Data Processing and Clustering Analysis - HW1

2023-07-02

```r
setwd('C:/Users/Yvonne/Desktop/UMN Courses/6131')
library(dplyr)
library(ggplot2)

wholesale <- read.csv('Wholesale customers data.csv')
```

GROUP PART – technical document

Table of contents:

Problem defining and our overall rationale to solve it

General goal for analysis

To get a deeper understanding of the spending patterns of the clients.

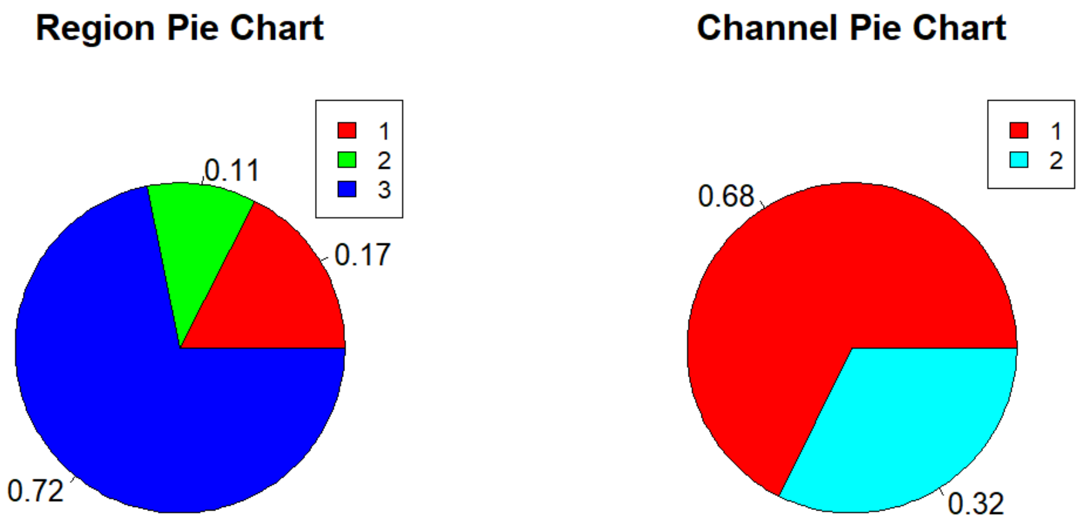
Specific goal for analysis:

1. conduct clustering analysis and discover different patterns of spending for each cluster.
2. we need to do EDA before moving on to further analysis, and we are going to combine analysis from that as well as clustering to provide data support for business solutions.
3. we will translate analyzing results into business solutions outline, and more detailed business strategies will be presented in our managerial document.

EDA before Clustering

Pie charts for Channels and Regions to see the proportion.

```r
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))
```



Region 3 is the largest compared to Region 1 and 2, being about 70%.

Channel 1 is about 2 times larger than channel 2.

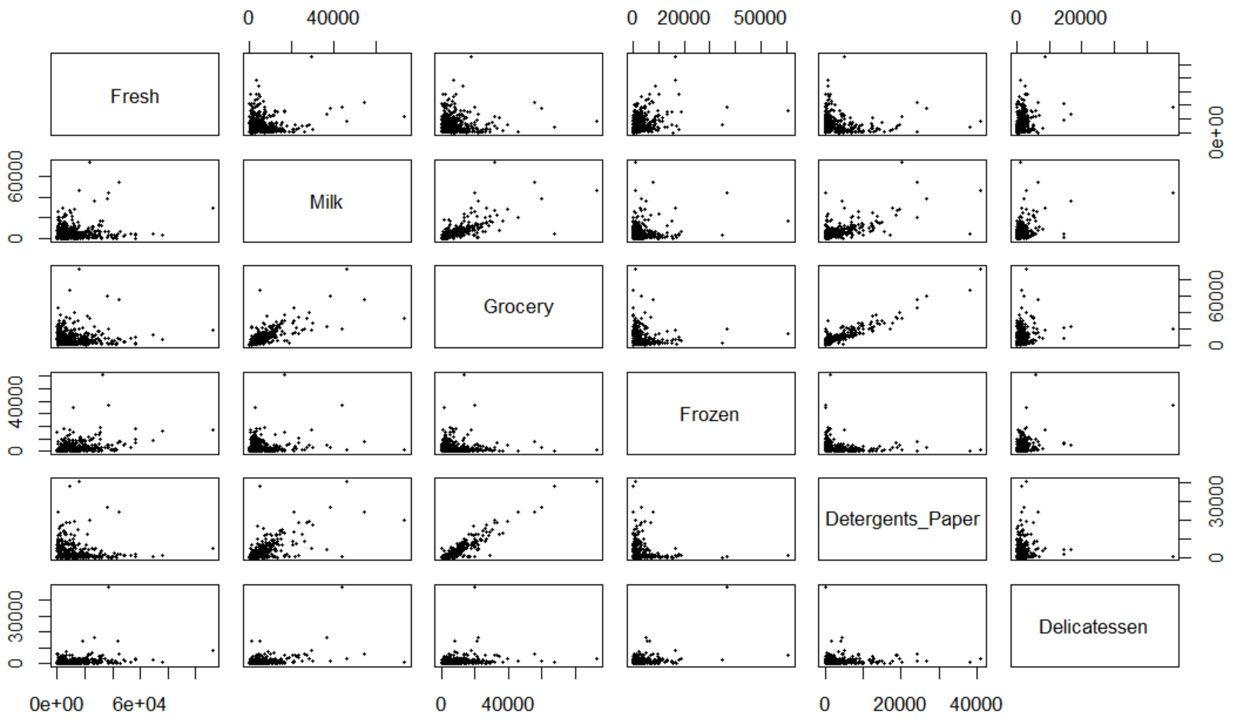
Before further analysis, we want to check the correlation among variables to get a conceptual and general understanding of our dataset.

```r
library(corrplot)
```

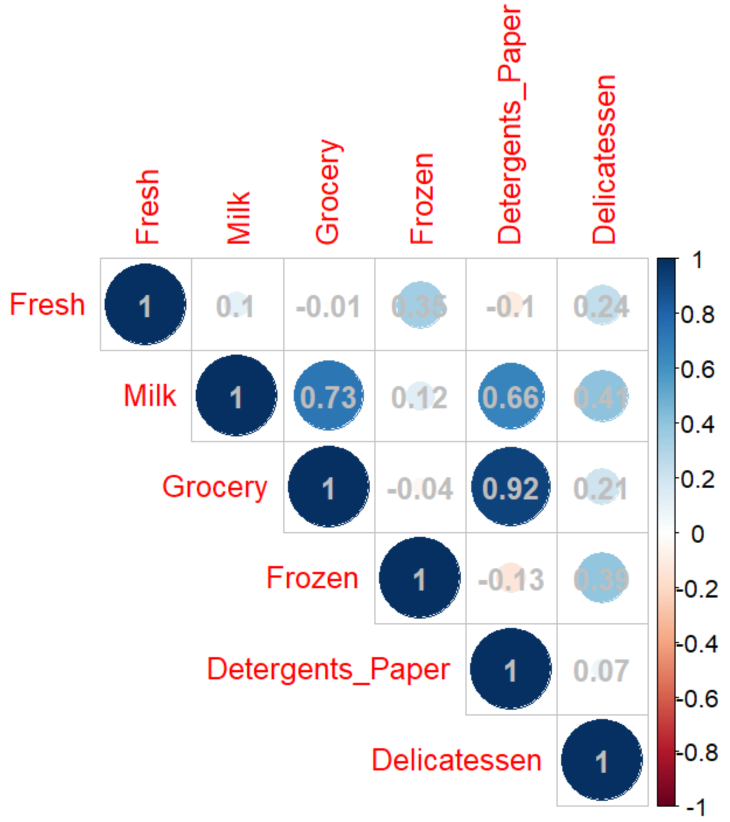
```
## corrplot 0.92 loaded
```

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

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



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



From the 2 plots above, we found no obvious negative correlation between variables.

What we believe more worth highlighting is the relatively strong positive correlation between Grocery and Detergents_paper, and then are Milk and Grocery, Milk and Detergents_paper.

Considering the categories of the products range widely, we will keep these correlation results in mind then conduct clustering analysis, and after that we will come back to see if those relatively strong positive correlations still exist, and whether their patterns worth mentioning if existing.

Hierarchical Clustering

Calculate distances

Normalization

Implement min-max normalization before computing our distance matrix.

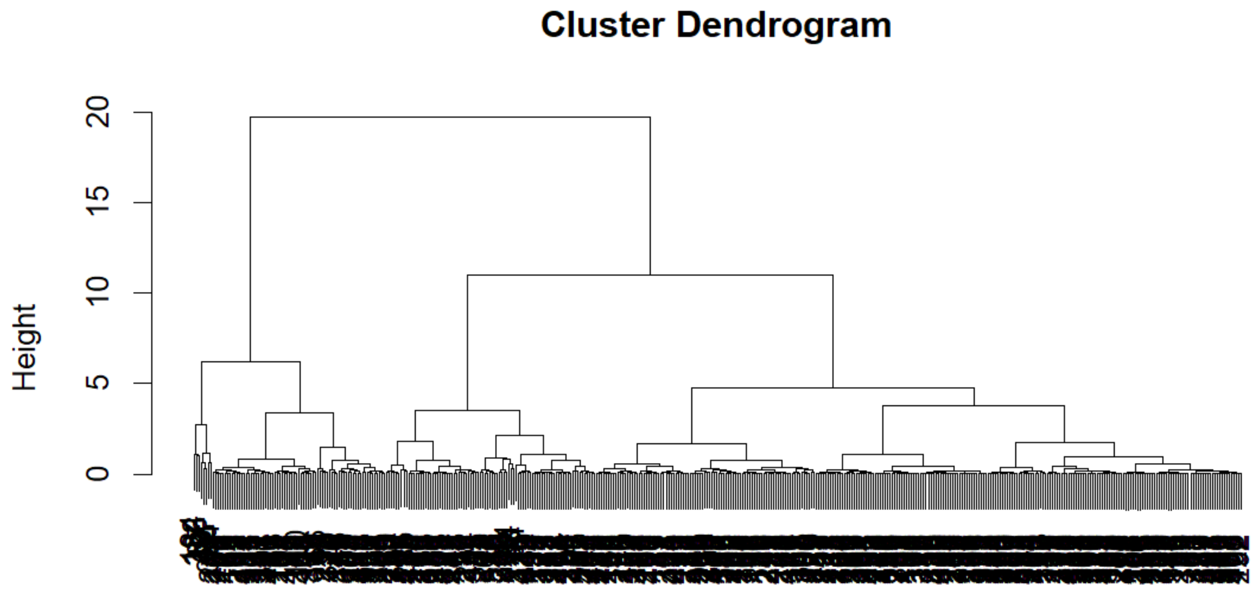
```
normalize = function(x) {
  return((x - min(x)) / (max(x) - min(x)))
}

# 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
# we use the euclidean distance, which is also used in k-means clustering later
distance_matrix <- dist(ws_normalized[, 3:8], method = "euclidean")
# View(as.matrix(distance_matrix)) # can view the distance matrix
```

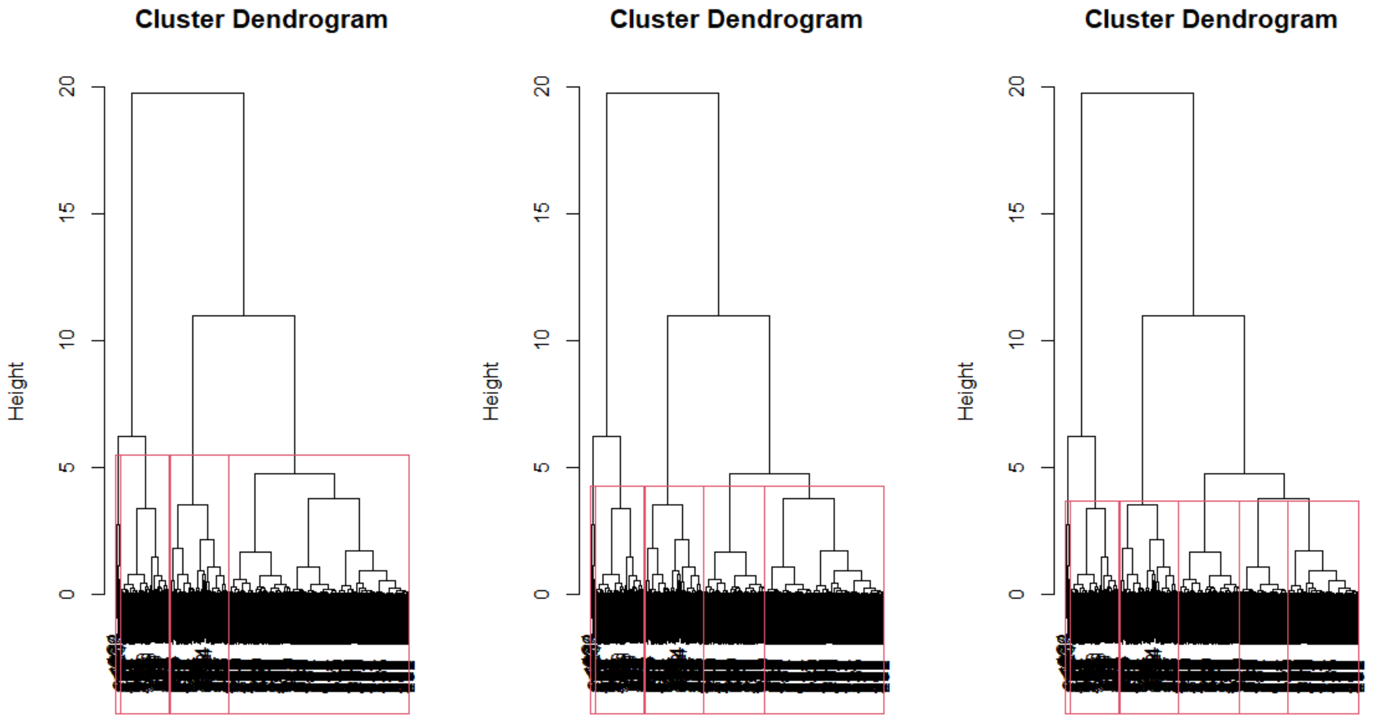
Conduct Hierarchical clustering and plot the dendrogram

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

# plot the dendrogram
plot(hierarchical)
```



From the dendrogram above, there seem to be 4 or 5 or 6 more obvious clusters, so we decided to check the cluster number of 4, 5, and 6 respectively.

```r
par(mfrow = c(1, 3))

# set cluster number = 4
plot(hierarchical)
# rect.hclust() can mark the clustering solution for a given number of clusters
rect.hclust(hierarchical, k = 4)

# set cluster number = 5
plot(hierarchical)
# rect.hclust() can mark the clustering solution for a given number of clusters
rect.hclust(hierarchical, k = 5)

# set cluster number = 6
plot(hierarchical)
# rect.hclust() can mark the clustering solution for a given number of clusters
rect.hclust(hierarchical, k = 6)
```



As shown above, a cluster number of 4 would form a strangely large group which we consider can still be divided into smaller groups. Thus at this point, a cluster number of 5 or 6 seems more reasonable for us.

We decided to choose 5 first (please note that it’s a relatively subjective choice), and will test our cluster number later in the evaluation part.

Combine the column that labels cluster to the normalized dataset.

```
# curtree() can cut the dendrogram and tell you which entities belong to which 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 rows
head(ws_normalized)

##   Channel Region      Fresh       Milk    Grocery      Frozen Detergents_Paper
## 1       2      3 0.11294004 0.13072723 0.08146416 0.003106305       0.06542720
## 2       2      3 0.06289903 0.13282409 0.10309667 0.028548419       0.08058985
## 3       2      3 0.05662161 0.11918086 0.08278992 0.039116429       0.08605232
## 4       1      3 0.11825445 0.01553586 0.04546385 0.104841891       0.01234568
## 5       2      3 0.20162642 0.07291369 0.07755155 0.063933995       0.04345483
## 6       2      3 0.08390698 0.11170568 0.05521843 0.010535139       0.04389575
##   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
```

Note that we will also do the k-means clustering to double check the number of cluster that we are more in favor of, and will give our corresponding reason later. And we’re also going to see whether a cluster number of 5 is reasonable and appropriate for business point-of-view.

Check the number of data in each cluster.

```
table(ws_normalized$hcluster)
##   1   2   3   4   5 
##  92  88  73 179   8
```

The cluster 4 is an obviously larger group than others, while the 5 is the smallest with only 8 rows. With this in mind, we move to the k-means clustering.

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.

5 clusters

Set a cluster number of 5 first.

```
# use a this normalized dataset that we've preserved previously, ws_normalized_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
```

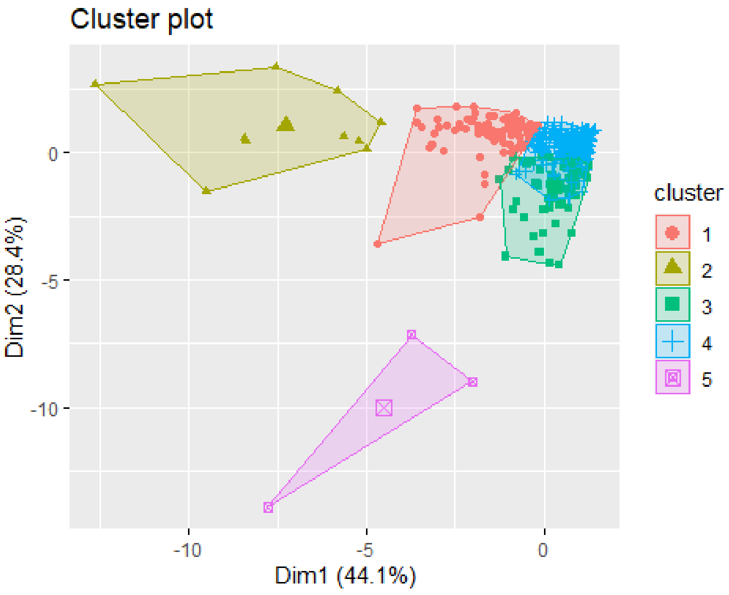
```
##        Fresh       Milk    Grocery    Frozen Detergents_Paper Delicatessen
## 1 0.04831988 0.14601276 0.18434265 0.02355312       0.18151814   0.03793272
## 2 0.14232889 0.47184211 0.52312427 0.04979291       0.60925436   0.06132249
## 3 0.28951801 0.06109561 0.05902438 0.08860389       0.02330097   0.04332125
## 4 0.07521079 0.04114892 0.04154015 0.04305447       0.02488650   0.02120851
## 5 0.54007799 0.40936962 0.18659438 0.62494795       0.05266510   0.43174107
```

```
head(kcluster$cluster) #can see the cluster that each entity belongs to
```

```
## [1] 4 4 4 4 3 4
```

Use fviz_cluster() to visualize the results of k-means clustering.

```
library(cluster)
library(fpc)
library(mclust)
library(FactoMineR)
library(factoextra)

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



This cluster plot is basically a PCA plot, indicating the first PC accounts for about 44.1% of the total variation, while the second counts for 28.4%.

We can see the relatively obvious difference between several clusters, thus we believe a cluster number of 5 is reasonable.

Then we also tried 6 clusters.

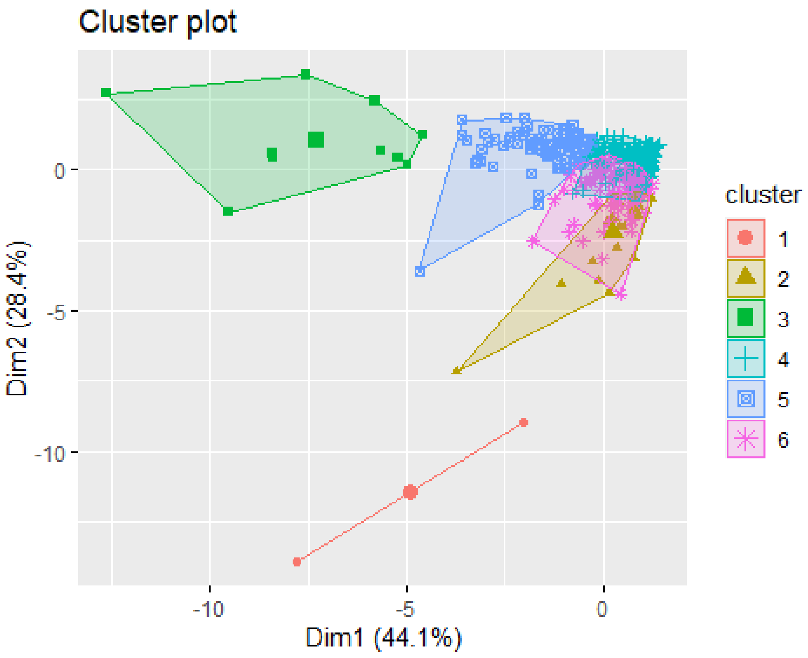
6 clusters

Set a cluster number of 6.

```
kcluster_6 <- kmeans(ws_normalized_k[, 3:8], centers = 6)
head(kcluster_6$centers) # can see the centroids

##        Fresh       Milk    Grocery     Frozen Detergents_Paper Delicatessen
## 1 0.31011699 0.41272824 0.18210332 0.80002137       0.01843278   0.55846892
## 2 0.44625568 0.05980705 0.05628599 0.09648669       0.02315260   0.05009102
## 3 0.14232889 0.47184211 0.52312427 0.04979291       0.60925436   0.06132249
## 4 0.05582426 0.04051000 0.04157230 0.03470480       0.02561912   0.01947448
## 5 0.04637980 0.15007369 0.18558231 0.02308957       0.18480565   0.03538810
## 6 0.18559040 0.05563062 0.05569048 0.08066515       0.02692948   0.03738370

head(kcluster_6$cluster) #can see the cluster that each entity belongs to

## [1] 4 4 4 6 6 4
```

Use fviz_cluster() to visualize the results of k-means clustering.

```
# cluster plot
fviz_cluster(kcluster_6, data = ws_normalized_k[, 3:8], geom = "point")
```



This plot didn’t make that much sense for us since there’s too many clusters on the top-right which are hard to interpret. Also, in some trails there will be only 2 data points in a cluster, and we’re afraid of overly interpreting them as a meaningful cluster.

Although there’s also a small cluster in a cluster number of 5, that way of partition in other clusters makes more sense to us.

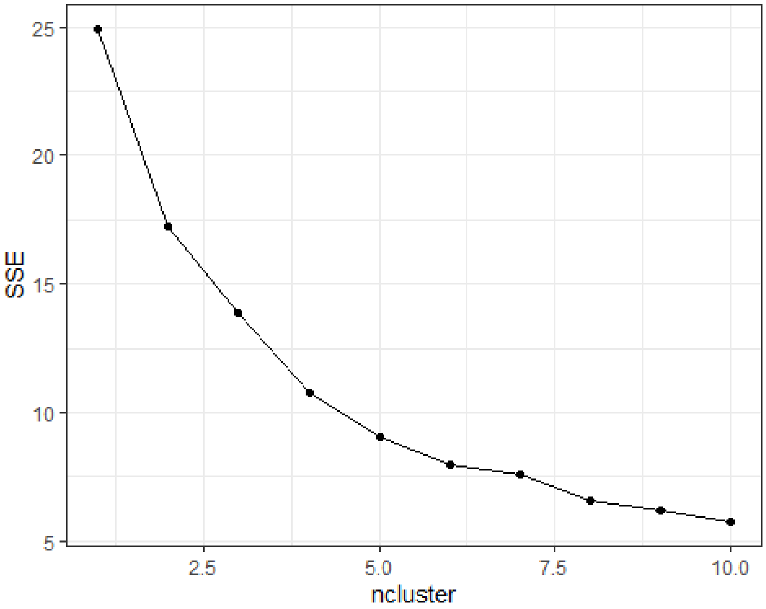
Thus until now, we are most in favor of 5 clusters in our mind.

Then we did the evaluation to see further details.

Evaluate Clustering Solutions: SSE, cluster plot, and Slihouette Coefficient

Evaluate Clustering Solutions: SSE Curve.

```r
# 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()
```



In the elbow plot, since the magnitude of SSE decreasing, i.e. slope, has a bigger change in 4-5 than 5-6, meaning that SSE decrease more when changing from 4 to 5 than from 5 to 6, thus we are more certain to say 5 clusters might be better and better enough.

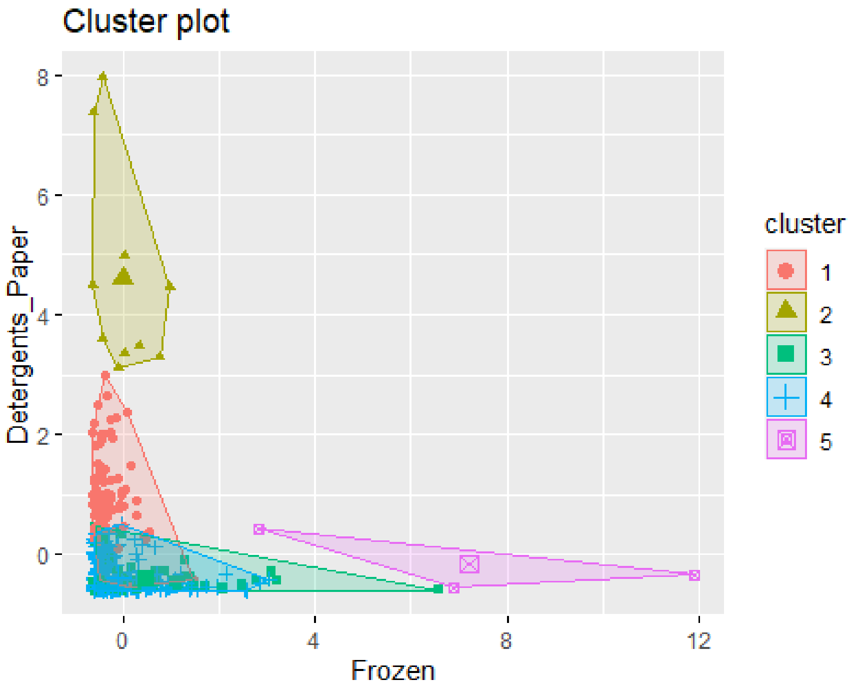
Evaluate Clustering Solutions: cluster plot

Note that since the algorithm starts by selecting k objects and set as the initial centers for clusters at random, the results might not be exactly the same through each time, so we can't use the labels of cluster 1, 2, 3, 4, 5 to refer to those in hierarchical clustering.

In short, cluster plot is used to evaluate whether a cluster number of 5 or 6 is more appropriate and reasonable for us to interpret and find insights in business point-of-view.

We now plot them according to 2 categories with the least positive correlation (and there's actually a really small negative correlation between them) so that we can interpret more easily.

```
fviz_cluster(kcluster, data = ws_normalized_k[, 3:8], geom = "point", choose.vars =
c("Frozen", "Detergents_Paper"))
```



Note that the 2 categories are not represent the 2 PCs, but generally speaking, a negative correlation can explain more dimensions of variation than a positive correlation, so it's better to choose the categories without positive correlation.

Note again that the labels here might not be the same as those in hierarchical clustering. If you redo it again, the labels might not be the same as the elaboration below, but the main patterns will be almost the same:

- Cluster 3, 5 has a bigger range in Frozen yet small range in Detergents_Paper.
- Cluster 2 spends really few on Frozen yet much on Detergents_Paper, compared to others.
- Cluster 2 and 5 seem to follow different spending patterns since their ranges varies a lot on the 2 dimensions respectively, so do 1 and 3.
- Cluster 4 seems relatively evenly falls on these 2 categories, indicating they have more tendency to buy both products.

This just an example of patterns we've found in terms of these 2 products. We will see if there's other business point-of-views using Excel chart in the managerial document.

Evaluate Clustering Solutions: Silhouette coefficient

Silhouette coefficient is a measure of both cohesion and separation with its range of [-1, 1]. 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.

Thus generally speaking, a higher Silhouette coefficient means clusters are well distinguished from each other cluster and having well cohesion within its cluster.

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

```
## Silhouette of 440 units in 5 clusters from silhouette.default(x = ws_normalized$h
cluster, 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
```

The result of Silhouette coefficient is not quite high, yet the previous 2 evaluation still seem reasonable for us, so we still keep our original clustering, With the result of Silhouette coefficient in mind, and we will look for further insights in terms business part.

Analysis after clustering

We want to dig deeper to see whether the correlation in the first part still exist after we cluster the data.

Because we want to see, more specifically, is if there are any meaningful clients group or types as well as their distribution across different channels of regions, we will split the dataset by regions and channels respectively.

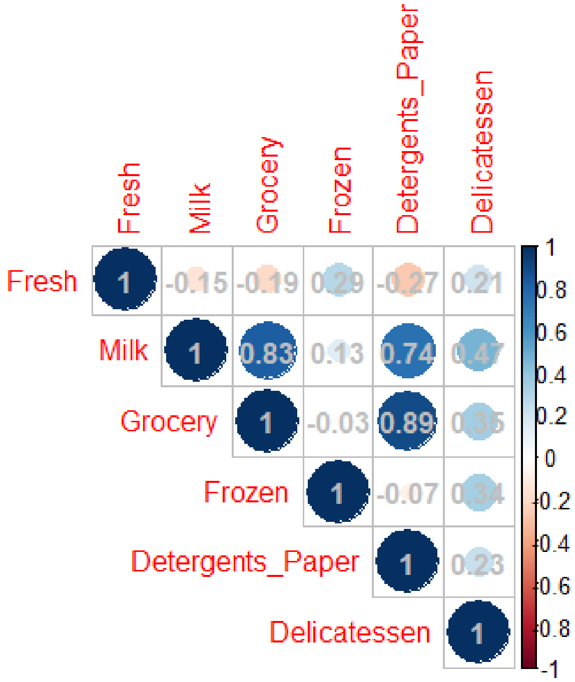
Split by region

first we split by regions.

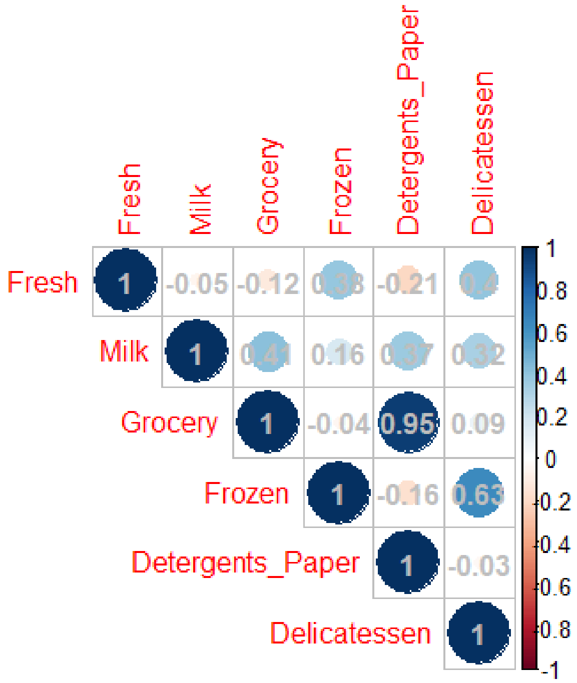
```r
# filter the regions (normalized dataset)
ws_normalized_re1 <-
  ws_normalized %>% filter(Region == 1)

ws_normalized_re2 <-
  ws_normalized %>% filter(Region == 2)

ws_normalized_re3 <-
  ws_normalized %>% filter(Region == 3)
```

Check correlation for each region.

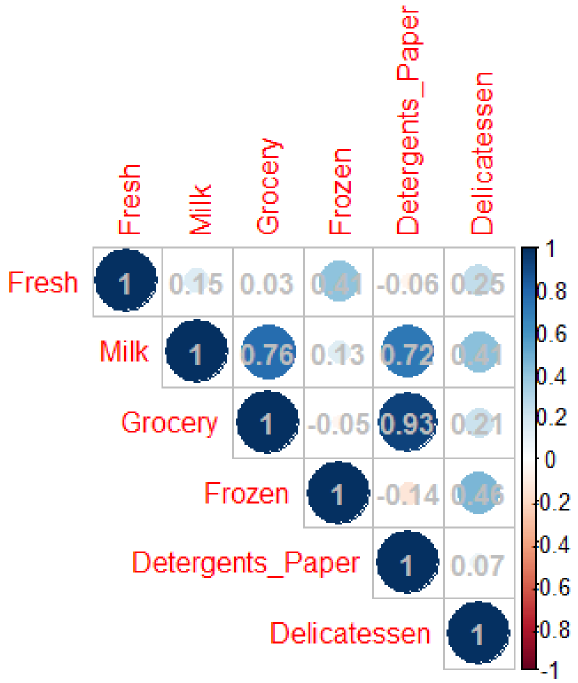
```r
# Region 1
corrplot(cor(ws_normalized_re1[, 3:8]), type = 'upper', addCoef.col = 'gray')
```



```r
# Region 2
corrplot(cor(ws_normalized_re2[, 3:8]), type = 'upper', addCoef.col = 'gray')
```



```r
# Region 3
corrplot(cor(ws_normalized_re3[, 3:8]), type = 'upper', addCoef.col = 'gray')
```



We've also done the plot through Excel yet we didn't find valuable business insights among regions. Thus we move on to split data by channels.

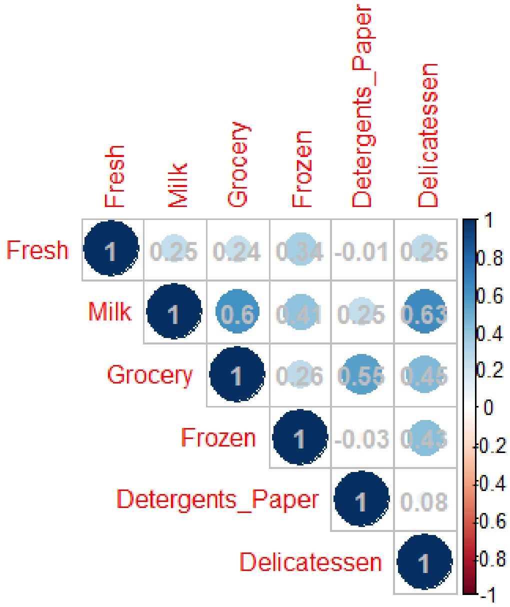
Split by channel

Then we split by channels.

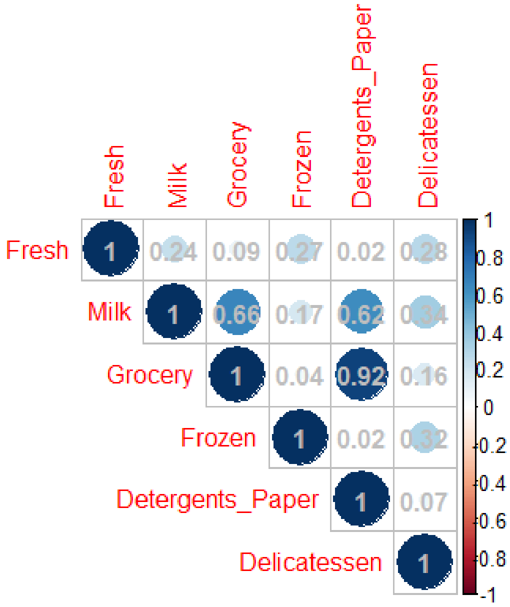
```r
# filter the channels (normalized dataset)
ws_normalized_ch1 <-
  ws_normalized %>% filter(Channel == 1)

ws_normalized_ch2 <-
  ws_normalized %>% filter(Channel == 2)
```

Check correlation for each channel.

```
# Channel 1
corrplot(cor(ws_normalized_ch1[, 3:8]), type = 'upper', addCoef.col = 'gray')
```



```
# Channel 2
corrplot(cor(ws_normalized_ch2[, 3:8]), type = 'upper', addCoef.col = 'gray')
```



According to the 2 correlation plots, the strongest positive correlation between Grocery and Detergents_Papers, 2 categories of products not belonging to food, is well preserved in channel 2. What's more, the stronger positive correlations between food products(Fresh, Milk, Frozen, and Delicatessen) appear in channel 1 only after splitting by channel.

After split by channel…

Since we believe it’s worth focusing on clusters in different channels, we draw the cluster distribution on the two channels respectively using Excel, and elaboration will be in our managerial document.

Here we take mean for each cluster in both of the 2 channels and output the files to Excel.

```
ws_normalized_mean_for_clusters_ch1 <-
  ws_normalized_ch1 %>% group_by(hcluster) %>% summarize_at(c(3:8), mean)

ws_normalized_mean_for_clusters_ch2 <-
  ws_normalized_ch2 %>% group_by(hcluster) %>% summarize_at(c(3:8), mean)
```

We output the two dataframe to Excel to plot.

```
library(openxlsx)

write.xlsx(ws_normalized_mean_for_clusters_ch1, 'ws_normalized_mean_for_clusters_ch
1.xlsx')

write.xlsx(ws_normalized_mean_for_clusters_ch2, 'ws_normalized_mean_for_clusters_ch
2.xlsx')
```

We did find business insights, thus come up with the recommendation. Main recommendation: channel segmentation management. It will be explained in the manegiral document.

Output the files needed (normalized and original).

```
# normalized dataset with the cluster labeled column from hierarchical clustering
write.xlsx(ws_normalized, 'ws_normalized.xlsx')

# original dataset with the cluster labeled column from hierarchical clustering
write.xlsx(wholesale, 'wholesale.xlsx')
```

Summary for clustering and other analysis results

correlation summary

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.

clustering summary

Through the clustering analysis, we believe there really exist different and specific spending patterns in each cluster, and we believe a cluster number of 5 is the most reasonable to make further business recommendations.

Translate analyzing results into business solutions outline

Concrete suggestions for business solutions based on results from statistical analysis:

1. suggest integration strategy according to positive correlation among product categories, since positive correlation indicates that spending patterns of the products are in similar directions (more or less), thus can accept the same business strategy. We do integration here to reduce cost for setting up different strategies for them.

2. focus on cluster with the most obvious patterns. If the patterns have valuable business interpretation, we should focus on that rather than suggest a too general solution.