Practicum 3

Jordan Lian

4/20/2021

Problem 2 (40 pts)

1. (0 pts) Download this data set on Whole Sale Customers (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: customers Channel - Horeca (Hotel/Restaurant/Cafe) or Retail channel (Nominal)
8) REGION: customers Region - Lisbon, Oporto or Other (Nominal)

Descriptive Statistics:

1. (Minimum, Maximum, Mean, Std. Deviation)

- FRESH (3, 112151, 12000.30, 12647.329)
- MILK (55, 73498, 5796.27, 7380.377)
- GROCERY (3, 92780, 7951.28, 9503.163)
- FROZEN (25, 60869, 3071.93, 4854.673)
- DETERGENTS_PAPER (3, 40827, 2881.49, 4767.854)
- DELICATESSEN (3, 47943, 1524.87, 2820.106)

2. REGION Frequency

- Lisbon 77
- Oporto 47
- Other Region 316
- Total 440

3. CHANNEL Frequency

- Horeca 298
- Retail 142

- Total 440

```
origin_customers <- read.csv('Wholesale customers data.csv')
customers <- origin_customers
# customers <- scale(origin_customers)
head(customers)
```

```
##   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
```

2. (0 pts) Build a new R Notebook named DA5030.P3-2.LastName.Rmd, where LastName is your last name.
3. (40 pts) Using an implementation of your choice of the k-means algorithm, determine clusters that may exist. Define 2, 3, and 4 clusters. What are some of the characteristics of the determined clusters? How would you label them?

I used a few different methods using fviz_nbclust() and NbClust() to determine the optimal amount of clusters.

Libraries

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.0.5
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.0.4
```

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

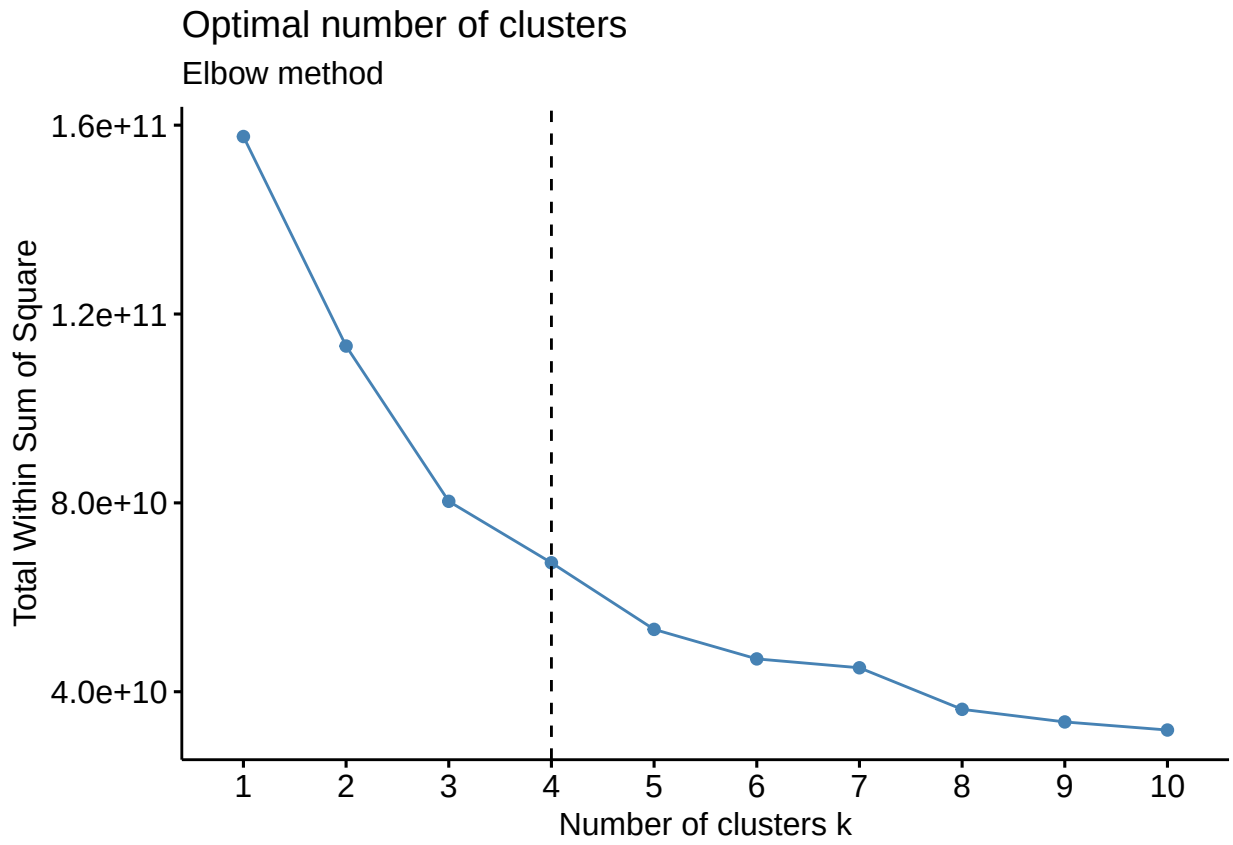
```
library(NbClust)
```

```
## Warning: package 'NbClust' was built under R version 4.0.3
```

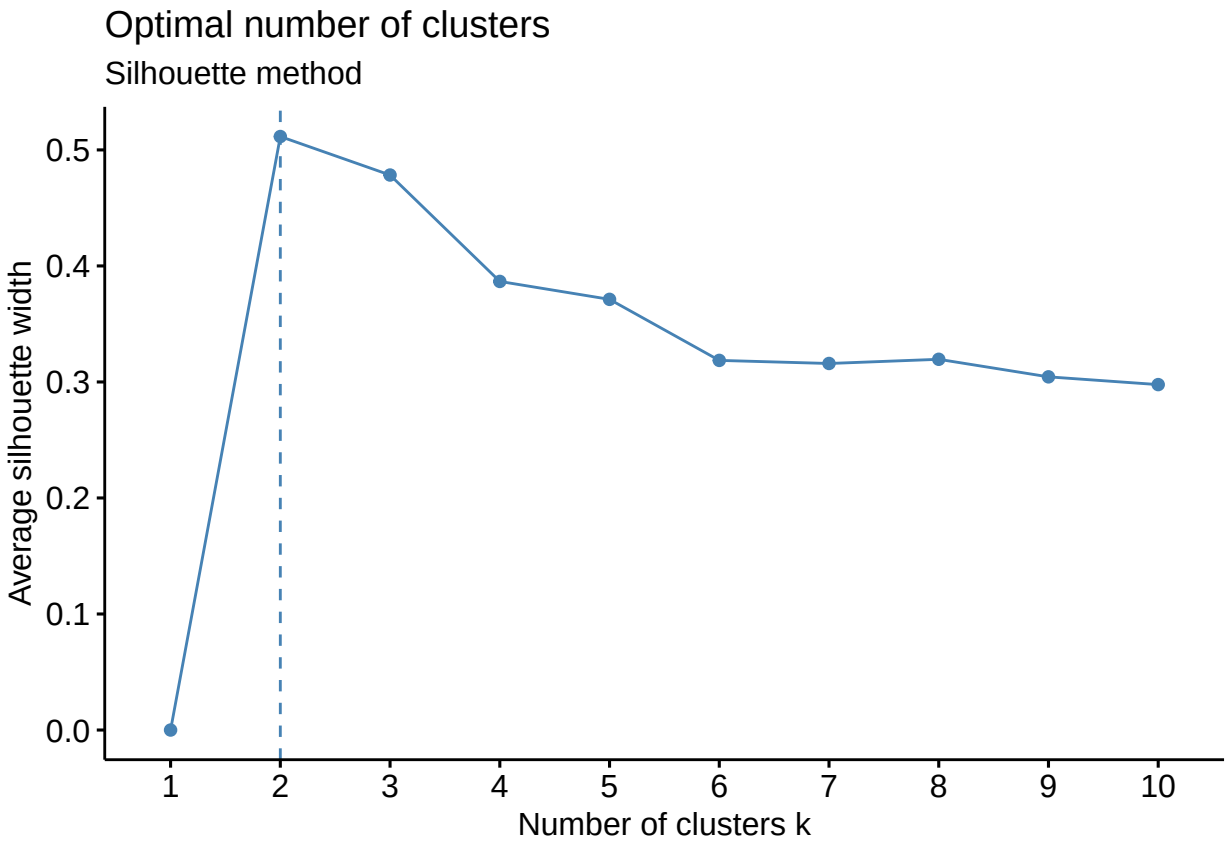
fviz_nbclust()

The Elbow and Silhouette methods had similar conclusions, while the Gap statistic method suggested to use 10 clusters. However, this was an outlier compared to the other values (2 and 4 respectively).

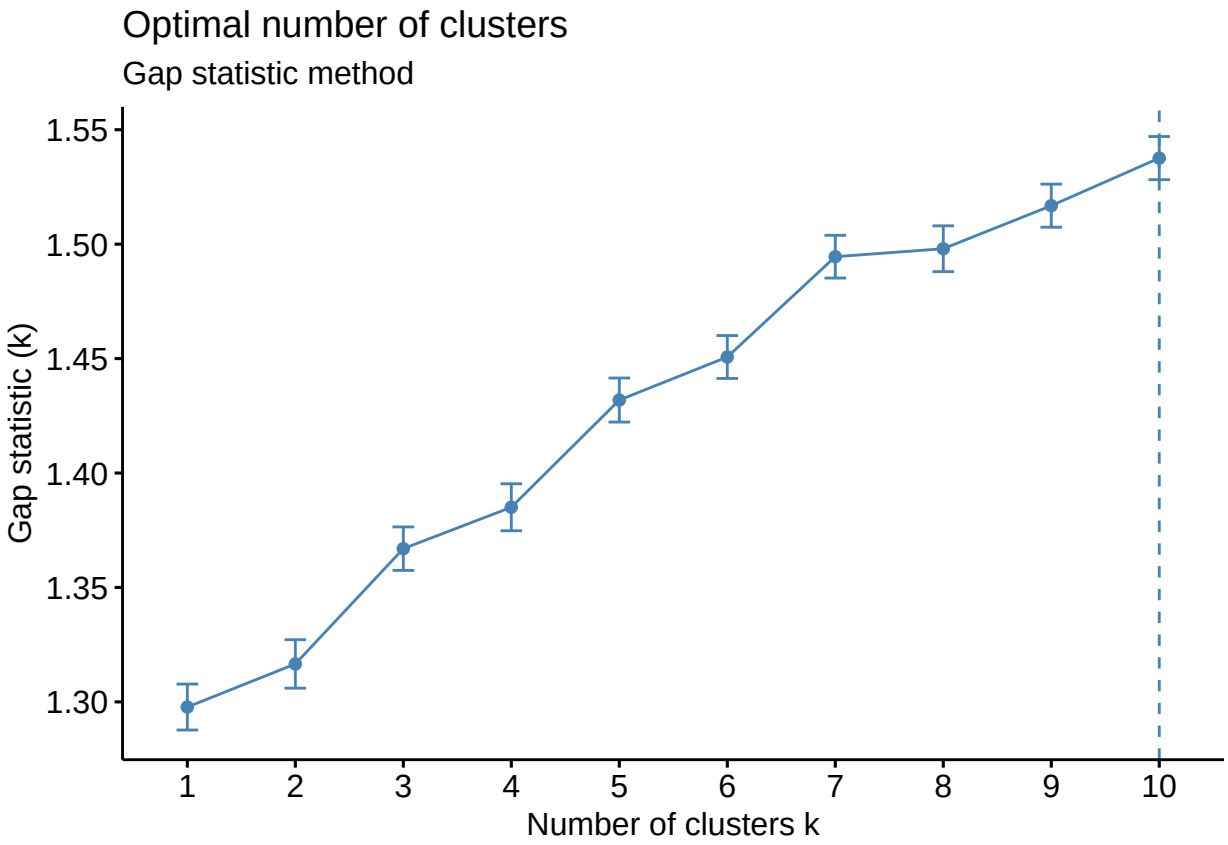
```
# Elbow method
fviz_nbclust(customers, kmeans, method = "wss") +
    geom_vline(xintercept = 4, linetype = 2)+
  labs(subtitle = "Elbow method")
```




```
# Silhouette method
fviz_nbclust(customers, kmeans, method = "silhouette")+
  labs(subtitle = "Silhouette method")
```



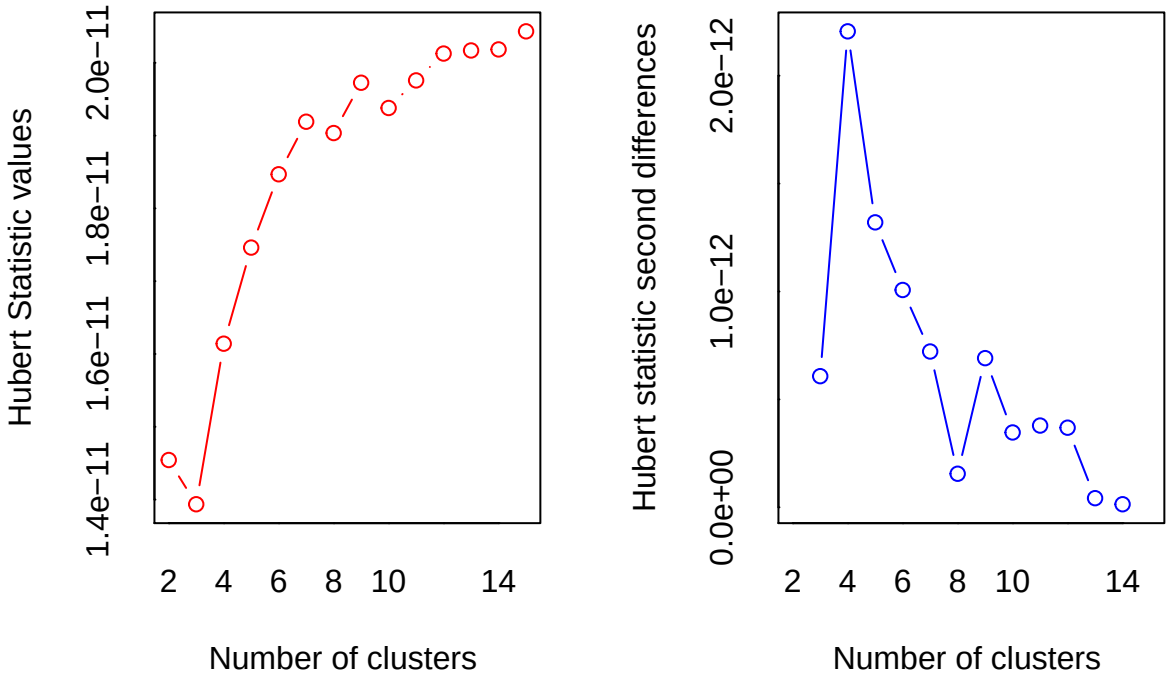

```
# Gap statistic
set.seed(123)
fviz_nbclust(customers, kmeans, nstart = 25,  method = "gap_stat", nboot = 50)+
  labs(subtitle = "Gap statistic method")
```



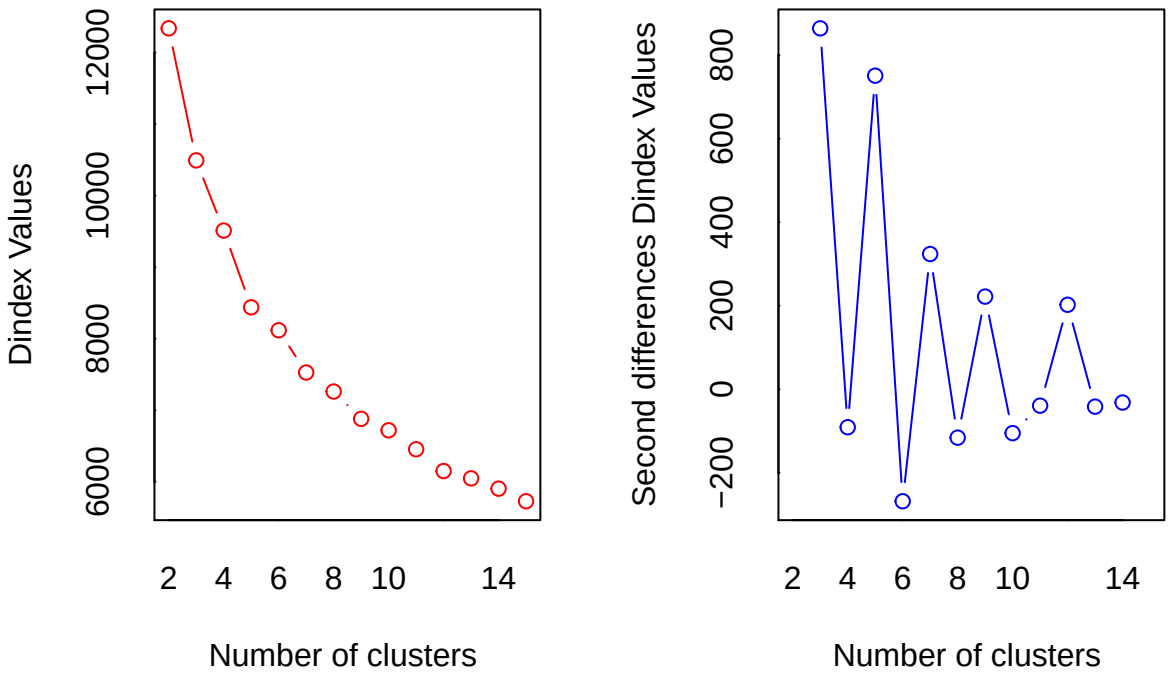
NbClust()

```
clusters <- NbClust(data = customers, diss = NULL, distance = "euclidean",
                    method = "kmeans")
```

```
## Warning in pf(beale, pp, df2): NaNs produced
```




```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```




```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 5 proposed 2 as the best number of clusters
## * 8 proposed 3 as the best number of clusters
## * 3 proposed 4 as the best number of clusters
## * 2 proposed 5 as the best number of clusters
## * 3 proposed 8 as the best number of clusters
## * 1 proposed 9 as the best number of clusters
## * 1 proposed 10 as the best number of clusters
## * 1 proposed 15 as the best number of clusters
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  3
##
##
## *******************************************************************
```

```
clusters$Best.partition
```

```
##   [1] 3 3 3 3 1 3 3 3 3 2 3 3 1 3 1 3 3 3 3 3 3 3 3 1 2 1 3 3 3 2 1 3 3 3 1 3 3 1
```

```
##  [38] 3 2 1 1 3 3 2 3 2 2 2 3 2 3 3 1 3 1 3 2 3 3 3 3 2 3 3 3 2 3 3 3 3 3 3 3 3
##  [75] 3 3 3 2 3 3 3 3 3 3 3 2 2 1 3 1 3 3 2 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 2 3
## [112] 2 3 3 3 3 3 3 3 3 3 3 3 3 1 1 3 3 3 1 3 3 3 3 3 3 3 3 3 3 1 1 3 3 2 3 3
## [149] 3 1 3 3 3 3 3 3 2 3 3 3 3 3 3 3 2 3 2 3 3 3 3 3 2 3 2 3 3 1 3 3 3 3 1 3 1 3
## [186] 3 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 2 2 1 3 3 2 3 3 3 2 3 2 3 3 3 3 2 3 3 3 3 3
## [223] 3 3 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 1 1 1 3 3 3 3 3 3 3 3 3 3 2 3 1 3 1 3 3 1
## [260] 1 3 3 1 3 3 2 2 3 2 3 3 3 3 1 3 3 1 3 3 3 3 3 3 1 1 1 1 3 3 3 1 3 3 3 3 3 3
## [297] 3 3 3 3 3 3 2 3 3 2 3 2 3 3 2 3 1 2 3 3 3 3 3 3 2 3 3 3 3 1 1 3 3 3 3 3 2 3
## [334] 2 3 1 3 3 3 3 3 3 3 2 3 3 3 1 3 2 3 2 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [371] 1 3 3 3 3 3 3 3 1 3 3 1 3 1 3 2 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 1 1 1 3 3 1
## [408] 2 3 3 3 3 3 3 3 3 3 3 3 2 3 3 3 1 3 3 3 3 3 1 3 3 3 3 3 3 3 3 1 1 2 3 3
```

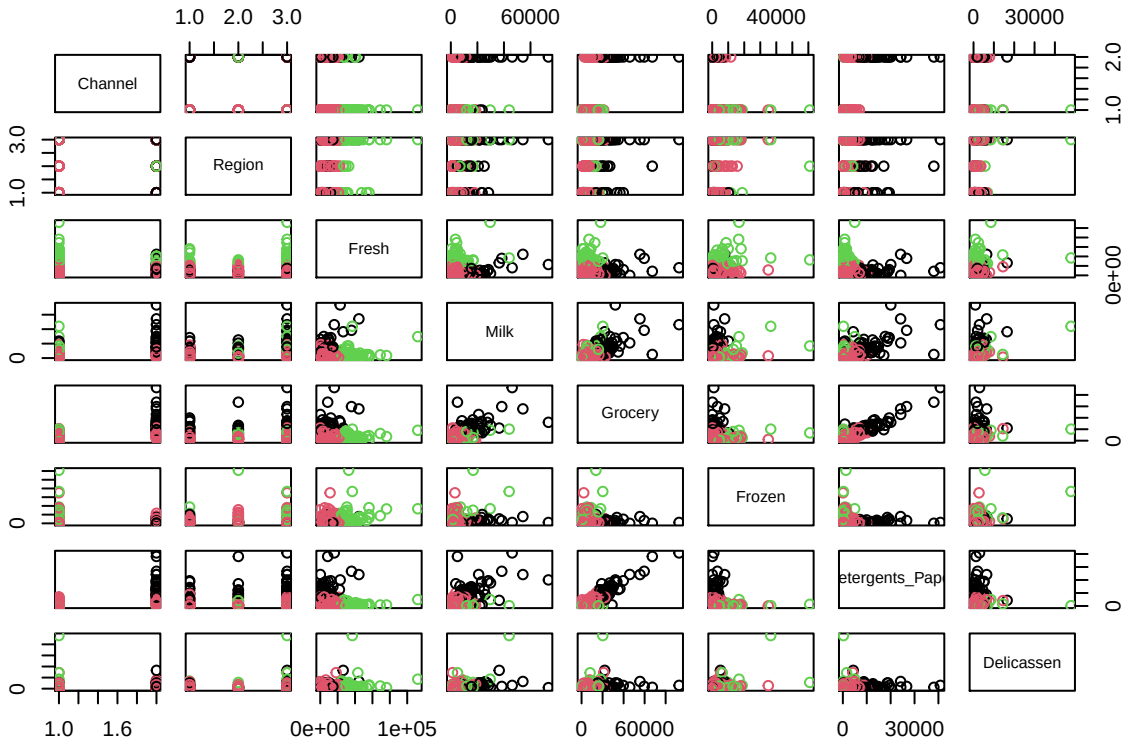
Based off of the 4 functions, I decided to go with 3 clusters and define/characterize them. I used kmeans to define the parameters for the clusters. I initially tried out using 2 clusters, but one of the clusters seemed too big.

```
# run K-Means
km <- kmeans(customers, 3, 15)

# print components of km
print(km)
```

```
## K-means clustering with 3 clusters of sizes 50, 330, 60
##
## Cluster means:
##    Channel   Region    Fresh      Milk   Grocery   Frozen Detergents_Paper
## 1 1.960000 2.440000  8000.04 18511.420 27573.900 1996.680        12407.360
## 2 1.260606 2.554545  8253.47  3824.603  5280.455 2572.661         1773.058
## 3 1.133333 2.566667 35941.40  6044.450  6288.617 6713.967         1039.667
##   Delicassen
## 1   2252.020
## 2   1137.497
## 3   3049.467
##
## Clustering vector:
##   [1] 2 2 2 2 3 2 2 2 2 1 2 2 3 2 3 2 2 2 2 2 2 2 3 1 3 2 2 2 1 3 2 2 2 3 2 2 3
##  [38] 2 1 3 3 2 2 1 2 1 1 1 2 1 2 2 3 2 3 2 1 2 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2
##  [75] 2 2 2 1 2 2 2 2 2 2 2 1 1 3 2 3 2 2 1 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 1 2
## [112] 1 2 2 2 2 2 2 2 2 2 2 2 2 3 3 2 2 2 3 2 2 2 2 2 2 2 2 2 2 3 3 2 2 1 2 2
## [149] 2 3 2 2 2 2 2 2 1 2 2 2 2 2 2 2 1 2 1 2 2 2 2 2 1 2 1 2 2 3 2 2 2 2 3 2 3 2
## [186] 2 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 1 1 3 2 2 1 2 2 2 1 2 1 2 2 2 2 1 2 2 2 2 2
## [223] 2 2 2 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 3 3 3 2 2 2 2 2 2 2 2 1 2 3 2 3 2 2 3
## [260] 3 2 2 3 2 2 1 1 2 1 2 2 2 2 3 2 2 3 2 2 2 2 2 3 3 3 3 2 2 2 3 2 2 2 2 2 2
## [297] 2 2 2 2 2 2 1 2 2 1 2 1 2 2 1 2 3 1 2 2 2 2 2 2 1 2 2 2 2 3 3 2 2 2 2 2 1 2
## [334] 1 2 3 2 2 2 2 2 2 2 1 2 2 2 3 2 1 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [371] 3 2 2 2 2 2 2 2 3 2 2 3 2 3 2 1 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 3 3 3 2 2 3
## [408] 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 3 2 2 2 2 3 2 2 2 2 2 2 2 2 3 3 1 2 2
##
## Within cluster sum of squares by cluster:
## [1] 26382784712 28184319111 25765310355
##  (between_SS / total_SS =  49.0 %)
##
## Available components:
```

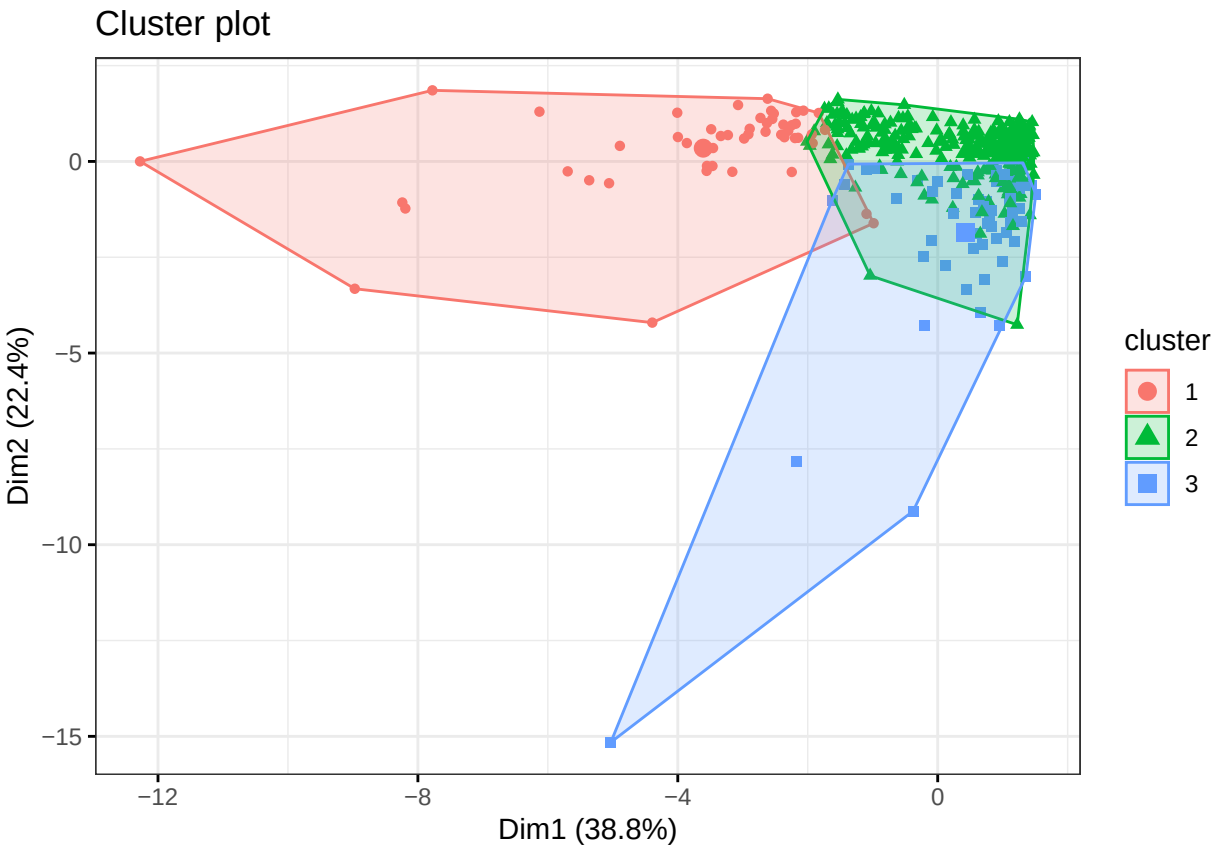
```
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
# plot clusters
plot(customers, col = km$cluster)

# plot centers
points(km$centers)
```



I plotted the clusters to get a better visual understanding of the clusters themselves.

```
fviz_cluster(km, data = customers,
             geom = "point",
             ellipse.type = "convex",
             ggtheme = theme_bw()
             )
```


Cluster plot
Dim2 (22.4%)
0
−5
−10
−15
−12
−8
−4
0
Dim1 (38.8%)
cluster
1
2
3