**Problem 1 (10 points). The k-means algorithm, which we discussed in the class, is being run on a small two-dimensional dataset. After a certain number of iterations, you have two clusters as shown below:**

**Run one more iteration of the k-Means clustering algorithm and show the two clusters at the end of the iteration. Use Manhattan distance when calculating the distance between objects.**

ID x y Cluster

*<chr>* *<dbl>* *<dbl>* *<dbl>*

1 a 3 6 1

2 b 7 2 2

3 c 5 6 1

4 d 4 5 1

5 e 1 5 1

6 f 5 3 2

7 g 6 7 1

**Code:**  
> data <- data.frame(

+ ID = c('a', 'b', 'c', 'd', 'e', 'f', 'g'),

+ x = c(3, 7, 5, 4, 1, 5, 6),

+ y = c(6, 2, 6, 5, 5, 3, 7),

+ Cluster = c(1, 1, 1, 1, 2, 2, 2)

+ )

>

> # Calculate Manhattan distance

> manhattan\_distance <- function(x1, y1, x2, y2) {

+ abs(x1 - x2) + abs(y1 - y2)

+ }

>

> # Function to assign clusters based on Manhattan distance

> assign\_clusters <- function(data, centroids) {

+ apply(data[,c('x', 'y')], 1, function(row) {

+ distances <- apply(centroids, 1, function(centroid) {

+ manhattan\_distance(row[1], row[2], centroid['x'], centroid['y'])

+ })

+ return(which.min(distances))

+ })

+ }

>

> # Calculate initial centroids

> centroids <- aggregate(cbind(x, y) ~ Cluster, data, median)

>

> # Assign points to the nearest centroids

> data$Cluster <- assign\_clusters(data, centroids)

>

> # Recalculate the centroids after reassigning the points

> new\_centroids <- aggregate(cbind(x, y) ~ Cluster, data, median)

>

> # Print out the updated data with new cluster assignments

> print(data)

ID x y Cluster

1 a 3 6 1

2 b 7 2 2

3 c 5 6 1

4 d 4 5 1

5 e 1 5 1

6 f 5 3 2

7 g 6 7 1

**Problem 2 (10 points). Consider the following two clusters:**

**(1). Compute the distance between the two clusters using the average method that we discussed in the class. Use the Euclidean distance measure when calculating the distance between objects.**

= √26

= √45

= √17

= √26

= 3

= √20

Average distance = (√26 + √45 + √17 + √26 + 3 + √20) / 6 = 4.7502

**(2). Compute the distance between the two clusters using the Ward's method that we discussed in the class. Use the Euclidean distance measure when calculating the distance between objects.**

Centroid1 = ((2+3+4)/3, (5+7+6)/3) = (3,6)

Centroid2 = ((7+8)/2, (6+8)/2) = (7.5,7)

Distance = = 4.61

**Problem 3 (10 points). Use hw8\_p3.csv file. (1). Standardize all columns using the scale function. (2). Run the k-means algorithm using fviz\_nbclust 10 times (this is default) with the wss method. (3). Show all 10 wss values. (4). Plot wss-k graph. Can you see an obvious elbow? (5). Run the k-means algorithm using fviz\_nbclust 10 times (this is default) with silhouette method and plot silhouette-k graph. What is the optimal number of clusters? (6). Run the k-means algorithm once more with the following code segment: set.seed(31) k3 <- kmeans(task.df, centers = 3, nstart = 25) From the clustering result, prepare the following table, which is a cluster profile:**

> library(factoextra)

> library(cluster)

>

> data <- read.csv('hw8\_p3.csv')

>

> # Standardize data

> data\_scaled <- scale(data)

>

> wss\_values <- numeric(10)

> for (k in 1:10) {

+ set.seed(123)

+ kmeans\_result <- kmeans(data\_scaled, centers = k, nstart = 25)

+ wss\_values[k] <- kmeans\_result$tot.withinss

+ }

>

> wss\_values

[1] 575.00000 282.45001 199.87648 161.72222 142.70938 126.75042 115.89747 100.63388 92.87873 83.51817

>

> # Plot the elbow method

> fviz\_nbclust(data\_scaled, kmeans, method = "wss")

图表, 折线图

描述已自动生成

I cannot see obvious elbow.

>

> # Compute average silhouette method

> silhouette\_scores <- fviz\_nbclust(data\_scaled, kmeans, method = "silhouette")

>

> print(silhouette\_scores)

> 图表, 折线图

描述已自动生成

Optimal number of clusters is 2

>

> set.seed(31)

> k2 <- kmeans(data\_scaled, centers = 2, nstart = 25)

>

> cluster\_stats <- data.frame(cluster=rep(1:2, each=5),

+ Variable=rep(colnames(data\_scaled), 2),

+ Mean=numeric(10), Max=numeric(10), Min=numeric(10))

>

> # Calculate the mean, max, and min for each variable in each cluster

> for(i in 1:2){

+ for(variable in colnames(data\_scaled)){

+ cluster\_data <- data\_scaled[k2$cluster == i, variable]

+ cluster\_stats[cluster\_stats$cluster == i & cluster\_stats$Variable == variable, "Mean"] <- mean(cluster\_data)

+ cluster\_stats[cluster\_stats$cluster == i & cluster\_stats$Variable == variable, "Max"] <- max(cluster\_data)

+ cluster\_stats[cluster\_stats$cluster == i & cluster\_stats$Variable == variable, "Min"] <- min(cluster\_data)

+ }

+ }

>

> cluster\_stats\_wide <- reshape(cluster\_stats, timevar = "Variable", idvar = "cluster", direction = "wide")

>

> print(cluster\_stats\_wide)

cluster Mean.Weight Max.Weight Min.Weight Mean.Turning\_Circle Max.Turning\_Circle Min.Turning\_Circle Mean.Displacement

1 1 0.9857914 2.4779934 -0.1075849 0.8287193 2.686332 -0.8257167 0.9880952

6 2 -0.6024281 0.9378475 -2.3571314 -0.5064396 1.089946 -2.1028253 -0.6038359

Max.Displacement Min.Displacement Mean.Horsepower Max.Horsepower Min.Horsepower Mean.Gas\_Tank\_Size Max.Gas\_Tank\_Size

1 3.17320683 -0.2865533 0.8436737 3.7115148 -0.7583223 0.8832909 3.4992224

6 0.01141647 -1.6108634 -0.5155784 0.9492559 -1.8883373 -0.5397889 0.5729262

Min.Gas\_Tank\_Size

1 -0.2724483

6 -2.2883412

**Problem 4 (10 points). Use hw8\_p4.csv. Run the pam (k-medoids) algorithm using the Gower distance 9 times with k = 2 through 10 and plot the silhouette-k graph. What is an optimal k?**

> library(cluster)

> library(factoextra)

>

> data <- read.csv("hw8\_p4.csv")

>

> # Convert character columns to factors

> data[] <- lapply(data, function(x) {

+ if(is.character(x)) factor(x) else x

+ })

>

> # Compute the Gower distance matrix

> gower\_dist <- daisy(data, metric = "gower")

>

> # Initialize a vector to store the average silhouette widths

> avg\_sil\_width <- numeric(9)

>

> # Run the PAM algorithm for k=2 to k=10 and calculate silhouette scores

> for(k in 2:10) {

+ pam\_fit <- pam(gower\_dist, diss = TRUE, k = k)

+ sil\_width <- silhouette(pam\_fit)

+ avg\_sil\_width[k - 1] <- mean(sil\_width[, 'sil\_width'])

+ }

>

> # Find the k value corresponding to the largest average silhouette width

> optimal\_k <- which.max(avg\_sil\_width) + 1

>

> # Plot the silhouette scores

> plot(2:10, avg\_sil\_width, type='b', pch=19, frame=FALSE,

+ xlab="Number of clusters k", ylab="Average Silhouette Width")

> title(main="Silhouette Method for Optimal k")

图表, 折线图

描述已自动生成

>

> # Print the optimal k value

> print(paste("The optimal number of clusters k is:", optimal\_k))

[1] "The optimal number of clusters k is: 2"

**Problem 5 (10 points). Use hw8\_p5.csv (1). Run the agnes hierarchical clustering algorithm using the Gower distance 4 times with average, single, complete, and ward methods. (2). Collect agglomerative coefficients and plot a bar graph of the coefficients. Which method has the highest coefficient? (5). Run the agnes algorithm once more with the best method you identified in the above Problem 5-(2). (6). Cut the tree with k = 3. (7). Show the row numbers for each cluster, which shows cluster membership. The format of your answer should look like: Cluster 1: rows {1, 5, 22, 30, … } Cluster 2: rows {2, 15, 17, 41, … } Cluster 3: rows {8, 52, 53, 61, … }**

> library(cluster)

>

> data <- read.csv('hw8\_p5.csv')

>

> # Convert categorical columns to factors

> categorical\_columns <- c("job", "marital", "education", "housing", "loan", "contact")

> data[categorical\_columns] <- lapply(data[categorical\_columns], as.factor)

>

> # Calculate the Gower distance

> gower\_dist <- daisy(data, metric = "gower")

>

> # Execute the clustering algorithm

> methods <- c("average", "single", "complete", "ward")

> agnes\_models <- lapply(methods, function(m) agnes(gower\_dist, method = m))

>

> # Collect and plot the agglomerative coefficients

> agglomerative\_coefficients <- sapply(agnes\_models, function(model) model$ac)

> barplot(agglomerative\_coefficients, main="Agglomerative Coefficients", names.arg = methods, ylab = "Coefficient")

图表, 条形图

描述已自动生成

Ward method have the highest coefficient.

>

> # From the chart, we see that the ward method is the best

> best\_method <- "ward"

>

> # Cluster using the best method

> best\_model <- agnes(gower\_dist, method = best\_method)

>

> # Cut the tree with k = 3.

> clusters <- cutree(as.hclust(best\_model), k = 3)

>

> # Output

> cluster\_list <- list()

> for (k in 1:max(clusters)) {

+ cluster\_rows <- which(clusters == k)

+ cluster\_list[[k]] <- cluster\_rows

+ cat(paste("Cluster", k, "rows:", toString(cluster\_rows)), "\n")

+ }

Cluster 1 rows: 1, 6, 8, 20, 23, 24, 25, 26, 32

Cluster 2 rows: 2, 4, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 22, 29, 30, 34, 35, 36, 38, 39, 40, 41, 42, 44, 45, 46, 47, 48

Cluster 3 rows: 3, 5, 19, 21, 27, 28, 31, 33, 37, 43, 49, 50