**Ex NO : INTRODUCTORY COMMANDS IN R**

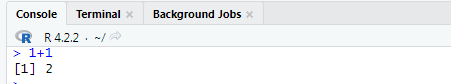
**DATE :**

**AIM :** The aim of the program to study Introductory Commands in R

**BASIC R COMMANDS :**

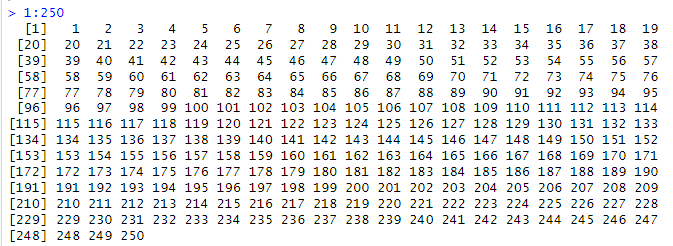
**Cmd 1:**  # Basic Maths - Addition

1+1



**Cmd 2:**  # Printing the Values in Range

1:250



**Cmd 3:**  #Printing String

print("Hello World")



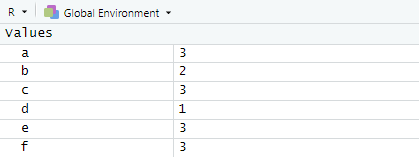
**Cmd 4:**  #Assigning Individual values

d<-1

2->b

c<-f<-e<-3





**Cmd 5:**  #Multiple Values

x <- c(1,2,5,9)

x



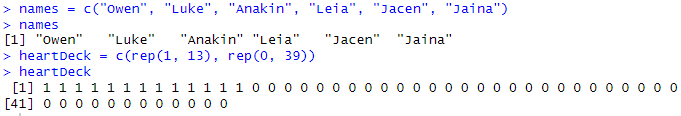
**Cmd 6:**  #Printing Values

names = c("Owen", "Luke", "Anakin", "Leia", "Jacen", "Jaina")

names

heartDeck = c(rep(1, 13), rep(0, 39))

heartDeck



**Cmd 7:**  #SEQUENCES

#Create Sequential Data

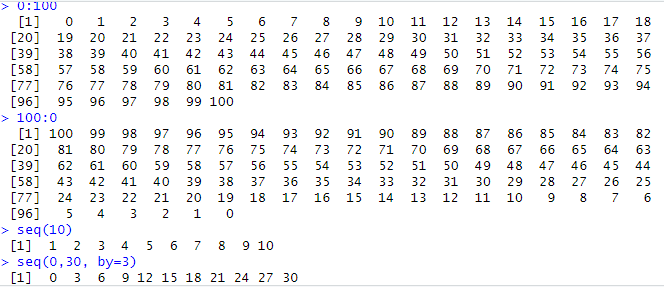
0:100

100:0

seq(10)

seq(0,30, by=3)

seq(30,0, by = - 2)





**Cmd 8:**  #MATHEMATICAL FUNCTIONS

(y <- c(5,1,0,10))

x+y

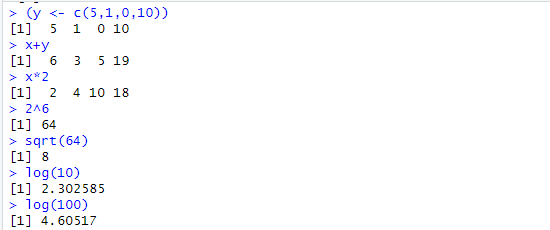
x\*2

2^6

sqrt(64)

log(10)

log(100)



**RESULT :** Thus the basic introductory commands in R has been studied and executed successfully.

**Ex NO : PROGRAM USING DESCRIPTIVE STATISTICS**

**DATE :**

**AIM :** The aim of the program is to perform Descriptive statistics analysis of dataset using R

**PRE- REQUISITES:** **INSTALL : ( FBasics ) Go To Tools - Install this package**

**PROCEDURE :**

**Step 1: Import the Dataset ->CardioGoodFitnss.csv Dataet File in R Studio**

### Step 2 : Use R functions for computing descriptive analysis.

### Step 3: Calculate mean , median , range, max, min , range, standard deviation, quantile, IQR , summary values using any one of the specific column data imported in the dataset

### Step 4: Save and Run all the commands

**PROGRAM CODE :**

data = read.csv(file.choose(),header = TRUE)

mean(data$Age)

median(data$Age)

max(data$Age)

min(data$Age)

range(data$Age)

print(r)

var(data$Age)

sd(data$Age)

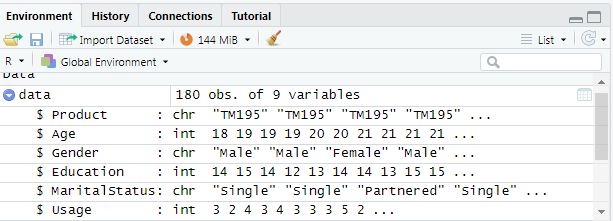
quantile(data$Age)

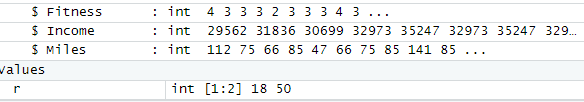
IQR(data$Age)

summary(data$Age)

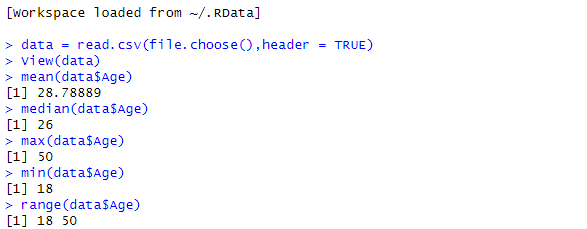
summary(data)

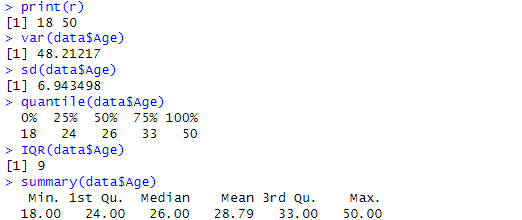
**DATASET VIEW :**

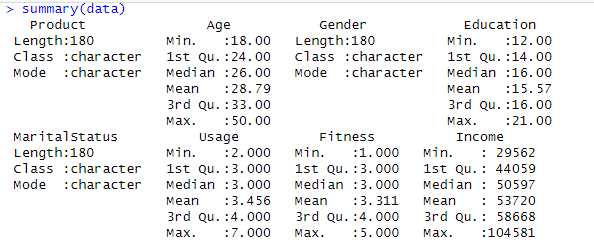


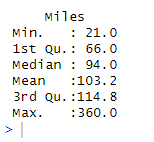


**OUTPUT :**









**RESULT :** Thus the R Program using Descriptive Statistics using dataset has been implemented and executed successfully

**Ex NO : PREPROCESSING ON DATASET MTCARS**

**DATE :**

**AIM :** The aim of the program is to perform Preprocessing of Data in MTCARS Datset using R

**PRE- REQUISITES:** **LOAD : MtCars Dataset Go To Tools - Install this package**

**PROCEDURE :**

**Step 1:** Load the Dataset MtCars in R Studio

### Step 2 : Use R functions for Preprocessing the data present in the dataset.

### Step 3: Preprocessing is done using head, nrow, ncol, head, tail, summary , quantile , var command functions for the imported MtCars dataset

### Step 4: Generate Histogram Graph for the preprocessed Data.

### Step 5: Save and Run all the commands

**PROGRAM CODE :**

data("mtcars")

head(mtcars)

nrow(mtcars)

ncol(mtcars)

head(mtcars)

tail(mtcars)

summary(mtcars)

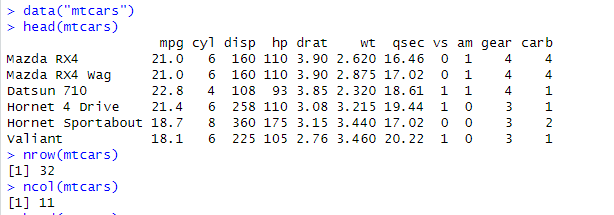
quantile(mtcars$wt)

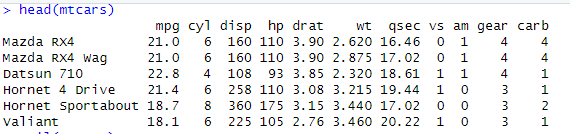
quantile(mtcars$wt, c(.2, .4, .8))

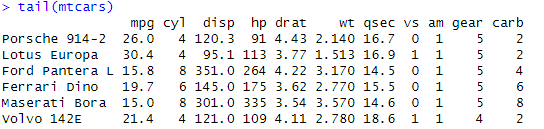
var(mtcars$wt)

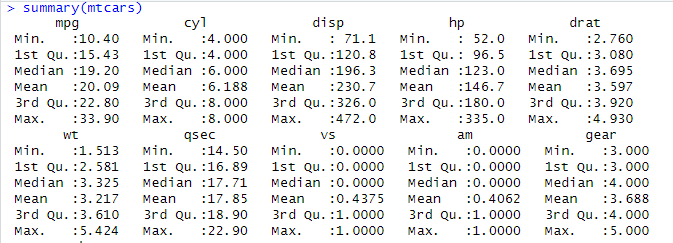
hist(mtcars$hp)

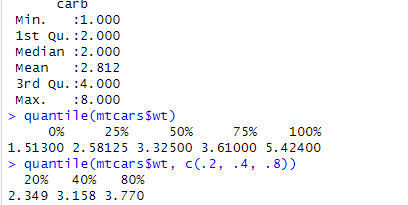
**OUTPUT :**



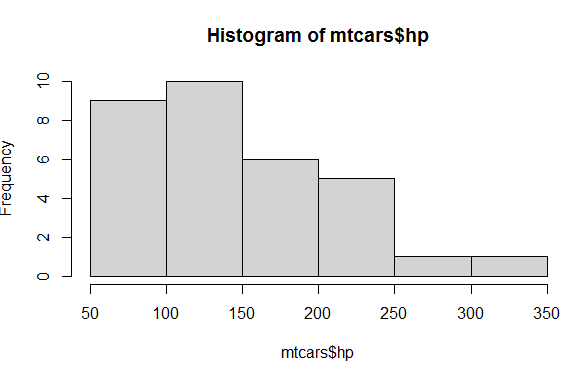












**RESULT :** Thus the R Program for Preprocessing the data using MtCars dataset has been implemented and executed successfully

**Ex NO : ASSOCIATION RULES ON GROCERIES DATASET USING APRIORI**

**DATE : ALGORITHM**

**AIM :** The aim of the program is to use Association rules on Groceries Dataset using APriori Algorithm.

**PREREQUISITES** : INSTALL : Arules, arulesviz, RColorBrewer (Go To Tools and Install)

**PROCEDURE :**

Step 1:Import the Dataset -> Groceries.csv File in R Studio

Step 2 :Load the groceries dataset

Step 3:Inspect the structure of the transactions

Step 4**:** Inspect the structure of the generated rules**.**

Step 5**:** Sort the rules by lift

Step 6: Inspect the top 10 rules

Step 7:Save and Run all the commands

**PROGRAM CODE:**

library(arules)

data = read.csv(file.choose(),header = TRUE)

str(data)

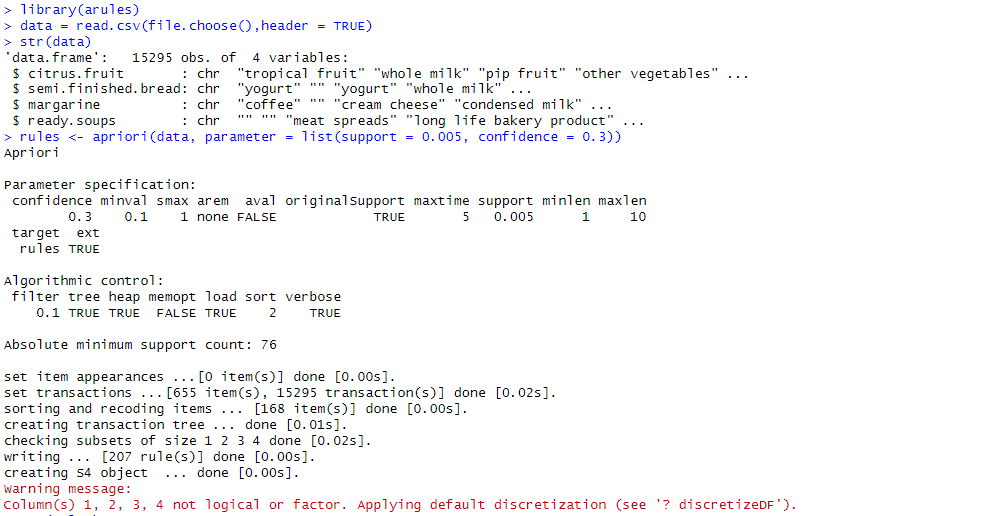
rules <- apriori(data, parameter = list(support = 0.005, confidence = 0.3))

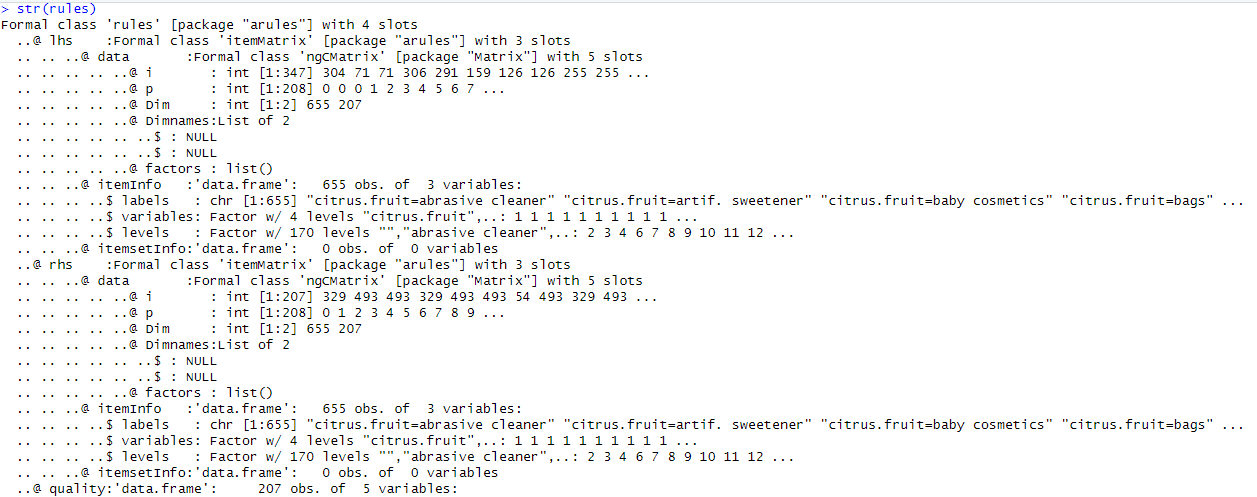
str(rules)

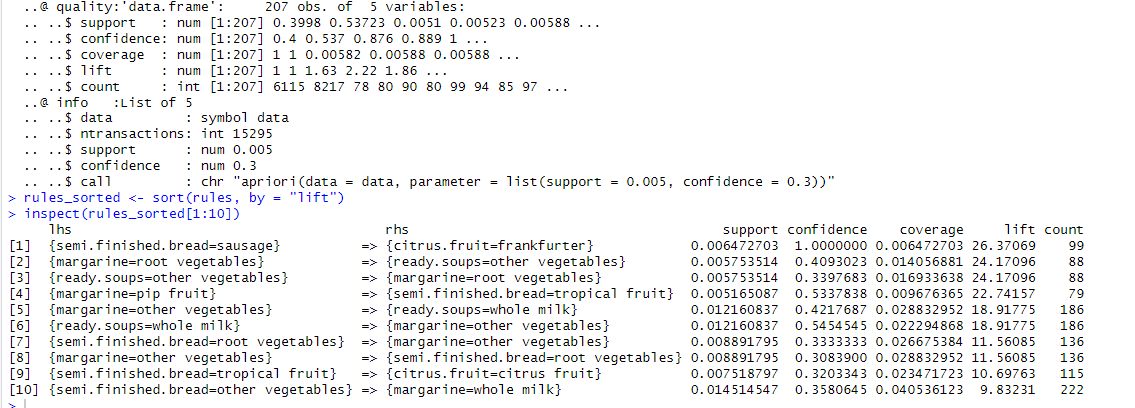
rules\_sorted <- sort(rules, by = "lift")

inspect(rules\_sorted[1:10])

**OUTPUT :**

****

****

****

**RESULT :** Thus the R Program to demonstrate Association rules on Groceries Dataset using Apriori algorithm has been implemented and executed successfully

**Ex NO : CLASSIFICATION RULES PROCESS ON DATASET TITANIC USING**

**DATE : ID3 ALGORITHM**

**AIM :** The aim of the Program is To Demonstrate Classification Rules Process on Titanic dataset using Id3 Algorithm

**PREREQUISITES :** LOAD and INSTALL titanic Package From Tools

**PROCEDURE :**

**Step 1:** Load the titanic dataset

**Step 2:** Create a new data frame that contains only the relevant variables

**Step 3:** Split the data into a training set and a testing set**.**

**Step 4:** Build the decision tree using the ID3 algorithm

**Step 5:** Visualize the decision tree using plot and text function

**Step 6:** Use the decision tree to make predictions on the testing set

**Step 7:** Evaluate the performance of the decision tree using a confusion matrix

**Step 8:** Save and Run all the commands

**PROGRAM CODE :**

install.packages("titanic")

data("titanic\_train")

titanic <- data.frame(Survived = as.factor(titanic\_train$Survived),

Gender = as.factor(titanic\_train$Sex),

Age = as.factor(ifelse(is.na(titanic\_train$Age), "Unknown", ifelse(titanic\_train$Age < 18, "Child", "Adult"))),

Class = as.factor(titanic\_train$Pclass))

set.seed(123)

train <- sample(nrow(titanic), nrow(titanic) \* 0.7)

train\_data <- titanic[train, ]

test\_data <- titanic[-train, ]

library(rpart)

tree <- rpart(Survived ~ Gender + Age + Class, data = train\_data, method = "class", control = rpart.control(minsplit = 5, minbucket = 2, maxdepth = 5))

plot(tree)

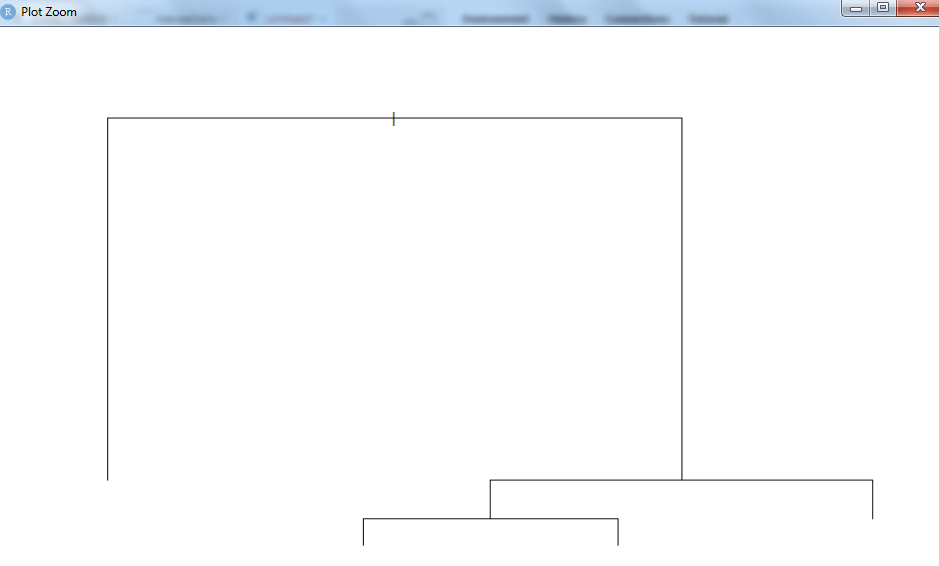
text(tree)

predictions <- predict(tree, test\_data, type = "class")

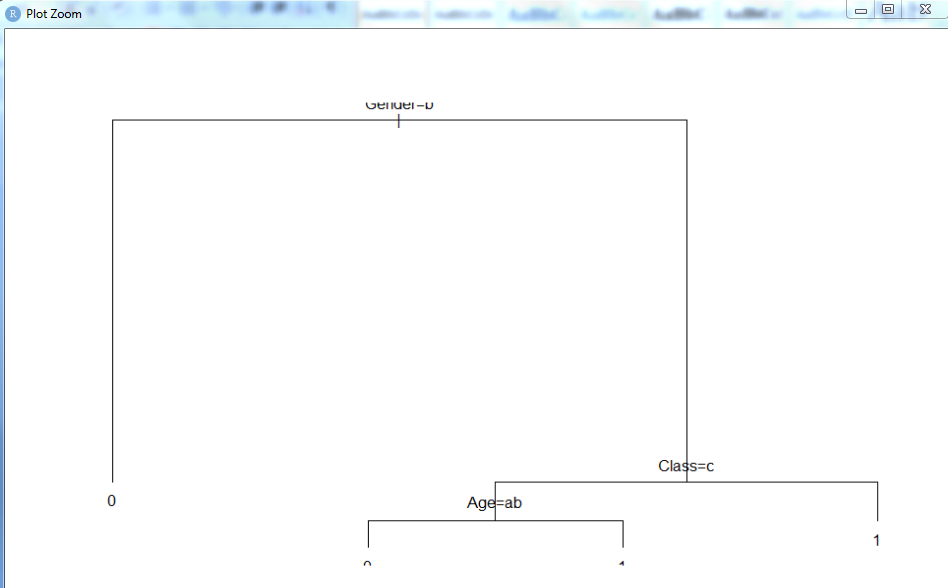
table(predictions, test\_data$Survived)

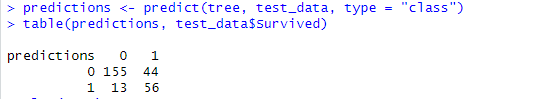
**OUTPUT:**

**PLOT TREE :**

****

**TEXT TREE :**

****

****

**RESULT :** Thus the R Program to demonstrate Classification rule process on Titanic Dataset using id3 algorithm has been implemented and executed successfully

**Ex NO : CLASSIFICATION RULES PROCESS ON DATASET TITANIC USING DATE : CART ALGORITHM**

**AIM :** The aim of the Program is To Demonstrate Classification Rules Process on Titanic dataset using CART Algorithm

**PREREQUISITES :** LOAD and INSTALL titanic dataset , rpart Package From Tools

**PROCEDURE :**

**Step 1:** Load the titanic dataset

**Step 2:** Load the necessary packages

**Step 3:** Convert the dataset to a data frame

**Step 4:** Set the seed for reproducibility**.**

**Step 5:** Split the dataset into training and testing sets

**Step 6:** Fit the CART model using the training set

**Step 7:** Plot the CART model

**Step 8:** Make predictions on the testing set

**Step 9:** Evaluate the model's performance

**Step 10:** Save and Run all the commands

**PROGRAM CODE :**

library(rpart)

library(rpart.plot)

library(caret)

data(Titanic)

Titanic\_df <- as.data.frame(Titanic)

set.seed(123)

trainIndex <- createDataPartition(Titanic\_df$Survived, p = 0.8, list = FALSE)

train <- Titanic\_df[trainIndex, ]

test <- Titanic\_df[-trainIndex, ]

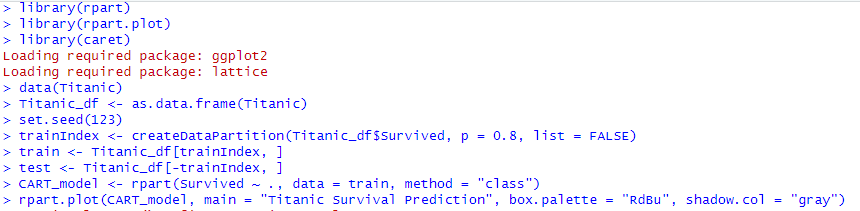
CART\_model <- rpart(Survived ~ ., data = train, method = "class")

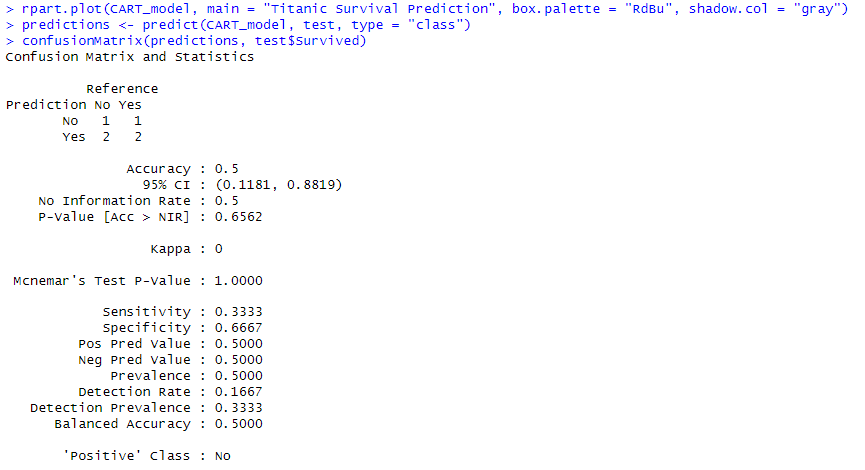
rpart.plot(CART\_model, main = "Titanic Survival Prediction", box.palette = "RdBu", shadow.col = "gray")

predictions <- predict(CART\_model, test, type = "class")

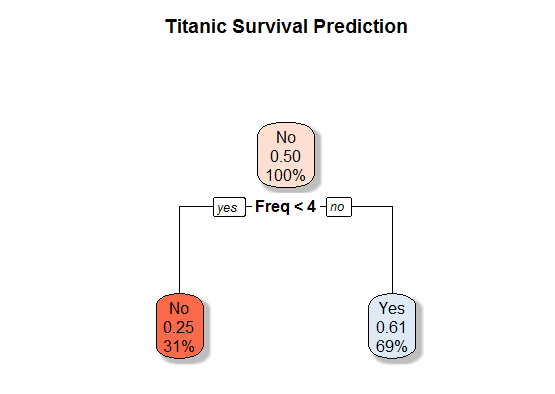
confusionMatrix(predictions, test$Survived)

**OUTPUT:**

****

****

**PLOT TREE :**

****

**RESULT :** Thus the R Program to demonstrate Classification rule process on Titanic Dataset using id3 algorithm has been implemented and executed successfully

**Ex NO : CLASSIFICATION RULE PROCESS ON BREAST CANCER**

**DATE : DATASET USING NAÏVE BAYESIAN ALGORITHM**

**AIM :** The aim of the Program is To Demonstrate Classification Rules Process on Breast Cancer dataset using Naïve Bayesian Algorithm.

**PREREQUISITES :** LOAD and INSTALL mlbench , BreastCancer Dataset From Tools

**PROCEDURE :**

**Step 1:** Install and load the mlbench package

**Step 2:** Load the breast cancer dataset

**Step 3:** Convert the class variable to a factor

**Step 4:** Convert the dataset to a data frame

**Step 5:** Split the data into a training set and a testing set

**Step 6:** Build the Naive Bayes model

**Step 7:** Use the model to make predictions on the testing set

**Step 8:** Evaluate the performance of the model using a confusion matrix

**Step 9:** Calculate the accuracy, precision, and recall of the model

**Step 10:** Plot the performance of the model

**Step 11:** Save and Run all the commands

**PROGRAM CODE :**

install.packages("mlbench")

library(mlbench)

data(BreastCancer)

BreastCancer$Class <- as.factor(BreastCancer$Class)

set.seed(123)

train <- sample(nrow(BreastCancer), nrow(BreastCancer) \* 0.7)

train\_data <- BreastCancer[train, ]

test\_data <- BreastCancer[-train, ]

library(e1071)

model <- naiveBayes(Class ~ ., data = train\_data)

predictions <- predict(model, test\_data)

table(predictions, test\_data$Class)

conf\_mat <- table(predictions, test\_data$Class)

accuracy <- sum(diag(conf\_mat)) / sum(conf\_mat)

precision <- diag(conf\_mat) / colSums(conf\_mat)

recall <- diag(conf\_mat) / rowSums(conf\_mat)

library(ggplot2)

perf\_df <- data.frame(Measure = c("Accuracy", "Precision", "Recall"),

Value = c(accuracy, mean(precision), mean(recall)),

Type = "Naive Bayes")

ggplot(perf\_df, aes(x = Measure, y = Value, fill = Type)) +

geom\_bar(stat = "identity", position = position\_dodge()) +

ylim(0, 1) +

ggtitle("Performance of Naive Bayes Model") +

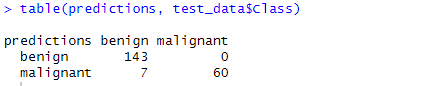
xlab("") +

ylab("") +

theme\_bw()

**OUTPUT:**

****

****

**RESULT :** Thus the R Program to demonstrate Classification rule process on Breast Cancer Dataset using Naïve Bayesian algorithm has been implemented and executed successfully

**Ex NO: CLUSTERING RULE PROCESS ON DATASET IRIS USING SIMPLE K DATE: MEANS**

**AIM :** The aim of the Program is To Demonstrate Clustering Rules Process on Iris dataset using Simple K-Means.

**PREREQUISITES : LOAD Iris Dataset Go To Tools - Install this package**

**PROCEDURE :**

**Step 1 :** Install and Load the Iris dataset

**Step 2:** Convert the species variable to a factor

**Step 3 :** Select the relevant variables for clustering

**Step 4:** Set the number of clusters

**Step 5:** Run the K-means algorithm

**Step 6:** Get the cluster assignments for each observation

**Step 7:** Evaluate the performance of the model

**Step 8:** Plot the clusters

**Step 9:** Save and Run all the commands

**PROGRAM CODE :**

data(iris)

iris$Species <- as.factor(iris$Species)

X <- iris[, -5]

k <- 3

set.seed(123)

kmeans\_model <- kmeans(X, centers = k)

cluster\_assignments <- kmeans\_model$cluster

table(cluster\_assignments, iris$Species)

library(ggplot2)

ggplot(iris, aes(x = Petal.Length, y = Petal.Width, color = Species)) +

geom\_point() +

ggtitle("Iris Clustering with K-Means") +

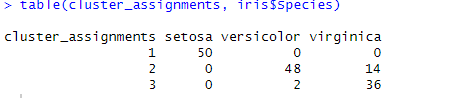
xlab("Petal Length") +

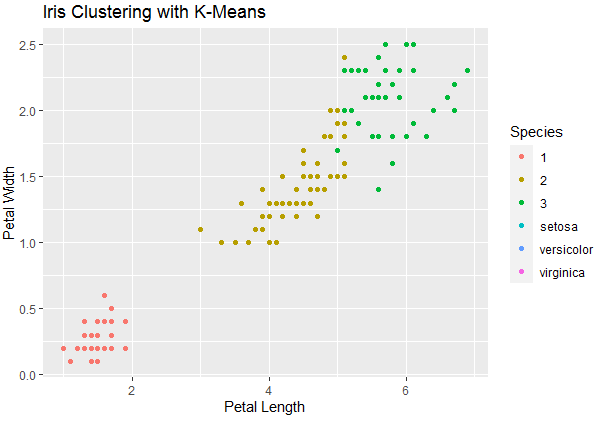
ylab("Petal Width") +

geom\_point(aes(x = Petal.Length, y = Petal.Width, color = factor(cluster\_assignments)),

data = cbind(iris, cluster\_assignments))

**OUTPUT :**

****

****

**RESULT :** Thus the R Program to demonstrate Clustering rule process on Iris Dataset using simple K-Means has been implemented and executed successfully

**Ex NO : CLUSTERING RULE PROCESS ON DATASET IRIS USING**

**DATE : HIERARCHICAL CLUSTERING**

**AIM :** The aim of the Program is To Demonstrate Clustering Rules Process on Iris dataset using Hierarchical Clustering in R.

**PROCEDURE :**

**Step 1 :** Install and Load the Iris dataset

**Step 2:** Convert the species variable to a factor

**Step 3 :** Select the relevant variables for clustering

**Step 4:** Set the number of clusters

**Step 5:** Run the hierarchical clustering

**Step 6:** Cut the dendrogram at a certain height to obtain clusters

**Step 7:** Evaluate the performance of the model

**Step 8:** Plot the clusters

**Step 9:** Save and Run all the commands

**PROGRAM CODE:**

data(iris)

iris$Species <- as.factor(iris$Species)

X <- iris[, -5]

hc\_model <- hclust(dist(X))

k <- 3

cluster\_assignments <- cutree(hc\_model, k)

table(cluster\_assignments, iris$Species)

library(ggplot2)

ggplot(iris, aes(x = Petal.Length, y = Petal.Width, color = Species)) +

geom\_point() +

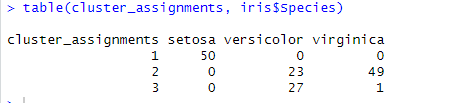
ggtitle("Iris Clustering with Hierarchical Clustering") +

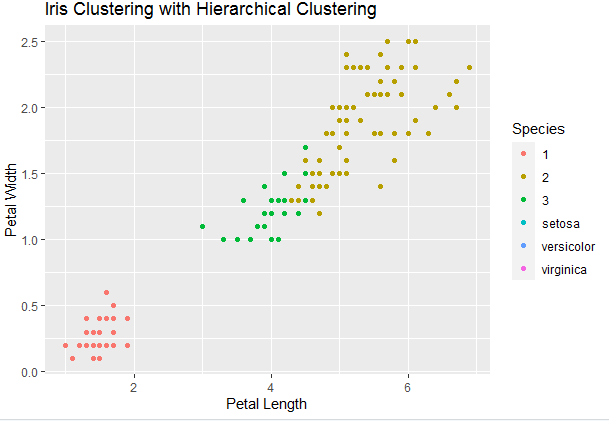
xlab("Petal Length") +

ylab("Petal Width") +

geom\_point(aes(x = Petal.Length, y = Petal.Width, color = factor(cluster\_assignments)))

**OUTPUT :**

****

****

**RESULT :** Thus the R Program to demonstrate Clustering rule process on Iris Dataset using hierarchical clustering has been implemented and executed successfully

**Ex NO : OUTLIER DETECTION USING DATASET IRIS**

**DATE :**

**AIM :** The aim of the Program is To Demonstrate Outlier Detection on Iris dataset using R

**PROCEDURE :**

**Step 1:** Install and Load the Iris dataset

**Step 2:** Convert the species variable to a factor

**Step 3:** Select the relevant variables for clustering

**Step 4:** Calculate the Mahalanobis distance for each observation

**Step 5:** Set the threshold for outlier detection

**Step 6:** Identify the outliers

**Step 7:** Print the number of outliers detected

**Step 8:** Plot the observations with outliers in red

**Step 9:** Save and Run all the commands

**PROGRAM CODE:**

data(iris)

iris$Species <- as.factor(iris$Species)

X <- iris[, -5]

mah\_dist <- mahalanobis(X, colMeans(X), cov(X))

threshold <- qchisq(0.99, df = ncol(X))

outliers <- which(mah\_dist > threshold)

cat(paste("Number of outliers detected:", length(outliers), "\n"))

library(ggplot2)

ggplot(iris, aes(x = Petal.Length, y = Petal.Width, color = Species)) +

geom\_point() +

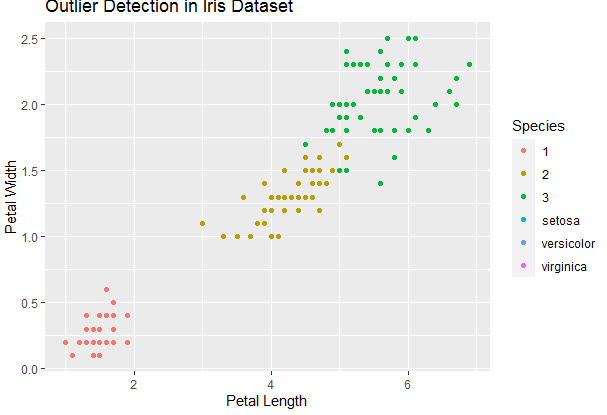
ggtitle("Outlier Detection in Iris Dataset") +

xlab("Petal Length") +

ylab("Petal Width") +

geom\_point(aes(x = Petal.Length, y = Petal.Width, color = factor(ifelse(seq\_along(iris[, 1]) %in% outliers, "Outlier", Species))))

**OUTPUT :**

****

****

**RESULT :** Thus the R Program to demonstrate Outlier Detection on Iris Dataset has been implemented and executed successfully.