```
set.seed(123)
```

USArrests Dataset and Hierarchical Clustering

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.

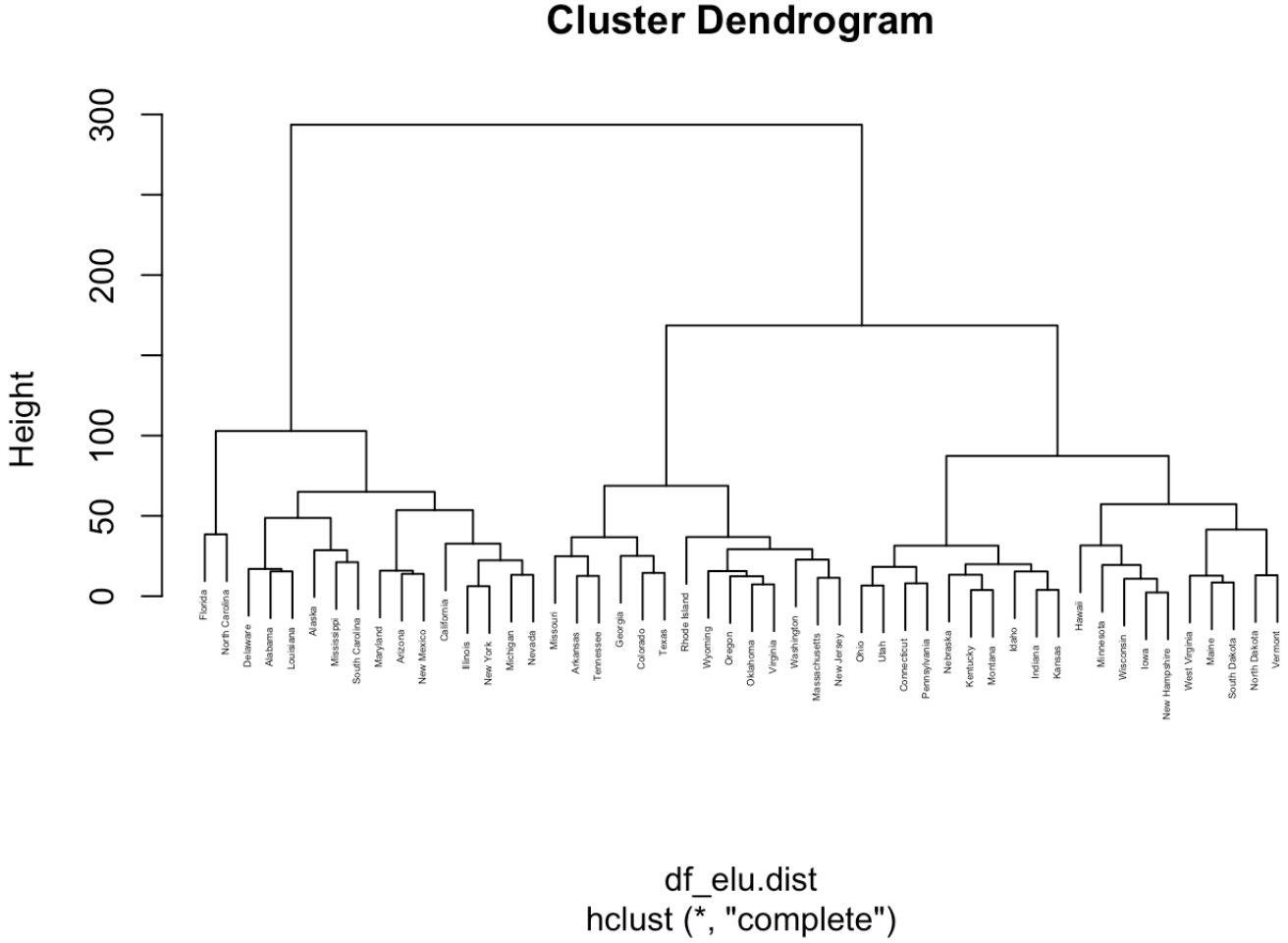
```
head(USArrests)
```

```
##            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
```

```
df = USArrests
```

Using hierarchical clustering with complete linkage and Euclidean distance, cluster the states.

```
df_elu.dist = dist(df,method = "euclidean")
df_elu.comp = hclust(df_elu.dist, method = "complete")
plot(df_elu.comp,cex = .3)
```

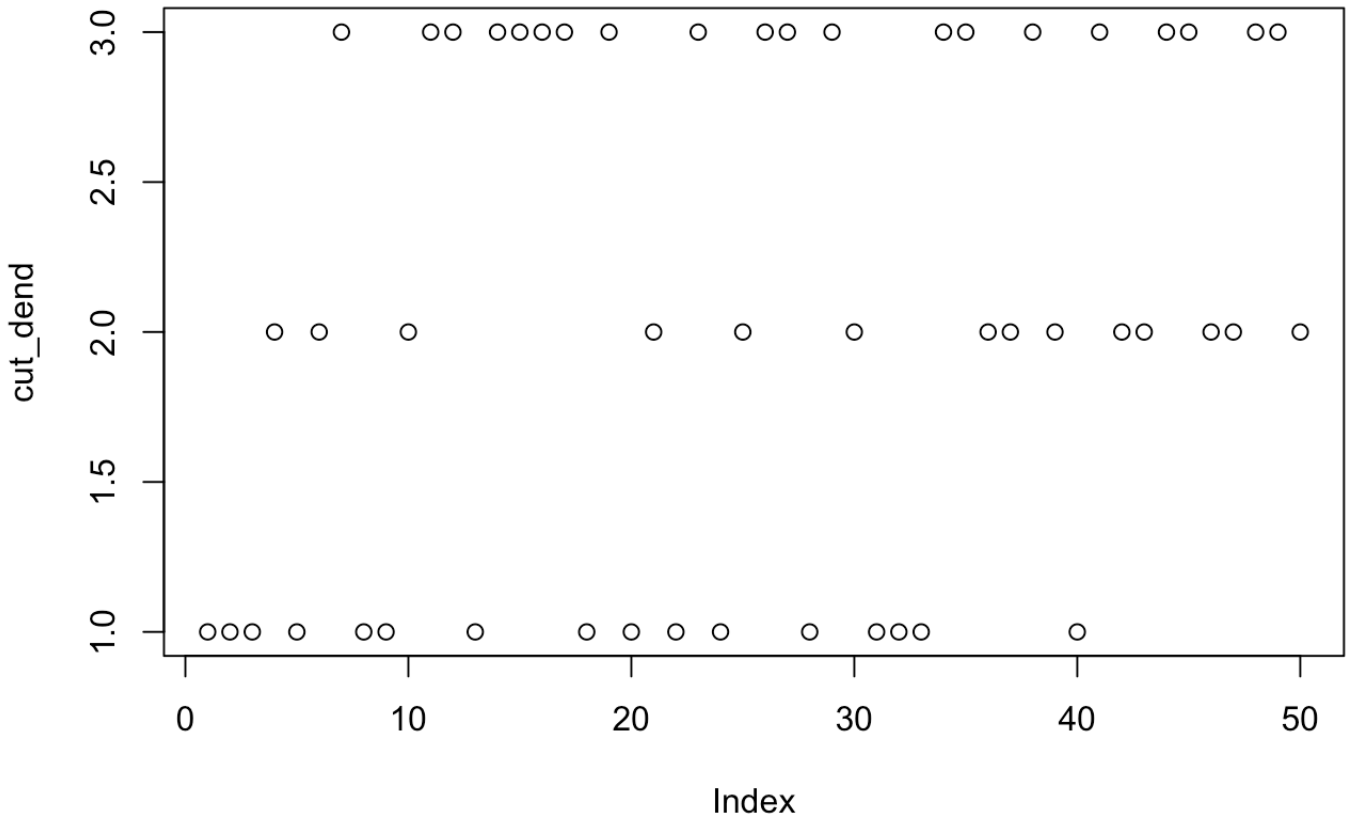


Cut the dendrogram at a height that results in three distinct clusters. Interpret the clusters

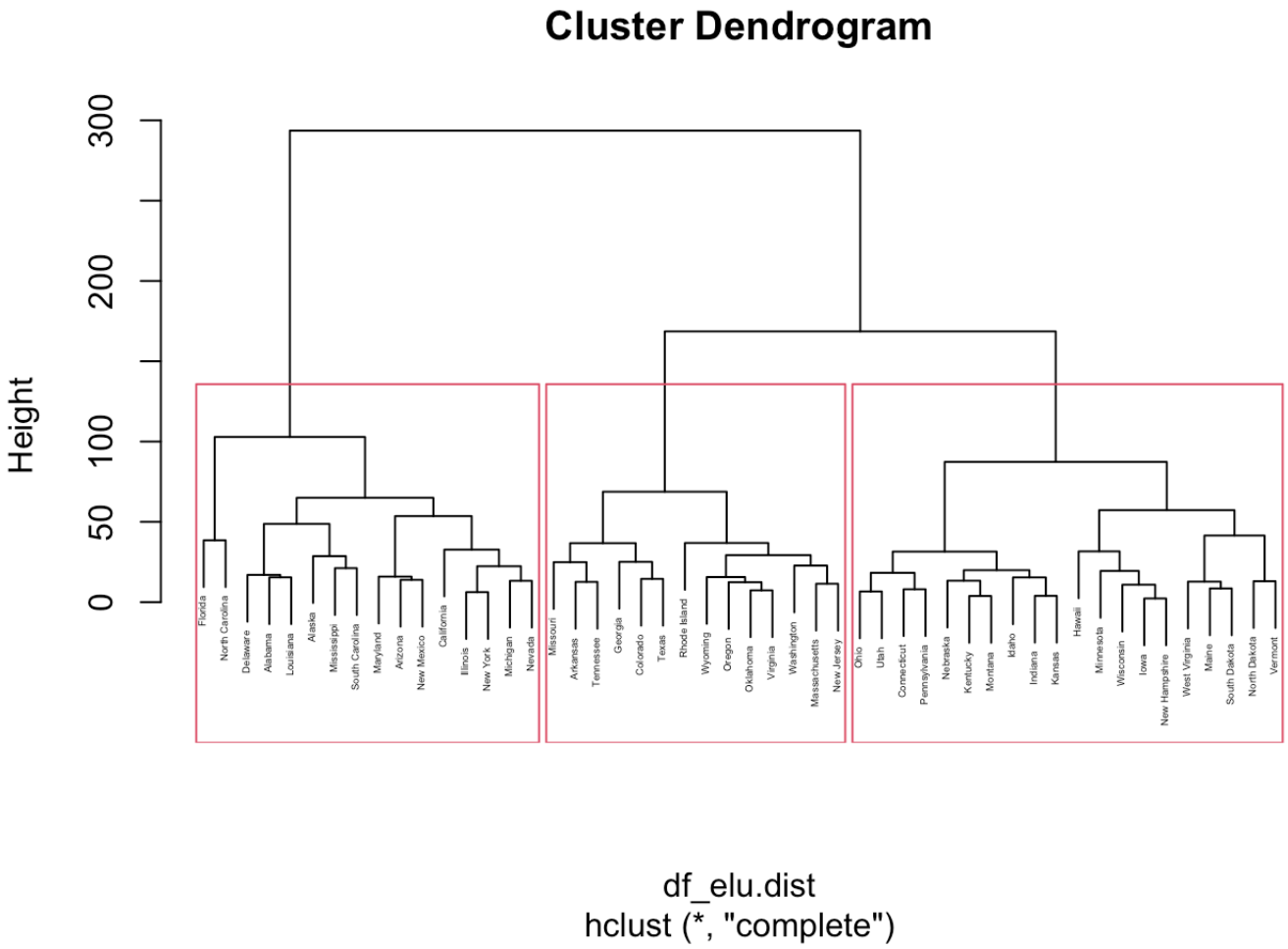
```
cut_dend = cutree(df_elu.comp,k = 3)
table(cut_dend)
```

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

```
plot(cut_dend)
```




```
plot(df_elu.comp,cex = .3)
rect.hclust(df_elu.comp, k = 3)
```




```
new_df <- data.frame(states =  rownames(USArrests),clusters = cut_dend)
new_df
```

```
##                       states clusters
## Alabama              Alabama        1
## Alaska                Alaska        1
## Arizona              Arizona        1
## Arkansas            Arkansas        2
## California         California        1
## Colorado             Colorado        2
## Connecticut       Connecticut        3
## Delaware             Delaware        1
## Florida               Florida        1
## Georgia               Georgia        2
## Hawaii                 Hawaii        3
## Idaho                   Idaho        3
## Illinois             Illinois        1
## Indiana               Indiana        3
## Iowa                     Iowa        3
```

```
## Kansas                 Kansas        3
## Kentucky               Kentucky      3
## Louisiana             Louisiana      1
## Maine                     Maine      3
## Maryland               Maryland      1
## Massachusetts    Massachusetts       2
## Michigan               Michigan      1
## Minnesota             Minnesota      3
## Mississippi         Mississippi      1
## Missouri               Missouri      2
## Montana                 Montana      3
## Nebraska               Nebraska      3
## Nevada                   Nevada      1
## New Hampshire    New Hampshire       3
## New Jersey           New Jersey      2
## New Mexico           New Mexico      1
## New York               New York      1
## North Carolina North Carolina        1
## North Dakota      North Dakota       3
## Ohio                       Ohio      3
## Oklahoma               Oklahoma      2
## Oregon                   Oregon      2
## Pennsylvania      Pennsylvania       3
## Rhode Island      Rhode Island       2
## South Carolina South Carolina        1
## South Dakota      South Dakota       3
## Tennessee             Tennessee      2
## Texas                     Texas      2
## Utah                       Utah      3
## Vermont                 Vermont      3
## Virginia               Virginia      2
## Washington           Washington      2
## West Virginia    West Virginia       3
## Wisconsin             Wisconsin      3
## Wyoming                 Wyoming      2
```

```
#NAMES OF ALL STATES SEPERATED BY EACH CLUSTERS.
group1 <- rownames(new_df[cut_dend == 1, ])
group2 <- rownames(new_df[cut_dend == 2, ])
group3 <- rownames(new_df[cut_dend == 3, ])

print(paste("Cluster 1:", paste(group1, collapse = ", ")))
```

```
## [1] "Cluster 1: Alabama, Alaska, Arizona, California, Delaware, Florida, Illinois,
Louisiana, Maryland, Michigan, Mississippi, Nevada, New Mexico, New York, North Carol
ina, South Carolina"
```

```
print(paste("Cluster 2:", paste(group2, collapse = ", ")))
```

```
## [1] "Cluster 2: Arkansas, Colorado, Georgia, Massachusetts, Missouri, New Jersey,
Oklahoma, Oregon, Rhode Island, Tennessee, Texas, Virginia, Washington, Wyoming"
```

```
print(paste("Cluster 3:", paste(group3, collapse = ", ")))
```

```
## [1] "Cluster 3: Connecticut, Hawaii, Idaho, Indiana, Iowa, Kansas, Kentucky, Main
e, Minnesota, Montana, Nebraska, New Hampshire, North Dakota, Ohio, Pennsylvania, Sou
th Dakota, Utah, Vermont, West Virginia, Wisconsin"
```

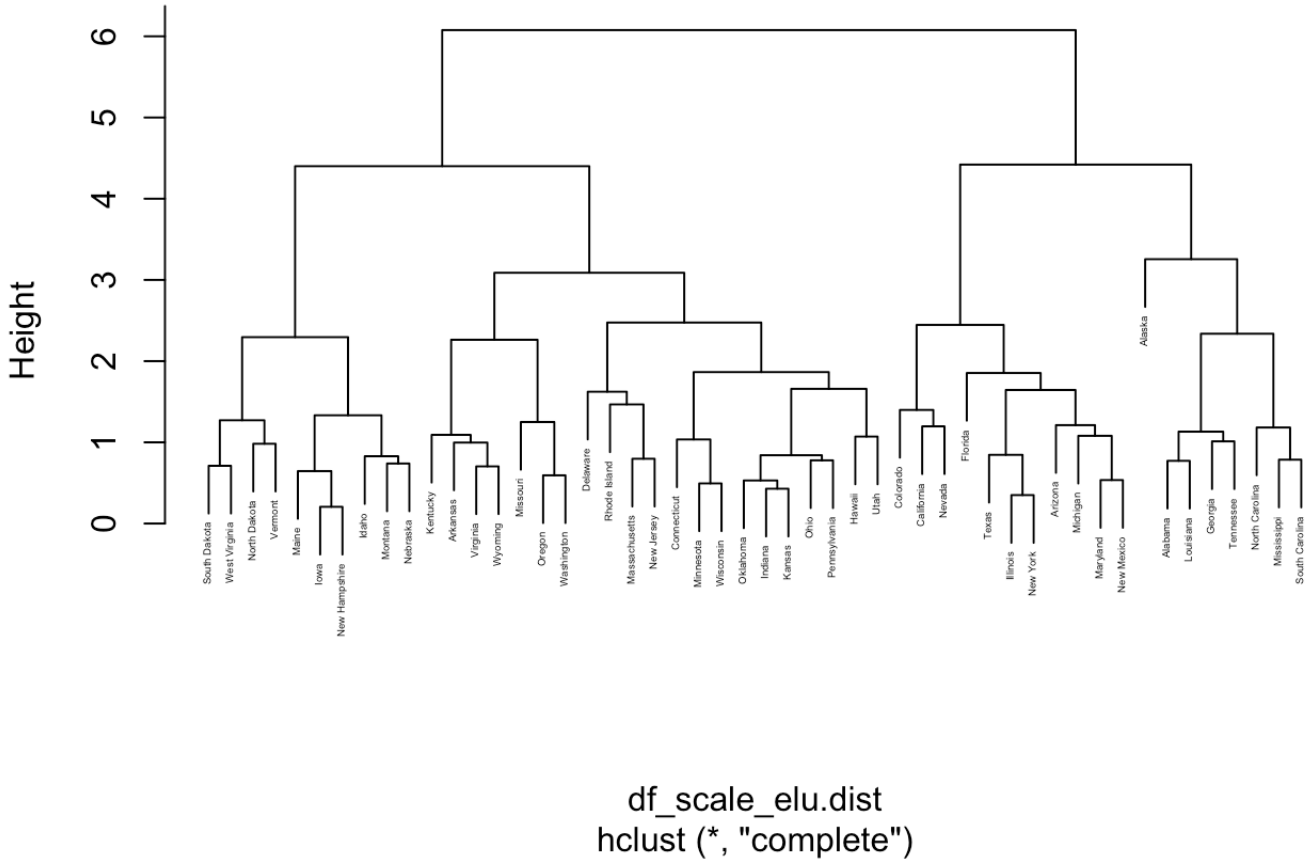
```
group_mean <- aggregate(df, by=list(cut_dend), FUN=mean)
colnames(group_mean) <- c("Cluster", "Assault", "UrbanPop", "Murder", "Rape")
group_mean
```

```
##   Cluster   Assault UrbanPop   Murder     Rape
## 1       1 11.812500 272.5625 68.31250 28.37500
## 2       2  8.214286 173.2857 70.64286 22.84286
## 3       3  4.270000  87.5500 59.75000 14.39000
```

Interpretation: The above average gives us an understanding of Cluster 1 contains states with high crime rates and high urban populations, Cluster 2 includes states with moderate crime rates and urban populations, and Cluster 3 includes states with low crime rates and low urban populations.

Also, we can see the clusters are almost equally disturbed the data of 16,14 and 20

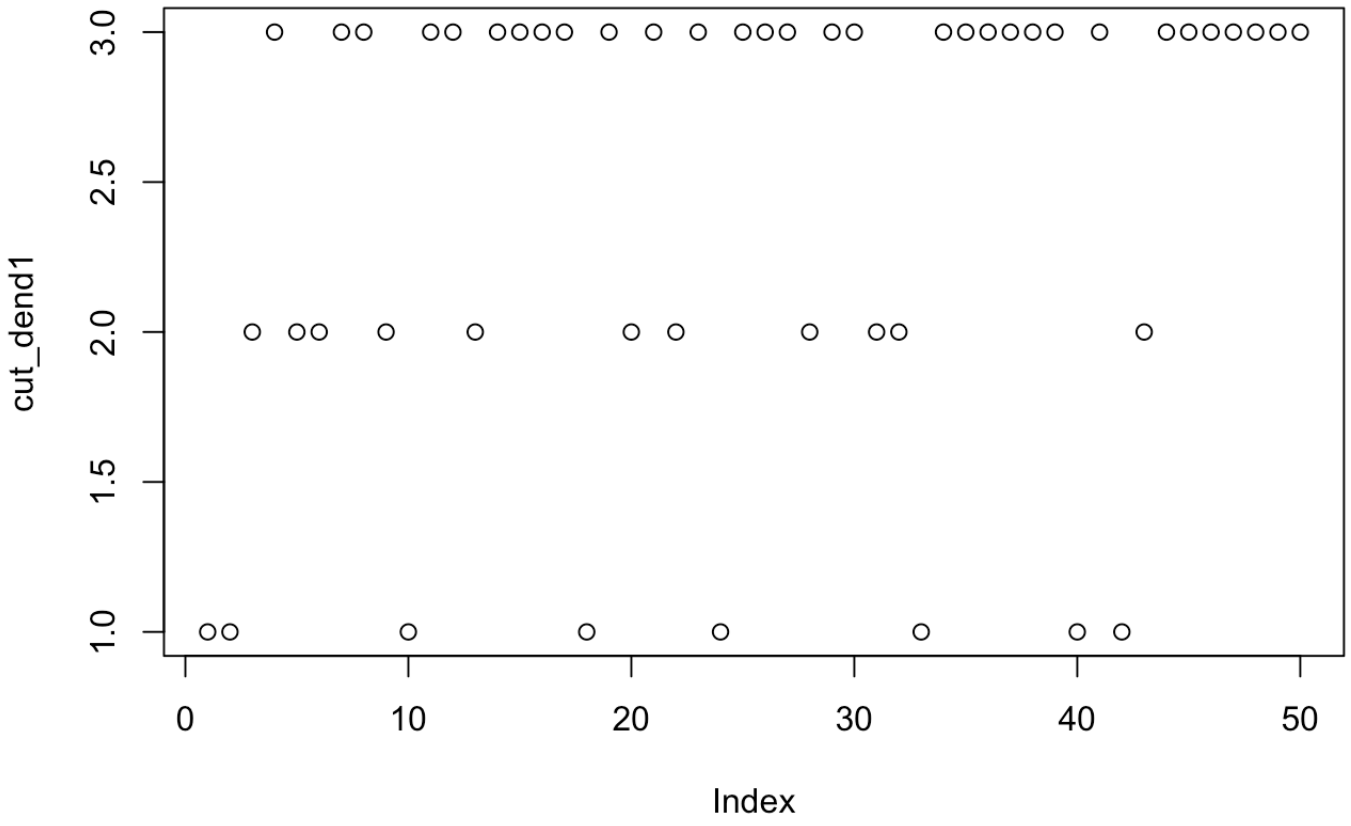
```
df_scale_elu.dist = dist(scale(df),method = "euclidean")
df_scale_elu.comp = hclust(df_scale_elu.dist, method = "complete")
plot(df_scale_elu.comp,cex = .3)
```



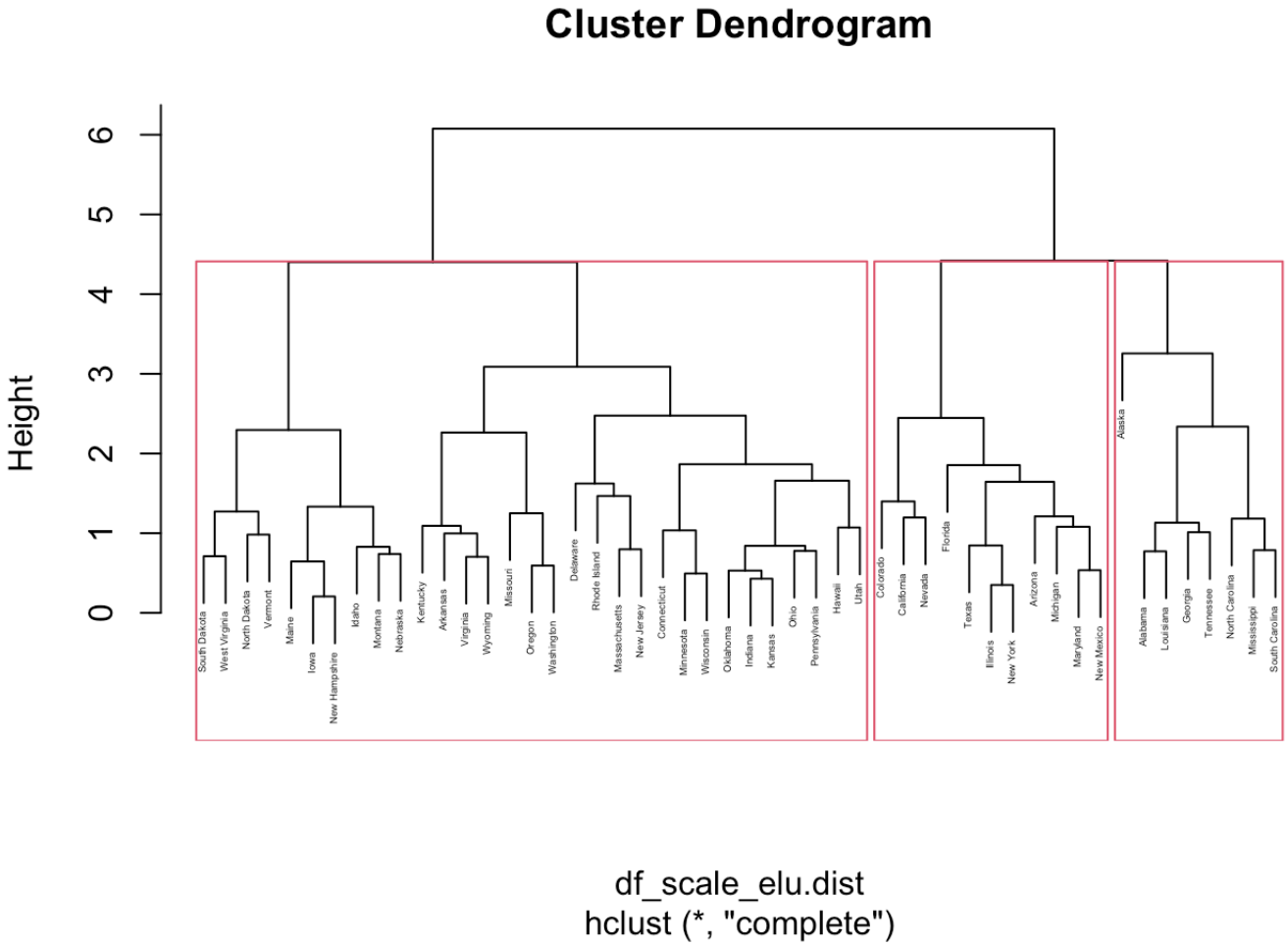

```
cut_dend1= cutree(df_scale_elu.comp,k = 3)
table(cut_dend1)
```

```
## cut_dend1
##  1  2  3
##  8 11 31
```

```
plot(cut_dend1)
```




```
new_df1 <- data.frame(states =  rownames(USArrests),clusters = cut_dend1)
plot(df_scale_elu.comp,cex = .3)
rect.hclust(df_scale_elu.comp, k = 3)
```




```
new_df1
```

```
##                    states clusters
## Alabama             Alabama        1
## Alaska               Alaska        1
## Arizona             Arizona        2
## Arkansas           Arkansas        3
## California       California        2
## Colorado           Colorado        2
## Connecticut     Connecticut        3
## Delaware           Delaware        3
## Florida             Florida        2
## Georgia             Georgia        1
## Hawaii               Hawaii        3
## Idaho                 Idaho        3
## Illinois           Illinois        2
## Indiana             Indiana        3
## Iowa                   Iowa        3
## Kansas               Kansas        3
```

```
## Kentucky              Kentucky        3
## Louisiana            Louisiana        1
## Maine                    Maine        3
## Maryland              Maryland        2
## Massachusetts   Massachusetts        3
## Michigan              Michigan        2
## Minnesota            Minnesota        3
## Mississippi       Mississippi        1
## Missouri              Missouri        3
## Montana                Montana        3
## Nebraska              Nebraska        3
## Nevada                  Nevada        2
## New Hampshire   New Hampshire        3
## New Jersey          New Jersey        3
## New Mexico          New Mexico        2
## New York              New York        2
## North Carolina North Carolina        1
## North Dakota     North Dakota        3
## Ohio                     Ohio        3
## Oklahoma              Oklahoma        3
## Oregon                  Oregon        3
## Pennsylvania     Pennsylvania        3
## Rhode Island     Rhode Island        3
## South Carolina South Carolina        1
## South Dakota     South Dakota        3
## Tennessee            Tennessee        1
## Texas                    Texas        2
## Utah                      Utah        3
## Vermont                Vermont        3
## Virginia              Virginia        3
## Washington          Washington        3
## West Virginia   West Virginia        3
## Wisconsin            Wisconsin        3
## Wyoming                Wyoming        3
```

```
#NAMES OF ALL STATES SEPERATED BY EACH CLUSTERS.
group1 <- rownames(new_df1[cut_dend1 == 1, ])
group2 <- rownames(new_df1[cut_dend1 == 2, ])
group3 <- rownames(new_df1[cut_dend1 == 3, ])

print(paste("Cluster 1:", paste(group1, collapse = ", ")))
```

```
## [1] "Cluster 1: Alabama, Alaska, Georgia, Louisiana, Mississippi, North Carolina,
South Carolina, Tennessee"
```

```
print(paste("Cluster 2:", paste(group2, collapse = ", ")))
```

```
## [1] "Cluster 2: Arizona, California, Colorado, Florida, Illinois, Maryland, Michig
an, Nevada, New Mexico, New York, Texas"
```

```
print(paste("Cluster 3:", paste(group3, collapse = ", ")))
```

```
## [1] "Cluster 3: Arkansas, Connecticut, Delaware, Hawaii, Idaho, Indiana, Iowa, Kan
sas, Kentucky, Maine, Massachusetts, Minnesota, Missouri, Montana, Nebraska, New Hamp
shire, New Jersey, North Dakota, Ohio, Oklahoma, Oregon, Pennsylvania, Rhode Island,
South Dakota, Utah, Vermont, Virginia, Washington, West Virginia, Wisconsin, Wyoming"
```

Murder numeric Murder arrests (per 100,000) Assault numeric Assault arrests (per 100,000) UrbanPop numeric Percent urban population Rape numeric Rape arrests (per 100,000)

The above tells us the feature unit, where murder,Rape,Assault are per 100,000 where as UrbanPop numeric Percent.As a result, it is critical to scale so that the ‘UrbanPop’ equally contribute to the hierarchical clustering process with other variables.

Scaling the variables has an effect on the produced clusters like branch lengths and tree height.The un-scaled tree stands 300 feet tall, whereas the scaled tree stands six feet tall. We cut the tree without scaling at a height around 140, whereas we cut the scaled tree at a height around 4 to generate 3 clusters.

The clusters before clustering as a consequence where clusters were almost similar, with each cluster including all states. This result reveals that variable scales, rather than underlying data connections, dominating the clustering process. Therefore, scaling is important for clustering where it represents the real relationships in the data and produces more interpretable clusters

Market Segmentation

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

```
#library(data.table)
df <- data.table::fread("m_marketing_campaign.csv")
head(df)
```

```
##      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
## 1:           88               8                 4        0
## 2:            6               1                 2        0
## 3:           42               8                10        0
## 4:            5               2                 4        0
## 5:           15               5                 6        0
## 6:           14               6                10        0
```

```
Today.date = as.Date("2014-07-01")
df$Age = 2014 - df$Year_Birth
```

```
df$Dt_Customer = as.Date(df$Dt_Customer, format = "%d-%m-%Y")
df$MembershipDays = difftime(Today.date,df$Dt_Customer)
```

```
Summarize_edu = table(df$Education)
print(Summarize_edu)
```

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

```
df$EducationLevel = recode(df$Education, HighSchool=13, Associate=15, Bachelor=17, Ma
ster=19,PhD=22)
```

```
Summarize_marital.status = table(df$Marital_Status)
print(Summarize_marital.status)
```

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

```
df = fastDummies::dummy_cols(df, select_columns ="Marital_Status")
```

```
df_sel <- subset(df, select = -c(ID, Year_Birth, Dt_Customer, Education, Marital_Stat
us))
```

```
df_sel$MembershipDays = as.numeric(df_sel$MembershipDays)
df_scale = data.frame(scale(df_sel))
```

```
head(df_scale)
```

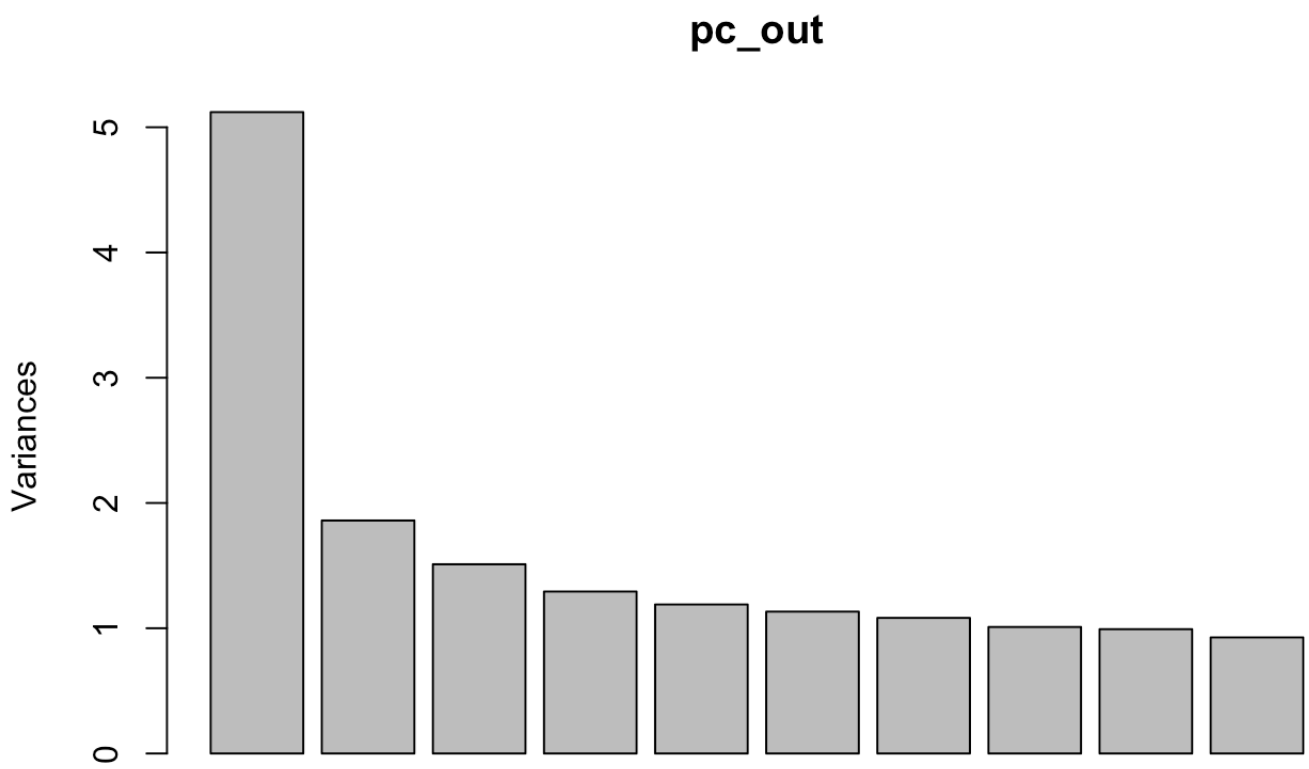
```
##       Income    Kidhome   Teenhome    Recency   MntWines  MntFruits
## 1  0.2339039 -0.8227362 -0.9281454  0.3082732  0.9766566  1.5488659
## 2 -0.2341403  1.0393789  0.9090170 -0.3826166 -0.8711997 -0.6370558
## 3  0.7686585 -0.8227362 -0.9281454 -0.7971505  0.3577432  0.5689700
## 4 -1.0158542  1.0393789 -0.9281454 -0.7971505 -0.8711997 -0.5616792
## 5  0.2400551  1.0393789 -0.9281454  1.5518749 -0.3914678  0.4182168
## 6  0.4075255 -0.8227362  0.9090170 -1.1425954  0.6361062  0.3930912
##   MntMeatProducts MntFishProducts MntSweetProducts MntGoldProds NumWebPurchases
## 1       1.6879549       2.4630607      1.481888649   0.85482704       1.4304941
## 2      -0.7180699      -0.6514171     -0.634215838  -0.73267383      -1.1252228
## 3      -0.1789421       1.3455128     -0.147755036  -0.03572223       1.4304941
## 4      -0.6556915      -0.5048534     -0.585569758  -0.75203360      -0.7601204
## 5      -0.2190425       0.1546831     -0.001816796  -0.55843593       0.3351868
## 6      -0.3081546      -0.6880580      0.363028806  -0.57779570       0.7002892
##   NumStorePurchases    Complain        Age MembershipDays EducationLevel
## 1        -0.5538715 -0.09794622  0.9853629      1.5300508     -0.4807422
## 2        -1.1683880 -0.09794622  1.2357656     -1.1889072     -0.4807422
## 3         1.2896778 -0.09794622  0.3176225     -0.2051387     -0.4807422
## 4        -0.5538715 -0.09794622 -1.2682609     -1.0603746     -0.4807422
## 5         0.0606449 -0.09794622 -1.0178582     -0.9516163      1.6431770
## 6         1.2896778 -0.09794622  0.1506875     -0.2990664      0.3688254
##   Marital_Status_Divorced Marital_Status_Married Marital_Status_Single
## 1              -0.3424856             -0.7959829             1.9205079
## 2              -0.3424856             -0.7959829             1.9205079
## 3              -0.3424856             -0.7959829            -0.5204599
## 4              -0.3424856             -0.7959829            -0.5204599
## 5              -0.3424856              1.2557397            -0.5204599
## 6              -0.3424856             -0.7959829            -0.5204599
##   Marital_Status_Together Marital_Status_Widow
## 1              -0.5916806           -0.1887179
## 2              -0.5916806           -0.1887179
## 3               1.6893359           -0.1887179
## 4               1.6893359           -0.1887179
## 5              -0.5916806           -0.1887179
## 6               1.6893359           -0.1887179
```

PCA

```
pc_out = prcomp(df_scale)
summary(pc_out)
```

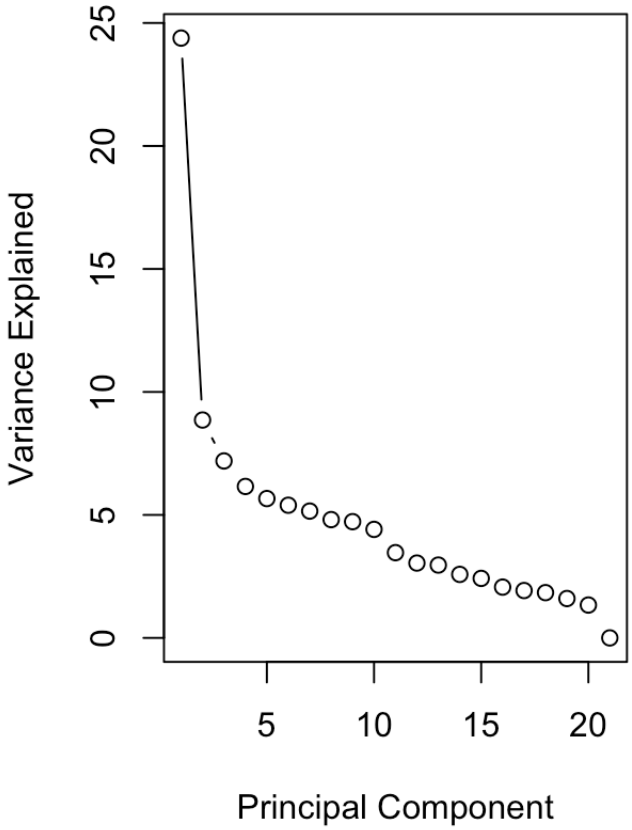
```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.2630 1.36390 1.22917 1.13715 1.09074 1.06431 1.04053
## Proportion of Variance 0.2439 0.08858 0.07195 0.06158 0.05665 0.05394 0.05156
## Cumulative Proportion  0.2439 0.33244 0.40438 0.46596 0.52261 0.57655 0.62811
##                           PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     1.0050 0.99623 0.96259 0.85315 0.79939 0.78861 0.73634
## Proportion of Variance 0.0481 0.04726 0.04412 0.03466 0.03043 0.02961 0.02582
## Cumulative Proportion  0.6762 0.72347 0.76759 0.80225 0.83268 0.86230 0.88812
##                          PC15    PC16    PC17   PC18    PC19    PC20       PC21
## Standard deviation     0.7128 0.65860 0.63594 0.6217 0.57981 0.52972 1.411e-15
## Proportion of Variance 0.0242 0.02065 0.01926 0.0184 0.01601 0.01336 0.000e+00
## Cumulative Proportion  0.9123 0.93297 0.95222 0.9706 0.98664 1.00000 1.000e+00
```

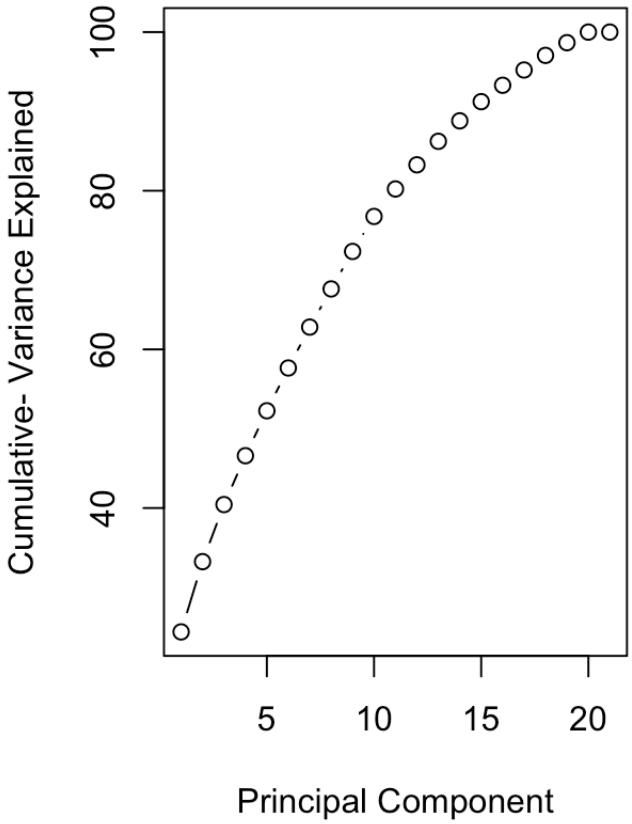
```
plot(pc_out)
```



scree plot

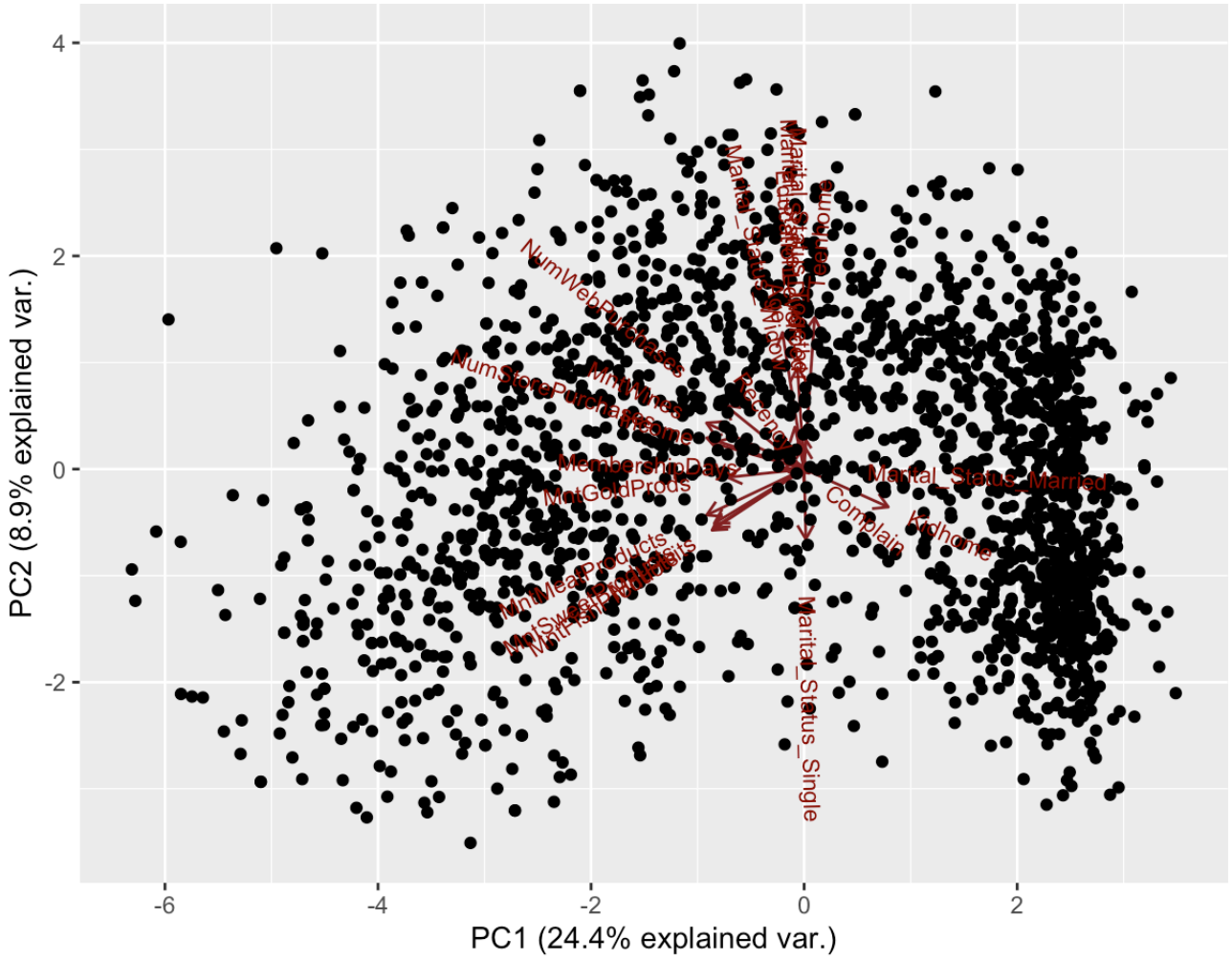
```
variance = pc_out$sdev^2
pve = 100 * variance / sum(variance)
par(mfrow = c(1, 2))
plot(pve, xlab = "Principal Component",
    ylab = "Variance Explained",
    type = "b")
plot(cumsum(pve), xlab = "Principal Component",
    ylab = "Cumulative- Variance Explained",
     type = "b")
```



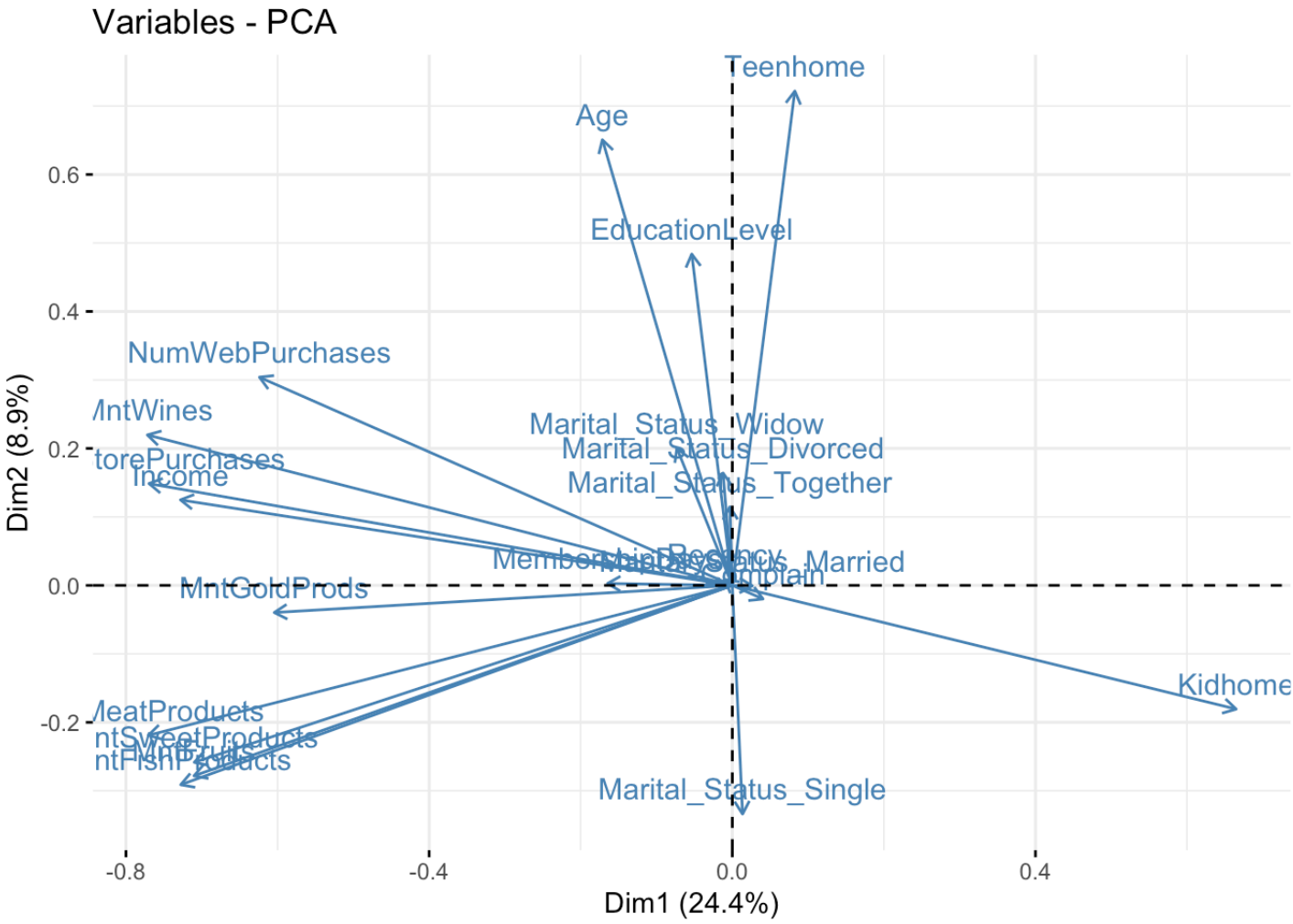


biplot

```
ggbiplot(pc_out, scale = 0, labels=rownames(pc_out$x))
```




```
fviz_pca_var(pc_out, col.var="steelblue")
```



I'm able to see the 2 clusters based on the density of the data points. where the points at right side of plot are more dense and near, which differences from the other cluster. Though, there is not clear distinction between clusters. Clusters may be poorly defined as they are densely packed and have overlap between points. Also elliptical shape of principal component explains the variance in data points of PC1 and PC2 is due to clusters present.The further analysis helps us to clearly explain the clusters available in the data set.

About PCA:

PC1 explains of 24.1 % of variance and PC2 - 8.9% and total it explains around 33% of total variance. for cumulative PC14 explains variance upto 90% depending on the needs of percentage of the variance needed we can chose the principal component. We can see features like age, education level martial status is more explained and co-related to PC2 and purchases,mntgoldprods,wines are more towards PC1

Selecting Number of Clusters

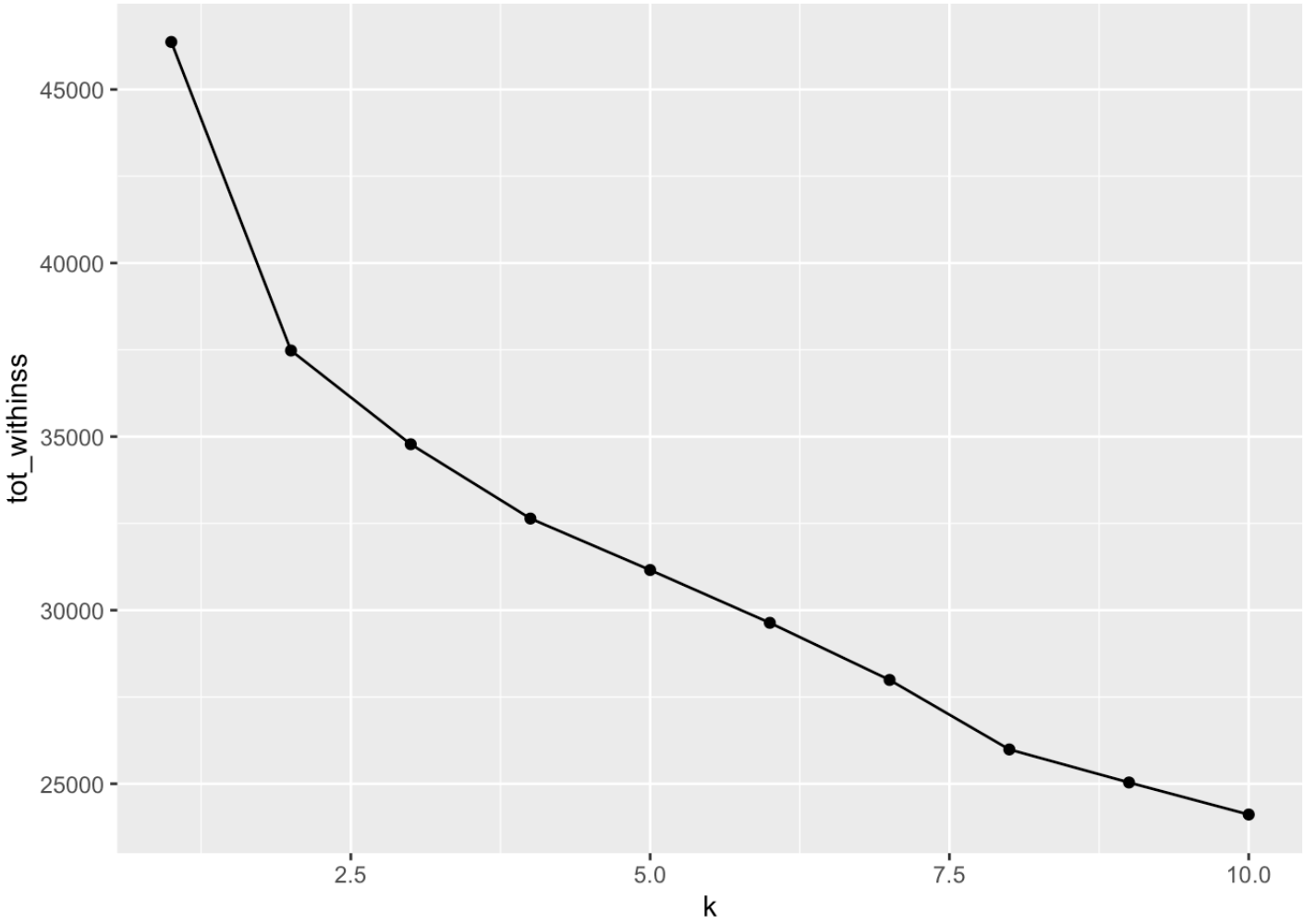
```
set.seed(123)
km_out_list <- lapply(1:10, function(k) list(
  k=k,
  km_out=kmeans(df_scale, k, nstart = 50)))
```

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

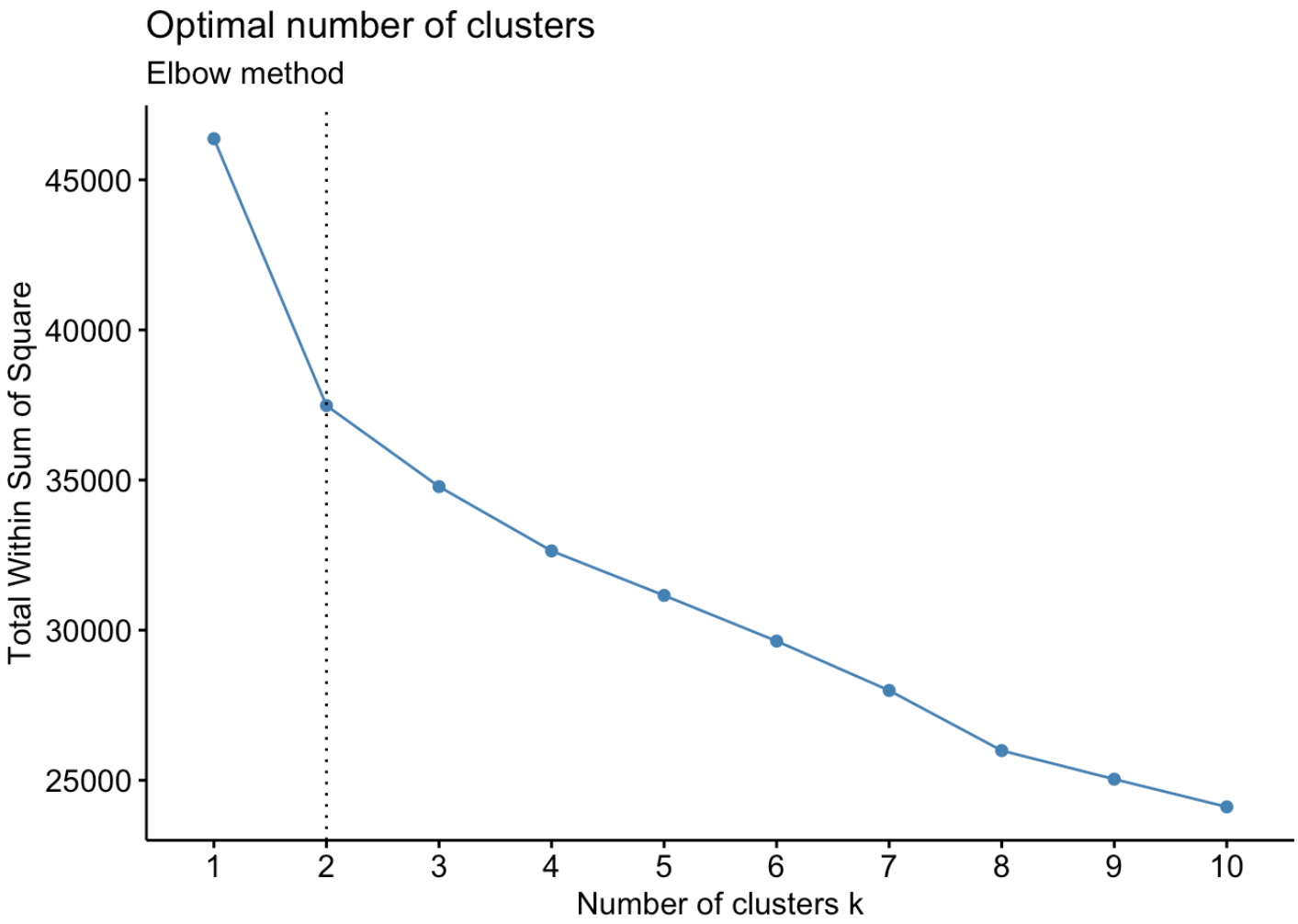
```
km_results <- data.frame(
  k=sapply(km_out_list, function(k) k$k),
  totss=sapply(km_out_list, function(k) k$km_out$totss),
  tot_withinss=sapply(km_out_list, function(k) k$km_out$tot.withinss)
  )
km_results
```

```
##     k totss tot_withinss
## 1   1 46368     46368.00
## 2   2 46368     37479.89
## 3   3 46368     34780.65
## 4   4 46368     32639.80
## 5   5 46368     31157.25
## 6   6 46368     29635.72
## 7   7 46368     27990.97
## 8   8 46368     25988.88
## 9   9 46368     25036.74
## 10 10 46368     24113.68
```

```
ggplot(km_results,aes(x=k,y=tot_withinss))+geom_line()+geom_point()
```

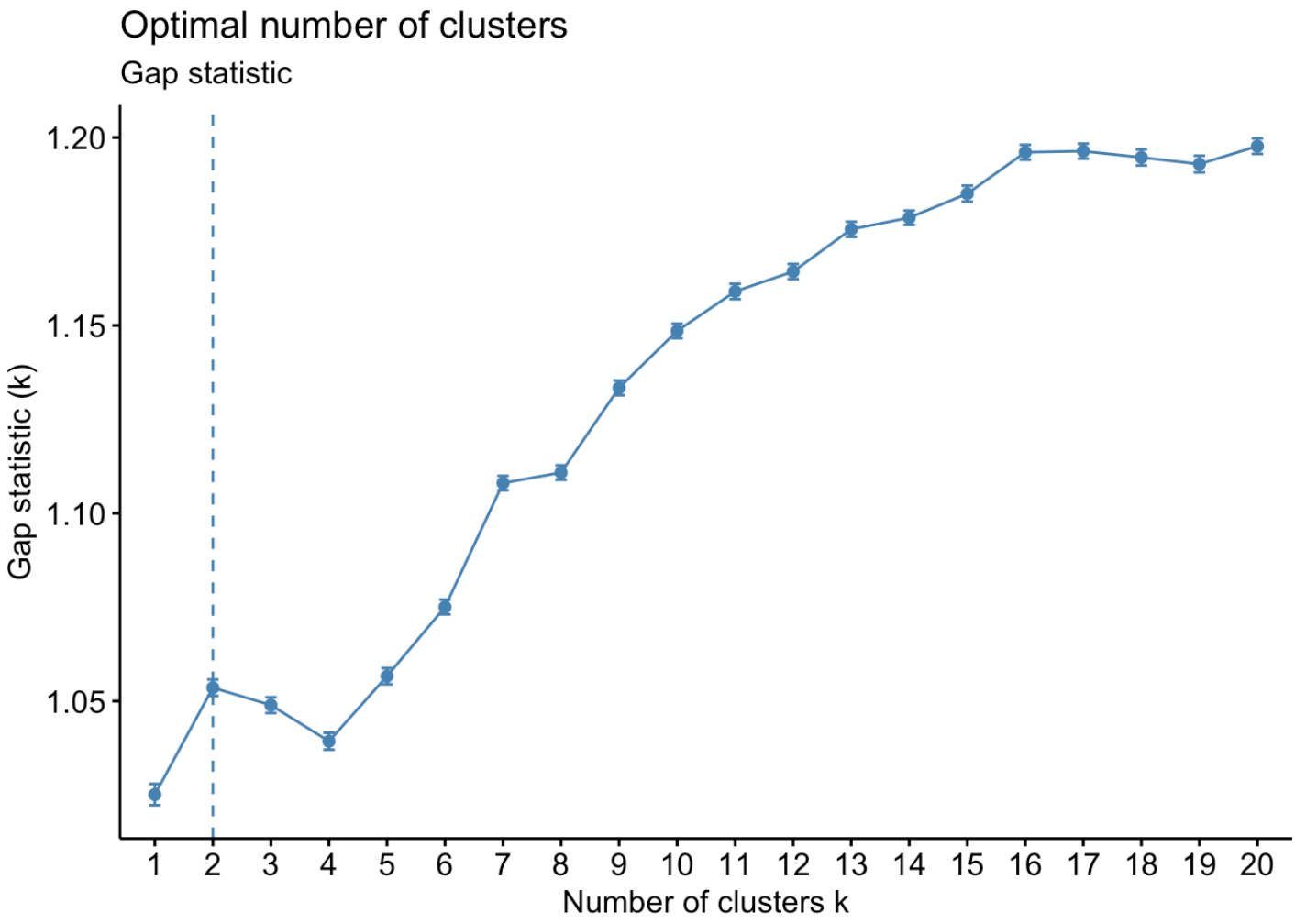



```
set.seed(1)
fviz_nbclust(df_scale, kmeans, method = "wss",k.max=10, nstart=50, iter.max=21) +
  geom_vline(xintercept = 2, linetype = 3)+
  labs(subtitle = "Elbow method")
```



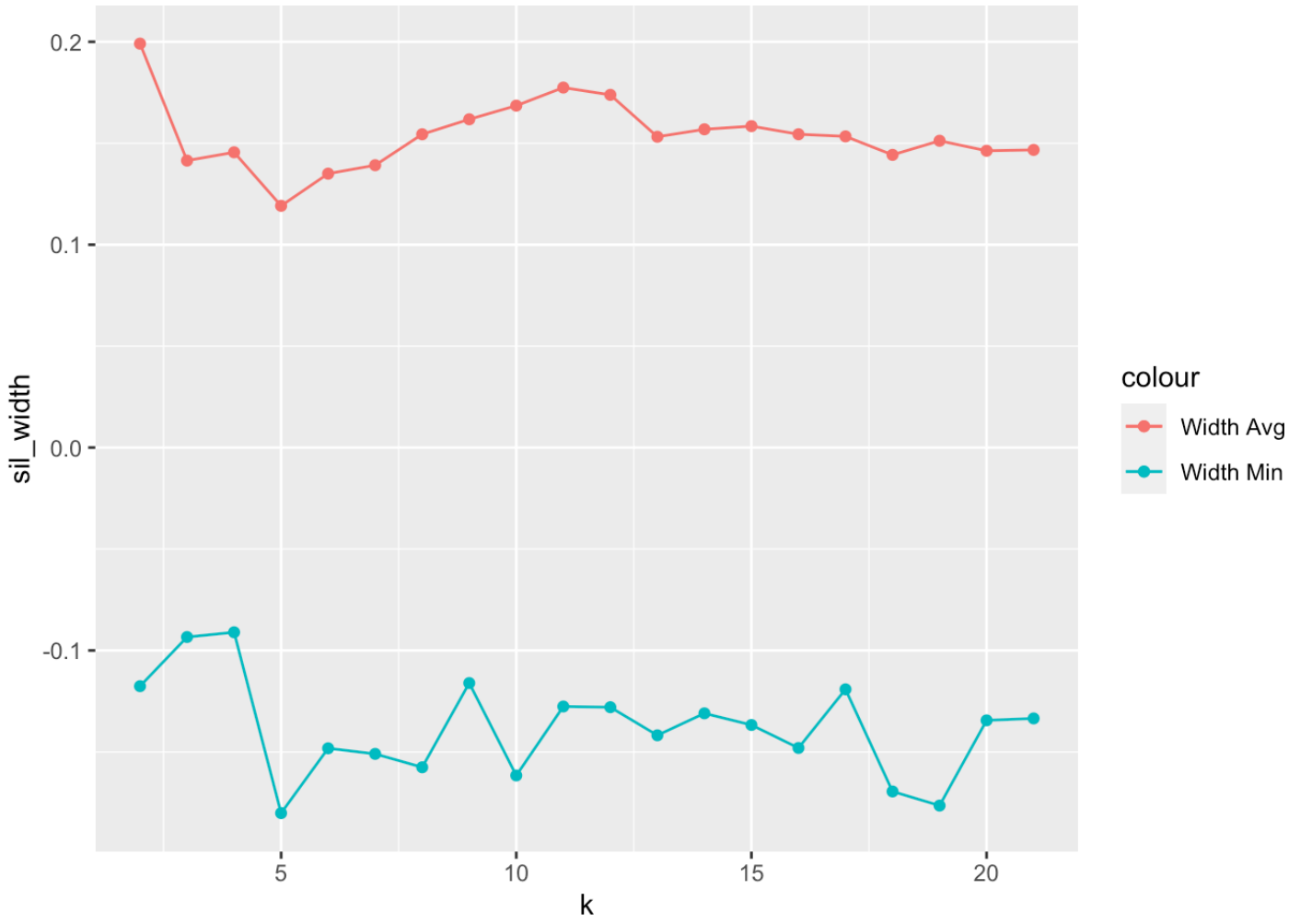
Optimal number of clusters using elbow method will be 2,8 as shown in the graph. 2 will be a better choice accroding to elbow method for our objective

```
set.seed(1)
fviz_nbclust(df_scale, kmeans, method = "gap_stat", nboot = 20,k.max=20, nstart=20, i
ter.max=40) +
  labs(subtitle = "Gap statistic")
```

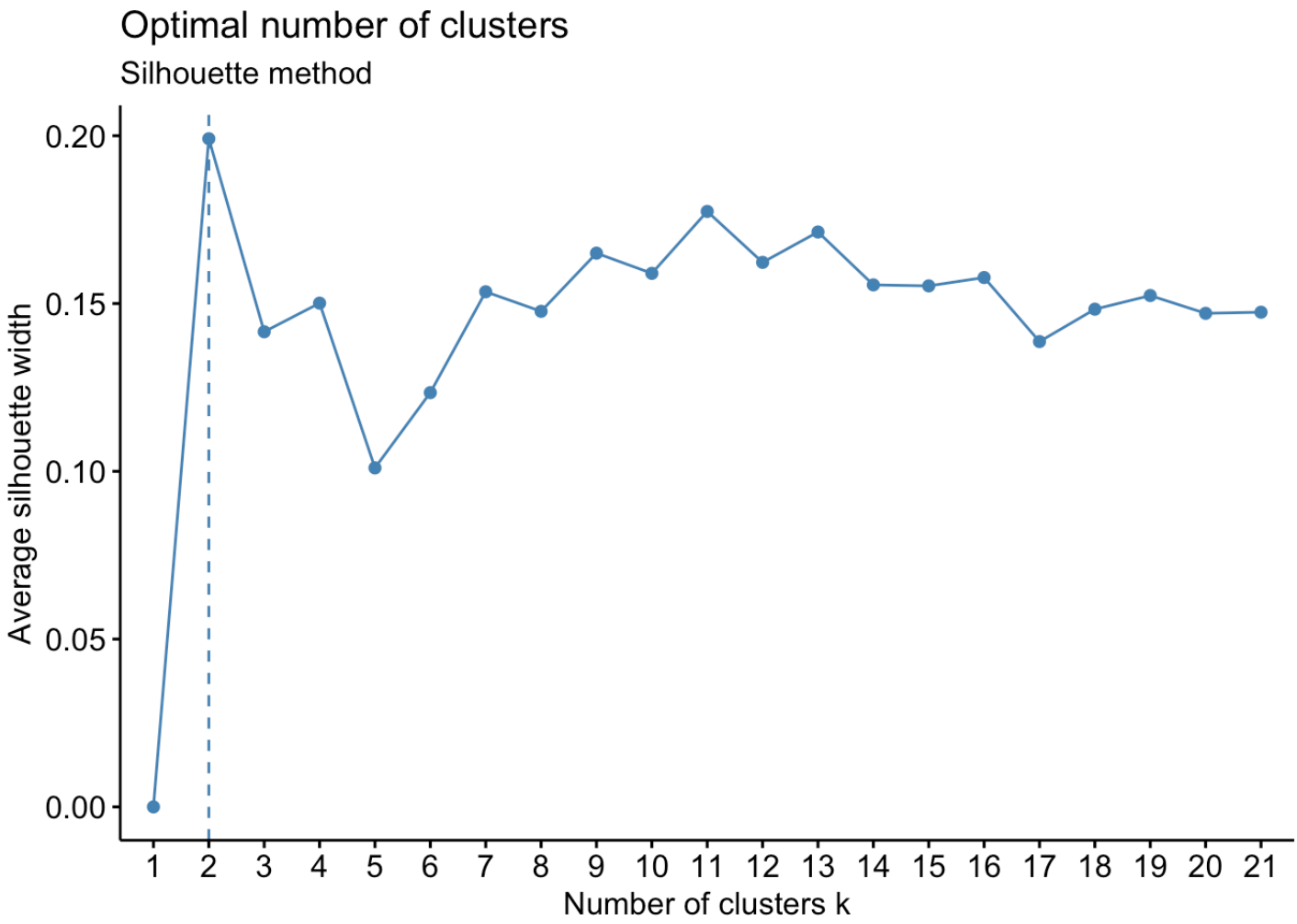


The choice of 2,16,20 or more will be a good choice 2 will be a better choice according to gap statistic for our objective

```
set.seed(1)
results <- lapply(2:21, function(k) {
  kmeans_cluster <- kmeans(df_scale, k, nstart=21, iter.max=21)
  si <- silhouette(kmeans_cluster$cluster, dist = dist(df_scale))
  data.frame(k=k,sil_width=mean(si[,'sil_width']),sil_width_min=min(si[,'sil_widt
h']))
})
si_df <- bind_rows(results)

ggplot(si_df, aes(x=k,y=sil_width,color="Width Avg"))+geom_point()+geom_line()+
  geom_point(aes(y=sil_width_min,color="Width Min"))+geom_line(aes(y=sil_width_min,co
lor="Width Min"))
```




```
set.seed(1)
fviz_nbclust(df_scale, kmeans, method = "silhouette", nboot = 21,k.max=21, nstart=21,
iter.max=40)+
  labs(subtitle = "Silhouette method")
```



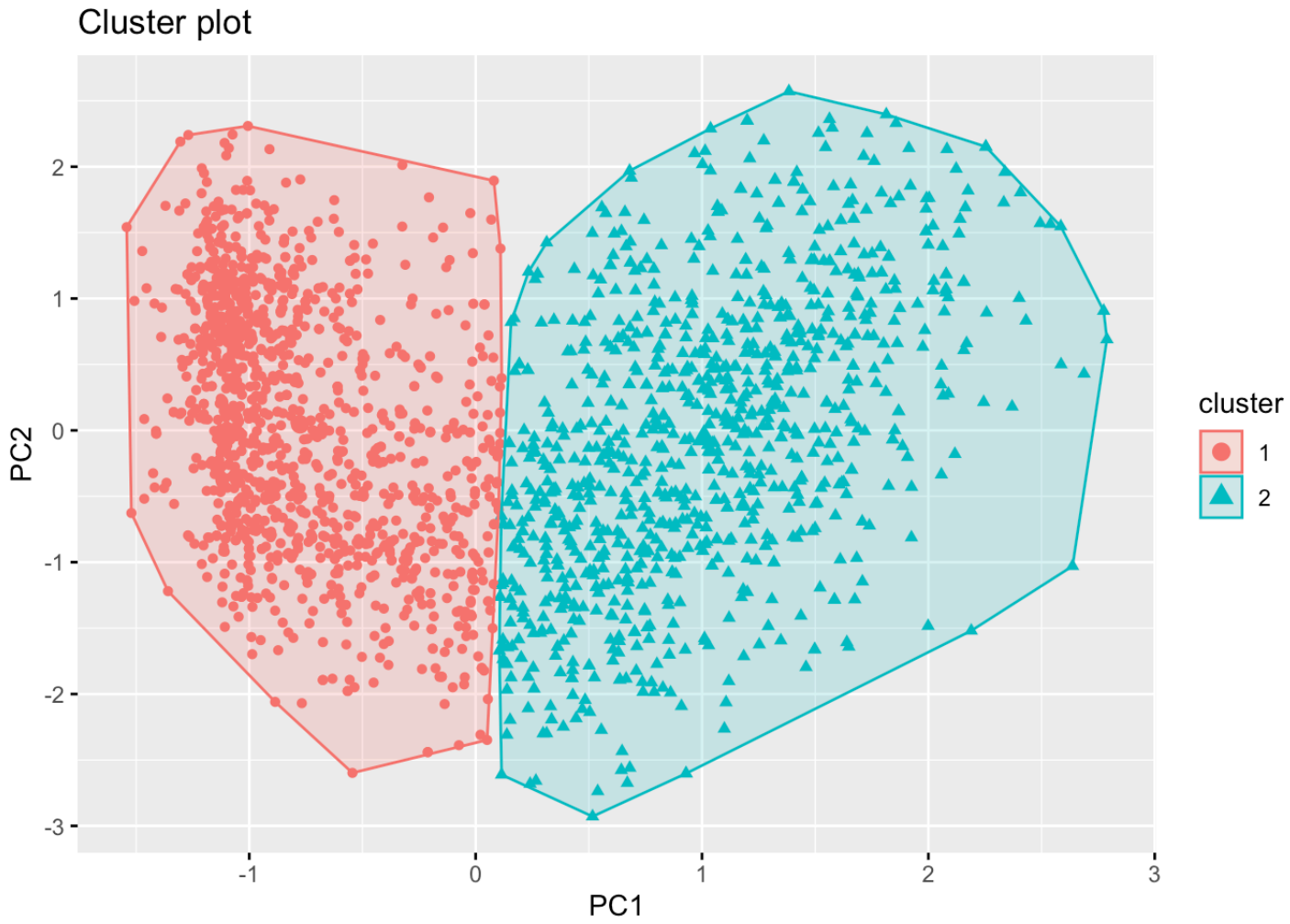
The choice of 2, 4, 7, 9, 11,13,19 and more will be a good option according Silhouette method optimal number of clusters using Silhouette method is 2 for our objective.

The elbow technique, gap statistics, and silhouette are all showing that two clusters are the best fit for this data.This implies that dividing the store's consumers into two unique groups based on the data set attributes and characteristics will be better for market segmentation

Though choosing 2 cluster over 1 might Over segment or unneeded complexity in market segmentation.Also, reducing segmentation into a single group may be a more practical solution but clustering by customer will help us in targeted Advertising which will improve our campaign. If we segment all in one campaign it won't help us in better understanding of customer and to create distinct marketing campaigns. The data is clustered based on customer features like purchase, age, income etc. Which will help in knowing customers better and take individual decision.

Clusters Visulalization

```
df_scale.transformed = as.data.frame(-pc_out$x[,1:2])
k = 2
km_out = kmeans(df_scale.transformed, centers = k, nstart = 50)
fviz_cluster(km_out, data = df_scale.transformed,geom = "point",)
```



I see some grouping in the biplot of PCA components PC1 and PC2 with K-Means cluster. In this two-dimensional space, data points from the same K-Means cluster tend to be closer together. While there is considerable overlap within clusters, data points of the same colour tend to cluster together, indicating that K-Means has discovered some significant categories within the data. Also, there is significant overlap, particularly between nearby clusters. This implies that, while K-Means has effectively identified clusters, there may be some resemblance or shared traits across nearby clusters, making segmentation more subtle.

Characterizing Cluster

```
km1 <- kmeans(scale(df_sel),2,nstart = 50)
df <- df_sel %>% mutate(Cluster = km1$cluster)
df_cluster1 <- subset(df, Cluster == 1)
df_cluster2 <- subset(df, Cluster == 2)
```

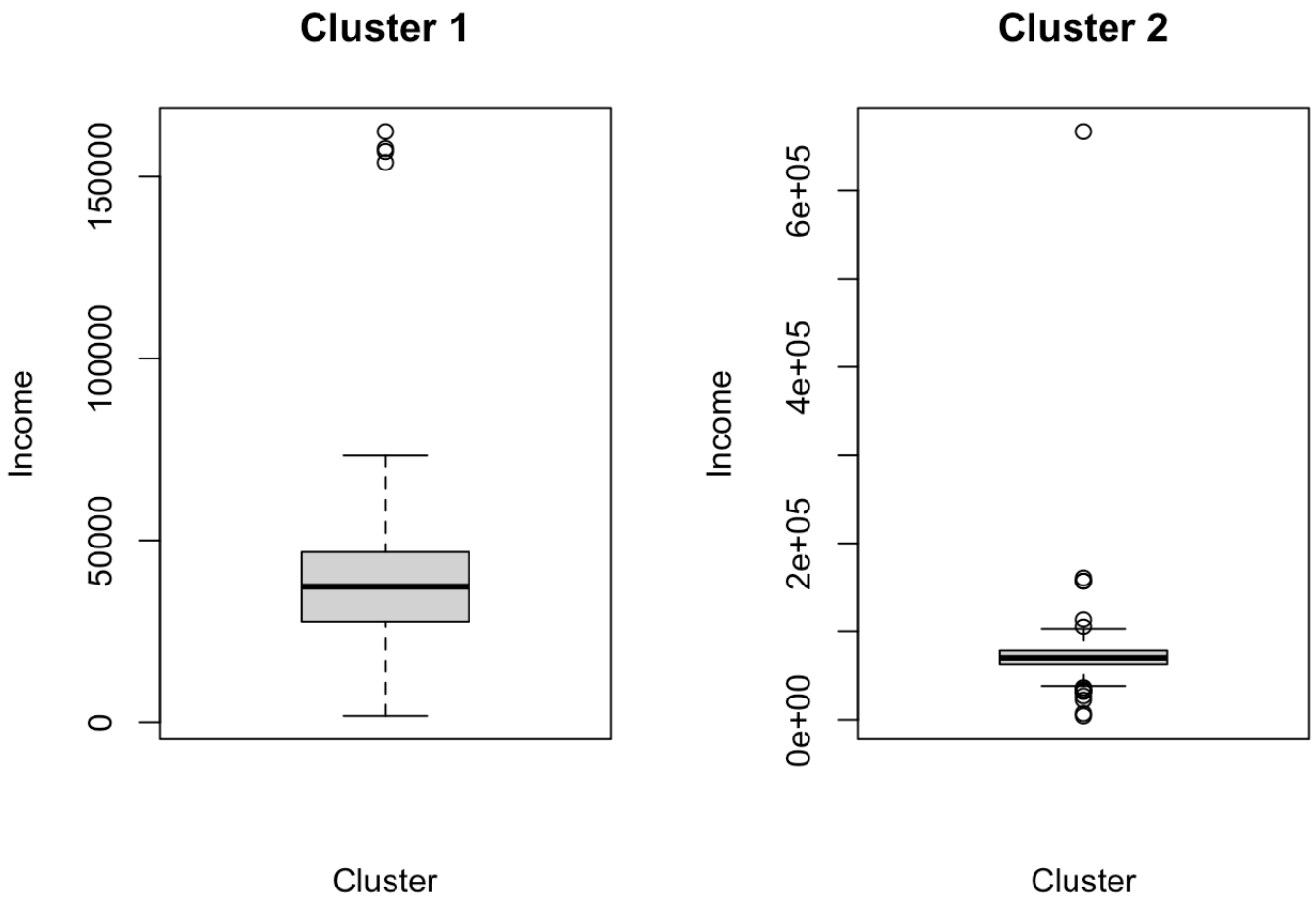
```
summary(df_cluster1)
```

```
##      Income          Kidhome         Teenhome        Recency
##  Min.   :  1730   Min.   :0.0000   Min.   :0.0000   Min.   : 0.00
##  1st Qu.: 27733   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:24.00
##  Median : 37292   Median :1.0000   Median :1.0000   Median :49.00
##  Mean   : 37789   Mean   :0.7213   Mean   :0.5317   Mean   :48.73
##  3rd Qu.: 46779   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:74.00
##  Max.   :162397   Max.   :2.0000   Max.   :2.0000   Max.   :99.00
##     MntWines        MntFruits      MntMeatProducts   MntFishProducts
##  Min.   :  0.00   Min.   : 0.000   Min.   :   0.00   Min.   :  0.000
##  1st Qu.: 10.00   1st Qu.: 1.000   1st Qu.:   9.00   1st Qu.:  2.000
##  Median : 30.00   Median : 3.000   Median :  18.00   Median :  4.000
##  Mean   : 82.25   Mean   : 5.977   Mean   :  33.99   Mean   :  8.806
##  3rd Qu.:109.00   3rd Qu.: 7.000   3rd Qu.:  45.00   3rd Qu.: 11.000
##  Max.   :750.00   Max.   :70.000   Max.   :1725.00   Max.   :150.000
##  MntSweetProducts  MntGoldProds    NumWebPurchases  NumStorePurchases
##  Min.   : 0.000   Min.   :  0.00   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 0.000   1st Qu.:  5.00   1st Qu.: 1.000   1st Qu.: 3.000
##  Median : 3.000   Median : 12.00   Median : 2.000   Median : 3.000
##  Mean   : 5.807   Mean   : 20.25   Mean   : 2.704   Mean   : 3.699
##  3rd Qu.: 8.000   3rd Qu.: 25.00   3rd Qu.: 4.000   3rd Qu.: 4.000
##  Max.   :78.000   Max.   :262.00   Max.   :11.000   Max.   :12.000
##     Complain            Age          MembershipDays  EducationLevel
##  Min.   :0.00000   Min.   : 18.00   Min.   :  2.0   Min.   :13.00
##  1st Qu.:0.00000   1st Qu.: 36.00   1st Qu.:155.0   1st Qu.:17.00
##  Median :0.00000   Median : 42.00   Median :326.0   Median :17.00
##  Mean   :0.01124   Mean   : 43.58   Mean   :334.2   Mean   :17.97
##  3rd Qu.:0.00000   3rd Qu.: 51.00   3rd Qu.:502.0   3rd Qu.:19.00
##  Max.   :1.00000   Max.   :121.00   Max.   :701.0   Max.   :22.00
##  Marital_Status_Divorced Marital_Status_Married Marital_Status_Single
##  Min.   :0.0000          Min.   :0.0000         Min.   :0.0000
##  1st Qu.:0.0000          1st Qu.:0.0000         1st Qu.:0.0000
##  Median :0.0000          Median :0.0000         Median :0.0000
##  Mean   :0.0988          Mean   :0.3984         Mean   :0.2177
##  3rd Qu.:0.0000          3rd Qu.:1.0000         3rd Qu.:0.0000
##  Max.   :1.0000          Max.   :1.0000         Max.   :1.0000
##  Marital_Status_Together Marital_Status_Widow    Cluster
##  Min.   :0.000           Min.   :0.0000       Min.   :1
##  1st Qu.:0.000           1st Qu.:0.0000       1st Qu.:1
##  Median :0.000           Median :0.0000       Median :1
##  Mean   :0.261           Mean   :0.0241       Mean   :1
##  3rd Qu.:1.000           3rd Qu.:0.0000       3rd Qu.:1
##  Max.   :1.000           Max.   :1.0000       Max.   :1
```

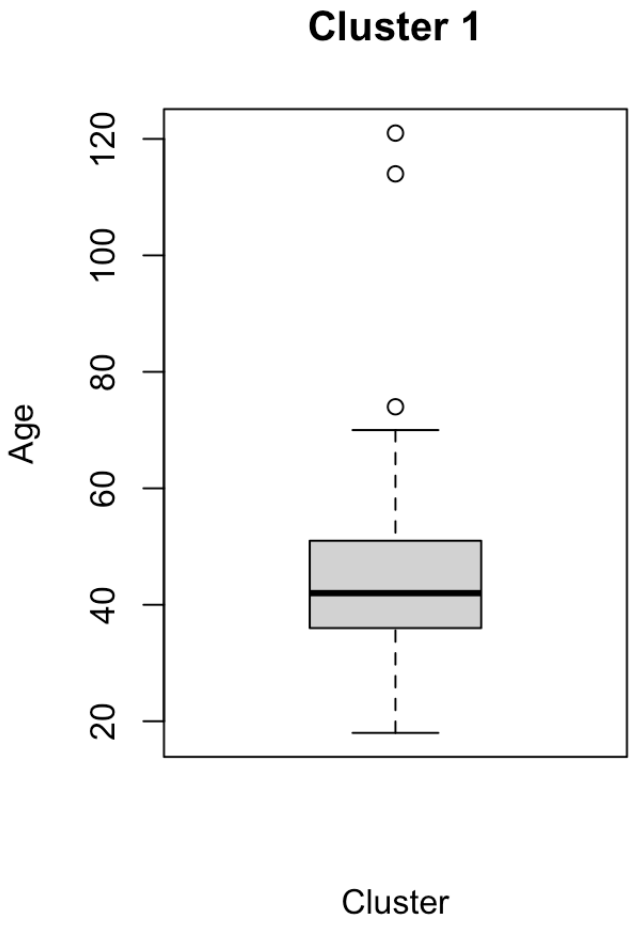
```
summary(df_cluster2)
```

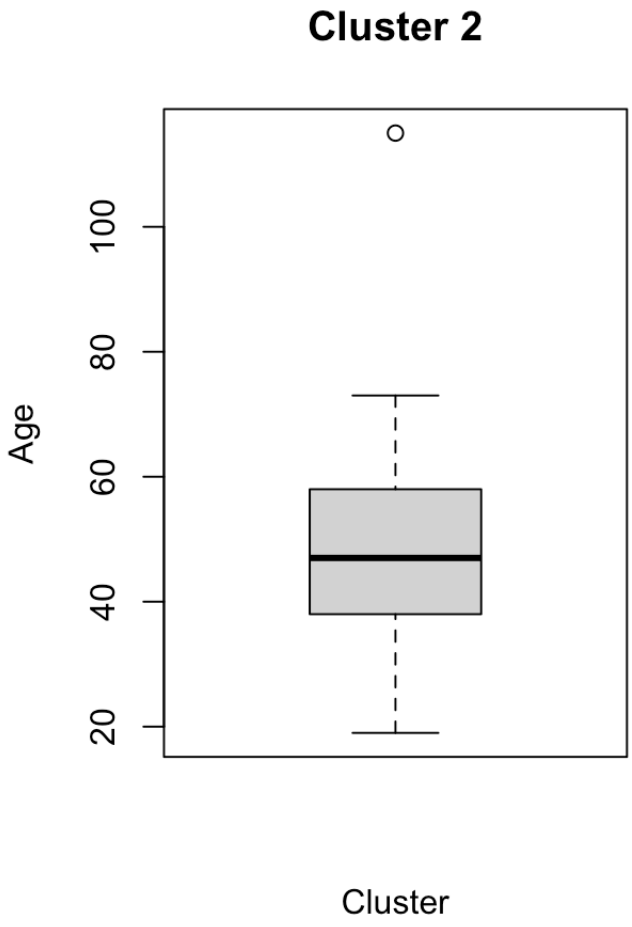
```
##      Income          Kidhome           Teenhome        Recency
##  Min.   :  4428   Min.   :0.00000   Min.   :0.000   Min.   : 0.00
##  1st Qu.: 62494   1st Qu.:0.00000   1st Qu.:0.000   1st Qu.:25.00
##  Median : 70430   Median :0.00000   Median :0.000   Median :51.00
##  Mean   : 70912   Mean   :0.08091   Mean   :0.471   Mean   :49.53
##  3rd Qu.: 78908   3rd Qu.:0.00000   3rd Qu.:1.000   3rd Qu.:74.00
##  Max.   :666666   Max.   :2.00000   Max.   :2.000   Max.   :99.00
##     MntWines        MntFruits      MntMeatProducts  MntFishProducts
##  Min.   :   1.0   Min.   :  0.00   Min.   :   3.0   Min.   :  0.00
##  1st Qu.: 356.0   1st Qu.: 16.00   1st Qu.: 144.8   1st Qu.: 23.00
##  Median : 546.5   Median : 35.00   Median : 270.0   Median : 58.00
##  Mean   : 593.1   Mean   : 52.67   Mean   : 339.2   Mean   : 74.69
##  3rd Qu.: 794.2   3rd Qu.: 80.00   3rd Qu.: 465.2   3rd Qu.:111.00
##  Max.   :1493.0   Max.   :199.00   Max.   :1725.0   Max.   :259.00
##  MntSweetProducts  MntGoldProds    NumWebPurchases  NumStorePurchases
##  Min.   :  0.00   Min.   :  0.00   Min.   : 0.000   Min.   : 0.00
##  1st Qu.: 16.00   1st Qu.: 30.00   1st Qu.: 4.000   1st Qu.: 6.00
##  Median : 38.50   Median : 54.00   Median : 6.000   Median : 9.00
##  Mean   : 54.54   Mean   : 74.32   Mean   : 5.861   Mean   : 8.52
##  3rd Qu.: 83.00   3rd Qu.:107.00   3rd Qu.: 7.000   3rd Qu.:11.00
##  Max.   :262.00   Max.   :321.00   Max.   :27.000   Max.   :13.00
##     Complain            Age         MembershipDays  EducationLevel
##  Min.   :0.000000   Min.   : 19.00   Min.   :  2.0   Min.   :13.00
##  1st Qu.:0.000000   1st Qu.: 38.00   1st Qu.:217.0   1st Qu.:17.00
##  Median :0.000000   Median : 47.00   Median :405.0   Median :17.00
##  Mean   :0.007261   Mean   : 47.28   Mean   :383.0   Mean   :18.34
##  3rd Qu.:0.000000   3rd Qu.: 58.00   3rd Qu.:557.5   3rd Qu.:19.00
##  Max.   :1.000000   Max.   :115.00   Max.   :700.0   Max.   :22.00
##  Marital_Status_Divorced Marital_Status_Married Marital_Status_Single
##  Min.   :0.0000          Min.   :0.0000         Min.   :0.0000
##  1st Qu.:0.0000          1st Qu.:0.0000         1st Qu.:0.0000
##  Median :0.0000          Median :0.0000         Median :0.0000
##  Mean   :0.1131          Mean   :0.3745         Mean   :0.2075
##  3rd Qu.:0.0000          3rd Qu.:1.0000         3rd Qu.:0.0000
##  Max.   :1.0000          Max.   :1.0000         Max.   :1.0000
##  Marital_Status_Together Marital_Status_Widow    Cluster
##  Min.   :0.0000          Min.   :0.00000      Min.   :2
##  1st Qu.:0.0000          1st Qu.:0.00000      1st Qu.:2
##  Median :0.0000          Median :0.00000      Median :2
##  Mean   :0.2573          Mean   :0.04772      Mean   :2
##  3rd Qu.:1.0000          3rd Qu.:0.00000      3rd Qu.:2
##  Max.   :1.0000          Max.   :1.00000      Max.   :2
```

```
#COMPARING INCOME
par(mfrow = c(1, 2))
boxplot(Income ~ Cluster, data = df[df$Cluster == 1,],
        main = "Cluster 1", ylab = "Income")
boxplot(Income ~ Cluster, data = df[df$Cluster == 2,],
        main = "Cluster 2", ylab = "Income")
```

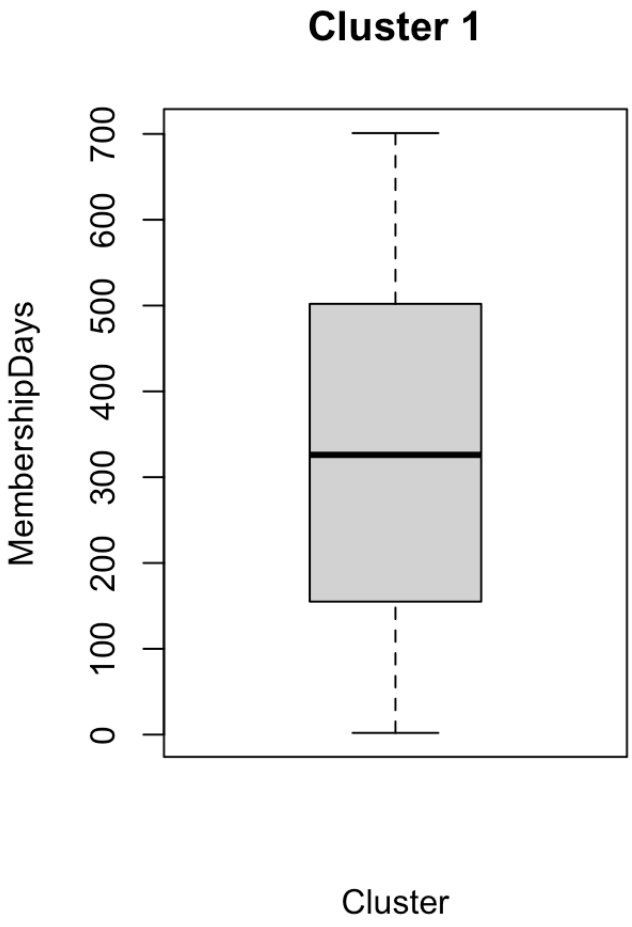


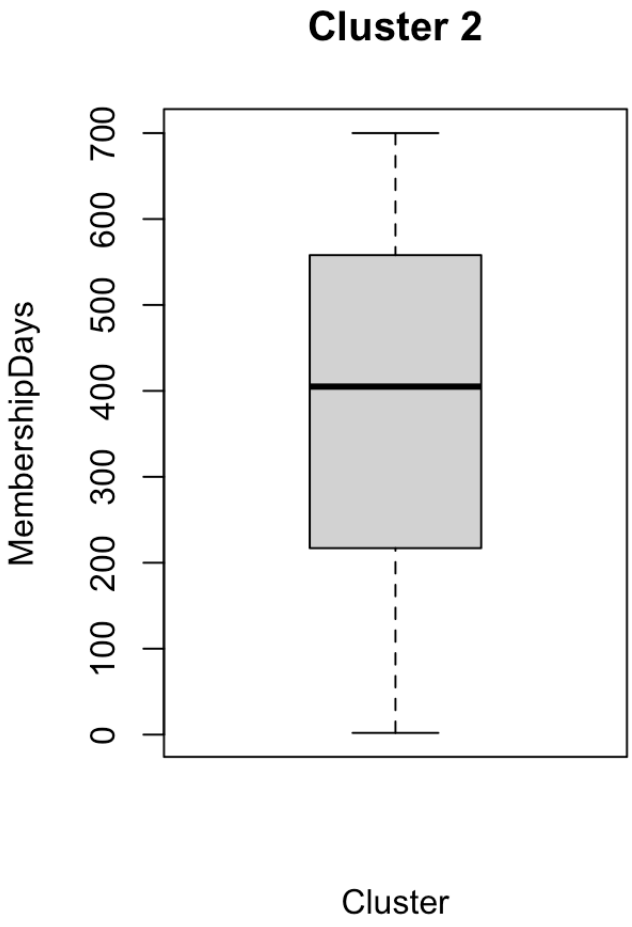

```
 #COMPARING AGE
par(mfrow = c(1, 2))
boxplot(Age ~ Cluster, data = df[df$Cluster == 1,],
        main = "Cluster 1", ylab = "Age")
boxplot(Age ~ Cluster, data = df[df$Cluster == 2,],
        main = "Cluster 2", ylab = "Age")
```



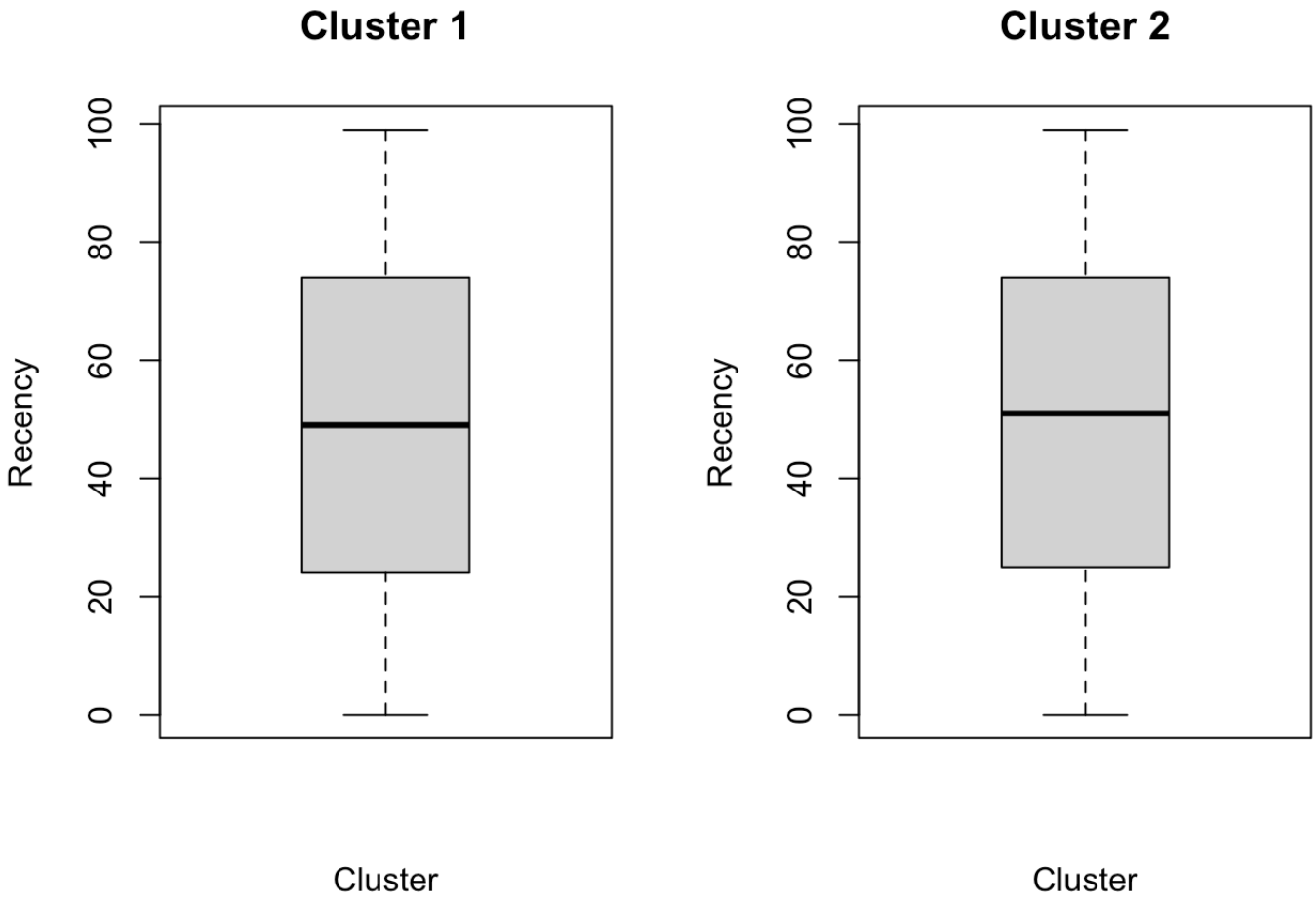



```
#COMPARING MEMBERSHIP DAYS
par(mfrow = c(1, 2))
boxplot(MembershipDays ~ Cluster, data = df[df$Cluster == 1,],
        main = "Cluster 1", ylab = "MembershipDays")
boxplot(MembershipDays ~ Cluster, data = df[df$Cluster == 2,],
        main = "Cluster 2", ylab = "MembershipDays")
```

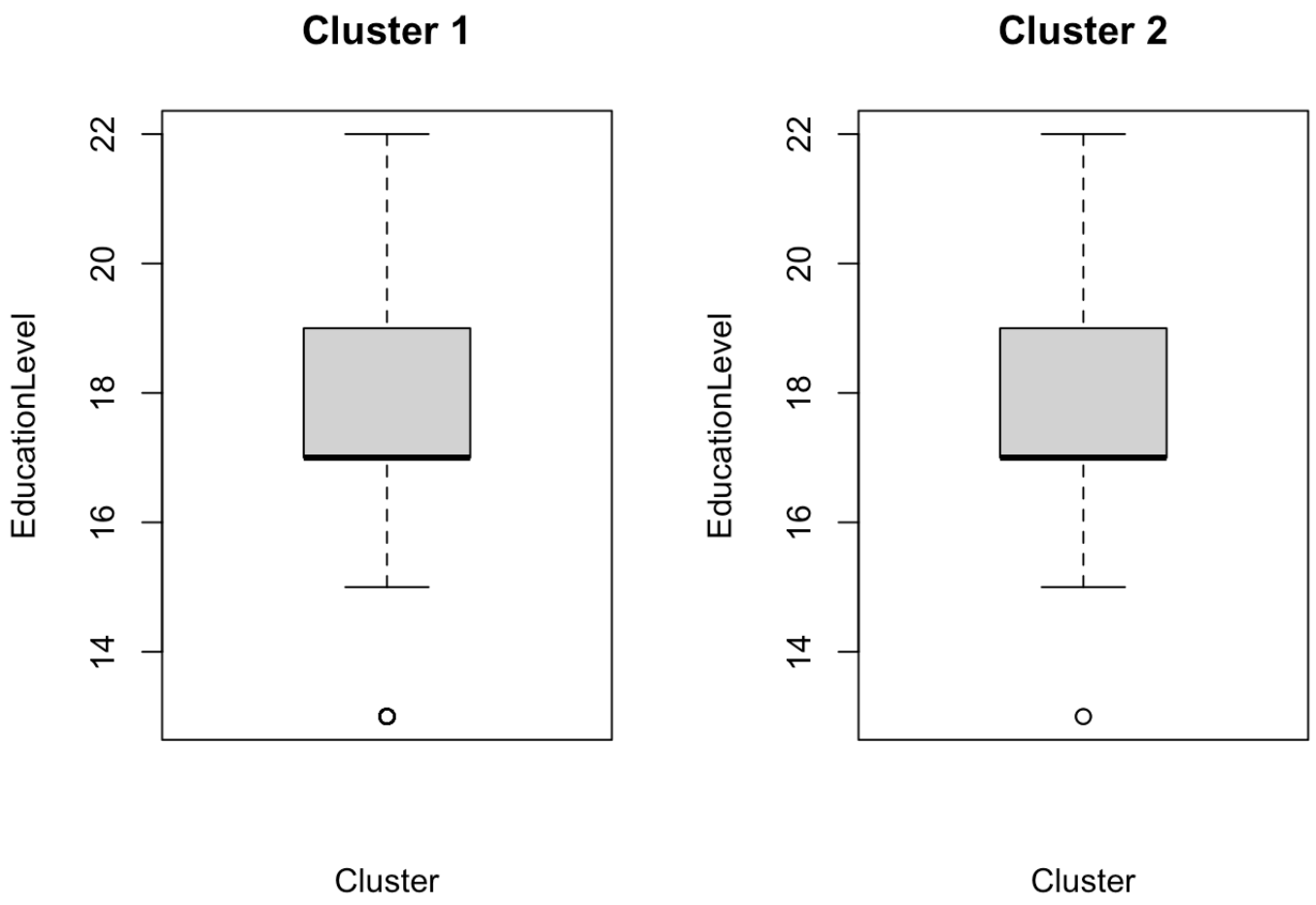




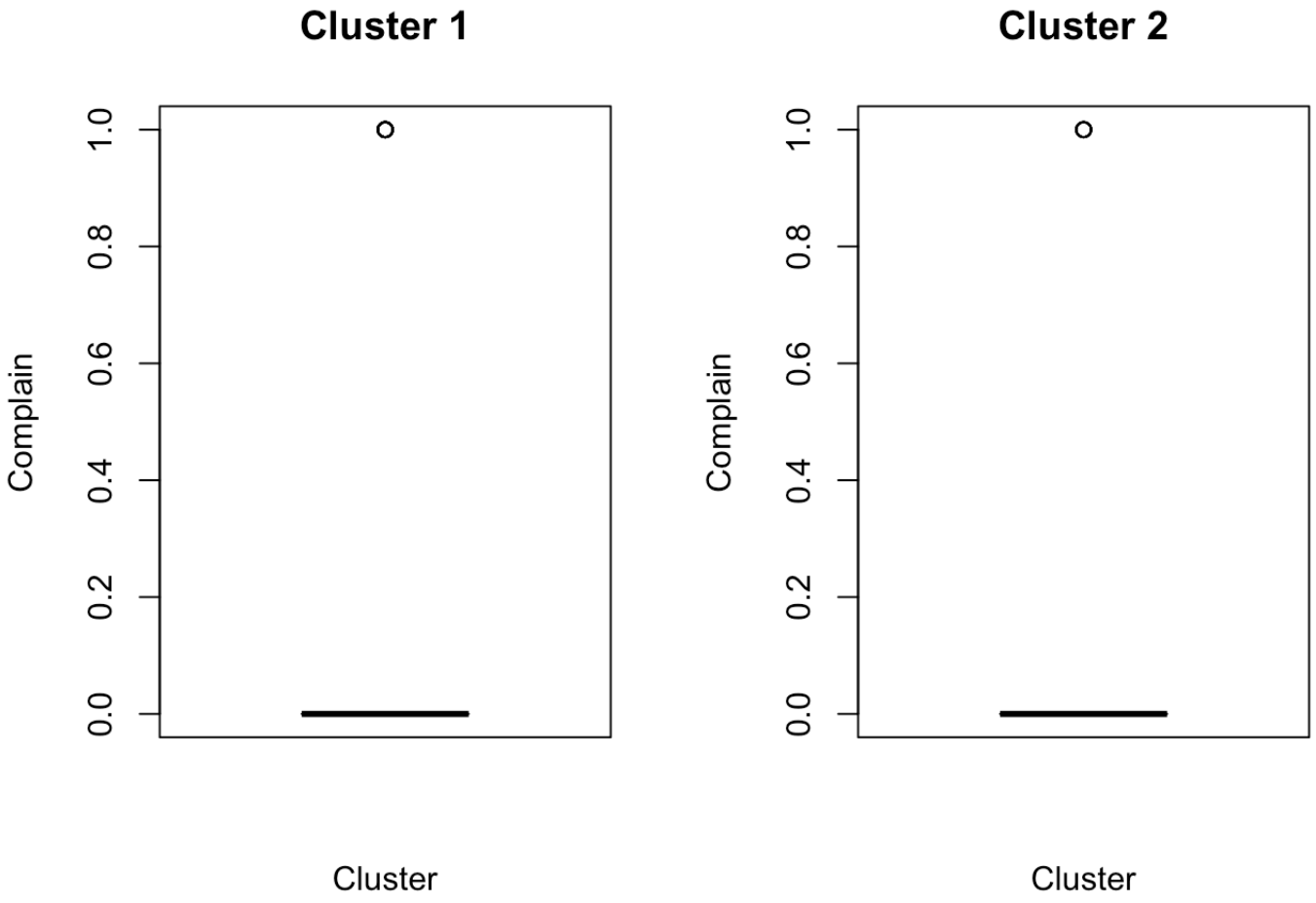

```
#COMPARING RECENCY
par(mfrow = c(1, 2))
boxplot(Recency ~ Cluster, data = df[df$Cluster == 1,],
        main = "Cluster 1", ylab = "Recency")
boxplot(Recency ~ Cluster, data = df[df$Cluster == 2,],
        main = "Cluster 2", ylab = "Recency")
```




```
#COMPARING EducationLevel
par(mfrow = c(1, 2))
boxplot(EducationLevel ~ Cluster, data = df[df$Cluster == 1,],
        main = "Cluster 1", ylab = "EducationLevel")
boxplot(EducationLevel ~ Cluster, data = df[df$Cluster == 2,],
        main = "Cluster 2", ylab = "EducationLevel")
```




```
#COMPARING Complain
par(mfrow = c(1, 2))
boxplot(Complain ~ Cluster, data = df[df$Cluster == 1,],
        main = "Cluster 1", ylab = "Complain")
boxplot(Complain ~ Cluster, data = df[df$Cluster == 2,],
        main = "Cluster 2", ylab = "Complain")
```



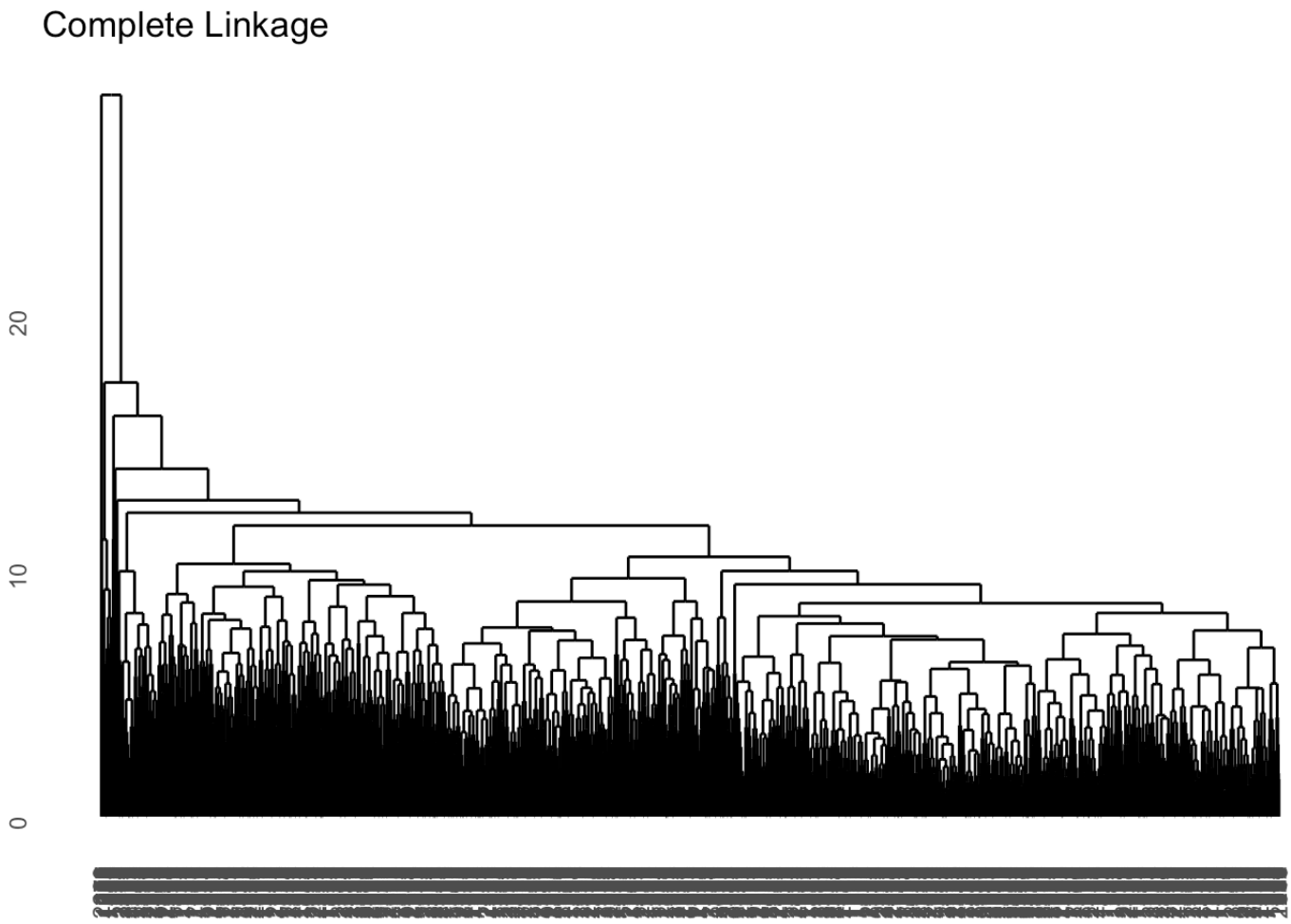
OBSERVATIONS:

Cluster 1

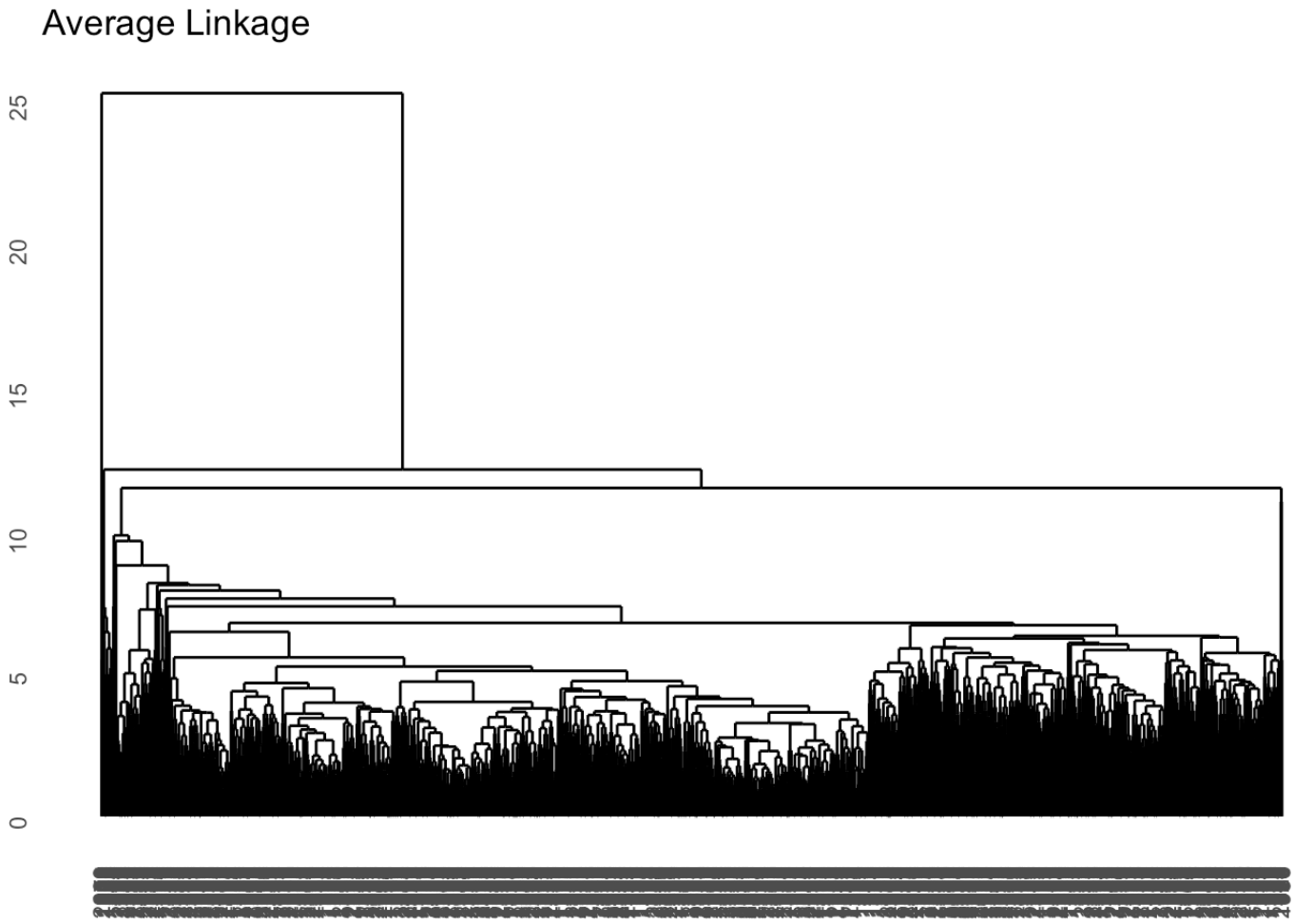
- The proportion of children and teens in families is greater with average of 0.7213 and 0.5317. *The customers in this cluster purchased less recently of average 48.73* This cluster has a lower average income with average 37789 *spending on various product categories is lower.* Customers in this cluster make less purchases online and in stores.
- The average age is relatively low of average 43
- The average length of membership is both relatively low of average 334 days *This group has additional complaints of average 0.011

Cluster2: *The presence of children and teens in families is lower with average of 0.08091 and 0.471* In this cluster, expenditure on numerous product categories is greater * This cluster has a greater average income with average of 70000 *spending on various product categories is higher* Customers in this cluster make more purchases online and in stores of average 5.861 and 8.52. *The average age is considerably higher of average 47 The average length of membership are higher of average 383 days* This cluster has fewer complaints 0.007261 comparetively

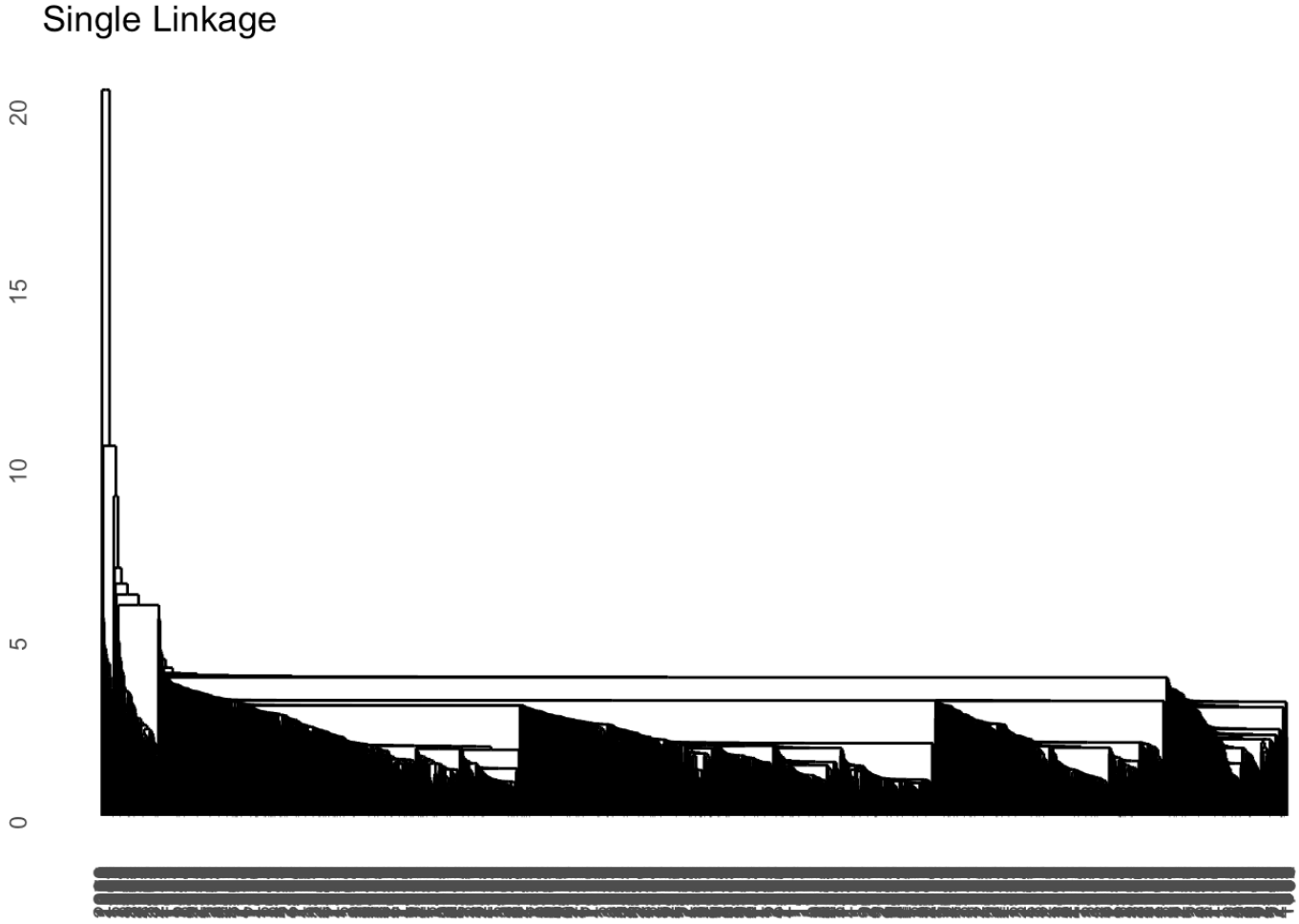
```
x_dist <- dist(df_scale, method = "euclidean")
hc.complete <- hclust(x_dist, method = "complete")
hc.average <- hclust(x_dist, method = "average")
hc.single <- hclust(x_dist, method = "single")

#par(mfrow = c(1, 3))
ggdendrogram(hc.complete, segements=TRUE, labels=TRUE, leaf_labels = TRUE, rotate=FAL
SE, theme_dendro = TRUE) +
 labs(title='Complete Linkage')
```




```
ggdendrogram(hc.average, segements=TRUE, labels=TRUE, leaf_labels = TRUE, rotate=FALS
E, theme_dendro = TRUE) +
 labs(title='Average Linkage')
```




```
ggdendrogram(hc.single, segements=TRUE, labels=TRUE, leaf_labels = TRUE, rotate=FALS
E, theme_dendro = TRUE) +
 labs(title='Single Linkage')
```



scaling data is preferred, and I will be utilizing it. As the data as columns like Age, income, Recency, number of purchased etc which are nominal data. And we have data like martial status, education level, Kidhome, teenhome are ordinal therefore, if we did not scale there is high chance of clustering being biased on values instead of actual relation.Therefore, scaling reveals that variable scales, rather than underlying data connections, dominating the clustering process.

The choice of average and complete linkage will be better option compare to single linkage as in single linkage we are not able differentiate and it is not balanced in size.Also it is more sensitive to outliers and no clarity and no sharp jumps to classify the clusters. If we have to compare between average and complete linkage, for our objective average linkage will be a better choice over complete linkage as it is helping us to differentiate clearly between clusters with sharp jumps. The length of vertical lines for cluster merging is modest, reflecting a balance between compactness and separation.

Choosing clusters: I will be choosing 2 clusters as average linkage offers a balanced solution with two primary customer segments that are moderately distinct and internally cohesive which correlates with our objective of customer segmentation and we can also confirm that by a tree-like structure with branches that merge progressively can be seen.