High value customers identification for an E-Commerce company

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 identificationof segments. Use the clustering methodology to segment customers into groups:

Domain: E-commerce

Dataset Description:

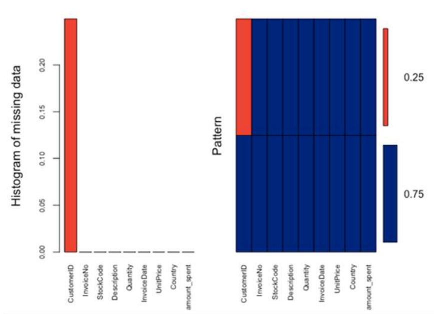
This is a transnational dataset that contains all the transactions occurring between Nov-2016 to Dec-2017 for a UK-based online retail store.

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)

```
# Installing necessary packages
installpackages("dplyr",dependencies = T)
install.packages("ggplot2", dependencies = T)
install.packages("scales", dependencies = T)
install.packages("NbClust", dependencies = T)

# reading installed packages and atttaching it to project
library(dplyr)
library(ggplot2)
library(NbClust)
library(scales)
```

High value customers identification for an E-Commerce company

```
# reading data from file
ecom_data <- read.csv("Ecommerce.csv",header = T)

# data exploration
class(ecom_data) # class of data
View(ecom_data)
str(ecom_data) # type of data stored by each columns
summary(ecom_data) # mean,median,min,max etc
head(ecom_data) # first 6 data
dim(ecom_data) # 541909  9

# Removing redundant column X
ecom_data_subset <- subset(ecom_data, select = -X)
View(ecom_data_subset)

# Checking for missing values
length(unique(ecom_data_subset$CustomerID)) # 4373
sum(is.na(ecom_data_subset$CustomerID)) # 135080

mean(is.na(ecom_data_subset)) # 0.02769633 only 2.76 % data are having missing values so
we can ignore it since its less than 5 %


# InvoiceNo    StockCode  Description    Quantity  InvoiceDate    UnitPrice   CustomerID
# 0.00000      0.00000     0.00000      0.00000      0.00000      0.00000      24.92669
# Country  amount_spent
# 0.00000      0.00000

# From the output we can see that CustomerID has around 25% of missing values which is
way too high.

# In order to visualize missing values
install.packages("VIM",dependencies = T)
library(VIM)

aggr_plot <- aggr(ecom_data_subset, col=c('navyblue','red'), numbers=TRUE,
sortVars=TRUE, labels=names(ecom_data_subset), cex.axis=.7, gap=3,
              ylab=c("Histogram of missing data","Pattern"))
```



High value customers identification for an E-Commerce company

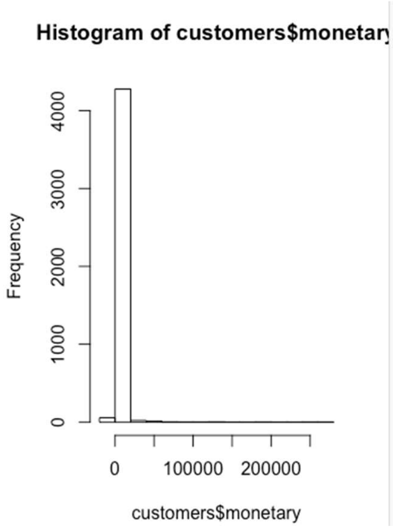
```
# Let's see the number of unique invoices and unique customers.
length(unique(ecom_data_subset$InvoiceNo))
length(unique(ecom_data_subset$CustomerID))
# We now have a dataset of 23,494 unique invoices and 3,951 unique customers.

# Remove Quantity with negative values
pos_quant <- ecom_data_subset[ecom_data_subset$Quantity > 0,]
nrow(pos_quant) # 5,31,285

# changing date format
ecom_data_subset$InvoiceDate <- as.Date(ecom_data_subset$InvoiceDate,
format = "%d-%b-%y") #23-Nov-17
str(ecom_data_subset$InvoiceDate) # Now data class is changed from factor to Date

ecom_data_subset$InvoiceNo <- as.integer(ecom_data_subset$InvoiceNo)
# Changing Invoice No from factor to int

# Customer clusters vary geographically
# So here we'll restrict the data to one geographical unit.
table(ecom_data_subset$Country)

# Let's see the number of unique invoices and unique customers.
length(unique(ecom_data_subset$InvoiceNo))
length(unique(ecom_data_subset$CustomerID))
# We now have a dataset of 25,900 unique invoices and 4,373 unique customers.

# We will calculate recency which is the no of days elapsed since customer last order
# and frequency which will refer to the no of invoices with purchase during the year
# It is necessary to distinguish invoices with purchases from invoices with returns.
ecom_data_subset$item.return <- grepl("C", ecom_data_subset$InvoiceNo, fixed=TRUE)

ecom_data_subset$purchase.invoice <- ifelse(ecom_data_subset$item.return == "TRUE",
0,1) # Identify returns

# Creating Customer Level Dataset

customers <- as.data.frame(unique(ecom_data_subset$CustomerID))
names(customers) <- "CustomerID"



# Adding a recency column by substracting the InvoiceDate from the (last InvoiceDate+1)
ecom_data_subset$recency <- as.Date("2017-12-08") -
as.Date(ecom_data_subset$InvoiceDate)
```

High value customers identification for an E-Commerce company

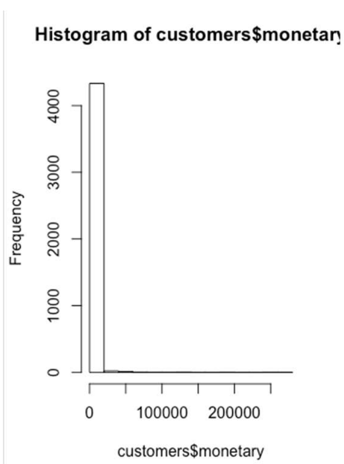
```
# remove returns so only consider the data of most recent "purchase"
temp <- subset(ecom_data_subset, purchase.invoice == 1)

# Obtain no of days since most recent purchase
recency <- aggregate(recency ~ CustomerID, data=temp, FUN=min, na.rm=TRUE)
remove(temp)

# Add recency to customer data
customers <- merge(customers, recency, by="CustomerID", all=TRUE, sort=TRUE)
remove(recency)

customers$recency <- as.numeric(customers$recency)

### frequency ##########
customer.invoices <- subset(ecom_data_subset, select = c("CustomerID","InvoiceNo",
"purchase.invoice"))
customer.invoices <- customer.invoices[!duplicated(customer.invoices), ]
customer.invoices <- customer.invoices[order(customer.invoices$CustomerID),]
row.names(customer.invoices) <- NULL

# Number of invoices/year (purchases only)
annual.invoices <- aggregate(purchase.invoice ~ CustomerID, data=customer.invoices,
FUN=sum, na.rm=TRUE)
names(annual.invoices)[names(annual.invoices)=="purchase.invoice"] <- "frequency"

# Adding of invoices to customers data
customers <- merge(customers, annual.invoices, by="CustomerID", all=TRUE, sort=TRUE)
remove(customer.invoices, annual.invoices)

range(customers$frequency) # NA NA
table(customers$frequency) # displays some values

# Remove customers who have not made any purchases in the past year
customers <- subset(customers, frequency > 0)



# Calculating Monetary Value for the customers
# Add the column - amount_spent
ecom_data_subset['amount_spent'] = ecom_data_subset['Quantity'] *
ecom_data_subset['UnitPrice']

# Aggregated total sales to customer and assigning it to monetary field
total.sales <- aggregate(amount_spent ~ CustomerID, data = ecom_data_subset, FUN=sum,
na.rm=TRUE)
names(total.sales)[names(total.sales)=="amount_spent"] <- "monetary"
```

High value customers identification for an E-Commerce company

```
# Adding monetary value to customers dataset
customers <- merge(customers, total.sales, by="CustomerID", all.x=TRUE, sort=TRUE)
remove(total.sales)

# Identify customers with negative monetary value numbers, as they were presumably
# returning purchases from the preceding year
hist(customers$monetary)
```




```
customers$monetary <- ifelse(customers$monetary < 0, 0, customers$monetary)
# reset negative numbers to zero
hist(customers$monetary)
```




```
# highly valued top 20%  and bottom 80 % customers
customers <- customers[order(-customers$monetary),]
high.cutoff <- 0.8 * sum(customers$monetary)
customers$high <- ifelse(cumsum(customers$monetary) <= high.cutoff, "Top 20%", "Bottom
80%")
```

High value customers identification for an E-Commerce company

```
customers$high <- factor(customers$high, levels=c("Top 20%", "Bottom 80%"),
ordered=TRUE)
levels(customers$high) # "Top 20%"    "Bottom 80%"
round(prop.table(table(customers$high)), 2) # Top 20% Bottom 80%
                                                 0.27      0.73
remove(high.cutoff)

customers <- customers[order(customers$CustomerID),]

###################
# Preprocessing Of DataSet #
###################

# Log-transform positively-skewed variables
customers$recency.log <- log(customers$recency)
customers$frequency.log <- log(customers$frequency)
customers$monetary.log <- customers$monetary + 0.1 # can't take log(0), so add a small
value to remove zeros
customers$monetary.log <- log(customers$monetary.log)

# Z-scores to scale each of the data
customers$recency.z <- scale(customers$recency.log, center=TRUE, scale=TRUE)
customers$frequency.z <- scale(customers$frequency.log, center=TRUE, scale=TRUE)
customers$monetary.z <- scale(customers$monetary.log, center=TRUE, scale=TRUE)

View(customers)
```

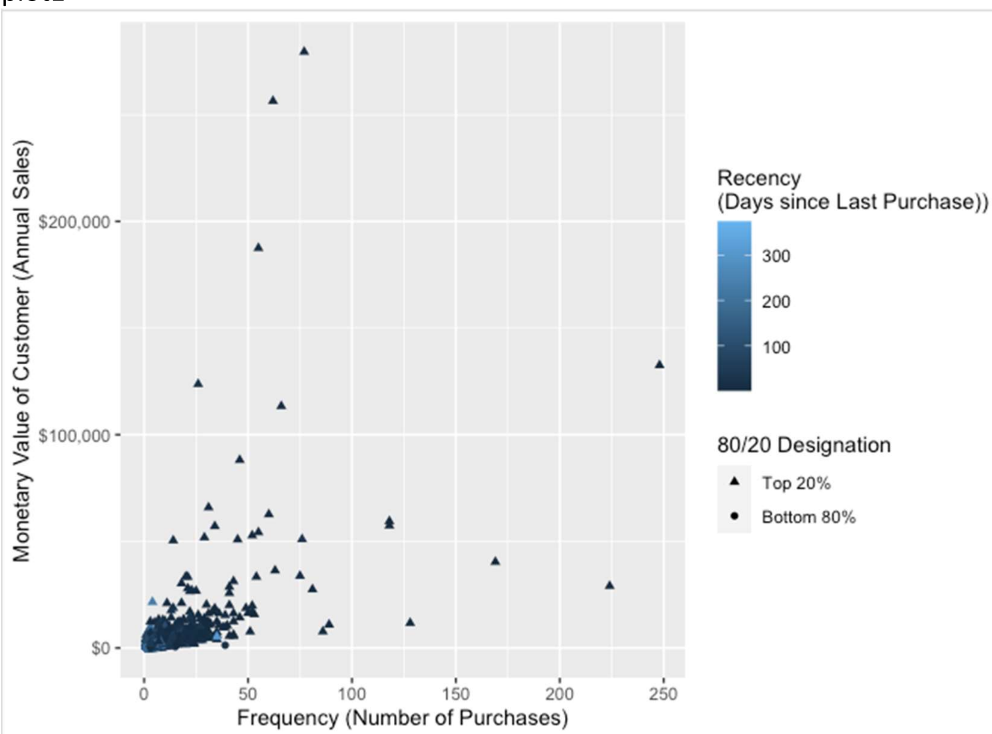
Filter

	CustomerID	recency	frequency	monetary	pareto	high	recency.log	frequency.log
1	12346	326	2	0.00	Bottom 80%	Bottom 80%	5.7868974	0.6931472
2	12347	3	7	4310.00	Top 20%	Top 20%	1.0986123	1.9459101
3	12348	76	4	1797.24	Top 20%	Top 20%	4.3307333	1.3862944
4	12349	19	1	1757.55	Top 20%	Top 20%	2.9444390	0.0000000
5	12350	311	1	334.40	Bottom 80%	Bottom 80%	5.7397929	0.0000000
6	12352	37	11	1545.41	Top 20%	Top 20%	3.6109179	2.3978953
7	12353	205	1	89.00	Bottom 80%	Bottom 80%	5.3230100	0.0000000
8	12354	233	1	1079.40	Bottom 80%	Bottom 80%	5.4510385	0.0000000
9	12355	215	1	459.40	Bottom 80%	Bottom 80%	5.3706380	0.0000000
10	12356	23	3	2811.43	Top 20%	Top 20%	3.1354942	1.0986123
11	12357	34	1	6207.67	Top 20%	Top 20%	3.5263605	0.0000000
12	12358	2	2	1168.06	Bottom 80%	Bottom 80%	0.6931472	0.6931472
13	12359	8	6	6245.53	Top 20%	Top 20%	2.0794415	1.7917595
14	12360	53	3	2662.06	Top 20%	Top 20%	3.9702919	1.0986123
15	12361	288	1	189.90	Bottom 80%	Bottom 80%	5.6629605	0.0000000

Showing 1 to 16 of 4,372 entries, 12 total columns

High value customers identification for an E-Commerce company

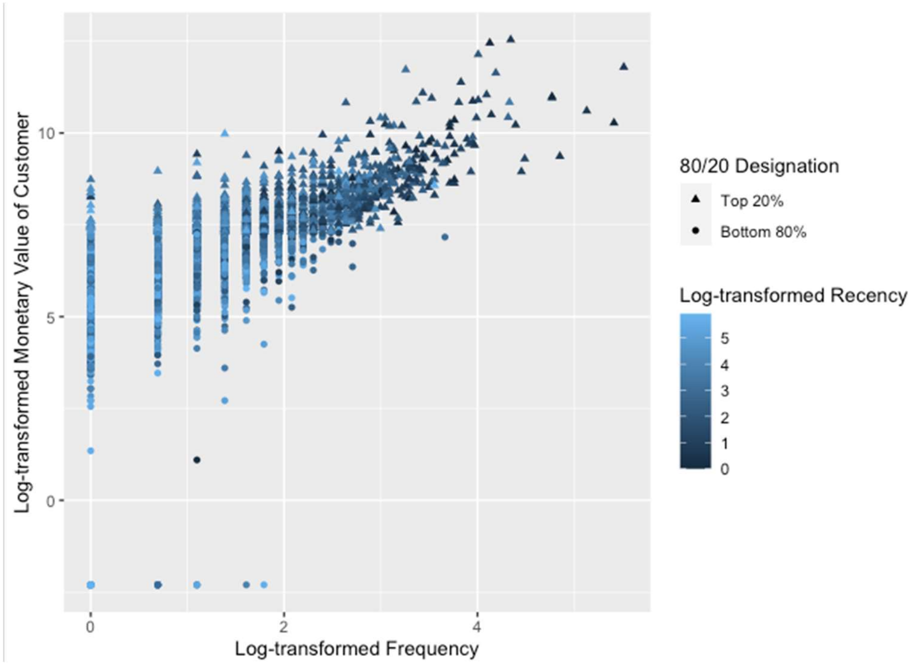
```
### Data Visualization ##########
library(ggplot2)
library(scales)

# Original scale
plot1 <- ggplot(customers, aes(x = frequency, y = monetary))
plot1 <- plot1 + geom_point(aes(colour = recency, shape = pareto))
plot1 <- plot1 + scale_shape_manual(name = "80/20 Designation", values=c(17, 16))
plot1 <- plot1 + scale_colour_gradient(name="Recency\n(Days since Last Purchase))")
plot1 <- plot1 + scale_y_continuous(label=dollar)
plot1 <- plot1 + xlab("Frequency (Number of Purchases)")
plot1 <- plot1 + ylab("Monetary Value of Customer (Annual Sales)")
plot1
```




```
# This first graph uses the variables' original metrics and is almost completely
uninterpretable.There's a clump of data points in the lower left-hand corner of the plot, and
then a few outliers.This is why we log-transformed the input variables.

# Log-transformed
plot2 <- ggplot(customers, aes(x = frequency.log, y = monetary.log))
plot2 <- plot2+ geom_point(aes(colour = recency.log, shape = pareto))
plot2<- plot2+ scale_shape_manual(name = "80/20 Designation", values=c(17, 16))
plot2<- plot2+ scale_colour_gradient(name="Log-transformed Recency")
plot2<- plot2+ xlab("Log-transformed Frequency")
plot2<- plot2+ ylab("Log-transformed Monetary Value of Customer")
```

High value customers identification for an E-Commerce company

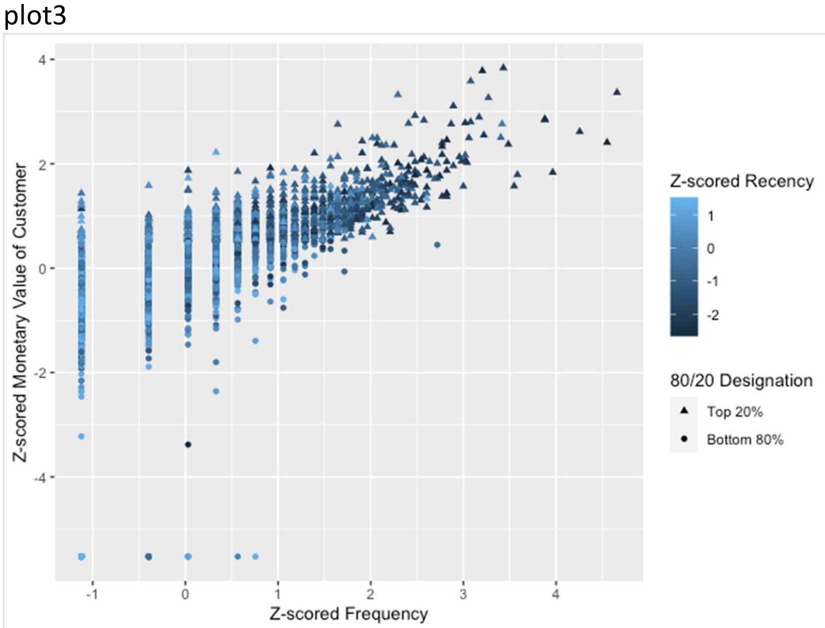
plot2



Scaled variables

```
plot3 <- ggplot(customers, aes(x = frequency.z, y = monetary.z))
plot3 <- plot3 + geom_point(aes(colour = recency.z, shape = pareto))
plot3 <- plot3 + scale_shape_manual(name = "80/20 Designation", values=c(17, 16))
plot3 <- plot3 + scale_colour_gradient(name="Z-scored Recency")
plot3 <- plot3 + xlab("Z-scored Frequency")
plot3 <- plot3 + ylab("Z-scored Monetary Value of Customer")
```

plot3



High value customers identification for an E-Commerce company

Analysis tasks to be performed:
Use the clustering methodology to segment customers into groups:
Use the following clustering algorithms:

K means
Hierarchical

- Identify the right number of customer segments.
- Provide the number of customers who are highly valued.
- Identify the clustering algorithm that gives maximum accuracy and explains robust clusters.
- If the number of observations is loaded in one of the clusters, break down that cluster further using the clustering algorithm.

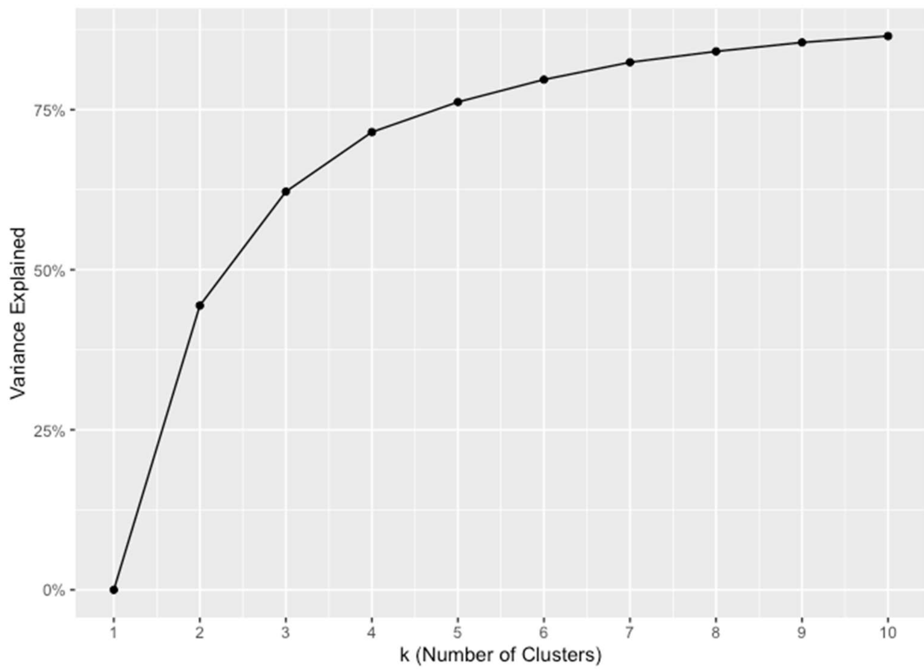
[hint: Here loaded means if any cluster has more number of data points as compared to other clusters then split that clusters by increasing the number of clusters and observe, compare the results with previous results.]

```
########## Determining number of clusters through K-Means ####################
preprocessed <- customers[,9:11]
clustmax <-  10 # specify the maximum number of clusters you want to try out

models <- data.frame(k=integer(),
                tot.withinss=numeric(),
                betweenss=numeric(),
                totss=numeric(),
                rsquared=numeric())

for (k in 1: clustmax)
 {

 print(k)

 # Run kmeans
 # nstart = number of initial configurations; the best one is used
 # $iter will return the iteration used for the final model
 output <- kmeans(preprocessed, centers = k, nstart = 20)

 # Add cluster membership to customers dataset
 var.name <- paste("cluster", k, sep="_")
 customers[,(var.name)] <- output$cluster
 customers[,(var.name)] <- factor(customers[,(var.name)], levels = c(1:k))

 # Graph clusters
 cluster_graph <- ggplot(customers, aes(x = frequency.log, y = monetary.log))
```

```
  cluster_graph <- cluster_graph + geom_point(aes(colour = customers[,(var.name)]))
  colors <-
c('red','orange','green3','deepskyblue','blue','darkorchid4','violet','pink1','tan3','black')
  cluster_graph <- cluster_graph + scale_colour_manual(name = "Cluster Group",
values=colors)
  cluster_graph <- cluster_graph + xlab("Log-transformed Frequency")
  cluster_graph <- cluster_graph + ylab("Log-transformed Monetary Value of Customer")
  title <- paste("k-means Solution with", k, sep=" ")
  title <- paste(title, "Clusters", sep=" ")
  cluster_graph <- cluster_graph + ggtitle(title)
  print(cluster_graph)

  # Cluster centers in original metrics
  library(plyr)
  print(title)
  cluster_centers <- ddply(customers, .(customers[,(var.name)]), summarize,
                           monetary=round(median(monetary),2),# use median b/c this is the
          raw,  heavily-skewed data
                  frequency=round(median(frequency),1),
                  recency=round(median(recency), 0))
  names(cluster_centers)[names(cluster_centers)=="customers[, (var.name)]"] <- "Cluster"
  print(cluster_centers)
  cat("\n")
  cat("\n")

  # Collect model information
  models[k,("k")] <- k
  models[k,("tot.withinss")] <- output$tot.withinss # the sum of all within sum of squares
  models[k,("betweenss")] <- output$betweenss
  models[k,("totss")] <- output$totss # betweenss + tot.withinss

  # percentage of variance explained by cluster membership
  models[k,("rsquared")] <- round(output$betweenss/output$totss, 3)
  assign("models", models, envir = .GlobalEnv)

}
```

High value customers identification for an E-Commerce company

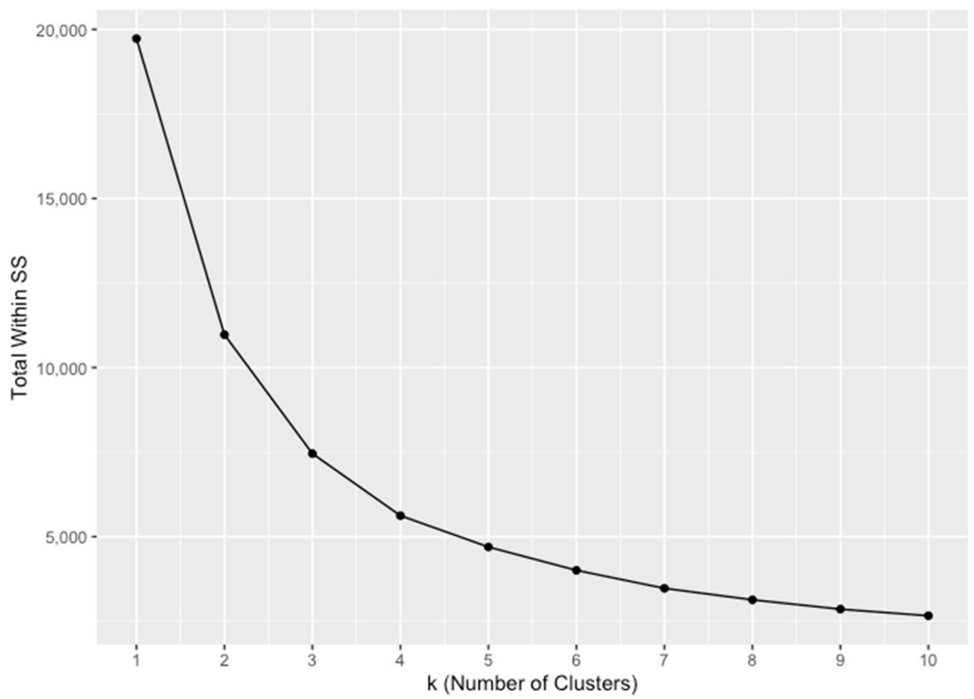
```
# Graph variance explained by number of clusters
r2_graph <- ggplot(models, aes(x = k, y = rsquared))
r2_graph <- r2_graph + geom_point() + geom_line()
r2_graph <- r2_graph + scale_y_continuous(labels = scales::percent)
r2_graph <- r2_graph + scale_x_continuous(breaks = 1:clustmax)
r2_graph <- r2_graph + xlab("k (Number of Clusters)")
r2_graph <- r2_graph + ylab("Variance Explained")
r2_graph
```

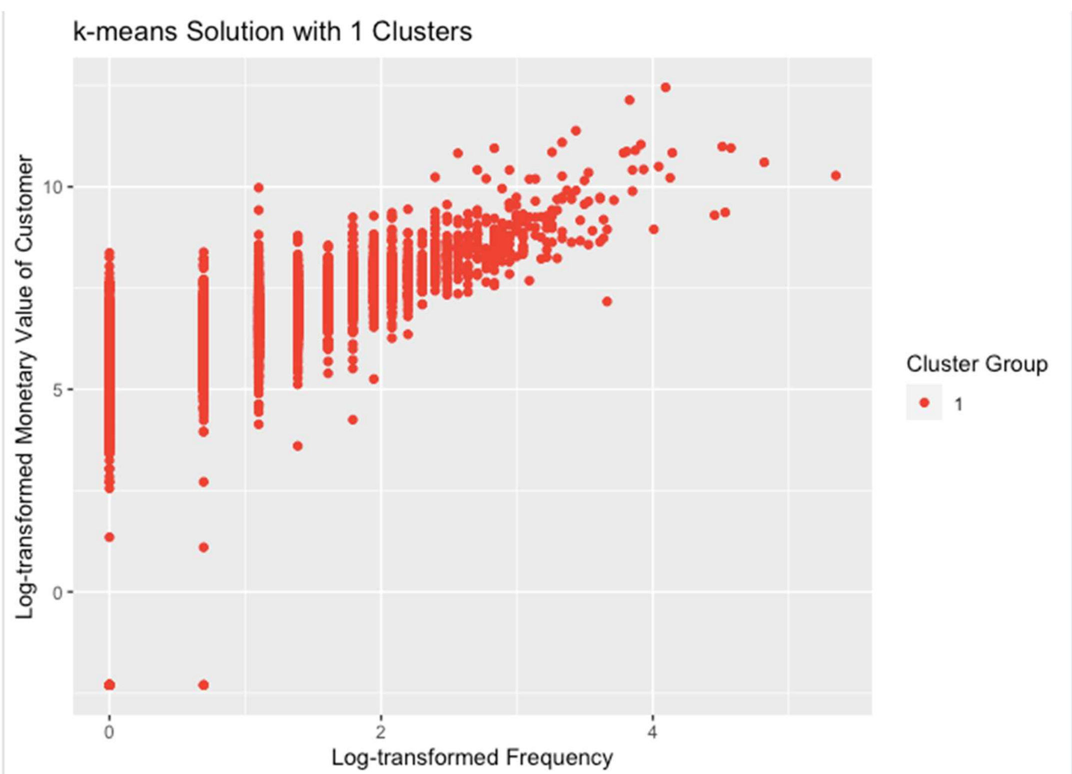


From the graphs we can see that a 2 cluster solution explains only 48% of variance and that cannot be taken into consideration with business strategy targeted for customers. Also 5 cluster solutions provides 76% variance around which is good but there is no break at that point as evident from the graph

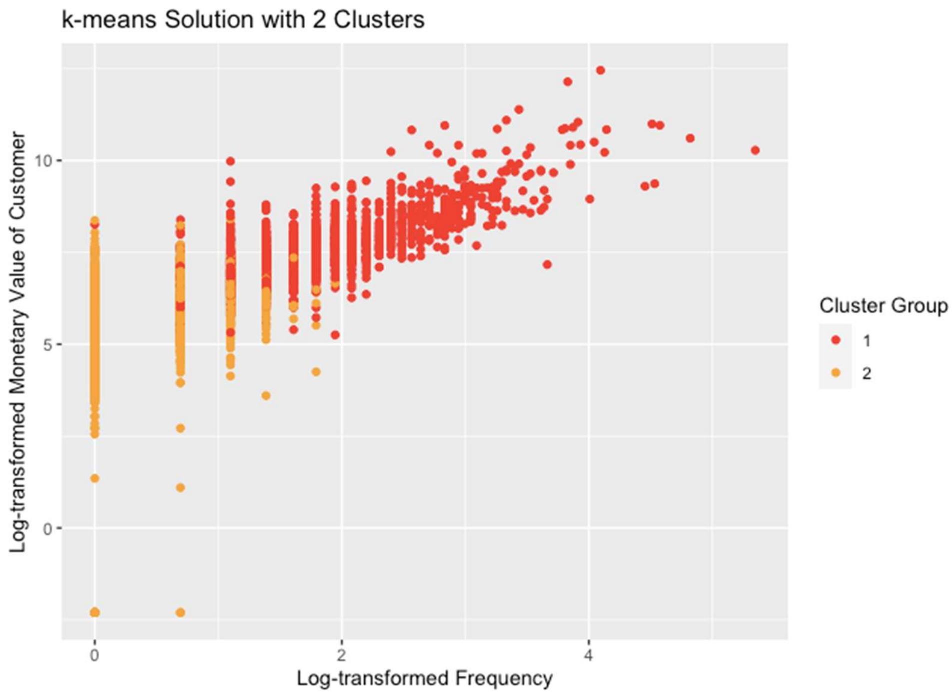
```
# Graph within sums of squares by number of clusters
# Look for a "bend" in the graph, as with a scree plot
ss_graph <- ggplot(models, aes(x = k, y = tot.withinss))
ss_graph <- ss_graph + geom_point() + geom_line()
ss_graph <- ss_graph + scale_x_continuous(breaks = 1:clustmax)
ss_graph <- ss_graph + scale_y_continuous(labels = scales::comma)
ss_graph <- ss_graph + xlab("k (Number of Clusters)")
ss_graph <- ss_graph + ylab("Total Within SS")
ss_graph
```

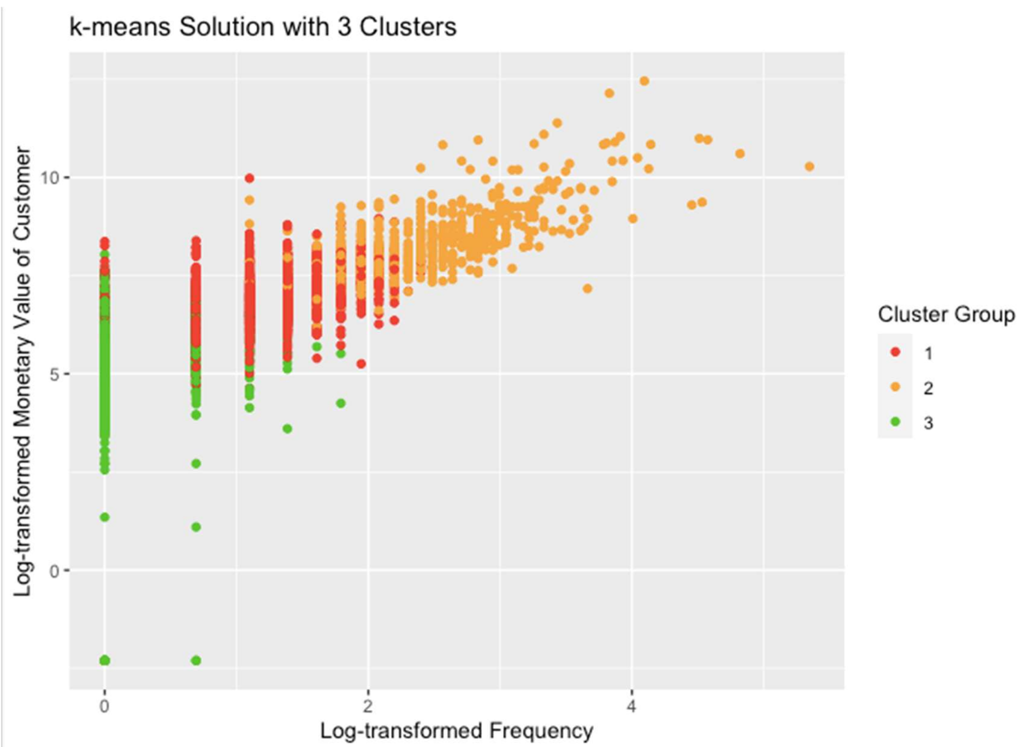
High value customers identification for an E-Commerce company



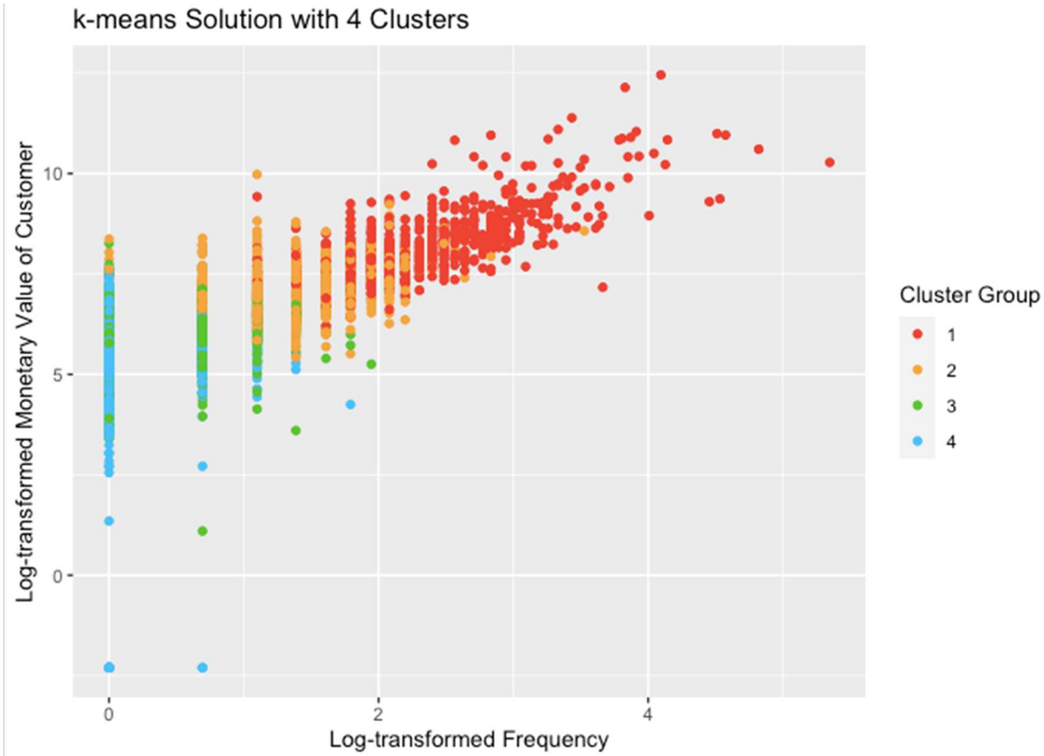


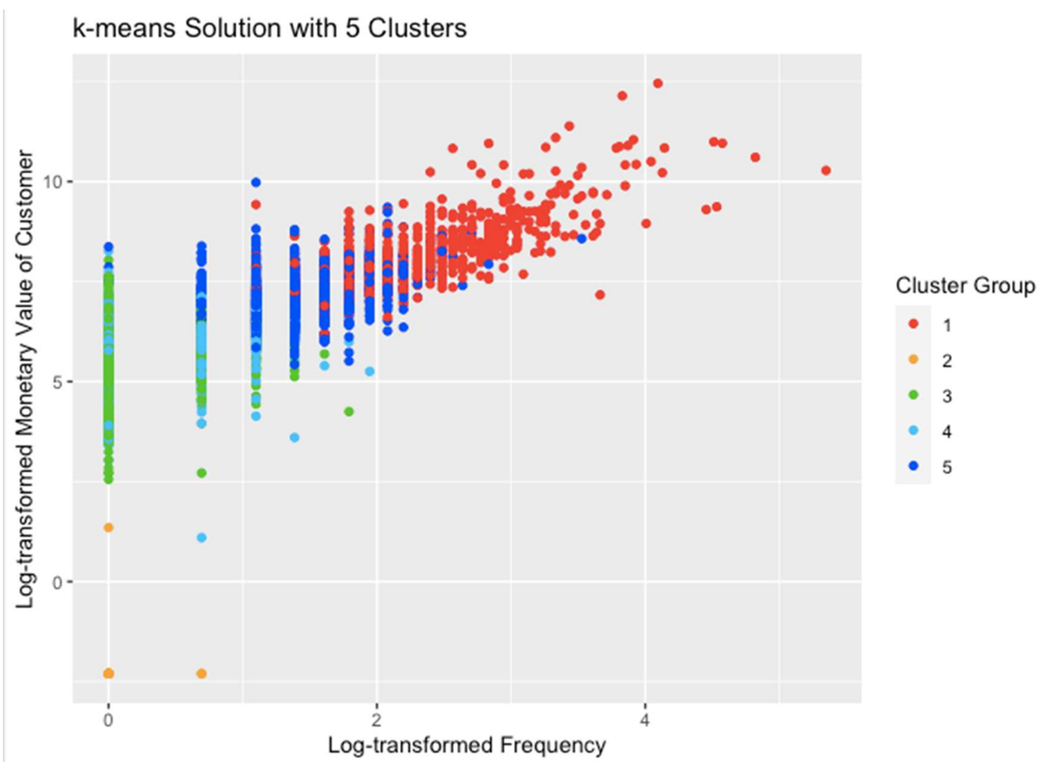
High value customers identification for an E-Commerce company





High value customers identification for an E-Commerce company





High value customers identification for an E-Commerce company

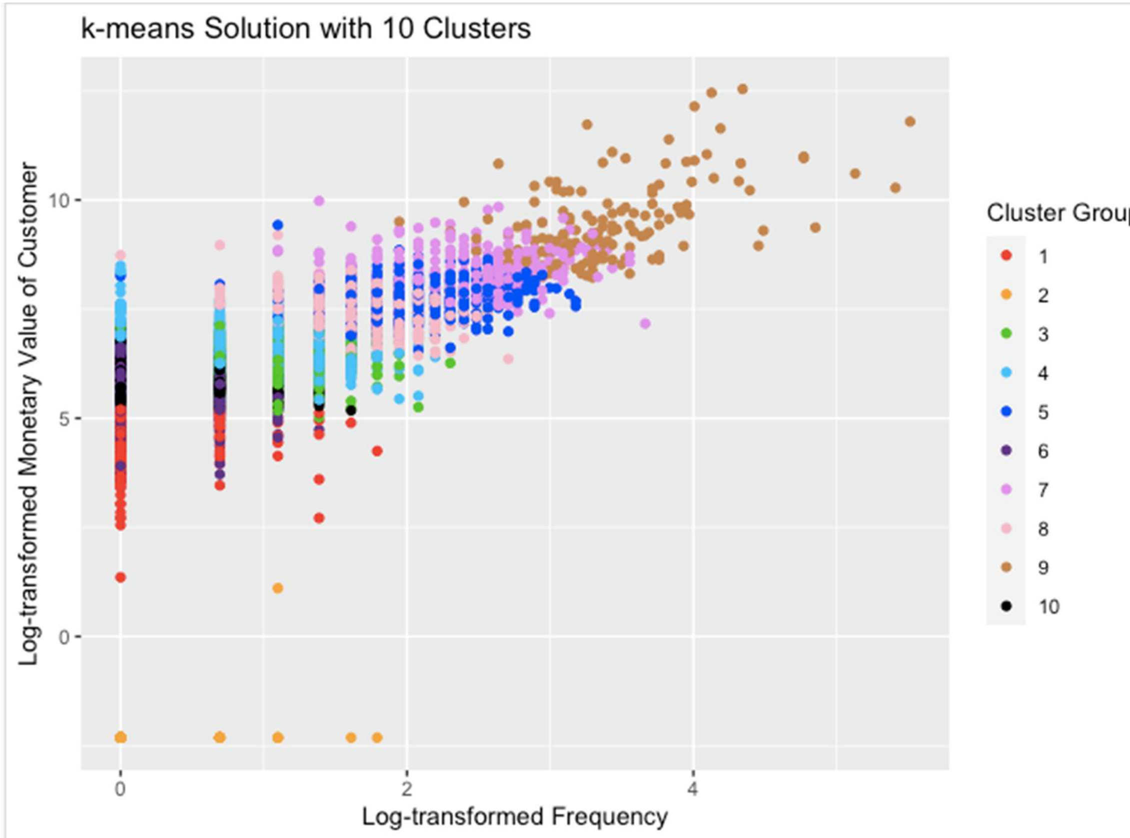
```
[1] 1
[1] "k-means Solution with 1 Clusters"
  Cluster monetary frequency recency
1       1   648.08         3      51


[1] 2
[1] "k-means Solution with 2 Clusters"
  Cluster monetary frequency recency
1       1  1803.86         6      18
2       2   318.89         1     107


[1] 3
[1] "k-means Solution with 3 Clusters"
  Cluster monetary frequency recency
1       1   339.52         1      96
2       2  1919.31         6      17
3       3     0.00         1     290
```

```
[1] 4
[1] "k-means Solution with 4 Clusters"
  Cluster monetary frequency recency
1       1   251.24         1     149
2       2     0.00         1     290
3       3  3315.71        11       9
4       4   908.03         3      39


[1] 5
[1] "k-means Solution with 5 Clusters"
  Cluster monetary frequency recency
1       1  1174.35         4      65
2       2     0.00         1     290
3       3  3543.26        11       9
4       4   249.93         1     165
5       5   547.06         2      13


[1] 6
[1] "k-means Solution with 6 Clusters"
  Cluster monetary frequency recency
1       1   336.42         2      25
2       2   215.83         1     213
3       3  1621.73         6      16
4       4   864.59         3      87
5       5     0.00         1     290
6       6  5664.57        17       8


[1] 7
[1] "k-means Solution with 7 Clusters"
  Cluster monetary frequency recency
1       1  1032.36         4       9
2       2  2118.22         7      40
3       3   693.74         3      90
4       4   275.56         1      30
5       5     0.00         1     290
6       6  5116.86        16       6
7       7   207.50         1     234
```

High value customers identification for an E-Commerce company

```
[1] 8
[1] "k-means Solution with 8 Clusters"
  Cluster monetary frequency recency
1       1  1837.28         6      47
2       2 10158.54        27       4
3       3   747.03         3      12
4       4   639.63         2     111
5       5   193.25         1     240
6       6  2777.41        10       9
7       7   248.61         1      36
8       8     0.00         1     290


[1] 9
[1] "k-means Solution with 9 Clusters"
  Cluster monetary frequency recency
1       1 10360.04        27       5
2       2  2385.80         7      37
3       3  2745.78        10       6
4       4   131.90         1     219
5       5   841.20         3      78
6       6     0.00         1     280
7       7   869.57         4      12
8       8   361.16         1     206
9       9   266.57         1      30


[1] 10
[1] "k-means Solution with 10 Clusters"
   Cluster monetary frequency recency
1        1   121.94         1     219
2        2     0.00         1     280
3        3   635.41         3      12
4        4   711.41         3      89
5        5  2063.16         8       5
6        6   247.64         1      37
7        7  4203.62        12      23
8        8  1605.75         5      41
9        9 10327.11        27       3
10      10   330.90         1     234
```

High value customers identification for an E-Commerce company



##
Using NbClust metrics to determine number of clusters

```
library(NbClust)
set.seed(1)
nc <- NbClust(preprocessed, min.nc=2, max.nc=7, 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.

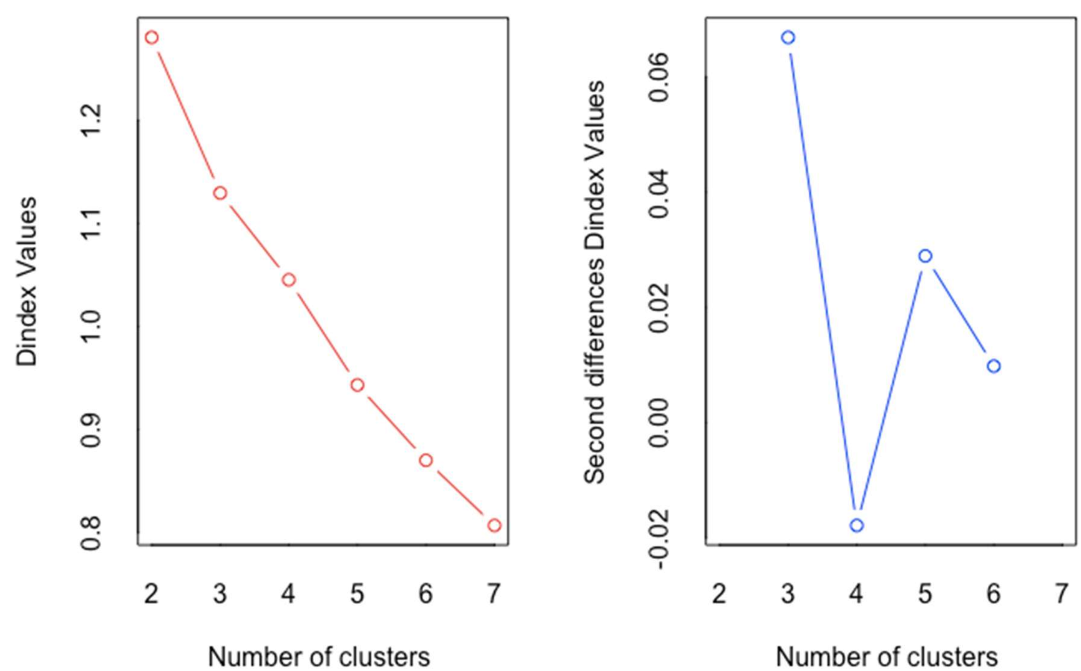
*** : The D index is a graphical method of determining the number of clusters.
In the plot of D index, we seek a significant knee (the significant peak in Dindex second differences plot) that corresponds to a significant increase of the value of the measure.

****
* Among all indices:
* 8 proposed 2 as the best number of clusters
* 2 proposed 3 as the best number of clusters
* 2 proposed 4 as the best number of clusters

High value customers identification for an E-Commerce company

* 9 proposed 5 as the best number of clusters
* 2 proposed 6 as the best number of clusters

***** Conclusion *****



* According to the majority rule, the best number of clusters is 5

****

table(nc$Best.n[1,])

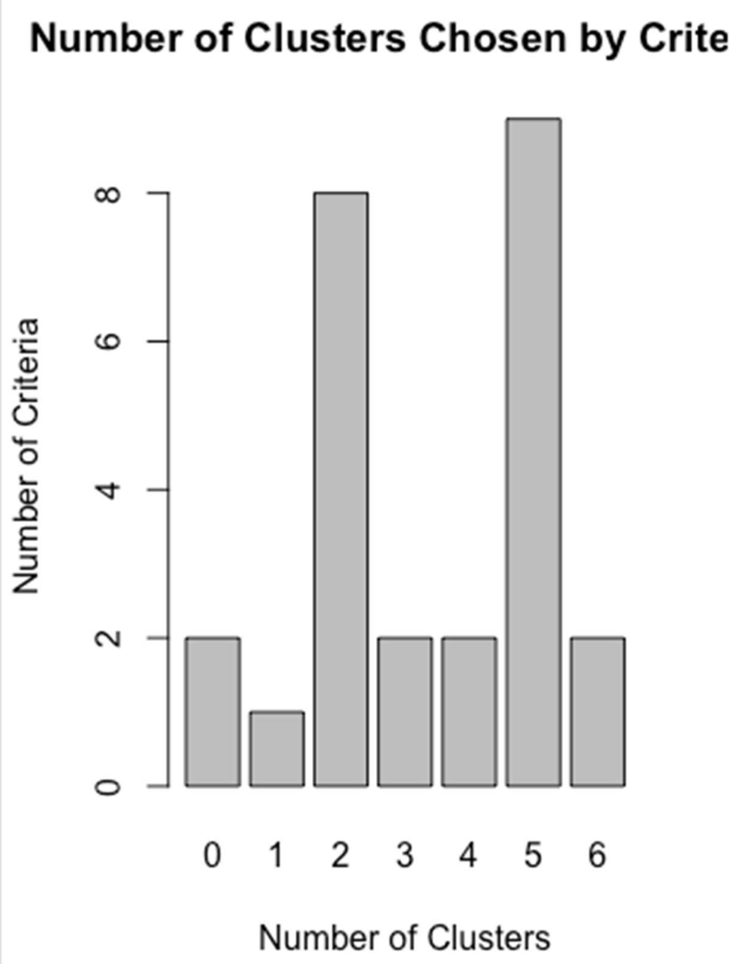
```
0 1 2 3 4 5 6
2 1 8 2 2 9 2
```

nc$All.index # estimates for each number of clusters on 26 different metrics of model fit

```
       KL       CH  Hartigan      CCC    Scott      Marriot   TrCovW    TraceW Friedman
2  2.7802 3486.596 1095.6885 114.3046 19831.20 105114922397 17500303 10972.858  36.5465
3  0.5926 2727.615  546.1660  84.0192 22113.74 140316828346  9531805  8773.166  42.9362
4  0.2366 2227.273 2887.5659  69.7877 24223.18 153973188620  9136622  7798.304  44.7512
5 12.9789 3495.838  733.9822  90.6195 29134.56  78233226810  2501591  4694.740 118.3917
6  0.8084 3412.735  687.4964  88.9212 30717.02  78443919424  1655283  4019.212 139.2931
7  3.4833 3405.573  468.0348  88.9662 32546.43  70263086623  1351303  3472.423 145.9306
    Rubin Cindex     DB Silhouette   Duda  Pseudot2   Beale Ratkowsky      Ball Ptbiserial
2 18.4177 0.1674 1.0986     0.4130 1.5685 -897.7817 -0.6167    0.4741 5486.4290     0.4203
3 23.0355 0.1644 1.3164     0.3067 1.3798 -496.3357 -0.4682    0.4293 2924.3885     0.4126
4 25.9152 0.1549 1.2373     0.2917 0.8334  167.4970  0.3399    0.3958 1949.5761     0.3666
5 43.0470 0.2082 0.9628     0.3350 1.3357 -417.2214 -0.4273    0.3818  938.9480     0.4421
6 50.2821 0.1952 1.0131     0.2951 1.4240 -334.0521 -0.5059    0.3561  669.8687     0.4105
7 58.1998 0.1909 1.0082     0.3015 1.5408 -413.4405 -0.5961    0.3379  496.0605     0.3875
     Frey McClain   Dunn Hubert SDindex Dindex   SDbw
2  0.8433  0.5495 0.0030  1e-04  2.0534 1.2805 1.5295
3  1.3992  0.9606 0.0029  1e-04  2.3273 1.1295 1.0221
4 -0.4374  1.4810 0.0026  1e-04  2.6159 1.0453 1.1863
5  0.8626  1.2075 0.0031  1e-04  2.5153 0.9433 0.8670
```

High value customers identification for an E-Commerce company

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




```
[1] 5
[1] "k-means Solution with 5 Clusters"
  Cluster monetary frequency recency
1       1  1174.35         4      65
2       2     0.00         1     290
3       3  3543.26        11       9
4       4   249.93         1     165
5       5   547.06         2      13
```

Inference : Cluster No 5 is high monetary value,high frequency ,recent purchase group and hence can be identified as a high valued customer segment and should be the most ideal to roll out the loyalty program