

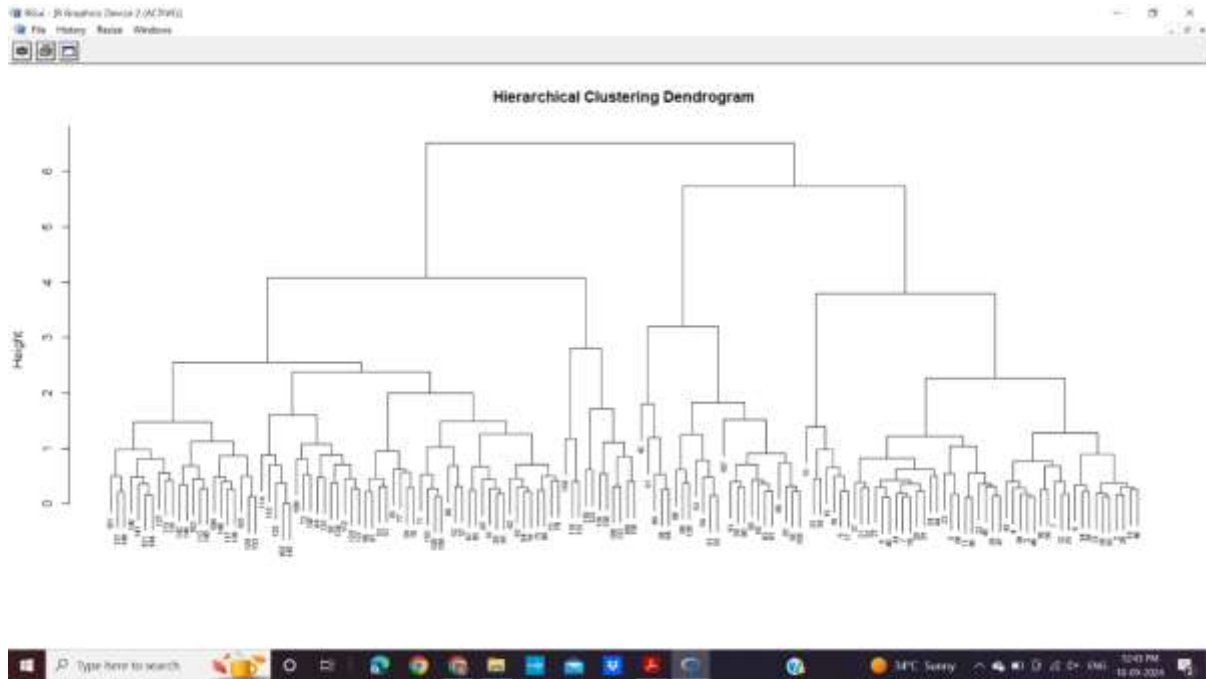
```
# Load the iris dataset
data(iris)
# Use only the numeric columns for clustering (exclude the Species column)
iris_data <- iris[, -5]
# Standardize the data
iris_scaled <- scale(iris_data)
# Compute the distance matrix
distance_matrix <- dist(iris_scaled, method = "euclidean")
# Perform hierarchical 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)
```

The screenshot shows an RStudio window with a script for hierarchical clustering. The script performs the following steps:

- Load the `iris` dataset.
- Use only the numeric columns for clustering (excluding the `Species` column).
- Standardize the data.
- Compute the distance matrix.
- Generate a dendrogram using the `dist` function.
- Perform hierarchical clustering using the `complete` linkage method.
- Print the dendrogram.
- Plot the dendrogram using `plot` and `text` functions.
- Use the `cutree` function to cut the dendrogram into 3 clusters.
- Print the cluster memberships.
- Print the clusters to the original dataset.
- Display the first few rows of the updated dataset.

The output of the script is a table showing the first few rows of the `iris` dataset with the `Species` column updated based on the clustering results.

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species	Cluster
1	5.1	3.5	1.4	0.2	setosa	1
2	4.9	3.0	1.4	0.2	setosa	1
3	4.7	3.1	1.3	0.2	setosa	1
4	4.6	3.1	1.3	0.2	setosa	1
5	5.1	3.6	1.4	0.2	setosa	1
6	5.0	3.8	1.7	0.2	setosa	1



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: