MSDS680 - Week 4 - Decision Trees

Benjamin Siebold

Regis University

MSDS680 - Week 4 - Decision Trees

Introduction

Methodology

Set Up

```
library(DataExplorer) #data exploration

library(factoextra) #building wss and silhouette plots

library(tidyverse) #data cleaning

library(cluster) #applies HCA

library(dendextend) #compares dendrograms

library(caret) #dummy variables


set.seed(422)

customers <- read.csv('../Week-6/Wholesale customers data.csv')
```

Data Clean

```
summary(customers)
```

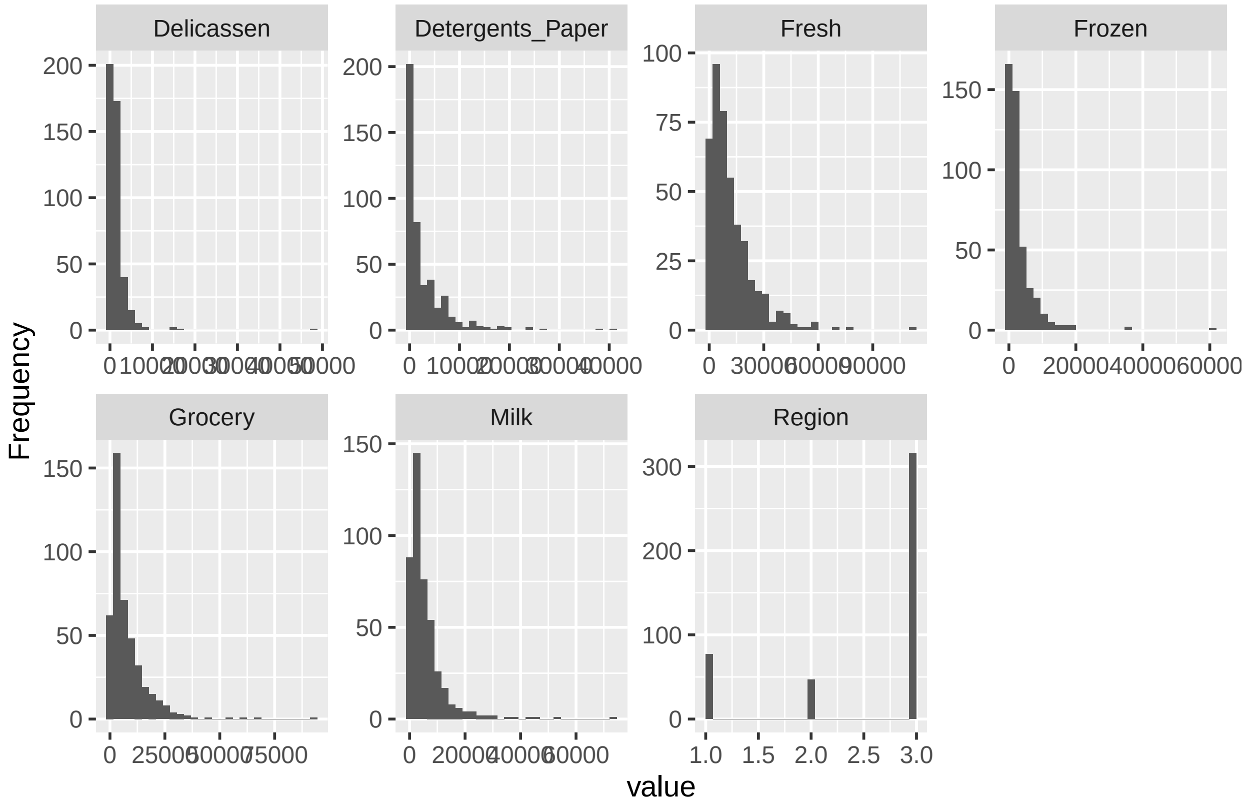
```
## Channel Region Fresh Milk

## Min.  :1.000 Min.  :1.000 Min.  : 3 Min.  : 55

## 1st Qu.:1.000 1st Qu.:2.000 1st Qu.: 3128 1st Qu.: 1533

## Median :1.000 Median :3.000 Median : 8504 Median : 3627

## Mean :1.323 Mean :2.543 Mean : 12000 Mean : 5796

## 3rd Qu.:2.000 3rd Qu.:3.000 3rd Qu.: 16934 3rd Qu.: 7190
## Max.  :2.000 Max.  :3.000 Max.  :112151 Max.  :73498
## Grocery Frozen Detergents_Paper Delicassen
## Min.  : 3 Min.  : 25.0 Min.  : 3.0 Min.  : 3.0
## 1st Qu.: 2153 1st Qu.: 742.2 1st Qu.: 256.8 1st Qu.:
408.2
## Median : 4756 Median : 1526.0 Median : 816.5 Median :
965.5
## Mean : 7951 Mean : 3071.9 Mean : 2881.5 Mean : 1524.9
## 3rd Qu.:10656 3rd Qu.: 3554.2 3rd Qu.: 3922.0 3rd Qu.:
1820.2
## Max.  :92780 Max.  :60869.0 Max.  :40827.0 Max.
:47943.0
```

```
str(customers)
```

```
## 'data.frame': 440 obs. of 8 variables:
## $ Channel : int 2 2 2 1 2 2 2 2 1 2 ...
## $ Region : int 3 3 3 3 3 3 3 3 3 3 ...
## $ Fresh : int 12669 7057 6353 13265 22615 9413 12126
7579 5963 6006 ...
## $ Milk : int 9656 9810 8808 1196 5410 8259 3199 4956
3648 11093 ...
## $ Grocery : int 7561 9568 7684 4221 7198 5126 6975 9426
6192 18881 ...
## $ Frozen : int 214 1762 2405 6404 3915 666 480 1669 425
1159 ...
## $ Detergents_Paper: int 2674 3293 3516 507 1777 1795
3140 3321 1716 7425 ...
```

```
## $ Delicassen : int 1338 1776 7844 1788 5185 1451 545
2566 750 2098 ...
```

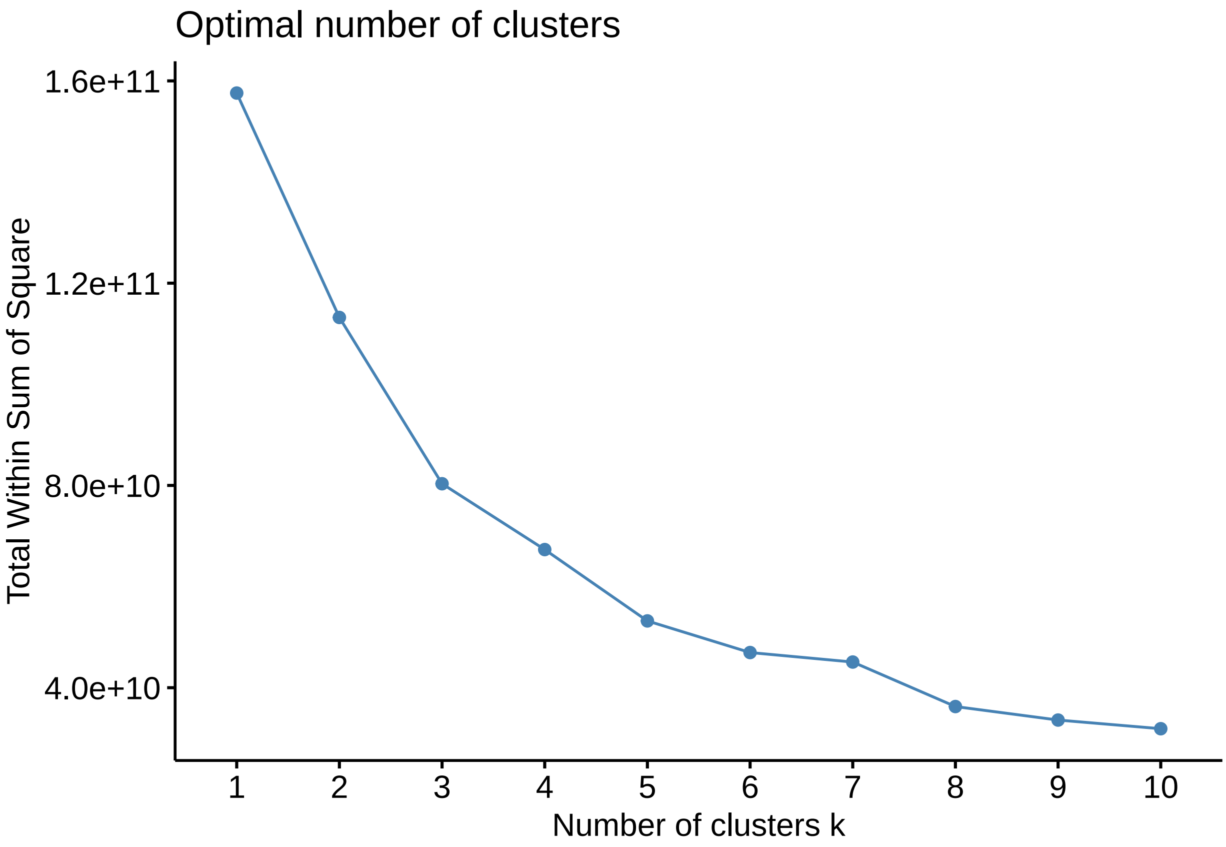
```
plot_histogram(customers)
```



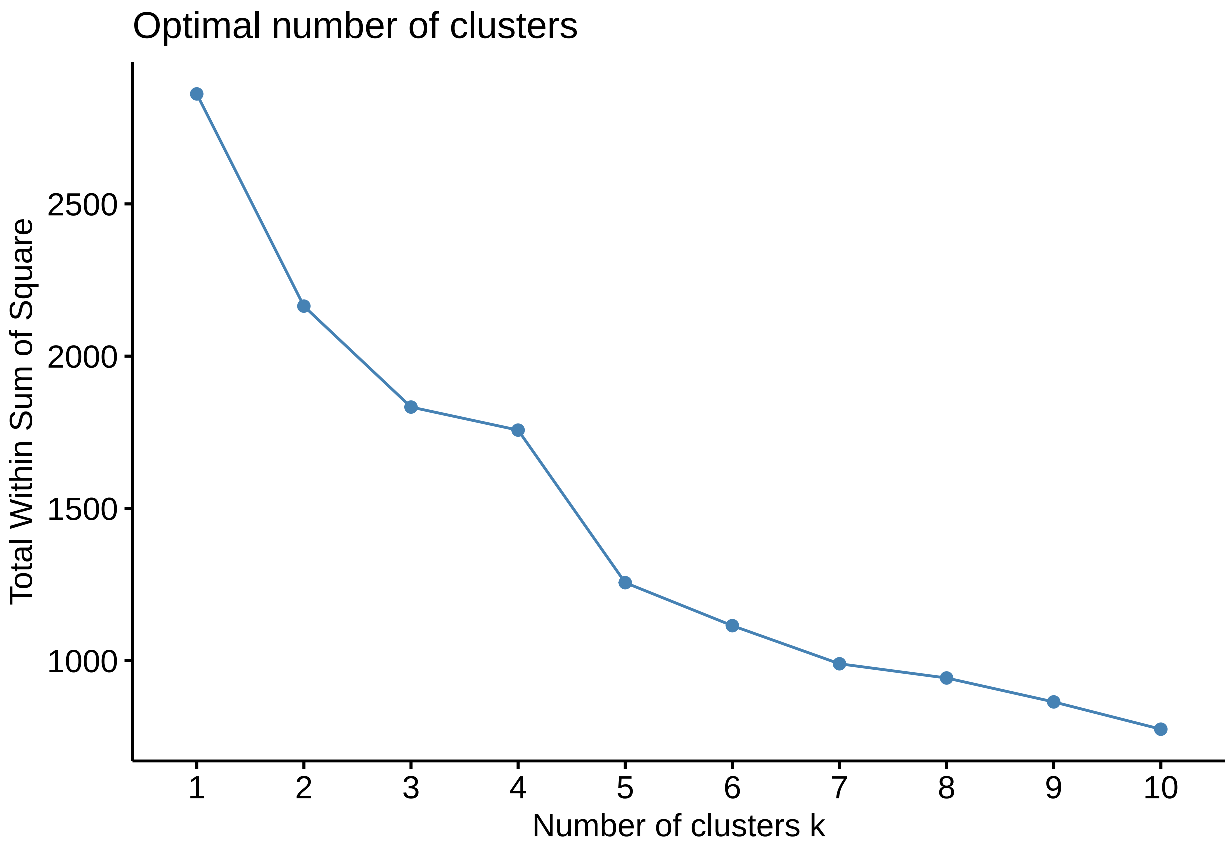
```
customers_factor <- as.data.frame(lapply(customers[,c(1,2)], as.factor))
customers_scaled <- as.data.frame(lapply(customers[,c(3:8)], scale))
dummy_vars <- dummyVars('~.', data=customers_factor,fullRank=T)
dummy_customers <- as.data.frame(predict(dummy_vars, newdata=customers_factor))

#creates dummy dataset without scales
clean_customers <- cbind(dummy_customers,customers[,c(3:8)])
#creates scaled dataset with dummy variables
scaled_clean_cust <- cbind(dummy_customers, customers_scaled)
```

Kmeans Cluster Decision

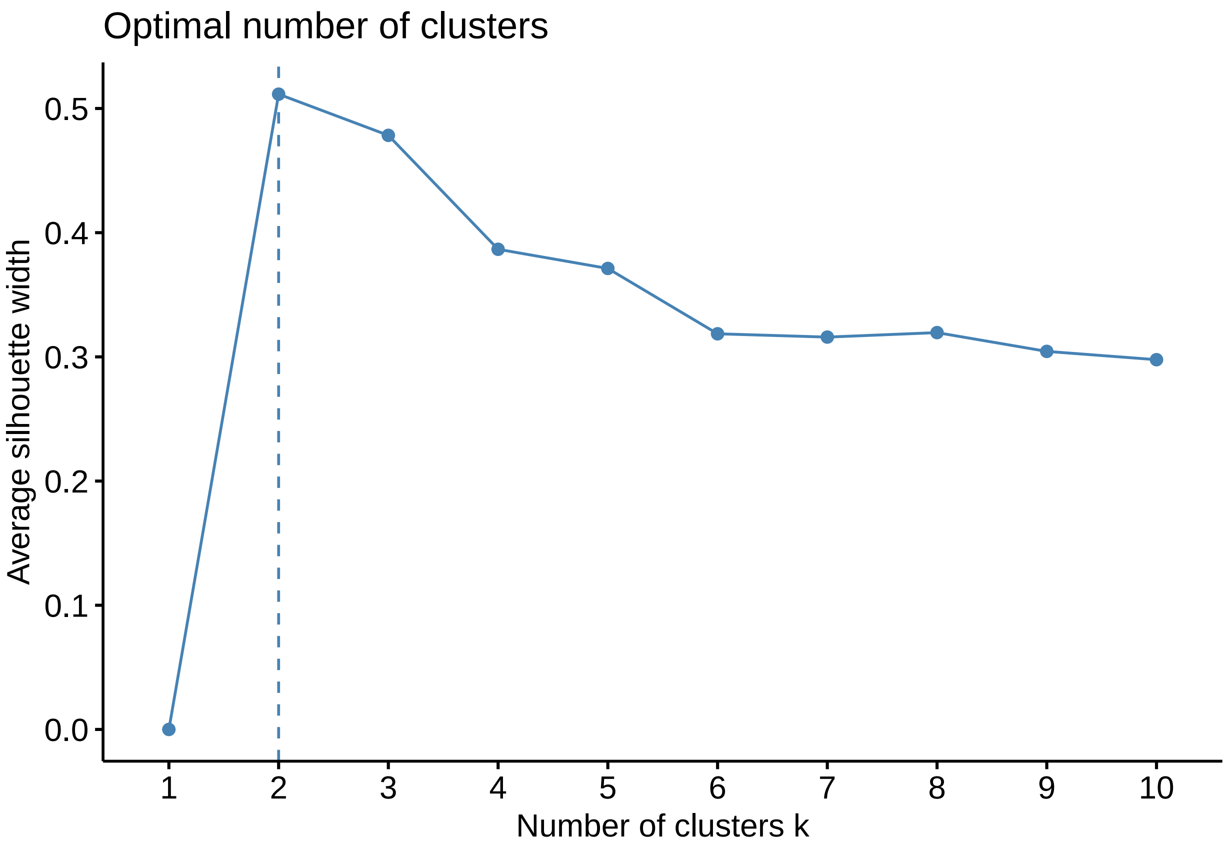
```r
fviz_nbclust(clean_customers, kmeans, method = 'wss')
```



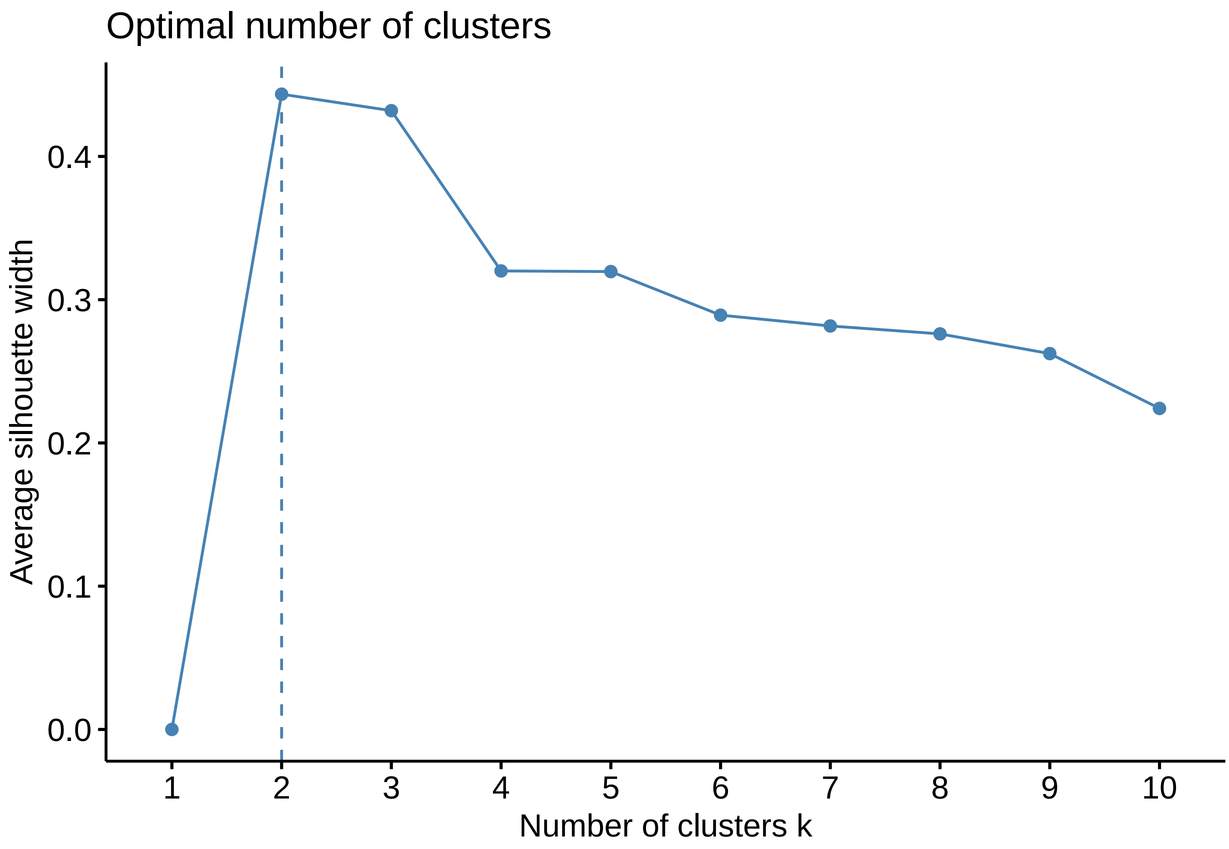
```r
fviz_nbclust(scaled_clean_cust, kmeans, method = 'wss')
```



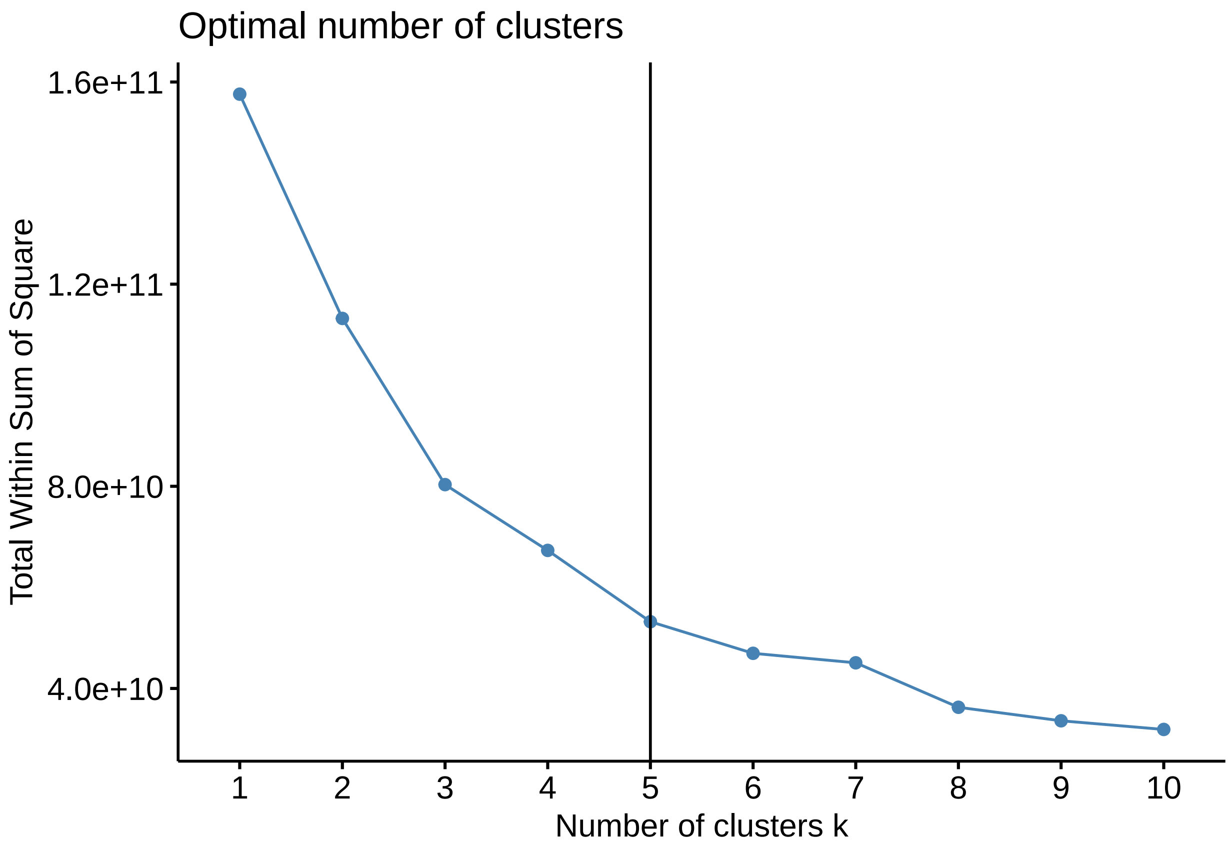
```
fviz_nbclust(clean_customers, kmeans, method = 'silhouette')
```



```
fviz_nbclust(scaled_clean_cust, kmeans, method = 'silhouette')
```



```
fviz_nbclust(clean_customers, kmeans, method = 'wss') +
  geom_vline(xintercept = 5, linetype = 1)
```



Kmeans Model

```
kmeans_fit <- kmeans(clean_customers, 5)


kmeans_fit$size
```

```
## [1] 113  24 227   5  71
```

```
kmeans_fit$centers
```

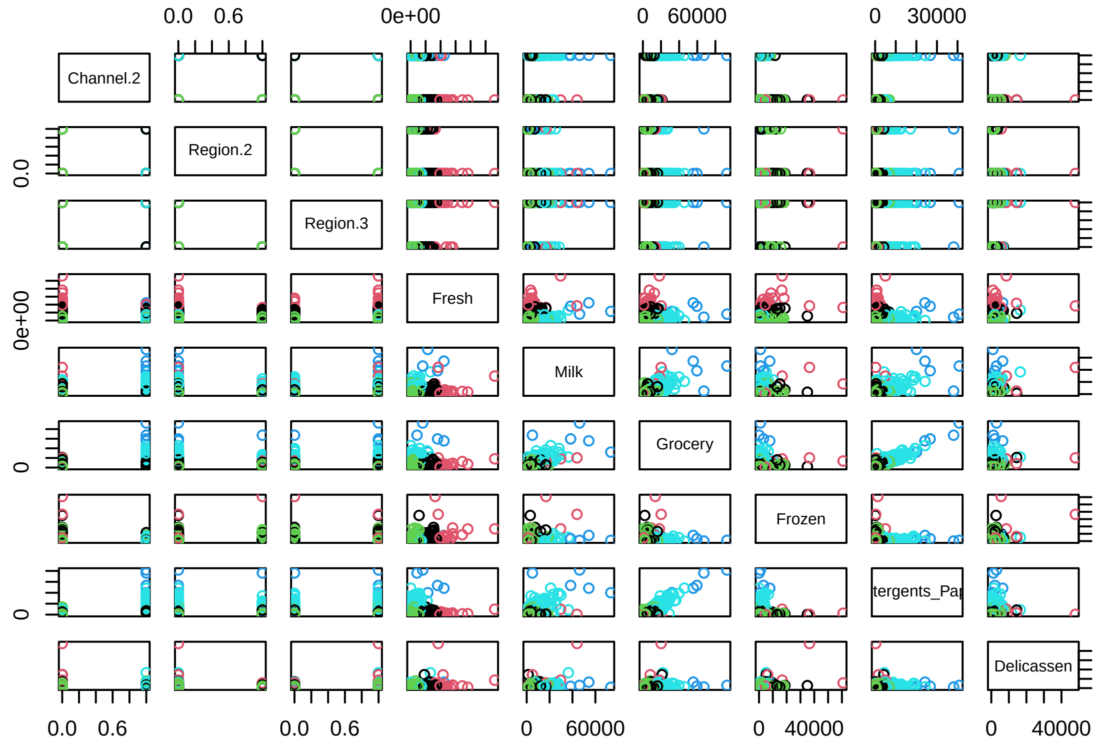
```
##    Channel.2   Region.2  Region.3     Fresh      Milk   Grocery   Frozen
## 1 0.19469027 0.11504425 0.7168142 20600.283  3787.832  5089.841 3989.071
## 2 0.08333333 0.04166667 0.8333333 48777.375  6607.375  6197.792 9462.792
## 3 0.20264317 0.09691630 0.7224670  5655.819  3567.793  4513.040 2386.529
## 4 1.00000000 0.20000000 0.8000000 25603.000 43460.600 61472.200 2636.000
## 5 0.94366197 0.14084507 0.6619718  5207.831 13191.028 20321.718 1674.028
##   Detergents_Paper Delicassen
## 1         1130.142   1639.071
## 2          932.125   4435.333
## 3         1437.559   1005.031
## 4        29974.200   2708.800
## 5         9036.380   1937.944
```

```
(kmeans_fit$betweenss/kmeans_fit$totss) #provides fit score
```

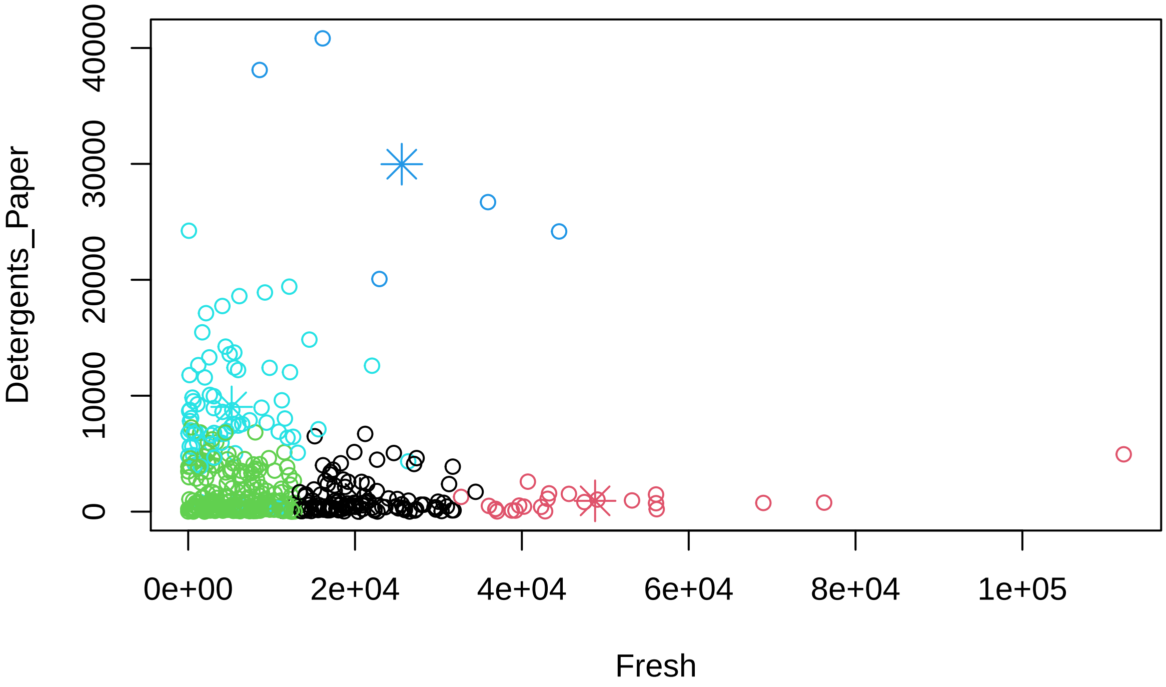
```
## [1] 0.6629548
```

Feature Compare

```
#gives all the plots to provide a good candidate for inspection
plot(clean_customers, col=kmeans_fit$cluster)
```



```
plot(clean_customers[c('Fresh','Detergents_Paper')], col=kmeans_fit$cluster)
#adds center of cluster to plot
points(kmeans_fit$centers[,c('Fresh', 'Detergents_Paper')],
       col=1:5, pch=8, cex=2)
```



Kmeans analysis

```
clean_customers$kmeans_cluster <- kmeans_fit$cluster


head(clean_customers)
```

```
##   Channel.2 Region.2 Region.3 Fresh Milk Grocery Frozen Detergents_Paper
## 1         1        0        1 12669 9656    7561    214             2674
## 2         1        0        1  7057 9810    9568   1762             3293
## 3         1        0        1  6353 8808    7684   2405             3516
## 4         0        0        1 13265 1196    4221   6404              507
## 5         1        0        1 22615 5410    7198   3915             1777
## 6         1        0        1  9413 8259    5126    666             1795
##   Delicassen kmeans_cluster
## 1       1338              3
## 2       1776              3
## 3       7844              3
## 4       1788              1
## 5       5185              1
## 6       1451              3
```

```
# summary statistics grouped by cluster
aggregate(clean_customers, list(clean_customers$kmeans_cluster), min)
```

```
##   Group.1 Channel.2 Region.2 Region.3 Fresh Milk Grocery Frozen
## 1       1         0        0        0 11314  134       3    118
## 2       2         0        0        0 32717  286     471    532
## 3       3         0        0        0     3   55     137     25
## 4       4         1        0        0  8565 4980   32114    131
## 5       5         0        0        0    18 1275   10487     33
##   Detergents_Paper Delicassen kmeans_cluster
## 1                3         57              1
## 2               20          3              2
## 3                3          3              3
## 4            20070        903              4
## 5              282          3              5
```

```
aggregate(clean_customers, list(clean_customers$kmeans_cluster), max)
```

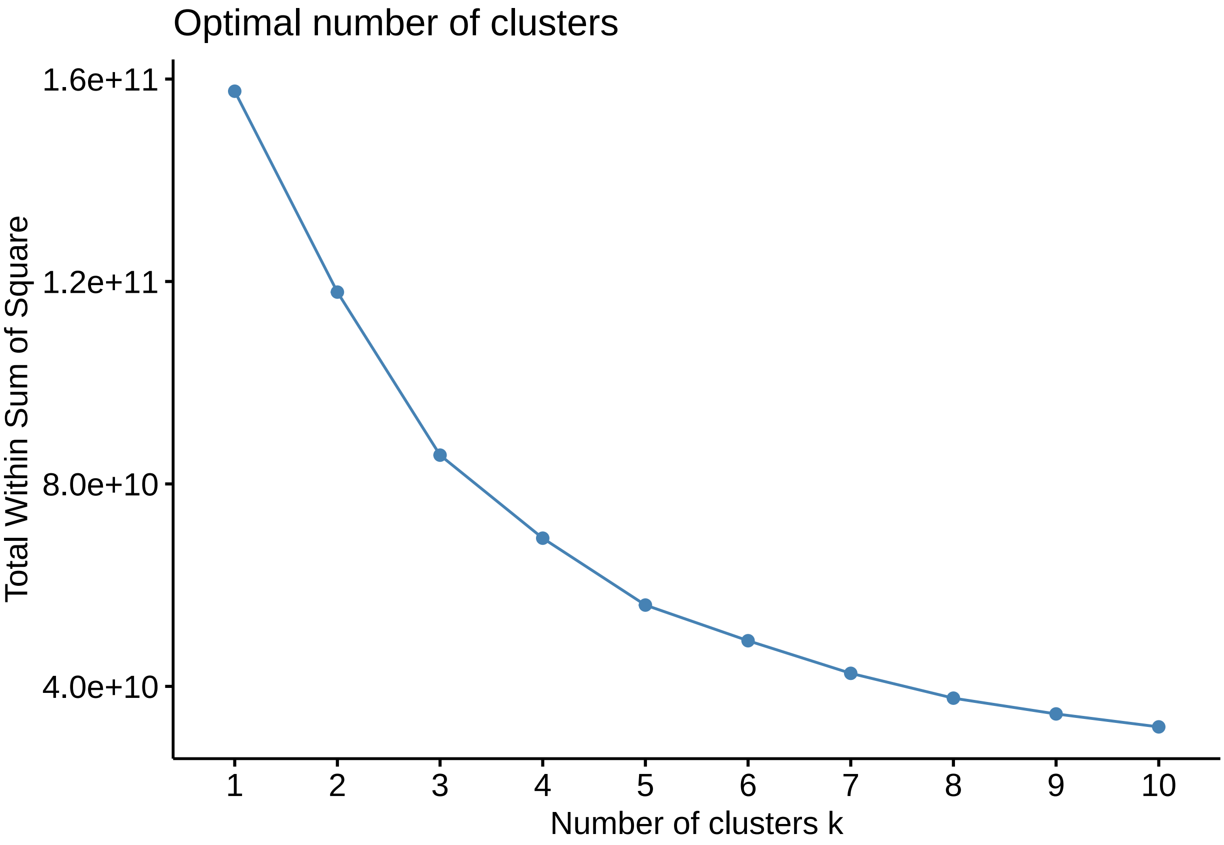
```
##   Group.1 Channel.2 Region.2 Region.3  Fresh  Milk Grocery Frozen
## 1       1         1        1         1  34454 16687   21042  35009
## 2       2         1        1         1 112151 43950   20170  60869
## 3       3         1        1         1  13146 18664   16483  17866
## 4       4         1        1         1  44466 73498   92780   7782
## 5       5         1        1         1  26373 36423   45828  10155
##   Detergents_Paper Delicassen kmeans_cluster
## 1             6707      14472              1
## 2             4948      47943              2
## 3             7271       7844              3
## 4            40827       6465              4
## 5            24231      16523              5
```

```r
aggregate(clean_customers, list(clean_customers$kmeans_cluster), mean)
```

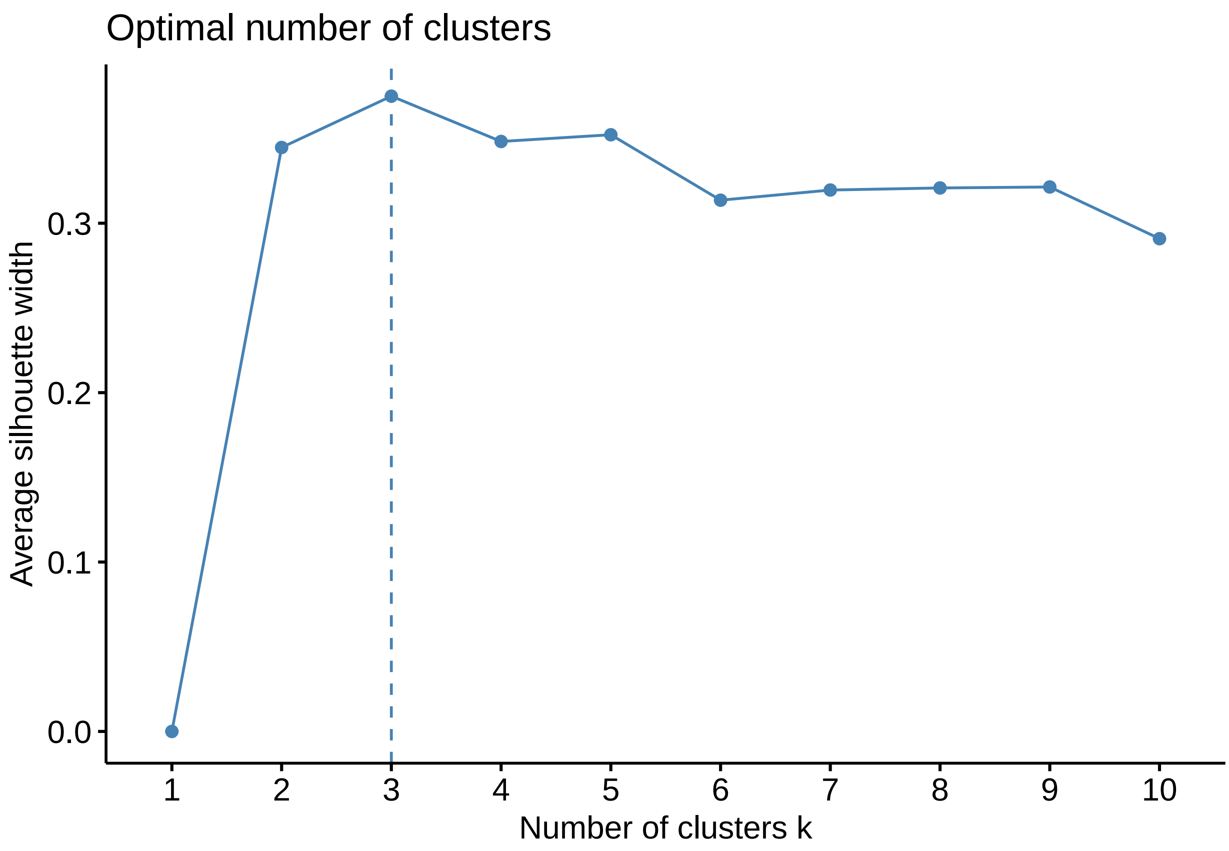
```
##   Group.1  Channel.2   Region.2  Region.3      Fresh      Milk   Grocery
## 1       1 0.19469027 0.11504425 0.7168142 20600.283  3787.832  5089.841
## 2       2 0.08333333 0.04166667 0.8333333 48777.375  6607.375  6197.792
## 3       3 0.20264317 0.09691630 0.7224670  5655.819  3567.793  4513.040
## 4       4 1.00000000 0.20000000 0.8000000 25603.000 43460.600 61472.200
## 5       5 0.94366197 0.14084507 0.6619718  5207.831 13191.028 20321.718
##     Frozen Detergents_Paper Delicassen kmeans_cluster
## 1 3989.071         1130.142   1639.071              1
## 2 9462.792          932.125   4435.333              2
## 3 2386.529         1437.559   1005.031              3
## 4 2636.000        29974.200   2708.800              4
## 5 1674.028         9036.380   1937.944              5
```

HCA Cluster Decision

```
fviz_nbclust(clean_customers, hcut, method  = 'wss') #hca plot with wss
```



```
fviz_nbclust(clean_customers, hcut, method  = 'silhouette') #hca silhouette
```

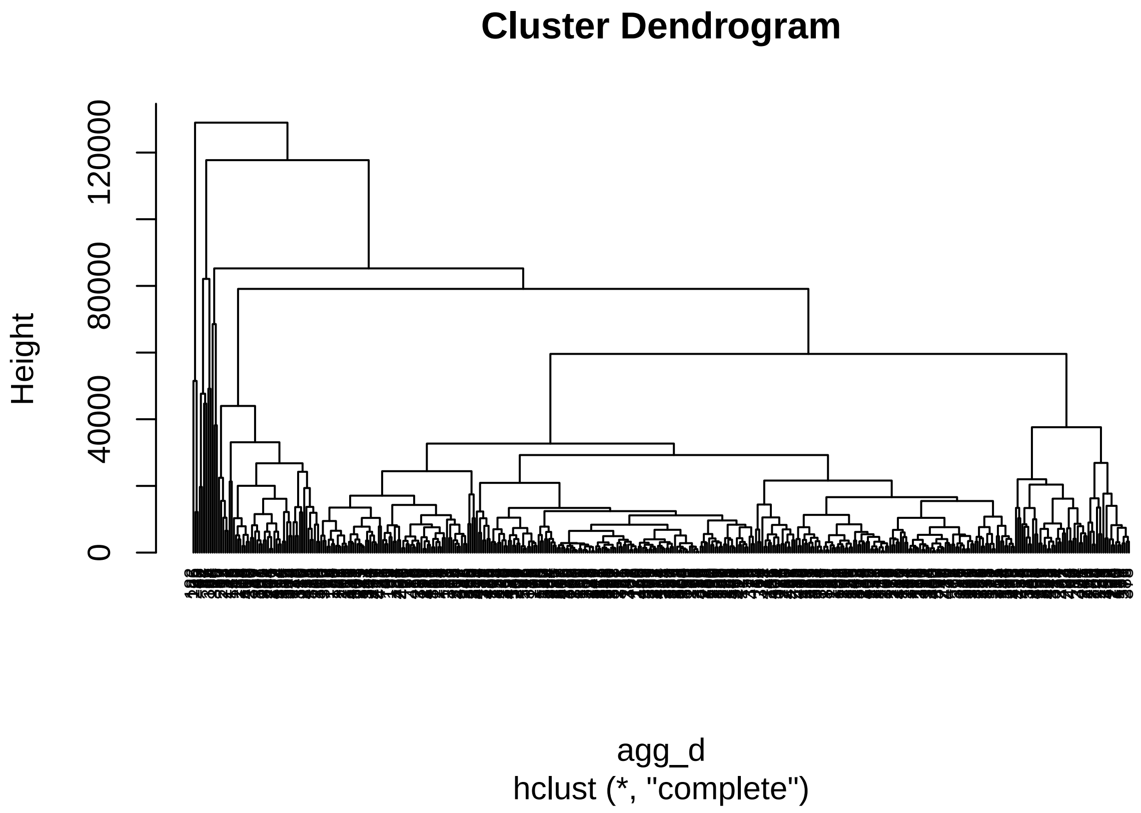


```
#agglomerative distance between points using euclidean
agg_d <- dist(clean_customers, method = 'euclidean')
```

Agglomerative

```
hc_complete_agg <- hclust(agg_d, method = 'complete') #HCA using complete method
plot(hc_complete_agg, cex =.6, hang = -1)
```

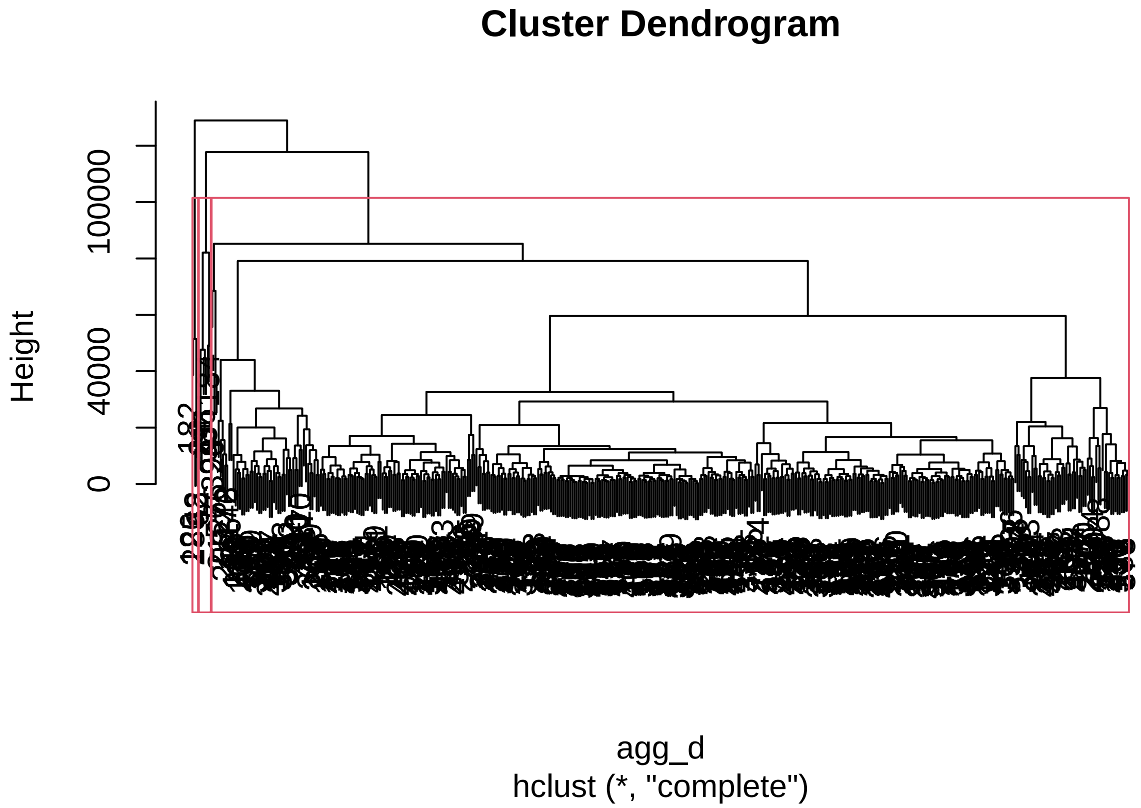
Complete.



```
hc_complete_agg_fit <- cutree(hc_complete_agg, k=3) #splits HCA into 3 clusters


table(hc_complete_agg_fit) #number of data points in each cluster
```

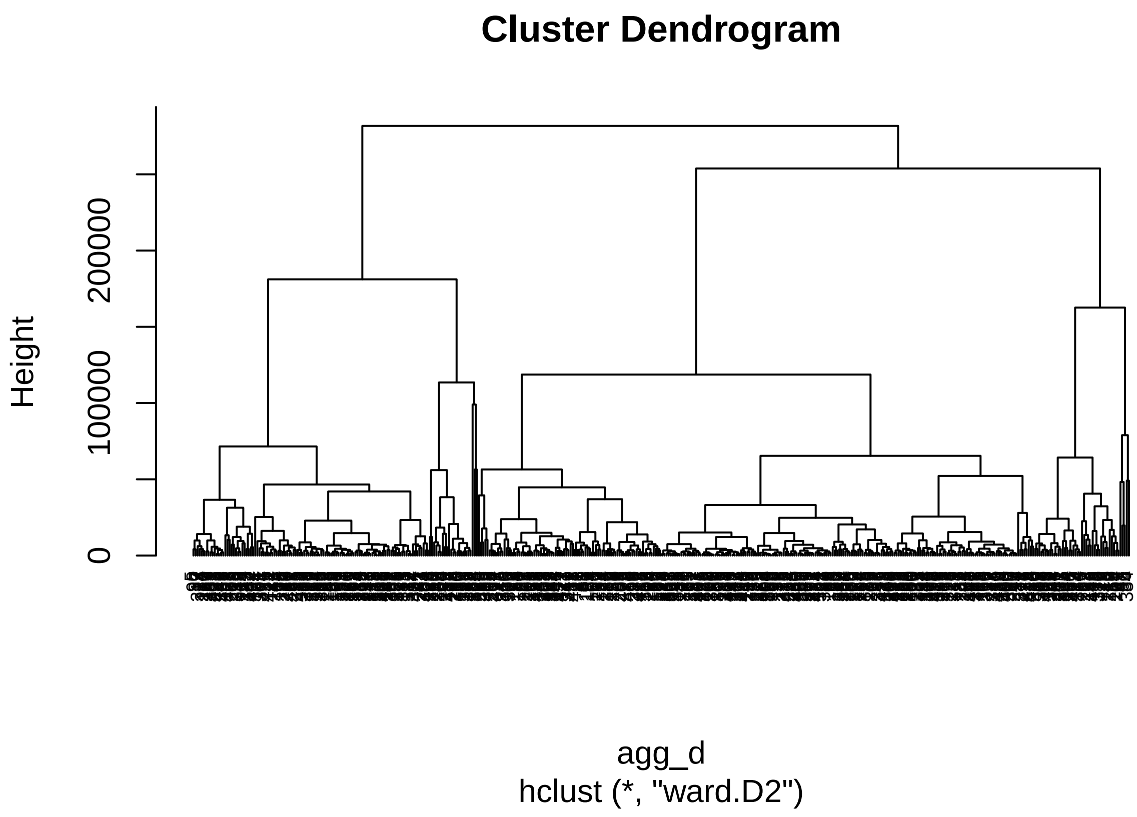
```
## hc_complete_agg_fit
##   1   2   3
## 431   6   3
```

```
plot(hc_complete_agg)
rect.hclust(hc_complete_agg, k=3) #addes cluster split
```



```
hc_wd2 <- hclust(agg_d, method = 'ward.D2')

plot(hc_wd2, cex =.6, hang = -1)
```

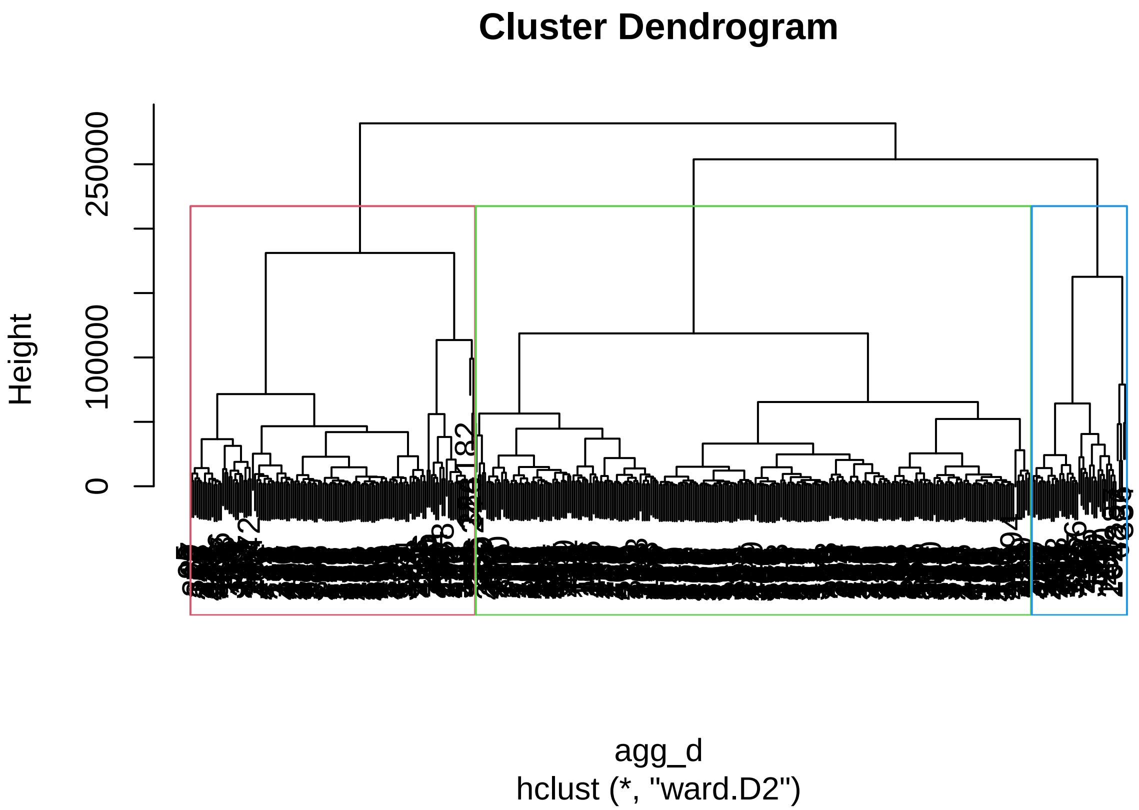
Ward-1.



```
hc_wd2_fit <- cutree(hc_wd2, k = 3)


table(hc_wd2_fit)
```

```
## hc_wd2_fit
##   1   2   3
## 261 134  45
```

```
plot(hc_wd2)
rect.hclust(hc_wd2, k = 3, border = 2:5)
```

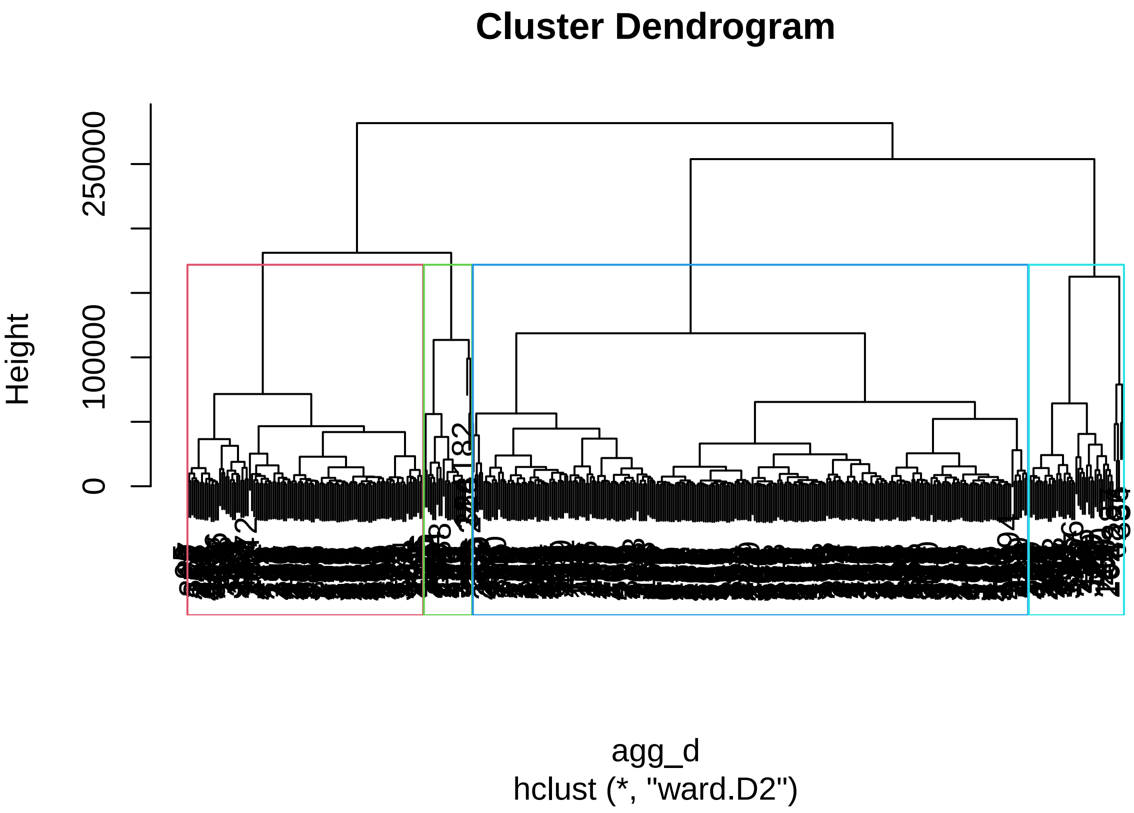


```
hc_wd2_new_fit <- cutree(hc_wd2, h = 175000) #splits hca based off height
table(hc_wd2_new_fit)
```

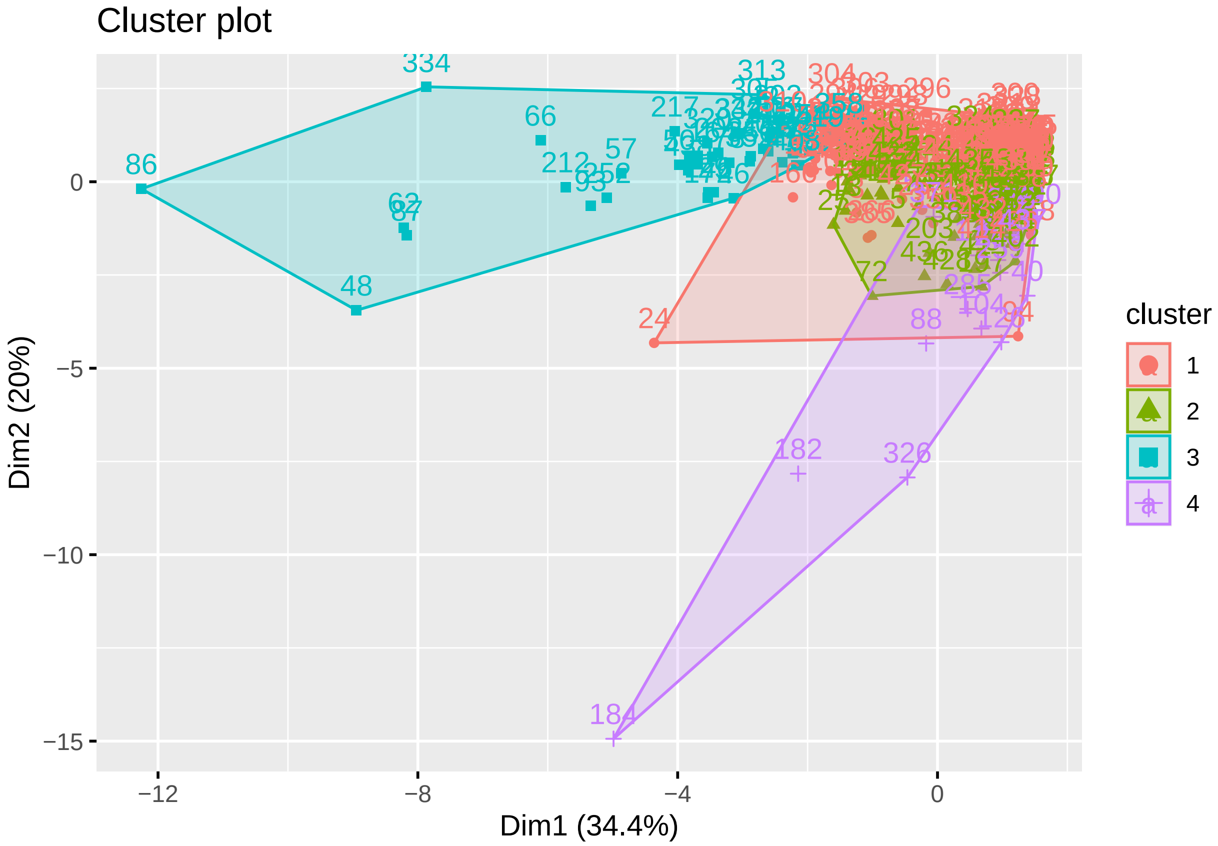
Ward-2.

```
## hc_wd2_new_fit
##   1   2   3   4
## 261 111  45  23
```

```
plot(hc_wd2)

rect.hclust(hc_wd2, h = 175000, border = 2:5)
```

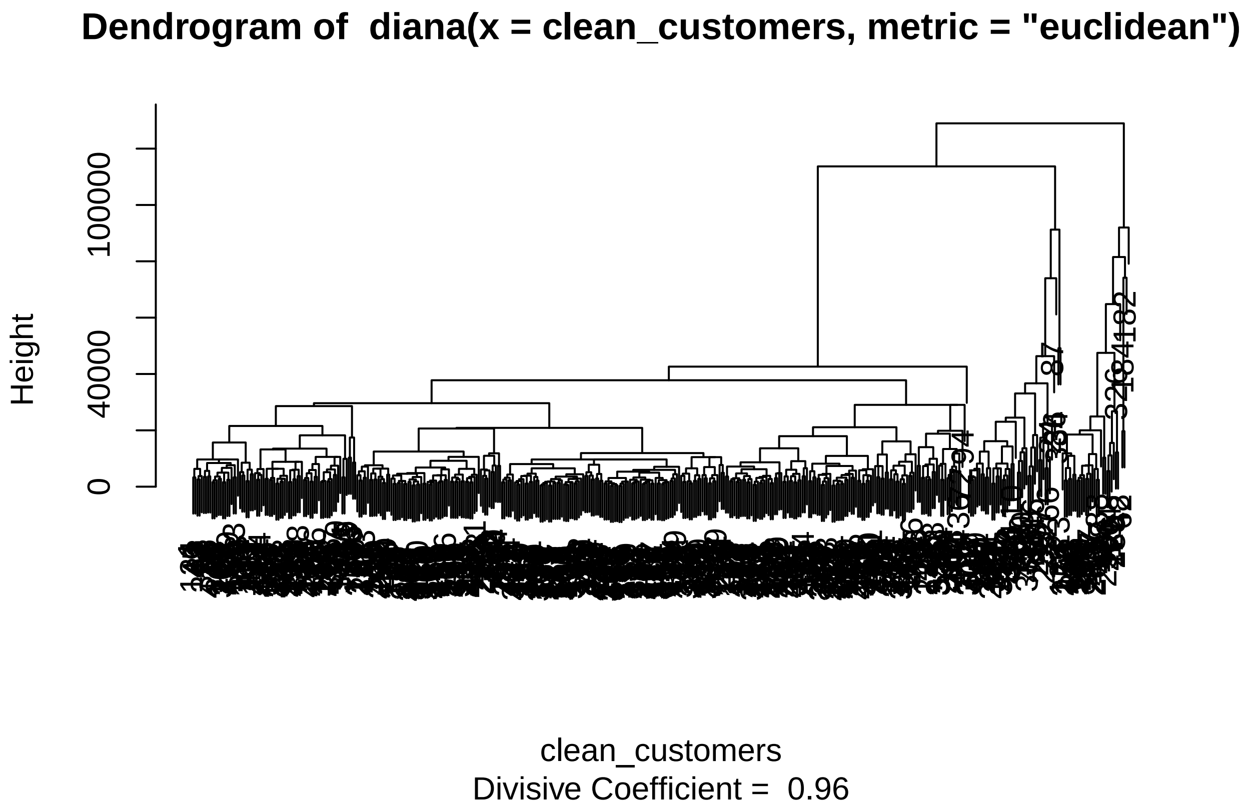


```
fviz_cluster(list(data = clean_customers[,c(1:9)],

                  cluster = hc_wd2_new_fit)) #provides cluster plot of clusters
```



Divisive

```
div_d <- diana(clean_customers, metric = 'euclidean') #divisive hca


plot(div_d, which.plots = 2)
```



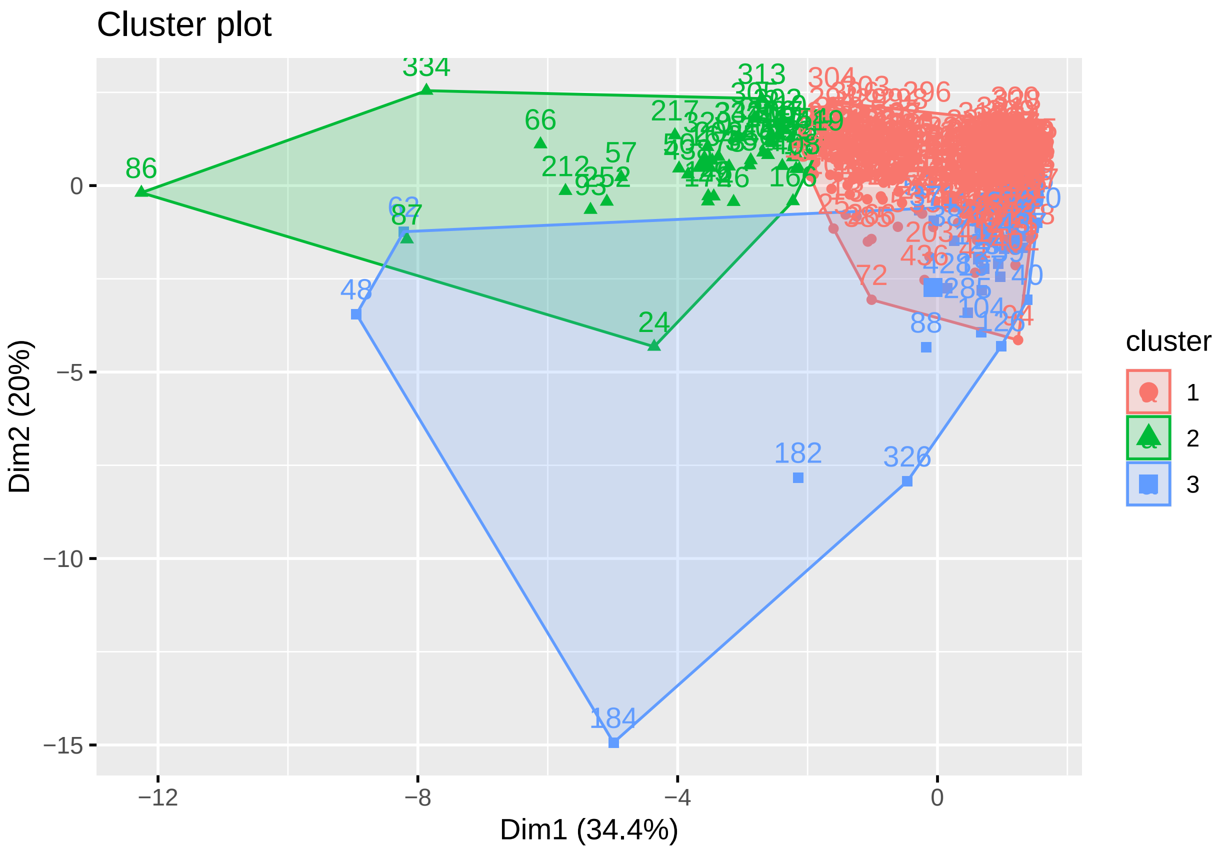
```
div_cut <- cutree(div_d, k=3)
table(div_cut)
```

```
## div_cut
##   1   2   3
## 364  44  32
```

```
div_d$dc
```

```
## [1] 0.9633628
```

```r
fviz_cluster(list(data = clean_customers[,c(1:9)], cluster = div_cut))
```



HCA Analysis

Conclusion

References