Customer Segmentation and Product Recommendation using K-means Clustering & Apriori Algorithm

Rutvik Sharedalal
Department of Industrial and Manufacturing Engineering
Pennsylvania State University
University Park, USA.
rns26@psu.edu

***Abstract*— Customer segmentation is the process of classifying the customers into different groups based on the similarities they share with respect to some characteristics relevant to a specific business. It is imperative from a business point of view to evaluate the purchasing behavior of its consumers to increase its profitability and efficiently meeting consumer requirements. Every retail company has the database containing the information about the transactional and demographic attributes of the customer. Data mining techniques can be used to extract the unknown hidden patterns which can then be analyzed to make important decisions for a retail company. This paper elaborates upon the use of data mining techniques such as k-means clustering and apriori algorithm to form customer segmentations and evaluate product preferences. It can later be mapped to the customers' demographic data to generate recommendations. The work flow involves defining the problem statement and discussing the background in the form of literature review of the previous related works. The possible approaches to tackle the problem at hand are discussed and implemented, followed by the results and inferences.**

Keywords—customer segmentation, k-means clustering, apriori algorithm, association rules mining, RFM analysis

I. INTRODUCTION

According to American Management Association, it is five times as much costlier to attract a new customer than retaining the old one [3]. In today's scenario, it is important for any business organization to evaluate its consumer preferences to increase profitability and meet consumer requirements efficiently. Customer segmentation and product recommendation are one of the important tools to successfully achieve the above goal. The key components required for developing proper segmentation is understanding business objectives and observing available consumer information. Proper segmentation can help achieve business goals such as identifying high profitable customer groups, improving product assortment and designing marketing campaigns. Customer segmentation can further be used to generate product recommendation based on the demographic data of the consumers. Recommendation systems are information filtering systems that filter out and recommend products to the consumer based on their purchasing behavior. The essence of the paper is to identify various data analytics procedures and apply them on the dataset at hand to perform customer segmentation and product recommendation.

II. PROBLEM STATEMENT

The dataset in the question has been divided into 2 sections: 1) Transactional data 2) Demographics data.

TransactionID	ItemType	VendorID	ItemID	CustomerID	Week	Day	StoreID	UnitsSold	CouponOrigin	CouponValue	CouponID
1	11	54500	5251	15100503	614	1	1522	1	0	0	1
2	5	17800	818	15101519	614	1	1542	2	0	0	1
2	8	27500	1486	15101519	614	1	1542	1	0	0	1
2	24	36632	2419	15101519	614	1	1542	2	0	0	1
3	2	36200	2352	15105601	614	1	1522	1	0	0	1
3	20	41545	3904	15105601	614	1	1522	1	0	0	1
4	2	27000	1442	15106500	614	1	1522	1	0	0	1
5	4	17800	839	15107144	614	1	1521	1	0	0	1
5	5	16000	624	15107144	614	1	1521	1	23	50	111
5	8	75350	6361	15107144	614	1	1521	1	0	0	1
5	17	28400	1697	15107144	614	1	1521	1	0	0	1
5	20	72870	6047	15107144	614	1	1521	1	0	0	1
5	20	72870	6050	15107144	614	1	1521	1	0	0	1

Figure 1.1- Snapshot of the Transaction Data

Customer ID	Family Siz	Income	Ethnicity	Dogs	Cats	TVs	Newspaper	Male(Age)	Female(Age)	Children	Cable TV
15100115	1	1	1	0	0	1	0	7	6	8	1
15100503	4	8	1	0	0	3	1	3	3	6	1
15100677	4	6	1	0	0	2	1	2	2	1	0
15100768	2	5	1	0	0	2	0	6	6	8	0
15101121	2	4	1	0	1	9	1	6	6	8	0
15101246	2	4	1	0	0	1	1	7	6	8	1
15101519	2	7	1	0	0	9	1	6	6	8	1
15102004	3	7	1	0	0	2	0	5	5	8	1
15102202	2	6	2	0	0	1	0	7	3	3	0
15102319	2	6	1	0	1	1	1	6	6	8	1
15102327	6	6	2	0	0	9	0	5	5	6	1

Figure 1.2 – Snapshot of the Demographics Data

Transaction data shows the purchasing history of various consumers. It consists of the information of item types and its quantity sold, time of purchasing, retail store number and coupon origin along with the amount given in the coupon. On the other hand, the demographics data provides insights about the family size, income, age, working hours, education and occupation of the consumer. Apart from this information, it also shows whether the consumer has newspaper and television subscription or not. The problem at the hand is to cluster the consumers into various groups based on the similarity with respect to some specific characteristics. There are various approaches to perform customer segmentation such as behavioral segmentation, geographic segmentation, demographic segmentation and price segmentation. One of the possible approaches that can be used is behavioral segmentation. The transaction data can be used to perform the market segmentation. Customers can be segmented according to their preferences for a particular set of items as well as their loyalty. Once the clusters are formed, various business strategies such as product assortment, promotional offers, and marketing campaign can be designed for consumers of each cluster. Demographic attributes of the customers can help generate more accurate recommendations.

III. BACKGROUND

In their paper, Pascal et al [1] emphasized on the requirement of applying data mining techniques for performing customer segmentation in the competitive business scenario. According to Pascal et al, more weight should be given to data mining techniques compared to conventional business processes for performing customer segmentation. The technique used by them was k-means clustering. The data used was collected from a mega business retail outlet that has many branches in the Akwa Ibom state of Nigeria. The data consisted of 2 features and 100 tuples representing 100 consumers. The features used were the average amount of goods purchased by consumer per month and the average number of consumer visits per month. 4 customer segments were identified and labeled as: High Buyers-Regular Visitors, High Buyers-Irregular Visitors, Low Buyers-Regular-Visitors and Low Buyers-Irregular Visitors. Purity measure was used to evaluate the validity of clustering. 95% of purity achieved resulting in accurate segmentation [1].

Malik et al [2] introduced a variant of k-means clustering in their paper. They discussed expectation minimization clustering and emphasized on its advantages over the conventional k-means clustering technique. Both the techniques were implemented using WEKA software on a dataset consisting of records of 600 consumers with attributes considered such as age, gender, income, marital status, number of children and car ownership. Expectation-Maximization technique proved better in terms of results when implemented on the dataset.

According to Ching-Hseu Cheng et al, customer relationship management and customer value analysis are vital tools for improving customer loyalty in today's competitive market [3]. As the RFM analysis is one of the important aspects of customer value analysis, Ching-Hseu et al based their model upon RFM analysis which is a widely used marketing strategy. In their model, RFM values corresponding to each consumer were used as inputs to the k-means clustering algorithm. The clusters obtained were labeled in terms of customer loyalty achieved.

Pramod Prasad et al. [4], emphasized on performing 'market basket analysis' to extract interesting patterns from the transaction data available to the retailers. According to them, extracting frequent item sets from the transactional data will help company take important decisions such as what items to put on sale, how to design coupons and how to place merchandise on shelves. They used apriori algorithm on a sample dataset to generate association rules using WEKA software and analyzed which items are frequently bought together.

IV. METHODS

1. k-means clustering

Pascal et al [1] used k-means clustering approach to cluster their dataset into 4 segments. The procedure was divided into 4 steps. In the first step, features were normalized using z-score to bring all the attributes to a common scale. Scaling improves the performance of the clustering algorithm as there is no bias towards any of the attribute. The second step was to generate initialization of the centroids according to the pre-defined number of clusters. Forgy method of random initialization was used for this purpose. The next stage of the procedure was assignment stage. Data points were assigned to a cluster for which the distance of the data point to the cluster centroid was minimum. The last stage was updating stage. Centroids were updated as new clusters formed and finally the algorithm terminated when there was no possibility of changing the position of centroids further. The attributes chosen for this study were the average amount of goods purchased by a customer per month and the average number of visits per month. 4 segments were formed namely High buyer-Regular visitor, High buyer- Irregular visitor, Low buyer – Regular visitor and Low buyer- Irregular visitor. The validity of the segmentation was checked by applying purity measure and 95% purity measure was achieved. Malik et al [2] introduced expectation-maximization algorithm which is a variant of k-means algorithm. It is a soft clustering method and hence elements can belong to multiple clusters with different probabilities of being in each cluster. The elements are clustered to a class for which it has more conditional probability.

2. RFM analysis

Ching-Hseu et al emphasized on using RFM analysis with the combination of k-means clustering to perform customer segmentation based on customer loyalty [3]. In the Recency-Frequency-Monetary analysis, recency stands for the latest date of the purchase; frequency stands for the average number of transaction per period and monetary stands for the average amount spent per period. In their study, Ching-Hseu et al applied this method on the database of a Taiwan based electronics company in the year 2006. All the three factors were given an equal weight and the continuous attributes of recent date of purchase, number of transactions and total amount spent were discretized by assigning a score in the range of 1-5 with 1 denoting very low and 5 denoting very high. Hence, the new attributes obtained were in the combination of discrete values from 1 to 5. K-means clustering was performed using these attribute values as input set and the customers were segmented based on customer loyalty index. 5 clusters were predefined and labeled as: Very high loyalty, High loyalty, Medium Loyalty, Low loyalty and Very low loyalty.

3. Apriori algorithm

This was the approach used by Pramod Prasad et al [4] in their paper on association rule mining for extracting product sales pattern in retail store transactions. They used this algorithm to generate association rules to evaluate which all items are frequently bought together. First of all, they converted the transactional data into market basket form i.e. for a particular transaction and for a particular item type, the attribute value is 1 if item is bought in that transaction and 0 if it is not bought. Then they generated frequent item sets based upon minimum support and confidence values. The threshold value for support and confidence were taken as 40% and 80% respectively. Support for a particular itemset is defined as the fraction of transactions that contain that itemset. Similarly, confidence for a particular item sets X and Y can be defined as a measure of how often items in Y appear in transactions which already contains itemset X. Apriori algorithm works on apriori principle that support of an itemset never exceeds the support of its subset. This helps achieving pruning of the dataset and hence computational time decreases. Now, confidence is only calculated for the item sets that are remained after pruning. These confidence values for each generated rule can be used as a reference to decide all possible frequent item sets which subsequently can be used by the retailers to design promotional offers and generate product recommendation.

V. Plan For Analysis

In our analysis, we will consider 2 approaches: 1) Item-based customer segmentation 2) Recency-Frequency based customer segmentation. In the first approach, we will emphasize more upon evaluating the preferences of the new customer with the help of their demographic attributes. In this analysis, we will perform customer segmentation with choosing product quantities as attributes. Clustering algorithm will be used to generate the clusters and then the formed clusters will be analyzed in terms of the demographic attributes. Hence, if the new customer visits the store, those particular items can be recommended to him. In the second approach, the focus will be on improving the customer loyalty of the current customers. In this analysis, customer segmentation will be performed on the basis of attributes such as customers' frequency of visits to the store and the recency index which shows how recently the customer has been to the store. This analysis will be performed separately on all the 4 retail stores owned by the company. After generating clusters, clusters with high frequency and high recency will be evaluated in terms of the product preferences and coupon use of the customers. These customers can prove to be potential profitable customers to the company. Apriori algorithm will be used to generate association rules which give an idea about the items frequently bought together. This can be used to generate product recommendation for the customers.

VI. Problem Approach-1

Item-based segmentation:

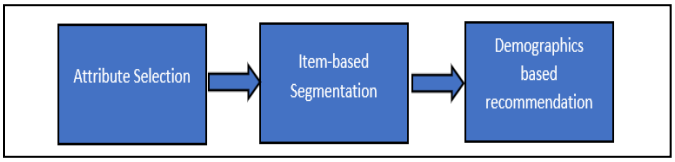

Figure 1.3- Flowchart for item-based segmentation

This approach consists of 3 steps: 1) Attribute selection 2) Item based segmentation 3) Demographics based recommendation. The idea behind this approach is to segment customers based on the quantities of a particular itemset bought by them and evaluating the clusters with reference to the demographic attributes. Then this demographic information can be used to generate product recommendation for the new user with similar demographic attributes.

1)Attribute selection:

In our approach, the attributes considered are the quantities of 2 particular items. We will consider the items cat food and cereals in our analysis.

2)Item based segmentation:

Now, we will perform customer segmentation based on the chosen attributes using k-means clustering algorithm. We will visualize the customers' purchasing behavior and make necessary transformations if required. It will be followed by the scaling of the attributes and generation of clusters.

2.1. Visualizing purchasing behavior:

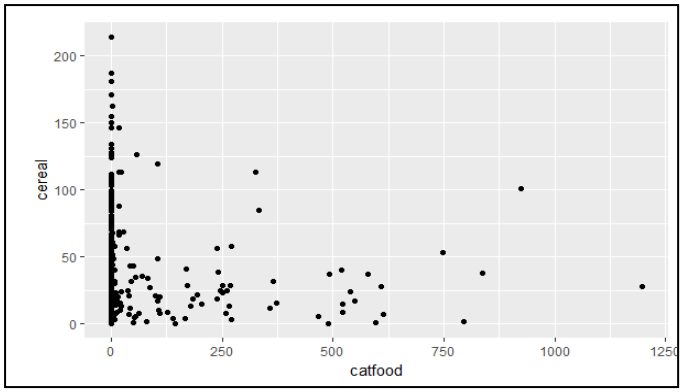

Figure 1.4- Visualizing customers' purchasing behavior

Figure 1.4 shows the purchasing behavior of the customers for 2 items cat food and cereal. It can be seen from the figure that the data is skewed with respect to both the attributes. This explains the Pareto's principle that the 80% of sales are accountable to 20% of company's customers [6].

2.2. Log-transforming the attribute:

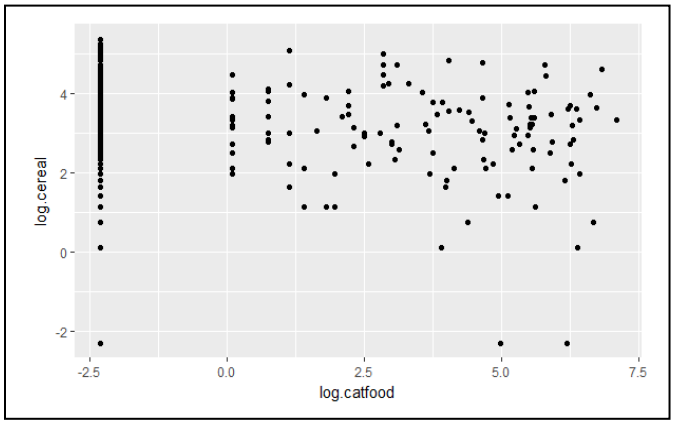

Figure 1.5 – Log-transformation of the attributes

As it can be seen from the figure 1.5, logarithmic transformation of the attributes improves the interpretability of the hidden patterns in the data. It also does not affect the accuracy of end results and hence is a reliable transformation method.

2.3. Scaling the attributes:

It is required to standardized the attributes before performing clustering to get accurate results. If a scale of one attribute is more than the scale of the other attribute, the results of the clustering will be biased towards the attribute with high scale [5,6]. Hence, some standardizing technique should be applied to the attributes. In our analysis, we will use normalizing technique to scale the attributes.

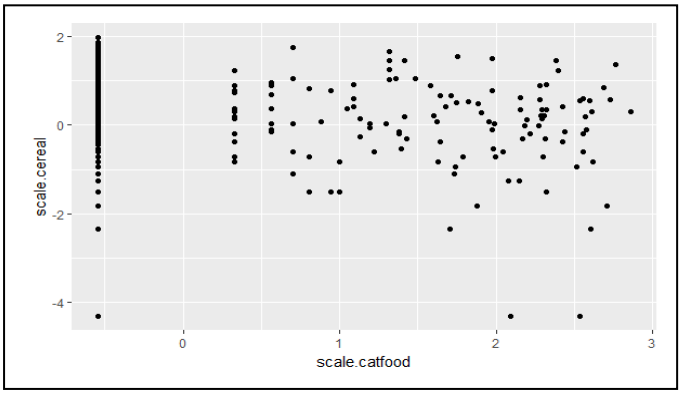

Figure 1.6- Scaling the log-transformed attributes

2.4. k-means clustering:

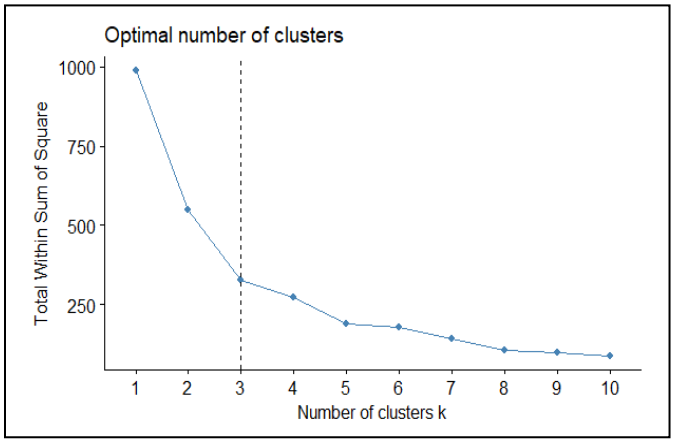

Figure 1.7-Deciding optimum number of clusters using elbow method

In this step, we will use k-means clustering algorithm to perform customer segmentation. First of all, we will decide the optimum number of clusters that minimizes the total within cluster sum of squares using elbow method. As it can be seen from the figure 1.7, the optimum number of clusters for our dataset is 3. Now, we will cluster the customers into 3 clusters based on the chosen attributes.

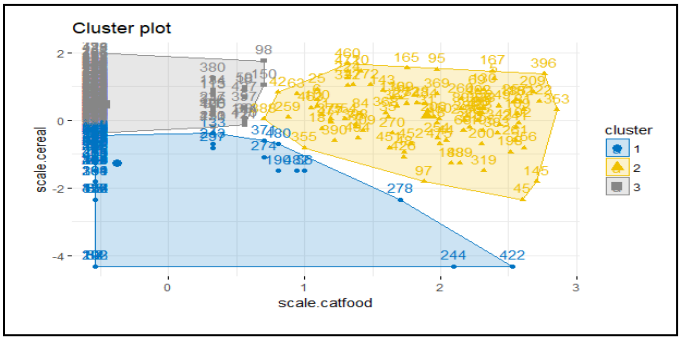

Figure 1.8- Customer segmentation using k-means clustering

3)Demographic based recommendation:

After obtaining the clusters as seen in figure 1.8, we will now evaluate the demographic characteristics of the customers of each cluster. These demographics can later be taken as a reference to generate recommendations for new users with similar demographic attributes.

Table 1.1–Comparison of the demographic attributes of the customers of each clusters

Demographic Attributes	Cluster-1(Moderate quantity cereal-Moderate quantity cat food)	Cluster-2(High quantity cat food-High quantity cereal)	Cluster-3(High quantity cereal-Low quantity cat food)
Median Income	25000-34999	35000-44999	25000-34999
Median Male Age	65-99	55-64	55-64
Median Female Age	55-64	45-54	55-64
Median Male Occupation	Operative	Clerical	Operative
Median Female Occupation	Retired	Retired	Retired
Median Male Education	Completed College	Completed College	Some College
Median Female Education	Completed High School	Some College	Some College
Cable TV subscription	55.65%	62.36%	55.94%
Newspaper Subscription	46.08%	58.06%	44.05%
Median/Max no. of cats	0,1	1,4	0,2

Table 1.1 shows the comparison of the demographic attributes of the customer of each cluster. The 3 clusters are labelled as: **Cluster-1**: Customers who are buying moderate amount of cereal and moderate amount of cat food. **Cluster-2:** Customers who are buying high amount of cereal and high amount of cat food. **Cluster-3:** Customers who are buying less amount of cat food and high amount of cereal. Here, median values of attributes are considered as the median value is not affected by the outliers. It can be inferred from the table that the customers who buy high quantity of cat food and high quantity of cereal have comparatively higher income than the other customer groups. In addition to income, these customers also have overall better education and well-to-do occupation. Their percentage of cable tv and newspaper subscription strengthens the above argument. Finally, their behavior of buying more cat food can be justified by their median and maximum value of cats owned. Talking about more general application of the data, it can be used as a reference for generating recommendation for the new customer based on his/her demographic attributes.

VII. Problem Approach-2

Recency -Frequency based Segmentation:

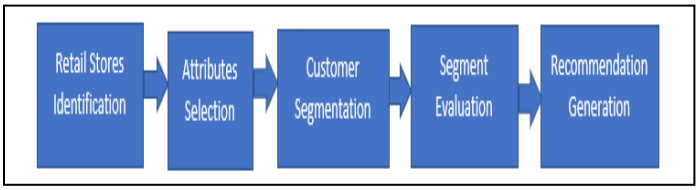

Figure 1.9-Flowchart for recency-frequency based segmentation

The approach to get the required results consists of 5 steps:

1)Retail stores identification:
As it can be seen from the transactional dataset, the company owns 4 different retail stores. We will evaluate the consumers' purchasing behavior separately for each of the retail stores to design the marketing strategy efficiently for each of them.

2)Attributes selection:
This is an important step required for performing customer segmentation. The attributes considered for this analysis are: 1) Frequency of the visit of a consumer 2) Recency of the customer considered for a given period. The frequency of each consumer can be calculated on the basis of number of transactions of that particular consumer. Similarly, recency index of the consumer can be found out by the 'week' attribute given in the data. For each consumer, recency will be calculated by subtracting minimum week number from the week number of that particular customer.

3)Customer segmentation:
Customer segmentation will be performed by applying k-means clustering algorithm to the transactional dataset with respect to the selected attributes. Customers will be segmented into the optimum number of clusters which will be found out by elbow method.

4)Segment Evaluation:
After performing the customer segmentation, cluster containing potential customers will be evaluated in terms of their product preferences and coupon use. Potential customers can be categorized as Frequent – Recent Visitors. In this way, customer segment evaluation performs a significant role in evaluating the customer loyalty towards the company. Accordingly, the company can strategize their marketing campaign to improve the customer loyalty. The way of achieving that is discussed in the next step.

5)Recommendation generation:
After evaluating potential customer cluster for each of the retail stores, marketing strategies can be formed for those particular clusters based on the product preferences and coupons used by the customers of these particular customers. Apriori algorithm can be applied on the transaction dataset to generate association rules and consequently it can be found out which items are frequently bought together.

VIII. ANALYSIS

(1) Identifying retail stores:

We can identify the total number of retail stores that are in the transactional dataset by applying 'unique' function on the dataset. It can be inferred after applying the function that the company owns below 4 retail stores.

Table 1.2- Identification of retail stores owned

Sr. No.	Retail Store ID
1	1521
2	1522
3	1542
4	1558

(2) Evaluating the attributes for each of the retail store:

The next step is to calculate the frequency of visits and the the recency index of each customer in each of the retail stores. This is done with a view to setup a basis for the customer segmentation. These attributes will be considered as a reference for performing customer segmentation and consequently identifying the potential customers for the company.

(3) Visualizing consumers' purchasing pattern:

The next step is to plot the frequency v/s recency graph for all the customers of each retail store. It will give an idea to evaluate the purchasing behavior of the consumers of a particular retail store. As it can be seen from the figure -1.10, the data is skewed. Almost all of the data points are accumulated on the top left corner of the plot. It follows the Pareto principle that 20% of the company's consumers are accountable for the 80% of the company's sale [6]. To interpret the purchasing pattern in a better way, some transformation techniques can be applied to the dataset which we will discuss in the next step.

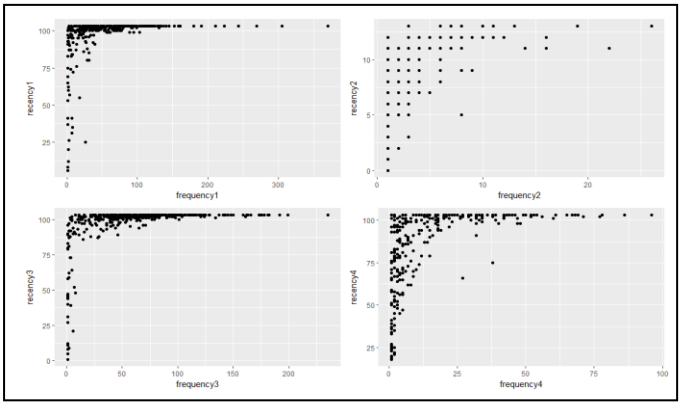

Figure 1.10– Purchasing pattern visualization

(4) Log-transforming the attributes:

Log- transformation is applied to the attributes with a reason to remove the skewness in the data and improve the interpretation of the customers' purchasing pattern. It has to be noted that the log-transformation does not affect the end results in any way and hence it is a reliable method.

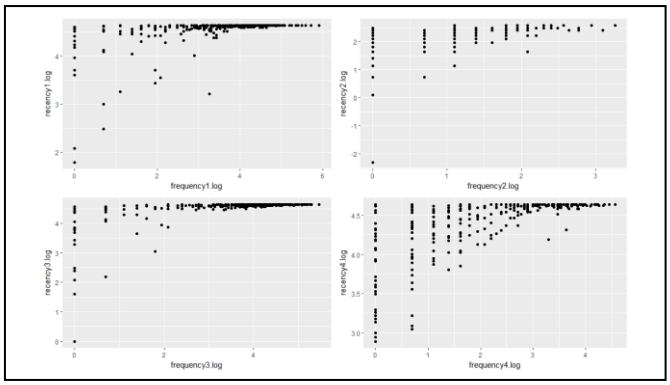

Figure 1.11- Log-transformation of the attributes

(5) Normalizing the log-transformed data:

Normalizing is a type of scaling in which center is taken as the mean of the array and scale is taken as the standard deviation of the array. Scaling is required for the clustering algorithm to work properly. If the values of one attribute are of different order compared to the values of the other attribute, the results of the clustering algorithm will be biased towards the attribute with higher order values [1,5]. After performing the scaling operation on the log-transformed attributes, we will get the attributes in the form of z-score. We will not remove the outliers i.e. the values with z-score out of the range [-3,3] from our consideration. Even though those are the potential outliers, those particular customer groups are still important for the company in terms of their frequency of visits and their recency index. They could turn out to be potential profitable customers.

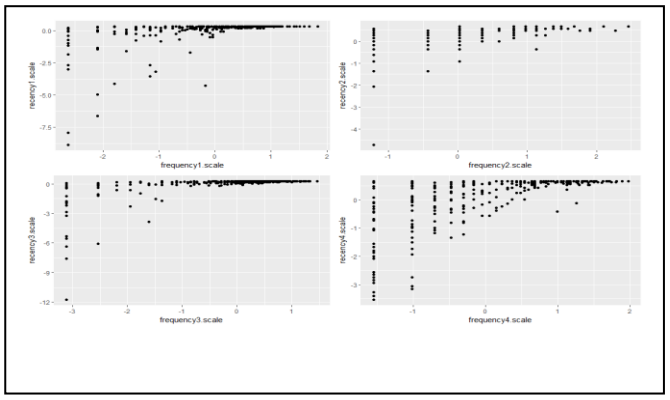

Figure 1.12-Normalizing the log-transformed attributes

IX. METHODOLOGY

1)Customer segmentation using k-means clustering:

In this step of the methodology, we will try to segment customers using k-means clustering algorithm. The attributes that will be used to perform segmentation are: 1) Customer's frequency of visits 2) Recency index of the customer. Accordingly, we will cluster the customers into the segments based on their loyalty.

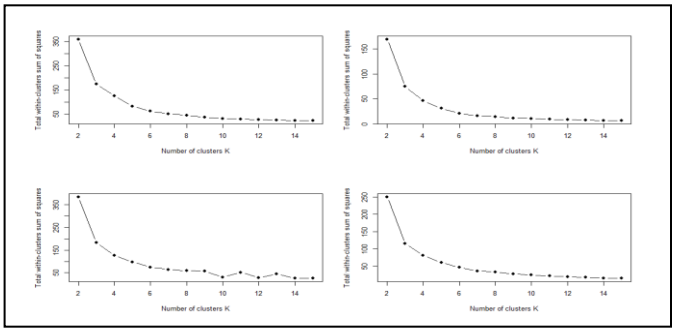

Figure 1.13– Plots formed using elbow method

First of all, we will evaluate what are the optimum number of clusters for each of the retail stores based on the total within cluster sum of squares. We will use elbow method to determine the optimum number of clusters for customers of each of the retail stores. It can be seen from the figure 1.13 that the elbow plot starts converging at around 4 number of clusters for each of the cases. Hence, 4 can be considered as optimum number of clusters for each of the cases. Now, we will apply k-means clustering algorithm on our selected attributes and visualize the results.

As the clusters having already been formed, our next step will be to examine which customers are potential profitable customers to the group. Once identifying these customers, the next step will be to identify what all items are frequently bought together by them. Accordingly, those particular items can be recommended to these customers to improve the profitability of the company.

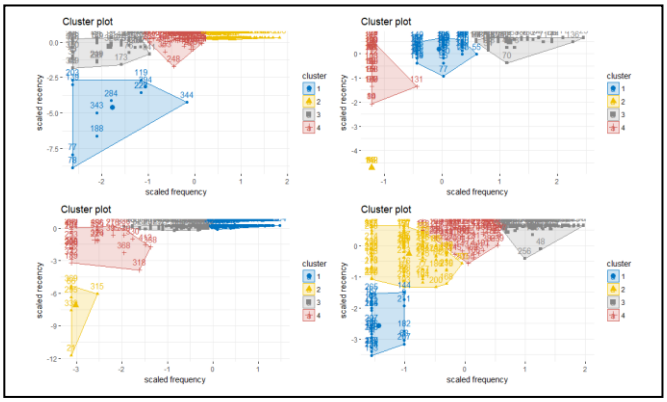

Figure 1.14-Customer segmentation using k-means clustering

2)Finding association rules using Apriori algorithm:

In this section, we will apply apriori algorithm on the selected potential customer segments and try to generate association rules to observe which set of items are frequently bought together by these potential profitable customers. First of all, we will try to examine which set of items are frequently bought by the profitable customers in each of the retail stores.

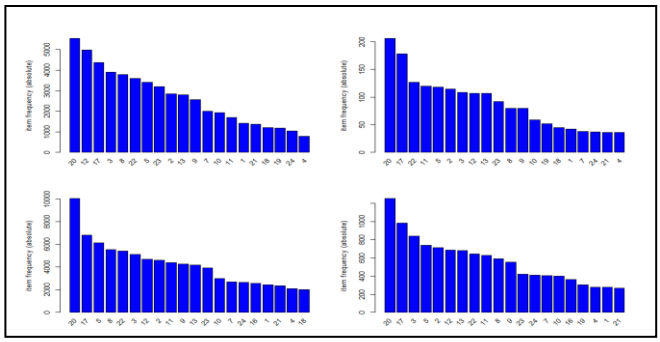

Figure 1.15- Top 20 frequently bought items in each store

Figure 1.15 shows the top 20 frequently bought items in each of the retail stores. It can be inferred that the item type 17 & 20 are the items which are frequently bought by the customers in each of the retail stores. Hence, the company should try to increase the profitability by focusing on the sales of these particular items.

Now, we will try to apply apriori algorithm on the selected customer segments of each of the retail stores to see which particular set of items are frequently bought together. In our analysis, we will set the threshold for support and confidence at 0.001 and 0.80 respectively. We will then generate top 10 association rules and sort them according to their lift value. Lift value for 2 item sets X and Y can be interpreted as the

ratio of probability of finding Y in all the transactions that has X to probability of finding Y alone in all the transactions [7]. Lift value indicates whether the items in the generated rules are correlated or not. If the value of lift is >1, then the items are positively correlated and hence the probability of them getting bought together is very high. Lift value gives better idea about associativity than the confidence value. This is the justification for selecting lift value as a criterion for evaluating items frequently bought together.

Table 1.3-Top 2 association rules for each retail store

Store ID	Association Rules	Support	Confidence	Lift
1521	{12,4,6}=>{23}	0.0010397	0.84	5.316
	{17,4,7}=>{23}	0.0011882	0.80	5.063
1522	{13,14,21}=>{15}	0.001424	1	54
	{14,21,23}=>{15}	0.001424	1	54
1542	{16,20,22,23,4,9}=>{10}	0.001520	0.8478	7.262
	{16,20,23,4,9}=>{10}	0.001598	0.8367	7.167
1558	{10,14,17,20}=>{18}	0.001145	1	27.28
	{10,12,17,20,22,8}=>{6}	0.001145	0.83333	26.74

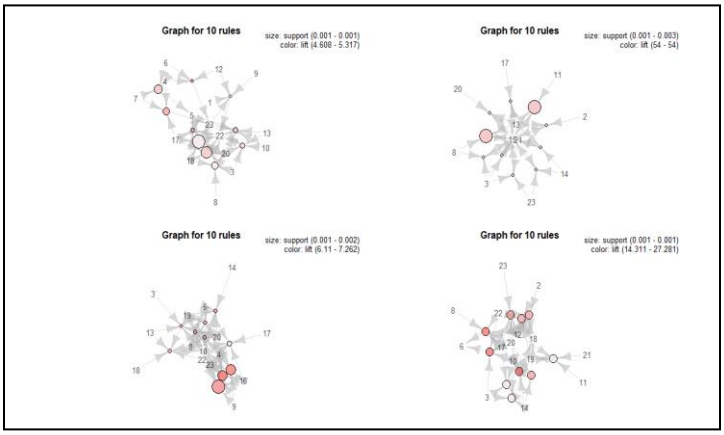

Figure 1.16-Top 10 association rules based on the lift value

Table 1.3 shows the top 2 association rules for each of the retail stores. The minimum support and confidence value considered here are 0.001 and 0.80 respectively. The rules were ranked in terms of the lift value. The lift value is greater than 1 for all the considered rules which shows that the probability of those items getting bought together is very high. Hence, these rules can be taken as a reference by the company to generate the product recommendation.

Now, we will examine what is the popular source of coupons among the profitable customers in each retail store.

It can be seen from the figure 1.17 that the most popular source of coupon among the profitable customers are 23 and 51 which are coupons offered by Sunday supplement vendors and coupons offered in newspaper advertisement by the company. Hence, company should focus on offering coupons through these sources frequently along with recommending preferable products to the customer.

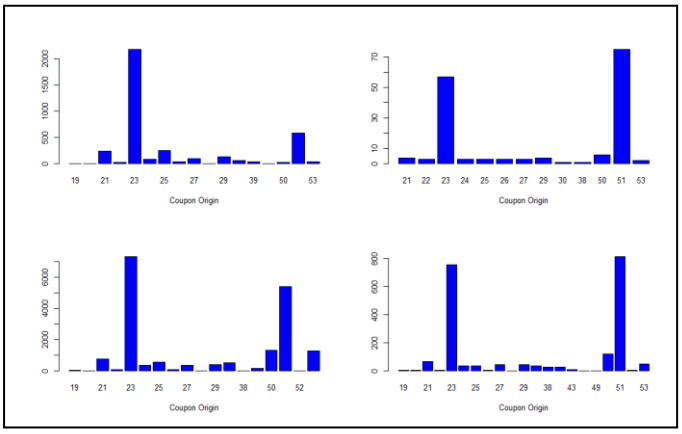

Figure 1.17- Evaluation of coupon sources for each retail store

X. Discussion & Conclusion

In our analysis, we considered 2 approaches: 1) Generating recommendation for a new customer 2) Generating recommendation for a current customer. In the first approach, customers were segmented based on the quantities of 2 item sets bought by them. The items considered were cereal and cat food. The idea was to evaluate the demographic attributes of the clustered customers and use them as a reference to generate recommendation for a new customer with similar attributes. It was inferred from the results that the customers who bought high amount of both cereal and cat food had comparatively high income compared to the other customer groups. Similarly, they also had overall better education level and well-to-do occupation. The better percentage of cable tv and newspaper subscriptions showed their affording power. Finally, their behavior of buying more amount of cat food was justified by the median and max values of cats owned by them. This were the specific inferences made from the obtained result. Talking about general application, this result can be used as a reference for generating recommendation for a new customer with similar demographic attributes. In the second approach, customers were segmented based on their frequency and recency attributes. In this approach, analysis was performed separately for each retail store owned by the company. After performing the segmentation using clustering algorithm, the clusters were analyzed in terms of the product preferences and coupon usage of the customers. It was found out that the items snacks and cold drinks are frequently bought in each retail store. Hence, it is advisable for the company to focus on these products to improve sales. The idea of frequently bought items is different from frequently bought together items. Apriori algorithm was used to generate

association rules which denoted frequently bought together items in each retail store.

Table 1.4- Frequent item sets for each retail store

Retail Store ID	Frequent Item set
1521	Cat food, cleansers, eggs, towels
1522	Ice creams, nuts, sugars, pills
1542	Cat food, crackers, detergents, pizza, soft drinks, tissue, towel
1558	Detergents, nuts, snack, soap, soft drinks

Table 1.4 shows frequent item sets for each of the retail stores which can be used as a reference to generate product recommendations. The popular source of coupons among the consumer is coupons sold by Sunday supplement vendors and coupons offered in newspaper by the company. Hence, company should focus on these sources of advertisement to increase their sales.

Problems Faced:

As the data on the unit cost for each of the items were not available, customer's monetary index could not be calculated. Monetary index could have been considered as one of the attributes for performing customer segmentation in the 2nd approach and could have provided more granularity. Logarithmic transformation had to be applied before performing clustering to remove the skewness in the data distribution. Even though there were some potential outliers identified after scaling the attributes, they were not removed s those outliers can prove to be potential profitable customers for the company.

XI. Future Work

The information on total amount of money spent by each customer can be found out from their transactions. This information then can be used to calculate monetary index of each customer. As the recency and frequency attributes of each customer are already available, RFM analysis technique can be used to perform customer segmentation. The other approach that can be considered is to perform customer segmentation based on the demographics of the customers. For example, the customers can be segmented based on their annual income and their occupation. After that, apriori algorithm can be applied on the obtained clusters to generate association rules which give an idea about the items frequently bought together.

References

[1] Pascal, C., Ozuomba, S., &Kalu, C. (2015). Application of k-means algorithm for efficient customer segmentation: A strategy for targeted customer services. International Journal of Advanced Research in Artificial Intelligence, 4(10).

[2] Malik, Latesh G. "Generating Customer Profiles for Retail Stores Using Clustering Techniques." *International Journal on Computer Science and Engineering (IJCSE)*: 2506-2510.

[3] Cheng, Ching-Hsue, and You-Shyang Chen. "Classifying the segmentation of customer value via RFM model and RS theory." *Expert systems with applications* 36.3 (2009): 4176-4184.

[4] Prasad, Pramod. "Using association rule mining for extracting product sales patterns in retail store transactions."(2011).

[5] An Introduction to Statistical Learning with Applications. R. G. James, D. Witten, T. Hastie, and R. Tibshirani (2013). Berlin: Springer. 440 pages, ISBN: 978-1-4614-7138-7.

[6] k-means Clustering for Customer Segmentation: A Practical Example(http://www.kimberlycoffey.com/blog/2016/8/k-means-clustering-for-customer-segmentation).

[7] Association Analysis Simplified. (n.d.). Retrieved December 08, 2017, from http://analyticstrainings.com/?p=151

APPENDIX

```
[1]  #Approach-1#
[2]  retail=read.csv("TransactionData.csv",header=TRUE)
[3]  library(reshape2)
[4]
[5]  #Generating item-based dataframe
[6]  retail.items=dcast(retail,CustomerID~ItemType,value.var = "UnitsSold",fun.aggregate = sum)
[7]  colnames(retail.items)=c("CustomerID","bacon","bbq","butter","catfood","cereal","cleansers","coffee","cook","crackers","de
     tergents","hotdogs","eggs","icecream","nuts","pill","pizza","snack","soap","soft","softdrinks","sugar","tissue","towel","yogu
     rt")
[8]  retailfiltered.items=data.frame(retail.items$catfood,retail.items$cereal)
[9]  colnames(retailfiltered.items)=c("catfood","cereal")
[10]
[11] #Visualizing customer purchasing behavior
[12] library(ggplot2)
[13] scatter.1=ggplot(retailfiltered.items,aes(x=catfood,y=cereal))+geom_point()
[14] scatter.1
[15]
[16] #log-transforming the attributes
[17] log.catfood=log(retail.items$catfood+0.1)
[18] log.cereal=log(retail.items$cereal+0.1)
[19] retail.log=data.frame(log.catfood,log.cereal)
[20] plot(log.cereal,log.catfood)
[21] scatter.2=ggplot(retail.log,aes(x=log.catfood,y=log.cereal))+geom_point()
[22] scatter.2
[23]
[24] #scaling the log transformed attributes
[25] scale.catfood=scale(log.catfood,center=TRUE,scale=TRUE)
[26] scale.cereal=scale(log.cereal,center=TRUE,scale=TRUE)
[27] plot(scale.catfood,scale.cereal)
[28] retail.scale=data.frame(scale.catfood,scale.cereal)
[29] scatter.3=ggplot(retail.scale,aes(x=scale.catfood,y=scale.cereal))+geom_point()
[30] scatter.3
[31]
[32] #Generating customer segments using k-means clustering
[33] set.seed(123)
[34] library(NbClust)
[35] library("factoextra")
[36]
[37] #Finding the optimum number of clusters
[38] fviz_nbclust(retail.scale,kmeans, method = "wss")+ geom_vline(xintercept = 3, linetype = 2)
[39]
[40] #Visualizing segmentation
[41] retail.cluster=kmeans(retail.scale,3,nstart=20)
[42] fviz_cluster(retail.cluster,data=retail.scale,ellipse.type="convex",palette="jco",ggtheme=theme_minimal())
[43]
[44] #Demographics based evaluation
[45] demo=read.csv("DemographicData.csv",header=TRUE)
[46] d1.c=demo[which(retail.cluster$cluster==2),]
[47] d2.c=demo[which(retail.cluster$cluster==3),]
[48] d3.c=demo[which(retail.cluster$cluster==1),]
[49]
[50] #Income
[51] d1.income=data.frame(segment="cluster-2",Income=d1.c$Income)
[52] d2.income=data.frame(segment="cluster-3",Income=d2.c$Income)
[53] d3.income=data.frame(segment="cluster-1",Income=d3.c$Income)
[54] plot.data=rbind(d1.income,d2.income,d3.income)
```

```
ggplot(plot.data, aes(x=segment, y=Income, fill=segment)) +geom_boxplot()

#MaleAge
d1.maleage=data.frame(segment="cluster-2",mage=d1.c[d1.c$Male.Age.!=7,]$Male.Age.)
d2.maleage=data.frame(segment="cluster-3",mage=d2.c[d2.c$Male.Age.!=7,]$Male.Age.)
d3.maleage=data.frame(segment="cluster-1",mage=d3.c[d3.c$Male.Age.!=7,]$Male.Age.)
plot.data=rbind(d1.maleage,d2.maleage,d3.maleage)
ggplot(plot.data, aes(x=segment, y=mage, fill=segment)) +geom_boxplot()

#FemaleAge
d1.femaleage=data.frame(segment="cluster-2",fmage=d1.c[d1.c$Female.Age.!=7,]$Female.Age.)
d2.femaleage=data.frame(segment="cluster-3",fmage=d2.c[d2.c$Female.Age.!=7,]$Female.Age.)
d3.femaleage=data.frame(segment="cluster-1",fmage=d3.c[d3.c$Female.Age.!=7,]$Female.Age.)
plot.data=rbind(d1.femaleage,d2.femaleage,d3.femaleage)
ggplot(plot.data, aes(x=segment, y=fmage, fill=segment)) +geom_boxplot()

#MaleOccupation
d1.maleocc=data.frame(segment="cluster-2",mocc=d1.c[d1.c$Male.Occupation!=11,]$Male.Occupation)
d2.maleocc=data.frame(segment="cluster-3",mocc=d2.c[d2.c$Male.Occupation!=11,]$Male.Occupation)
d3.maleocc=data.frame(segment="cluster-1",mocc=d3.c[d3.c$Male.Occupation!=11,]$Male.Occupation)
plot.data=rbind(d1.maleocc,d2.maleocc,d3.maleocc)
ggplot(plot.data, aes(x=segment, y=mocc, fill=segment)) +geom_boxplot()

#FemaleOccupation
d1.femaleocc=data.frame(segment="cluster-2",fmocc=d1.c$Female.Occupation)
d2.femaleocc=data.frame(segment="cluster-3",fmocc=d2.c$Female.Occupation)
d3.femaleocc=data.frame(segment="cluster-1",fmocc=d3.c$Female.Occupation)
plot.data=rbind(d1.femaleocc,d2.femaleocc,d3.femaleocc)
ggplot(plot.data, aes(x=segment, y=fmocc, fill=segment)) +geom_boxplot()

#Maleeducation
d1.maleed=data.frame(segment="cluster-2",med=d1.c$Male.Education)
d2.maleed=data.frame(segment="cluster-3",med=d2.c$Male.Education)
d3.maleed=data.frame(segment="cluster-1",med=d3.c$Male.Education)
plot.data=rbind(d1.maleed,d2.maleed,d3.maleed)
ggplot(plot.data, aes(x=segment, y=med, fill=segment)) +geom_boxplot()

#Femaleeducation
d1.femaleed=data.frame(segment="cluster-2",fmed=d1.c$Female.Education)
d2.femaleed=data.frame(segment="cluster-3",fmed=d2.c$Female.Education)
d3.femaleed=data.frame(segment="cluster-1",fmed=d3.c$Female.Education)
plot.data=rbind(d1.femaleed,d2.femaleed,d3.femaleed)
ggplot(plot.data, aes(x=segment, y=fmed, fill=segment)) +geom_boxplot()

#No. of cats
d1.cats=data.frame(segment="cluster-2",cats=d1.c[d1.c$Cats!=9,]$Cats)
d2.cats=data.frame(segment="cluster-3",cats=d2.c[d2.c$Cats!=9,]$Cats)
d3.cats=data.frame(segment="cluster-1",cats=d3.c[d3.c$Cats!=9,]$Cats)
plot.data=rbind(d1.cats,d2.cats,d3.cats)
ggplot(plot.data, aes(x=segment, y=cats, fill=segment)) +geom_boxplot()

#cabel tv subscription
count1.ct=table(d1.c$Cable.TV)
count2.ct=table(d2.c$Cable.TV)
count3.ct=table(d3.c$Cable.TV)
count1.ct
count2.ct
count3.ct
```

```
par(mfrow=c(2,1))
barplot(count1.ct,xlab="cabel tv subscription",col="blue")
barplot(count2.ct,xlab="cabel tv subscription",col="blue")
barplot(count3.ct,xlab="cabel tv subscription",col="blue")

#newspapersubscription
count1.np=table(d1.c$Newspaper.subscriber)
count2.np=table(d2.c$Newspaper.subscriber)
count3.np=table(d3.c$Newspaper.subscriber)
count1.np
count2.np
count3.np
par(mfrow=c(2,1))
barplot(count1.np,xlab="newspaper subscription",col="blue")
barplot(count2.np,xlab="newspaper subscription",col="blue")
barplot(count3.np,xlab="newspaper subscription",col="blue")

#Approach-2#
#subsetting transaction data according to 4 retail stores
retail=read.csv("TransactionData.csv",header=TRUE)
unique(retail$StoreID)
retail1=subset(retail,StoreID==1521)
retail2=subset(retail,StoreID==1522)
retail3=subset(retail,StoreID==1542)
retail4=subset(retail,StoreID==1558)

#Evaluating store No. 1521
retail1$purchase=grepl("\\d+", retail1$TransactionID, fixed=TRUE)
retail1$purchase.invoice=ifelse(retail1$purchase=="TRUE", 0, 1)
retail1$recency=retail1$Week-614
customer1.invoices=subset(retail1,select=c("TransactionID","CustomerID","purchase.invoice"))
customer1.invoices=customer1.invoices[!duplicated(customer1.invoices),]
total1.invoices=aggregate(purchase.invoice ~ CustomerID, data=customer1.invoices, FUN=sum, na.rm=TRUE)
recency1=aggregate(recency~CustomerID,data=retail1,FUN=max)
recency1=recency1$recency

#Evaluating store No. 1522
retail2$purchase=grepl("\\d+", retail2$TransactionID, fixed=TRUE)
retail2$purchase.invoice=ifelse(retail2$purchase=="TRUE", 0, 1)
retail2$recency=retail2$Week-614
customer2.invoices=subset(retail2,select=c("TransactionID","CustomerID","purchase.invoice"))
customer2.invoices=customer2.invoices[!duplicated(customer2.invoices),]
total2.invoices=aggregate(purchase.invoice ~ CustomerID, data=customer2.invoices, FUN=sum, na.rm=TRUE)
recency2=aggregate(recency~CustomerID,data=retail2,FUN=max)
recency2=recency2$recency

#Evaluating store No. 1542
retail3$purchase=grepl("\\d+", retail3$TransactionID, fixed=TRUE)
retail3$purchase.invoice=ifelse(retail3$purchase=="TRUE", 0, 1)
retail3$recency=retail3$Week-614
customer3.invoices=subset(retail3,select=c("TransactionID","CustomerID","purchase.invoice"))
customer3.invoices=customer3.invoices[!duplicated(customer3.invoices),]
total3.invoices=aggregate(purchase.invoice ~ CustomerID, data=customer3.invoices, FUN=sum, na.rm=TRUE)
recency3=aggregate(recency~CustomerID,data=retail3,FUN=max)
recency3=recency3$recency

#Evaluating store No. 1558
retail4$purchase=grepl("\\d+", retail4$TransactionID, fixed=TRUE)
```

```
retail4$purchase.invoice=ifelse(retail4$purchase=="TRUE", 0, 1)
retail4$recency=retail4$Week-614
customer4.invoices=subset(retail4,select=c("TransactionID","CustomerID","purchase.invoice"))
customer4.invoices=customer4.invoices[!duplicated(customer4.invoices),]
total4.invoices=aggregate(purchase.invoice ~ CustomerID, data=customer4.invoices, FUN=sum, na.rm=TRUE)
recency4=aggregate(recency~CustomerID,data=retail4,FUN=max)
recency4=recency4$recency


#Plotting data for each case
frequency1=total1.invoices$purchase.invoice
df1=as.data.frame(cbind(frequency1,recency1))
frequency2=total2.invoices$purchase.invoice
df2=as.data.frame(cbind(frequency2,recency2))
frequency3=total3.invoices$purchase.invoice
df3=as.data.frame(cbind(frequency3,recency3))
frequency4=total4.invoices$purchase.invoice
df4=as.data.frame(cbind(frequency4,recency4))
library(ggplot2)
library(grid)
library(gridExtra)
scatter.1=ggplot(df1,aes(x=frequency1,y=recency1))+geom_point()
scatter.2=ggplot(df2,aes(x=frequency2,y=recency2))+geom_point()
scatter.3=ggplot(df3,aes(x=frequency3,y=recency3))+geom_point()
scatter.4=ggplot(df4,aes(x=frequency4,y=recency4))+geom_point()
grid.arrange(scatter.1,scatter.2,scatter.3,scatter.4,ncol=2)


#log-transforming data
frequency1.log=log(frequency1)
recency1.log=log(recency1)
df1.log=as.data.frame(cbind(frequency1.log,recency1.log))
frequency2.log=log(frequency2)
recency2.log=log(recency2+0.1)
df2.log=as.data.frame(cbind(frequency2.log,recency2.log))
frequency3.log=log(frequency3)
recency3.log=log(recency3)
df3.log=as.data.frame(cbind(frequency3.log,recency3.log))
frequency4.log=log(frequency4)
recency4.log=log(recency4)
df4.log=as.data.frame(cbind(frequency4.log,recency4.log))
scatterlog.1=ggplot(df1.log,aes(x=frequency1.log,y=recency1.log))+geom_point()
scatterlog.2=ggplot(df2.log,aes(x=frequency2.log,y=recency2.log))+geom_point()
scatterlog.3=ggplot(df3.log,aes(x=frequency3.log,y=recency3.log))+geom_point()
scatterlog.4=ggplot(df4.log,aes(x=frequency4.log,y=recency4.log))+geom_point()
grid.arrange(scatterlog.1,scatterlog.2,scatterlog.3,scatterlog.4,ncol=2)


#scaling the log-transformed data
frequency1.scale=scale(frequency1.log,center=TRUE,scale=TRUE)
recency1.scale=scale(recency1.log,center=TRUE,scale=TRUE)
df1.scale=as.data.frame(cbind(frequency1.scale,recency1.scale))
colnames(df1.scale)=c("scaled frequency","scaled recency")
frequency2.scale=scale(frequency2.log,center=TRUE,scale=TRUE)
recency2.scale=scale(recency2.log,center=TRUE,scale=TRUE)
df2.scale=as.data.frame(cbind(frequency2.scale,recency2.scale))
colnames(df2.scale)=c("scaled frequency","scaled recency")
frequency3.scale=scale(frequency3.log,center=TRUE,scale=TRUE)
recency3.scale=scale(recency3.log,center=TRUE,scale=TRUE)
```

```
df3.scale=as.data.frame(cbind(frequency3.scale,recency3.scale))
colnames(df3.scale)=c("scaled frequency","scaled recency")
frequency4.scale=scale(frequency4.log,center=TRUE,scale=TRUE)
recency4.scale=scale(recency4.log,center=TRUE,scale=TRUE)
df4.scale=as.data.frame(cbind(frequency4.scale,recency4.scale))
colnames(df4.scale)=c("scaled frequency","scaled recency")
scatterscale.1=ggplot(df1.scale,aes(x=frequency1.scale,y=recency1.scale))+geom_point()
scatterscale.2=ggplot(df2.scale,aes(x=frequency2.scale,y=recency2.scale))+geom_point()
scatterscale.3=ggplot(df3.scale,aes(x=frequency3.scale,y=recency3.scale))+geom_point()
scatterscale.4=ggplot(df4.scale,aes(x=frequency4.scale,y=recency4.scale))+geom_point()
grid.arrange(scatterscale.1,scatterscale.2,scatterscale.3,scatterscale.4,ncol=2)

#customer segmentation for each of the retail stores
library(dplyr)
set.seed(123)
kmax=15
twss1=sapply(2:kmax,function(k){kmeans(df1.scale,k,nstart=20, iter.max=1000000)$tot.withinss})
twss1
km.res1=kmeans(df1.scale,4,nstart=25)
library("factoextra")
fviz_cluster(km.res1,data=df1.scale,ellipse.type="convex",palette="jco",ggtheme=theme_minimal())


set.seed(123)
kmax=15
twss2=sapply(2:kmax,function(k){kmeans(df2.scale,k,nstart=20, iter.max=1000000)$tot.withinss})
twss2
km.res2=kmeans(df2.scale,4,nstart=25)
fviz_cluster(km.res2,data=df2.scale,ellipse.type="convex",palette="jco",ggtheme=theme_minimal())

set.seed(123)
kmax=15
twss3=sapply(2:kmax,function(k){kmeans(df3.scale,k,nstart=20, iter.max=1000000)$tot.withinss})
twss3
km.res3=kmeans(df3.scale,4,nstart=25)
fviz_cluster(km.res3,data=df3.scale,ellipse.type="convex",palette="jco",ggtheme=theme_minimal())


set.seed(123)
kmax=15
twss4=sapply(2:kmax,function(k){kmeans(df4.scale,k,nstart=20, iter.max=1000000)$tot.withinss})
twss4
km.res4=kmeans(df4.scale,4,nstart=25)
fviz_cluster(km.res4,data=df4.scale,ellipse.type="convex",palette="jco",ggtheme=theme_minimal())

par(mfrow=c(2,2))
plot(2:kmax,twss1,type="b", pch = 19, xlab="Number of clusters K", ylab="Total within-clusters sum of squares")
plot(2:kmax,twss2,type="b", pch = 19, xlab="Number of clusters K", ylab="Total within-clusters sum of squares")
plot(2:kmax,twss3,type="b", pch = 19, xlab="Number of clusters K", ylab="Total within-clusters sum of squares")
plot(2:kmax,twss4,type="b", pch = 19, xlab="Number of clusters K", ylab="Total within-clusters sum of squares")

grid.arrange(
  fviz_cluster(km.res1,data=df1.scale,ellipse.type="convex",palette="jco",ggtheme=theme_minimal()),
  fviz_cluster(km.res2,data=df2.scale,ellipse.type="convex",palette="jco",ggtheme=theme_minimal()),
  fviz_cluster(km.res3,data=df3.scale,ellipse.type="convex",palette="jco",ggtheme=theme_minimal()),
  fviz_cluster(km.res4,data=df4.scale,ellipse.type="convex",palette="jco",ggtheme=theme_minimal()),ncol=2)

#identifying cluster members
```

```
which(km.res1$cluster==2)
which(km.res2$cluster==3)
which(km.res3$cluster==1)
which(km.res4$cluster==3)

#Generating association rules
library(plyr)
itemList1=ddply(retail1,c("TransactionID"), function(t1)paste(t1$ItemType, collapse = ","))
itemList1$TransactionID=NULL
colnames(itemList1)=NULL
write.csv(itemList1,"marketbasket1.csv", quote = FALSE, row.names =FALSE)

library(tidyverse)
library(readxl)
library(knitr)
library(lubridate)
library(arules)
library(arulesViz)

tr1=read.transactions('marketbasket1.csv',format='basket',sep=',')
tr1
summary(tr1)
itemFrequencyPlot(tr1, topN=20, type='absolute')
rules1=apriori(tr1, parameter = list(supp=0.001, conf=0.80))
rules1=sort(rules1, by='lift', decreasing = TRUE)
summary(rules1)
inspect(rules1[1:10])
toprules1=rules1[1:10]
plot(toprules1,method="graph")


itemList2=ddply(retail2,c("TransactionID"), function(t1)paste(t1$ItemType, collapse = ","))
itemList2$TransactionID=NULL
colnames(itemList2)=NULL
write.csv(itemList2,"marketbasket2.csv", quote = FALSE, row.names =FALSE)

tr2=read.transactions('marketbasket2.csv',format='basket',sep=',')
tr2
summary(tr2)
itemFrequencyPlot(tr2, topN=20, type='absolute')
rules2=apriori(tr2, parameter = list(supp=0.001, conf=0.80))
rules2=sort(rules2, by='lift', decreasing = TRUE)
summary(rules2)
inspect(rules2[1:10])
toprules2=rules2[1:10]
plot(toprules2,method="graph")


itemList3=ddply(retail3,c("TransactionID"), function(t1)paste(t1$ItemType, collapse = ","))
itemList3$TransactionID=NULL
colnames(itemList3)=NULL
write.csv(itemList3,"marketbasket3.csv", quote = FALSE, row.names =FALSE)

tr3=read.transactions('marketbasket3.csv',format='basket',sep=',')
tr3
summary(tr3)
itemFrequencyPlot(tr3, topN=20, type='absolute')
rules3=apriori(tr3, parameter = list(supp=0.001, conf=0.80))
```

```
rules3=sort(rules3, by='lift', decreasing = TRUE)
summary(rules3)
inspect(rules3[1:10])
toprules3=rules3[1:10]
plot(toprules3,method="graph")


itemList4=ddply(retail4,c("TransactionID"), function(t1)paste(t1$ItemType, collapse = ","))
itemList4$TransactionID=NULL
colnames(itemList4)=NULL
write.csv(itemList4,"marketbasket4.csv", quote = FALSE, row.names =FALSE)

tr4=read.transactions('marketbasket4.csv',format='basket',sep=',')
tr4
summary(tr4)
itemFrequencyPlot(tr4, topN=20, type='absolute')
rules4=apriori(tr4, parameter = list(supp=0.001, conf=0.80))
rules4=sort(rules4, by='lift', decreasing = TRUE)
summary(rules4)
inspect(rules4[1:10])
toprules4=rules4[1:10]
plot(toprules4,method="graph")

par(mfrow=c(2,2))
itemFrequencyPlot(tr1, topN=20, type='absolute',col="blue")
itemFrequencyPlot(tr2, topN=20, type='absolute',col="blue")
itemFrequencyPlot(tr3, topN=20, type='absolute',col="blue")
itemFrequencyPlot(tr4, topN=20, type='absolute',col="blue")

par(mfrow=c(2,2))
plot(toprules1,method="graph")
plot(toprules2,method="graph")
plot(toprules3,method="graph")
plot(toprules4,method="graph")

#Evaluating coupon origins
coupon1=retail1$CouponOrigin
coupon1=coupon1[coupon1!=0]
count1=table(coupon1)
count1
coupon2=retail2$CouponOrigin
coupon2=coupon2[coupon2!=0]
count2=table(coupon2)
count2
coupon3=retail3$CouponOrigin
coupon3=coupon3[coupon3!=0]
count3=table(coupon3)
count3
coupon4=retail4$CouponOrigin
coupon4=coupon4[coupon4!=0]
count4=table(coupon4)
count4
par(mfrow=c(2,2))
barplot(count1,xlab="Coupon Origin",col="blue")
barplot(count2,xlab="Coupon Origin",col="blue")
barplot(count3,xlab="Coupon Origin",col="blue")
barplot(count4,xlab="Coupon Origin",col="blue")
```