# Wholesale - K-means

*Brittani Wilson*

*June 12, 2019*

```
## Loading required package: tidyverse

## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang

## Registered S3 method overwritten by 'rvest':
##   method            from
##   read_xml.response xml2

## -- Attaching packages ------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.1       v purrr   0.3.2
## v tibble  2.1.1       v dplyr   0.8.0.1
## v tidyr   0.8.3       v stringr 1.4.0
## v readr   1.3.1       v forcats 0.4.0

## -- Conflicts ---------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

## Loading required package: cluster

## Loading required package: factoextra

## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ

## Loading required package: gridExtra

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

## Loading required package: animation

## Loading required package: RColorBrewer

## Loading required package: dendextend

##
## ---------------------
## Welcome to dendextend version 1.12.0
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## Or contact: <tal.galili@gmail.com>
##
```

```
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
```

```
##
## Attaching package: 'dendextend'
```

```
## The following object is masked from 'package:stats':
##
##     cutree
```

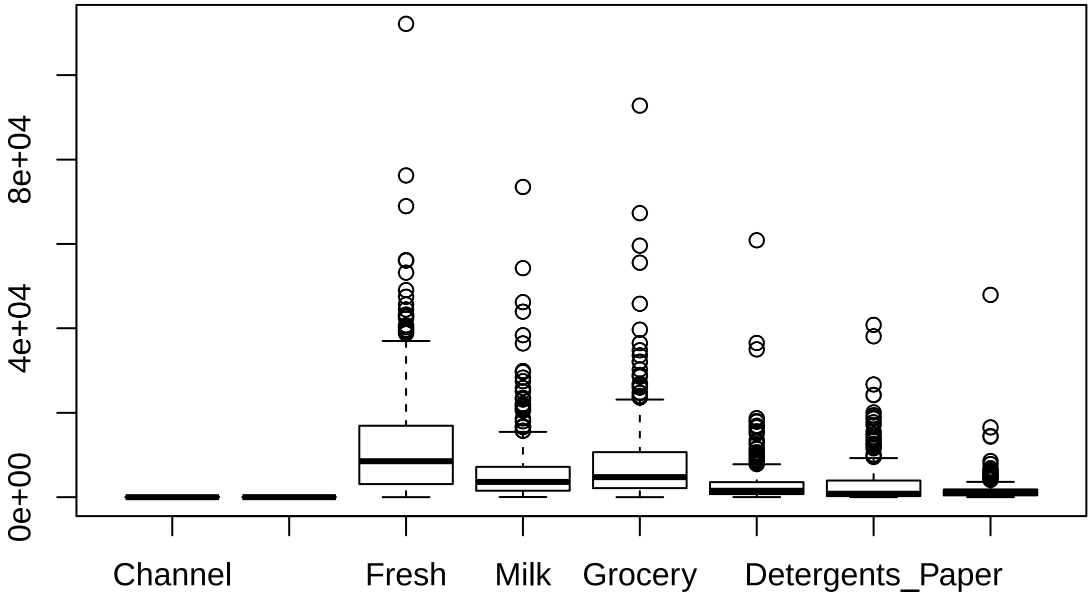
```
##   Channel Region Fresh Milk Grocery Frozen Detergents_Paper Delicassen
## 1       2      3 12669 9656    7561    214             2674       1338
## 2       2      3  7057 9810    9568   1762             3293       1776
## 3       2      3  6353 8808    7684   2405             3516       7844
##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

I made a box plot to cehck for any outliers, and since there are some so I set parameters to leave them out after reviewing histograms for the individual variables. I also made sure to omit any missing values and dropped columns "channel" and "region" since they don't contribute much. I then used scale() to standardize the data frame and set the mean to zero. I then plotted distance matrix using Euclidean distance to check out correlation.

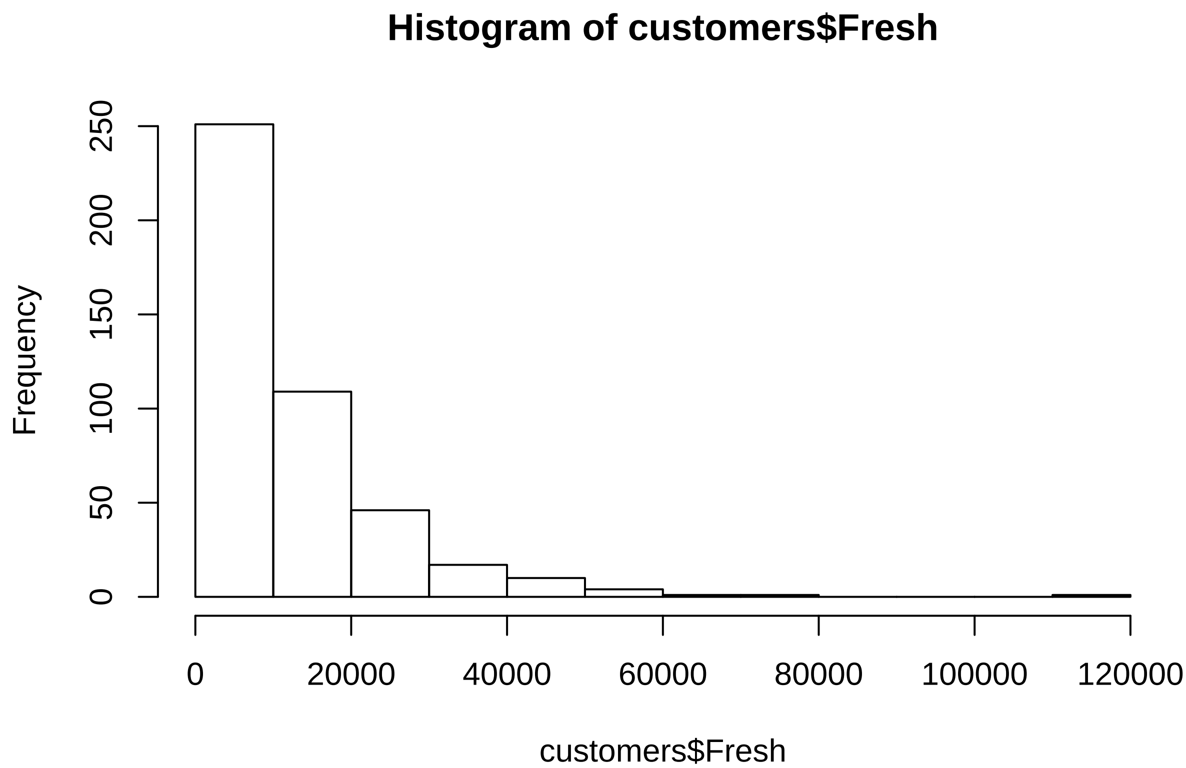
```r
str(customers)
```

```
## '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 ...
```

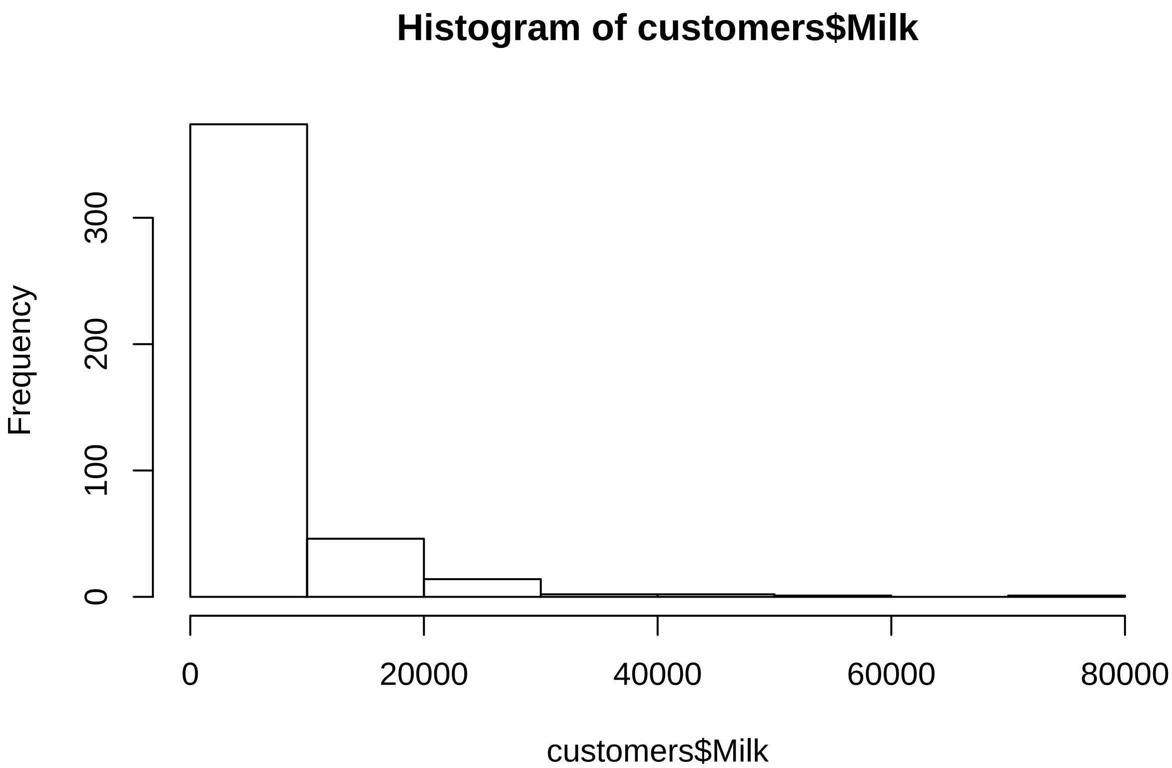
```r
boxplot(customers)
```



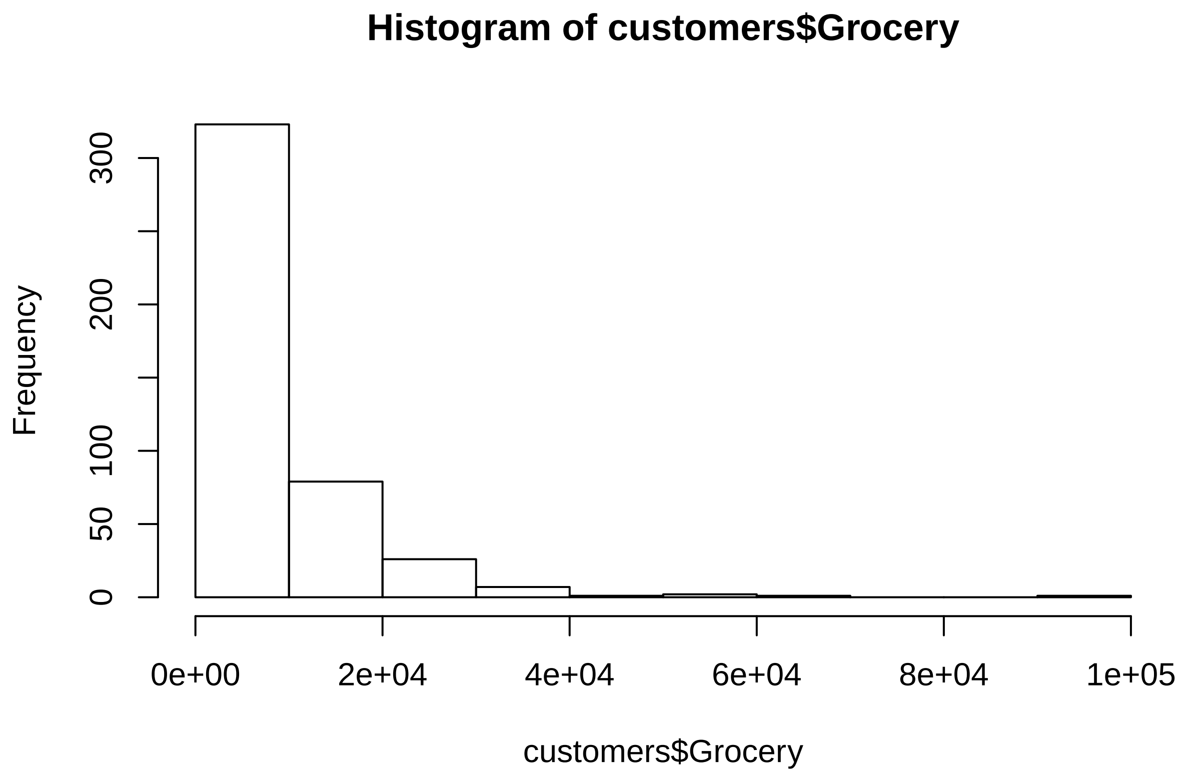

```r
hist(customers$Fresh)
```



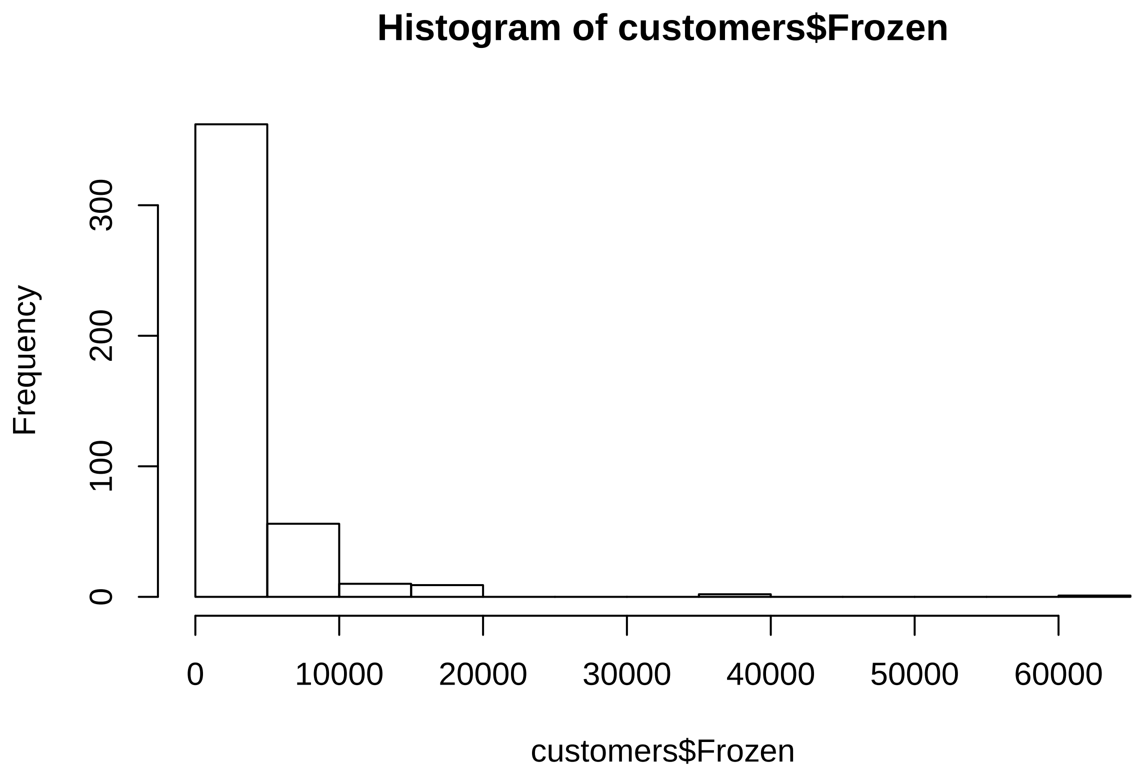

```
hist(customers$Milk)
```



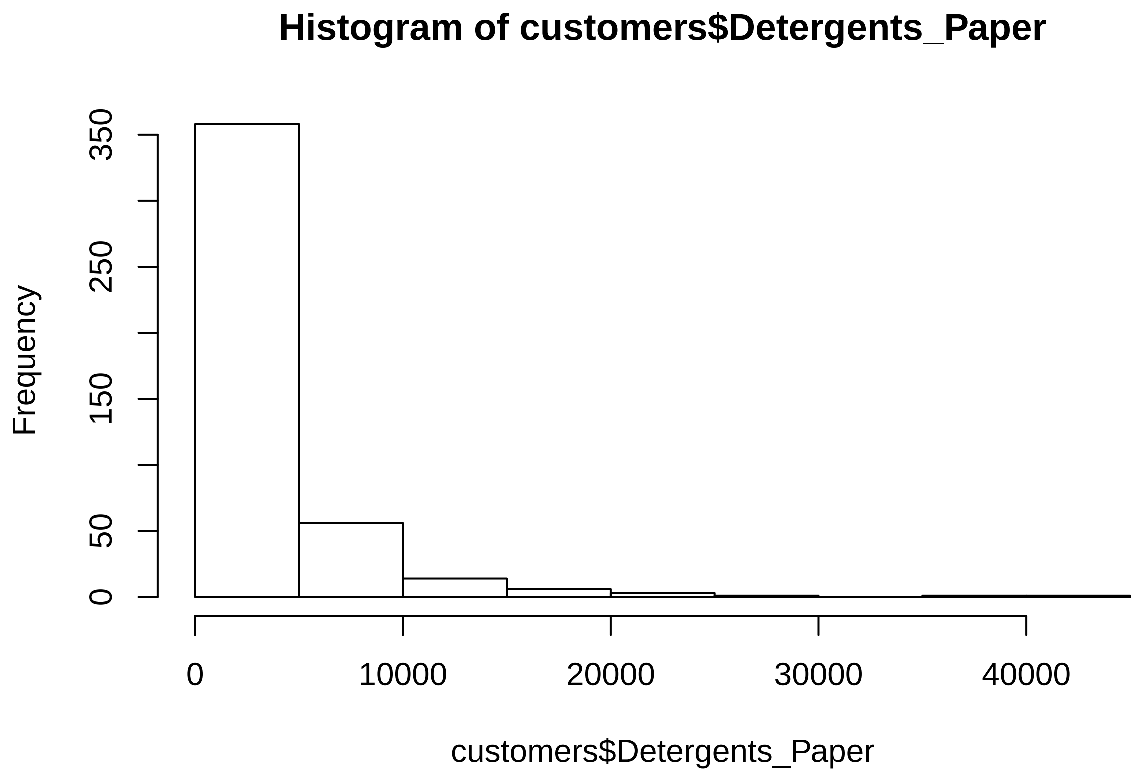

```r
hist(customers$Grocery)
```



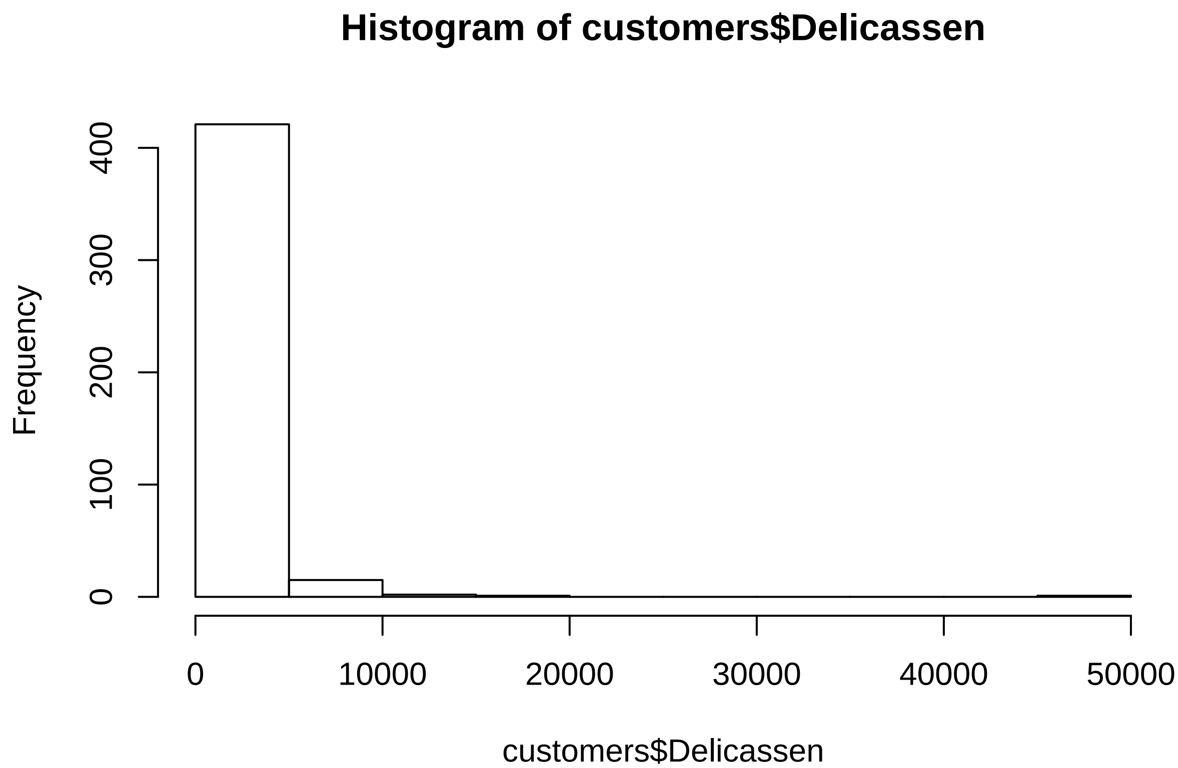

```r
hist(customers$Frozen)
```




```r
hist(customers$Detergents_Paper)
```



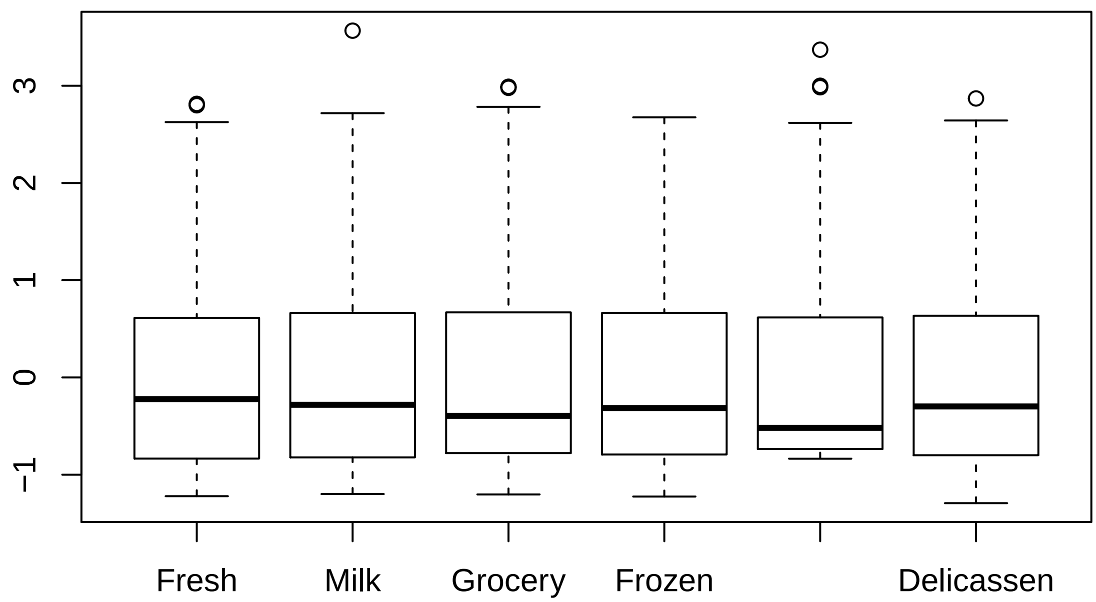

```r
hist(customers$Delicassen)
```




```r
customers2<- subset(customers, Channel & Region & Fresh<30000 &
                        Milk<20000 & Grocery<20000 & Frozen<5000
                      & Detergents_Paper<10000 & Delicassen<3000)


customers3<-customers2[-1:-2]

customers3 <- na.omit(customers3)

customers4 <- scale(customers3)
summary(customers4)
```

```
##      Fresh               Milk              Grocery             Frozen
##  Min.   :-1.2220   Min.   :-1.2000   Min.   :-1.2034   Min.   :-1.2248
##  1st Qu.:-0.8335   1st Qu.:-0.8213   1st Qu.:-0.7793   1st Qu.:-0.7903
##  Median :-0.2245   Median :-0.2815   Median :-0.3969   Median :-0.3169
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.6044   3rd Qu.: 0.6582   3rd Qu.: 0.6671   3rd Qu.: 0.6532
##  Max.   : 2.8139   Max.   : 3.5664   Max.   : 2.9895   Max.   : 2.6749
##  Detergents_Paper    Delicassen
##  Min.   :-0.8354   Min.   :-1.2938
##  1st Qu.:-0.7375   1st Qu.:-0.7963
##  Median :-0.5201   Median :-0.2981
##  Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.6132   3rd Qu.: 0.6326
##  Max.   : 3.3708   Max.   : 2.8691
```

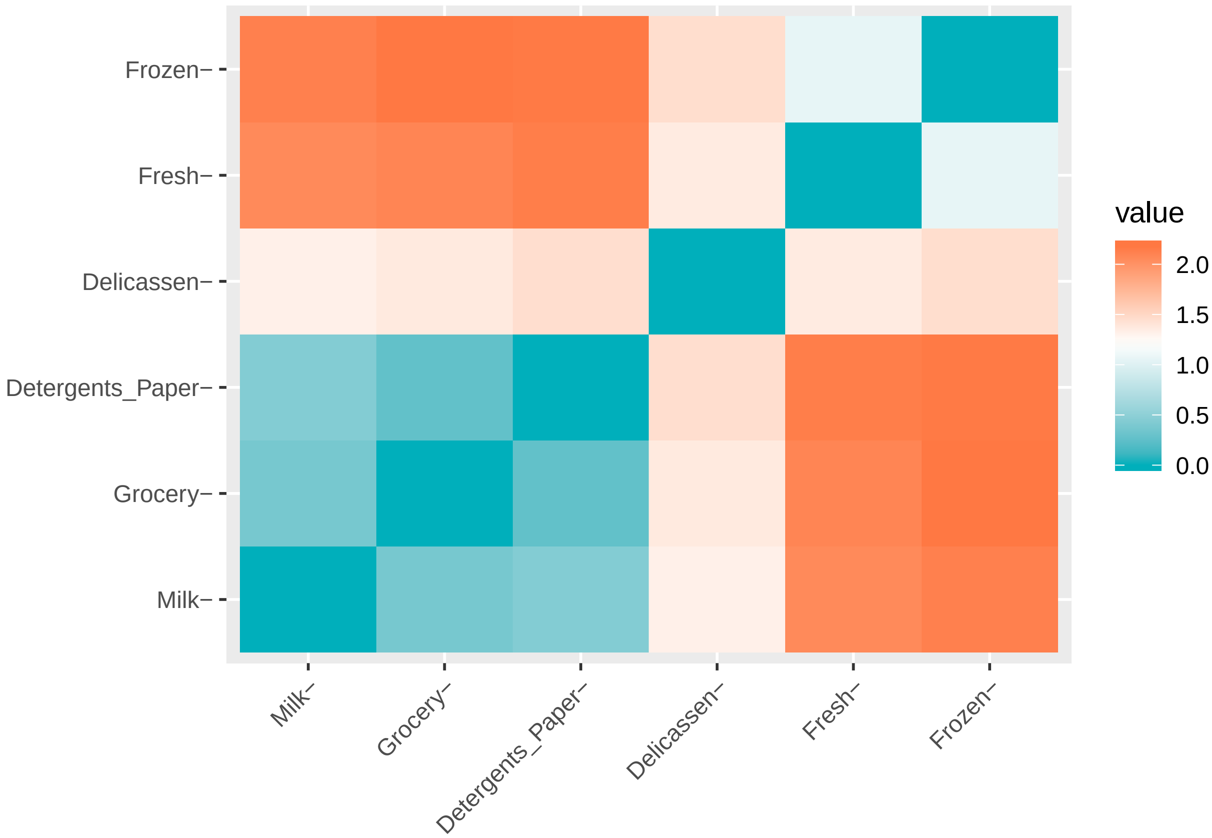
```r
boxplot(customers4)
```




```r
customers_cor<- cor(customers4)
customers_cor
```

```
##                        Fresh       Milk     Grocery      Frozen
## Fresh             1.00000000 -0.1253835 -0.07145758  0.25694093
## Milk             -0.12538349  1.0000000  0.76039355 -0.16488160
## Grocery          -0.07145758  0.7603936  1.00000000 -0.16682531
## Frozen            0.25694093 -0.1648816 -0.16682531  1.00000000
## Detergents_Paper -0.19251717  0.7145396  0.85841788 -0.15581614
## Delicassen        0.15628415  0.2682731  0.27976701  0.09886296
##                  Detergents_Paper Delicassen
## Fresh                  -0.1925172 0.15628415
## Milk                    0.7145396 0.26827313
## Grocery                 0.8584179 0.27976701
## Frozen                 -0.1558161 0.09886296
## Detergents_Paper        1.0000000 0.16623513
## Delicassen              0.1662351 1.00000000
```

```r
distance <- get_dist(customers_cor)
fviz_dist(distance, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))
```

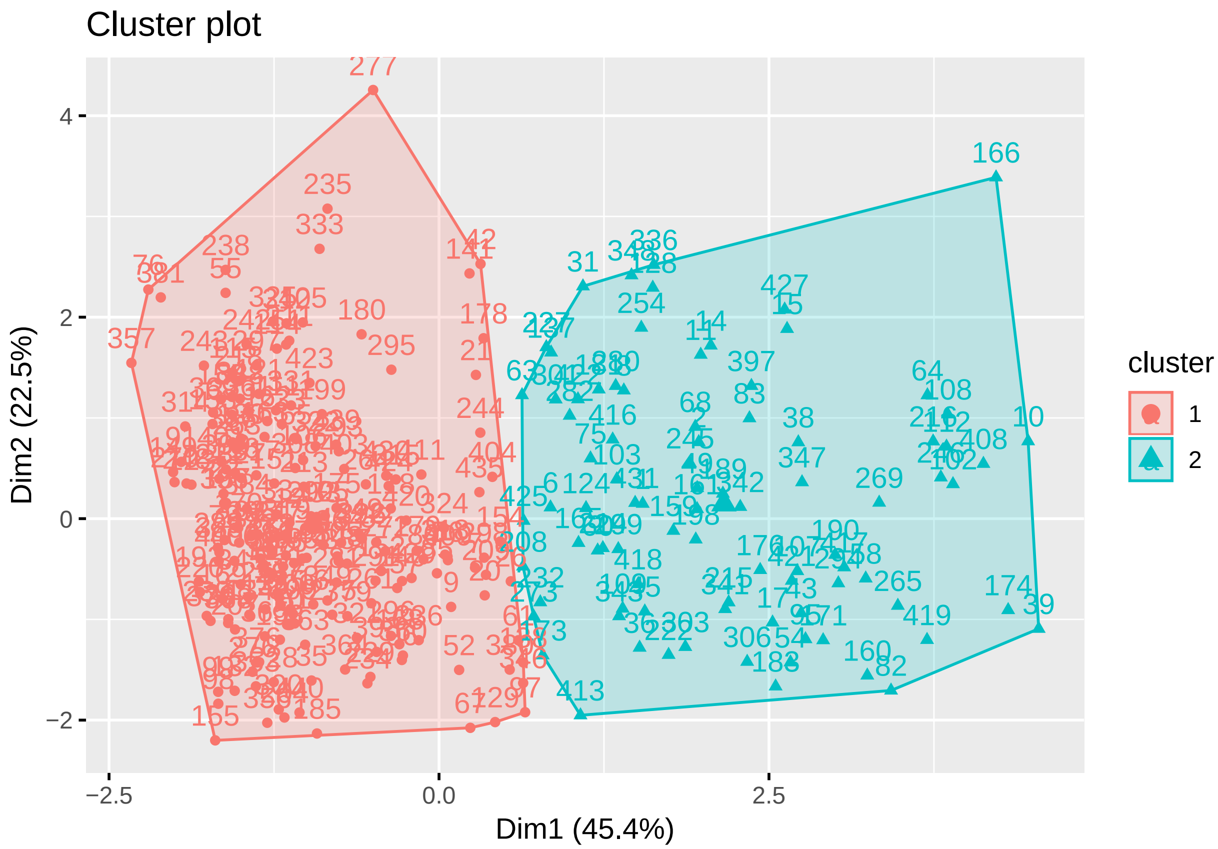



The R software uses 10 as the default value for the maximum number of iterations. An nstart of 25 is recommended and this serves as the number of initial configurations. According to the elbow method 4 looks to be the optimal number of clusters, and 2 maximizes the average silhoutette values for the average silhoutete method. However, the gap statistics recommended 10 clusters.

```
set.seed(34)
k2 <- kmeans(customers4, centers = 2, nstart = 25)
str(k2)
```

```
## List of 9
##  $ cluster     : Named int [1:284] 2 2 2 1 2 1 2 2 1 2 ...
##   ..- attr(*, "names")= chr [1:284] "1" "2" "6" "7" ...
##  $ centers     : num [1:2, 1:6] 0.0872 -0.1975 -0.5173 1.1714 -0.5453 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : chr [1:6] "Fresh" "Milk" "Grocery" "Frozen" ...
##  $ totss       : num 1698
##  $ withinss    : num [1:2] 679 431
##  $ tot.withinss: num 1111
##  $ betweenss   : num 587
##  $ size        : int [1:2] 197 87
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```r
fviz_cluster(k2, data = customers4)
```

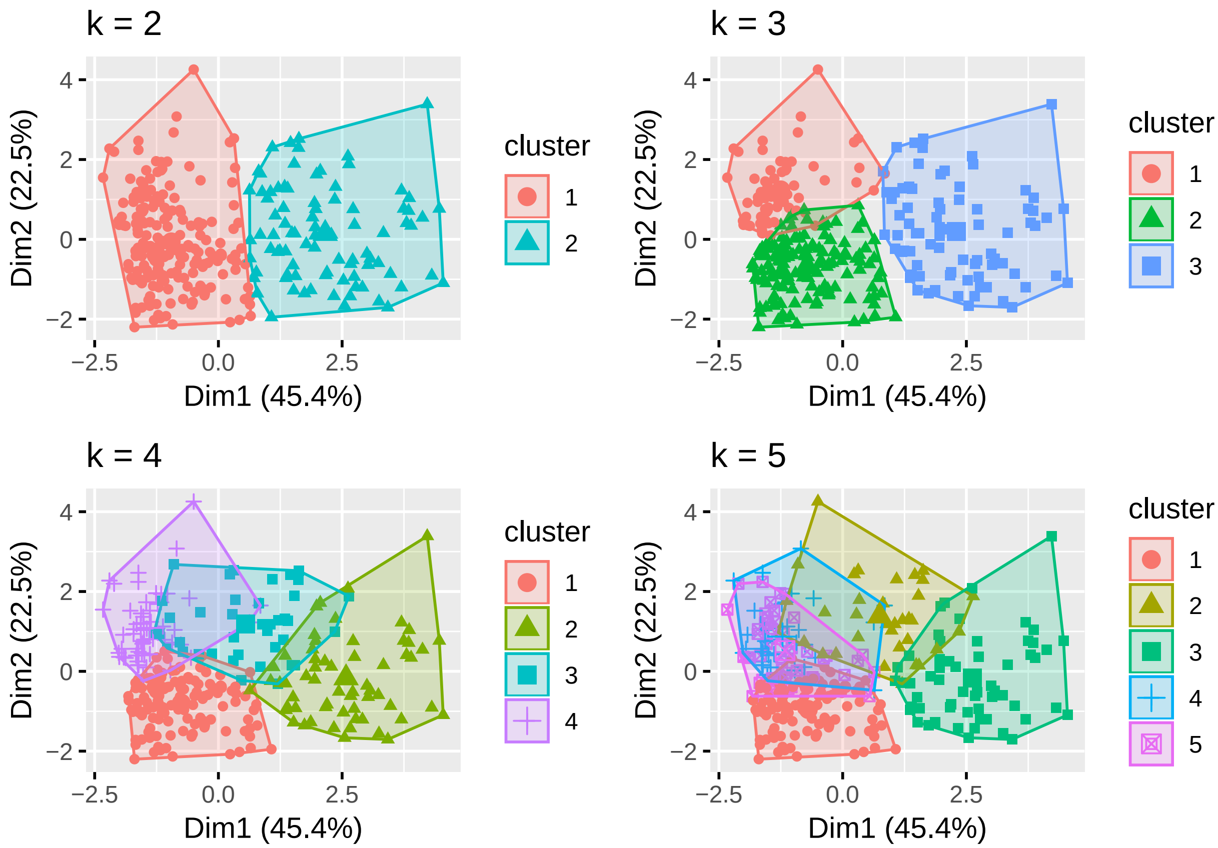



```r
k2
```

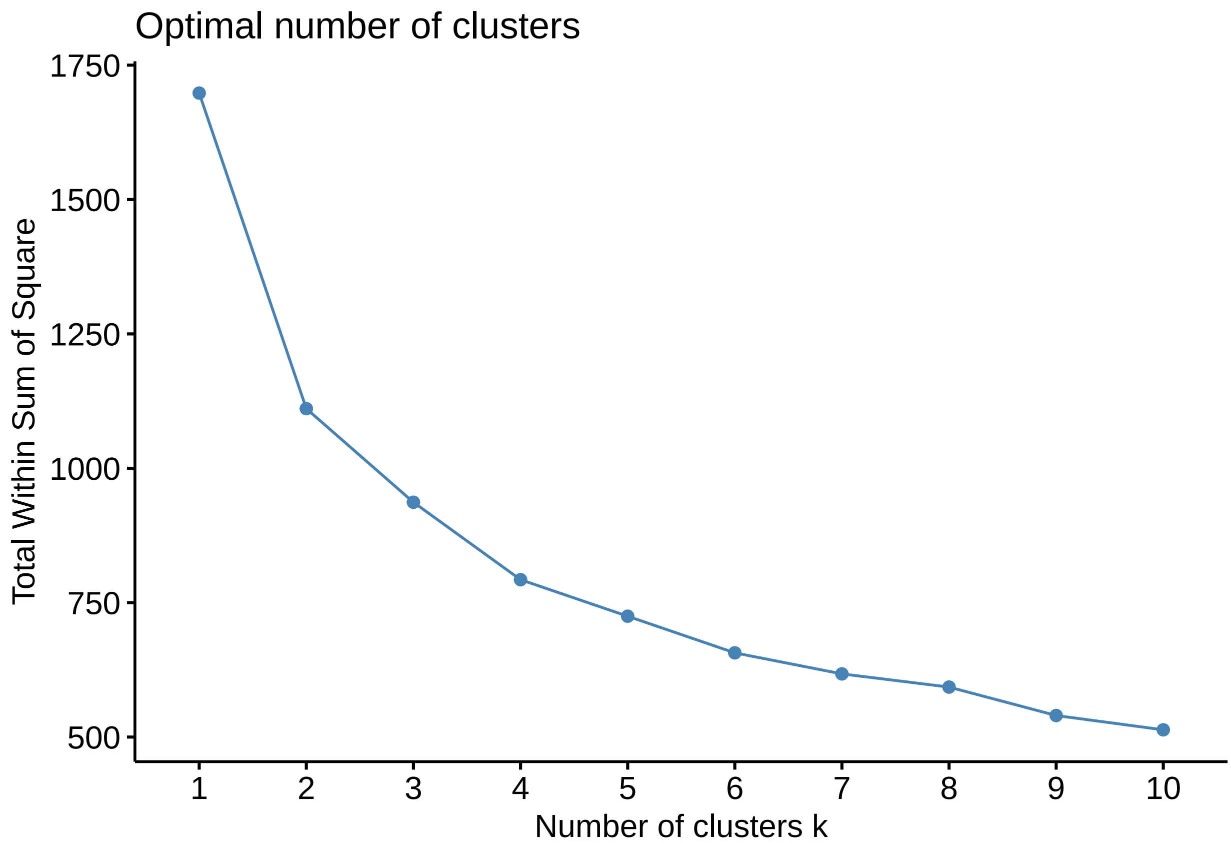
```
## K-means clustering with 2 clusters of sizes 197, 87
##
## Cluster means:
##         Fresh       Milk    Grocery     Frozen Detergents_Paper Delicassen
## 1  0.08724214 -0.5172997 -0.5452735  0.1009782       -0.5328879 -0.2163532
## 2 -0.19754830  1.1713569  1.2346997 -0.2286518        1.2066541  0.4899032
##
## Clustering vector:
##   1   2   6   7   8   9  10  11  12  14  15  16  17  20  21  22  26  27
##   2   2   2   1   2   1   2   2   1   2   2   1   2   1   1   1   1   1
##  28  31  32  33  35  36  38  39  42  43  45  49  51  52  54  55  56  58
##   1   2   1   1   1   2   2   2   1   2   2   2   1   1   2   1   1   2
##  59  60  61  63  64  65  67  68  70  75  76  79  80  81  82  83  84  85
##   1   2   1   2   2   1   1   2   1   2   1   1   1   1   2   2   1   1
##  91  95  96  97  98  99 102 103 105 106 107 108 109 111 112 114 115 116
##   1   2   1   1   1   1   2   2   1   1   2   2   2   1   2   1   1   1
## 117 118 120 121 122 123 124 128 129 132 133 134 135 136 137 138 140 141
##   1   1   1   1   1   1   2   2   1   1   1   1   1   1   2   1   1   1
## 145 147 148 149 151 152 153 154 155 158 159 160 161 162 163 165 166 168
##   1   1   1   1   1   1   1   1   1   1   2   2   2   1   1   2   2   1
## 169 170 171 173 174 175 176 178 179 180 181 183 185 186 187 189 190 192
##   1   1   2   2   2   1   2   1   1   1   2   2   1   1   1   2   2   1
```

```
## 193 195 198 199 200 204 205 207 208 209 211 213 214 215 216 218 220 221
##   1   1   2   1   1   1   1   1   2   1   1   1   2   2   2   1   1   1
## 222 225 226 227 228 229 232 233 234 235 236 237 238 239 242 243 244 245
##   2   1   1   2   1   1   2   1   1   1   1   1   1   1   1   1   1   2
## 246 247 248 249 251 254 257 261 263 264 265 269 270 272 273 275 276 277
##   2   1   1   1   1   2   1   1   1   1   2   2   1   1   2   1   1   1
## 280 281 282 287 289 291 292 293 294 295 296 297 298 299 300 301 303 306
##   2   1   2   1   1   1   1   1   2   1   1   1   1   2   1   2   2   2
## 308 309 312 314 315 317 318 319 321 322 323 324 325 327 328 331 333 336
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   2
## 337 341 342 343 345 346 347 348 349 351 353 356 357 360 361 362 363 364
##   1   2   2   2   1   1   2   2   1   1   1   1   1   1   1   1   1   1
## 365 367 368 369 370 375 376 379 380 381 384 386 387 388 389 390 392 393
##   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1
## 395 396 397 400 401 403 404 405 406 408 409 411 413 416 417 418 419 420
##   1   1   2   1   1   1   1   1   1   2   1   1   2   2   2   2   2   1
## 421 422 423 424 425 427 429 430 431 433 434 435 439 440
##   2   2   1   1   2   2   1   1   2   1   1   1   1   1
##
## Within cluster sum of squares by cluster:
## [1] 679.2310 431.3098
##  (between_SS / total_SS =  34.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

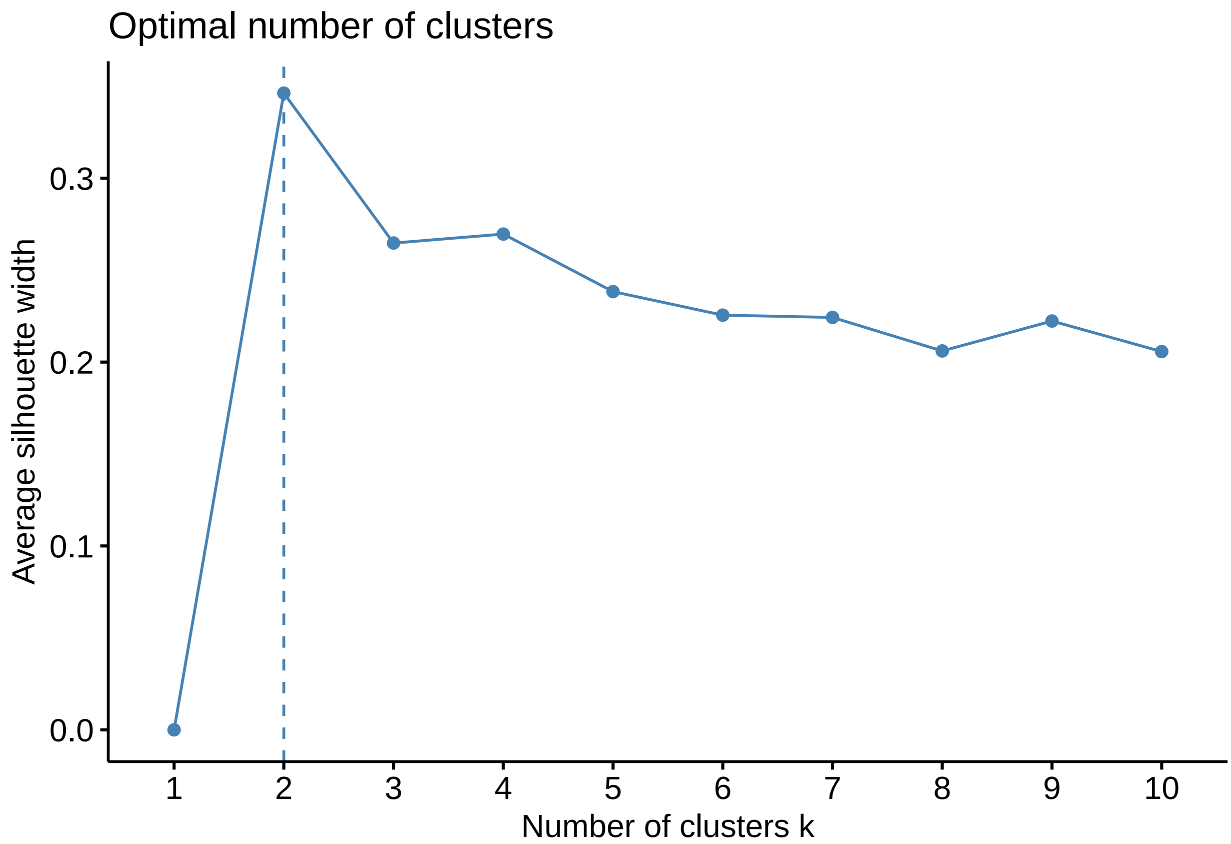
```r
k3 <- kmeans(customers4, centers = 3, nstart = 25)
k4 <- kmeans(customers4, centers = 4, nstart = 25)
k5 <- kmeans(customers4, centers = 5, nstart = 25)

p1 <- fviz_cluster(k2, geom = "point", data = customers4) + ggtitle("k = 2")
p2 <- fviz_cluster(k3, geom = "point",  data = customers4) + ggtitle("k = 3")
p3 <- fviz_cluster(k4, geom = "point",  data = customers4) + ggtitle("k = 4")
p4 <- fviz_cluster(k5, geom = "point",  data = customers4) + ggtitle("k = 5")
grid.arrange(p1, p2, p3, p4, nrow = 2)
```



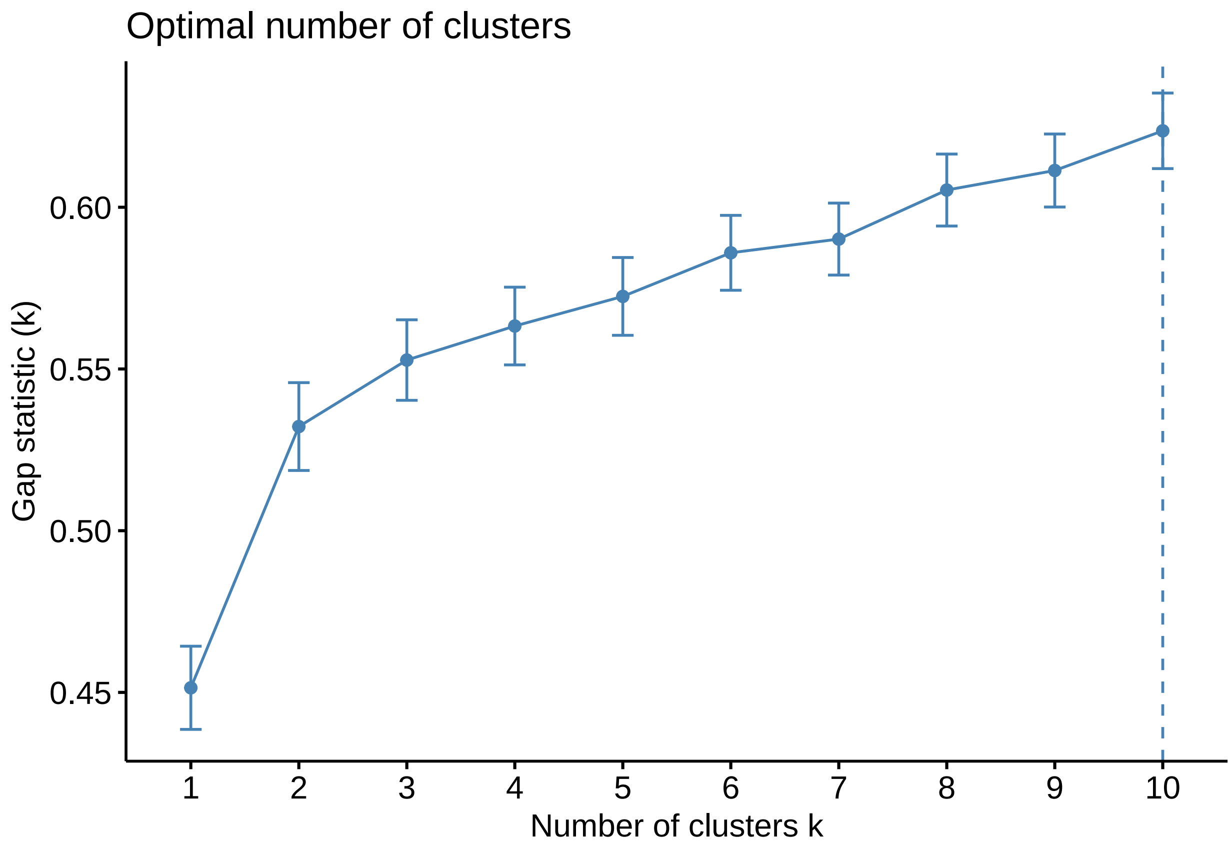

```r
set.seed(34)
fviz_nbclust(customers4, kmeans, method = "wss")
```




```
set.seed(34)
fviz_nbclust(customers4, kmeans, method = "silhouette")
```




```
set.seed(34)
gap_stat <- clusGap(customers4, FUN = kmeans, nstart = 25,
                    K.max = 10, B = 50)
fviz_gap_stat(gap_stat)
```

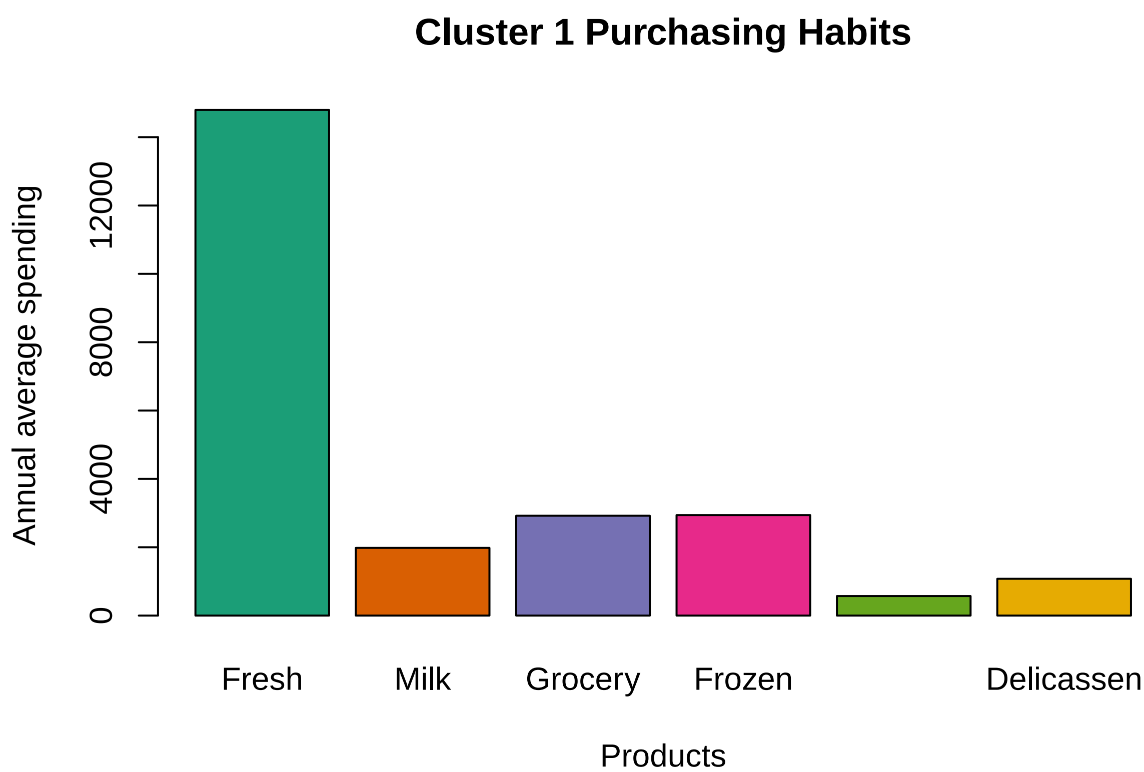



Let's go with a cluster of 3. We can see from the product preferences in each of our three clusters that cluster 1 prefers to mainly buy fresh foods, and while this is a favorite of cluster 2 they tend to also buy more in the milk and grocery department. Cluster 3 tends to buy more of a mixture, the least in the fresh department, and the most of Detergents_Paper(the missing named value).

```
set.seed(34)
jBrewColors <- brewer.pal(n = 8, name = "Dark2")
cluster1 <- (customers3[k3$cluster==1,])
cluster1_avg <- (sapply(cluster1, mean, na.rm=TRUE))
cluster1_avg
```

```
##            Fresh             Milk          Grocery           Frozen
##       14796.6364        1981.1061        2921.2424        2941.2879
## Detergents_Paper       Delicassen
##         572.1061        1076.8939
```

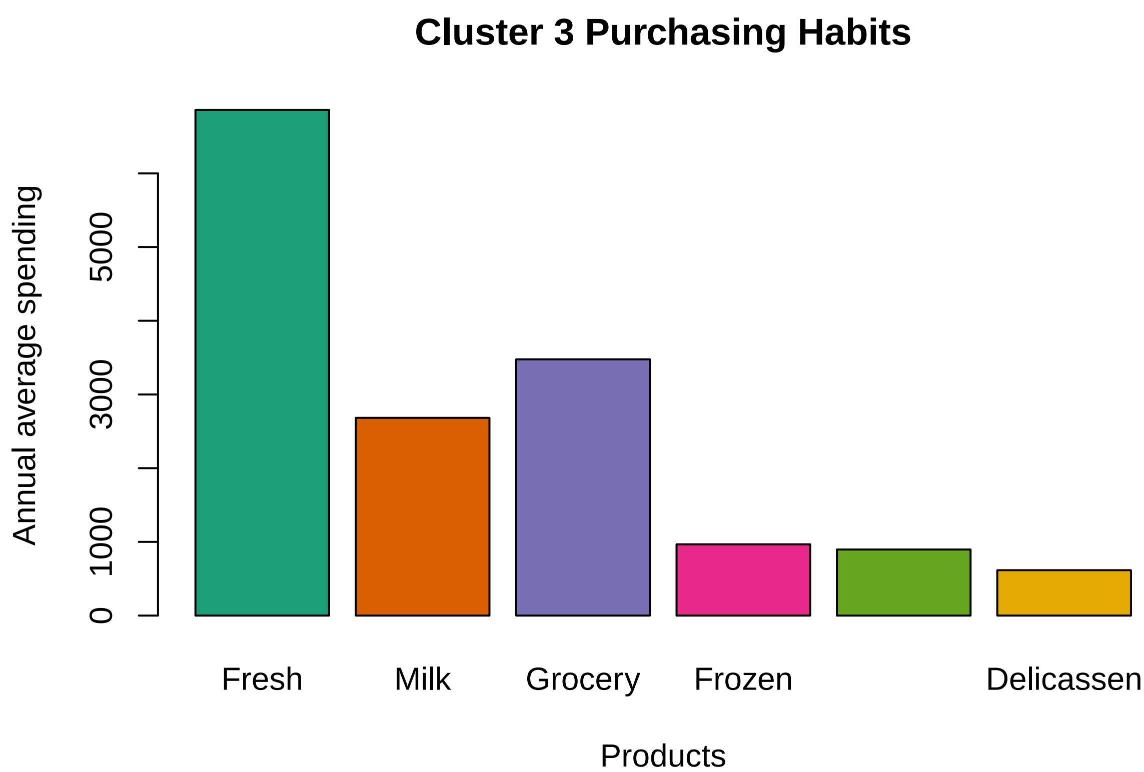
```
barplot(cluster1_avg, main="Cluster 1 Purchasing Habits", xlab="Products",ylab="Annual average spending
```




```r
cluster2 <- (customers3[k3$cluster==2,])
cluster2_avg <- (sapply(cluster2, mean, na.rm=TRUE))
cluster2_avg
```

```
##            Fresh             Milk          Grocery           Frozen
##        6860.1942        2682.6115        3476.9065         967.0863
## Detergents_Paper       Delicassen
##         896.8417         615.9281
```

```r
barplot(cluster2_avg, main="Cluster 3 Purchasing Habits", xlab="Products",ylab="Annual average spending
```




```
cluster3 <- (customers3[k3$cluster==3,])
cluster3_avg <- (sapply(cluster3, mean, na.rm=TRUE))
cluster3_avg
```

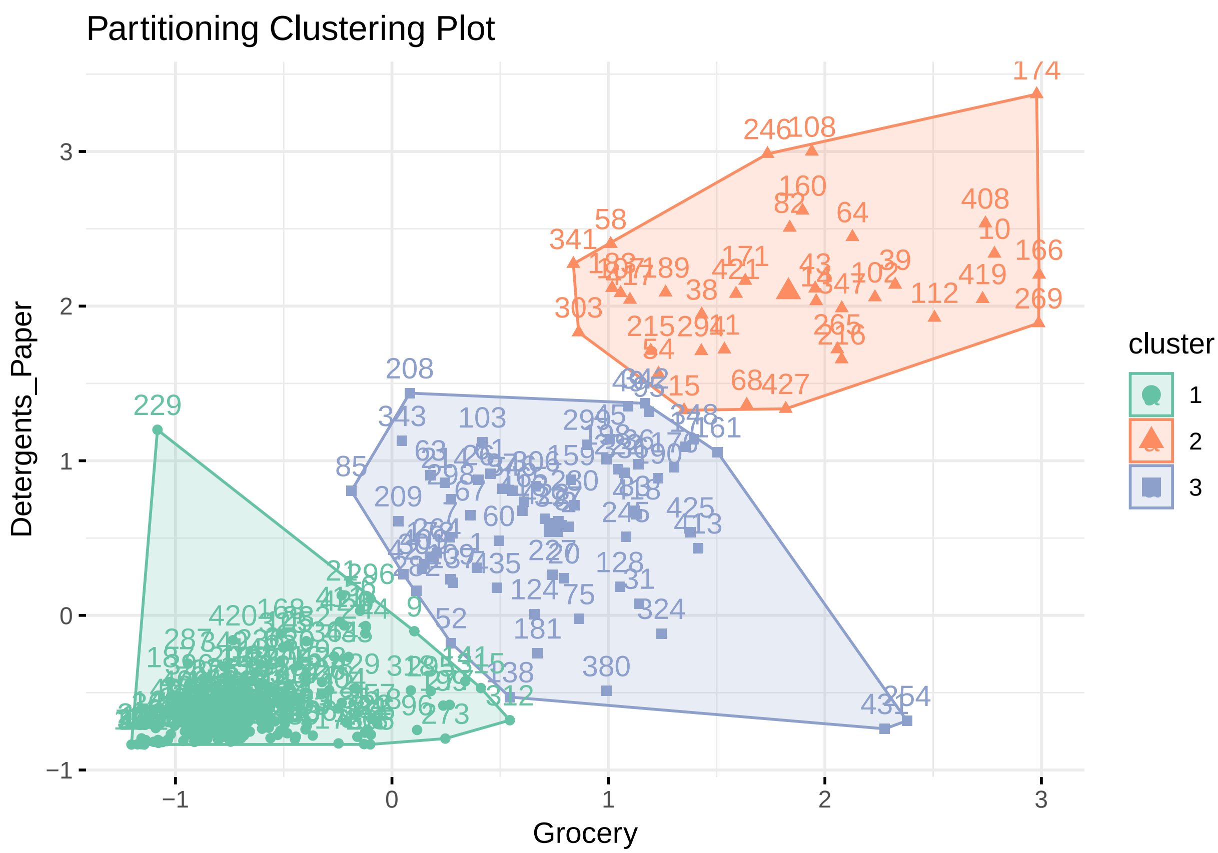
```
##          Fresh             Milk          Grocery           Frozen
##       7832.443         8008.747        11939.101         1134.835
## Detergents_Paper       Delicassen
##       4975.456         1334.532
```

```
barplot(cluster3_avg, main="Cluster 3 Purchasing Habits", xlab="Products",ylab="Annual average spending
```


We can tell in the comparision between grocery and detergents_paper that customers in cluter 2 purchase most of these items while customers in cluster 1 don't purchase these items.

While looking fresh and frozen plot we can see that while those in cluster 1 purchase the most in frozen they also purchase the least in fresh. Those in cluster 2 don't purchase from either.

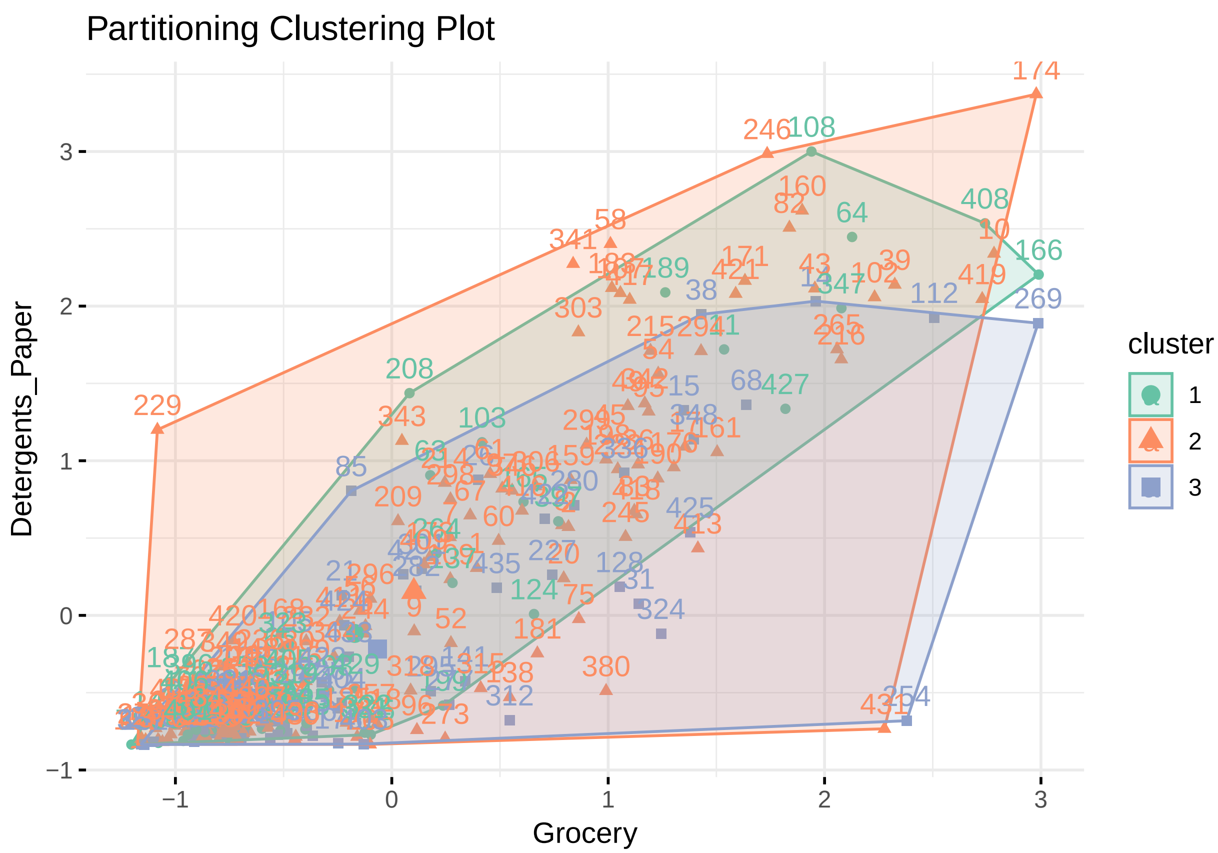
```
set.seed(34)

customers4.subset<-as.data.frame(customers4[,c("Grocery","Detergents_Paper")])

customers5 = kmeans(customers4.subset, centers = 3, nstart = 25)

fviz_cluster(customers5, customers4.subset[, -5],
   palette = "Set2", ggtheme = theme_minimal(), main = "Partitioning Clustering Plot")
```




```
customers5$centers
```

```
##      Grocery Detergents_Paper
## 1 -0.6323915       -0.6200515
## 2  1.8314249        2.0926637
## 3  0.7449084        0.5683860
```

```
customers6.subset<-as.data.frame(customers4[,c("Frozen","Fresh")])

customers7 = kmeans(customers6.subset, centers = 3, nstart = 10)

fviz_cluster(customers7, customers4.subset[, -5],
   palette = "Set2", ggtheme = theme_minimal(), main = "Partitioning Clustering Plot")
```




```
customers7$centers
```

```
##       Frozen       Fresh
## 1  1.4486538 -0.07722363
## 2 -0.5966744 -0.61758971
## 3  0.0155243  1.33354928
```

In building the dendogram we find that “ward” seems to be the best while “complete” is the second best linkage method. We cut the tree to find subgroups, and this is similar to finding the K in the k-means analysis. The color lines surrounding is to better define the borders for each cluster, and you can see this change as the number of clusters change with each cut.

We can see that Fresh, Frozen, and Delicatessen are the most similar to each other, but the most dissimlar to Milk, Grocery, and Detergent_Paper.

```
set.seed(34)
customers_cor.d <- dist(customers_cor, method = "euclidean")
hc2 <- agnes(customers_cor.d, method = "complete")
hc2$ac
```

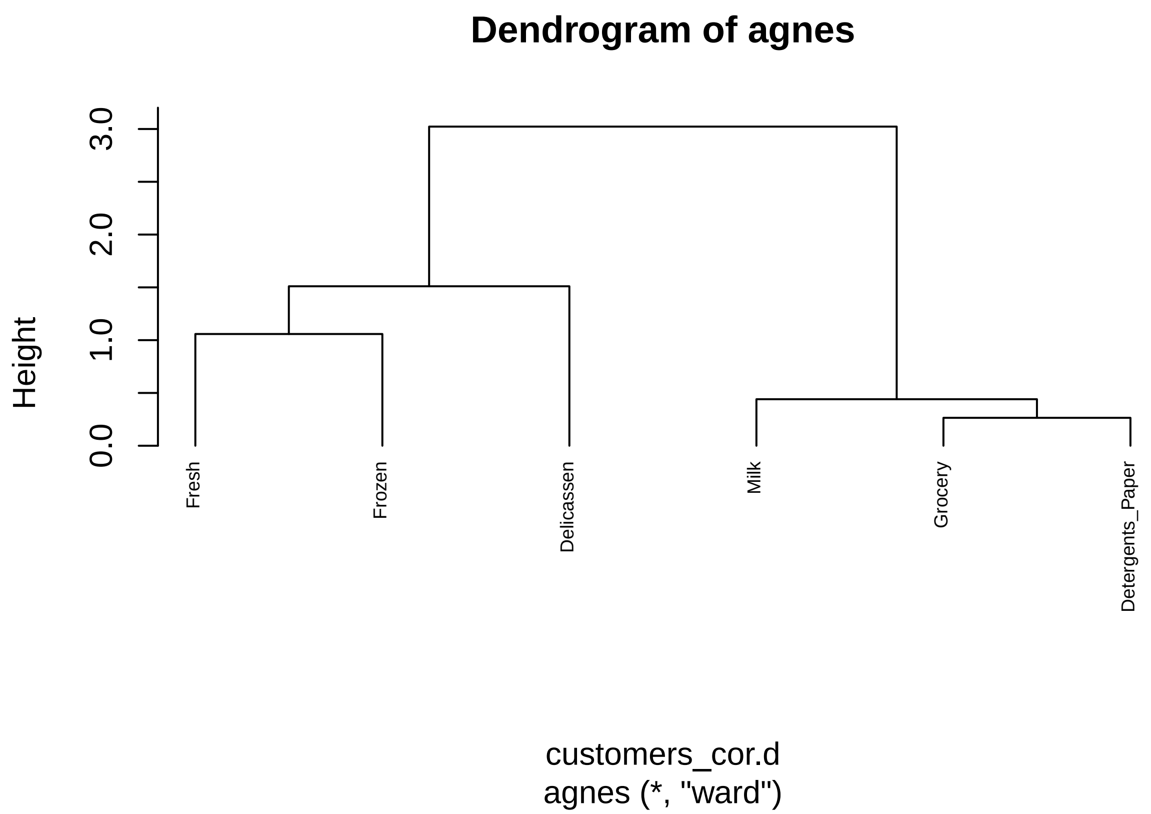
```
## [1] 0.6534604
```

```
m <- c( "average", "single", "complete", "ward")
names(m) <- c( "average", "single", "complete", "ward")

ac <- function(x) {
  agnes(customers_cor.d, method = x)$ac
}
```

```r
map_dbl(m, ac)
```

```
##   average    single  complete      ward
## 0.6209483 0.4689597 0.6534604 0.7465921
```

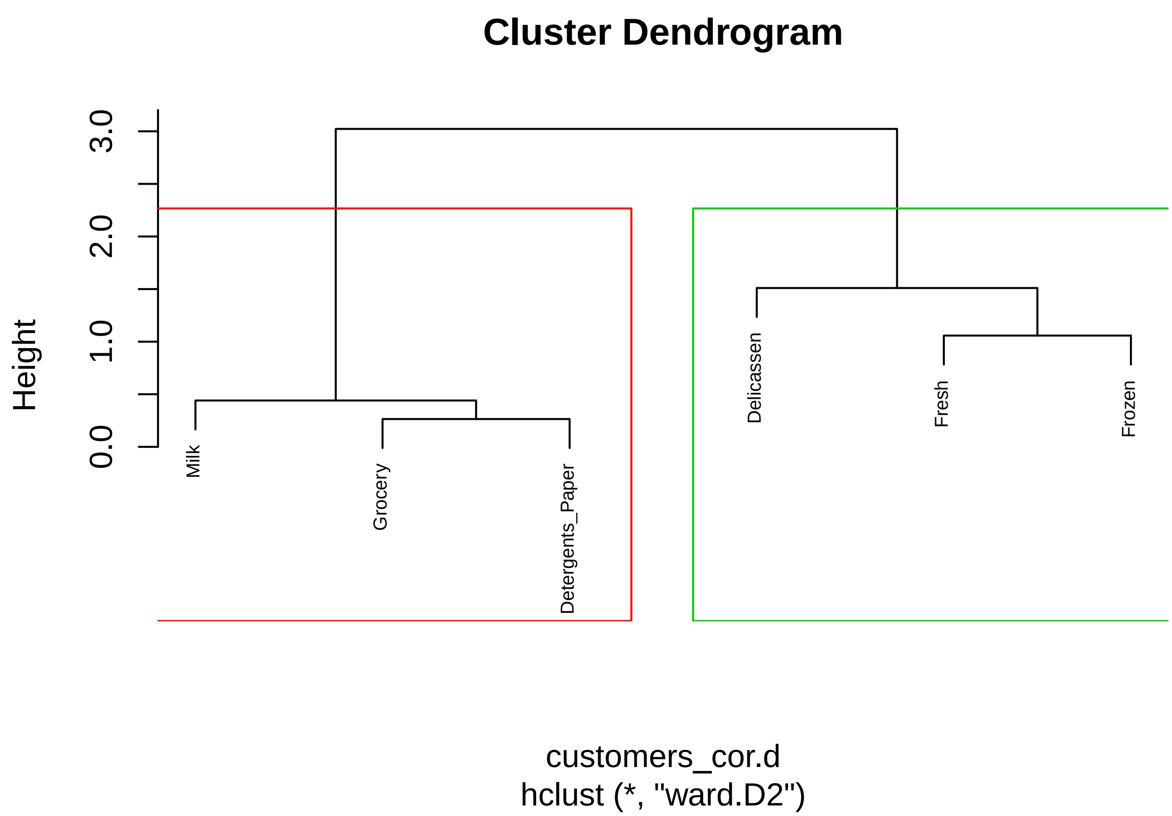
```r
hc3 <- agnes(customers_cor.d, method = "ward")
pltree(hc3, cex = 0.6, hang = -1, main = "Dendrogram of agnes")
```




```r
hc5 <- hclust(customers_cor.d, method = "ward.D2" )

sub_grp2 <- cutree(hc5, k = 2)
table(sub_grp2)
```

```
## sub_grp2
## 1 2
## 3 3
```

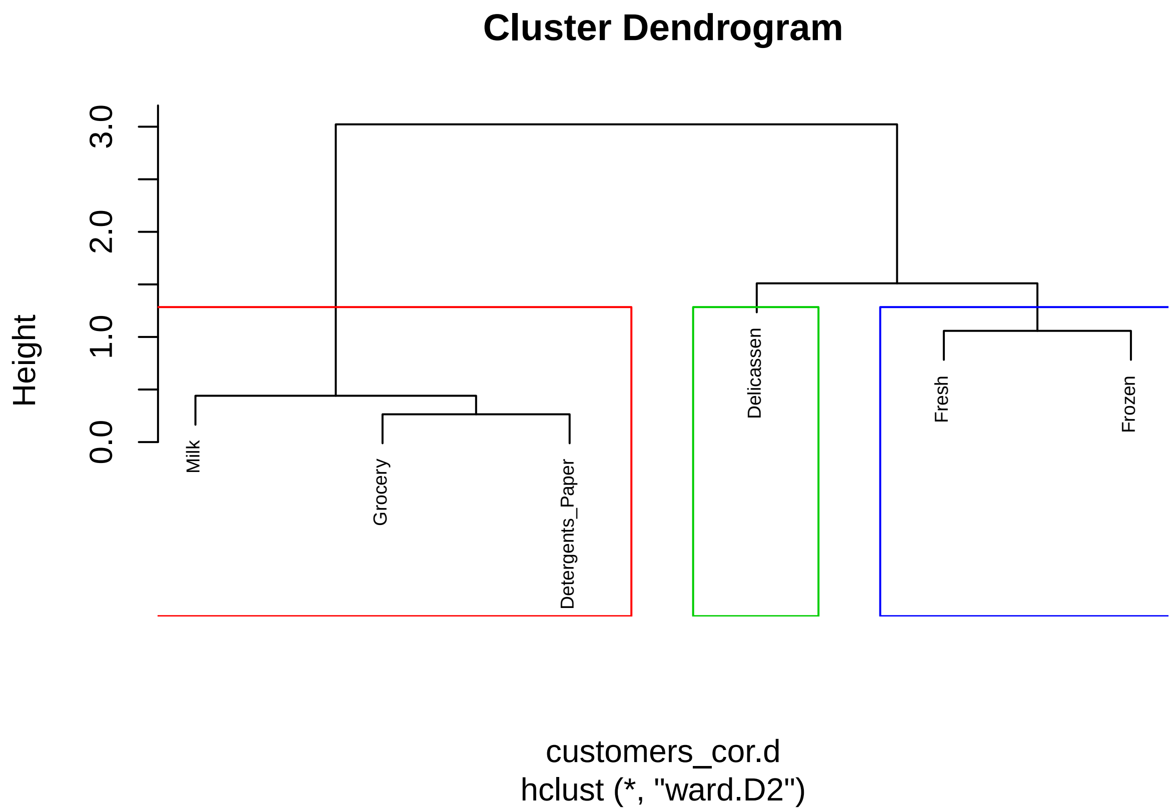
```r
plot(hc5, cex = 0.6)
rect.hclust(hc5, k = 2, border = 2:5)
```




```r
sub_grp3 <- cutree(hc5, k = 3)
table(sub_grp3)
```

```
## sub_grp3
## 1 2 3
## 2 3 1
```

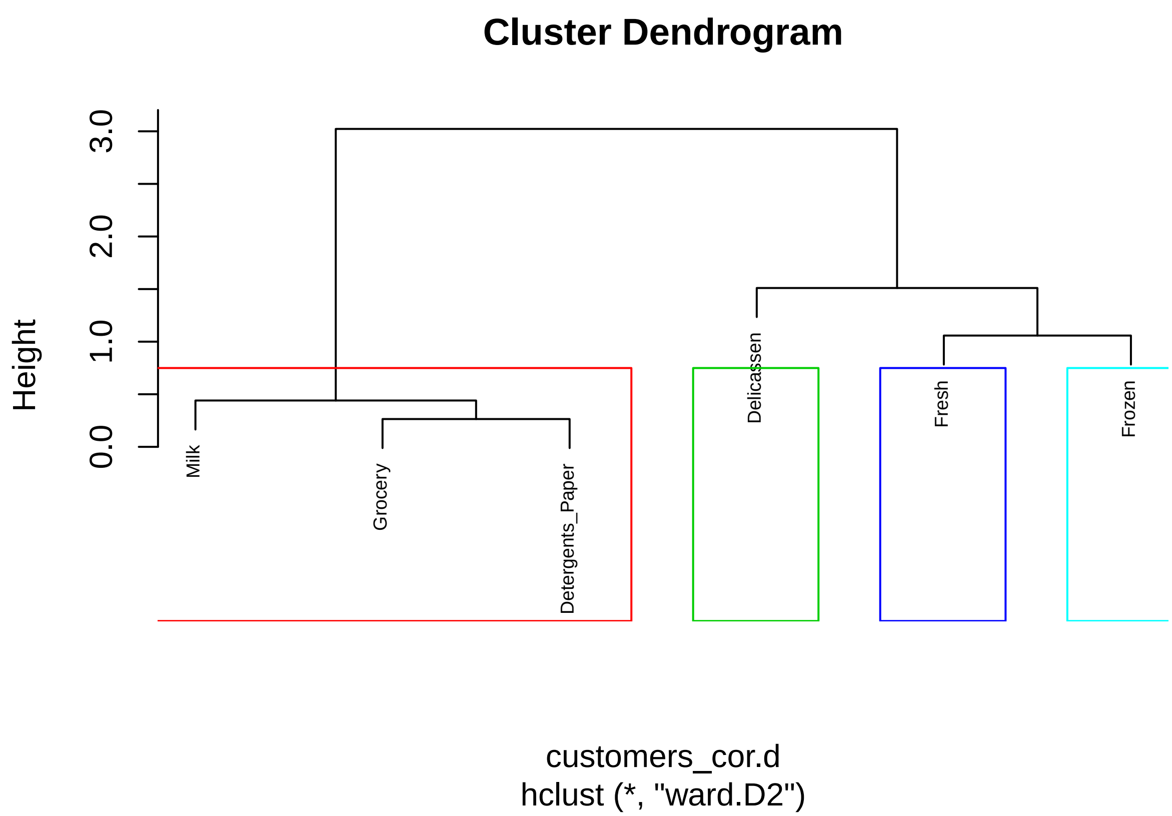
```r
plot(hc5, cex = 0.6)
rect.hclust(hc5, k = 3, border = 2:5)
```




```r
sub_grp4 <- cutree(hc5, k = 4)
table(sub_grp4)
```

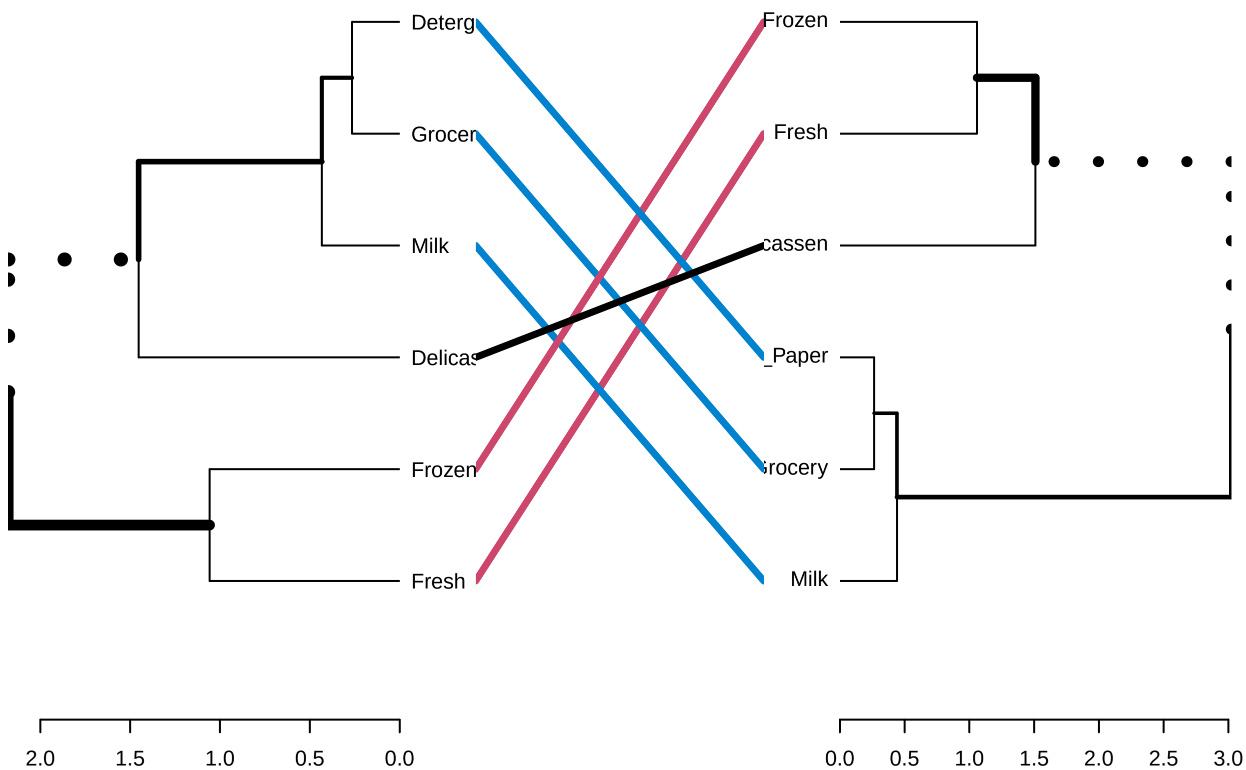
```
## sub_grp4
## 1 2 3 4
## 1 3 1 1
```

```r
plot(hc5, cex = 0.6)
rect.hclust(hc5, k = 4, border = 2:5)
```




Now we compare the 2 linkage methods "complete" and "ward" by measuring their entaglement, which is a measure between 1 (full entanglement) and 0 (no entanglement). A lower entanglement coefficient means there is a good alignment.

```
res.dist <- dist(customers_cor, method = "euclidean")
hc11 <- hclust(res.dist, method = "complete")
hc22 <- hclust(res.dist, method = "ward.D2")
dend1 <- as.dendrogram (hc11)
dend2 <- as.dendrogram (hc22)

tanglegram(dend1, dend2)
```


Deterg
Grocer
Milk
Delicas
Frozen
Fresh
Frozen
Fresh
cassen
_Paper
rocery
Milk
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0