UK Retail E-commerce Sales

Submitted By:

Joseph Gyamfi
Masih Haqdad
Fatima Shah
Rashmi Sharma
Kaptan Singh
Deepa Sreekumar

January 12, 2019

TABLE OF CONTENTS

Section	Page
Abstract	3
Background and Introduction	3
Objective and Methodology	4
Data Understanding	4
Data Description and Exploration	5
Modeling & Interpretation	7
* Methodology/Approach	9
Observation and Plot Interpretation	16-18
Assessment of Ward Agglomeration Method	18-26
Deployment Strategy	27
Recommendation	28

Abstract

Our analysis is based on a UK based online retail store that sells specialty product that mostly caters to wholesalers. The business aspect of collecting this data from online retail customers goes beyond information gathering, record keeping and monitoring. Our team is working together to identify the best utilization of marketing budget for existing customer base. Our main objective is to develop an improved, informed marketing strategy that is tailored towards each customer group. We will be extracting value from data set that allows the retail store to efficiently and effectively allocate marketing budget while also reducing waste of resources.

Background and Introduction

E-commerce is a massively large area that has been growing exponentially as people are becoming more comfortable with online shopping. As per Statista, the e-commerce sales worldwide valued to 2.3 trillion US dollars and e-retail revenue between 2014 to 2021. It is projected to grow 4.88 trillion US dollars in 2021. Online shopping is becoming a widely popular activity online worldwide, however usage may vary by region. (See Figure 1)

Retail e-commerce sales worldwide from 2014 to 2021 (in billion U.S. Dollars)

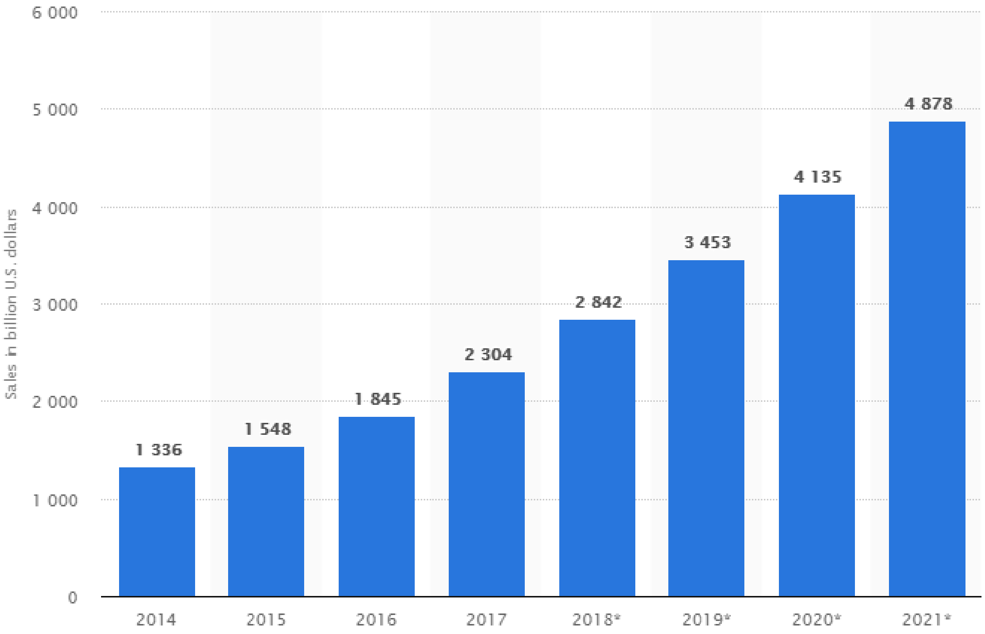


Figure 1.0

Online retail stores have abundance of data to play with, to better understand customers purchasing trends and habits in order to project the future positioning of the business in the market. With more retail stores offering their services online, the competition has evolved and becoming more aggressively challenging.

An average consumer is bombarded with over 3000 marketing messages every day causing a severe information overload. Hence, more consumers are choosing the option to opting out of receiving marketing ads. These consumers are throwing out direct mails, email notifications and video alerts that doesn't cater to their personal needs and wants. Yet, companies spend billions of dollars on marketing campaigns that yield little results.

Objective and Methodology

The objective of this analysis is to develop a model that clusters customers into different groups that can be utilized to apply customized marketing strategies to increase revenue. Directing marketing budget towards customers that will increase the return on investment. Our goal is to identify sunk costs of marketing to customers that won't spend online regardless of the amount of marketing strategy used to increase revenue. The step by step research methodology is outlined below:

1. Understanding the data structure, it's distribution and correlations
2. Developing three new Key Principle Variables
3. Developing a model with all relevant variables
4. Evaluating Model Performance
5. Recommending deployment strategy and its respected benefits for the company.

Data Understanding

The data set is multivariate sequential. There are 541909 instances with 8 real / integer attributes. The attributes are listed below:

1. Invoice No: Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.
2. Stock Code: Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
3. Description: Product (item) name. Nominal.
4. Quantity: The quantities of each product (item) per transaction. Numeric.
5. Invoice Date: Invoice Date and time. Numeric, the day and time when each transaction was generated.
6. Unit Price: Unit price. Numeric, Product price per unit in sterling.
7. Customer ID: Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.
8. Country: Country name. Nominal, the name of the country where each customer resides.

The following code snippet was used to read the data into R, and to obtain the structure of the data such as number of variables and rows, and the types of the variables as read into R:

# Read data into R
raw.data <- read.csv(file.choose(), header = TRUE, stringsAsFactors = FALSE, na.strings = c("NA","","#NA"))

data <- raw.data #keep raw.data as a backup

```
> head(data,5)
  InvoiceNo StockCode                        Description Quantity    Invoice
Date UnitPrice CustomerID        Country
1    536365    85123A  WHITE HANGING HEART T-LIGHT HOLDER        6 12/1/2010
8:26      2.55      17850 United Kingdom
2    536365     71053                 WHITE METAL LANTERN        6 12/1/2010
8:26      3.39      17850 United Kingdom
3    536365    84406B      CREAM CUPID HEARTS COAT HANGER        8 12/1/2010
8:26      2.75      17850 United Kingdom
4    536365    84029G KNITTED UNION FLAG HOT WATER BOTTLE        6 12/1/2010
8:26      3.39      17850 United Kingdom
5    536365    84029E      RED WOOLLY HOTTIE WHITE HEART.        6 12/1/2010
8:26      3.39      17850 United Kingdom
```

```
> str(data)
'data.frame':	541909 obs. of  8 variables:
 $ InvoiceNo  : chr  "536365" "536365" "536365" "536365" ...
 $ StockCode  : chr  "85123A" "71053" "84406B" "84029G" ...
 $ Description: chr  "WHITE HANGING HEART T-LIGHT HOLDER" "WHITE METAL LANTER
N" "CREAM CUPID HEARTS COAT HANGER" "KNITTED UNION FLAG HOT WATER BOTTLE" ...
 $ Quantity   : int  6 6 8 6 6 2 6 6 6 32 ...
 $ InvoiceDate: chr  "12/1/2010 8:26" "12/1/2010 8:26" "12/1/2010 8:26" "12/1
/2010 8:26" ...
 $ UnitPrice  : num  2.55 3.39 2.75 3.39 3.39 7.65 4.25 1.85 1.85 1.69 ...
 $ CustomerID : int  17850 17850 17850 17850 17850 17850 17850 17850 17850 13
047 ...
 $ Country    : chr  "United Kingdom" "United Kingdom" "United Kingdom" "Unit
ed Kingdom" ...
```

Data Description and Exploration

As with any data modeling exercise, the first step is to explore the data in order to understand the type of data one is working with. Which is to know the variable types, the unique values for categorical types, the summary stats of the numerical types (e.g. minimum, mean, median, maximum, standard deviation, range, etc.), the date range of any variables, and lastly if there are any missing values, outliers or wrongly coded values.

```
> # Missing values
> colSums(is.na(data))
  InvoiceNo   StockCode Description    Quantity InvoiceDate   UnitPrice  Cust
omerID     Country
          0           0        1509           0           0           0
135080           0
```

The above code snippet and results show that the “Description” and “CustomerID” fields had 1,509 and 135,080 missing values, respectively.

```
> length(unique(data$StockCode))
```

```
[1] 4070
> length(unique(data$Description))
[1] 4221
> length(unique(data$CustomerID))
[1] 4373
> length(unique(data$Country))
[1] 38
```

There are over 4,000 unique values each of StockCode, Description and CustomerID, and 38 different countries as shown above.

```
> summary(data$Quantity)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-80995.00      1.00      3.00      9.55     10.00  80995.00
> summary(data$UnitPrice)
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-11062.06      1.25      2.08      4.61      4.13  38970.00
> table(data$Country)

           Australia              Austria              Bahrain             B
elgium               Brazil               Canada
                1259                  401                   19
2069                   32                  151
     Channel Islands               Cyprus       Czech Republic             D
enmark                 EIRE   European Community
                 758                  622                   30
389                 8196                   61
             Finland               France              Germany
Greece            Hong Kong              Iceland
                 695                 8557                 9495
146                  288                  182
              Israel                Italy                Japan             L
ebanon            Lithuania                Malta
                 297                  803                  358
45                   35                  127
         Netherlands               Norway               Poland            Po
rtugal                  RSA         Saudi Arabia
                2371                 1086                  341
1519                   58                   10
           Singapore                Spain               Sweden         Switz
erland United Arab Emirates       United Kingdom
                 229                 2533                  462
2002                   68               495478
         Unspecified                  USA
                 446                  291
```

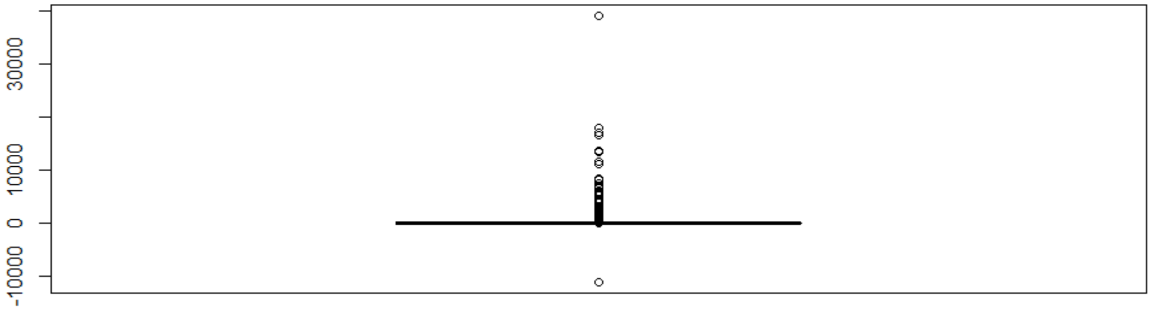
The number of items purchased (quantity) range from -80,995 to 80,995 where negative quantities were treated as returns and cancellations. UnitPrice of items fell in the range -11,062.06 to 38,970, and negative numbers were treated in the same way as quantities.
The distribution of the Country column shows that the number of customers are primarily from the United Kingdom, with 495,478 out of the 541,909 total number, representing 91.4%.
Since the InvoiceDate was a character field, additional work needed to be done to convert it to a proper and usable date format.

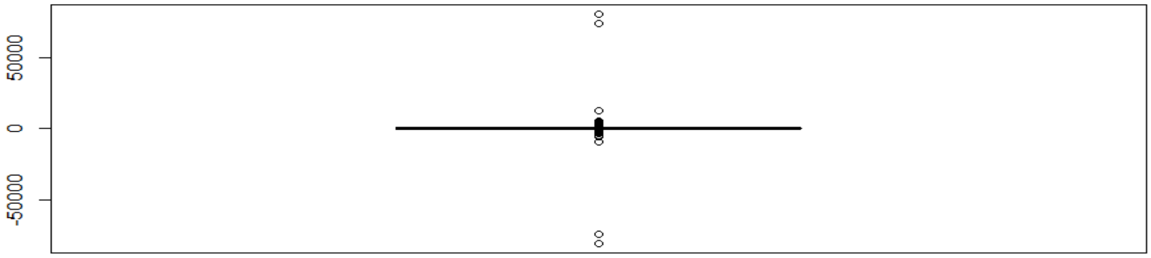
Modeling & Interpretation

Initial Plots (Charts)

Histogram for numeric variables, bar charts for factor variables, and scatter plots to explore relationship between the original variables were not useful due to the nature of the original variables. However, a box plot was obtained for our two numeric variables, Unit Price and Quantity.

```
> boxplot(data$UnitPrice) # we can use ylim=c(0,100) to zoom in
> boxplot(data$Quantity)
```





It is interesting to see that there fewer negative values of Unit Price than Quantity.

Further Data Manipulation

Convert Customer ID to Factor and InvoiceDate to date with MM/DD/YYYY format:

```
> # Convert CustomerID from Integer to Factor
> data$CustomerID <- as.factor(data$CustomerID)
>
> # Convert InvoiceDate to Date
> #data$InvoiceDate <- as.Date(data$InvoiceDate, "%m/%d/%y") #Outputs wrong d
ates
> #str(data)
>
> data <- transform(data, lapply({l<-list(InvoiceDate);names(l)=c('InvoiceDat
e');l},
+                                function(x)do.call(rbind, strsplit(x, ' ', fi
xed=TRUE))), stringsAsFactors=F)
>
> #str(data)
```

```
>
> to_drop1 <- c("InvoiceDate","InvoiceDate.1.1", "InvoiceDate.2", "InvoiceDat
e.2.1")
> data <- data[, !(names(data) %in% to_drop1)]
> library(lubridate)
> data$InvoiceDate <- mdy(data$InvoiceDate.1)
> to_drop2 <- c("InvoiceDate.1")
> data <- data[, !(names(data) %in% to_drop2)]
> str(data)
'data.frame':	541909 obs. of  8 variables:
 $ InvoiceNo  : chr  "536365" "536365" "536365" "536365" ...
 $ StockCode  : chr  "85123A" "71053" "84406B" "84029G" ...
 $ Description: chr  "WHITE HANGING HEART T-LIGHT HOLDER" "WHITE METAL LANTER
N" "CREAM CUPID HEARTS COAT HANGER" "KNITTED UNION FLAG HOT WATER BOTTLE" ...
 $ Quantity   : int  6 6 8 6 6 2 6 6 6 32 ...
 $ UnitPrice  : num  2.55 3.39 2.75 3.39 3.39 7.65 4.25 1.85 1.85 1.69 ...
 $ CustomerID : Factor w/ 4372 levels "12346","12347",..: 4049 4049 4049 4049
4049 4049 4049 4049 4049 541 ...
 $ Country    : chr  "United Kingdom" "United Kingdom" "United Kingdom" "Unit
ed Kingdom" ...
 $ InvoiceDate: Date, format: "2010-12-01" "2010-12-01" "2010-12-01" "2010-12
-01" ...
```

```
> range(data$InvoiceDate)
[1] "2010-12-01" "2011-12-09"
```

The date range indicates that we have a little over one year's worth of data.

Preprocessing Data

With the number of variables that we have, and in line with our main goal of customer segmentation, we sought to use clustering algorithms to achieve our objective. The two most common clustering algorithms are Hierarchical Clustering (HC) and K-Means Clustering (KC). While HC can work with both numerical and factor variable types, KC requires continuous variables, and they both work best when:

- There are no missing values in the data – they must be removed or estimated.
- The data is standardized (i.e., scaled) to make variables comparable. Standardization consists of transforming the variables such that they have a mean of zero and standard deviation as one. Standardizing the input variables is quite important; otherwise, input variables with larger variances will have commensurately greater influence on the results.

We have already discussed that missing CustomerID values were removed, and the three RFM variables have been transformed using the log functions. Next, we transformed our three input variables to reduce positive skew and then standardize them as z-scores. Below is a snippet of the code:

```
> # 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
> 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=TR
UE)
```

* Methodology/Approach

Selected Approach and Data Enhancement

Upon further deliberation, the group decided that it was best to do a segmentation analysis at the customer level since the results will be more meaningful, practical and useful for the UK-based company. Further research revealed several parameters that are typically used for customer segmentation. Three segmentations calculated from our original data are Recency, Frequency and Monetary Value (RFM).

With customer-level segmentation, the variable we will use to the profiling will be CustomerID, which had 135, 080 missing values out of the 541,909 total values, representing approximately 25% missingness. While we considered imputing the missing values using a non-parametric algorithm called k-nearest-neighbors (KNN), the group decided it was best to remove the missing rows of CustomerID and to work with the remaining 75% (approximately 406,820 rows or data points) in order preserve the information in the original data and not introduce any further bias.

```
> length(unique(data$CustomerID))
[1] 4373
> sum(is.na(data$CustomerID))
[1] 135080
> data <- subset(data, !is.na(data$CustomerID))
```

Since this was a UK-based company, with about 92% percent of the purchases made by customers from the UK, the group decided to focus on UK customers, thus a subset of the data with only UK customers was created, and unique customers decreased to 3,950 (from the initial 4,373) with 19,857 of unique invoices from final total number of rows of 361,878.

```
> data <- subset(data, Country == "United Kingdom")
> length(unique(data$CustomerID))
[1] 3950
```

RFM Variables

As stated above, since this is customer level segmentation, we needed to add three new features that is typically used to segment customers – namely Recency, Frequency and Monetary Value, (i.e. RFM). The recency variable refers 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 during the year. Monetary value is the amount that the customer spent during the year. Some customers have negative monetary values. These customers probably returned something during the year that they had purchased before the year started, so I reset their monetary value to zero.

To calculate the recency and frequency variables, it will be necessary to distinguish invoices related to actual purchases from invoices related to returns, and the “grepl” function in R was used as follows:

```
> data$item.return <- grepl("C", data$InvoiceNo, fixed=TRUE)
> data$purchase.invoice <- ifelse(data$item.return=="TRUE", 0, 1)
> head(data)
```

```
  InvoiceNo StockCode                         Description Quantity UnitPrice
CustomerID        Country InvoiceDate item.return
1    536365    85123A  WHITE HANGING HEART T-LIGHT HOLDER        6      2.55
17850 United Kingdom  2010-12-01       FALSE
2    536365     71053                 WHITE METAL LANTERN        6      3.39
17850 United Kingdom  2010-12-01       FALSE
3    536365    84406B      CREAM CUPID HEARTS COAT HANGER        8      2.75
17850 United Kingdom  2010-12-01       FALSE
4    536365    84029G KNITTED UNION FLAG HOT WATER BOTTLE        6      3.39
17850 United Kingdom  2010-12-01       FALSE
5    536365    84029E      RED WOOLLY HOTTIE WHITE HEART.        6      3.39
17850 United Kingdom  2010-12-01       FALSE
6    536365     22752        SET 7 BABUSHKA NESTING BOXES        2      7.65
17850 United Kingdom  2010-12-01       FALSE
  purchase.invoice
1                1
2                1
3                1
4                1
5                1
6                1
```

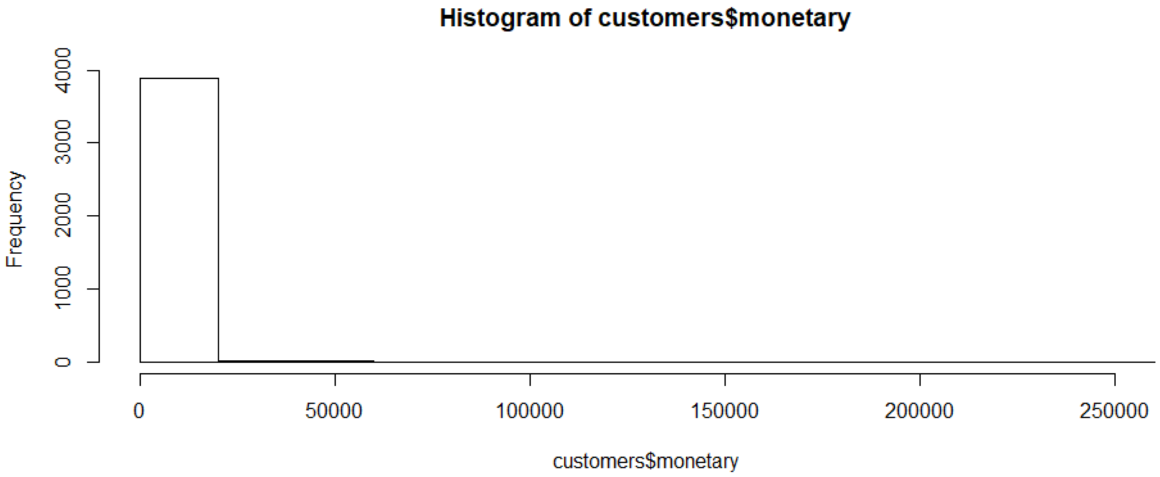
The subsequent code snippets were then used to compute the RFM variables.

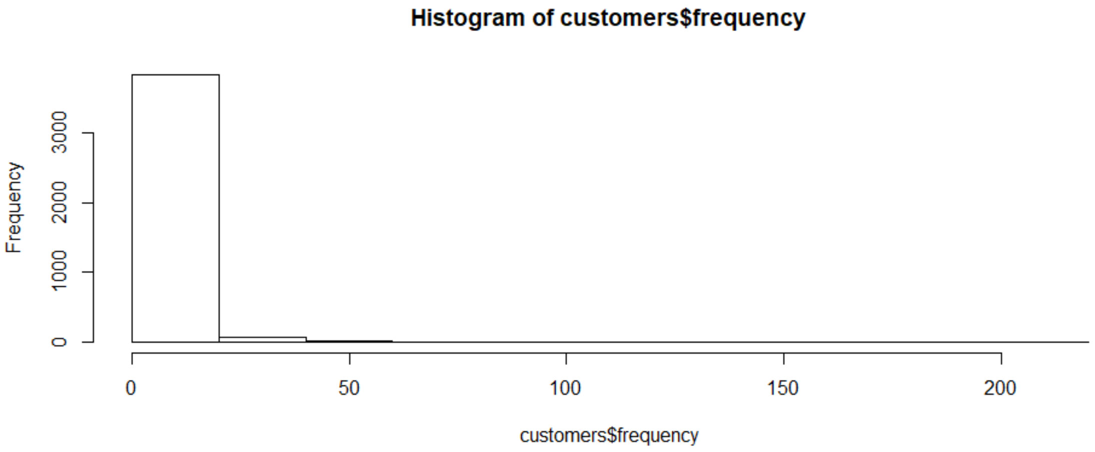
```
> ################################
> # Create customer-level dataset #
> ################################
>
> # create a customer-level dataset and add recency, frequency, and monetary
value data to it.
>
> customers <- as.data.frame(unique(data$CustomerID))
> names(customers) <- "CustomerID"
>
>
> ###########
> # Recency #
> ###########
>
> #data$recency <- as.Date("2011-12-10") - as.Date(data$InvoiceDate)
> data$recency <- as.Date("2011-12-10") - data$InvoiceDate
>
> # remove returns so only consider the data of most recent *purchase*
> temp <- subset(data, purchase.invoice == 1)
>
> # Obtain # of days since most recent purchase
> recency <- aggregate(recency ~ CustomerID, data=temp, FUN=min, na.rm=TRUE)
>
> # Add recency to customer data
> customers <- merge(customers, recency, by="CustomerID", all=TRUE, sort=TRUE
)
>
> customers$recency <- as.numeric(customers$recency)
```

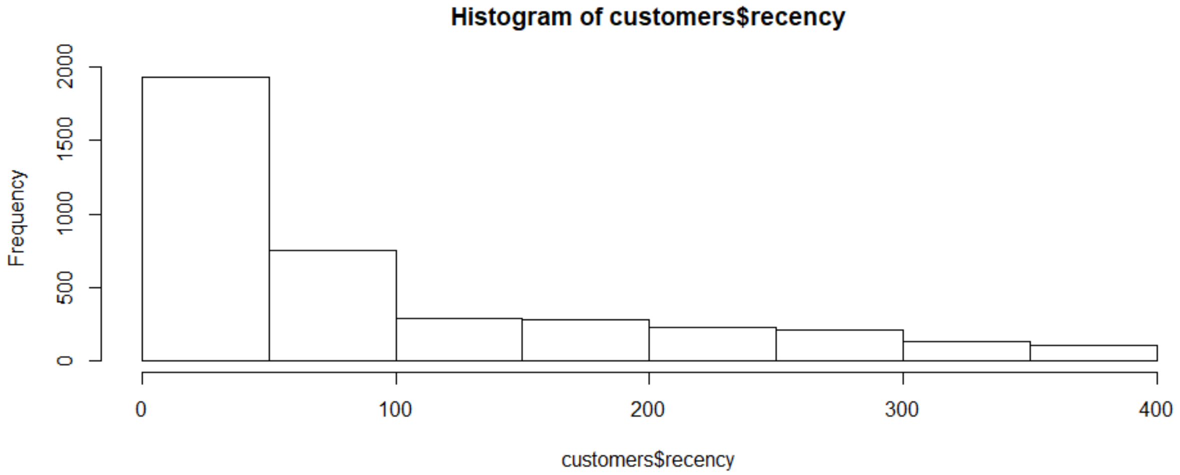
```
>
>
> #############
> # Frequency #
> #############
>
> customer.invoices <- subset(data, select = c("CustomerID","InvoiceNo", "pur
chase.invoice"))
> customer.invoices <- customer.invoices[!duplicated(customer.invoices), ]
> customer.invoices <- customer.invoices[order(customer.invoices$CustomerID),
]
> row.names(customer.invoices) <- NULL
>
> # Number of invoices (purchases only)
> annual.invoices <- aggregate(purchase.invoice ~ CustomerID, data=customer.i
nvoices, FUN=sum, na.rm=TRUE)
> names(annual.invoices)[names(annual.invoices)=="purchase.invoice"] <- "freq
uency"
>
> # Add # of invoices to customers data
> customers <- merge(customers, annual.invoices, by="CustomerID", all=TRUE, s
ort=TRUE)
>
> range(customers$frequency)
[1]   0 210
> table(customers$frequency)

   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14 
15   16   17   18   19   20   21   22   23   24   25   26 
  29 1351  746  465  351  217  158  130   87   61   46   48   41   28   20 
23   11   17   12   12   11   10    4    5    3    7    7 
  27   28   29   30   31   32   33   34   35   37   38   39   41   44   45 
46   47   48   50   51   55   57   60   62   63   86   91 
   3    6    1    3    2    2    2    3    1    3    2    2    1    1    1 
1    2    1    1    1    1    1    1    1    1    1    1 
  93   97  124  210 
   1    1    1    1 
>
> # Remove customers who have not made any purchases in the past year
> customers <- subset(customers, frequency > 0)
>
>
> ###############################
> # Monetary Value of Customers #
> ###############################
>
> # Total spent on each item on an invoice
> data$Amount <- data$Quantity * data$UnitPrice
>
> # Aggregated total sales to customer
> annual.sales <- aggregate(Amount ~ CustomerID, data=data, FUN=sum, na.rm=TR
UE)
> names(annual.sales)[names(annual.sales)=="Amount"] <- "monetary"
>
> # Add monetary value to customers dataset
> customers <- merge(customers, annual.sales, by="CustomerID", all.x=TRUE, so
rt=TRUE)
```

We then identified negative monetary values in the customers dataset and reset them to zero, after which obtained the summary statistics and made histogram plots of the RFM variables to see their distribution as follows:

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

> summary(customers$monetary)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
     0.0    289.8    633.7   1729.1   1530.8 256438.5 
> summary(customers$frequency)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  1.000   1.000   2.000   4.246   5.000 210.000 
> summary(customers$recency)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   1.00   18.00   51.00   92.72  143.00  374.00 

> hist(customers$monetary)
> hist(customers$frequency)
> hist(customers$recency)
```

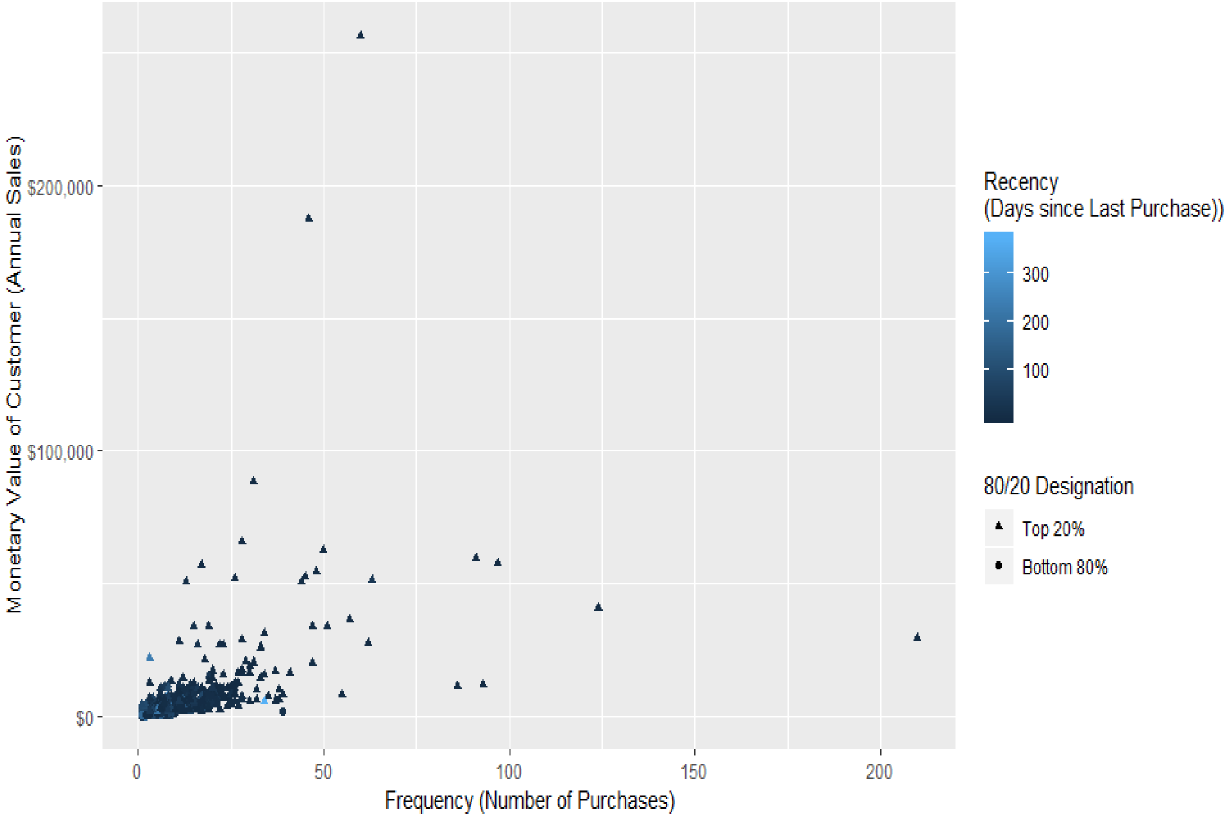


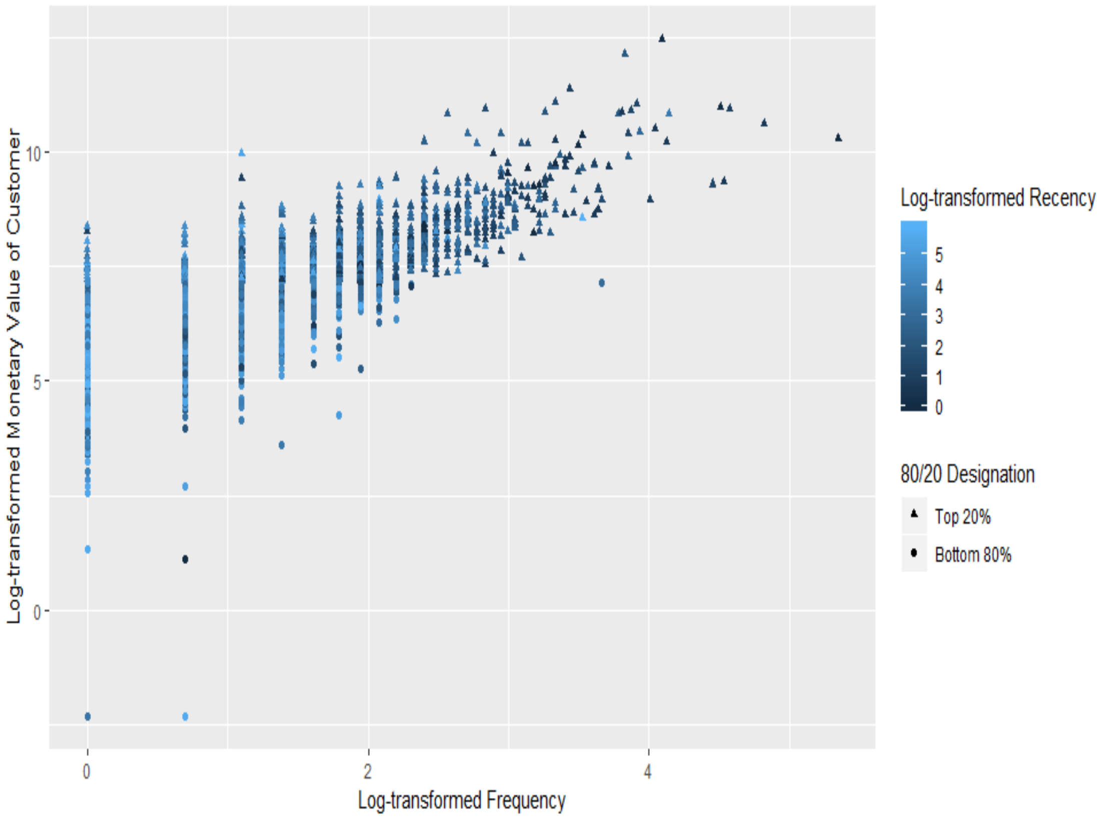


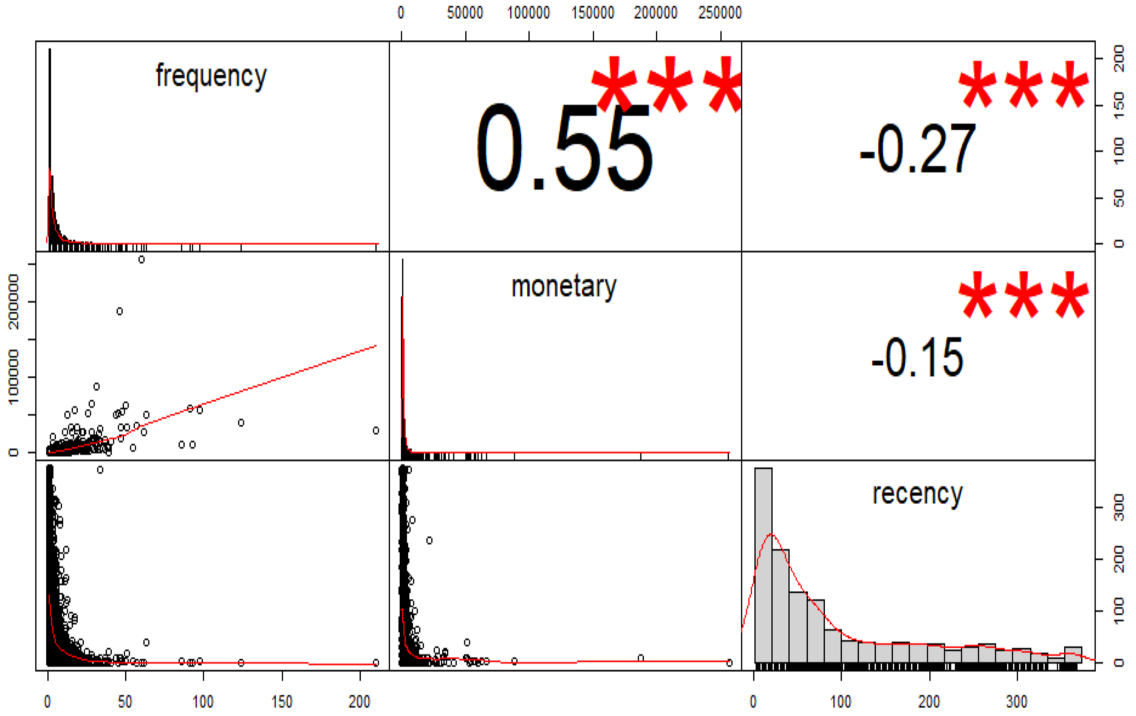


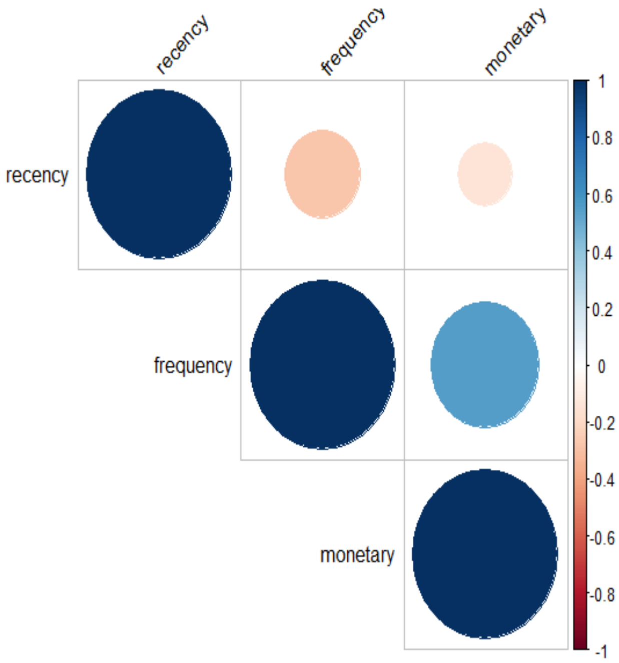
The Histograms (frequency plots) indicate that the three variables are highly skewed, with few positive extreme values. The main observation or interpretation is that the company has majority customers making total purchases less than $25,000, most of which are less than 100 days as of December 9, 2011, and several times for most purchases less than 25.

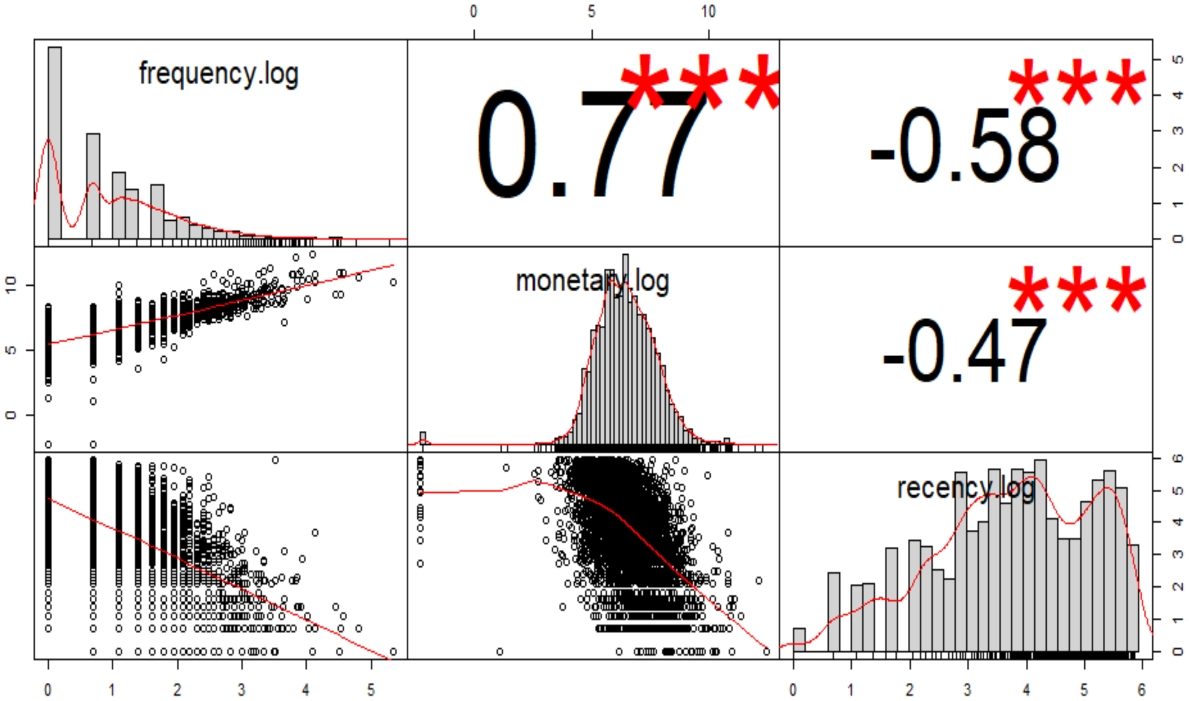
We employed scatter plots and correlations to further explore the relationships among the three variables as well as their log-transformed values. Here are few selected plots for comparative analysis:

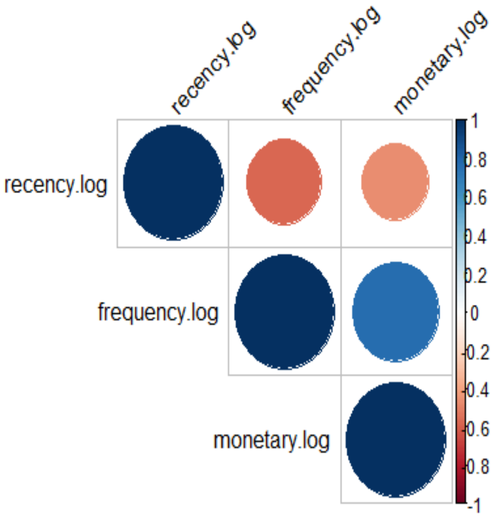



Log-transformed Monetary Value of Customer
10
5
0
0
2
4
Log-transformed Frequency
Log-transformed Recency
5
4
3
2
1
0
80/20 Designation
▲ Top 20%
● Bottom 80%


0
50000
100000
150000
200000
250000
frequency
0.55
***
-0.27
***
200
150
100
50
0
200000
100000
0
monetary
-0.15
***
recency
300
200
100
0
0
50
100
150
200
0
100
200
300


recency
frequency
monetary
recency
frequency
monetary
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1





Observation and Plot Interpretation

The scatter plot with the original RFM metrics are uninterpretable. There's a clump of data points in the lower left-hand corner of the plot, and a few outliers. This is why we log-transformed the input variables. In the scatter plot with the log-transformed variables, we can now 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. Importantly, we can see that the data points are fairly continuously-distributed.

There really aren't clear clusters. This means that any cluster groupings we create won't exactly reflect some true, underlying group membership - they'll be somewhat arbitrary (albeit reasonable) distinctions that we draw for our own purposes.

Thus, it is obvious that the log transformation provides some key advantages such as:

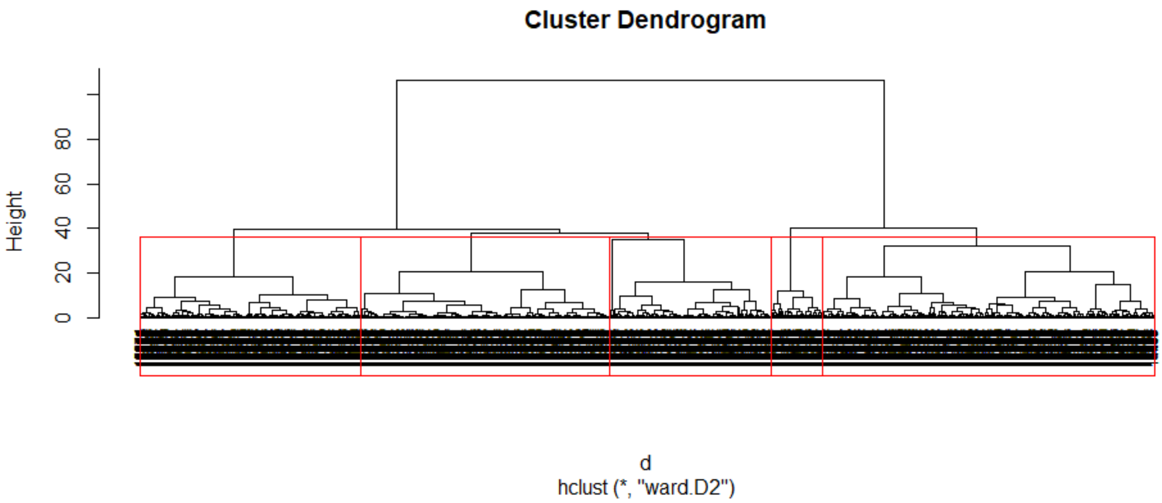
1. It allows us to work with the three variables on comparable scale
2. It amplifies linear relationship between the variables, especially the positive linear relationship (high +ve correlation) between log.frequency and log.monetary value.
3. The positive correlation between frequency and monetary value means the two variables rise and fall together, i.e. customers who made fewer number of purchases spent smaller amount while highly frequent customers spent higher amount. The negative correlation between monetary and recency (meaning higher value of one is associated with lower value of the other) is interpreted as follows: customers who made more recent purchases (lower recency values) spent higher amount than customers whose purchases are not relatively recent. Likewise for negative correlation between recency and frequency suggests that customers with more recent purchases (lower recency values) made a number of purchases many times more (higher frequency) than those who did not make recent purchases, relative speaking.
4. Based on these observations, even before moving forward, our initial hypothesis is that the company will be better served focusing on or targeting more recent customers since they seem to have higher purchase amount as a result of making several purchases.
5. It is very important to note that these observations are only limited to this data and in the date range but only not general observations and conclusions for other online retail problems.

Hierarchical Clustering

We performed agglomerative HC with `hclust`. First, we compute the dissimilarity values with `dist` and then feed these values into `hclust` and specify the agglomeration method to be used (i.e. “complete”, “average”, “single”, “ward.D”). Then, we can plot the dendrogram. Below is the full code and the dendogram.

```
> scaled_customers_2 <- customers[, c("CustomerID", "frequency.z", "monetary.
z", "recency.z")]
> #head(scaled_customers_2)
> d <- dist(scaled_customers_2, method = "euclidean") #dissimilarity (distanc
e) matrix
> h_clust <- hclust(d, method = "ward.D2") #clustering using Ward’s minimum v
ariance method
> h_clust
```

```
Call:
hclust(d = d, method = "ward.D2")

Cluster method   : ward.D2
Distance         : euclidean
Number of objects: 3921

>
> #Plot dendogram
> plot(h_clust, cex = 0.6, hang = -1, labels = customers$CustomerID)
> rect.hclust(h_clust,k=5)
>
>
> #extract clusters
> groups <- cutree(h_clust,k=5)
> table(groups)
groups
   1    2    3    4    5
 622 1283  199  962  855
```



We chose a 5-solution cluster which is shown in the output of the code (5 groups) and selected with rectangles drawn around them in the dendogram. We believe a 4- or 5-cluster solution is a reasonable number of clusters as very well shown the dendogram because a 2-cluster solution will be very simplistic and higher number of clusters would tend to overly complicate the delineation boundary and the interpretation.

Assessment of Ward Agglomeration Method

A number of different cluster agglomeration methods (i.e, linkage methods) have been developed to measure the dissimilarity between two clusters of observations. Each method computes all pairwise dissimilarities between the elements in cluster 1 and the elements in cluster 2. The most common types are:

a) **Maximum or complete linkage clustering** – which considers the largest (i.e., maximum value of the dissimilarities as the distance between the two clusters

b) **Minimum or single linkage clustering** – which uses the minimum value of dissimilarities

c) **Mean or average linkage clustering** – based on the mean value of dissimilarities

d) **Centroid linkage clustering -** computes the dissimilarity between the centroid for cluster 1 (a mean vector of length p variables) and the centroid for cluster 2.

e) **Ward's minimum variance method –** minimizes the total within-cluster variance. At each step the pair of clusters with minimum between-cluster distance are merged.

We assessed our choice of Ward's method using the "agnes" function as shown in the following code snippet and output. This allows us to find certain hierarchical clustering methods that can identify stronger clustering structures.

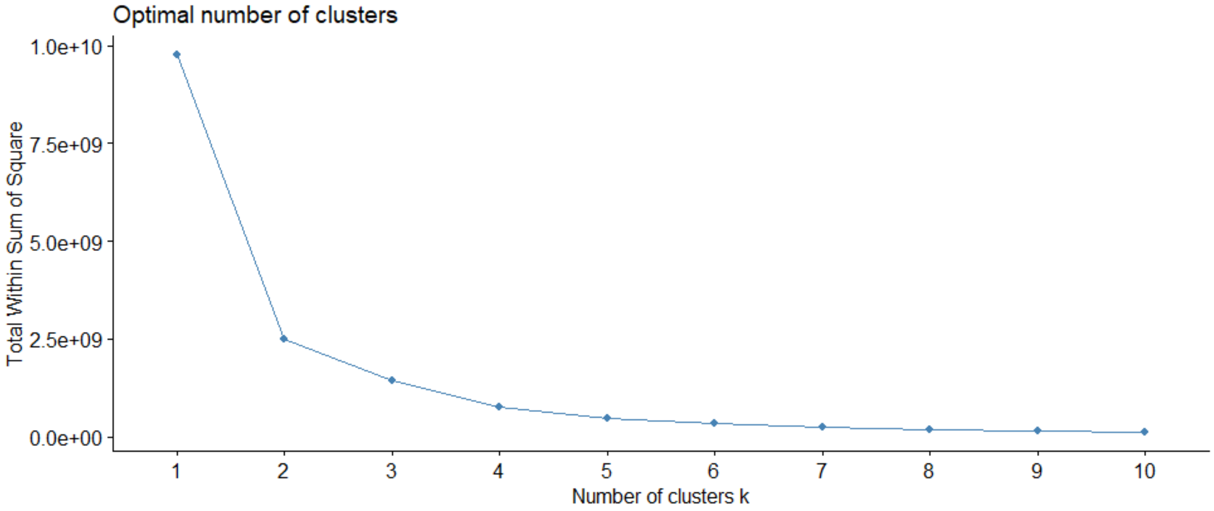
```
> # methods to assess
> m <- c( "average", "single", "complete", "ward")
> names(m) <- c( "average", "single", "complete", "ward")
>
> # function to compute coefficient
> ac <- function(x) {
+   agnes(scaled_customers_2, method = x)$ac
+ }
>
> map_dbl(m, ac)
  average    single  complete      ward 
0.9988746 0.9941975 0.9994960 0.9999746 
```

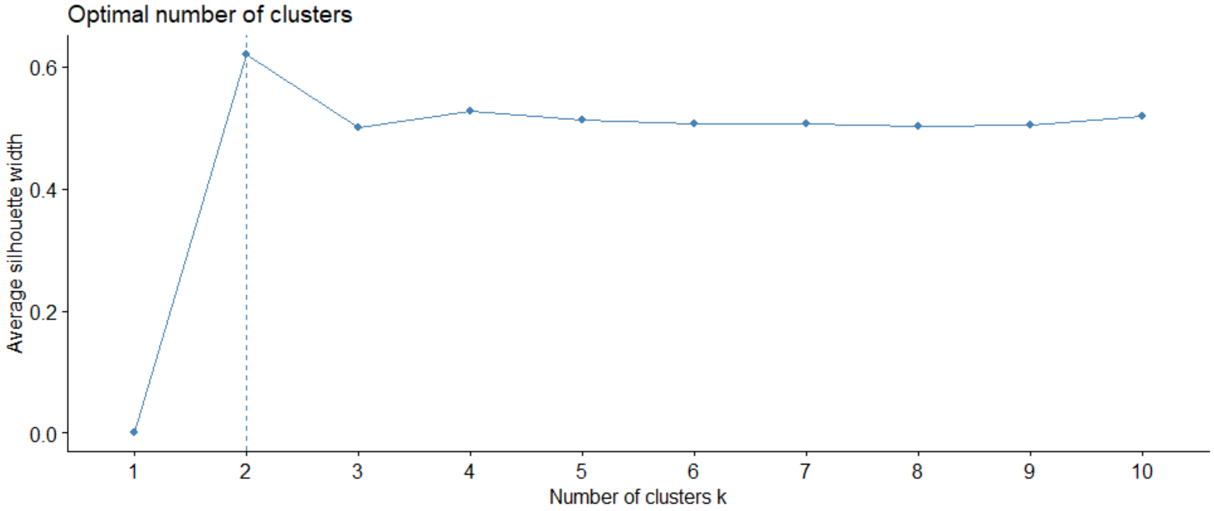
The output shows that Ward's method has the maximum value, and thus identifies the strongest clustering structure of the four methods assessed.

Determining Optimal Clusters (Hierarchical Clustering)

Two methods, the Elbow method and Average Silhouette method, were used to obtain two plots as shown below:

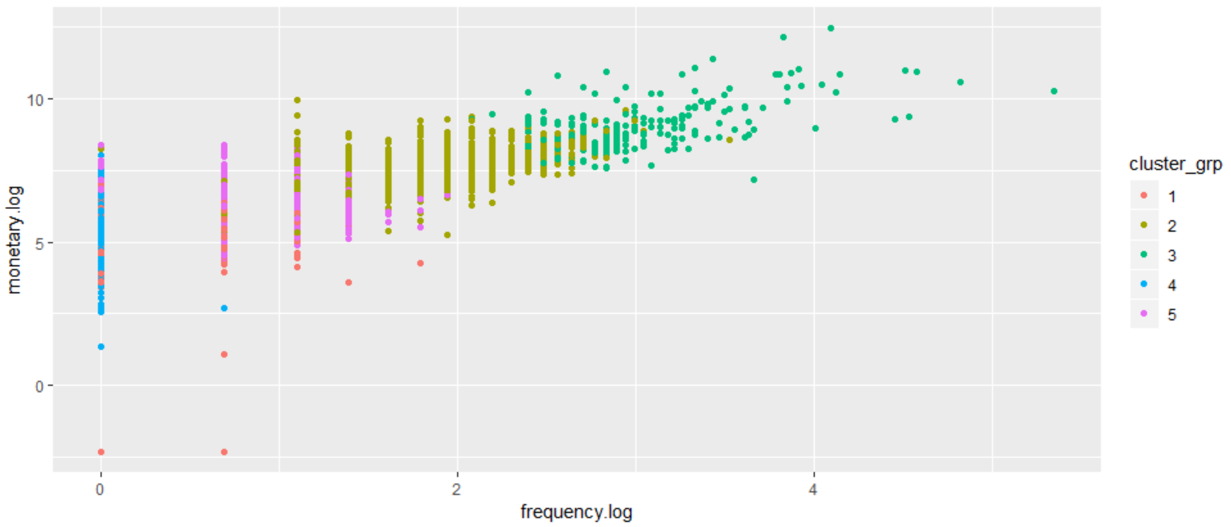
```
> #Determining Optimal Clusters
> # Use Elbow method
> fviz_nbclust(hc_customers, FUN = hcut, method = "wss")

> #Average Silhouette Method
> fviz_nbclust(hc_customers, FUN = hcut, method = "silhouette")
```

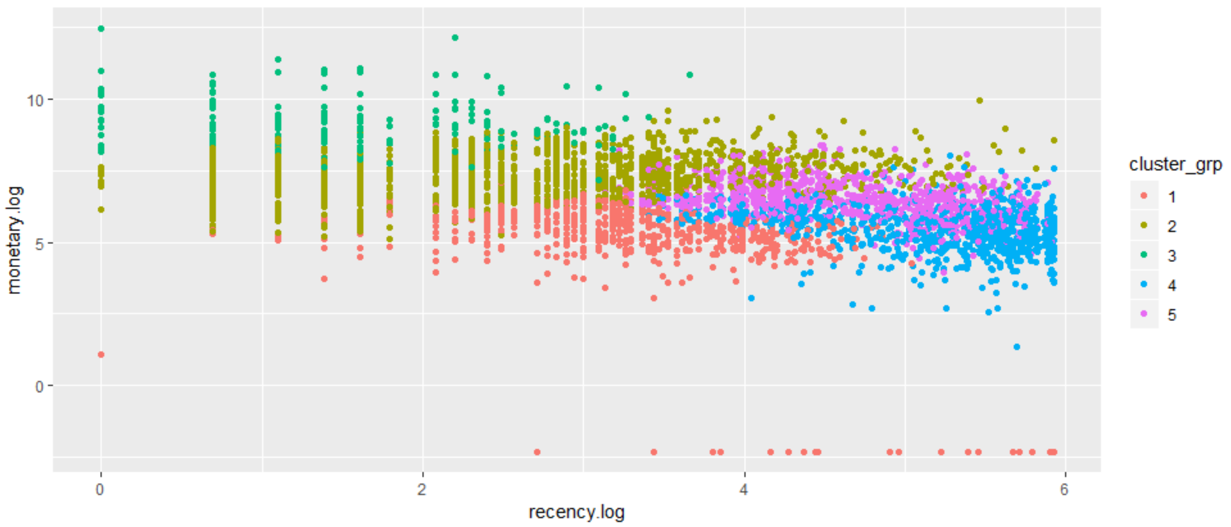




In the Elbow plot, we seek the cluster-solution with a minimal "within sum of squares" but below which the change doesn't come at a cost of complexity and interpretability. While in the Average Silhouette Method, the maximum value of the index is used to determine the optimal number of clusters in the data. While both plots make a strong case for a 2-cluster solution, we thought a 2-cluster solution is too simplistic, and that 4 or 5 clusters would be optimal and appropriate. We tried a third method, the Gap Statistic Method, that calculates a goodness of clustering measure, the "gap" statistic.

We tried visualizing the result in a scatter plot using the "fviz_cluster" function from the factoextra package but there was an error that we couldn't fix. Thus, a scatter plot of customers grouped by clusters was produced with ggplot2 for log-transformed variables - frequency versus monetary and recency versus monetary. The two plots are shown below:





K-Means Clustering

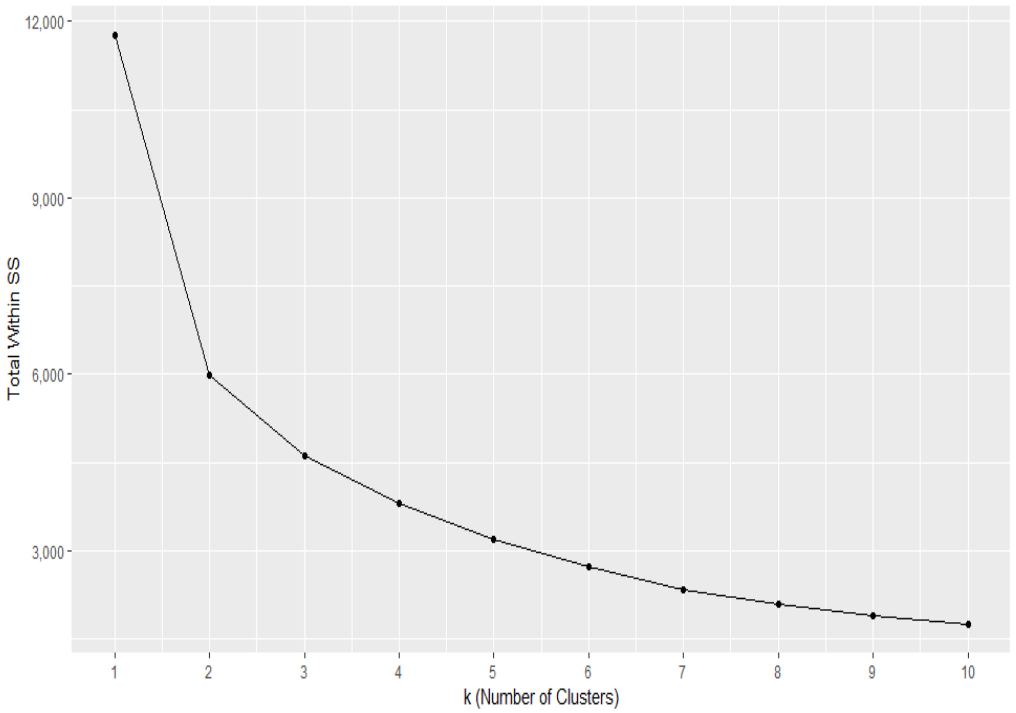
As with the hierarchical clustering, the standardized RFM variables were used in the k-means clustering algorithm. A loop was created to run the k-means analysis with increasing numbers of clusters, each time generating a graph of the clusters, the cluster centers for each model, and information about the variance explained. The information generated and stored or plotted were used to assist in selecting the optimal number of clusters. The code below was used:

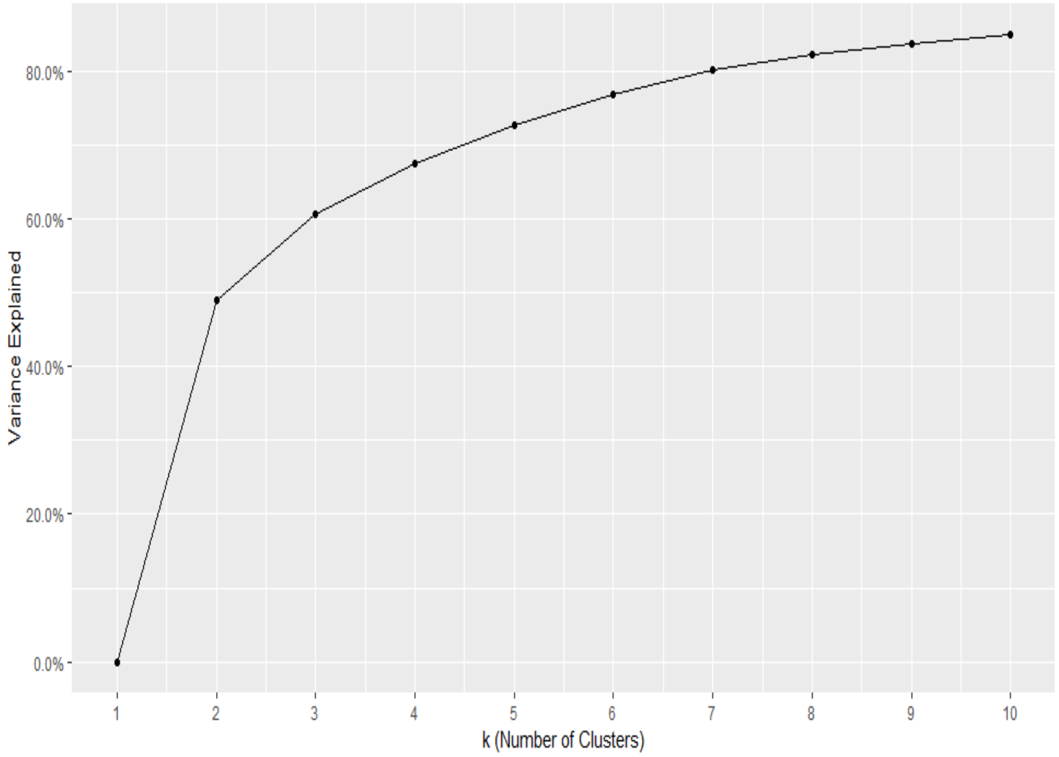
```
> preprocessed <- customers[,9:11]
> j <- 10 # specify the maximum number of clusters you want to try out
>
> models <- data.frame(k=integer(),
+                      tot.withinss=numeric(),
+                      betweenss=numeric(),
+                      totss=numeric(),
+                      rsquared=numeric())
>
> for (k in 1:j ) {
+
+   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.lo
g))
+   cluster_graph <- cluster_graph + geom_point(aes(colour = customers[,(var.
name)]))
+   colors <- c('red','orange','green3','deepskyblue','blue','darkorchid4','v
iolet','pink1','tan3','black')
+   cluster_graph <- cluster_graph + scale_colour_manual(name = "Cluster Grou
p", 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)
```

Determining Optimal Clusters (Hierarchical Clustering)

We used the Elbow method as in the HC. Below shows two plots – one for the “Total Within Variance (or Sum of Squares Errors)”, and the other the “Variance Explained” at each cluster solution as the number of clusters increases.





These graphs are two different ways of visualizing the same information - in both cases, we're looking for an "elbow" or bend in the graph beyond which additional clusters add little additional explanatory power. Both graphs look to have elbows at around 2 clusters, but a 2-cluster solution explains only 49% of the variance and, once again, a 2-cluster solution may be too much of a simplification to really help the business with targeted marketing.

Thus, we decided to go with a 5-cluster solution, which explains over 73% of the variance, even though there are no clear elbows in the graphs at this point. We sought to use additional metrics to evaluate and validate our selected cluster solution.

Validation of Optimal Cluster Solution

We used an R package to generate a host of different fit indices. Typically, a majority rule is used to suggest the number of clusters based on what most indices recommend. The code is shown below:

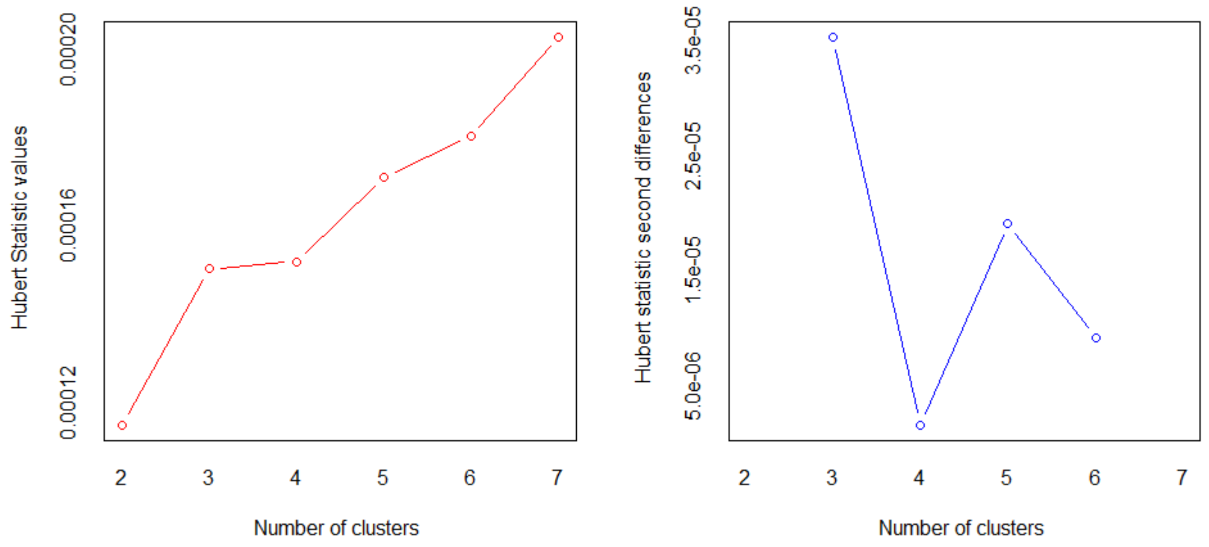
```
> library(NbClust)
> set.seed(1)
> nc <- NbClust(preprocessed, min.nc=2, max.nc=7, method="kmeans")
> table(nc$Best.n[1,])
>
> nc$All.index # estimates for each number of clusters on 26 different metric
s of model fit
> barplot(table(nc$Best.n[1,]),
+         xlab="Number of Clusters", ylab="Number of Criteria",
+         main="Number of Clusters Chosen by Criteria")
```

26 indices are generated. For the lack of space, two of the indices (statistics) are discussed below along with the output from the code.

The Hubert Index

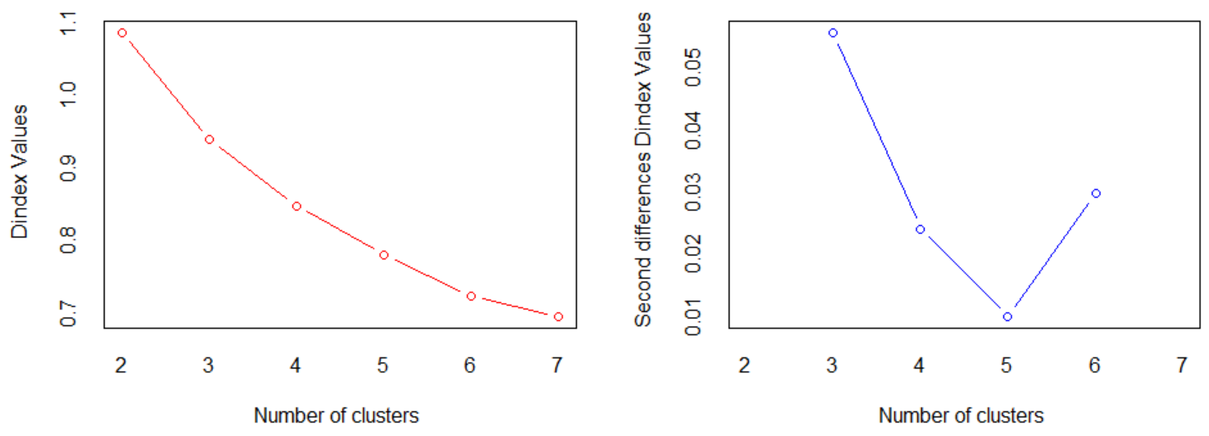
Hubert’s statistic is the point serial correlation coefficient between any two matrices. High values of normalized Hubert statistics indicate the existence of compact clusters. Thus, in the plot of normalized Hubert’s statistic versus the number of clusters, we seek a significant knee that corresponds to a significant increase of normalized Hubert’s statistic as the number of clusters varies from 2 to the maximum possible number of clusters. The number of clusters at which the knee occurs is an indication of the number of clusters that underlie the data.

In the NbClust package, second differences values of normalized Hubert’s statistics are plotted to help distinguish the knee from other anomalies. A significant peak in this plot indicates the optimal number of clusters.



The Dindex

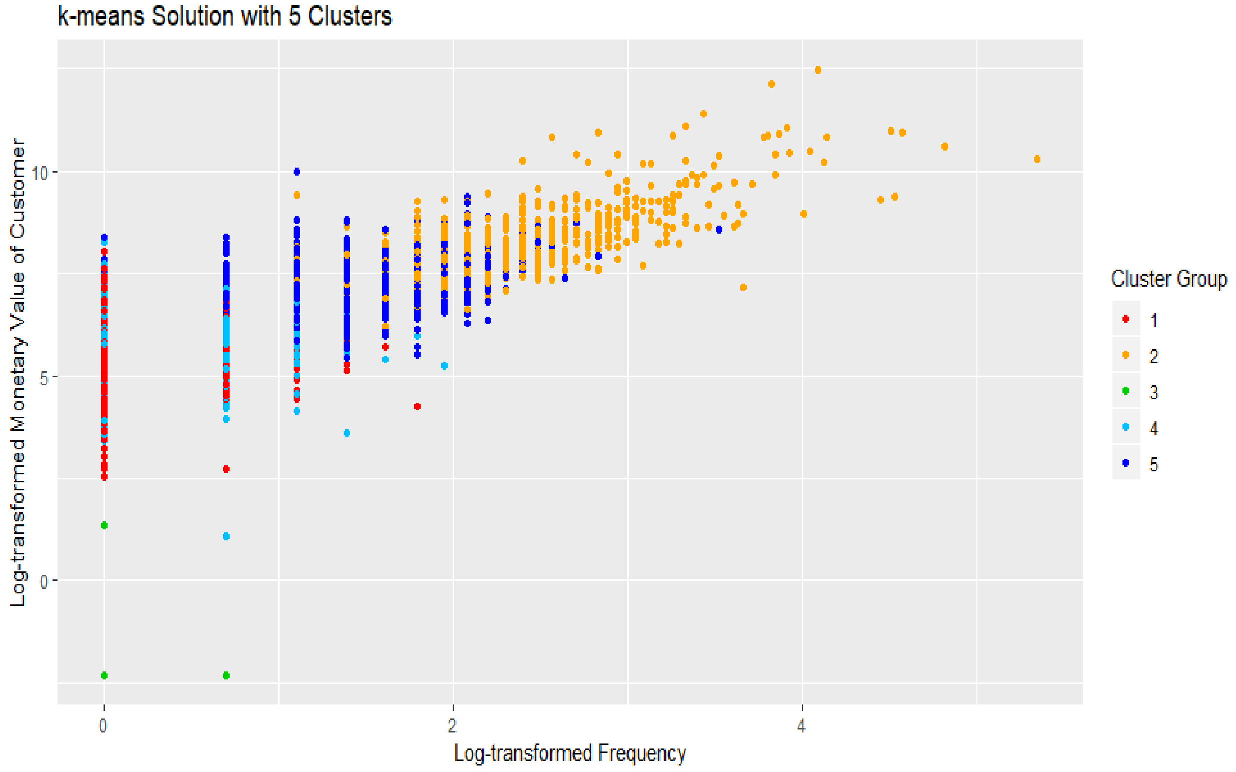
The Dindex is based on clustering gain on intra-cluster inertia which measures the degree of homogeneity between the data associated with a cluster. It calculates their distances compared to the reference point representing the profile of the cluster, i.e., the cluster centroid in general. As with the Hubert's statistics, the optimal cluster configuration can be identified by the sharp knee that corresponds to a significant decrease of the first differences of clustering gain versus the number of clusters. This knee or great jump of gain values can be identified by a significant peak in second differences of clustering gain.

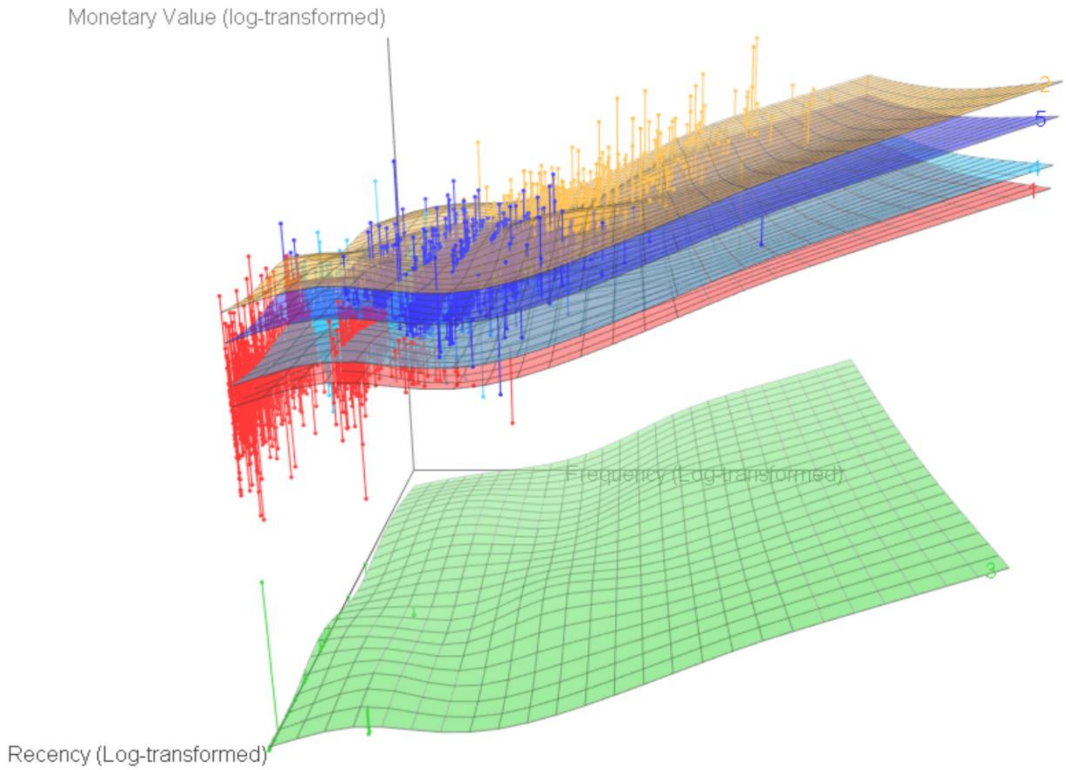


While the Hubert's statistic suggests a 5-cluster solution, the Dindex statistic recommends a 6-cluster solution. We settled on the 5-cluster solution because its interpretation will be simpler than the 6-cluster solution while also agreeing with the optimal 5-cluster solution in the hierarchical clustering.

Having selected a 5-cluster solution in the k-means clustering, we created 3-D scatter plots to visualize the clusters as well.

```
>
> scatter3d(x = customers$frequency.log,
+           y = customers$monetary.log,
+           z = customers$recency.log,
+           groups = customers$cluster_5,
+           xlab = "Frequency (Log-transformed)",
+           ylab = "Monetary Value (log-transformed)",
+           zlab = "Recency (Log-transformed)",
+           surface.col = colors,
+           axis.scales = FALSE,
+           surface = TRUE, # produces the horizonal planes through the graph
at each level of monetary value
+           fit = "smooth",
+           # ellipsoid = TRUE, # to graph ellipses uses this command and com
ment out "surface = TRUE"
+           grid = TRUE,
+           axis.col = c("black", "black", "black"))
```

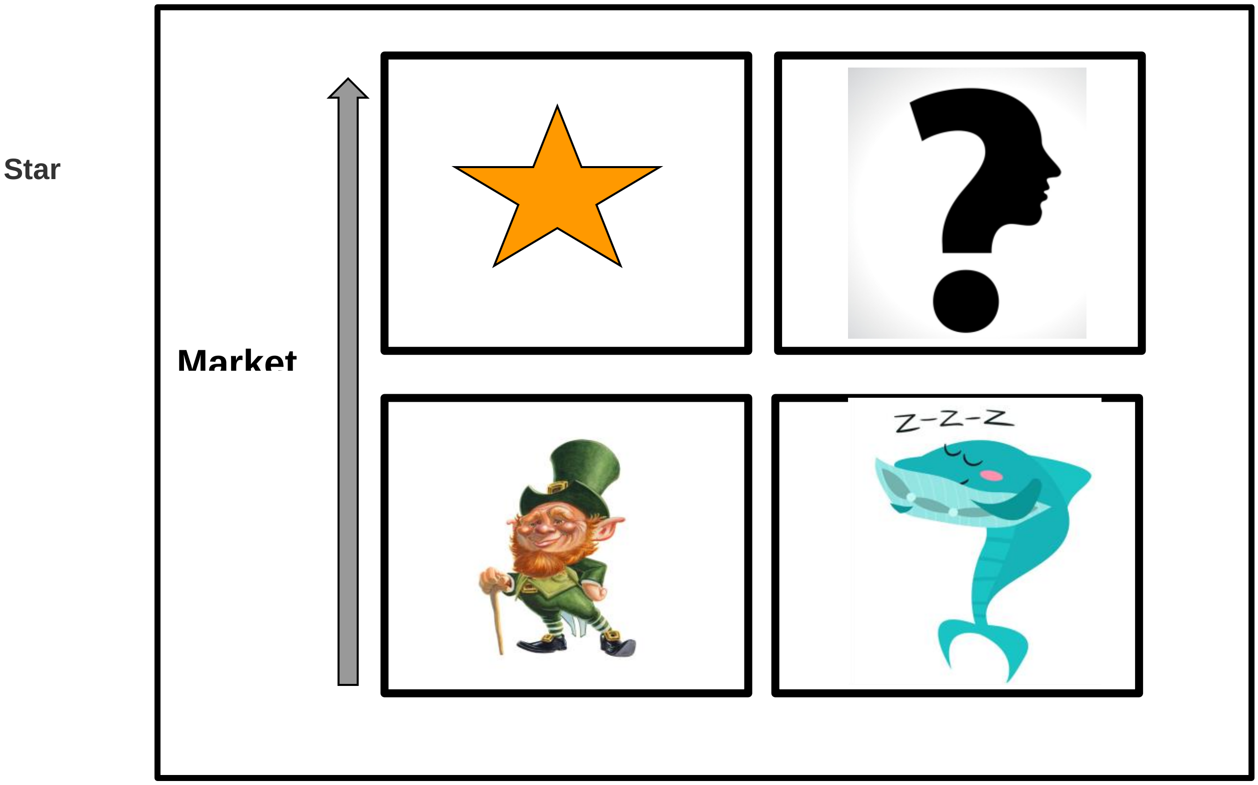




Deployment Strategy

Based on the results extracted from our analysis, we suggests that the company allocate their funds based on where the customer falls on the marketing matrix we have developed below in Figure 1.1. The matrix is designed to assist marketing strategists with long term strategic planning to help the UK based online retail store grow in the competitive online market. The matrix identifies the star consumers, the unsure consumers, the moneymaker leprechauns and the sleeping sharks.

Marketing
Investment Matrix



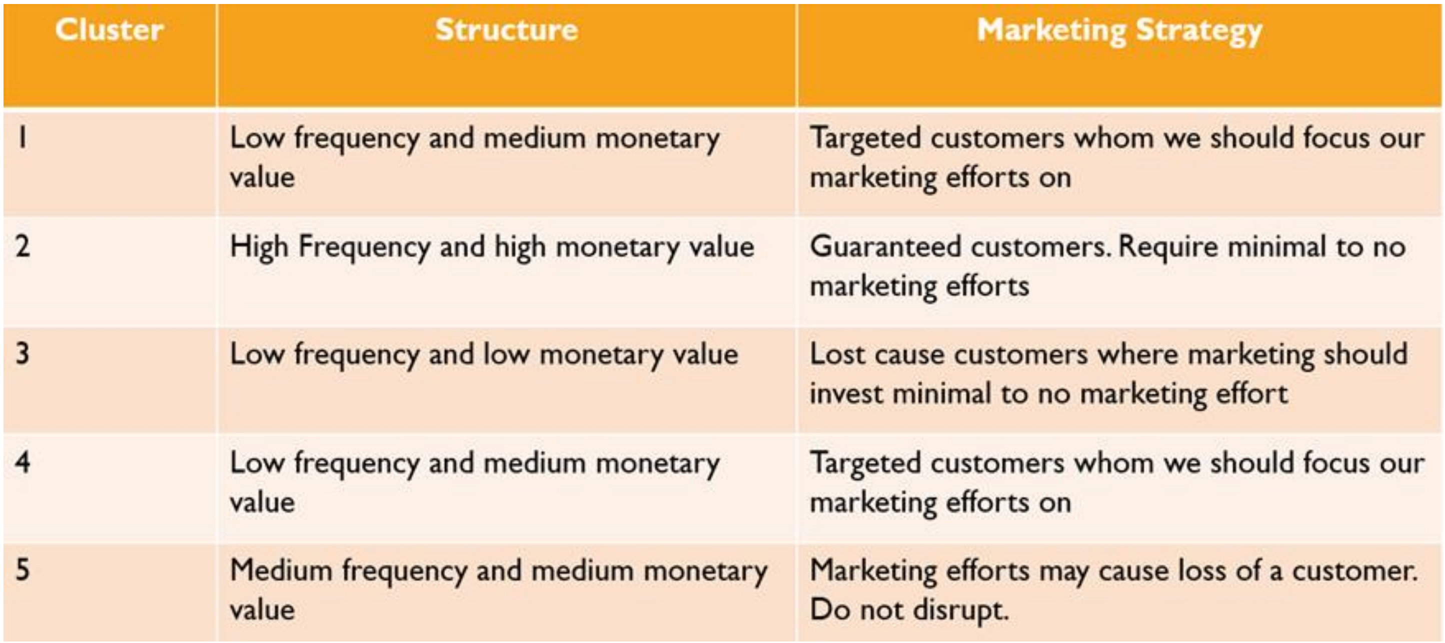
Consumers
Customer base that are loyal to the brand, have consistent purchasing habit and are not going anywhere anytime soon. These are guaranteed clients who will purchase products from the retail store regardless of marketing investment.

Unsure Consumers
Customer base that will either become a star or a sleeping shark. Usually require high investment but ROI is not guaranteed. This is where board of experts should truly evaluate a product to measure whether it would generate profit.

Leprechaun
Customer base that have resources and will purchase when Marketing efforts are made.

Sleeping Sharks
Customers that purchase bare minimum due to minimal financial resources, skepticism for online retail, etc. These group are the consumer base will react negatively the most if bombarded with marketing propagandas. Marketing force may result in losing all business or bad word of mouth recommendation. These are the sleeping sharks. Do Not Disturb!

Recommendations



Cluster	Structure	Marketing Strategy
1	Low frequency and medium monetary value	Targeted customers whom we should focus our marketing efforts on
2	High Frequency and high monetary value	Guaranteed customers. Require minimal to no marketing efforts
3	Low frequency and low monetary value	Lost cause customers where marketing should invest minimal to no marketing effort
4	Low frequency and medium monetary value	Targeted customers whom we should focus our marketing efforts on
5	Medium frequency and medium monetary value	Marketing efforts may cause loss of a customer. Do not disrupt.

Reference:

NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set,

Journal of Statistical Software, October 2014, Volume 61, Issue 6.
(*https://www.jstatsoft.org/article/view/v061i06/v61i06.pdf*)

Daqing Chen, Sai Liang Sain, and Kun Guo, Data mining for the online retail industry: A case study of RFM model-based customer segmentation using data mining, Journal of Database Marketing and Customer Strategy Management, Vol. 19, No. 3, pp. 197â€“208, 2012 (Published online before print: 27 August 2012. doi: 10.1057/dbm.2012.17).