Data Science with R

Final Project

Project: 6 High value customers identification for an E-Commerce company

Prepared by -

Niket Khuthia

BUSINESS SCENARIO:

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.

EXPECTATION /GOALS:

Find the significant customers for the business who make high purchases of their favourite products.

The organization wants to roll out an offer to the high-value customers after identification of segments. 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 clustering algorithm.

For this project, I have decided to calculate and work with the following metrics for each customer:

- Recency of last purchase
- Frequency of purchase
- Monetary value

These three variables, collectively known as FRM, are often used in customer segmentation for marketing purposes.

Code with Output:

```
#########################
## My Final R Project ##
########################


##Installing necessary packages

install.packages("plyr")
install.packages("ggplot2")
install.packages("scales")
install.packages("NbClust")
install.packages('cluster')

library(dplyr)
library(ggplot2)
library(scales)
library(NbClust)
library(cluster)

###  Importing dataset

ecom <- read.csv(file.choose()) # Read the CSV file from the data base
head(ecom) # view head of the file(to get to know the data headings)
str(ecom) #viewing the structure of the data

###  Data cleaning & Manuplation
```

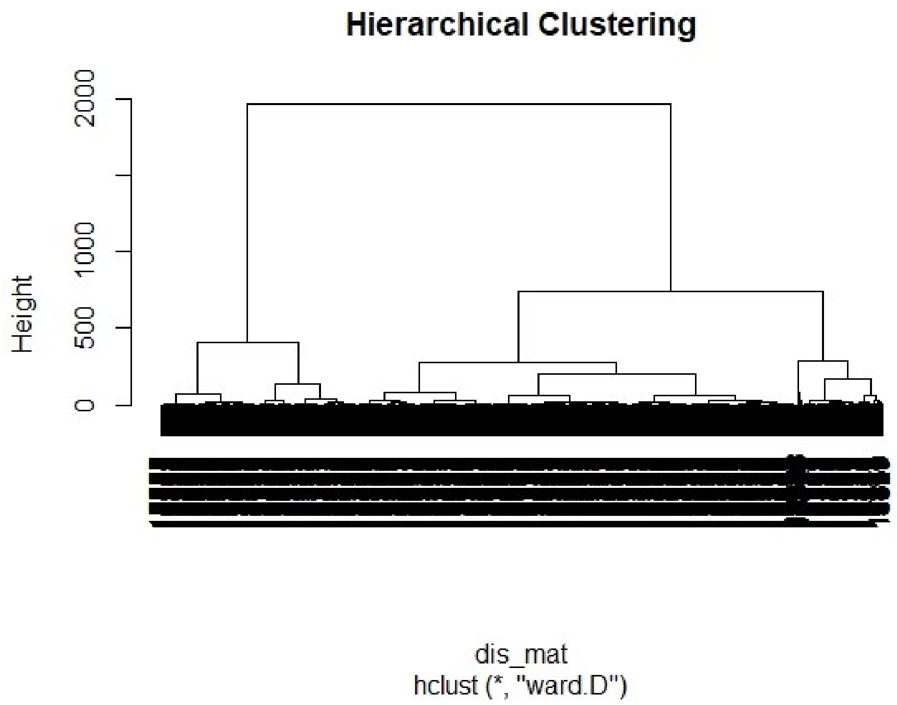
```
# Removing unnecessary column "X"

ecom <- subset(ecom, select = -X) # column X removed from the data set

# Checking customer ID data

length(levels(as.factor(ecom$CustomerID))) # to know count of unique customers
length(unique(ecom$CustomerID)) # to know count of unique values in the data set
#Some invoices have missing CustomerID numbers because there is one unique observations extra.
#We will be removing any observations with missing ID numbers.

sum(is.na(ecom$CustomerID)) # to check no of missing values
ecom <- subset(ecom, !is.na(ecom$CustomerID)) #Data manuplation
length(unique(ecom$CustomerID)) # to know count of unique values in the data set
#hence now the count is equal that means our customer data is cleaned and now we can go forward

# Checking Quantity data

length(levels(as.factor(ecom$Quantity))) # to know count of unique customers
length(unique(ecom$Quantity)) # to know count of unique values in the data set
#there are no missing quantities because the data is equal to 3950 unique customers
#now checking for negative or zero quantaties

sum(ecom$Quantity <= 0) # to check wheather there are negative or zero Quantities
# we got 7533 entries with zero or negative data
ecom <- subset(ecom, !ecom$Quantity <= 0) #Data manuplation
length(unique(ecom$Quantity)) # to know count of unique values in the data set
```

```r
#hence now the count is different, that means our Quantity data is cleaned and now we can go forward

#checking the price data

length(unique(ecom$UnitPrice)) # to know count of unique values in the data set
sum(is.na(ecom$UnitPrice)) # to check no of missing values
# there are no missing values in price
sum(ecom$UnitPrice <= 0) # to check wheather there are negative or zero Pricies
#there are 24 such entries. hence deleting them
ecom <- subset(ecom, !ecom$UnitPrice <= 0) #Data manuplation
length(unique(ecom$UnitPrice)) # to know count of unique values in the data set
#hence now the count is different, that means our Price data is cleaned and now we can go forward

# checking stock code data

length(levels(as.factor(ecom$StockCode))) # to know count of unique stocks
sum(is.na(ecom$StockCode)) # to check no of missing values
# hence there are no missing values, Stockcode data is okay.

# Manuplating country data

length(levels(as.factor(ecom$Country))) # to know count of unique Countries
table(ecom$Country) # there are only 38 countries so viewing them by count will be better
# as we see that UK contains maximum no of customers, hence elimnating other countries customer data wont harm our data significantly.
#Hence we will only consider UK customers for our analysis.
ecom <- subset(ecom, Country == "United Kingdom")
```

```
#checking for return items
sum(grepl("C", ecom$InvoiceNo, fixed=TRUE)) # checking for any return items
#the sum is zero there are no return items



###############################################################################
## To Create new customer database -  # to estimate valuable customers ##
###############################################################################



### code for Receancy of customers ###

# converting data into std date format
ecom$InvoiceDate <- as.Date(ecom$InvoiceDate, "%d-%b-%y") #to convert from character to date format
#creating new column for recent transaction and calculating no of days since last transaction
ecom$Recency <- ((max(ecom$InvoiceDate) + 1) - ecom$InvoiceDate) # creating new table column for recency
range(ecom$Recency) # Viewing range of difference in days
# creating new data frame for customers data
customers <- as.data.frame(unique(ecom$CustomerID)) # creating unique data frame of customer ID
names(customers) <- "CustomerID" # naming the Column name
#calculating for recency
# Obtain no of days since most recent purchase
Recency <-  aggregate(Recency ~ CustomerID, data=ecom, FUN=min)
#merging data by CustomerID
customers <- merge(customers, Recency, by="CustomerID", all=TRUE, sort=TRUE)#merging operation
remove(Recency) # removing the unnecssary data
```

```
#converting data into integer format

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

str(customers)


### Code for Frequency of customers ###


custinvoice <- subset(ecom, select = c("CustomerID","InvoiceNo")) #creating new data frame

custinvoice <- custinvoice[!duplicated(custinvoice),] # removing dublicates

custinvoice <- arrange(custinvoice, CustomerID) # arranging by customerID

row.names(custinvoice) <- NULL #removing rownames

custinvoice$Frequency <- 1 # adding extra column of value 1 to find sum

#making new dataframe of unique customers with frequency table

invoices <- aggregate(Frequency ~ CustomerID, data=custinvoice, FUN=sum)

# Add & merge no of invoices to customers data

customers <- merge(customers, invoices, by="CustomerID", all=TRUE, sort=TRUE)

remove(invoices, custinvoice) #removing unnecessary data

table(customers$Frequency)

# Removing customers who have not made any purchases in the past year or feilds with NA

customers <- subset(customers, Frequency > 0)

# now our customer data has frequency, as well as recency


### code for Monetary Value of Customers ###


# total Spending per transaction

ecom$Amount <- (ecom$Quantity*ecom$UnitPrice) # To creat amount table to know about the total spending in a transaction

# creating new dataset total sales to customer

totalsales <- aggregate(Amount ~ CustomerID, data=ecom, FUN=sum)

names(totalsales)[names(totalsales)=="Amount"] <- "Monetary"
```

```
# Add & Merge Monetary value to customers dataset

customers <- merge(customers, totalsales, by="CustomerID", all.x=TRUE, sort=TRUE) #merging operation

remove(totalsales) # removing unnecssary data

str(customers)

View(customers)



###########################################

### Clustering to find no of divisions ###

###########################################



# standardise data


sapply(customers[,-1], mean) # Viewing Mean of customers data

sapply(customers[,-1], sd) # Viewing SD of customers data

scale_cust <- scale(customers[,-1]) # creating standerized data by scaling

round(apply(scale_cust, 2, mean)) # checking for standarized mean

apply(scale_cust, 2, sd) # checking for standarized SD


# Hiearchical Clustering


dis_mat <- dist(scale_cust, method = 'euclidean')

hclus <- hclust(dis_mat, method = 'ward.D')


# dendogram


plot(hclus, labels = as.character(customers$CustomerID),
    main = 'Hierarchical Clustering')
```

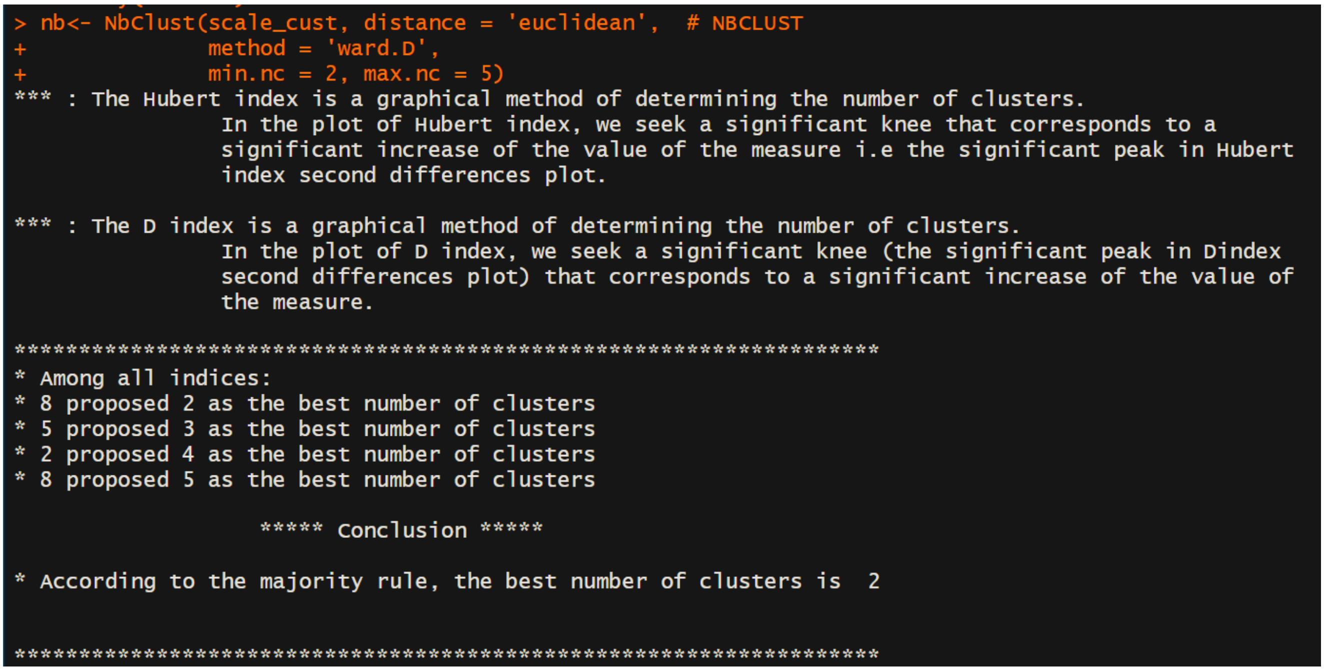


nb<- NbClust(scale_cust, distance = 'euclidean', # NBCLUST

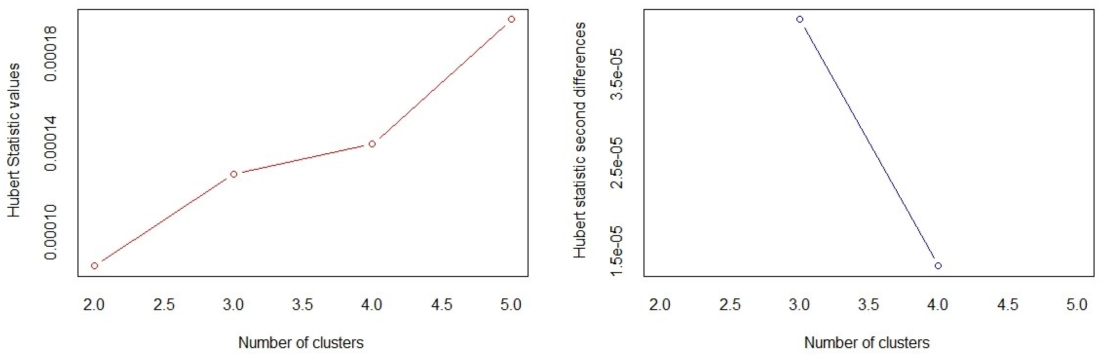
method = 'ward.D',

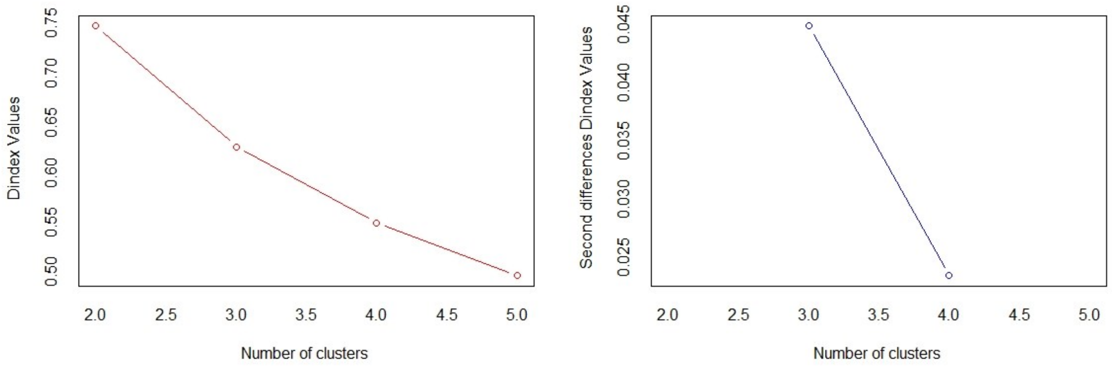
min.nc = 2, max.nc = 5)

#considering maximum value = 5 clusters



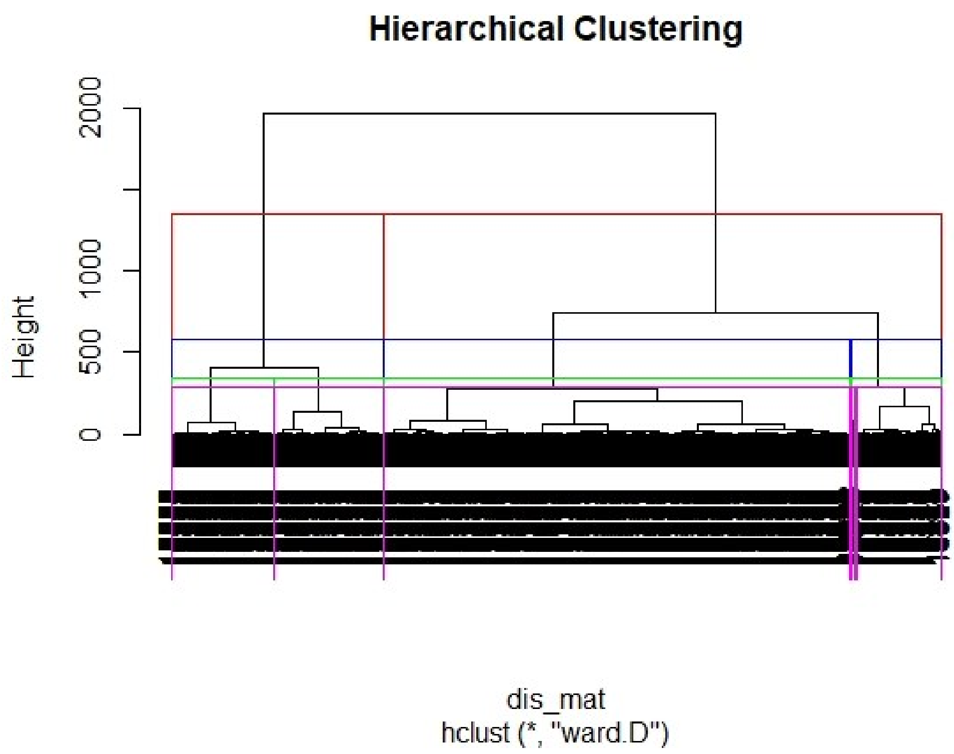
```
> nb<- NbClust(scale_cust, distance = 'euclidean',  # NBCLUST
+              method = 'ward.D',
+              min.nc = 2, max.nc = 5)
*** : The Hubert index is a graphical method of determining the number of clusters.
                In the plot of Hubert index, we seek a significant knee that corresponds 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
* 5 proposed 3 as the best number of clusters
* 2 proposed 4 as the best number of clusters
* 8 proposed 5 as the best number of clusters

                   ***** Conclusion *****

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


*******************************************************************
```





```
# concludes based on 3920 indices
```

```
plot(hclus, labels = as.character(customers$CustomerID),
    main = 'Hierarchical Clustering')
rect.hclust(hclus, k = 2, border = 'red')
rect.hclust(hclus, k = 3, border = 'blue')
rect.hclust(hclus, k = 4, border = 'green')
rect.hclust(hclus, k = 5, border = 'magenta')
```

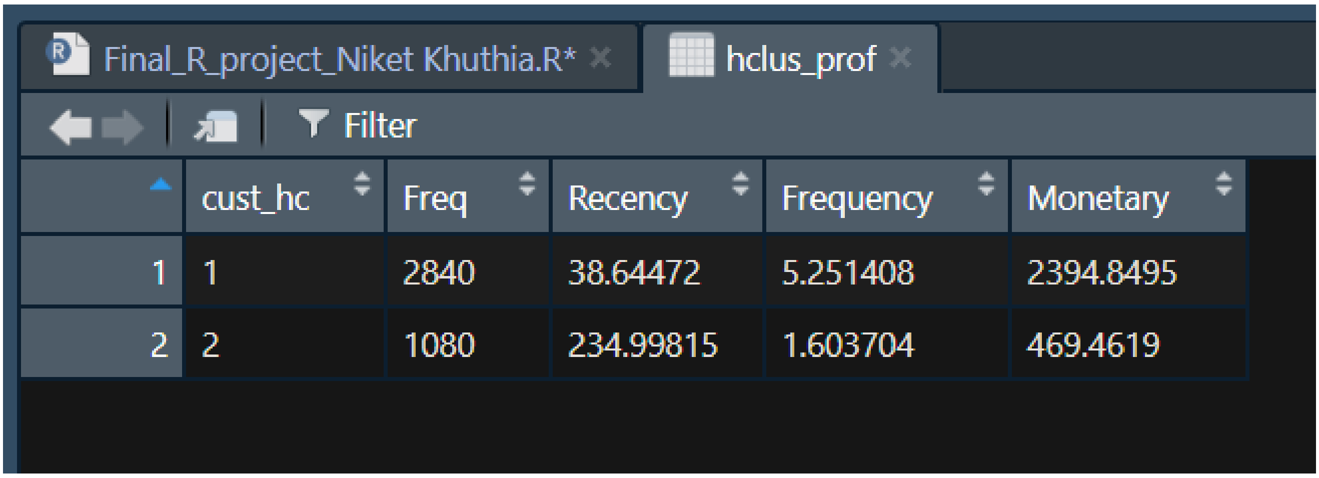


#hence going with majority and considering 2 clusters

# Premium customers # Silver Customers

# cluster profile

```
customers['cust_hc'] <- cutree(hclus, k = 2) # get cluster labels

hclus_prof<- customers%>%
  dplyr::select(-CustomerID)%>%
  group_by(cust_hc)%>%
  summarise_all(mean)%>%
  mutate(Freq = as.vector(table(customers$cust_hc)))%>%
  dplyr::select(cust_hc,Freq, Recency, Frequency, Monetary)%>%
  data.frame()
View(hclus_prof)
```



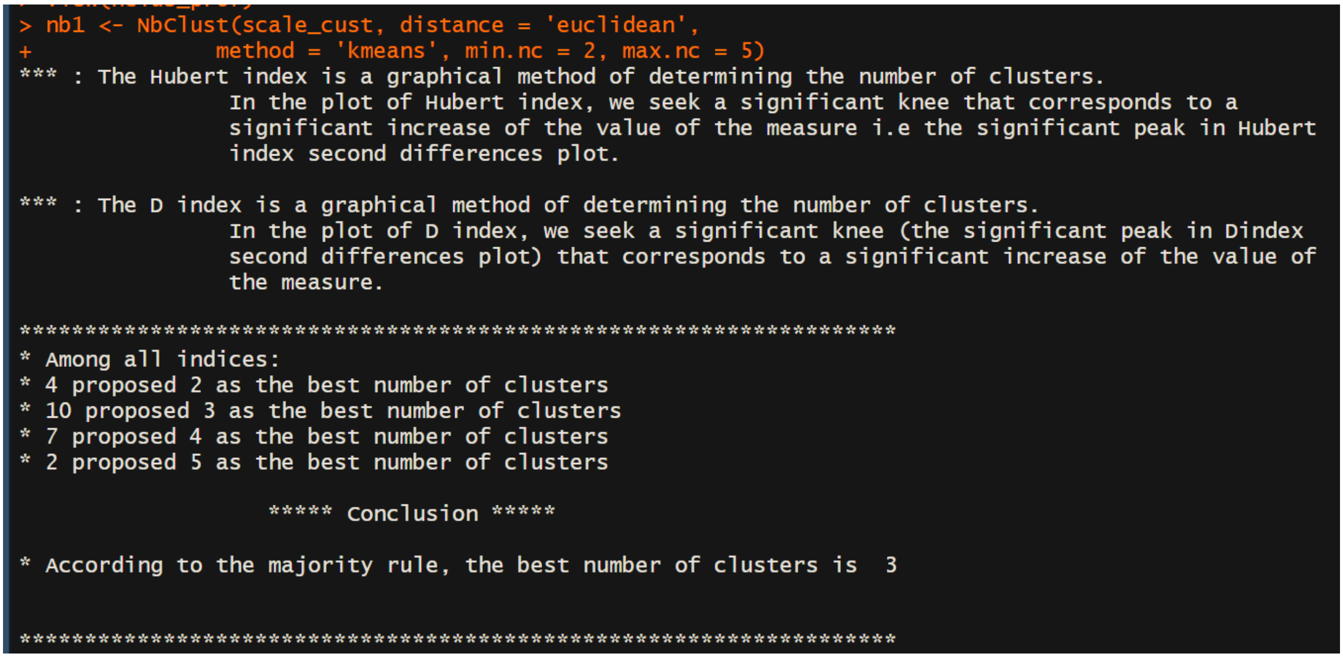
	cust_hc	Freq	Recency	Frequency	Monetary
1	1	2840	38.64472	5.251408	2394.8495
2	2	1080	234.99815	1.603704	469.4619

# Also comparing the above dataset by kmeans clustring

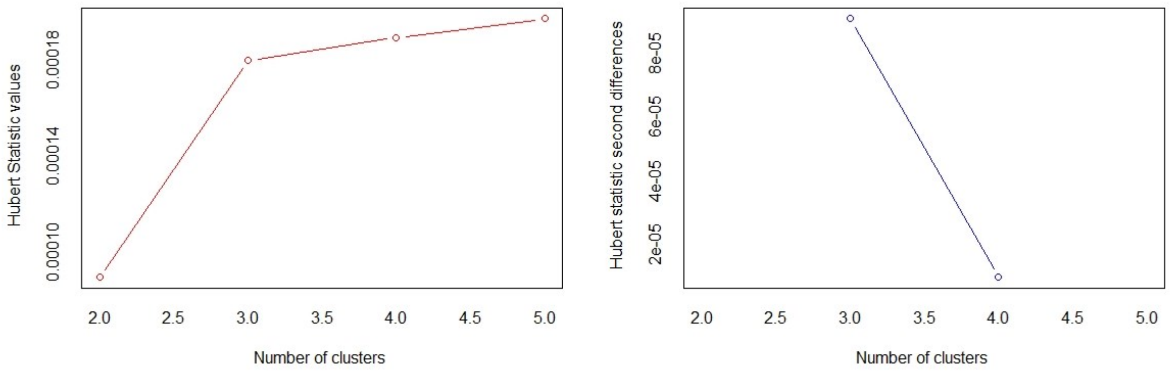
nb1 <- NbClust(scale_cust, distance = 'euclidean',

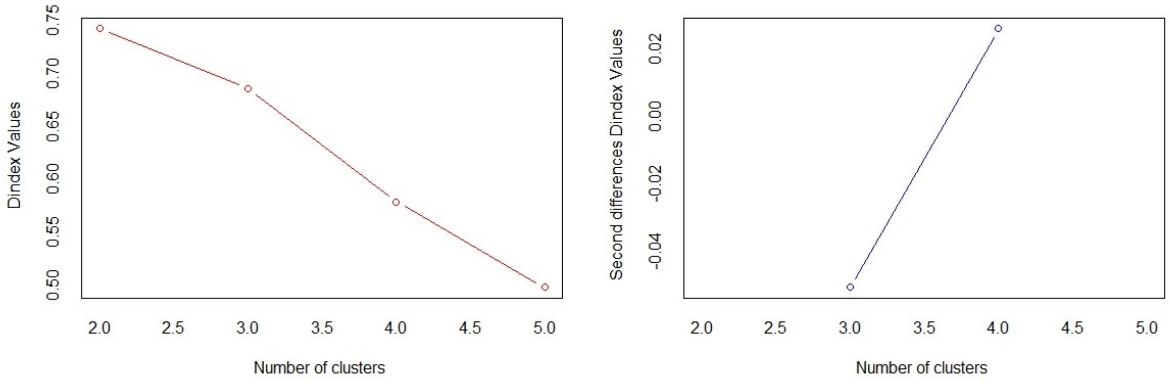
method = 'kmeans', min.nc = 2, max.nc = 5)

# again considering maximum value = 5 clusters



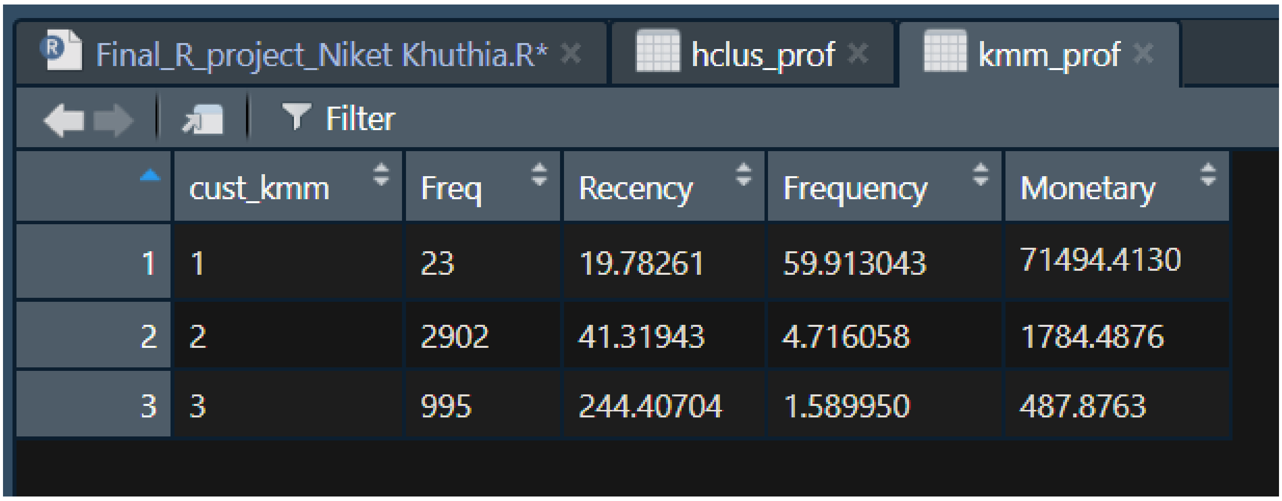
```
> nb1 <- NbClust(scale_cust, distance = 'euclidean',
+               method = 'kmeans', min.nc = 2, max.nc = 5)
*** : The Hubert index is a graphical method of determining the number of clusters.
                In the plot of Hubert index, we seek a significant knee that corresponds to a 
                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:                                                
* 4 proposed 2 as the best number of clusters 
* 10 proposed 3 as the best number of clusters 
* 7 proposed 4 as the best number of clusters 
* 2 proposed 5 as the best number of clusters 

                   ***** Conclusion *****                            
 
* According to the majority rule, the best number of clusters is  3 
 
 
******************************************************************* 
```





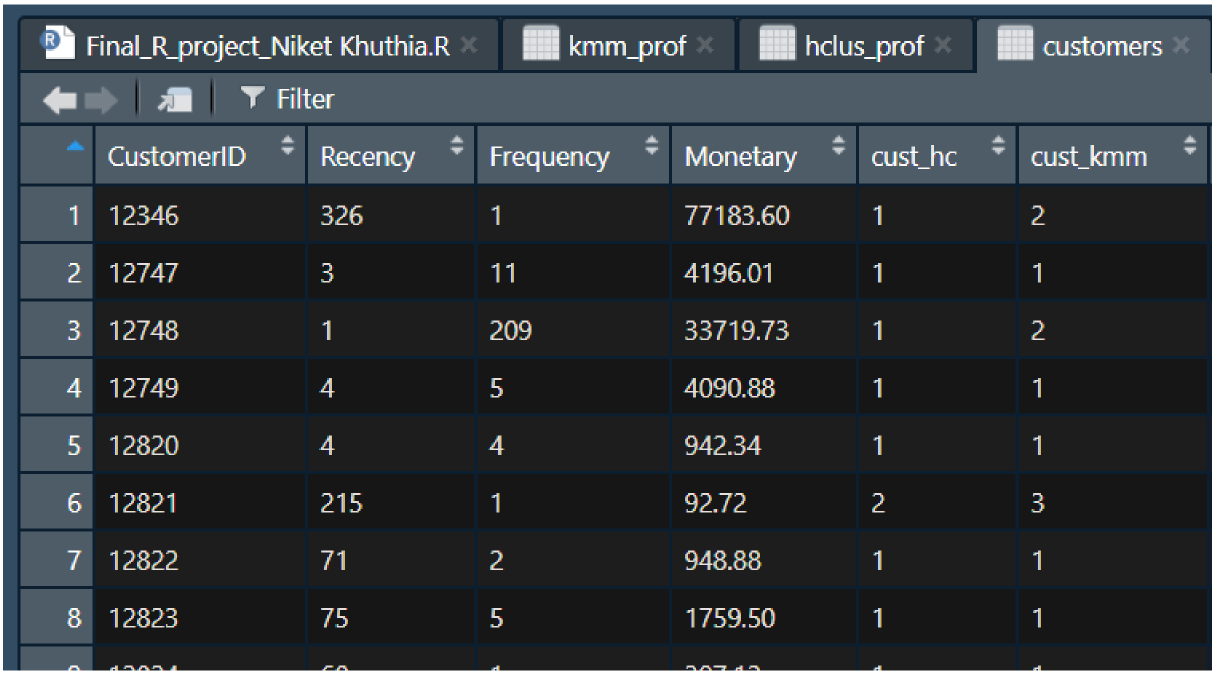
```
#result is 3 clusters from the majority the three classifications may be below

#Platinum # Gold # Silver

# perform kmeans

kmm <- kmeans(scale_cust, centers = 3)

kmm


# cluster profile


customers['cust_kmm'] <- kmm$cluster


kmm_prof<- customers %>%

  dplyr::select(-CustomerID)%>%

  group_by(cust_kmm)%>%
  summarise_all(mean)%>%
  mutate(Freq = as.vector(table(customers$cust_kmm)))%>%
  dplyr::select(cust_kmm,Freq, Recency, Frequency, Monetary)%>%
  data.frame()
```

```
View(kmm_prof)
```



	cust_kmm	Freq	Recency	Frequency	Monetary
1	1	23	19.78261	59.913043	71494.4130
2	2	2902	41.31943	4.716058	1784.4876
3	3	995	244.40704	1.589950	487.8763

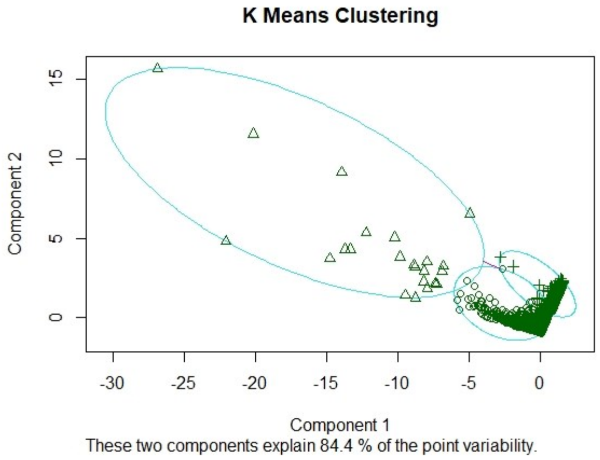
```
View(customers)
```

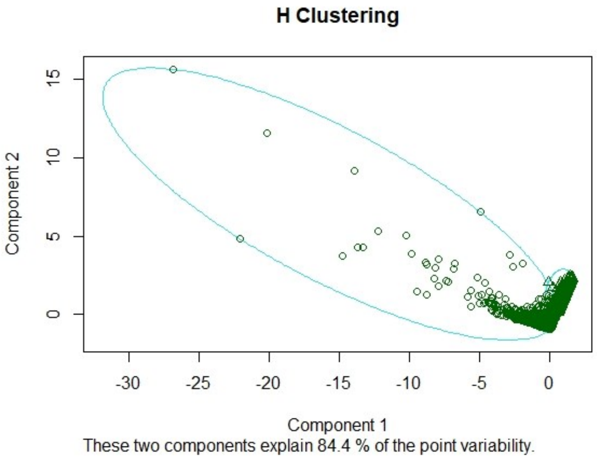


	CustomerID	Recency	Frequency	Monetary	cust_hc	cust_kmm
1	12346	326	1	77183.60	1	2
2	12747	3	11	4196.01	1	1
3	12748	1	209	33719.73	1	2
4	12749	4	5	4090.88	1	1
5	12820	4	4	942.34	1	1
6	12821	215	1	92.72	2	3
7	12822	71	2	948.88	1	1
8	12823	75	5	1759.50	1	1

```
# Visualize the clusters
```

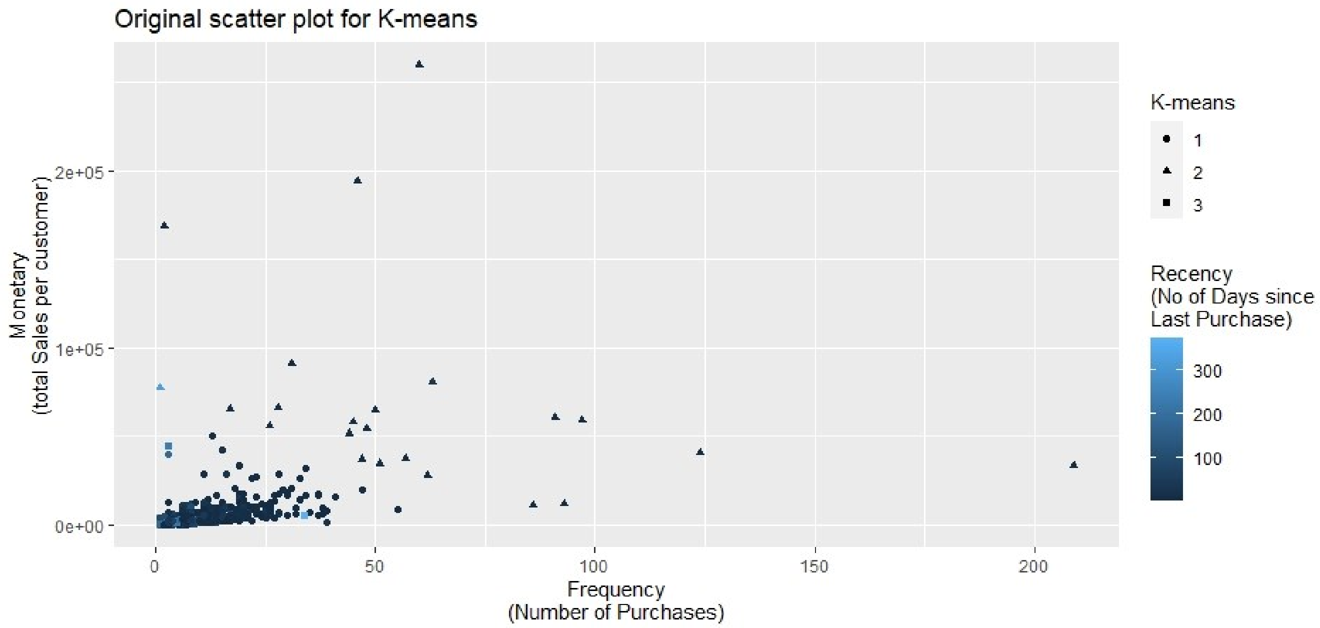
```
cluster::clusplot(scale_cust, kmm$cluster, main = 'K Means Clustering')

cluster::clusplot(scale_cust, customers$cust_hc, main = 'H Clustering')
```



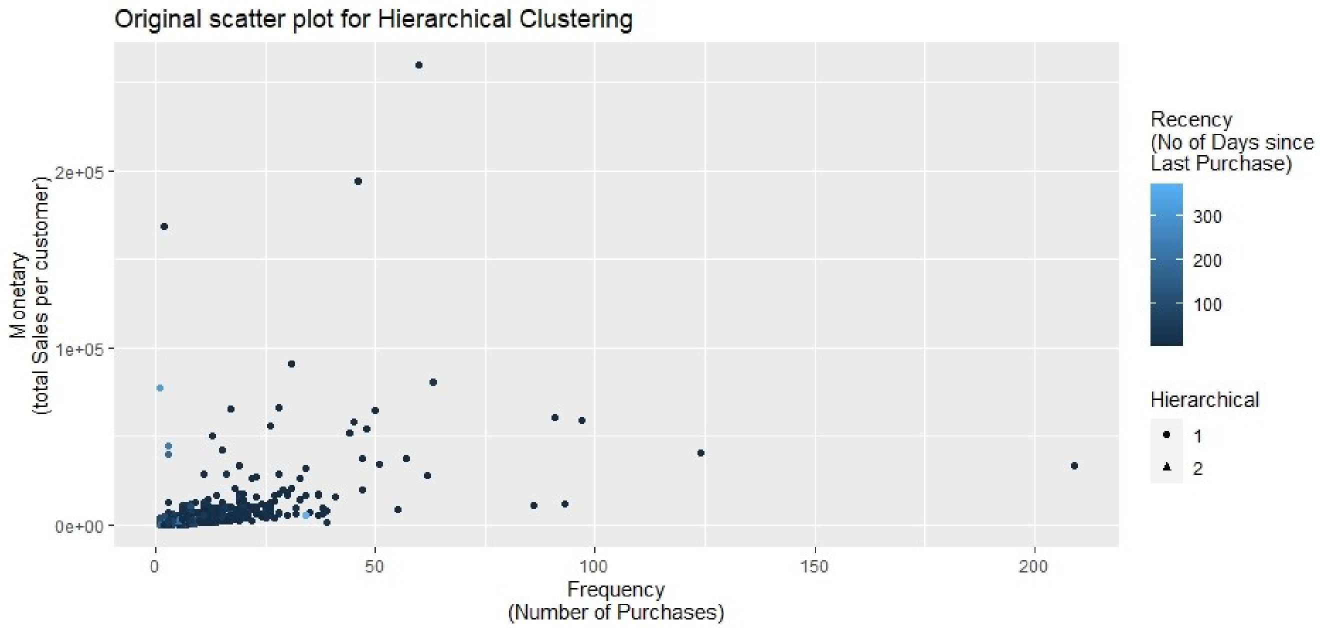


```
## to know propotion of the data from H clustering


round(prop.table(table(customers$cust_hc)), 2)

sum(customers$cust_hc == 1)


# there are 2840 top Customers according to H clustering

# this indicates that

#premium customers are approx 72 % of the total customers

#silver customers are approx 28 % of the total customers


## to know propotion of the data from K Means Clustering


round(prop.table(table(customers$cust_kmm)), 2)

sum(customers$cust_kmm == 2)


# there are 23 top Customers according to Kmeans clustering

# this indicates that

#pletinum customers are approx 1 % of the total customers
```

```
#gold customers are approx 74 % of the total customers

#Silver customers are approx 25 % of the total customers


remove(dis_mat) # removing the unwanted data

################################

## Visualization of the data ##

################################


# Original scale


customers$cust_hc <- as.factor(customers$cust_hc) #converting cluster into factor so that it can be readebale by ggplot

customers$cust_kmm <- as.factor(customers$cust_kmm) #converting cluster into factor so that it can be readebale by ggplot


## sactter plot for original values using ggplot tool


# Scatter plot for Kmeans clustering


og_scatter_k <- ggplot(customers, aes(x = Frequency, y = Monetary))

og_scatter_k <- og_scatter_k + geom_point(aes(colour = Recency, shape = cust_kmm))

og_scatter_k <- og_scatter_k + scale_shape_discrete(name = "K-means")

og_scatter_k <- og_scatter_k + scale_colour_gradient(name="Recency\n(No of Days since Last Purchase)")

og_scatter_k <- og_scatter_k + xlab("Frequency\n(Number of Purchases)")

og_scatter_k <- og_scatter_k + ylab("Monetary\n(total Sales per customer)")

og_scatter_k <- og_scatter_k + ggtitle("Original scatter plot for K-means")

og_scatter_k
```



```
# Scatter plot for Hierarchical clustering

og_scatter_h <- ggplot(customers, aes(x = Frequency, y = Monetary))

og_scatter_h <- og_scatter_h + geom_point(aes(colour = Recency, shape = cust_hc))

og_scatter_h <- og_scatter_h + scale_shape_discrete(name = "Hierarchical")

og_scatter_h <- og_scatter_h + scale_colour_gradient(name="Recency\n(No of Days since Last Purchase)")

og_scatter_h <- og_scatter_h + xlab("Frequency\n(Number of Purchases)")

og_scatter_h <- og_scatter_h + ylab("Monetary\n(total Sales per customer)")

og_scatter_h <- og_scatter_h + ggtitle("Original scatter plot for Hierarchical Clustering")

og_scatter_h
```



# we cant understand the data because it is concentrated on one side and dispersed on the other

#This first graph uses the variable is original metrics and is almost completely uninterpretable.

#There is a clump of data points in the lower left-hand corner of the plot, and then a few outliers.

# hence we are applying for log transformation

## Preprocessing the data ##

# Log-transformation

customers$Recency.log <- log(customers$Recency)

customers$Frequency.log <- log(customers$Frequency)

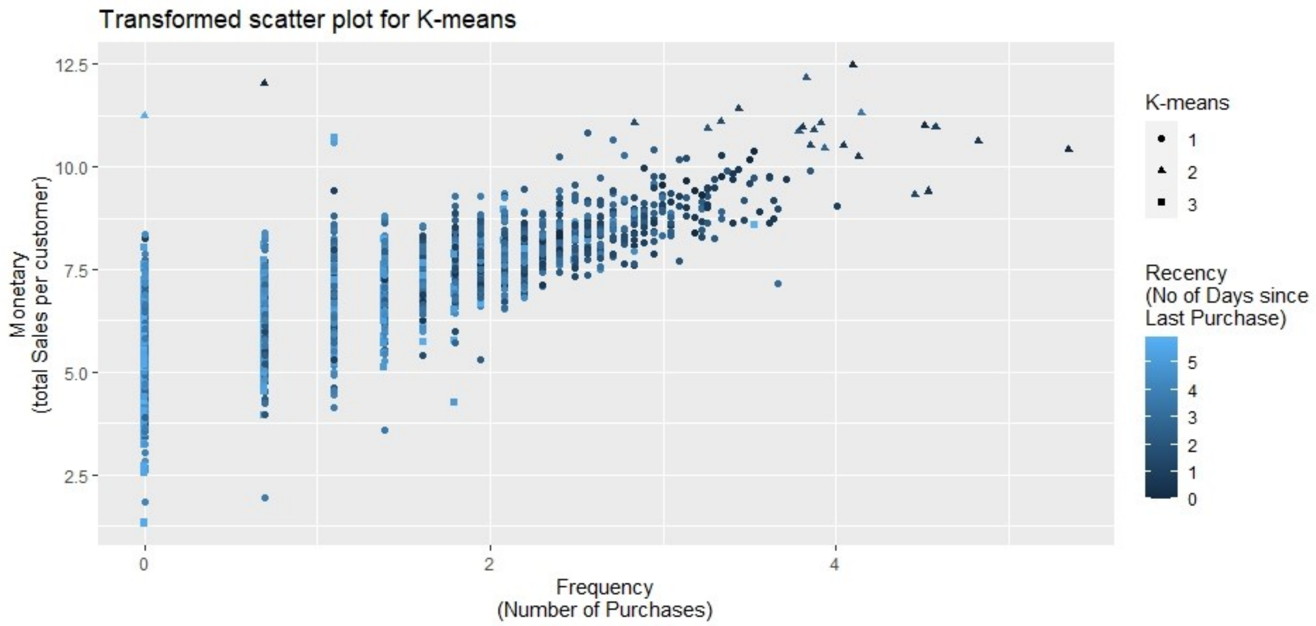
customers$Monetary.log <- log(customers$Monetary)

## sactter plot for transformed values using ggplot tool

# Scatter plot for Kmeans clustering

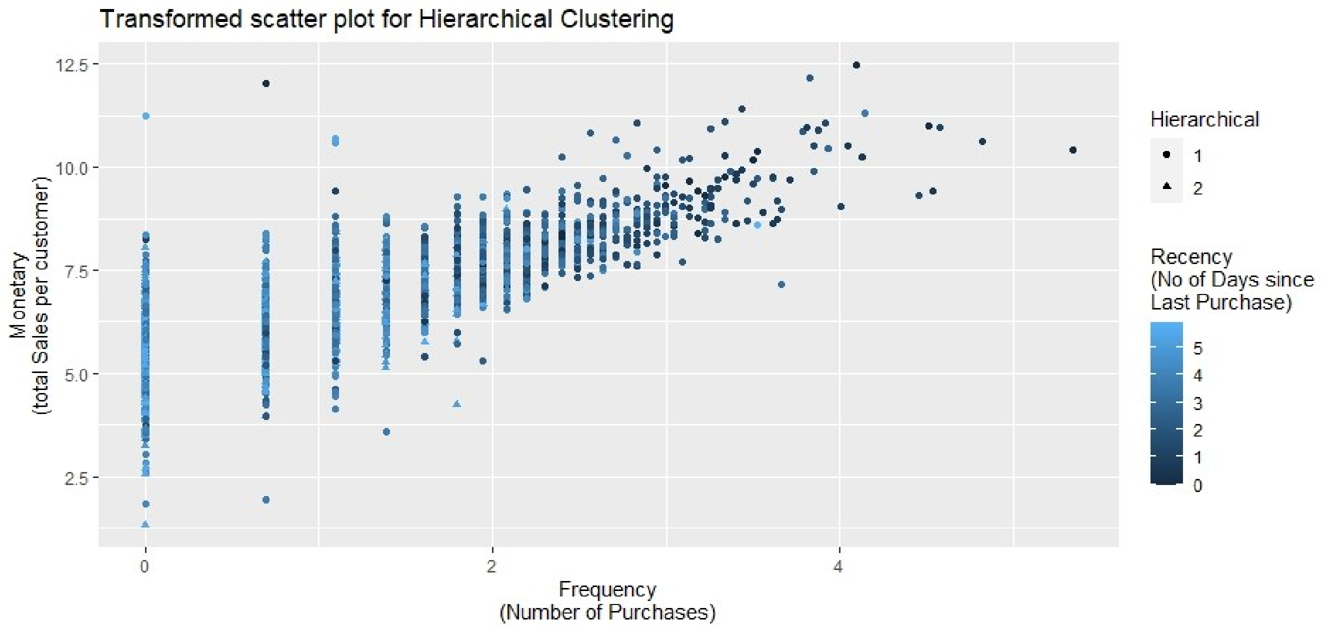
log_scatter_k <- ggplot(customers, aes(x = Frequency.log, y = Monetary.log))

```
log_scatter_k <- log_scatter_k + geom_point(aes(colour = Recency.log, shape = cust_kmm))

log_scatter_k <- log_scatter_k + scale_shape_discrete(name = "K-means")

log_scatter_k <- log_scatter_k + scale_colour_gradient(name="Recency\n(No of Days since Last Purchase)")

log_scatter_k <- log_scatter_k + xlab("Frequency\n(Number of Purchases)")

log_scatter_k <- log_scatter_k + ylab("Monetary\n(total Sales per customer)")

log_scatter_k <- log_scatter_k + ggtitle("Transformed scatter plot for K-means")

log_scatter_k
```



```
# Scatter plot for Hierarchical clustering


log_scatter_h <- ggplot(customers, aes(x = Frequency.log, y = Monetary.log))

log_scatter_h <- log_scatter_h + geom_point(aes(colour = Recency.log, shape = cust_hc))

log_scatter_h <- log_scatter_h + scale_shape_discrete(name = "Hierarchical")

log_scatter_h <- log_scatter_h + scale_colour_gradient(name="Recency\n(No of Days since Last Purchase)")

log_scatter_h <- log_scatter_h + xlab("Frequency\n(Number of Purchases)")

log_scatter_h <- log_scatter_h + ylab("Monetary\n(total Sales per customer)")

log_scatter_h <- log_scatter_h + ggtitle("Transformed scatter plot for Hierarchical Clustering")
```

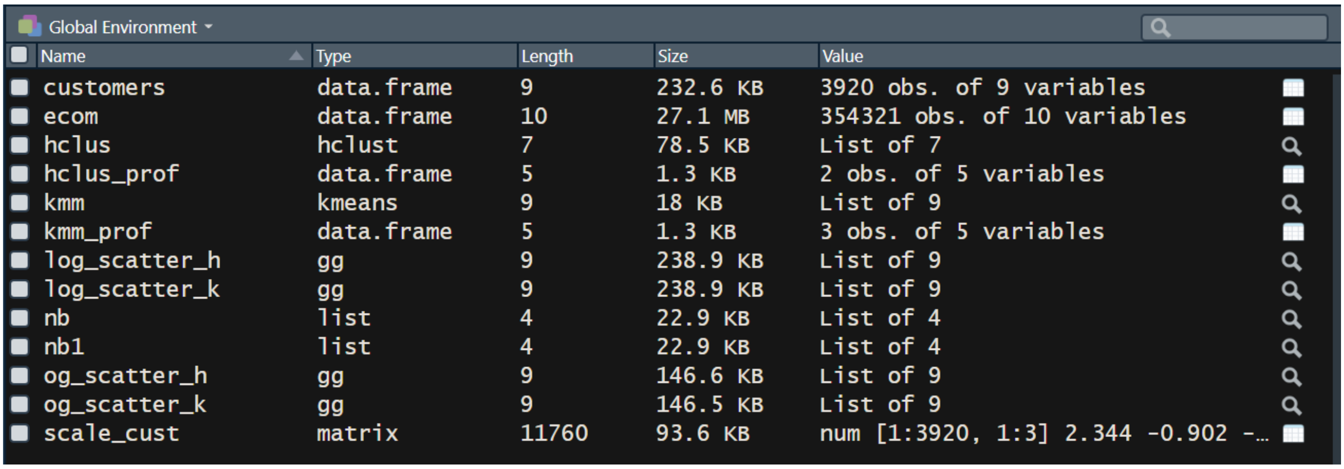
log_scatter_h



#################

### Thank you ###

#################



Name	Type	Length	Size	Value
customers	data.frame	9	232.6 KB	3920 obs. of 9 variables
ecom	data.frame	10	27.1 MB	354321 obs. of 10 variables
hclus	hclust	7	78.5 KB	List of 7
hclus_prof	data.frame	5	1.3 KB	2 obs. of 5 variables
kmm	kmeans	9	18 KB	List of 9
kmm_prof	data.frame	5	1.3 KB	3 obs. of 5 variables
log_scatter_h	gg	9	238.9 KB	List of 9
log_scatter_k	gg	9	238.9 KB	List of 9
nb	list	4	22.9 KB	List of 4
nb1	list	4	22.9 KB	List of 4
og_scatter_h	gg	9	146.6 KB	List of 9
og_scatter_k	gg	9	146.5 KB	List of 9
scale_cust	matrix	11760	93.6 KB	num [1:3920, 1:3] 2.344 -0.902 -…

## Above attached is the snapshot of Global Environment

ANALYSIS:

FRM Variables

The original dataset was organized long, with invoices nested within customer. To tackle this, I have created a customer-level dataset and added recency, frequency, and monetary value data to it.

• The recency variable denotes to the number of days that have elapsed since the customer last purchased something (so, smaller numbers indicate more recent activity on the customer's account).

• Frequency refers to the number of invoices with purchases.

• Monetary value is the amount that the customer spent.

I have Also filtered the data according to UK country only because this is a UK based company and approx. 95 % customers are from UK. Also it would be easy to analyse for a single country.

Clustering data Insights

I have applied Hierarchical Clustering as well as K-means Clustering on the dataset of 3920 unique Customers.

A. Hierarchical Clustering

In Hierarchical Clustering, it was advised by the majority to divide the customers data into 2 clusters. Two clusters are categorised as Premium customers & Silver Customers.

Here, Premium customers consisted of 2840 customers which holds 72% of the total customers. Whereas Silver customers consisted of 1080 customers which holds about 28% of the total customers

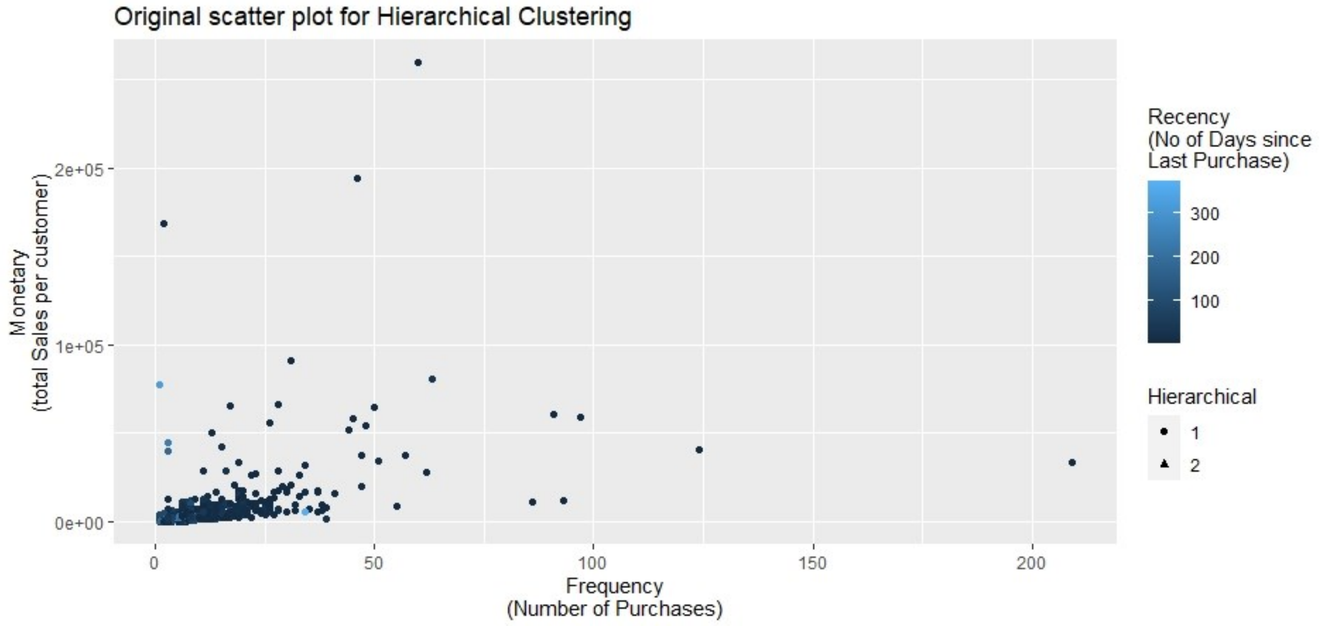
B. K-means Clustering

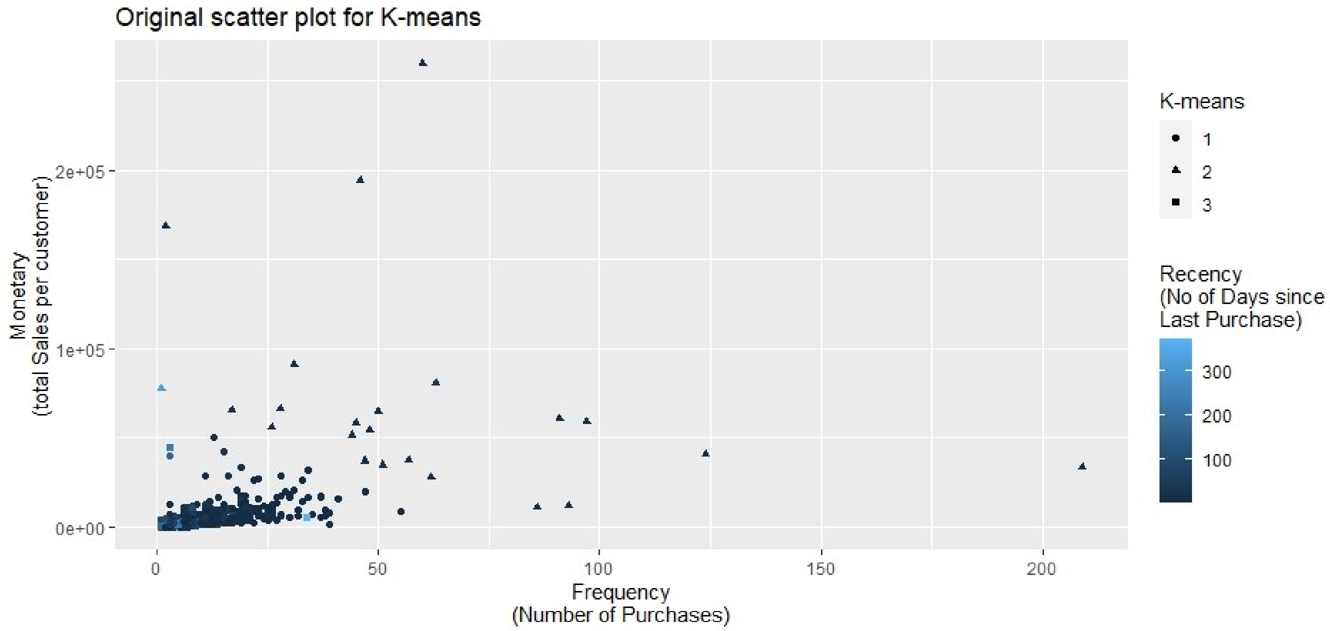
In K-means clustering, it was advised by the majority to divide the customers data into 3 clusters. the clusters are categorised as Premium customers, Gold Customers & Silver Customers.

Here, Premium customers consisted of 23 customers which holds approximately 1% of the total customers. Probably all these are dealers because the frequency, recency and Monetary is very high. Whereas Gold customers consisted of 1902 customers which holds about approximately 74% of the total customers and similarly Silver customers consisted of 995 customers which holds about approximately 25% of the total customers

Visualization of data

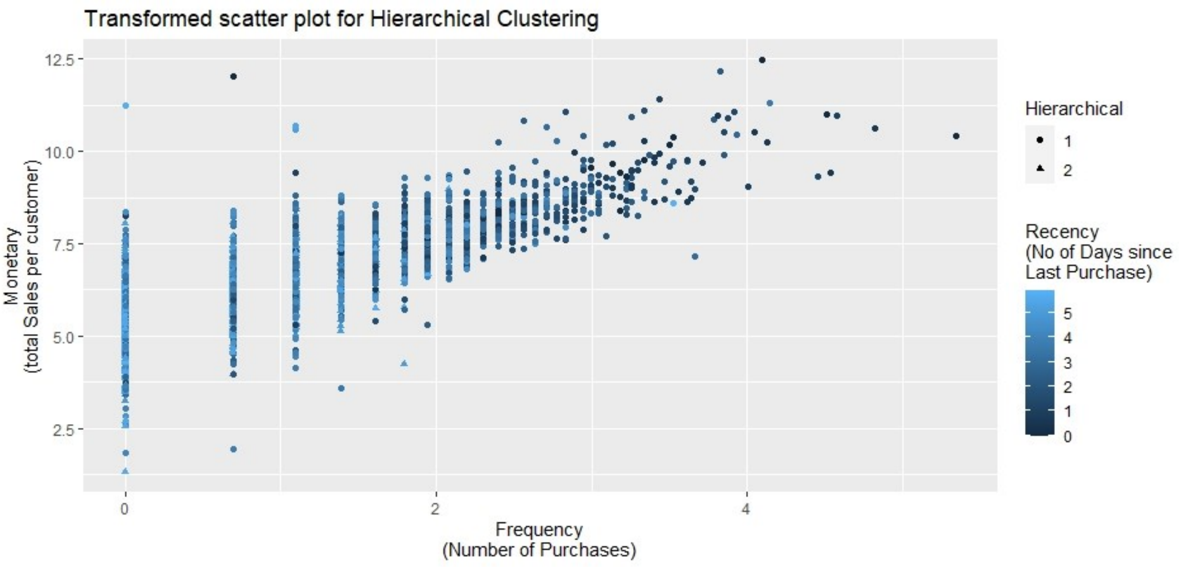
The user has to select the number of clusters with k-means clustering. Looking at the data can give a sense for what we are dealing with and how many clusters we might have. In the graphs below, the outcome we're probably most interested in, customer monetary value, is plotted on the y-axis. Frequency of purchases is on the x-axis, and the third variable, recency of purchase, is represented by color-coding the data points. Lastly, we have also included the cluster segments using different shapes, so we could map those designations on to customer monetary value, frequency, and recency.

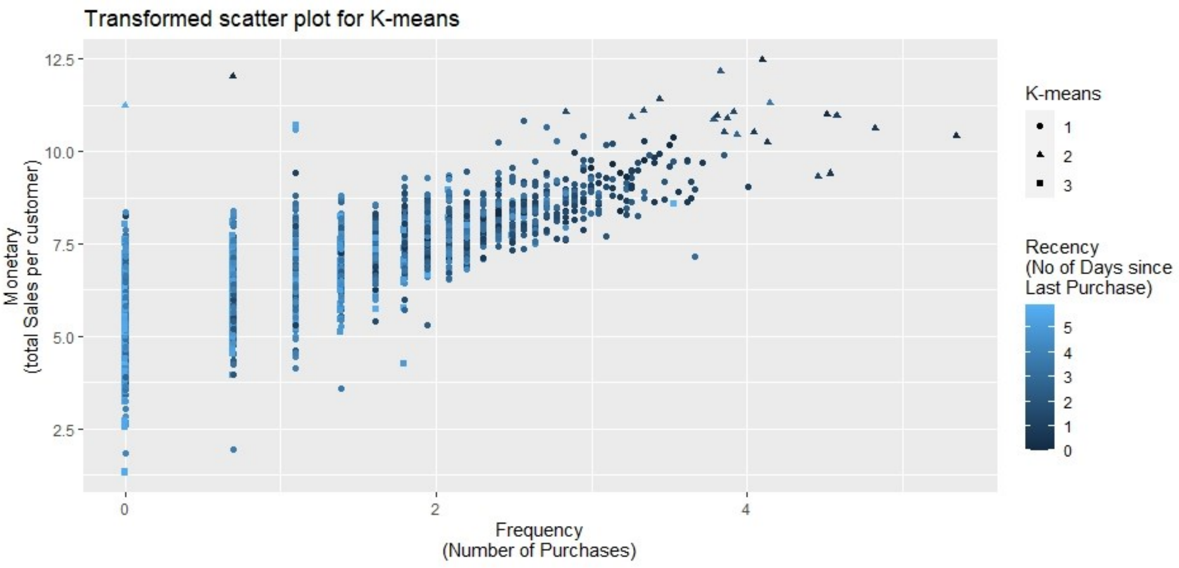




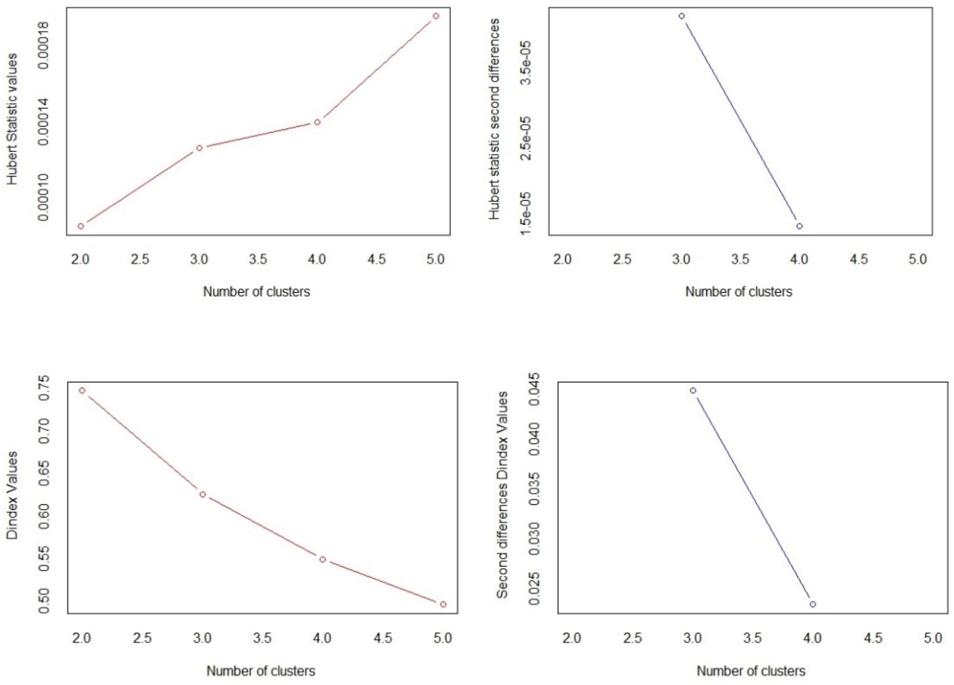
This first & the second graph represents Hierarchical Clustering & K-means Clustering respectively and reuses the variables from original metrics and as we can see, the result is almost completely uninterpretable. There's a clump of data points in the lower left-hand corner of the plot, and then a few outliers. This is reason we have log-transformed our input variables.

Below are the log transformed graphs

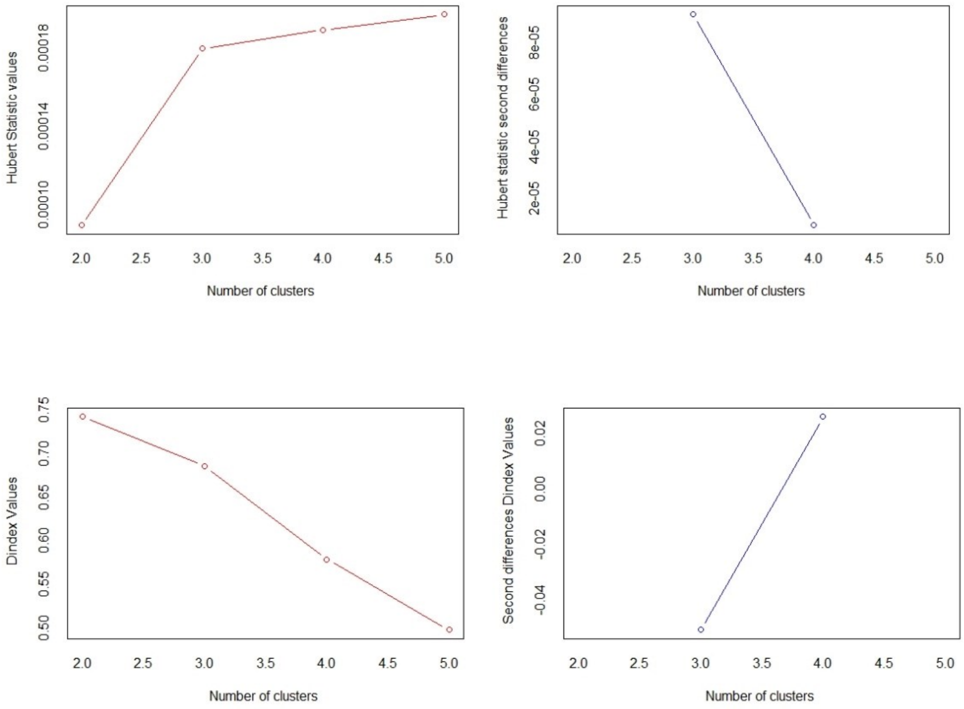




This looks better now. Now we can see a scattering of high-value, high-frequency customers in the top, right-hand corner of the graph. These data points are dark, indicating that they've purchased something recently. In the bottom, left-hand corner of the plot, we can see a couple of low-value, low frequency customers who haven't purchased anything recently, with a range of values in between. Notably, we can also see that the data points are fairly continuously-distributed. There aren't really any clear clusters formations. This means that any cluster groupings we create won't exactly reflect some true, underlying group membership – they'll be somewhat arbitrary distinctions that we draw for our own purposes.



The above two graphs are formed by NbClust from Hierarchical Clustering here we do not see any typical Knee point and the graph is almost linear. This means the variance is not big.



Similarly, here in above two graphs are formed by NbClust from K-means Clustering we see that there is a clean Knee type formation on 3rd cluster. This means the variance after 3 cluster is high and by choosing 3 clusters will be better solution.

CONCLUSION

If the business wants to use the results to understand a range of customer behaviour from high-to-low value customers, I'd probably recommend the 3-cluster solution from k-means clustering. I like that it distinguishes the no-value group of customers, whom the business probably wants to eliminate as much as possible, and also separates low-value, low-frequency customers who have purchased recently from those who have not. It may be easier to encourage recently-active customers to re-engage with the business and possibly develop into medium-value customers. That said, there isn't just one correct decision here. With regards to our business decision on whom to deploy customer loyalty programs: Cluster No. 2 is a 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 group to roll out the loyalty program and then followed by group no 1 for encouraging them for buying more as premium customers by giving a different offer on event basis. Lastly, group 3 is a tough nut to crack in which they are very low in all the three categories, we can take their feedback and see where we failed and how we can improve our product and service.