Online Retail Selling Cluster Report

MCDA 5580

Team member:

Mitkumar Patel A00444857

Chirag Panasuriya A00442907

Wensho Li A00445457

Contents

Executive summary

An analysis of online retail sales data was performed to better understand customers purchasing behaviours at online. The analysis focused on a world-wide range of areas, which customers are from the following countries:

United Kingdom	Japan	Switzerland	Israel
France	Iceland	Spain	Finland
Australia	Channel Island	Poland	Bahrain
Netherlands	Denmark	Portugal	Greece
Germany	Cyprus	Italy	Hong Kong
Norway	Sweden	Belgium	Singapore
EIRE	Austria	Lithuania	Lebanon

Table-1 Countries Customers from

And analysis is also focusing on the top 2,000 products and customers by overall revenue. Though use of the K-Means clustering algorithm, 5 customer segments and 6 product segments were deemed optimal based on underlying data. These segments represent partially homogenous clusters of customers and products that have similar characteristics. A brief overview of the segments is presented below:

Customer Segments:

	# of customers	Description
Vendors	163(15.73%)	Customers who bought cheap products in bulks.
Normal Buyers	293(28.28%)	Customers buy moderate price product but in less quantity.
Refunder	88(8.49%)	Customers who come for Return the product
Potential Buyers	134(12.93%)	Customer who bought lots of things and has many visit
Impulsive Buyers	358(34.56%)	Customer bought expensive product

Table-2 Customer Segments

Suggestion for customers:

- For the 1st cluster, send deal flyers every week with the newspaper, send them emails related to new exciting discount, give them loyalty discount every week based on their shopping.
- For the 2nd clusters, inspire to buy new products and try to increase total quantity; give coupon of new items, so they will visit store in few days and exposed with the new items.
- For the 3rd cluster of customers, few customers buy product for one time use and then they return the product once they are done with their motive so company should form new rules for expensive product e.g. if customer return the product without any solid motive, company gives only 50% of total cost.
- For the 4th cluster of customers, show advertisements on the social media based on their purchased and related products(product clusters), which helps to increase number of visits and profit.

- The last category of customers care about product quality and their need so we can find their need based on their web surfing. Buy user's product search data from Search engines such as Google and Bing, display these products with some special offer on the YouTube, Facebook, Instagram.

Product Segments:

	# of products	Description	Product Example
Popular Product	43(6.14%)	These Products give good revenues and customer buy these products after second visits. This cluster's products have a high average unit price, but it did not influence the selling quantity. That is why this cluster has the second high revenue among all clusters.	MINI FUNKY DESIGN TAPES
High Potential	108(15.42%)	Products which is cheap and buy in high quantity. It requires less visits for selling and sell in high quantity but generate less revenue.	RED RETROSPOT PURSE
Advertisement Needer	164(23.43%)	Products has moderate price with less revenue and less average quantity. This product cluster has low average unit price which influence the revenue of it. Besides, it attracts only a few customers.	DOGGY RUBBER
Specific attraction product	80(11.43%)	Products have moderate revenue; few customers buy it frequently in high quantity. So, it will attract a specific kind of customers.	GREEN POLKADOT BOWL
Champion product	38(5.42%)	These kind of products are high quality and required in daily life. Most favourite products, price is high, and customer buy in most of the visit so this cluster offers maximum in profit.	SANDALWOOD FAN
Loss leader	267(38.14%)	These products have less number of visits and revenues. This cluster has the lowest number of customers, average selling quantity and average unit price.	INFLATABLE POLITICAL GLOBE

Table-3 Product Segments

Suggestion for product:

- We can increase selling of first cluster's products using second cluster's products. If customer buy 2nd cluster's product in bulk (decide minimum amount of quantity), then customer will be rewarded with the coupons of 1st cluster's product. Customer will get some discount for first cluster products in next visit.
- Sell 3rd cluster's product in packing of 2-3 similar products with some discount because of high quantity. For example, pack green colour's Doggy rubber with red colour's doggy rubber.
- We can reduce margin for 4th cluster's product to raise its selling because it has good revenue.
- 5th cluster's products are most favourable among customer so product of 5th cluster and 6th cluster sell together. Customer will use new product and get new experience. If Customer like 5th cluster's product, then they will buy 5th and 6th cluster's product separately.

Understanding these segments allows for better alignment of corporate initiatives, such as marketing and revenue generation. These segments also lend themselves to products based on specific characteristics and early purchasing behaviour. Although beyond the scope of this initial analysis, the authors note that a more robust analysis of world-wide regions with additional descriptive and qualitative information for customers and products would produce and even more robust analysis that could benefit the entire Online Retail Network.

Objective

Online Retail, a new retail way which is getting popular among the world, has requested an analysis be performed on their sales data with and expressed interest in understanding the nature of customers shopping online. Specifically, Online Retail wants to understand more about their customers and products they are purchasing so they can better align corporate initiatives, such as marketing and revenue generation. It is the report author's intent to perform various data preparation, cleansing, and segmentation producers to produce an analysis that will provide beneficial information to the business owner. In the absence of specific requirements from the business, the report's authors will consider the goal of customer segmentation to be: knowledge generation for target marketing purposes, and the goal of production segmentation to be: knowledge generation for new product development, optimization of product placement on retail shelves, and elimination of low-revenue items that contribute to excessive storage costs. A successful analysis will produce delineated and communicable profiles for customers and products that will aid in decision making and support future analyses for Online Retail.

About the data

To support the analysis, Online Retail has provided transactional sales data captured between December 1,2010 and February 2,2011 from various regions. The sales data provides purchasing behaviour as it relates to both customers and the products they are purchasing. A brief description of each table is provided herein:

Online Retail Table

Provides additional information for each item in a transaction, such a description of the product, its unit price, quantity and customer who bought this product. And each row also contains information about customers, such country that customers come from, invoice date and time.

Number of records:	79118
Number of unique records:	79118
Reference Field:	N/A(only one table)

Design/Methodology/Approach

When performing a large analysis with foreign data, it is important to devise a technique that supports the overall objective, is communicable, and is repeatable. When analysing customer segmentation, the most frequently used section technique is the RFM-model, which categorizes customers based on Recency, Frequency, and Monetary value. Using the Online Retail as an example. Recency measures the number of days since the customers has last visited the store, Frequency measures the total numbers of visits in the dataset, and Monetary value measures the total amount spent by customer. A popular technique for employing RFM is to assign scores to categorize customers. However, this approach has the disadvantage of only using three selection variables (Recency, Frequency, and Monetary value), when other characteristics and measurements may be available and equally as informative.

This report's authors have chosen to enrich the dataset by engineering the RFM features; however, these features will serve to compliment other measures and characteristics available in the data. In this regard, the RFM process of categorizing based on assigned scores will be substituted for a more generic clustering analysis supported by K-Means clustering algorithm. The K-Means algorithm was chosen because it is fast and easy to explain to business owners. The algorithm is unsupervised, so prior knowledge of the data and existing classification is not required which allows for objective and potentially surprising patterns to emerge. A brief overview of the analytical process followed in this report is outlined below:

Subset the data for all transactions at Online Retail

Select and engineer the appropriate features to support the analysis.

Pull out the top 2,000 customers and top 2,000 products based on revenue.

Clean the data and remove outliers.

Normalize the data.

Determine appropriate number of clusters and perform clustering.

De-normalize the data.

Use metadata from the original dataset to create customer and product profiles based on clustering.

Feature selection, Feature engineering and Feature definition

As previously mentioned, the chosen approach will loosely follow the popular RFM-model; however, rankings are not employed, and additional complimentary features are added. After analysing the transactional sales data, the following datasets were created for the purposes of the analysis:

Customers Dataset:

Feature Name	Measurement	Description
CustomerID	N/A	Unique field for identify customer
TOTAL_QUANTITY	SUM(Quantity)	Product total quantity that customer bought
DISTINCT_PRODUCT	SUM(UnitPrice * Quantity)	Number of product kinds that customer bought
REVENUES	SUM(UnitPrice * Quantity)	Revenue that customer contribute to online retail
NUM_VISITS	COUNT(InvoiceDateTime)	Number of that customer visit the online retail
NUM_REFUND	COUNT(Quantity < 0 OR NULL)	How many times that the customer get refund
AVG_PRICE_ITEM	AVERAGE	The average price of all products that customer bought
WEEKDAY	(showing in below graph)	Weekday in which customer bought products at most times.

Table-4 Customers Dataset

Products Dataset:

Feature Name	Measurement	Description
DISTINCT_CUSTOMER	COUNT DISTINCT	How many customers have bought this product
REVENUES	SUM(UnitPrice * Quantity)	How much revenue this product made
NUM_VISITS	COUNT	How many times this product has been browsed
AVG_QUANTITY	AVERAGE	Average quantity of selling in all transaction this product involved in.
AVG_UNITPRICE	AVERAGE	Average price of selling in all transaction this product involved in.
StockCode	N/A	Unique field for identify customer

Table-4 Product Dataset

The following analysis was performed using these two datasets exclusively, with the original dataset reserved for referral purposes and cluster profiling. Both datasets have been reduced to the top 2,000 by total revenue (the original distinct customers are no more than 2,000) to expedite analysis and produce meaningful results against the products and customers most relevant for revenue generation.

Data Cleansing and Outlier Removal

As the name implies, the K-Means algorithm determines clustering by calculating the mean of clustered points, which as statistical measurement for central tendency is sensitive to outliers. For this reason, outliers in the data should be removed to avoid distorting clustering results; however, outlier removal in transactional data is a contentious subject. Form a business perspective, the outliers offer much of the information one hopes to glean from the data in terms of identifying high-value/low-value customers and products. To compensate, one could employ the K-Medians algorithm, which is less sensitive to outliers; however, the report authors have chosen to stay with K-Means algorithm and provide ample justification for any records that are removed.

Common approaches for removing outliers involve somewhat arbitrarily determining acceptable upper and lower bounds, generally based on distribution, and removing datum that fall outside these bounds. However, as mentioned previously, this can remove datum important for business decisions. This approach was evaluated through plotting the data in a series of boxplots:

Product Boxplot (before and after outlier removal)

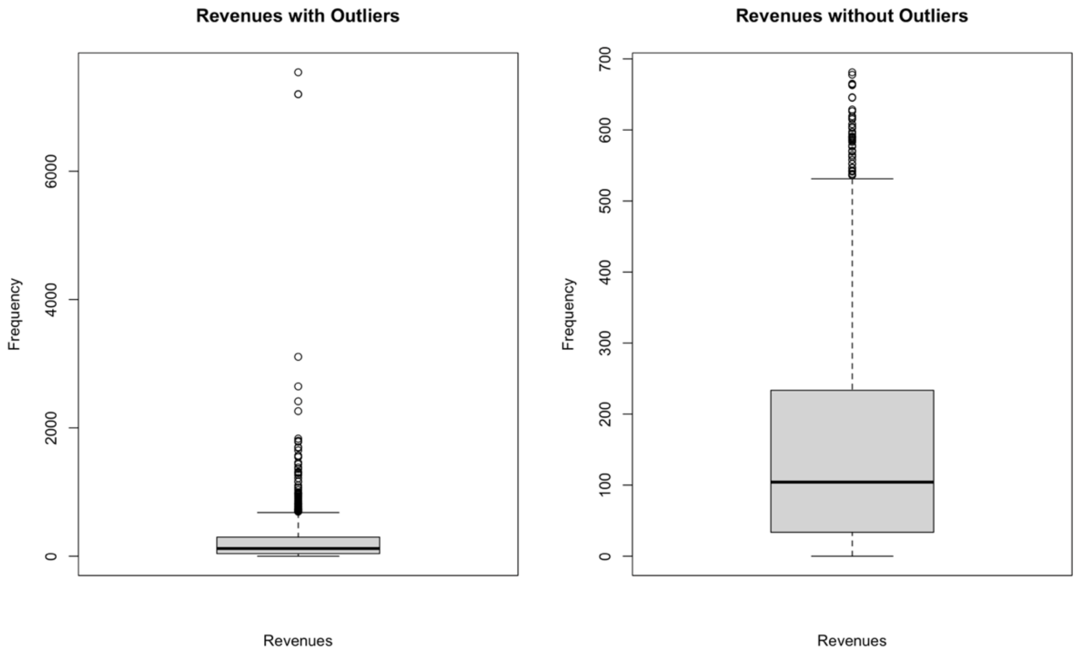


Figure-1 Boxplot before and after removal outliers for revenue

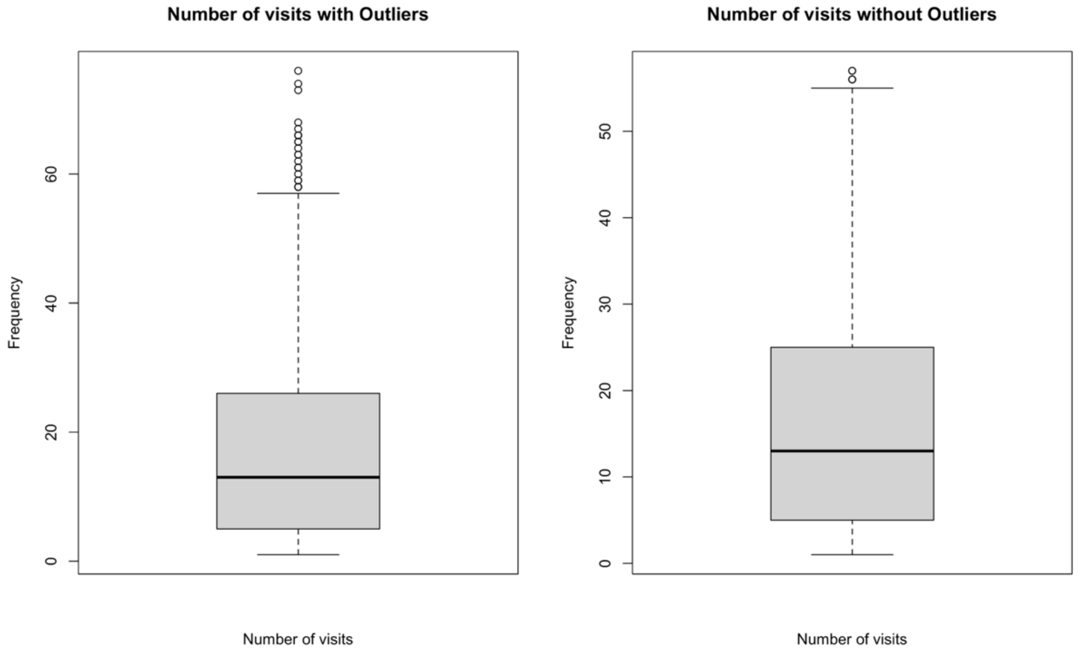


Figure-2 Boxplot before and after removal outliers for number of visits

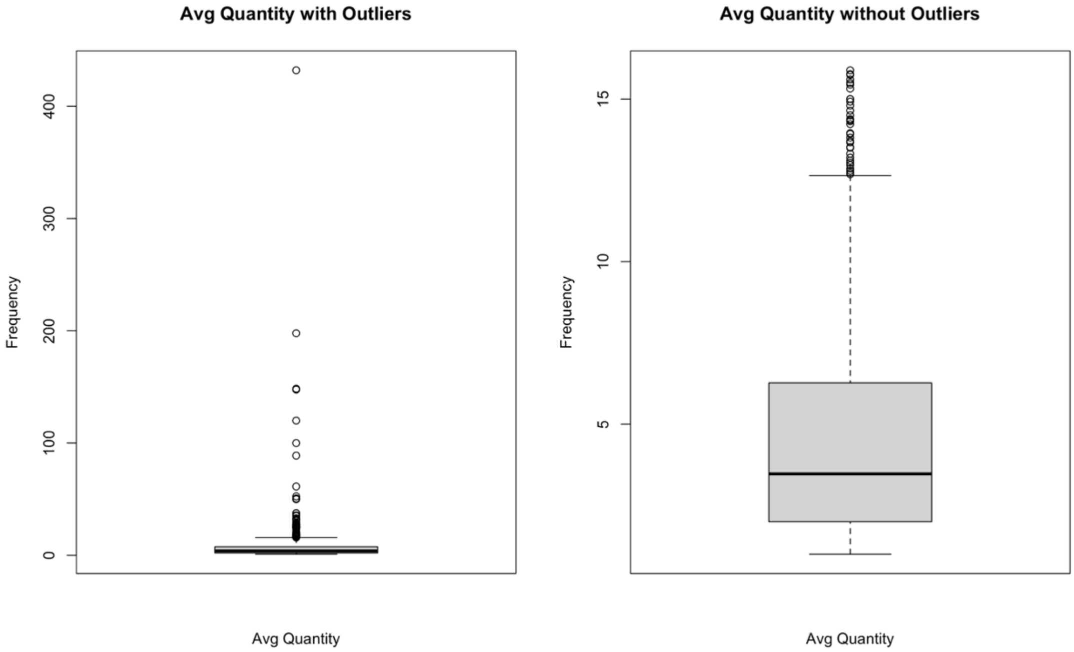


Figure-3 Boxplot before and after removal outliers for average quantity

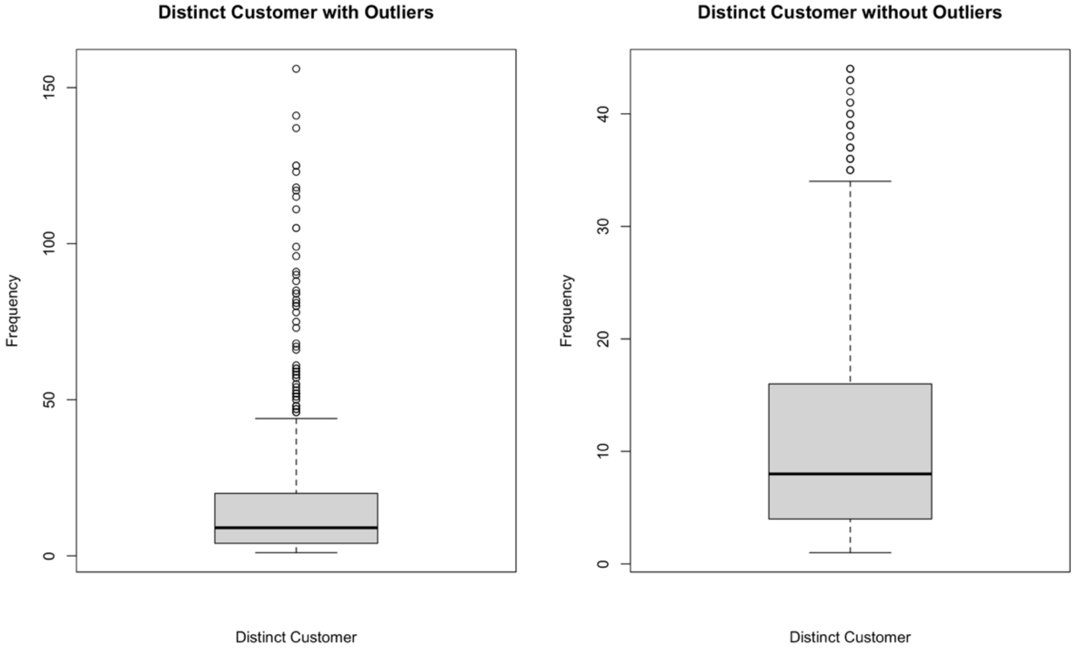


Figure-4 Boxplot before and after removal outliers for distinct customers

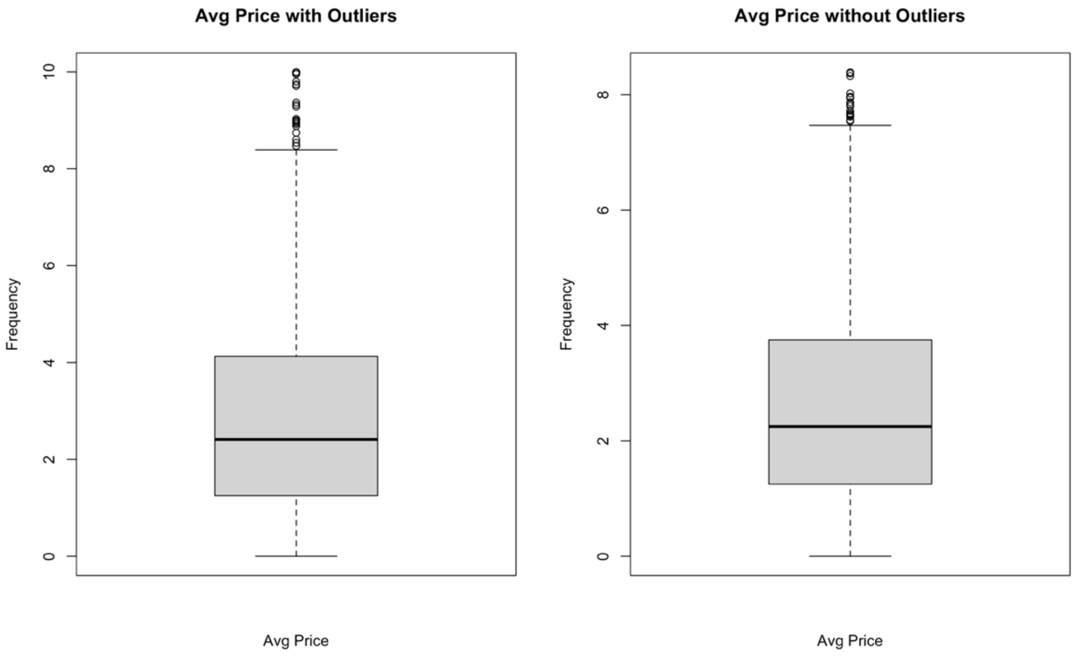


Figure-5 Boxplot before and after removal outliers for average price

Customer Boxplot (before and after outlier removal)

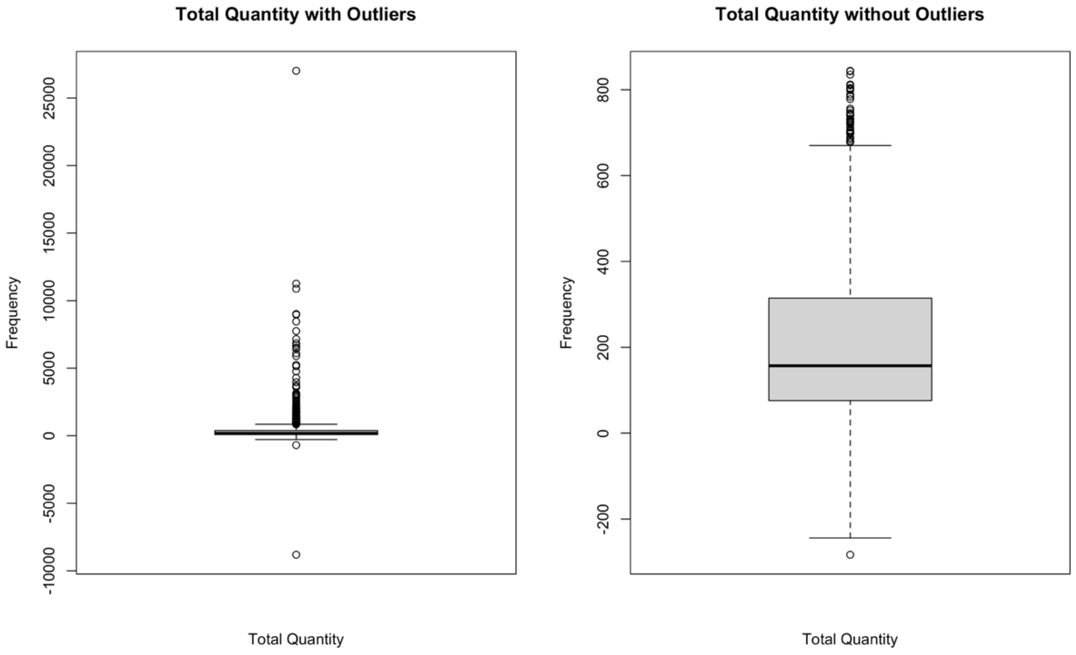


Figure-6 Boxplot before and after removal outliers for total quantity

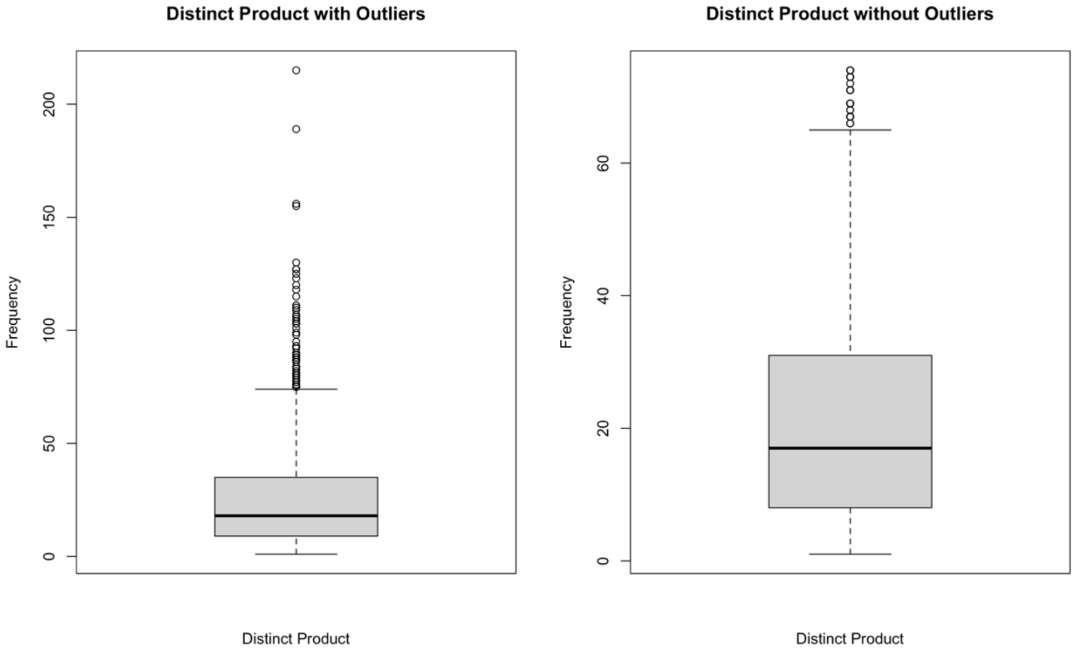


Figure-7 Boxplot before and after removal outliers for distinct products

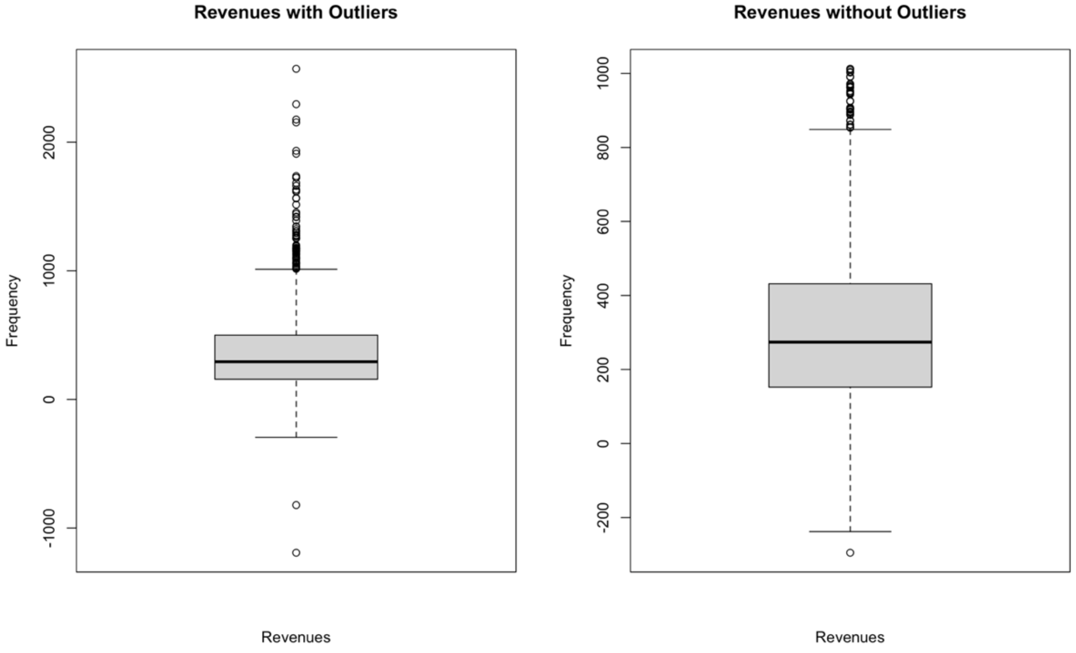


Figure-8 Boxplot before and after removal outliers for revenues

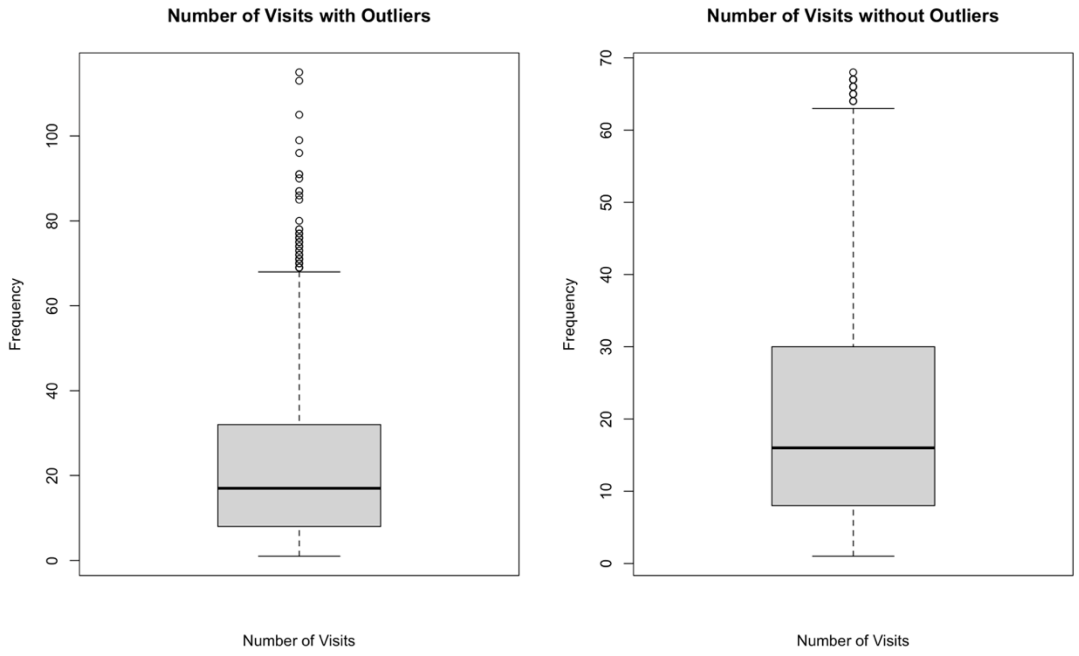


Figure-9 Boxplot before and after removal outliers for number of visits

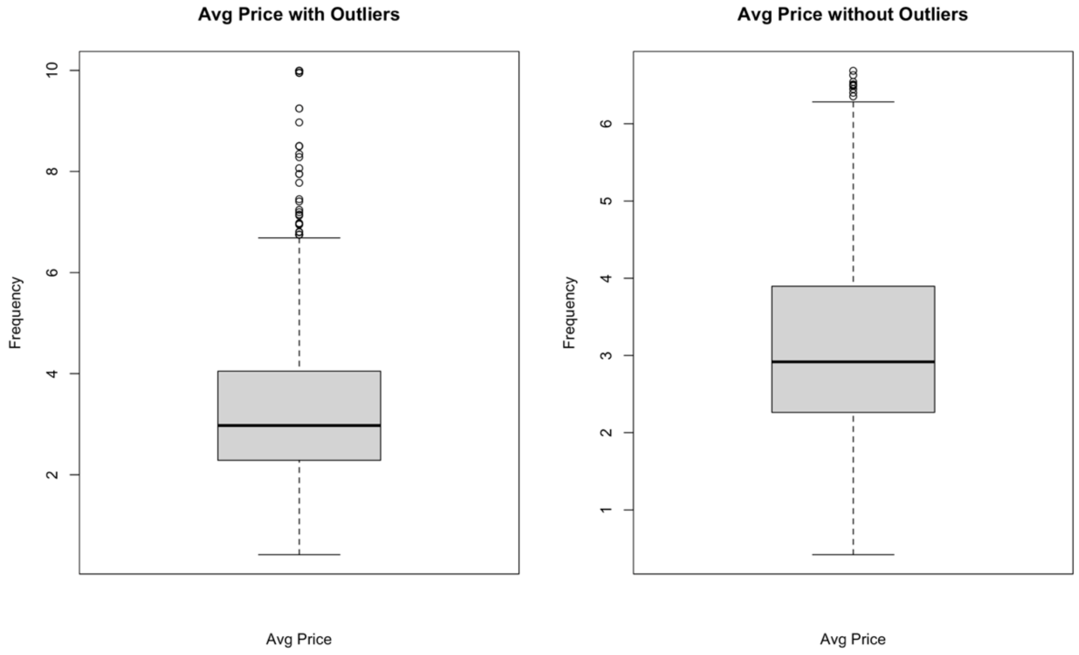


Figure-10 Boxplot before and after removal outliers for average price

Given the identified risks associated with removing outliers using the above approach, the authors employed a simplistic and effective method for identifying outliers by creating a matrix of plots that compares all features. Visually inspecting these plots will help identifying outliers. The initial plots for our customer and product datasets are shown here.

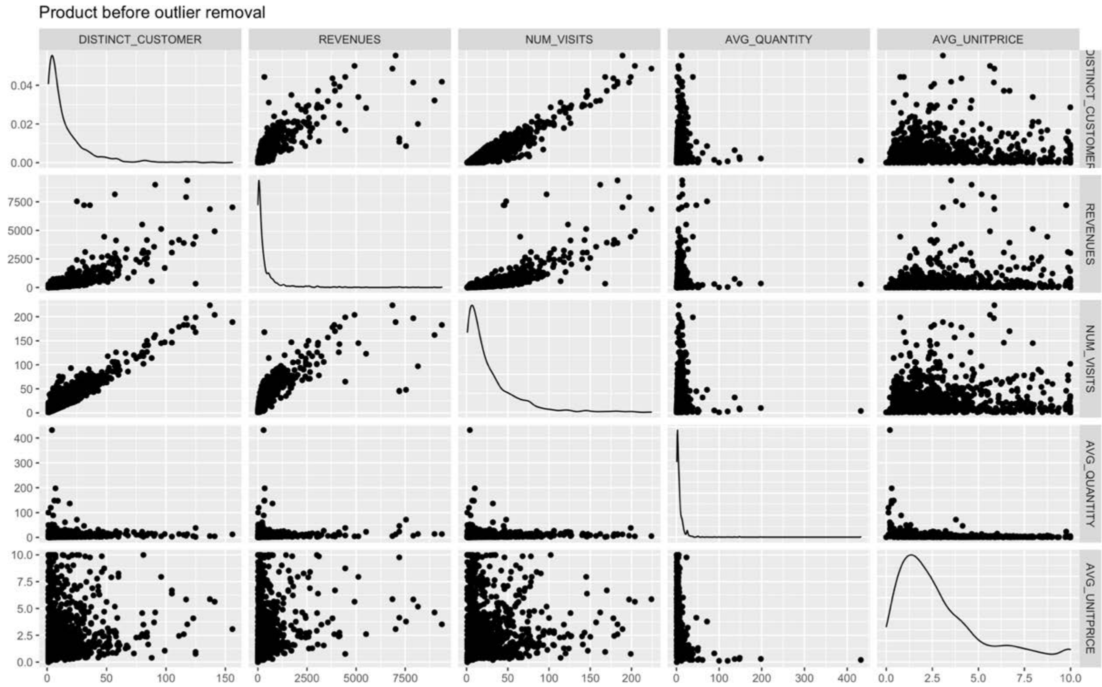


Figure-11 Scatter Plot of Customer's attributes with outliers

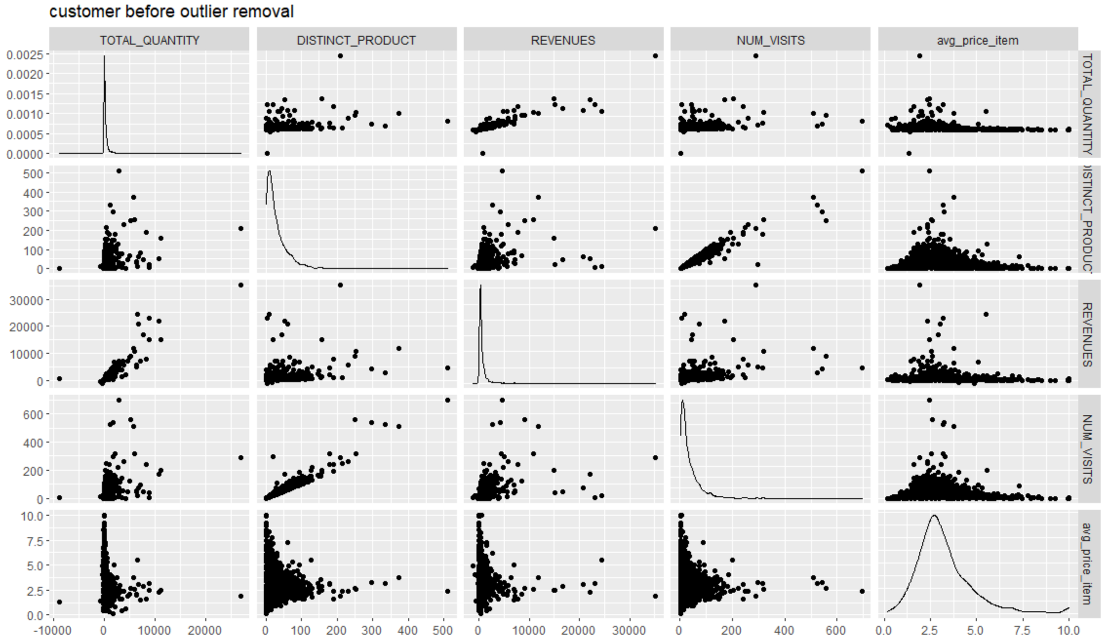


Figure-12 Scatter Plot of Customer's attributes without outliers

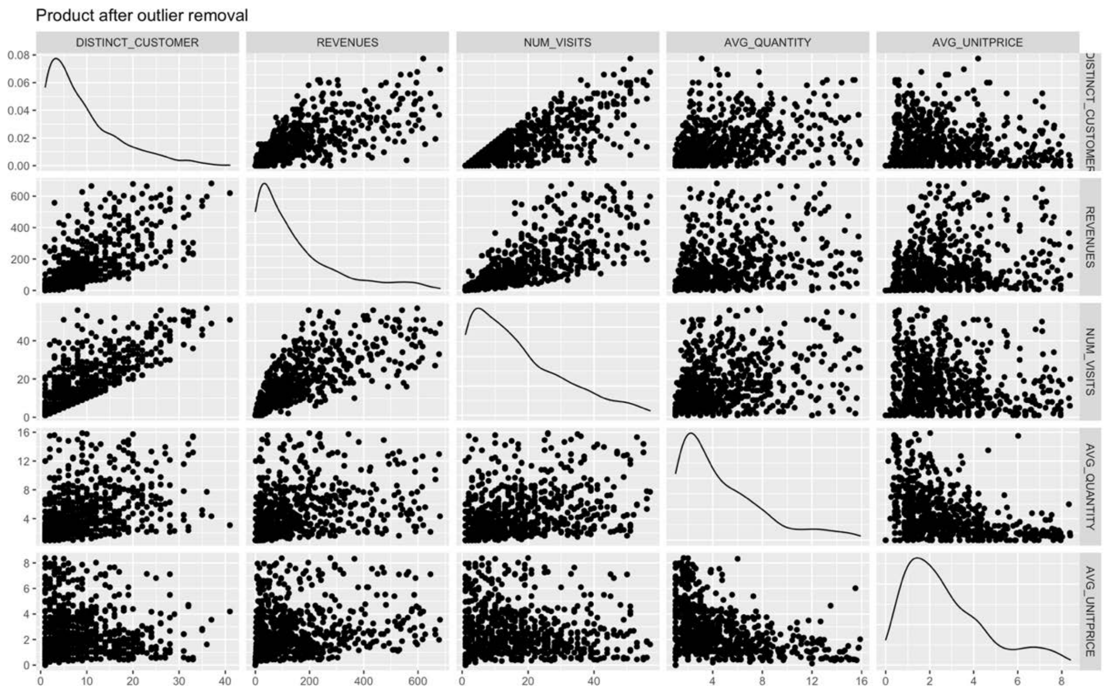


Figure-13 Scatter Plot of Product's attributes with outliers

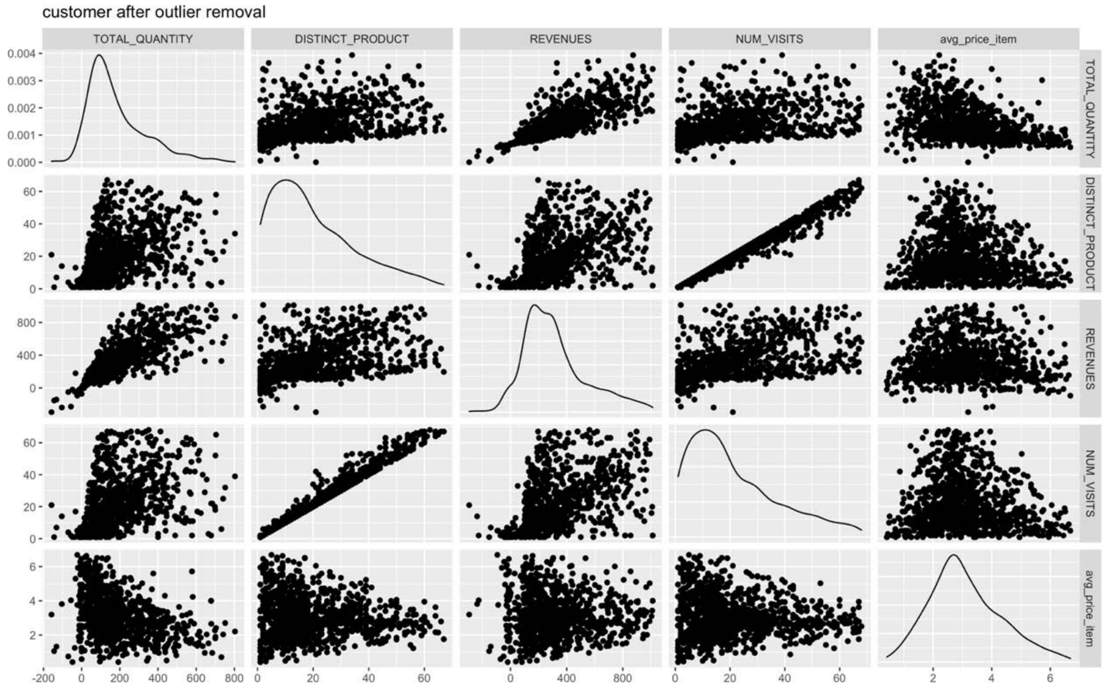


Figure-14 Scatter Plot of Product's attributes without outliers

Explain on how we identify the outliers:

A boxplot helps to visualize a quantitative variable by displaying five common location summary (minimum, median, first and third quartiles and maximum) and any observation that was classified as a suspected outlier using the interquartile range (IQR) criterion. The IQR criterion means that all observations above $q_{0.75}+1.5 \cdot \mathrm{IQR}$ or below $q_{0.25}-1.5 \cdot \mathrm{IQR}$ (where $q0.25$ and $q0.75$ correspond to first and third quartile respectively, and IQR is the difference between the third and first quartile) are considered as potential outliers by R. In other words, all observations outside of the following interval will be considered as potential outliers:

$$I = [q_{0.25} - 1.5 \cdot IQR; q_{0.75} + 1.5 \cdot IQR]$$

Example of outliers that we judge:

Removal product 21733, which is 'RED HANGING HEART T-LIGHT HOLDER'

We remove it because it attracts number of customers. And it also falls out of the formula above.

Removal customer 14646, which create enormous revenue and contribute a lot to number of visit and product quantity.

We remove this customer because we consider he/she could be a vender.

Cluster Analysis

The result of a single clustering run will find a local optimum – a locally best clustering – but will be dependent upon the initial centroid locations. For this reason, k-means is usually run many times, starting with different random centroids each time.

There are several methods used for determining the appropriate number of clusters for k-means analysis. The authors chose to use the “elbow plot” method due to its simplicity and ease of use. While the optimal point cannot always be objectively defined through this approach, the intent is to find the point at which the total sum of squares is minimized within clusters while not overfitting.

The plots below show that the marginal benefit of adding a cluster after the 6th point does not improve the analysis by a significant enough margin to warrant consideration. Due to this, 6 was select as the optimal number of clusters for analysis on both datasets.

As the elbow curve for customers showing below, we can see that the inflection point is 5 and 6. We formed the clusters for K=5 and K=6 and decided to go with K=5. We found the overlapping in the K=6.

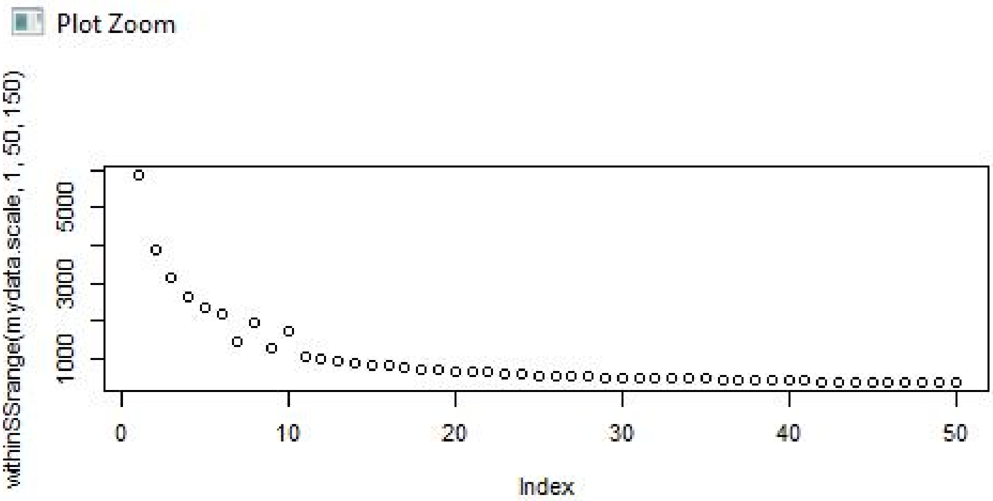


Figure-15 Elbow Graph of Customer data using K-mean

Product

As the elbow curve for products showing below, we can see elbow at K = 6. So we have formed 6 clusters.

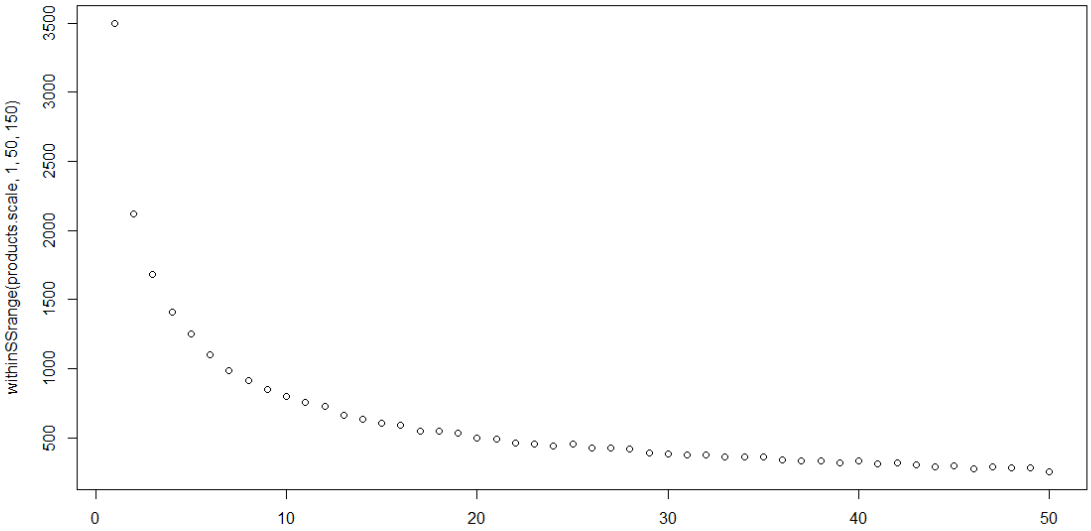


Figure-16 Elbow Graph of Product data using K-mean

Cluster Profiling

Once the desired number of clusters have been produced, we should create profiles. And we choose characteristic profiling. Characteristic profiling involves either determining a single data point that best represents the entire cluster or using the centroid to describe the average characteristics of data points in the cluster. Differential profiling is supervised process where the cluster labels are combined with the existing data to determine what characteristics best explain the differences in the profiles.

Customer Clusters Overview

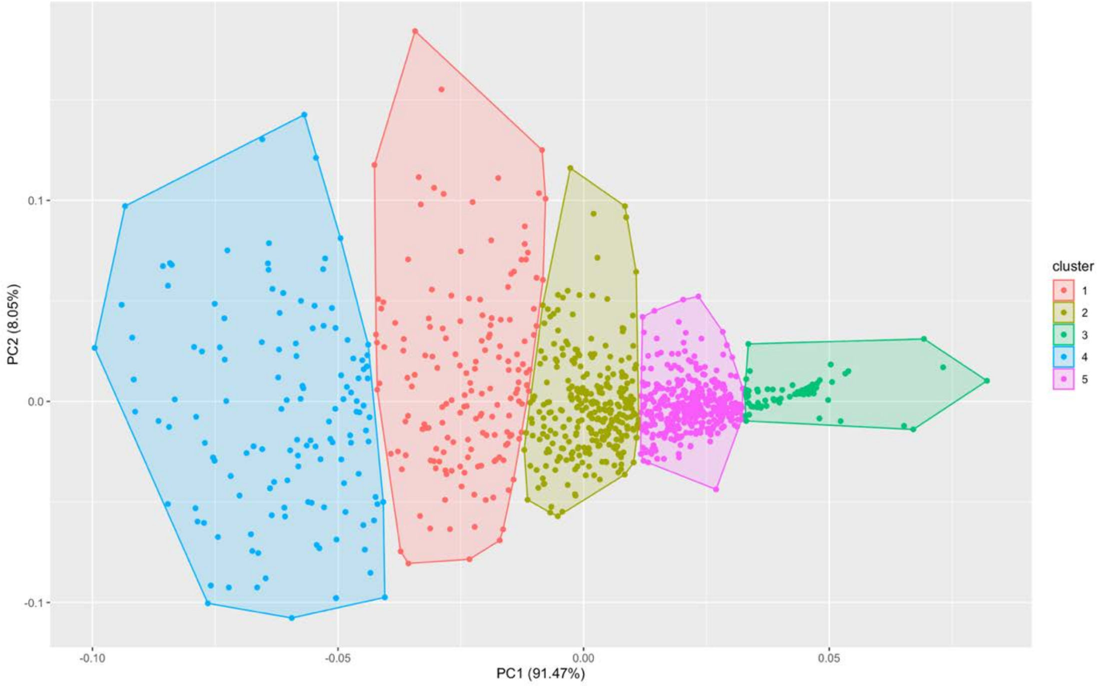
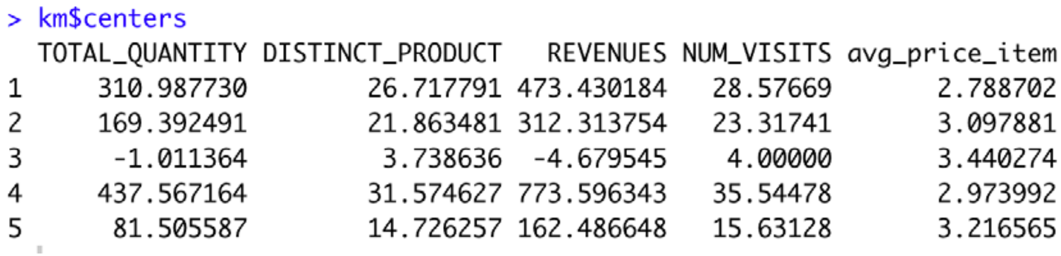


Figure-17 Clusters of Customers



```
> km$centers
  TOTAL_QUANTITY DISTINCT_PRODUCT   REVENUES NUM_VISITS avg_price_item
1     310.987730         26.717791 473.430184   28.57669       2.788702
2     169.392491         21.863481 312.313754   23.31741       3.097881
3      -1.011364          3.738636  -4.679545    4.00000       3.440274
4     437.567164         31.574627 773.596343   35.54478       2.973992
5      81.505587         14.726257 162.486648   15.63128       3.216565
```

Figure-18 Five Centroids of Product clusters

We used characteristic profiling to create profile of each cluster.

1. Customer who bought cheap products in bulks
2. Customers buy moderate price product but in less quantity.
3. Who come to return the product.
4. Customer who bought lots of things and has many visits.
5. Customer bought expensive product.

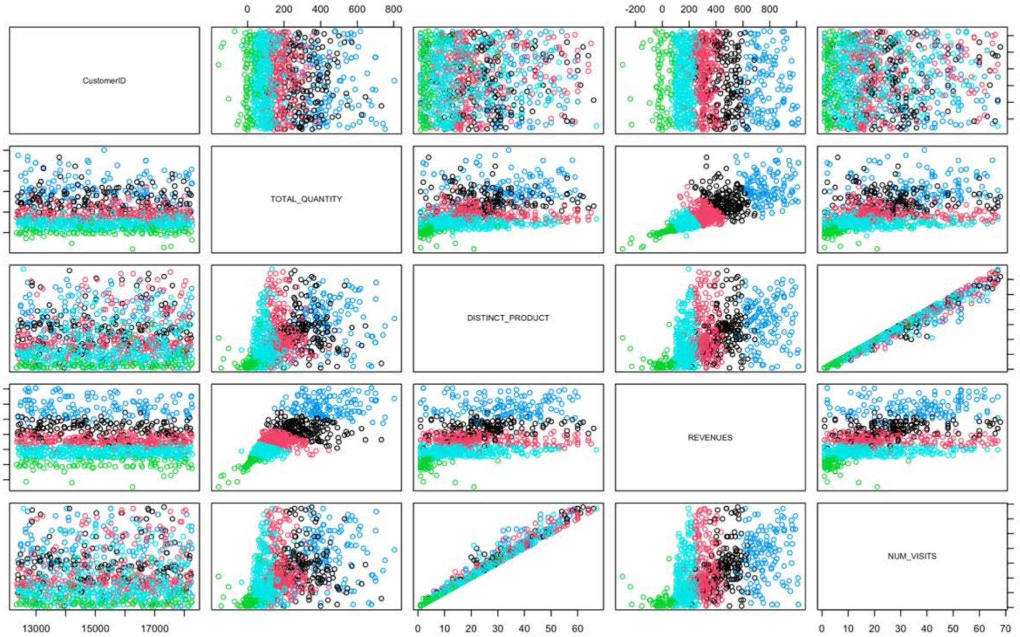


Figure-19 Graphical representation of Customers attributes

Product Cluster Overview

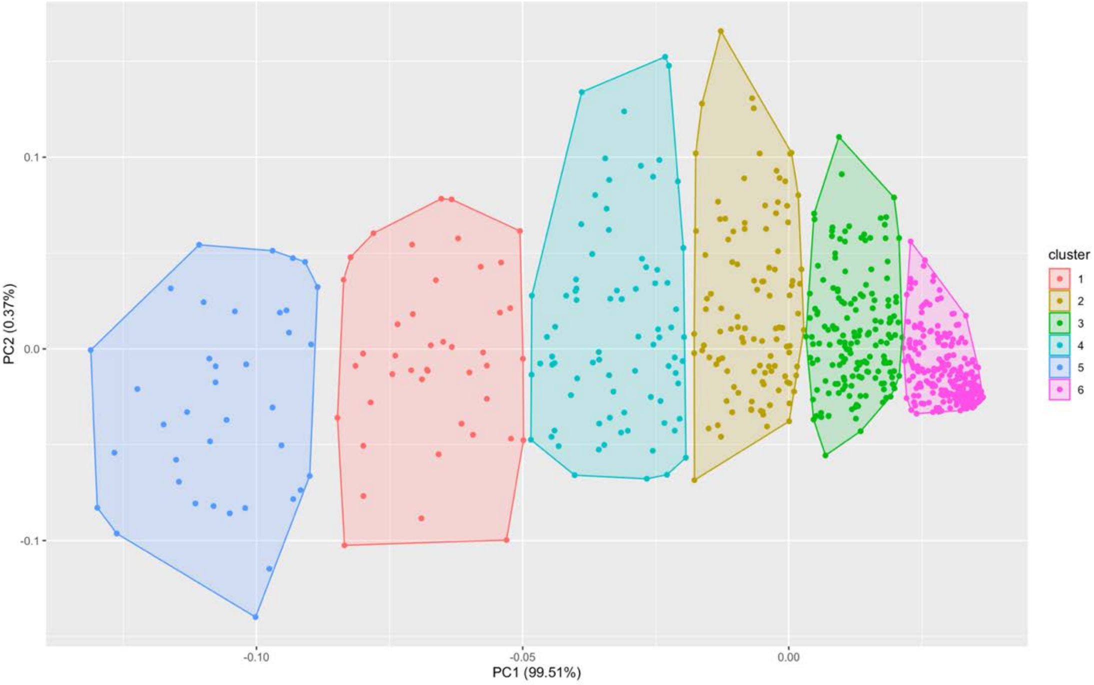
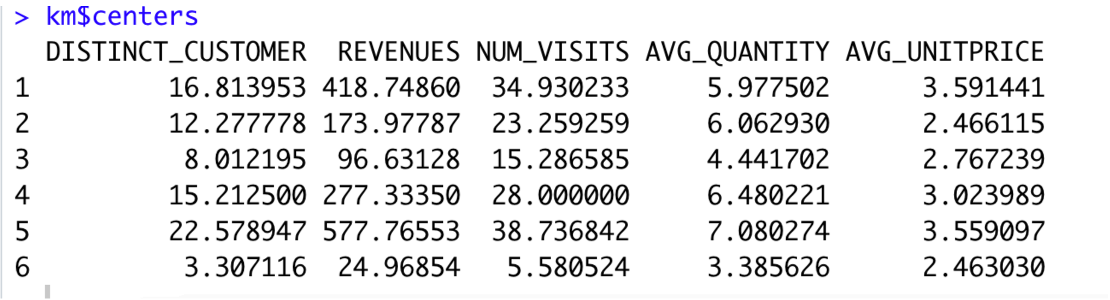


Figure-20 Clusters of Products



```
> km$centers
  DISTINCT_CUSTOMER  REVENUES NUM_VISITS AVG_QUANTITY AVG_UNITPRICE
1         16.813953 418.74860  34.930233     5.977502      3.591441
2         12.277778 173.97787  23.259259     6.062930      2.466115
3          8.012195  96.63128  15.286585     4.441702      2.767239
4         15.212500 277.33350  28.000000     6.480221      3.023989
5         22.578947 577.76553  38.736842     7.080274      3.559097
6          3.307116  24.96854   5.580524     3.385626      2.463030
```

Figure-21 Six Centroids of Product clusters

We used characteristic profiling to create profile of each cluster.

1. Product which is cheap and buy in 5-6 quantity
2. Most favourite Product among users but it requires many visits and price is high
3. Product has moderate price but requires more visits
4. Product require few visits and give good revenues
5. Product has less price and revenues is also less
6. Product which is not buy by most of the customer but few customer buy it frequently.

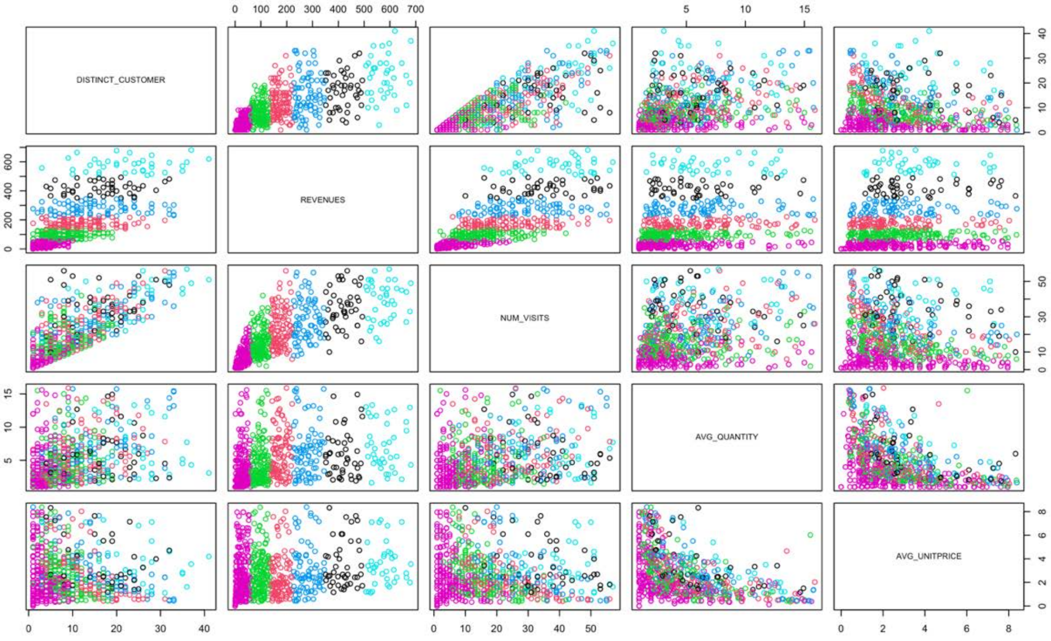


Figure-22 Graphical representation of Product attributes

Conclusion and Next Steps

We have created five profile of customers. First group of Customer, who bought cheap products in bulks for their shop. Second group of Customer buy moderate price products but in less quantity. This group buy only those products which are required in daily routine. Third group come for returning the product. This group use product for temporary need and give it back for refund once they complete their work. Forth group buy lots of things and have frequent visits for shopping. Final group buy expensive products. Products can be divided into six categories. First class of products give good revenues and customers buy these product after second visits. Products of second cluster shows that price and revenue are low and customer buy in good quantity. Third group has moderate price with less revenue and less average quantity. Forth set of products have good revenue, few customer buy it frequently in high quantity. Fifth batch of products play significant role in getting high profit. Last group is less likely among customers though this products have minimum average unit price. These kind of products are one time investment or these products have some quality issues, hence customer not buying them frequently.

Next step is evaluate CSV file of product and Customer. Analyse the files and offer them discount, deals and try to sell products. Few suggestions are already mentioned in Executive summary. One of the analysis is Sixth cluster of product has INFLATABLE POLITICAL GLOBE and it’s one time investment so we can say people are not buying it frequently.

Appendix-A SQL QUERRY

Create a view showing the weekday in which customer bought products at most times:

```
create view week_days as
SELECT
WEEKDAY(InvoiceDateTime) AS WEEKDAY,
COUNT(WEEKDAY(InvoiceDateTime)) AS num,
t.CustomerID
FROM
u49.OnlineRetail t
WHERE
t.CustomerID != 0
GROUP BY CustomerID , WEEKDAY(InvoiceDateTime)
ORDER BY CustomerID,num desc;
```

Then, connect the weekday with customer:

```
SELECT
r.CustomerID,
SUM(r.Quantity) AS TOTAL_QUANTITY,
COUNT(DISTINCT r.StockCode) AS DISTINCT_PRODUCT,
SUM(r.UnitPrice * r.Quantity) AS REVENUES,
COUNT(r.InvoiceDateTime) AS NUM_VISITS,
COUNT(r.Quantity < 0 OR NULL) AS NUM_REFUND,
AVG(r.UnitPrice) AS avg_price_item,
a.WEEKDAY
FROM
u49.OnlineRetail r
LEFT JOIN
(SELECT
WEEKDAY, t.CustomerID
FROM
u49.week_days t
WHERE
t.NUM IN (SELECT
MAX(num)
FROM
u49.week_days a
WHERE
a.CustomerID = t.CustomerID
GROUP BY a.CustomerID)) a ON a.CustomerID = r.CustomerID
WHERE
r.CustomerID != 0
GROUP BY r.CustomerID , a.WEEKDAY
ORDER BY SUM(UnitPrice * Quantity) DESC
LIMIT 2000;
```

Appendix-B R Scripts

```
library(stats)
library(dplyr)
library(ggplot2)
library(ggfortify)
library(GGally)
mydata <- read.csv(file = './customer.csv')

mydata2 <- mydata[mydata$CustomerID != '', ]
#mydata$InvoiceDateTime <- strftime(mydata$InvoiceDateTime, format="%H:%M")
#outlier
ggpairs(mydata2, upper = list(continuous = ggally_points),lower = list(continuous = "points"), title = "customer before outlier removal")
attach(mtcars)
par(mfrow=c(1,2))
boxplot(mydata$TOTAL_QUANTITY, plot=TRUE, main="Total Quantity with Outliers", xlab="Total Quantity", ylab="Frequency")
outliers <- boxplot(mydata$TOTAL_QUANTITY, plot=FALSE)$out
mydata=mydata[-which(mydata$TOTAL_QUANTITY %in% outliers),]
boxplot(mydata$TOTAL_QUANTITY, plot=TRUE, main="Total Quantity without Outliers", xlab="Total Quantity", ylab="Frequency")

attach(mtcars)
par(mfrow=c(1,2))
boxplot(mydata$DISTINCT_PRODUCT, plot=TRUE,main="Distinct Product with Outliers", xlab="Distinct Product", ylab="Frequency")
outliers <- boxplot(mydata$DISTINCT_PRODUCT, plot=FALSE)$out
mydata=mydata[-which(mydata$DISTINCT_PRODUCT %in% outliers),]
boxplot(mydata$DISTINCT_PRODUCT, plot=TRUE, main="Distinct Product without Outliers", xlab="Distinct Product", ylab="Frequency")

attach(mtcars)
par(mfrow=c(1,2))
boxplot(mydata$REVENUES, plot=TRUE, main="Revenues with Outliers", xlab="Revenues", ylab="Frequency")
outliers <- boxplot(mydata$REVENUES, plot=FALSE)$out
mydata=mydata[-which(mydata$REVENUES %in% outliers),]
boxplot(mydata$REVENUES, plot=TRUE, main="Revenues without Outliers", xlab="Revenues", ylab="Frequency")

attach(mtcars)
par(mfrow=c(1,2))
boxplot(mydata$NUM_VISITS, plot=TRUE,main="Number of Visits with Outliers", xlab="Number of Visits", ylab="Frequency")
outliers <- boxplot(mydata$NUM_VISITS, plot=FALSE)$out
mydata=mydata[-which(mydata$NUM_VISITS %in% outliers),]
boxplot(mydata$NUM_VISITS, plot=TRUE, main="Number of Visits without Outliers", xlab="Number of Visits", ylab="Frequency")

attach(mtcars)
par(mfrow=c(1,2))
boxplot(mydata$avg_price_item, plot=TRUE,main="Avg Price with Outliers", xlab="Avg Price", ylab="Frequency")
outliers <- boxplot(mydata$avg_price_item, plot=FALSE)$out
mydata=mydata[-which(mydata$avg_price_item %in% outliers),]
boxplot(mydata$avg_price_item, plot=TRUE, main="Avg Price without Outliers", xlab="Avg Price", ylab="Frequency")

mydata2 <- mydata[2:(length(mydata)) ]
```

```
ggpairs(mydata2, upper = list(continuous = ggally_points),lower = list(continuous = "points"), title = "customer after outlier removal")

mydata.scale = scale(mydata2)

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(mydata.scale,1,50,150))

km=kmeans((mydata2),5,150)
autoplot(km,(mydata), frame=TRUE)
km$centers

clusteredval=cbind(mydata2,km$cluster)
plot(clusteredval[,1:5],col=km$cluster)
write.csv(clusteredval, file = './customer2.csv',col.names = FALSE)

count(subset(clusteredval, km$cluster==1))
count(subset(clusteredval, km$cluster==2))
count(subset(clusteredval, km$cluster==3))
count(subset(clusteredval, km$cluster==4))
count(subset(clusteredval, km$cluster==5))
```

R-script for customers

```
#install.packages('stats')
#install.packages('dplyr')
#install.packages('ggplot2')
#install.packages('ggfortify')
#install.packages('GGally')

library(stats)
library(dplyr)
library(ggplot2)
library(ggfortify)
library(GGally)

mydata <- read.csv(file = './product.csv')
mydata2=mydata[1:(length(mydata)-1)]
ggpairs(mydata2, upper = list(continuous = ggally_points),lower = list(continuous = "points"), title = "Product before outlier removal")
attach(mtcars)
par(mfrow=c(1,2))
#outlier
boxplot(mydata$DISTINCT_CUSTOMER, plot=TRUE, main="Distinct Customer with Outliers", xlab="Distinct Customer", ylab="Frequency")
outliers <- boxplot(mydata$DISTINCT_CUSTOMER, plot=FALSE)$out
mydata=mydata[-which(mydata$DISTINCT_CUSTOMER %in% outliers),]
boxplot(mydata$DISTINCT_CUSTOMER, plot=TRUE, main="Distinct Customer without Outliers", xlab="Distinct Customer", ylab="Frequency")

attach(mtcars)
```

```
par(mfrow=c(1,2))
boxplot(mydata$REVENUES, plot=TRUE,main="Revenues with Outliers", xlab="Revenues", ylab="Frequency")
outliers <- boxplot(mydata$REVENUES, plot=FALSE)$out
mydata=mydata[-which(mydata$REVENUES %in% outliers),]
boxplot(mydata$REVENUES, plot=TRUE, main="Revenues without Outliers", xlab="Revenues",
ylab="Frequency")

attach(mtcars)
par(mfrow=c(1,2))
boxplot(mydata$NUM_VISITS, plot=TRUE,main="Number of visits with Outliers", xlab="Number of visits",
ylab="Frequency")
outliers <- boxplot(mydata$NUM_VISITS, plot=FALSE)$out
mydata=mydata[-which(mydata$NUM_VISITS %in% outliers),]
boxplot(mydata$NUM_VISITS, plot=TRUE, main="Number of visits without Outliers", xlab="Number of visits",
ylab="Frequency")

attach(mtcars)
par(mfrow=c(1,2))
boxplot(mydata$AVG_QUANTITY, plot=TRUE,main="Avg Quantity with Outliers", xlab="Avg Quantity",
ylab="Frequency")
outliers <- boxplot(mydata$AVG_QUANTITY, plot=FALSE)$out
mydata=mydata[-which(mydata$AVG_QUANTITY %in% outliers),]
boxplot(mydata$AVG_QUANTITY, plot=TRUE, main="Avg Quantity without Outliers", xlab="Avg Quantity",
ylab="Frequency")

attach(mtcars)
par(mfrow=c(1,2))
boxplot(mydata$AVG_UNITPRICE, plot=TRUE,main="Avg Price with Outliers", xlab="Avg Price",
ylab="Frequency")
outliers <- boxplot(mydata$AVG_UNITPRICE, plot=FALSE)$out
mydata=mydata[-which(mydata$AVG_UNITPRICE %in% outliers),]
boxplot(mydata$AVG_UNITPRICE, plot=TRUE, main="Avg Price without Outliers", xlab="Avg Price",
ylab="Frequency")

mydata2=mydata[1:(length(mydata)-1)]
ggpairs(mydata2, upper = list(continuous = ggally_points),lower = list(continuous = "points"), title = "Product
after outlier removal")

mydata.scale = scale(mydata2)

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(mydata.scale,1,50,150))

km=kmeans(mydata2,6,150)
autoplot(km,mydata2, frame=TRUE)
km$centers

clusteredval=cbind(mydata2,km$cluster)
write.csv(clusteredval, file = './product2.csv', col.names = "FALSE")
plot(clusteredval[,1:5],col=km$cluster)
```

```
count(subset(clusteredval, km$cluster==1))
count(subset(clusteredval, km$cluster==2))
count(subset(clusteredval, km$cluster==3))
count(subset(clusteredval, km$cluster==4))
count(subset(clusteredval, km$cluster==5))
count(subset(clusteredval, km$cluster==6))
```

R-script for products