Travail Pratique de Fouille de donnees

Hugues Kanda Madimba

Introduction

Dans ce travail pratique, il vous a ete demande de travailler su les points ci-dessous:

- Choisir un jeu de donnees qui constitue l'objet d'etudes pour tous les TPs du cours de fouille de donnees;
- Faire une description detaillee de ce jeu de donnees (description des attributs, nombre d'individus, valeurs manquantes, probleme pose);
- Faire une analyse exploratoire de chaque attribut;
- Faire une analyse de lien entre chaque paire d'atributs.

1. Choix de jeu de donnees

We chose an open dataset that UCI offered for academic and research purposes: this dataset includes online activities related to a group of retailers (http://archive.ics.uci.edu/ml/datasets/online+retail).

```
library(readxl)
library(lubridate)
library(tibble)
library(magrittr)
library(tidyverse)
library(tidyquant)
library(modelr)
library(gridExtra)
library(grid)


dataload <- read_xlsx("~/assignment_one/data_online_retail.xlsx")
data <- dataload %>% as_tibble()
```

The data structure is described as follows: 541909 observations and 14 number of features.

This is a transnational data set which contains all the transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail.The company mainly sells unique all-occasion gifts. Many customers of the company are wholesalers.

The data has been collected by Dr Daqing Chen (http://www.lsbu.ac.uk/about-us/people-finder/dr-daqing-chen), Director: Public Analytics group, School of Engineering, London South Bank University, London SE1 0AA, UK.

2. Dataset description

2.1. Variable description

- InvoiceNo: Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter ‘c’, it indicates a cancellation.
- StockCode: Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
- Description: Product (item) name. Nominal.
- Quantity: The quantities of each product (item) per transaction. Numeric.
- InvoiceDate: Invice Date and time. Numeric, the day and time when each transaction was generated.
- UnitPrice: Unit price. Numeric, Product price per unit in sterling.
- CustomerID: Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.
- Country: Country name. Nominal, the name of the country where each customer resides.

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    541909 obs. of  14 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 COAT HANGER" "KNITTED UNION FLAG HOT WATER BOTTLE" ...
##  $ Quantity   : num  6 6 8 6 6 2 6 6 6 32 ...
##  $ InvoiceDate: POSIXct, format: "2010-12-01 08:26:00" "2010-12-01 08:26:00" ...
##  $ UnitPrice  : num  2.55 3.39 2.75 3.39 3.39 7.65 4.25 1.85 1.85 1.69 ...
##  $ CustomerID : num  17850 17850 17850 17850 17850 ...
##  $ Country    : chr  "United Kingdom" "United Kingdom" "United Kingdom" "United
Kingdom" ...
##  $ day        : Date, format: "2010-12-01" "2010-12-01" ...
##  $ day_of_week: Ord.factor w/ 7 levels "Sun"<"Mon"<"Tue"<..: 4 4 4 4 4 4 4 4 4
4 ...
##  $ time       :Classes 'hms', 'difftime'  atomic [1:541909] 30360 30360 30360 30
360 30360 ...
##   .. ..- attr(*, "units")= chr "secs"
##  $ month      : chr  "12" "12" "12" "12" ...
##  $ income     : num  15.3 20.3 22 20.3 20.3 ...
##  $ return     : chr  "income" "income" "income" "income" ...
```

2.2. Missing values

- The total count of missing values is 136534
- Below, the total count of missing values per variable

```
##   InvoiceNo   StockCode Description    Quantity InvoiceDate   UnitPrice
##           0           0        1454           0           0           0
##  CustomerID     Country         day day_of_week        time       month
##      135080           0           0           0           0           0
##      income      return
##           0           0
```

- In case we want to omit missing values, the number of occurrences that will remain is:

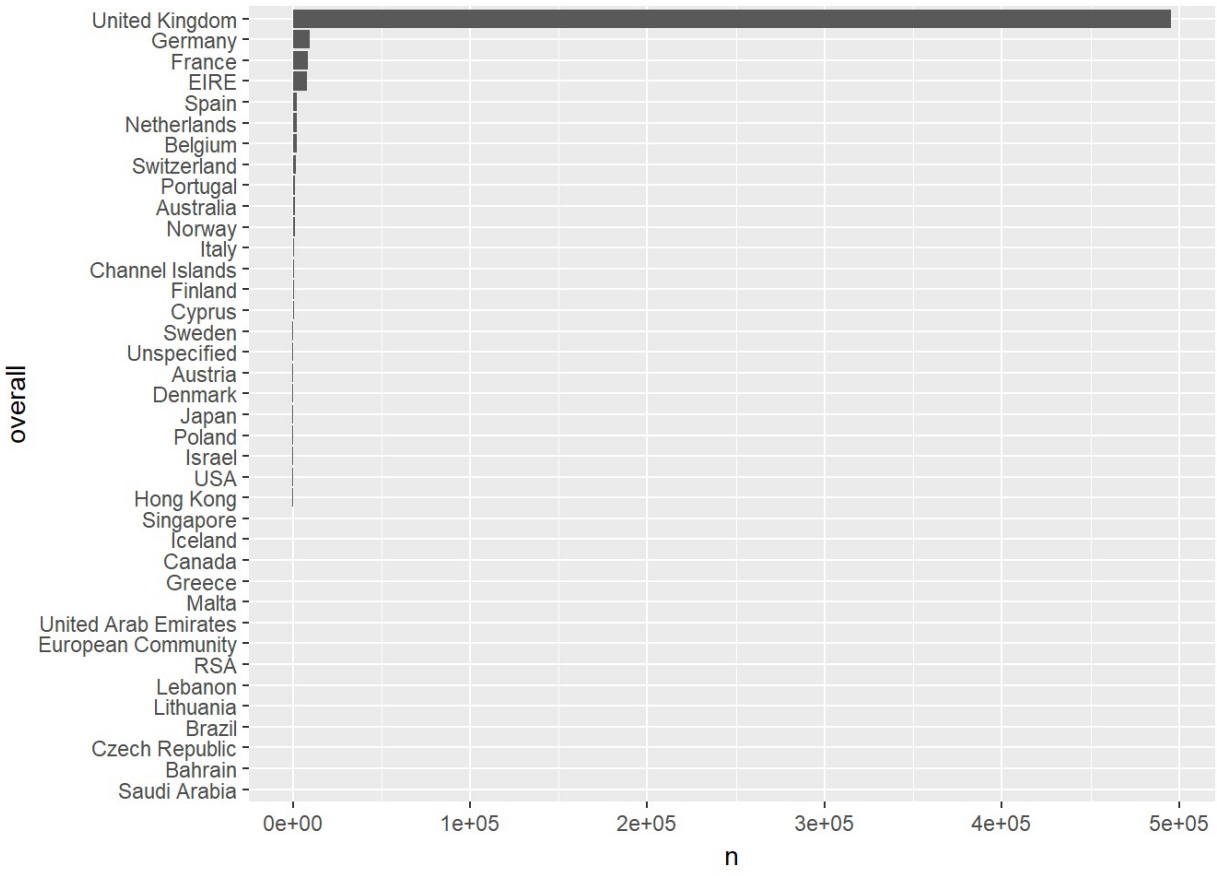
```
## [1] 406829
```

2.3. Variables details

- Total of count of distinct `CustomerID` in overall is 4373 while the total distinct of number of `CustomerID` without missing values is 4372. Despite the missing values the number of distinct customers is slightly the same.
- Total of count of distinct `Country` in overall is 38 while the total distinct of number of `Country` without missing values is 37.

The number of transaction by country in overall…

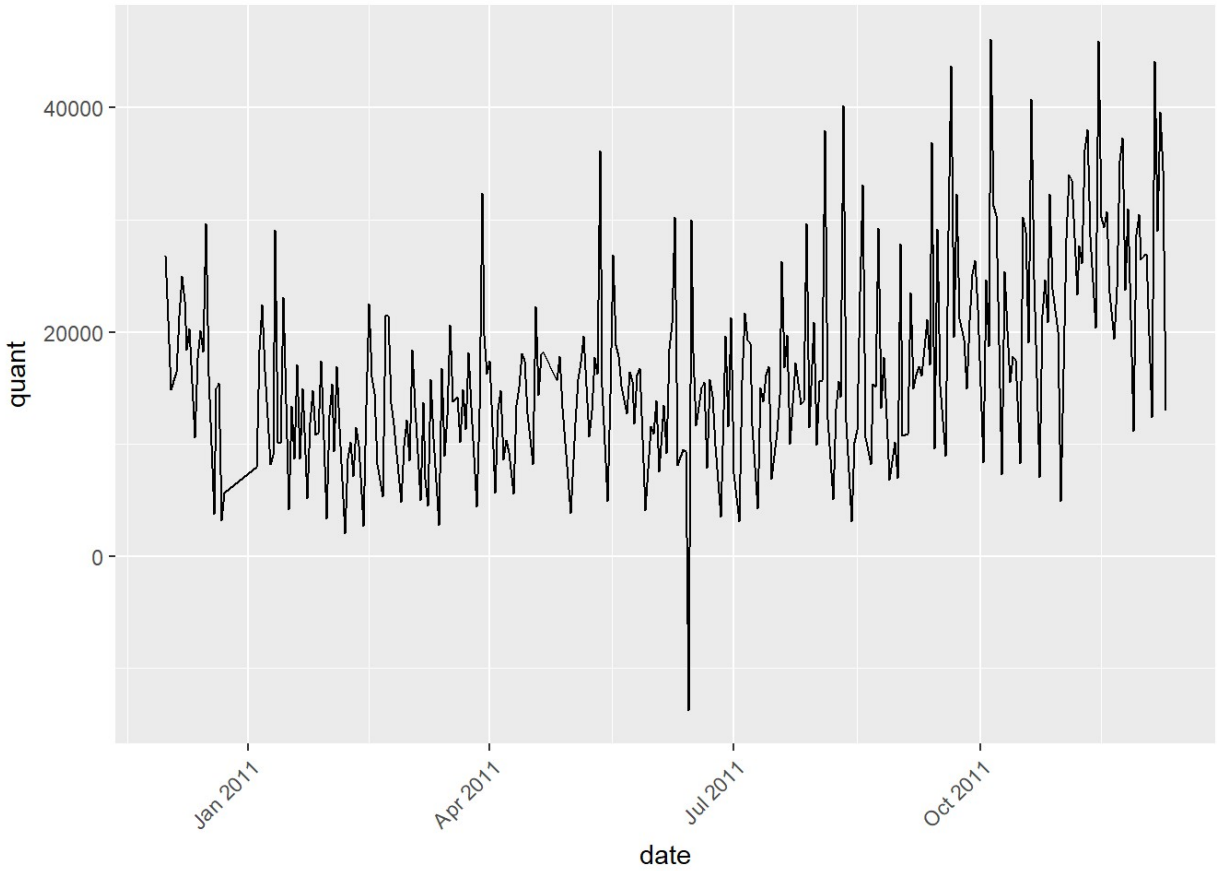
```
pdata <- data %>%  count(Country)

p <- ggplot(pdata, aes(y=n))
pdata$overall <- reorder(pdata$Country, pdata$n)
p + geom_bar(aes(x=overall), data=pdata, stat="identity") + coord_flip()
```



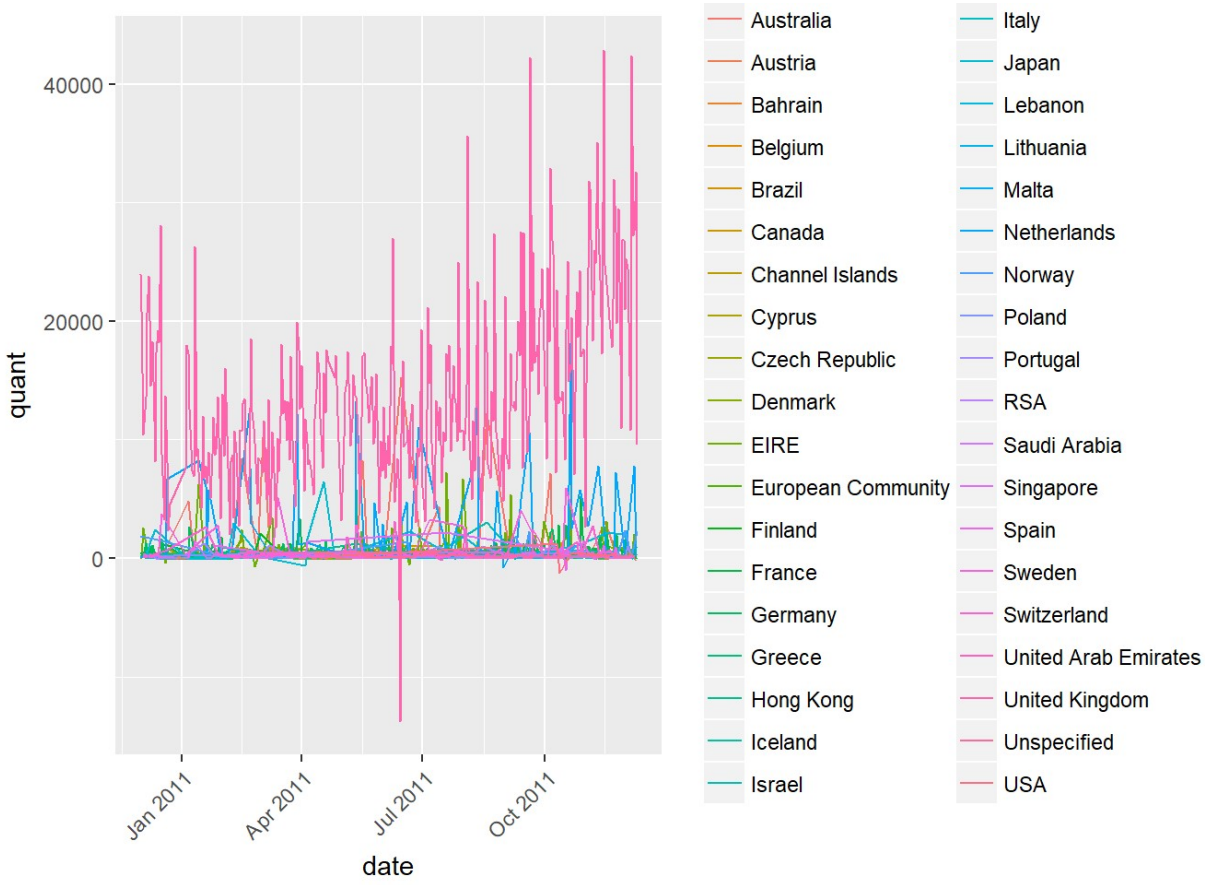
As we can see: UK are largely leading the number of transactions and widely leave a big gap with Germany which is the second in the ascending order. Otherwise, Saudi Arabia is the last country with an insignificant number of transactions.

- Visualization of the evolution of the quantities of each product per transaction from 01/12/2010 to 09/12/2011.

```
data %>% mutate(date = as.Date(InvoiceDate))  %>%
  group_by(date)  %>%
  summarise(quant = sum(Quantity)) %>%
  ggplot(aes(x=date, y=quant)) +
  geom_line() +  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



- Visualization of the evolution of the quantities of each product per transaction from 01/12/2010 and 09/12/2011, including the country where the transaction has been run.



3. Exploratory Data Analysis

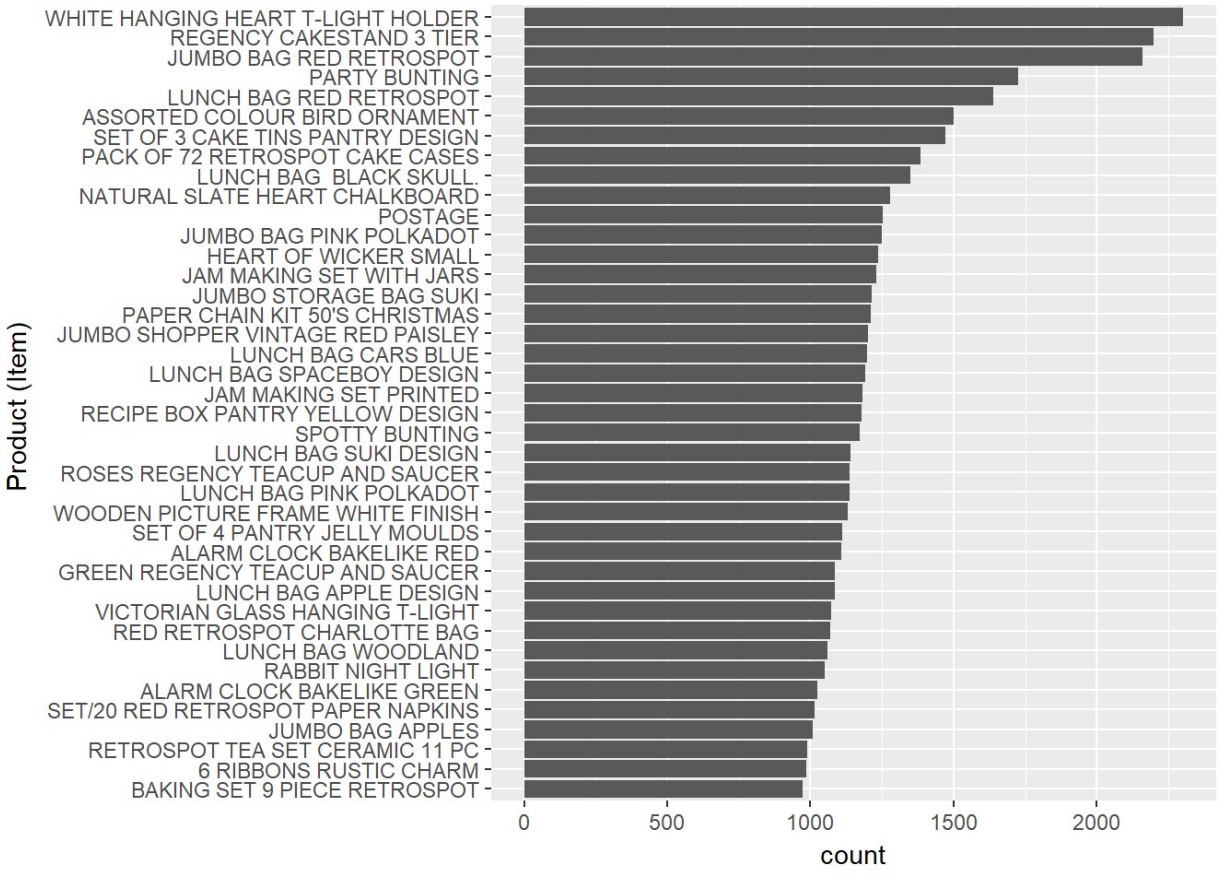
- Dataset summary

```
summary(data)
```

```
##   InvoiceNo          StockCode         Description
##  Length:541909      Length:541909      Length:541909
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     Quantity          InvoiceDate                    UnitPrice
##  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 13:34:57   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              day             day_of_week
##  Min.   :12346    Length:541909      Min.   :2010-12-01   Sun: 64375
##  1st Qu.:13953    Class :character   1st Qu.:2011-03-28   Mon: 95111
##  Median :15152    Mode  :character   Median :2011-07-19   Tue:101808
##  Mean   :15288                       Mean   :2011-07-04   Wed: 94565
##  3rd Qu.:16791                       3rd Qu.:2011-10-19   Thu:103857
##  Max.   :18287                       Max.   :2011-12-09   Fri: 82193
##  NA's   :135080                                           Sat:     0
##      time             month              income
##  Length:541909     Length:541909      Min.   :-168469.60
##  Class1:hms        Class :character   1st Qu.:      3.40
##  Class2:difftime   Mode  :character   Median :      9.75
##  Mode  :numeric                       Mean   :     17.99
##                                       3rd Qu.:     17.40
##                                       Max.   : 168469.60
##
##     return
##  Length:541909
##  Class :character
##  Mode  :character
##
##
##
##
```

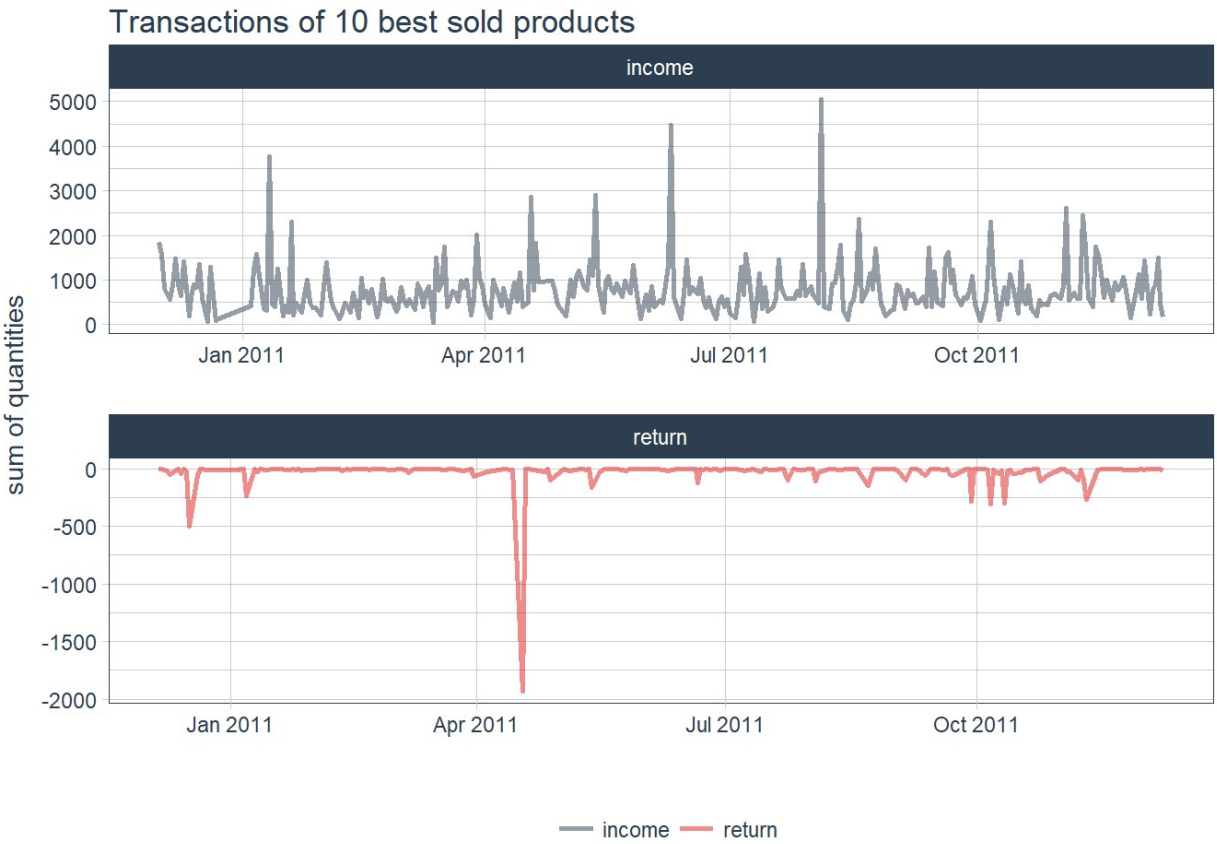
- The 40 best sold product

```
tmp <- data %>%
  group_by(StockCode, Description) %>%
  summarize(count = n()) %>%
  arrange(desc(count))
tmp <- head(tmp, n=40)

tmp %>%
  ggplot(aes(x=reorder(Description,count), y=count))+
  geom_bar(stat="identity") +
  coord_flip()  +  xlab("Product (Item) ") + scale_fill_continuous(guide=FALSE)
```

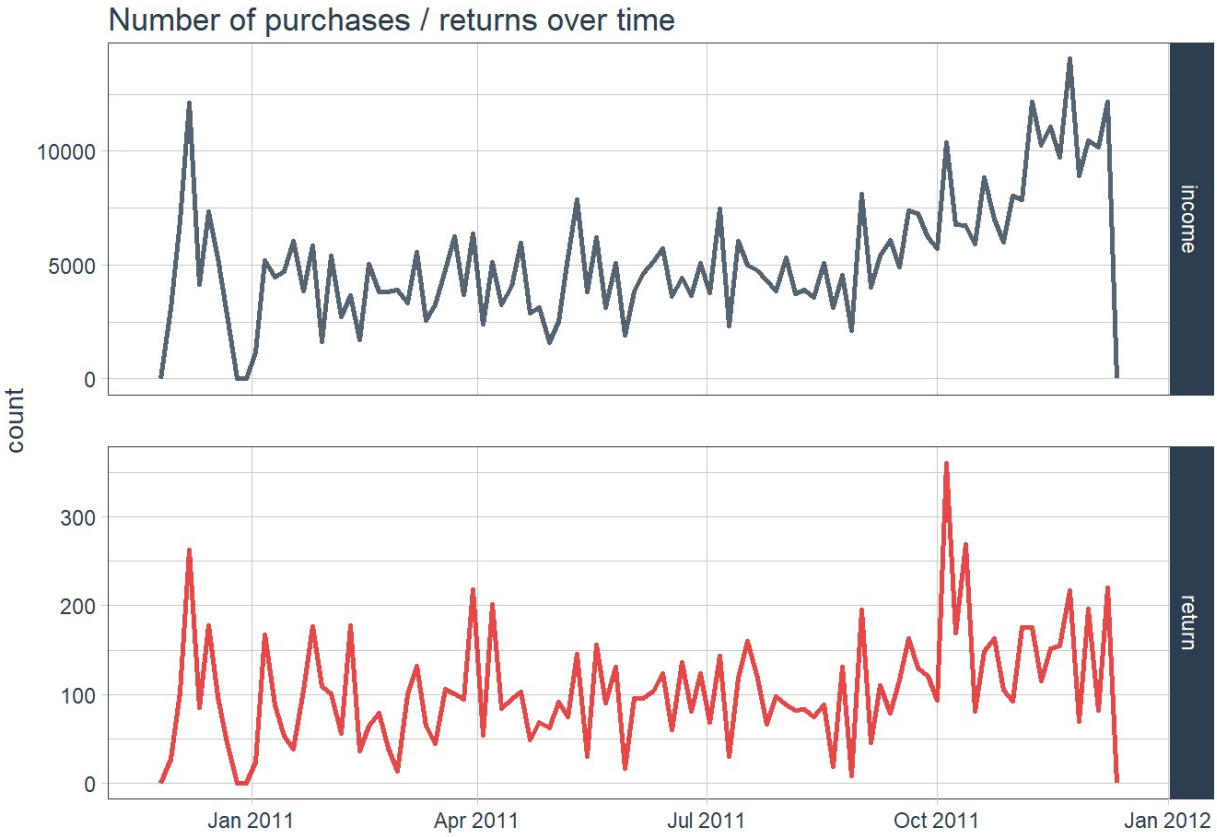


- Transactions of 10 best sold products: We added variables namely `income` and `return`. The `income` describes the amount of money realized from a single transaction (Quantity * UnitPrice) while the `return` shows if a particular transaction has generated an income or recorded a loss.

```
data %>%
  filter(StockCode %in% c("85123A","22423","85099B", "47566", "20725", "84879", "22
720", "21212","20727", "22457")) %>%
  group_by(day, return) %>%
  summarise(sum = sum(Quantity)) %>%
  ggplot(aes(x = day, y = sum, color = return)) +
  facet_wrap(~ return, ncol = 1, scales = "free") +
  geom_line(size = 1, alpha = 0.5) +
  scale_color_manual(values = palette_light()) +
  theme_tq() +
  labs(x = "", y = "sum of quantities",
       color = "", title = "Transactions of 10 best sold products")
```



- Number of purchases / returns over time



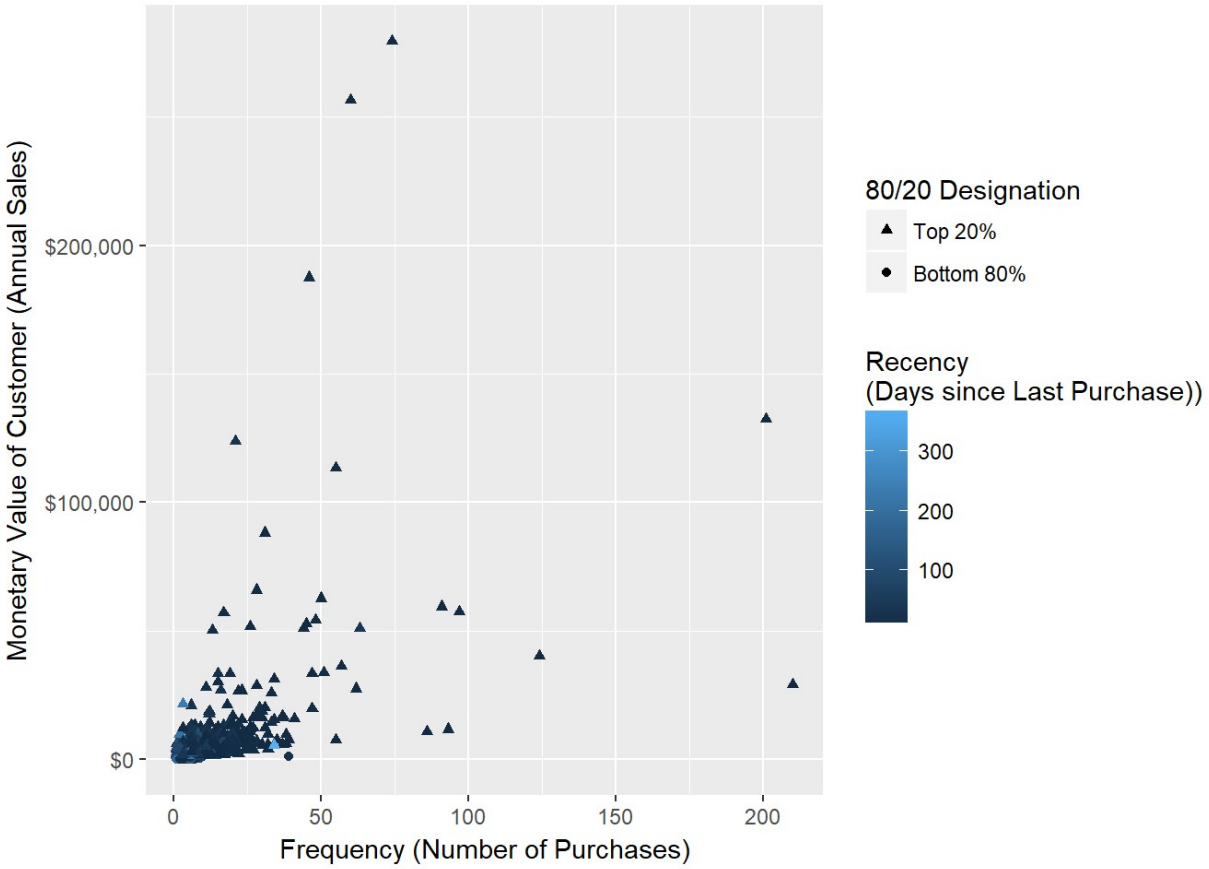
- Test

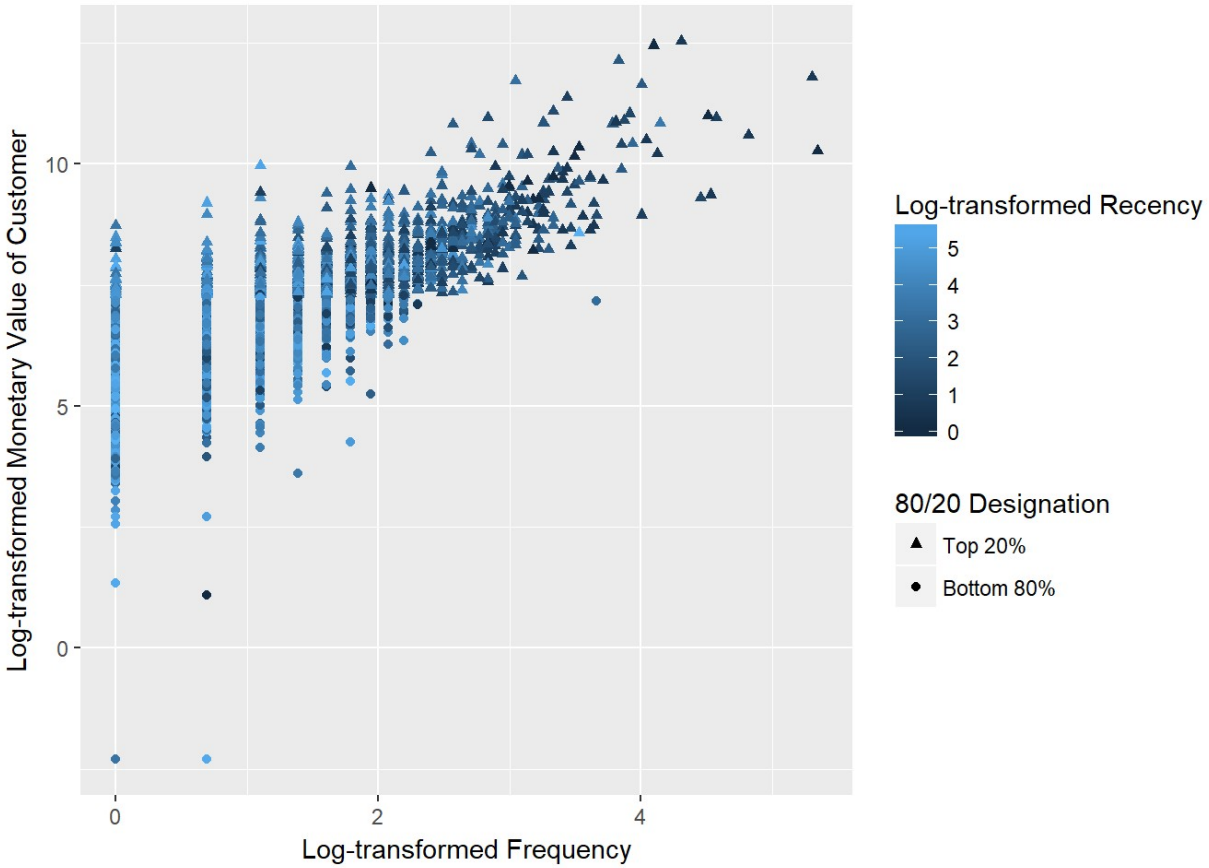
```
## [1] NA NA
```

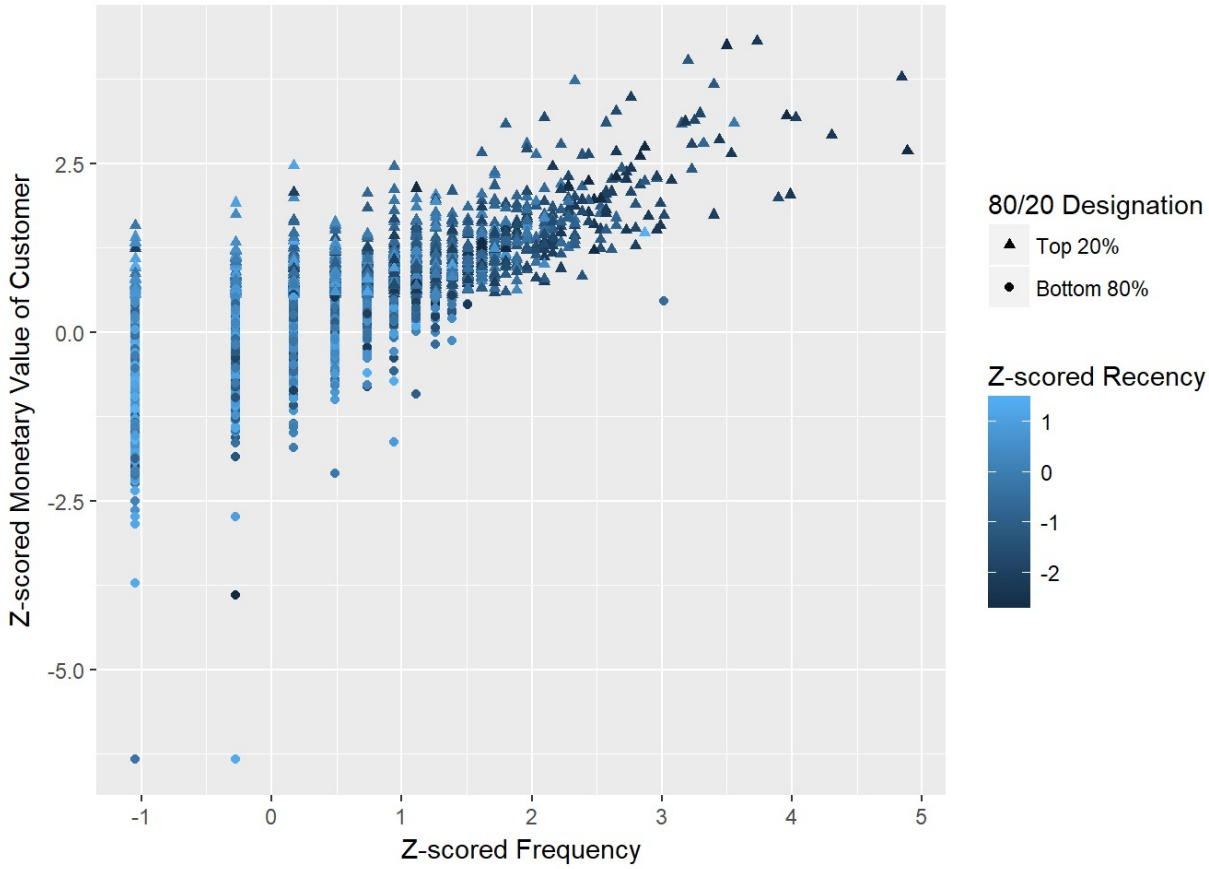
```
##
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14
##   33 1494  835  508  387  243  172  143   98   68   54   52   45   30   20
##   15   16   17   18   19   20   21   22   23   24   25   26   27   28   29
##   28   11   18   14   12   12   11    5    5    3    8    7    3    6    1
##   30   31   32   33   34   35   37   38   39   41   44   45   46   47   48
##    4    3    3    2    3    1    3    2    2    1    1    1    1    2    1
##   50   51   55   57   60   62   63   74   86   91   93   97  124  201  210
##    1    1    2    1    1    1    1    1    1    1    1    1    1    1    1
```

```
## [1] "Top 20%"    "Bottom 80%"
```

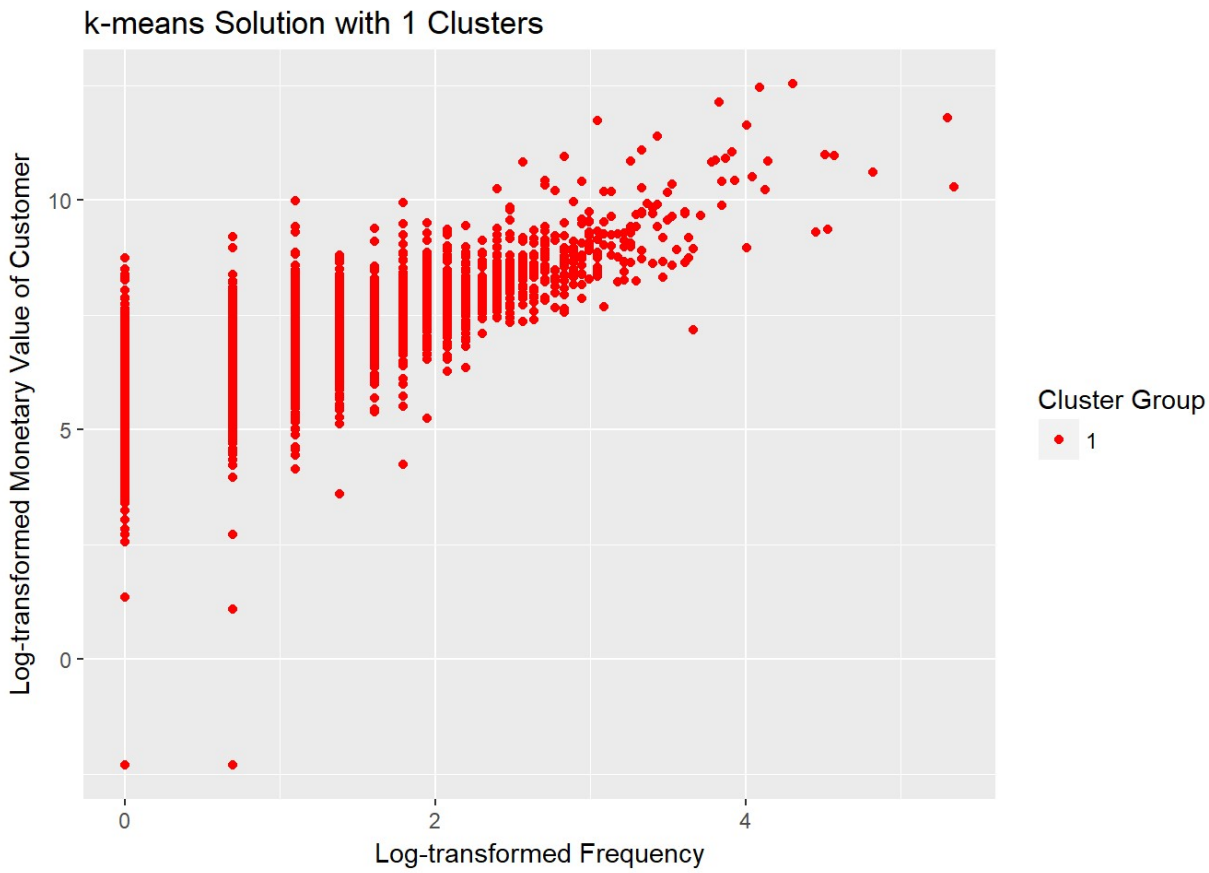
```
##
##    Top 20% Bottom 80%
##       0.27       0.73
```


Monetary Value of Customer (Annual Sales)
$200,000
$100,000
$0
0
50
100
150
200
Frequency (Number of Purchases)
80/20 Designation
Top 20%
Bottom 80%
Recency
(Days since Last Purchase))
300
200
100

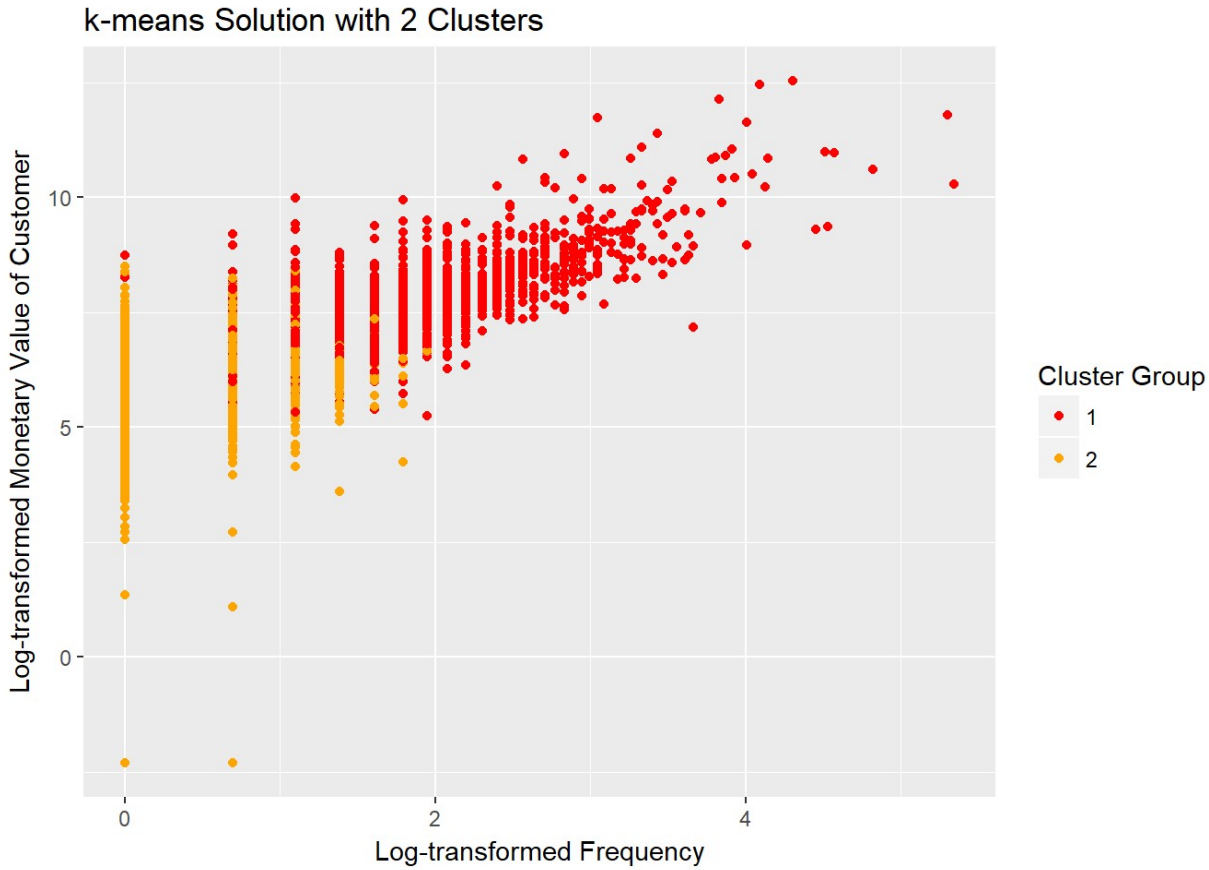

Log-transformed Monetary Value of Customer
10
5
0
0
2
4
Log-transformed Frequency
Log-transformed Recency
5
4
3
2
1
0
80/20 Designation
Top 20%
Bottom 80%


Z-scored Monetary Value of Customer
2.5
0.0
-2.5
-5.0
-1
0
1
2
3
4
5
Z-scored Frequency
80/20 Designation
Top 20%
Bottom 80%
Z-scored Recency
1
0
-1
-2

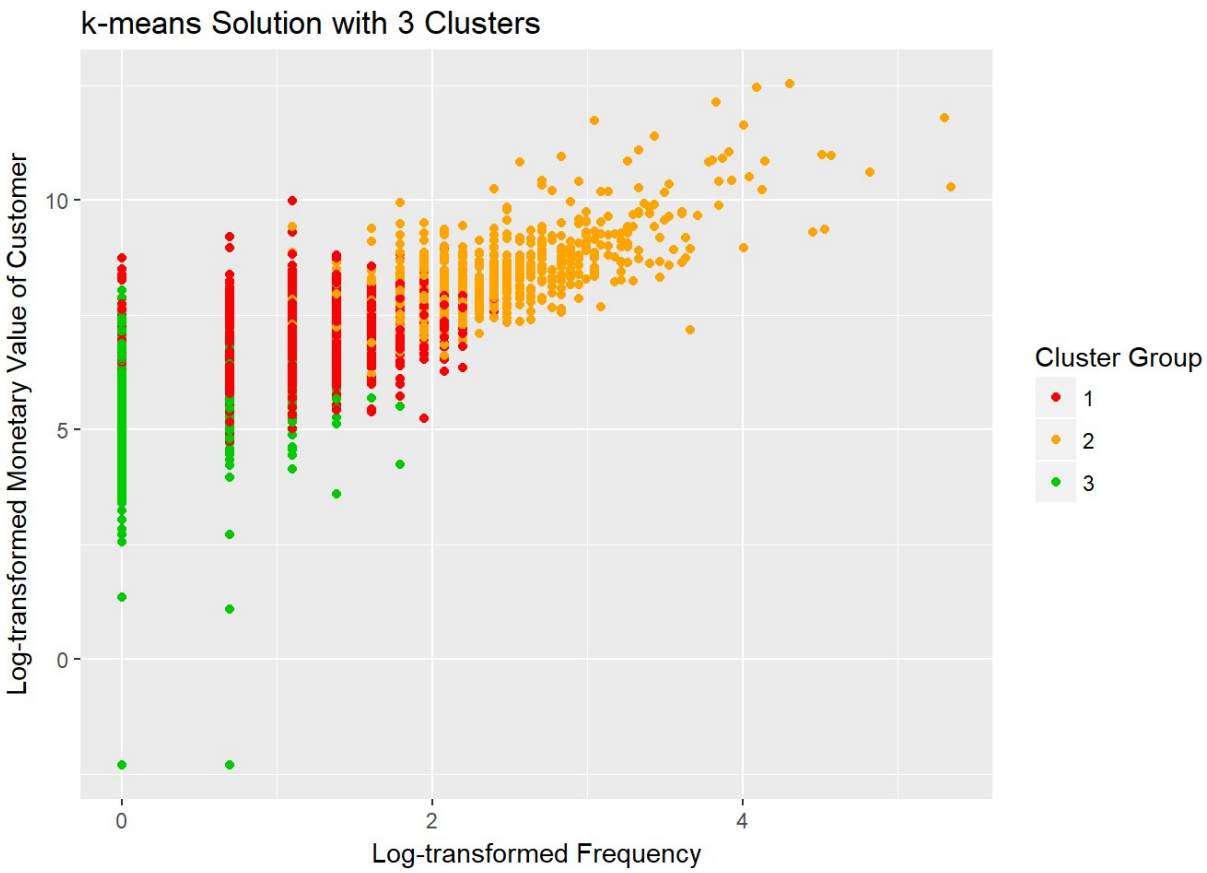
```
## [1] 1
```


k-means Solution with 1 Clusters
Log-transformed Monetary Value of Customer
10
5
0
0
2
4
Log-transformed Frequency
Cluster Group
1

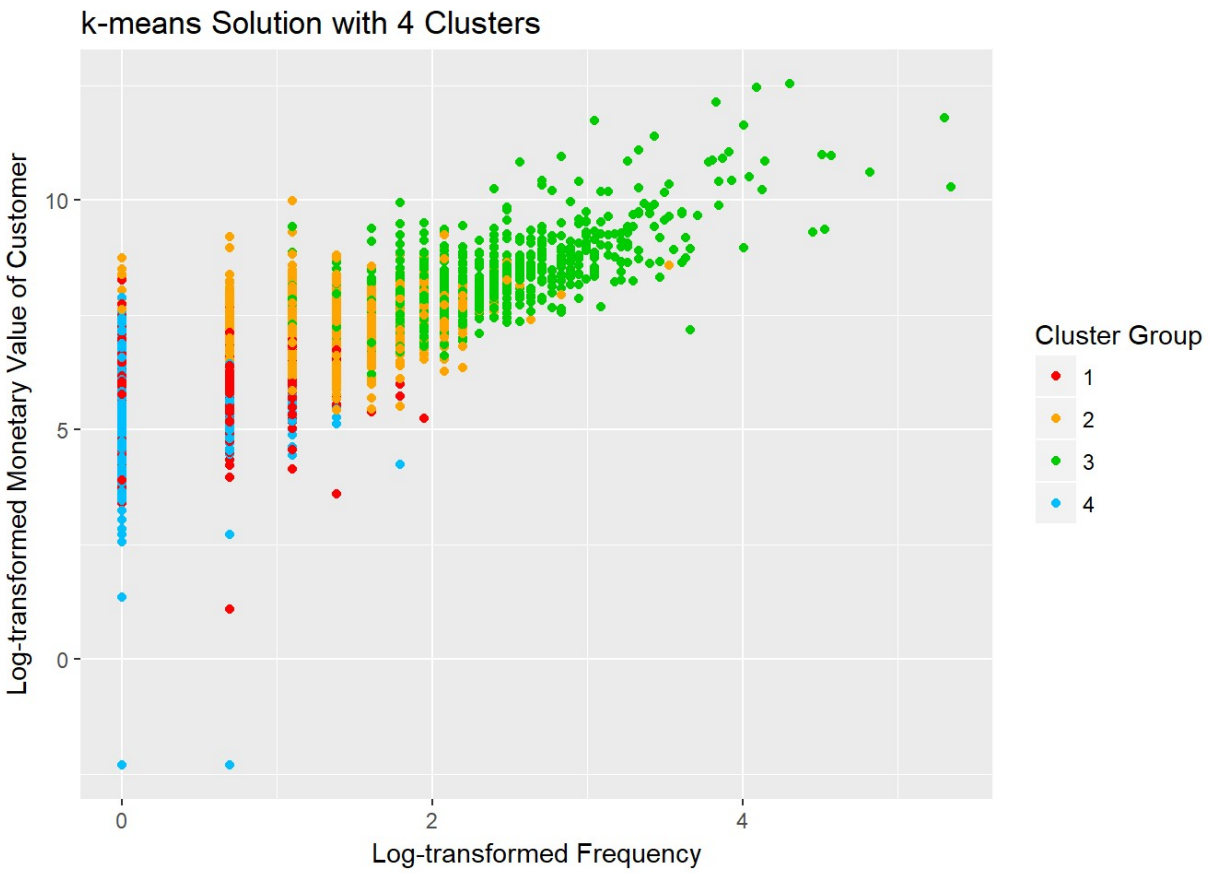
```
## [1] "k-means Solution with 1 Clusters"
##   Cluster monetary frequency recency
## 1       1   654.92         2      51
##
##
## [1] 2
```



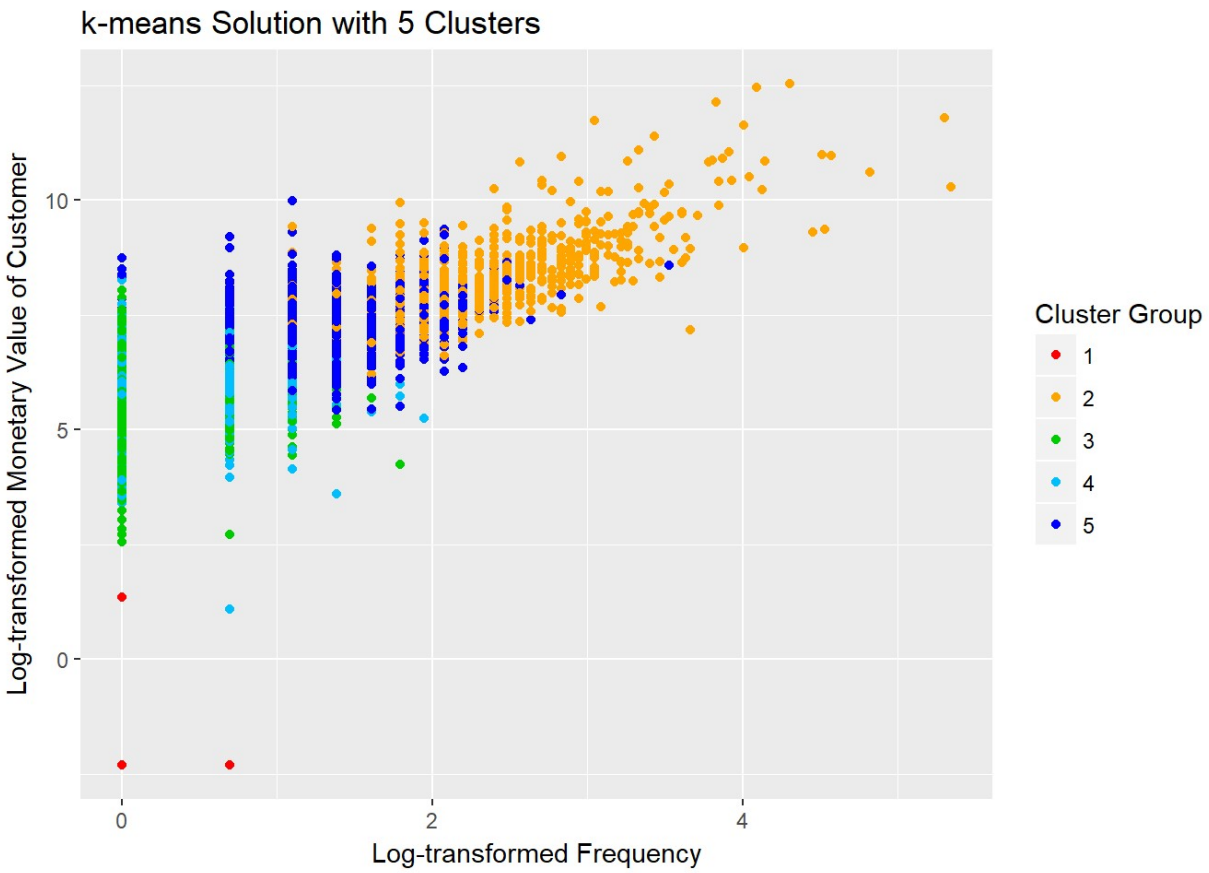

```
## [1] "k-means Solution with 2 Clusters"
##   Cluster monetary frequency recency
## 1       1  1898.52         5      18
## 2       2   344.24         1     106
##
##
## [1] 3
```



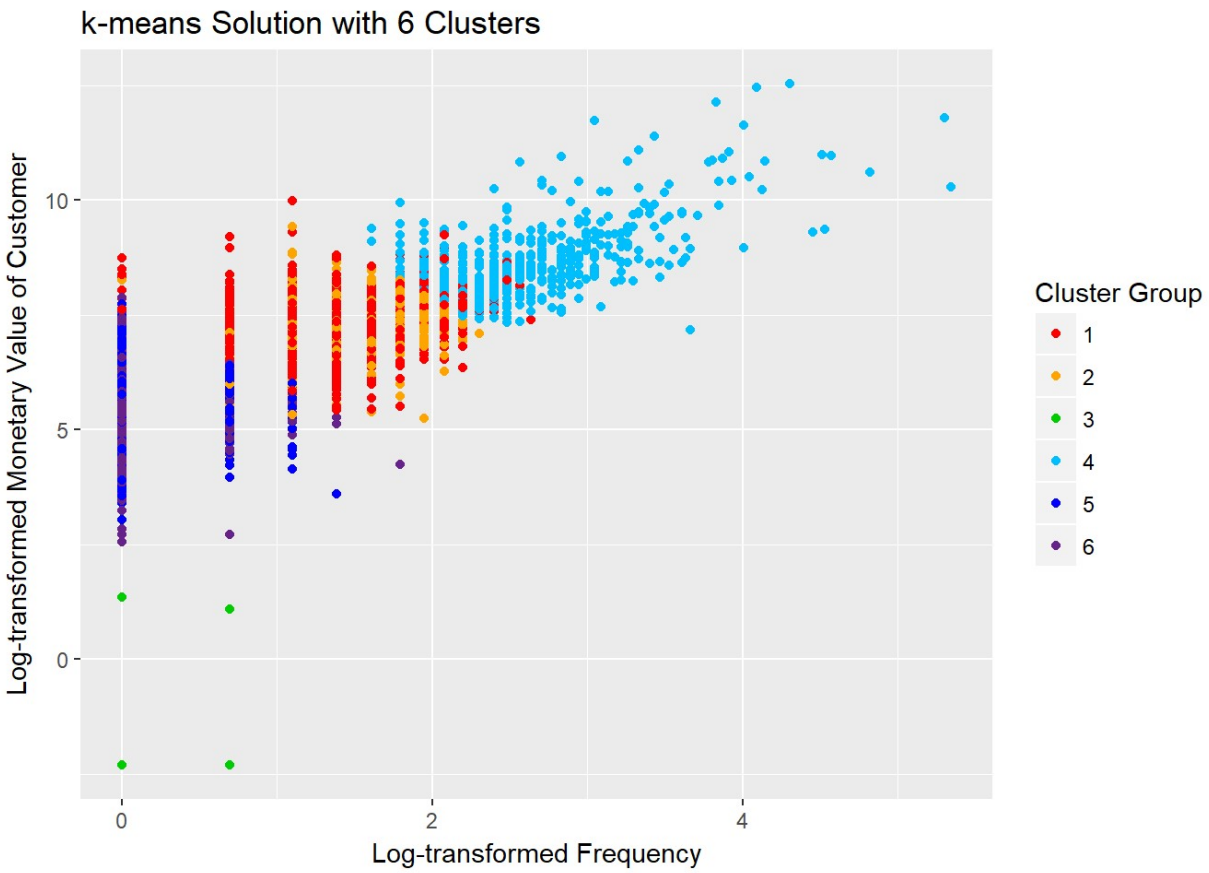

```
## [1] "k-means Solution with 3 Clusters"
##   Cluster monetary frequency recency
## 1       1   932.91         3      37
## 2       2  3355.60         9       9
## 3       3   267.18         1     157
##
##
## [1] 4
```



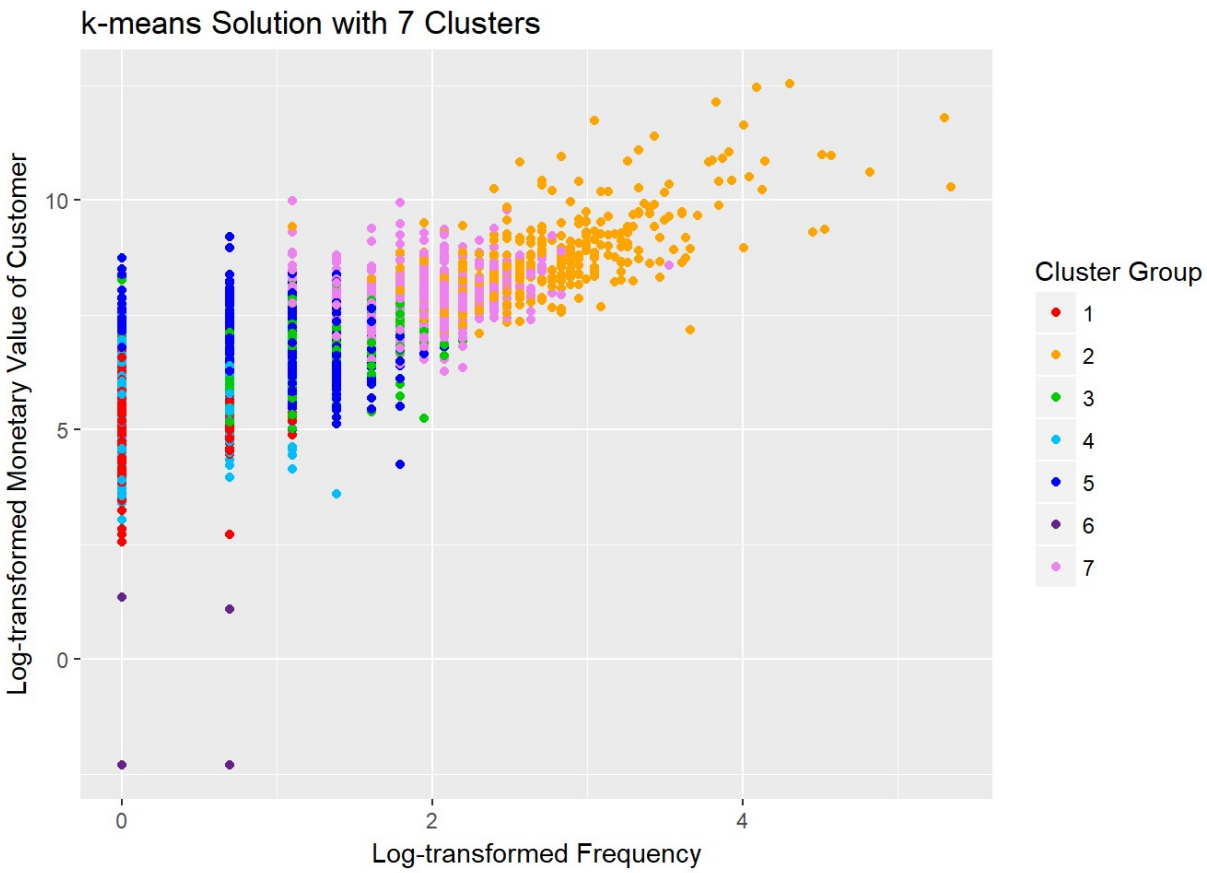

```
## [1] "k-means Solution with 4 Clusters"
##   Cluster monetary frequency recency
## 1       1   400.64         2      20
## 2       2  1177.22         4      59
## 3       3  3191.04         9       9
## 4       4   266.40         1     188
##
##
## [1] 5
```




```
## [1] "k-means Solution with 5 Clusters"
##   Cluster monetary frequency recency
## 1       1     0.00         1      86
## 2       2  3320.22         9       8
## 3       3   289.96         1     187
## 4       4   389.45         2      20
## 5       5  1246.05         4      54
##
##
## [1] 6
```



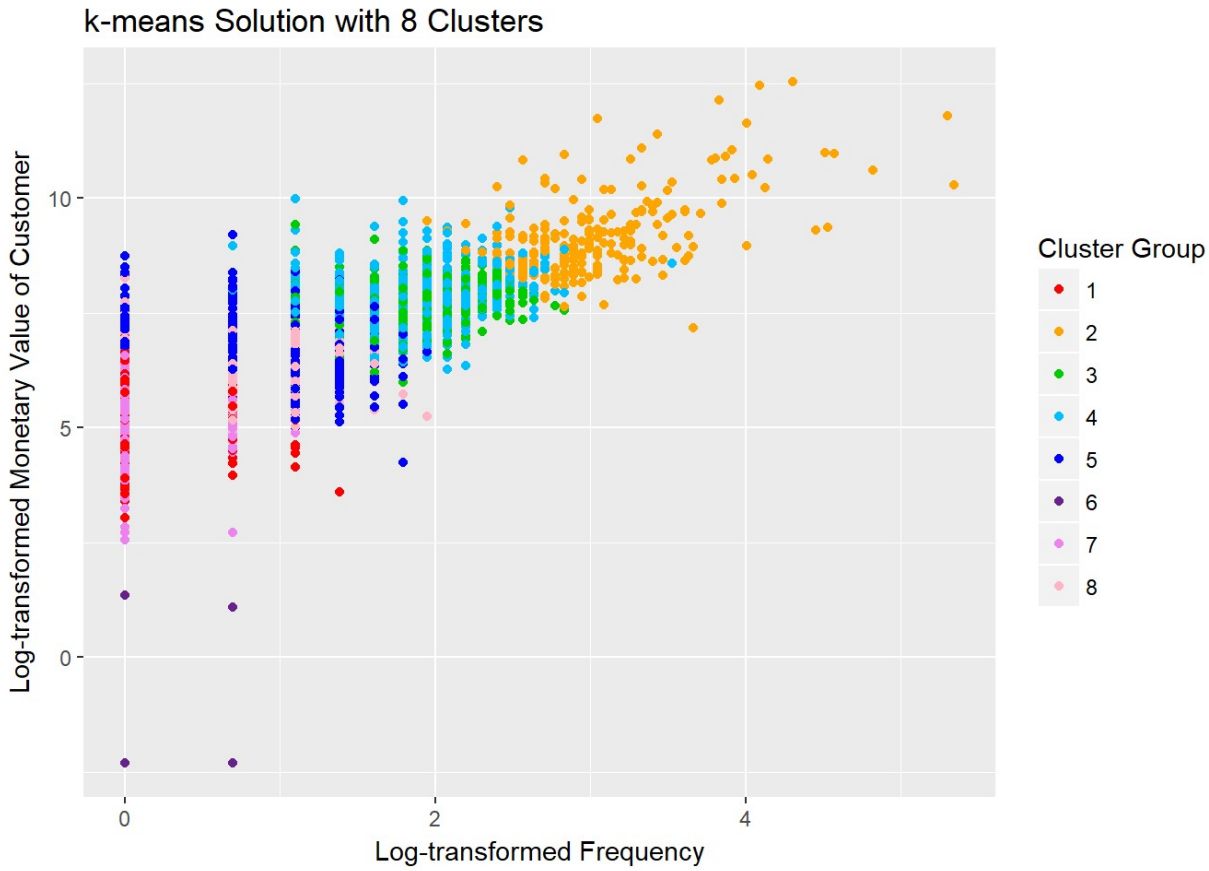

```
## [1] "k-means Solution with 6 Clusters"
##   Cluster monetary frequency recency
## 1       1  1142.38         3      68
## 2       2  1221.46         4      11
## 3       3     0.00         1      86
## 4       4  4734.26        12       9
## 5       5   318.00         1      37
## 6       6   264.70         1     233
##
##
## [1] 7
```

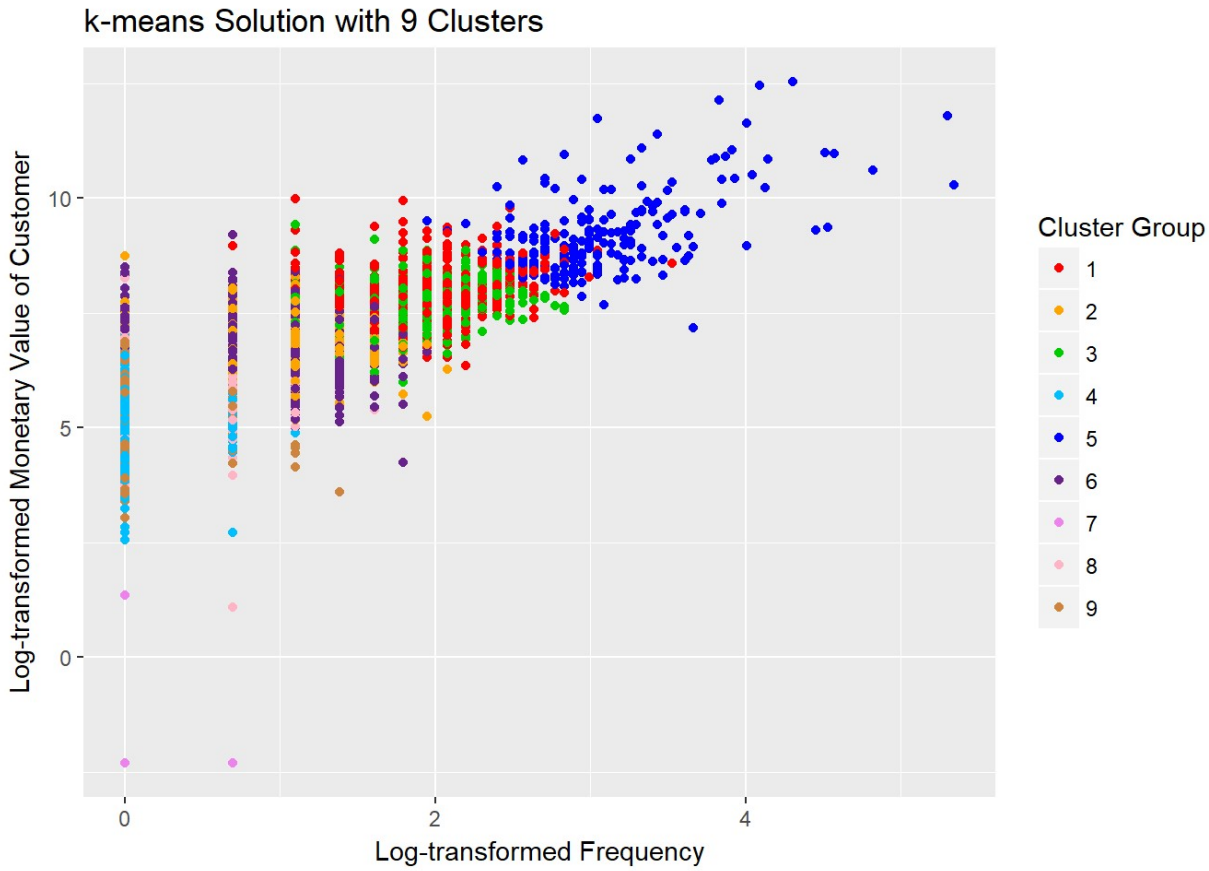



```
## [1] "k-means Solution with 7 Clusters"
##   Cluster monetary frequency recency
## 1       1   230.40         1     243
## 2       2  5126.25        14       5
## 3       3   848.55         3      10
## 4       4   281.62         1      40
## 5       5   808.54         3      90
## 6       6     0.00         1      86
## 7       7  2301.71         6      30
##
##
## [1] 8
```

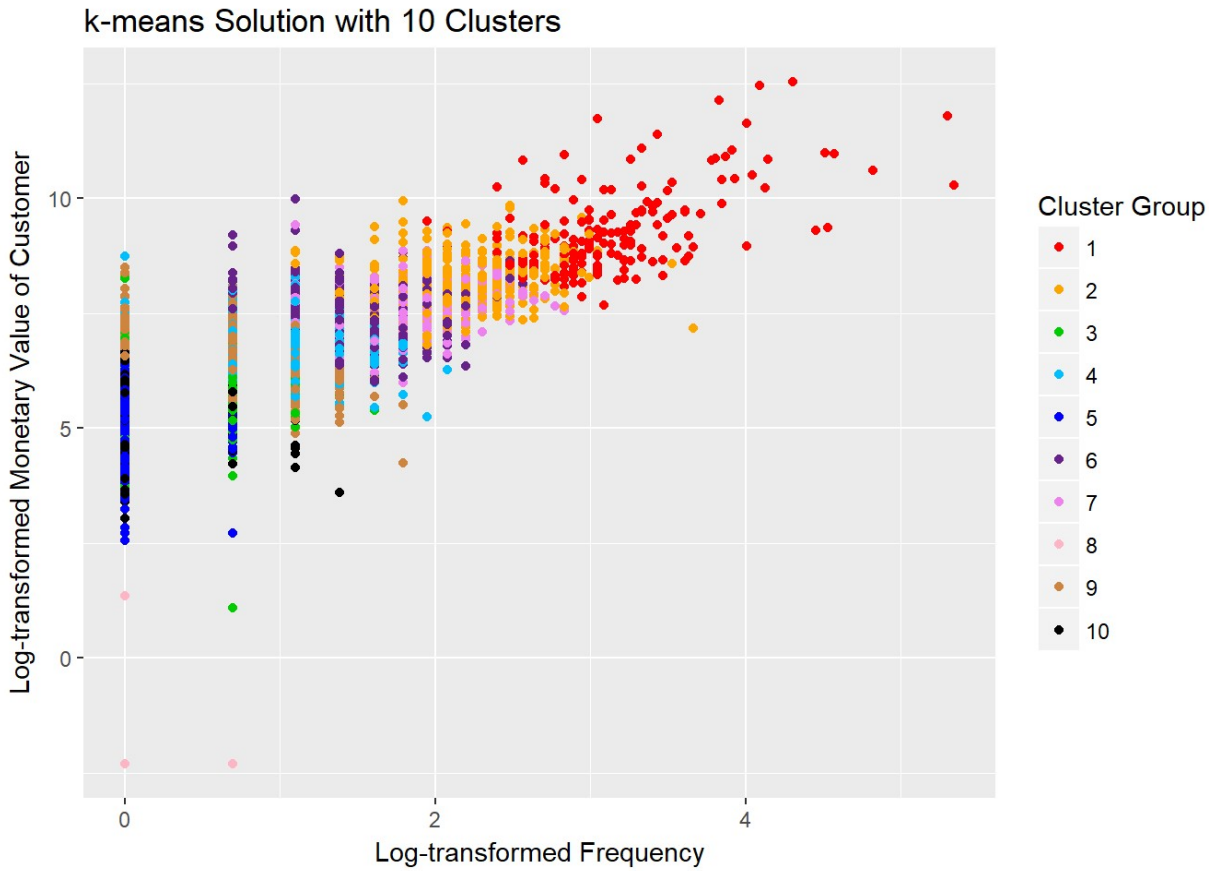
```
## [1] "k-means Solution with 8 Clusters"
##   Cluster monetary frequency recency
## 1       1   259.76         1      47
## 2       2  7490.17        19       5
## 3       3  1990.12         6       5
## 4       4  2186.20         6      33
## 5       5   770.53         2     107
## 6       6     0.00         1      86
## 7       7   225.15         1     247
## 8       8   639.02         2      18
##
##
## [1] 9
```

```
## Warning: did not converge in 10 iterations
```

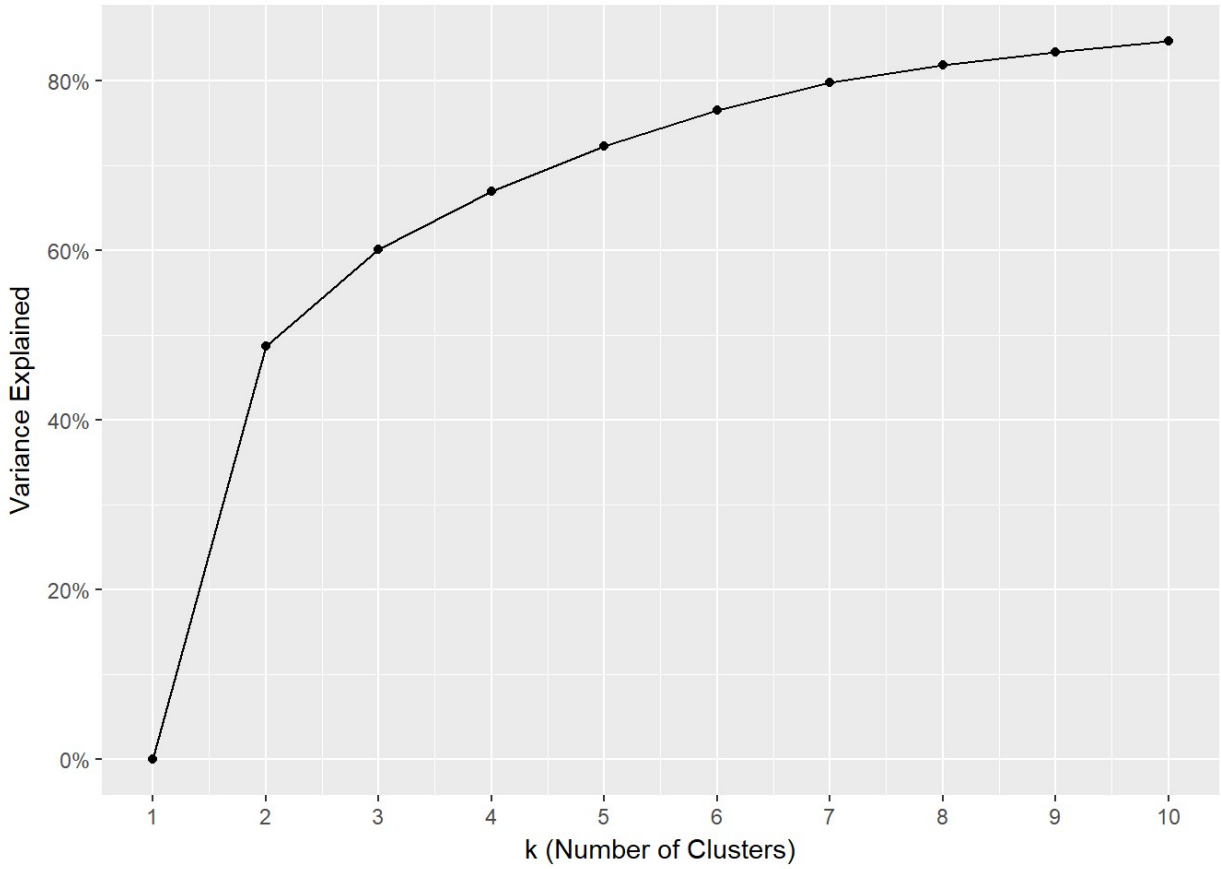


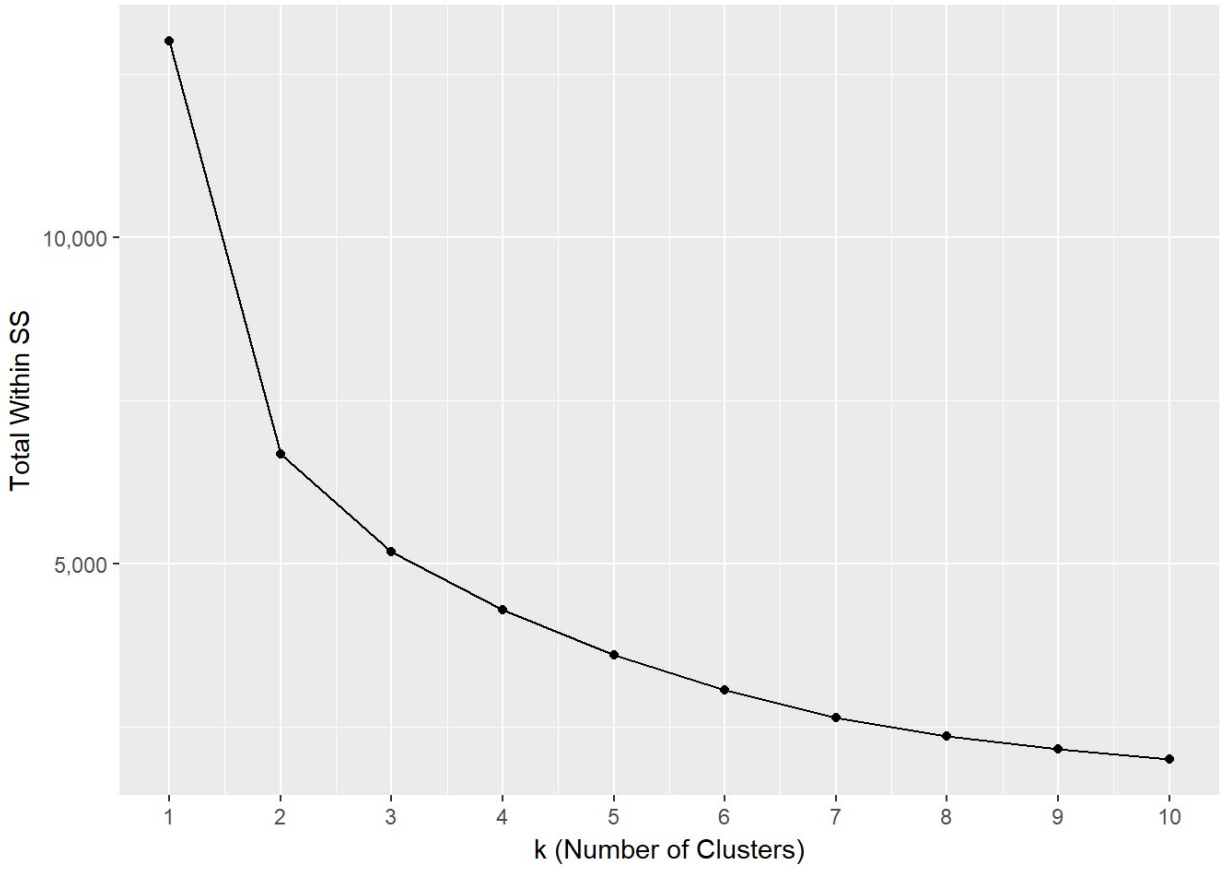


```
## [1] "k-means Solution with 9 Clusters"
##   Cluster monetary frequency recency
## 1       1  2595.72         7      36
## 2       2   872.40         3      26
## 3       3  2059.20         7       6
## 4       4   218.52         1     248
## 5       5  8108.99        19       5
## 6       6   766.20         2     123
## 7       7     0.00         1      86
## 8       8   384.81         2       8
## 9       9   285.62         1      51
##
##
## [1] 10
```



```
## [1] "k-means Solution with 10 Clusters"
##    Cluster monetary frequency recency
## 1        1  8587.42        21       4
## 2        2  3001.52         8      20
## 3        3   363.17         2       9
## 4        4   827.39         3      25
## 5        5   205.59         1     250
## 6        6  1641.72         4      74
## 7        7  1826.21         6       4
## 8        8     0.00         1      86
## 9        9   644.97         2     134
## 10      10   256.40         1      48
```




Total Within SS
10,000
5,000
1
2
3
4
5
6
7
8
9
10
k (Number of Clusters)