**PROGRAM 1**

library(editrules)

d = read.table(file.choose(),header=TRUE) #select people.txt

attach(d) #to avoid using $ symbol with dataset

E <- editset(expression(

age >= 0,

age <= 150,

age > yearsmarried,

status %in% c('single','married','widowed'),

if(age <= 18) agegroup %in% c('child'),

if(age > 18 && age < 65 ) agegroup %in% c('adult'),

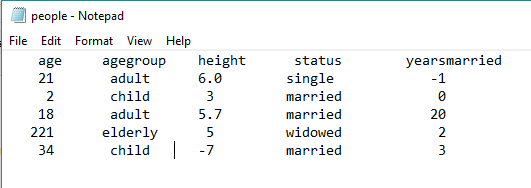
if(age >= 65) agegroup %in% c('elderly')

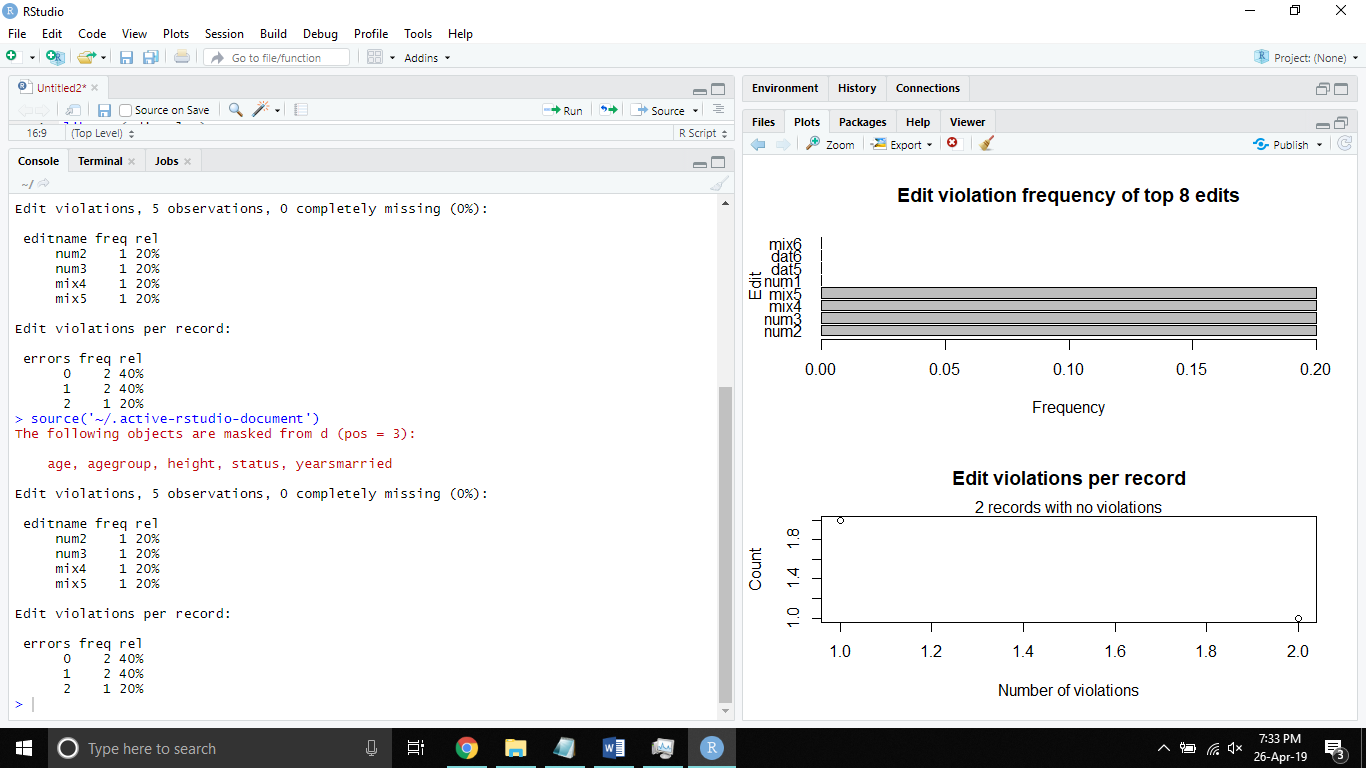
))

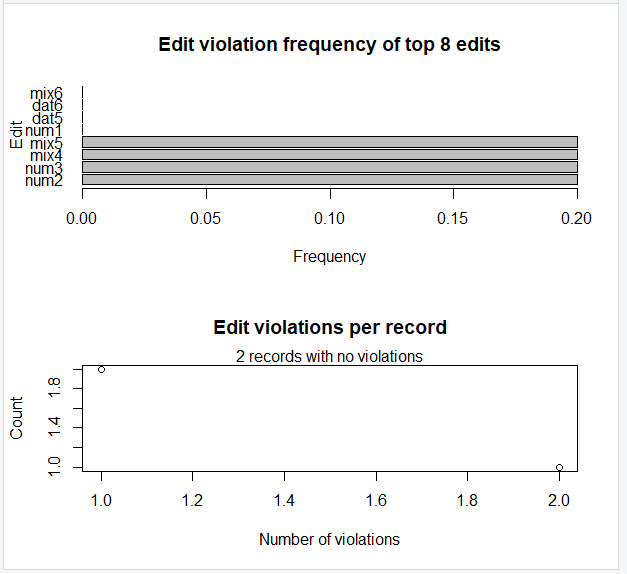
sm <- violatedEdits(E,d)

summary(sm)

plot(sm)







**PROGRAM 2**

x = read.csv(file.choose()) #select dirty\_iris.csv

#replace special values with NA

x[,-5] = lapply(x[,-5], function(y) as.numeric(as.character(y)))

#total number of complete observations

c = sum(complete.cases(x))

cat("Number of complete observations : ", c, "\n")

#percentage of complete observations

cat("Number of complete observations : ", c/(dim(x)[1])\*100, "\n\n")

x = na.omit(x) #delete records with NAs

library(editrules)

edit2 <- editfile(file.choose()) #select rules2.txt

sm <- violatedEdits(edit2,x)

summary(sm)

plot(sm)

boxplot(iris$Sepal.Length)

boxplot.stats(iris$Sepal.Length)

CSV FILE

Sepal Length Sepal Width Petal Length Petal Width Species

5.1 3.5 1.4 0.2 setosa

4.9 3 1.4 0.2 setosa

@ 1.3 0.2 setosa

4.6 3.1 1.5 0.2 ABC

5 3.6 1.4 0.2 setosa

5.4 1.7 0.4 setosa

4.6 3.4 1.4 0.3 setosa

\_ 3.4 1.5 0.2 XYZ

4.4 2.9 1.4 0.2 setosa

4.9 @ 1.5 0.1 setosa

$ 3.7 1.5 0.2 setosa

4.8 3.4 1.6 0.2 setosa

4.8 3 1.4 0.1 setosa

4.3 3 1.1 0.1 setosa

5.8 4 1.2 0.2 setosa

5.7 4.4 -1.5 0.4 setosa

5.4 3.9 -1.3 0.4 setosa

5.1 3.5 6 0.3 setosa

48 3.8 1.7 0.3 setosa

5.1 3.8 1.5 0.3 setosa

44 3.4 0.2 0.2 setosa

5.1 3.7 1.5 0.4 setosa

4.6 3.6 1 0.2 setosa

5.1 3.3 1.7 0.5 setosa

4.8 3.4 1.9 0.2 setosa

5 3 1.6 0.2 setosa

5 3.4 1.6 0.4 setosa

5.2 3.5 1.5 0.2 setosa

5.2 3.4 1.4 0.2 setosa

4.7 3.2 1.6 0.2 setosa

4.8 3.1 1.6 0.2 setosa

5.4 3.4 1.5 0.4 setosa

5.2 4.1 1.5 0.1 setosa

5.5 4.2 1.4 0.2 setosa

4.9 3.1 1.5 0.1 setosa

5 3.2 1.2 0.2 setosa

5.5 3.5 1.3 0.2 setosa

4.9 3.1 1.5 0.1 setosa

4.4 3 1.3 0.2 setosa

5.1 3.4 1.5 0.2 setosa

5 3.5 1.3 0.3 setosa

4.5 2.3 1.3 0.3 setosa

4.4 3.2 1.3 0.2 setosa

5 3.5 1.6 0.6 setosa

5.1 3.8 1.9 0.4 setosa

4.8 3 1.4 0.3 setosa

5.1 3.8 1.6 0.2 setosa

4.6 3.2 1.4 0.2 setosa

5.3 3.7 1.5 0.2 setosa

5 3.3 1.4 0.2 setosa

7 3.2 4.7 1.4 versicolor

6.4 3.2 4.5 1.5 versicolor

6.9 3.1 4.9 1.5 versicolor

5.5 2.3 4 1.3 versicolor

6.5 2.8 4.6 1.5 versicolor

5.7 2.8 4.5 1.3 versicolor

6.3 3.3 4.7 1.6 versicolor

4.9 2.4 3.3 1 versicolor

6.6 2.9 4.6 1.3 versicolor

5.2 2.7 3.9 1.4 versicolor

5 2 3.5 1 versicolor

5.9 3 4.2 1.5 versicolor

6 2.2 4 1 versicolor

6.1 2.9 4.7 1.4 versicolor

5.6 2.9 3.6 1.3 versicolor

6.7 3.1 4.4 1.4 versicolor

5.6 3 4.5 1.5 versicolor

5.8 2.7 4.1 1 versicolor

6.2 2.2 4.5 1.5 versicolor

5.6 2.5 3.9 1.1 versicolor

5.9 3.2 4.8 1.8 versicolor

6.1 2.8 4 1.3 versicolor

6.3 2.5 4.9 1.5 versicolor

6.1 2.8 4.7 1.2 versicolor

6.4 2.9 4.3 1.3 versicolor

6.6 3 4.4 1.4 versicolor

6.8 2.8 4.8 1.4 versicolor

6.7 3 5 1.7 versicolor

6 2.9 4.5 1.5 versicolor

5.7 2.6 3.5 1 versicolor

5.5 2.4 3.8 1.1 versicolor

5.5 2.4 3.7 1 versicolor

5.8 2.7 3.9 1.2 versicolor

6 2.7 5.1 1.6 versicolor

5.4 3 4.5 1.5 versicolor

6 3.4 4.5 1.6 versicolor

6.7 3.1 4.7 1.5 versicolor

6.3 2.3 4.4 1.3 versicolor

5.6 3 4.1 1.3 versicolor

5.5 2.5 4 1.3 versicolor

5.5 2.6 4.4 1.2 versicolor

6.1 3 4.6 1.4 versicolor

5.8 2.6 4 1.2 versicolor

5 2.3 3.3 1 versicolor

5.6 2.7 4.2 1.3 versicolor

5.7 3 4.2 1.2 versicolor

5.7 2.9 4.2 1.3 versicolor

6.2 2.9 4.3 1.3 versicolor

5.1 2.5 3 1.1 versicolor

5.7 2.8 4.1 1.3 versicolor

6.3 3.3 6 2.5 virginica

5.8 2.7 5.1 1.9 virginica

7.1 3 5.9 2.1 virginica

6.3 2.9 5.6 1.8 virginica

6.5 3 5.8 2.2 virginica

7.6 3 6.6 2.1 virginica

4.9 2.5 4.5 1.7 virginica

7.3 2.9 6.3 1.8 virginica

6.7 2.5 5.8 1.8 virginica

7.2 3.6 6.1 2.5 virginica

6.5 3.2 5.1 2 virginica

6.4 2.7 5.3 1.9 virginica

6.8 3 5.5 2.1 virginica

5.7 2.5 5 2 virginica

5.8 2.8 5.1 2.4 virginica

6.4 3.2 5.3 2.3 virginica

6.5 3 5.5 1.8 virginica

7.7 3.8 6.7 2.2 virginica

7.7 2.6 6.9 2.3 virginica

6 2.2 5 1.5 virginica

6.9 3.2 5.7 2.3 virginica

5.6 2.8 4.9 2 virginica

7.7 2.8 6.7 2 virginica

6.3 2.7 4.9 1.8 virginica

6.7 3.3 5.7 2.1 virginica

7.2 3.2 6 1.8 virginica

6.2 2.8 4.8 1.8 virginica

6.1 3 4.9 1.8 virginica

6.4 2.8 5.6 2.1 virginica

7.2 3 5.8 1.6 virginica

7.4 2.8 6.1 1.9 virginica

7.9 3.8 6.4 2 virginica

6.4 2.8 5.6 2.2 virginica

6.3 2.8 5.1 1.5 virginica

6.1 2.6 5.6 1.4 virginica

7.7 3 6.1 2.3 virginica

6.3 3.4 5.6 2.4 virginica

6.4 3.1 5.5 1.8 virginica

6 3 4.8 1.8 virginica

6.9 3.1 5.4 2.1 virginica

6.7 3.1 5.6 2.4 virginica

6.9 3.1 5.1 2.3 virginica

5.8 2.7 5.1 1.9 virginica

6.8 3.2 5.9 2.3 virginica

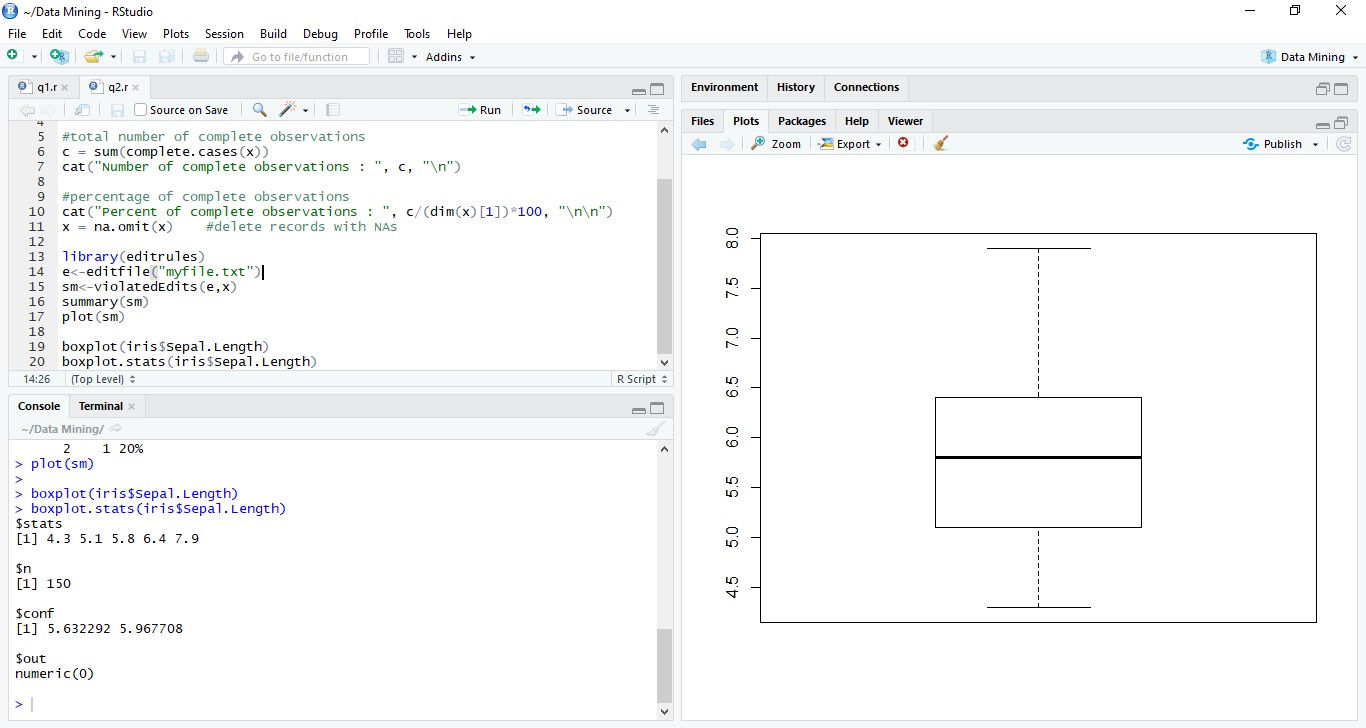
6.7 3.3 5.7 2.5 virginica

6.7 3 5.2 2.3 virginica

6.3 2.5 5 1.9 virginica

6.5 3 5.2 2 virginica

6.2 3.4 5.4 2.3 virginica

 5.9 3 5.1 1.8 virginica

**PROGRAM 3**

iris1 <- iris[,-5]

summary(iris1)

s <- sapply(iris1,sd)

s

hist(iris1$Sepal.Width) #check bell shape curve for all attributes

data\_std <- function(x) { (x-mean(x))/sd(x) }

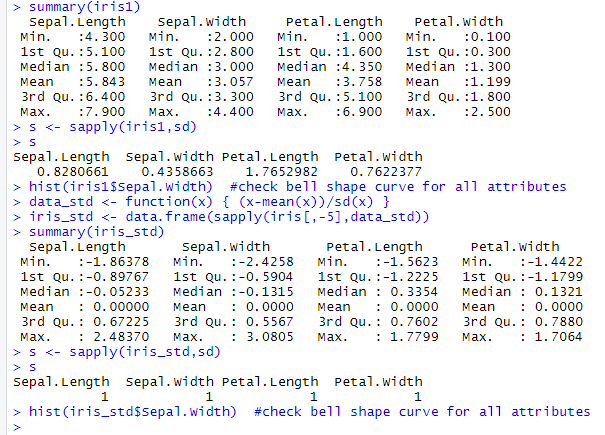
iris\_std <- data.frame(sapply(iris[,-5],data\_std))

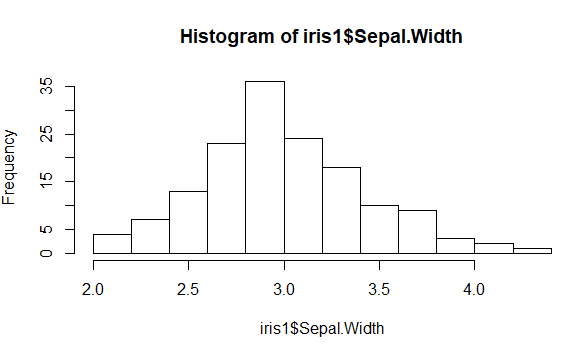
summary(iris\_std)

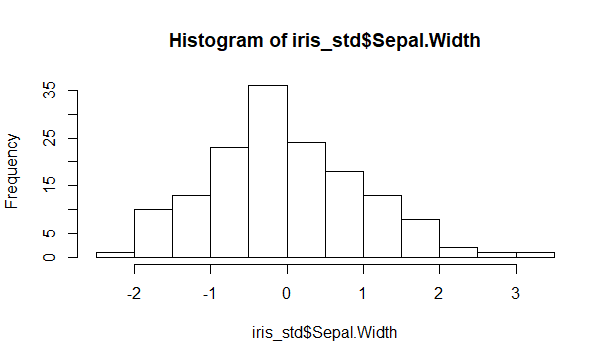
s <- sapply(iris\_std,sd)

s

hist(iris\_std$Sepal.Width) #check bell shape curve for all attributes







**PROGRAM 4**

library(arules)

data(Groceries)

#SUPPORT=0.1% CONFIDENCE=80%

itemFrequencyPlot(Groceries, topN=20, type="absolute")

rules <- apriori(Groceries, parameter=list(sup=0.001,conf=0.8))

inspect(head(rules))

rules <- sort(rules, by="confidence", decreasing = TRUE)

inspect(head(rules))

#SUPPORT=0.5% CONFIDENCE=70%

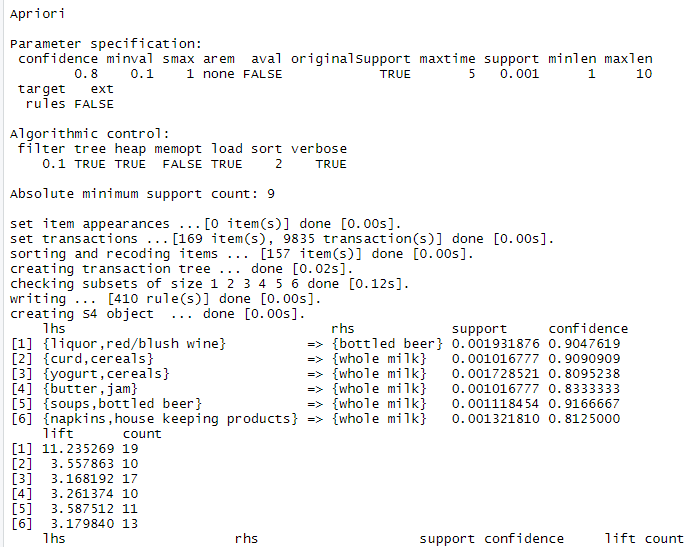
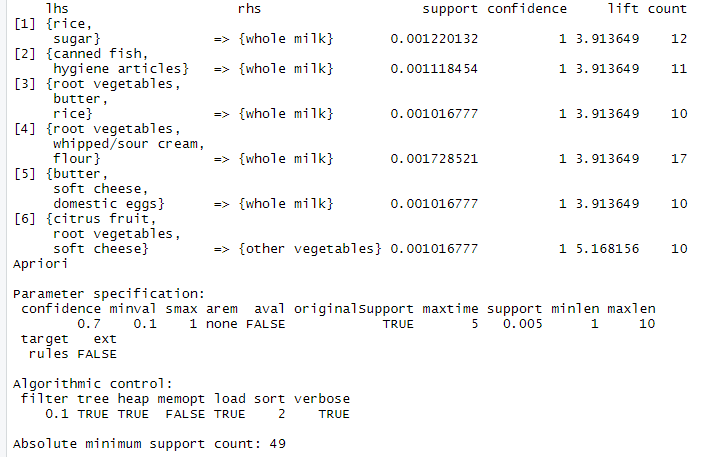
itemFrequencyPlot(Groceries, topN=20, type="absolute")

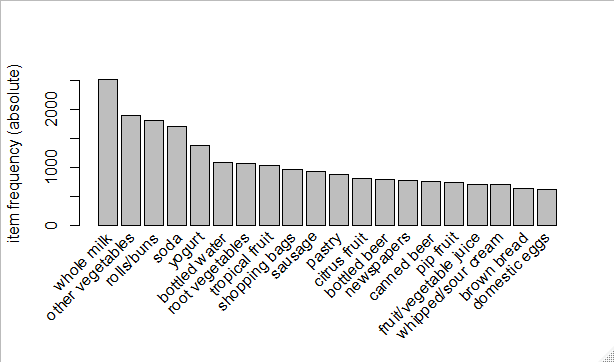
rules <- apriori(Groceries, parameter=list(sup=0.005,conf=0.7))

inspect(head(rules))

rules <- sort(rules, by="confidence", decreasing = TRUE)

inspect(head(rules))





**PROGRAM 5**

#Naive Bayes

library(naivebayes)

library(caret)

set.seed(1234)

id <- sample(2,150,prob=c(0.7,0.3),replace = T)

train <- iris[id==1,]

test <- iris[id==2,]

imp\_nb = naive\_bayes(Species~., data=train)

pre3 <- predict(imp\_nb,test)

confusionMatrix(table(pre3, test$Species))

mean(pre3==test[,5])

#KNN

library(class)

normalize <- function(x) { (x-min(x))/(max(x)-min(x)) }

iris\_norm <- sapply(iris[,-5], normalize)

s <- sample(150,120)

iris\_train <- iris\_norm[s,]

iris\_test <- iris\_norm[-s,]

iris\_pred <- knn(iris\_train,iris\_test,iris[s,5],k=13)

table(iris\_pred,iris[-s,5])

mean(iris\_pred==iris[-s,5])

#DECISION TREE

library(rpart)

library(rpart.plot)

s <- sample(150, 100)

train <- iris[s,]

test <- iris[-s,]

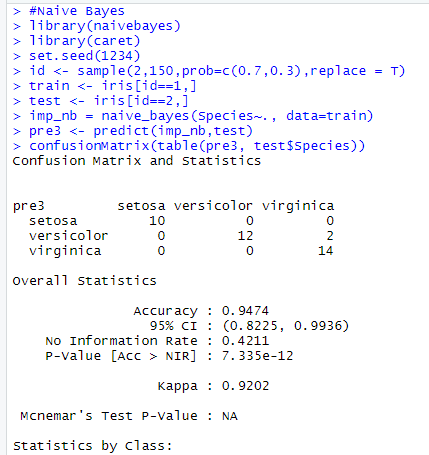
d <- rpart(Species~., train, method = "class")

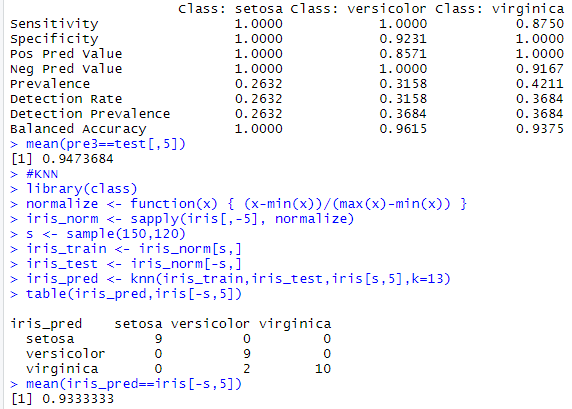
rpart.plot(d)

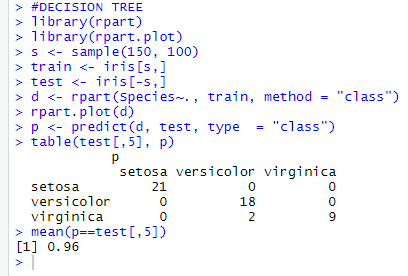
p <- predict(d, test, type = "class")

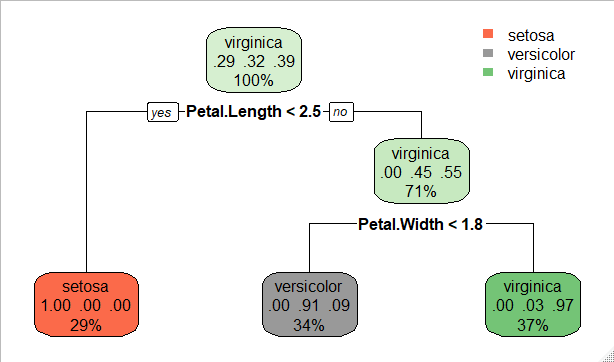
table(test[,5], p)

mean(p==test[,5])









**PROGRAM 6**

#Kmeans

library(cluster)

iris1 <- iris[,1:4]

results <- kmeans(iris1,3)

results

table(iris$Species,results$cluster)

plot(iris[,-5],col=results$cluster)

#DBScan

library(dbscan)

iris\_m <- iris[,1:4]

kNNdistplot(iris\_m,k=1)

abline(h=0.4,col="red")

db <- dbscan(iris\_m,0.4,4)

db

hullplot(iris\_m,db$cluster)

table(iris$Species,db$cluster)

#Hierarchical Clustering

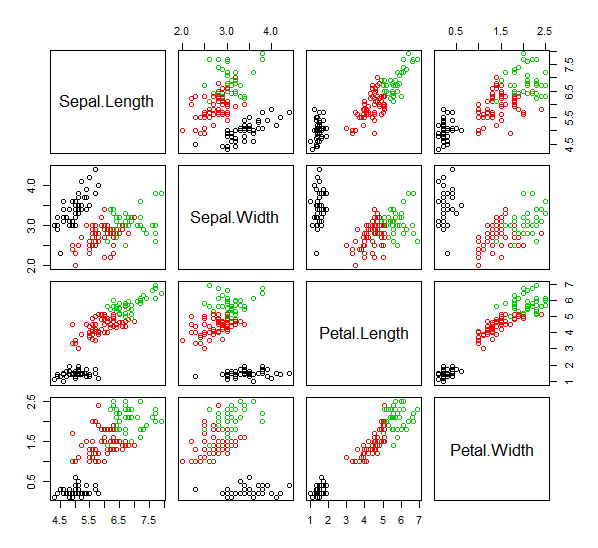
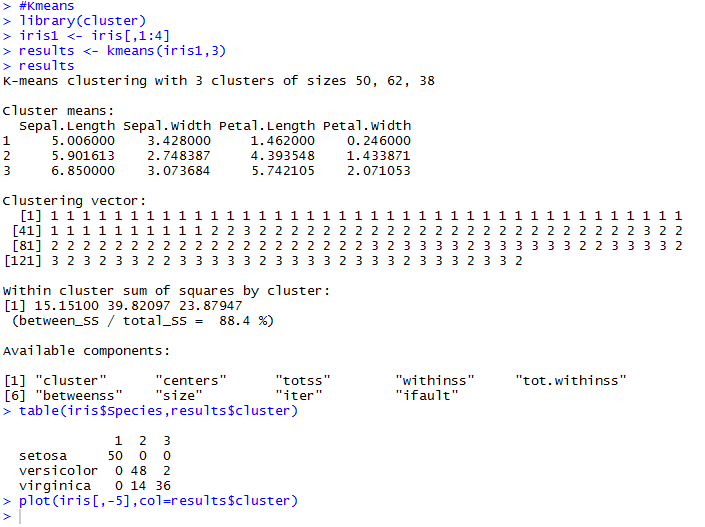
library(cluster)

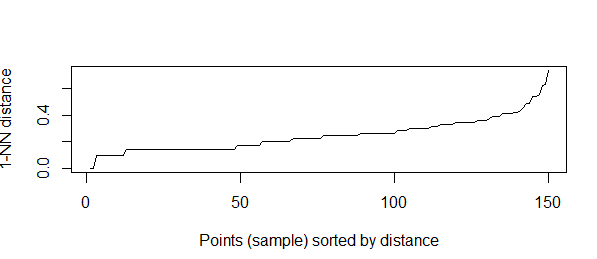
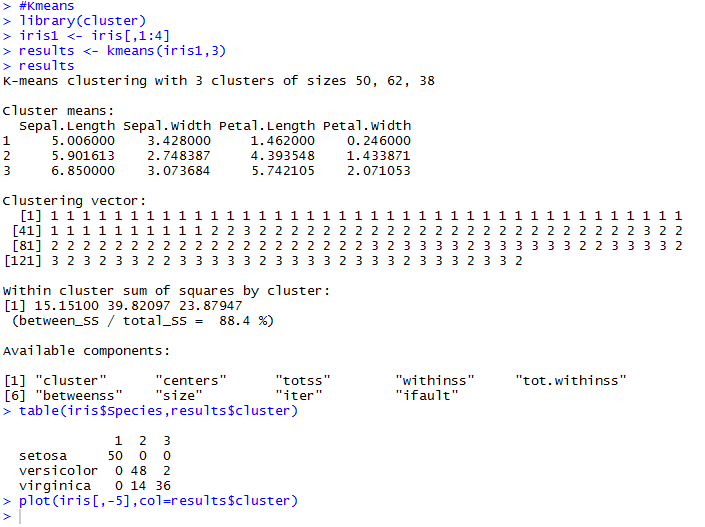
hc\_complete <- hclust(dist(iris),method = "complete")

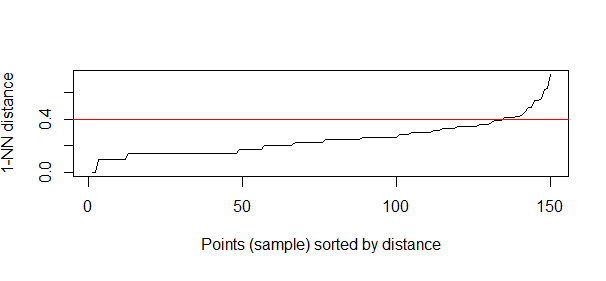
hc\_single <- hclust(dist(iris),method = "single")

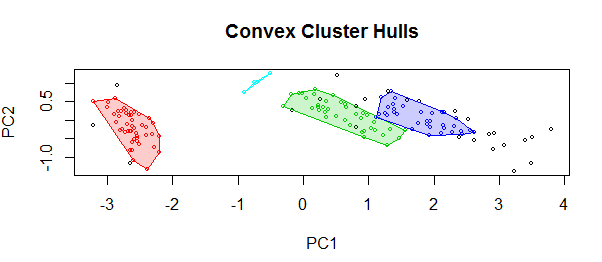
plot(hc\_complete, main = "Hierarchical Clustering Complete", cex = 0.9)

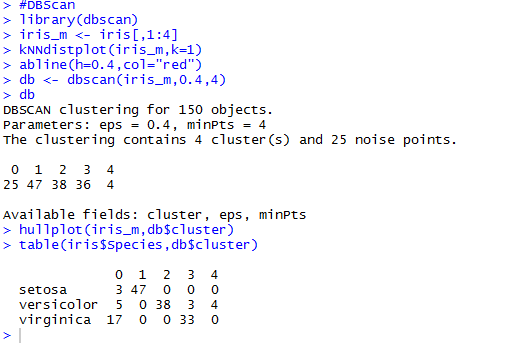
plot(hc\_single, main = "Hierarchical Clustering Single", cex = 0.9)











1. 