Project --- Wholesale Customer Analysis

Introduction:

wholesale is the resale (sale without transformation) of goods to retailers, to industrial, commercial, institutional or professional users, or to other wholesalers. Analyzing wholesale customer information can help wholesalers making decisions, increasing work efficiency, and improving market competition.

I downloaded this wholesale customer dataset from https://www.kaggle.com/binovi/wholesale-customers-data-set . The data set refers to clients of a wholesale distributor. It includes the annual spending in monetary units on diverse product categories.

I got relevant variables information from https://archive.ics.uci.edu/ml/datasets/wholesale+customers.

Attribute Information:

1. FRESH: annual spending (m.u.) on fresh products (Continuous)
2. MILK: annual spending (m.u.) on milk products (Continuous)
3. GROCERY: annual spending (m.u.) on grocery products (Continuous)
4. FROZEN: annual spending (m.u.) on frozen products (Continuous)
5. DETERGENTS_PAPER: annual spending (m.u.) on detergents and paper products (Continuous)
6. DELICATESSEN: annual spending (m.u.) on and delicatessen products (Continuous)
7. CHANNEL: customersale Channel - Horeca (Hotel/Restaurant/Cafe) or Retail channel (Nominal)
8. REGION: customersale Region - Lisnon, Oporto or Other (Nominal)

Objective:

My goal is to use various clustering techniques to segment customers. Clustering is an unsupervised learning algorithm that tries to cluster data based on their similarity. In this dataset, clustering analysis can determine which types of products customers tend to buy together, and which types of customers purchase certain products. The result can then be used to better market products to customers based on their types, and to advertise similarly purchased products. This approach can also help in predicting and ordering inventory by knowing that certain types of products are often purchased together.

Data Description:

This data set include 440 observations and 8 variables. In channel column “1” is horeca (Hotel/Restaurant/Cafe), and “2” is retail. Region represents lisnon, oporto or other (Nominal). Other columns indicate customers’ spending on different products. Based on descriptive statistics for data, I find customers’ spending behaviors on different product categories are different. Further, I split data into two parts based on their channels. I find different channels also have unique spending behaviors. In channel 1, customers spent more on fresh, grocery, and frozen; in channel 2, customers spent more on grocery, milk, and fresh. Moreover, grocery and detergents-paper are highly corelated in channel 2. After clustering, we would find further information.

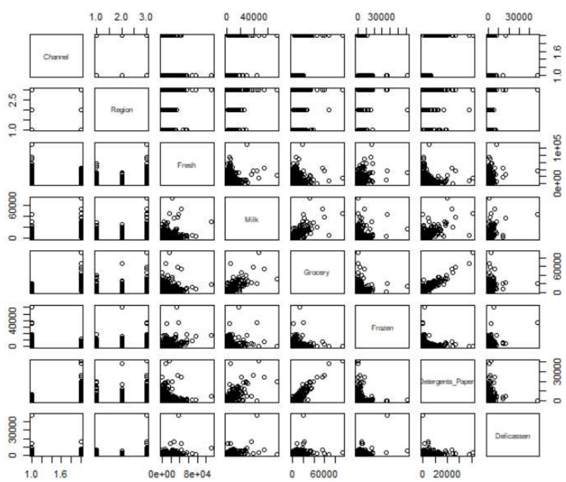
```
str(data)
```

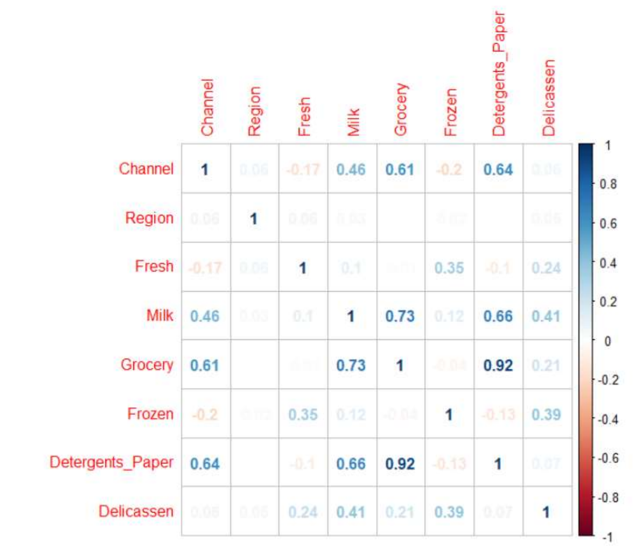
```
'data.frame':   440 obs. of  8 variables:
 $ Channel         : int  2 2 2 1 2 2 2 2 1 2 ...
 $ Region          : int  3 3 3 3 3 3 3 3 3 3 ...
 $ Fresh           : int  12669 7057 6353 13265 22615 9413 12126 7579 5963 6006 ...
 $ Milk            : int  9656 9810 8808 1196 5410 8259 3199 4956 3648 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 ...
```

```
> summary(data)
    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  
> 
```

```
> stat.desc(data[,-c(1,2)])
                    Fresh         Milk      Grocery       Frozen Detergents_Paper   Delicassen
nbr.val      4.400000e+02 4.400000e+02 4.400000e+02 4.400000e+02     4.400000e+02 4.400000e+02
nbr.null     0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00     0.000000e+00 0.000000e+00
nbr.na       0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00     0.000000e+00 0.000000e+00
min          3.000000e+00 5.500000e+01 3.000000e+00 2.500000e+01     3.000000e+00 3.000000e+00
max          1.121510e+05 7.349800e+04 9.278000e+04 6.086900e+04     4.082700e+04 4.794300e+04
range        1.121480e+05 7.344300e+04 9.277700e+04 6.084400e+04     4.082400e+04 4.794000e+04
sum          5.280131e+06 2.550357e+06 3.498562e+06 1.351650e+06     1.267857e+06 6.709430e+05
median       8.504000e+03 3.627000e+03 4.755500e+03 1.526000e+03     8.165000e+02 9.655000e+02
mean         1.200030e+04 5.796266e+03 7.951277e+03 3.071932e+03     2.881493e+03 1.524870e+03
SE.mean      6.029377e+02 3.518457e+02 4.530455e+02 2.314375e+02     2.272985e+02 1.344433e+02
CI.mean.0.95 1.185003e+03 6.915113e+02 8.904077e+02 4.548631e+02     4.467286e+02 2.642325e+02
var          1.599549e+08 5.446997e+07 9.031010e+07 2.356785e+07     2.273244e+07 7.952997e+06
std.dev      1.264733e+04 7.380377e+03 9.503163e+03 4.854673e+03     4.767854e+03 2.820106e+03
coef.var     1.053918e+00 1.273299e+00 1.195174e+00 1.580332e+00     1.654647e+00 1.849407e+00
>
```

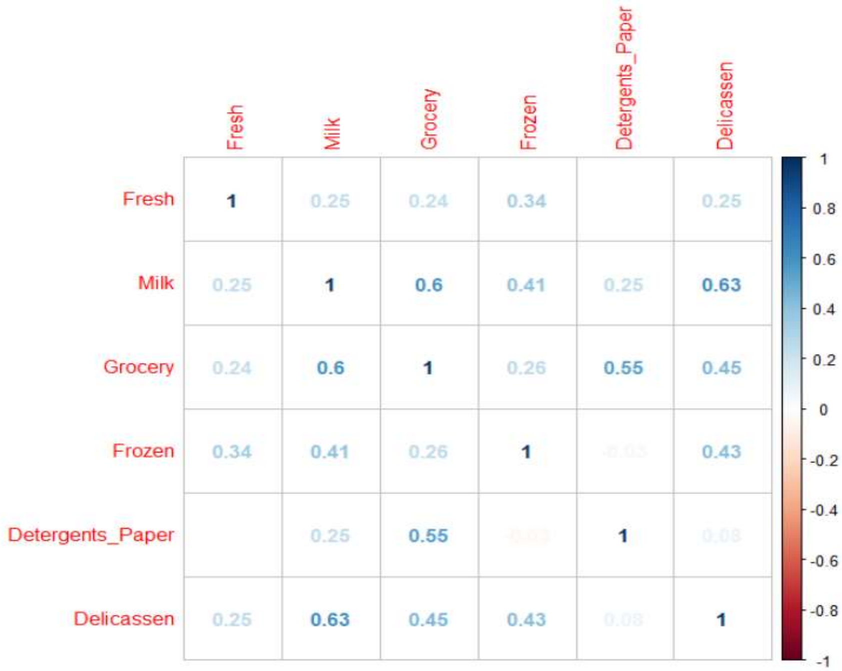
```
plot(data)
corrplot(corrmatrix, method = 'number')
```



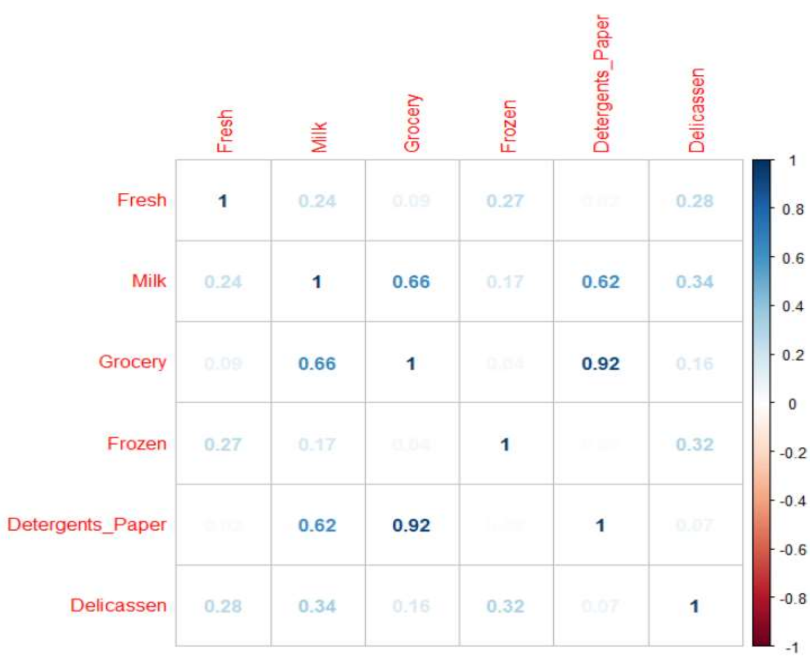


```
> horeca.data <- subset(data, Channel == 1)
> retail.data <- subset(data, Channel == 2)
> summary(horeca.data)
    Channel      Region          Fresh             Milk            Grocery     
 Min.   :1   Min.   :1.00   Min.   :     3   Min.   :   55   Min.   :    3  
 1st Qu.:1   1st Qu.:2.00   1st Qu.:  4070   1st Qu.: 1164   1st Qu.: 1704  
 Median :1   Median :3.00   Median :  9582   Median : 2157   Median : 2684  
 Mean   :1   Mean   :2.51   Mean   : 13476   Mean   : 3452   Mean   : 3962  
 3rd Qu.:1   3rd Qu.:3.00   3rd Qu.: 18275   3rd Qu.: 4030   3rd Qu.: 5077  
 Max.   :1   Max.   :3.00   Max.   :112151   Max.   :43950   Max.   :21042  
     Frozen      Detergents_Paper   Delicassen   
 Min.   :   25   Min.   :   3.0   Min.   :    3  
 1st Qu.:  830   1st Qu.: 183.2   1st Qu.:  379  
 Median : 2058   Median : 385.5   Median :  821  
 Mean   : 3748   Mean   : 790.6   Mean   : 1416  
 3rd Qu.: 4559   3rd Qu.: 899.5   3rd Qu.: 1548  
 Max.   :60869   Max.   :6907.0   Max.   :47943  
> summary(retail.data)
    Channel      Region           Fresh            Milk          Grocery     
 Min.   :2   Min.   :1.000   Min.   :   18   Min.   :  928   Min.   : 2743  
 1st Qu.:2   1st Qu.:2.000   1st Qu.: 2348   1st Qu.: 5938   1st Qu.: 9245  
 Median :2   Median :3.000   Median : 5994   Median : 7812   Median :12390  
 Mean   :2   Mean   :2.613   Mean   : 8904   Mean   :10716   Mean   :16323  
 3rd Qu.:2   3rd Qu.:3.000   3rd Qu.:12230   3rd Qu.:12163   3rd Qu.:20184  
 Max.   :2   Max.   :3.000   Max.   :44466   Max.   :73498   Max.   :92780  
     Frozen          Detergents_Paper   Delicassen     
 Min.   :   33.0   Min.   :  332     Min.   :    3.0  
 1st Qu.:  534.2   1st Qu.: 3684     1st Qu.:  566.8  
 Median : 1081.0   Median : 5614     Median : 1350.0  
 Mean   : 1652.6   Mean   : 7270     Mean   : 1753.4  
 3rd Qu.: 2146.8   3rd Qu.: 8662     3rd Qu.: 2156.0  
 Max.   :11559.0   Max.   :40827     Max.   :16523.0  
> 
```

```
corrmatrix <- cor(horeca.data[,-c(1,2)])
corrplot(corrmatrix, method = 'number')
```



```
corrmatrix <- cor(retail.data[,-c(1,2)])
> corrplot(corrmatrix, method = 'number')
```



Examining and Preparing the Data:

We will first check whether there are observations with missing values. If missing values are present, they should be either removed or imputed (imputation is the process of replacing missing data with substituted values). Next, I want to observe the frequency of the categorical variables.

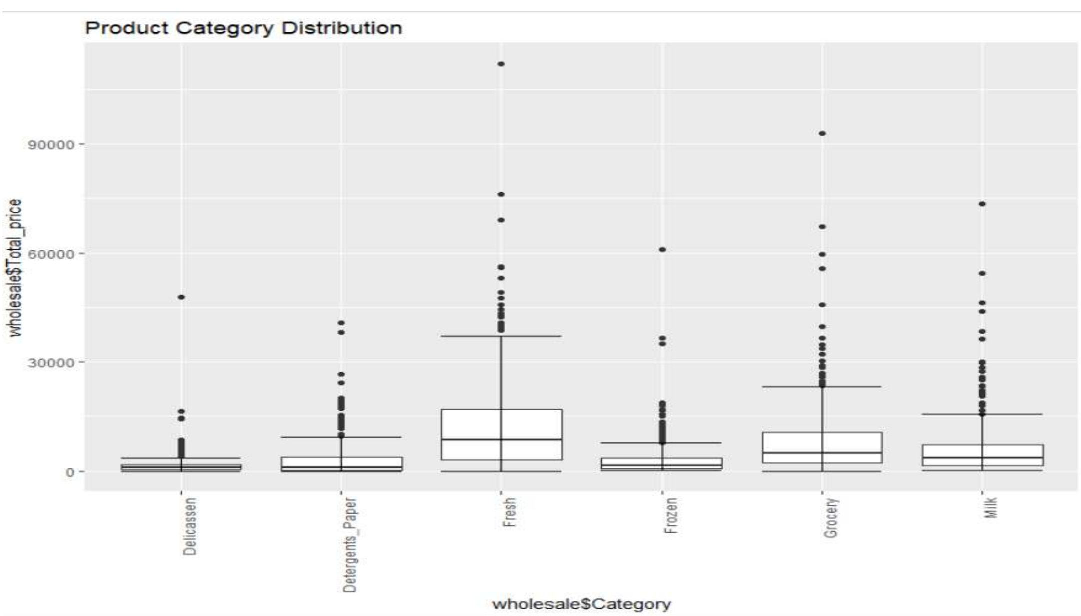
```
> which(complete.cases(data)==F)
integer(0)
```

```
> table(data$Channel)

  1   2
298 142
> table(data$Region)

  1   2   3
 77  47 316
> data %>%
+   group_by(Channel,Region) %>%
+   summarise(total_fresh = sum(data$Fresh), total_Milk = sum(data$Milk),
+             total_Grocery= sum(data$Grocery),total_Frozen=sum(data$Frozen),
+             total_Detergents_Paper=sum(data$Detergents_Paper), total_Delicassen= sum(data$Delicassen))
# A tibble: 6 x 8
# Groups:   Channel [2]
  Channel Region total_fresh total_Milk total_Grocery total_Frozen total_Detergents_Paper total_Delicassen
    <int>  <int>       <int>      <int>         <int>        <int>                  <int>            <int>
1       1      1     5280131    2550357       3498562      1351650                1267857           670943
2       1      2     5280131    2550357       3498562      1351650                1267857           670943
3       1      3     5280131    2550357       3498562      1351650                1267857           670943
4       2      1     5280131    2550357       3498562      1351650                1267857           670943
5       2      2     5280131    2550357       3498562      1351650                1267857           670943
6       2      3     5280131    2550357       3498562      1351650                1267857           670943
```

```
wholesale <- reshape(data, direction="long",
                     varying=c("Fresh","Milk","Grocery","Frozen","Detergents_Paper", "Delicassen"),
                     v.names= "Total_price", timevar="Category",
                     time=c("Fresh", "Milk","Grocery","Frozen","Detergents_Paper", "Delicassen"))
ggplot(wholesale, aes(x=wholesale$Category,
                      y =wholesale$Total_price)) +geom_boxplot() +stat_boxplot(geom ='errorbar') +
                      theme(axis.text.x= element_text(angle=90,hjust=1))+
                      ggtitle("Product Category Distribution")
```



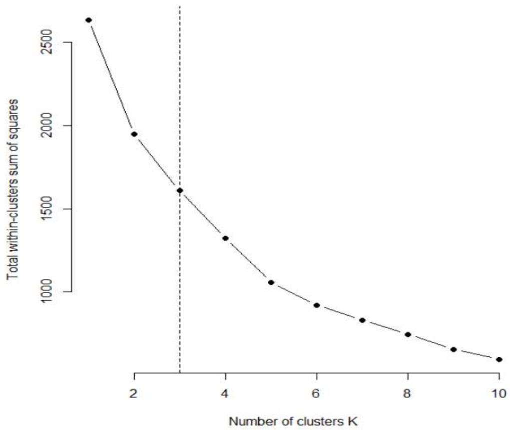
```
> data1<-scale(data[,-c(1,2)])
> summary(data1)
     Fresh              Milk             Grocery             Frozen          Detergents_Paper     Delicassen
 Min.   :-0.9486   Min.   :-0.7779   Min.   :-0.8364   Min.   :-0.62763   Min.   :-0.6037   Min.   :-0.5396
 1st Qu.:-0.7015   1st Qu.:-0.5776   1st Qu.:-0.6101   1st Qu.:-0.47988   1st Qu.:-0.5505   1st Qu.:-0.3960
 Median :-0.2764   Median :-0.2939   Median :-0.3363   Median :-0.31844   Median :-0.4331   Median :-0.1984
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000
 3rd Qu.: 0.3901   3rd Qu.: 0.1889   3rd Qu.: 0.2846   3rd Qu.: 0.09935   3rd Qu.: 0.2182   3rd Qu.: 0.1047
 Max.   : 7.9187   Max.   : 9.1732   Max.   : 8.9264   Max.   :11.90545   Max.   : 7.9586   Max.   :16.4597
> |
```

In this dataset, there are no missing values. The data is clean. There are 298 observations that purchased via Horeca (Hotel/Restaurant/Café) and 142 by retail. On Region, there are 77 annual transactions from Lisbon, 47 from Oporto and 316 from other Regions. Above input shows the sum of cost for different product categories. Fresh and Grocery categories are the top spending. "Channel" and "Region" variables are not related customers' spending behaviors. Although different channel customers have different spending characteristics, channel and region don't affect clustering results. So, I remove the columns of channel and region before clustering. I choose to scale the variables in order to avoid bias although the data only include values measured in monetary units (scale step can be skipped).

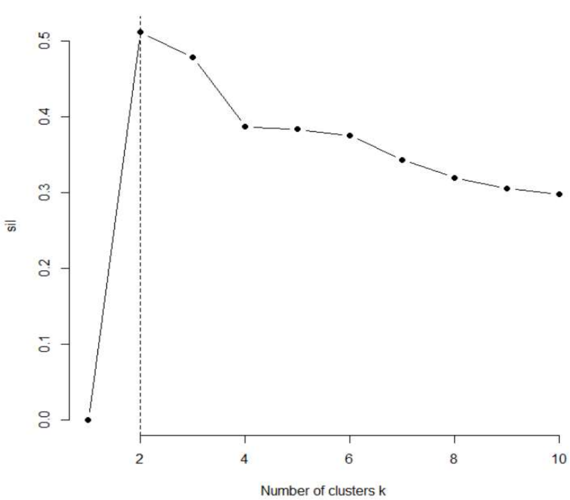
Choose optimal number of clusters:

Determining the **optimal number of clusters** in a data set is a fundamental issue in partitioning clustering. The Elbow method looks at the total WSS as a function of the number of clusters. The idea of the elbow method is to run k-means clustering on the dataset for a range of values of *k* (*k* from 1 to 10 in following code), and for each value of *k* calculate the sum of squared errors (SSE). The average silhouette approach, it measures the quality of a clustering. That is, it determines how well each object lies within its cluster. A high average silhouette width indicates a good clustering. NbClust package provides 30 indices for determining the relevant number of clusters and proposes to users the best clustering scheme from the different results obtained by varying all combinations of number of clusters, distance measures, and clustering methods. The gap statistic compares the total within intra-cluster variation for different values of k with their expected values under null reference distribution of the data. I will use these methods to get optimal number of clusters.

```
> # Elbow method for k-means clustering
> set.seed(123)
> # Compute and plot wss for k = 2 to k = 10
> k.max <- 10 # Maximal number of clusters
> wss <- sapply(1:k.max,
+               function(k){kmeans(data1, k, nstart=10 )$tot.withinss})
> plot(1:k.max, wss,
+      type="b", pch = 19, frame = FALSE,
+      xlab="Number of clusters K",
+      ylab="Total within-clusters sum of squares")
> abline(v = 3, lty =2)
```

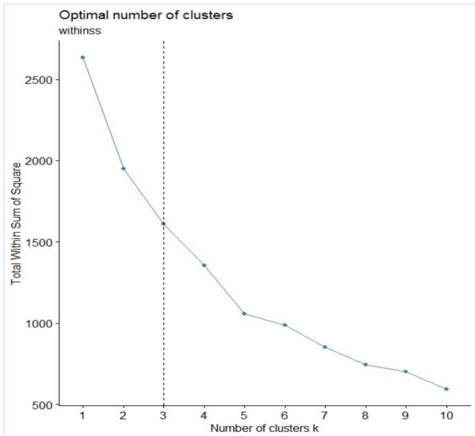



```
> k.max <- 10
> sil <- rep(0, k.max)
> # Compute the average silhouette width for
> # k = 2 to k = 10
> for(i in 2:k.max){
+   km.res <- kmeans(data, centers = i, nstart = 25)
+   ss <- silhouette(km.res$cluster, dist(data))
+   sil[i] <- mean(ss[, 3])
+ }
> # Plot the  average silhouette width
> plot(1:k.max, sil, type = "b", pch = 19,
+      frame = FALSE, xlab = "Number of clusters k")
> abline(v = which.max(sil), lty = 2)
> fviz_nbclust(data1, hcut, method = "silhouette",
+              hc_method = "complete")
>
```

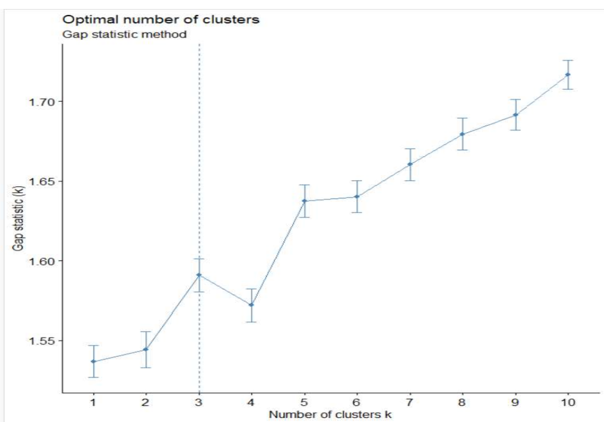



```
# using NbClust for number of clusters

bestK <- NbClust(data1, min.nc=2, max.nc=15, method="kmeans",index='ch')
bestK$Best.nc
fviz_nbclust(data1,kmeans, method="wss")+
  geom_vline(xintercept = 3, linetype=2)+
  labs(subtitle ="withinss")
```




```
> # Gap statistic
> set.seed(123)
> fviz_nbclust(data1, kmeans, nstart = 25,  method = "gap_stat", nboot = 50)+
+   labs(subtitle = "Gap statistic method")
Clustering k = 1,2,..., K.max (= 10): .. done
Bootstrapping, b = 1,2,..., B (= 50)  [one "." per sample]:
```



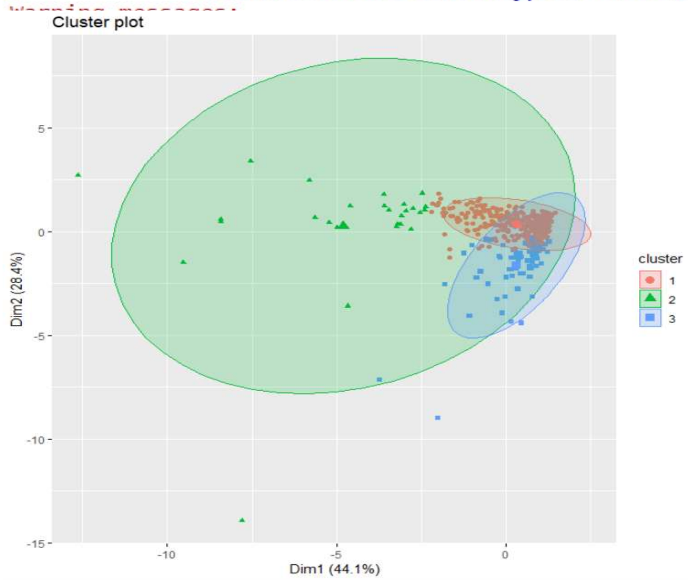
Based on multiple methods for determining the optimal number of clusters, k=2 and k=3 are good for this data set. I would choose K=3 clusters as the following analysis.

Multiple methods clustering results:

K means is an iterative clustering algorithm that aims to find local maxima in each iteration. The k-means algorithm is sensitive to outliers. Hierarchical clustering, as the name suggests it is an algorithm that builds hierarchy of clusters. Hierarchical clustering can't handle big data well. PAM (Partitioning Around Medoid) Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering. PAM works effectively for small data set, but it does not scale well for large data sets (due to the computational complexity).

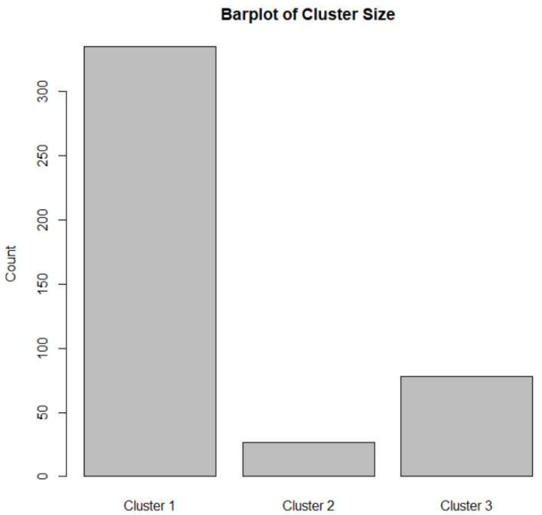
1. Kmeans K=3 clustering

```
> Clusters <- kmeans(data1, 3)
> Clusters$size
[1] 335  27  78
> Clusters$centers
        Fresh       Milk    Grocery      Frozen Detergents_Paper Delicassen
1 -0.34163918 -0.1693818 -0.1568942 -0.23079110       -0.1261322 -0.1556199
2  0.01319547  2.5877169  2.7450998  0.07358958        2.7725390  1.1396932
3  1.46272883 -0.1682749 -0.2763866  0.96574487       -0.4180036  0.2738582
> str(Clusters)
List of 9
 $ cluster     : int [1:440] 1 1 1 1 3 1 1 1 1 1 ...
 $ centers     : num [1:3, 1:6] -0.3416 0.0132 1.4627 -0.1694 2.5877 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:3] "1" "2" "3"
  .. ..$ : chr [1:6] "Fresh" "Milk" "Grocery" "Frozen" ...
 $ totss       : num 2634
 $ withinss    : num [1:3] 525 648 479
 $ tot.withinss: num 1651
 $ betweenss   : num 983
 $ size        : int [1:3] 335 27 78
 $ iter        : int 4
 $ ifault      : int 0
 - attr(*, "class")= chr "kmeans"
> fviz_cluster(Clusters, data = data1, geom = "point",
+              stand = FALSE, frame.type = "norm")
```




```
> barplot(Clusters$size, names.arg=c("Cluster 1","Cluster 2","Cluster 3"), ylab="Count",
+         main="Barplot of Cluster Size")
> Clusters$size
[1] 335  27  78
> table(Clusters$size)

 27  78 335
  1   1   1
> cluster1size<-((Clusters$size)[1]/sum(Clusters$size))*100
> cluster2size<-((Clusters$size)[2]/sum(Clusters$size))*100
> cluster3size<-((Clusters$size)[3]/sum(Clusters$size))*100
> Clusters$size
[1] 335  27  78
> cmatrix <- cbind(cluster1size,cluster2size,cluster3size)
> cmatrix
     cluster1size cluster2size cluster3size
[1,]     76.13636     6.136364     17.72727
```




```
> mydata<-data %>%
+ mutate(CLUSTER = Clusters$cluster)
> library(dplyr)
> df<-mydata %>%
+   group_by(CLUSTER) %>%
+   summarise(fresh = mean(Fresh), milk = mean(Milk),
+             grocery= mean(Grocery),frozen=mean(Frozen),
+             detergents_Paper=mean(Detergents_Paper),
+             delicassen= mean(Delicassen))
> df
# A tibble: 3 x 7
  CLUSTER  fresh   milk grocery frozen detergents_Paper delicassen
    <int>  <dbl>  <dbl>   <dbl>  <dbl>            <dbl>      <dbl>
1       1  7679.  4546.   6460.  1952.            2280.      1086.
2       2 12167. 24895.  34038.  3429.           16101.      4739.
3       3 30500.  4554.   5325.  7760.             889.      2297.
```

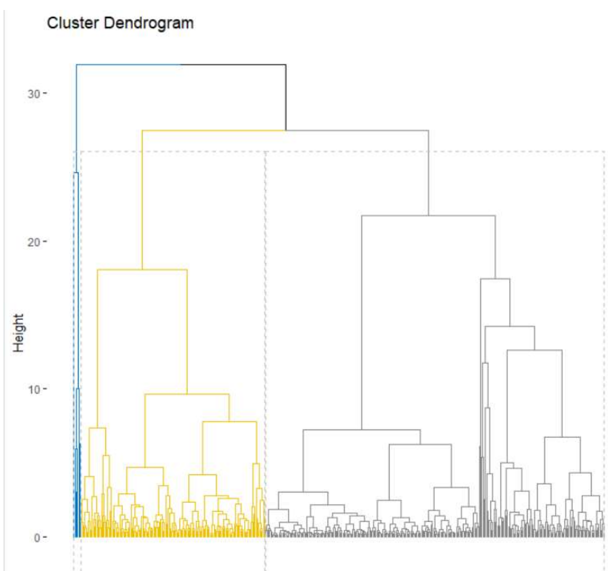
2. PAM k=3 clustering:

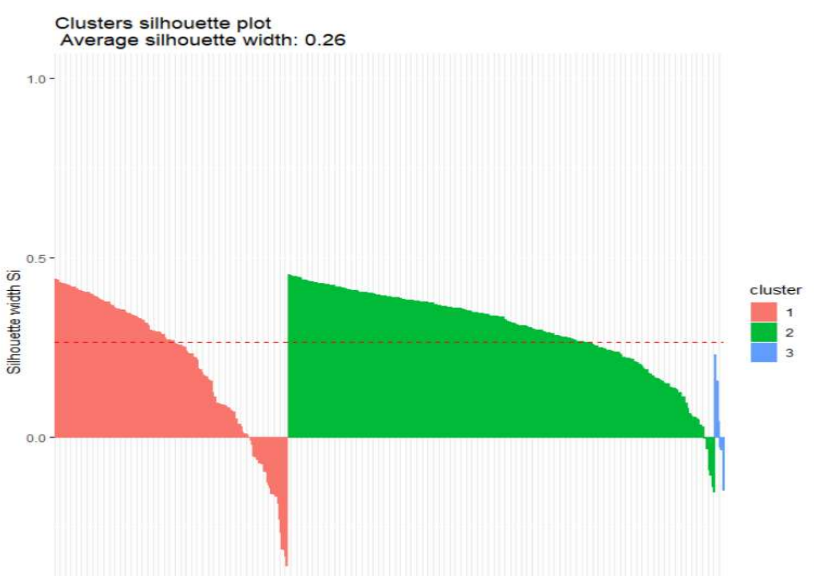
```
> summary(pam.clusters)
Medoids:
      ID      Fresh       Milk    Grocery     Frozen Detergents_Paper Delicassen
[1,] 375 -0.4477070 -0.4796863 -0.6611775 -0.3254455       -0.5391300 -0.2974606
[2,]  10 -0.4739576  0.7176780  1.1501142 -0.3940392        0.9529458  0.2032298
[3,] 119  0.6363954 -0.5291418 -0.5881492  0.4678107       -0.5181562 -0.2098753
Clustering vector:

Numerical information per cluster:
     size  max_diss   av_diss  diameter separation
[1,]  204  2.740689 0.7557963  3.286193  0.1566052
[2,]  108 18.585660 2.0531932 21.244265  0.1566052
[3,]  128 11.835609 1.5974184 12.871817  0.2443120
```

3. Hierarchical clustering:

```
> hc <- data1 %>%
+
+   eclust("hclust", k = 3, graph = FALSE)
> # Visualize with factoextra
> fviz_dend(hc, palette = "jco",
+           rect = TRUE, show_labels = FALSE)
> fviz_silhouette(hc)
  cluster size ave.sil.width
1       1  153          0.21
2       2  281          0.30
3       3    6          0.04
>
```





Result Analysis and Recommendation:

According to k-means, k- medoid (PAM), and hierarchical clustering, the customers groups indicate some different features. I use K-mean clusters as example to interpret.
When running the k-means algorithm in R, three distinct customer profiles emerge. The cluster centers are listed for each of the types of products and cluster assignments. The k-means approach calculates cluster centers based on the average values across the variables. The output generates clusters that are most mathematically homogeneous within the clusters, and distinct between clusters.

Cluster size of K-means clustering:

	Cluster1	Cluster2	Cluster3
Size (numbers of customers)	**335**	27	78
Percent (%)	**76.14**	6.14	17.72

Average spending on all products for each of clusters:

	Fresh	Milk	Grocery	Frozen	Detergents_Paper	Delicassen
Cluster1	7679	4546	6460	1952	2280	1086
Cluster2	12167	24895	34038	3429	16101	4739
Cluster3	30500	4554	5325	7760	889	2297

- Cluster 1: customer size is largest, but spending is less on all products.
- Cluster 2: Prefers grocery, milk and detergents/paper.
- Cluster 3: Prefers fresh foods, frozen foods.

These results indicate that fresh and frozen foods tend to be purchased in large amounts by similar customers, whereas groceries, milk and detergents/paper are linked to another large group of customers. Another group of customers tends to spend less overall on all products.

When mapped back to original data for the customer types, the results further reveal that cluster 3 is mostly hotels, restaurants and cafés, who, logically, tend to order raw food such as fresh and frozen foods for cooking, whereas cluster 2 is largely comprised of retail customers such as grocery stores, which would explain for purchasing large amount at grocery, milk and detergents/paper. The lower average spending cluster 1 is the largest group, 76% of total customers, which may indicate that this customer purchasers are smaller restaurants or cafe rather than larger retailers. We can make a conclusion that the retail stores and horeca have different requirement for products.

This clustering analysis helps wholesalers more efficiently target their products for different customer groups. For example, wholesalers can arrange fresh and frozen products at same area to meet similar customers' demand. Moreover, new fresh and frozen products should market towards restaurants, hotels and cafes; grocery and milk target to retailers. This would greatly reduce market and service cost. On the other hand, customer clustering analysis also helps wholesaler to order inventory by knowing that certain types of products are often bought together. For example, wholesalers can order grocery, milk, and detergents together for reducing shipping cost.

So, clustering analysis is an important method in data mining. it is broadly used in many applications such as market research, pattern recognition, data analysis, and image processing.

Reference:

e-textbook: Data Mining Algorithms In R
https://bb.csueastbay.edu/bbcswebdav/pid-4586312-dt-content-rid-49505176_1/xid-49505176_1

http://www.kimberlycoffey.com/blog/2016/8/k-means-clustering-for-customer-segmentation
http://www.kimberlycoffey.com/blog/2016/8/k-means-clustering-for-customer-segmentation
https://rstudio-pubs-static.s3.amazonaws.com/183721_c69be1add01344bd816c67b086de8454.html

https://developer.ibm.com/articles/os-weka2

https://www.r-bloggers.com/in-depth-introduction-to-machine-learning-in-15-hours-of-expert-videos/