High value customers identification for an E-Commerce company

Andrei Enescu

11/2/2020

DESCRIPTION

Background of Problem Statement:

A UK-based online retail store has captured the sales data for different products for the period of one year (Nov 2016 to Dec 2017). The organization sells gifts primarily on the online platform. The customers who make a purchase consume directly for themselves. There are small businesses that buy in bulk and sell to other customers through the retail outlet channel.

Project Objective:

Find significant customers for the business who make high purchases of their favourite products. The organization wants to roll out a loyalty program to the high-value customers after identification of segments. Use the clustering methodology to segment customers into groups:

Domain: E-commerce

Dataset Description:

This is a transnational dataset that contains all the transactions occurring between Nov-2016 to Dec-2017 for a UK-based online retail store.

Attribute Description
InvoiceNo Invoice number (A 6-digit integral number uniquely assigned to each transaction)
StockCode Product (item) code
Description Product (item) name
Quantity The quantities of each product (item) per transaction
InvoiceDate The day when each transaction was generated
UnitPrice Unit price (Product price per unit)
CustomerID Customer number (Unique ID assigned to each customer)
Country Country name (The name of the country where each customer resides)

Analysis tasks to be performed:

1. Use the clustering methodology to segment customers into groups: Use the following clustering algorithms:

1.1 K means
1.2 Hierarchical

2. Identify the right number of customer segments.

3. Provide the number of customers who are highly valued.

4. Identify the clustering algorithm that gives maximum accuracy and explains robust clusters.

5. If the number of observations is loaded in one of the clusters, break down that cluster further using the clustering algorithm. [hint: Here loaded means if any cluster has more number of data points as compared to other clusters then split that clusters by increasing the number of clusters and observe, compare the results with previous results.]

These are the libraries I used:

```
library(rio)
library(DataExplorer)
library(ggplot2)
library(factoextra)
library(NbClust)
```

Importing the data using RIO package

```
EcommDF <- import("Ecommerce.csv")
```

Checking the data structure

```
head(EcommDF)
```

```
##   InvoiceNo StockCode                         Description Quantity InvoiceDate
## 1    536365    85123A  WHITE HANGING HEART T-LIGHT HOLDER        6   29-Nov-16
## 2    536365     71053                 WHITE METAL LANTERN        6   29-Nov-16
## 3    536365    84406B      CREAM CUPID HEARTS COAT HANGER        8   29-Nov-16
## 4    536365    84029G KNITTED UNION FLAG HOT WATER BOTTLE        6   29-Nov-16
## 5    536365    84029E      RED WOOLLY HOTTIE WHITE HEART.        6   29-Nov-16
## 6    536365     22752        SET 7 BABUSHKA NESTING BOXES        2   29-Nov-16
##   UnitPrice CustomerID        Country V9
## 1      2.55      17850 United Kingdom NA
## 2      3.39      17850 United Kingdom NA
## 3      2.75      17850 United Kingdom NA
## 4      3.39      17850 United Kingdom NA
## 5      3.39      17850 United Kingdom NA
## 6      7.65      17850 United Kingdom NA
```

```
summary(EcommDF)
```

```
##   InvoiceNo          StockCode         Description           Quantity
##  Length:541909      Length:541909      Length:541909      Min.   :-80995.00
```

```
##  Class :character   Class :character   Class :character   1st Qu.:     1.00
##  Mode  :character   Mode  :character   Mode  :character   Median :     3.00
##                                                          Mean   :     9.55
##                                                          3rd Qu.:    10.00
##                                                          Max.   : 80995.00
##
##  InvoiceDate          UnitPrice           CustomerID       Country
##  Length:541909      Min.   :-11062.06   Min.   :12346    Length:541909
##  Class :character   1st Qu.:     1.25   1st Qu.:13953    Class :character
##  Mode  :character   Median :     2.08   Median :15152    Mode  :character
##                     Mean   :     4.61   Mean   :15288
##                     3rd Qu.:     4.13   3rd Qu.:16791
##                     Max.   : 38970.00   Max.   :18287
##                                         NA's   :135080
##     V9
##  Mode:logical
##  NA's:541909
##
##
##
##
##
```

```
str(EcommDF)
```

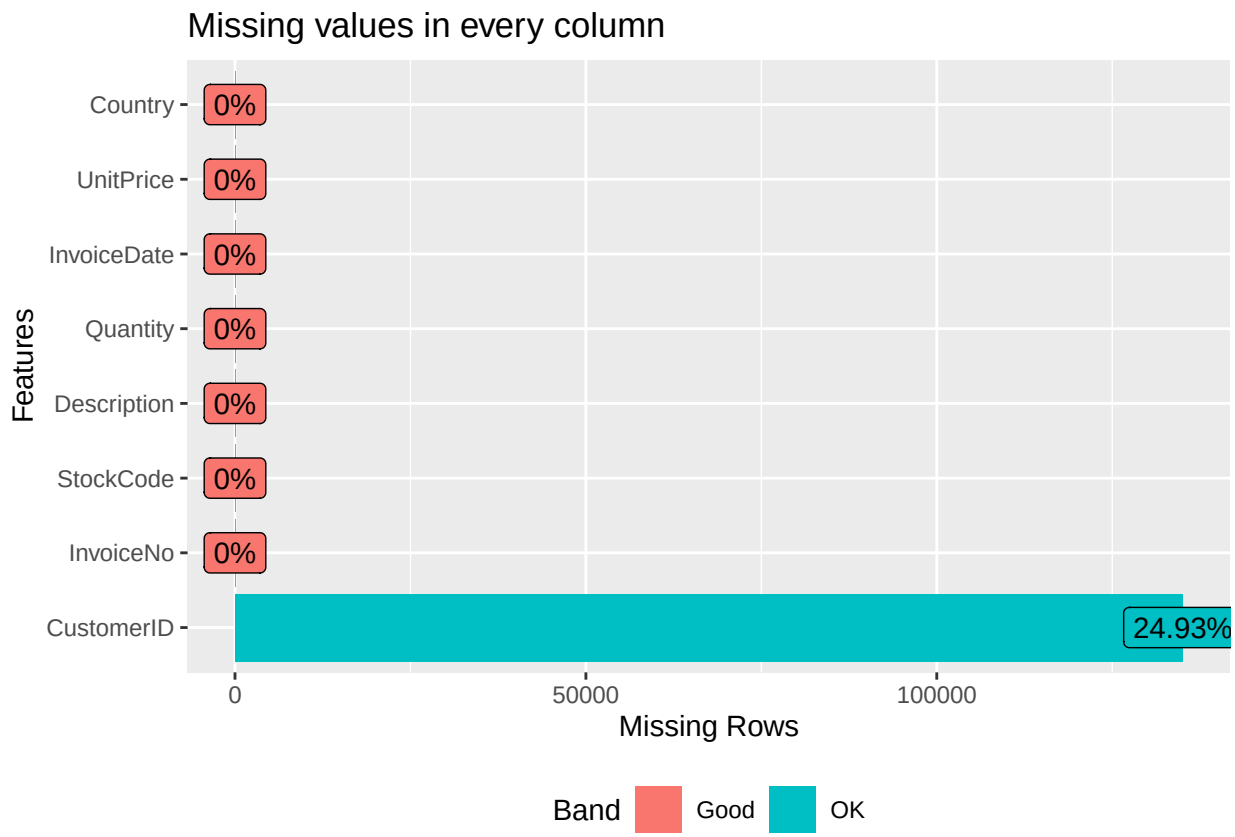
```
## 'data.frame':    541909 obs. of  9 variables:
##  $ InvoiceNo  : chr  "536365" "536365" "536365" "536365" ...
##  $ StockCode  : chr  "85123A" "71053" "84406B" "84029G" ...
##  $ Description: chr  "WHITE HANGING HEART T-LIGHT HOLDER" "WHITE METAL LANTERN" "CREAM CUPID HEARTS C
##  $ Quantity   : int  6 6 8 6 6 2 6 6 6 32 ...
##  $ InvoiceDate: chr  "29-Nov-16" "29-Nov-16" "29-Nov-16" "29-Nov-16" ...
##  $ UnitPrice  : num  2.55 3.39 2.75 3.39 3.39 7.65 4.25 1.85 1.85 1.69 ...
##  $ CustomerID : int  17850 17850 17850 17850 17850 17850 17850 17850 17850 13047 ...
##  $ Country    : chr  "United Kingdom" "United Kingdom" "United Kingdom" "United Kingdom" ...
##  $ V9         : logi  NA NA NA NA NA NA ...
```

Removing empty column

```
EcommDF <- EcommDF[, 1:8]
```

Checking and removing NA values

```
options(scipen = 999)
plot_missing(EcommDF, title = "Missing values in every column")
```




```
EcommDF <- na.omit(EcommDF)
```

Leaving only unique customers

```
EcommDF <- EcommDF[unique(EcommDF$CustomerID), ]
```

Remove Quantity with negative values

```
EcommDF <- EcommDF[EcommDF$Quantity >= 0,]
```

Add a "Total Spent" column

```
EcommDF$TotalSpent <- EcommDF$Quantity * EcommDF$UnitPrice
```

Variables should be numeric

```
EcommDF_Num <- EcommDF[, -c(3, 5, 8)]
str(EcommDF_Num)
```

```
## 'data.frame':    4244 obs. of  6 variables:
##  $ InvoiceNo : chr  "538533" "537885" "537845" "538051" ...
##  $ StockCode : chr  "22067" "20675" "21755" "22945" ...
##  $ Quantity  : int  2 8 3 12 3 1 2 36 2 1 ...
##  $ UnitPrice : num  1.65 1.25 5.95 0.85 0.85 1.65 2.95 1.25 1.25 4.95 ...
##  $ CustomerID: int  14796 12942 17690 18041 17827 12748 14702 13209 12748 16907 ...
##  $ TotalSpent: num  3.3 10 17.85 10.2 2.55 ...
```

```
EcommDF_Num$InvoiceNo <- as.numeric(EcommDF_Num$InvoiceNo)
EcommDF_Num$StockCode <- as.numeric(EcommDF_Num$StockCode)
```

```
## Warning: NAs introduced by coercion
```

```
EcommDF_Num$Quantity <- as.numeric(EcommDF_Num$Quantity)
EcommDF_Num$CustomerID <- as.numeric(EcommDF_Num$CustomerID)

summary(EcommDF_Num)
```

```
##    InvoiceNo        StockCode        Quantity         UnitPrice
##  Min.   :537822   Min.   :10002   Min.   :   1.00   Min.   :  0.100
##  1st Qu.:538056   1st Qu.:21755   1st Qu.:   2.00   1st Qu.:  1.250
##  Median :538205   Median :22423   Median :   4.00   Median :  2.100
##  Mean   :538228   Mean   :28375   Mean   :  11.09   Mean   :  3.066
##  3rd Qu.:538418   3rd Qu.:22793   3rd Qu.:  12.00   3rd Qu.:  3.750
##  Max.   :538634   Max.   :90204   Max.   :1728.00   Max.   :175.000
##                   NA's   :409
##    CustomerID      TotalSpent
##  Min.   :12429   Min.   :   0.21
##  1st Qu.:14256   1st Qu.:   4.20
##  Median :15547   Median :  10.08
##  Mean   :15527   Mean   :  20.75
##  3rd Qu.:17126   3rd Qu.:  17.85
##  Max.   :18225   Max.   :3794.40
##
```

Outlier removal

```
Quantity_LT <- mean(EcommDF_Num$Quantity) - 2 * sd(EcommDF_Num$Quantity)
Quantity_UT <- mean(EcommDF_Num$Quantity) + 2 * sd(EcommDF_Num$Quantity)

UnitPrice_LT <- mean(EcommDF_Num$UnitPrice) - 2 * sd(EcommDF_Num$UnitPrice)
UnitPrice_UT <- mean(EcommDF_Num$UnitPrice) + 2 * sd(EcommDF_Num$UnitPrice)
```

Threshold

```
EcommDF_Num$Quantity <-
  ifelse(EcommDF_Num$Quantity > Quantity_UT, Quantity_UT, EcommDF_Num$Quantity)
EcommDF_Num$Quantity <-
  ifelse(EcommDF_Num$Quantity < Quantity_LT, Quantity_LT, EcommDF_Num$Quantity)

EcommDF_Num$UnitPrice <-
  ifelse(EcommDF_Num$UnitPrice > UnitPrice_UT, UnitPrice_UT, EcommDF_Num$UnitPrice)
EcommDF_Num$UnitPrice <-
  ifelse(EcommDF_Num$UnitPrice < UnitPrice_LT, UnitPrice_LT, EcommDF_Num$UnitPrice)
```

Remove scaling effect from data

```
EcommDF_Num <- scale(EcommDF_Num)
summary(EcommDF_Num)
```

```
##    InvoiceNo          StockCode          Quantity         UnitPrice
##  Min.   :-1.73565   Min.   :-1.0106   Min.   :-0.5350   Min.   :-1.1097
##  1st Qu.:-0.73609   1st Qu.:-0.3642   1st Qu.:-0.4708   1st Qu.:-0.6549
##  Median :-0.09962   Median :-0.3274   Median :-0.3423   Median :-0.3188
##  Mean   : 0.00000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.81023   3rd Qu.:-0.3071   3rd Qu.: 0.1715   3rd Qu.: 0.3338
##  Max.   : 1.73291   Max.   : 3.4010   Max.   : 6.2985   Max.   : 3.7156
##                     NA's   :409
##    CustomerID         TotalSpent
##  Min.   :-1.79579   Min.   :-0.23833
##  1st Qu.:-0.73659   1st Qu.:-0.19203
##  Median : 0.01187   Median :-0.12381
##  Mean   : 0.00000   Mean   : 0.00000
##  3rd Qu.: 0.92729   3rd Qu.:-0.03365
##  Max.   : 1.56444   Max.   :43.78485
##
```

Removing NA values

```
EcommDF_Num <- na.omit(EcommDF_Num)
```

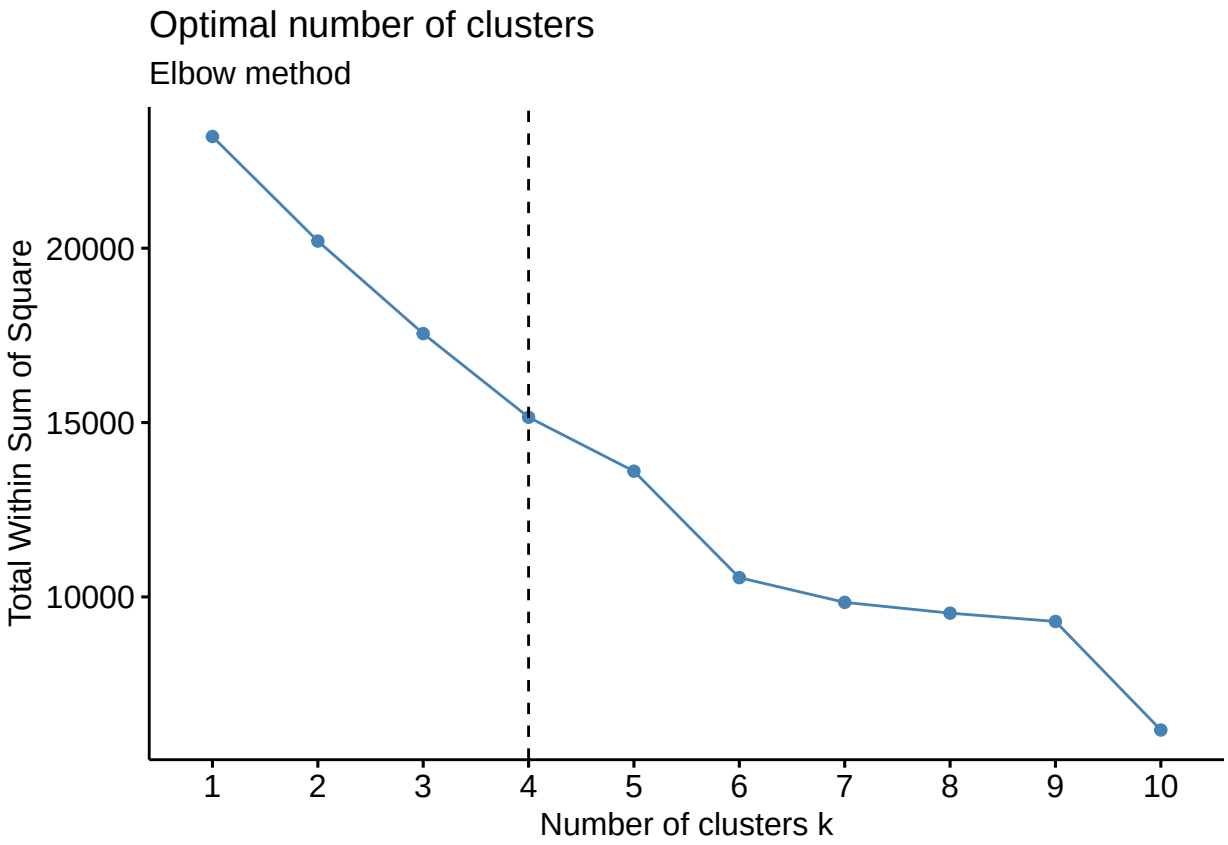
1. Use the clustering methodology to segment customers into groups:

Use the following clustering algorithms:

1.1 K means

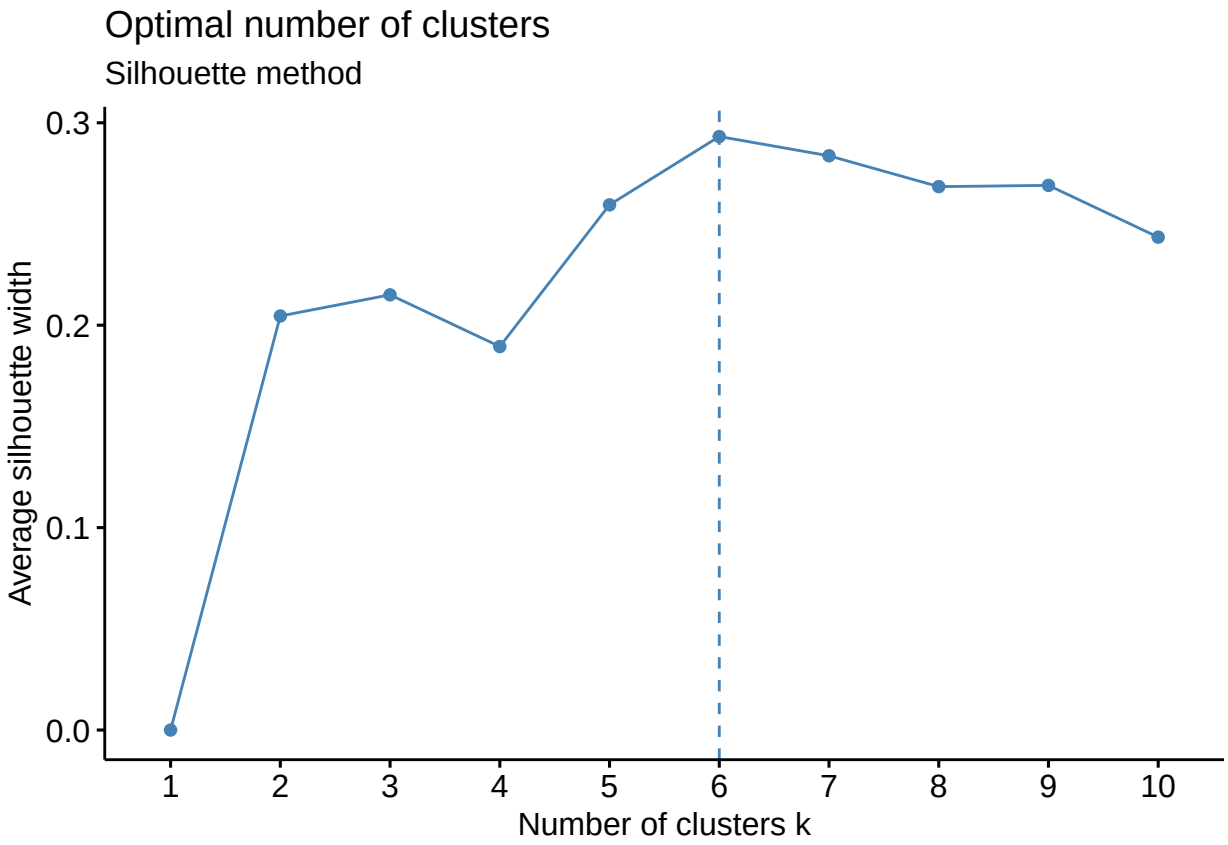
Elbow method

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



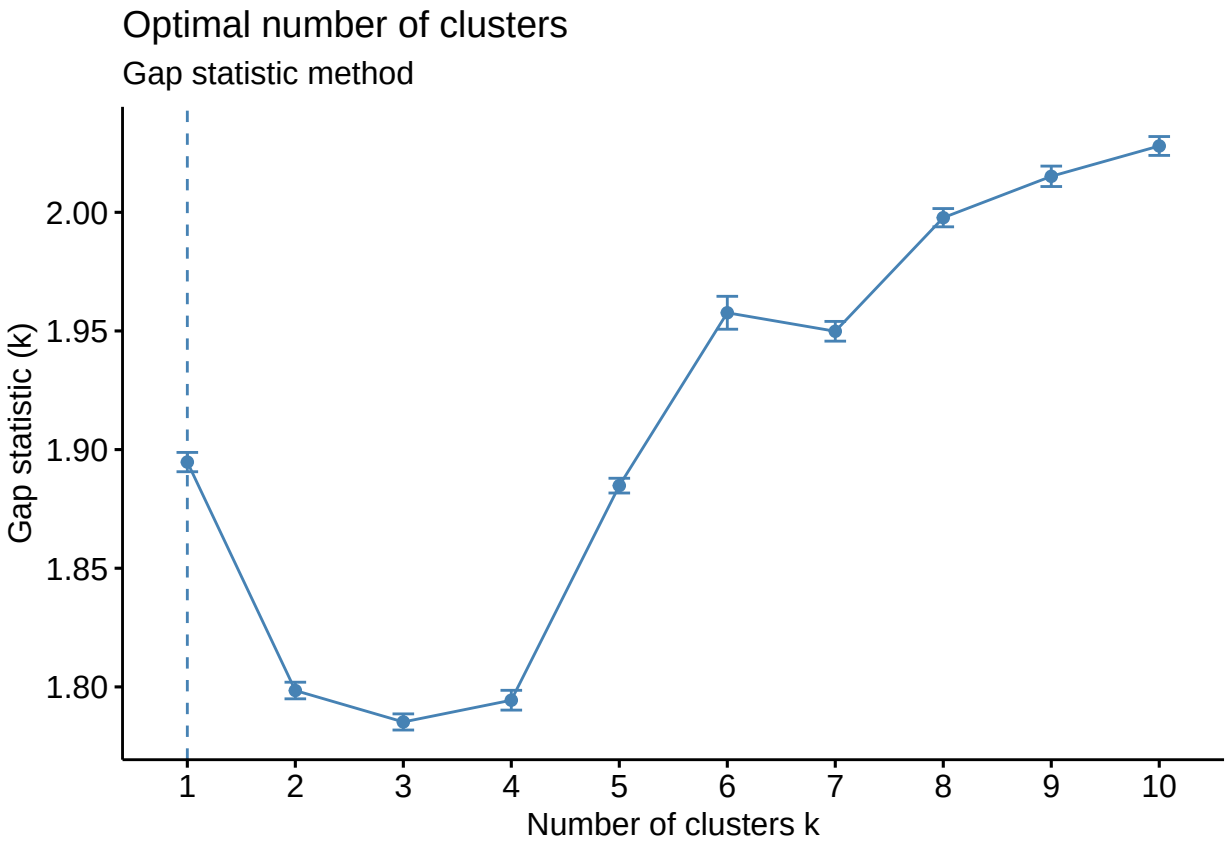
Silhouette method

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



Gap statistic

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

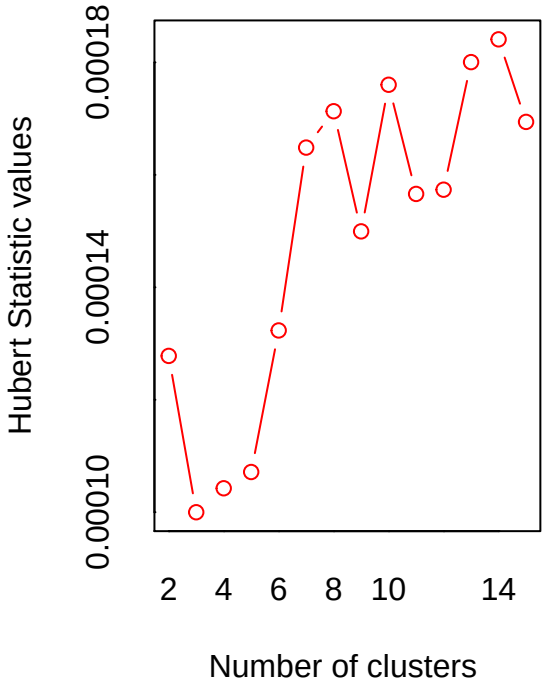


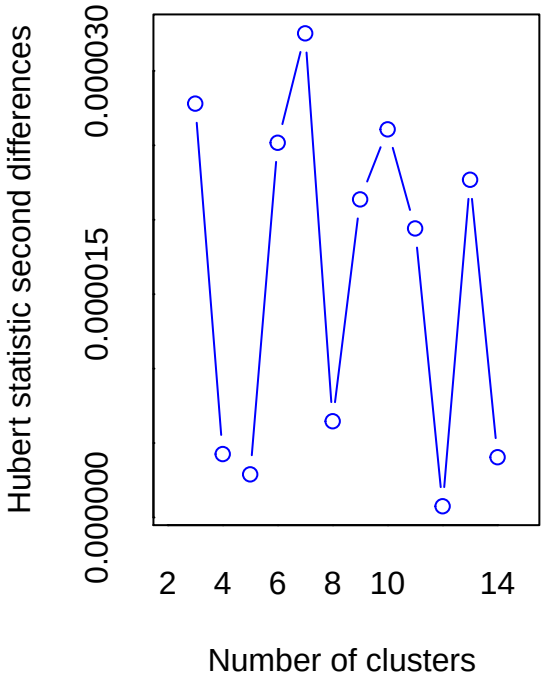
Comments:

We can observe that these three methods do not necessarily lead to the same result. All 3 approaches suggest a different number of clusters.

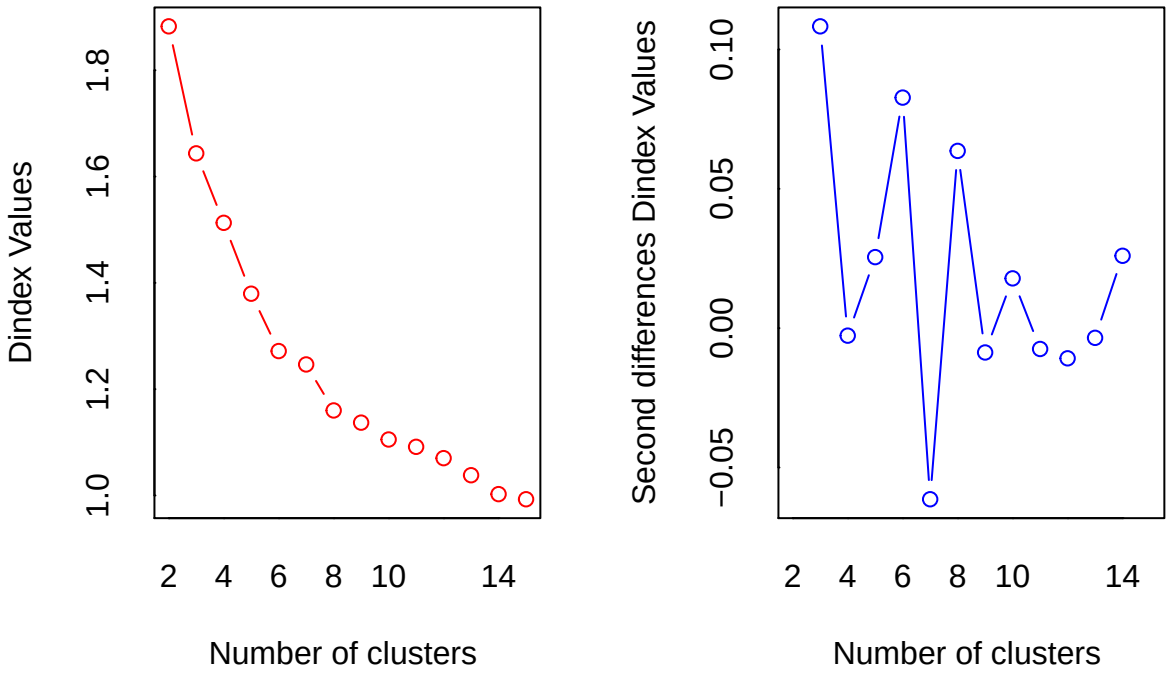
In this case we can use a 4th alternative - *NbClust* function, which provides 30 indices for choosing the best number of clusters.

```
NC <- NbClust(EcommDF_Num, distance = "euclidean",
        min.nc = 2, max.nc = 15, method = "kmeans")
```






```
## *** : 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:
## * 7 proposed 2 as the best number of clusters
## * 5 proposed 3 as the best number of clusters
## * 2 proposed 5 as the best number of clusters
## * 3 proposed 7 as the best number of clusters
## * 4 proposed 10 as the best number of clusters
## * 1 proposed 11 as the best number of clusters
## * 1 proposed 14 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  2
##
##
## *******************************************************************
```

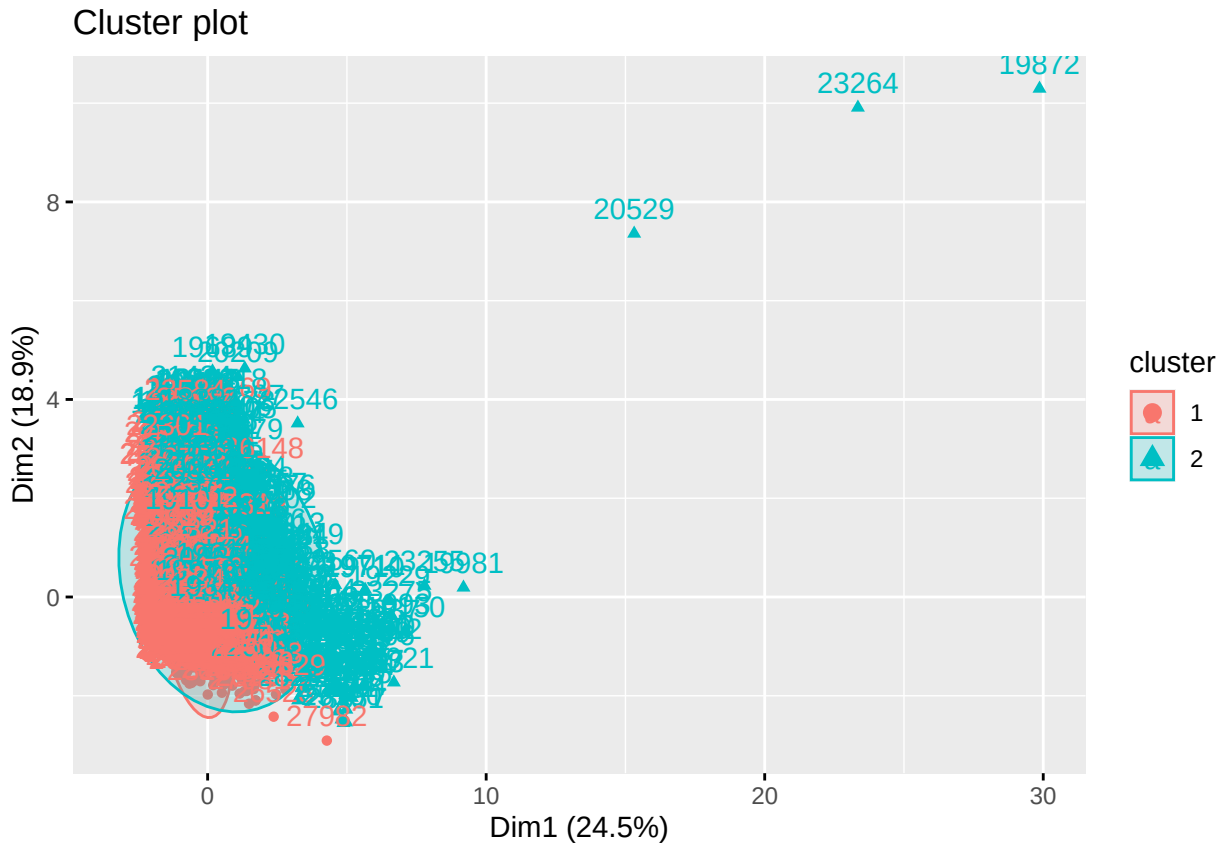
```
barplot(table(NC$Best.n[1,]),
        xlab="Number of Clusters", ylab="Number of Criteria",
```

```
        main="Number of Clusters Chosen by Criteria")

KM_res <- kmeans(EcommDF_Num, centers = 2)
fviz_cluster(KM_res, EcommDF_Num, ellipse.type = "norm")

KM_res$centers
```

```
##    InvoiceNo   StockCode    Quantity   UnitPrice CustomerID TotalSpent
## 1  0.6652679 -0.03921182 -0.2102278 -0.07788155  0.4480131 -0.1033226
## 2 -0.7881534  0.04671809  0.2541170  0.06941187 -0.5175253  0.1177332
```



Comments:

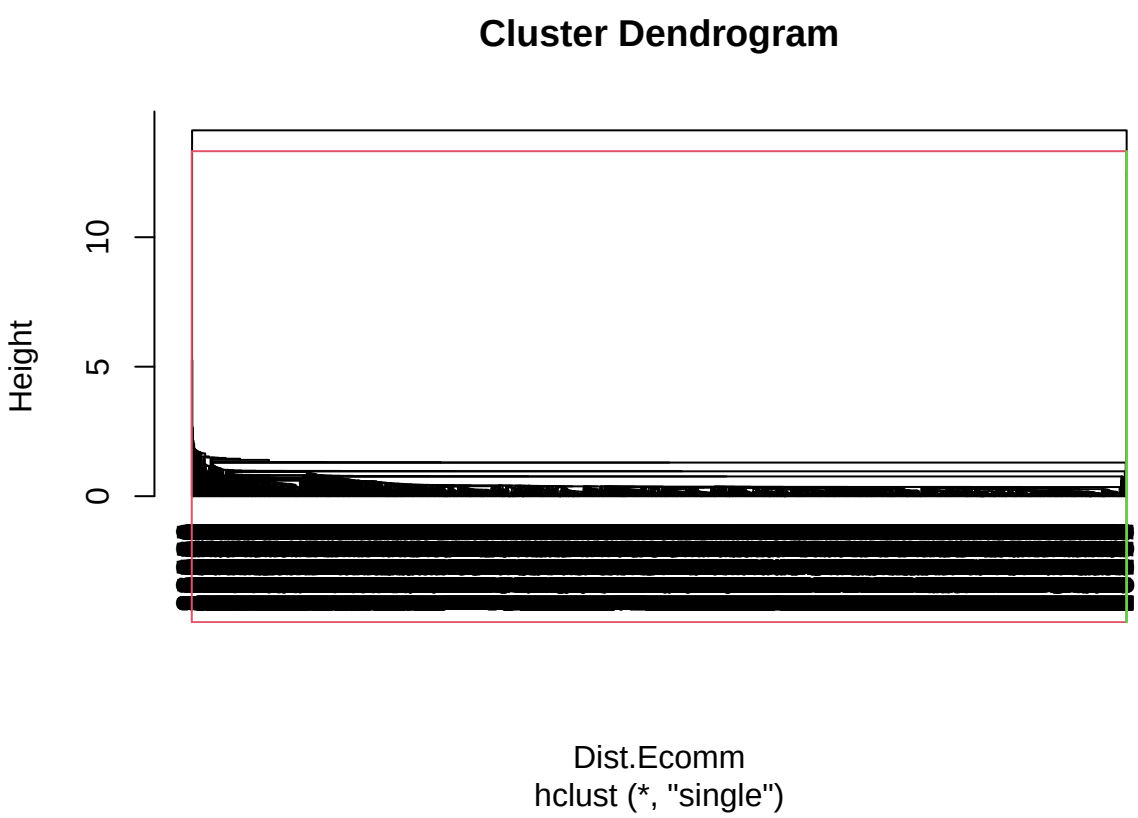
Based on all 30 indices, the best number of clusters is 2 clusters.
By looking at the centers of the two clusters we can observe that there is no overlapping.

1.2 Hierarchical

Calculate distances between observations

```
Dist.Ecomm <- dist(EcommDF_Num, method = 'euclidean')

HC <- hclust(Dist.Ecomm, method = "single")
plot(HC, hang = -1)
rect.hclust(HC, k = 2, border = 2:5)
```



2. Identify the right number of customer segments.

According to the majority rule, the best number of clusters is 2

3. Provide the number of customers who are highly valued.

```
sum(EcommDF$TotalSpent > mean(EcommDF$TotalSpent))
```

```
## [1] 868
```

The most valuable customers buy more or higher-value products than the average customer. So we can conclude that there are *868* highly valued clients

4. Identify the clustering algorithm that gives maximum accuracy and explains robust clusters.

For the given dataset the maximum accuracy is obtained by using partitioning method clustering, more exactly the K-means clustering