BUSINESS INTELLIGENCE AND ANALYTICS

ASSIGNMENT-3

GROUP MEMBERS:

LASVITHA PREGADA

SAI ROHITH YADAV KALASANI

SMITHA KANNUR ASHOK

SNEHA VENKATAPATHY

1. Read data from a CSV file. The dataset has variables of mixed types. If required, set up the working directory using setwd()

```
setwd("C:\\Users\\sneha\\OneDrive\\Documents\\Business intelligence and analytics\\ASSIGNMENT-3")
df=read.csv("Registered_Business_Locations_-_San_Francisco_20231014 (1).csv")
```

- We setup the directory and read the file into the df.

2. Remove the row(s) with missing values using and na.omit()

```
sum(is.na(df))
df1=na.omit(df)
sum(is.na(df1))
str(df1)
summary(df1)
dim(df1)
names(df1)
```

- We used na.omit to remove the missing values from the df and after removing we created df1.

OUTPUT:

```
> sum(is.na(df))
[1] 349388
> df1=na.omit(df)
> sum(is.na(df1))
[1] 0
```

3. Show the imported data's brief description and summary statistics using str() and summary().

```
sum(is.na(df))
df1=na.omit(df)
sum(is.na(df1))
str(df1)
summary(df1)
dim(df1)
names(df1)
```

We used str(df1), summary(df1) and got the below information.

OUTPUT:

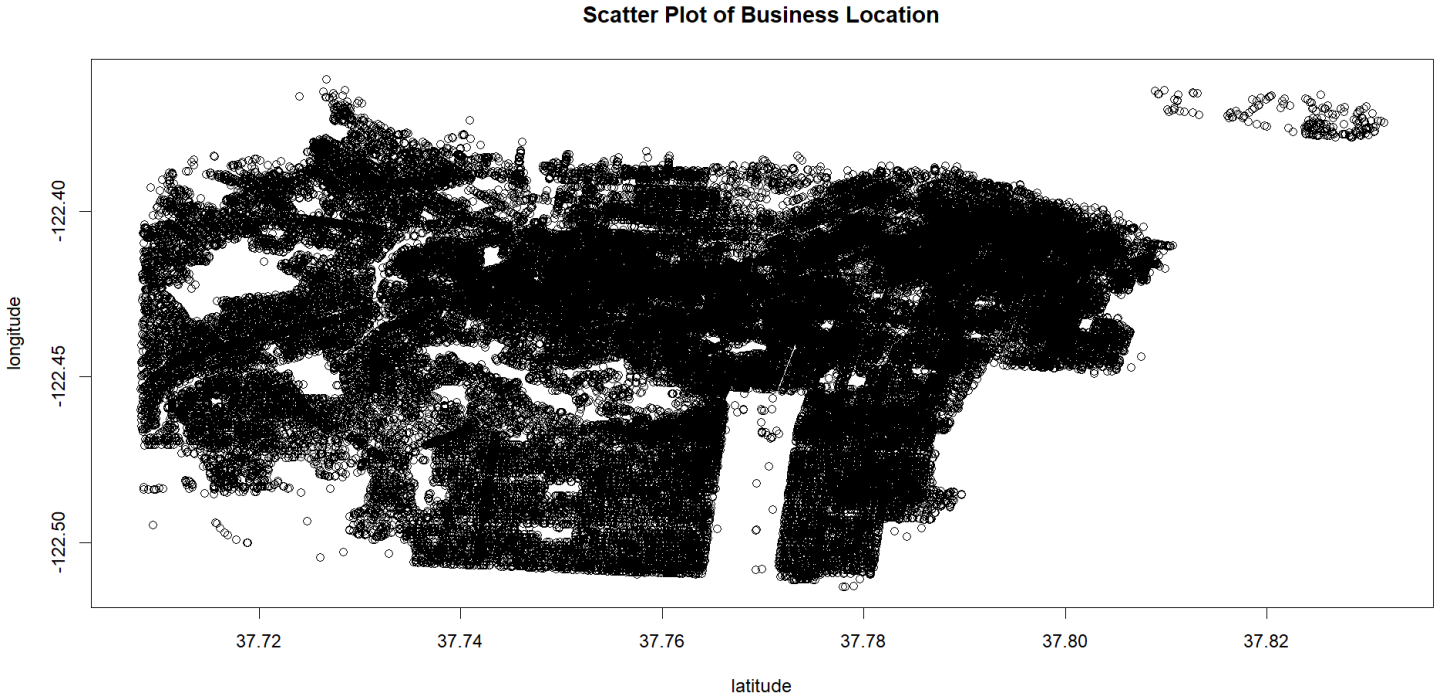
```
> sum(is.na(df))
[1] 349388
> df1=na.omit(df)
> sum(is.na(df1))
[1] 0
> str(df1)
'data.frame':   255899 obs. of  32 variables:
 $ Location.Id                          : chr  "0441029-02-001" "0335342-01-001" "0002881-01-001" "0441029-01-001" ...
 $ Business.Account.Number              : int  441029 335342 2881 441029 1088599 11913 1089379 401832 1089379 1089405 ...
 $ Ownership.Name                       : chr  "Stiger Lynnell L" "Hinh Michael" "Benson & Neff-Cpas-A Prof Corp" "Stiger Lynnell L" ...
 $ DBA.Name                             : chr  "L. Steiger" "Executive Mercedes Sedan Srvc" "Benson & Neff-Cpas-A Prof Corp" "Bad Intentions" ...
 $ Street.Address                       : chr  "328 Haight St" "1759 Greenwich St" "1 Post St 2150" "328 Haight St" ...
 $ City                                 : chr  "San Francisco" "San Francisco" "San Francisco" "San Francisco" ...
 $ State                                : chr  "CA" "CA" "CA" "CA" ...
 $ Source.Zipcode                       : chr  "94102-6157" "94123-3613" "94104" "94102-6157" ...
 $ Business.Start.Date                  : chr  "07/24/2009" "03/25/2000" "10/01/1968" "07/24/2009" ...
 $ Business.End.Date                    : chr  "11/11/2014" "01/01/2016" "08/31/2018" "11/11/2014" ...
 $ Location.Start.Date                  : chr  "06/28/2013" "03/25/2000" "10/01/1996" "07/24/2009" ...
 $ Location.End.Date                    : chr  "11/11/2014" "01/01/2016" "08/31/2018" "11/11/2014" ...
 $ Mail.Address                         : chr  "" "" "" "" ...
 $ Mail.City                            : chr  "" "" "" "" ...
 $ Mail.Zipcode                         : chr  "" "" "" "" ...
 $ Mail.State                           : chr  "" "" "" "" ...
 $ NAICS.Code                           : chr  "" "" "" "" ...
 $ NAICS.Code.Description               : chr  "" "" "" "" ...
 $ Parking.Tax                          : chr  "false" "false" "false" "false" ...
 $ Transient.Occupancy.Tax              : chr  "false" "false" "false" "false" ...
 $ LIC.Code                             : chr  "" "" "" "" ...
 $ LIC.Code.Description                 : chr  "" "" "" "" ...
 $ Supervisor.District                  : int  5 2 3 5 3 2 3 2 3 3 ...
 $ Neighborhoods...Analysis.Boundaries: chr  "Hayes Valley" "Marina" "Financial District/South Beach" "Hayes Valley" ...
 $ Business.Corridor                    : chr  "" "" "" "" ...
 $ Business.Location                    : chr  "POINT (-122.42768 37.772488)" "POINT (-122.4286 37.79985)" "POINT (-122.40211 37.789062)" "POINT (-122.42768 37.772488)" ...
 $ UniqueID                             : chr  "0441029-02-001-0441029--06-28-2013" "0335342-01-001-0335342--03-25-2000" "0002881-01-001-0002881--10-01-1996" "0441029-01-001-0441029--07-2
4-2009" ...
 $ SF.Find.Neighborhoods                : int  26 15 19 26 77 10 77 15 77 77 ...
 $ Current.Police.Districts             : int  4 4 6 4 6 8 6 4 6 6 ...
 $ Current.Supervisor.Districts         : int  11 6 3 11 3 6 3 6 3 3 ...
 $ Analysis.Neighborhoods               : int  9 13 8 9 8 31 8 13 8 8 ...
 $ Neighborhoods                        : int  26 15 19 26 77 10 77 15 77 77 ...
 - attr(*, "na.action")= 'omit' Named int [1:59309] 1 2 3 16 19 20 26 28 37 38 ...
  ..- attr(*, "names")= chr [1:59309] "1" "2" "3" "16" ...
```

```
> summary(df1)
 Location.Id        Business.Account.Number Ownership.Name       DBA.Name         Street.Address         City
 Length:255899      Min.   :     28         Length:255899      Length:255899      Length:255899      Length:255899
 Class :character   1st Qu.: 414977         Class :character   Class :character   Class :character   Class :character
 Mode  :character   Median : 491022         Mode  :character   Mode  :character   Mode  :character   Mode  :character
                    Mean   : 718719
                    3rd Qu.:1077149
                    Max.   :1151597
    State           Source.Zipcode     Business.Start.Date Business.End.Date  Location.Start.Date Location.End.Date
 Length:255899      Length:255899      Length:255899       Length:255899      Length:255899       Length:255899
 Class :character   Class :character   Class :character    Class :character   Class :character    Class :character
 Mode  :character   Mode  :character   Mode  :character    Mode  :character   Mode  :character    Mode  :character


 Mail.Address        Mail.City         Mail.Zipcode        Mail.State         NAICS.Code         NAICS.Code.Description
 Length:255899      Length:255899      Length:255899      Length:255899      Length:255899      Length:255899
 Class :character   Class :character   Class :character   Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character


 Parking.Tax        Transient.Occupancy.Tax   LIC.Code         LIC.Code.Description Supervisor.District
 Length:255899      Length:255899           Length:255899      Length:255899        Min.   : 1.000
 Class :character   Class :character        Class :character   Class :character     1st Qu.: 3.000
 Mode  :character   Mode  :character        Mode  :character   Mode  :character     Median : 6.000
                                                                                    Mean   : 5.582
                                                                                    3rd Qu.: 8.000
                                                                                    Max.   :11.000
 Neighborhoods...Analysis.Boundaries Business.Corridor  Business.Location    UniqueID         SF.Find.Neighborhoods
 Length:255899                       Length:255899      Length:255899      Length:255899      Min.   :  1.0
 Class :character                    Class :character   Class :character   Class :character   1st Qu.: 31.0
 Mode  :character                    Mode  :character   Mode  :character   Mode  :character   Median : 53.0
                                                                                             Mean   : 57.7
                                                                                             3rd Qu.: 95.0
                                                                                             Max.   :117.0
 Current.Police.Districts Current.Supervisor.Districts Analysis.Neighborhoods Neighborhoods
 Min.   : 1.000           Min.   : 1.000               Min.   : 1.00          Min.   :  1.0
 1st Qu.: 3.000           1st Qu.: 3.000               1st Qu.: 8.00          1st Qu.: 31.0
 Median : 6.000           Median : 6.000               Median :20.00          Median : 53.0
 Mean   : 5.258           Mean   : 6.253               Mean   :18.84          Mean   : 57.7
 3rd Qu.: 8.000           3rd Qu.:10.000               3rd Qu.:30.00          3rd Qu.: 95.0
 Max.   :10.000           Max.   :11.000               Max.   :41.00          Max.   :117.0
> dim(df1)
[1] 255899     32
```

4. Perform data pre-processing and explore the data through visualizations of a scatter plot of the 'Business.Location' variable. Interpret the graph regarding the potential structure in which data points can be grouped together. You may use the pairs( ) function as well.

```
library(dplyr)
df1 <- df1 %>%
  mutate(
    latitude = as.numeric(sub(".*\\(([^ ]+) ([^)]+)\\)", "\\2", Business.Location)),
    longitude = as.numeric(sub(".*\\(([^ ]+) ([^)]+)\\)", "\\1", Business.Location))
  )
names(df1)


df2=subset(df1, select = -c(NAICS.Code,Ownership.Name,DBA.Name,Street.Address,City,State,Source.Zipcode,Business.Start.Date,
                            Business.End.Date,Location.Start.Date,Location.End.Date,Mail.Address,Mail.City,Mail.Zipcode,Mail.State,NAICS.Code.Description,LIC.Code.Description,
                            Current.Supervisor.Districts,LIC.Code,Neighborhoods...Analysis.Boundaries,Business.Corridor,Analysis.Neighborhoods))
names(df2)
df2 <- replace(df2, df2=='', NA)
sum(is.na(df2))
df3=na.omit(df2)
sum(is.na(df3))

df3=replace(df3,df3=="true",1)
df3=replace(df3,df3=="false",0)

summary(df3)
names(df3)
any(is.na(df3$Business.Location))
class(df3$Business.Location)
class(df3$latitude)

# Select relevant columns
#selected_data = df3[, c("Business.Location", "Location.ID", "Business.Account.Number", "Parking.Tax","Transient.Occupancy.Tax","Supervisor.District","Business.Location","UniqueI
plot(df3$latitude,df3$longitude,
     main = "Scatter Plot of Business Location",
     xlab = "latitude",
     ylab = "longitude"
)
str(df3)
```

OUTPUT:



The scatter plot shows the distribution of business locations in San Francisco, with dense clusters indicating commercial areas and gaps corresponding to less commercialized zones. These patterns suggest that businesses are concentrated in specific neighborhoods or districts. Outliers may indicate businesses in remote areas or data errors. This distribution is useful for identifying natural groupings of businesses for market analysis or urban planning. Before clustering, you should check for and remove data errors and consider the geographic scale of coordinates. Spatial clustering techniques may be appropriate due to the geographic nature of the data.

6. Choose a set of appropriate variables for processing data and provide interpretation regarding your selection.

"Location.Id", "Business.Account.Number", "Parking.Tax", "Transient.Occupancy.Tax", "Supervisor.District", "Business.Location" "UniqueID", "SF.Find.Neighborhoods", "Current.Police.Districts", "Neighborhoods", "latitude", "longitude" these variables are selected. The latitude and longitude are obtained from the business location category. Remaining city, state, address variables are dropped as the location is giving precise values.

Location.Id, Business.Account.Number, UniqueId are the identifiers and the rest of the columns are used for clustering as these are the ones which contributed effectively to the clustering.

7. Use random sampling to select a subset of rows from your dataset to avoid freezing your computer or running into memory issues.

```
df3$Location.Id <- as.factor(df3$Location.Id)
df3$Parking.Tax <- as.factor(df3$Parking.Tax)
df3$Transient.Occupancy.Tax <- as.factor(df3$Transient.Occupancy.Tax)
df3$Business.Location <- as.factor(df3$Business.Location)
df3$UniqueID <- as.factor(df3$UniqueID)

# Load the dplyr package for data manipulation
library(dplyr)
# Set the fraction of rows you want to sample
sample_fraction <- 0.01  # For example, sample 20% of the data

# Use the sample function to randomly select a subset of rows
sampled_df <- df3 %>%
  sample_n(size = floor(n() * sample_fraction))

# Print the sampled data frame
print(sampled_df)

df4=subset(sampled_df, select = -c(Location.Id,Business.Account.Number,Business.Location,UniqueID))
names(df4)

df4$Parking.Tax=as.numeric(df4$Parking.Tax)
df4$Transient.Occupancy.Tax=as.numeric(df4$Transient.Occupancy.Tax)
```

OUTPUT:

```
> # Print the sampled data frame
> print(sampled_df)
      Location.Id Business.Account.Number Parking.Tax Transient.Occupancy.Tax Supervisor.District
1  0462353-01-001                  462353           0                       0                   4
2  1216716-03-191                 1099885           0                       0                   8
3  1214544-02-191                 1098966           0                       0                   3
4  1269361-02-211                 1121298           0                       0                   8
5  1207406-12-181                 1095848           0                       0                   3
6  0444896-02-001                  444896           0                       0                   3
7  0314873-01-001                  314873           0                       0                   2
8  0472988-01-001                  472988           0                       0                  10
9  1299786-03-221                 1133011           0                       0                  11
10 1130296-12-161                 1062537           0                       0                   2
11 1137563-02-171                 1066110           0                       0                   3
12 1264415-11-201                 1119225           0                       1                   9
13 1308171-06-221                 1136319           0                       0                   9
14 0459998-01-001                  459998           0                       0                   4
15 1200119-09-181                 1092493           0                       0                  10
16 1255426-07-201                 1115634           0                       0                   5
17 1190257-05-181                 1088476           0                       0                   3
18 1055713-02-161                 1027143           0                       0                   2
19 0484513-01-001                  484513           0                       0                   6
20 0927982-01-001                  927982           0                       0                  10
21 0394274-02-001                  394274           0                       0                   3
22 1247950-03-201                 1084141           0                       0                   3
23 0434678-01-001                  434678           0                       0                   6
24 1299662-03-221                 1132963           0                       0                  11
25 0474544-01-001                  474544           0                       0                   6
26 1279861-06-211                 1125254           0                       0                   5
27 1124751-10-161                  303652           0                       0                   6
28 1024147-03-151                  194331           0                       0                   3
29 0490096-01-001                  490096           0                       0                   5
30 1205019-11-181                 1094755           0                       0                  10
31 0321317-01-001                  321317           0                       0                   3
32 1223864-05-191                 1102654           0                       0                  11
33 0477619-01-001                  477619           0                       0                   3
34 1182701-03-181                 1085258           0                       0                   9
35 0388913-03-001                  388913           0                       0                   6
36 0035158-03-001                   35158           0                       0                   3
37 1153819-07-171                 1073081           0                       0                   6
38 1017645-01-151                 1008491           0                       1                   8
39 1266242-12-201                 1119969           0                       0                  10
40 1077196-06-161                 1020639           0                       0                   3
```

8. To ensure the integrity of your clustering analysis, it's imperative to normalize the selected data set, which comprises a diverse mix of integer, character, and numeric variables. Additionally, pay special attention to the handling of dummy variables to mitigate potential sources of bias.

Input:

```
# Normalize numerical variables
num_cols <- sapply(df4, is.numeric)
df4[num_cols] <- scale(df4[num_cols]) # This centers and scales the numerical data

# Convert factors to dummy variables for categorical variables
# This is assuming that categorical variables have been converted to factors
df4 <- data.frame(model.matrix(~ . - 1, data = df4))
```

OUTPUT:

```
# Normalize numerical variables
num_cols <- sapply(df4, is.numeric)
df4[num_cols] <- scale(df4[num_cols]) # This centers and scales the numerical data

# Convert factors to dummy variables for categorical variables
# This is assuming that categorical variables have been converted to factors
df4 <- data.frame(model.matrix(~ . - 1, data = df4))
```

9. Calculate the distance matrix by the dist() function and pay close attention to the method you choose since you have a mixed data set.

```
install.packages("FD")
library(FD)
install.packages("cluster")
library(cluster)
install.packages("cluster")
library(cluster)
# Calculate the Gower distance matrix
gower_dist <- daisy(df4, metric = "gower")
```

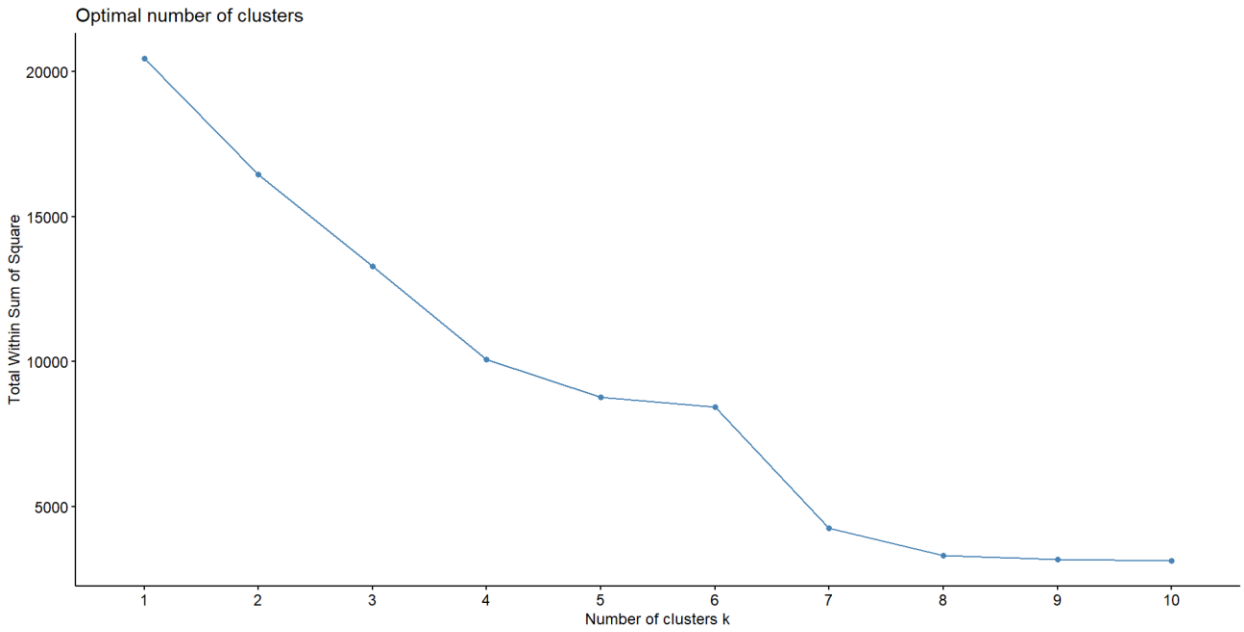
OUTPUT:

```
> gower_dist <- daisy(df4, metric = "gower")
```

Values

gower_dist	Large dissimilarity (327…
- attr(*, "Labels")= chr [1:2558]…	
- attr(*, "Size")= int 2558	
- attr(*, "Metric")= chr "mixed"	
- attr(*, "Types")= chr [1:8] "I"…	

10. calculate how many clusters by interpreting the elbow plot. You may install the 'factoextra' function



OUTPUT:

The elbow plot is commonly used to select the optimal number of clusters for K-means clustering. It shows the total within-cluster sum of squares (WSS) for different numbers of clusters (k). The "elbow" of the plot typically indicates the number of clusters after which adding more clusters does not lead to a significant decrease in the total WSS. This is considered a good balance between the number of clusters and the within-cluster variance.

Looking at plot, the WSS rapidly decreases as the number of clusters increases from 1 to 4. After k=4, the rate of decrease slows down significantly, suggesting that additional clusters do not add as much value. The elbow seems to be around k=4, which is where the plot starts to flatten out. Therefore, based on this elbow plot, the optimal number of clusters to use for your K-means clustering analysis appears to be 4.

11. Run Kmeans clustering with the optimal number of clusters obtained from the previous section

```
# Create an elbow plot to determine the optimal number of clusters
fviz_nbclust(df4, kmeans, method = "wss")
final_kmeans <- kmeans(df4, centers = 4)
```

OUTPUT:

```
final_kmeans          List of  9
  $ cluster     : Named int [1:2558] 2 2 2 1 3 3 3 1 2 2 ...
   ..- attr(*, "names")= chr [1:2558] "1" "2" "3" "4" ...
  $ centers     : num [1:4, 1:8] -0.03066 -0.0212 0.05423 0.00146 0.01889 ...
   ..- attr(*, "dimnames")=List of 2
   .. ..$ : chr [1:4] "1" "2" "3" "4"
   .. ..$ : chr [1:8] "Parking.Tax" "Transient.Occupancy.Tax" "Supervisor.Distri..
  $ totss       : num 20456
  $ withinss    : num [1:4] 1695 3760 1965 2689
  $ tot.withinss: num 10109
  $ betweenss   : num 10347
  $ size        : int [1:4] 436 1030 637 455
  $ iter        : int 3
  $ ifault      : int 0
  - attr(*, "class")= chr "kmeans"
```

12. Visualize the clustering algorithm result and interpret the graph.

OUTPUT:

The cluster plot shows four distinct groups, indicating the dataset has been segmented into clusters with similar characteristics. The axes suggest that the plot is based on principal component analysis, with the first two components explaining just over half of the data's variance. Some overlap between clusters is visible, particularly between clusters 1 and 3, which may require further analysis. Overall, the plot suggests a meaningful division of the data into four categories.

13. Run hierarchical clustering and create a dendrogram

```
# Create a cluster plot
fviz_cluster(final_kmeans, data = df4, geom = "point")
hierarchical_clusters <- hclust(gower_dist, method = "ward.D2")
dendrogram <- as.dendrogram(hierarchical_clusters)
plot(dendrogram, main = "Dendrogram", xlab = "Data Points", ylab = "Height")
```

OUTPUT:

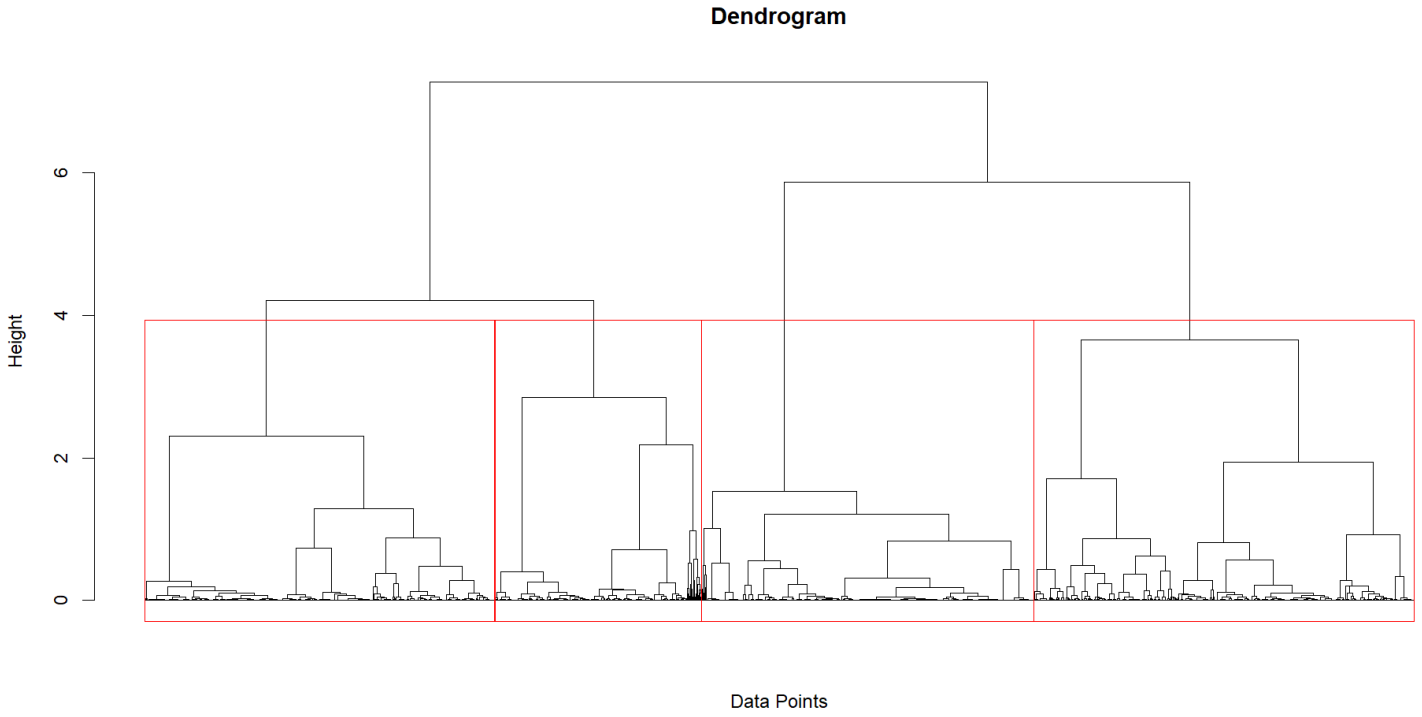
hierarchical_clusters	list [7] (S3: hclust)	List of length 7
merge	integer [2557 x 2]	-2 -1876 -2088 -5 -1116 -12 -1408 1 2 -575 4 -959 ...
height	double [2557]	0 0 0 0 0 0 ...
order	integer [2558]	2030 538 1868 2281 1048 1255 ...
labels	character [2558]	'1' '2' '3' '4' '5' '6' ...
method	character [1]	'ward.D2'
call	language	hclust(d = gower_dist, method = "ward.D2")
dist.method	NULL	Pairlist of length 0

dendrogram	list [2] (S3: dendrogram)	List of length 2
[[1]]	list [2] (S3: dendrogram)	List of length 2
[[2]]	list [2] (S3: dendrogram)	List of length 2

14. Plot the dendrogram and highlight the optimal number of clusters in the graph

```
# Create a cluster plot
fviz_cluster(final_kmeans, data = df4, geom = "point")
hierarchical_clusters <- hclust(gower_dist, method = "ward.D2")
dendrogram <- as.dendrogram(hierarchical_clusters)
plot(dendrogram, main = "Dendrogram", xlab = "Data Points", ylab = "Height")
```

OUTPUT:


Dendrogram
Height
0
2
4
6
Data Points