main = "Line Graph")


#scatter plot
plot(x = student_marks, y = NULL, xlab = "Students",
     ylab = "Marks", main = "Scatter Plot",
     col.lab = "darkgreen", col.main = "darkgreen",
     col.axis = "darkgreen")
```

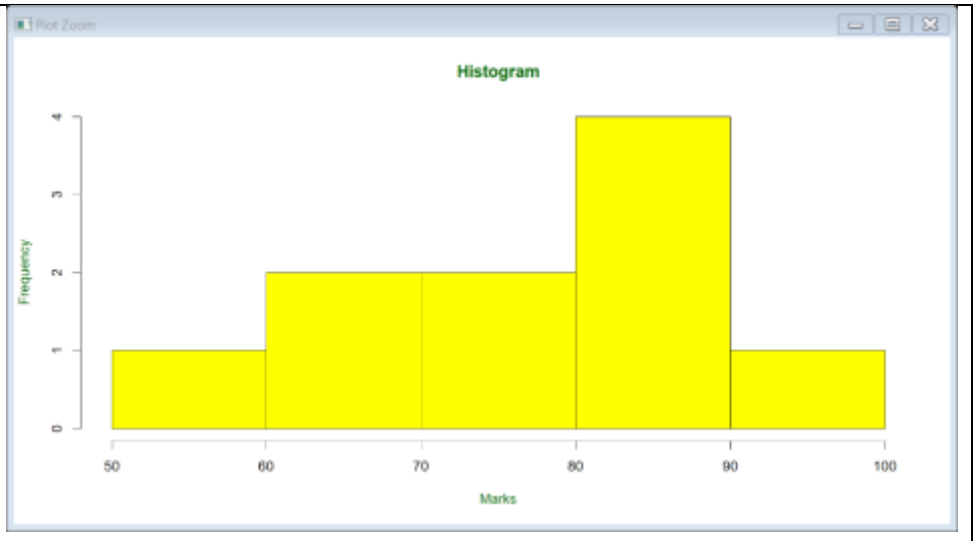


Output:	**Pie chart**  **Bar chart**  **Box plot**

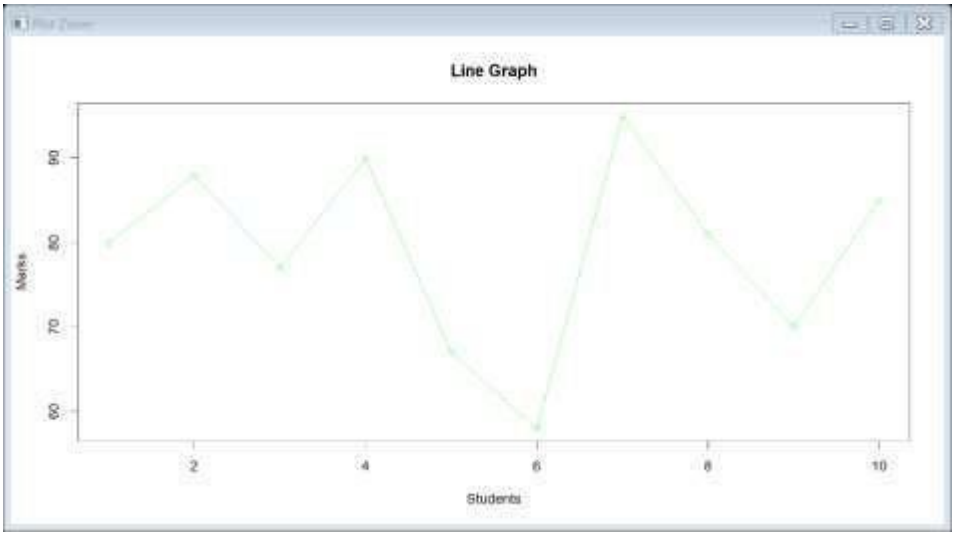

student_marks
90
80
70
60
Aditya
Jigna
Krutika
Prabha
Pratiksha
Purva
Rahul
Shaan
Siddhi
Vinayak
names(student_marks)


Age
90
80
70
60
Box Plot

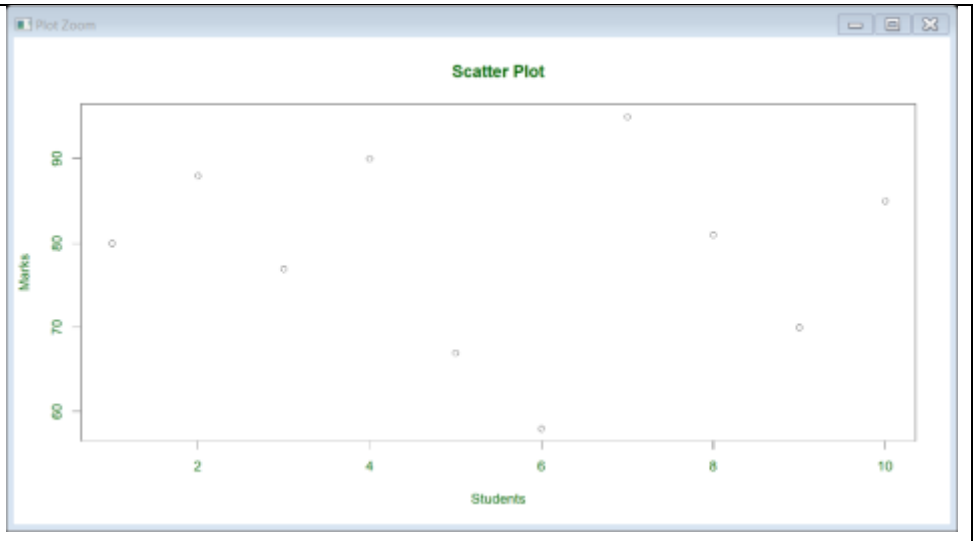
Histogram:



Line chart:



Scatter plot:


Plot Zoom
Scatter Plot
Marks
90
80
70
60
2
4
6
8
10
Students

Department of Information Technology

Semester	B.E. Semester VIII – INFT
Subject	R programming lab
Laboratory Teacher:	Shruti Agrawal
Laboratory	-

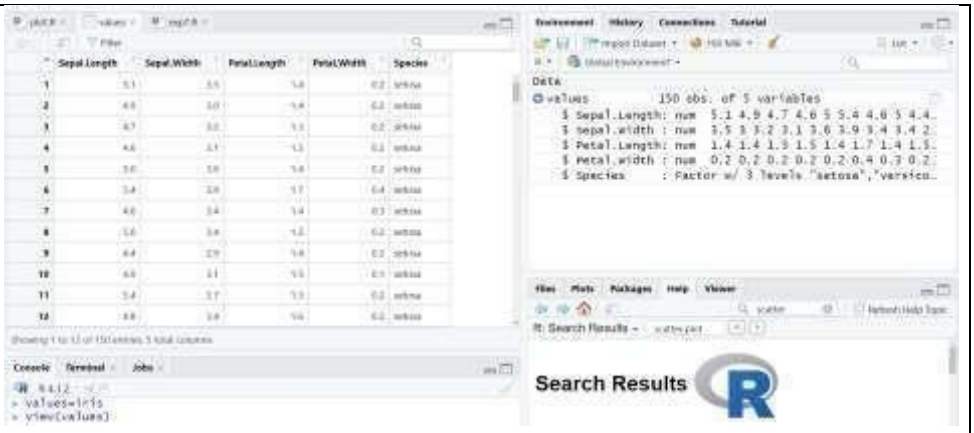
Student Name	Soham Sahare	
Roll Number	18101B0010	
Grade and Subject Teacher's Signature		

Experiment Number	8	
Experiment Title	To understand exploratory data analysis in R	
Problem Statement	Write a R program to import a data set of minimum 100 tuples and perform exploratory data analysis in it. Provide proper screenshot to every function.	
Resources / Apparatus Required	Hardware: Desktop/Laptop	Software: R studio

Code:	values = iris #(to load iris dataset i.e. inbuild)
	View(values) #to see all the data
	head(values) #to see top 6 values of data
	str(values) #to see structure of dataset
	min(values$Sepal.Length)
	max(values$Sepal.Length)
	range(values$Sepal.Length) #data ranging between	

	#or
	max(values$Sepal.Length) - min(values$Sepal.Length)
	mean(values$Sepal.Length)
	median(values$Sepal.Length)
	quantile(values$Sepal.Length)
	#0% 25% 50% 75% 100%
	#4.3 5.1 5.8 6.4 7.9
	quantile(values$Sepal.Length,0.25) #first quartile
	#25%
	#5.1
	quantile(values$Sepal.Length,0.5) #second quartile
	#50%
	#5.8
	sd(values$Sepal.Length) #standard deviation
	var(values$Sepal.Length) #variance
	IQR(values$Sepal.Length)
	summary(values)
	values$Sepal.Width
	
	##Plotting
	vect = c(12,4,10,9,3,8,20,18,24)
	hist(vect)
	hist(vect,col = "blue",border = "Green")
	
	#line graph
	plot(vect) #will show dots
	plot(vect,type = "o") #for line
	
	#mtcars is also a dataset in R
	head(mtcars) #here you'll find mpg and cyl columns
	#Boxplot
	boxplot(mpg~cyl,data=mtcars,xlab="Number of cyl",
	ylab = "Number of mpg", main="mtcars Boxplot"

Output:	**values = iris** **View(values)**



plot.R | values | exp7.R

Filter

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa

Showing 1 to 12 of 150 entries, 5 total columns

Console | Terminal | Jobs

R 4.1.2 · ~/

```
> values=iris
> view(values)
```

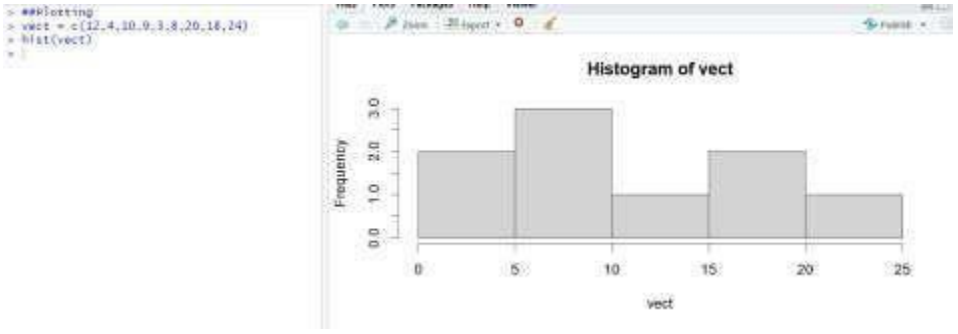
```
> head(values) #to see top 6 values of data
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa
> str(values) #to see structure of dataset
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1
 1 1 1 1 1 1 ...
```

```
> min(values$Sepal.Length)
[1] 4.3
> max(values$Sepal.Length)
[1] 7.9
> range(values$Sepal.Length) #data ranging between
[1] 4.3 7.9
> mean(values$Sepal.Length)
[1] 5.843333
> median(values$Sepal.Length)
[1] 5.8
```

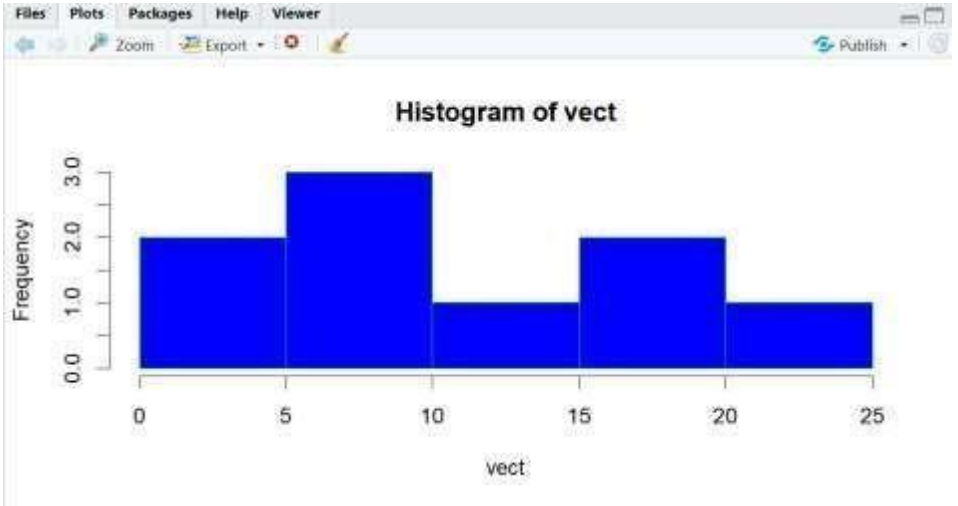
```
> quantile(values$Sepal.Length)
  0%  25%  50%  75% 100%
 4.3  5.1  5.8  6.4  7.9
> #0%  25%  50%  75% 100%
> #4.3  5.1  5.8  6.4  7.9
> quantile(values$Sepal.Length,0.25) #first quartile
25%
5.1
> #25%
> #5.1
> quantile(values$Sepal.Length,0.5) #second quartile
50%
5.8
> #50%
> #5.8
> sd(values$Sepal.Length) #standard deviation
[1] 0.8280661
> var(values$Sepal.Length) #variance
[1] 0.6856935
> IQR(values$Sepal.Length)
[1] 1.3
```

```
> summary(values)
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
 Median :5.800   Median :3.000   Median :4.350   Median :1.300
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
       Species
 setosa    :50
 versicolor:50
 virginica :50
```

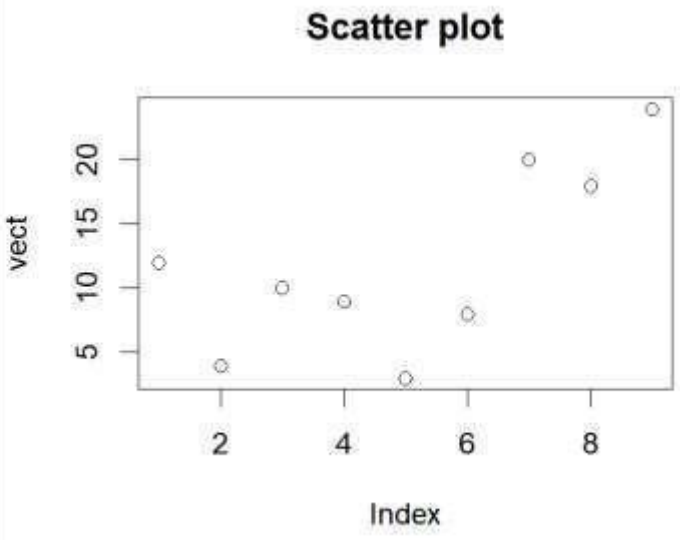
Plotting



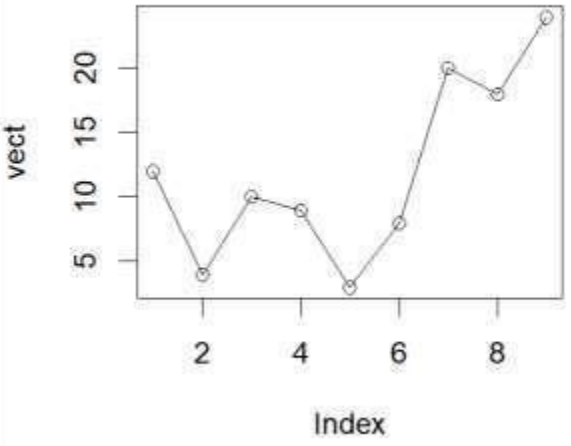
hist(vect,col = "blue",border = "Green")



```
Console
R 4.1.2 · ~/
> #line graph
> plot(vect,
 main = "Scat
ter plot") #w
ill show dots
>
```




```
Console  Terminal
R 4.1.2 · ~/
> #line graph
> plot(vect) #will s
how dots
> plot(vect,type =
 "o") #for line
>
```



```
> #mtcars is also a dataset in R
> head(mtcars) #here you'll find mpg and cyl columns
                   mpg cyl disp
Mazda RX4         21.0   6  160
Mazda RX4 Wag     21.0   6  160
Datsun 710        22.8   4  108
Hornet 4 Drive    21.4   6  258
Hornet Sportabout 18.7   8  360
Valiant           18.1   6  225
                   hp drat
Mazda RX4         110 3.90
Mazda RX4 Wag     110 3.90
Datsun 710         93 3.85
Hornet 4 Drive    110 3.08
Hornet Sportabout 175 3.15
Valiant           105 2.76
                     wt  qsec
Mazda RX4         2.620 16.46
Mazda RX4 Wag     2.875 17.02
Datsun 710        2.320 18.61
Hornet 4 Drive    3.215 19.44
Hornet Sportabout 3.440 17.02
Valiant           3.460 20.22
                  vs am gear
Mazda RX4          0  1    4
```

```
> #Boxplot
> boxplot(mpg~cyl,data=mtcars,xlab="Number of cyl",
+         ylab = "Number of mpg", main="mtcars Boxplot")
> |
```