Customer Segmentation. Application of Unsupervised Learning Methods for Trend Exploration

by Ketao Li, Kush Halani, Josue Romain, Juan Peña

Abstract Customer segmentation is the process of dividing customers into groups based on common characteristics so companies can market to each group effectively and appropriately.

Background

Without a deep understanding of how a company's best current customers are segmented, a business often lacks the market focus needed to allocate and spend its precious human and capital resources efficiently. Furthermore, a lack of best current customer segment focus can cause diffused go-to-market and product development strategies that hamper a company's ability to fully engage with its target segments. Together, all of those factors can ultimately impede a company's growth.
RFM (recency, frequency, monetary) analysis is a marketing technique used to determine quantitatively which customers are the best ones by examining how recently a customer has purchased (recency), how often they purchase (frequency), and how much the customer spends (monetary).

Objective

The objective of customers segment according to their purchase history, is to turn them into loyal customers by recommending products of their choice.

Data Analysis

Typically e-commerce datasets are proprietary and consequently hard to find among publicly available data. However, The UCI Machine Learning Repository has made this dataset containing actual transactions from 2010 and 2011.The data set used for this research contains 540k of transaction from UK retailer. The data has been sourced from Kaggle.

Data Dictionary

Column Name	Column Description
InvoiceNo	Invoice No
StockCode	Stock Code
Description	Description for the stock
Quantity	Quantity of products sold
InvoiceDate	Invoice Date
UnitPrice	Unit Price
CustomerID	Customer ID
Country	Country where the products are sold

Data Exploration

Firstly we are going to load and examine content and statistics of the data set

```
data = read.csv("../data/data.csv", header = T,
               na.strings = c("NA","","#NA"),sep=",")
```

Table 2: Online Retail Dataset Summary

No	Variable	Stats / Values	Freqs (% of Valid)	Missing
1	InvoiceNo [factor]	1. 536365 2. 536366 3. 536367 [25897 others]	7 (0.0%) 2 (0.0%) 12 (0.0%) 541888 (100.0%)	0 (0%)
2	StockCode [factor]	1. 10002 2. 10080 3. 10120 [4067 others]	73 (0.0%) 24 (0.0%) 30 (0.0%) 541782 (100.0%)	0 (0%)
3	Description [factor]	1. ·4 PURPLE FLOCK DINNER CA 2. ·50'S CHRISTMAS GIFT BAG 3. ·DOLLY GIRL BEAKER [4220 others]	41 (0.0%) 130 (0.0%) 181 (0.0%) 540103 (99.9%)	1454 (0.27%)
4	Quantity [integer]	Mean (sd) : 9.6 (218.1) min < med < max: -80995 < 3 < 80995 IQR (CV) : 9 (22.8)	722 distinct values	0 (0%)
5	InvoiceDate [factor]	1. 1/10/2011 10:04 2. 1/10/2011 10:07 3. 1/10/2011 10:08 [23257 others]	1 (0.0%) 1 (0.0%) 1 (0.0%) 541906 (100.0%)	0 (0%)
6	UnitPrice [numeric]	Mean (sd) : 4.6 (96.8) min < med < max: -11062.1 < 2.1 < 38970 IQR (CV) : 2.9 (21)	1630 distinct values	0 (0%)
7	CustomerID [integer]	Mean (sd) : 15287.7 (1713.6) min < med < max: 12346 < 15152 < 18287 IQR (CV) : 2838 (0.1)	4372 distinct values	135080 (24.93%)
8	Country [factor]	1. Australia 2. Austria 3. Bahrain [35 others]	1259 (0.2%) 401 (0.1%) 19 (0.0%) 540230 (99.7%)	0 (0%)

From the above summary, we can find that there are some negative values for Quantity and UnitPrice.These values don't make sense, so we'll delete them directly. There are some missing data for CustomerID, we just remove them directly considering we have enough data.

```
customerData <- data %>%
  mutate(Quantity = replace(Quantity, Quantity<=0, NA),
         UnitPrice = replace(UnitPrice, UnitPrice<=0, NA))


customerData = customerData %>%filter(complete.cases(.))
```

Table 3: Online Retail Dataset Summary

No	Variable	Stats / Values	Freqs (% of Valid)	Missing
1	InvoiceNo [factor]	1. 536365 2. 536366 3. 536367 [25897 others]	7 (0.0%) 2 (0.0%) 12 (0.0%) 397863 (100.0%)	0 (0%)
2	StockCode [factor]	1. 10002 2. 10080 3. 10120 [4067 others]	49 (0.0%) 21 (0.0%) 30 (0.0%) 397784 (100.0%)	0 (0%)

No	Variable	Stats / Values	Freqs (% of Valid)	Missing
3	Description [factor]	1. ·4 PURPLE FLOCK DINNER CA 2. ·50'S CHRISTMAS GIFT BAG 3. ·DOLLY GIRL BEAKER [4220 others]	39 (0.0%) 109 (0.0%) 138 (0.0%) 397598 (99.9%)	0 (0%)
4	Quantity [integer]	Mean (sd) : 13 (179.3) min < med < max: 1 < 6 < 80995 IQR (CV) : 10 (13.8)	301 distinct values	0 (0%)
5	InvoiceDate [factor]	1. 1/10/2011 10:04 2. 1/10/2011 10:07 3. 1/10/2011 10:08 [23257 others]	0 (0.0%) 0 (0.0%) 0 (0.0%) 397884 (100.0%)	0 (0%)
6	UnitPrice [numeric]	Mean (sd) : 3.1 (22.1) min < med < max: 0 < 2 < 8142.8 IQR (CV) : 2.5 (7.1)	440 distinct values	0 (0%)
7	CustomerID [integer]	Mean (sd) : 15294.4 (1713.1) min < med < max: 12346 < 15159 < 18287 IQR (CV) : 2826 (0.1)	4338 distinct values	0 (0%)
8	Country [factor]	1. Australia 2. Austria 3. Bahrain [35 others]	1182 (0.3%) 398 (0.1%) 17 (0.0%) 396287 (99.6%)	0 (0%)

Data Preparation

We need do some some data transformation and add one new variant total.

```
customerData <- customerData %>%
  mutate( InvoiceDate=as.Date(InvoiceDate, '%m/%d/%Y %H:%M'),
          CustomerID=as.factor(CustomerID))

customerData <- customerData %>%
  mutate(total = Quantity*UnitPrice)

glimpse(customerData)
```

```
Observations: 397,884
Variables: 9
$ InvoiceNo   <fct> 536365, 536365, 536365, 536365, 536365, 536365, 536365,...
$ StockCode   <fct> 85123A, 71053, 84406B, 84029G, 84029E, 22752, 21730, 22...
$ Description <fct> WHITE HANGING HEART T-LIGHT HOLDER, WHITE METAL LANTERN...
$ Quantity    <int> 6, 6, 8, 6, 6, 2, 6, 6, 6, 32, 6, 6, 8, 6, 6, 3, 2, 3, ...
$ InvoiceDate <date> 2010-12-01, 2010-12-01, 2010-12-01, 2010-12-01, 2010-1...
$ UnitPrice   <dbl> 2.55, 3.39, 2.75, 3.39, 3.39, 7.65, 4.25, 1.85, 1.85, 1...
$ CustomerID  <fct> 17850, 17850, 17850, 17850, 17850, 17850, 17850, 17850,...
$ Country     <fct> United Kingdom, United Kingdom, United Kingdom, United ...
$ total       <dbl> 15.30, 20.34, 22.00, 20.34, 20.34, 15.30, 25.50, 11.10,...
```

Calculate RFM

To implement the RFM analysis, we need to take steps to get the rfm values:

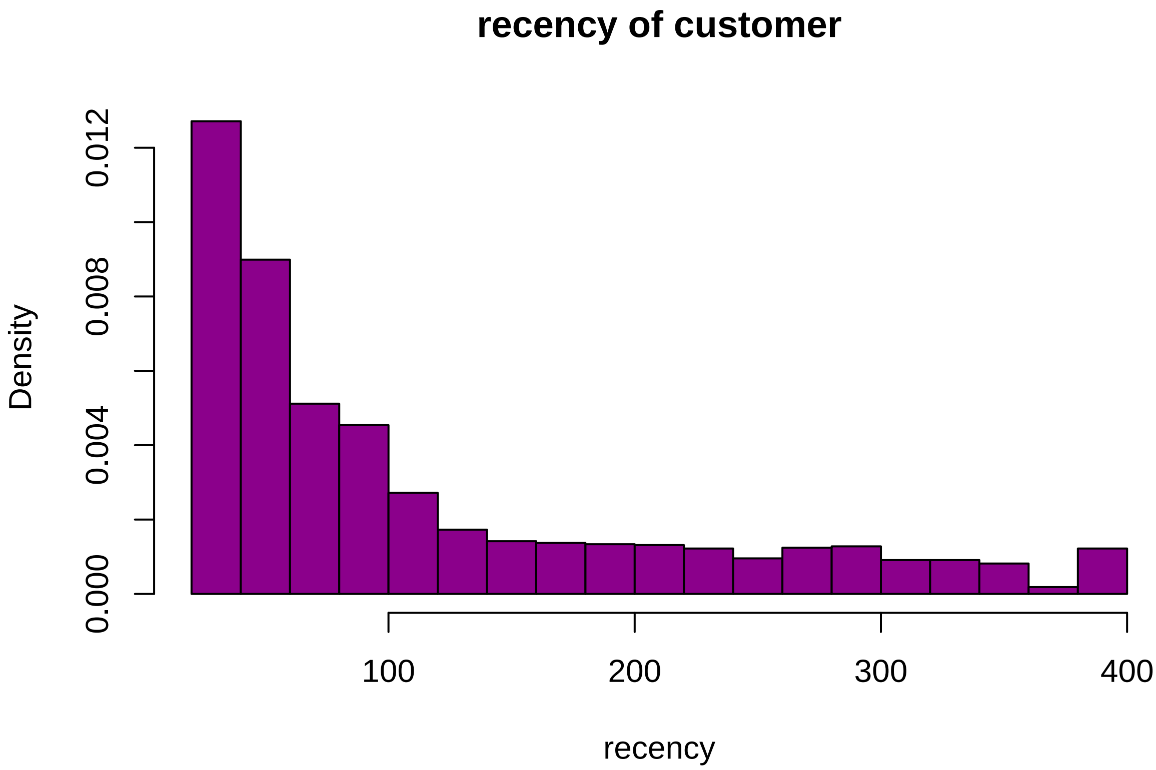
1. Find the most recent date for each customer ID and calculate the days to the 2012-01-01, to get the recency data.
2. Calculate the quantity of transactions of a customer, to get the frequency data
3. Sum the amount of money a customer spent and divide it by frequency, to get the amount per transaction on average, that is the monetary data.

```
cd_RFM <- customerData %>%
  group_by(CustomerID) %>%
  summarise(recency=as.numeric(as.Date("2012-01-01")-max(InvoiceDate)),
            frequenci=n_distinct(InvoiceNo), monitery= sum(total)/n_distinct(InvoiceNo))

summary(cd_RFM)

head(cd_RFM)

   CustomerID      recency        frequenci          monitery
 12346  :   1   Min.   : 23.0   Min.   :  1.000   Min.   :    3.45
 12347  :   1   1st Qu.: 40.0   1st Qu.:  1.000   1st Qu.:  178.62
 12348  :   1   Median : 73.0   Median :  2.000   Median :  293.90
 12349  :   1   Mean   :115.1   Mean   :  4.272   Mean   :  419.17
 12350  :   1   3rd Qu.:164.8   3rd Qu.:  5.000   3rd Qu.:  430.11
 12352  :   1   Max.   :396.0   Max.   :209.000   Max.   :84236.25
 (Other):4332
# A tibble: 6 x 4
  CustomerID recency frequenci monitery
  <fct>        <dbl>     <int>    <dbl>
1 12346          348         1   77184.
2 12347           25         7     616.
3 12348           98         4     449.
4 12349           41         1    1758.
5 12350          333         1     334.
6 12352           59         8     313.
```

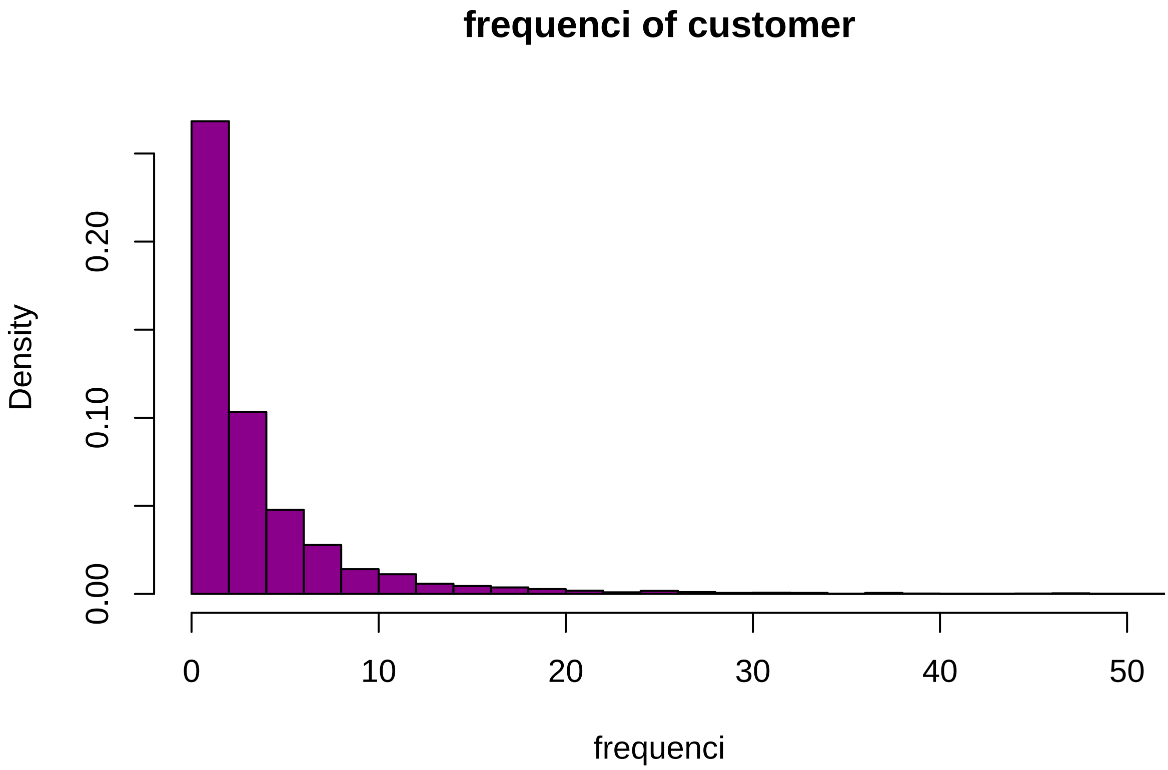
Table 4: Online Retail Dataset Summary

No	Variable	Stats / Values	Freqs (% of Valid)	Missing
1	CustomerID [factor]	1. 12346 2. 12347 3. 12348 [4335 others]	1 (0.0%) 1 (0.0%) 1 (0.0%) 4335 (99.9%)	0 (0%)
2	recency [numeric]	Mean (sd) : 115.1 (100) min < med < max: 23 < 73 < 396 IQR (CV) : 124.8 (0.9)	304 distinct values	0 (0%)
3	frequenci [integer]	Mean (sd) : 4.3 (7.7) min < med < max: 1 < 2 < 209 IQR (CV) : 4 (1.8)	59 distinct values	0 (0%)
4	monitery [numeric]	Mean (sd) : 419.2 (1796.5) min < med < max: 3.5 < 293.9 < 84236.2 IQR (CV) : 251.5 (4.3)	4249 distinct values	0 (0%)

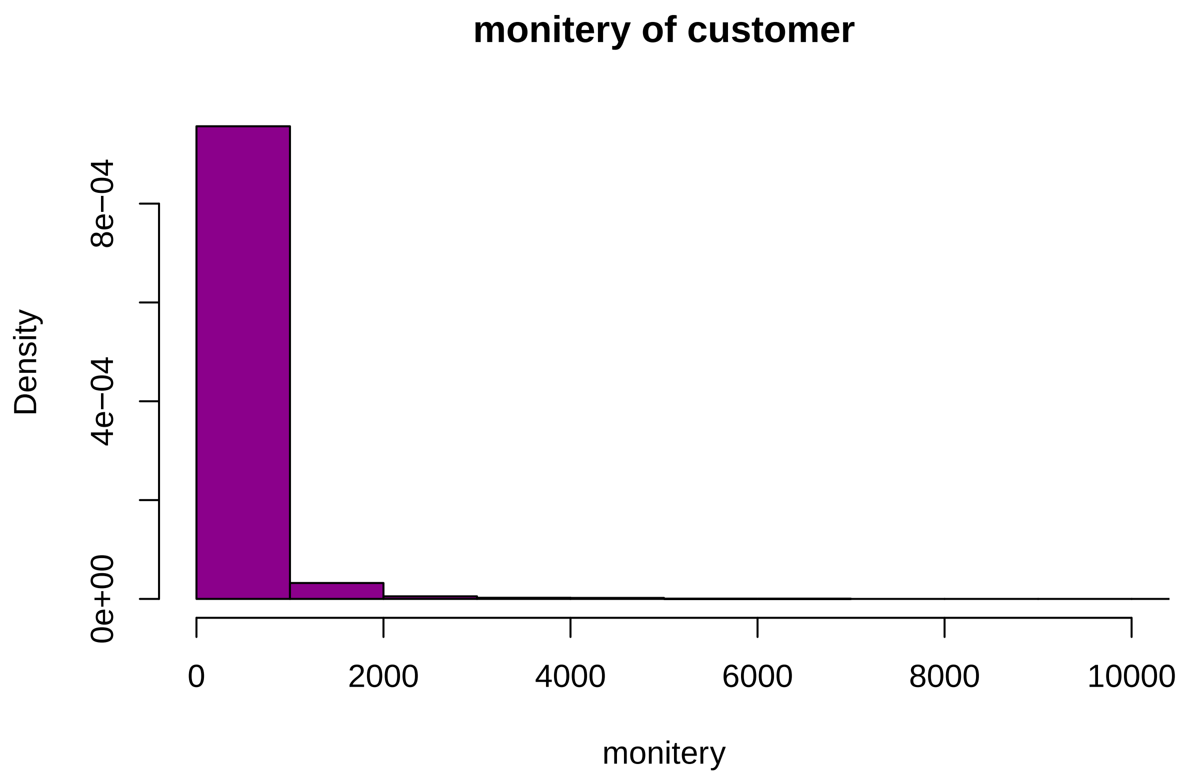
```
# histogram with added parameters
hist(cd_RFM$recency,
main="recency of customer",
xlab="recency",
xlim=c(20,400),
col="darkmagenta",
freq=FALSE
)
```




```
# histogram with added parameters
hist(cd_RFM$frequenci,
main="frequenci of customer",
xlab="frequenci",
breaks=100,
xlim=c(0,50),
col="darkmagenta",
freq=FALSE
)
```

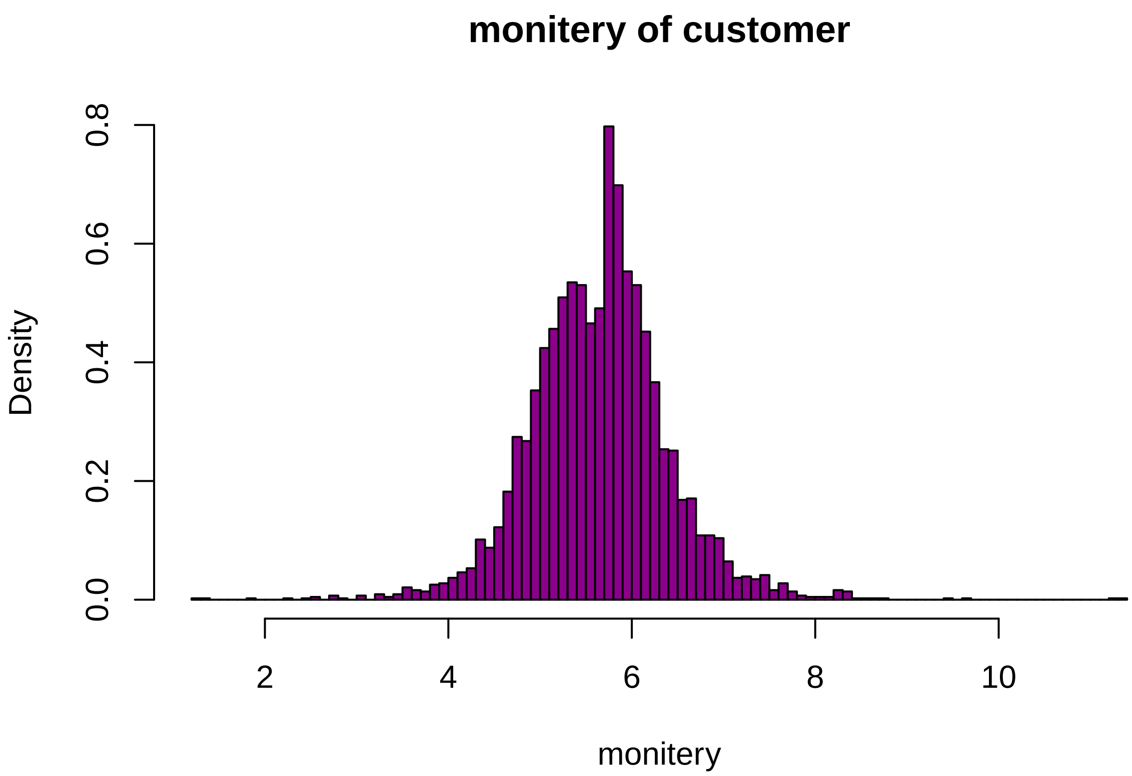



```
# histogram with added parameters
hist(cd_RFM$monitery,
main="monitery of customer",
xlab="monitery",
breaks=100,
xlim=c(0,10000),
col="darkmagenta",
freq=FALSE
)
```



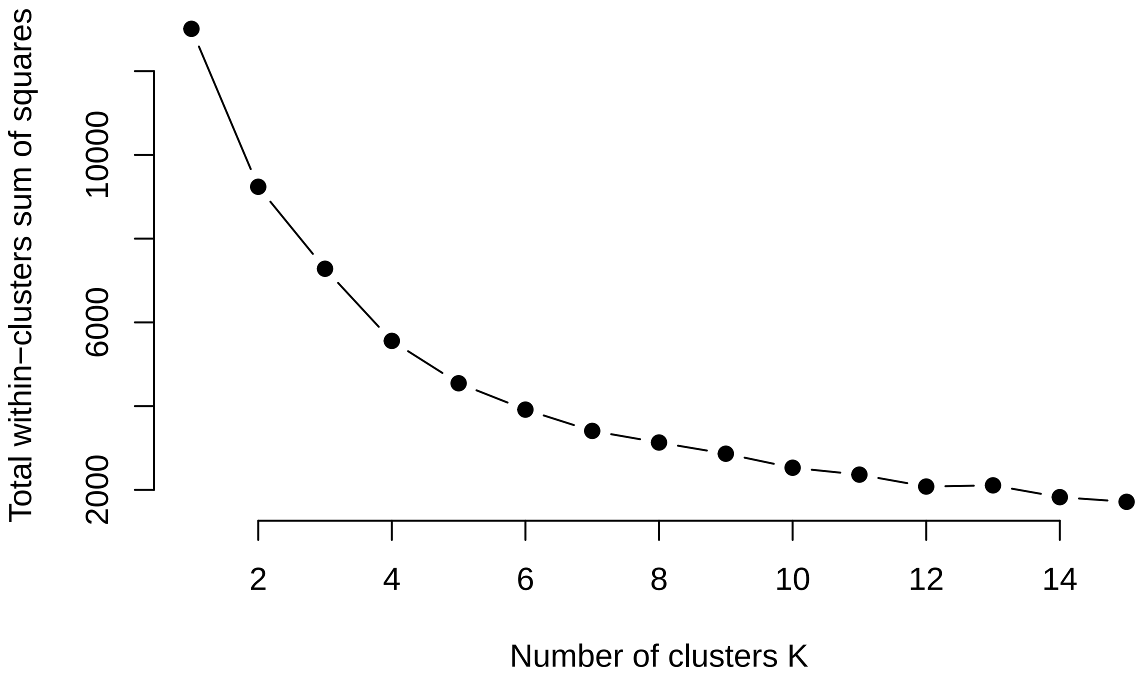
Becouse the data is realy skewed, we use log scale to normalize

```
cd_RFM$monitery <- log(cd_RFM$monitery)
hist(cd_RFM$monitery,
main="monitery of customer",
xlab="monitery",
breaks=100,
col="darkmagenta",
freq=FALSE
)
```

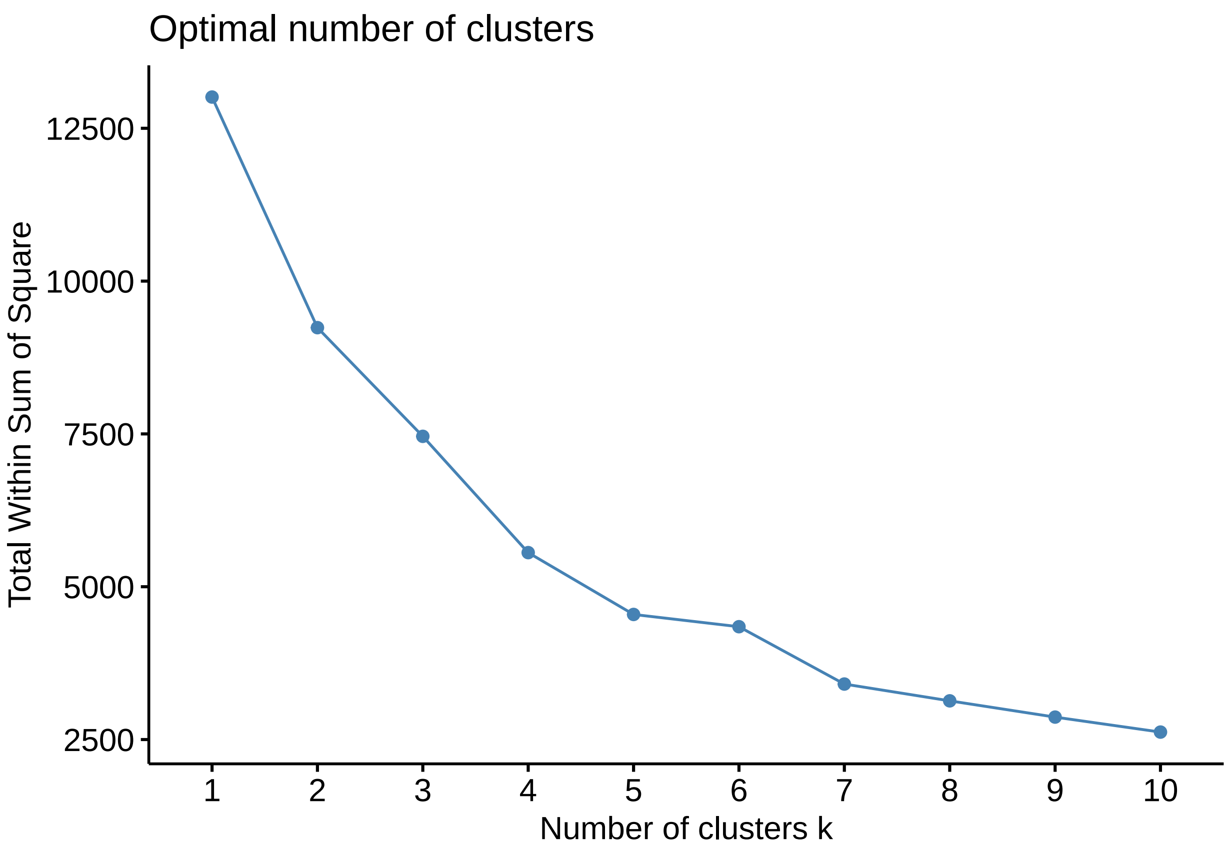



```
cd_RFM1 = cd_RFM%>%
dplyr::select(-CustomerID)

summary(cd_RFM1)

    recency        frequenci          monitery
 Min.   : 23.0   Min.   :  1.000   Min.   : 1.238
 1st Qu.: 40.0   1st Qu.:  1.000   1st Qu.: 5.185
 Median : 73.0   Median :  2.000   Median : 5.683
 Mean   :115.1   Mean   :  4.272   Mean   : 5.646
 3rd Qu.:164.8   3rd Qu.:  5.000   3rd Qu.: 6.064
 Max.   :396.0   Max.   :209.000   Max.   :11.341

cd_RFM2 <- cd_RFM1 %>%
mutate(recency = scale(recency),
       frequenci = scale(frequenci),
       monitery = scale(monitery)

)

summary(cd_RFM2)

      recency.V1          frequenci.V1          monitery.V1
 Min.   :-0.9204819   Min.   :-0.425048   Min.   :-5.883231
 1st Qu.:-0.7505027   1st Qu.:-0.425048   1st Qu.:-0.615310
 Median :-0.4205432   Median :-0.295144   Median : 0.049302
 Mean   : 0.0000000   Mean   : 0.000000   Mean   : 0.000000
 3rd Qu.: 0.4968443   3rd Qu.: 0.094568   3rd Qu.: 0.557567
 Max.   : 2.8090607   Max.   :26.594965   Max.   : 7.601186

set.seed(123)

# function to compute total within-cluster sum of square
wss <- function(k) {
  kmeans(cd_RFM2, k, nstart = 10 )$tot.withinss
```

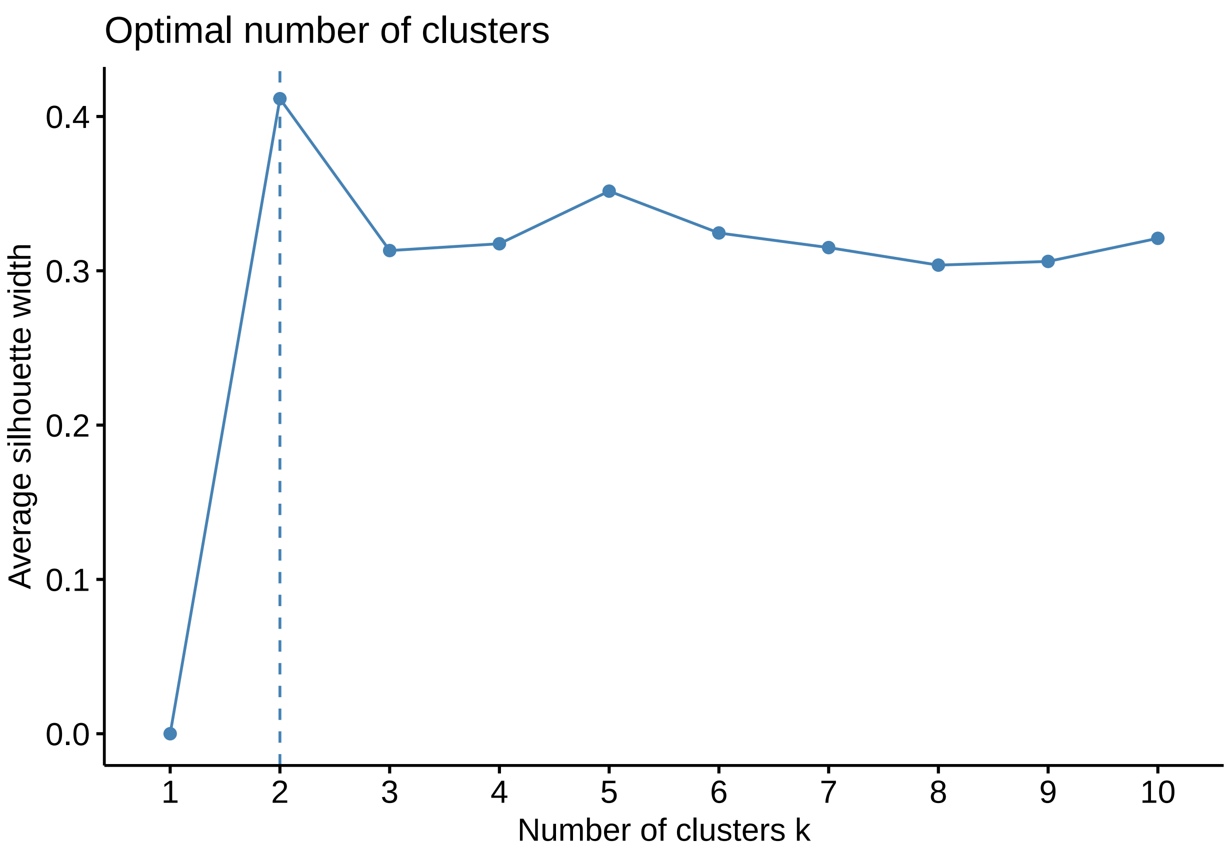
```
}

# Compute and plot wss for k = 1 to k = 15
k.values <- 1:15

# extract wss for 2-15 clusters
wss_values <- map_dbl(k.values, wss)

plot(k.values, wss_values,
       type="b", pch = 19, frame = FALSE,
       xlab="Number of clusters K",
       ylab="Total within-clusters sum of squares")
```



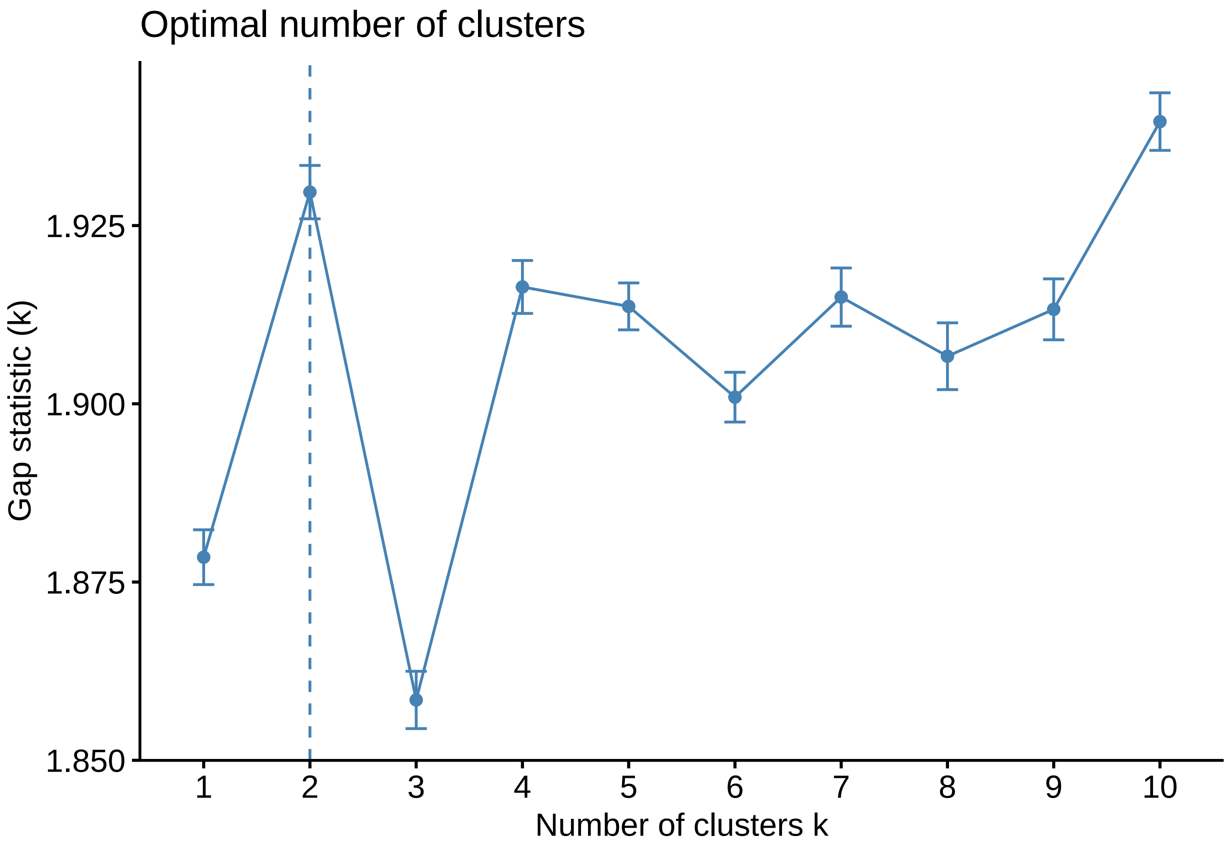

```
set.seed(123)

fviz_nbclust(cd_RFM2, kmeans, method = "wss")
```




```
set.seed(123)
fviz_nbclust(cd_RFM2, kmeans, method = "silhouette")
```

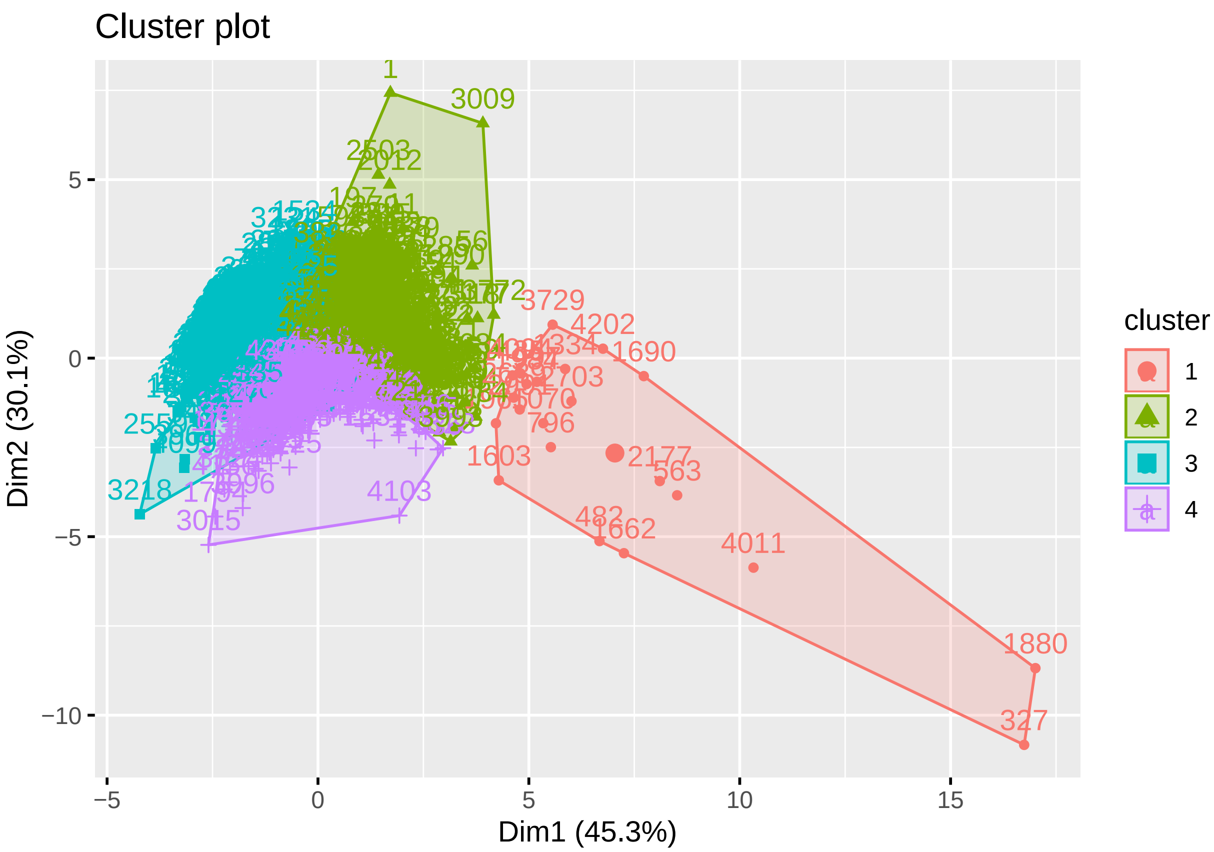


```
set.seed(123)
gap_stat <- clusGap(cd_RFM2, FUN = kmeans, nstart = 25,
                    K.max = 10, B = 50)
# Print the result
print(gap_stat, method = "firstmax")
fviz_gap_stat(gap_stat)
```




```
Clustering Gap statistic ["clusGap"] from call:
clusGap(x = cd_RFM2, FUNcluster = kmeans, K.max = 10, B = 50,     nstart = 25)
B=50 simulated reference sets, k = 1..10; spaceH0="scaledPCA"
 --> Number of clusters (method 'firstmax'): 2
          logW   E.logW       gap      SE.sim
 [1,] 7.665127 9.543614 1.878487 0.003841697
 [2,] 7.428966 9.358659 1.929692 0.003743205
 [3,] 7.385286 9.243754 1.858468 0.004020079
 [4,] 7.221823 9.138214 1.916391 0.003710861
 [5,] 7.142412 9.056076 1.913664 0.003289008
 [6,] 7.070778 8.971714 1.900935 0.003488891
 [7,] 6.992752 8.907722 1.914971 0.004083335
 [8,] 6.940304 8.846979 1.906675 0.004682498
 [9,] 6.896881 8.810133 1.913252 0.004272970
[10,] 6.836043 8.775627 1.939584 0.004036459
```

```
k2 <- kmeans(cd_RFM2, centers = 4, nstart = 25)

fviz_cluster(k2, data = cd_RFM2)
```



group 1: Champions
Bought recently, buy often and spend the most!
Reward them. Can be early adopters for new products. Will promote your brand.

group 2: Recent Customers
Bought most recently, but not often.
Provide on-boarding support, give them early success, start building relationship.

group 3: Hibernating Last purchase was long back, low spenders and low number of orders. Offer other relevant products and special discounts. Recreate brand value.

group 4: Promising
Recent shoppers, but haven't spent much.
Create brand awareness, offer free trials

```
cd_RFM3 <- cbind(cd_RFM, k2$cluster)
```

Note from the Authors

This file was generated using *The R Journal* style article template, additional information on how to prepare articles for submission is here - Instructions for Authors. The article itself is an executable R Markdown file that could be downloaded from Github with all the necessary artifacts.

Ketao Li
York University

liketao@yahoo.com

Kush Halani
York University

kush.halani@ontariotechu.net

Josue Romain

York University

josue.rolland.romain@gmail.com

Juan Peña
York University

jppena62@my.yorku.ca