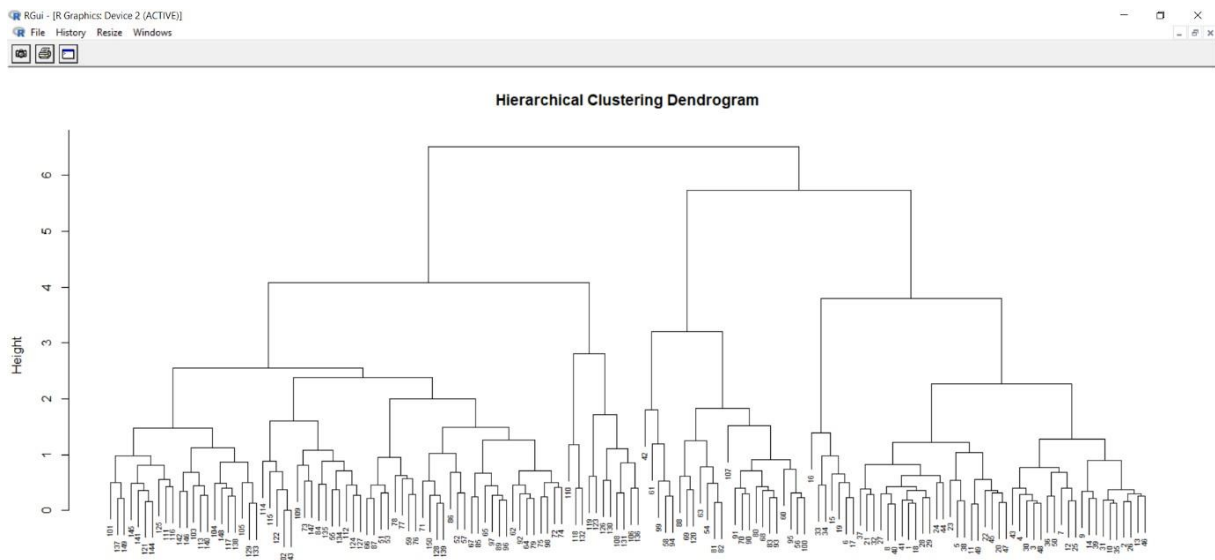


```
# Load the iris dataset
data(iris)
# Use only the numeric columns for clustering (exclude the Species column)
iris_data <- iris[, -5]
# Standardize the data
iris_scaled <- scale(iris_data)
# Compute the distance matrix
distance_matrix <- dist(iris_scaled, method = "euclidean")
# Perform hierarchical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

- R Console:** Contains R code for data analysis and clustering. The code includes:
  - Printing a confusion matrix for the iris dataset.
  - Calculating accuracy using `sum(diag(confusion_matrix)) / sum(confusion_matrix)`.
  - Loading the iris dataset and selecting numeric columns for clustering.
  - Standardizing the data using `scale()`.
  - Computing the distance matrix using `dist()`.
  - Performing hierarchical clustering using the "complete" linkage method with `hclust()`.
  - Plotting the dendrogram using `plot(hc_complete, main = "Hierarchical Clustering Dendrogram", xlab = "", sub = "", cex = 0.6)`.
  - Cutting the tree to form 3 clusters using `cutree(hc_complete, k = 3)`.
  - Printing the cluster memberships using `print(clusters)`.
  - Adding the clusters to the original dataset using `iris$Cluster <- as.factor(clusters)`.
  - Displaying the first few rows of the updated dataset using `head(iris)`.
- R Editor:** Displays the output of the R code, showing the confusion matrix, accuracy, and the first few rows of the updated dataset.
 

| Head.Length | Sepal.Width | Petal.Length | Petal.Width | Species | Cluster |
|-------------|-------------|--------------|-------------|---------|---------|
| 5.1         | 3.5         | 1.4          | 0.2         | setosa  | 1       |
| 4.5         | 3.0         | 1.4          | 0.2         | setosa  | 1       |
| 4.7         | 3.2         | 1.3          | 0.2         | setosa  | 1       |
| 4.6         | 3.1         | 1.5          | 0.2         | setosa  | 1       |
| 5.0         | 3.6         | 1.4          | 0.2         | setosa  | 1       |
| 5.4         | 3.9         | 1.7          | 0.4         | 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:

