MCDA 5580

Assignment-1

Online Retail Cluster Analysis Report

Authors:

Thapa, Vedant - A00457249

James, Rubin - A00455851

Singh, Karnjot - A00457246

Table of Contents

1. Executive Summary

The report contains a detailed analysis of online retail data for the purpose of inferring customer behavior and product trends. The source dataset has been restricted to the region of the United Kingdom and spans over a period of 1 year from 2010 to 2011. The analysis has been carried out using the k-means algorithm and the customer and product data have been segmented into 4 respectively. The optimal number of clusters was determined using the elbow plot. A certain degree of parity and homogeneity can be deduced from customers and products of a particular segment. A summary of the findings is documented below:

Customer Segments:

	Customer Share	Description
Crème de la crème	773 (19.76%)	High loyalty customers, visit often and purchase a wide number of products
The Contenders	1324 (33.84%)	Purchase significant amount of products, generate good revenue and have the potential of becoming loyal customers
The Sale Seekers	1342(34.30%)	The frequency of purchase and visits is quite low, average spending habits is low
The Blue Moons	473(12.10%)	Rarely purchase any products, spending is inadequate and been a long time since they purchased

Product Segments:

	Customer Share	Description
Jewel in the Crown	818 (22.54%)	Highest selling products, very popular and generate the most revenue
The Budding Artists	1265 (34.85%)	Potentially high selling products, purchase quite often by customers and generates significant revenue
The Low Takers	964 (26.56%)	Low popularity products, generates modest revenue.
The Unknowns	582 (16.03%)	Rarely bought by customers and generate a negligible share of the revenue.

The segments can be used to comprehend the different categories of customers and market habits followed by them. The product segmentation helps stakeholders perceive the merits of various items. The report can be utilized by stakeholders to visualize and predict the different qualitative and quantitative indicators that affect the revenue and implement actionable measures to maximize the same.

2. Objective

An Online Retail store has invited to analyze its sales data with an intention to understand its customers better by inferring their shopping habits. They are also interested in exploring the different product trends that directly impact logistics. The authors of the report endeavor to provide a detailed analysis by performing data preparation, data cleansing, feature selection, feature engineering, and application of established algorithms to segment the data into insightful customer and product clusters. It is the authors' understanding that the purpose of the customer segmentation is to identify the different classes of customers, that will be utilized by stakeholders for better marketing strategies and the purpose of product segmentation is to ascertain the products that garner maximum revenue, phase out low selling products and develop insights that would aid administration in acting in the interests of the stakeholders.

3. About The Data:

The online retail store has provided data about the transactions that were recorded from 1st December 2010 to 09 December 2011. The dataset consists of customer information and product sales. A summary of the table is as follows:

No of records: 541909

Column Descriptions

InvoiceNo
Type: Integer
The column consists of all invoices generated for the above time frame. It does not contain unique values as a record exists for each item in the invoice.

StockCode
Type: Varchar(6)
The column consists of all items that were purchased/returned in the transactions that occurred for the specified time frame.

Description
Type: Varchar(35)
The column consists of descriptions and names of items in the transactions.

Quantity
Type: Integer
The column specifies the count of a specific item in a transaction. A negative value for quantity is assumed that the product has been returned.

InvoiceDate
Type: Varchar(14)
The column records the date on which the particular invoice was generated.

UnitPrice
Type: Decimal(3,2)
The column specifies the cost of a singular unit of a particular item.

CustomerID
Type: Integer
The customer IDs are recorded in this column. It contains non-unique values as an entry exists for each stockcode in a particular invoice.

Country
Type: Varchar(14)
The country where the transaction occurred is recorded in this column.

InvoiceDateTime
Type: DateTime
The column consists of the date along with the time component on which the invoice was generated.

4. Design/Methodology/Approach

There are a lot of ways to find valuable information or patterns from datasets like using Excel, SQL, plots, etc. But when are working with large volumes of data, It is almost impossible to find patterns manually. However, it all becomes very easy when Machine Learning is introduced.

Machine Learning provides a lot of different algorithms which use complex mathematics and statistics to find patterns and information from vast amounts of data. Here data can be in any format, it can be text, documents, database files, sensor data, system logs, images, videos, or anything which can be stored digitally. If it is digital, it can be processed by ML algorithms to find insights into that data. Since data can be of diverse types, ML provides us three categories of algorithms -

- *Supervised Learning:* This type is useful when it is already known what to look for in the data. Basically, when data is well labeled, supervised learning is used to extract possible patterns and predict the outcome for a similar set of data distribution.
- *Unsupervised learning:* When dealing with data where little is known about what to look for, unsupervised learning takes responsibility to find hidden patterns from data. It helps in finding patterns in unlabelled datasets.
- *Reinforcement learning:* It is a technique in which an algorithm learns by trial and error method. Data is provided and then it works on feedback. Algorithms predict and then rely on feedback to determine if it is predicted correctly or not and keeps learning and improving.

To find the patterns in the provided OnlineRetail dataset, an unsupervised learning technique was used. Clustering groups unlabeled data based on differences and similarities. Since OnlineRetail data was also one of the unlabeled datasets, clustering is the best approach to find information. Clustering finds information by naturally grouping the data based on the inputs provided to it.

For the OnlineRetail dataset, the authors employ the K-means algorithm that groups data points into K groups, where K denotes the number of clusters. All clusters have their own centroids and all the data points within those clusters are desired to be near to their centroids. Each cluster represents a distinct category.

A big K value indicates a large number of little clusters, while a small K value indicates a large number of large clusters. It is not intelligent to directly use K-means on raw data. Data needs to be processed first to make sure that it will give us correct results. In any event, finding and removing outliers before employing K-means is crucial since these data points might have a significant influence on the final findings. But before that, data will also be processed and refined to be free of any missing values, incorrect data types, and more.

Relevant columns are first extracted using SQL and then other preprocessing steps such as data transformations and centering are applied. Results are viewed using density and pair plots. Then, optimal k values are determined before clustering. Authors use the Elbow method to find the optimal value for k. Finally, data is clustered using K-means, and results are interpreted and presented below.

5. Feature Selection / Engineering / Definition

5.1. Assumptions

Before starting out with the clustering, it is crucial to select the appropriate features from the dataset. Since the majority of the transactions in the data were from the United Kingdom (nearly 91%), the authors have excluded other

Appendix-B.

transactions from other countries as customer spending patterns and product demand may vary significantly across different countries.

Furthermore, transactions, where Customer ID is equal to 0, are excluded because it is assumed that those transactions are anonymous transactions i.e, buyers who do not have an account with the store and such buyers may distort the patterns in the store’s customers. Transactions, where InvoiceNo is equal to 0, are assumed to be returns or internal transactions of the store.

After grouping, it was observed that there were few instances where the derived feature, average revenue, had negative values. Such instances were observed to be returned/canceled transactions, i.e, customers who had purchased the product before the date range of the dataset were either returning/canceling their orders.

5.2. Selection

The authors engineer new features before clustering. The features were derived in accordance with customers and products.
The features related to customers are -

- Total Products
- Baskets
- Total revenue
- Average spent
- Days since last purchase

The features related to products are -

- Customer count
- Total product revenue
- Popularity
- Average revenue

5.3. Engineering

To derive those features, SQL queries are implemented. For product data, groups are made on the basis of distinct “StockCode”. Moreover, aggregate functions are used to get the count of distinct customers, total revenue, total quantity, and average revenue.

Similarly, rationale is used to extract customer data, distinct “CustomerID” and aggregation functions are used to get total products, orders, revenue, and average revenue. To view the implementation of our SQL queries please refer to Appendix-B.

5.4. Definition

The extracted features are defined in the following table -

Feature Type	Name of feature	Definition of feature	How is it selected?
Customer	TOTAL_PRODS	Total number of products bought by each customer	It is selected by summing up all the product quantity
	BASKETS	Total number of unique products in a customer's bag	By counting distinct StockCode
	TOTAL REVENUE	Total revenue generated by each customer	By summing up the product of quantity and unit price
	NO_OF_VISITS	Total number of times each customer shopped	By counting distinct InvoiceNo
	AVG_SPEND	Average revenue generated per order by each customer	By dividing total revenue by total baskets
	DAYS_SINCE_LSAT_PURCHASE	Number of days since the last purchase of a customer	By finding the difference between the last transaction of the customer and the reference date i.e, 2011-12-31 00:00:00
Product	CUST_COUNT	Total customers bought each product	By counting distinct customers who bought the product
	TOTALPRODREVENUE	Total revenue generated by each product	By summing up the product of units sold and unit price of the product
	POPULARITY	All baskets that had this product	Sum of all the InvoiceNo that included that product
	AVG_REVENUE	Average revenue generated by the sale of each product	By dividing the total revenue by the total quantity of the product

6. Data Cleansing and Outlier Treatment

6.1. Customer Data

6.1.1. Incorrect Data Types

As the first step in data cleaning, we ensure that all variables have the correct data types i.e, since all the variables that are being used are continuous in nature, the expectation is that they will be either an integer or a numeric data type in R. However while importing the dataset it was observed that variables containing decimal values such as “revenue” and “avg_spend” were being read as “character” data type. Hence such variables were cast to their relevant (numeric) data types.

6.1.2. Missing Information

Since K-Means cannot handle null values in the dataset, any observation even with one missing dimension must be specially handled. Nonetheless, as shown in Figure 3.1.2, it was discovered that the dataset had all of the necessary information.

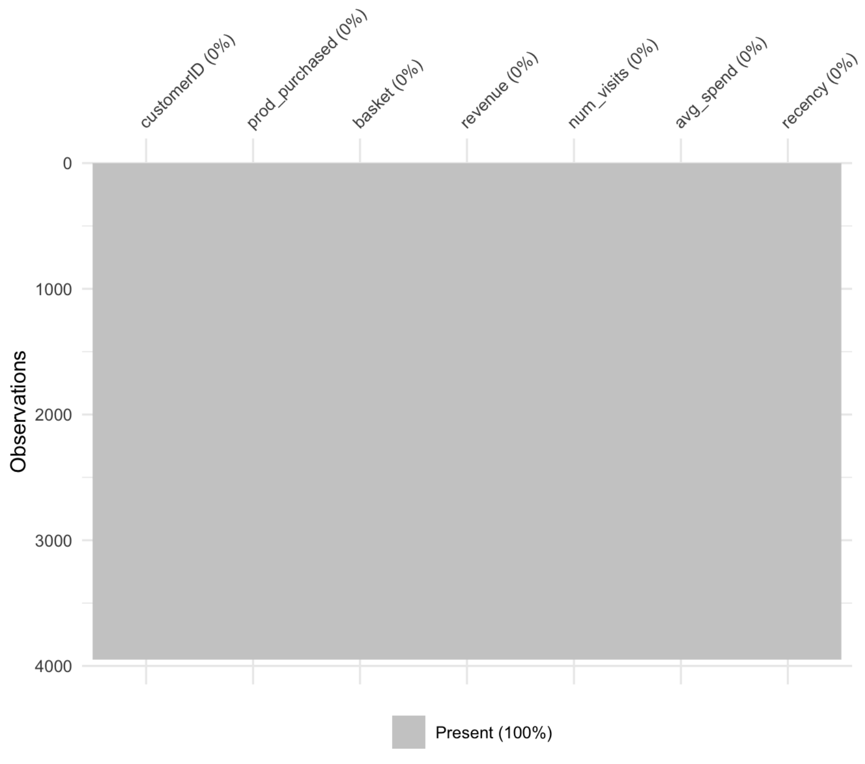


Figure 3.1.2.1

6.1.3. Data Transformation and Standardization

K-Means is an unsupervised machine learning algorithm that segments data points into k clusters by calculating the Euclidean distance between the individual data points and the centroids. This distance computation in K-Means weighs each dimension equally and therefore, extra care is needed to ensure that the units of dimension should be on a similar scale. However, as shown in Figure 3.1.3.1, it was noticed that not only did the scale of the variables varied differently but also their distributions were heavily skewed towards the right.

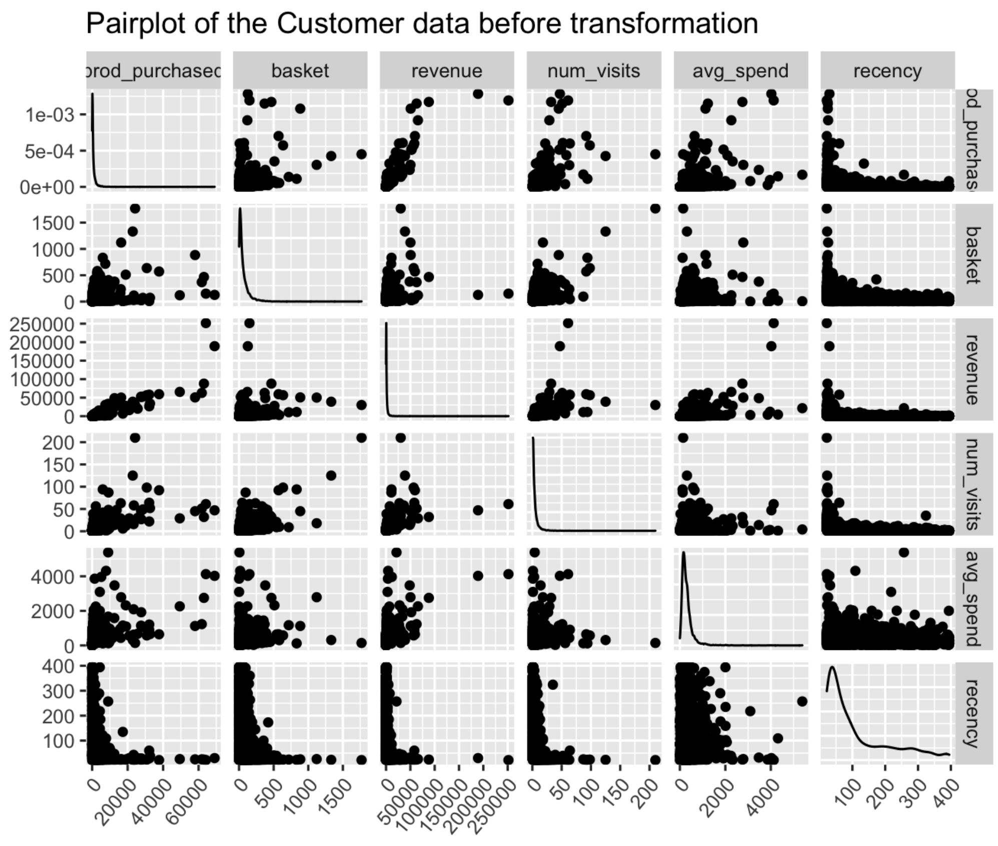


Figure 3.1.3.1

It is also evident from the above plot (Fig. 3.1.3.1) that the majority of the features have outliers and since K-Means is based on distance computation, it is extremely sensitive to them. Upon close investigation of these outliers, it is observed that these data points were not erroneous points, in fact, they were genuine customers with valid transactional data. Excluding them from the dataset would mean losing out on occasional spending patterns of the customers which were not desired. The authors, therefore, opt for a more sophisticated approach of applying the transformation to the data to reduce the effect of such outliers.

Upon transformation of the data by taking a log of each variable, the effect of the outliers on the overall dataset is reduced. Moreover, the distribution of the data also appears to be more Gaussian-like. Even so, the variables still have different ranges and different means which may affect final clusters in K-Means. Therefore, scaling is applied by centering the dataset to a mean value of 0. This method gives the most distinct clusters as compared to other techniques like standardization and normalization. Figure 3.1.3.2 compares the distribution of the variables before and after applying transformation and centering.

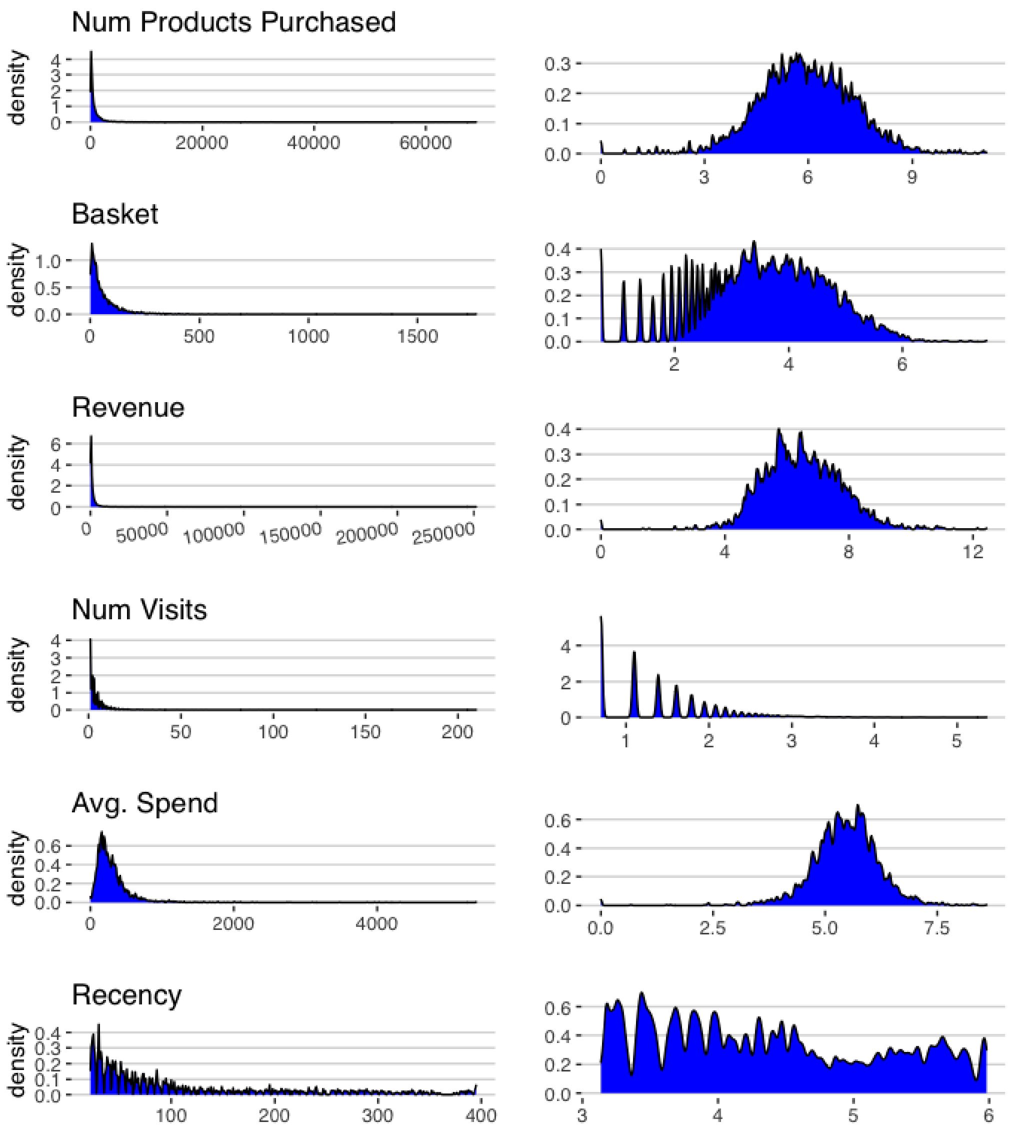


Figure 3.1.3.2

6.2. Product Data

6.2.1. Incorrect Data Types

Similar to the Customer data, while importing the dataset it was observed that variables containing decimal values such as “revenue” and “avg_spend” were being read as “character” data types. Hence such variables were cast to their relevant (numeric) data types.

6.2.2. Missing Information

The Product dataset also did not have any missing information. This can be verified by referring to the plot (Figure 3.2.1) below.

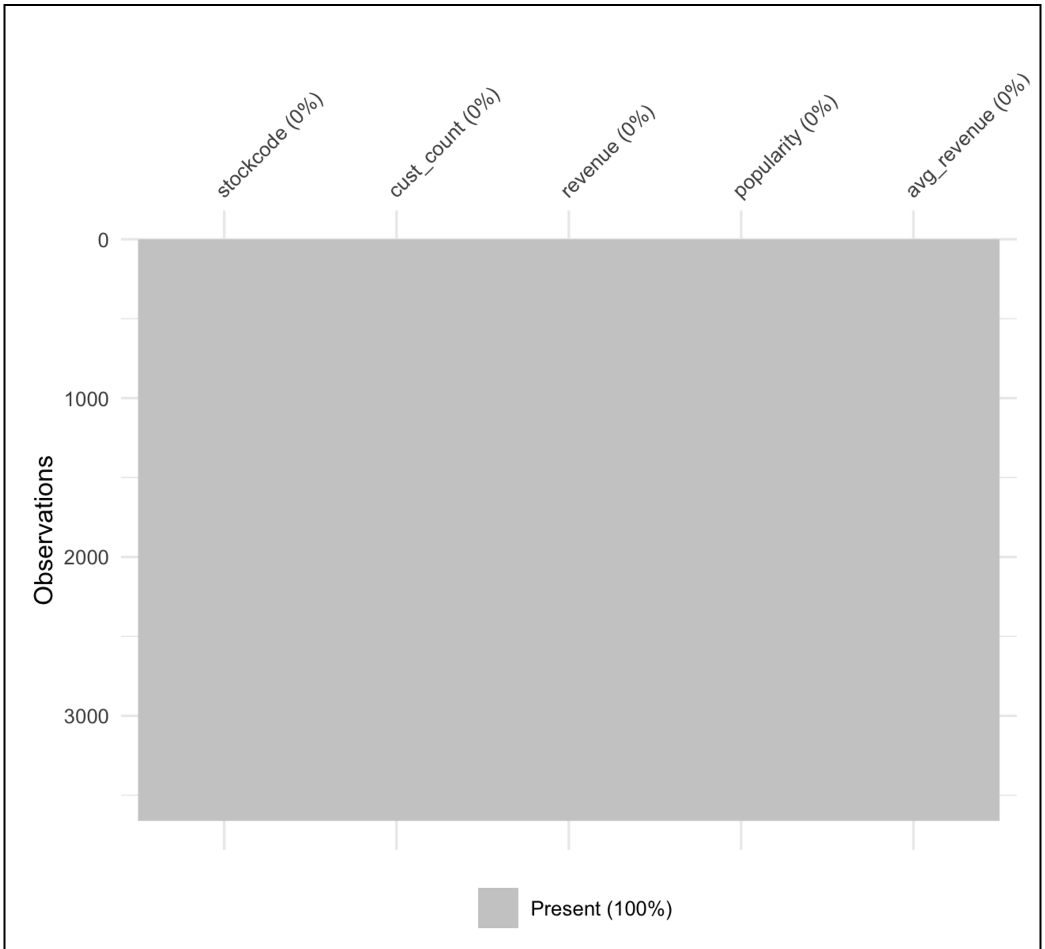


Figure 3.2.2.1

6.2.3. Data Transformation and Standardization

The variables in the Product Dataset were distributed similarly to those in the Customer Dataset. Furthermore, the same rationale is applied for dealing with outliers here as well. That is, the authors intend to preserve the patterns in occasional purchases of the products. Figure 3.2.3.1 displays a pair plot of the data.

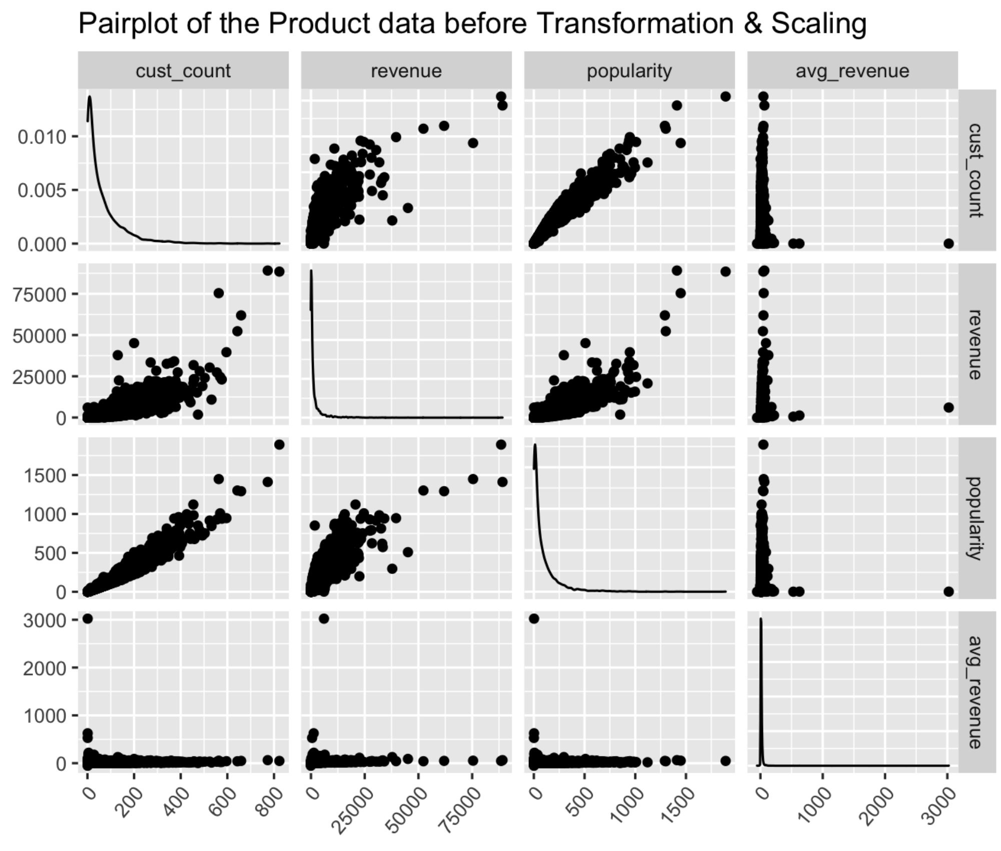


Figure 3.2.3.1

Figure 3.1.3.2 compares the distribution of the variables before and after applying transformation and centering.

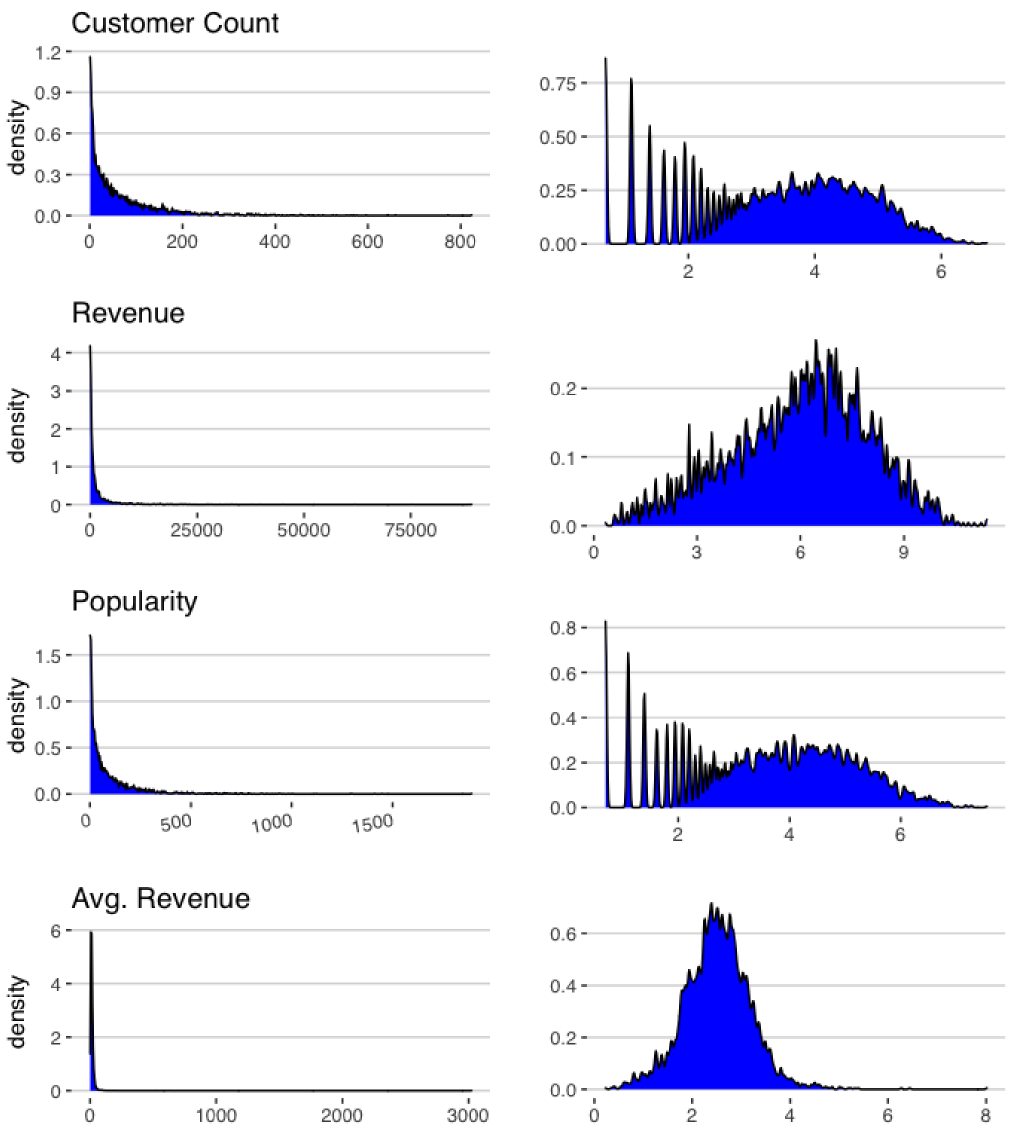


Figure 3.1.3.2

7. Cluster Analysis

The purpose of performing clustering is to segment customers and products into relatable brackets. As part of cluster analysis, the authors have performed k-means multiple times using random centroids to obtain a suitable number of clusters.

The elbow plot was selected as the method of determining the optimum number of clusters by the authors. The elbow plot gives us a visualization of the sum of squared error against different k-values. The most favorable k-value is selected when the total sum of squared errors starts becoming minimal or moves parallel to the horizontal axis.

The elbow plot was recorded separately for the customer and product segments and shows that the optimal k-values are 4 respectively. Refer to Figure 4.1.

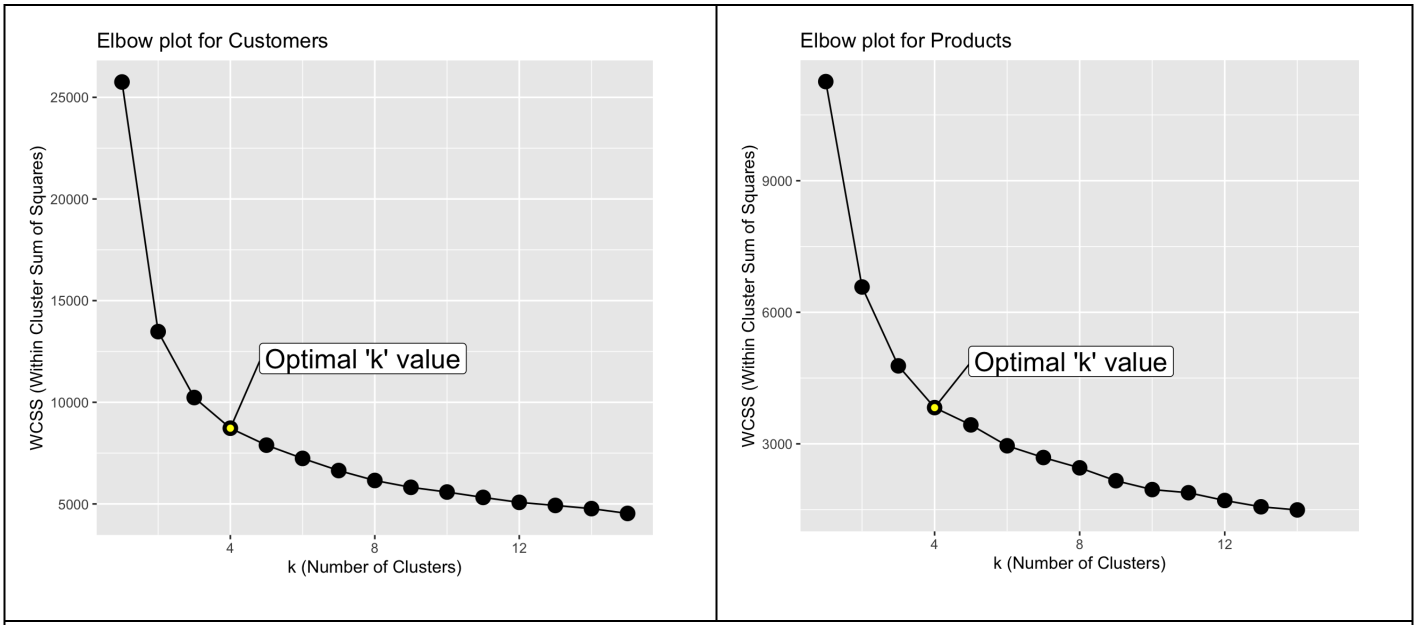
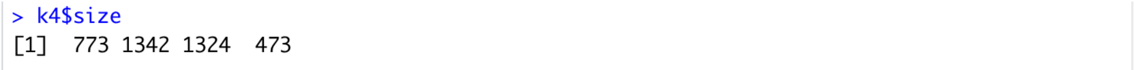


Figure 4.1

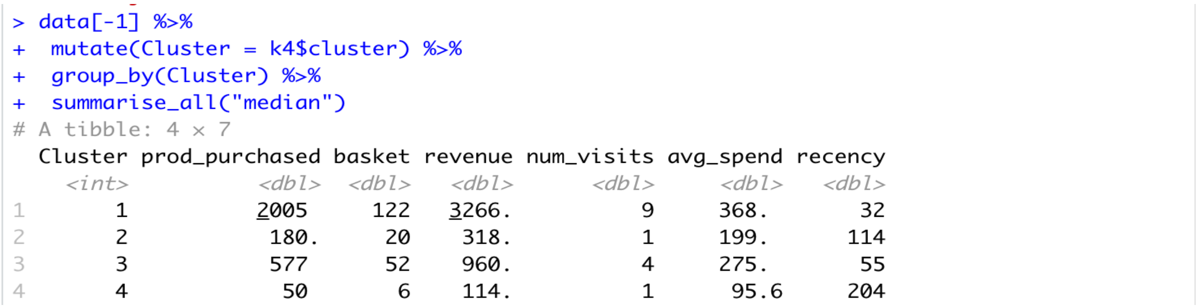
8. Cluster Profiling

Post obtaining the suitable k values, the clusters are formed for customers and products. Based on these, analysis was conducted to classify the customers and products.

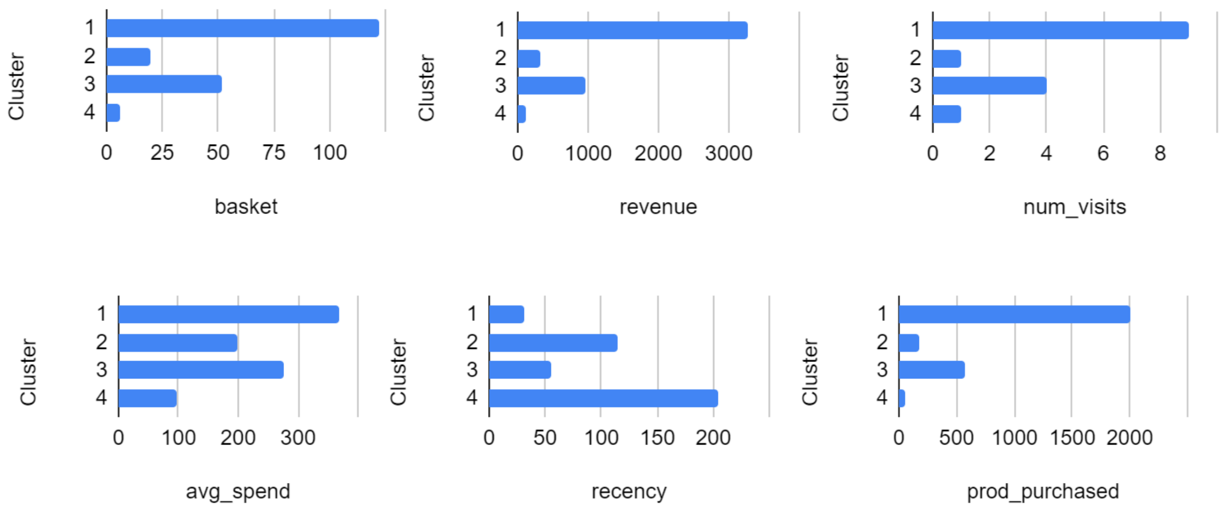
8.1. Customer Clusters

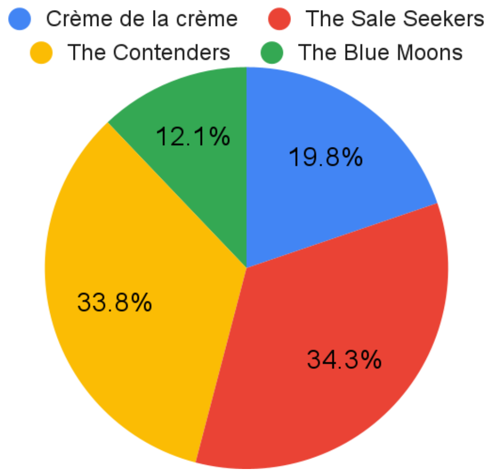


```
> k4$size
[1]  773 1342 1324  473
```



```
> data[-1] %>%
+  mutate(Cluster = k4$cluster) %>%
+  group_by(Cluster) %>%
+  summarise_all("median")
# A tibble: 4 × 7
  Cluster prod_purchased basket revenue num_visits avg_spend recency
    <int>          <dbl>  <dbl>   <dbl>      <dbl>     <dbl>   <dbl>
1       1           2005    122   3266.          9     368.       32
2       2           180.     20    318.          1     199.      114
3       3            577     52    960.          4     275.       55
4       4             50      6    114.          1      95.6     204
```





Customer Segment	Insights	Recommendations
Cluster 1 **Crème de la crème**	• Top Spenders • High-frequency visitors • Most recent visits • Highest revenue generators • Most products purchased • Highly diversified product purchases	Highly loyal customers, no immediate actions required.
Cluster 3 **The Contenders**	• Moderate frequency visitors • Significant number of products purchased • Decent selection of products • Potentially high revenue generators	Customers have a very high potential to become the top segment. Roll out offers on products seldomly purchased to attract them.

Cluster 2 **The Sale Seekers**	● Poor frequency of purchase ● Low average spend ● Narrow choice of products ● Modest revenue generators	Need to engage these customers more by rolling out frequent promotions and offers on a variety of products.
Cluster 4 **The Blue Moons**	● Poor frequency of purchase ● Abysmal spending habits ● Minimal choice of products ● Inferior revenue generators ● Long time since their last visit	These customers require immediate attention as there is a risk of losing them. Product discounts and combos can capture the customer base.

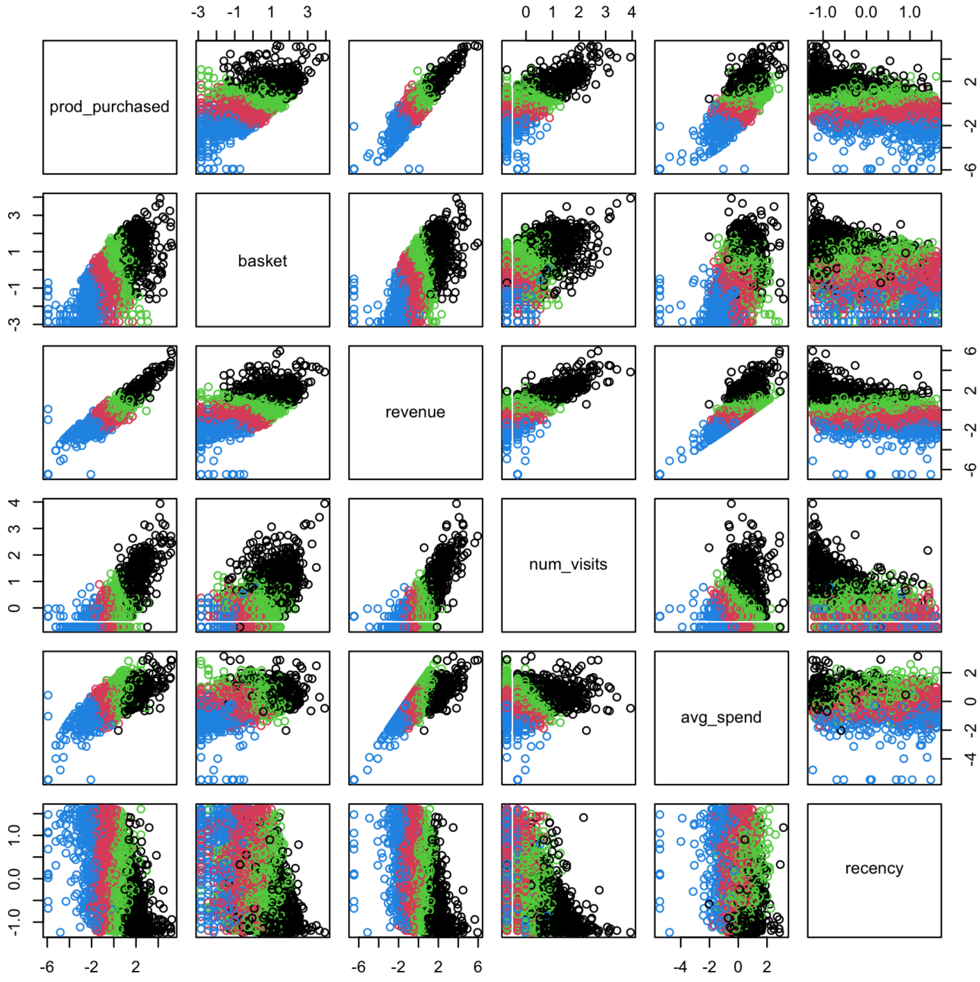
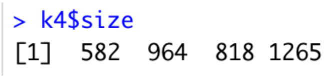
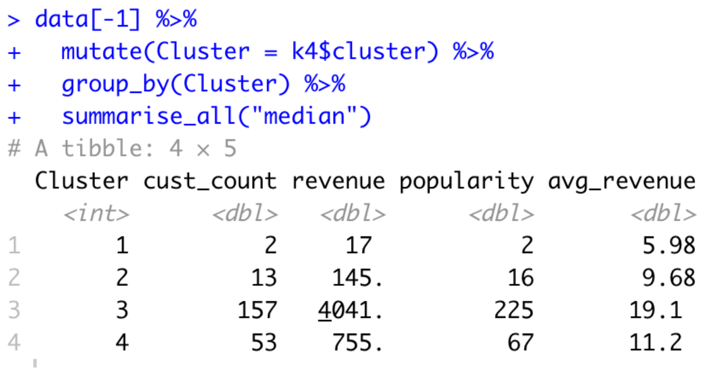


Figure 8.1.1 : Customer Cluster Distribution Across Attributes

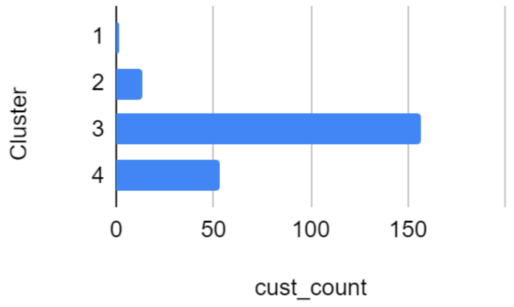
8.2. Product Clusters

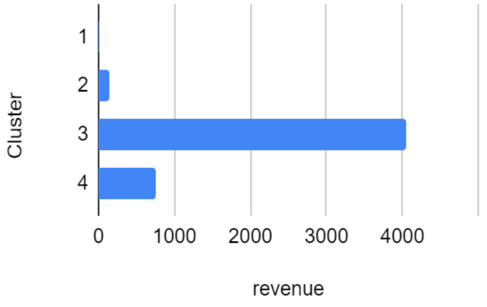


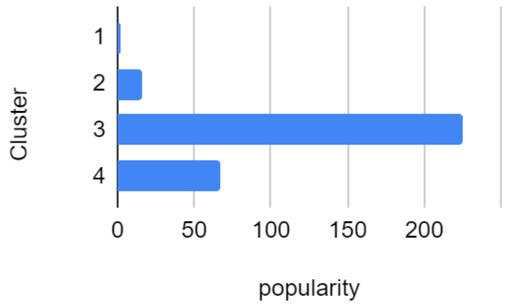
```
> k4$size
[1]  582  964  818 1265
```

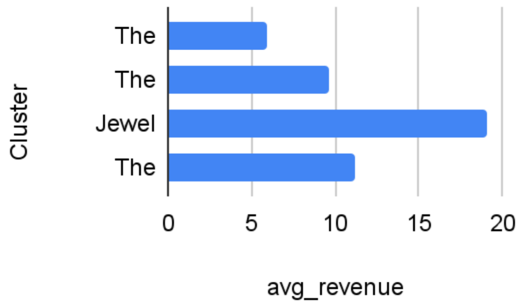


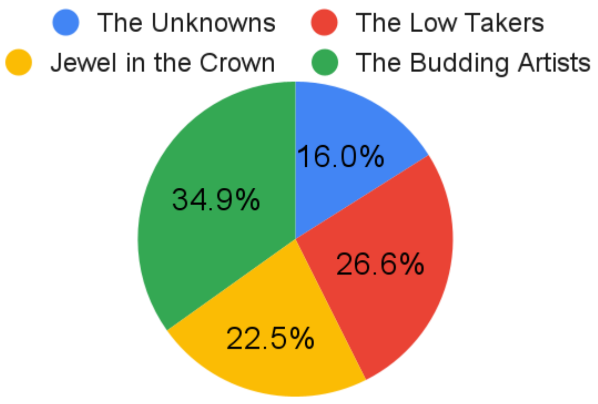
```
> data[-1] %>%
+   mutate(Cluster = k4$cluster) %>%
+   group_by(Cluster) %>%
+   summarise_all("median")
# A tibble: 4 × 5
  Cluster cust_count revenue popularity avg_revenue
    <int>      <dbl>   <dbl>      <dbl>       <dbl>
1       1          2     17           2        5.98
2       2         13    145.         16        9.68
3       3        157   4041.        225       19.1 
4       4         53    755.         67       11.2 
```











Product Segment	Insights	Recommendations
Cluster 3 **Jewel in the Crown**	● Highest revenue generators ● Attracted the most number of customers ● Very high popularity ● Highest average revenue	Very popular products, no prompt actions required.
Cluster 4 **The Budding Artists**	● Moderate popularity ● Significant amount of revenue generated ● A fair number of customers captured ● Potential average revenue generators	The products have the potential of becoming top selling products via promotions that would engage customers.
Cluster 2 **The Low Takers**	● Low number of customers attracted ● Low revenue generated ● insufficient popularity	Need to roll out offers, discounts, and product bundles to improve the visibility of these products amongst customers.
Cluster 1 **The Unknowns**	● Poor frequency of purchase ● Abysmal spending habits ● Minimal choice of products ● Inferior revenue generators ● Long time since their last visit	Need to be shelved out as a negligible percentage of revenue being generated through these products.

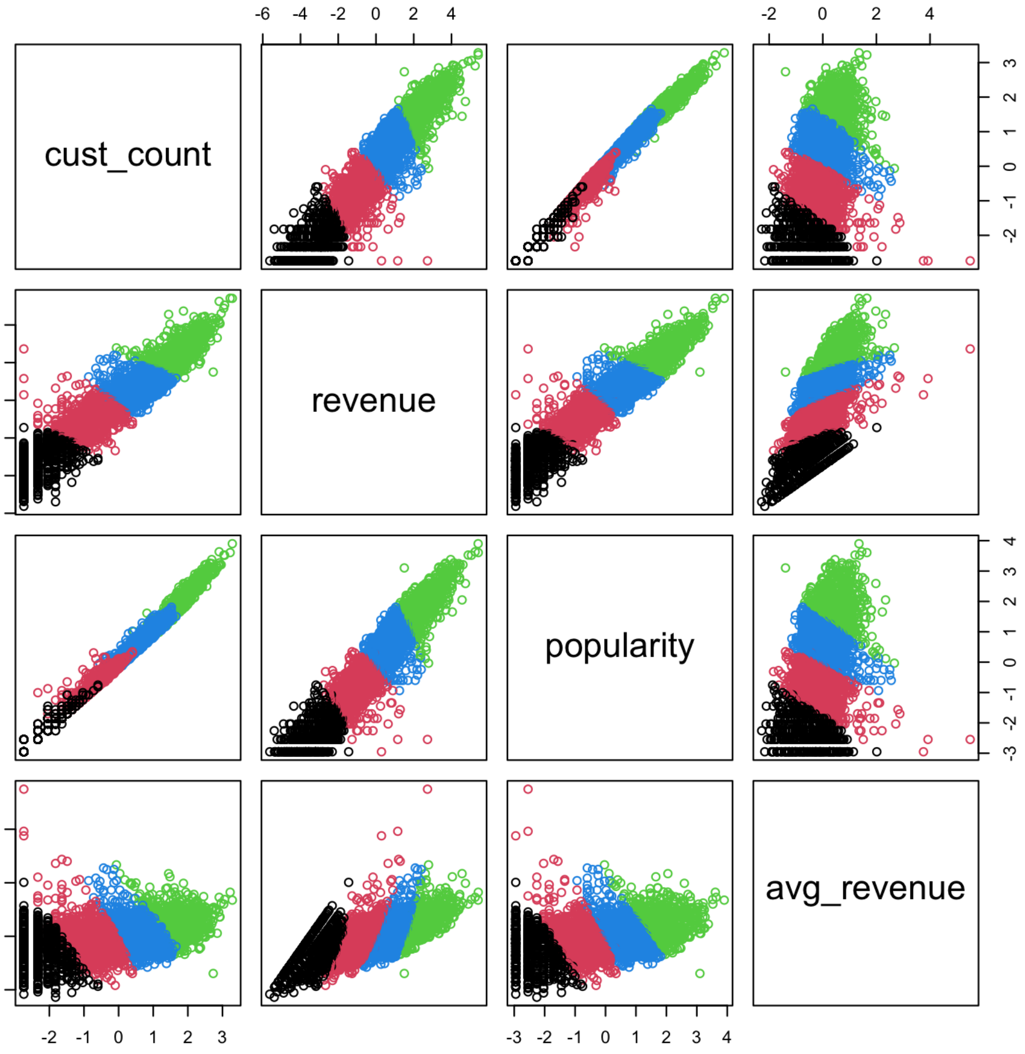


Figure 8.2.1 : Product Cluster Distribution Across Attributes

9. Conclusion and Next Steps

The analysis conducted by the authors has produced string actionable insights, easily comprehensible data tables, and well-defined visualizations, which was the intended objective. The mining and subsequent findings of the online retail store data set can be used by the stakeholders for inferring customer behavior & product sales and implementing administrative decisions that would affect the social and economic indicators in alignment with the organization’s values and interests.

The customer and product data were segmented into 4 categories, which were derived through the implementation of the elbow method. Outlier treatment was conducted using log transformation and scaling that would reduce the effects of outliers. The plots of the raw data set, transformed data set and clusters help in visualizing the procedures that were performed.

Data sampling was performed by the authors to focus on a particular region and assumptions were made due to the absence of additional information required to conduct the analysis. To enhance the studies, it is suggested that the authors be provided with qualitative data such as demographic data to help analyze customer trends in a more efficient manner and design effective targeted product sales and promotions

10. Appendix-A [References]

01. https://www.rdocumentation.org/
02. https://www.researchgate.net/publication/263376827_The_evolution_of_direct_data_and_digital_marketing
03. https://jakevdp.github.io/PythonDataScienceHandbook/05.11-k-means.html
04. https://jtr13.github.io/cc20/customized-plot-matrix-pairs-and-ggpairs.html
05. https://statsandr.com/blog/clustering-analysis-k-means-and-hierarchical-clustering-by-hand-and-in-r/#visualizations

11. Appendix-B [SQL - Queries]

```sql
-- Customer data
SELECT CustomerID,
        sum(Quantity) as TOTAL_PRODS,
        count(distinct StockCode) as BASKETS,
        sum(Quantity*UnitPrice) as TOTALREVENUE,
        count(distinct InvoiceNo) as NO_OF_VISITS,
        (select sum(Quantity*UnitPrice)/count(distinct InvoiceNo)) as
AVG_SPEND,
        DATEDIFF("2011-12-31 00:00:00", max(InvoiceDateTime)) as
DAYS_SINCE_LAST_PURCHASE
FROM OnlineRetail
WHERE UnitPrice>0
        AND CustomerID != 0
    AND Country='United Kingdom'
GROUP BY CustomerID
ORDER BY 1;

-- Product data
SELECT StockCode,
    COUNT(DISTINCT CustomerID) as CUST_COUNT,
    SUM(Quantity*UnitPrice) as TOTALPRODREVENUE,
    COUNT(DISTINCT InvoiceNo) as POPULARITY,
    (SELECT SUM(Quantity*UnitPrice)/SUM(Quantity)) as AVG_REVENUE
FROM OnlineRetail
WHERE Country='United Kingdom'
    AND CustomerID != 0
GROUP BY StockCode
ORDER BY TOTALPRODREVENUE DESC;
```

12. Appendix-C [R Code]

12.1. Github

https://github.com/vedantthapa/online-retail-segmentation

12.2. custcluster.R

```
# imports
library(ggplot2)
library(GGally)
library(dplyr)
library(caret)
library(naniar)
library(ggrepel)
source("~/Downloads/online-retail-segmentation/utils.r")

# read data
data =
read.csv2("~/Downloads/online-retail-segmentation/custCluster/cus
tcluster.csv",
                  col.names = c("customerID", "prod_purchased",
"basket",
                                "revenue", "num_visits",
"avg_spend", "recency"),
                  stringsAsFactors = FALSE)
head(data, 2)
dim(data)

# convert columns to numeric
sapply(data, class)
cols.num = c("revenue", "avg_spend")
data[cols.num] = sapply(data[cols.num],as.numeric)
sapply(data, class)

# checking for missing values
vis_miss(data)

# keep only revenue > 0
data = data[data$revenue >= 0, ]
dim(data)
```

```r
ggpairs(data[which(names(data) != "customerID")],
        upper = list(continuous = ggally_points),
        lower = list(continuous = "points"),
        title = "Pairplot of the Customer data before
transformation")+
        theme(axis.text.x = element_text(angle = 50, hjust = 1))

# Apply transformation
data.transformed = log(1 + data)

# centering data
pp = preProcess(data.transformed[-1], method = c("center"))
data.scale = predict(pp, as.data.frame(data.transformed[-1]))

summary(data.scale)

# plot elbow
err = multiKmeans(data.scale, 1, 15, 1000)
err = as.data.frame(err)
err$k = seq(1:15)

ggplot(err, aes(x = k, y = err)) +
  geom_line() +
  geom_point(size = 4) +
  geom_label_repel(aes(label = "Optimal 'k' value"),
                   data = subset(err, k == 4),
                   box.padding = 6,
                   position = position_dodge(width=0.9),
                   size = 6) +
  geom_point(data = subset(err, k == 4), color = "blue") +
  ggtitle("Elbow plot for Customers") +
  ylab("WCSS (Within Cluster Sum of Squares)") +
  xlab("k (Number of Clusters)")

# run kmeans with optimal k
set.seed(42)
k4 = kmeans(data.scale, centers = 4, iter.max = 1000)
k4$size

# view median for each feature
data[-1] %>%
 mutate(Cluster = k4$cluster) %>%
 group_by(Cluster) %>%
 summarise_all("median")

# visualize clusters
fviz_cluster(k4, data = data.scale)
plot(data.scale, col = k4$cluster)

# save the results
data$cluster = k4$cluster
write.csv(data,
"~/Downloads/online-retail-segmentation/custCluster/Cluster.csv",
          row.names = FALSE)
```

12.3. prodcluster.R

```
# imports
library(ggplot2)
library(GGally)
library(dplyr)
library(caret)
library(factoextra)
source("~/Downloads/online-retail-segmentation/utils.r")

# read data
data =
read.csv2("Downloads/online-retail-segmentation/prodCluster/prodclus
ter.csv",
                 col.names = c("stockcode", "cust_count", "revenue",
                               "popularity", "avg_revenue"),
                 stringsAsFactors = FALSE)


head(data, 2)
dim(data)

# convert columns to numeric
sapply(data, class)
cols.num = c("revenue", "avg_revenue")
data[cols.num] = sapply(data[cols.num],as.numeric)
sapply(data, class)
```

```r
# checking for missing values
vis_miss(data)

# filter rows
data = data[!(data$stockcode %in% c('POST', 'D', 'C2', 'DOT', 'M',
'S', 'm', 'PADS', 'B', 'CRUK')),]
dim(data)

data = data[data$avg_revenue > 0, ]
dim(data)

ggpairs(data[which(names(data) != "stockcode")],
        upper = list(continuous = ggally_points),
        lower = list(continuous = "points"),
        title = "Pairplot of the Product data before Transformation
& Scaling")+
        theme(axis.text.x = element_text(angle = 50, hjust = 1))


# apply transformation
data.transformed = log(1 + data[-1])


# centering data
pp = preProcess(data.transformed, method = c("center"))
data.scale = predict(pp, as.data.frame(data.transformed))

summary(data.scale)

# plot elbow
err = multiKmeans(data.scale, 2, 15, 1000)
err = as.data.frame(err)
err$k = seq(2:15)

ggplot(err, aes(x = k, y = err)) +
  geom_line() +
  geom_point(size = 4) +
  geom_label_repel(aes(label = "Optimal 'k' value"),
                   data = subset(err, k == 4),
                   box.padding = 6,
                   position = position_dodge(width=0.9),
                   size = 6) +
  geom_point(data = subset(err, k == 4), color = "blue") +
  ggtitle("Elbow plot for Products") +
  ylab("WCSS (Within Cluster Sum of Squares)") +
  xlab("k (Number of Clusters)") +
  xlim(1, 15)

# train kmeans with optimal k
set.seed(42)
k4 = kmeans(data.scale, centers = 4, iter.max = 1000)
k4$size

# view median for each feature
data[-1] %>%
  mutate(Cluster = k4$cluster) %>%
  group_by(Cluster) %>%
  summarise_all("median")

# visualize clusters
fviz_cluster(k4, data = data.scale)
plot(data.scale, col=k4$cluster)

# save the results
data$cluster = k4$cluster
write.csv(data,
"~/Downloads/online-retail-segmentation/prodCluster/Cluster.csv",
          row.names = FALSE)
```

12.4. utils.R

```
# credits: Trishla Shah (Trishla.Shah@smu.ca)
# train kmeans for different values of k
multiKmeans <- function(data,lo,hi, iter)
{
   err=array((hi-lo+1)*2,dim=c((hi-lo+1),2))
   for(i in lo:hi)
   {
       rowNum=i-lo+1
       err[rowNum,1]=i
       set.seed(42)
       err[rowNum,2]=kmeans(data,i,iter)$tot.withinss
   }
   err[,2]}
```