IE425 Data Mining

Homework 3

01.06.2023

Tevfik Buğra Türker 2019402120

Ömercan Mısırlıoğlu 2020402261

Initial Codes to Modify the Data

```
library(readxl)
library(data.table)
library(ggplot2)
library(cluster)
library(dplyr)

data = read_excel("EastWestAirlines.xlsx") #reading the data


data = data[c(-1)] # removing ID variable from the data
setDT(data) # converting categorical miles data into numerical data based on
given ranges and their mean values

data$cc1_miles = ifelse(data$cc1_miles==1,2500,
                        ifelse(data$cc1_miles==2,7500,
                               ifelse(data$cc1_miles==3,17500,
                                      ifelse(data$cc1_miles==4,32500,
                                             ifelse(data$cc1_miles==5,50000,0)
))))

data$cc2_miles = ifelse(data$cc2_miles==1,2500,
                        ifelse(data$cc2_miles==2,7500,
                               ifelse(data$cc2_miles==3,17500,
                                      ifelse(data$cc2_miles==4,32500,
                                             ifelse(data$cc2_miles==5,50000,0)
))))

data$cc3_miles = ifelse(data$cc3_miles==1,2500,
                        ifelse(data$cc3_miles==2,7500,
                               ifelse(data$cc3_miles==3,17500,
                                      ifelse(data$cc3_miles==4,32500,
                                             ifelse(data$cc3_miles==5,50000,0)
))))


summary(data) # checking if there is any problem

data_sc <- (scale(data)) # scaling the data
summary(data_sc) # checking if there is any problem

dist_mat <- dist(data_sc, method = 'euclidean') # distance matrix
```

PART A

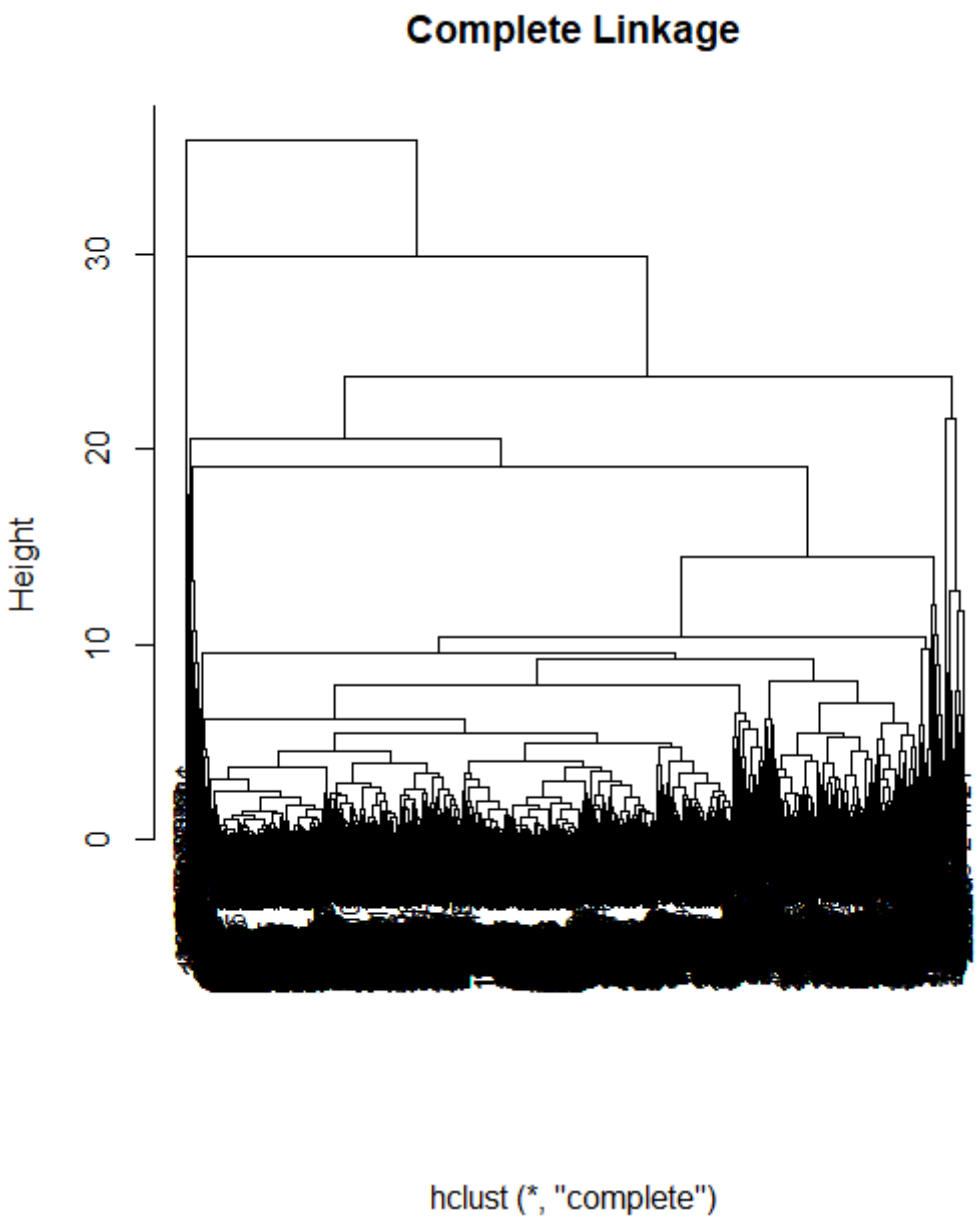
Question: Apply hierarchical clustering with Euclidean distance and complete linkage. How many clusters appear to be appropriate? Use silhoutte index.

Code:

```
hc_complete=hclust(dist_mat, method="complete") # Hierarchical Clustering with
plot(hc_complete,main="Complete Linkage", xlab="", cex=.9) #Euclidean distance
and complete linkage


# finding the best k values based on silhoutte index
silhout=c()
for (k in 2:8){
  clust=cutree(hc_complete,k=k)
  X_sil=silhouette(clust, dist_mat)
  silhout[k-1]=mean(X_sil[,3])
}
data.frame(k=2:8,silhout) # making it a dataframe

Best_k = which.max(silhout) + 1 # obtaining the best k value
cat("Appropriate # of clusters:", Best_k)


# Clusters:
best_set = cutree(hc_complete,k=Best_k)
best_set = table(best_set)
best_set
```

Output:

```
  k   silhout
1 2 0.8383266
2 3 0.7733689
3 4 0.5777837
4 5 0.5166354
5 6 0.5184992
6 7 0.5288008
7 8 0.5290171

Appropriate # of clusters: 2

best_set
   1    2
3997    2
```



Comment: According to the silhouette index, 2 clusters appear to be appropriate. At k=2 the index takes the maximum value which is 0.8383266.

PART B

Question: Compare the cluster centroids to characterize the different clusters and try to give each cluster a label.

Code:

```
hic_cluster = cutree(hc_complete,k=Best_k)
Clustcentro <- aggregate(data,list(hic_cluster),mean) #Finding the mean of all
the values

#Appending the size of clusters with the mean of all other values
dataF <-
data.frame(Cluster=Clustcentro[,1],Observations_in_this_cluster=as.vector(tabl
e(hic_cluster)),Clustcentro[,-1])

transpose_df <- t(dataF) #transposing

df_round <- function(x, digits) {
  numeric_columns    <- sapply(x, class) == 'numeric'
  x[numeric_columns] <-  round(x[numeric_columns], digits)
  x}

Clustcentros <- df_round(transpose_df, 2)
Clustcentros[2,] = as.numeric(Clustcentros[2,])
Clustcentros
```

Output:

```
                                [,1]     [,2]
Cluster                         1.00      2.0
Observations_in_this_cluster  3997.00      2.0
Balance                      73584.78 106673.0
Qual_miles                     143.84    694.0
cc1_miles                    12511.26  17500.0
cc2_miles                     2591.32   2500.0
cc3_miles                     2623.22   2500.0
Bonus_miles                  17115.23  76325.0
Bonus_trans                     11.57     75.5
Flight_miles_12mo              447.05  26458.5
Flight_trans_12                  1.35     49.0
Days_since_enroll             4119.32   2602.0
Award                            0.37      1.0
```

Comment: According to our output, customers in the 2nd cluster use airlines more frequently than the 1st cluster. Thus:

1st cluster: Irregular Customers

2nd cluster: Premium Customers

PART C

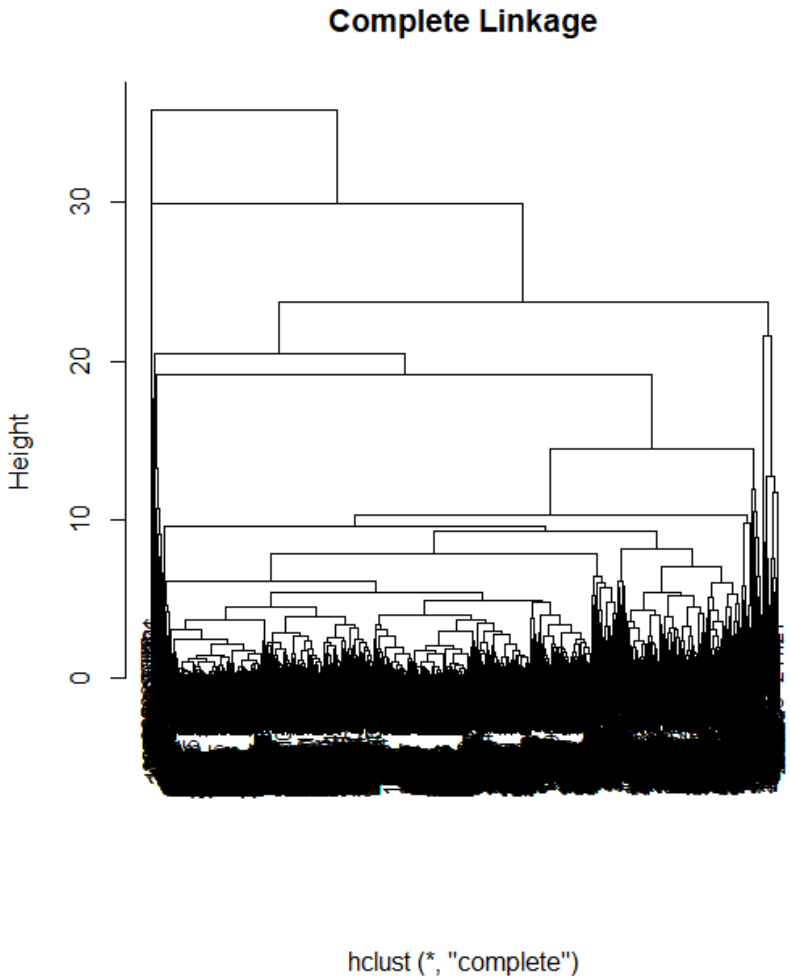
Question: To check the stability of the clusters, remove a random 5% of the data (by taking a random sample of 95% of the records, namely 200 records), and repeat the analysis. Does the same picture emerge? Use 425 as the seed.

Code:

```
# removing 5% of the data randomly
set.seed(425)
rem_indices <- sample(1:nrow(data), 200) #since 200 is the 5% of the data
data_c <- data[-rem_indices,]
summary(data_c)

# scaling the data
data_scaled_c <- (scale(data_c))
summary(data_scaled_c)

# distance matrix
dist_matrix_c <- dist(data_scaled_c, method = 'euclidean')

# Hierarchical Clustering with Euclidean distance and complete linkage for
part c
hc_complete_c=hclust(dist_matrix_c, method="complete")
plot(hc_complete,main="Complete Linkage", xlab="", cex=.8)

silhoutc=c()
for (k in 2:8){
  clustc = cutree(hc_complete_c,k=k)
  Sil = silhouette(clustc, dist_matrix_c)
  silhoutc[k-1] = mean(Sil[,3])
}
data.frame(k=2:8,silhoutc)

#finding the best k
best_k = which.max(silhoutc) + 1
cat("Suitable number of clusters for part c:", best_k)


#obtaining the table for the best clustering set
best_clust_set = cutree(hc_complete_c,k=best_k)
best_clust_set = table(best_clust_set)
best_clust_set


hc_cluster_c = cutree(hc_complete_c,k=best_k) #Finding the mean of all the
values
```

```
Clustcentro <- aggregate(data_c,list(hc_cluster_c),mean) #Appending the size
of clusters with the mean of all other values

dataf <-
data.frame(Cluster=Clustcentro[,1],Observations_in_this_cluster=as.vector(tabl
e(hc_cluster_c)),Clustcentro[,-1])
transpose_df <- t(dataf)
df_round <- function(x, digits) {
  numeric_columns    <- sapply(x, class) == 'numeric'
  x[numeric_columns] <-  round(x[numeric_columns], digits)
  x
}
Clustcentros_c <- df_round(transpose_df, 2)
Clustcentros[2,] = as.numeric(Clustcentros[2,])

#Comparison
Clustcentros_c
Clustcentros
```

Output:



```
  k  silhoutc
1 2 0.8127294
2 3 0.8046575
3 4 0.6656101
4 5 0.6711499
5 6 0.6558561
6 7 0.6250696
7 8 0.6122809

Suitable number of clusters for part c: 2

best_clust_set
   1    2
3795    4
```

Centroids of the new clusters: and the previous ones:

	[,1]	[,2]	[,1]	[,2]
Cluster	1.00	2.00	1.00	2.0
Observations_in_this_cluster	3795.00	4.00	3997.00	2.0
Balance	73612.02	131999.50	73584.78	106673.0
Qual_miles	144.23	347.00	143.84	694.0
cc1_miles	12565.88	17500.00	12511.26	17500.0
cc2_miles	2588.27	2500.00	2591.32	2500.0
cc3_miles	2617.26	2500.00	2623.22	2500.0
Bonus_miles	17103.51	65634.25	17115.23	76325.0
Bonus_trans	11.53	69.25	11.57	75.5
Flight_miles_12mo	437.92	19960.00	447.05	26458.5
Flight_trans_12	1.31	49.25	1.35	49.0
Days_since_enroll	4122.56	2200.25	4119.32	2602.0
Award	0.37	1.00	0.37	1.0

Comment: In this question, 5% of the data is randomly deleted to obtain new clusters. The results for these new clusters are given above.

When these two different clustering results are compared, we can see that the new one is almost the same as the previous one. As in previous clusters, the 1st cluster continues to represent “Irregular Customers” and the 2nd cluster continues to represent “Premium Customers”.

Thus, we can say that these clusters are reasonably stable.

PART D

Question: Use k-means algorithm with the number of clusters you found in part (a). Does the same picture emerge?

Code:

```
best_km = kmeans(x=data_sc,centers=2,nstart=20)
best_km$size # the cluster distribution


km_centroid = aggregate(data,list(best_km$cluster),mean)
vec = c(sum(best_km$cluster==1), sum(best_km$cluster==2))
hc_cluster = cutree(hc_complete,k=the_best_k_value)

#Finding the mean of all the values
Cluster_centroid <- aggregate(data,list(hc_cluster),mean)

#Appending the size of clusters with the mean of all other values
df =
data.frame(Cluster=km_centroid[,1],Observations_in_this_cluster=vec,km_centroi
d[,-1])
trans_df = t(df)
df_round <- function(x, digits) {
  numeric_columns    <- sapply(x, class) == 'numeric'
  x[numeric_columns] <-  round(x[numeric_columns], digits)
  x
}
km_centroids = df_round(trans_df, 2)
km_centroids
```

Output:

KNN

	[,1]	[,2]
Cluster	1.00	2.00
Observations_in_this_cluster	2906.00	1093.00
Balance	46727.89	145050.74
Qual_miles	85.53	299.87
cc1_miles	5548.86	31031.56
cc2_miles	2601.51	2564.04
cc3_miles	2501.72	2946.02
Bonus_miles	6423.26	45650.72
Bonus_trans	8.03	21.09
Flight_miles_12mo	213.66	1115.15
Flight_trans_12	0.64	3.33
Days_since_enroll	3776.18	5028.85
Award	0.24	0.72

Hierarchical

[,1]	[,2]
1.00	2.0
3997.00	2.0
73584.78	106673.0
143.84	694.0
12511.26	17500.0
2591.32	2500.0
2623.22	2500.0
17115.23	76325.0
11.57	75.5
447.05	26458.5
1.35	49.0
4119.32	2602.0
0.37	1.0

Comment: The clusters obtained by 2 different methods are compared in the table above. According to these data, the clusters obtained with KNN seem to be more inclusive than hierarchical clustering because the number of observations in KNN is more balanced. According to the clusters obtained by KNN, the members of the 2nd cluster have more balance, earned more miles, flied more, used their cards more and enrolled for a longer time. Thus, the label “Premium Customers” is still valid for this cluster. On the other side, the members of the 1st cluster have less balance, earned less miles, flied less, used their cards less and enrolled for a shorter time. Thus, the label “Irregular Customers” is still valid for this cluster.

Overall, KNN and Hierarchical Clustering algorithms provided similar results on the “EastWestAirlines” dataset. In both algorithms, the clusters represented premium and irregular customers. However, since KNN has more evenly distributed clusters, the clusters it provides are seemed to be more reliable.

PART E

Question: Which clusters would you target for offers, and what type of offers would you target to customers in that cluster?

Comment: I would target the cluster of "Irregular Customers" for offers and promotions. This group do not use the airline as frequently as “Premium Customers”. Therefore, with offers and promotions, there is an opportunity to attract their attention and encourage them to develop a greater loyalty to the airline. By offering targeted discounts or promotional codes for future flights, we can increase the engagement of these customers and inspire them to choose this airline more often.