Exp. No : 9

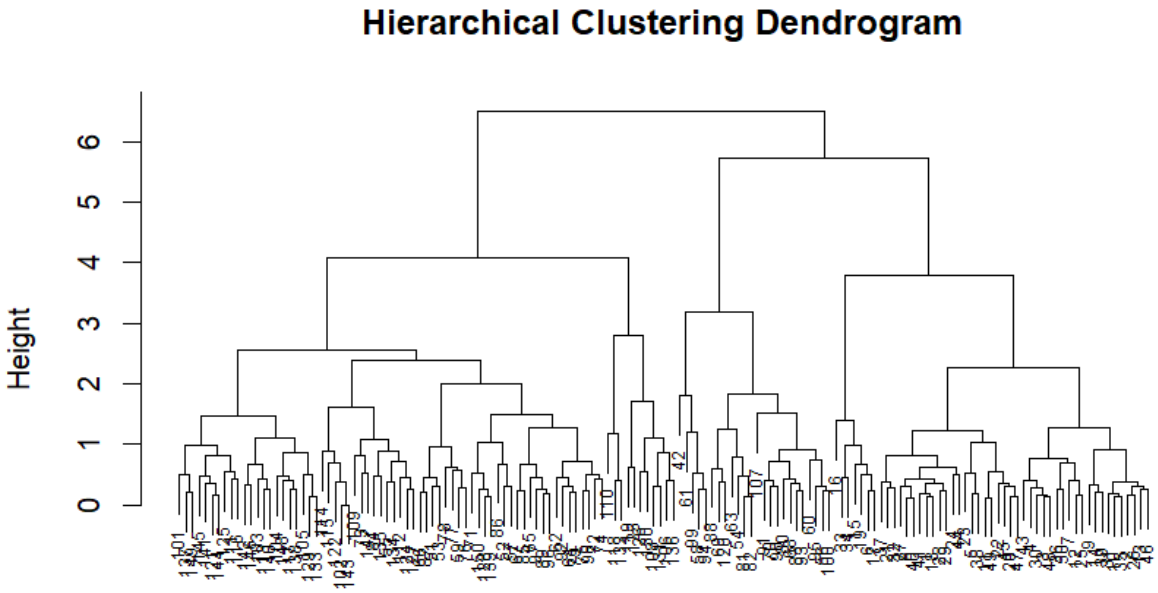
Implement clustering techniques – Hierarchical and K-Means

a) Hierarchical 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)
```

Output :

```
> print(clusters)
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1
 [46] 1 1 1 1 1 3 3 3 2 3 2 3 2 3 2 2 3 2 3 3 3 3 2 2 2 3 3 3 3 3 3 3 3 3 3 2 2 2 3
 [91] 2 3 2 2 2 3 3 3 2 2 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3
[136] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
>
```



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 :

```
> iris_scaled <- scale(iris_data)
> set.seed(123)  # For reproducibility
> k <- 3  # Number of clusters
> kmeans_result <- kmeans(iris_scaled, centers = k, nstart = 25)
> print(kmeans_result)
K-means clustering with 3 clusters of sizes 50, 53, 47

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1  -1.01119138  0.85041372   -1.3006301  -1.2507035
2  -0.05005221 -0.88042696    0.3465767   0.2805873
3   1.13217737  0.08812645    0.9928284   1.0141287

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [46] 1 1 1 1 1 3 3 3 2 2 2 3 2 2 2 2 2 2 2 3 2 2 2 2 3 2 2 2 2 3 3 3 2 2 2 2 2 2
 [91] 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3 3 2 3 3 3 3 3 3 2 2 3 3 3 3 2 3 2 3 2 3 3 2 3
[136] 3 3 3 2 3 3 3 2 3 3 3 2 3 3 2

Within cluster sum of squares by cluster:
[1] 47.35062 44.08754 47.45019
 (between_SS / total_SS =  76.7 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"
[7] "size"         "iter"         "ifault"
>
```

```
The downloaded binary packages are in
        C:\Users\super\AppData\Local\Temp\Rtmp4kW9ap\downloaded_packages
> 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")
> 
```