|  |
| --- |
| Online Retail Cluster Analysis |
| MCDA 5580 Data Mining Assignment 1 Report |

|  |
| --- |
|  |

****

|  |  |
| --- | --- |
| ­­ | |
| Contents  [Executive Summary 2](#_Toc9607123)  [Data Summary 2](#_Toc9607124)  [Data Preparing 2](#_Toc9607125)  [Approach 4](#_Toc9607126)  [Attributes Selection 4](#_Toc9607127)  [Data Cleaning / Outlier Removal 6](#_Toc9607128)  [Cluster Analysis 9](#_Toc9607129)  [Cluster Profiling 13](#_Toc9607130)  [Conclusion & Next Steps 20](#_Toc9607131)  [Appendix 21](#_Toc9607132) | |

# Executive Summary

In this report, we are going to clustering analysis online retail dataset based on selected attributes to cluster customer and product in order to analysis the attribute of a different cluster of customer and product. We can use cluster profiling to perform further research on customer services and future marketing strategy.

# Data Summary

The online retail dataset is a table in a database which included the online retail store’s invoice transaction. The table has nine columns to store InvoiceNo (An unique number to identify order invoice), StockCode (An unique text to identify a product), Description (Detail description about the product), Quantity (The number of product in this transaction), InvoiceDate (The date and time when the transaction occurs), UnitPrice (Unit price of the product), CustomerID (The unique ID to identify a customer), Country (The nation which the customer residence), InvoiceDateTime (The date and time when the transaction occurs).

There is a total of 541909 transaction records in this table.



Figure 1 Data source

# Data Preparing

The first part of the analysis is clustering the customers based on the following attributes:

1. Number of products bought;
2. Number of distinct products bought;
3. Revenues;
4. Number of visits;
5. Average spend on each visit.
6. Quantity purchase per visit.

To prepare the data for customers cluster analysis we designed a SQL query to create a new table called customercluster which included the following column:

1. Count the Quantity for each customerID as ItemPurchased;
2. Count the distinct StockCode for each customerID as ItemVariety;
3. Sum the Quantity multiply by the UnitPrice for each customerID as Revenue;
4. Count the distinct InvoiceNo for each customerID as TotalVisit;
5. Sum the Quantity multiply by the UnitPrice divided by the count of distinct InvoiceNo for each customerID as RevenuePerVisit.
6. Sum of the quantity purchased divided by distinct InvoiceNo for each customerID as ItemPerVisit.

alter table onlineretail

add column TotalPrice double;

SET SQL\_SAFE\_UPDATES = 0;

update onlineretail set TotalPrice = Quantity \* UnitPrice;

select CustomerID,

sum(Quantity) as ItemPurchased,

count(distinct StockCode) as ItemVariety,

sum(TotalPrice) as Revenue,

count(distinct InvoiceNo) as TotalVisit,

sum(TotalPrice)/count(distinct InvoiceNo) as RevenuePerVisit,

sum(Quantity)/count(distinct InvoiceNo) as ItemPerVisit

from onlineretail

group by CustomerID;

The second part of the analysis is clustering the products based on the following attributes:

1. Number of distinct customers who buy the product;
2. Revenues;
3. Number of visits in which the product is bought;
4. Revenue per visit.

To prepare the data for products cluster analysis we designed another SQL query to create a new table called productcluster which included the following column:

1. Count distinct CustomerID for each StockCode as TotalCustomer;
2. Sum the TotalPrice for each StockCode as Revenue;
3. Count distinct InvoiceNo for each StockCode as TotalVisit;
4. Sum the TotalPrice divided by count distinct InvoiceNo for each StockCode as RevenuePerVisit.

select StockCode,

count(distinct CustomerID) as TotalCustomer,

sum(TotalPrice) as Revenue,

count(distinct InvoiceNo) as TotalVisit,

sum(TotalPrice)/count(distinct InvoiceNo) RevenuePerVisit

from onlineretail

group by StockCode;

# Approach

This report was designed to showcase the clustering analysis process for online retail dataset focus on products and customer and conclude the profile for each cluster.

We have reviewed the dataset to determine the data structure and understand each fields’ meaning, we have a total 541909 transaction to record online retail activity.

We have done the data preparing to extract the relative data from MySQL database table, using SQL query to calculate the data for our analysis.

Two more attributes we have selected for customer cluster which is the average spend on each visit and the quantity purchased per visit to make the attributes more meaningful for our cluster. We also selected one more attribute revenue per visit for our product cluster.

In the data cleaning and outlier removal part, we utilizing Open Refine to remove the following outlier: CustomerID is 0, 150 of customers are deemed as potential business customers. For the product part, we have removed 15 rows of products which are services related and not typical merchandise. Total 19 rows products which generate negative total revenue with negative values in purchased quantity we deemed it as abnormal products.  
Move to the cluster analysis part, we normalize data using the scale function. The withinSSrange function can calculate the sum of a square within the clusters with a different number of clusters, from the elbow angle of the plot, we determine 6 is the best number of the cluster for customer and 7 is the best number of the cluster for the product. In the end, we denormalized the data and visualized with the plot for cluster profiling.

# Attributes Selection

Except for existing attributes for customers and products, we have selected two more attributes for customers which is the average spends on each visit and the quantity purchase per visit. The average spends on each visit can measure the customer’s shopping habit, a high value for this field can show the customer has contributed a high revenue with less visit, a low value for this field can tell the customer visit more frequently but spend less. The quantity purchase per visit can use to measure the customer is intended to purchase limited items or purchase various items. We have also selected one more attributes for products which is the revenue per visit, this attribute can tell for each visit one product can contribute more or less revenue.

A close up of text on a black background

Description automatically generated

Figure 2 GGpairs for Customers

From figure 2 GGpairs for Customers, we can observe the linear relationship between Revenue and ItemPurchased which means customers with more item purchased normally will contribute more revenue to the company. On the other hand, those customers have more visit normally will purchase more items and products will also create more revenue, but there definitely has an exemption.

A black and silver text on a white background

Description automatically generated

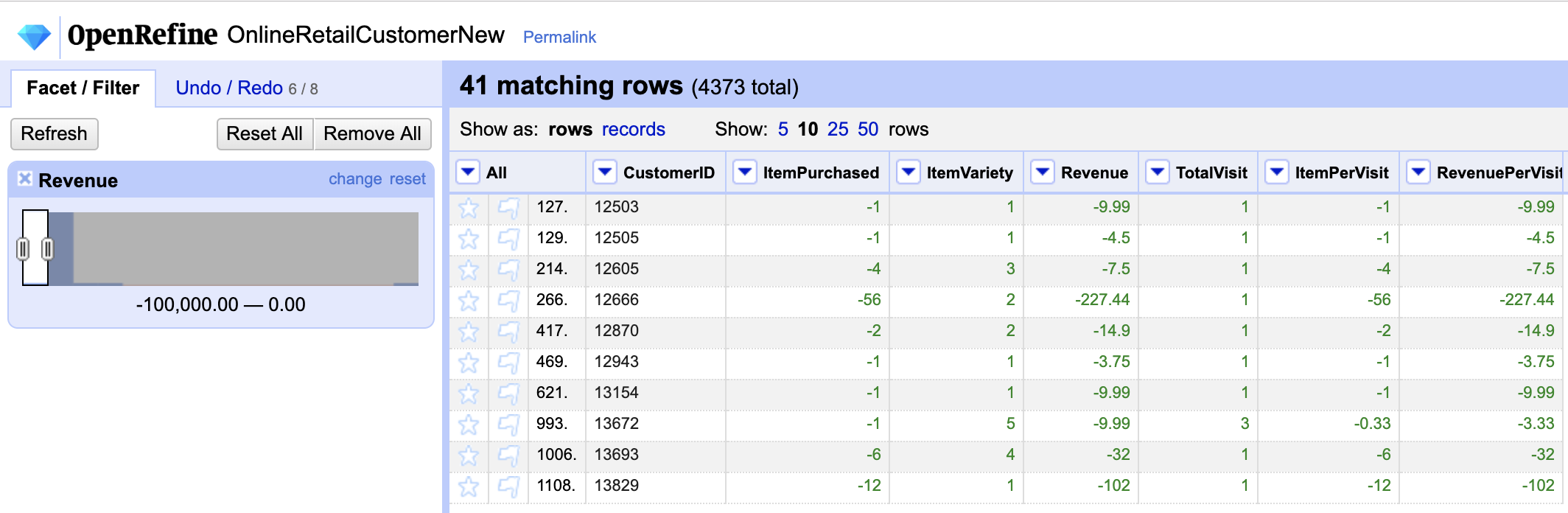
Figure 3 GGpairs for Products

From figure 3 GGpairs form Products we can see the linear relationship between Revenue and TotalCustomer, this can tell for one product if more customers purchased it, this product will bring in more revenue. There also has a linear relationship between TotalVisit and TotalCustomer, between TotalVisit and Revenue because more visit absolutely means more customers have purchased one particular product and more visit also likely contribute more revenue.

# Data Cleaning / Outlier Removal

Customer Data

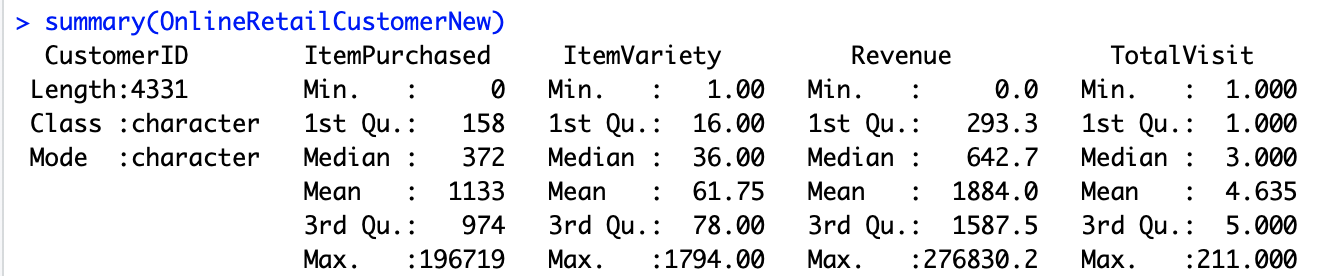
1. In the observation of the customer data using OpenRefine, 41 rows of customers have negative values in the attribute of total revenue. These customers are deemed as outliers and the reason of negative revenue will be further researched in future work.



1. The row with 0 as CustomerID is a collection of all customers without membership and deemed as an outlier.



1. In the observation of customer data in Rstudio, there are huge numbers of outliers with values much higher than 3rd Qu in all four existing attributes. Therefore, some of the customers can be identified as business customers. However, it’s not precise to just clean the outliers based on any of these four attributes.



1. To identify potential business customers, two attributes are introduced to find the outliers. We assume that the business customers can be best identified using the attributes of Revenue Per Visit and Item Quantity Per Visit, because individual customers are not likely to purchase a large number of items and spend a huge amount of money on each visit. We take 3 as the coefficient to find out extremely abnormal data. As a result, 150 of customers are deemed as potential business customers as well as outliers. The R scripts and the boxplots of two attributes before and after the cleaning are as follows.

QR1 <- quantile(OnlineRetailCustomerNew$RevenuePerVisit, probs = 0.25)

QR3 <- quantile(OnlineRetailCustomerNew$RevenuePerVisit, probs = 0.75)

QR31 <- QR3-QR1

hiR <- QR3+3\*QR31

CusRevClean <- OnlineRetailCustomerNew[-which(OnlineRetailCustomerNew$RevenuePerVisit>hiR),]

QI1 <- quantile(OnlineRetailCustomerNew$ItemPerVisit, probs = 0.25)

QI3 <- quantile(OnlineRetailCustomerNew$ItemPerVisit, probs = 0.75)

QI31 <- QI3 - QI1

hiI <- QI3+3\*QI31

CusTotalClean <- CusRevClean[-which(CusRevClean$ItemPerVisit>hiI),]

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Revenue Per Visit (Before/After)

A close up of a person

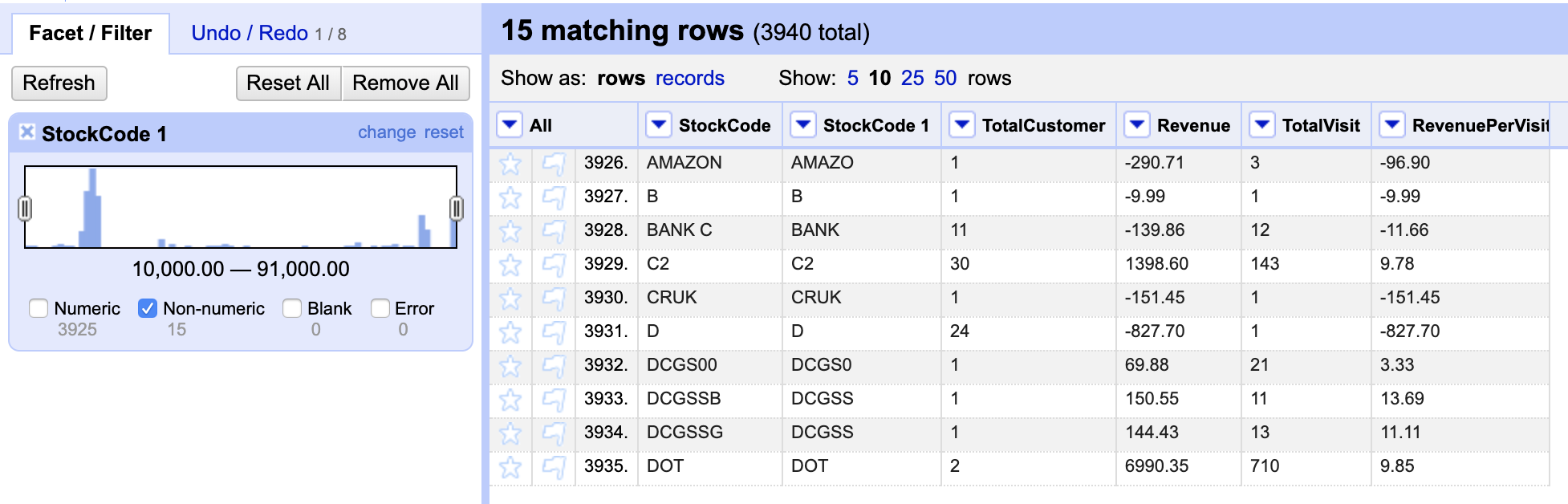
Description automatically generatedA screenshot of a cell phone

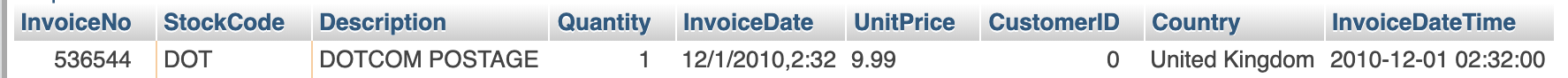
Description automatically generated

Item Quantity Per Visit (Before/After)

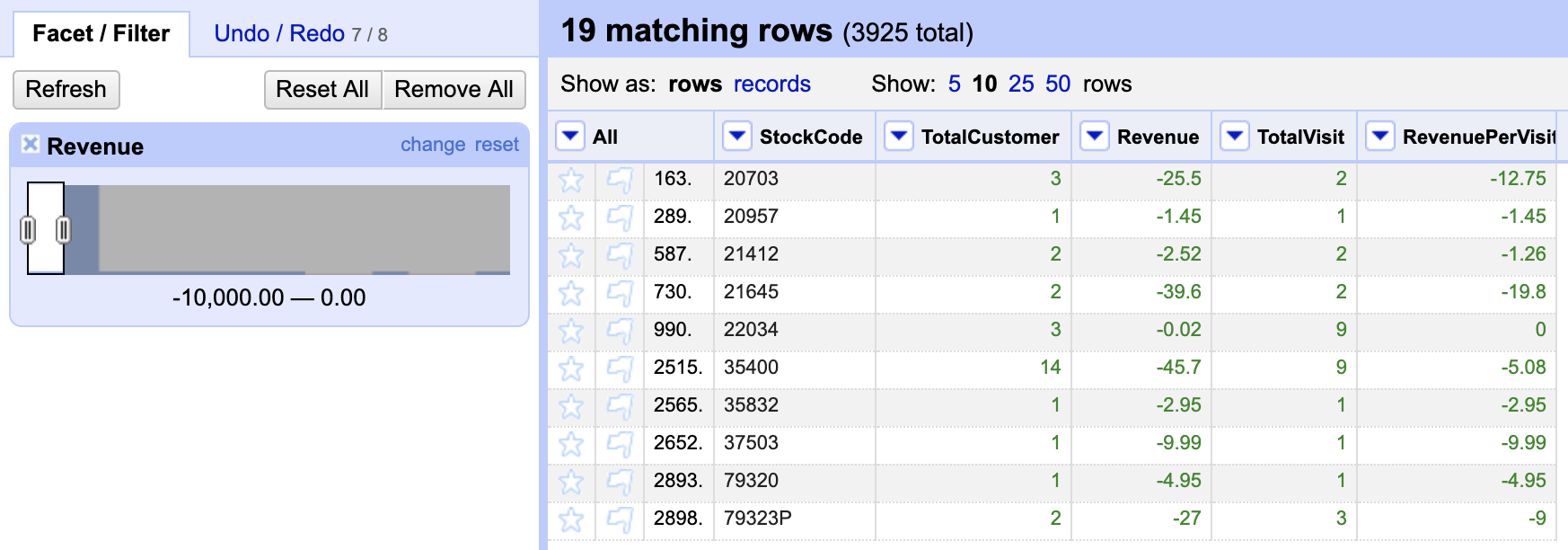
Product Data

1. In the observation of product data using OpenRefine, all products are featured with StockCode of 5 digits and one optional character. By using the function of splitting the column based on the length of 5, the products with non-numeric values are identified. By further looking into the data, those rows with unusual StockCode are not regular products. They are services such as postage, carriage or discounts, and can’t be used in product cluster analysis. As a result, these 15 rows are removed.





1. By further observation in OpenRefine, 19 rows of products generate negative total revenue with negative values in purchased quantity attribute and removed accordingly.

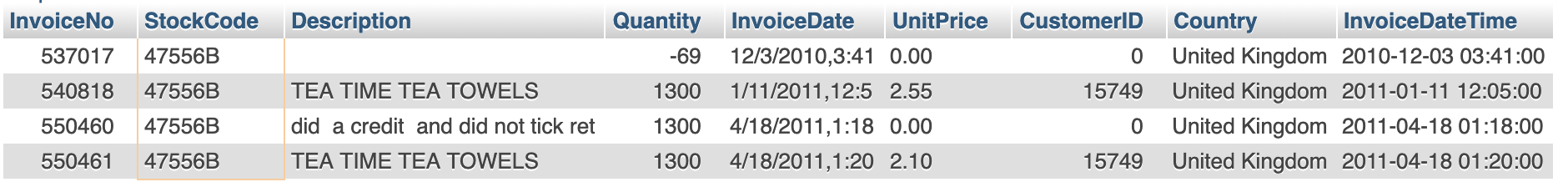




1. In the observation of the cluster analysis for the first time, one piece of data is noticed to be clustered by itself. By further looking into this product, it’s featured with the highest Revenue Per Visit and a high value of Quantity in rows with different InvoiceNo, but the Total Visit, Total Customer and Revenue is relatively low. As a result, it’s deemed as an abnormal product and removed.

A screenshot of a cell phone

Description automatically generated



# Cluster Analysis

Customer Cluster Analysis

1. The customer data is normalized using the scale function and the level of the importance of each attribute is assumed to be the same. The R scripts are as follows.

CusTotalClean.scale = scale(CusTotalClean[-1])

1. A function is defined to calculate the sum of square within the clusters with a different number of clusters, the K value. By observing the plot, 6 is identified as the best number of clusters. The R scripts are as follows.

withinSSrange <- function(data,low,high,maxIter)

{

withinss=array(0,dim=c(high-low+1));

for(i in low:high)

{

withinss[i-low+1] <-kmeans(data,i,maxIter)$tot.withinss

}

withinss

}

plot(withinSSrange(CusTotalClean.scale,1,50,150))

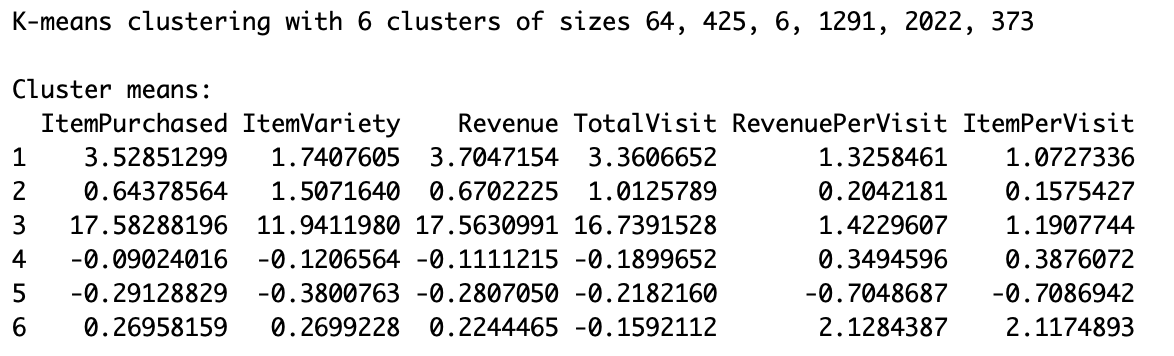
A picture containing screenshot

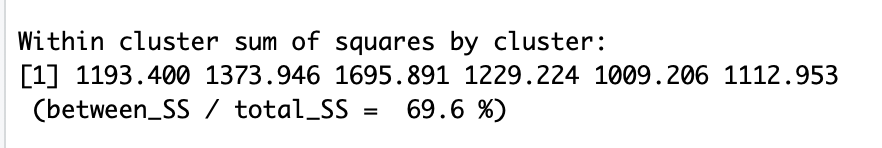
Description automatically generated

The normalized data is clustered using K-means function into 6 clusters. The sizes and the sum of squares within the cluster of each cluster are shown below. In the 3rd cluster, the size is small with the value of 6 comparing with others’ and the values in Item Purchased, Item Variety Revenue and Total Visit are much higher than others. The sum of squares within this cluster is also higher than others’. As a result, this cluster needs further attention to identify whether it’s a cluster of outliers and whether it’s the same with removed outliers which are deemed as potential business customers. The R scripts are as follows.

Cuskm = kmeans(CusTotalClean.scale,6,150)

Cuskm



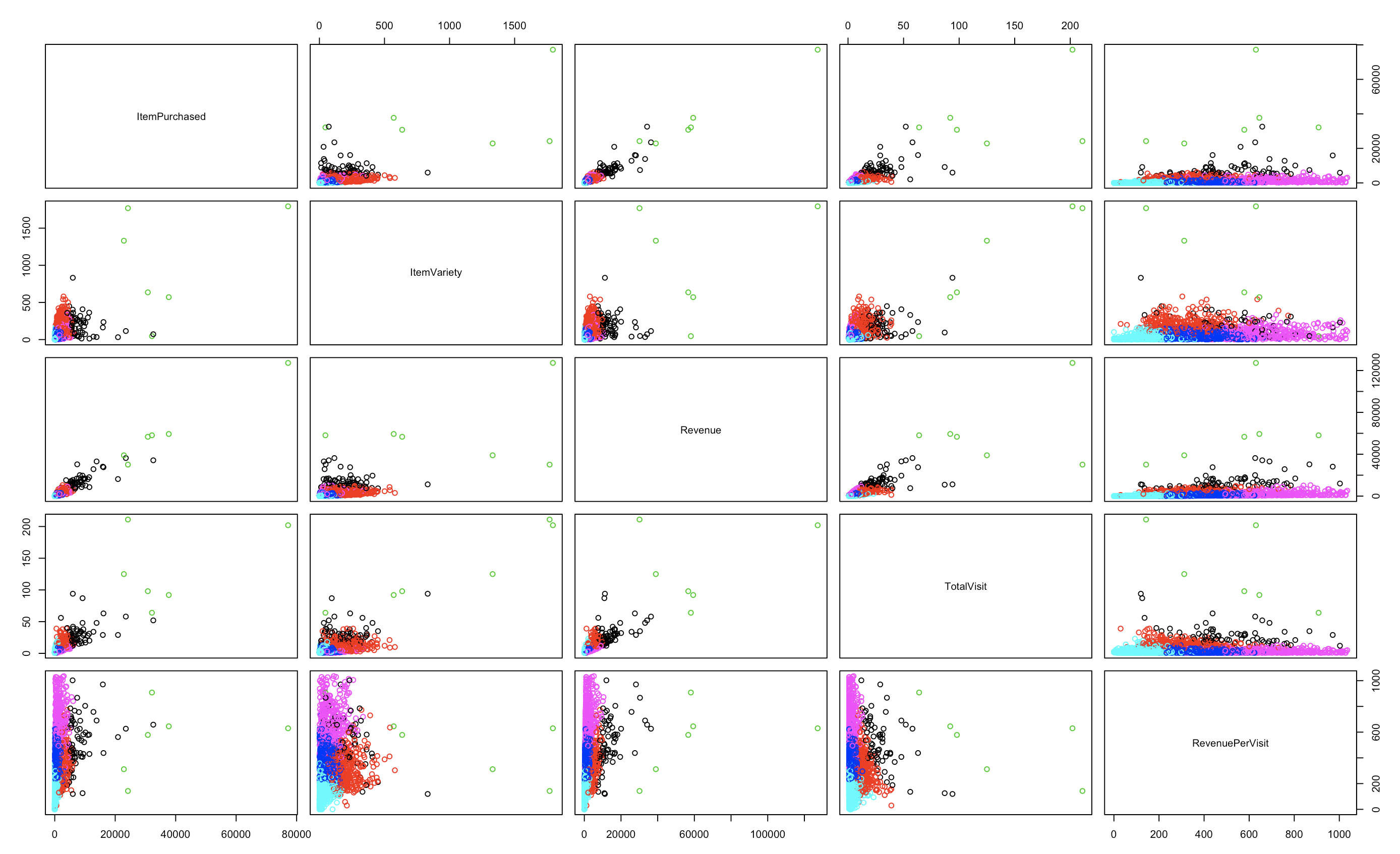


The customer data is denormalized and visualized using plot function. The R scripts are as follows.

CusTotalclean.realCenters = unscale(Cuskm$centers,CusTotalClean.scale)

clusteredCus = cbind(CusTotalClean,Cuskm$cluster)

plot(clusteredCus[,2:6],col=Cuskm$cluster)



Product Cluster Analysis

1. The product data is normalized using the scale function and the level of the importance of each attribute is assumed to be the same. The R scripts are as follows.

ProTotalClean.Scale= scale(ProTotalClean[-1])

1. A function is defined to calculate the sum of square within the clusters with a different number of clusters, the K value. By observing the plot, 7 is identified as the best number of clusters. The R scripts are as follows.

withinSSrange <- function(data,low,high,maxIter)

{

withinss=array(0,dim=c(high-low+1));

for(i in low:high)

{

withinss[i-low+1] <-kmeans(data,i,maxIter)$tot.withinss

}

withinss

}

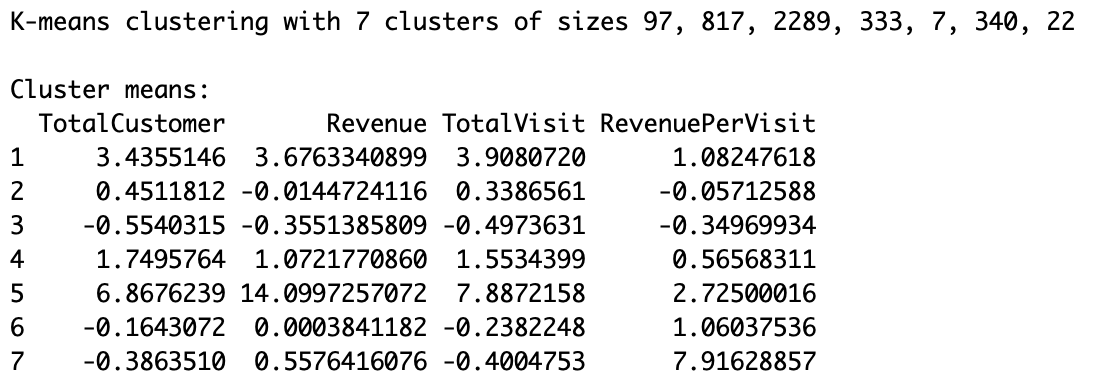
plot(withinSSrange(ProTotalClean.Scale,1,50,150))A picture containing screenshot

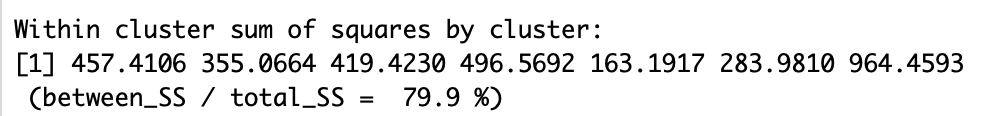
Description automatically generated

1. The normalized data is clustered using K-means function into 7 clusters. The sizes and the sum of squares within the cluster of each cluster are shown below. In the 5th cluster, the size is small with the value of 7 comparing with others’ and the values in Total Customer, Revenue and Total Visit are much higher than others. The sum of squares within this cluster is the smallest among others’. As a result, this cluster needs further attention to identify whether it’s a cluster of outliers and what’s the common feature of these 7 products. The R scripts are as follows.

Prokm = kmeans(ProTotalClean.Scale,7,150)

Prokm



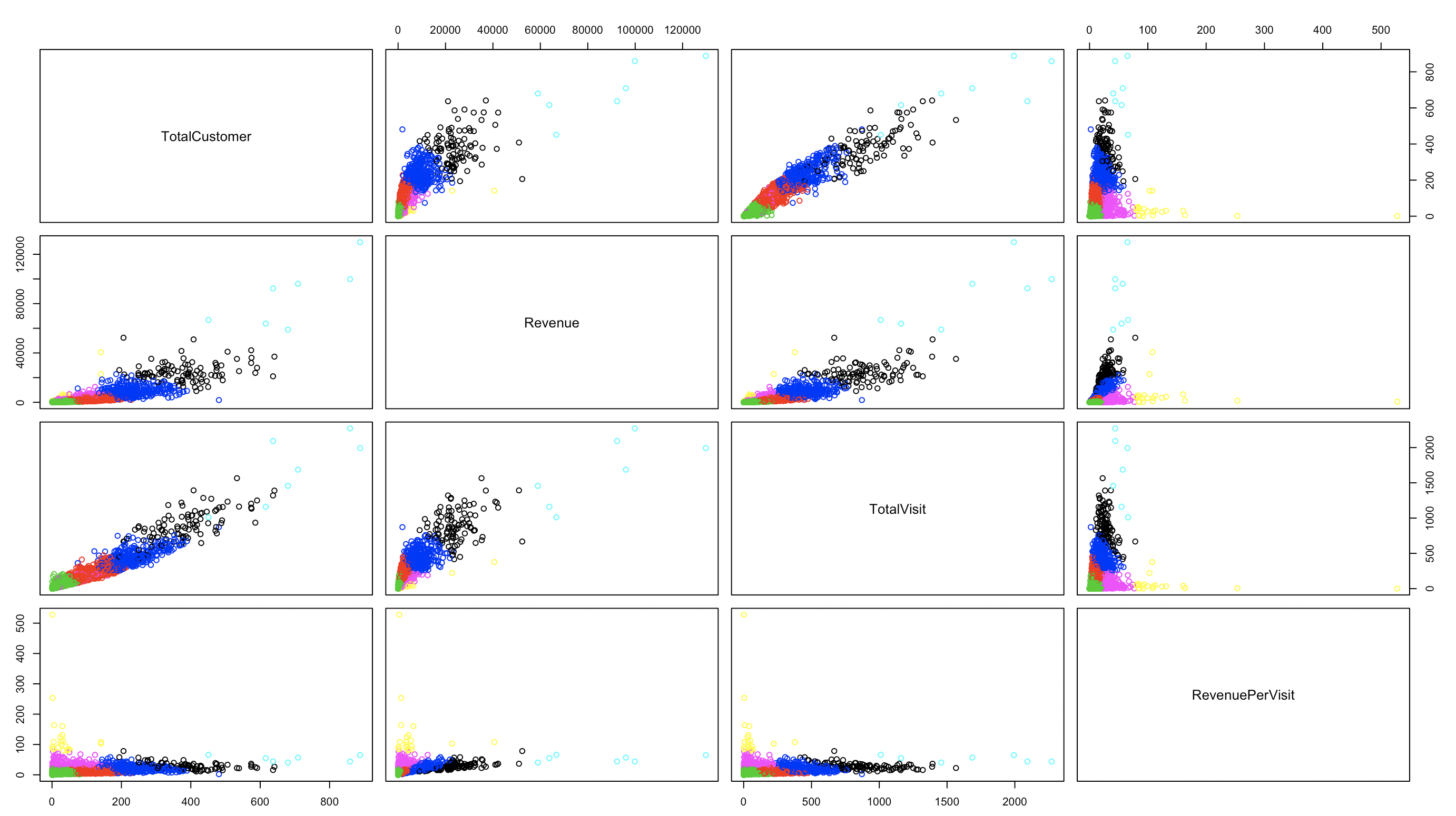


1. The product data is denormalized and visualized using plot function. The R scripts are as follows.

ProTotalClean.realCenters = unscale(Prokm$centers,ProTotalClean.Scale)

clusteredPro = cbind(ProTotalClean,Prokm$cluster)

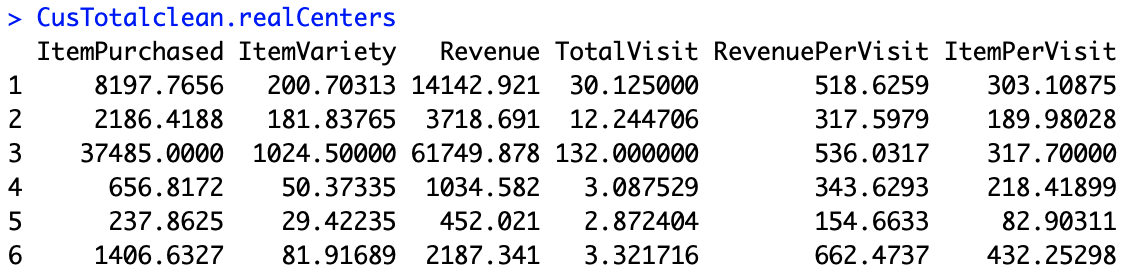
plot(clusteredPro[,2:5],col=Prokm$cluster)

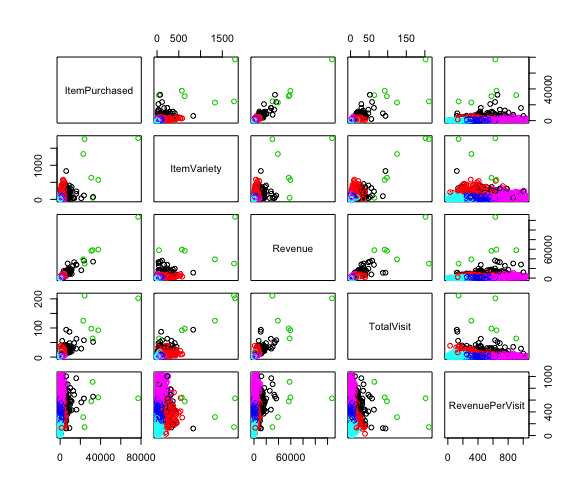


# Cluster Profiling

Customer Cluster Profiling

The customer behavioral data is clustered into 6 groups using 5 attributes except for Item Per Visit. The graph of clustering is shown in former section as well as below and the real centers of each group, the mean values, are listed as follows.





Loyal Family Customer (Size: 64, Within\_SS: 1193.40, Col = Black)

Item Purchased: 8197.77

Item Variety: 200.70

Revenue: 14142.92

Total Visit: 30.13

Revenue Per Visit: 518.63

1.53% of the customers belong to this cluster. The average values listed above are all in the second place among all 6 groups except for the Revenue Per Visit which is in the 3rd place. The quantity and variety of the purchased items are relatively high. Thus, the total revenue is also high. Although they visit the store more frequently than others, the average revenue on each visit is still high. To conclude, we can come to the assumption that these customers buy most of their daily needs here.

Loyal Individual Customer (Size: 425, Within\_SS: 1373.95, Col = Red)

Item Purchased: 2186.42

Item Variety: 181.84

Revenue: 3718.69

Total Visit: 12.24

Revenue Per Visit: 317.60

10.17% of the customers belong to this cluster. The average values listed above are all in the third place except for the Revenue Per Visit which is in fourth place but close to the third. The variety of purchased items is slightly smaller than Loyal Family Customers but the quantity and Revenue are nearly one-fourth of the former ones. We can assume that with similar variety but much smaller revenue, these customers are individual customers and the former ones are family customers.

1. Long Term Business Customer (Size: 6, Within\_SS: 1695.89, Col = Green)

Item Purchased: 37485.00

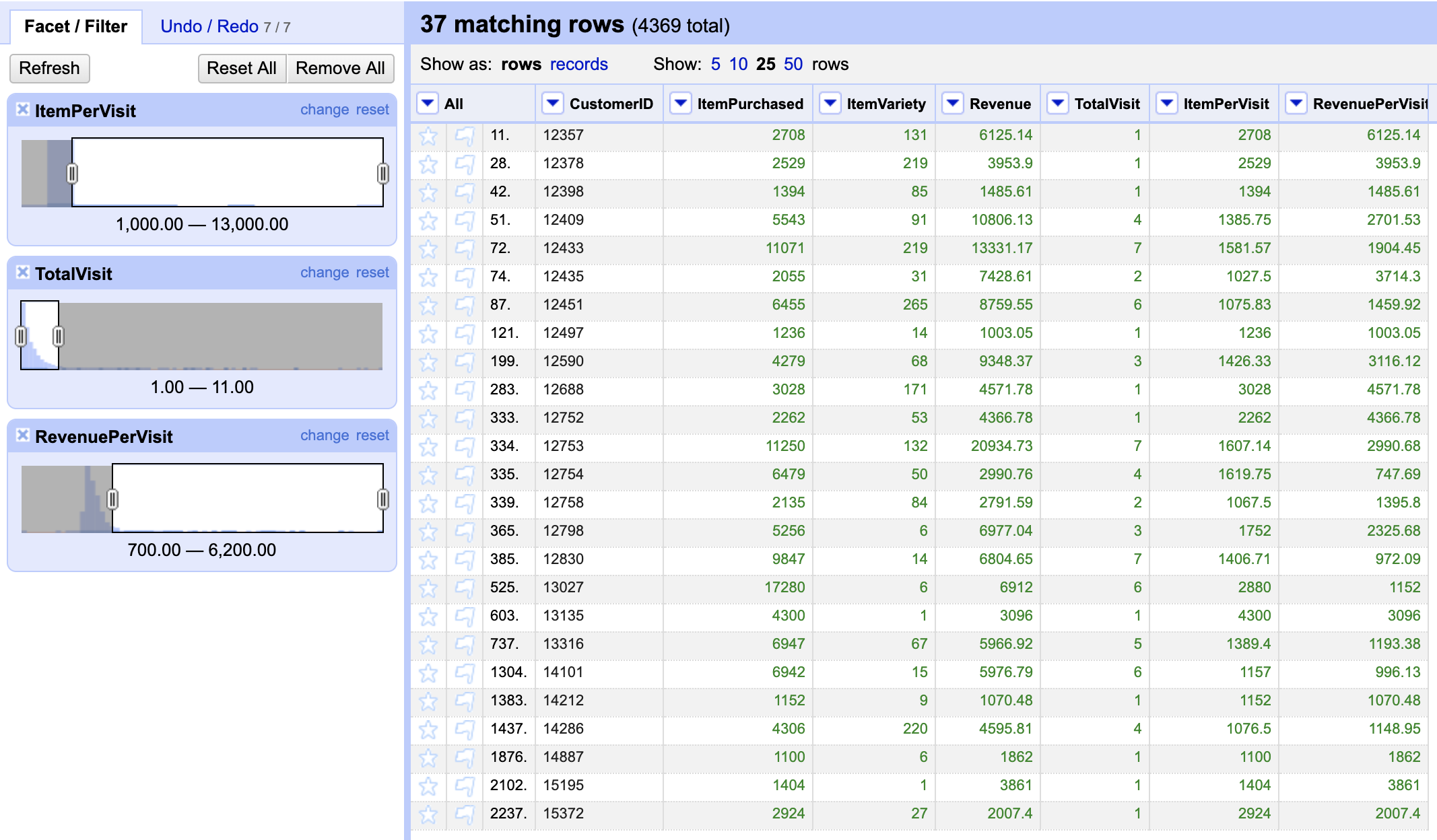
Item Variety: 1024.50

Revenue: 61749.88

Total Visit: 132.00

Revenue Per Visit: 536.03

Only 6 customers, 0.01% of the total customers, belong to these clusters. They feature extremely high frequency in visiting the store, as well as the Revenue Per Visit in second place. As a result, all other values of this cluster rank the first place. However, compared to the outliers moved in data cleaning, the Total Visit is much higher than those outliers. Based on the Total Visit, we can assume that these 6 customers are long term business customers. And the outliers with fewer visits are assumed to be short term business customers or new residents in this area.



1. Occasion Customer (Size: 1291, Within\_SS: 1229.22, Col = Deep Blue)

Item Purchased: 656.82

Item Variety: 50.37

Revenue: 1034.58

Total Visit: 3.09

Revenue Per Visit: 343.63

30.88% of the customers belong to this cluster. The quantity and variety of the purchased items are nearly one-third of the values of Loyal Individual Customers. But the Revenue Per Visit is even higher than Loyal Individual Customers. The reason might be the few visits which is around 3, one-fourth of the value of Loyal Individual Customers. We assume that they visit the online store only when the items are not available offline. As a result, they only purchase particular kinds of items in which their daily needs are not included.

1. Poor Performance Customer (Size: 2022, Within\_SS: 1009.21, Col = Light Blue)

Item Purchased: 237.86

Item Variety: 29.42

Revenue: 452.02

Total Visit: 2.87

Revenue Per Visit: 154.66

52.54% of the customers belong to this cluster. They perform the worst in every attribute. We can assume that the loyalty of these customers is low and this online store is not their first choice or they rarely do online shopping.

1. Potential Loyal Customer (Size: 373, Within\_SS: 1112.95, Col = Purple)

Item Purchased: 1406.63

Item Variety: 81.92

Revenue: 2187.34

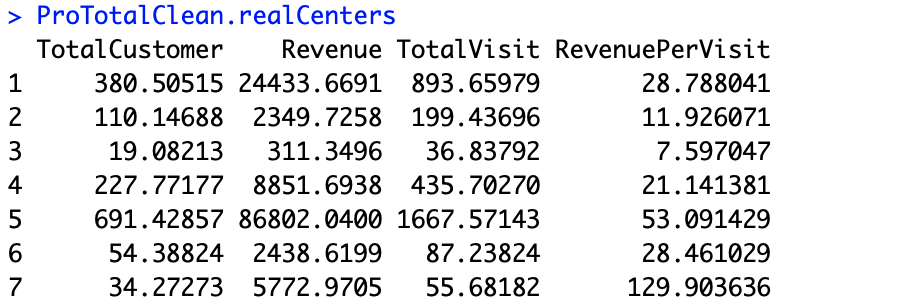
Total Visit: 3.32

Revenue Per Visit: 662.47

8.90% of the customers belong to this cluster. The average values listed above are in the fourth place except for the Revenue Per Visit which is the highest. The Total Visit is similar to the values of Occasion Customers and Poor Performance Customers. They feature great potential and would generate huge revenue if the visit frequency increases. We assume they have not had a clear preference in online stores or even between online and offline shopping. But they are more likely to become loyal customers.

Product Cluster Profiling

The product sale data is clustered into 7 groups using all 4 attributes. The graph of clustering is shown in former section as well as below and the real centers of each group, the mean values, are listed as follows.



A screenshot of a video game

Description automatically generated

1. Better Profiting Product (Size: 97, Within\_SS: 457.41, Col = Black)

Total Customer: 380.50

Revenue: 24433.67

Total Visit: 893.66

Revenue Per Visit: 28.79

2.48% of the products belong to this cluster. The average values of three attributes are in the second place except for Revenue Per Visit which is in the 3rd place but really close to the second. The overall performance of these products is good and steady.

1. Non Profiting Product (Size: 817, Within\_SS: 355.07, Col = Red)

Total Customer: 110.15

Revenue: 2349.73

Total Visit: 199.44

Revenue Per Visit: 11.93

20.92 of the products belong to this cluster. The average values of Revenue and Revenue Per visit are both in the second last place and the average values of Total Customer and Total Visit are in the fourth place. These products are popular to some extent but not profitable.

1. Poor Performance Product (Size: 2289, Within\_SS: 419.42, Col = Green)

Total Customer: 19.08

Revenue: 311.35

Total Visit: 36.84

Revenue Per Visit: 7.60

58.62% of the products are in this cluster. They feature the least values in every attribute. Few customers are interested in those items and they are not profiting in each visit as well.

1. Profiting Product (Size: 333, Within\_SS: 496.57, Col = Deep Blue)

Total Customer: 227.77

Revenue: 8851.69

Total Visit: 435.70

Revenue Per Visit: 21.14

8.53% of the products belong to this cluster. Compared to Non-Profiting Products, these products feature twofold values in Total Customer and Total Visit, and three times of the values in Revenue Per Visit. The average value of Revenue is in third place among all 7 groups. They performed slightly worse than the Better Profiting Products but still good in profiting.

1. Star Product (Size: 7, Within\_SS: 163.19, Col = Light Blue)

Total Customer: 691.43

Revenue: 86802.04

Total Visit: 1667.57

Revenue Per Visit: 53.09

Only 7 products, 0.18% of all products, belong to this cluster. They feature the best in all four attributes except for the Revenue Per Visit which is in the second place. They are the most popular products as well as the profiting ones.

1. Potential Profiting Product (Size: 340, Within\_SS: 283.98, Col = Purple)

Total Customer: 54.39

Revenue: 2438.62

Total Visit: 87.24

Revenue Per Visit: 28.46

8.71 of the products belong to this cluster. The average Revenue of these products is like Non-Profiting Products but the average Revenue Per Visit is close to Better Profiting Products. However, the values of Total Customer and Total Visit are both at the fifth place which is relatively small. These products are not popular but have the potential to profit.

1. Treasure Product (Size: 22, Within\_SS: 964.46, Col = Yellow)

Total Customer: 34.28

Revenue: 5772.97

Total Visit: 55.68

Revenue Per Visit: 129.90

Only 22 products, 0.56% of all products, belong to this cluster. These products feature the second last popularity in Total Customer and Total Visit. But the Revenue is slightly smaller than Profiting Products. The value of Revenue Per Visit of these products is the highest among all 7 groups. Minor of the customers are interested in these products but they are profiting.

# Conclusion & Next Steps

To conclude we have 6 clusters of the customer and 7 clusters of product. For example, we have differentiated loyal family customer who purchased a high variety and quantity of product also contribute a high revenue for daily family use and the loyal individual customer who purchased slightly smaller quantity and variety product also revenue probably for daily individual use. We also identify different types of products and the features behind as well as purchasing behaviors.

In future work, we can utilize those attributes for customer cluster to perform targeted marketing to those potential royal customers and attract them to become a loyal customer. We can do more analysis on the products which potential royal customer more frequently buy but not the loyal customer to optimize the inventory management strategy. We can also develop marketing strategies such as promotion and advertising on different types of products.

# Reference

[1] Michael, G., 2018. Understanding Boxplots. [Online] Available at:

<https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51> [accessed May 2019]

# Appendix

MySQL Queries

alter table onlineretail

add column TotalPrice double;

SET SQL\_SAFE\_UPDATES = 0;

update onlineretail set TotalPrice = Quantity \* UnitPrice;

select CustomerID,

sum(Quantity) as ItemPurchased,

count(distinct StockCode) as ItemVariety,

sum(TotalPrice) as Revenue,

count(distinct InvoiceNo) as TotalVisit,

sum(TotalPrice)/count(distinct InvoiceNo) as RevenuePerVisit,

sum(Quantity)/count(distinct InvoiceNo) as ItemPerVisit

from onlineretail

group by CustomerID;

select StockCode,

count(distinct CustomerID) as TotalCustomer,

sum(TotalPrice) as Revenue,

count(distinct InvoiceNo) as TotalVisit,

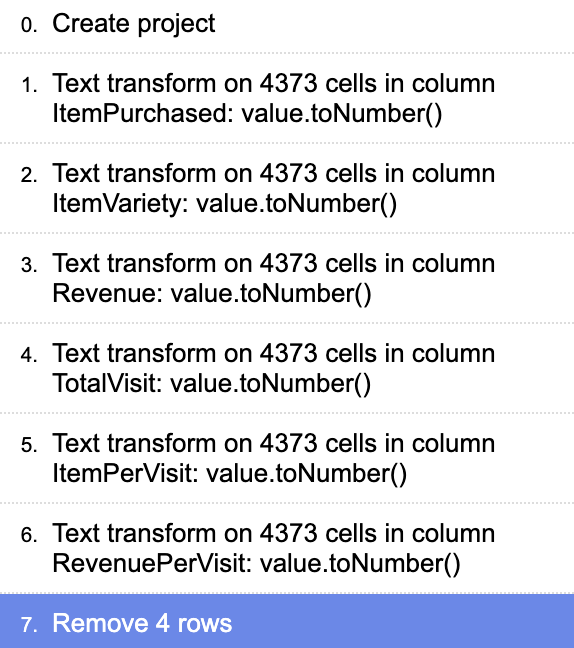
sum(TotalPrice)/count(distinct InvoiceNo) RevenuePerVisit

from onlineretail

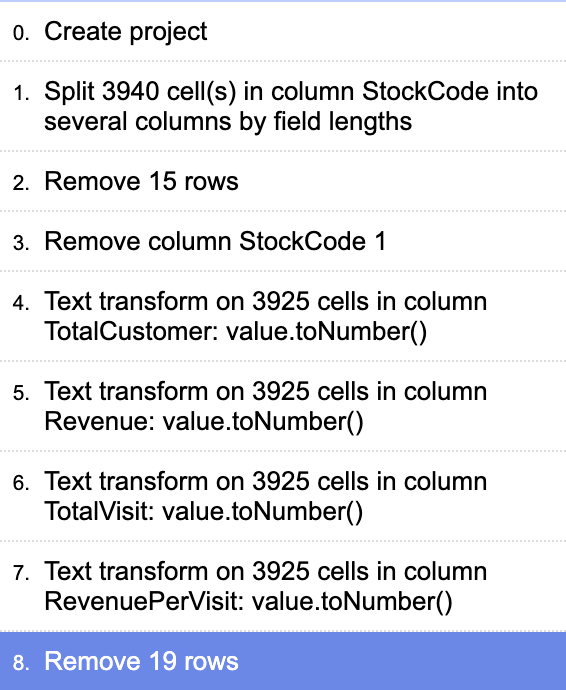
group by StockCode;

OpenRefine Project

Customer



Product



R Scripts

library(ggplot2)

library(GGally)

library(DMwR)

set.seed(5580)

boxplot(OnlineRetail$Quantity)

summary(OnlineRetail$Quantity)

summary(OnlineRetailCustomerNew)

boxplot(OnlineRetailCustomerNew$RevenuePerVisit)

QR1 <- quantile(OnlineRetailCustomerNew$RevenuePerVisit, probs = 0.25)

QR3 <- quantile(OnlineRetailCustomerNew$RevenuePerVisit, probs = 0.75)

QR31 <- QR3-QR1

hiR <- QR3+3\*QR31

CusRevClean <- OnlineRetailCustomerNew[-which(OnlineRetailCustomerNew$RevenuePerVisit>hiR),]

summary(CusRevClean$RevenuePerVisit)

boxplot(OnlineRetailCustomerNew$ItemPerVisit)

QI1 <- quantile(OnlineRetailCustomerNew$ItemPerVisit, probs = 0.25)

QI3 <- quantile(OnlineRetailCustomerNew$ItemPerVisit, probs = 0.75)

QI31 <- QI3 - QI1

hiI <- QI3+3\*QI31

boxplot(CusItemClean$ItemPerVisit)

CusTotalClean <- CusRevClean[-which(CusRevClean$ItemPerVisit>hiI),]

ggpairs(CusTotalClean[,which(names(CusTotalClean)!="CustomerID")],upper = list(continous = ggally\_points), lower = list(continous = "points"))

CusTotalClean.scale = scale(CusTotalClean[-1])

withinSSrange <- function(data,low,high,maxIter)

{

withinss=array(0,dim=c(high-low+1));

for(i in low:high)

{

withinss[i-low+1] <-kmeans(data,i,maxIter)$tot.withinss

}

withinss

}

plot(withinSSrange(CusTotalClean.scale,1,50,150))

Cuskm = kmeans(CusTotalClean.scale,6,150)

CusTotalclean.realCenters = unscale(Cuskm$centers,CusTotalClean.scale)

clusteredCus = cbind(CusTotalClean,Cuskm$cluster)

plot(clusteredCus[,2:6],col=Cuskm$cluster)

Cuskm

CusTotalclean.realCenters

summary(OnlineRetailProductNew)

boxplot(OnlineRetailProductNew$RevenuePerVisit)

boxplot(OnlineRetailProductNew$TotalCustomer)

boxplot(OnlineRetailProductNew$Revenue)

boxplot(OnlineRetailProductNew$TotalVisit)

ProTotalClean <- OnlineRetailProductNew

ProTotalClean <- ProTotalClean[-which(ProTotalClean$RevenuePerVisit>700),]

ggpairs(ProTotalClean[,which(names(ProTotalClean)!="StockCode")],upper = list(continous = ggally\_points), lower = list(continous = "points"))

ProTotalClean.Scale= scale(ProTotalClean[-1])

plot(withinSSrange(ProTotalClean.Scale,1,50,150))

Prokm = kmeans(ProTotalClean.Scale,7,150)

ProTotalClean.realCenters = unscale(Prokm$centers,ProTotalClean.Scale)

clusteredPro = cbind(ProTotalClean,Prokm$cluster)

plot(clusteredPro[,2:5],col=Prokm$cluster)

summary(clusteredPro)

Prokm

ProTotalClean.realCenters