

**STA401**

**Fall 2024**

**Dr. Ayman Alzaatreh**

**Homework 5**

**Halla, Rumisal, Eleonara**

**3/11/2024**

**Question 3:**

NOTE: ChatGPT was used to assist in the tabulation and the plotting of the data

#a

# Step 1: Create the data frame

data <- data.frame(

Obs = 1:6,

X1 = c(1, 1, 0, 5, 6, 4),

X2 = c(4, 3, 4, 1, 2, 0)

)

# Print the data frame as a table

print("Data Table:")

print(data)

# Step 2: Plot the data

plot(data$X1, data$X2,

xlab = "X1", ylab = "X2",

main = "Scatter Plot of Observations",

pch = 4, col = "blue", xlim = c(-1, 7), ylim = c(-1, 5))

# Step 3: Add labels to each point

text(data$X1 + 0.2, data$X2 + 0.2, labels = data$Obs, cex = 0.8, col = "red")

#b

set.seed(1) # Set seed for reproducibility

data$Cluster <- sample(1:2, nrow(data), replace = TRUE)

# Display the data with the randomly assigned clusters

print("Data with Randomly Assigned Clusters:")

print(data)

#c: Compute the centroids for each cluster

centroids <- aggregate(cbind(X1, X2) ~ Cluster, data = data, FUN = mean)

# Print the centroids

print("Centroids of Each Cluster:")

print(centroids)

# Step 5: Assign each observation to the nearest centroid

# Initialize a vector to hold the closest cluster labels

closest\_clusters <- numeric(nrow(data))

# Loop through each observation

for (i in 1:nrow(data)) {

# Calculate the Euclidean distances from the observation to each centroid

distances <- sqrt((data$X1[i] - centroids$X1)^2 + (data$X2[i] - centroids$X2)^2)

# Find the index of the minimum distance

closest\_clusters[i] <- which.min(distances)

}

# Add the closest cluster labels to the data frame

data$ClosestCluster <- closest\_clusters

# Print the observations with their closest cluster labels

print("Observations with Closest Cluster Labels:")

print(data)

# Step 6: Plot the data colored by the closest cluster labels

custom\_colors <- c("red", "blue")

plot(data$X1, data$X2,

xlab = "X1", ylab = "X2",

main = "Scatter Plot of Observations by Cluster",

pch = 16, # Use filled circles for points

col = custom\_colors[data$ClosestCluster], # Use custom colors to color by closest cluster

xlim = c(-1, 7), ylim = c(-1, 5))

# Add labels to each point

text(data$X1 + 0.2, data$X2 + 0.2, labels = data$Obs, cex = 0.8, col = "black")

# Optionally, add legend for clusters

legend("topright", legend = unique(data$ClosestCluster), col = unique(custom\_colors[data$ClosestCluster]), pch = 16, title = "Cluster")

A diagram of a cluster

Description automatically generated

**Question 9:**

#a

setwd("/Users/halla.d/Library/Mobile Documents/com~apple~CloudDocs/Desktop/STA401")

data("USArrests")

head(USArrests)

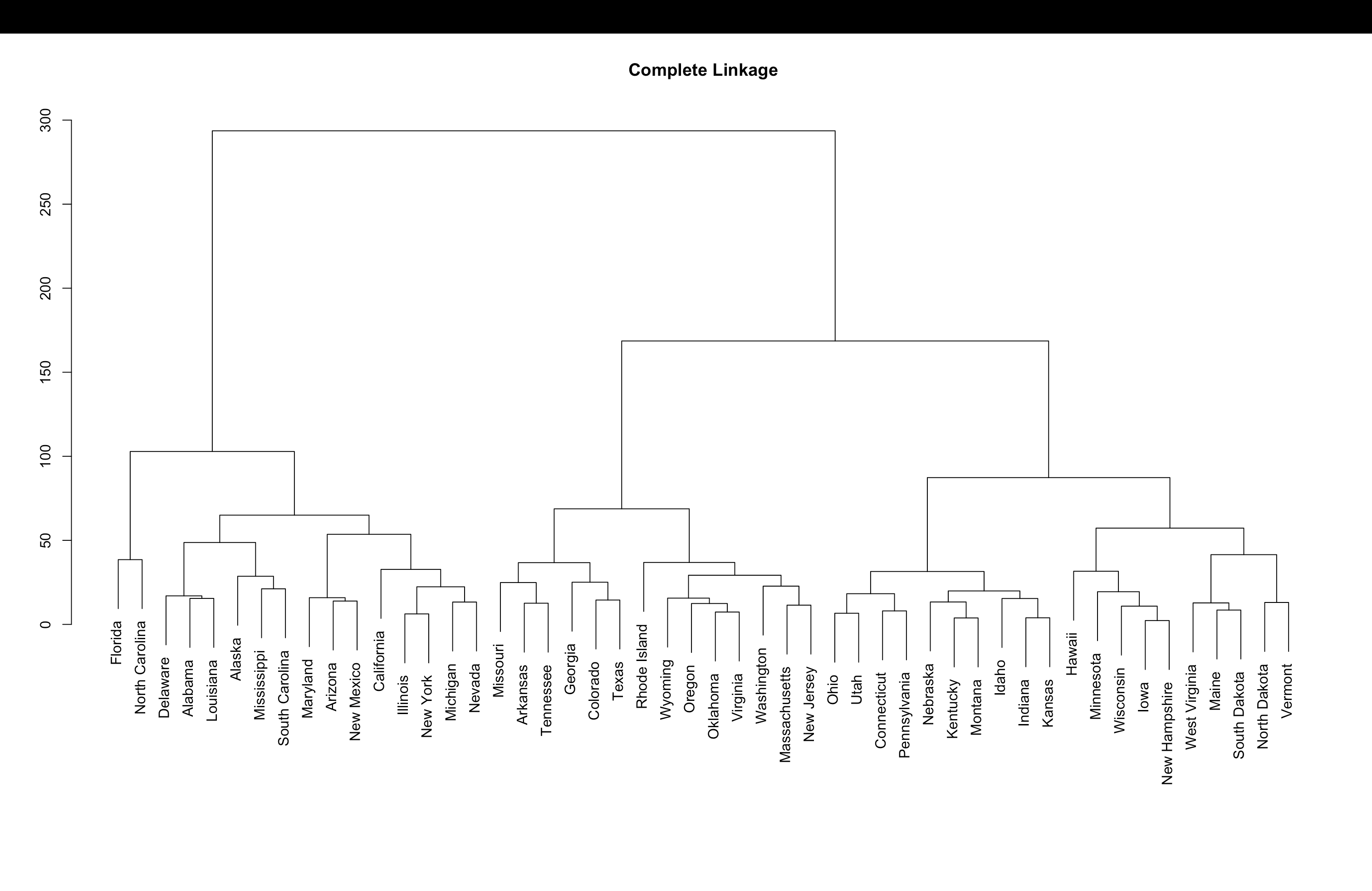
#sd.data=scale(USArrests)

USArrests.rowname <- rownames(USArrests)

data.dist=dist(USArrests) #computes Euclidean distance matrix of the scaled data

hclust(data.dist) #default method = complete

plot(hclust(data.dist), labels= USArrests.rowname, main="Complete Linkage", xlab="", sub="",ylab="")



#b

abline(h=180, col="red")

A screenshot of a computer screen

Description automatically generated

#c

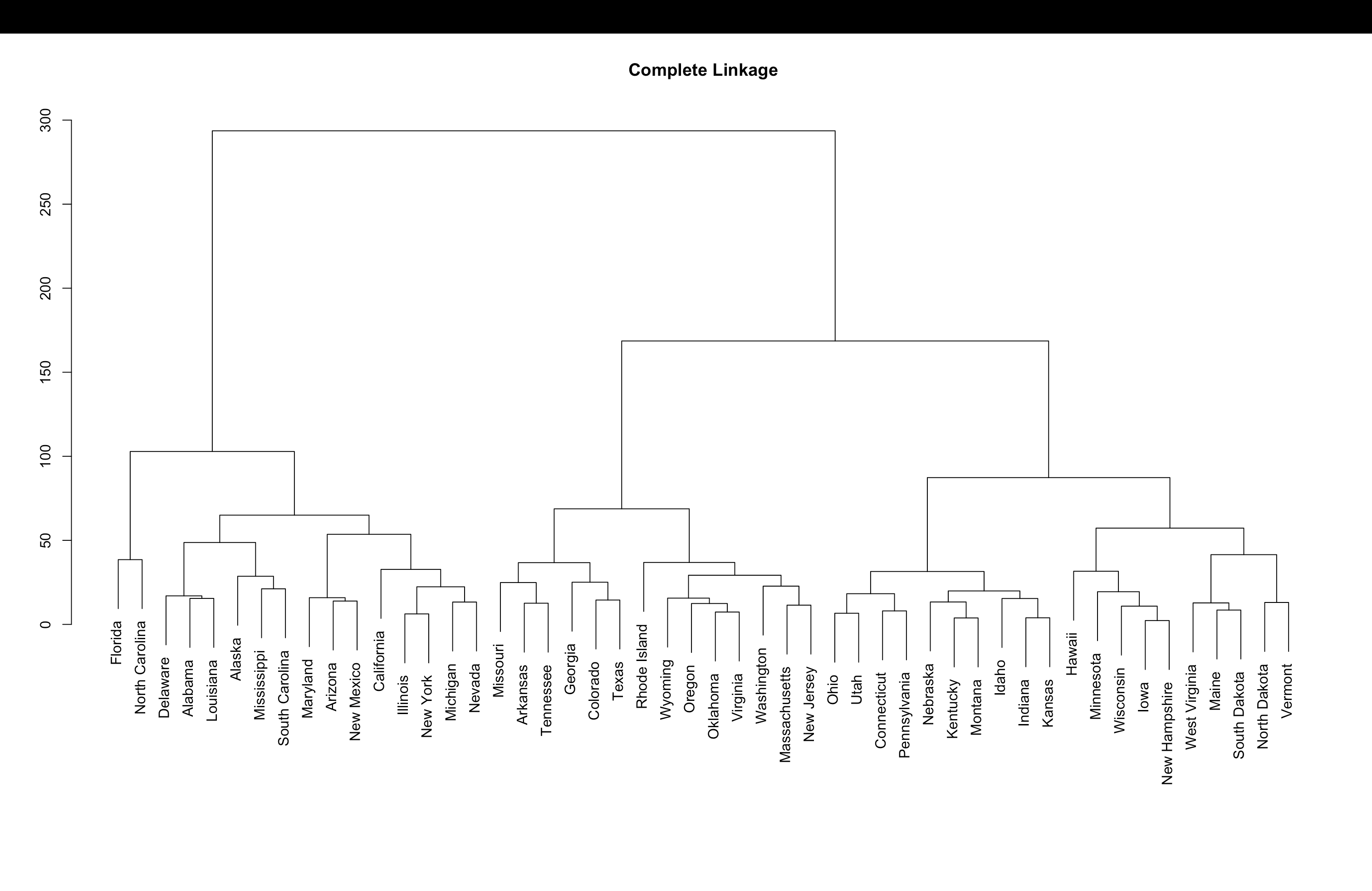
sd.data=scale(USArrests)

USArrests.rowname <- rownames(USArrests)

data.dist=dist(USArrests) #computes Euclidean distance matrix of the scaled data

hclust(data.dist) #default method = complete

plot(hclust(data.dist), labels= USArrests.rowname, main="Complete Linkage", xlab="", sub="",ylab="")



d) Scaling ensures that each variable contributes equally to the distance metric. When variables are not scaled, the clustering results may be skewed, leading to different clusters than those obtained with scaled variables. Additionally, when distances are computed on scaled data, visualizations like dendrograms or cluster plots are easier to interpret, making it clearer how observations group together. For these reasons, I think variables should generally be scaled before computing inter-observation dissimilarities in hierarchical clustering.