EX.NO:9 **210701167**

Implement clustering techniques – Hierarchical and K-Means

AIM:

To Implement clustering techniques Hierarchical and K-Means using R programming in R Studio.

a) HIERARCHIAL CLUSTERING

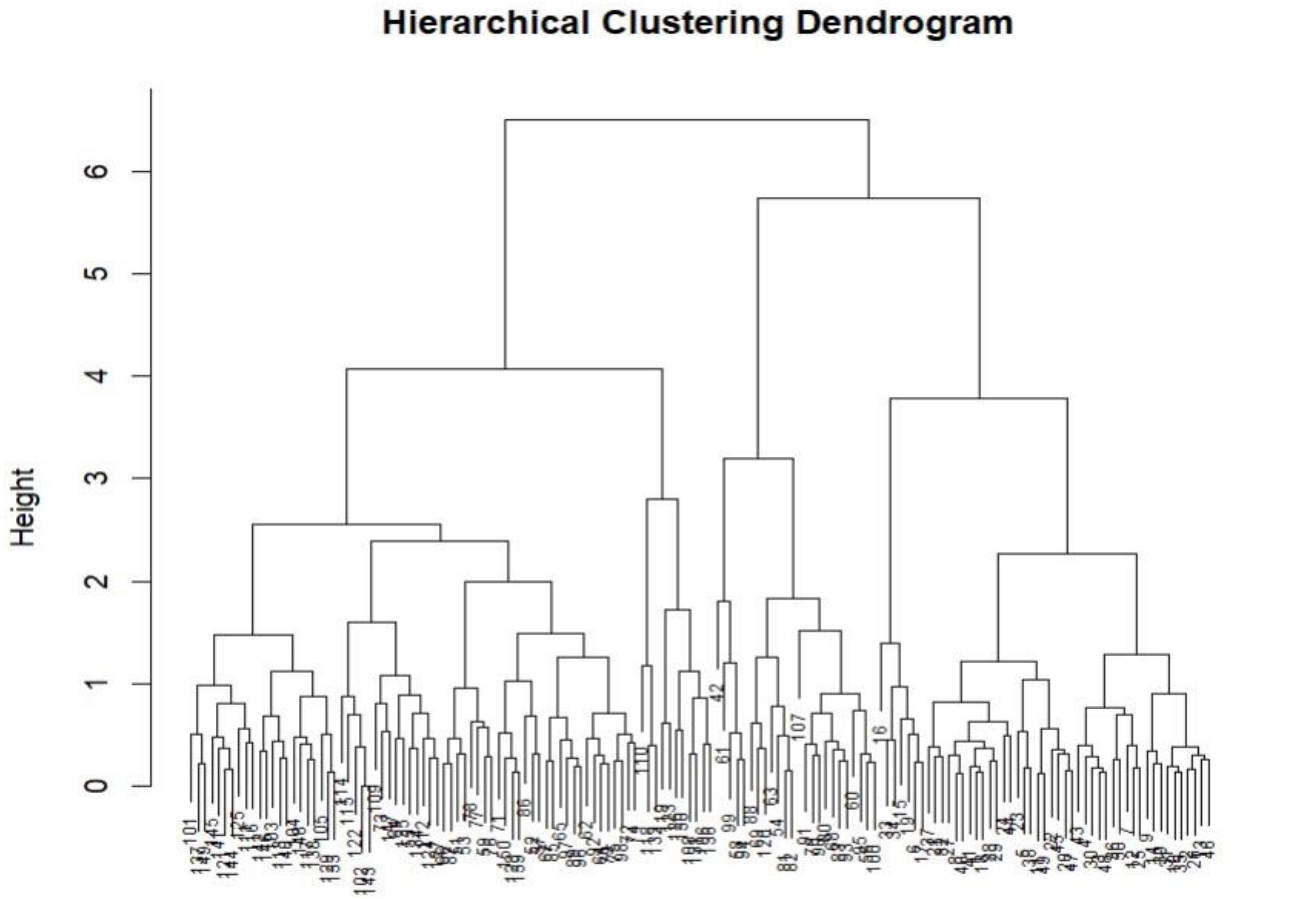
```
# Load the iris dataset data(iris)

# Use only the numeric columns for clustering (exclude the Species column) iris_data <-
iris[, -5]

# Standardize the data  iris_scaled
<- scale(iris_data)

# Compute the distance matrix distance_matrix <- dist(iris_scaled,
method = "euclidean")

# Perform hierarchical clustering using the "complete" linkage method hc_complete <-
hclust(distance_matrix, method = "complete")

# Plot the dendrogram plot(hc_complete, main = "Hierarchical Clustering Dendrogram", xlab
= "", sub = "", cex =
0.6)

# Cut the tree to form 3 clusters
clusters <- cutree(hc_complete, k = 3)

# Print the cluster memberships print(clusters)

# Add the clusters to the original dataset  iris$Cluster
<- as.factor(clusters)

# Display the first few rows of the updated dataset head(iris)
```

EX.NO:9 **210701167**

OUTPUT:

Hierarchial.R

```r
# Load the iris dataset
data(iris)
# Use only the numeric columns for clustering (exclude the Species column)
iris_data <- iris[, -5]
# Standardize the data
iris_scaled <- scale(iris_data)
# Compute the distance matrix
distance_matrix <- dist(iris_scaled, method = "euclidean")
# Perform hierarchical clustering using the "complete" linkage method
hc_complete <- hclust(distance_matrix, method = "complete")
# Plot the dendrogram
plot(hc_complete, main = "Hierarchical Clustering Dendrogram", xlab = "", sub = "", cex =
     0.6)
# Cut the tree to form 3 clusters
clusters <- cutree(hc_complete, k = 3)
# Print the cluster memberships
print(clusters)
# Add the clusters to the original dataset
iris$Cluster <- as.factor(clusters)
# Display the first few rows of the updated dataset
head(iris)
```



EX.NO:9 **210701167**

b) K-MEANS CLUSTERING

```
# Load the iris dataset data(iris)

# Use only the numeric columns for clustering (exclude the Species column) iris_data <-
iris[, -5]

# Standardize the data
iris_scaled <- scale(iris_data)

# Set the number of clusters set.seed(123)
# For reproducibility  k <- 3
# Number of clusters

# Perform K-Means clustering
kmeans_result <- kmeans(iris_scaled, centers = k, nstart = 25)

# Print the K-Means result
print(kmeans_result)

# Print the cluster centers
print(kmeans_result$centers)

# Add the cluster assignments to the original dataset iris$Cluster
<- as.factor(kmeans_result$cluster)

# Display the first few rows of the updated dataset head(iris)

# Plot the clusters library(ggplot2)
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Cluster)) +
geom_point(size = 3) +
  labs(title = "K-Means Clustering of Iris Dataset", x = "Sepal Length", y = "Sepal Width")
```

OUTPUT:

EX.NO:9 **210701167**

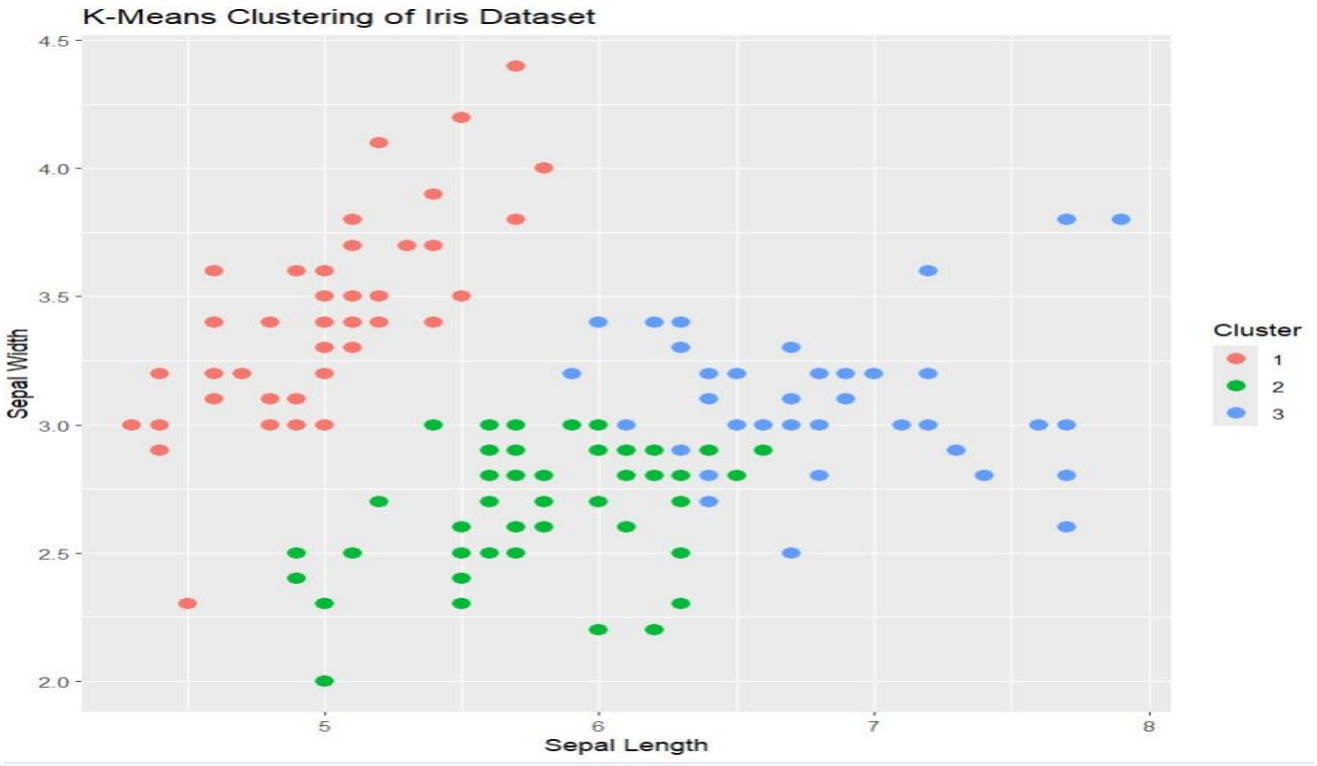
Hierarchial.R | K-Means.R*

Source on Save | Run | Source

```
# Load the iris dataset
data(iris)
# Load the iris dataset
data(iris)
# Use only the numeric columns for clustering (exclude the Species column)
iris_data <- iris[, -5]
# Standardize the data
iris_scaled <- scale(iris_data)
# Set the number of clusters
set.seed(123) # For reproducibility
k <- 3 # Number of clusters
# Perform K-Means clustering
kmeans_result <- kmeans(iris_scaled, centers = k, nstart = 25)
# Print the K-Means result
print(kmeans_result)
# Print the cluster centers
print(kmeans_result$centers)
# Add the cluster assignments to the original dataset
iris$Cluster <- as.factor(kmeans_result$cluster)
# Display the first few rows of the updated dataset
head(iris)
# Plot the clusters
library(ggplot2)
ggplot(iris, aes(x = Sepal.Length, y = Sepal.Width, color = Cluster)) +
  geom_point(size = 3) +
  labs(title = "K-Means Clustering of Iris Dataset", x = "Sepal Length", y = "Sepal Width")
```

Files | Plots | Packages | Help | Viewer | Presentation

Zoom | Export | Publish



RESULT:

Thus, the Implement clustering techniques Hierarchical and K-Means using R programming in R Studio have been successfully executed.