DATA SCIENCE

VidyaVikas Education Society's

VIKAS COLLEGE OF ARTS, SCIENCE & COMMERCE

Affliated to University of Mumbai

RE-ACCREDITED 'A' GRADE BY NAAC

ISO 9001 : 2008 CERTIFIED

Vikas High School Marg, Kannamwar Nagar No 2, Vikhroli (E), Mumbai – 400083

Dr. R. K. Patra
Principal

Hon' ble: **Shri P. M. Raut**
Chairman. V. V. Edu. Society

This is to certify that,

____

Student of T.Y.B.Sc. (Computer Science) (Sem-VI) with college enrolled Roll no.__________________has satisfactorily completed the practical work for the Subject Data Science in the program of Computer Science from the UNIVERSITY OF MUMBAI for the academic year 2022-2023.

Guided By

____

____

Head Of Department

____

Internal Examiner

____

External Examiner

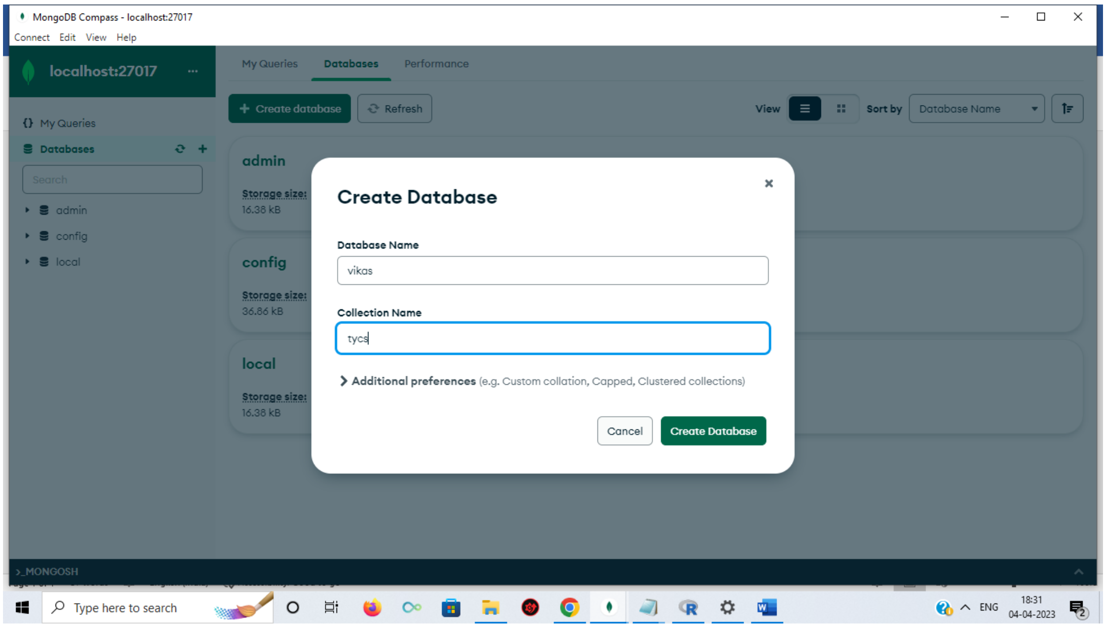
Index

Sr. No.	Aim	Sign
1	Practical of Data collection, Data curation and management for Unstructured (NoSQL)	
2	Data Curation and Management using MongoDB and R.	
3	To perform practical of Principal Component Analysis (PCA).	
4	To perform practical of Clustering.	
5	To perform practical of Time-series forecasting.	
6	To perform practical of Simple/Multiple Linear Regression	
7	To perform practical of Logistics Regression	
8	To perform practical of Hypothesis testing	
9	To perform practical of Analysis of Variance	
10	To perform practical of Decision Tree	

Practical No 1

<u>Aim</u>: Practical of Data collection, Data curation and management for Unstructured data (NoSQL)

Step 1: Create database and collection in mongodb 4.0 by using MongoDB Compass.



Step 2: Run the following code in R or Rstudio

```
install.packages('mongolite')

# Load the mongolite package
library(mongolite)

# Connect to the MongoDB database vikas
conn <- mongo(collection = "tycs", url ="mongodb://localhost:27017/vikas")

# Insert the document into the collection
doc <- '{"name": "Aniket", "age": 20}'
conn$insert(doc)

# Insert the document into the collection
doc <- '{"name": "Mayur", "age": 20}'
conn$insert(doc)

# Insert the document into the collection
doc <- '{"name": "Kirti", "age": 20}'
conn$insert(doc)
```

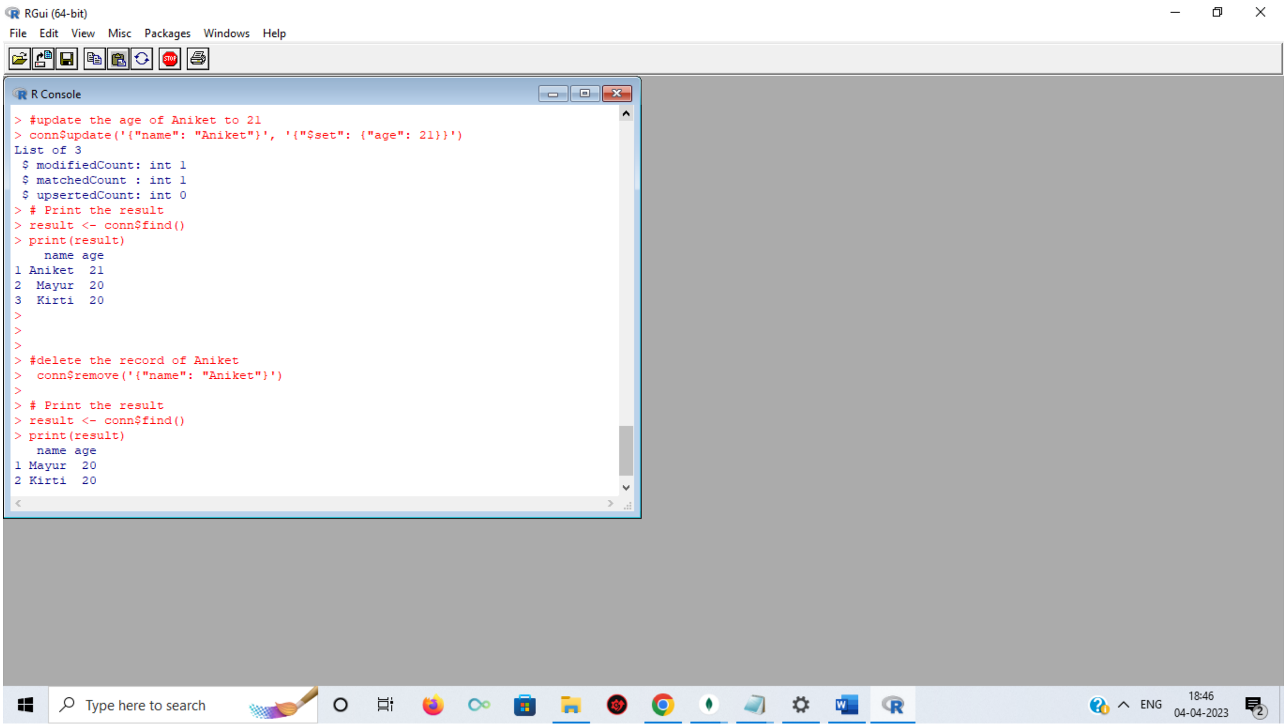
# Confirm that the records have been inserted
result <- conn$find()
print(result)

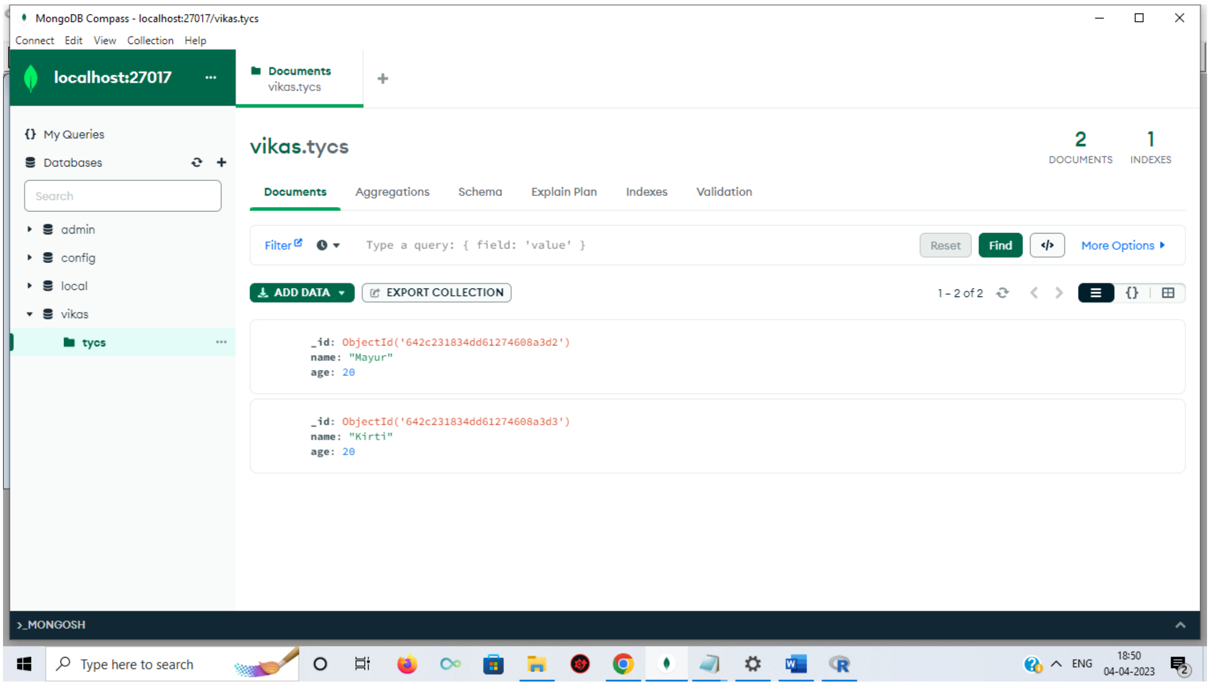
# Find the record of aniket in the collection
result <- conn$find('{"name": "Aniket"}')
print(result)

#update the age of Aniket to 21
conn$update('{"name": "Aniket"}', '{"$set": {"age": 21}}')
# Print the result
result <- conn$find()
print(result)

#delete the record of Aniket
 conn$remove('{"name": "Aniket"}')

# Print the result
result <- conn$find()
print(result)




MongoDB Compass - localhost:27017/vikas.tycs
Connect Edit View Collection Help
localhost:27017
Documents
vikas.tycs
My Queries
Databases
Search
admin
config
local
vikas
tycs
vikas.tycs
2
DOCUMENTS
1
INDEXES
Documents
Aggregations
Schema
Explain Plan
Indexes
Validation
Filter
Type a query: { field: 'value' }
Reset
Find
More Options
ADD DATA
EXPORT COLLECTION
1 - 2 of 2
_id: ObjectId('642c231834dd61274608a3d2')
name: "Mayur"
age: 20
_id: ObjectId('642c231834dd61274608a3d3')
name: "Kirti"
age: 20
>_MONGOSH
Type here to search
ENG
18:50
04-04-2023

Practical No:2

Aim: Practical of Data collection, Data curation and management for Large-scale Data system

```
use tycs
db.dropDatabase()
db.collection1.insert({id:001, Name:”Rajat”})
show collections;

db.collection1.insert({ id: 002, name:"Raj", course:[{name:"CS", duration:7},
 {name:"Java", duration:5}]})

var ins = [ {"StudentID" : 100, "Name" : "Mayur"} , {"StudentID" : 101,"Name" : "kirti"}];

db.collections1.insert(ins);

 db.collection1.find()

db.collection1.find().pretty()

db.collection1.find({"id":{$gt:1}}).pretty()

db.collection1.find({"id":{$lt:2}}).pretty()

db.collection1.update({"Name":"Rajat"},{$set:{"Name":"Aniket"}})

db.collection1.find({},{"name":1})

db.collection1.find()

db.collection1.find().sort({id:-1})
```

Command Prompt - mongo

```
---

> use tycs;
switched to db tycs
> db.dropDatabase()
{ "ok" : 1 }
> db.collection1.insert({id:001, Name:"Rajat"})
2023-04-04T19:01:32.355+0530 E QUERY    [js] SyntaxError: illegal character @(shell):1:36
> show collections
> show collections;
> db.collection1.insert({id:001, Name:"Rajat"})
WriteResult({ "nInserted" : 1 })
> show collections;
collection1
> db.collection1.insert({ id: 002, name:"Raj", course:[{name:"CS", duration:7}, {name:"Java", duration:5}]})
WriteResult({ "nInserted" : 1 })
> var ins = [ {"StudentID" : 100, "Name" : "Rajaa"} , {"StudentID" : 101,"Name" : "Raju"}];
>
> db.collections1.insert(ins);
BulkWriteResult({
        "writeErrors" : [ ],
        "writeConcernErrors" : [ ],
        "nInserted" : 2,
        "nUpserted" : 0,
        "nMatched" : 0,
        "nModified" : 0,
        "nRemoved" : 0,
        "upserted" : [ ]
})
```

Command Prompt - mongo

```
> db.collection1.find()
{ "_id" : ObjectId("642c26ff6b92d8fcf8971a4c"), "id" : 1, "Name" : "Rajat" }
{ "_id" : ObjectId("642c275e6b92d8fcf8971a4d"), "id" : 2, "name" : "Raj", "course" : [ { "name" : "CS", "duration" : 7 }
, { "name" : "Java", "duration" : 5 } ] }
> db.collection1.find().pretty()
{ "_id" : ObjectId("642c26ff6b92d8fcf8971a4c"), "id" : 1, "Name" : "Rajat" }
{
        "_id" : ObjectId("642c275e6b92d8fcf8971a4d"),
        "id" : 2,
        "name" : "Raj",
        "course" : [
                {
                        "name" : "CS",
                        "duration" : 7
                },
                {
                        "name" : "Java",
                        "duration" : 5
                }
        ]
}
> db.collection1.find({"id":{$gt:1}}).pretty()
{
        "_id" : ObjectId("642c275e6b92d8fcf8971a4d"),
        "id" : 2,
        "name" : "Raj",
        "course" : [
                {
                        "name" : "CS",
                        "duration" : 7
```

```
> db.collection1.find({"id":{$lt:2}}).pretty()
{ "_id" : ObjectId("642c26ff6b92d8fcf8971a4c"), "id" : 1, "Name" : "Rajat" }
> db.collection1.update({"Name":"Rajat"},{$set:{"Name":"Anikey"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.collection1.find({},{"name":1})
{ "_id" : ObjectId("642c26ff6b92d8fcf8971a4c") }
{ "_id" : ObjectId("642c275e6b92d8fcf8971a4d"), "name" : "Raj" }
> db.collection1.find()
{ "_id" : ObjectId("642c26ff6b92d8fcf8971a4c"), "id" : 1, "Name" : "Anikey" }
{ "_id" : ObjectId("642c275e6b92d8fcf8971a4d"), "id" : 2, "name" : "Raj", "course" : [ { "name" : "CS", "duration" : 7 }
, { "name" : "Java", "duration" : 5 } ] }
> db.collection1.find().sort({id:-1})
{ "_id" : ObjectId("642c275e6b92d8fcf8971a4d"), "id" : 2, "name" : "Raj", "course" : [ { "name" : "CS", "duration" : 7 }
, { "name" : "Java", "duration" : 5 } ] }
{ "_id" : ObjectId("642c26ff6b92d8fcf8971a4c"), "id" : 1, "Name" : "Anikey" }
>
```

Practical No 3

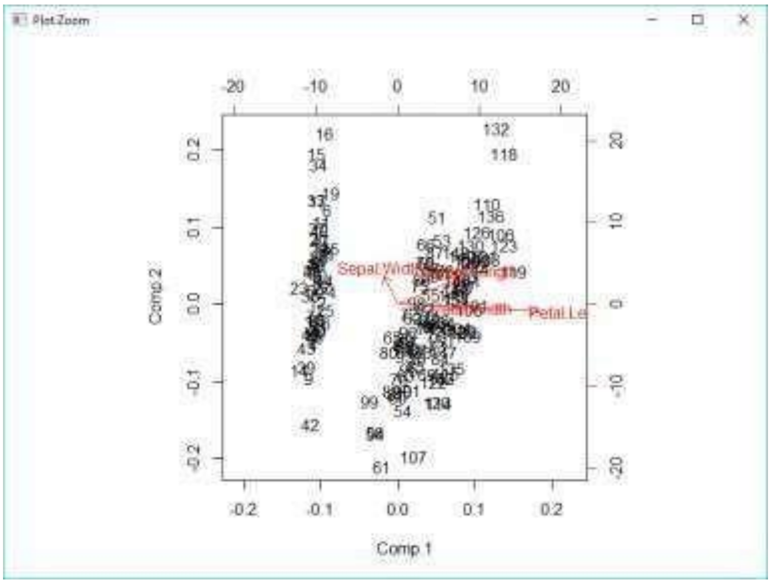
Aim: To perform practical of Principal Component Analysis (PCA).

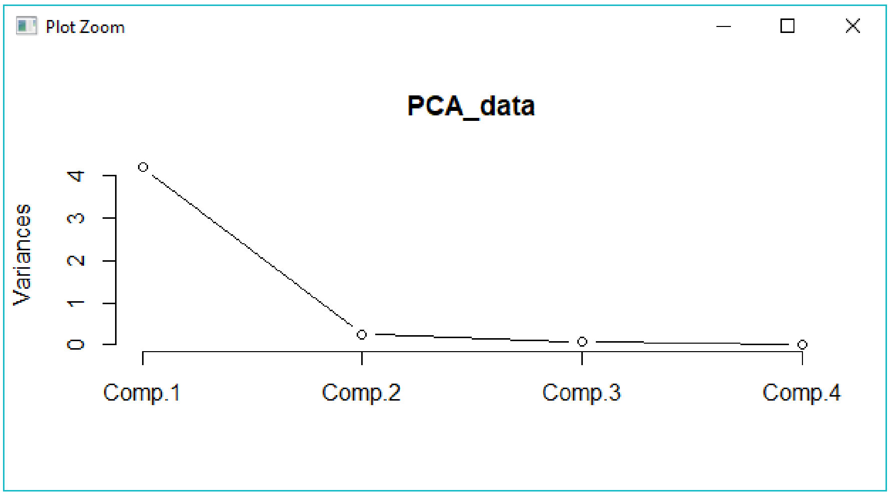
Program Code:

```
data_iris <- iris[1:4]
cov_data <- cov(data_iris)
Eigen_data <- eigen(cov_data)
PCA_data <- princomp(data_iris,cor = "False")
Eigen_data$values
PCA_data$sdev^2
PCA_data$loadings[,1:4]
Eigen_data$vectors
summary(PCA_data)
biplot(PCA_data)
screeplot(PCA_data,type = 'lines')
model2 = PCA_data$loadings[,1]
model2_scores <- as.matrix(data_iris)%*%model2
library(class)
install.packages("e1071")
library(e1071)
mod1 <- naiveBayes(iris[,1:4],iris[,5])
mod2 <- naiveBayes(model2_scores,iris[,5])
table(predict(mod1,iris[,1:4]),iris[,5])
table(predict(mod2,model2_scores),iris[,5])
```

Conclusion: Practical of Principal Component Analysis (PCA) has been executed successfully.

OUTPUT: -


Plot Zoom
Comp.2
Comp.1
Sepal.Width
Petal.Le


Plot Zoom
PCA_data
Variances
Comp.1
Comp.2
Comp.3
Comp.4

Practical No 4

Aim: To perform practical of Clustering.

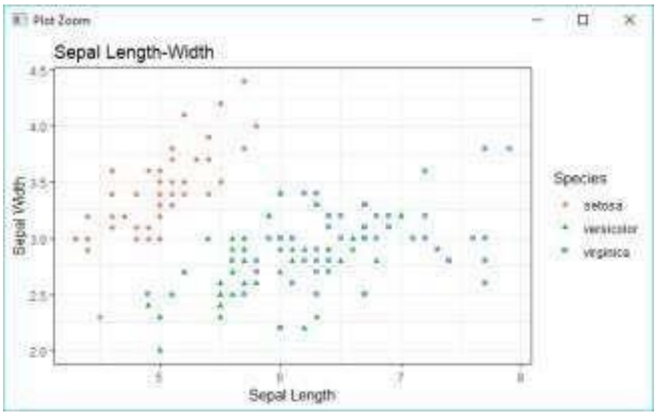
Program Code:

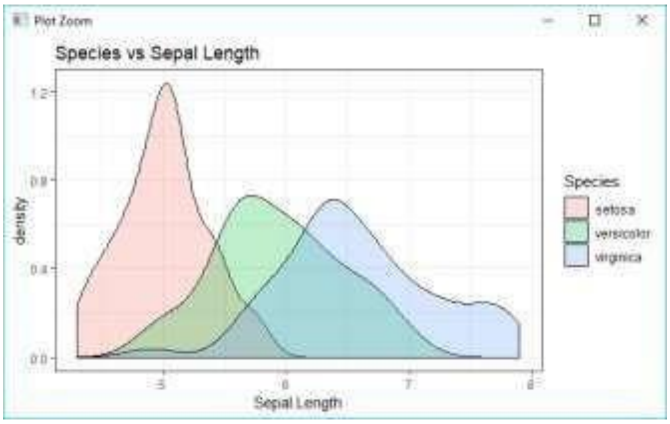
```
install.packages("ggplot2")
library(ggplot2)
scatter <- ggplot(data=iris,aes(x=Sepal.Length,y=Sepal.Width))
scatter + geom_point(aes(color=Species,shape=Species))+
theme_bw()+
xlab("Sepal Length")+ylab("Sepal Width")+
ggtitle("Sepal Length-Width")
ggplot(data=iris,aes(Sepal.Length,fill=Species))+
theme_bw()+
geom_density(alpha=0.25)+
labs(x="Sepal.Length",title="Species vs Sepal Length")
vol <- ggplot(data=iris,aes(x=Sepal.Length))
vol + stat_density(aes(ymax=..density..,ymin=-
..density..,fill=Species,color=Species),geom="ribbon",position="identity")+
facet_grid(.~Species)+coord_flip()+theme_bw()+labs(x="Sepal Length",title="Species vs
Sepal Length")
vol <- ggplot(data=iris,aes(x=Sepal.Width))
vol + stat_density(aes(ymax=..density..,ymin=-
..density..,fill=Species,color=Species),geom="ribbon",position="identity")+
facet_grid(.~Species)+coord_flip()+theme_bw()+labs(x="Sepal Width",title="Species vs
Sepal Width")
irisData <- iris[,1:4]
totalwSS<-c()
for(i in 1:15)
{clusterIRIS<- kmeans(irisData,centers = i)
totalwSS[i] <-clusterIRIS$tot.withinss}
plot(x=1:15,y=totalwSS,type="b",xlab="Number of Clusters",ylab="Within groups sum-of-
squares")
install.packages("NbClust")
library(NbClust)
par(mar=c(2,2,2,2))
nb<-NbClust(irisData,method="kmeans")
hist(nb$Best.nc[1,],breaks=15,main="Histogram for Number of Clusters")
install.packages("vegan")
library(vegan)
modelData<-cascadeKM(irisData,1,10,iter=100)
plot(modelData,sortg=TRUE)
```

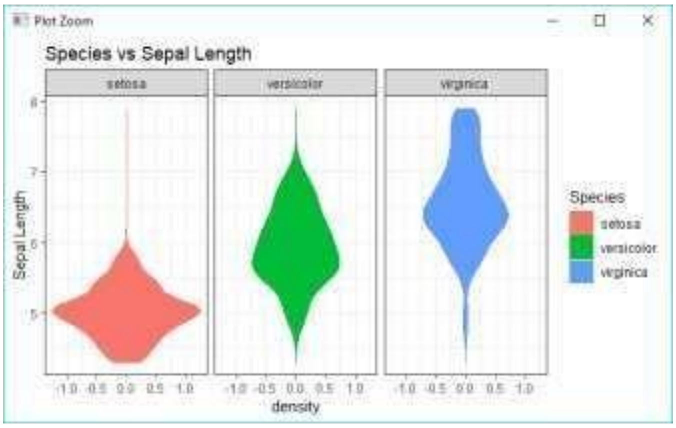
```
modelData$results[2,]
which.max(modelData$results[2,])
library(cluster)
cl<-kmeans(iris[,-5],2)
dis<-dist(iris[,-5])^2
sil=silhouette(cl$cluster,dis)
plot(sil,main="Clustering Data with silhoutte plot using 2 Clusters",col=c("cyan","blue"))
library(cluster)
cl<-kmeans(iris[,-5],8)
dis<-dist(iris[,-5])^2
sil=silhouette(cl$cluster,dis)
plot(sil,main="Clustering Data with silhoutte plot using 8
Clusters",col=c("cyan","blue","orange","yellow","red","gray","green","maroon"))
install.packages("factoextra")
library(factoextra)
install.packages("clustertend")
library(clustertend)
genx<-function(x){
runif(length(x),min(x),(max(x)))}
random_df<-apply(iris[,-5],2,genx)
random_df<-as.data.frame(random_df)
iris[,-5]<-scale(iris[,-5])
random_df<-scale(random_df)
res<-get_clust_tendency(iris[,-5],n=nrow(iris)-1,graph=FALSE)
res$hopkins_stat
hopkins(iris[,-5],n=nrow(iris)-1)
res<-get_clust_tendency(random_df,n=nrow(random_df)-1,graph=FALSE)
res$hopkins_stat
```

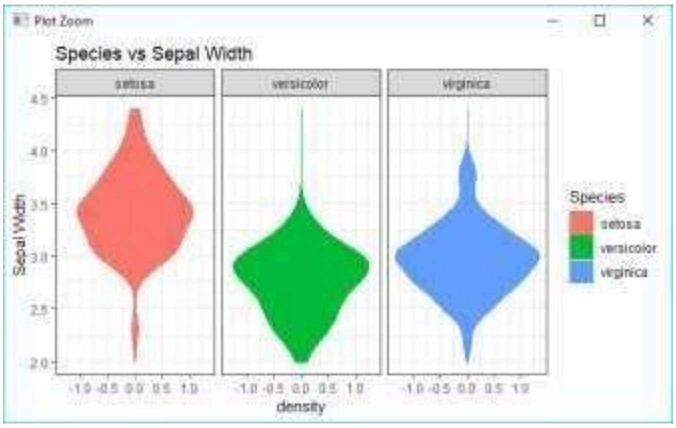
<u>Conclusion</u>: Practical of Clustering has been executed successfully.

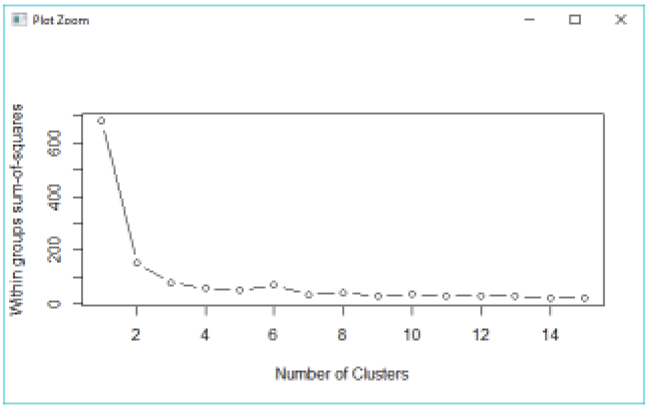
OUTPUT: -

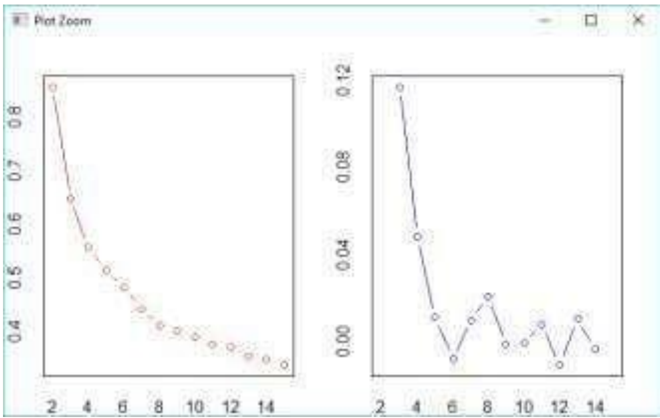

Plot Zoom
Sepal Length-Width
Sepal Width
Sepal Length
Species
setosa
versicolor
virginica

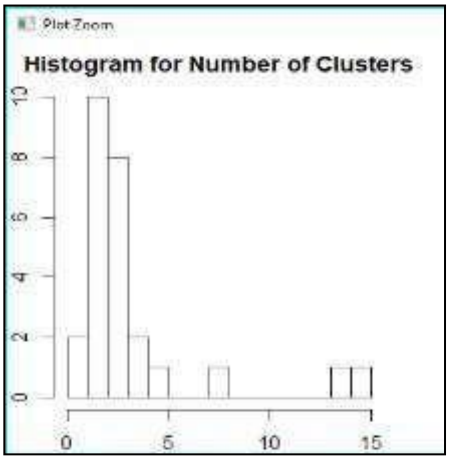

Plot Zoom
Species vs Sepal Length
density
Sepal Length
Species
setosa
versicolor
virginica

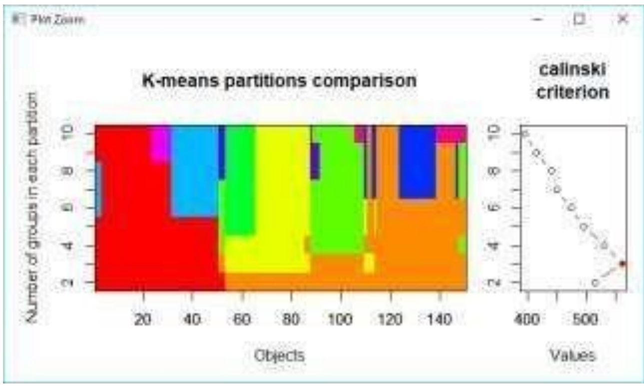

Plot Zoom
Species vs Sepal Length
setosa
versicolor
virginica
Sepal Length
density
Species
setosa
versicolor
virginica

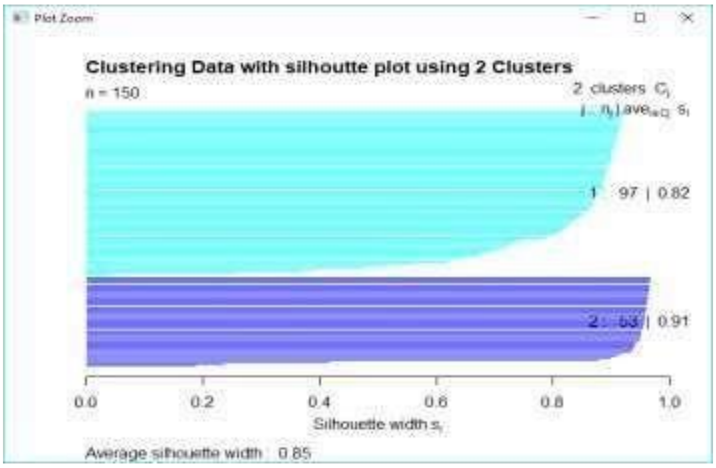

Plot Zoom
Species vs Sepal Width
setosa
versicolor
virginica
Sepal Width
density
Species
setosa
versicolor
virginica

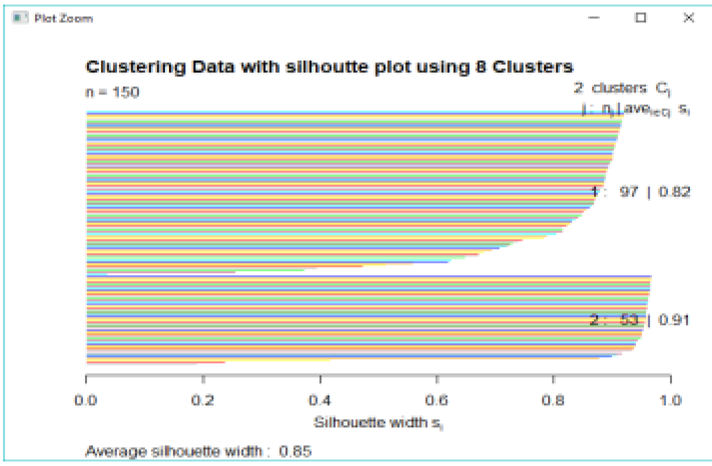

Plot Zoom
Within groups sum-of-squares
Number of Clusters


Plot Zoom


Plot Zoom
Histogram for Number of Clusters
0
2
4
6
8
10
0
5
10
15


Plot Zoom
K-means partitions comparison
calinski criterion
Number of groups in each partition
2
4
6
8
10
20
40
60
80
100
120
140
Objects
400
500
Values


Plot Zoom
Clustering Data with silhoutte plot using 2 Clusters
n = 150
2 clusters C_j
j : n_j | $ave_{i \in Cj}$ s_i
1 : 97 | 0.82
2 : 53 | 0.91
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette width s_i
Average silhouette width : 0.85


Plot Zoom
Clustering Data with silhoutte plot using 8 Clusters
n = 150
2 clusters C_j
j : n_j | $ave_{i \in Cj}$ s_i
1 : 97 | 0.82
2 : 53 | 0.91
0.0
0.2
0.4
0.6
0.8
1.0
Silhouette width s_i
Average silhouette width : 0.85

Practical no 5

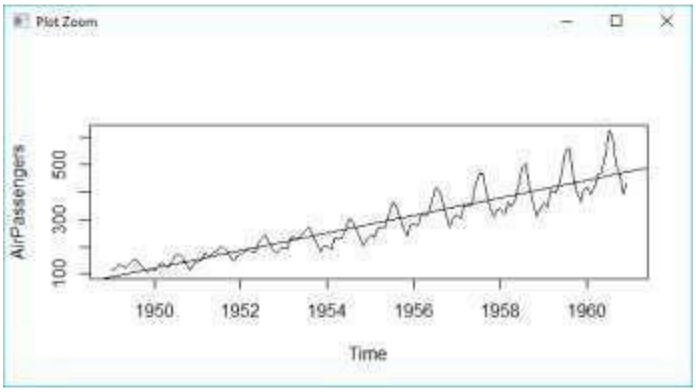
Aim: To perform practical of Time-series forecasting.

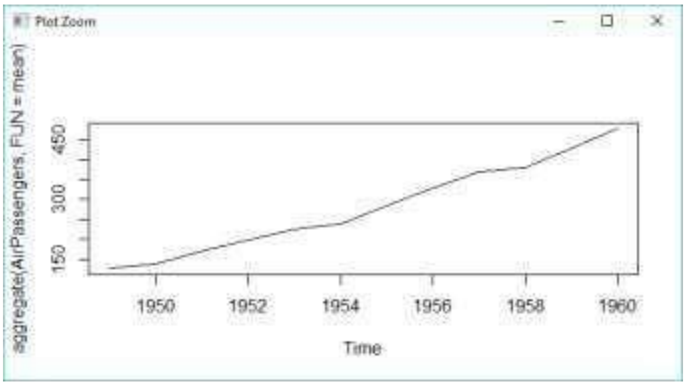
Program Code:

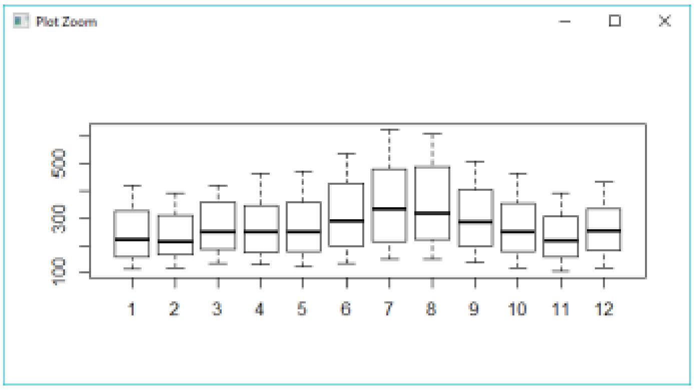
```
data(AirPassengers)
class(AirPassengers)
start(AirPassengers)
end(AirPassengers)
frequency(AirPassengers)
summary(AirPassengers)
plot(AirPassengers)
abline(reg=lm(AirPassengers~time(AirPassengers)))
cycle(AirPassengers)
plot(aggregate(AirPassengers,FUN=mean))
boxplot(AirPassengers~cycle(AirPassengers))
acf(log(AirPassengers))
(fit<-arima(log(AirPassengers),c(0,1,1),seasonal=list(order=c(0,1,1),period=12)))
pred<-predict(fit,n.ahead=10*12)
ts.plot(AirPassengers,2.718^pred$pred,log="y",lty=c(1,3))
```

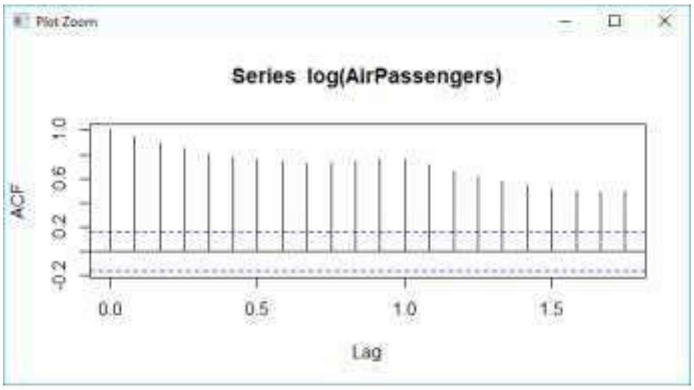
Conclusion: Practical of Time-series forecasting has been executed successfully.

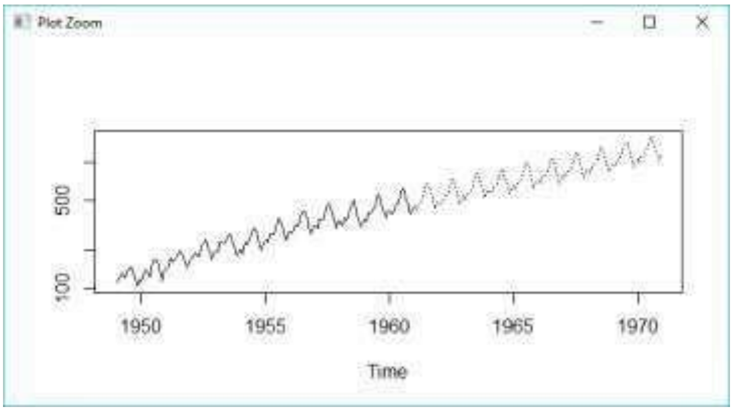
OUTPUT:


Plot Zoom
AirPassengers
100
300
500
1950
1952
1954
1956
1958
1960
Time


Plot Zoom
aggregate(AirPassengers, FUN = mean)
150
300
450
1950
1952
1954
1956
1958
1960
Time


Plot Zoom
100
300
500
1
2
3
4
5
6
7
8
9
10
11
12


Plot Zoom
Series log(AirPassengers)
ACF
-0.2
0.2
0.6
1.0
0.0
0.5
1.0
1.5
Lag


Plot Zoom
100
500
1950
1955
1960
1965
1970
Time

Practical no 6

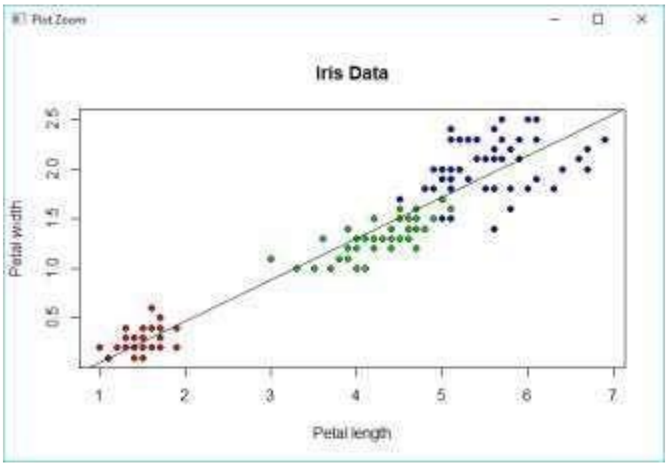
Aim: To perform practical of Simple/Multiple Linear Regression.

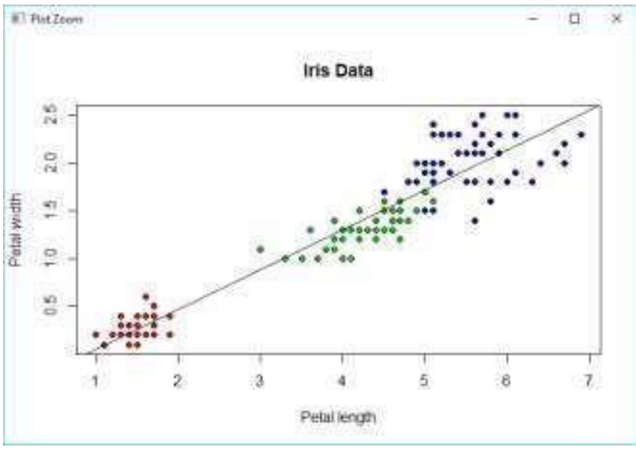
Program Code:

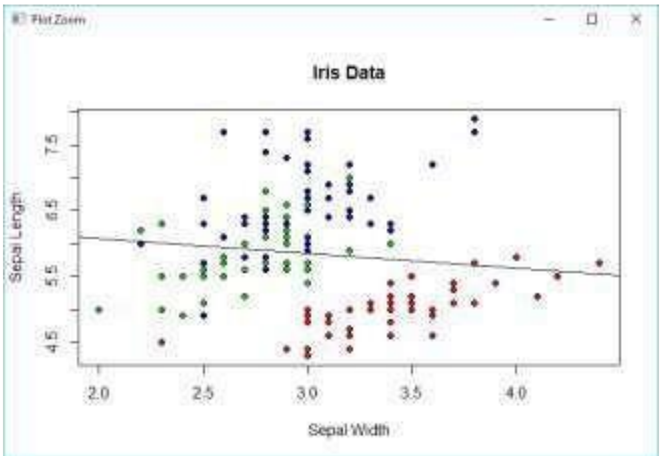
```
lsfit(iris$Petal.Length,iris$Petal.Width)$coefficients
plot(iris$Petal.Length,iris$Petal.Width,pch=21,bg=c("red","green3","blue")[unclass(iris$Spe
cies)],main="Iris Data",xlab="Petal length",ylab="Petal width")
abline(lsfit(iris$Petal.Length,iris$Petal.Width)$coefficients,col="black")
lm(Petal.Width~Petal.Length,data=iris)$coefficients
plot(iris$Petal.Length,iris$Petal.Width,pch=21,bg=c("red","green3","blue")[unclass(iris$Spe
cies)],main="Iris Data",xlab="Petal length",ylab="Petal width")
abline(lm(Petal.Width~Petal.Length,data=iris)$coefficients,col="black")
summary(lm(Petal.Width~Petal.Length,data=iris))
plot(iris$Sepal.Width,iris$Sepal.Length,pch=21,bg=c("red","green3","blue")[unclass(iris$Sp
ecies)],main="Iris Data",xlab="Sepal Width",ylab="Sepal Length")
abline(lm(Sepal.Length~Sepal.Width,data=iris)$coefficients,col="black")
summary(lm(Sepal.Length~Sepal.Width,data=iris))
plot(iris$Sepal.Width,iris$Sepal.Length,pch=21,bg=c("red","green3","blue")[unclass(iris$Sp
ecies)],main="Iris Data",xlab="Petal length",ylab="Sepal length")
abline(lm(Sepal.Length~Sepal.Width,data=iris)$coefficients,col="black")
abline(lm(Sepal.Length~Sepal.Width,
   data=iris[which(iris$Species=="setosa"),])$coefficients,col="red")
abline(lm(Sepal.Length~Sepal.Width
      data=iris[which(iris$Species=="versicolor"),])$coefficients,col="green3")
abline(lm(Sepal.Length~Sepal.Width,
      data=iris[which(iris$Species=="virginica"),])$coefficients,col="blue")
lm(Sepal.Length~Sepal.Width,data=iris[which(iris$Species=="setosa"),])$coefficients
lm(Sepal.Length~Sepal.Width,data=iris[which(iris$Species=="versicolor"),])$coefficients
lm(Sepal.Length~Sepal.Width,data=iris[which(iris$Species=="virginica"),])$coefficients
lm(Sepal.Length~Sepal.Width:Species+Species-1,data=iris)$coefficients
summary(lm(Sepal.Length~Sepal.Width:Species+Species-1,data=iris))
summary(step(lm(Sepal.Length~Sepal.Width*species,data=iris)))
lm(Sepal.Length~Sepal.Width:Species+Species-1,data=iris)$coefficients
lm(Sepal.Length~Sepal.Width:Species+Species,data=iris)$coefficients
```

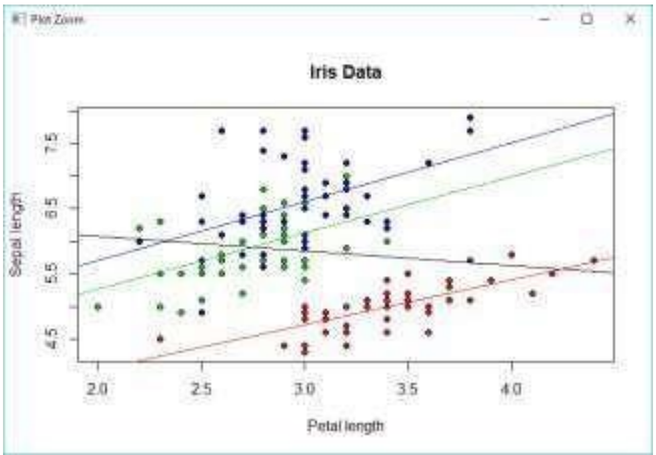
Conclusion: Practical of Simple/Multiple Linear Regression has been executed successfully.

OUTPUT:


Plot Zoom
Iris Data
Petal width
Petal length
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7


Plot Zoom
Iris Data
Petal width
Petal length
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7


Plot Zoom
Iris Data
Sepal Length
Sepal Width
4.5
5.5
6.5
7.5
2.0
2.5
3.0
3.5
4.0


Plot Zoom
Iris Data
Sepal length
Petal length
4.5
5.5
6.5
7.5
2.0
2.5
3.0
3.5
4.0

Practical No 7

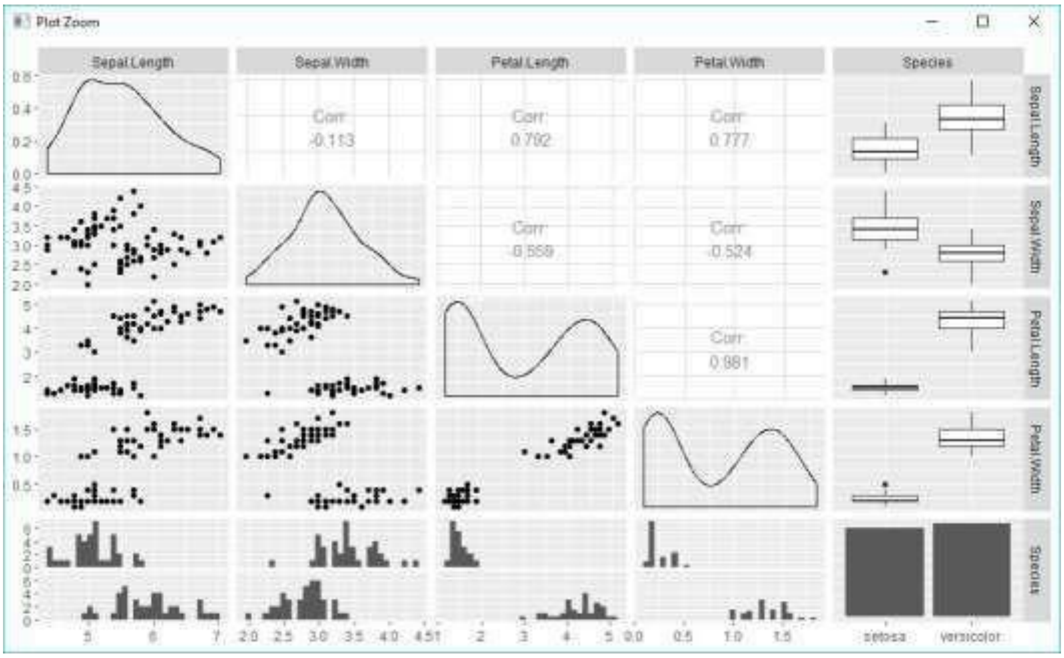
Aim: To perform practical of Logistics Regression.

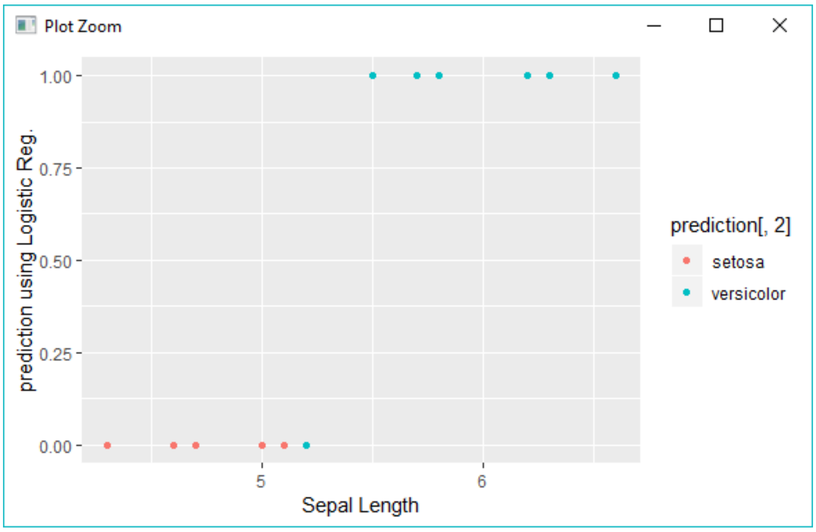
Program Code:

```
library(datasets)
ir_data<-iris
head(ir_data)
str(ir_data)
levels(ir_data$species)
sum(is.na(ir_data))
ir_data<-ir_data[1:100,]
set.seed(100)
samp<-sample(1:100,80)
ir_test<-ir_data[samp,]
ir_ctrl<-ir_data[-samp,]
install.packages("ggplot2")
library(ggplot2)
install.packages("GGally")
library(GGally)
ggpairs(ir_test)
y<-ir_test$Species;x<-ir_test$Sepal.Length
glfit<-glm(y~x,family='binomial')
summary(glfit)
newdata<-data.frame(x=ir_ctrl$Sepal.Length)
predicted_val<-predict(glfit,newdata,type="response")
prediction<-data.frame(ir_ctrl$Sepal.Length,ir_ctrl$Species,predicted_val)
prediction
qplot(prediction[,1],round(prediction[,3]),col=prediction[,2],xlab='Sepal
Length',ylab='prediction using Logistic Reg.')
```

Conclusion: Practical of Logistics Regression has been executed successfully.

OUTPUT


Plot Zoom
Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
Species
Corr:
-0.113
Corr:
0.792
Corr:
0.777
Corr:
-0.559
Corr:
-0.524
Corr:
0.981
setosa
versicolor


Plot Zoom
prediction using Logistic Reg.
1.00
0.75
0.50
0.25
0.00
5
6
Sepal Length
prediction[, 2]
setosa
versicolor

Practical no 8

<u>Aim</u>: To perform practical of Hypothesis testing.

<u>Program Code</u>:

```
x=c(6.2,6.6,7.1,7.4,7.6,7.9,8,8.3,8.4,8.5,8.6,
   8.8,8.8,9.1,9.2,9.4,9.4,9.7,9.9,10.2,10.4,10.8,11.3,11.9)
t.test(x-9,alternative ="two.sided",conf.level = 0.95)
x=c(418,421,421,422,425,427,431,434,437,439,446,447,448,453,454,463,465)
y=c(429,430,430,431,36,437,440,441,445,446,447)
test2<-t.test(x,y,alternative = "two.sided",mu=0,var.equal=F,conf.level=0.95)
test2
```

<u>Conclusion</u>: Practical of Hypothesis testing has been executed successfully.

OUTPUT-

```
> x=c(6.2,6.6,7.1,7.4,7.6,7.9,8,8.3,8.4,8.5,8.6,
+     8.8,8.8,9.1,9.2,9.4,9.4,9.7,9.9,10.2,10.4,10.8,11.3,11.9)
> t.test(x-9,alternative ="two.sided",conf.level = 0.95)

        One Sample t-test

data:  x - 9
t = -0.35687, df = 23, p-value = 0.7244
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -0.7079827  0.4996494
sample estimates:
 mean of x
-0.1041667

> x=c(418,421,421,422,425,427,431,434,437,439,446,447,448,453,454,463,465)
> y=c(429,430,430,431,36,437,440,441,445,446,447)
> test2<-t.test(x,y,alternative = "two.sided",mu=0,var.equal=F,conf.level=0.95)
> test2

        Welch Two Sample t-test

data:  x and y
t = 1.0123, df = 10.202, p-value = 0.3348
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -44.46343 118.86984
sample estimates:
mean of x mean of y
 438.2941  401.0909
```

Practical No 9

Aim: To perform practical of Analysis of Variance .

Program Code:

```
y1=c(18.2,20.1,17.6,16.8,18.8,19.7,19.1)
y2=c(17.4,18.7,19.1,16.4,15.9,18.4,17.7)
y3=c(15.2,18.8,17.7,16.5,15.9,17.1,16.7)
y=c(y1,y2,y3)
n=rep(7,3)
n
group =rep(1:3,n)
group
tmp=tapply(y,group,stem)
stem(y)
tmpfn=function(x)c(sum=sum(x),mean=mean(x),var=var(x),n=length(x))
tapply(y,group,tmpfn)
tmpfn(y)
data=data.frame(y=y,group=factor(group))
fit=lm(y~group,data)
anova(fit)
df=anova(fit)[,"Df"]
names(df)=c("trt","err")
df
alpha=c(0.05,0.01)
qf(alpha,df["trt"],df["err"],lower.tail=FALSE)
anova(fit)["Residuals","Sum Sq"]
anova(fit)["Residuals","Sum Sq"]/qchisq(c(0.025,0.975),18,lower.tail=FALSE)
```

Conclusion: Practical of Analysis of Variance has been executed successfully.

OUTPUT –

```
> y1=c(18.2,20.1,17.6,16.8,18.8,19.7,19.1)
> y2=c(17.4,18.7,19.1,16.4,15.9,18.4,17.7)
> y3=c(15.2,18.8,17.7,16.5,15.9,17.1,16.7)
> y=c(y1,y2,y3)
> n=rep(7,3)
> n
[1] 7 7 7
> group =rep(1:3,n)
> group
 [1] 1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 3 3 3 3 3 3
> tmp=tapply(y,group,stem)

  The decimal point is at the |

  16 | 8
  17 | 6
  18 | 28
  19 | 17
  20 | 1


  The decimal point is at the |

  15 | 9
  16 | 4
  17 | 47
  18 | 47
  19 | 1


  The decimal point is at the |

  15 | 29
  16 | 57
  17 | 17
  18 | 8
```

```
> stem(y)

  The decimal point is at the |

  15 | 299
  16 | 4578
  17 | 14677
  18 | 24788
  19 | 117
  20 | 1

> tmpfn=function(x)c(sum=sum(x),mean=mean(x),var=var(x),n=length(x))
> tapply(y,group,tmpfn)
$`1`
       sum       mean        var          n
130.300000  18.614286   1.358095   7.000000

$`2`
       sum       mean        var          n
123.600000  17.657143   1.409524   7.000000

$`3`
       sum       mean        var          n
117.900000  16.842857   1.392857   7.000000

> tmpfn(y)
       sum       mean        var          n
371.800000  17.704762   1.798476  21.000000
> data=data.frame(y=y,group=factor(group))
> fit=lm(y~group,data)
> anova(fit)
Analysis of Variance Table

Response: y
          Df Sum Sq Mean Sq F value  Pr(>F)
group      2 11.007  5.5033  3.9683 0.03735 *
Residuals 18 24.963  1.3868
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
> df=anova(fit)[,"Df"]
> names(df)=c("trt","err")
> df
trt err
  2  18
> alpha=c(0.05,0.01)
> qf(alpha,df["trt"],df["err"],lower.tail=FALSE)
[1] 3.554557 6.012905
> anova(fit)["Residuals","Sum Sq"]
[1] 24.96286
> anova(fit)["Residuals","Sum Sq"]/qchisq(c(0.025,0.975),18,lower.tail=FALSE)
[1] 0.7918086 3.0328790
```

Practical No 10

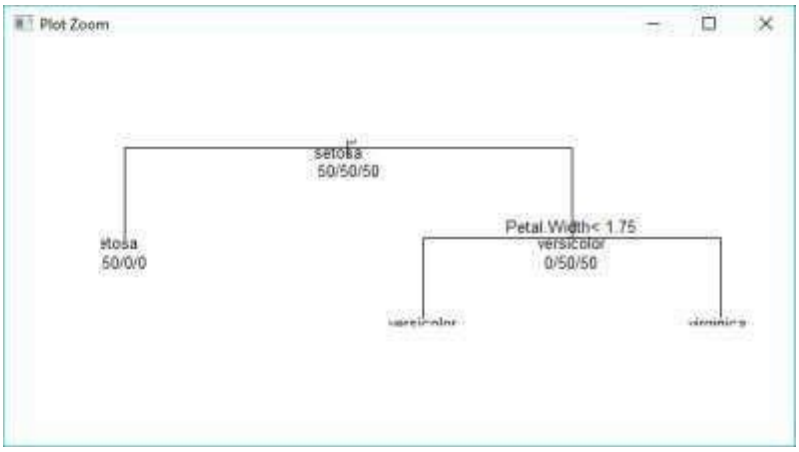
Aim: To perform practical of Decision Tree .

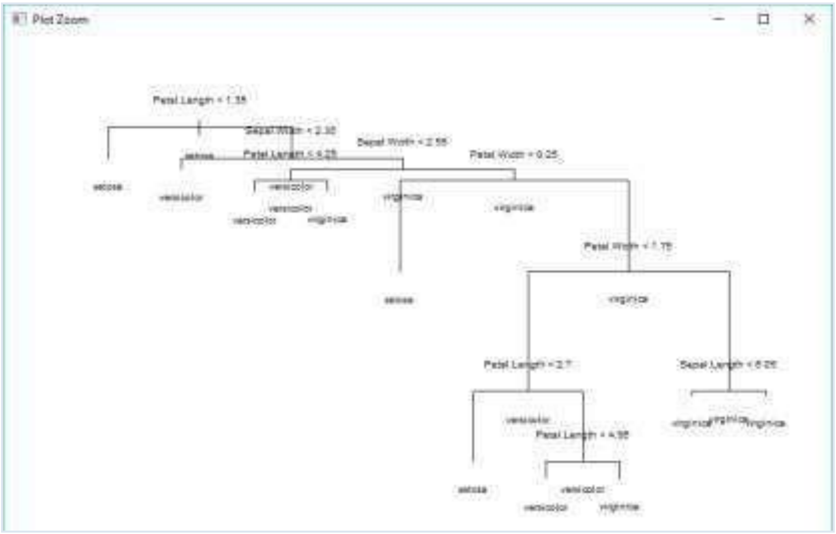
Program Code:

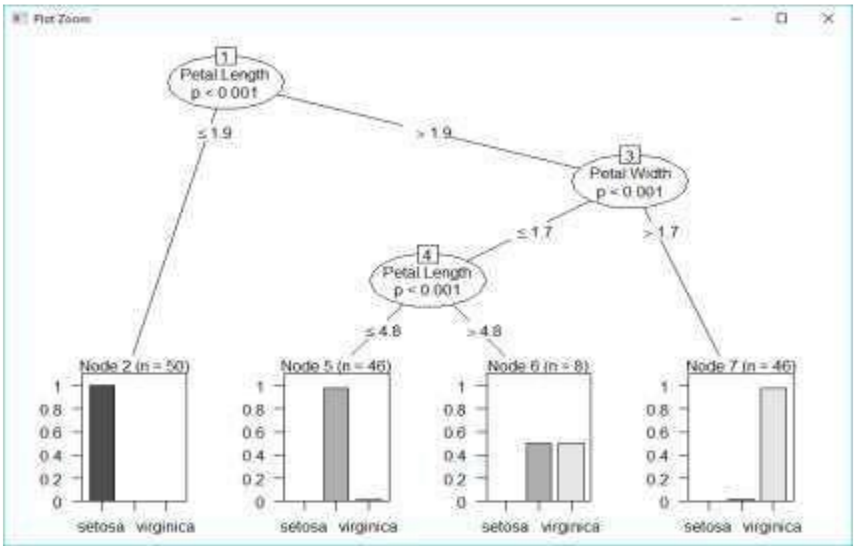
```
mydata<-data.frame(iris)
attach(mydata)
install.packages("rpart")
library(rpart)
model<-
rpart(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=mydata,method="
class")
plot(model)
text(model,use.n=TRUE,all=TRUE,cex=0.8)
install.packages("tree")
library(tree)
model1<-
tree(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=mydata,method="cl
ass",split="gini")
plot(model1)
text(model1,all=TRUE,cex=0.6)
install.packages("party")
library(party)
model2<-ctree(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=mydata)
plot(model2)
library(tree)
mydata<-data.frame(iris)
attach
model1<-
tree(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=mydata,method="cl
ass",control=tree.control(nobs=150,mincut=10))
plot(model1)
text(model1,all=TRUE,cex=0.6)
predict(model1,iris)(mydata)
model2<-
ctree(Species~Sepal.Length+Sepal.Width+Petal.Length+Petal.Width,data=mydata,controls=c
tree_control(maxdepth=2))
plot(model2)
```

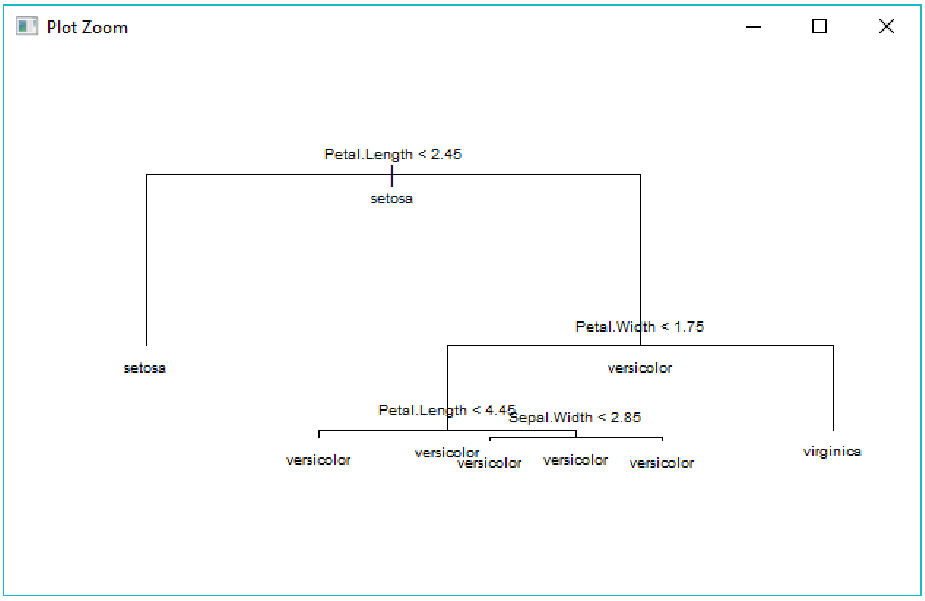
Conclusion: Practical of Decision tree has been executed successfully.

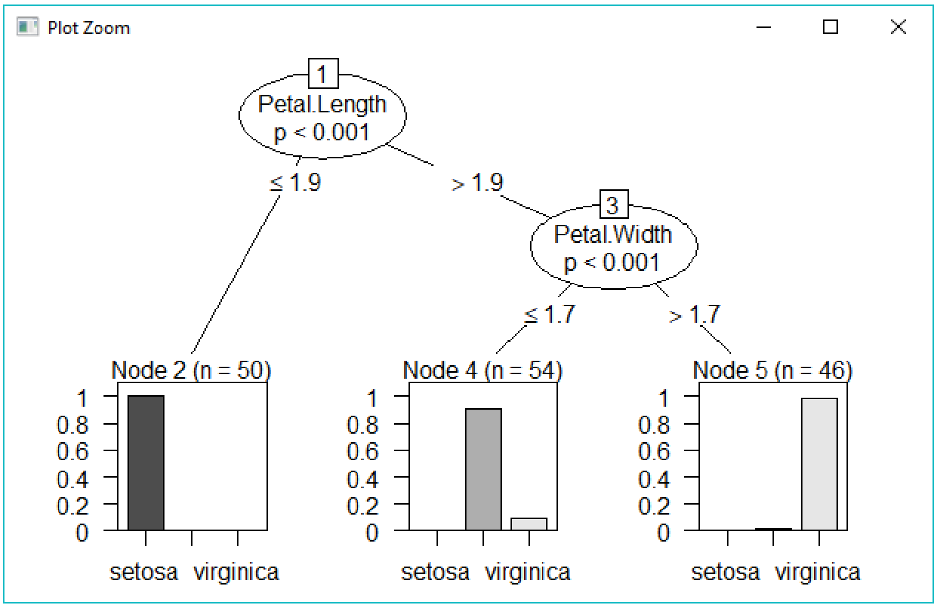
OUTPUT


Plot Zoom
setosa
50/50/50
Petal.Width< 1.75
versicolor
0/50/50
50/0/0




1
Petal.Length
p < 0.001
≤ 1.9
> 1.9
3
Petal.Width
p < 0.001
≤ 1.7
> 1.7
4
Petal.Length
p < 0.001
≤ 4.8
> 4.8
Node 2 (n = 50)
Node 5 (n = 46)
Node 6 (n = 8)
Node 7 (n = 46)
1
0.8
0.6
0.4
0.2
0
setosa
virginica


Plot Zoom
Petal.Length < 2.45
setosa
setosa
Petal.Width < 1.75
versicolor
Petal.Length < 4.45
Sepal.Width < 2.85
versicolor
versicolor
versicolor
versicolor
versicolor
virginica


Plot Zoom
1
Petal.Length
p < 0.001
≤ 1.9
> 1.9
3
Petal.Width
p < 0.001
≤ 1.7
> 1.7
Node 2 (n = 50)
Node 4 (n = 54)
Node 5 (n = 46)
1
0.8
0.6
0.4
0.2
0
setosa virginica