Implementation of Clustering techniques using - cluster, factoextra, magrittr packages for USArrests dataset.

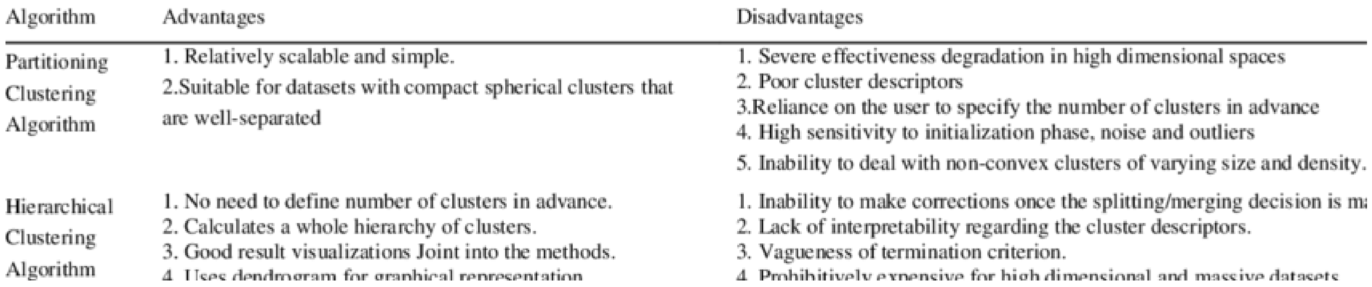
Expt No: 8 May 15,2019
Author: Subalakshmi Shanthosi S (186001008)

Aim

Implementation of different clustering techniques like - Partitioning and Hierarchical using R packages- cluster, factoextra, magrittr.

Description

1. What is Clustering?
   - Clustering is the classification of data objects into similarity groups (clusters) according to a defined distance measure.
   - It is used in many fields, such as machine learning, data mining, pattern recognition, image analysis, genomics, systems biology, etc.
   - Machine learning typically regards data clustering as a form of unsupervised learning.
2. Why Clustering and Data Mining in R?
   - Efficient data structures and functions for clustering.
   - Reproducible and programmable.
   - Comprehensive set of clustering and machine learning libraries.
   - Integration with many other data analysis tools.
3. Advantages and Disadvantages of clustering methodologies:



Algorithm	Advantages	Disadvantages
Partitioning Clustering Algorithm	1. Relatively scalable and simple. 2.Suitable for datasets with compact spherical clusters that are well-separated	1. Severe effectiveness degradation in high dimensional spaces 2. Poor cluster descriptors 3.Reliance on the user to specify the number of clusters in advance 4. High sensitivity to initialization phase, noise and outliers 5. Inability to deal with non-convex clusters of varying size and density.
Hierarchical Clustering Algorithm	1. No need to define number of clusters in advance. 2. Calculates a whole hierarchy of clusters. 3. Good result visualizations Joint into the methods. 4. Uses dendrogram for graphical representation	1. Inability to make corrections once the splitting/merging decision is m 2. Lack of interpretability regarding the cluster descriptors. 3. Vagueness of termination criterion. 4. Prohibitively expensive for high dimensional and massive datasets

Figure 1: Pros and cons of clustering.

Tools and Packages

1. Tools
   - RStudio.
   - R Version 1.1.463
2. Clustering and other Packages:
   - cluster
   - factoextra (Visualisation)
   - magrittr (pipe operator in R)
   - USArrests Dataset

Dataset Description - USArrests

- This data set contains statistics, in arrests per 100,000 residents for assault, murder, and rape in each of the 50 US states in 1973. Also given is the percent of the population living in urban areas.
- USArrests is a data frame with :
  - 50 observations(rows).
  - 4 features/variables(columns):
    * Murder : numeric, Murder arrests (per 100,000).
    * Assault: numeric, Assault arrests (per 100,000).
    * UrbanPop: numeric, Percent urban population(per 100,000).
    * Rape : numeric, Rape arrests (per 100,000).

Procedure

1. Data preparation:
   - Remove missing and junk values.
   - Scale the variables to make them equal.
2. Split the data set as (Optional):
   - Training dataset.
   - Testing dataset.
3. Implementation of clustering methodologies:
   - Computation and visualisation of k-means clustering.
   - Computation and visualisation of k-medoids clustering.
   - Computation and visualisation of PAM clustering.
   - Computation and visualisation of Hierarchial clustering.
4. Enhanced clustering visualisation using factoextra.
5. Determining optimal number of clusters.

Clustering Algorithms

- **The k-Means Algorithm - Introduction**:
  1. Split the input as k categories.
  2. Required: A distance measure - Eucilidean distance
  3. Processing : Clustering based on seed point computed by mean average.

- Pseudocode :

```
• Initialisation:
  * Choose a value for k.
  * Choose k random positions in the input space.
  * Assign the cluster centres $\mu_j$ to those positions.
• Training
- Repeat
  * for each datapoint x i :
      · compute the distance to each cluster centre.
      · assign the datapoint to the nearest cluster centre with distance.
            $d_i = min_j d(x_i, \mu_j).$   (1)
  * for each cluster centre :
      · move the position of the centre to the mean of the points in that cluster (N_j is the number of points in cluster j):
            $\mu_j = \frac{1}{N_j} \sum_{i=1}^{N_j} x_i$   (2)
- until the cluster centres stop moving
• Usage
  * for each test point :
      · Compute the distance to each cluster centre.
      · Assign the datapoint to the nearest cluster centre with distance.
            $d_i = min_j d(x_i, \mu_j).$   (3)
```

Algorithm 1: The k-Means Algorithm

- **K-medoids Algorithm - Introduction**:
  1. K-means is sensitive to outliers and compute large distance.
  2. Change in distance measure as follows :
     - Given $x = x_1, x_2, \ldots, x_N$
     - $median(x) \in \arg\min_z \sum_{i=1}^{N} |x_i - z|$
     - so could cluster by using the median to construct distance measure. More generally, we can assign cluster center to be the cluster medoid so we set $m_k := x_{i_k}$ where

$$i_k = \arg\min{}_l \sum_{i \in C_k} d(x_l, x_i). where\ l \in C_k$$

  3. Form new clusters by assigning points to the nearest cluster center.

$$C(i) = \arg\min{}_{limit} d(x_i, m_k) where\ limit - 1 \leq k \leq K$$

– Pseudocode :

- **Algorithm :**

1. For a given cluster assignment C find the observation in the cluster minimizing total distance to other points in that cluster.

$$i_\mathrm{k} = \arg\min{}_\mathrm{l} \sum_{i \in C_\mathrm{k}} D(x_\mathrm{i}, x_{\mathrm{i}'}). \tag{4}$$

where, $l \in C_\mathrm{k}, m_\mathrm{k} := x_{\mathrm{i_k}}$ are the current estimates of the cluster centres.

2. Given a current set of cluster centers = $\mathrm{m}_1, \ldots, \mathrm{m}_\mathrm{K}$ minimize the total error by assigning each observation to the closest (current) cluster center:

$$C(i) = \arg\min{}_\mathrm{limit} d(x_\mathrm{i}, m_\mathrm{k}) where\ limit - 1 \leq k \leq K \tag{5}$$

3. Repeat step 1 and 2 until no change in assignment.

Algorithm 2: The k-Medoids Clustering

- **Hierarcial clustering- Introduction**:

1. Clusters at a given level are created by merging clusters at the next lower level.
2. Each cluster at the lowest level contains a single observation.
3. At the highest level there is just one cluster containing all the data.
4. There are two basic paradigms for constructing hierarchical clusters
   - Agglomerative : bottom-up
   - Divisive : top-down

- Pseudocode :

- **Algorithm :**

1. Begin with the disjoint clustering having level L(0) = 0 and sequence number m = 0.
2. Find the least dissimilar pair of clusters in the current clustering, say pair (r), (s), according to.

$$d[(r), (s)] = mind[(i), (j)] \tag{6}$$

where the minimum is over all pairs of clusters in the current clustering.

3. Increment the sequence number : m = m +1. Merge clusters (r) and (s) into a single cluster to form the next clustering m. Set the level of this clustering to

$$L(m) = d[(r), (s)] \tag{7}$$

4. Update the proximity matrix, D, by deleting the rows and columns corresponding to clusters (r) and (s) and adding a row and column corresponding to the newly formed cluster. The proximity between the new cluster, denoted (r,s) and old cluster (k) is defined in this way:

$$d[(k), (r, s)] = mind[(k), (r)], d[(k), (s)] \tag{8}$$

5. If all objects are in one cluster, stop. Else, go to step 2.

Algorithm 3: Hierarcial clustering

Confusion Matrix

- A confusion matrix is a table that can be generated for a classifier on a Data Set

  True Positives(TP)- These are the cases where the predicted and actual both are yes.
  True Negatives(TN)- These are the cases where the predicted value is no and actual value is yes.
  False Positive(FP)- These are the cases where the predicted value is yes and actual value is no.
  False Negative(FN)- These are the cases where prediction is no and actual value is no.

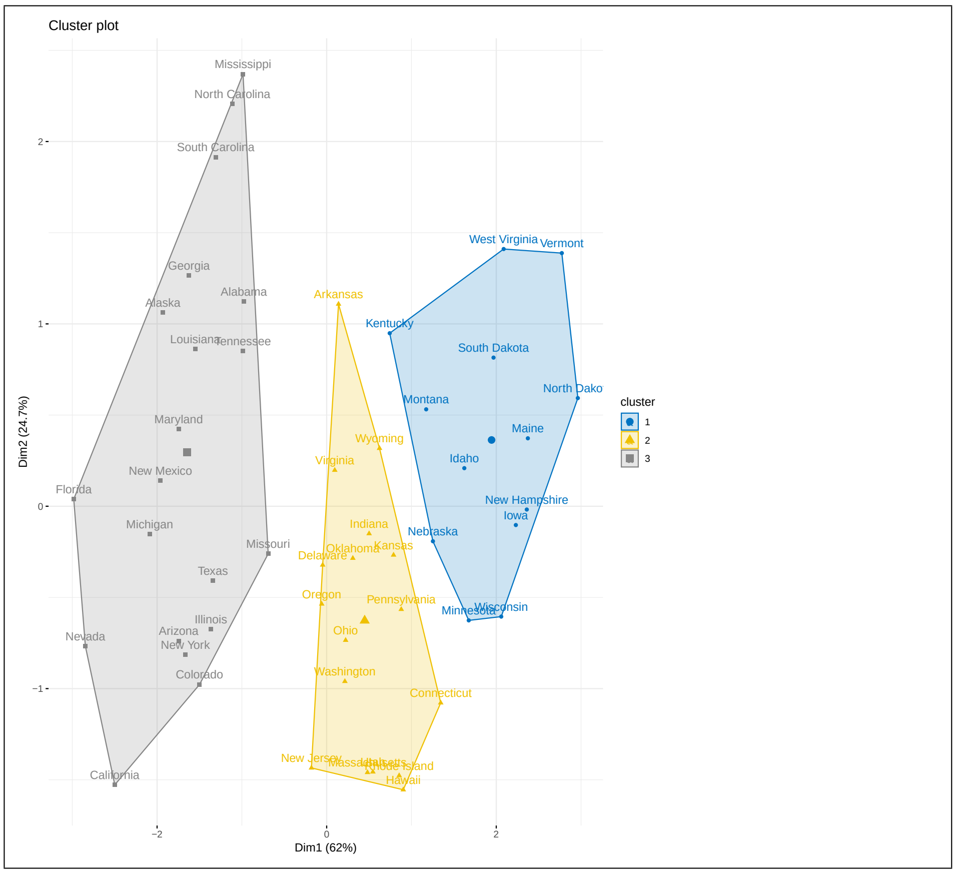
Coding

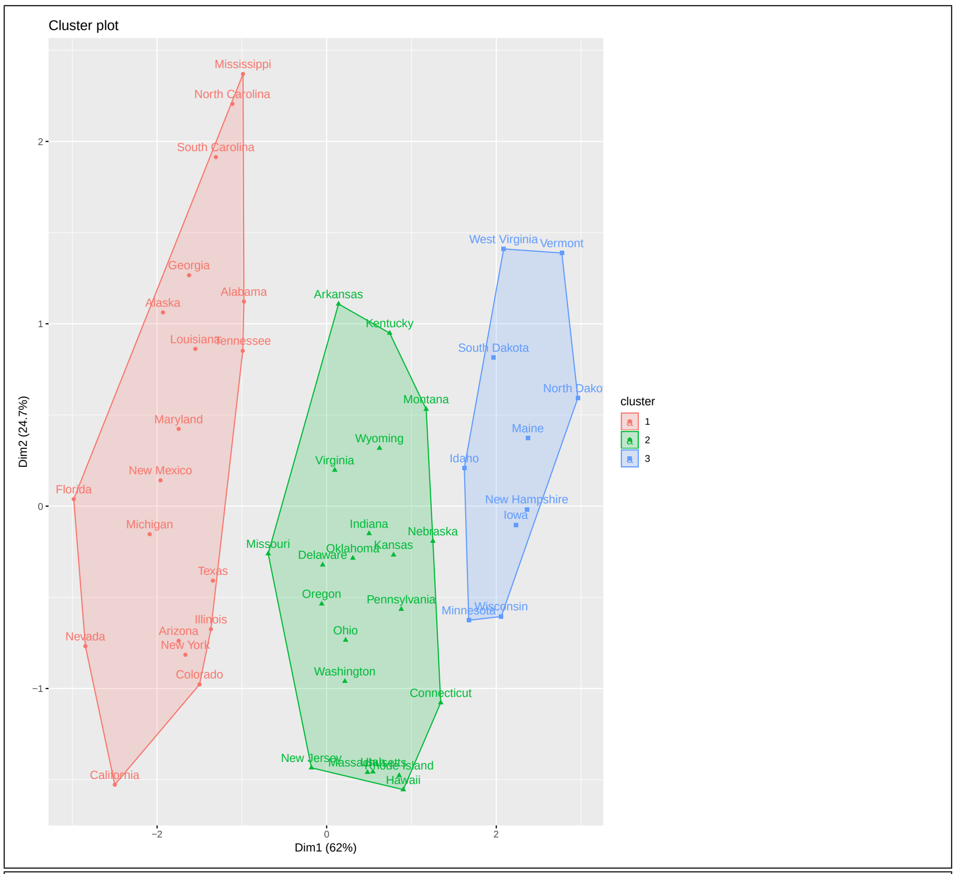
```
#Installing and loading required R packages
# ----------------------------------------------------------------------------------
#We'll use mainly two R packages:
#cluster package: for computing clustering
#factoextra package : for elegant ggplot2-based data visualization.
# ----------------------------------------------------------------------------------
#Install:
# ----------------------------------------------------------------------------------
#install.packages("factoextra")
#install.packages("cluster")
#install.packages("magrittr")
#install.packages("NbClust")
#Load packages:
# ----------------------------------------------------------------------------------

library("TeachingDemos")
txtStart("clusteringOutput.txt")
library("cluster")
library("factoextra")
library("magrittr")

#Data preparation
# ----------------------------------------------------------------------------------
#Demo data set: the built-in R dataset named USArrest
#Remove missing data
#Scale variables to make them comparable
# ----------------------------------------------------------------------------------

# Load  and prepare the data
data("USArrests")

my_data <- USArrests %>%
  na.omit() %>%          # Remove missing values (NA)
  scale()                # Scale variables

# View the firt 3 rows
head(my_data, n = 3)

#Partitioning clustering
# --------------------------------------------------------------------------------------------------------------------
#Partitioning algorithms are clustering techniques that subdivide the data sets into a set
#K-means clustering
#K-medoids clustering or PAM (Partitioning Around Medoids)
# --------------------------------------------------------------------------------------------------------------------

fviz_nbclust(my_data, kmeans, method = "gap_stat")
```

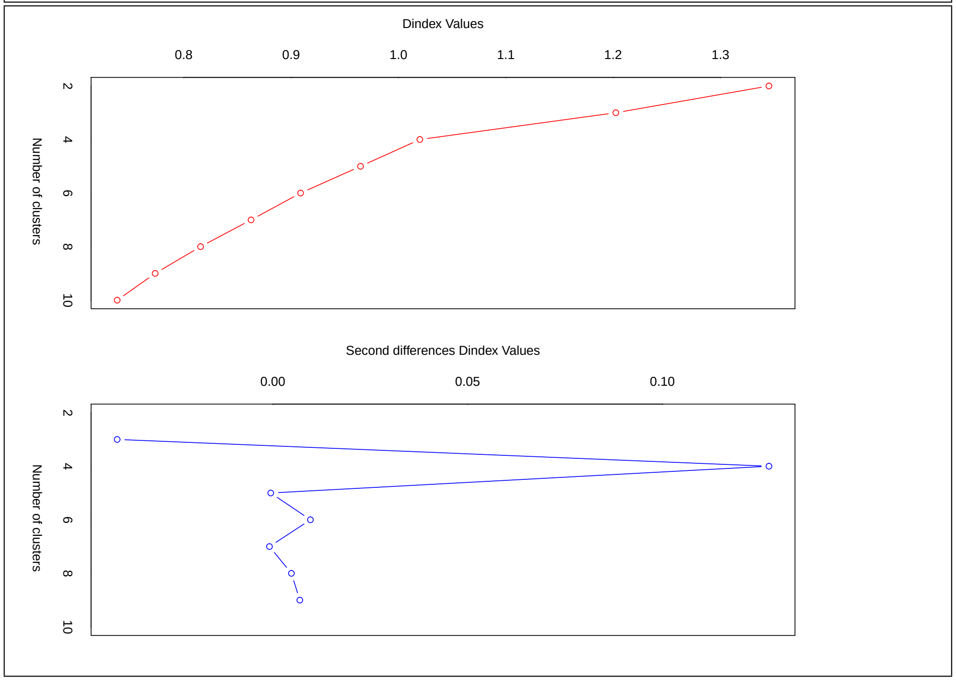
```
#Compute and visualize k-means clustering
# ------------------------------------------------------------------------

set.seed(123)
km.res <- kmeans(my_data, 3, nstart = 25)
# Visualize
fviz_cluster(km.res, data = my_data, ellipse.type = "convex", palette = "jco", ggtheme = t


#Compute and visualize k-medoids clustering
# ------------------------------------------------------------------------

# Compute PAM
library("cluster")
pam.res <- pam(my_data, 3)
# Visualize
fviz_cluster(pam.res)

#Hierarchical clustering
#Hierarchical clustering is an alternative approach to partitioning clustering for identify
# ------------------------------------------------------------------------
#compute and visualize hierarchical clustering:
# ------------------------------------------------------------------------
# Compute hierarchical clustering
res.hc <- USArrests %>%
  scale() %>%                    # Scale the data
  dist(method = "euclidean") %>% # Compute dissimilarity matrix
  hclust(method = "ward.D2")     # Compute hierachical clustering

# Visualize using factoextra
# Cut in 4 groups and color by groups
fviz_dend(res.hc, k = 4, # Cut in four groups
          cex = 0.5, # label size
          k_colors = c("#2E9FDF", "#00AFBB", "#E7B800", "#FC4E07"),
          color_labels_by_k = TRUE, # color labels by groups
          rect = TRUE # Add rectangle around groups
)

#Determining the optimal number of clusters
# ------------------------------------------------------------------------
set.seed(123)

# Compute
library("NbClust")
res.nbclust <- USArrests %>%
  scale() %>%
  NbClust(distance = "euclidean",
          min.nc = 2, max.nc = 10,
          method = "complete", index ="all")

# Visualize
library(factoextra)
fviz_nbclust(res.nbclust, ggtheme = theme_minimal())
txtStop()
```

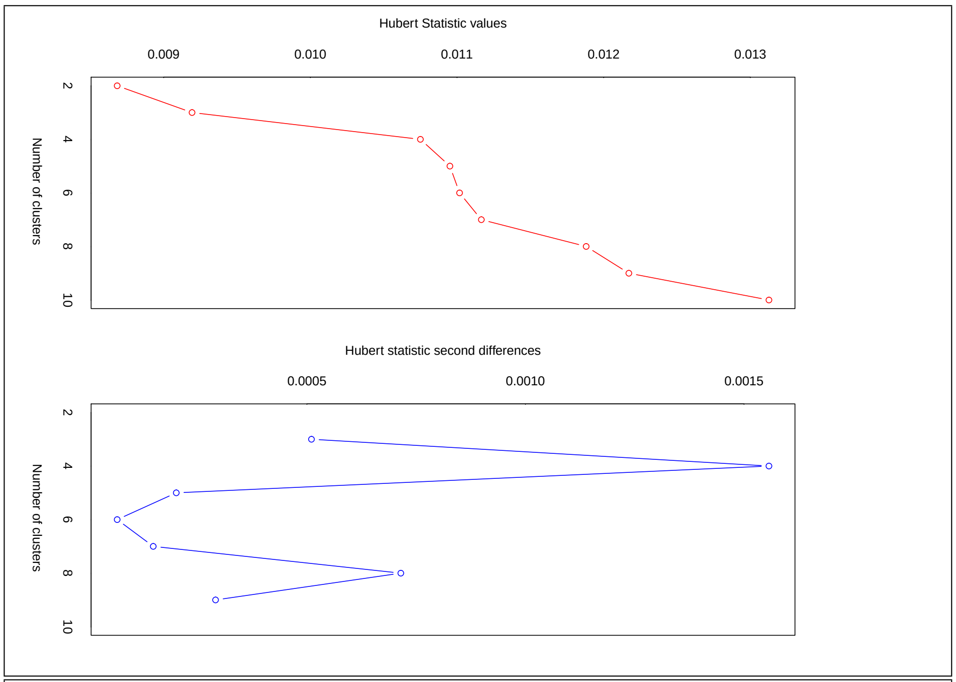
Output

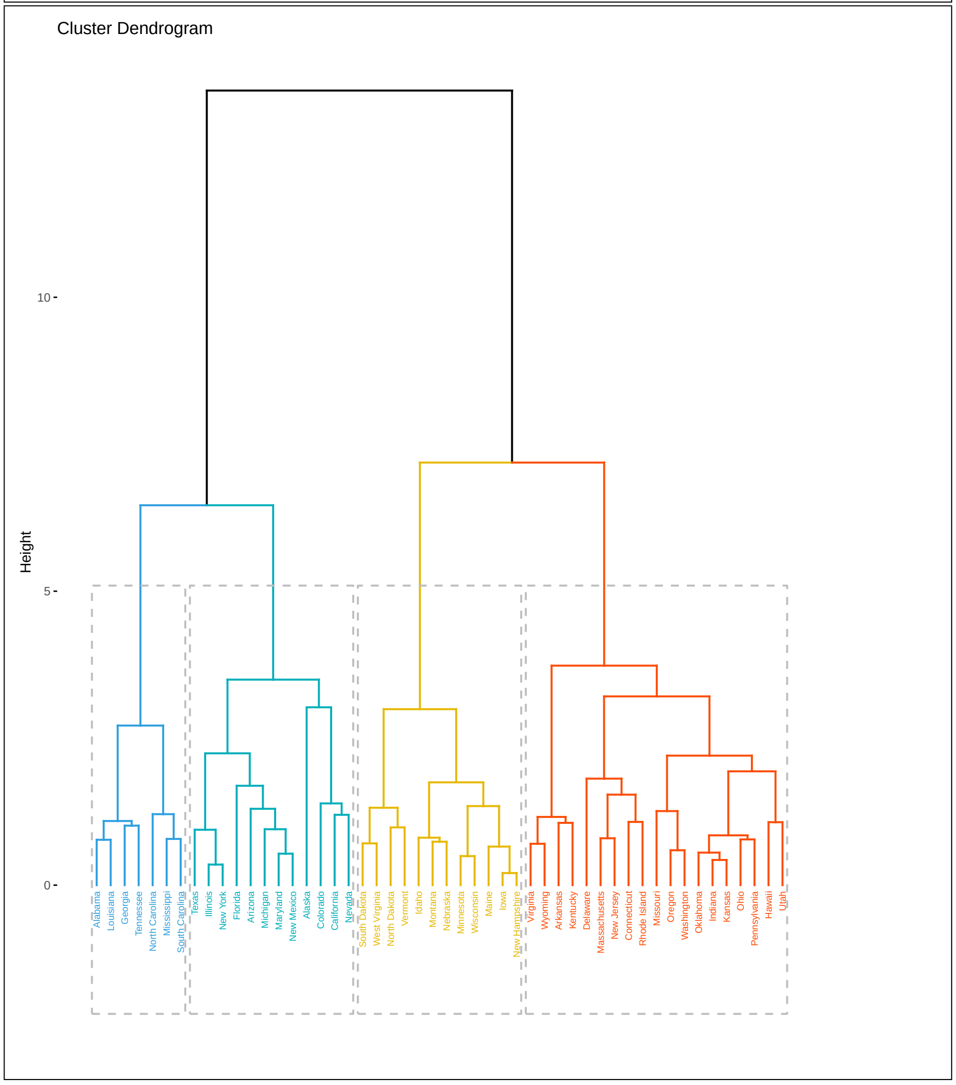
```
> library("cluster")
> library("factoextra")
> library("magrittr")
> data("USArrests")
> my_data <- USArrests %>% na.omit() %>% scale()
> head(my_data, n = 3)
            Murder   Assault   UrbanPop         Rape
Alabama 1.24256408 0.7828393 -0.5209066 -0.003416473
Alaska  0.50786248 1.1068225 -1.2117642  2.484202941
Arizona 0.07163341 1.4788032  0.9989801  1.042878388
> fviz_nbclust(my_data, kmeans, method = "gap_stat")
Clustering k = 1,2,..., K.max (= 10): .. done
Bootstrapping, b = 1,2,..., B (= 100)  [one "." per sample]:
.................................................. 50
.................................................. 100
> set.seed(123)
> km.res <- kmeans(my_data, 3, nstart = 25)
> fviz_cluster(km.res, data = my_data, ellipse.type = "convex",
+ palette = "jco", ggtheme = theme_minimal())
> library("cluster")
> pam.res <- pam(my_data, 3)
> fviz_cluster(pam.res)
> res.hc <- USArrests %>% scale() %>% dist(method = "euclidean") %>%
+ hclust(method = "ward.D2")
> fviz_dend(res.hc, k = 4, cex = 0.5, k_colors = c("#2E9FDF", "#00AFBB",
+ "#E7B800", "#FC4E07"), color_labels_by_k = TRUE, rect = TRUE)
> set.seed(123)
> library("NbClust")
> res.nbclust <- USArrests %>% scale() %>% NbClust(distance = "euclidean",
+ min.nc = 2, max.nc = 10, method = "complete", index = "all")
*** : The Hubert index is a graphical method of determining the number of clusters.
                In the plot of Hubert index, we seek a significant knee that corresponds to a
                significant increase of the value of the measure i.e the significant peak in Hubert
                index second differences plot.

*** : The D index is a graphical method of determining the number of clusters.
                In the plot of D index, we seek a significant knee (the significant peak in Dindex
                second differences plot) that corresponds to a significant increase of the value of
                the measure.

*******************************************************************
* Among all indices:
* 9 proposed 2 as the best number of clusters
* 4 proposed 3 as the best number of clusters
* 6 proposed 4 as the best number of clusters
* 2 proposed 5 as the best number of clusters
* 1 proposed 8 as the best number of clusters
* 1 proposed 10 as the best number of clusters

                   ***** Conclusion *****

* According to the majority rule, the best number of clusters is  2


*******************************************************************
> library(factoextra)
> fviz_nbclust(res.nbclust, ggtheme = theme_minimal())
Among all indices:
===================
* 2 proposed  0 as the best number of clusters
* 1 proposed  1 as the best number of clusters
* 9 proposed  2 as the best number of clusters
* 4 proposed  3 as the best number of clusters
* 6 proposed  4 as the best number of clusters
* 2 proposed  5 as the best number of clusters
* 1 proposed  8 as the best number of clusters
* 1 proposed  10 as the best number of clusters

Conclusion
=========================
* According to the majority rule, the best number of clusters is  2 .
```

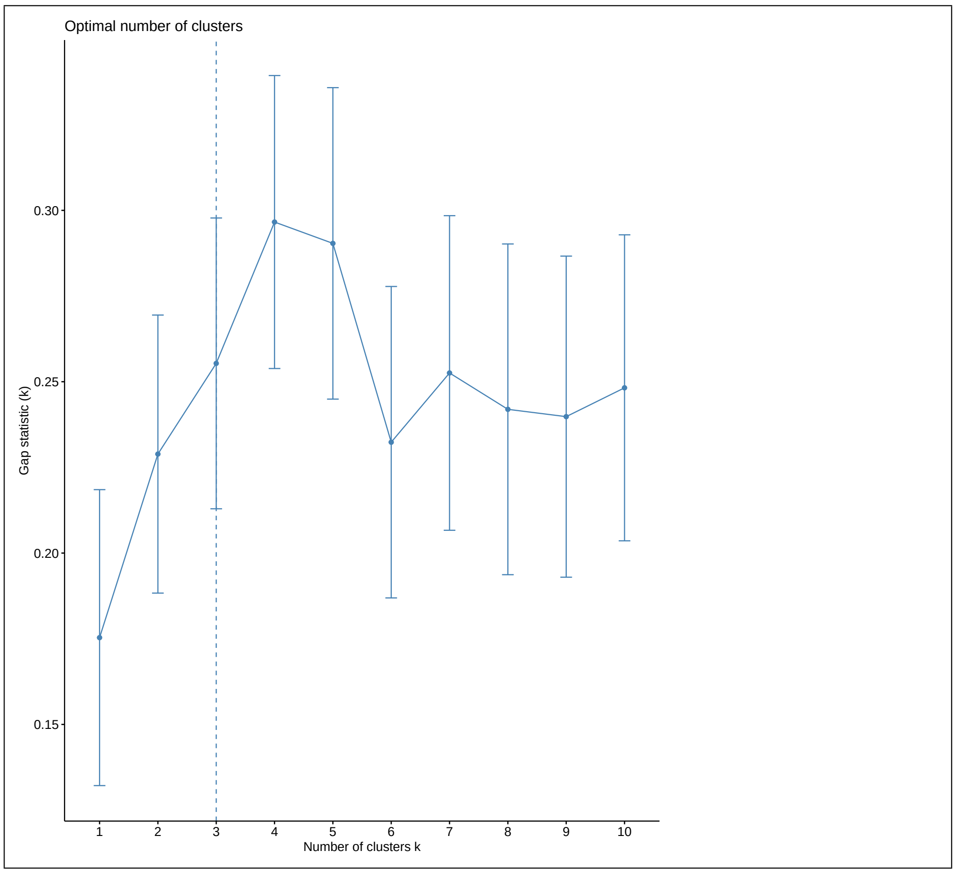



Cluster plot
Mississippi
North Carolina
South Carolina
Georgia
Alabama
Alaska
Louisiana
Tennessee
Maryland
New Mexico
Florida
Michigan
Texas
Illinois
Arizona
Nevada
New York
Colorado
California
Arkansas
Kentucky
Montana
Wyoming
Virginia
Indiana
Nebraska
Missouri
Oklahoma
Kansas
Delaware
Oregon
Pennsylvania
Ohio
Washington
Connecticut
New Jersey
Massachusetts
Rhode Island
Hawaii
West Virginia
Vermont
South Dakota
North Dako
Maine
Idaho
New Hampshire
Iowa
Minnesota
Wisconsin
cluster
1
2
3
Dim1 (62%)
Dim2 (24.7%)


Dindex Values
0.8
0.9
1.0
1.1
1.2
1.3
Number of clusters
2
4
6
8
10
Second differences Dindex Values
0.00
0.05
0.10
Number of clusters


Hubert Statistic values
0.009
0.010
0.011
0.012
0.013
Number of clusters
2
4
6
8
10
Hubert statistic second differences
0.0005
0.0010
0.0015
Number of clusters
2
4
6
8
10


Cluster Dendrogram
Height
0
5
10
Alabama
Louisiana
Georgia
Tennessee
North Carolina
Mississippi
South Carolina
Texas
Illinois
New York
Florida
Arizona
Michigan
Maryland
New Mexico
Alaska
Colorado
California
Nevada
South Dakota
West Virginia
North Dakota
Vermont
Idaho
Montana
Nebraska
Minnesota
Wisconsin
Maine
Iowa
New Hampshire
Virginia
Wyoming
Arkansas
Kentucky
Delaware
Massachusetts
New Jersey
Connecticut
Rhode Island
Missouri
Oregon
Washington
Oklahoma
Indiana
Kansas
Ohio
Pennsylvania
Hawaii
Utah



Result

Thus the implementation of Support Vector machine is executed successfully using R program.