****

S.I.W.S

N.R SWAMY COLLEGE OF COMMERCE & ECONOMICS

AND SMT. THIRUMALAI COLLEGE OF SCIENCE.

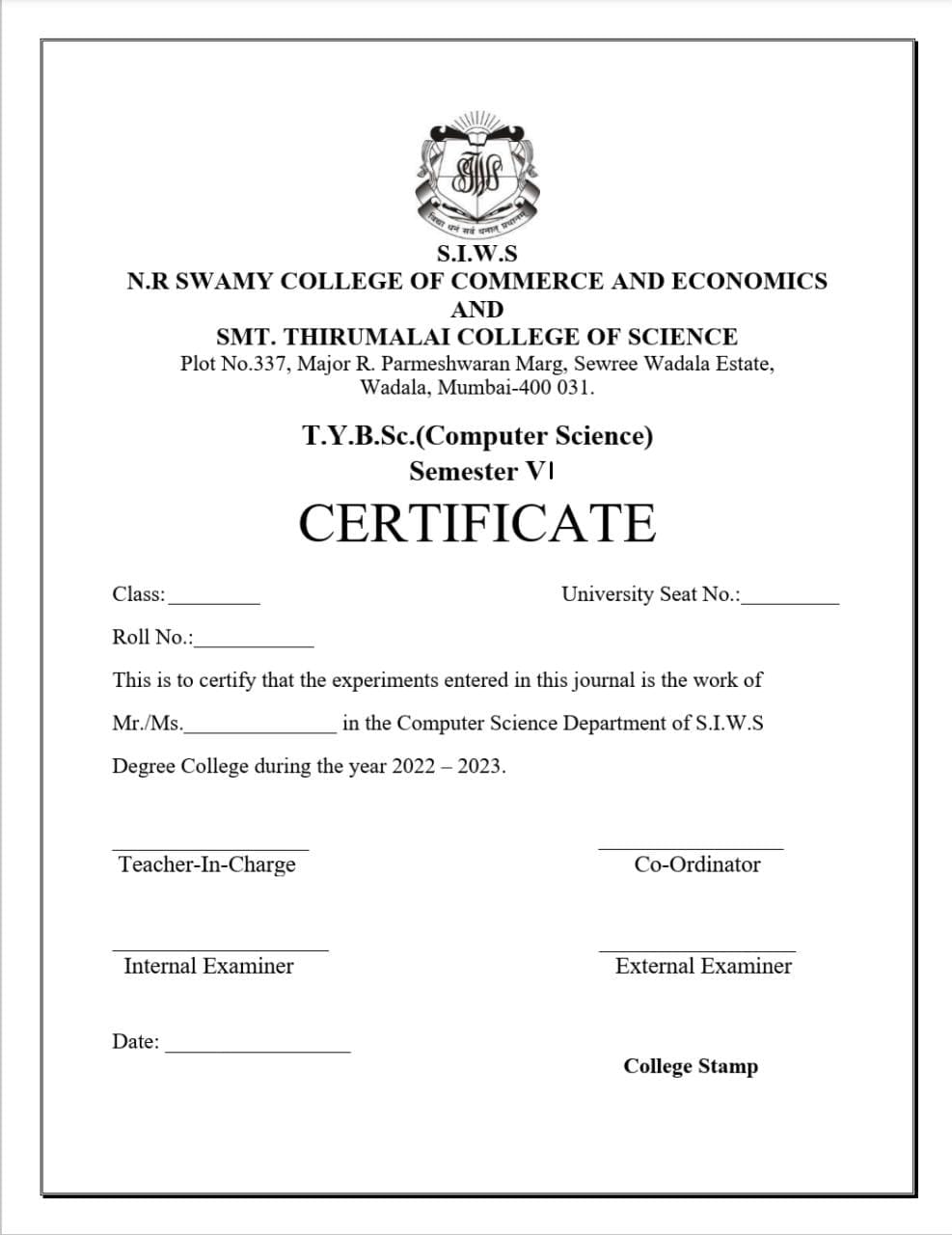
Data Science

ABHISHEK TRIPATHI

T.Y.BSC C.S.

Roll No.: 39034

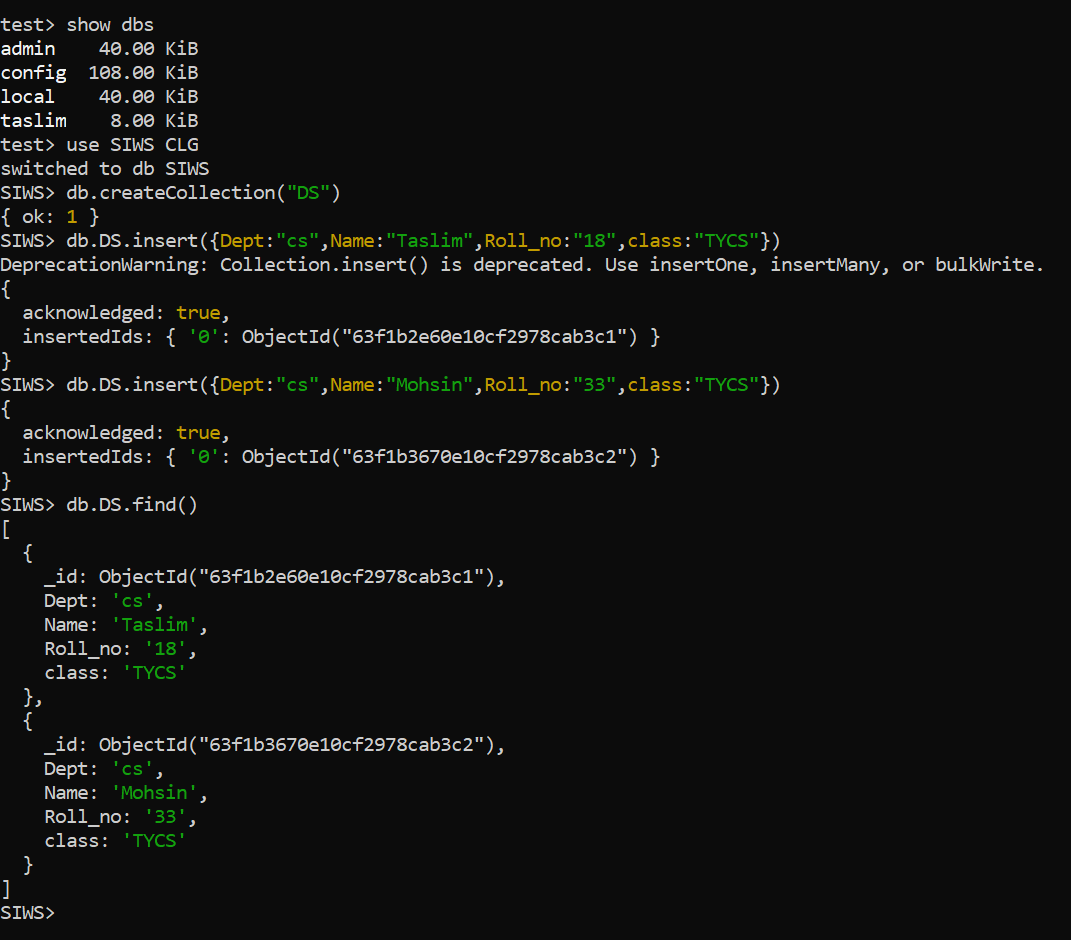
Year 2022-2023



**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **SR.NO.** | **DATE** | **PRACTICAL** | **PAGE** | **SIGN** |
| 1. |  | Practical of data collection, Data curation and management for Large-scale Data  system(such as MongoDB) |  |  |
| 2. |  | Practical of Data collection, Data curation and management for Unstructured data (No SQL) |  |  |
| 3. |  | Practical of Principal Component Analysis(PCA) |  |  |
| 4. |  | Perform the data clustering using clustering algorithm |  |  |
| 5. |  | Practical of Time – series forecasting |  |  |
| 6. |  | Practical of Simple and Multiple Linear Regression |  |  |
| 7. |  | Practical of Logistics Regression |  |  |
| 8. |  | Practical of Hypothesis Testing |  |  |
| 9. |  | Practical of Analysis of Variance |  |  |
| 10. |  | Practical of Decision Tree |  |  |

**OUTPUT**:



**PRACTICAL NO:- 1**

**Data curation and management for Unstructured data.**

**AIM:-** Practical of Data collection, Data curation and management for Unstructured data.

**DESCRIPTION:-** Data collection is the process of gathering and measuring information on variables of interest, in an established systematic fashion that enables one to answer stated research questions, test hypotheses, and evaluate outcomes.

A NoSQL database provides a mechanism for storage and retrieval of data that is

modeled in means other than the tabular relations used in relational databases.

When people use the term “NoSQL database”, they typically use it to refer to any

non-relational database. Some say the term “NoSQL” stands for “non SQL” while others say

it stands for “not only SQL.” Either way, most agree that NoSQL databases are databases that

store data in a format other than relational tables.

A common misconception is that NoSQL databases or non-relational databases don’t

store relationship data well. NoSQL databases can store relationship data—they just store it

differently than relational databases do. In fact, when compared with SQL databases, many

find modeling relationship data in NoSQL databases to be easier than in SQL databases,

because related data doesn’t have to be split between tables.

NoSQL data models allow related data to be nested within a single data structure.

**How NoSQL Databases Work:**

One way of understanding the appeal of NoSQL databases from a design perspective

is to look at how the data models of a SQL and a NoSQL database might look in an

oversimplified example using address data.

The SQL Case. For an SQL database, setting up a database for addresses begins with

the logical construction of the format and the expectation that the records to be stored are

going to remain relatively unchanged. After analyzing the expected query patterns, an SQL

database might optimize storage in two tables, one for basic information and one pertaining

to being a customer, with last name being the key to both tables. Each row in each table is a

single customer, and each column has the following fixed attributes:

1. Last name :: first name :: middle initial :: address fields :: email address :: phone number
2. Last name :: date of birth :: account number :: customer years :: communication preferences
3. Each type of NoSQL database would be designed with a specific customer situation in

mind, and there would be technical reasons for how each kind of database would be

organized. The simplest type to describe is the document database, in which it would

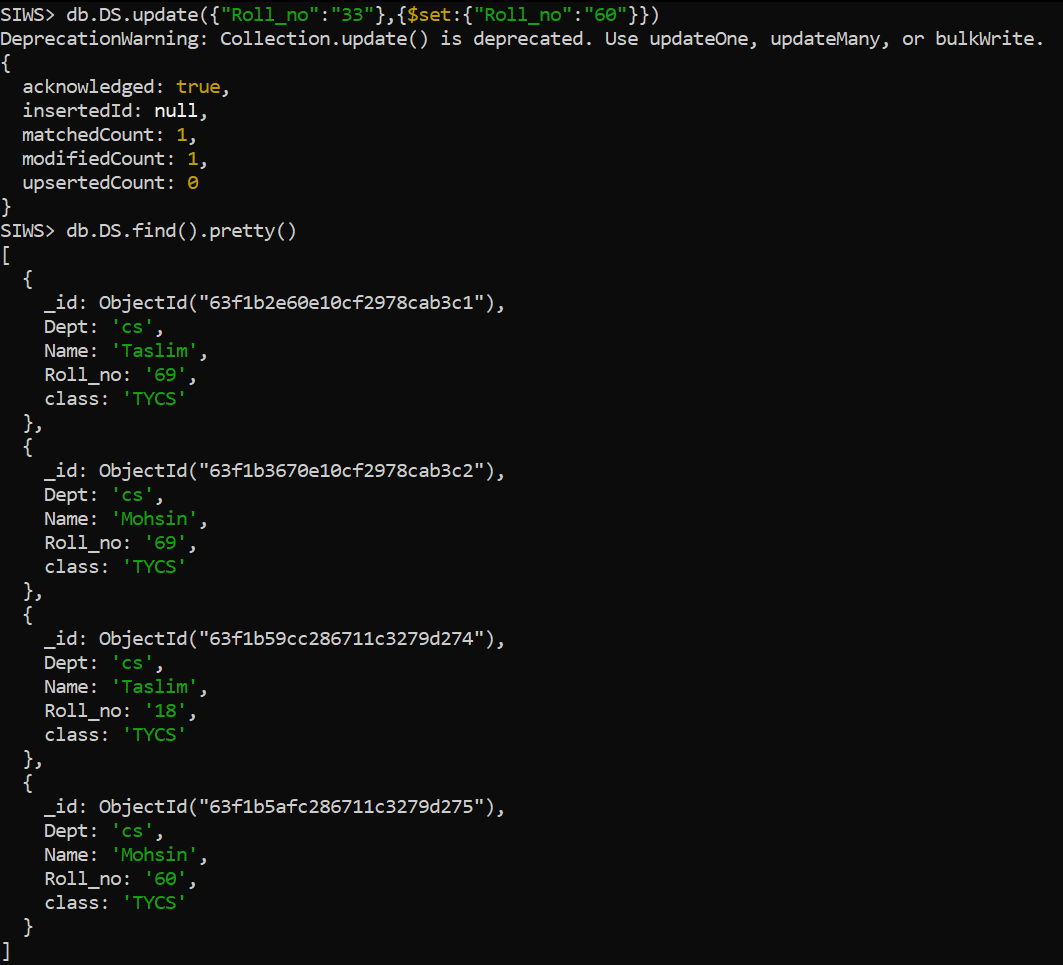
be natural to combine both the basic information and the customer information in one

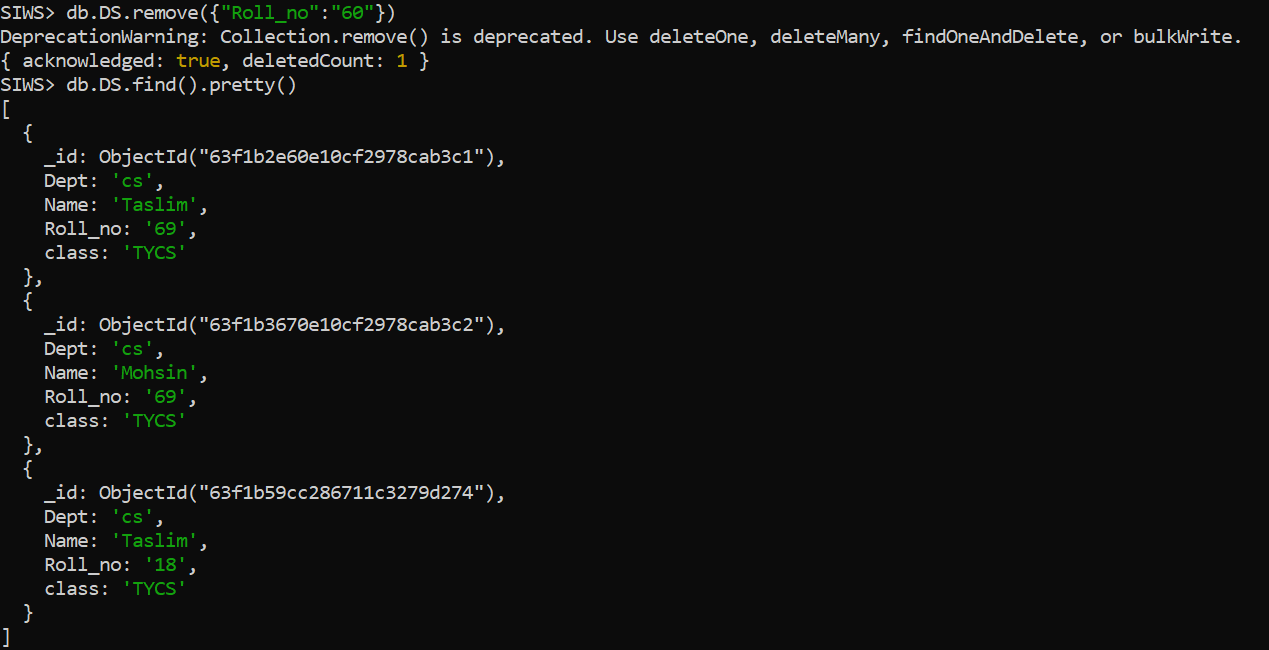
JSON document. In this case, each of the SQL column attributes would be fields and

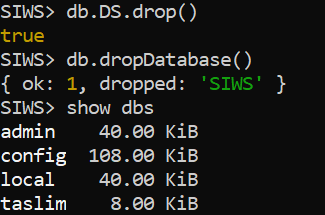
the details of a customer’s record would be the data values associated with each field.

For example: Last\_name: "Jones", First\_name: "Mary", Middle\_initial: "S"

OUTPUT:







**PRACTICAL NO:- 2**

**Data curation and management for Large-scale Data system**

**AIM:-** Practical of Data collection, Data curation and management for Large-scale Data system (such as MongoDB).

**DESCRIPTION:-** Big data management refers to the organisation, administration and governance of large volumes of unstructured and structured data. A high level of data quality and accessibility for business intelligence and big data analytics applications is the aim of big data management.

**PROGRAM:-**

To create a database in MongoDB

Use: DATABASE\_NAME

If you want to check your databases list

Use: show dbs.

create a collection using MongoDB.

Use: db.createCollection(name, options)

To Insert document in MongoDB collection.

Use: db.COLLECTION\_NAME.insert(document)

To query document from MongoDB collection.

Use: db.COLLECTION\_NAME.find()

To update document

Use: db.COLLECTION\_NAME.insert({name},{$set:{new naem}})

To delete collection: To drop Collection

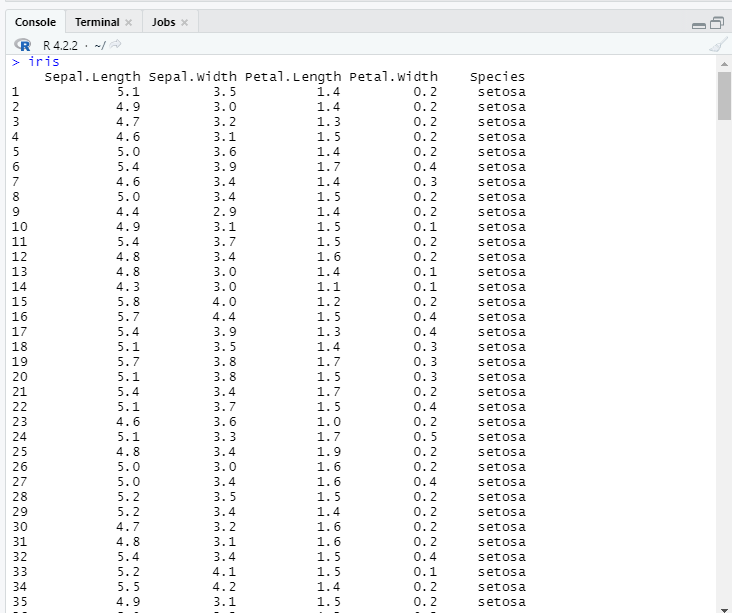
Use: db.COLLECTION\_NAME.remove()

To drop DB

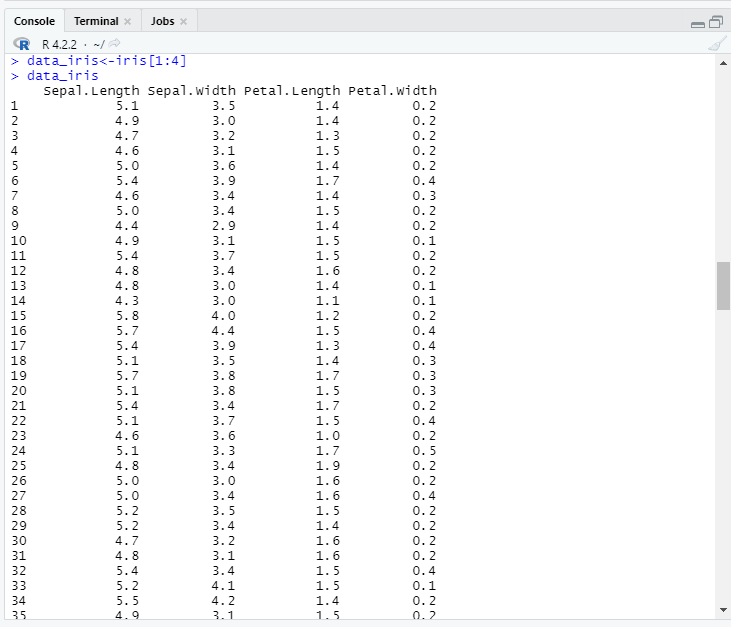
Use: db.COLLECTION\_NAME.drop()

Outputs:-

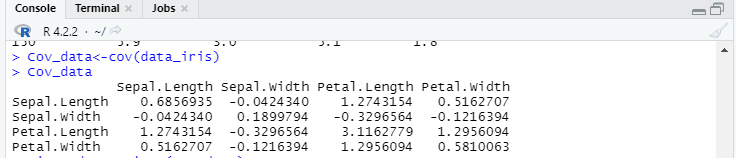
**1.**

****

**2.**

****

**3.**

****

**PRACTICAL NO:- 3**

**Principal Component Analysis**

**AIM:-** Practical of Principal Component Analysis(PCA).

**DESCRIPTION:-** Principal component analysis is used to extract the important information from a multivariate data table and to express this information as a set of few new variables called principal components. These new variables correspond to a linear combination of the originals.

**PROGRAM:-**

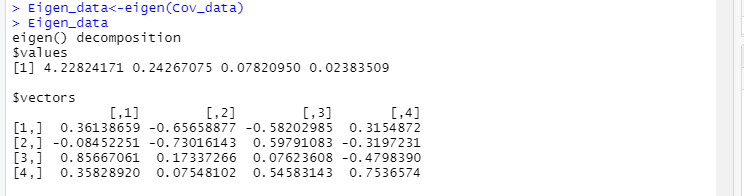
1. **Iris**
2. **data\_iris<-iris[1:4]**

**data\_iris**

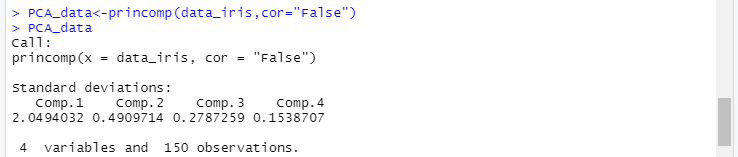
1. **Cov\_data<-cov(data\_iris)**

**Cov\_data**

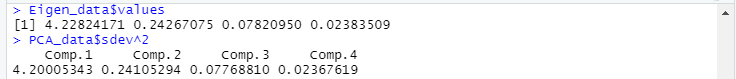
**4.**

****

**5.**

****

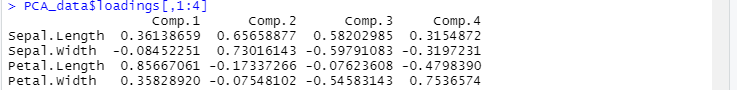
**6.**

****

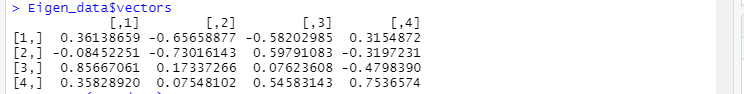
**7.**

****

**8.**

****

**9.**

****

1. **Eigen\_data<-eigen(Cov\_data)**

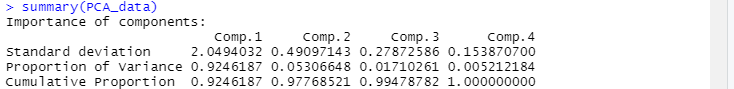
**Eigen\_data**

1. **PCA\_data<-princomp(data\_iris,cor="False")**

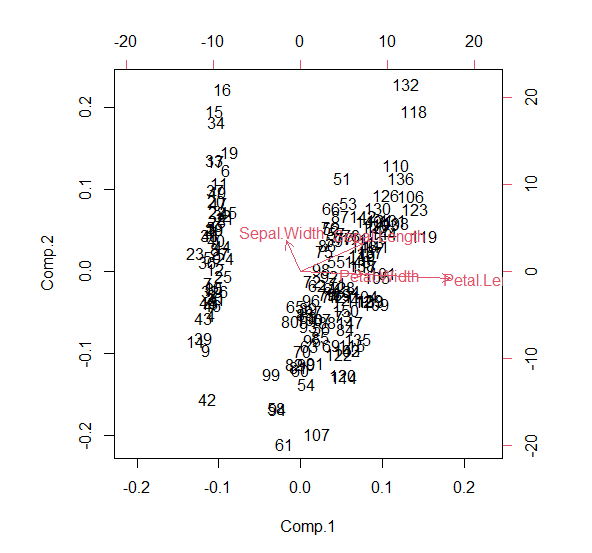
**PCA\_data**

1. **Eigen\_data$values**
2. **PCA\_data$sdev^2**
3. **PCA\_data$loadings[,1:4]**
4. **Eigen\_data$vectors**

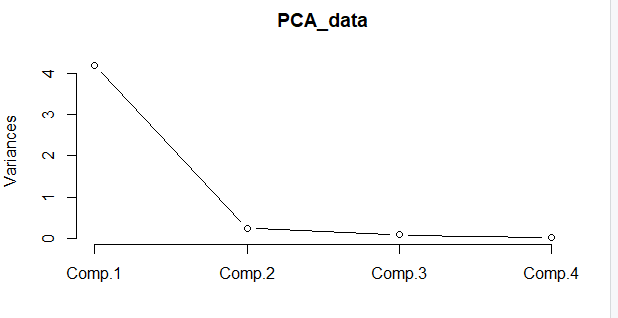
**10.**

****

**11.**

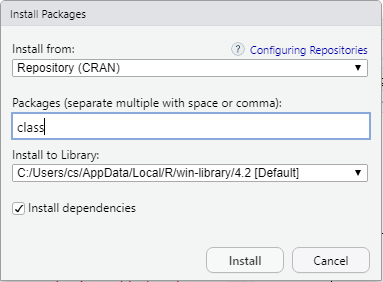
****

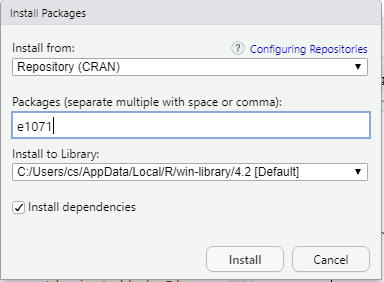
**12.**

****

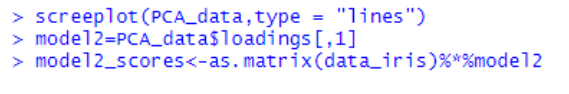
1. **summary(PCA\_data)**
2. **biplot(PCA\_data)**
3. **screeplot(PCA\_data,type = "lines")**

**13.**

****

****

**14.**

****

1. **Install Following 2 packages.**

**1) class**

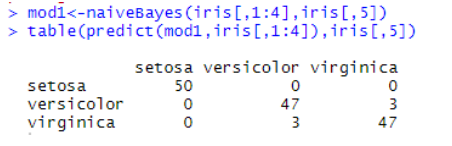
**2) e1071**

1. **screeplot(PCA\_data,type = "lines")**

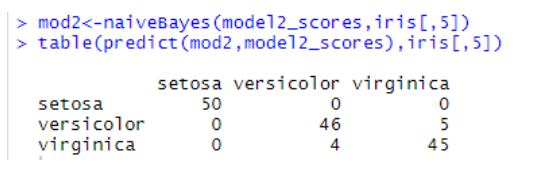
**model2=PCA\_data$loadings[,1]**

**model2\_scores<-as.matrix(data\_iris)%\*%model2**

**15.**

****

**16.**

****

1. **mod1<-naiveBayes(iris[,1:4],iris[,5])**

**table(predict(mod1,iris[,1:4]),iris[,5]**

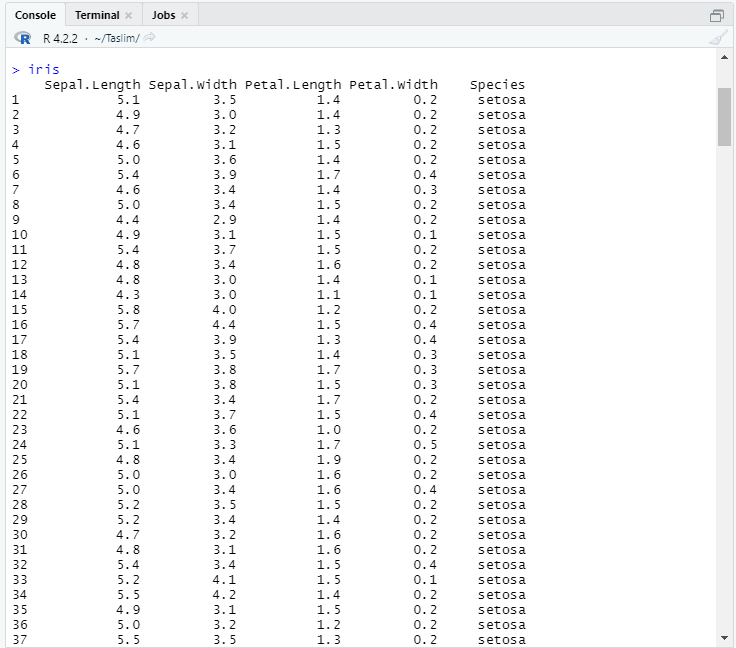
1. **mod2<-naiveBayes(model2\_scores,iris[,5])**

**table(predict(mod2,model2\_scores),iris[,5])**

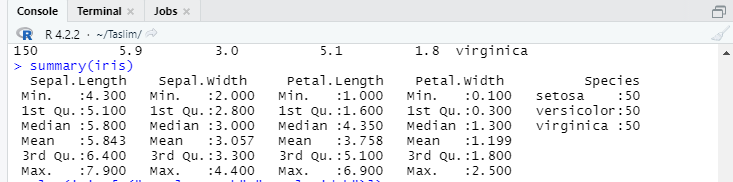
**CONCLUSION:-** The above program has been executed successfully.

**Outputs:-**

**1.**

****

**2.**

****

**PRACTICAL NO:- 4**

**Data Clustering Using Clustering Algorithm**

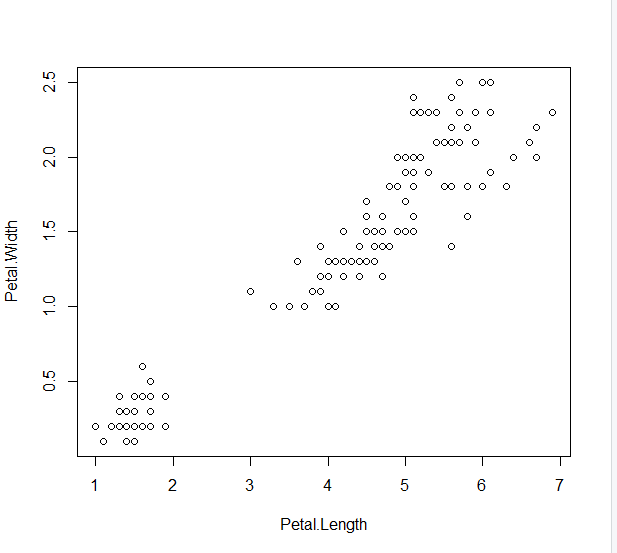
**AIM:-** Perform the data clustering using clustering algorithm.

**DESCRIPTION:-** Data clustering is the most commonly used clustering algorithm. It's a centroid-based algorithm and the simplest unsupervised learning algorithm. This algorithm tries to minimize the variance of data points within a cluster.

**PROGRAM:-**

1. **Iris**
2. **summary(iris)**

**3.**

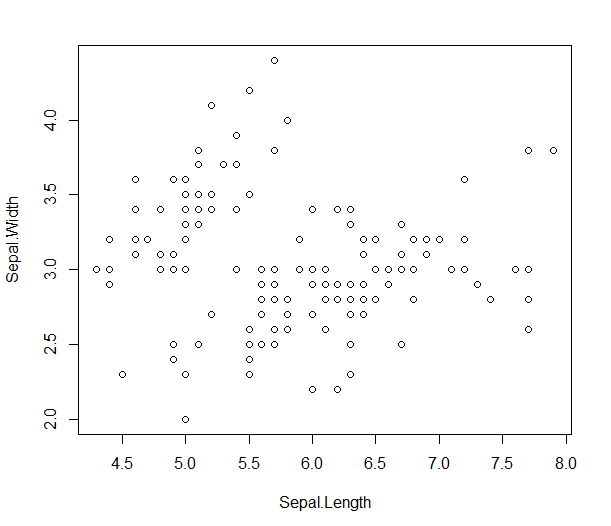
****

**4.**

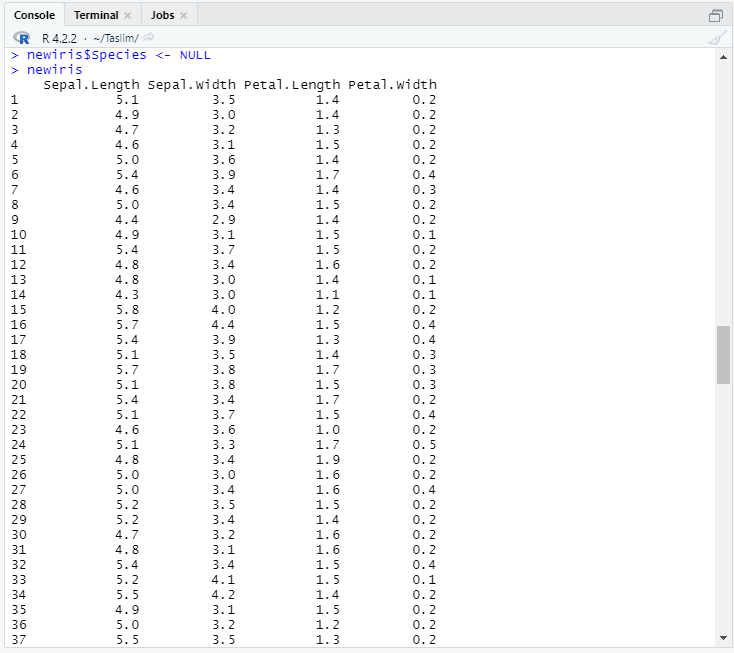
****

1. **plot(iris [c("Petal.Length","Petal.Width")])**
2. **plot(iris [c("Sepal.Length","Sepal.Width")])**

**5.**

****

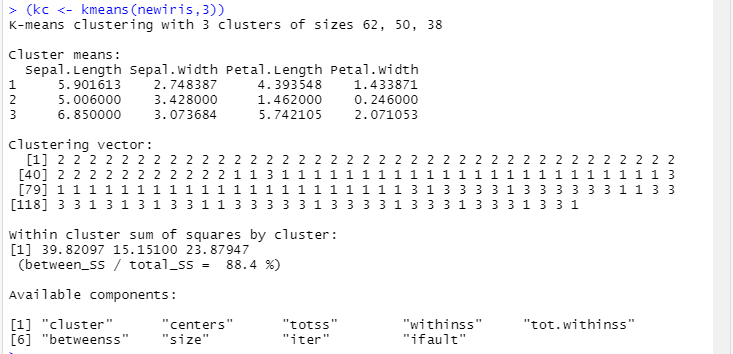
**6.**

****

1. **newiris <- iris**
2. **newiris$Species <- NULL**

**newiris**

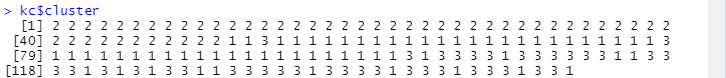
**7.**

****

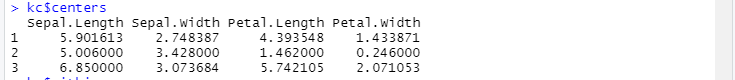
**8.**

****

**9.**

****

**10.**

****

**11.**

****

**12.**

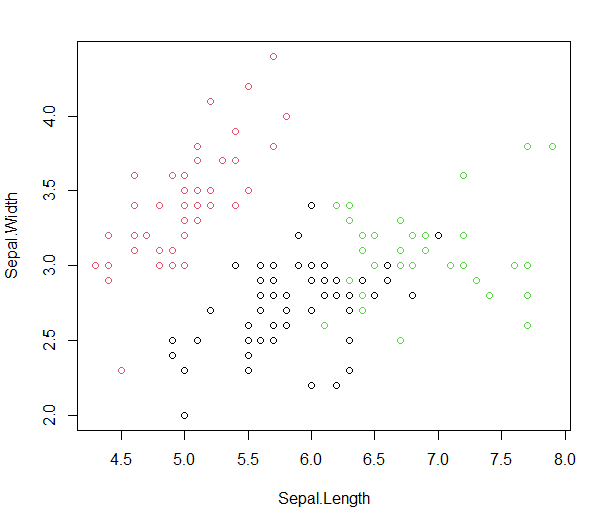
****

**13.**

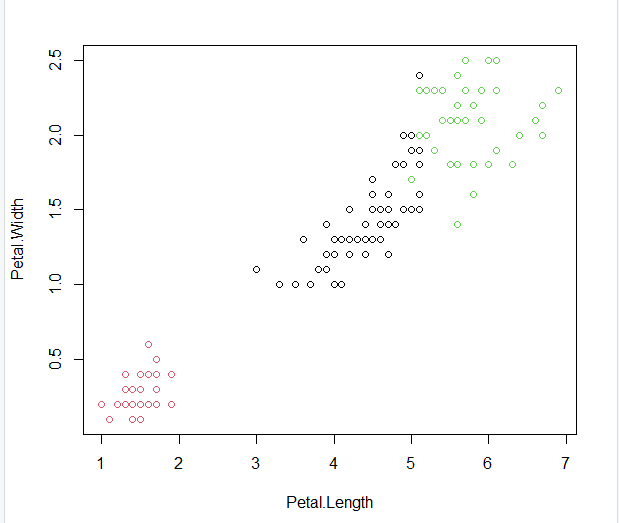
****

1. **(kc <- kmeans(newiris,3))**
2. **kc$size**
3. **kc$cluster**
4. **kc$centers**
5. **kc$withinss**
6. **kc$betweenss**
7. **table(iris$Species,kc$cluster)**

**14.**

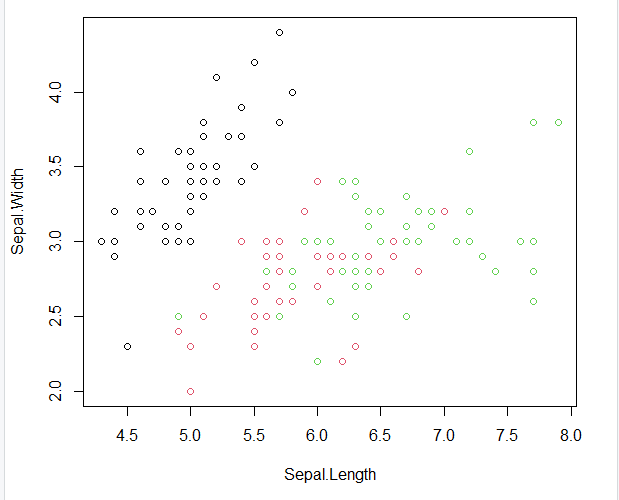
****

**15.**

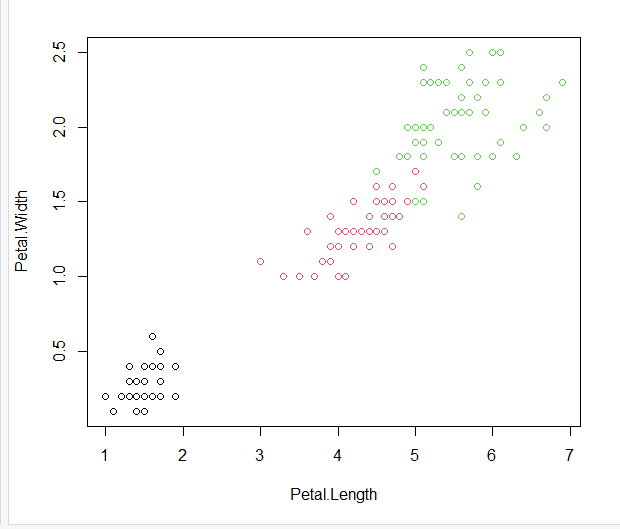
****

1. **plot(newiris[c("Sepal.Length","Sepal.Width")],col=kc$cluster)**
2. **plot(newiris[c("Petal.Length","Petal.Width")],col=kc$cluster)**

**16.**

****

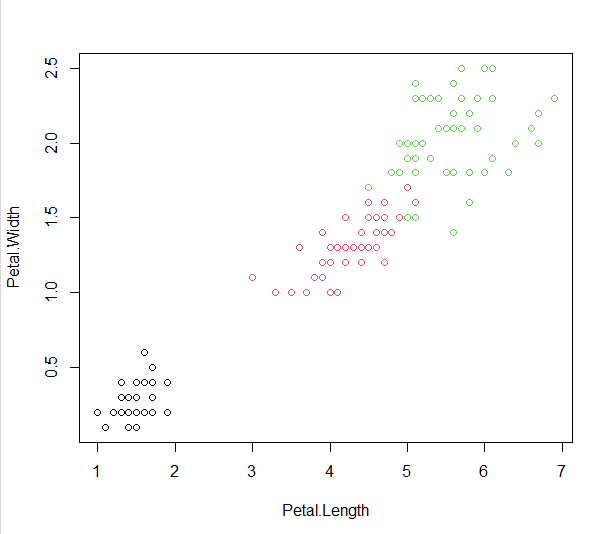
**17.**

****

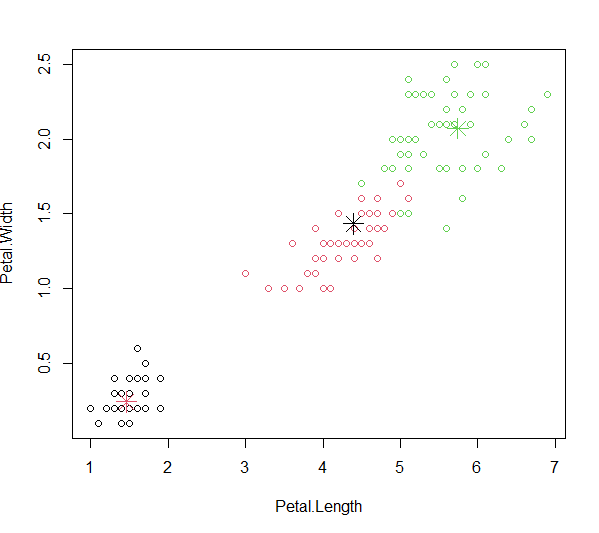
**16. plot(newiris[c("Sepal.Length","Sepal.Width")],col=iris$Species)**

**17. plot(newiris[c("Petal.Length","Petal.Width")],col=iris$Species)**

**18.**

****

**19.**

****

**+**

1. **points(kc$centers[,c("Sepal.Length","Sepal.Width")],col=1:3,pch=8,cex=2)**
2. **points(kc$centers[,c("Petal.Length","Petal.Width")],col=1:3,pch=8,cex=2)**

**CONCLUSION:-** The above program has been executed successfully.

**Outputs:-**

**1.**

****

**2.**

****

**3.**

****

**4.**

****

**5.**

****

**PRACTICAL NO:- 5**

**Time – Series Forecasting**

**AIM:-** Practical of Time – series forecasting.

**DESCRIPTION:-** Time series data analysis is increasingly important due to the massive production of such data through the internet of things, the digitalization of healthcare, and the rise of smart cities.

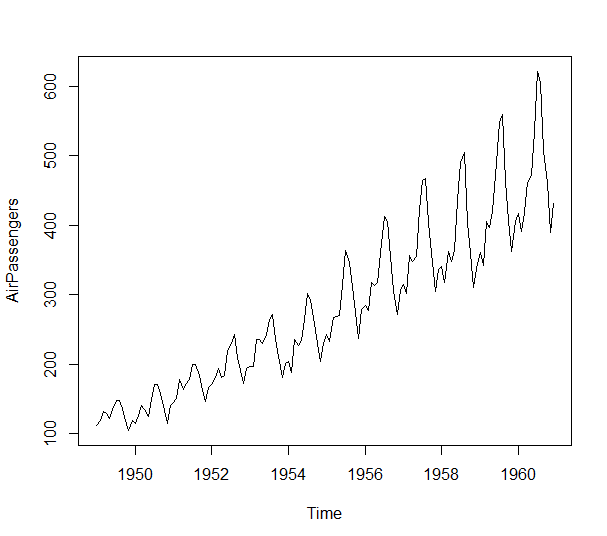
**PROGRAM:-**

1. **data("AirPassengers")**

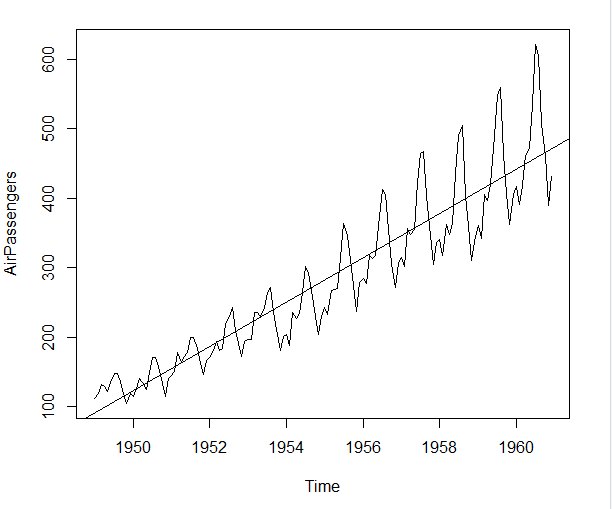
**class(AirPassengers)**

1. **start(AirPassengers)**
2. **end(AirPassengers)**
3. **frequency(AirPassengers)**
4. **summary(AirPassengers)**

**6.**

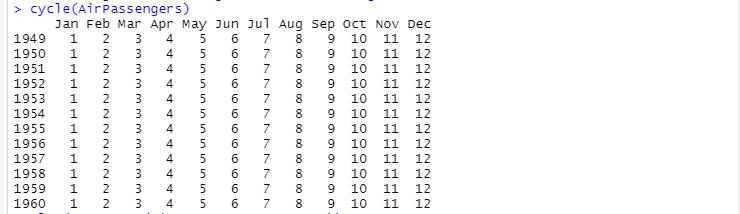
****

**7.**

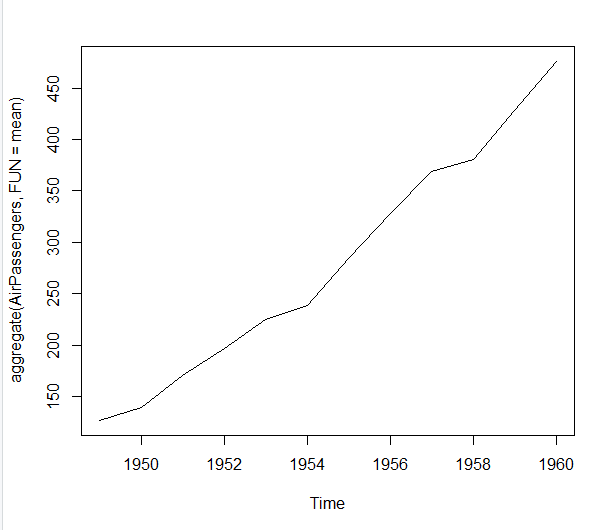
****

1. **plot(AirPassengers)**
2. **abline(reg=lm(AirPassengers~time(AirPassengers)))**

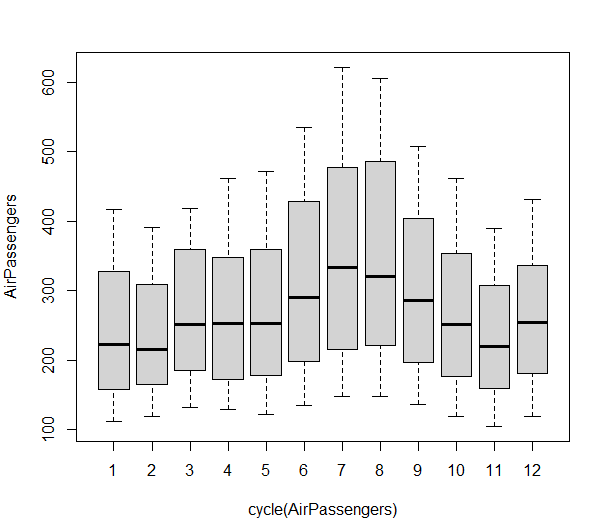
**8.**

****

**9.**

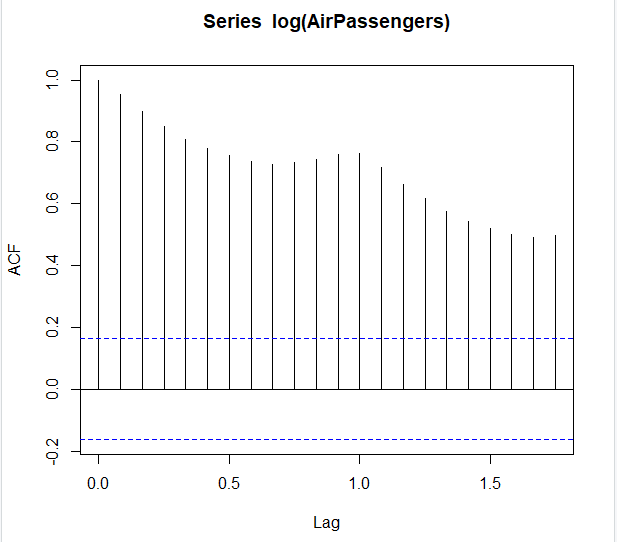
****

**10.**

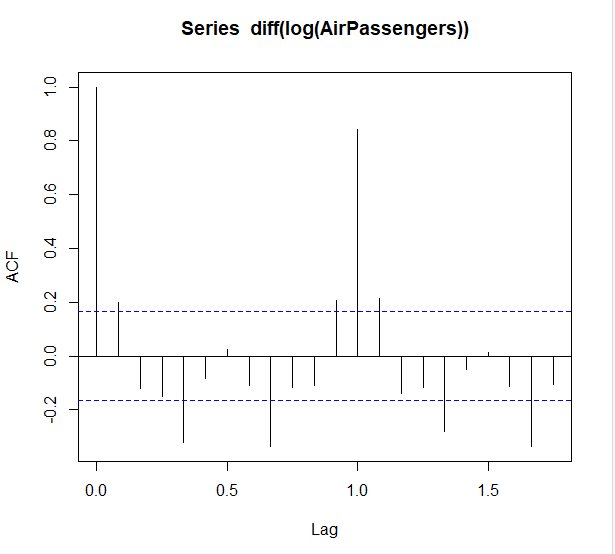
****

1. **cycle(AirPassengers)**
2. **plot(aggregate(AirPassengers,FUN=mean))**
3. **boxplot(AirPassengers~cycle(AirPassengers))**

**11.**

****

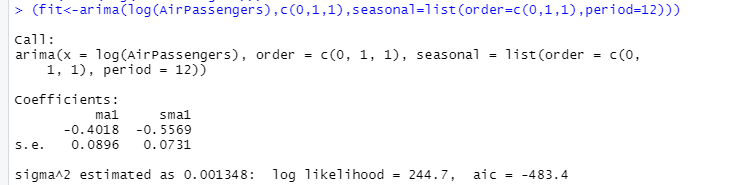
**12.**

****

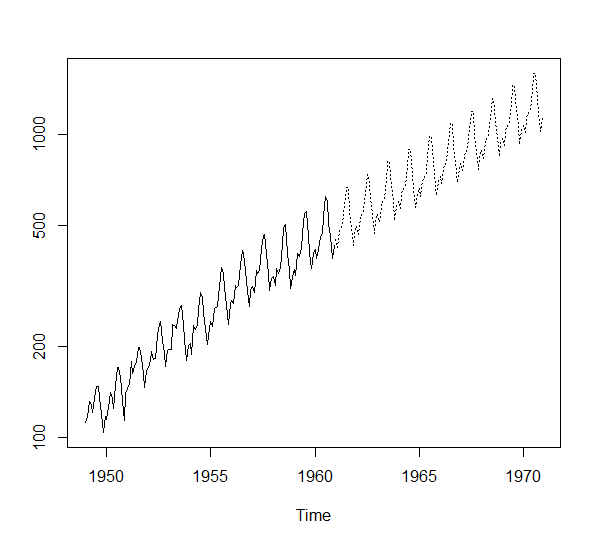
**11. acf(log(AirPassengers))**

**12.acf(diff(log(AirPassengers)))**

**13.**

****

**14.**

****

1. **(fit<-arima(log(AirPassengers),c(0,1,1),seasonal=list(order=c(0,1,1),period=12)))**

**pred<-predict(fit,n.ahead = 10\*12)**

1. **ts.plot(AirPassengers,2.718^pred$pred,log="y",lty=c(1,3))**

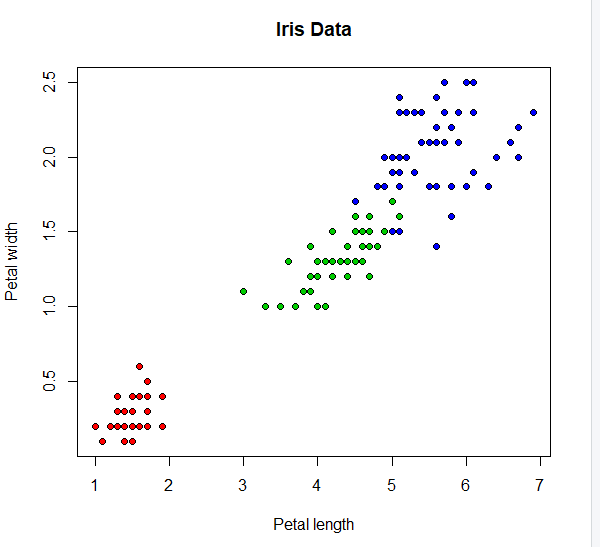
**CONCLUSION:-** The above program has been executed successfully.

**Output:-**

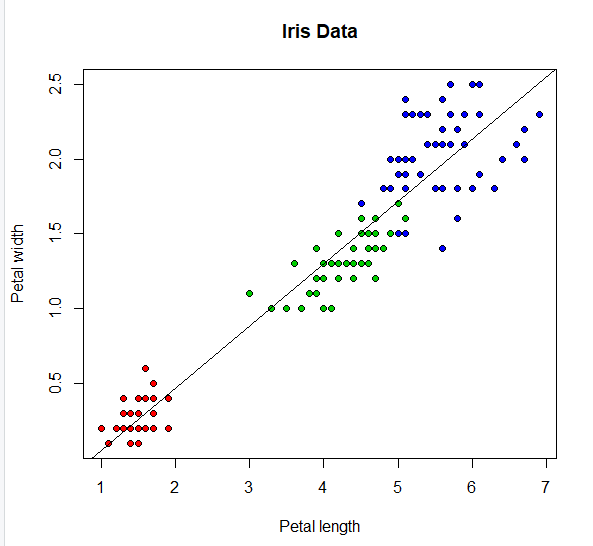
**1.**

****

**2.**

****

**3**

****

**PRACTICAL NO:- 6**

**Simple And Multiple Linear Regression**

**AIM:-** Practical of Simple and Multiple Linear Regression.

**DESCRIPTION:-** What's the difference between simple and multiple linear regression techniques? As noted above, the simple linear method measures one independent variable against one dependent variable. The multiple linear technique is used when there are at least two independent variables against the one dependent variable.

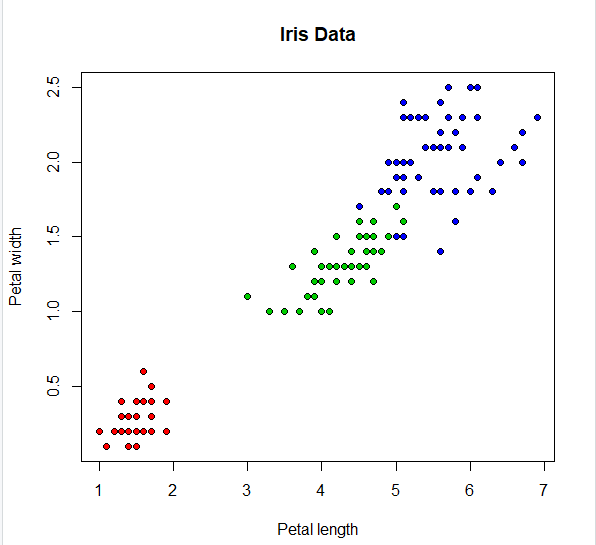
**PROGRAM:-**

1. **lsfit(iris$Petal.Length, iris$Petal.Width)$coefficients**
2. **plot(iris$Petal.Length,iris$Petal.Width,pch=21,bg=c("red","green3","blue")[unclass(iris$Species)],main="IrisData",xlab="Petal length",ylab="Petal width")**
3. **abline(lsfit(iris$Petal.Length,iris$Petal.Width)$coefficients, col="black")**

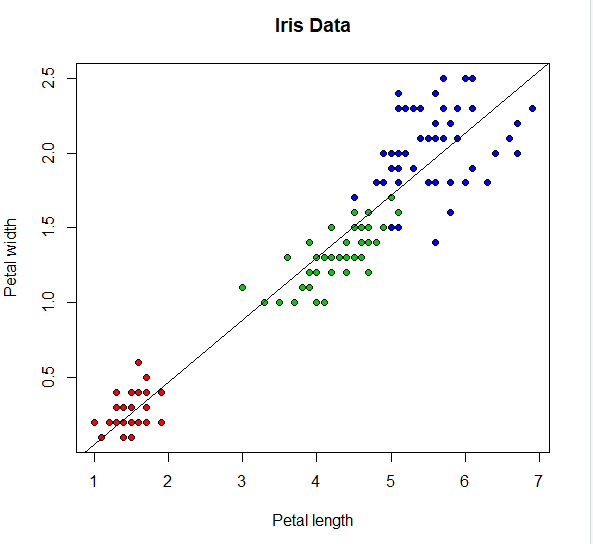
**4.**

****

**5.**

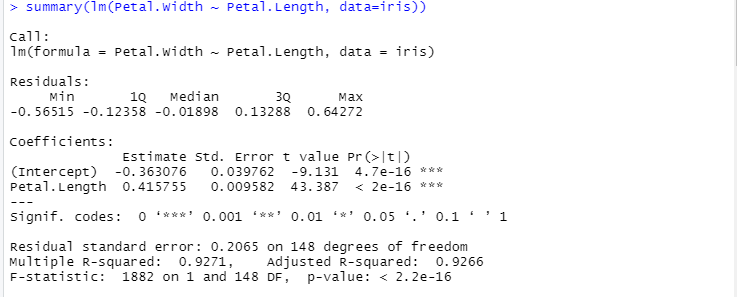
****

**6.**

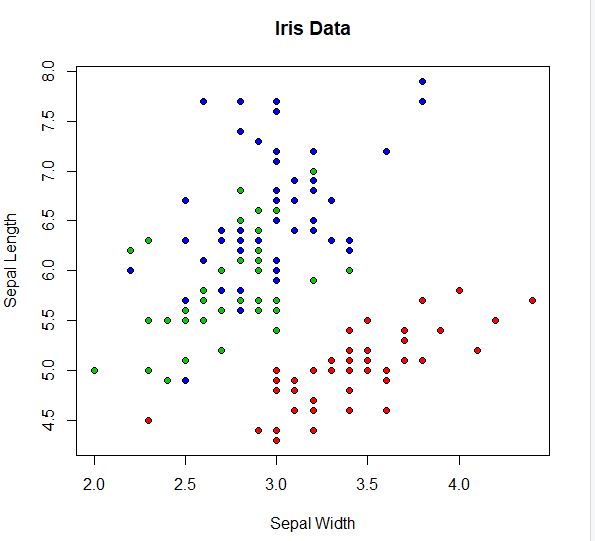
****

1. **lm(Petal.Width ~ Petal.Length, data=iris)$coefficients**
2. **plot(iris$Petal.Length, iris$Petal.Width, pch=21, bg=c("red","green3","blue")[unclass(iris$Species)], main="Iris Data", xlab="Petal length", ylab="Petal width")**
3. **abline(lm(Petal.Width ~ Petal.Length, data=iris)$coefficients, col="black")**

**7.**

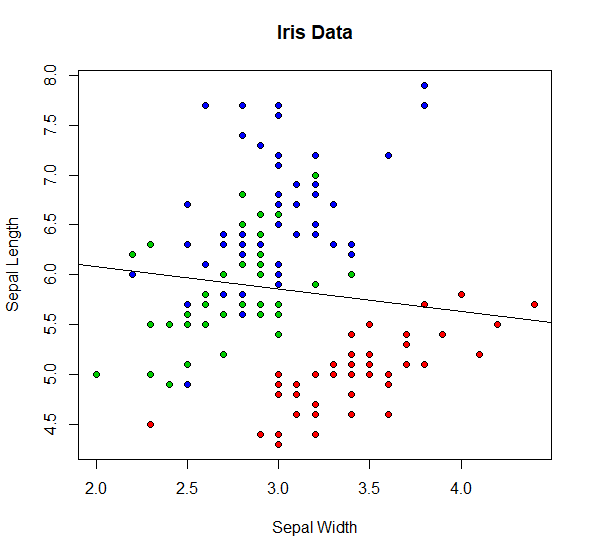
****

**8.**

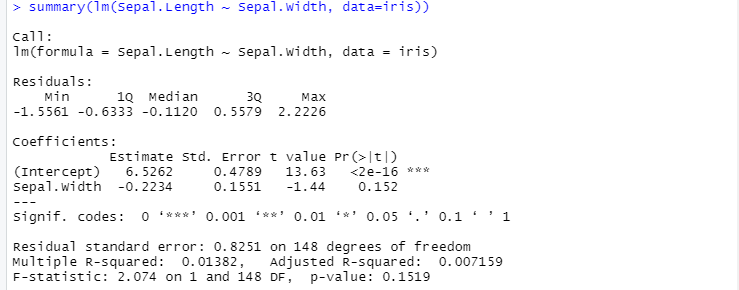
****

1. **summary(lm(Petal.Width ~ Petal.Length, data=iris))**
2. **plot(iris$Sepal.Width, iris$Sepal.Length, pch=21, bg=c("red","green3","blue")[unclass(iris$Species)], main="Iris Data", xlab="Sepal Width", ylab="Sepal Length")**

**9.**

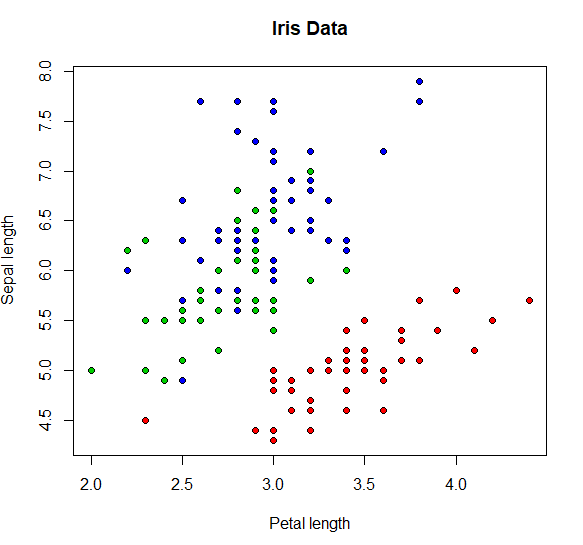
****

**10.**

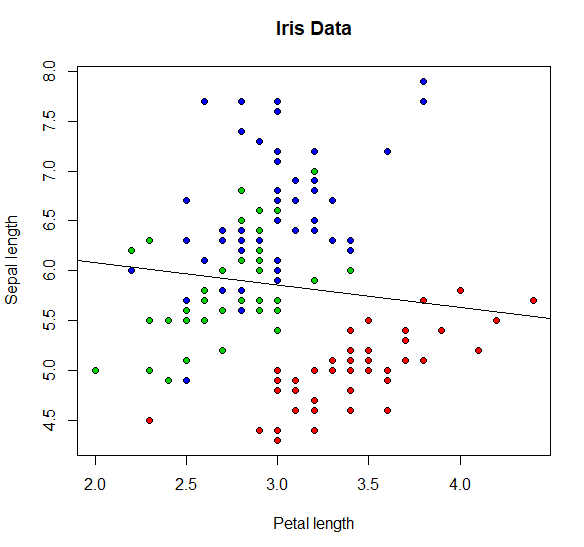
****

1. **abline(lm(Sepal.Length ~ Sepal.Width, data=iris)$coefficients, col="black")**
2. **summary(lm(Sepal.Length ~ Sepal.Width, data=iris))**

**11.**

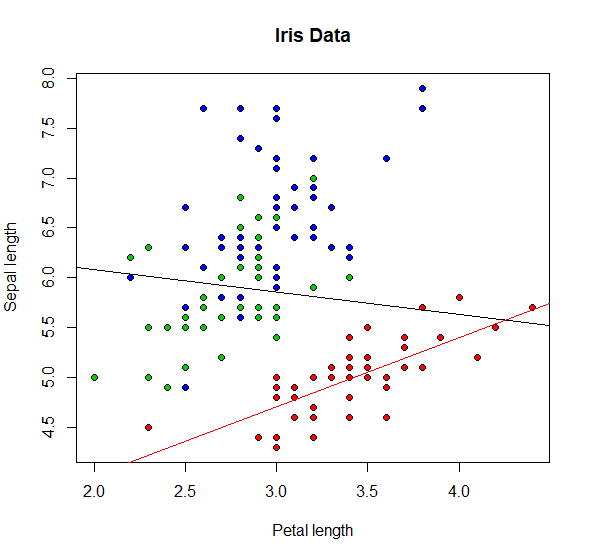
****

**12.**

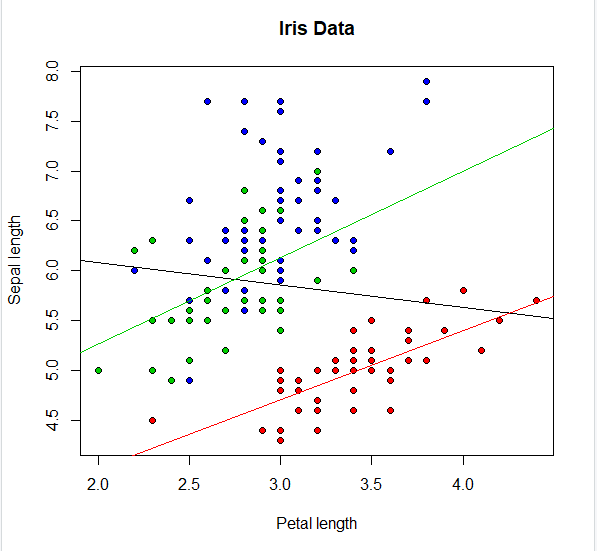
****

1. **plot(iris$Sepal.Width, iris$Sepal.Length, pch=21, bg=c("red","green3","blue")[unclass(iris$Species)], main="Iris Data", xlab="Petal length", ylab="Sepal length")**
2. **abline(lm(Sepal.Length ~ Sepal.Width, data=iris)$coefficients, col="black")**

**13.**

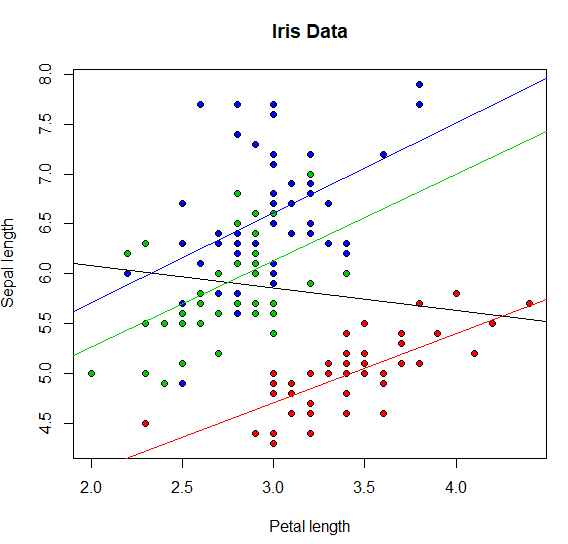
****

**14.**

****

1. **abline(lm(Sepal.Length~Sepal.Width,data=iris[which(iris$Species=="setosa"),])$coefficients, col="red"z**
2. **abline(lm(Sepal.Length~Sepal.Width,data=iris[which(iris$Species=="versicolor"),])$coefficients, col="green3")**

**15.**

****

**16.**

****

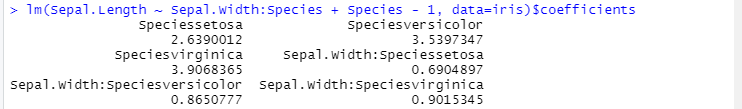
**17.**

****

**18.**

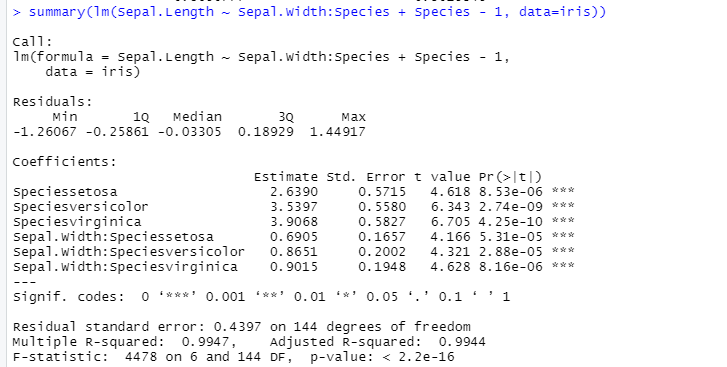
****

**19.**

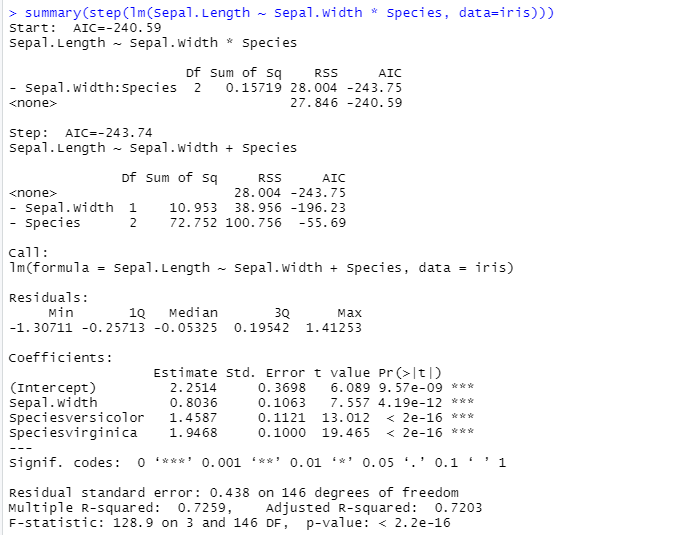
****

1. **abline(lm(Sepal.Length~Sepal.Width,data=iris[which(iris$Species=="virginica"),])$coefficients, col="blue")**
2. **lm(Sepal.Length~Sepal.Width,data=iris[which(iris$Species=="setosa"),])$coefficients**
3. **lm(Sepal.Length~Sepal.Width,data=iris[which(iris$Species=="versicolor"),])$coefficients**
4. **lm(Sepal.Length~Sepal.Width,data=iris[which(iris$Species=="virginica"),])$coefficients**
5. **lm(Sepal.Length ~ Sepal.Width:Species + Species - 1, data=iris)$coefficients**

**20.**

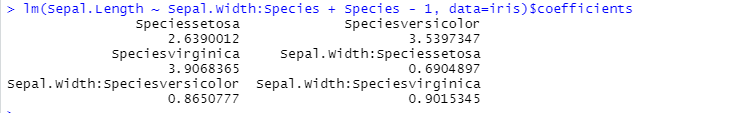
****

**21.**

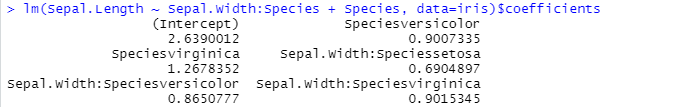
****

1. **summary(lm(Sepal.Length ~ Sepal.Width:Species + Species - 1, data=iris)**
2. **summary(step(lm(Sepal.Length ~ Sepal.Width \* Species, data=iris)))**

**22.**

****

**23.**

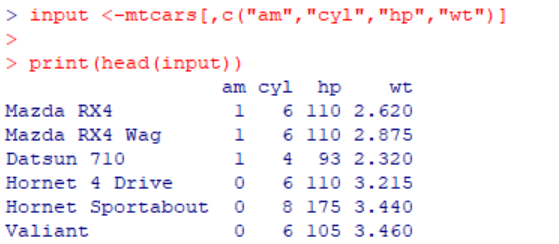
****

1. **lm(Sepal.Length ~ Sepal.Width:Species + Species - 1, data=iris)$coefficients**
2. **lm(Sepal.Length ~ Sepal.Width:Species + Species, data=iris)$coefficients**

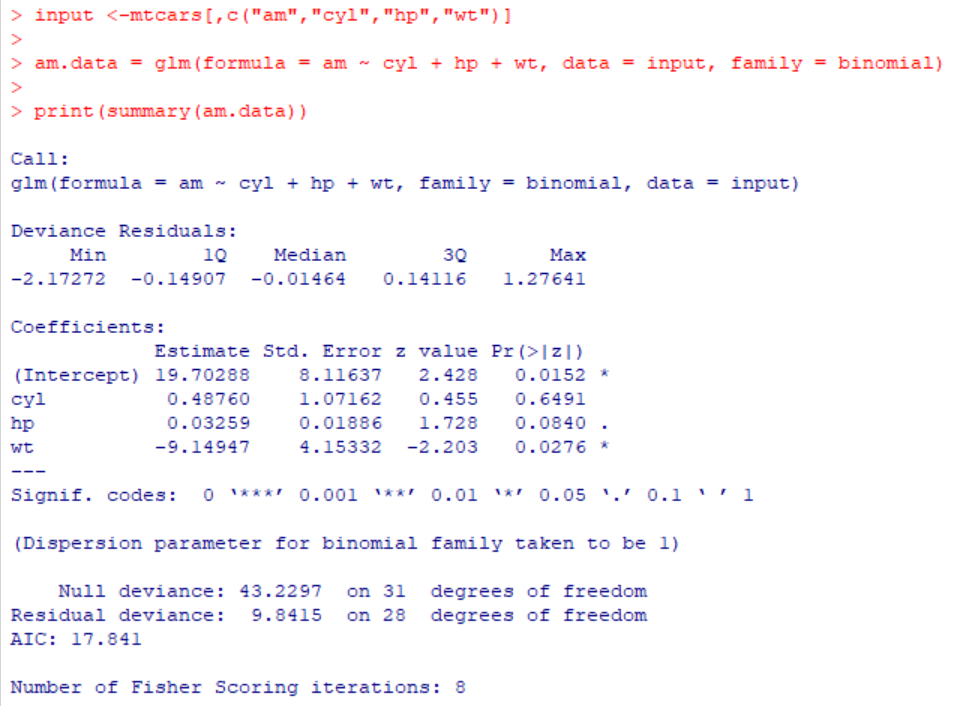
**CONCLUSION:-** The above program has been executed successfully.

**Outputs:-**

**1.**

****

**2.**

****

**PRACTICAL NO:- 7**

**Logistics Regression**

**AIM:-** Practical of Logistics Regression

**DESCRIPTION:-** Logistic regression is a data analysis technique that uses mathematics to find the relationships between two data factors. It then uses this relationship to predict the value of one of those factors based on the other. The prediction usually has a finite number of outcomes, like yes or no.

**PROGRAM:-**

1. **input <-mtcars[,c("am","cyl","hp","wt")]**

**print(head(input))**

1. **input <-mtcars[,c("am","cyl","hp","wt")]**

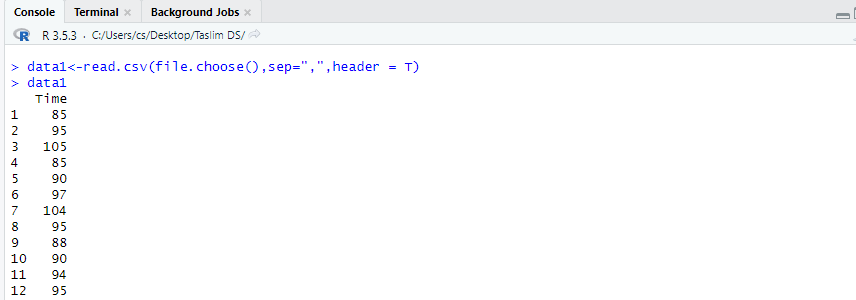
**am.data = glm(formula = am ~ cyl + hp + wt, data = input, family = binomial)**

**print(summary(am.data))**

**CONCLUSION:-** The above program has been executed successfully.

**Outputs:-**

**1.**

****

**PRACTICAL NO:- 8**

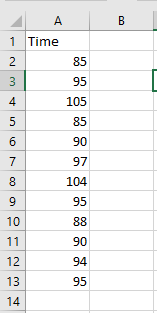
**Hypothesis Testing**

**AIM:- Practical of Hypothesis Testing.**

**DESCRIPTION:- Hypothesis testing is the process used to evaluate the strength of evidence from the sample and provides a framework for making determinations related to the population, ie, it provides a method for understanding how reliably one can extrapolate observed findings in a sample under study to the larger population.**

**PROGRAM:-**

**File:1 sample t test Hypo.csv**

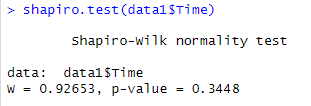
****

1. **"test for normal distribution"**

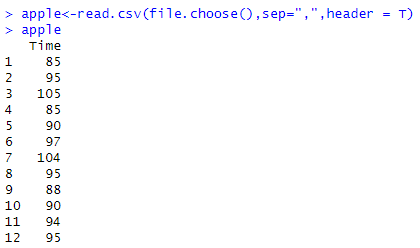
**data1<-read.csv(file.choose(),sep=",",header = T)**

****

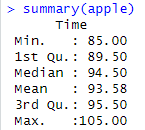
**2.**

****

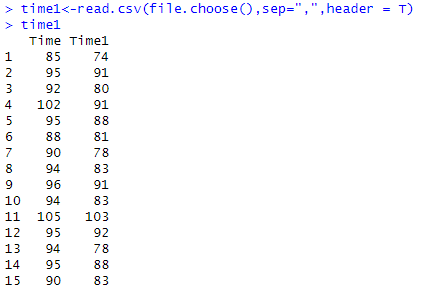
**3.**

****

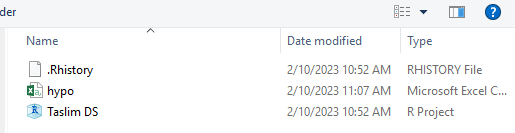
**4.**

****

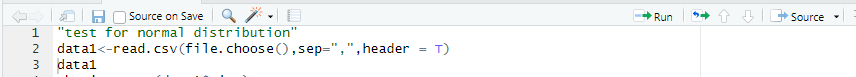
**5.**

****

**Select csv File**

****

**data1**

****

1. **shapiro.test(data1$Time)**
2. **File: ONE SAMPLE t Test Hypo.csv**

**"one sample t test"**

**apple<-read.csv(file.choose(),sep=",",header = T)**

**apple**

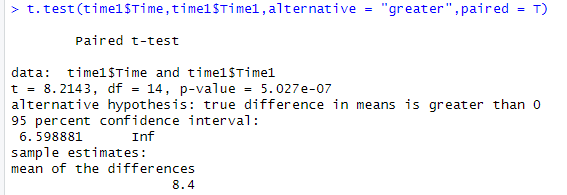
1. **summary(apple)**
2. **File: PAIRED t TEST.csv**

**"paired t test"**

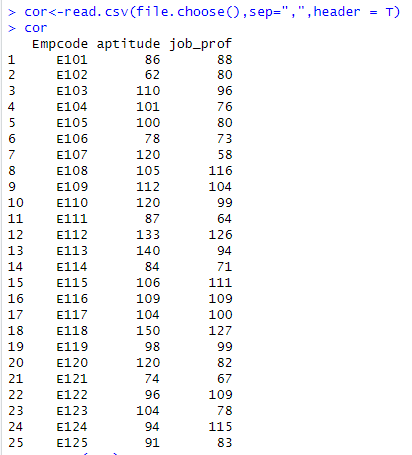
**time1<-read.csv(file.choose(),sep=",",header = T)**

**time1**

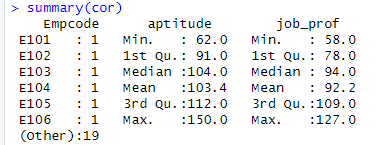
**6.**

****

**7.**

****

**8.**

****

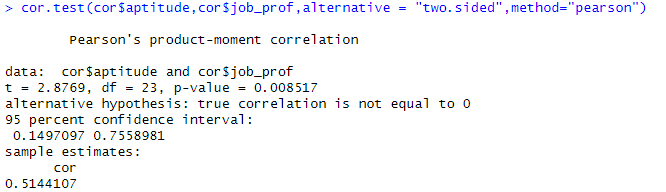
1. **t.test(time1$time\_before,time1$time\_after,alternative = "greater",paired = T)**
2. **File: t test for correlation JOBPROF.csv**

**"t test for correlation " cor<-read.csv(file.choose(),sep=",",header = T)**

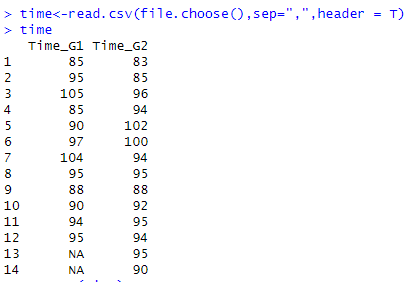
**Cor**

1. **summary(cor)**

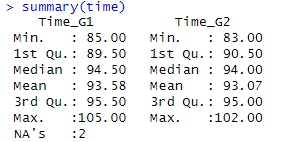
**9.**

****

**10.**

****

**11.**

****

1. **cor.test(cor$aptitude,cor$job\_prof,alternative = "two.sided",method="pearson")**
2. **File: INDEPENDENT SAMPLES t TEST.csv**

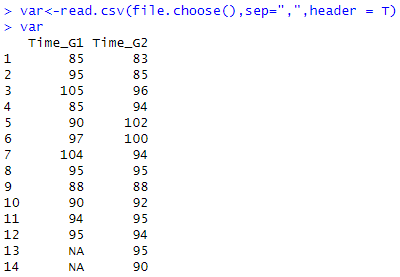
**"independent t test "**

**time<-read.csv(file.choose(),sep=",",header = T)**

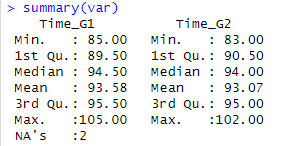
**time**

1. **summary(time)**

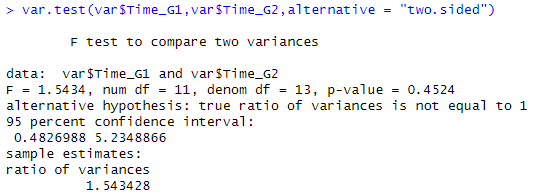
**12.**

****

**13.**

****

**14.**

****

1. **"t test for variance "**

**var<-read.csv(file.choose(),sep=",",header = T)**

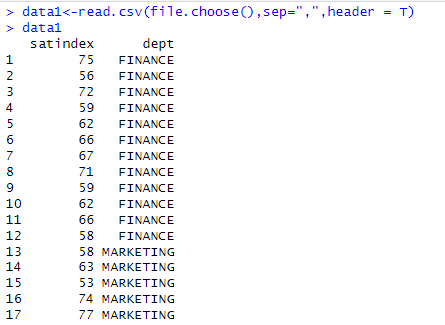
**var**

1. **summary(var)**
2. **var.test(var$Time\_G1,var$Time\_G2,alternative = "two.sided")**

**CONCLUSION:-** The above program has been executed successfully.

**Outputs:-**

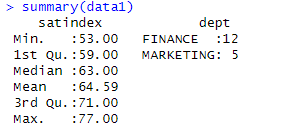
**1.**

****

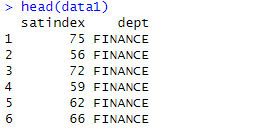
**2.**

****

**3.**

****

**4.**

****

**PRACTICAL NO:- 9**

**Analysis of Variance**

**AIM:-** Practical of Analysis of Variance.

**DESCRIPTION:-** Analysis of variance, or ANOVA, is a statistical method that separates observed variance data into different components to use for additional tests. A one-way ANOVA is used for three or more groups of data, to gain information about the relationship between the dependent and independent variables.

**PROGRAM:-**

1. **Code:**

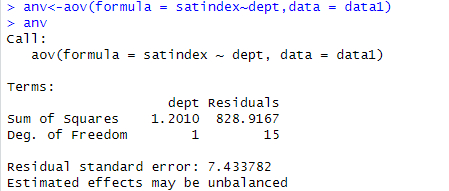
**"One way anova"**

**data1<-read.csv(file.choose(),sep=",",header = T)**

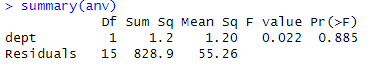
**data1**

1. **names(data1)**
2. **summary(data1)**
3. **head(data1)**

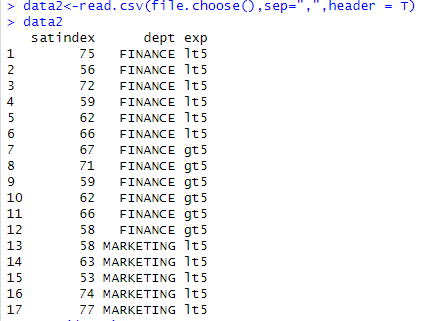
**5.**

****

**6.**

****

**7.**

****

**8.**

****

1. **anv<-aov(formula = satindex~dept,data = data1)**

**anv**

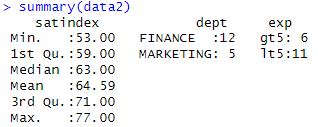
1. **summary(anv)**
2. **"Two way anova"**

**data2<-read.csv(file.choose(),sep=",",header = T)**

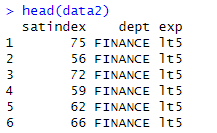
**data2**

1. **names(data2)**

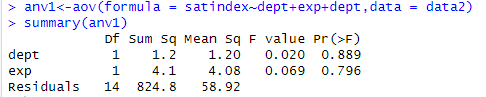
**9.**

****

**10.**

****

**11.**

****

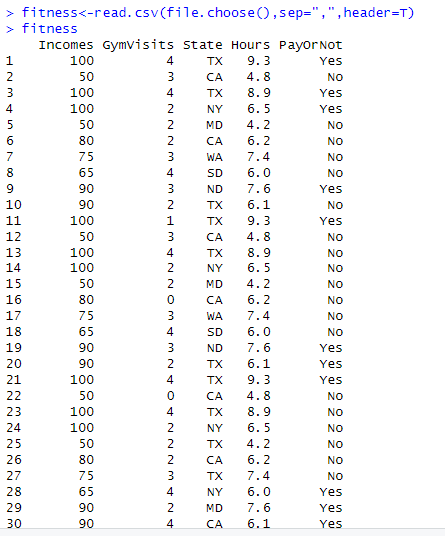
1. **names(data2)**
2. **head(data2)**
3. **anv1<-aov(formula = satindex~dept+exp+dept,data = data2)**

**summary(anv1)**

**CONCLUSION:-** The above program has been executed successfully.

**Outputs:-**

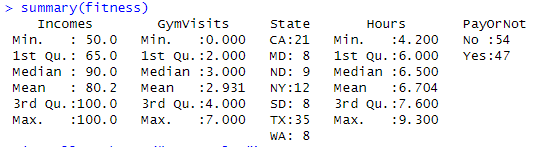
**1.**

****

**2.**

****

**3.**

****

**PRACTICAL NO:- 10**

**Decision Tree**

**AIM:-** Practical of Decision Tree.

**DESCRIPTION:-** A decision tree is one of the supervised machine learning algorithms. This algorithm can be used for regression and classification problems — yet, is mostly used for classification problems. A decision tree follows a set of if-else conditions to visualize the data and classify it according to the conditions.

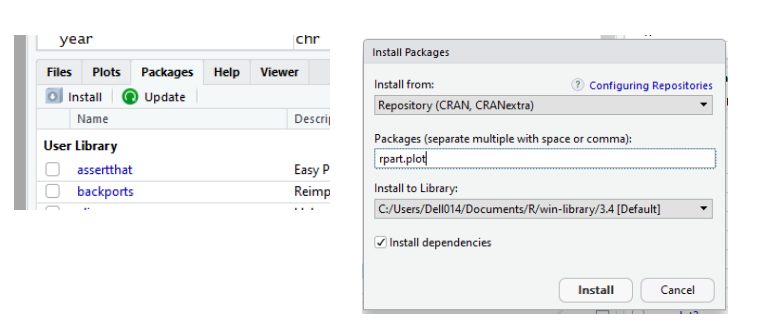
**PROGRAM:-**

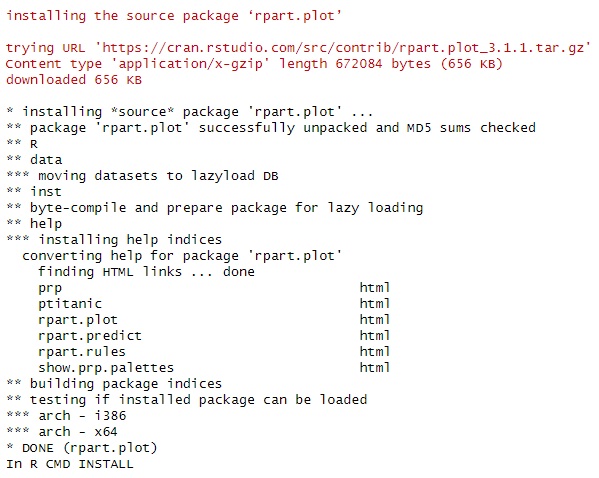
1. **fitness<-read.csv(file.choose(),sep=",",header=T)**

**fitness**

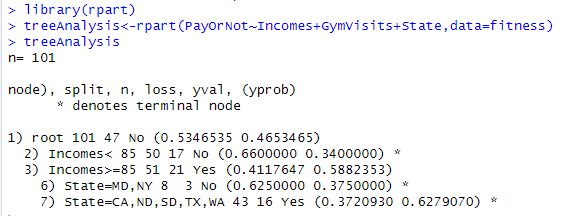
1. **names(fitness)**
2. **summary(fitness)**

**4.**

****

****

**5.**

****

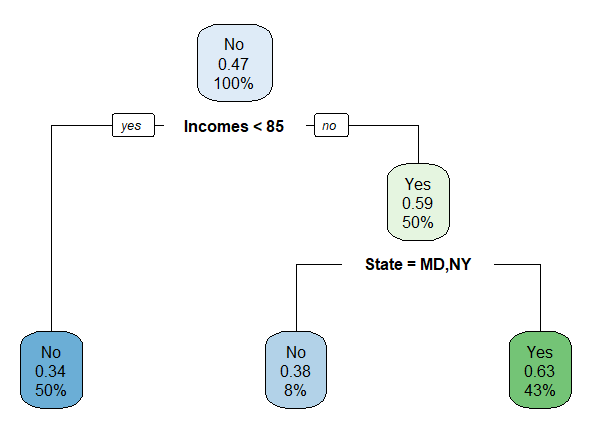
1. **Install library(rpart.plot)**
2. **library(rpart)**

**treeAnalysis<-rpart(PayOrNot~Incomes+GymVisits+State,data=fitness)**

**treeAnalysis**

**6.**

****

****

1. **library("rpart.plot")**

**rpart.plot(treeAnalysis**

**CONCLUSION:-** The above program has been executed successfully.