Online Retail II RFM K-Means Clustering

Bima Putra Pratama

10/6/2019

Contents

Overview

This project is a part of final project from the HarvardX PH125.9x Data Science:Capstone course by Rafael Irizarry. The aim of this project is to do customer segmentation analysis from online retail II dataset from UCI ML repository. This analysis will focus on getting RFM values and clustering it using K-Means algorithms.

Customer segmentation is a method to grouping customers based on desired criteria. In this project the customers was divided into some groups based on their recency, frequency and monetary value. Recency is about when the last time customers make an order. It means the number of days since a customer made the last purcase. Frequency is the total number of customer purcase in a given period. Then monetary is the total amount of money customer spent in that period. These three values that are used as features to conduct K-Means clustering.

Several step was taken to complete this project. First is prepare the library and download the required dataset. Then the raw data are cleaned up and scaled use standardization and normalization prior to modelling. Furthermore the optimum values of cluster was determined by produced an elbow plot. Lastly the modelling was performed and summarized the results.

Dataset

Dataset Preparation

This project use Online Retail II dataset that will be downloaded from UCI Repository here http://archive.ics.uci.edu/ml/machine-learning-databases/00502/online_retail_II.xlsx. This dataset contains all the transactions occurring for a UK-based and registered, non-store online retail between 01/12/2009 and 09/12/2011.The company mainly sells unique all-occasion gift-ware. Many customers of the company are wholesalers. However this project only used the data from year 2010 - 2011 in the second worksheet of this dataset.

```r
# Prepare Required
if(!require(tidyverse)) install.packages("tidyverse",
                                         repos = "http://cran.us.r-project.org")
if(!require(lubridate)) install.packages("lubridate",
                                       repos = "http://cran.us.r-project.org")
if(!require(readxl)) install.packages("readxl",
                                        repos = "http://cran.us.r-project.org")
if(!require(GGally)) install.packages("GGally",
                                        repos = "http://cran.us.r-project.org")

# Import Dataset
files <- tempfile()
download.file("http://archive.ics.uci.edu/ml/machine-learning-databases/00502/online_retail_II.xlsx", f
df <- read_excel(files, sheet = 'Year 2010-2011', col_names = TRUE)
df <- df %>% rename(CustomerID = `Customer ID`) # Rename CustomerID Column Names
glimpse(df)
```

```
## Observations: 541,910
## Variables: 8
## $ Invoice     <chr> "536365", "536365", "536365", "536365", "536365", ...
## $ StockCode   <chr> "85123A", "71053", "84406B", "84029G", "84029E", "...
## $ Description <chr> "WHITE HANGING HEART T-LIGHT HOLDER", "WHITE METAL...
## $ Quantity    <dbl> 6, 6, 8, 6, 6, 2, 6, 6, 6, 6, 3, 3, 3, 32, 6, 6, 8...
## $ InvoiceDate <dttm> 2010-12-01 08:26:00, 2010-12-01 08:26:00, 2010-12...
## $ Price       <dbl> 2.55, 3.39, 2.75, 3.39, 3.39, 7.65, 4.25, 1.85, 1....
## $ CustomerID  <dbl> 17850, 17850, 17850, 17850, 17850, 17850, 17850, 1...
## $ Country     <chr> "United Kingdom", "United Kingdom", "United Kingdo...
```

Data summaries can be seen to get initial understanding as follow:

```r
summary(df)
```

```
##    Invoice            StockCode         Description
##  Length:541910      Length:541910      Length:541910
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     Quantity          InvoiceDate                      Price
##  Min.   :-80995.00   Min.   :2010-12-01 08:26:00   Min.   :-11062.06
##  1st Qu.:     1.00   1st Qu.:2011-03-28 11:34:00   1st Qu.:     1.25
##  Median :     3.00   Median :2011-07-19 17:17:00   Median :     2.08
##  Mean   :     9.55   Mean   :2011-07-04 13:35:22   Mean   :     4.61
##  3rd Qu.:    10.00   3rd Qu.:2011-10-19 11:27:00   3rd Qu.:     4.13
##  Max.   : 80995.00   Max.   :2011-12-09 12:50:00   Max.   : 38970.00
##
##    CustomerID       Country
##  Min.   :12346    Length:541910
##  1st Qu.:13953    Class :character
##  Median :15152    Mode  :character
##  Mean   :15288
##  3rd Qu.:16791
##  Max.   :18287
```

```
##  NA's   :135080
```

As can be seen on the summary of the dataset, several step need to be taken to prepare the dataset before clustering the data. Firstly, row's with negative values in the 'Quantity' and 'Price' columns will be removed. Then row's with NA's values also will be excluded from this dataset. Furthermore some column also need to be recoded to factor and changes the date and time to date type only prior to clustering by code below:

```
# Remove Rows that have Negative Values of Quantity and Price
clean_df <- df %>%
  filter(Quantity > 0 & Price > 0) %>%
  drop_na()

# Recode Dataset
Recode_df <- clean_df %>%
  mutate(Invoice = as.factor(Invoice), StockCode = as.factor(StockCode),
         InvoiceDate = date(InvoiceDate), CustomerID = as.factor(CustomerID),
         Country = as.factor(Country))

summary(Recode_df)
```

```
##     Invoice         StockCode      Description           Quantity
##  576339 :   542   85123A :  2035   Length:397885      Min.   :    1.00
##  579196 :   533   22423  :  1723   Class :character   1st Qu.:    2.00
##  580727 :   529   85099B :  1618   Mode  :character   Median :    6.00
##  578270 :   442   84879  :  1408                      Mean   :   12.99
##  573576 :   435   47566  :  1396                      3rd Qu.:   12.00
##  567656 :   421   20725  :  1317                      Max.   :80995.00
##  (Other):394983   (Other):388388
##   InvoiceDate             Price            CustomerID
##  Min.   :2010-12-01   Min.   :   0.001   17841  :  7847
##  1st Qu.:2011-04-07   1st Qu.:   1.250   14911  :  5675
##  Median :2011-07-31   Median :   1.950   14096  :  5111
##  Mean   :2011-07-10   Mean   :   3.117   12748  :  4595
##  3rd Qu.:2011-10-20   3rd Qu.:   3.750   14606  :  2700
##  Max.   :2011-12-09   Max.   :8142.750   15311  :  2379
##                                          (Other):369578
##            Country
##  United Kingdom:354321
##  Germany       :  9040
##  France        :  8342
##  EIRE          :  7236
##  Spain         :  2484
##  Netherlands   :  2359
##  (Other)       : 14103
```

Lastly, the total spend will be calculated by multiply price and quantity per transaction.

```
Recode_df <- Recode_df %>%
  mutate(TotalSpend = Quantity * Price)

summary(Recode_df)
```

```
##     Invoice         StockCode      Description           Quantity
##  576339 :   542   85123A :  2035   Length:397885      Min.   :    1.00
##  579196 :   533   22423  :  1723   Class :character   1st Qu.:    2.00
##  580727 :   529   85099B :  1618   Mode  :character   Median :    6.00
```

```
##  578270 :   442   84879  :  1408                      Mean   :   12.99
##  573576 :   435   47566  :  1396                      3rd Qu.:   12.00
##  567656 :   421   20725  :  1317                      Max.   :80995.00
##  (Other):394983   (Other):388388
##   InvoiceDate             Price            CustomerID
##  Min.   :2010-12-01   Min.   :   0.001   17841  :  7847
##  1st Qu.:2011-04-07   1st Qu.:   1.250   14911  :  5675
##  Median :2011-07-31   Median :   1.950   14096  :  5111
##  Mean   :2011-07-10   Mean   :   3.117   12748  :  4595
##  3rd Qu.:2011-10-20   3rd Qu.:   3.750   14606  :  2700
##  Max.   :2011-12-09   Max.   :8142.750   15311  :  2379
##                                          (Other):369578
##            Country         TotalSpend
##  United Kingdom:354321   Min.   :     0.00
##  Germany       :  9040   1st Qu.:     4.68
##  France        :  8342   Median :    11.80
##  EIRE          :  7236   Mean   :    22.40
##  Spain         :  2484   3rd Qu.:    19.80
##  Netherlands   :  2359   Max.   :168469.60
##  (Other)       : 14103
```

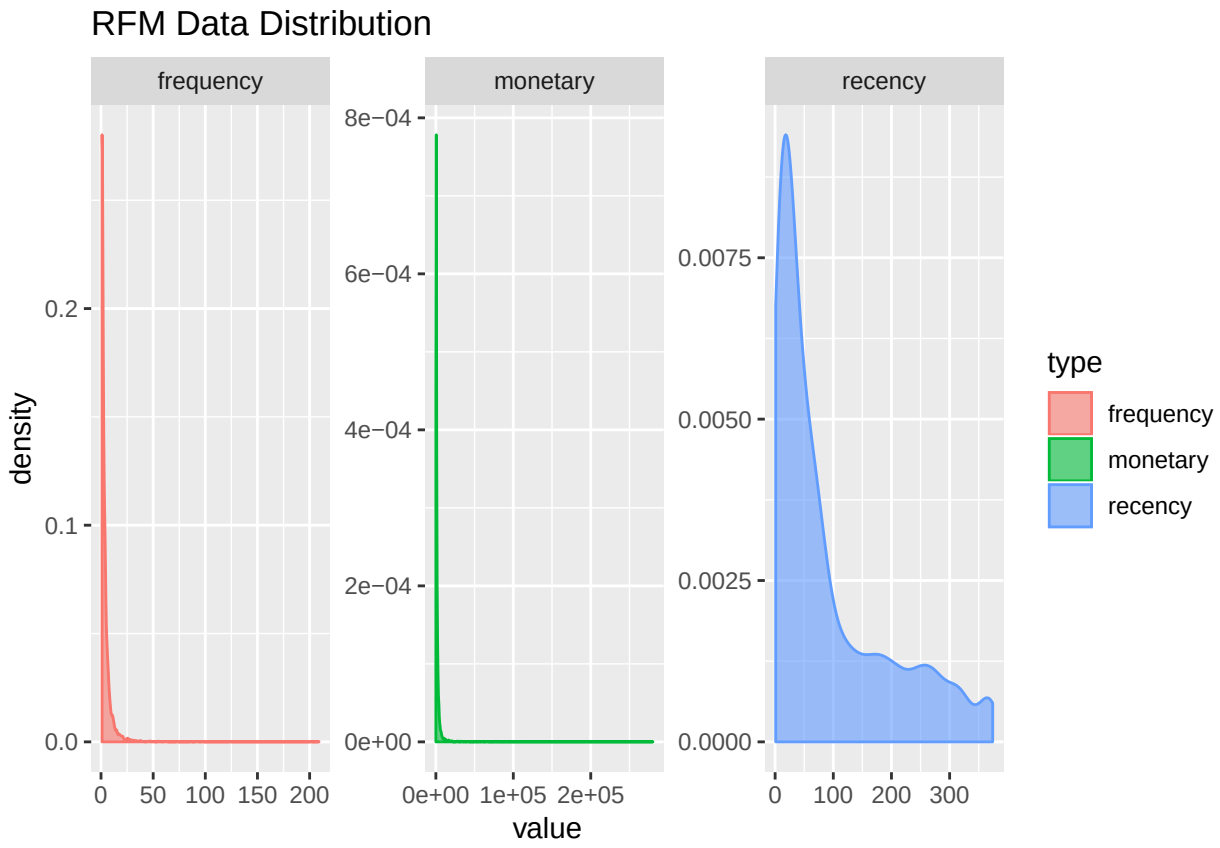
RFM Data Preparation

In order to perform the RFM Analysis, further process to the data set is required as follows: 1. Get the day after last InvoiceDate and used it as reference date. 2. Find the latest transaction date and calculate the day to the reference date per customer to get recency values. 3. Determine how many transaction has been done per customer and named as frequency values. 4. Total spend per customer that named as monetary values.

```
# Get Analysis Reference Date (One Day After Last Transaction)
Max_Date <- date(max(Recode_df$InvoiceDate)) + 1

# Calculate RFM Values
rfm_df <- Recode_df %>%
  group_by(CustomerID) %>%
  summarise(recency = as.numeric(Max_Date - max(InvoiceDate)),
            frequency = n_distinct(Invoice), monetary = sum(TotalSpend))

head(rfm_df)
```

```
## # A tibble: 6 x 4
##   CustomerID recency frequency monetary
##   <fct>        <dbl>     <int>    <dbl>
## 1 12346          326         1   77184.
## 2 12347            3         7    4310
## 3 12348           76         4    1797.
## 4 12349           19         1    1758.
## 5 12350          311         1     334.
## 6 12352           37         8    2506.
```

```
rfm_df %>%
  gather(type,value,recency:monetary) %>%
  ggplot(aes(x = value, color = type, fill = type)) +
  geom_density(alpha = 0.6) +
  facet_wrap(~type, nrow = 1, scales="free") +
  labs(title = 'RFM Data Distribution')
```



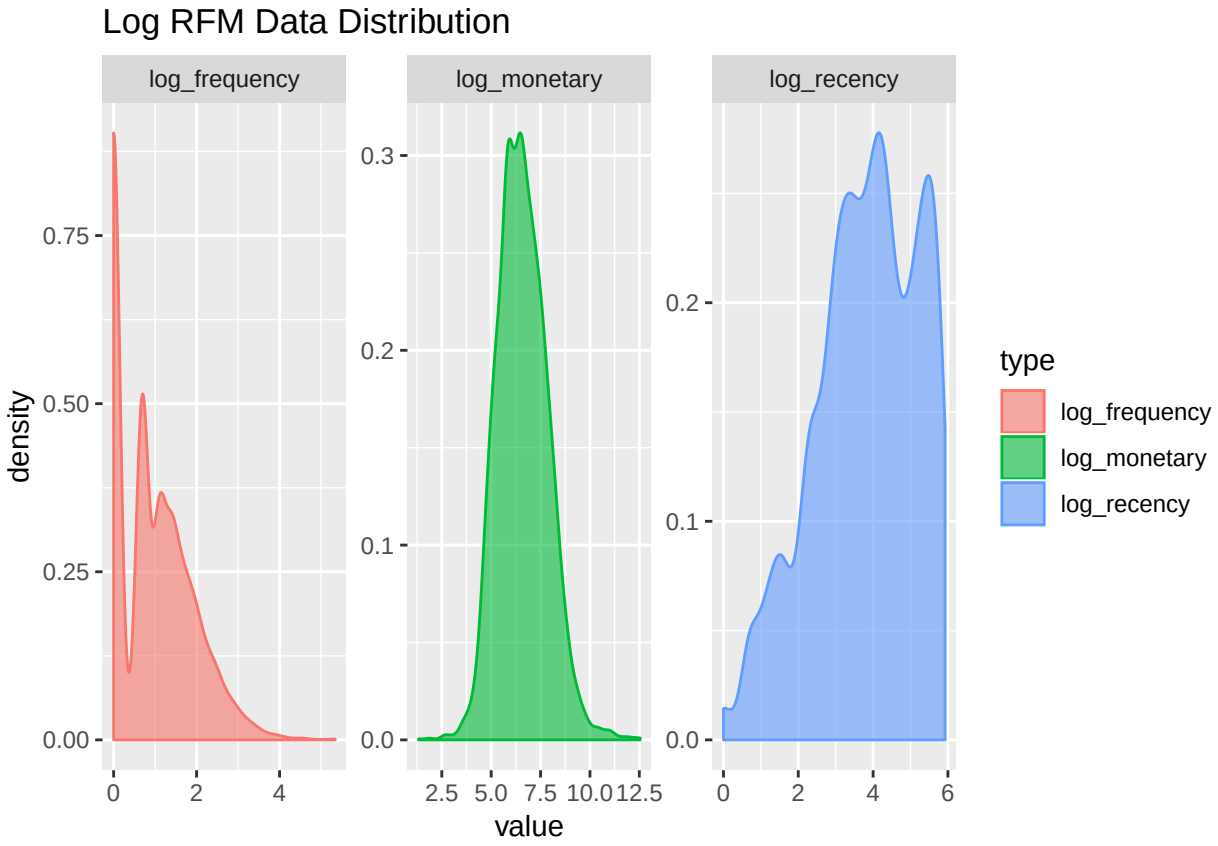
Method and Analysis

In this project, K-Means method was used to identify groups accross all customer. K-Means clustering is one type of unsupervised learning algorithms, which makes groups based on the distance between the points.

Data Transformation

As can be seen on RFM Data Distribution graph, the data is highly skewed especially on frequency and monetary. In order to get more sense on the data, the log transformation can be applied:

```
log_rfm <- rfm_df %>%
  mutate(log_recency = log(recency), log_frequency = log(frequency), log_monetary = log(monetary))

log_rfm %>%
  gather(type,value,log_recency:log_monetary) %>%
  ggplot(aes(x = value, color = type, fill = type)) +
  geom_density(alpha = 0.6) +
  facet_wrap(~type, nrow = 1, scales="free") +
  labs(title = 'Log RFM Data Distribution')
```

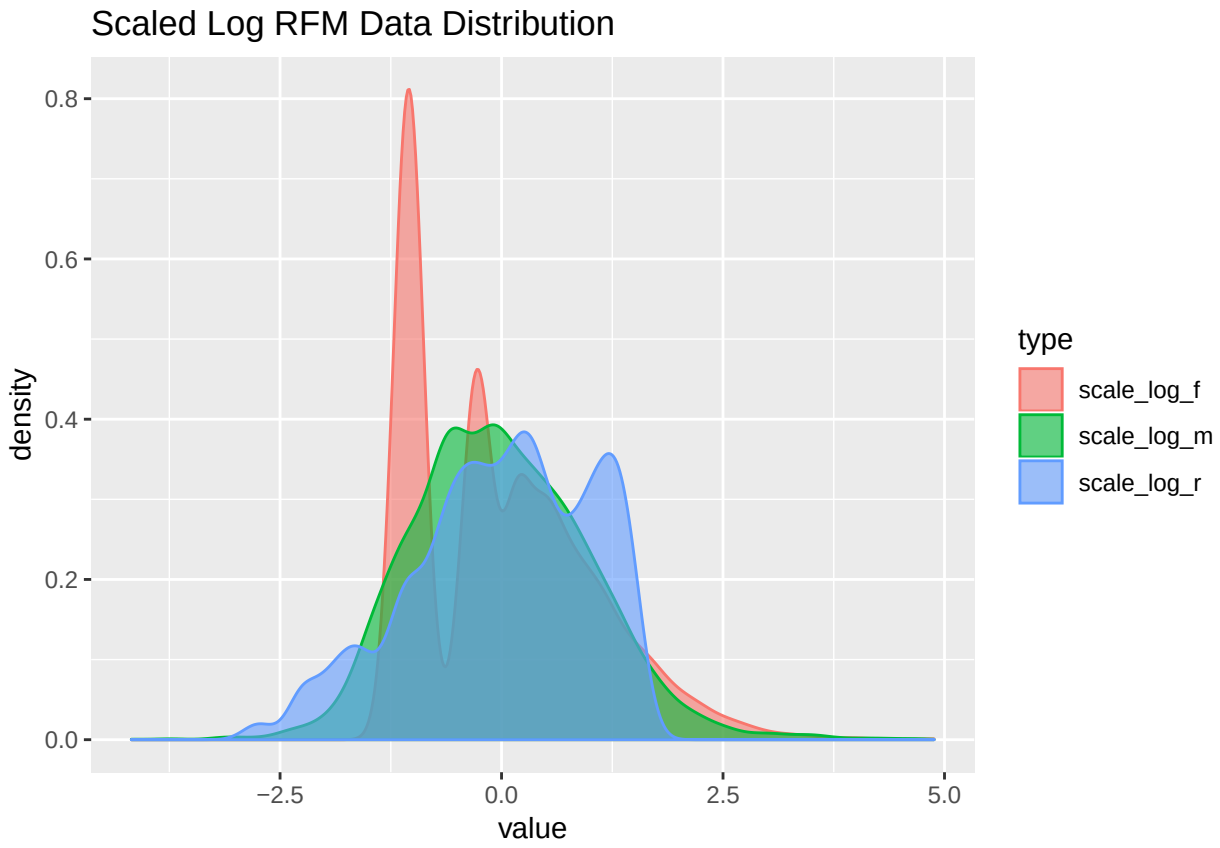


Moreover, Due to the used of distance in K-Means method, the features unit scale is important. Hence it is required to do standardization and normalisation by finding z-score of features prior to do clustering. This can be done by calculating by using this formula :

$$z = \frac{x - \mu}{\sigma}$$

That calculation can be done by using scale function in R.

```
scale_df <- log_rfm %>%
 mutate(scale_log_r = scale(log_recency), scale_log_f = scale(log_frequency), scale_log_m = scale(log_m

scale_df %>%
  gather(type,value,scale_log_r:scale_log_m) %>%
  ggplot(aes(x = value, color = type, fill = type)) +
  geom_density(alpha = 0.6) +
  labs(title = 'Scaled Log RFM Data Distribution')
```

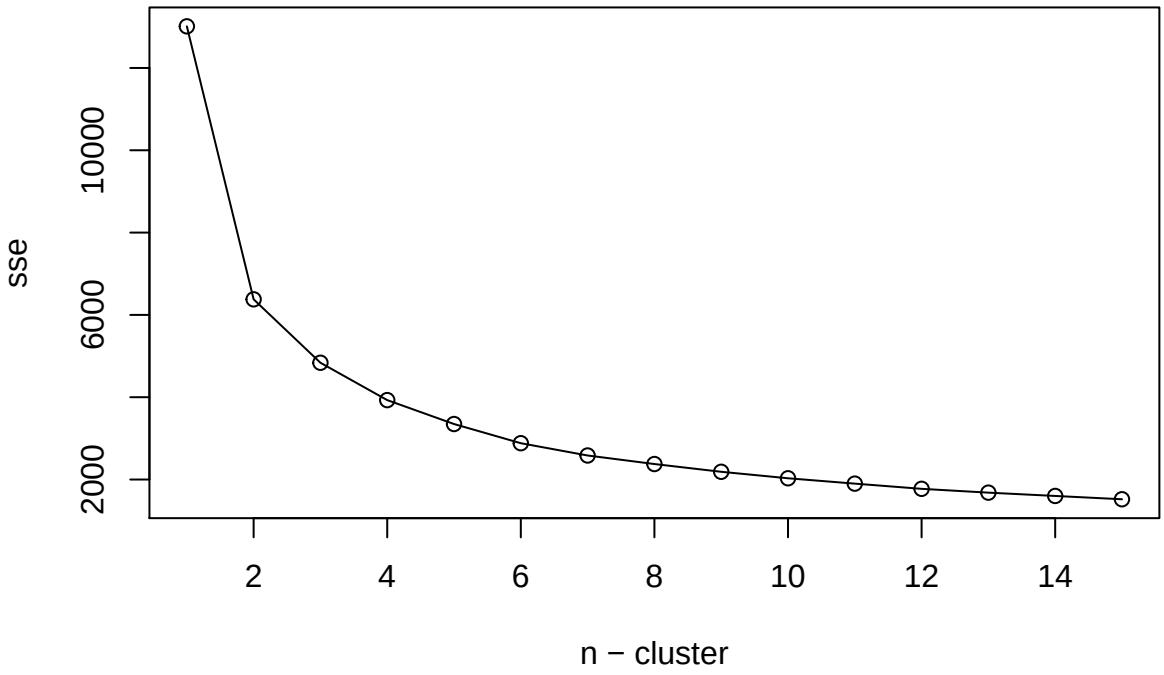


All the features already on the same scale after standardization and normalization. So, the preprocessing data has been done and clustering process can be performed by this scaled dataset.

Clustering with K-Means

The first step in this clustering is to find the right number of cluster. This process can be done by make a elbow curve and choose the most optimized cluster based on that curve. Elbow curve is a curve that made by plotting Sum Square Error (SSE) from the K-Means alghoritm. This number represent the sum square value of the actual point distance to the central of each cluster.

```
# Iterate from 1 to 15 to find the most optimum cluster by elbow curve
set.seed(100)
used_var = c("scale_log_r","scale_log_f","scale_log_m")
sse <- sapply(1:15,
              function(k)
              {
                kmeans(x=scale_df[used_var], k, nstart=25)$tot.withinss
              }
)

plot(sse, type = "o", xlab = "n - cluster", main = 'Elbow Curves')
```



Observing from the elbow curve, the most optimum cluster was pictured as the elbow of the curve somewhere SSE dramatically decrease but not to much. In this case 4 was choosen to be the most optimum cluster.

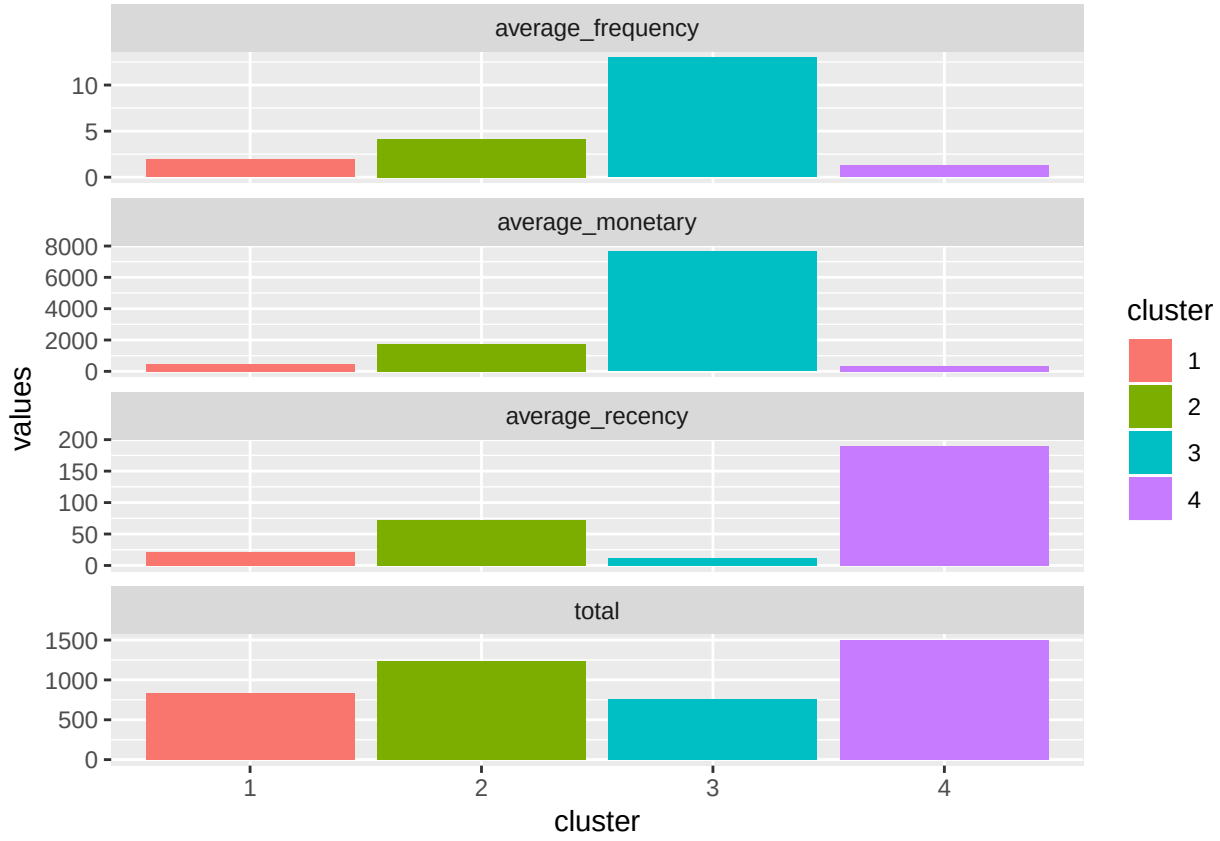
After decided the cluster numbers, a model can be build and make an actual cluster like below:

```
# Calculate Cluster group with 4 cluster
segment_4 <- kmeans(x=scale_df[used_var], 4, nstart=25)
cluster <- as.factor(segment_4$cluster)
rfm_clustered4 <- cbind(scale_df,cluster)

rfm_clust4_summary <- rfm_clustered4 %>%
  group_by(cluster) %>%
  summarise(total = n_distinct(CustomerID),
            average_recency = round(mean(recency),2),
            average_frequency = round(mean(frequency),2),
            average_monetary = round(mean(monetary),2)
  )

rfm_clust4_summary %>% knitr::kable()
```

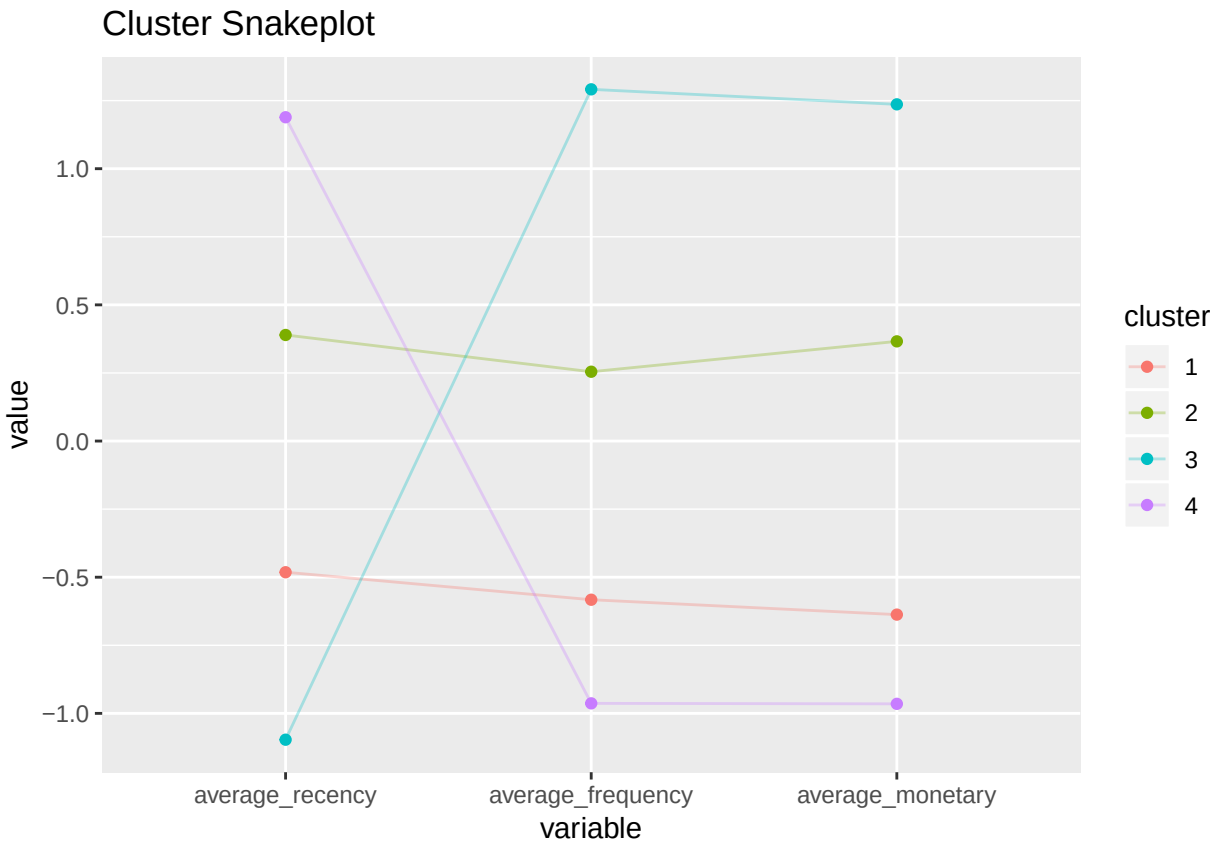
cluster	total	average_recency	average_frequency	average_monetary
1	833	22.18	1.90	483.53
2	1239	72.74	4.13	1743.12
3	762	11.13	13.01	7653.64
4	1504	190.56	1.27	343.64

```
rfm_clust4_summary %>%
  gather(key = 'measure', value = 'values',c(5,4,3,2)) %>%
  ggplot(aes(x = cluster , y = values, fill = cluster)) +
```

```
  geom_col() +
  facet_wrap(~measure, ncol = 1, scales="free_y")
```




```
rfm_clustered4 %>%
  group_by(cluster) %>%
  summarise(average_recency = round(mean(scale_log_r),2),
            average_frequency = round(mean(scale_log_f),2),
            average_monetary = round(mean(scale_log_m),2)) %>%
  ggparcoord(columns = 2:4, groupColumn = 'cluster',
           showPoints = TRUE,
           alphaLines = 0.3, title = "Cluster Snakeplot")
```



Results

As a result, there are 4 categories of customers generated in this project. Cluster 1 can be categorize as most valuable customer. Customer in this category have spent most frequently and spent the most money. On the other Cluster 2 have less frequent and less value of money compared to Cluster 1. However they haven't transact recently. Cluster 3 recently have transaction but not too frequent and only spend small amount of money. This Cluster can be a new customer that just done the transaction. Lastly, cluster 4 become our loss customer with the least frequent and monetary value and havent done any transaction for a while.

As a summary the detail for each cluster can be seen as follow:

```
rfm_clust4_summary %>% knitr::kable()
```

cluster	total	average_recency	average_frequency	average_monetary
1	833	22.18	1.90	483.53
2	1239	72.74	4.13	1743.12
3	762	11.13	13.01	7653.64
4	1504	190.56	1.27	343.64

Conclusion

Finnaly, a cluster have successfuly build and each Customer can be categories based on their recency, frequency and monetary values. Furthermore this cluster also can be used as a basis to give different treatment to gain more benefit to the business.

Further analysis also can be applied and introduce more variable like tenure or how many days since the customer doing first transaction to the last day of their transaction. More detail analysis also can be done by give more specific time range like RFM for yearl, monthly or weekly to see how our customer perform during that period.