|  |  |
| --- | --- |
| **EX. NO:** | **IMPLEMENTATION OF K-MEANS CLUSTERING ALGORITHMS USING R TOOL** |
| **DATE :** |

**AIM:** To implement K-Means Clustering Algorithms using R Tool.

## ALGORITHM:

K-Means clustering groups the data on similar groups. The algorithm is as follows:

1. Choose the number K clusters.
2. Select at random K points, the centroids(Not necessarily from the given data).
3. Assign each data point to closest centroid that forms K clusters.
4. Compute and place the new centroid of each centroid.
5. Reassign each data point to new cluster.
6. After final reassignment, name the cluster as Final cluster.

## PROGRAM:

# Loading data data(iris)

# Structure str(iris)

# Installing Packages install.packages("Cluste rR")

install.packages("cluster ")

# Loading package library(ClusterR) library(cluster)

# Removing initial label of # Species from original dataset iris\_1 <**-** iris[, **-**

5]

# Fitting K-Means clustering Model # to training dataset set.seed(240) # Setting seed

kmeans.re <**-** kmeans(iris\_1, centers **=** 3, nstart **=**

1. kmeans.re # Cluster

identification for # each observation kmeans.re$cluster # Confusion Matrix

cm <**-** table(iris$Species, kmeans.re$cluster) cm

# Model Evaluation and visualization plot(iris\_1[c("Sepal.Length", "Sepal.Width")]) plot(iris\_1[c("Sepal.Length", "Sepal.Width")],

col **=** kmeans.re$cluster) plot(iris\_1[c("Sepal.Length", "Sepal.Width")],

col **=** kmeans.re$cluster, main **=** "K-means with 3

clusters") ## Plotiing cluster centers kmeans.re$centers

kmeans.re$centers[, c("Sepal.Length", "Sepal.Width")] # cex is font size, pch is symbol

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

## Visualizing clusters y\_kmeans <**-** kmeans.re$cluster

clusplot(iris\_1[, c("Sepal.Length", "Sepal.Width")], y\_kmeans,

lines **=** 0, shade **=** TRUE,

color **=**

TRUE,

labels **=**

2,

plotchar **=**

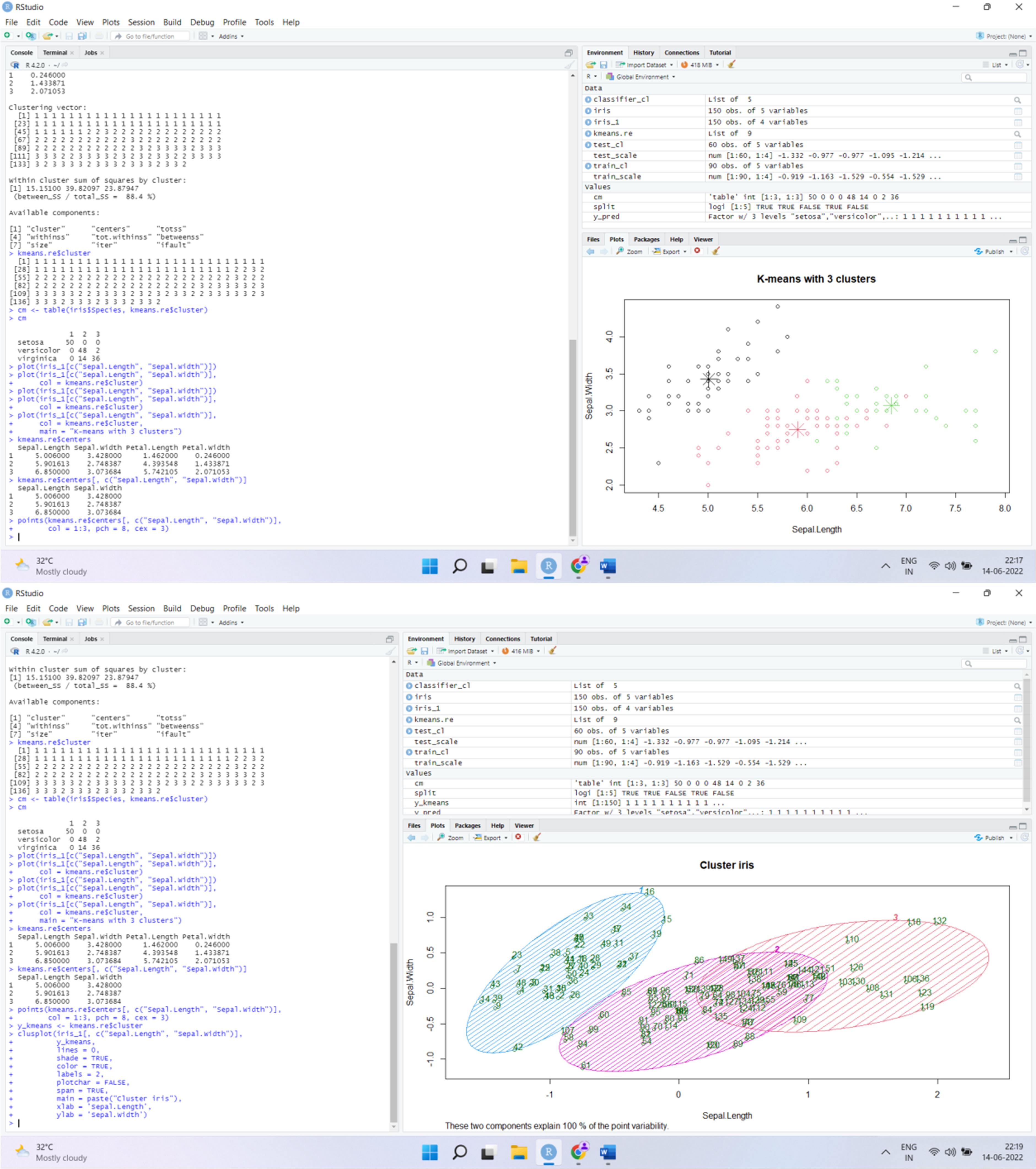
FALSE, span

**=** TRUE,

main **=** paste("Cluster iris"), xlab **=** 'Sepal.Length',

ylab **=** 'Sepal.Width')

# Output:



## RESULT:

Thus, the K-Means Clustering Algorithms using R Tool is implemented

|  |  |
| --- | --- |
| **EX. NO:** | **IMPLEMENTATION OF APRIORI ASSOCIATION ALGORITHM USING R TOOL** |
| **DATE :** |

**AIM:** To Implement the Apriori association algorithm using R Tool.

## ALGORITHM:

* 1. Read each item in the transaction.
  2. Calculate the support of every item.
  3. If support is less than minimum support, discard the item. Else, insert it into frequent itemset.
  4. Calculate confidence for each non- empty subset.
  5. If confidence is less than minimum confidence, discard the subset. Else, it into strong rules

## PROGRAM:

# Loading Libraries library(arules) library(arulesViz) library(RColorBrewer) # import dataset data("Groceries")

# using apriori() function

rules <- apriori(Groceries, parameter = list(supp = 0.01, conf = 0.2)) # using inspect() function

inspect(rules[1:10])

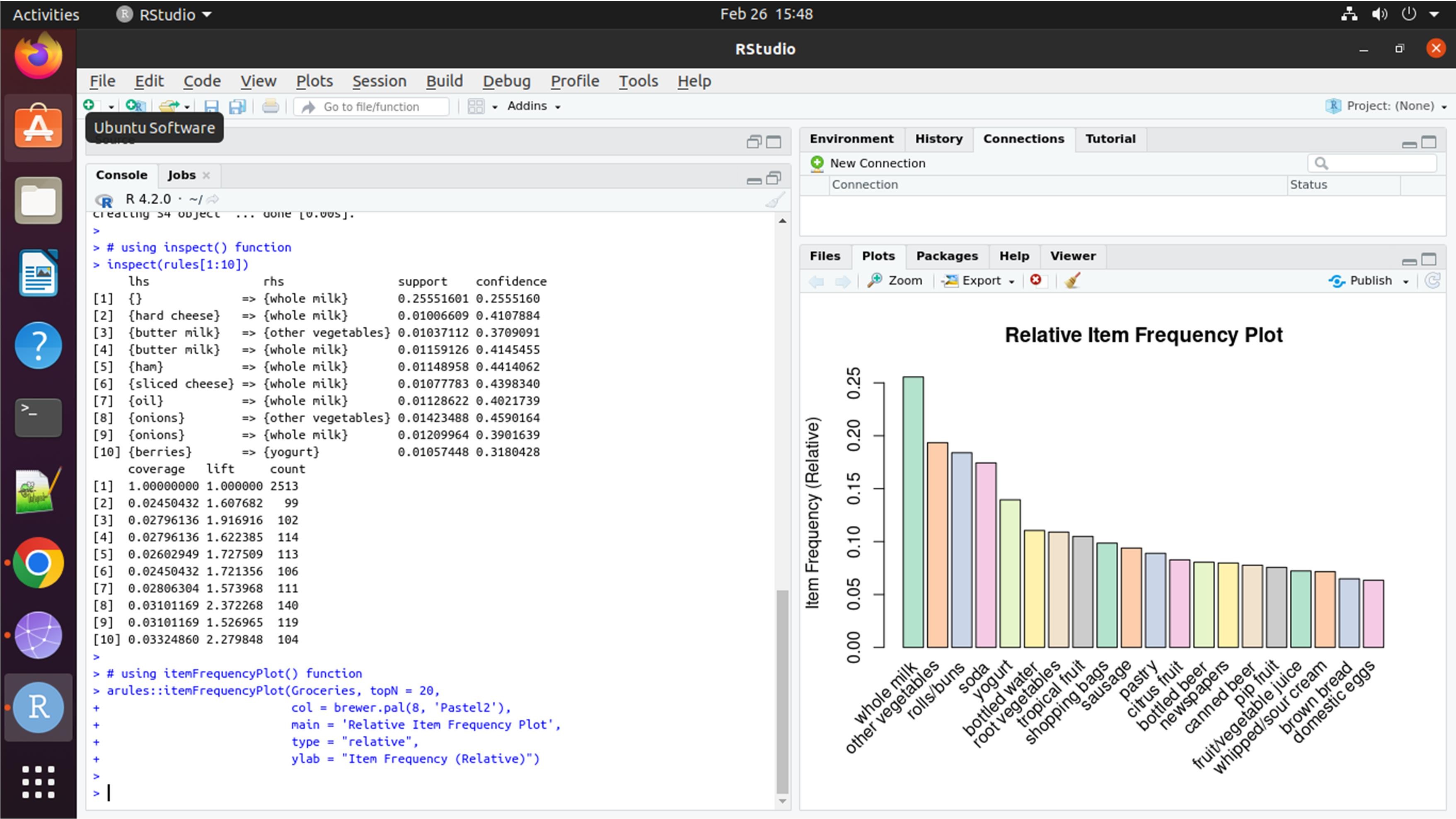
# using itemFrequencyPlot() function arules::itemFrequencyPlot(Groceries, topN = 20,

col = brewer.pal(8, 'Pastel2'),

main = 'Relative Item Frequency Plot', type = "relative",

ylab = "Item Frequency (Relative)")

# OUTPUT:



**RESULT:** Thus, Apriori association algorithm was implemented using R Tool.

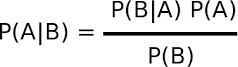
|  |  |
| --- | --- |
| **EX. NO:** | **IMPLEMENTATION OF NAIVE BAYES CLASSIFICATION**  **USING R TOOL** |
| **DATE:** |

**AIM:** To implement the Naive Bayes classification using R tool.

# Naive Bayes Classification:

Naive Bayes is a Supervised Non-linear classification algorithm in R Programming. Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Baye’s theorem with strong(Naive) independence assumptions between the features or variables. The Naive Bayes algorithm is called “Naive” because it makes the assumption that the occurrence of a certain feature is independent of the occurrence of other features.

Naive Bayes algorithm is based on Bayes theorem. Bayes theorem gives the conditional probability of an event A given another event B has occurred.



where,

P(A|B) = Conditional probability of A given B. P(B|A) = Conditional probability of B given A. P(A) = Probability of event A.

P(B) = Probability of event B.

# Program:

data(iris) str(iris)

install.packages("e1071") install.packages("caTools") install.packages("caret") library(e1071) library(caTools) library(caret)

split <- sample.split(iris, SplitRatio = 0.7) train\_cl <- subset(iris, split == "TRUE") test\_cl <- subset(iris, split == "FALSE") train\_scale <- scale(train\_cl[, 1:4]) test\_scale <- scale(test\_cl[, 1:4])

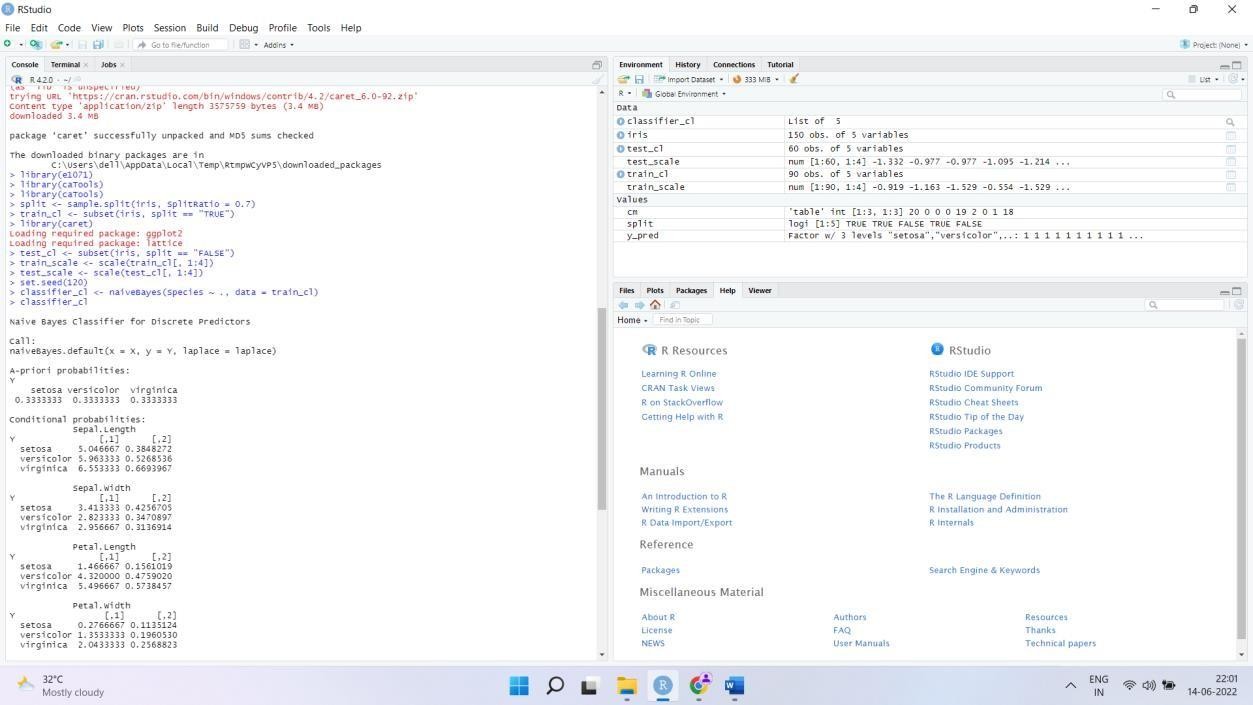
set.seed(120)

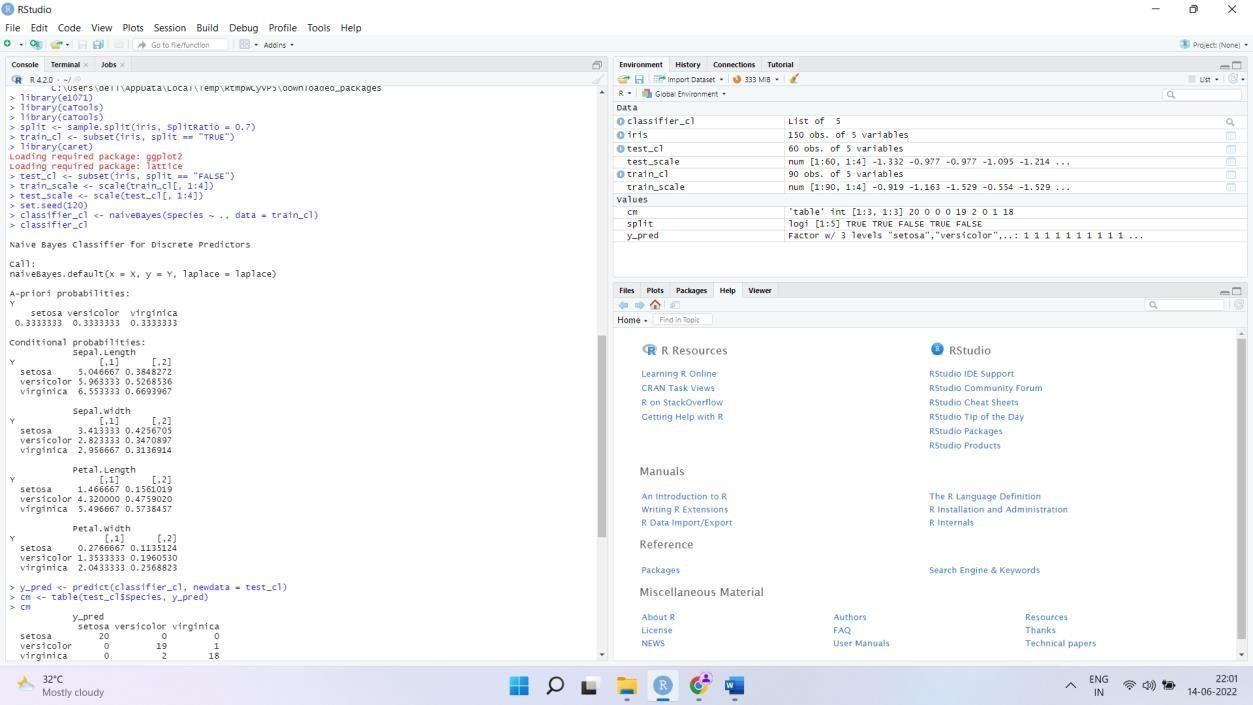
classifier\_cl <- naiveBayes(Species ~ ., data = train\_cl) y\_pred <- predict(classifier\_cl, newdata = test\_cl)

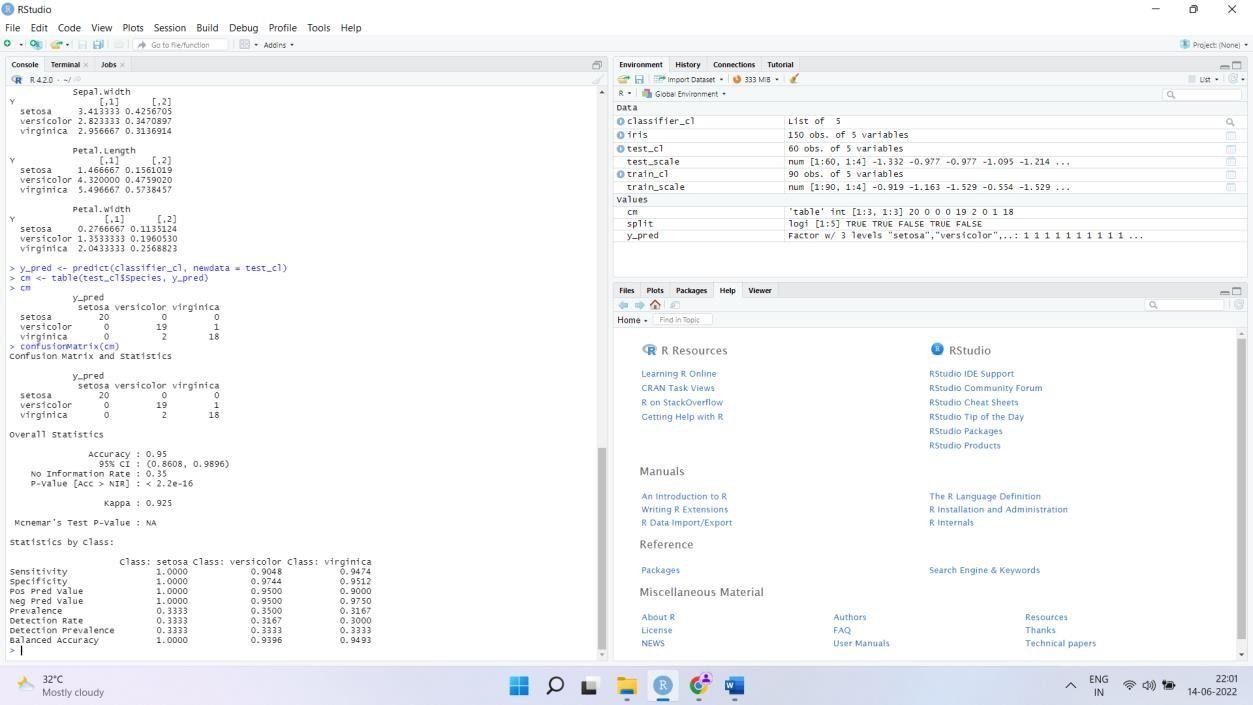
cm <- table(test\_cl$Species, y\_pred) cm

confusionMatrix(cm)

## OUTPUT:







**RESULT:** Thus, Naive Bayes was implemented using R Tool.