Wholesale Customers

1. Introduction

This study presents the customers´ buying behaviors on diverse products in a wholesale distributor, by segmenting the dataset in clusters according to each buying group characteristic.

To section the data set, the methodology to be applied is Kmeans, an unsupervised machine learning technique to group the similar data, and to segment and find the appropriate number of clusters, some methodologies will be used and then, this calculated number of groups will be applied on the Kmeans algorithm to indicate the most relevant clusters, regions and channels (see attribute information) through the total average spent by the customers for each product.

2. Attribute information

a. FRESH: annual spending (m.u.) on fresh products (Continuous)
b. MILK: annual spending (m.u.) on milk products (Continuous)
c. GROCERY: annual spending (m.u.) on grocery products (Continuous)
d. FROZEN: annual spending (m.u.) on frozen products (Continuous)
e. DETERGENTS_PAPER: annual spending (m.u.) on detergents and paper products (Continuous)
f. DELICATESSEN: annual spending (m.u.) on delicatessen products (Continuous)
g. CHANNEL: customers Channel - Horeca (Hotel/Restaurant/Cafe) or Retail channel (Nominal)
h. REGION: customers Region - Lisbon, Porto or Other (Nominal)

REGION Frequency: Lisbon 77, Porto 47, Other Region 316, Total 440

CHANNEL Frequency: Horeca 298, Retail 142, Total 440

3. Importing the data and categorizing some variables

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

raw.data$Channel <- as.factor(raw.data$Channel)
raw.data$Region <- as.factor(raw.data$Region)
summary(raw.data)
```

```
##  Channel Region      Fresh             Milk           Grocery
##  1:298   1: 77   Min.   :     3   Min.   :   55   Min.   :    3
##  2:142   2: 47   1st Qu.:  3128   1st Qu.: 1533   1st Qu.: 2153
##          3:316   Median :  8504   Median : 3627   Median : 4756
##                  Mean   : 12000   Mean   : 5796   Mean   : 7951
##                  3rd Qu.: 16934   3rd Qu.: 7190   3rd Qu.:10656
##                  Max.   :112151   Max.   :73498   Max.   :92780
##

        Frozen          Detergents_Paper     Delicassen
##  Min.   :   25.0   Min.   :    3.0   Min.   :    3.0
##  1st Qu.:  742.2   1st Qu.:  256.8   1st Qu.:  408.2
##  Median : 1526.0   Median :  816.5   Median :  965.5
##  Mean   : 3071.9   Mean   : 2881.5   Mean   : 1524.9
##  3rd Qu.: 3554.2   3rd Qu.: 3922.0   3rd Qu.: 1820.2
##  Max.   :60869.0   Max.   :40827.0   Max.   :47943.0
```

```
head(raw.data)
```

```
##   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
```

4. Applying Gower´s distance

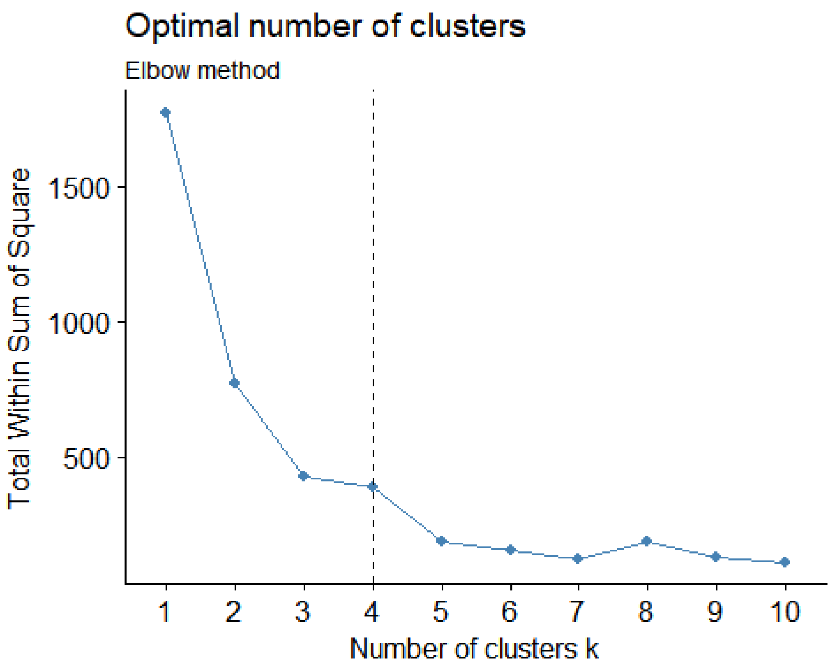
```
library(cluster)
wholesale.dist <- daisy(raw.data, metric = 'gower')
wholesale.matrix <- as.matrix(wholesale.dist)
```

5. Determining the optimal number of clusters by using:

5.1 Elbow method

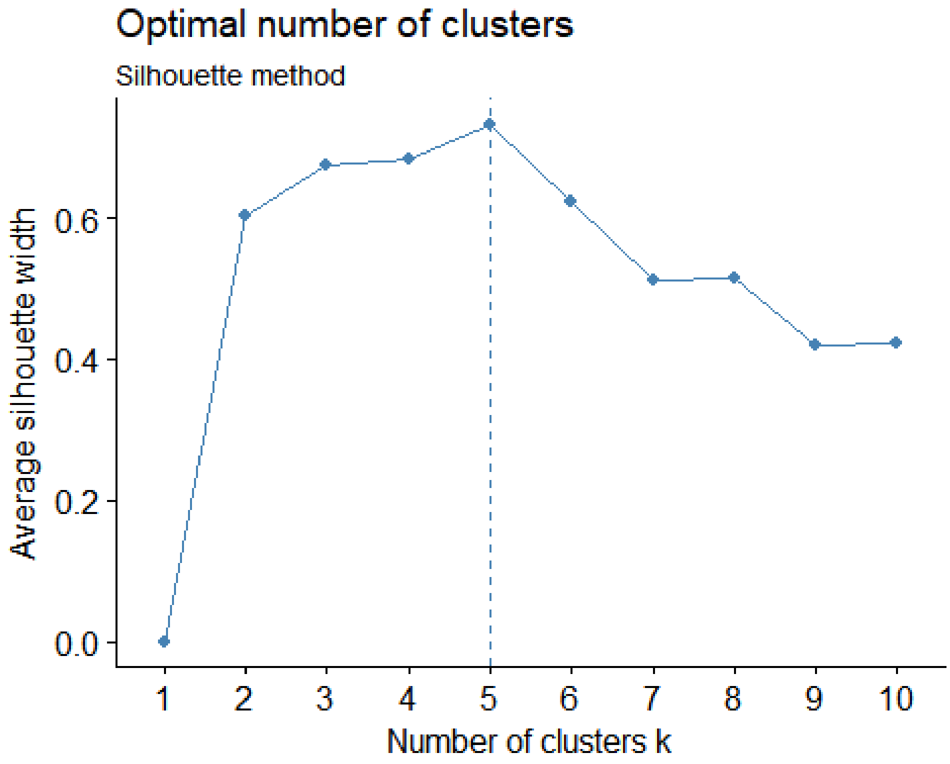
```
library(factoextra)

fviz_nbclust(wholesale.matrix, kmeans, method = "wss") +
  geom_vline(xintercept = 4, linetype = 2)+
  labs(subtitle = "Elbow method")
```



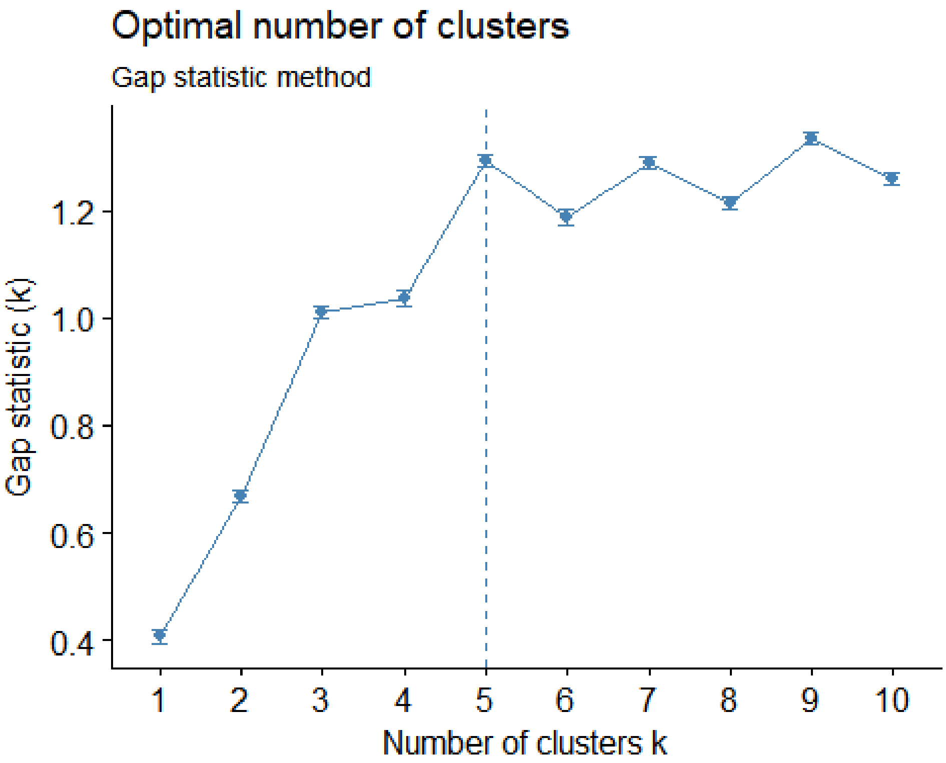
5.2 Silhouette method

```
fviz_nbclust(wholesale.matrix, kmeans, method = "silhouette")+
  labs(subtitle = "Silhouette method")
```



5.3 Gap statistic method

```r
# nboot = 50 to keep the function speedy.
# recommended value: nboot= 500 for your analysis.
# Use verbose = FALSE to hide computing progression.
fviz_nbclust(wholesale.matrix, kmeans,  method = "gap_stat")+
  labs(subtitle = "Gap statistic method")
```

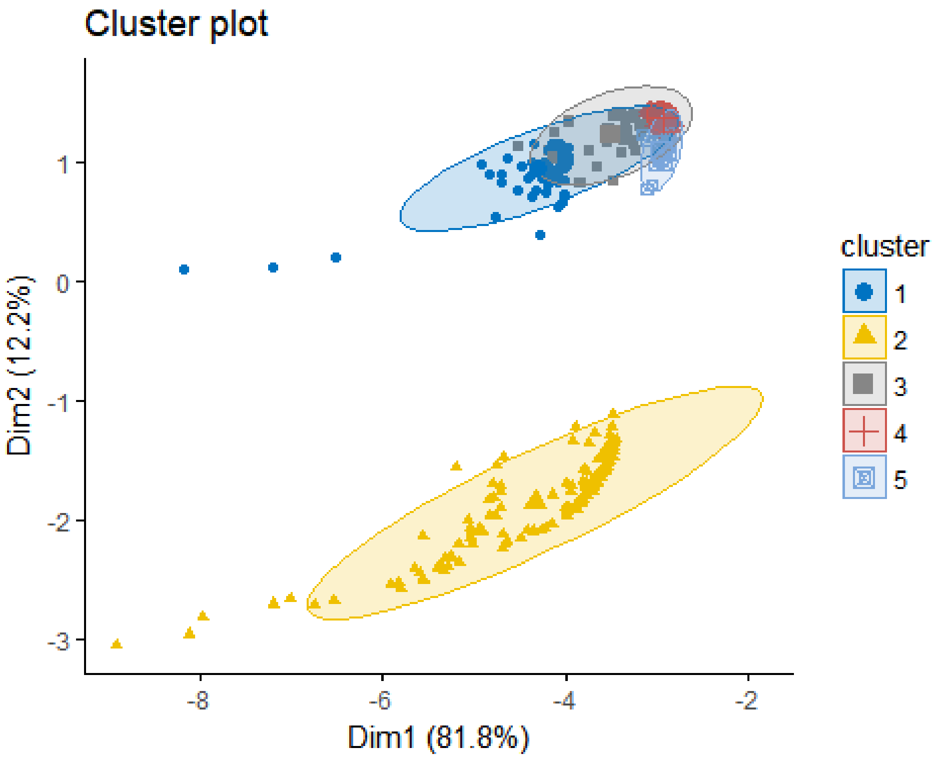


Note: According to the applied methods, it is suitable to utilize **5 centers**.

6. Applying Kmeans algorithm and visualizing the clusters

```r
library(factoextra)
set.seed(55)
km.res1 <- kmeans(wholesale.matrix, 5, iter.max = 5000)

fviz_cluster(list(data = wholesale.matrix, cluster = km.res1$cluster),
             ellipse.type = 'norm',
             geom = "point", stand = FALSE,
             palette = "jco", ggtheme = theme_classic())
```



7. Counting the quantities per clusters

```
library(dplyr)

raw.data$Cluster <- km.res1$cluster
raw.data$Cluster <- as.factor(raw.data$Cluster)
count(raw.data, Cluster)

## # A tibble: 5 x 2
##   Cluster     n
##   <fct>   <int>
## 1 1          89
## 2 2         142
## 3 3          35
## 4 4         145
## 5 5          29
```

8. Determining the most relevant clusters

The total values represent the sum of the averages spent by the customers for each product per cluster, region and channel.

```
z <- raw.data %>%
  group_by(Cluster,Region,Channel) %>%
  summarise_all(funs(mean(.))) %>%
  rowwise() %>%
  mutate(Total = sum(c(Fresh, Milk, Grocery, Frozen, Detergents_Paper, Delicassen)))
%>%
  arrange(desc(Total))

  z %>%
    select(Cluster, Region, Channel, Total)
```

```
## # A tibble: 9 x 4
##   Cluster Region Channel  Total
##   <fct>   <fct>  <fct>    <dbl>
## 1 1       3      1       187926
## 2 3       3      1        55823
## 3 2       1      2        47137
## 4 2       3      2        47005
## 5 2       2      2        43997
## 6 1       1      1        26074
## 7 1       2      1        25684
## 8 5       3      1        25042
## 9 4       3      1        18526
```

9. Total spent values per Region and Channel

```
a <- raw.data %>%
  rowwise() %>%
  mutate(Total = sum(c(Fresh, Milk, Grocery, Frozen, Detergents_Paper, Delicassen)))
%>%
  filter(Region == 1, Channel == 1) %>%
  arrange(desc(Total)) %>%
  summarise(TotalValue = sum(Total))

b <- raw.data %>%
  rowwise() %>%
  mutate(Total = sum(c(Fresh, Milk, Grocery, Frozen, Detergents_Paper, Delicassen)))
%>%
  filter(Region == 1, Channel == 2) %>%
  arrange(desc(Total)) %>%
  summarise(TotalValue = sum(Total))

c <- raw.data %>%
  rowwise() %>%
  mutate(Total = sum(c(Fresh, Milk, Grocery, Frozen, Detergents_Paper, Delicassen)))
%>%
  filter(Region == 2, Channel == 1) %>%
  arrange(desc(Total)) %>%
  summarise(TotalValue = sum(Total))
```

```
d <- raw.data %>%
  rowwise() %>%
  mutate(Total = sum(c(Fresh, Milk, Grocery, Frozen, Detergents_Paper, Delicassen)))
%>%
  filter(Region == 2, Channel == 2) %>%
  arrange(desc(Total)) %>%
  summarise(TotalValue = sum(Total))

e <- raw.data %>%
  rowwise() %>%
  mutate(Total = sum(c(Fresh, Milk, Grocery, Frozen, Detergents_Paper, Delicassen)))
%>%
  filter(Region == 3, Channel == 1) %>%
  arrange(desc(Total)) %>%
  summarise(TotalValue = sum(Total))

f <- raw.data %>%
  rowwise() %>%
  mutate(Total = sum(c(Fresh, Milk, Grocery, Frozen, Detergents_Paper, Delicassen)))
%>%
  filter(Region == 3, Channel == 2) %>%
  arrange(desc(Total)) %>%
  summarise(TotalValue = sum(Total))

a <- as.numeric(a[1,1])
b <- as.numeric(b[1,1])
c <- as.numeric(c[1,1])
d <- as.numeric(d[1,1])
e <- as.numeric(e[1,1])
f <- as.numeric(f[1,1])

table.format <- matrix(c(a,b,c,d,e,f), byrow = TRUE, ncol = 2)
colnames(table.format) <- c('Channel1', 'Channel2')
row.names(table.format) <- c('Region1', 'Region2', 'Region3')
table.format

##         Channel1 Channel2
## Region1  1538342   848471
## Region2   719150   835938
## Region3  5742077  4935522
```

10. Proportion of the total values per Region and Channel

```
prop.table(table.format)

##           Channel1   Channel2
## Region1 0.10522535 0.05803694
## Region2 0.04919115 0.05717966
## Region3 0.39276836 0.33759855
```

11. Total spent values per Cluster

```
g <- raw.data %>%
  rowwise() %>%
  mutate(Total = sum(c(Fresh, Milk, Grocery, Frozen, Detergents_Paper, Delicassen)))
%>%
  filter(Cluster == 1) %>%
  arrange(desc(Total)) %>%
  summarise(TotalValue = sum(Total))

h <- raw.data %>%
  rowwise() %>%
  mutate(Total = sum(c(Fresh, Milk, Grocery, Frozen, Detergents_Paper, Delicassen)))
%>%
  filter(Cluster == 2) %>%
  arrange(desc(Total)) %>%
  summarise(TotalValue = sum(Total))

i <- raw.data %>%
  rowwise() %>%
  mutate(Total = sum(c(Fresh, Milk, Grocery, Frozen, Detergents_Paper, Delicassen)))
%>%
  filter(Cluster == 3) %>%
  arrange(desc(Total)) %>%
  summarise(TotalValue = sum(Total))

j <- raw.data %>%
  rowwise() %>%
  mutate(Total = sum(c(Fresh, Milk, Grocery, Frozen, Detergents_Paper, Delicassen)))
%>%
  filter(Cluster == 4) %>%
  arrange(desc(Total)) %>%
  summarise(TotalValue = sum(Total))

k <- raw.data %>%
  rowwise() %>%
  mutate(Total = sum(c(Fresh, Milk, Grocery, Frozen, Detergents_Paper, Delicassen)))
%>%
  filter(Cluster == 5) %>%
  arrange(desc(Total)) %>%
  summarise(TotalValue = sum(Total))

g <- as.numeric(g[1,1])
h <- as.numeric(h[1,1])
i <- as.numeric(i[1,1])
j <- as.numeric(j[1,1])
k <- as.numeric(k[1,1])


table.format1 <- matrix(c(g,h,i,j,k), byrow = TRUE, ncol = 1)
rownames(table.format1) <- c('Cluster1', 'Cluster2', 'Cluster3', 'Cluster4',
'Cluster5')
```

```
colnames(table.format1) <- c('TotalValue')


table.format1

##            TotalValue
## Cluster1    2633344
## Cluster2    6619931
## Cluster3    1953811
## Cluster4    2686208
## Cluster5     726206
```

12. Proportion of the total values per Cluster

```
l <- prop.table(table.format1)
colnames(l) <- c('Proportion')
l

##          Proportion
## Cluster1 0.18012545
## Cluster2 0.45281514
## Cluster3 0.13364417
## Cluster4 0.18374144
## Cluster5 0.04967379
```

13. Example of business segmentation

To evaluate a specific product, it can be selected any product and its respective segment to assess the data. For example, analyzing the Delicassen product and segmenting the total sells to each cluster, as shown below.

```
raw.data %>%
  select(Delicassen, Cluster) %>%
  group_by(Cluster) %>%
  summarise(TotalValue = sum(Delicassen))

## # A tibble: 5 x 2
##   Cluster TotalValue
##   <fct>        <int>
## 1 1           158090
## 2 2           248988
## 3 3            93973
## 4 4           131975
## 5 5            37917
```

14. Conclusion

In this study, it was substantiated the usefulness of the Kmeans methodology by segmenting the customers behaviors into similar clusters illustrated by the total average spent for each product, it can be helpful to business strategy, assisting on the decisions of the most relevant clusters that shall be observed and where to focus the investments.