# Clustering Analysis on Wholesale Customer (Hierarchical & K-Means)

yumi yu

```
library(dplyr)

library(ggplot2)
wholesale <- read.csv('wholesale_customers_data.csv')
```

### Procedure of Data Analysis:

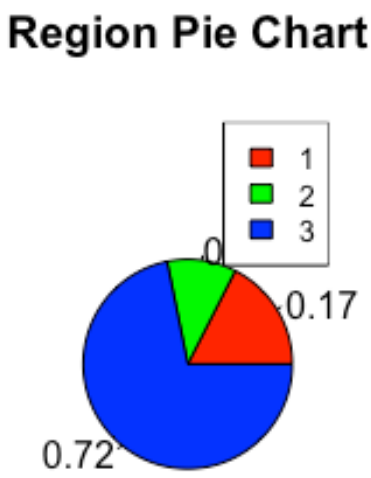
1. EDA before Clustering
2. Hierarchical Clustering
3. K-Means Clustering
4. Evaluate Clustering Solutions: SSE and Slihouette Coefficient
5. Analysis after clustering
6. Summary for clustering and other analysis results

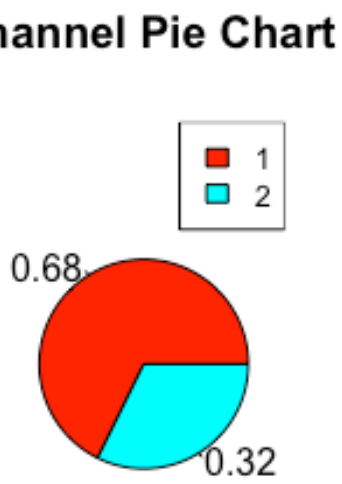
### Variable Name Description

Channel: Client channel ("1" means Horeca (Hotel/Restaurant/Cafe) and "2" means Retail)
Region: Client region ("1" means Lisbon, "2" means Oporto, and "3" means other regions)
Fresh: Annual spending on fresh products. Milk: Annual spending on milk products.
Grocery: Annual spending on grocery products. Frozen: Annual spending on frozen products. Detergents Paper: Annual spending on detergents and paper products.
Delicatessen: Annual spending on deli products.

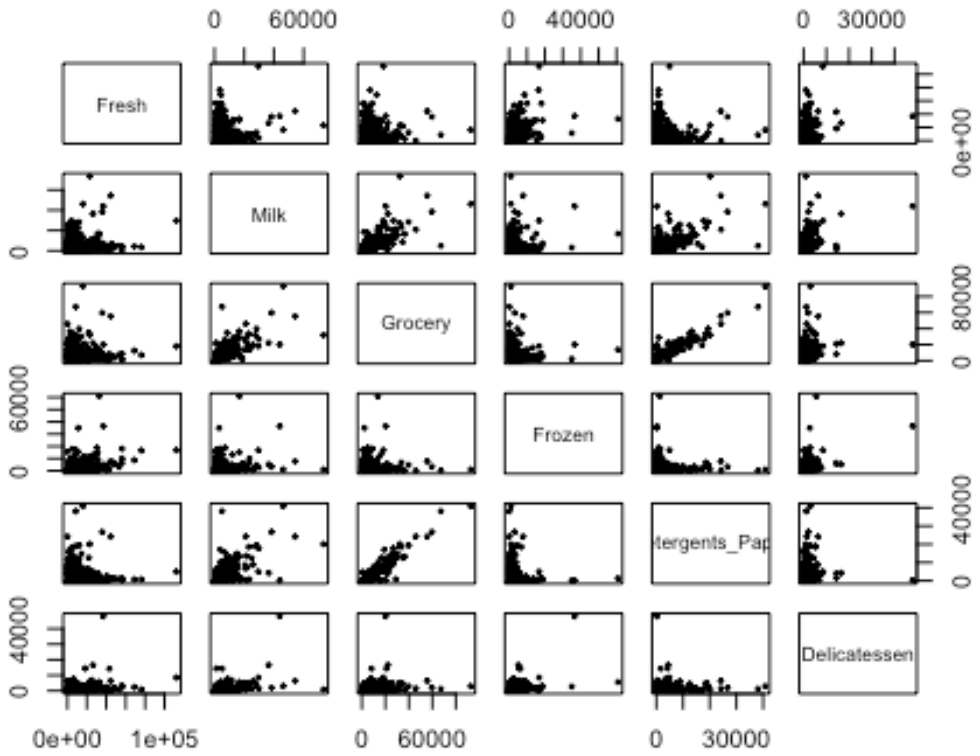
### EDA before Clustering

```
par(mfrow = c(1, 2))

# pie chart for region
pie(table(wholesale$Region), labels = round(table(wholesale$Region)/440, 2),
main = "Region Pie Chart", col = rainbow(3))
legend("topright", c("1","2","3"), cex = 0.8, fill = rainbow(3))

# pie chart for channel
pie(table(wholesale$Channel), labels = round(table(wholesale$Channel)/440, 2)
, main = "Channel Pie Chart", col = rainbow(2))
legend("topright", c("1","2"), cex = 0.8, fill = rainbow(2))
```

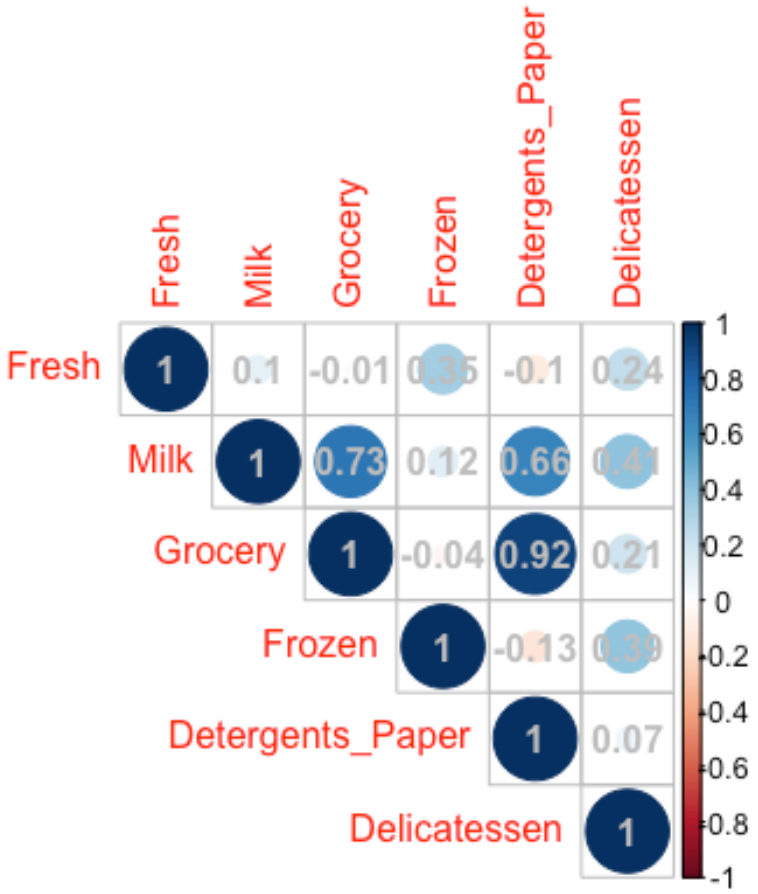



```r
# check correlation among variables
pairs(wholesale[, 3:8], cex = 0.5, pch = 20)
```



```
library(corrplot)
## corrplot 0.92 loaded
corrplot(cor(wholesale[, 3:8]), type = 'upper', addCoef.col = 'gray')
```




## Hierarchical Clustering

### Normalization

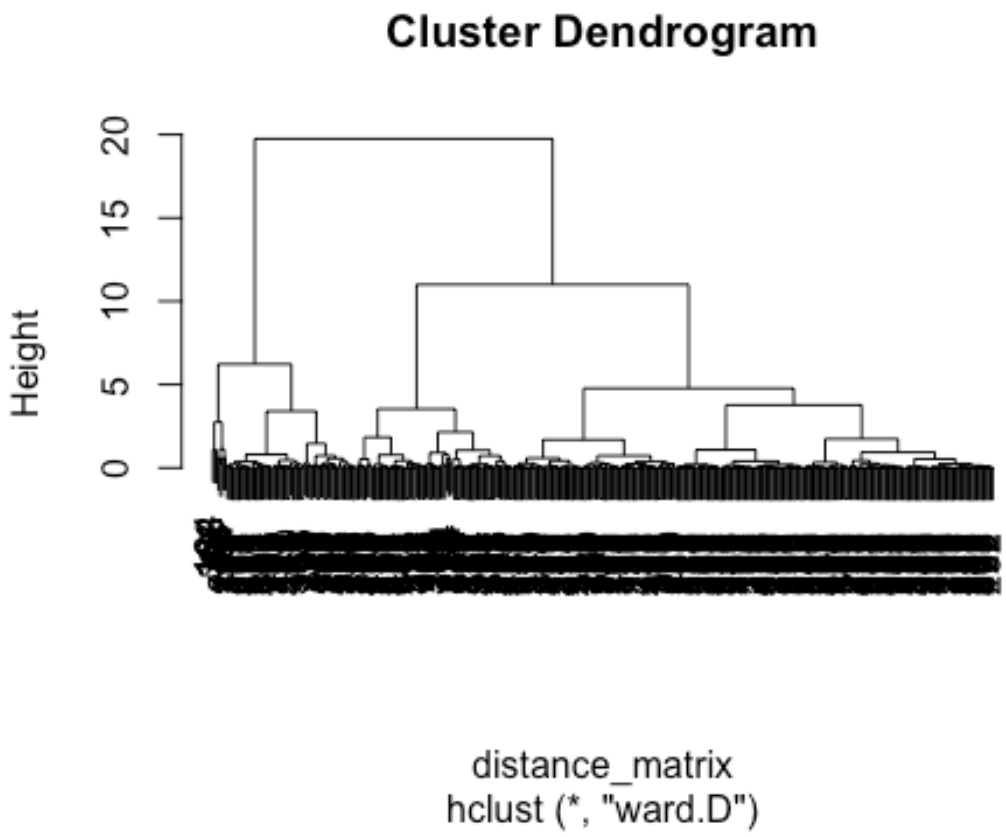
```
normalize = function(x) {
return((x - min(x)) / (max(x) - min(x)))
}
# use the mutate_at() to specify the indexes of columns needed normalization
ws_normalized <- wholesale %>% mutate_at(c(3:8), normalize)
# we also preserve a normalized dataset for k-means later
ws_normalized_k <- wholesale %>% mutate_at(c(3:8), normalize)
```

### Distance Matrix

```
# dist() from package stats can generate distance matrix
library(stats)
# prepare the distance matrix
# the euclidean distance method
distance_matrix <- dist(ws_normalized[, 3:8], method = "euclidean")
```

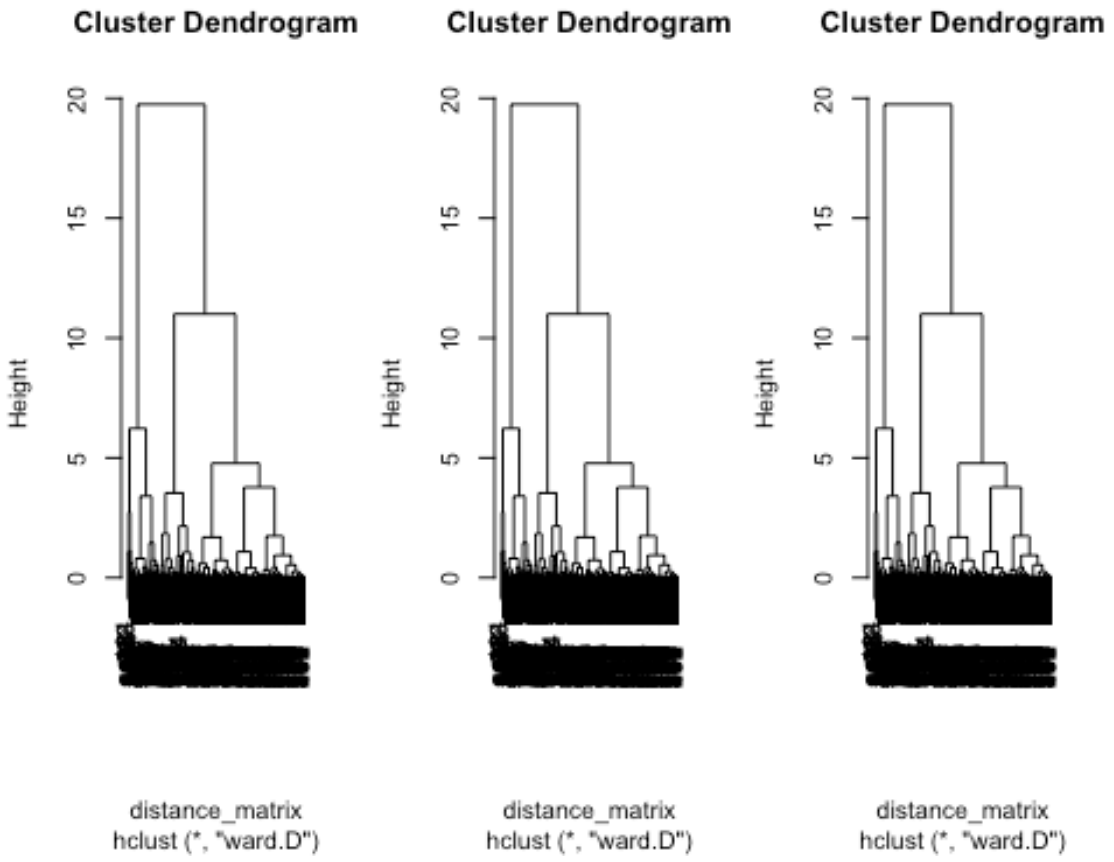
Hierarchical Clustering

```
# we use Ward's Method to measure distances
# plot the dendrogram
hierarchical = hclust(distance_matrix, method = "ward.D")
plot(hierarchical)
```




Check the Cluster Number of 4, 5, and 6 Respectively

```
par(mfrow = c(1, 3))
# set cluster number = 4
plot(hierarchical)
# rect.hclust() can mark the clustering solution for a given number of cluste
rs rect.hclust(hierarchical, k = 4)
# set cluster number = 5
plot(hierarchical)
# rect.hclust() can mark the clustering solution for a given number of cluste
rs rect.hclust(hierarchical, k = 5)
# set cluster number = 6
plot(hierarchical)
```




Check the number of data in each cluster

```
# curtree() can cut the dendrogram and tell you which entities belong to whic
h clust er
ws_normalized$hcluster <- cutree(hierarchical, k = 5)
# also append the cluster labels on the original dataset, maybe we will need
this
wholesale$hcluster <- cutree(hierarchical, k = 5) # just show the head of 6 r
ows
head(ws_normalized)
```

```
##   Channel Region      Fresh       Milk    Grocery      Frozen Detergents_P
aper
## 1       2      3 0.11294004 0.13072723 0.08146416 0.003106305       0.0654
2720
## 2       2      3 0.06289903 0.13282409 0.10309667 0.028548419       0.0805
8985
## 3       2      3 0.05662161 0.11918086 0.08278992 0.039116429       0.0860
5232
## 4       1      3 0.11825445 0.01553586 0.04546385 0.104841891       0.0123
4568
## 5       2      3 0.20162642 0.07291369 0.07755155 0.063933995       0.0434
```

```
5483
## 6       2      3 0.08390698 0.11170568 0.05521843 0.010535139       0.0438
9575
##   Delicatessen hcluster
## 1   0.02784731        1
## 2   0.03698373        1
## 3   0.16355861        1
## 4   0.03723404        2
## 5   0.10809345        2
## 6   0.03020442        1

table(ws_normalized$hcluster)

##
##   1   2   3   4   5
##  92  88  73 179   8
```

### K-Means Clustering

Based on the results of the previous hierarchical clustering, we are more in favor of 5 or 6 clusters rather than 4, and 5 is more than 6.

*Set a Cluster Number of 5 First*

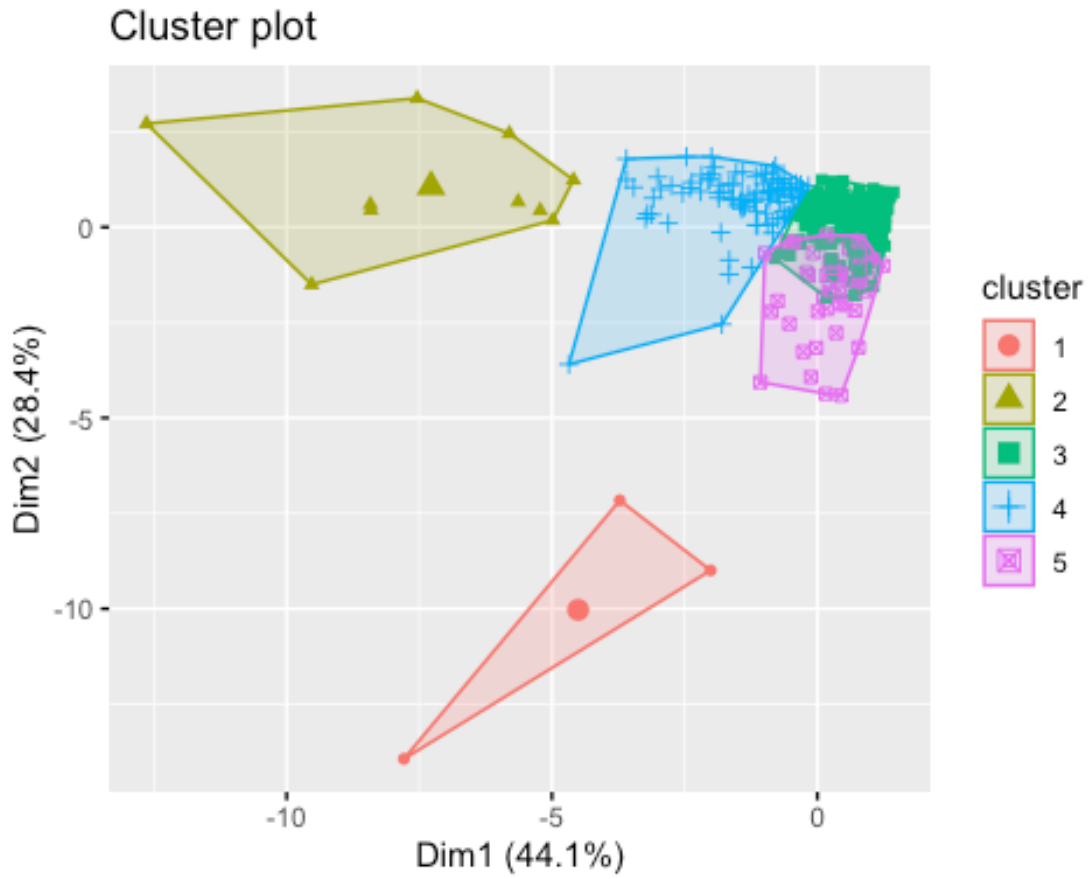
```
# use a this normalized dataset that we've preserved previously, ws_normalize
d_k # note that kmeans() works only with Euclidean distance
kcluster <- kmeans(ws_normalized_k[, 3:8], centers = 5)
head(kcluster$centers) # can see the centroids
```

```
##        Fresh       Milk    Grocery     Frozen Detergents_Paper Delicatesse
n
## 1 0.54007799 0.40936962 0.18659438 0.62494795       0.05266510   0.4317410
7
## 2 0.14232889 0.47184211 0.52312427 0.04979291       0.60925436   0.0613224
9
## 3 0.07896195 0.04101507 0.04145936 0.04354059       0.02474872   0.0217047
4
## 4 0.05181527 0.14567365 0.18337739 0.02360488       0.18012570   0.0388978
6
## 5 0.30814781 0.06249623 0.05936277 0.09574838       0.02039198   0.0430013
0
```

Visualize the Results

```
library(cluster)
library(fpc)
library(mclust)

## Package 'mclust' version 6.0.0
## Type 'citation("mclust")' for citing this R package in publications.
```

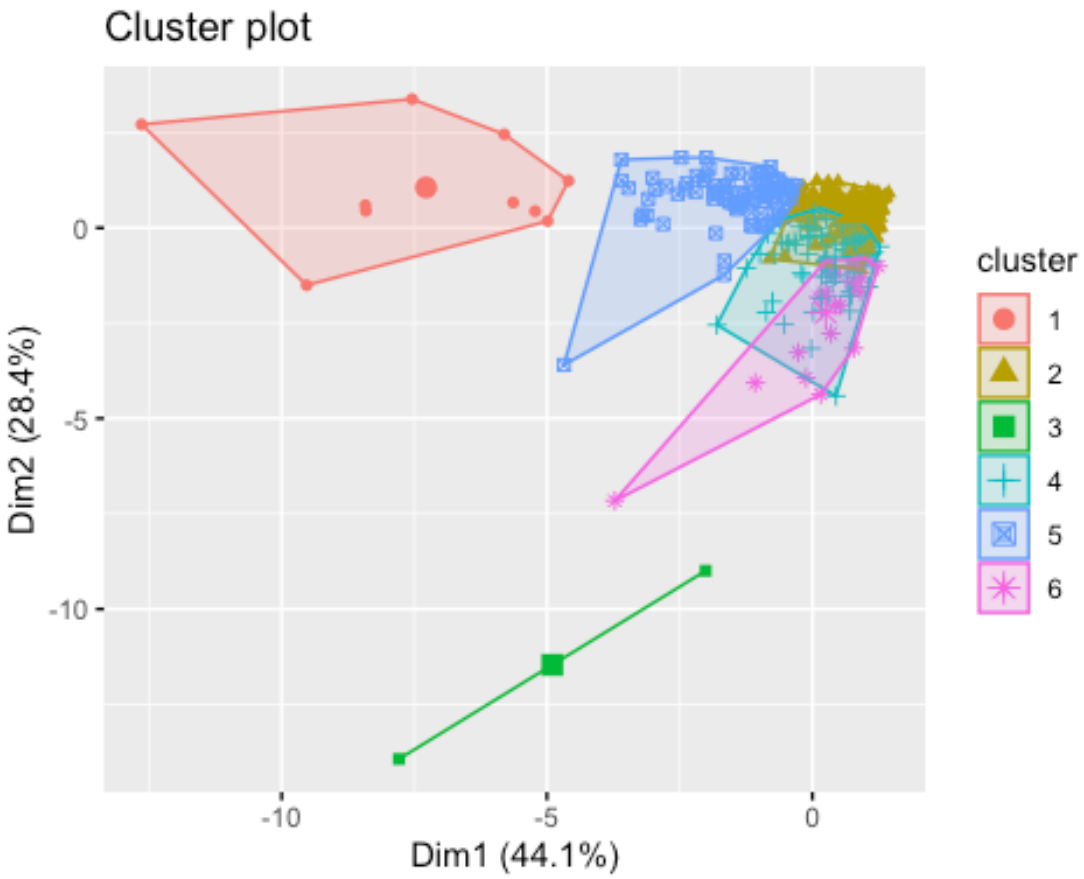
```
library(FactoMineR)
library(factoextra)

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

# cluster plot
fviz_cluster(kcluster, data = ws_normalized_k[, 3:8], geom = "point")
```




The first PC accounts for about 44.1% of the total variation, while the second counts for 28.4%. There is abvious vairance between different clusters. It is reasonable to have 5 clusters.

### Set a Cluster Number of 6

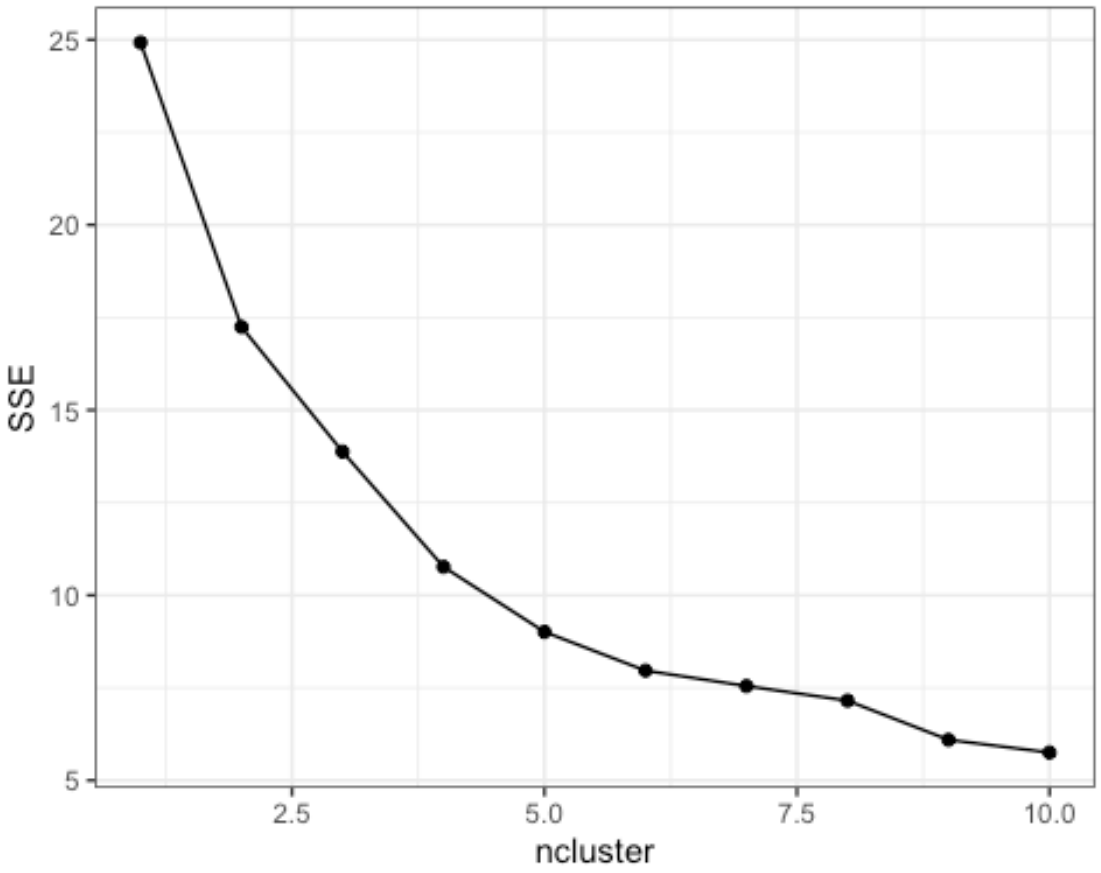
```
kcluster_6 <- kmeans(ws_normalized_k[, 3:8], centers = 6)
fviz_cluster(kcluster_6, data = ws_normalized_k[, 3:8], geom = "point")
```




For 6 clusters, 4 clusters are on the top-right, which are hard to interpret. Also, there is only 2 data points in a cluster. The performance of 5 clusters is better than 6.

### Evaluate Clustering Solutions: SSE Curve

```
# the vector to store the SSE
SSE_curve = c()
for (n in 1:10){
  kc = kmeans(ws_normalized_k[, 3:8], centers = n)
  SSE_curve[n] = kc$tot.withinss
}
# do the plot
plot_data = data.frame(ncluster = 1:10, SSE = SSE_curve)
ggplot(plot_data, aes(x = ncluster, y = SSE)) + geom_line() + geom_point() +
theme_bw()
```




From the elbow plot, it shows that 5 clusters is good enough.

### Silhouette coefficient

Silhouette coefficient = 1 indicates the data point x is very compact within its own cluster and far away from other clusters. Silhouette coefficient = -1 indicates the opposite situation.

```
library(cluster)
sc <- silhouette(ws_normalized$hcluster, dist = distance_matrix)
summary(sc)
```

```
## Silhouette of 440 units in 5 clusters from silhouette.default(x = ws_norma
lized$hcluster, dist = distance_matrix) :
##  Cluster sizes and average silhouette widths:
##          92          88          73         179           8
##  0.16970290 -0.01045482  0.13952127  0.33046600 -0.09460692
## Individual silhouette widths:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.48362  0.04804  0.25625  0.18926  0.36564  0.48338
```

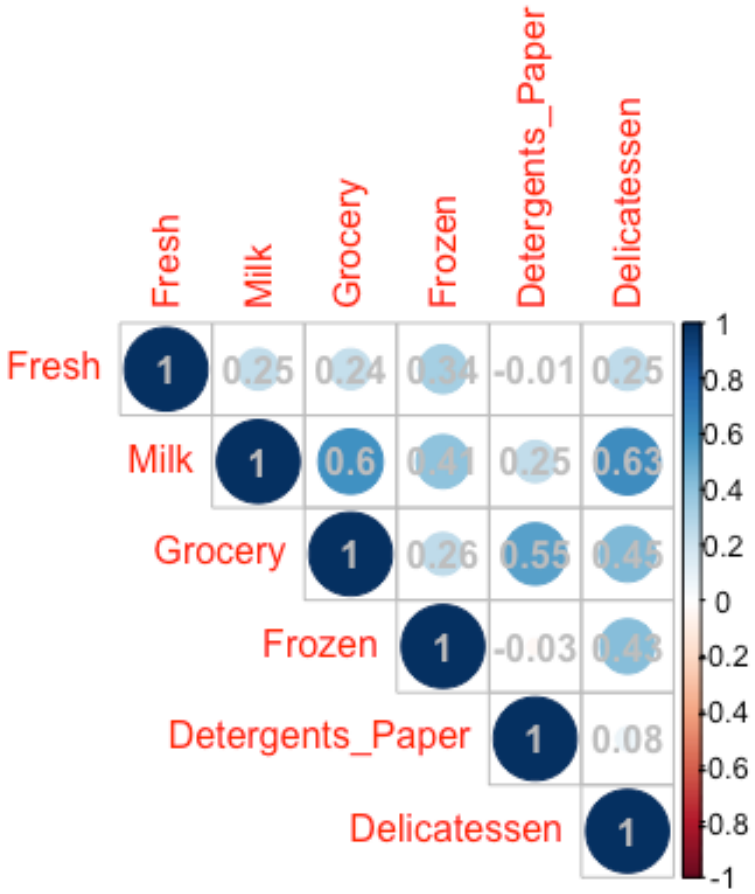
## Analysis after Clustering

### Split by region

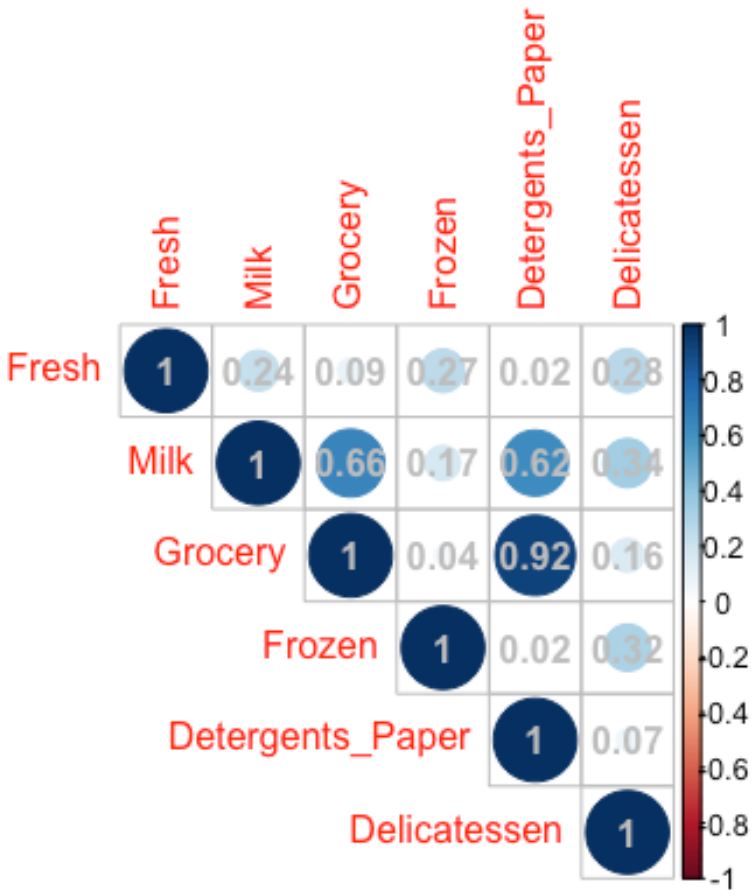
```
ws_normalized_re1 <- ws_normalized %>% filter(Region == 1)
ws_normalized_re2 <- ws_normalized %>% filter(Region == 2)
ws_normalized_re3 <- ws_normalized %>% filter(Region == 3)
```

### Split by channel

```
ws_normalized_ch1 <- ws_normalized %>% filter(Channel == 1)
ws_normalized_ch2 <- ws_normalized %>% filter(Channel == 2)
corrplot(cor(ws_normalized_ch1[, 3:8]), type = 'upper', addCoef.col = 'gray')
```




```
corrplot(cor(ws_normalized_ch2[, 3:8]), type = 'upper', addCoef.col = 'gray')
```




According to the 2 correlation plots, there is strongest positive correlation between Grocery and Detergents_Papers. The strong positive correlations between food products(Fresh, Milk, Frozen, and Delicatessen) appears in channel 1 only after splitting by channel.

**Through analyzing the correlation, we believe the channels segmentation could preserve or even create meaningful correlation for the data, thus it might be a good idea to do different business strategies in channel 1 and 2, respectively.**

**However, we didn't find valuable insights in terms of region, so we might not suggest do further strategies in different regions.**