FACULTY OF SCIENCE AND HUMANITIES

Tiruchirappalli, Tamil Nadu-621105.

Department of Computer Applications

PRACTICAL RECORD

NAME :

REGISTER NO : RA23322410500

COURSE : MCA

SEMESTER / YEAR : II / I

SUBJECT CODE : PCA20S02J

SUBJECT NAME : DATA ANALYSIS USING R

APRIL 2024

FACULTY OF SCIENCE AND HUMANITIES

Tiruchirappalli, Tamil Nadu-621105.

Department of Computer Applications

BONAFIDE CERTIFICATE

This is to certify that the bonafide work is done by Mr. / Ms. _____

Register Number._____in Data Analysis Using R (Subject Code: PCA20S02J)

at Computer Lab, SRM Institute of Science & Technology, Tiruchirappalli Campus, in April 2024.

STAFF IN-CHARGE **HEAD OF THE DEPARTMENT**

Submitted for the University Practical Examination held at Department of Computer Applications, SRM Institute of Science & Technology, Tiruchirappalli on_____.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

INDEX

S.NO	DATE	NAME OF THE PROGRAMS	PAGE.NO	SIGNATURE
1	04.01.2024	Implementation of R program-basic		
2	08.01.2024	Implementation of Data Types in R		
3	11.01.2024	Implementation of Data Structures in R		
4	18.01.2024	Implementation of Control Statements in R		
5	29.01.2024	Implementation of Looping Statements in R		
6	05.02.2024	Implementation of Decision Tree Classification Algorithm		
7	12.02.2024	Implementation of Naïve Bayes Classification Algorithms		
8	15.02.2024	Implementation of K-NN Classification Algorithm		
9	19.02.2024	Implementation of K-means Clustering Algorithm		
10	22.02.2024	Implementation of K-medoids Clustering Algorithm		
11	07.03.2024	Implementation of Data Visualization in R-Bar plot and Histogram		
12	18.03.2024	Implementation of Data Visualization in R-Box plot and Scatter plot		

EX.NO: 1

IMPLEMENTATION OF R PROGRAM-BASIC

AIM:

To implement the basic R function like printing the statements and using comments in the part of program.

SOURCE CODE:

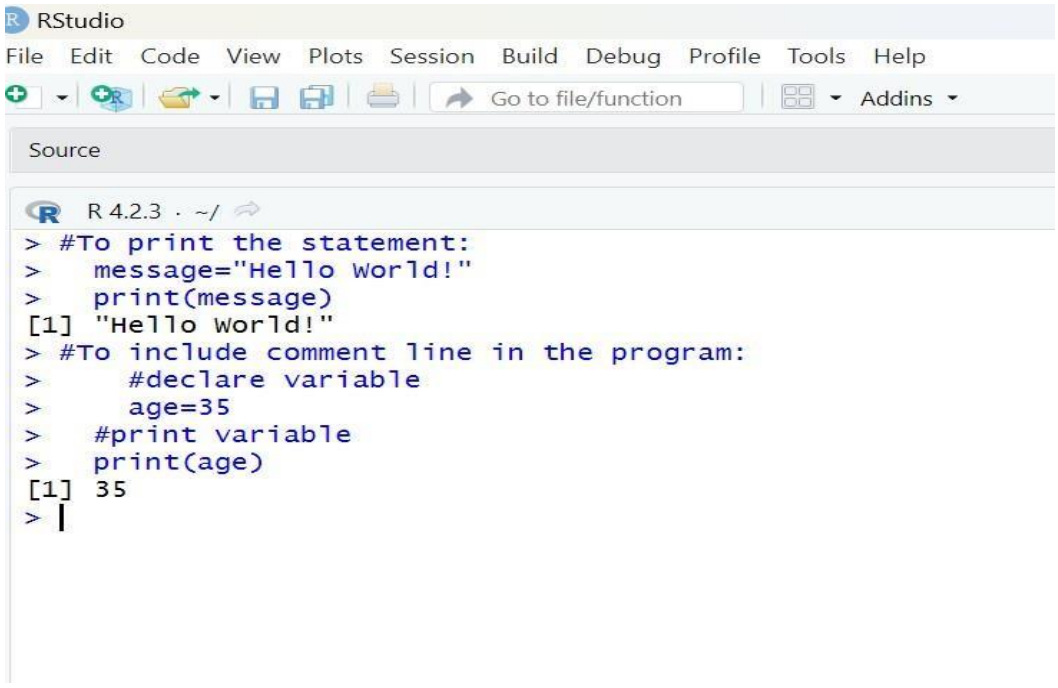
(i) To print the statement:

```
message="Hello World!"
print(message)
```

(ii) To include comment line in the program:

```
#declare variable
age=35
#print variable
print(age)
```

OUTPUT:



Result:

Thus, the basic R functions have been implemented successfully

EX.NO: 2

IMPLEMENTATION OF DATA TYPES IN R

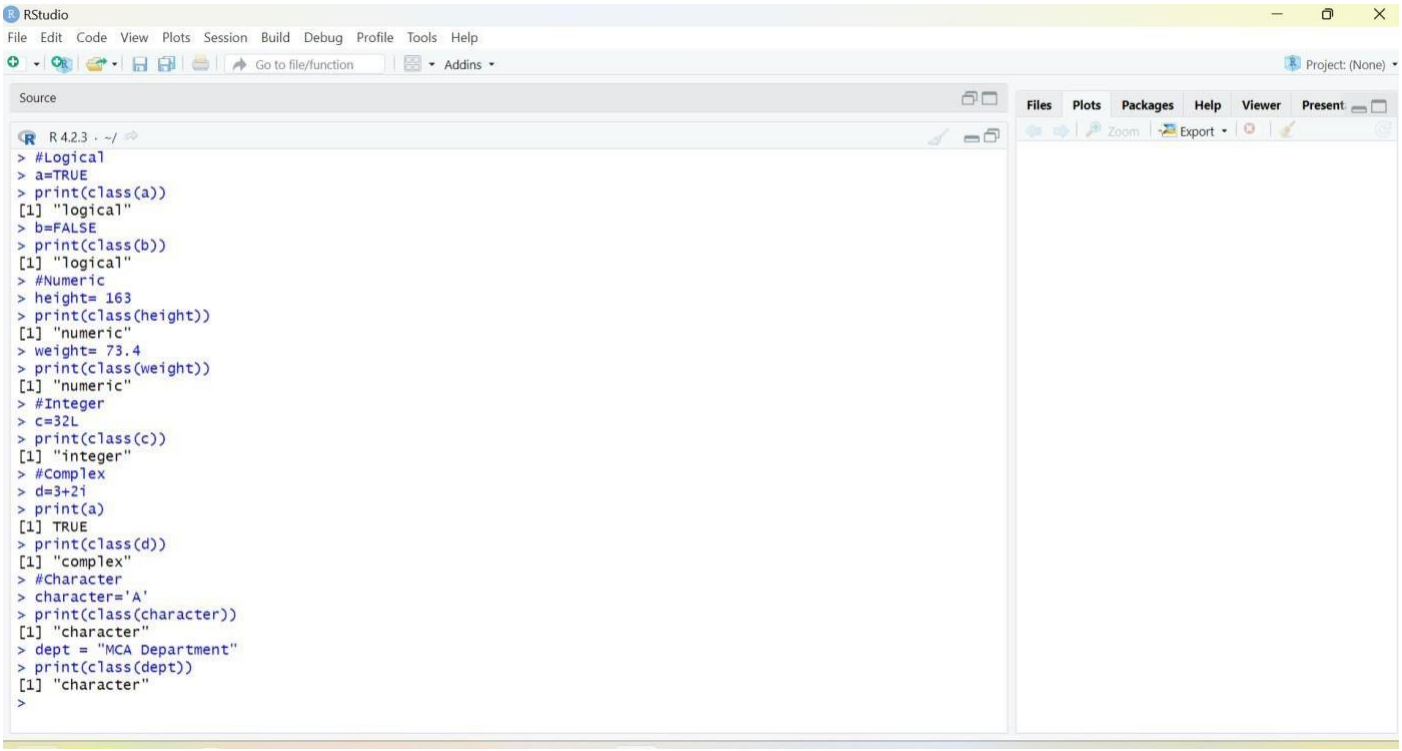
AIM:

To implement the basic data types in R programming.

SOURCE CODE:

```
#Logical
        a=TRUE
        print(class(a))
        b=FALSE
        print(class(b))
#Numeric height= 163
        rint(class(height))
        weight= 73.4
        print(class(weigh
        t))
#Integer
        c=32L
        print(class(c))
#Complex d=3+2i
        print(a)
        print(class(
        d))
#Character character='A'
        print(class(character))
        dept = "MCA
        Department"
        print(class(dept))
```

OUTPUT:



RESULT:

Thus, the basic data types in R programming have been successfully implemented.

EX.NO: 3

IMPLEMENTATION OF DATA STRUCTURES IN R

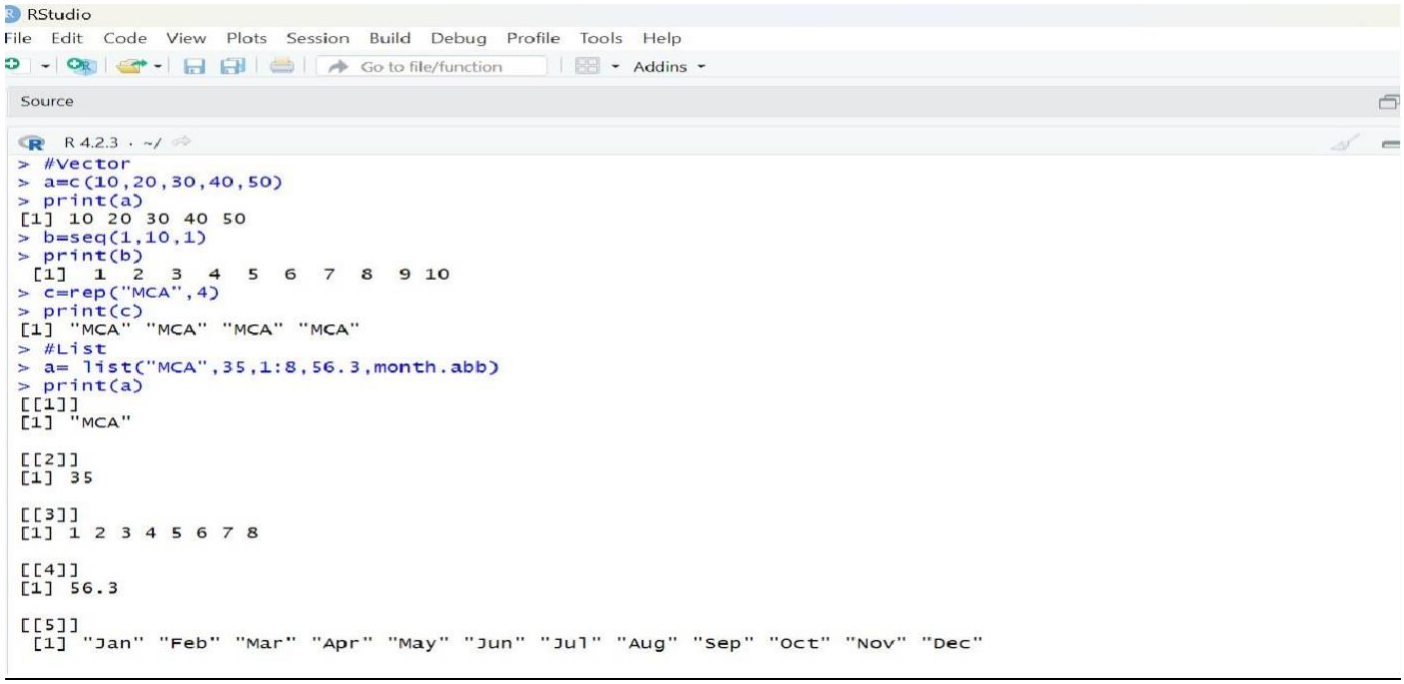
AIM:

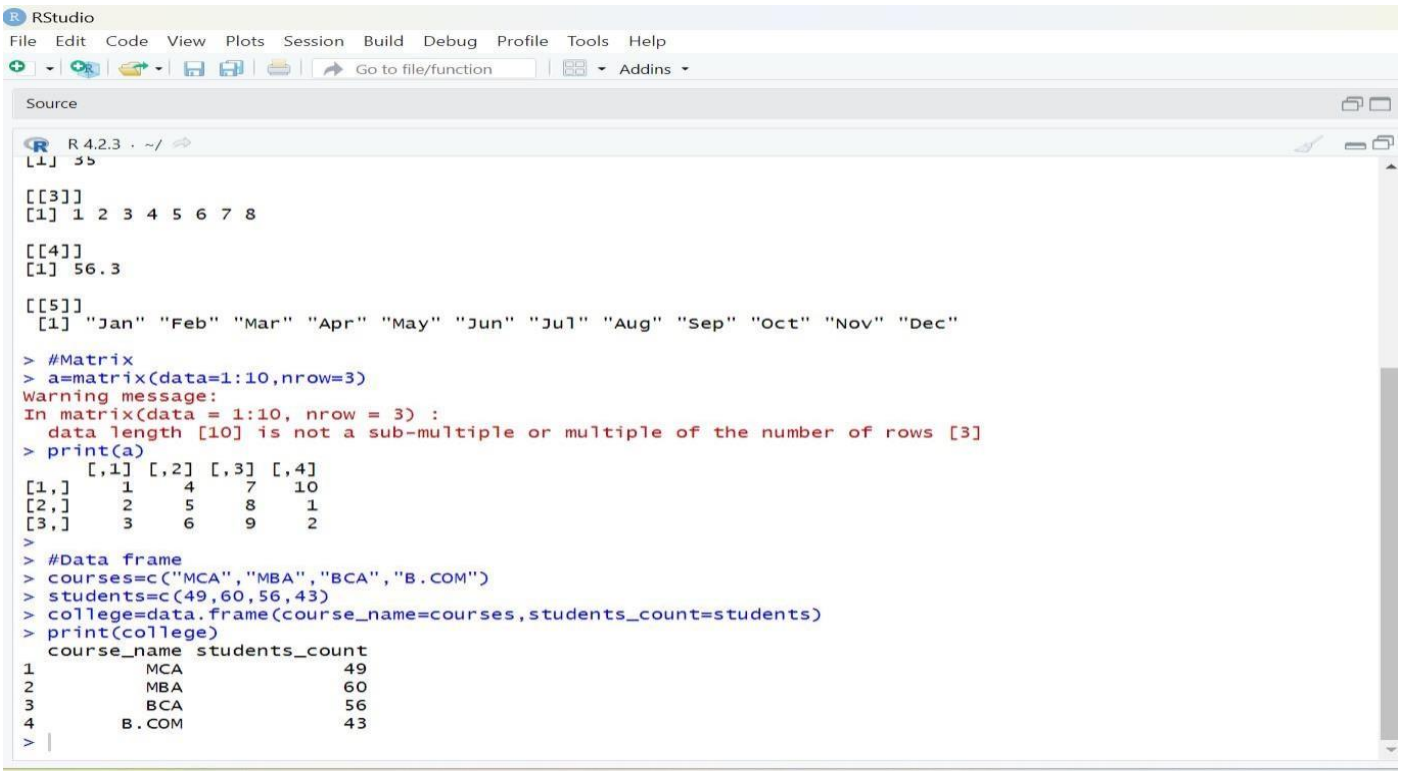
To implement the data structures in R programming.

SOURCE CODE:

```
#Vector
    a=c(10,20,30,40,50
    ) print(a)
    b=seq(1,10,1)
    print(b)
    c=rep("MCA",4)
    print(c)
#List a=
    list("MCA",35,1:8,56.3,month.abb)
    print(a)
#Matrix
    a=matrix(data=1:10,nrow=3)
    print(a)
#Data frame
    courses=c("MCA","MBA","BCA","B.COM")
    students=c(49,60,56,43)
    college=data.frame(course_name=courses,students_count=students)
    print(college)
```

OUTPUT:





RESULT:

Thus, the data structures in R programming have been successfully implemented.

EX.NO: 4

IMPLEMENTATION OF CONTROL STATEMENTS IN R

AIM:

To implement the control statements in R.

SOURCE CODE:

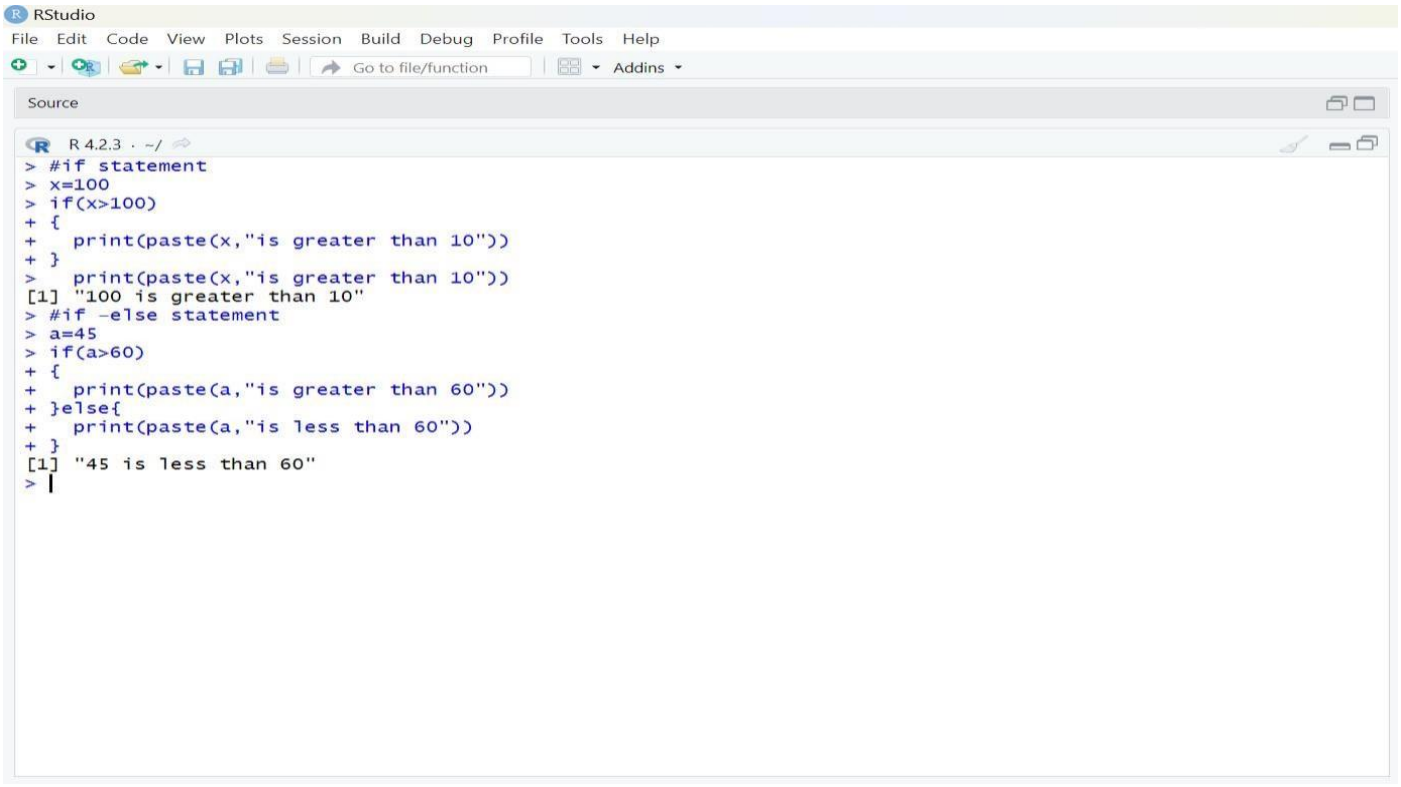
#if statement

```
x=100
if(x>100)
{
print(paste (x,"is greater than 10"))
}
```

#if –else statement

```
a=45
if(a>60)
{
 print(paste(a,"is greater than 60"))
}else{
print(paste(a,"is less than 60"))
}
```

OUTPUT:



RESULT:

Thus, the data structures in R programming have been successfully implemented.

EX.NO: 5

IMPLEMENTATION OF LOOPING STATEMENTS IN R

AIM: To implement the looping statements in R.

SOURCE CODE:

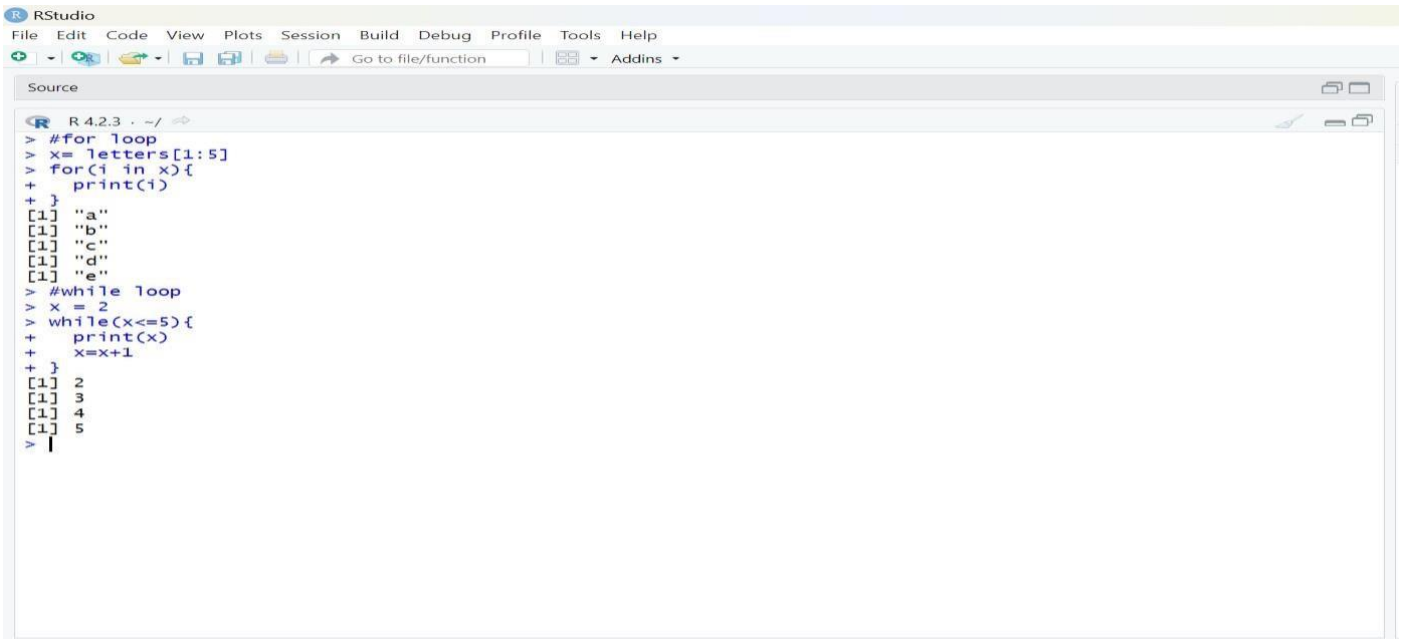
#for loop

```
x= letters[1:5]
for(i in x){
print(i)
}
```

#while loop

```
x = 2
while(x<=5)
{ print(x)
x=x+1
}
```

OUTPUT:-



RESULT:

Thus, the looping statements in R programming have been successfully implemented.

EX.NO: 6

IMPLEMENTATION OF DECISION TREE CLASSIFICATION ALGORITHM

AIM:

To classify the new instance (new sample) to a target data (grape fruit or orange) using decision tree algorithm in R Studio.

PROCEDURE:

Step 1: Install the necessary package in R Studio such as "rpart" and "rpart.plot"

Step 2: Set the current working directory

Step 3: Save the dataset in the current working directory

Step 4: Read the dataset in the form of csv file using **read.csv ()**

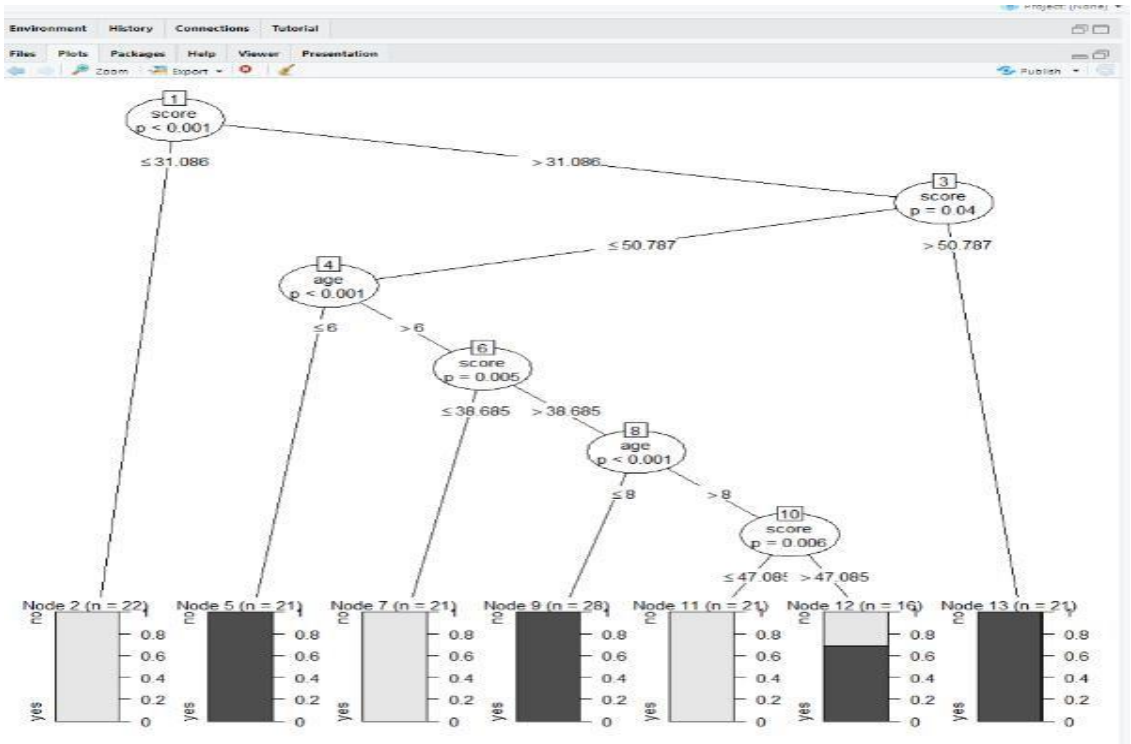
Step 5: Finally create the decision tree and predict the category for the new instance.

SOURCE CODE:

```
library(datasets)
library(caTools)
library(party)
library(dplyr)
library(magrittr)
data("readingSkills")
head(readingSkills)
sample_data = sample.split(readingSkills, SplitRatio = 0.8)
train_data <- subset(readingSkills, sample_data == TRUE)
test_data <- subset(readingSkills, sample_data == FALSE)
model<- ctree(nativeSpeaker ~ ., train_data)
plot(model)
ctree(formula, data)
```

OUTPUT:

```
Warning in install.packages :
  package 'closure' is not available for this version of R

A version of this package for your version of R might be available elsewhere,
see the ideas at
https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages
> library(datasets)
> library(caTools)
> library(party)
> library(dplyr)
> library(magrittr)
>
> data("readingSkills")
> head(readingSkills)
  nativeSpeaker age shoeSize    score
1           yes   5 24.83189 32.29385
2           yes   6 25.95238 36.63105
3            no  11 30.42170 49.60593
4           yes   7 28.66450 40.28456
5           yes  11 31.88207 55.46085
6           yes  10 30.07843 52.83124
> sample_data = sample.split(readingSkills, SplitRatio = 0.8)
> train_data <- subset(readingSkills, sample_data == TRUE)
> test_data <- subset(readingSkills, sample_data == FALSE)
> model<- ctree(nativeSpeaker ~ ., train_data)
> plot(model)
> ctree(formula, data)
Error in x$terms : object of type 'closure' is not subsettable
```



RESULT:

Thus the new instance has been classified to a target attribute (grape fruit) using Decision Tree algorithm.

EX.NO: 7

IMPLEMENTATION OF NAÏVE BAYES CLASSIFICATION ALGORITHM

AIM:

To classify the new instance (new sample) to a target data using Naïve Bayes classification algorithm in R Studio.

PROCEDURE:

Step 1: Install the necessary package in R Studio such as "e1071"

Step 2: Set the current working directory

Step 3: Save the dataset in the current working directory

Step 4: Read the dataset in the form of csv file using **read.csv().**

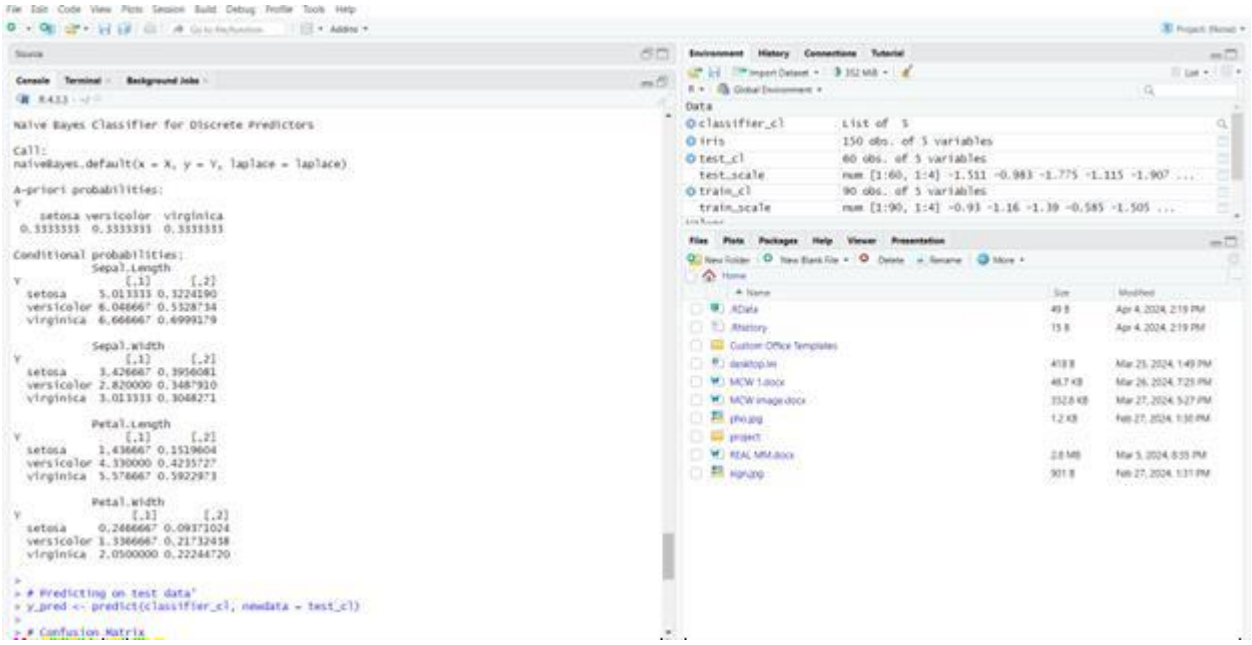
Step 5: Finally predict the category for the new instance using Naïve Bayes classifier.

SOURCE CODE:

```
# Loading package
library(e1071)
library(caTools)
library(caret)
# Splitting data into train
# and test data
split <- sample.split(iris, SplitRatio = 0.7)
train_cl <- subset(iris, split == "TRUE")
test_cl <- subset(iris, split == "FALSE")
# Feature Scaling
train_scale <- scale(train_cl[, 1:4])
test_scale <- scale(test_cl[, 1:4])

# Fitting Naive Bayes Model
# to training dataset
```

```
set.seed(120) # Setting Seed

classifier_cl <- naiveBayes(Species ~ ., data = train_cl)

classifier_cl

# Predicting on test data'

y_pred <- predict(classifier_cl, newdata = test_cl)

# Confusion Matrix

cm <- table(test_cl$Species, y_pred)

cm

# Model Evaluation

confusionMatrix(cm)
```

OUTPUT:



RESULT:

Thus, the new instance has been classified to a target attribute using Naïve Bayes classification algorithm.

EX.NO: 8

IMPLEMENTATION OF K-NN CLASSIFICATION ALGORITHM

AIM:

To classify the test data to a target attribute based on the model built using the training data with the help K-NN classification algorithm.

PROCEDURE:

Step 1: Install the necessary package in R Studio such as "class"

Step 2: Set the current working directory

Step 3: Save the dataset in the current working directory

Step 4: Read the dataset in the form of csv file using **read.csv ()**

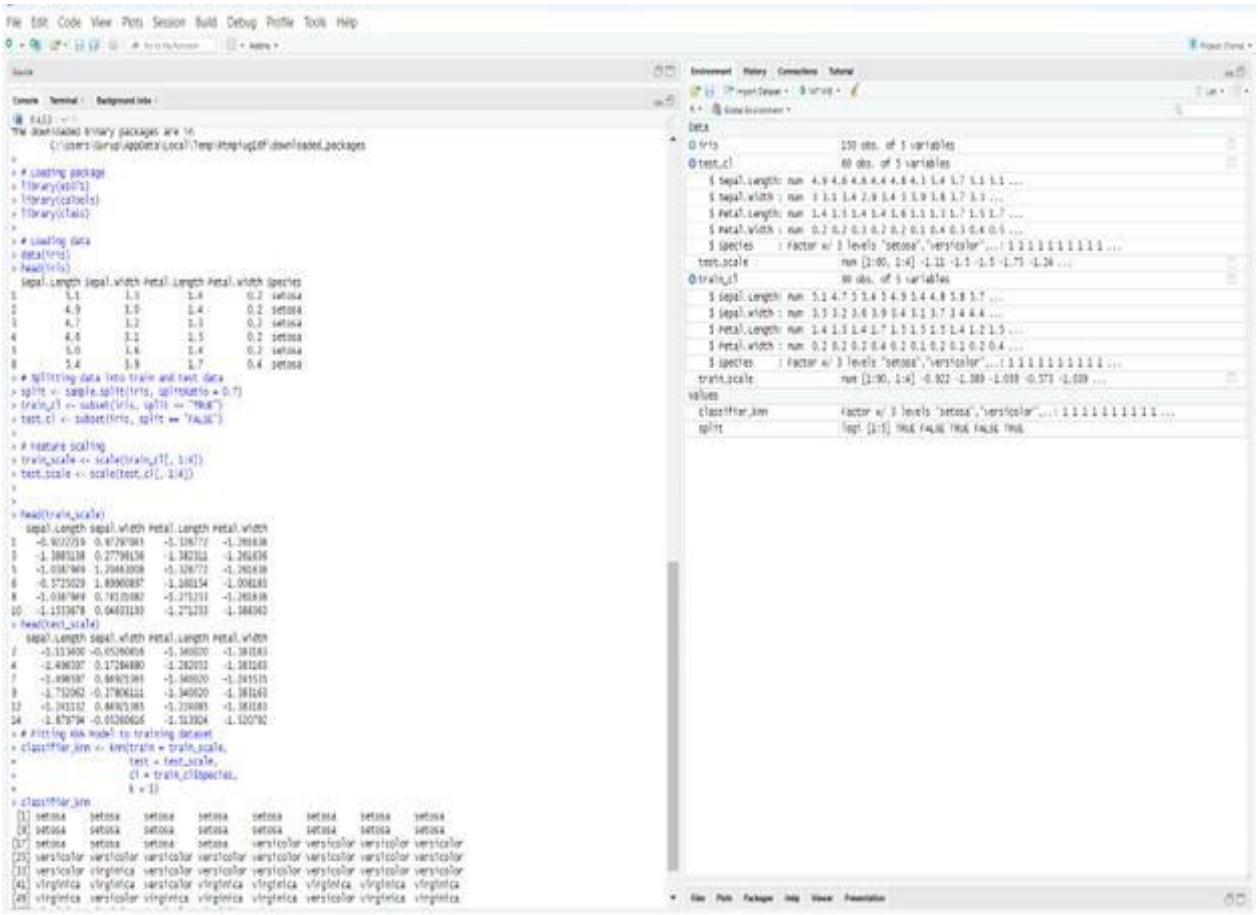
Step 5: Finally create the model using kNN algorithm with the help of training data.

SOURCE CODE:

```
# Loading data

data(iris)

# Structure

str(iris)

# Installing Packages

install. Packages("e1071")

install. packages("caTools")

install.packages("class")

# Loading package

library(e1071)

library(caTools)

library(class)
```

```
# Loading data
data(iris)
head(iris)
# Splitting data into train and test data
split <- sample. split(iris, SplitRatio = 0.7)
train_cl <- subset(iris, split == "TRUE")
test_cl <- subset(iris, split == "FALSE")
# Feature Scaling
train_scale <- scale(train_cl[, 1:4])
test_scale <- scale(test_cl[, 1:4])
head(train_scale)
head(test_scale)
# Fitting KNN Model to training dataset
classifier_knn <- knn(train = train_scale,
                      test = test_scale,
                      cl = train_cl$Species,
                      k = 1)
classifier_knn
```

OUTPUT:



RESULT:

Thus, the test data are classified successfully based on the model built using K-NN classification algorithm.

EX.NO: 9

IMPLEMENTATION OF K-MEANS CLUSTERING ALGORITHM

AIM: To group the IRIS (flower) data points into two different clusters with the help k-means clustering algorithm.

PROCEDURE:

Step1: Install the necessary packages in R Studio such as "stats","dplyr", "ggplot2", "ggfortify"

Step 2: Set the current working directory

Step 3: Save the IRIS dataset in the current working directory

Step 4: Read the dataset in the form of csv file using **read.csv ()**

Step 5: Finally create the model (k=2) with the help of k-means clustering algorithm

SOURCE CODE:

```
# Loading data
data(iris)
# Structure
str(iris)
# Installing Packages
install.packages("ClusterR")
install.  packages("cluster")
# Loading package
library(ClusterR)
library(cluster)
# Removing initial label of
# Species from original dataset
iris_1 <- iris[, -5]
# Fitting K-Means clustering Model
# to training dataset
set.seed(240) # Setting seed
```

```
kmeans.re <- kmeans(iris_1, centers = 3, nstart = 20)
kmeans.re
# Cluster identification for
# each observation
kmeans.re$cluster
# Confusion Matrix
cm <- table(iris$Species, kmeans.re$cluster)
cm
# Model Evaluation and visualization
plot (iris_1[c("Sepal.Length", "Sepal.Width")])
plot (iris_1[c("Sepal.Length", "Sepal.Width")],
col = kmeans.re$cluster)
plot(iris_1[c("Sepal.Length", "Sepal.Width")],
    col = kmeans.re$cluster,
    main = "K-means with 3 clusters")
## Plotiing cluster centers
kmeans.re$centers
kmeans.re$centers[, c("Sepal.Length", "Sepal.Width")]
# cex is font size, pch is symbol
points(kmeans.re$centers[, c("Sepal.Length", "Sepal.Width")],
      col = 1:3, pch = 8, cex = 3)
## Visualizing clusters
y_kmeans <- kmeans.re$cluster
clusplot(iris_1[, c("Sepal.Length", "Sepal.Width")],
        y_kmeans,
        lines =  0,
        shade = TRUE,
        color = TRUE,
```

```
labels =  2,
plotchar = FALSE,
span = TRUE,
main = paste("Cluster iris"),
xlab = 'Sepal.Length',
ylab = 'Sepal.Width')
```

OUTPUT:

RESULT:

Thus, the IRIS data points are successfully grouped into two different clusters with the help of kmeans clustering algorithm.

EX.NO: 10

IMPLEMENTATION OF K-MEDOIDS CLUSTERING ALGORITHM

AIM:

To group the customers of a bank based on their credit card details into three different clusters with the help k-medoids clustering algorithm.

PROCEDURE:

Step1: Install the necessary packages in R Studio such as “cluster” and “factoextra”

Step 2: Set the current working directory

Step 3: Save the Credit Card details dataset in the current working directory

Step 4: Read the dataset in the form of csv file using **read.csv ()**

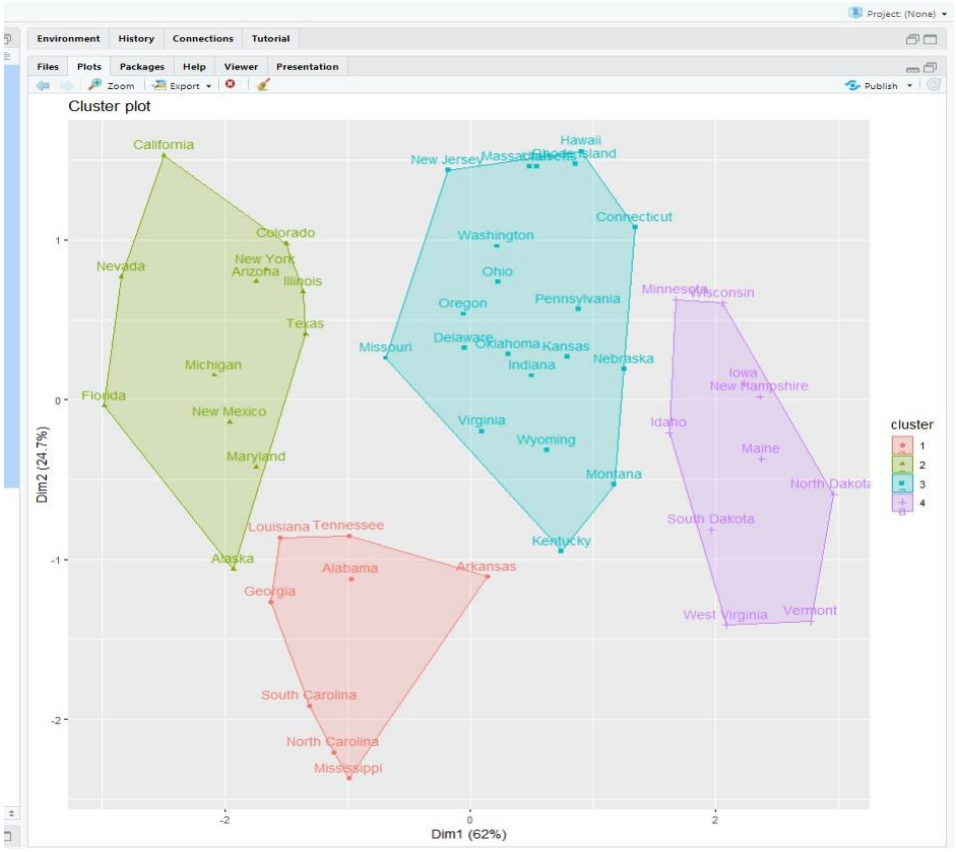
Step 5: Finally create the model (k=3) with the help of k-medoids clustering algorithm

SOURCE CODE:

```
install.packages("factoextra")

library(factoextra)

#load data

df <- USArrests

#remove rows with missing values

df <- na.omit(df)

#scale each variable to have a mean of 0 and sd of 1

df <- scale(df)

#view first six rows of dataset

head(df)

fviz_nbclust(df, pam, method = "wss")

#calculate gap statistic based on number of clusters

gap_stat <- clusGap(df,

                    FUN = pam,

                    K.max = 10, #max clusters to consider B = 50) #total bootstrapped iterations

#plot number of clusters vs. gap statistic
```

```
fviz_gap_stat(gap_stat)

#make this example reproducible

set.seed(1)

#perform k-medoids clustering with k = 4 clusters

kmed <- pam(df, k = 4)

#view results

kmed

#plot results of final k-medoids model

fviz_cluster(kmed, data = df)
```

OUTPUT:



RESULT:

Thus, the three different clusters are created with the helpof k-medoids clustering algorithm.

EX.NO: 11

IMPLEMENTATION OF DATA VISUALIZATION IN R- BAR PLOT AND HISTOGRAM

AIM:

To implement the Data Visualization (Bar plot and Histogram) using R Studio.

PROCEDURE:

Step1: Install the necessary packages in R Studio such as "vcd"

Step 2: Set the current working directory

Step 3: Save the dataset in the form of csv file in the current working directory

Step 4: Read the csv file using **read.csv ()**

Step 5: Finally create the bar plot and histogram for the dataset.

SOURCE CODE:

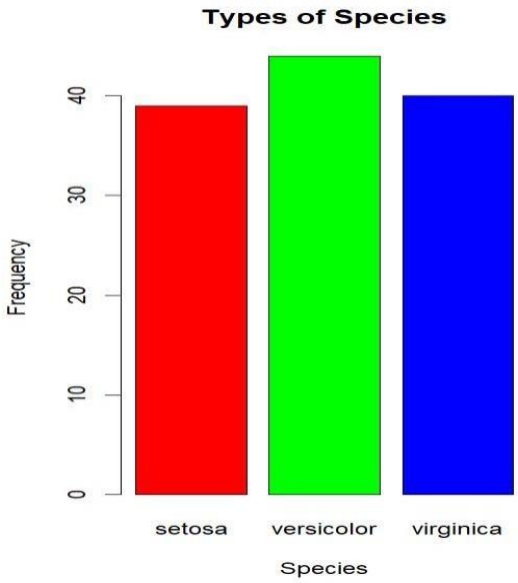
(i) Bar Plot:

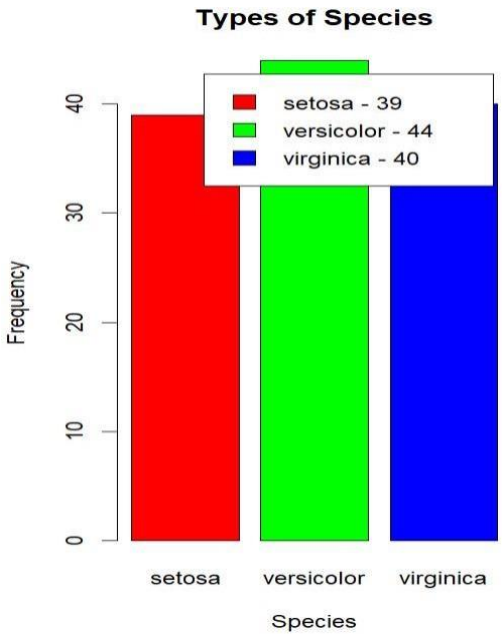
```
library(vcd)
setwd("E:\\SRMIST\\DA using R\\knn")
data=read.csv("iris.csv") View(data)
plot=table(data$Species)
barplot(plot,col=rainbow(length(plot)),xlab="Species",ylab="Frequency",main="Types of Species")
a=paste(rownames(plot),"-",plot)
barplot(plot,col=rainbow(length(plot)),xlab="Species",ylab="Frequency",main="Types of Species",legend.text = a)
```

(ii) Histogram:

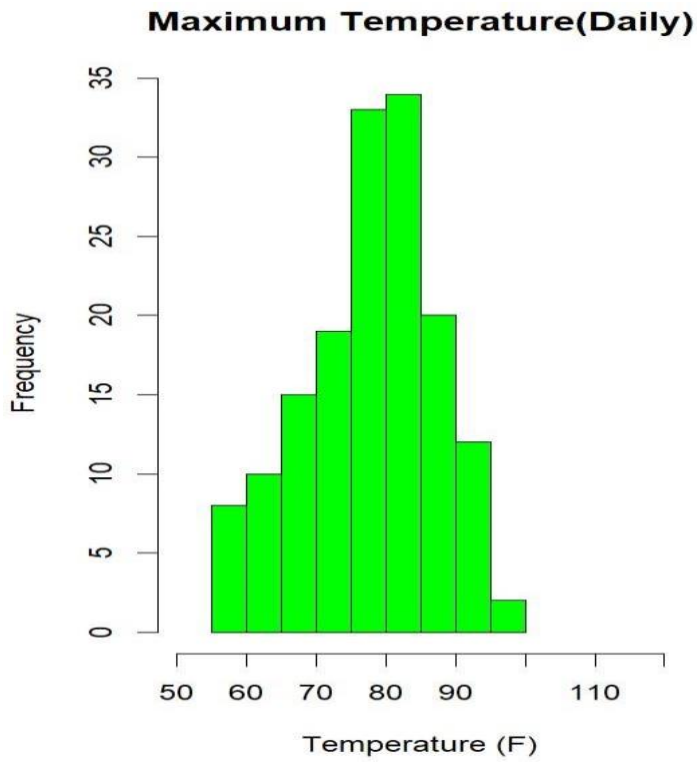
```
library(vcd)
setwd("E:\\SRMIST\\DA using R\\knn")
data=read.csv("airquality.csv") View(data)
hist(airquality$Temp,main="Maximum Temperature(Daily)",xlab="Temperature (F)",xlim=c(50,120),col="green")
```

(i) Bar Plot:





(ii) Histogram:



RESULT:

Thus, the Bar Plot and Histogram have been successfully created in R Studio.

EX.NO: 12

IMPLEMENTATION OF DATA VISUALIZATION IN R- BOX PLOT AND SCATTER PLOT

AIM:

To implement the Data Visualization (Box plot and Scatter Plot) using R Studio.

PROCEDURE:

Step1: Install the necessary packages in R Studio such as “vcd”

Step 2: Set the current working directory

Step 3: Save the dataset in the form of csv file in the current working directory

Step 4: Read the csv file using **read.csv ()**

Step 5: Finally create the bar plot and histogram for the dataset.

SOURCE CODE:

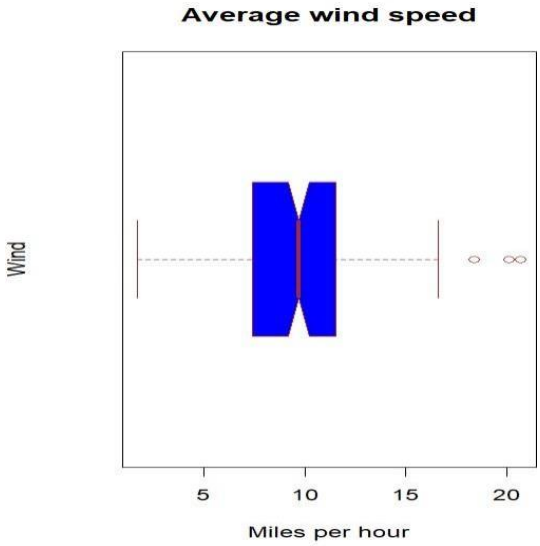
(i) Box Plot:

```
library(vcd)
library(ggplot2)
setwd("E:\\SRMIST\\DA using R\\knn")
data=read.csv("airquality.csv")
boxplot(airquality$Wind, main = "Average wind speed",
      xlab = "Miles per hour", ylab = "Wind",
col = "orange", border = "brown",
horizontal = TRUE, notch = TRUE)
```

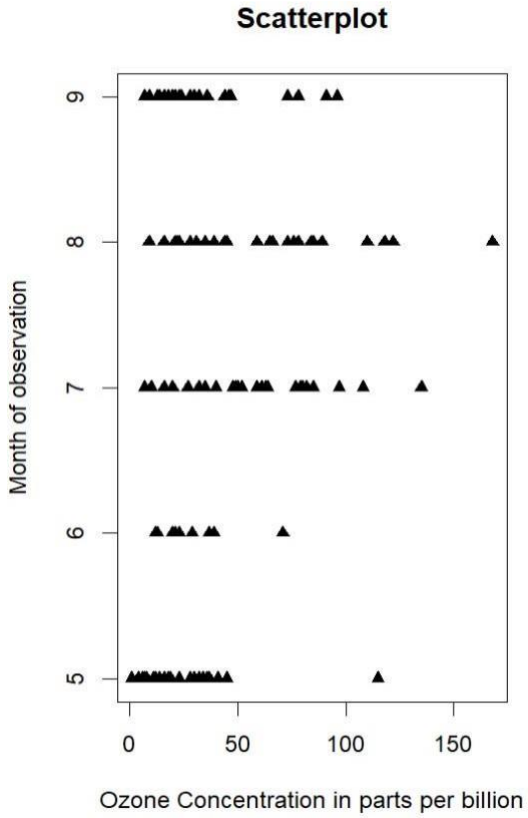
(ii) Scatter Plot:

```
library(vcd) library(ggplot2)
setwd("E:\\SRMIST\\DA using R\\knn")
data=read.csv("airquality.csv") plot(airquality$Ozone,
airquality$Month,
   main ="Scatterplot",
   xlab ="Ozone Concentration in parts per billion",
ylab =" Month of observation ", pch = 17)
```

(i) Box Plot:



(ii) Scatter Plot:



RESULT:

Thus, the Box Plot and Scatter Plot have been successfully created in R Studio.