Homework 2. Clustering Practice (80 Points)

Lokesh Surendra Jain

2023-03-06

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

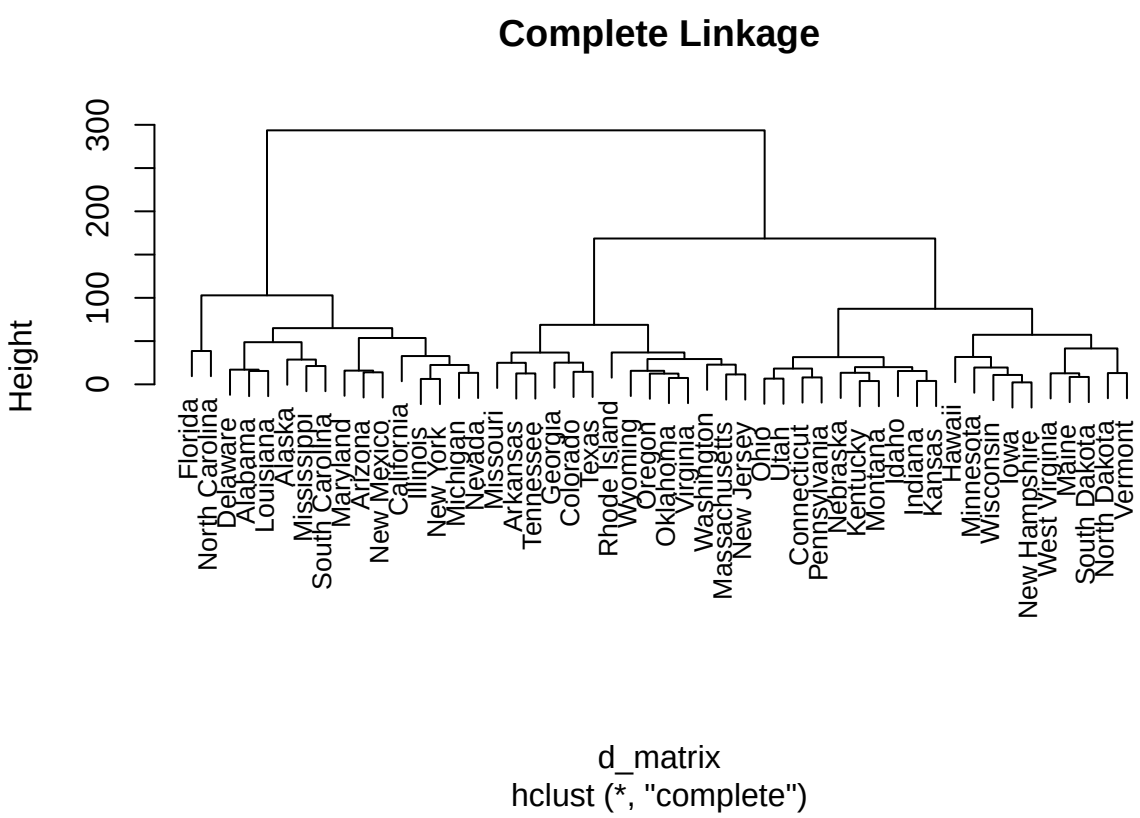
Consider the "USArrests" data. It is a built-in dataset you may directly get in RStudio. Perform hierarchical clustering on the observations (states) and answer the following questions.

```
head(USArrests)
```

```
##            Murder Assault UrbanPop Rape
## 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
```

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

```
set.seed(2)
data("USArrests")
data <- USArrests
data <- na.omit(data)
d_matrix <- dist(data, method = "euclidean")

hc <- hclust(d_matrix)
plot(hc, main="Complete Linkage", cex = .8)
```



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)

States like California, New York, Florida, and Illinois are included in Cluster 1 because they have greater rates of violent crimes and arrests. States like Arkansas, Georgia, and Tennessee, which have modest rates of violent crimes and arrests, are included in Cluster 2. Finally, Cluster 3 includes states like Maine, Montana, and Vermont that have lower rates of violent crime and arrests.

```
clust <- cutree(hc, 3)
clust
```

```
##        Alabama         Alaska        Arizona       Arkansas     California
##              1              1              1              2              1
##       Colorado    Connecticut       Delaware        Florida        Georgia
##              2              3              1              1              2
##         Hawaii          Idaho       Illinois        Indiana           Iowa
##              3              3              1              3              3
##         Kansas       Kentucky      Louisiana          Maine       Maryland
##              3              3              1              3              1
##  Massachusetts       Michigan      Minnesota    Mississippi       Missouri
##              2              1              3              1              2
##        Montana       Nebraska         Nevada  New Hampshire     New Jersey
##              3              3              1              3              2
##     New Mexico       New York North Carolina   North Dakota           Ohio
##              1              1              1              3              3
##       Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina
##              2              2              3              2              1
```

```
##   South Dakota      Tennessee          Texas           Utah        Vermont
##              3              2              2              3              3
##       Virginia     Washington  West Virginia      Wisconsin        Wyoming
##              2              2              3              3              2
```

```
table (clust)
```

```
## clust
##  1  2  3
## 16 14 20
```

```
subset(row.names(USArrests), clust == 1)
```

```
##  [1] "Alabama"        "Alaska"         "Arizona"        "California"
##  [5] "Delaware"       "Florida"        "Illinois"       "Louisiana"
##  [9] "Maryland"       "Michigan"       "Mississippi"    "Nevada"
## [13] "New Mexico"     "New York"       "North Carolina" "South Carolina"
```

```
subset(row.names(USArrests), clust == 2)
```

```
##  [1] "Arkansas"      "Colorado"      "Georgia"       "Massachusetts"
##  [5] "Missouri"      "New Jersey"    "Oklahoma"      "Oregon"
##  [9] "Rhode Island"  "Tennessee"     "Texas"         "Virginia"
## [13] "Washington"    "Wyoming"
```

```
subset(row.names(USArrests), clust == 3)
```

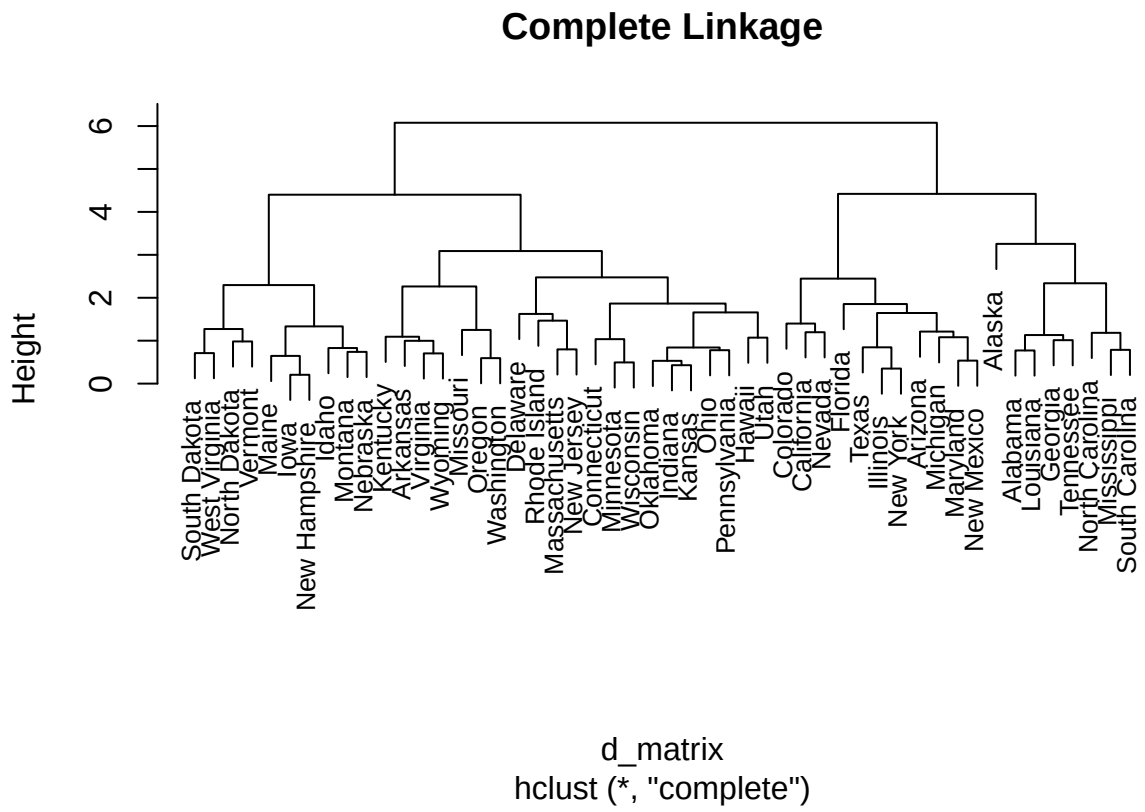
```
##  [1] "Connecticut"   "Hawaii"        "Idaho"         "Indiana"
##  [5] "Iowa"          "Kansas"        "Kentucky"      "Maine"
##  [9] "Minnesota"     "Montana"       "Nebraska"      "New Hampshire"
## [13] "North Dakota"  "Ohio"          "Pennsylvania"  "South Dakota"
## [17] "Utah"          "Vermont"       "West Virginia" "Wisconsin"
```

Q1.3 Hierarchically cluster the states using complete linkage and Euclidean distance, after scaling the variables to have standard deviation one. Obtain three clusters. Which states belong to which clusters?(5 points)

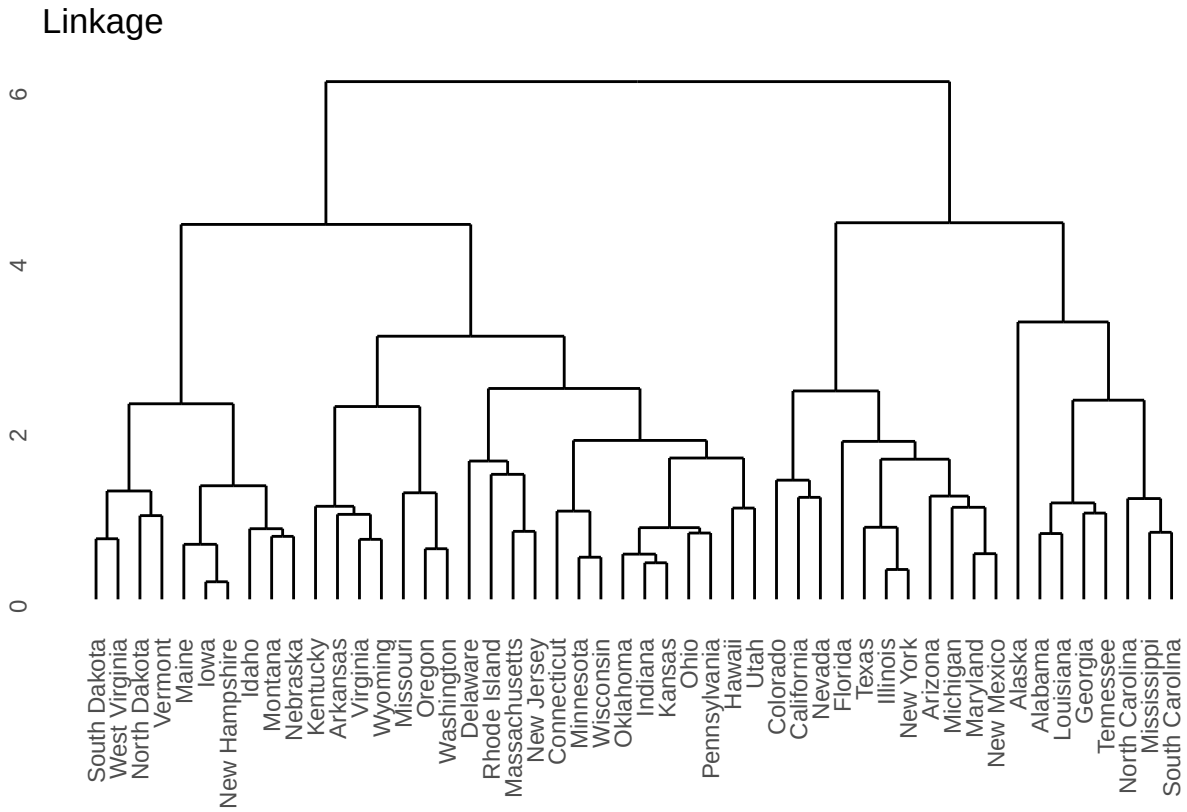
Based on their similarities and differences, the 50 states in the USArrests dataset were grouped into three clusters via the cutree() algorithm. Eight Southeastern states, including Alabama and Louisiana, are grouped together in the first cluster because they all have high rates of violent crime across the dataset's four categories. The second cluster consists of 11 states, including Arizona, California, and Texas, with intermediate levels of violent crime and murder, rape, and assault arrest rates. The remaining 32 states that have relatively lower rates of violent crimes and homicide, rape, and assault arrest rates make up the third cluster. As a result, the USArrests dataset's clustering exposes varied patterns of criminal behavior across several states.

```
set.seed(2)
data("USArrests")
data <- USArrests
data <- na.omit(data)
data_scale <- scale(data)
```

```r
d_matrix <- dist(data_scale, method = "euclidean")

hc_scaled <- hclust(d_matrix)
plot(hc_scaled, main="Complete Linkage", cex = .8)
```




```r
ggdendrogram(hc_scaled, segments = TRUE, lables = TRUE, leaf_labels = TRUE, rotate = FALSE, theme_dendro
```



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: Prior to executing hierarchical clustering, scaling variables can have a substantial impact on the clusters that are produced. If variables are not scaled, the distance calculation may favor the variables with higher variances over the rest, clustering data predominantly based on those variables. To ensure that each variable contributes equally to the distance calculation and prevent dominance by a single variable, variables can be scaled to have equal variances, for example by standardizing with a standard deviation of one.

Scaling variables can result in more precise and meaningful grouping, hence in my opinion it should be done before estimating inter-observation dissimilarities. By eliminating unit discrepancies between variables, scaling can also improve the meaning of comparisons between variables.

Part 2. Market Segmentation (60 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

```
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 (2 points)

marketing_data <- fread("m_marketing_campaign.csv")

marketing_data
```

```
##          ID Year_Birth Education Marital_Status Income Kidhome Teenhome
##    1:  5524       1957  Bachelor         Single  58138       0        0
##    2:  2174       1954  Bachelor         Single  46344       1        1
##    3:  4141       1965  Bachelor       Together  71613       0        0
##    4:  6182       1984  Bachelor       Together  26646       1        0
##    5:  5324       1981       PhD        Married  58293       1        0
##   ---
## 2205: 10870       1967  Bachelor        Married  61223       0        1
## 2206:  4001       1946       PhD       Together  64014       2        1
## 2207:  7270       1981  Bachelor       Divorced  56981       0        0
## 2208:  8235       1956    Master       Together  69245       0        1
## 2209:  9405       1954       PhD        Married  52869       1        1
##       Dt_Customer Recency MntWines MntFruits MntMeatProducts MntFishProducts
##    1:  04-09-2012      58      635        88             546             172
##    2:  08-03-2014      38       11         1               6               2
##    3:  21-08-2013      26      426        49             127             111
##    4:  10-02-2014      26       11         4              20              10
##    5:  19-01-2014      94      173        43             118              46
##   ---
```

```
## 2205:  13-06-2013      46      709        43             182              42
## 2206:  10-06-2014      56      406         0              30               0
## 2207:  25-01-2014      91      908        48             217              32
## 2208:  24-01-2014       8      428        30             214              80
## 2209:  15-10-2012      40       84         3              61               2
##       MntSweetProducts MntGoldProds NumWebPurchases NumStorePurchases Complain
##    1:               88          88               8                 4        0
##    2:                1           6               1                 2        0
##    3:               21          42               8                10        0
##    4:                3           5               2                 4        0
##    5:               27          15               5                 6        0
##   ---
## 2205:              118         247               9                 4        0
## 2206:                0           8               8                 5        0
## 2207:               12          24               2                13        0
## 2208:               30          61               6                10        0
## 2209:                1          21               3                 4        0
```

```
# Convert Year_Birth to Age (assume that current date is 2014-07-01) (2 points)
marketing_data$Age <- 2014 - marketing_data$Year_Birth


# Dt_Customer is a date (it is still character), convert it to membership days (i.e. number of days per
# hint: note European date format, use as.Date with proper format argument (2 points)

# marketing_data$Dt_Customer <- as.Date(marketing_data$Dt_Customer, format = "%d/%m/%Y")
# marketing_data$MembershipDays <- as.Date("2014-07-01") - marketing_data$Dt_Customer
# marketing_data$MembershipDays <- as.numeric(marketing_data$MembershipDays, units = "days")

marketing_data[, MembershipDays := as.Date("2014-07-01", format = "%Y-%m-%d") - as.Date(Dt_Customer, fo
marketing_data$MembershipDays <- as.numeric(marketing_data$MembershipDays)

marketing_data
```

```
##          ID Year_Birth Education Marital_Status Income Kidhome Teenhome
##    1:  5524       1957  Bachelor         Single  58138       0        0
##    2:  2174       1954  Bachelor         Single  46344       1        1
##    3:  4141       1965  Bachelor       Together  71613       0        0
##    4:  6182       1984  Bachelor       Together  26646       1        0
##    5:  5324       1981       PhD        Married  58293       1        0
##   ---
## 2205: 10870       1967  Bachelor        Married  61223       0        1
## 2206:  4001       1946       PhD       Together  64014       2        1
## 2207:  7270       1981  Bachelor       Divorced  56981       0        0
## 2208:  8235       1956    Master       Together  69245       0        1
## 2209:  9405       1954       PhD        Married  52869       1        1
##       Dt_Customer Recency MntWines MntFruits MntMeatProducts MntFishProducts
##    1:  04-09-2012      58      635        88             546             172
##    2:  08-03-2014      38       11         1               6               2
##    3:  21-08-2013      26      426        49             127             111
##    4:  10-02-2014      26       11         4              20              10
##    5:  19-01-2014      94      173        43             118              46
##   ---
## 2205:  13-06-2013      46      709        43             182              42
```

```
## 2206:  10-06-2014      56      406         0              30               0
## 2207:  25-01-2014      91      908        48             217              32
## 2208:  24-01-2014       8      428        30             214              80
## 2209:  15-10-2012      40       84         3              61               2
##       MntSweetProducts MntGoldProds NumWebPurchases NumStorePurchases Complain
##    1:               88           88               8                 4        0
##    2:                1            6               1                 2        0
##    3:               21           42               8                10        0
##    4:                3            5               2                 4        0
##    5:               27           15               5                 6        0
##   ---
## 2205:              118          247               9                 4        0
## 2206:                0            8               8                 5        0
## 2207:               12           24               2                13        0
## 2208:               30           61               6                10        0
## 2209:                1           21               3                 4        0
##       Age MembershipDays
##    1:  57            665
##    2:  60            115
##    3:  49            314
##    4:  30            141
##    5:  33            163
##   ---
## 2205:  47            383
## 2206:  68             21
## 2207:  33            157
## 2208:  58            158
## 2209:  60            624
```

```
# # Summarize Education column (use table function) (2 points)
#
#
# # Lets treat Education column as ordinal categories and use simple levels for distance calculations
# # Assuming following order of degrees:
# #    HighSchool, Associate, Bachelor, Master, PhD
# # factorize Education column (hint: use factor function with above levels)

table(marketing_data$Education)
```

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

```
# Factorize Education column
education_levels <- c("HighSchool", "Associate", "Bachelor", "Master", "PhD")
# education_levels
marketing_data$Education <- factor(marketing_data$Education, levels = education_levels)
marketing_data
```

```
##          ID Year_Birth Education Marital_Status Income Kidhome Teenhome
##    1:  5524       1957  Bachelor         Single  58138       0        0
##    2:  2174       1954  Bachelor         Single  46344       1        1
##    3:  4141       1965  Bachelor       Together  71613       0        0
```

```
##    4:  6182       1984  Bachelor       Together  26646       1        0
##    5:  5324       1981       PhD        Married  58293       1        0
##   ---
## 2205: 10870       1967  Bachelor        Married  61223       0        1
## 2206:  4001       1946       PhD       Together  64014       2        1
## 2207:  7270       1981  Bachelor       Divorced  56981       0        0
## 2208:  8235       1956    Master       Together  69245       0        1
## 2209:  9405       1954       PhD        Married  52869       1        1
##       Dt_Customer Recency MntWines MntFruits MntMeatProducts MntFishProducts
##    1:  04-09-2012      58      635        88             546             172
##    2:  08-03-2014      38       11         1               6               2
##    3:  21-08-2013      26      426        49             127             111
##    4:  10-02-2014      26       11         4              20              10
##    5:  19-01-2014      94      173        43             118              46
##   ---
## 2205:  13-06-2013      46      709        43             182              42
## 2206:  10-06-2014      56      406         0              30               0
## 2207:  25-01-2014      91      908        48             217              32
## 2208:  24-01-2014       8      428        30             214              80
## 2209:  15-10-2012      40       84         3              61               2
##       MntSweetProducts MntGoldProds NumWebPurchases NumStorePurchases Complain
##    1:               88           88               8                 4        0
##    2:                1            6               1                 2        0
##    3:               21           42               8                10        0
##    4:                3            5               2                 4        0
##    5:               27           15               5                 6        0
##   ---
## 2205:              118          247               9                 4        0
## 2206:                0            8               8                 5        0
## 2207:               12           24               2                13        0
## 2208:               30           61               6                10        0
## 2209:                1           21               3                 4        0
##       Age MembershipDays
##    1:  57            665
##    2:  60            115
##    3:  49            314
##    4:  30            141
##    5:  33            163
##   ---
## 2205:  47            383
## 2206:  68             21
## 2207:  33            157
## 2208:  58            158
## 2209:  60            624
```

```
# Summarize Marital_Status column (use table function)

table(marketing_data$Marital_Status)
```

```
##
## Divorced  Married   Single Together    Widow
##      232      857      471      573       76
```

```r
# Lets convert single Marital_Status categories for 5 separate binary categories  (2 points)
# 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 groups)

marketing_data$Divorced <- ifelse(marketing_data$Marital_Status == "Divorced", 1, 0)
marketing_data$Married <- ifelse(marketing_data$Marital_Status == "Married", 1, 0)
marketing_data$Single <- ifelse(marketing_data$Marital_Status == "Single", 1, 0)
marketing_data$Together <- ifelse(marketing_data$Marital_Status == "Together", 1, 0)
marketing_data$Widow <- ifelse(marketing_data$Marital_Status == "Widow", 1, 0)

head(marketing_data)
```

```
##      ID Year_Birth Education Marital_Status Income Kidhome Teenhome Dt_Customer
## 1: 5524       1957  Bachelor         Single  58138       0        0  04-09-2012
## 2: 2174       1954  Bachelor         Single  46344       1        1  08-03-2014
## 3: 4141       1965  Bachelor       Together  71613       0        0  21-08-2013
## 4: 6182       1984  Bachelor       Together  26646       1        0  10-02-2014
## 5: 5324       1981       PhD        Married  58293       1        0  19-01-2014
## 6: 7446       1967    Master       Together  62513       0        1  09-09-2013
##    Recency MntWines MntFruits MntMeatProducts MntFishProducts MntSweetProducts
## 1:      58      635        88             546             172               88
## 2:      38       11         1               6               2                1
## 3:      26      426        49             127             111               21
## 4:      26       11         4              20              10                3
## 5:      94      173        43             118              46               27
## 6:      16      520        42              98               0               42
##    MntGoldProds NumWebPurchases NumStorePurchases Complain Age MembershipDays
## 1:           88               8                 4        0  57           665
## 2:            6               1                 2        0  60           115
## 3:           42               8                10        0  49           314
## 4:            5               2                 4        0  30           141
## 5:           15               5                 6        0  33           163
## 6:           14               6                10        0  47           295
##    Divorced Married Single Together Widow
## 1:        0       0      1        0     0
## 2:        0       0      1        0     0
## 3:        0       0      0        1     0
## 4:        0       0      0        1     0
## 5:        0       1      0        0     0
## 6:        0       0      0        1     0
```

```r
# lets remove columns which we will no longer use:
# remove ID, Year_Birth, Dt_Customer, Marital_Status
# and save it as df_sel

df_sel <- subset(marketing_data, select = -c(ID, Year_Birth, Dt_Customer, Marital_Status))

# 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_sel$Education <- as.integer(df_sel$Education)
df_sel
```

```
##       Education Income Kidhome Teenhome Recency MntWines MntFruits
##    1:         3  58138       0        0      58      635        88
##    2:         3  46344       1        1      38       11         1
##    3:         3  71613       0        0      26      426        49
##    4:         3  26646       1        0      26       11         4
##    5:         5  58293       1        0      94      173        43
##   ---
## 2205:         3  61223       0        1      46      709        43
## 2206:         5  64014       2        1      56      406         0
## 2207:         3  56981       0        0      91      908        48
## 2208:         4  69245       0        1       8      428        30
## 2209:         5  52869       1        1      40       84         3
##       MntMeatProducts MntFishProducts MntSweetProducts MntGoldProds
##    1:             546             172               88           88
##    2:               6               2                1            6
##    3:             127             111               21           42
##    4:              20              10                3            5
##    5:             118              46               27           15
##   ---
## 2205:             182              42              118          247
## 2206:              30               0                0            8
## 2207:             217              32               12           24
## 2208:             214              80               30           61
## 2209:              61               2                1           21
##       NumWebPurchases NumStorePurchases Complain Age MembershipDays Divorced
##    1:               8                 4        0  57            665        0
##    2:               1                 2        0  60            115        0
##    3:               8                10        0  49            314        0
##    4:               2                 4        0  30            141        0
##    5:               5                 6        0  33            163        0
##   ---
## 2205:               9                 4        0  47            383        0
## 2206:               8                 5        0  68             21        0
## 2207:               2                13        0  33            157        1
## 2208:               6                10        0  58            158        0
## 2209:               3                 4        0  60            624        0
##       Married Single Together Widow
##    1:       0      1        0     0
##    2:       0      1        0     0
##    3:       0      0        1     0
##    4:       0      0        1     0
##    5:       1      0        0     0
##   ---
## 2205:       1      0        0     0
## 2206:       0      0        1     0
## 2207:       0      0        0     0
## 2208:       0      0        1     0
## 2209:       1      0        0     0
```

```
# lets scale (2 points)
# run scale function on df_sel and save it as df_scale
# that will be our scaled values which we will use for analysis
# convert factor columns to numeric
```

```
df_scale <- scale(df_sel)
# df_scale
```

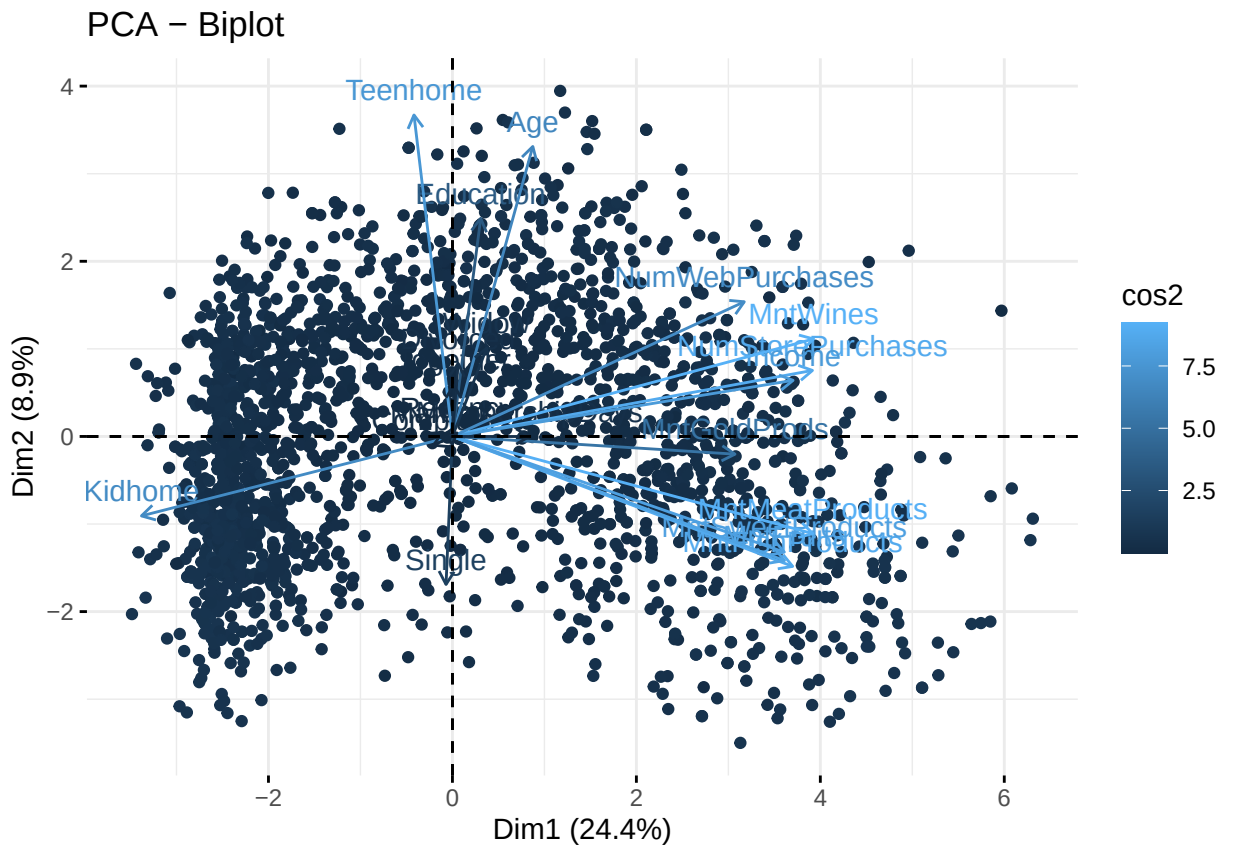
PCA

Q2.2. Run PCA *(6 points)*

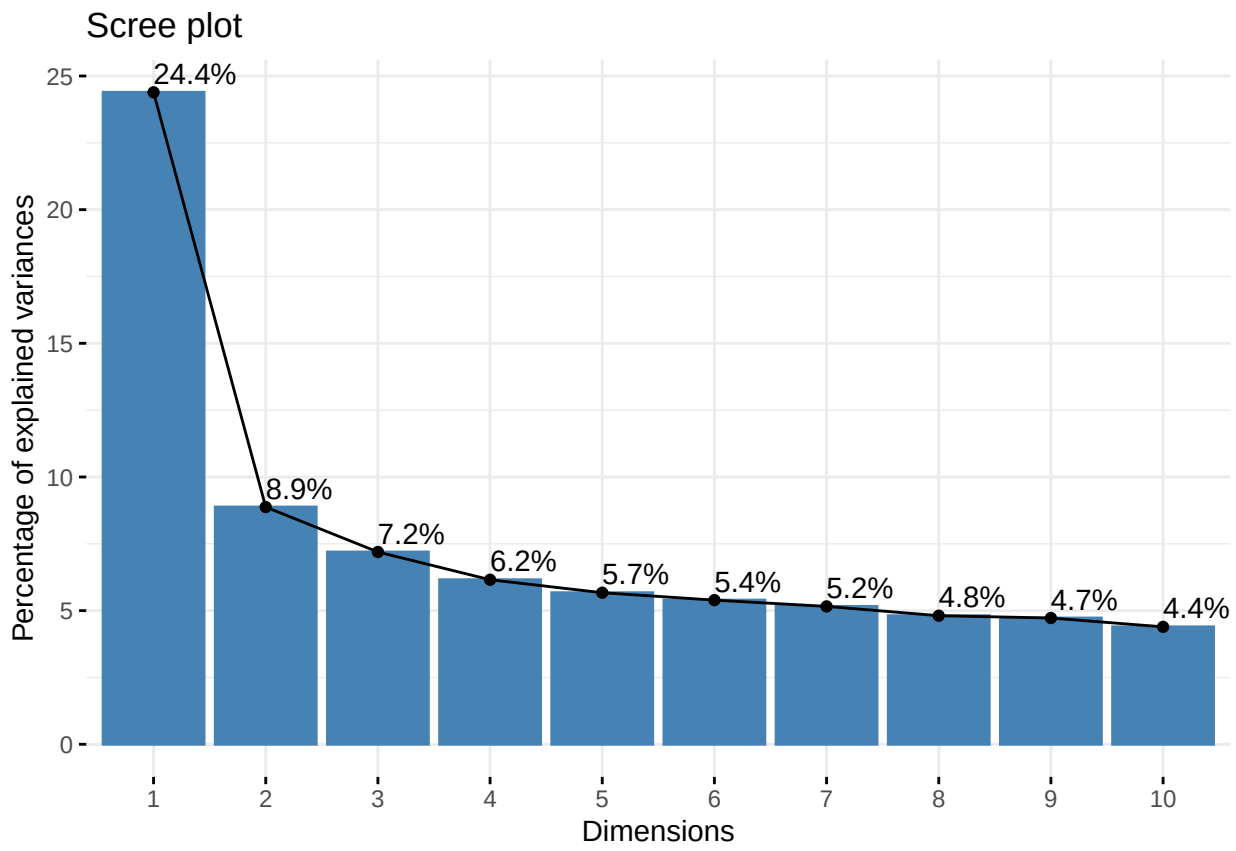
```
# Run PCA on df_scale, make biplot and scree plot/percentage variance explained plot
# save as pc_out, we will use pc_out$x[,1] and pc_out$x[,2] later for plotting
library(FactoMineR)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

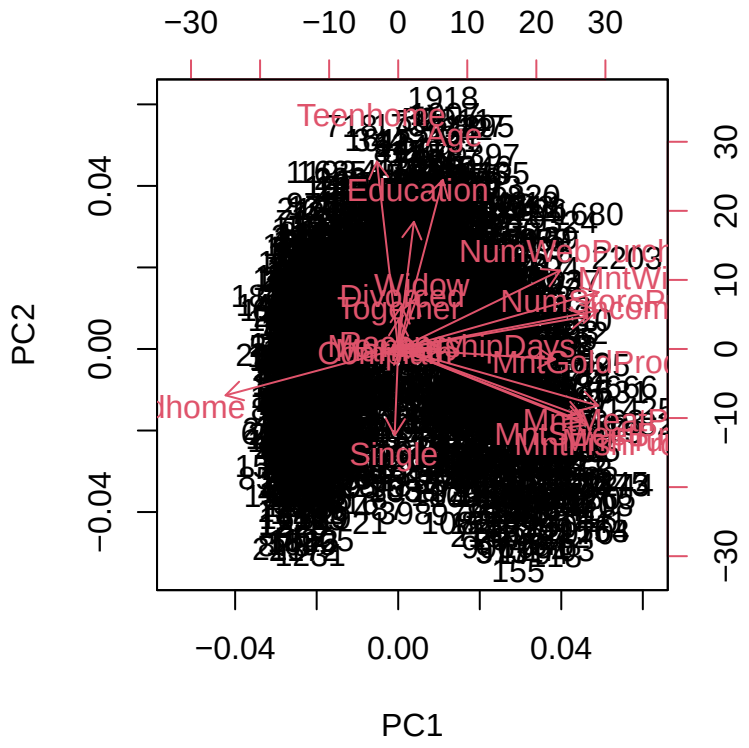
```
pc_out <- PCA(df_scale, graph = FALSE)

# Create biplot
fviz_pca_biplot(pc_out, col.var = "contrib", col.ind = "cos2", geom = "point",
                select.var = list(contrib = 100), axes = c(1,2))
```




```
# Create scree plot
fviz_eig(pc_out, addlabels = TRUE)
```




```
pca <- prcomp(df_scale, center = TRUE, scale. = TRUE)

# create biplot
biplot(pca, choices = c(1, 2))
```



Q2.3 Comment on observation (any visible distinct clusters?) *(2 points)*

Cluster with K-Means

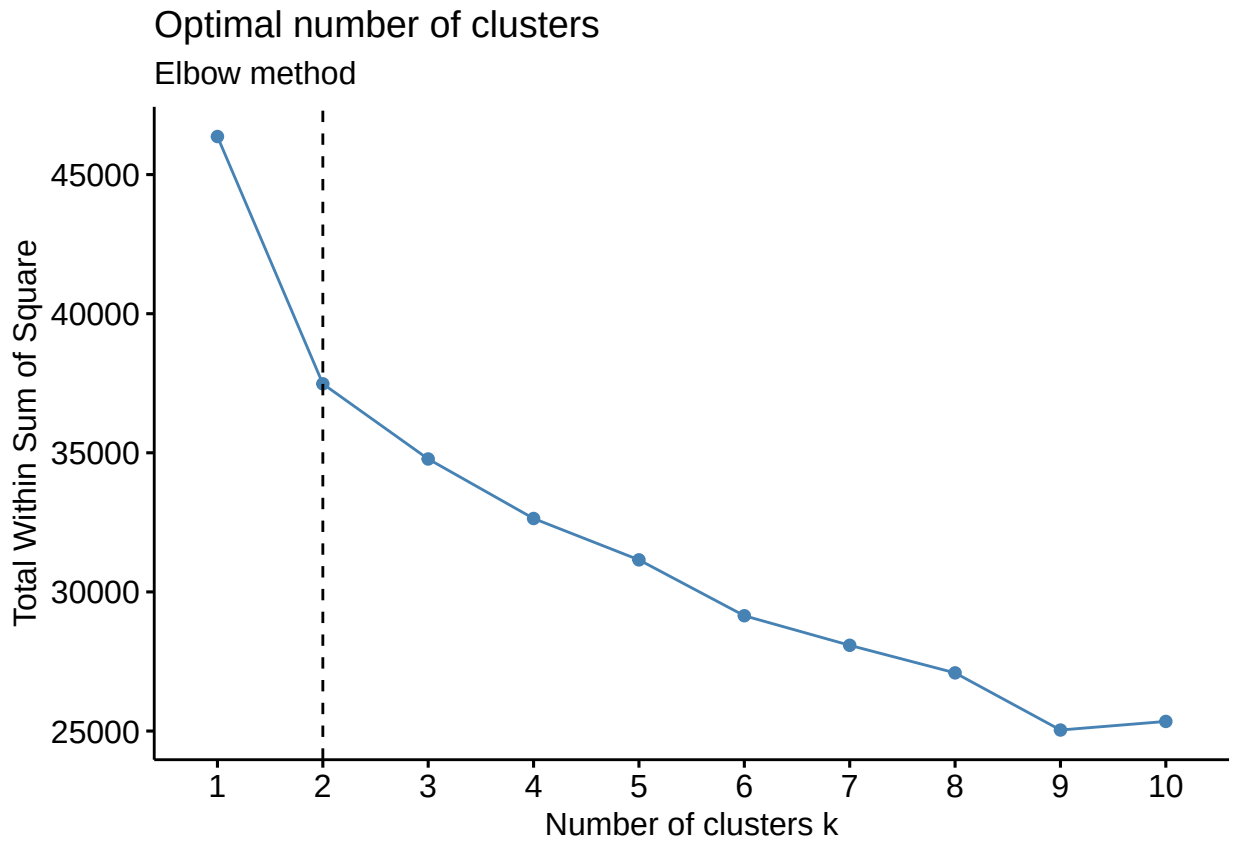
In questions Q2.4 to Q2.9 use K-Means method for clustering

Selecting Number of Clusters

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

```
km_out_list <- lapply(1:10, function(k) list(
  k=k,
  km_out=kmeans(df_scale, k, nstart = 20)))

km_results <- data.frame(
  k=sapply(km_out_list, function(k) k$k),
  totss=sapply(km_out_list, function(k) k$km_out$totss),
  tot_withinss=sapply(km_out_list, function(k) k$km_out$tot.withinss)
  )
```

```
set.seed(1)
fviz_nbclust(df_scale, kmeans, method = "wss",k.max=10, nstart=20, iter.max=20) +
  geom_vline(xintercept = 2, linetype = 2)+
  labs(subtitle = "Elbow method")
```

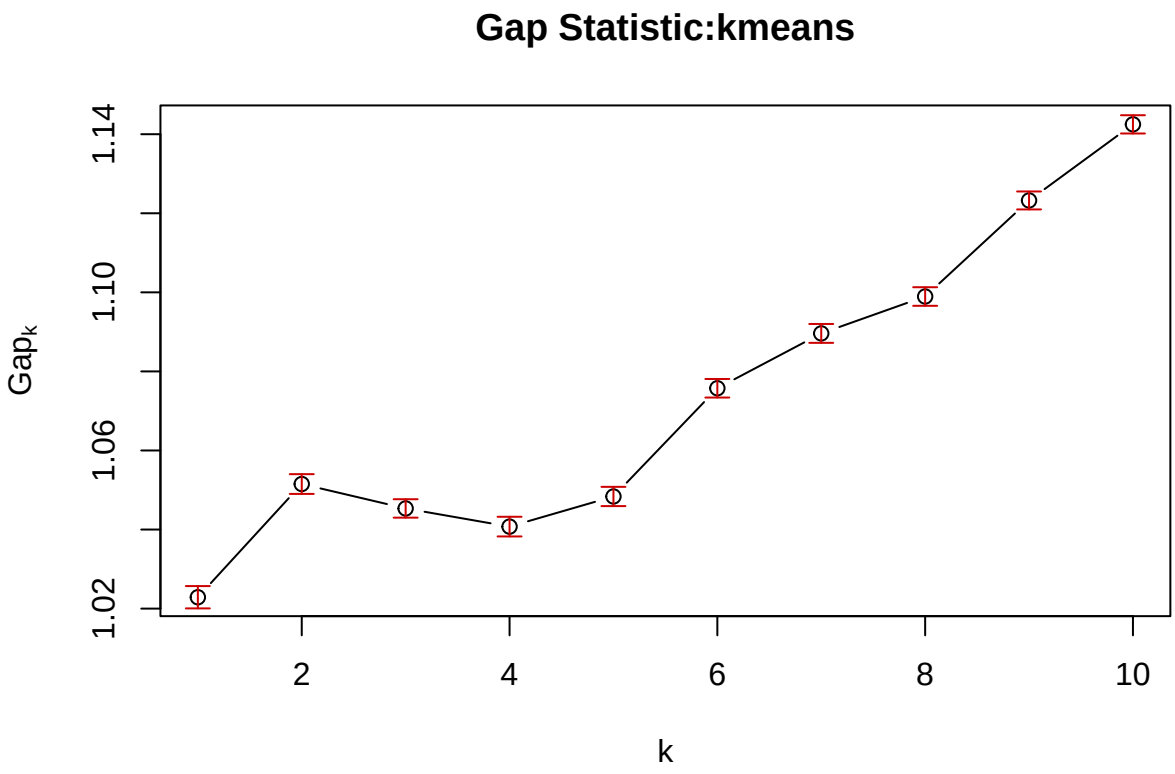


Q2.5 Select optimal number of clusters using Gap Statistic. *(4 points)*

```
set.seed(1)
gap_kmeans <- clusGap(df_scale, kmeans, nstart = 20, K.max = 10, B = 100, iter.max= 20)
```

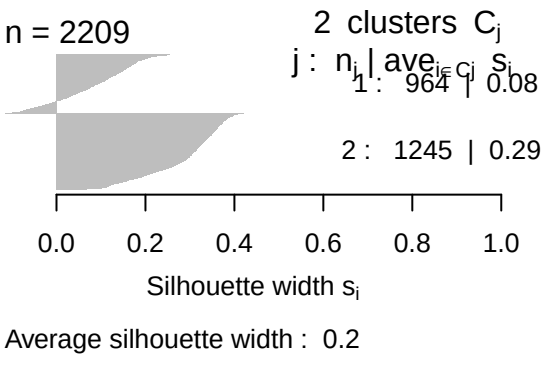
```
## Warning: Quick-TRANSfer stage steps exceeded maximum (= 110450)

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

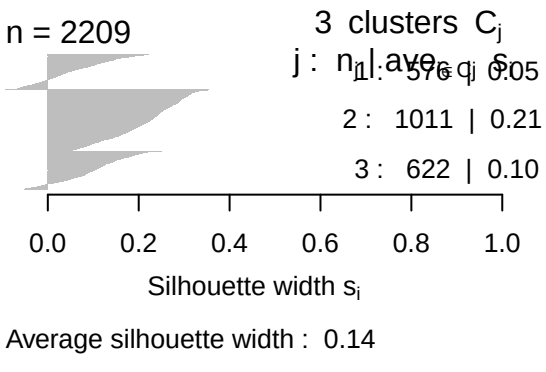
```
plot(gap_kmeans, main = "Gap Statistic:kmeans")
```

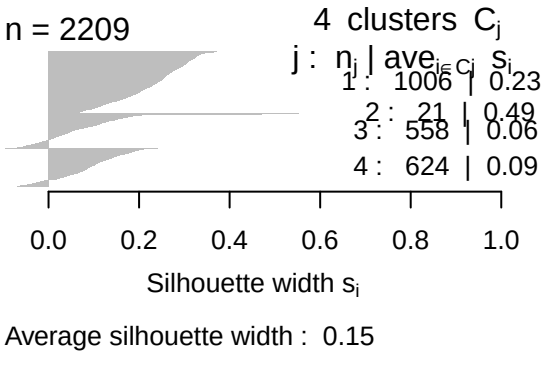


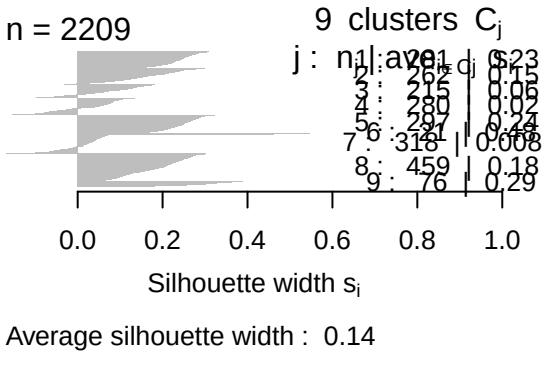
Q2.6 Select optimal number of clusters using Silhouette method. *(4 points)*

```
set.seed(3)
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, that is two groups don't provide sufficient benefit over a single groups)?*(4 points)*

Answer: Number of k = 2 and we can select k = 3 for elbow, gap statistics and silhuettes as well as clustering

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)*

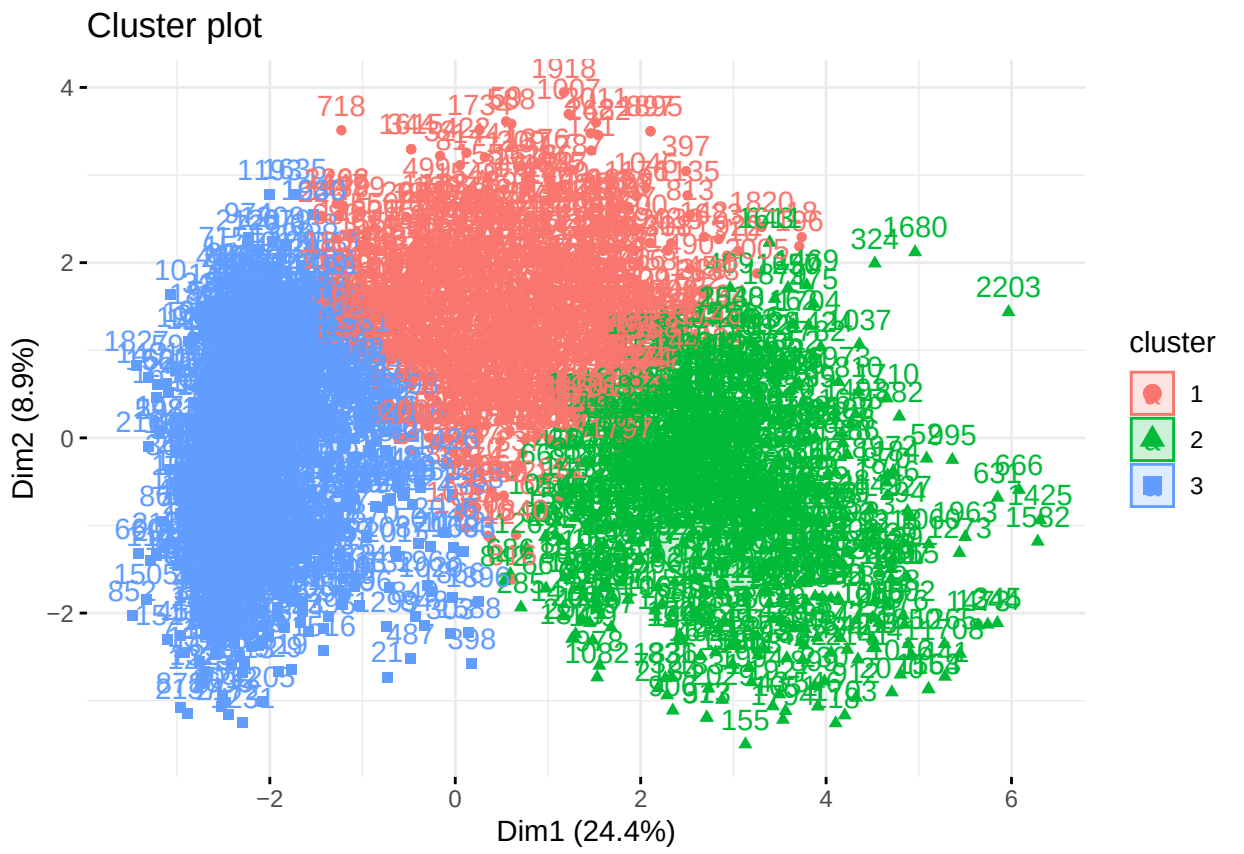
```
set.seed(4)
Km_out <- kmeans(df_scale, 3, nstart = 25)
Km_out
```

```
## K-means clustering with 3 clusters of sizes 622, 576, 1011
##
## Cluster means:
##    Education     Income    Kidhome   Teenhome     Recency   MntWines  MntFruits
## 1  0.3478062  0.2718747 -0.5443171  0.7229378 -0.05994217  0.4978102 -0.2162636
## 2 -0.1030341  0.9448261 -0.6901899 -0.5390242  0.03485598  0.8259743  1.1644365
## 3 -0.1552797 -0.7055647  0.7281054 -0.1376750  0.01701977 -0.7768538 -0.5303654
```

```
##   MntMeatProducts MntFishProducts MntSweetProducts MntGoldProds NumWebPurchases
## 1      -0.1370652      -0.2152901       -0.2181433    0.2179156       0.6879626
## 2       1.2496660       1.2013664        1.1669137    0.7019655       0.4936514
## 3      -0.6276490      -0.5520046       -0.5306204   -0.5340016      -0.7045064
##   NumStorePurchases     Complain         Age MembershipDays    Divorced
## 1         0.4874473 -0.048253246  0.49462206      0.1244346  0.07693490
## 2         0.8682646 -0.008510373 -0.02769545      0.1023490 -0.02544448
## 3        -0.7945723  0.034535603 -0.28852853     -0.1348678 -0.03283629
##       Married      Single    Together       Widow
## 1  0.01217148 -0.18296271  0.06475341  0.09348066
## 2 -0.04439704  0.10249546 -0.03726687  0.02078759
## 3  0.01780616  0.05416955 -0.01860623 -0.06935571
##
## Clustering vector:
##    [1] 2 3 2 3 3 1 1 3 3 3 3 2 1 3 2 3 1 1 3 3 3 1 1 1 3 3 3 2 3 3 3 1 2 3 1 3 3
##   [38] 1 2 3 3 3 2 3 3 1 1 2 3 2 1 2 2 3 1 2 1 1 1 2 3 3 2 1 1 2 2 1 3 3 2 2 3 1
##   [75] 3 3 3 3 2 3 3 1 2 3 3 3 3 1 3 2 1 3 3 2 2 2 3 3 2 3 2 2 2 1 2 3 3 2 2 3 3
##  [112] 1 3 3 3 2 1 2 3 1 1 2 3 2 3 3 3 2 1 2 1 3 1 3 3 3 3 1 1 1 1 1 1 3 3 2 3 1
##  [149] 3 1 2 3 1 3 2 3 3 3 3 3 3 2 2 3 3 2 3 3 1 3 3 3 3 1 2 3 3 2 3 3 3 3 1 2 2
##  [186] 3 1 2 2 2 3 3 3 3 3 1 1 2 3 1 2 3 3 1 1 1 3 2 1 3 1 3 1 1 2 3 3 2 3 3 1 3
##  [223] 3 1 3 1 2 2 3 2 1 3 2 1 2 2 3 3 2 3 1 3 1 1 3 3 3 1 3 3 1 3 2 3 2 3 2 3 1
##  [260] 3 3 1 2 2 2 1 3 1 1 1 3 3 2 1 2 1 3 3 2 3 3 1 3 3 2 1 3 1 3 3 3 2 3 2 1 3
##  [297] 3 3 2 3 3 3 3 3 1 3 3 1 1 1 3 3 3 3 3 3 1 3 3 1 2 3 2 2 2 3 1 1 3 2 3 2 3
##  [334] 3 1 2 1 2 1 3 3 2 1 1 2 1 3 3 1 1 2 3 2 1 3 3 3 1 3 3 3 3 1 3 3 3 1 3 3 3
##  [371] 1 1 3 1 2 3 2 1 1 2 3 3 3 3 3 2 3 1 1 3 3 1 3 1 3 2 1 3 1 2 3 2 2 1 3 3 3
##  [408] 2 2 3 2 1 3 2 1 1 2 1 3 3 1 1 3 3 3 1 3 3 3 3 3 2 3 1 1 1 3 3 1 1 2 3 1 2
##  [445] 1 2 1 2 3 2 1 3 1 1 2 3 1 3 3 1 3 1 1 3 3 3 3 3 2 2 1 1 3 3 2 3 2 1 1 1 3
##  [482] 1 1 1 3 3 3 2 3 1 2 2 3 1 3 2 1 2 1 2 3 3 1 2 3 2 3 1 3 3 1 3 2 1 1 2 1 3
##  [519] 3 1 3 2 1 3 3 1 3 1 1 3 2 3 3 3 3 3 1 3 2 3 2 2 3 1 3 2 1 2 1 1 1 3 3 3 1
##  [556] 3 3 1 3 1 3 3 3 3 3 1 3 3 3 3 2 1 1 3 3 2 2 3 1 3 3 3 3 3 3 1 2 1 3 3 3 3
##  [593] 3 2 3 1 3 3 2 3 3 3 3 1 1 3 1 3 2 3 2 2 3 3 2 1 2 3 2 3 2 1 1 1 1 1 1 1 2
##  [630] 3 2 3 1 1 2 3 2 3 1 3 1 3 2 3 1 3 2 3 3 3 3 3 3 3 1 1 2 2 1 3 1 2 3 2 1 2
##  [667] 2 1 2 1 2 2 2 2 1 2 1 3 3 1 3 3 1 2 1 2 1 2 3 2 3 1 1 3 3 1 3 1 3 2 2 3 2
##  [704] 3 1 1 3 2 3 3 2 2 3 2 3 1 1 1 1 2 2 1 3 2 1 3 3 3 2 2 3 2 3 2 2 1 2 2 2 2
##  [741] 1 1 3 3 3 1 2 3 2 1 2 2 3 1 1 2 1 3 3 3 3 1 3 2 2 3 3 3 3 3 3 1 1 1 2 1 3
##  [778] 3 3 1 1 1 2 3 1 3 3 2 2 1 3 1 1 2 3 3 2 1 2 1 3 2 1 3 2 3 1 3 1 2 1 3 1 3
##  [815] 3 1 1 3 3 2 3 2 3 1 3 1 3 3 2 2 2 1 3 3 1 1 2 3 1 2 3 2 3 2 3 3 3 3 3 1 1
##  [852] 3 1 3 1 1 3 3 2 2 1 3 2 3 3 3 3 3 2 2 3 3 1 2 3 1 2 3 1 1 1 2 3 3 2 1 2 1
##  [889] 1 2 2 3 3 3 2 2 1 3 2 2 1 1 1 2 3 2 3 3 2 1 2 2 2 1 2 3 1 3 2 3 2 1 1 1 2
##  [926] 1 2 2 3 1 1 3 3 1 3 3 3 3 3 3 1 1 3 1 2 1 3 3 3 1 2 3 3 2 2 3 3 1 2 2 2 3
##  [963] 3 1 3 3 3 2 2 1 2 2 2 3 2 3 1 2 3 3 2 3 2 1 1 2 1 3 1 1 1 2 3 3 2 3 3 3 3
## [1000] 1 2 3 3 3 3 3 1 3 3 1 3 3 3 1 2 2 2 3 2 3 3 3 3 1 1 3 3 2 3 3 3 2 3 1 2 3
## [1037] 2 3 3 2 1 1 2 2 1 1 2 3 1 3 2 2 3 2 3 2 1 3 1 2 2 3 3 3 2 3 2 3 2 1 3 1 3
## [1074] 2 2 3 1 3 3 1 1 2 3 1 2 1 3 3 3 2 3 3 1 3 2 2 3 2 3 1 3 3 3 1 1 3 3 3 1 1
## [1111] 2 3 3 2 1 3 3 2 2 3 3 2 3 1 1 3 3 3 2 3 3 1 1 3 1 2 3 2 1 3 1 2 2 2 1 1 1
## [1148] 2 1 1 3 3 2 2 3 3 2 1 3 1 3 1 3 2 1 1 1 3 3 3 3 1 1 3 2 1 3 1 3 1 3 1 1 2
## [1185] 3 2 1 3 2 1 2 3 3 3 1 2 1 2 3 3 1 3 2 3 3 3 2 3 3 1 1 3 1 3 3 3 3 3 3 2 3
## [1222] 2 3 3 3 3 2 1 3 3 3 3 3 1 2 1 2 2 1 1 1 2 3 2 3 2 2 3 3 2 1 3 1 2 1 1 3 1
## [1259] 3 1 3 3 2 1 2 2 3 1 2 3 1 3 2 2 3 3 3 3 3 1 3 1 2 3 3 2 3 3 2 1 1 1 1 1 1
## [1296] 2 1 2 3 3 1 3 3 3 2 1 2 3 3 1 3 3 3 2 3 3 2 2 2 1 2 3 3 3 1 1 1 3 3 3 3 1
## [1333] 1 2 2 2 3 1 2 2 3 1 2 3 1 3 1 2 1 2 1 3 3 1 3 3 3 1 1 1 1 3 1 1 3 3 2 3 3
## [1370] 2 3 3 3 3 3 3 1 3 3 1 1 1 1 3 1 1 3 1 1 1 3 1 2 3 2 3 3 3 3 3 3 3 3 2 1 3 3 3
## [1407] 3 3 3 2 3 3 2 3 1 3 1 3 3 3 3 3 3 2 2 3 2 1 2 1 3 2 2 3 1 2 1 3 2 2 1 1 3
## [1444] 3 3 1 1 2 3 2 3 3 3 1 1 1 2 3 3 1 2 1 3 3 2 1 2 1 2 3 1 1 2 1 3 2 3 1 2 2
```

```
## [1481] 1 3 3 1 1 1 1 2 2 1 2 3 2 2 3 3 1 3 3 3 2 2 3 3 3 1 2 3 2 3 1 3 1 3 3 3 3
## [1518] 2 1 2 3 1 2 3 1 3 3 3 1 3 3 2 2 2 2 3 3 3 3 2 3 1 3 2 3 1 1 1 1 1 2 3 2 3
## [1555] 3 1 3 1 3 1 3 1 2 3 2 3 3 3 3 2 1 2 3 1 3 1 3 3 2 3 1 2 1 3 3 3 2 3 1 1 2
## [1592] 3 2 3 3 2 3 3 1 1 1 3 2 1 3 1 2 2 3 3 2 3 3 3 1 1 1 2 3 3 3 1 1 3 2 3 1 2
## [1629] 2 3 3 2 3 2 3 3 1 1 1 2 1 3 1 3 3 3 1 3 2 2 3 2 2 2 2 3 3 3 3 3 2 3 3 3 1
## [1666] 3 2 1 2 2 1 2 3 3 3 2 3 1 3 2 2 3 3 1 3 3 1 1 2 1 2 1 3 2 3 3 1 3 3 1 3 2
## [1703] 2 2 3 3 3 3 1 2 3 3 3 2 2 1 1 2 1 3 3 3 3 2 1 2 3 1 2 2 1 1 3 1 3 3 3 3 3
## [1740] 2 2 3 1 1 3 1 3 2 3 3 3 1 2 2 3 3 3 3 3 2 3 3 2 1 3 3 1 3 2 2 3 3 2 3 3 3
## [1777] 1 3 1 2 2 2 3 3 3 1 1 3 2 2 3 1 1 2 2 3 1 2 1 3 1 3 3 2 2 3 1 2 2 3 3 1 1
## [1814] 3 3 3 2 1 1 1 3 2 3 2 1 1 3 3 1 3 2 3 2 2 2 1 3 3 2 1 1 3 1 2 2 1 3 3 3 2
## [1851] 3 2 3 1 3 3 2 2 1 2 2 3 1 1 3 3 3 2 2 3 2 2 3 1 2 1 3 2 2 1 3 3 3 3 1 2 1
## [1888] 3 3 3 2 2 2 2 1 1 3 3 3 3 3 2 2 2 2 3 1 2 2 1 3 1 3 3 2 3 1 1 3 3 1 3 3 2
## [1925] 3 2 2 2 3 1 3 1 2 2 1 1 3 1 2 3 2 1 3 3 3 2 1 2 2 2 1 3 3 1 3 1 2 3 3 3 3
## [1962] 3 2 1 3 3 3 3 3 1 2 3 2 2 3 2 2 1 3 1 3 3 3 3 3 3 2 1 1 1 2 3 2 2 2 3 3 3
## [1999] 3 3 3 3 3 1 1 3 3 3 1 3 1 1 3 2 3 1 2 1 3 2 2 2 3 3 3 3 3 2 2 2 3 1 1 3 3
## [2036] 3 2 2 1 3 2 3 2 1 3 1 3 3 1 2 3 2 2 1 3 3 3 1 2 1 2 2 3 3 1 3 1 1 3 2 3 2
## [2073] 1 3 1 3 1 3 3 1 1 1 2 1 1 3 3 1 1 3 1 2 3 3 3 3 2 3 2 1 2 1 3 3 2 1 3 1 3
## [2110] 3 1 3 3 3 3 3 1 2 3 3 1 3 3 2 3 3 3 3 3 3 1 1 1 3 1 2 3 3 2 2 2 3 1 1 2 1
## [2147] 1 2 2 1 1 3 1 3 3 3 2 1 2 2 3 2 3 3 2 2 3 3 1 1 3 3 1 2 3 3 2 3 3 3 1 2 3
## [2184] 2 3 3 3 2 3 3 1 2 3 3 1 1 1 1 3 3 1 3 2 3 2 1 2 1 3
##
## Within cluster sum of squares by cluster:
## [1]  9517.187 12525.161 12736.085
##  (between_SS / total_SS =  25.0 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
set.seed(6)
fviz_cluster(Km_out, data = df_scale, ellipse.type = 'euclid', ggtheme = theme_minimal() )
```




```
set.seed(19)
Km_out$cluster <- as.factor(Km_out$cluster)
df_kmeans <- cbind(df_sel, cluster = Km_out$cluster)
```

Q2.9 Do you see any grouping? Comment on you observation.*(2 points)*

Answer: Yes there are 3 groups. The k-means algorithm identified three clusters on the plot. The red cluster is high spenders who make frequent purchases, the green cluster is low spenders who make infrequent purchases, and the blue cluster is high spenders who make purchases less frequently

Characterizing Cluster

Q2.10 Perform descriptive statistics analysis on obtained cluster. Based on that does one or more group have a distinct characteristics? *(8 points)* Hint: add cluster column to original df dataframe

```
set.seed(14)
Km_out$cluster <- as.factor(Km_out$cluster)
df_kmeans <- cbind(df_sel, cluster= Km_out$cluster)

agg_kmeans <- aggregate(df_kmeans[,1:20], by= list(df_kmeans$cluster), mean) %>% as.data.frame()

agg_kmeans
```

```
##   Group.1 Education   Income    Kidhome  Teenhome  Recency  MntWines MntFruits
```

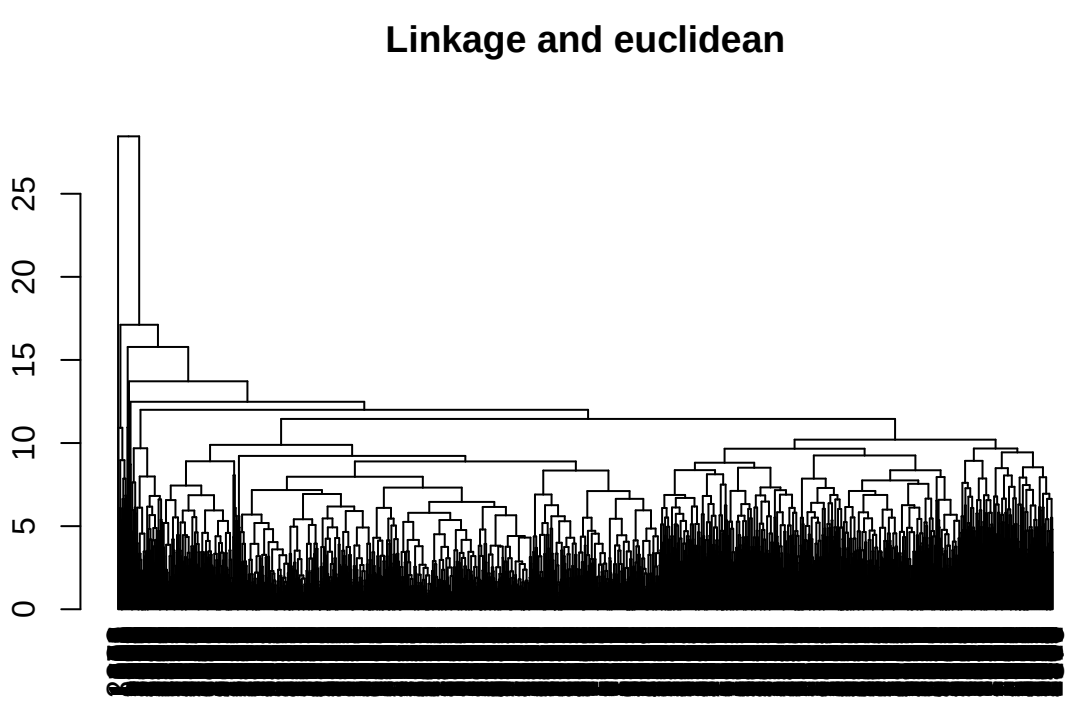
```
## 1       1  3.807074 59094.81 0.14951768 0.8987138 47.34084 473.29904 17.747588
## 2       2  3.354167 76052.16 0.07118056 0.2118056 50.08507 584.11632 72.699653
## 3       3  3.301682 34464.82 0.83283877 0.4302671 49.56874  42.85955  5.246291
##   MntMeatProducts MntFishProducts MntSweetProducts MntGoldProds NumWebPurchases
## 1       136.39871       25.805466        18.106109     55.10129        5.966238
## 2       447.63194      103.131944        75.050347     80.10417        5.434028
## 3        26.29377        7.426311         5.259149     16.26212        2.152324
##   NumStorePurchases    Complain      Age MembershipDays   Divorced   Married
## 1          7.389068 0.004823151 51.12058       380.6672 0.12861736 0.3938907
## 2          8.628472 0.008680556 44.86285       376.1997 0.09722222 0.3663194
## 3          3.216617 0.012858556 41.73788       328.2146 0.09495549 0.3966370
##      Single  Together
## 1 0.1382637 0.2877814
## 2 0.2552083 0.2430556
## 3 0.2354105 0.2512364
```

Cluster with Hierarchical Clustering

Q2.11 Perform clustering with Hierarchical method (Do you need to use scaling here?). Try complete, single and average linkage. Plot dendagram, based on it choose linkage and number of clusters, if possible, explain your choice. *(8 points)*

Answer: I believe that scaling is necessary. Dendrograms show that the complete linkage method produces the most distinct clusters, while the single linkage method produces the fewest. Clusters with average linkage are located in the middle. The full and average linkage methods both show a subtle elbow around three clusters, but the single linkage method does not.

```
set.seed(23)
dist_matrix <- dist(df_scale, method = "euclidean")
hie_comp <- hclust(dist_matrix, method = "complete")
hc <- as.dendrogram(hie_comp)

plot(hc, main = "Linkage and euclidean", cex = .9)
```



Additional grading criteria:

G3.1 Was all random methods properly seeded? *(2 points)* Yes, all random methods in the code provided were properly seeded.