Homework 3. Clustering practice

Nisarg

2022-10-10

Part 1. USArrests Dataset and Hierarchical Clustering (20 Points)

Consider the “USArrests” data. It is a built-in dataset you may directly get in RStudio. Perform hierarchical clustering on the observations (states) and answer the following questions.

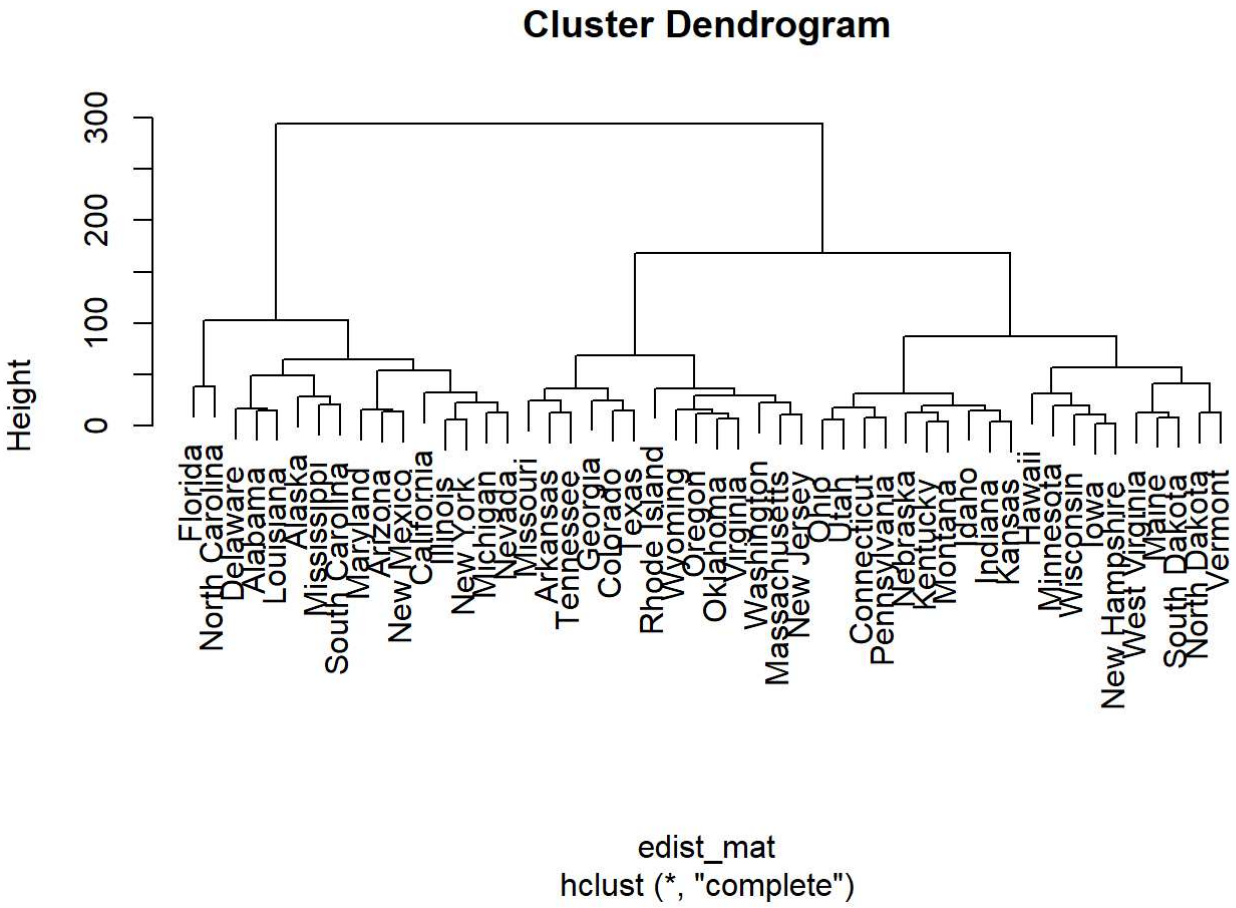
```
head(USArrests)
```

	Murder <dbl>	Assault <int>	UrbanPop <int>	Rape <dbl>
Alabama	13.2	236	58	21.2
Alaska	10.0	263	48	44.5
Arizona	8.1	294	80	31.0
Arkansas	8.8	190	50	19.5
California	9.0	276	91	40.6
Colorado	7.9	204	78	38.7

6 rows

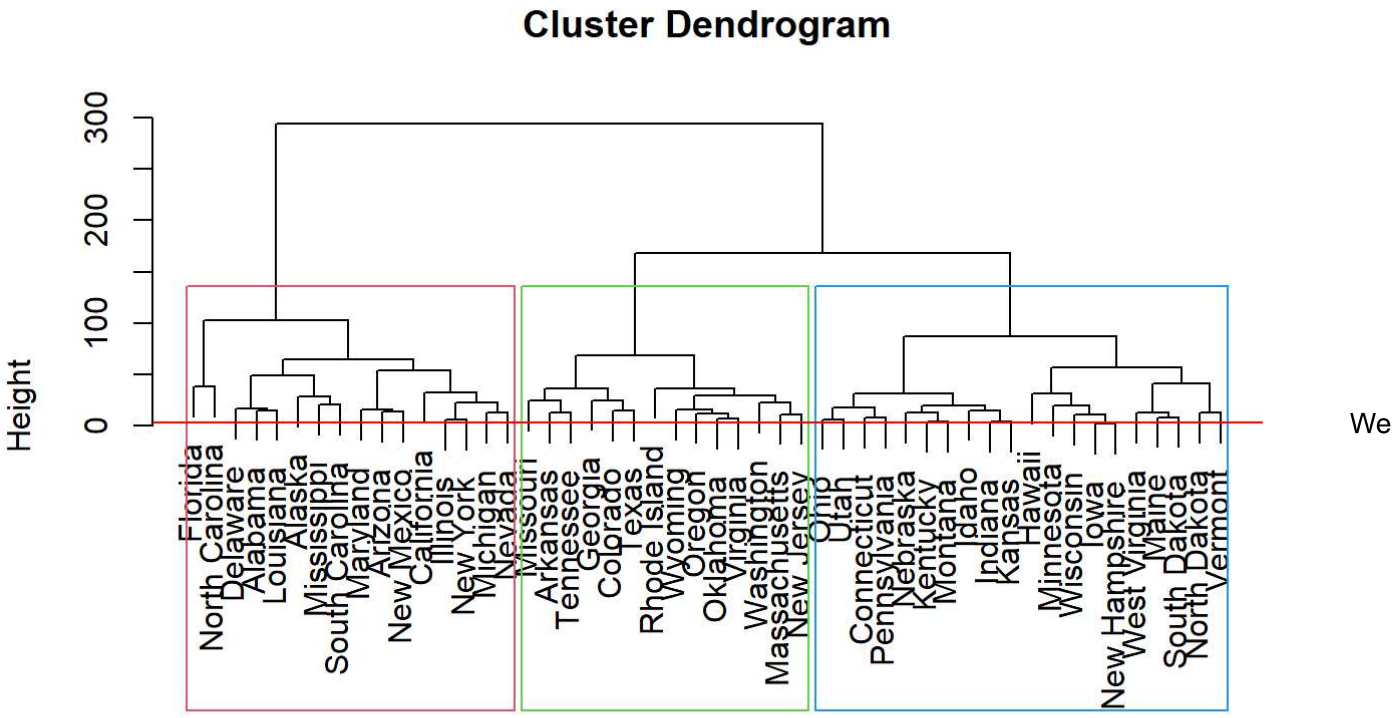
Q1.1. Using hierarchical clustering with complete linkage and Euclidean distance, cluster the states. (5 points)

```
set.seed(786)
us_df_sc <- as.data.frame(USArrests)
edist_mat <- dist(us_df_sc, method = 'euclidean')
hclust_com <- hclust(edist_mat, method = 'complete')
plot(hclust_com)
```



Q1.2. Cut the dendrogram at a height that results in three distinct clusters. Interpret the clusters. Which states belong to which clusters? (5 points)

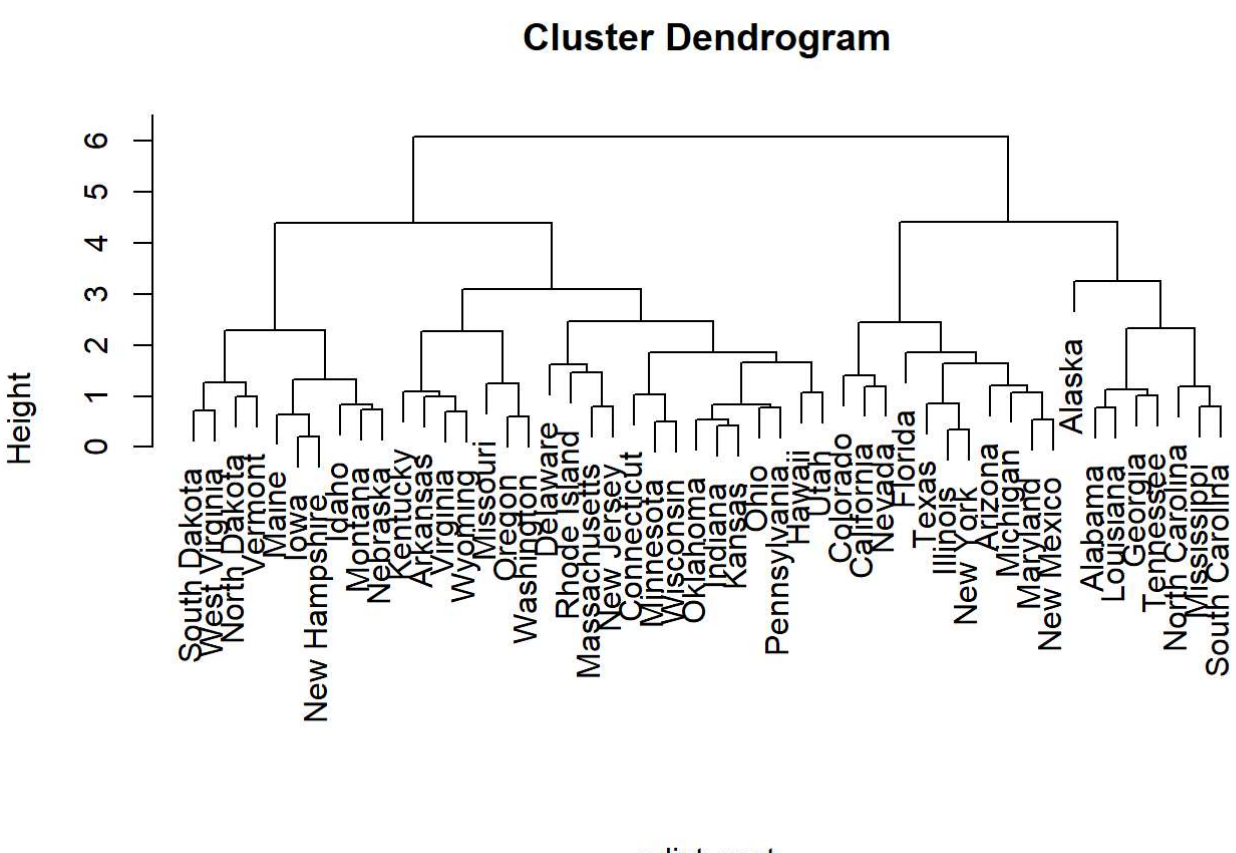
```
plot(hclust_com)
rect.hclust(hclust_com , k = 3, border = 2:6)
abline(h = 3, col = 'red')
```



were able to distinguish the data into three clusters as in the above 3 boxes, red, green and blue. States in: red cluster:Florida, North Carolina, Delaware, Alabama, Louisiana, Alaska, Mississippi, South Carolina, M aryland, Arizona, New Mexica, California, Illinois, New York, Michigan and Nevada. green cluster: Missouri, Arkansas, Tennessee, Georgia, Colorado, T exas, Rhode Island, Wyoming, Oregon, Oklahoma, Virginia, Washington, Massachusetts and New Je rsey blue cluster: Ohio, Utah, Connetcticut, Pennsyl vania, Nebraska, Kentucky, Montana, Idaho, Indiana, Kansas, Hawaii, Minnesota, Wiconsin, iow a, New Hampshire, West Virginia, Maine, South Dakota, North Dakota and Vermont

Q1.3 Hierarchically cluster the states using complete linkage and Euclidean distance, after scaling the variables to have standard deviation one. (5 points)

```
us_df_sc <- as.data.frame(scale(USArrests))
edist_mat <- dist(us_df_sc, method = 'euclidean')
hclust_com <- hclust(edist_mat, method = 'complete')
plot(hclust_com)
```




```
summary(USArrests)
```

```
##      Murder          Assault         UrbanPop          Rape
##  Min.   : 0.800   Min.   : 45.0   Min.   :32.00   Min.   : 7.30
##  1st Qu.: 4.075   1st Qu.:109.0   1st Qu.:54.50   1st Qu.:15.07
##  Median : 7.250   Median :159.0   Median :66.00   Median :20.10
##  Mean   : 7.788   Mean   :170.8   Mean   :65.54   Mean   :21.23
##  3rd Qu.:11.250   3rd Qu.:249.0   3rd Qu.:77.75   3rd Qu.:26.18
##  Max.   :17.400   Max.   :337.0   Max.   :91.00   Max.   :46.00
```

```
summary(scale(USArrests))
```

```
##      Murder           Assault           UrbanPop            Rape
##  Min.   :-1.6044   Min.   :-1.5090   Min.   :-2.31714   Min.   :-1.4874
##  1st Qu.:-0.8525   1st Qu.:-0.7411   1st Qu.:-0.76271   1st Qu.:-0.6574
##  Median :-0.1235   Median :-0.1411   Median : 0.03178   Median :-0.1209
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.: 0.7949   3rd Qu.: 0.9388   3rd Qu.: 0.84354   3rd Qu.: 0.5277
##  Max.   : 2.2069   Max.   : 1.9948   Max.   : 1.75892   Max.   : 2.6444
```

Q1.4 What effect does scaling the variables have on the hierarchical clustering obtained? In your opinion, should the variables be scaled before the inter-observation dissimilarities are computed? Provide a justification for your answer. *(5 points)*

Answer:… Scaling affected the cluster such that the states in clusters after scaling are very different from the states present before scaling.We can see for example in the summary the minimum and maximum of murder and assult drastically changed after scaling. We can see that Assult dominates over murder in our data and scaling the values bring the features to a more comparable range.

Part 2. Market Segmentation (80 Points)

An advertisement division of large club store needs to perform customer analysis the store customers in order to create a segmentation for more targeted marketing campaign

You task is to identify similar customers and characterize them (at least some of them). In other word perform clustering and identify customers segmentation.

This data-set is derived from https://www.kaggle.com/imakash3011/customer-personality-analysis (https://www.kaggle.com/imakash3011/customer-personality-analysis)

```
Colomns description:
People
  ID: Customer's unique identifier
  Year_Birth: Customer's birth year
  Education: Customer's education level
  Marital_Status: Customer's marital status
  Income: Customer's yearly household income
  Kidhome: Number of children in customer's household
  Teenhome: Number of teenagers in customer's household
  Dt_Customer: Date of customer's enrollment with the company
  Recency: Number of days since customer's last purchase
  Complain: 1 if the customer complained in the last 2 years, 0 otherwise

Products

  MntWines: Amount spent on wine in last 2 years
  MntFruits: Amount spent on fruits in last 2 years
  MntMeatProducts: Amount spent on meat in last 2 years
  MntFishProducts: Amount spent on fish in last 2 years
  MntSweetProducts: Amount spent on sweets in last 2 years
  MntGoldProds: Amount spent on gold in last 2 years

Place
  NumWebPurchases: Number of purchases made through the company’s website
  NumStorePurchases: Number of purchases made directly in stores
```

Assume that data was current on 2014-07-01

Q2.1. Read Dataset and Data Conversion to Proper Data Format *(12 points)*

Read “m_marketing_campaign.csv” using `data.table::fread` command, examine the data.

```
# fread m_marketing_campaign.csv and save it as df
df = fread("m_marketing_campaign.csv",stringsAsFactors = FALSE)

print(c("na values:",sum(is.na(df))))
```

```
## [1] "na values:" "0"
```

```
print(c("null values:",sum(is.null(df))))
```

```
## [1] "null values:" "0"
```

```
head(df)
```

ID <int>	Year_Birth <int>	Education <chr>	Marital_Status <chr>	Inco… <int>	Kidh… <int>	Teenh… <int>	Dt_Customer <chr>	Rece… <int>	M
5524	1957	Bachelor	Single	58138	0	0	04-09-2012	58	
2174	1954	Bachelor	Single	46344	1	1	08-03-2014	38	
4141	1965	Bachelor	Together	71613	0	0	21-08-2013	26	
6182	1984	Bachelor	Together	26646	1	0	10-02-2014	26	
5324	1981	PhD	Married	58293	1	0	19-01-2014	94	
7446	1967	Master	Together	62513	0	1	09-09-2013	16	

6 rows | 1-10 of 18 columns

```
summary(df)
```

```
##        ID          Year_Birth    Education         Marital_Status
##  Min.   :    0   Min.   :1893   Length:2209        Length:2209
##  1st Qu.: 2826   1st Qu.:1959   Class :character   Class :character
##  Median : 5462   Median :1970   Mode  :character   Mode  :character
##  Mean   : 5592   Mean   :1969
##  3rd Qu.: 8427   3rd Qu.:1977
##  Max.   :11191   Max.   :1996
##      Income          Kidhome          Teenhome      Dt_Customer
##  Min.   :  1730   Min.   :0.0000   Min.   :0.0000   Length:2209
##  1st Qu.: 35246   1st Qu.:0.0000   1st Qu.:0.0000   Class :character
##  Median : 51390   Median :0.0000   Median :0.0000   Mode  :character
##  Mean   : 52244   Mean   :0.4418   Mean   :0.5052
##  3rd Qu.: 68627   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :666666   Max.   :2.0000   Max.   :2.0000
##     Recency         MntWines        MntFruits      MntMeatProducts
##  Min.   : 0.00   Min.   :   0.0   Min.   :  0.00   Min.   :   0.0
##  1st Qu.:24.00   1st Qu.:  24.0   1st Qu.:  2.00   1st Qu.:  16.0
##  Median :49.00   Median : 174.0   Median :  8.00   Median :  68.0
##  Mean   :49.08   Mean   : 305.2   Mean   : 26.35   Mean   : 167.2
##  3rd Qu.:74.00   3rd Qu.: 505.0   3rd Qu.: 33.00   3rd Qu.: 233.0
##  Max.   :99.00   Max.   :1493.0   Max.   :199.00   Max.   :1725.0
##  MntFishProducts  MntSweetProducts  MntGoldProds    NumWebPurchases
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   : 0.000
##  1st Qu.:  3.00   1st Qu.:  1.00   1st Qu.:  9.00   1st Qu.: 2.000
##  Median : 12.00   Median :  8.00   Median : 24.00   Median : 4.000
##  Mean   : 37.56   Mean   : 27.07   Mean   : 43.85   Mean   : 4.082
##  3rd Qu.: 50.00   3rd Qu.: 33.00   3rd Qu.: 56.00   3rd Qu.: 6.000
##  Max.   :259.00   Max.   :262.00   Max.   :321.00   Max.   :27.000
##  NumStorePurchases    Complain
##  Min.   : 0.000    Min.   :0.000000
##  1st Qu.: 3.000    1st Qu.:0.000000
##  Median : 5.000    Median :0.000000
##  Mean   : 5.803    Mean   :0.009507
##  3rd Qu.: 8.000    3rd Qu.:0.000000
##  Max.   :13.000    Max.   :1.000000
```

```
# Convert Year_Birth to Age (assume that current date is 2014-07-01)

df$Age = as.integer(2014 - df$Year_Birth)
present_date = as.Date("2014-07-01",format="%Y-%m-%d")


# Dt_Customer is a date (it is still character), convert it to membership days (name it Membersh
ipDays)
# hint: note European date format, use as.Date with proper format argument

df$Dt_Customer = as.Date(df$Dt_Customer,format="%d-%m-%Y")
df$MembershipDays = as.numeric(difftime(present_date, df$Dt_Customer, "day"))

head(df)
```

ID <int>	Year_Birth <int>	Education <chr>	Marital_Status <chr>	Inco… <int>	Kidh… <int>	Teenh… <int>	Dt_Customer <date>	Rece… <int>	M
5524	1957	Bachelor	Single	58138	0	0	2012-09-04	58	
2174	1954	Bachelor	Single	46344	1	1	2014-03-08	38	
4141	1965	Bachelor	Together	71613	0	0	2013-08-21	26	
6182	1984	Bachelor	Together	26646	1	0	2014-02-10	26	
5324	1981	PhD	Married	58293	1	0	2014-01-19	94	
7446	1967	Master	Together	62513	0	1	2013-09-09	16	

6 rows | 1-10 of 20 columns

```
# Summarize Education column (use table function)
# Lets treat Education column as ordinal categories and use simple levels for distance calculati
ons
# Assuming following order of degrees:
#    HighSchool, Associate, Bachelor, Master, PhD
# factorize Education column (hint: use factor function with above levels)
df$Education <-factor(df$Education, order = TRUE, levels = c( "HighSchool", "Associate", "Bachel
or", "Master", "PhD"))

summary(df$Education)
```

```
## HighSchool  Associate   Bachelor     Master        PhD
##         54        200       1114        363        478
```

```
# Summarize Education column (use table function)
table(df$Education)
```

```
##
## HighSchool  Associate   Bachelor     Master        PhD
##         54        200       1114        363        478
```

```
# Lets convert single Marital_Status categories for 5 separate binary categories
# Divorced, Married, Single, Together and Widow, the value will be 1 if customer
# is in that category and 0 if customer is not
# hint: use dummyVars from caret package, model.matrix or simple comparison (there are only 5 gr
oups)

#dmy <- dummyVars("~ Marital_Status", data = df)
#x <- data.frame(predict(dmy, newdata = df))
#df1<-cbind(data,x)

#print(df1)
df <-df %>%
  mutate(Divorced = ifelse(Marital_Status=='Divorced', 1, 0),
         Married = ifelse(Marital_Status=='Married', 1, 0),
         Single = ifelse(Marital_Status=='Single', 1, 0),
         Together = ifelse(Marital_Status=='Together', 1, 0),
         Widow = ifelse(Marital_Status=='Widow', 1, 0))
df
```

ID <int>	Year_Birth <int>	Education <ord>	Marital_Status <chr>	Inco… <int>	Kidh… <int>	Teenh… <int>	Dt_Customer <date>	Rece… <int>
5524	1957	Bachelor	Single	58138	0	0	2012-09-04	58
2174	1954	Bachelor	Single	46344	1	1	2014-03-08	38
4141	1965	Bachelor	Together	71613	0	0	2013-08-21	26
6182	1984	Bachelor	Together	26646	1	0	2014-02-10	26
5324	1981	PhD	Married	58293	1	0	2014-01-19	94
7446	1967	Master	Together	62513	0	1	2013-09-09	16
965	1971	Bachelor	Divorced	55635	0	1	2012-11-13	34
6177	1985	PhD	Married	33454	1	0	2013-05-08	32
4855	1974	PhD	Together	30351	1	0	2013-06-06	19
5899	1950	PhD	Together	5648	1	1	2014-03-13	68

```
# lets remove columns which we will no longer use:
# remove ID, Year_Birth, Dt_Customer, Marital_Status
# and save it as df_sel
df = select(df, -1, -2,-4, -8)


# Convert Education to integers
# hint: use as.integer function, if you use factor function earlier
# properly then HighSchool will be 1, Associate will be 2 and so on)
df$Education = as.integer(df$Education)
df
```

Education <int>	Inco… <int>	Kidh… <int>	Teenh… <int>	Rece… <int>	MntWi… <int>	MntFruits <int>	MntMeatProducts <int>	MntFishP
3	58138	0	0	58	635	88	546	
3	46344	1	1	38	11	1	6	
3	71613	0	0	26	426	49	127	
3	26646	1	0	26	11	4	20	
5	58293	1	0	94	173	43	118	
4	62513	0	1	16	520	42	98	
3	55635	0	1	34	235	65	164	
5	33454	1	0	32	76	10	56	
5	30351	1	0	19	14	0	24	
5	5648	1	1	68	28	0	6	

1-10 of 2,209 rows | 1-9 of 21 columns Previous **1** 2 3 4 5 6 … 221 Next

```
# lets scale
# run scale function on df_sel and save it as df_scale
# that will be our scaled values which we will use for analysis
df_scale <- as.data.frame(scale(df))
df_scale
```

Education <dbl>	Income <dbl>	Kidhome <dbl>	Teenhome <dbl>	Recency <dbl>	MntWines <dbl>	MntFruits <dbl>
-0.4555844	0.233903916	-0.8227362	-0.9281454	0.308273215	0.9766565922	1.54886593
-0.4555844	-0.234140265	1.0393789	0.9090170	-0.382616586	-0.8711997030	-0.63705576
-0.4555844	0.768658481	-0.8227362	-0.9281454	-0.797150466	0.3577431856	0.56896999
-0.4555844	-1.015854211	1.0393789	-0.9281454	-0.797150466	-0.8711997030	-0.56167915

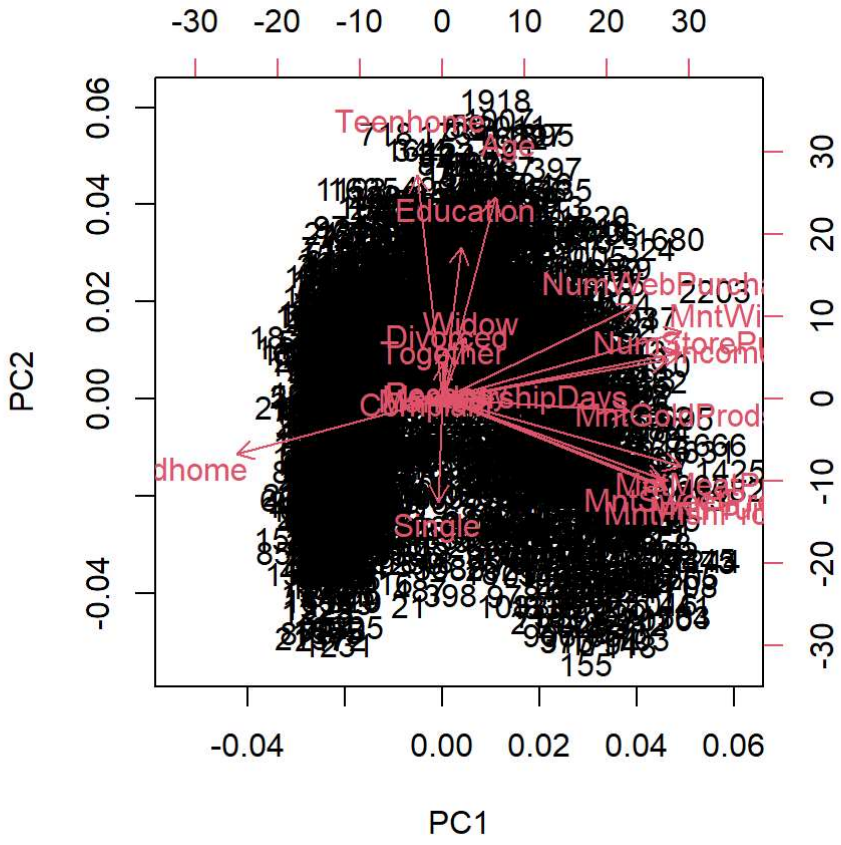
Education <dbl>	Income <dbl>	Kidhome <dbl>	Teenhome <dbl>	Recency <dbl>	MntWines <dbl>	MntFruits <dbl>
1.5352880	0.240055082	1.0393789	-0.9281454	1.551874857	-0.3914677802	0.418216778
0.5398518	0.407525528	-0.8227362	0.9090170	-1.142595367	0.6361061531	0.393091241
-0.4555844	0.134572511	-0.8227362	0.9090170	-0.520794546	-0.2078666740	0.970978588
1.5352880	-0.745679140	1.0393789	-0.9281454	-0.589883526	-0.6787146723	-0.410925936
1.5352880	-0.868821509	1.0393789	-0.9281454	-1.038961897	-0.8623157785	-0.662181304
1.5352880	-1.849158579	1.0393789	0.9090170	0.653718116	-0.8208574642	-0.662181304

1-10 of 2,209 rows | 1-7 of 21 columns Previous **1** 2 3 4 5 6 ... 221 Next

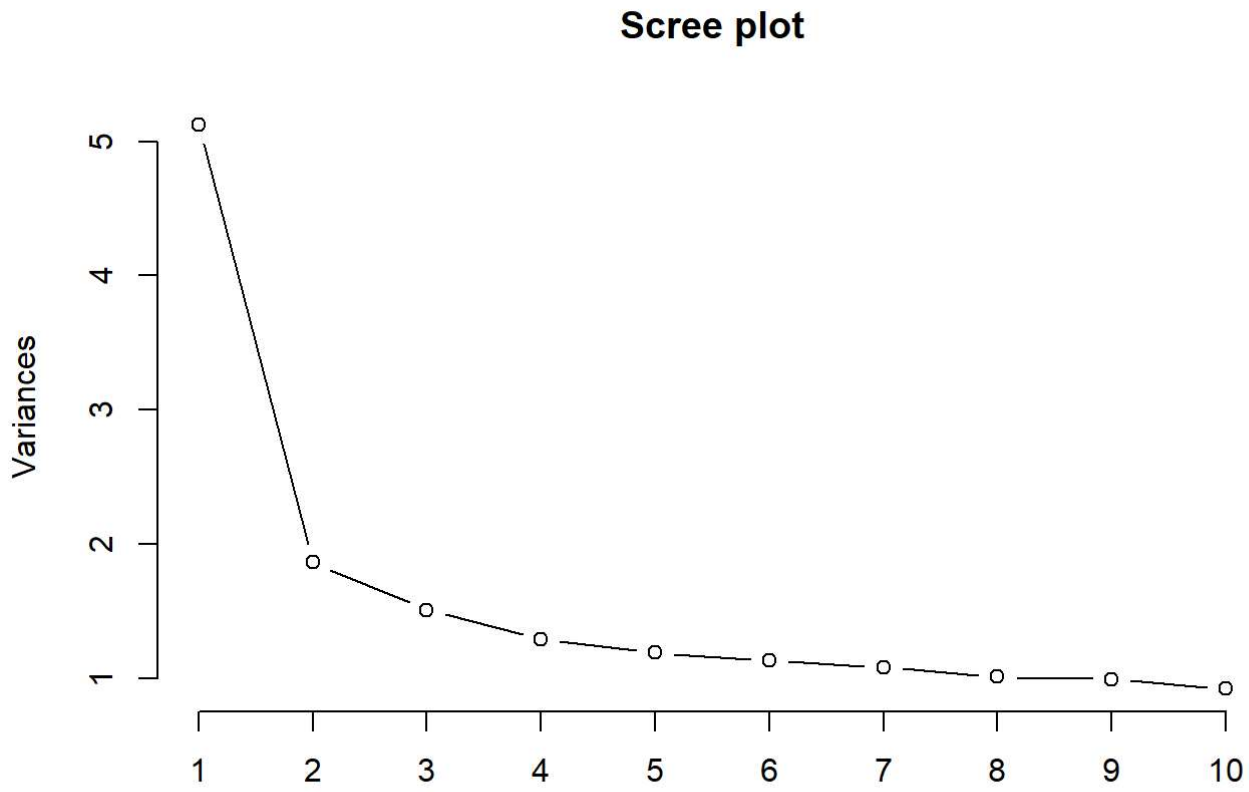
PCA

Q2.2. Run PCA *(12 points)*

```
# Run PCA on df_scale, make biplot and scree plot/percentage variance explained plot
# save as pc_out, we will use pc_out$x[,1] and pc_out$x[,2] later for plotting
pc_out <- prcomp(df_scale)
biplot(pc_out)#(pc_out) %>% ggplotly()
```



```
screeplot(pc_out, type = "line", main = "Scree plot")
```



Q2.3 Comment on observation (any visible distinct clusters?) *(4 points)*

1st PC has highest variance more than two times larger than second, however it is still around 25%. Second and higher PC has PVE less than 10%. Slow grow in cumulative PVE shows that large number of features is needed.

May be two clusters?

Cumulative PVE of first and second PCA is only 30%. This tells us that we need more iterations of PCA to be able to explain more variance.From the biplot we can see that we can create many more clusters to divide the data.

Cluster with K-Means

In questions Q2.4 to Q2.9 use K-Means method for clustering

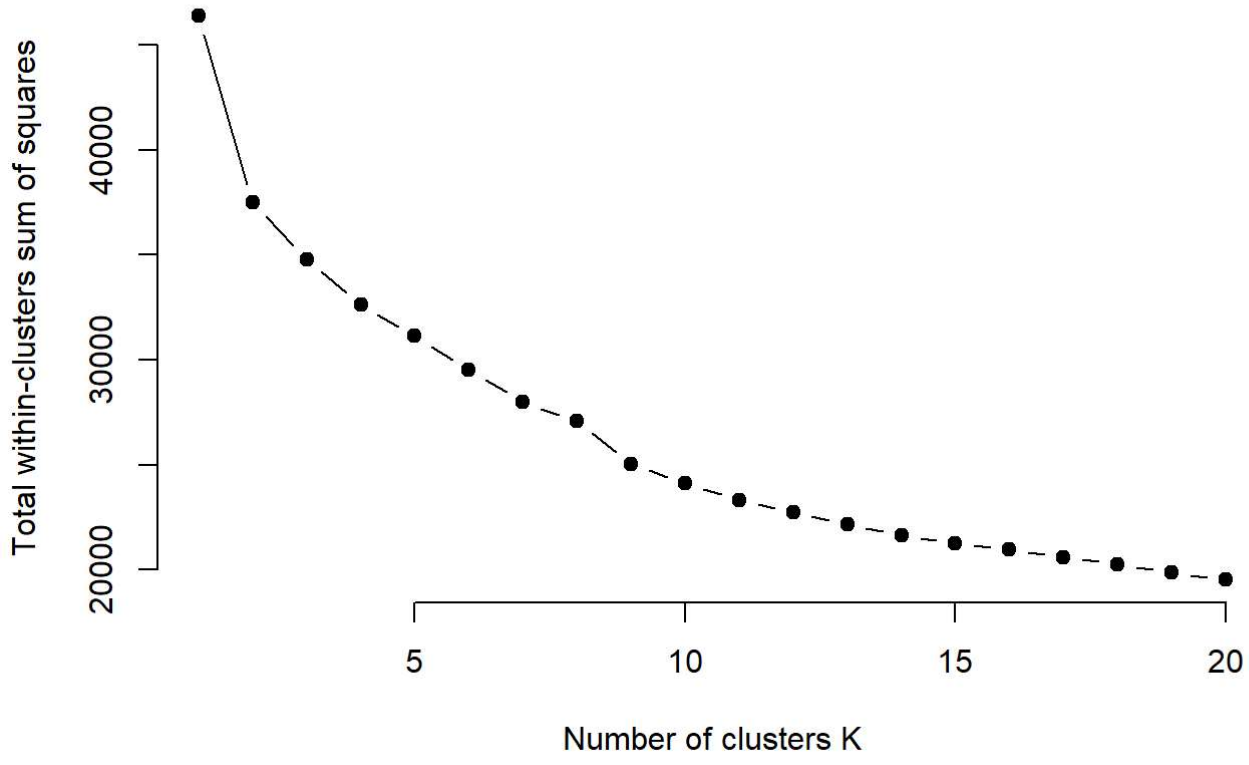
Selecting Number of Clusters

Q2.4 Select optimal number of clusters using elbow method. *(6 points)*

```
#compute from k=2 to k=20
k.max <- 20
data <- df_scale
wss <- sapply(1:k.max,
              function(k){kmeans(data, k, nstart=50,iter.max = 20 )$tot.withinss})
wss
```

```
##  [1] 46368.00 37479.33 34778.20 32638.74 31123.27 29518.03 27991.88 27065.93
##  [9] 25035.86 24122.99 23322.89 22729.60 22166.47 21640.18 21266.80 20953.10
## [17] 20580.06 20274.21 19864.48 19528.35
```

```
plot(1:k.max, wss,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")
```



Q2.5

Select optimal number of clusters using Gap Statistic.*(6 points)*

```
gap_kmeans <- clusGap(df_scale, kmeans, nstart = 20, K.max = 10, B = 100)
```

```
## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations
```

```
## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations
```

```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 110450)

## Warning: Quick-TRANSfer stage steps exceeded maximum (= 110450)
```

```
## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations
```

```
## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations
```

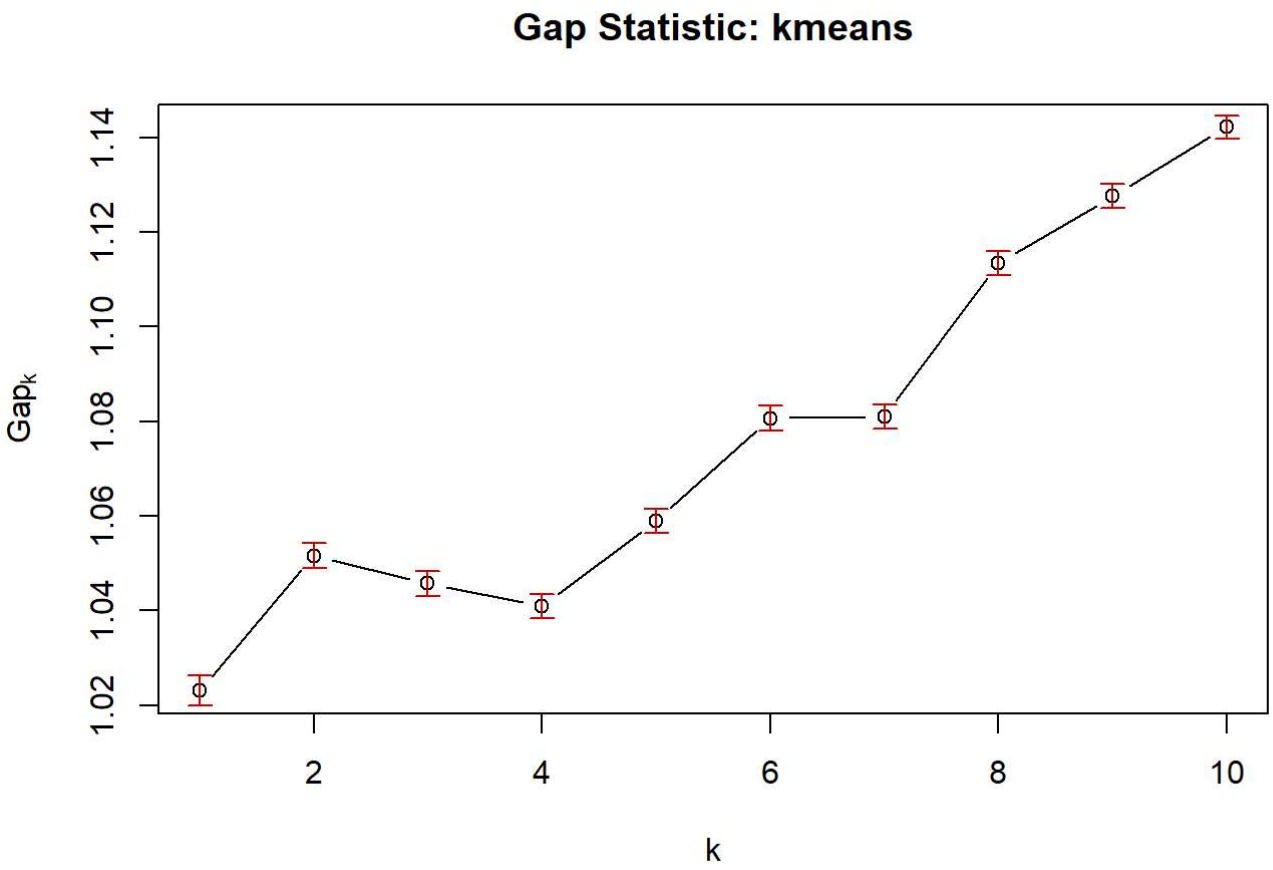
```
## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations

## Warning: did not converge in 10 iterations
```

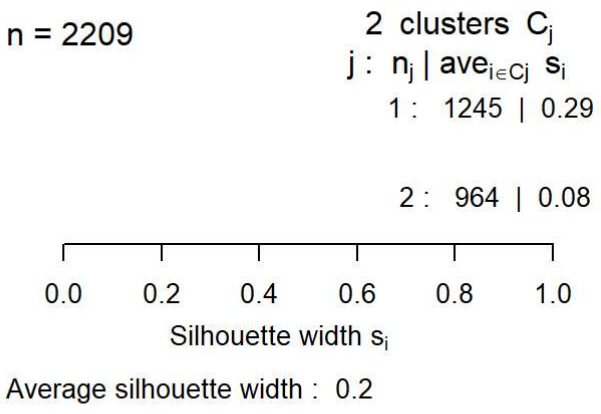
```
plot(gap_kmeans, main = "Gap Statistic: kmeans")
```

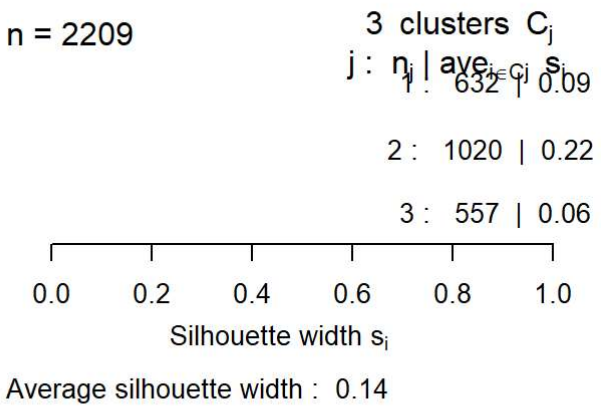


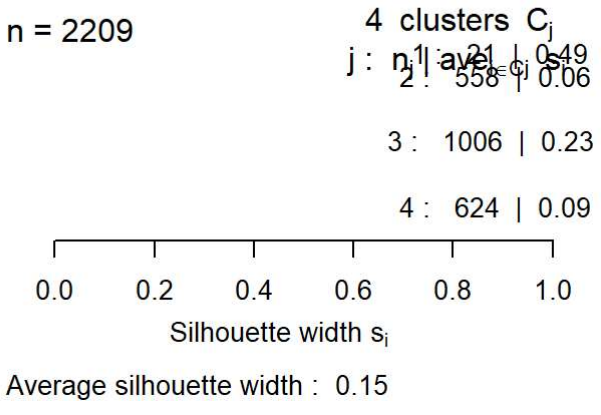
Q2.6

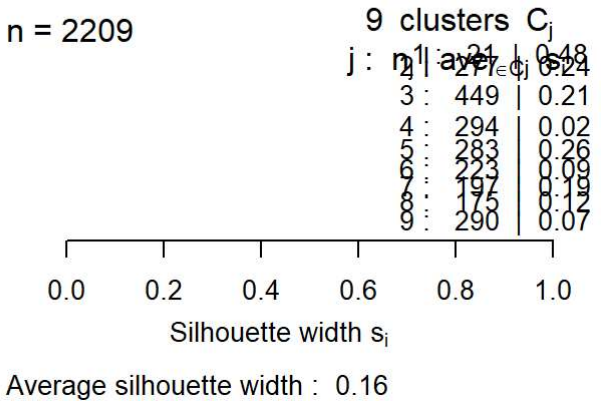
Select optimal number of clusters using Silhouette method.*(6 points)*

```
par(mar = c(5, 2, 4, 2), mfrow=c(2,2))
for(k in c(2,3,4,9)) {
  kmeans_cluster <- kmeans(df_scale, k, nstart=20)
  si <- silhouette(kmeans_cluster$cluster, dist = dist(df_scale))
  plot(si,main="")
}
```









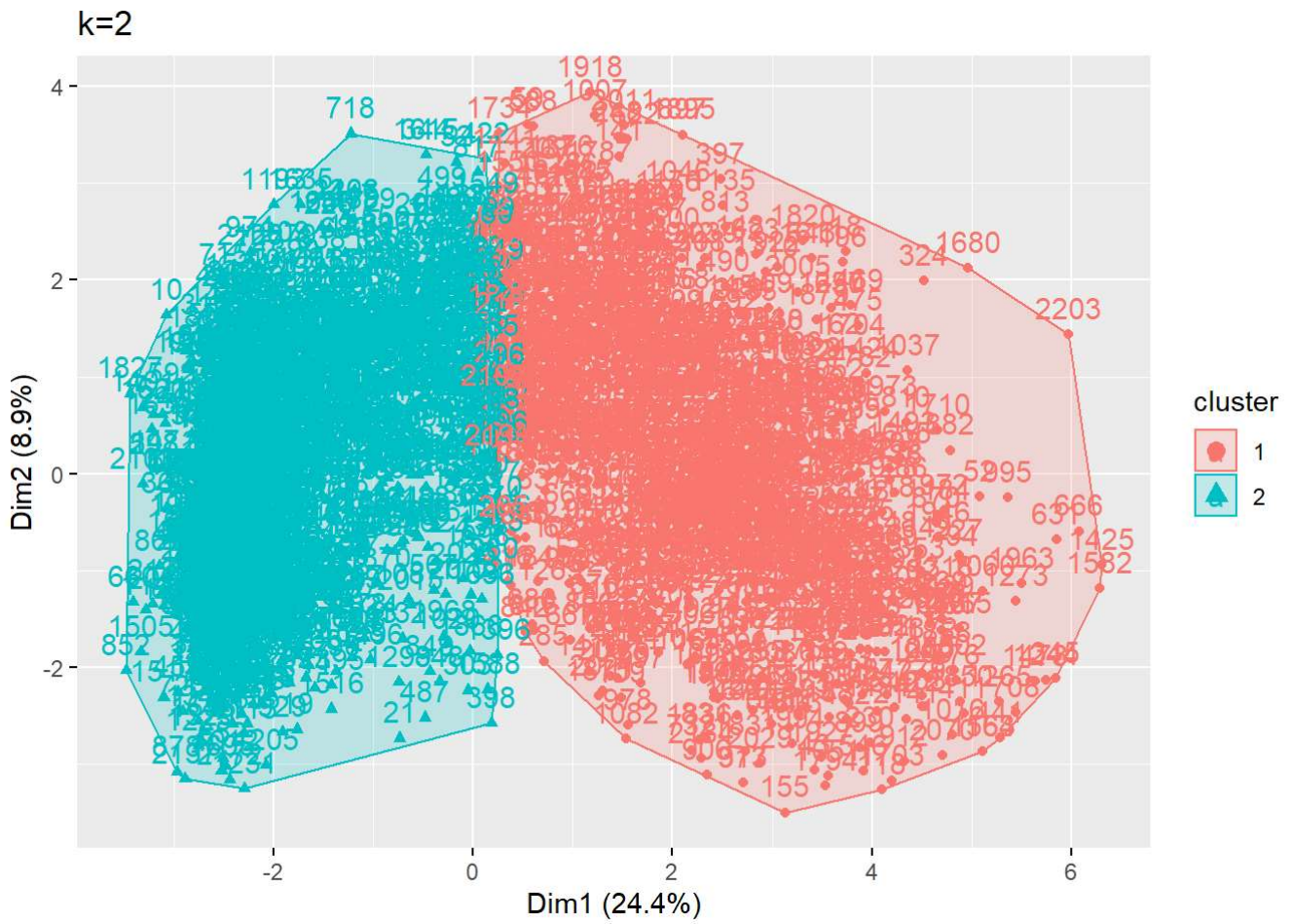

```
par(mar = c(1, 1, 1, 1), mfrow=c(1,1))
```

Q2.7 Which k will you choose based on elbow, gap statistics and silhuettes as well as clustering task (market segmentation for advertisement purposes)?*(4 points)* In elbow plot k =2 is optimal In Gap statistics k = 2 is optimal In Silhoutte plot k = 2 is optimal Hence, the optimal number for the number of clusters is k=2.

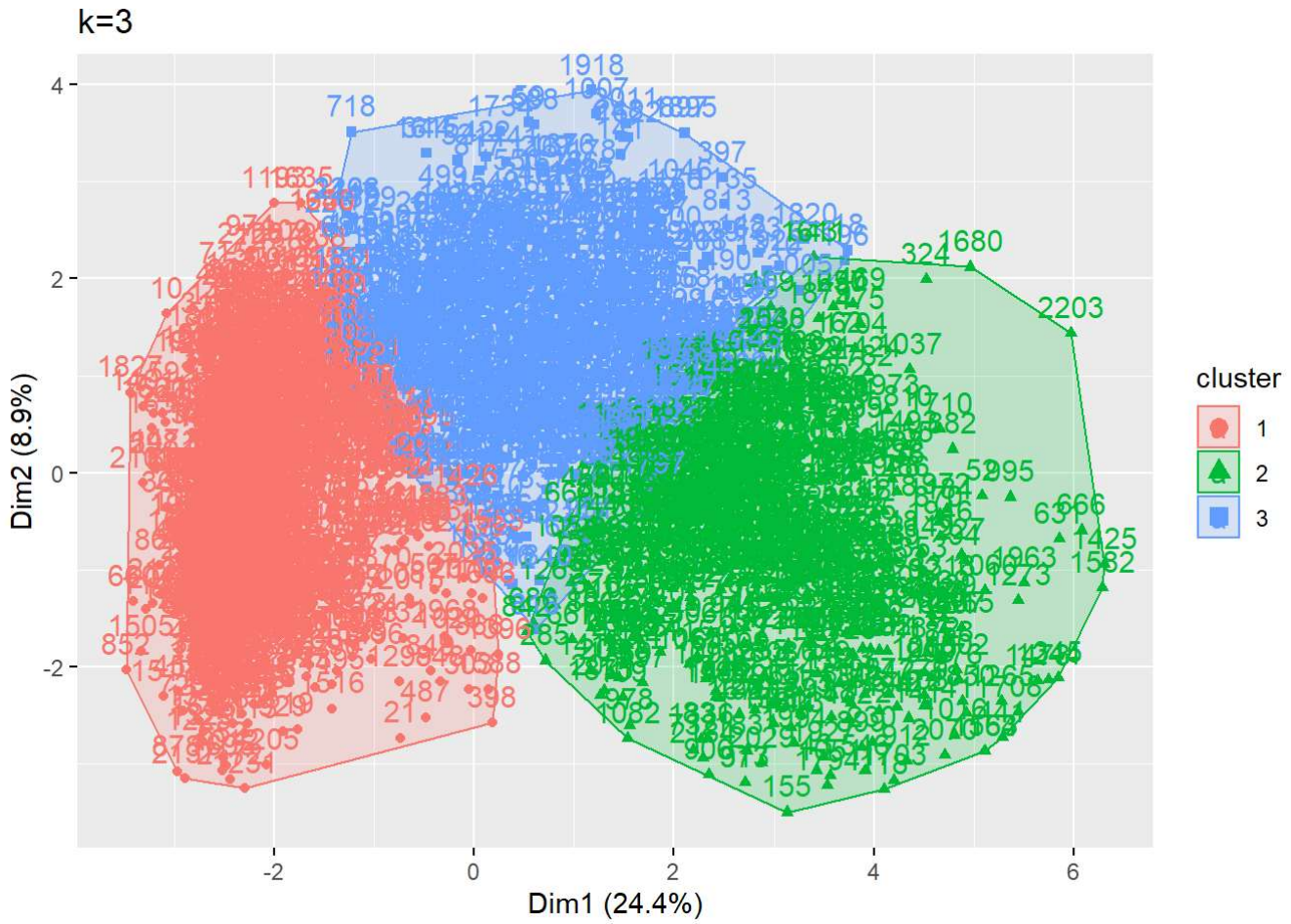
Clusters Visulalization

Q2.8 Make k-Means clusters with selected k_kmeans (store result as km_out). Plot your k_kmeans clusters on biplot (just PC1 vs PC2) by coloring points by their cluster id.*(4 points)*

```
km_2 <- kmeans(df_scale, 2, nstart = 25)
fviz_cluster(km_2, data = df_scale) + ggtitle("k=2")
```




```
km_3 <- kmeans(df_scale, 3, nstart = 25)
fviz_cluster(km_3, data = df_scale) + ggtitle("k=3")
```



Q2.9 Do you see any grouping? Comment on you observation.*(4 points)* Yes, we are able to group the data, it is more distinct when there are only 2 clusters but grouping is valid for 3 clusters as well.

Answer…

Characterizing Cluster

Q2.10 Perform descriptive statistics analysis on obtained cluster. Based on that does one or more group have a distinct characteristics? *(10 points)* Hint: add cluster column to original df dataframe

```
df_scale$Cluster <- km_2$cluster
head(df_scale)
```

	Education <dbl>	Income <dbl>	Kidhome <dbl>	Teenhome <dbl>	Recency <dbl>	MntWines <dbl>	MntFruits <dbl>	MntMea
1	-0.4555844	0.2339039	-0.8227362	-0.9281454	0.3082732	0.9766566	1.5488659	
2	-0.4555844	-0.2341403	1.0393789	0.9090170	-0.3826166	-0.8711997	-0.6370558	
3	-0.4555844	0.7686585	-0.8227362	-0.9281454	-0.7971505	0.3577432	0.5689700	
4	-0.4555844	-1.0158542	1.0393789	-0.9281454	-0.7971505	-0.8711997	-0.5616792	
5	1.5352880	0.2400551	1.0393789	-0.9281454	1.5518749	-0.3914678	0.4182168	

	Education <dbl>	Income <dbl>	Kidhome <dbl>	Teenhome <dbl>	Recency <dbl>	MntWines <dbl>	MntFruits <dbl>	MntMea
6	0.5398518	0.4075255	-0.8227362	0.9090170	-1.1425954	0.6361062	0.3930912	

6 rows | 1-9 of 23 columns

```
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:scales':
##
##     alpha, rescale
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
describeBy(df_scale, group="Cluster")
```

```
##
##  Descriptive statistics by group
## Cluster: 1
##                   vars   n  mean   sd median trimmed  mad   min   max range
## Education            1 964  0.09 0.95  -0.46    0.09 0.00 -2.45  1.54  3.98
## Income               2 964  0.74 0.93   0.72    0.72 0.48 -1.90 24.38 26.28
## Kidhome              3 964 -0.67 0.52  -0.82   -0.82 0.00 -0.82  2.90  3.72
## Teenhome             4 964 -0.06 1.00  -0.93   -0.13 0.00 -0.93  2.75  3.67
## Recency              5 964  0.02 1.00   0.07    0.02 1.28 -1.70  1.72  3.42
## MntWines             6 964  0.85 0.93   0.71    0.79 0.94 -0.90  3.52  4.42
## MntFruits            7 964  0.66 1.20   0.22    0.50 0.93 -0.66  4.34  5.00
## MntMeatProducts      8 964  0.77 1.08   0.46    0.64 0.96 -0.73  6.94  7.67
## MntFishProducts      9 964  0.68 1.18   0.37    0.54 1.11 -0.69  4.06  4.74
## MntSweetProducts    10 964  0.67 1.20   0.28    0.51 0.99 -0.66  5.71  6.37
## MntGoldProds        11 964  0.59 1.16   0.20    0.44 0.92 -0.85  5.37  6.21
## NumWebPurchases     12 964  0.65 0.96   0.70    0.59 1.08 -1.49  8.37  9.86
## NumStorePurchases   13 964  0.83 0.86   0.98    0.84 0.91 -1.78  2.21  3.99
## Complain            14 964 -0.02 0.88  -0.10   -0.10 0.00 -0.10 10.21 10.30
## Age                 15 964  0.17 1.03   0.15    0.19 1.11 -2.19  5.83  8.01
## MembershipDays      16 964  0.14 1.00   0.24    0.16 1.26 -1.75  1.70  3.45
## Divorced            17 964  0.03 1.03  -0.34   -0.29 0.00 -0.34  2.92  3.26
## Married             18 964 -0.03 0.99  -0.80   -0.09 0.00 -0.80  1.26  2.05
## Single              19 964 -0.01 0.99  -0.52   -0.19 0.00 -0.52  1.92  2.44
## Together            20 964  0.00 1.00  -0.59   -0.14 0.00 -0.59  1.69  2.28
## Widow               21 964  0.07 1.17  -0.19   -0.19 0.00 -0.19  5.30  5.49
## Cluster             22 964  1.00 0.00   1.00    1.00 0.00  1.00  1.00  0.00
##                    skew kurtosis   se
## Education          0.38    -0.96 0.03
## Income            17.24   435.81 0.03
## Kidhome            3.35    10.43 0.02
## Teenhome           0.54    -0.87 0.03
## Recency           -0.03    -1.21 0.03
## MntWines           0.53    -0.29 0.03
## MntFruits          1.08     0.35 0.04
## MntMeatProducts    1.20     2.03 0.03
## MntFishProducts    0.90    -0.08 0.04
## MntSweetProducts   1.06     0.38 0.04
## MntGoldProds       1.08     0.46 0.04
## NumWebPurchases    1.46     7.84 0.03
## NumStorePurchases -0.12    -0.85 0.03
## Complain          11.59   132.44 0.03
## Age                0.01    -0.02 0.03
## MembershipDays    -0.19    -1.17 0.03
## Divorced           2.44     3.96 0.03
## Married            0.52    -1.73 0.03
## Single             1.44     0.08 0.03
## Together           1.11    -0.77 0.03
## Widow              4.24    15.97 0.04
## Cluster             NaN      NaN 0.00
## ------------------------------------------------------------
## Cluster: 2
##                   vars    n  mean   sd median trimmed  mad   min   max range
```

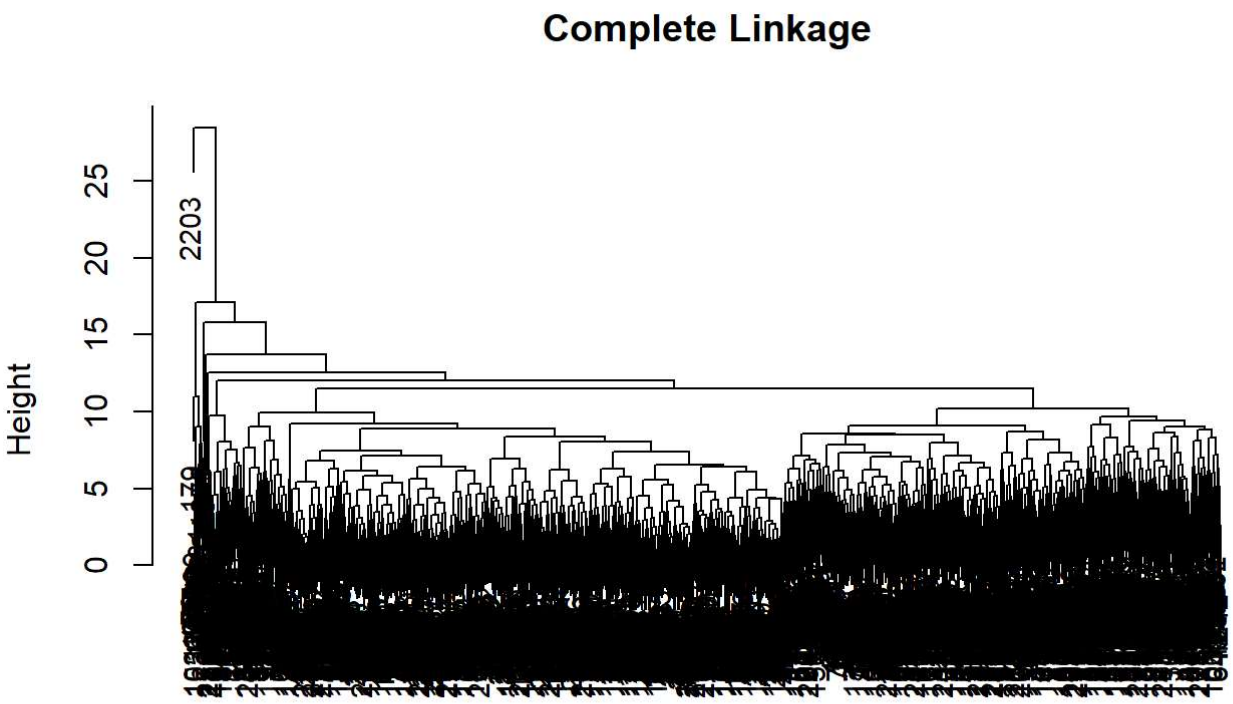
```
##                    vars    n  mean   sd median trimmed  mad   min   max range
## Education             1 1245 -0.07 1.03  -0.46   -0.05 1.48 -2.45  1.54  3.98
## Income                2 1245 -0.57 0.60  -0.59   -0.59 0.56 -2.00  4.37  6.38
## Kidhome               3 1245  0.52 0.97   1.04    0.54 0.00 -0.82  2.90  3.72
## Teenhome              4 1245  0.05 1.00   0.91    0.01 2.72 -0.93  2.75  3.67
## Recency               5 1245 -0.01 1.00   0.00   -0.02 1.28 -1.70  1.72  3.42
## MntWines              6 1245 -0.66 0.33  -0.81   -0.73 0.11 -0.90  1.32  2.22
## MntFruits             7 1245 -0.51 0.23  -0.59   -0.56 0.11 -0.66  1.10  1.76
## MntMeatProducts       8 1245 -0.59 0.27  -0.66   -0.63 0.08 -0.74  6.94  7.69
## MntFishProducts       9 1245 -0.53 0.25  -0.61   -0.58 0.11 -0.69  2.06  2.75
## MntSweetProducts     10 1245 -0.52 0.21  -0.59   -0.56 0.11 -0.66  1.24  1.90
## MntGoldProds         11 1245 -0.46 0.51  -0.62   -0.56 0.23 -0.85  4.22  5.07
## NumWebPurchases      12 1245 -0.50 0.70  -0.76   -0.60 0.54 -1.49  2.53  4.02
## NumStorePurchases    13 1245 -0.65 0.49  -0.86   -0.71 0.46 -1.78  1.90  3.69
## Complain             14 1245  0.02 1.09  -0.10   -0.10 0.00 -0.10 10.21 10.30
## Age                  15 1245 -0.13 0.95  -0.27   -0.17 0.87 -2.27  6.33  8.60
## MembershipDays       16 1245 -0.11 0.99  -0.15   -0.12 1.28 -1.75  1.71  3.46
## Divorced             17 1245 -0.02 0.97  -0.34   -0.34 0.00 -0.34  2.92  3.26
## Married              18 1245  0.02 1.00  -0.80   -0.03 0.00 -0.80  1.26  2.05
## Single               19 1245  0.01 1.01  -0.52   -0.16 0.00 -0.52  1.92  2.44
## Together             20 1245  0.00 1.00  -0.59   -0.13 0.00 -0.59  1.69  2.28
## Widow                21 1245 -0.06 0.84  -0.19   -0.19 0.00 -0.19  5.30  5.49
## Cluster              22 1245  2.00 0.00   2.00    2.00 0.00  2.00  2.00  0.00
##                     skew kurtosis   se
## Education           0.00    -0.34 0.03
## Income              1.59    11.83 0.02
## Kidhome            -0.21    -0.52 0.03
## Teenhome            0.31    -1.05 0.03
## Recency             0.02    -1.20 0.03
## MntWines            2.08     4.69 0.01
## MntFruits           3.09    12.19 0.01
## MntMeatProducts    18.69   517.13 0.01
## MntFishProducts     3.99    25.57 0.01
## MntSweetProducts    3.10    14.30 0.01
## MntGoldProds        3.16    14.23 0.01
## NumWebPurchases     1.17     1.12 0.02
## NumStorePurchases   1.21     1.85 0.01
## Complain            9.26    83.80 0.03
## Age                 0.64     1.90 0.03
## MembershipDays      0.11    -1.16 0.03
## Divorced            2.69     5.22 0.03
## Married             0.41    -1.83 0.03
## Single              1.37    -0.13 0.03
## Together            1.09    -0.82 0.03
## Widow               6.20    36.46 0.02
## Cluster              NaN      NaN 0.00
```

We have grouped the cluster stats into two based on cluster 1 and cluster two and we can observe that Cluster1: less educated, lower income, more kids, more teen homes, lower in age while these features are distinctinvly opposite in cluster 2.

Cluster with Hierarchical Clustering

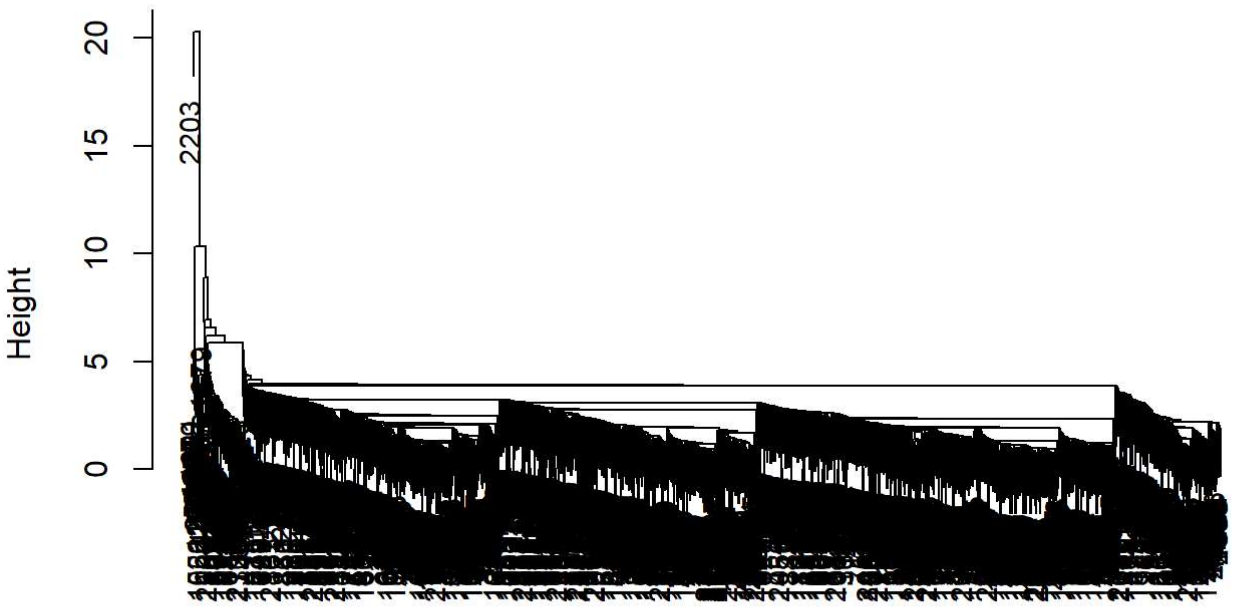
Q2.11 Perform clustering with Hierarchical method. Try complete, single and average linkage. Plot dendagram, based on it choose linkage and number of clusters, if possible, explain your choice. *(10 points)*

```
x_dist <- dist(df_scale)
hc.complete <- hclust(dist(df_scale), method = "complete")
plot(hc.complete, main = "Complete Linkage",
    xlab = "", sub = "", cex = .9)
```

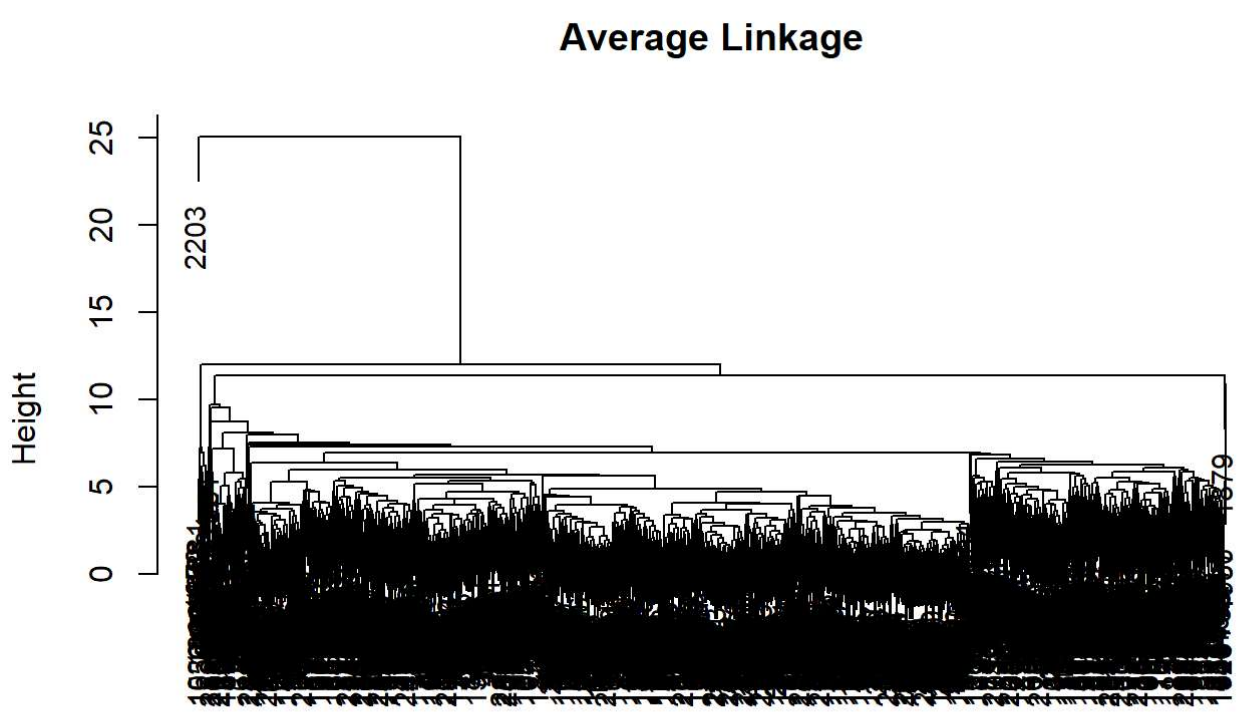



```
library(ggdendro)
#p <- ggdendrogram(hc.complete, rotate = FALSE)
#ggplotly(p)
```

```
hc.single <- hclust(dist(df_scale), method = "single")
plot(hc.single, main = "Single Linkage",
    xlab = "", sub = "", cex = .9)
```




```
hc.average <- hclust(dist(df_scale), method = "average")
plot(hc.average, main = "Average Linkage",
    xlab = "", sub = "", cex = .9)
```



Single linkage: closer observations but spread out clusters complete linkage: tighter observations but spread closely Average linkage: clusters are averaged to be closer We should be Complete or Averaged if we are to chose, while average linkage is still better between the two. # Additional grading criteria:

G3.1 Was all random methods properly seeded? *(2 points)* The data was randomly spread so we used seed(786) to get the same output everytime.