Customer Segmentation:

Implementation using RFM Analysis and K-means Clustering Algorithm

Abhishek Gupta, Adriana Rosas Cordoba, Jash Shah, Manvir Singh, Noopur Mishra, and Revati Deshpande

Department of Analytics, Harrisburg University of Science & Technology

ANLY 506-51- B-2021/Summer - Exploratory Data Analysis

Dr. Doaa Taha

August 16, 2021

Contents

Introduction

The coronavirus disease (COVID-19) was first found in Wuhan, China in December 2019. The COVID-19 virus is a strand of a large family of coronaviruses known as SARS-CoV-2 which is highly contagious (CDC, 2021). The Coronavirus pandemic has caused more than 190 million people to get sick with the virus and more than 4 million deaths since 2019 (Worldmeters, 2021). Since the beginning of the COVID-19 pandemic, health officials such as CDC(Centers for Disease Control) and WHO (World Health Organization) have shared many guidelines to keep people safe from the virus, for example, travel restrictions, face masks, social distancing, closing the public places, no large gatherings, etc. (CDC, 2021). Fortunately, Vaccinations have started, and more than 3.58 billion vaccines have been administered so far worldwide (Hannah Ritchie, 2020). During covid-19, many countries have suffered many crises from normal day-to-day needs to supply of medical equipment and vaccines. The whole world is still recovering from the severe damage done by the pandemic. The COVID-19 has affected everyone and every business in unimaginable ways.

According to the research done by the USC Center for Risk and Economic Analysis of Terrorism Events(CREATE), “The businesses closures and partial reopening due to COVID-19 pandemic could result in net losses from $3.2 trillion to $4.8 trillion in the USA over two years” (Gersema, 2020). Businesses have also ushered to make many changes in their online marketing strategies as most people are staying at home and buying products online.

Businesses must adopt the revised marketing strategies which can target the right group of customers (Brodbeck, 2020). The marketing strategies for in-stores will not work efficiently in an online environment during a pandemic where we need to make decisions considering safety for everyone. For example, many companies provide more offers and discounts on their app and

website than in-store purchases (Brodbeck, 2020). It is important to learn the efficient ways to understand the customers better and their segmentation. The information from customer segmentation can be highly useful to increase customer loyalty as the customer will be getting customized content. Businesses can find their most valuable customers and can also focus on high-quality customer service to keep them with the business for long.

Background

An effective marketing strategy involves dividing their customers into different groups to understand their needs better. The customer groups can be made based on their purchase history, how much money they spent in the past, how many times they make the purchase as well as their demographic information. This process of dividing the customer into groups is called customer segmentation. Segmentation can offer great insights into customer's purchase patterns, habits, behavior, and preferences which allow businesses to build customized marketing strategies and improving the overall customer experience.

One of the great papers by Sulekha Goyal (Goyal, 2011) compared the process of customer segmentation with the strategy of divide and conquer. Distinguishing the products, services, and customers to design or redesign new services or products is highly useful to meet the market goals. Customer segmentation is a great way of identifying the most and least profitable customers as well as products. It helps to make the best use of marketing resources and budget (Goyal, 2011).

One of the best use cases we found explains how PepsiCo, a multinational beverage company that supplies more than 200 countries uses big data, customer segmentation, and predictive analysis to make informed decisions to identify and target the customers who are likely to be highly interested in a specific brand. They revised their marketing strategies after

finding the target customers that drove 80% of the product's sales growth in one year after its launch (Gavin M. , 2019). Many other case studies support the strength of customer segmentation and data-driven marketing strategies. Our project is inspired with the same intuition to understand the process of customer segmentation. This project will emphasize the process of customer segmentation using clustering techniques.

Objective

There are several techniques available to perform customer segmentation. The RFM (Recency, Frequency, and Monetary) analysis and K-means clustering algorithm are widely used algorithms for customer segmentation to develop efficient market strategies. This project proposes to analyze the sales of an online store in the UK from 01/12/2010 to 9/12/2011 to understand how customer segmentation using RFM analysis and K-means clustering help businesses make informed decisions to increase their revenue and to enhance their customer's overall shopping experience.

Some guiding questions include:

- What are the available algorithms for customer segmentation?
- How will we make sure the businesses provide a great customer experience?
- What are those important factors which help to understand customer's relationship with the businesses?
- How can we segment customers based on their purchase history?
- How does customer segmentation help businesses to come up with more effective marketing strategies?

Data Description

In our quest of finding the dataset, which could be useful to implement the algorithms and methods that we learned in the course. We stumbled upon this dataset which will be greatly suited for implementing the clustering algorithms. We intend to perform customer segmentation using RFM analysis and K-means clustering using this dataset. The dataset is available on the website of the UCI machine learning repository at https://archive.ics.uci.edu/ml/machine-learning-databases/00502/ (Chen, n.d.). The data is collected by Dr. Daqing Chen, School of Engineering, London South Bank University. The sample of the raw data is shown in Figure 1.

The data set contains the sales transactions for two years 2009 and 2010 for the UK-based online retail all-occasion gift store. We will use the data set from 01/12/2010 to 9/12/2011 to implement this project. The data has 541909 observations and 8 variables. The variable's information is as follows:

InvoiceNo is a 6-digit number assigned to each transaction.

StockCode is a product code, uniquely assigned to each product.

Description is the information about the product bought by customers.

Quantity shows how many items are bought per transaction.

InvoiceDate is a date and time for the purchase for each transaction.

UnitPrice is the price for each unit.

CustomerID is a 5-digit unique number assigned to each customer.

Country has information of the origin country for each order.

Figure 1

Sample of the dataset before data preprocessing

InvoiceNo	StockCode	Descriptio	Quantity	InvoiceDate	UnitPrice	Customer	Country
536365	85123A	WHITE HA	6	12/1/2010 8:26	2.55	17850	United Kingdom
536365	71053	WHITE ME	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84406B	CREAM CU	8	12/1/2010 8:26	2.75	17850	United Kingdom
536365	84029G	KNITTED U	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84029E	RED WOO	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	22752	SET 7 BAB	2	12/1/2010 8:26	7.65	17850	United Kingdom
536365	21730	GLASS STA	6	12/1/2010 8:26	4.25	17850	United Kingdom
536366	22633	HAND WA	6	12/1/2010 8:28	1.85	17850	United Kingdom
536366	22632	HAND WA	6	12/1/2010 8:28	1.85	17850	United Kingdom
536367	84879	ASSORTED	32	12/1/2010 8:34	1.69	13047	United Kingdom

Note. This figure shows a sample of the raw dataset for an online store in the United Kingdom from 2010 to 2011 (Chen, n.d.).

Data Preprocessing

We have utilized RStudio to perform data preprocessing, data visualization, and project implementation using the R programming language. The data set contains 541909 observations and 8 variables. The dataset had 135,080 null values only in the “CustomerID” column. We decided to remove the observations with the null values as the dataset is quite large. We will still have 406,829 observations and 8 variables for further analysis.

After checking the summary of the data set, we found that the column “Quantity” has negative values and “UnitPrice” has 0 values in few observations. There was no description in the data dictionary for these values o be 0 or negative. We decided to remove those observations from the data set. The dataset has 397,884 observations and 8 columns after cleaning the numerical columns. We added one column as ’InvoiceTotal’ for the total amount on the invoice as the data set involves the number of units bought and the price for each item.

We found that the column “InvoiceDate” is a character type. We created a new column “InvoiceDateTime” using mdy_hm() function in the “lubridate” package from “InvoiceDate” column. We also extracted the date, time, weekday, month, year, and hour of the day from the “InvoiceDateTime” column to obtain better insights into the data set. We also changed the “Country” column to the factor type from the character type as well.

The cleaned “eCommData” dataset is further filtered on transactions for the United Kingdom to “eCommDataUK” dataset with 354,321 observations and 14 variables for project implementation**.** Figure 2 is a snapshot of the sample of cleaned data. The data preprocessing codebook is attached in Appendix A.

Figure 2

Sample of the dataset after data preprocessing

	InvoiceNo	StockCode	Descriptio	Quantity	UnitPrice	Customer	Country	TotalInvoi	InvoiceDateTime	InvoiceFullDate	InvoiceMonth	InvoiceWeekDay	InvoiceYear	HourOfDay
1	536365	85123A	WHITE HA	6	2.55	17850	United Kin	15.3	12/1/2010 8:26	12/1/2010	December	Wednesday	2010	8
2	536365	71053	WHITE ME	6	3.39	17850	United Kin	20.34	12/1/2010 8:26	12/1/2010	December	Wednesday	2010	8
3	536365	84406B	CREAM CU	8	2.75	17850	United Kin	22	12/1/2010 8:26	12/1/2010	December	Wednesday	2010	8
4	536365	84029G	KNITTED U	6	3.39	17850	United Kin	20.34	12/1/2010 8:26	12/1/2010	December	Wednesday	2010	8
5	536365	84029E	RED WOO	6	3.39	17850	United Kin	20.34	12/1/2010 8:26	12/1/2010	December	Wednesday	2010	8
6	536365	22752	SET 7 BAB	2	7.65	17850	United Kin	15.3	12/1/2010 8:26	12/1/2010	December	Wednesday	2010	8
7	536365	21730	GLASS STA	6	4.25	17850	United Kin	25.5	12/1/2010 8:26	12/1/2010	December	Wednesday	2010	8
8	536366	22633	HAND WA	6	1.85	17850	United Kin	11.1	12/1/2010 8:28	12/1/2010	December	Wednesday	2010	8
9	536366	22632	HAND WA	6	1.85	17850	United Kin	11.1	12/1/2010 8:28	12/1/2010	December	Wednesday	2010	8
10	536367	84879	ASSORTED	32	1.69	13047	United Kin	54.08	12/1/2010 8:34	12/1/2010	December	Wednesday	2010	8
11	536367	22745	POPPY'S P	6	2.1	13047	United Kin	12.6	12/1/2010 8:34	12/1/2010	December	Wednesday	2010	8
12	536367	22748	POPPY'S P	6	2.1	13047	United Kin	12.6	12/1/2010 8:34	12/1/2010	December	Wednesday	2010	8

Note. This figure shows a sample of the cleaned dataset after data preprocessing.

Data Visualization

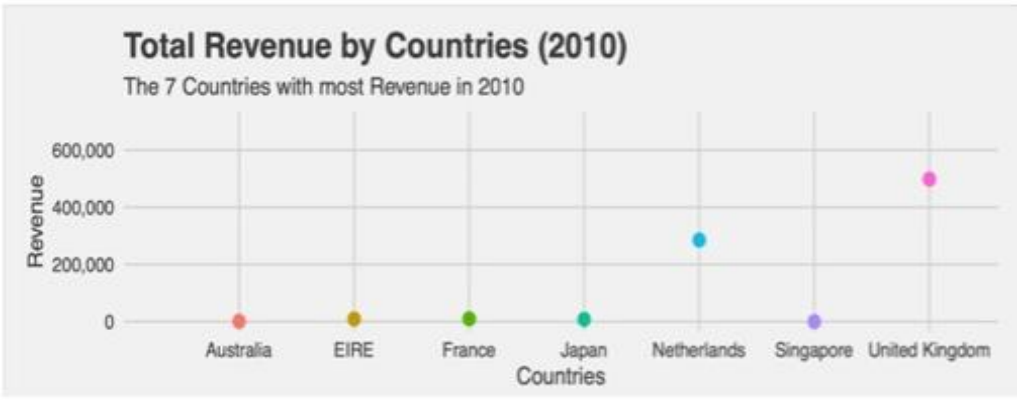
The dataset has transactional data of an online gift store for multiple countries. Visualizing the dataset is helpful to find any pattern as well as to get a better understanding of the data. The visualization task is performed by using ggplot2, Plotly, and R's base plotting packages. Sqldf package is used for querying the dataset. Visualization helped us to derive important information about the dataset.

We examined many important aspects of the dataset. The total revenue for each country in 2010 and 2011 is shown in Figure 3 and Figure 4. The number of daily transactions by each country is shown in Figure 5. The top 10 customers responsible for maximum revenue are found using their customer IDs and the total Invoice price shown in Figure 6. The number of daily transactions for each month is shown in Figure 7. The number of daily transactions for each weekday is shown in Figure 8. The number of daily transactions by the hour of the day is shown in Figure 9.

The data visualization helped us to conclude that the dataset has maximum transactions(354,321) for the United Kingdom with the maximum revenue. We also checked who are the top 10 customer IDs responsible for maximum revenue. We also looked into timely distribution in the dataset to check for any seasonality. October, November, and December months have the maximum number of sales in the dataset. Wednesday and Thursday are the popular days for sales. The data visualization codebook is included in Appendix B.

Figure 3

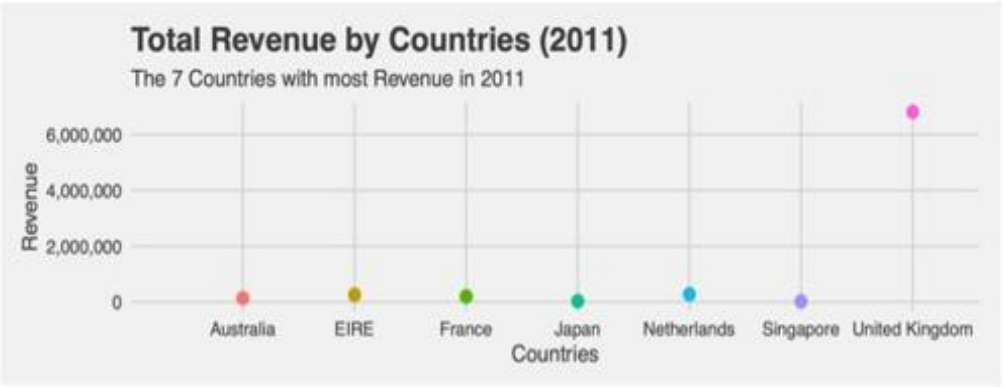
Total revenue for each country in 2010



Note. The graph shows the total revenue in 2010 generated on the y-axis with the corresponding country name on the x-axis.

Figure 4

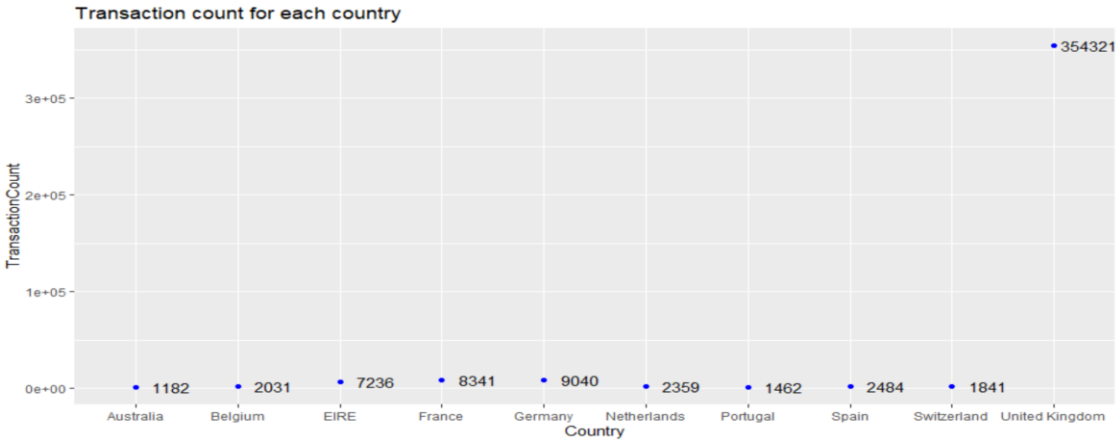
Total revenue for each country in 2011



Note. The graph shows the total revenue in 2011 generated on the y-axis with the corresponding country name on the x-axis.

Figure 5

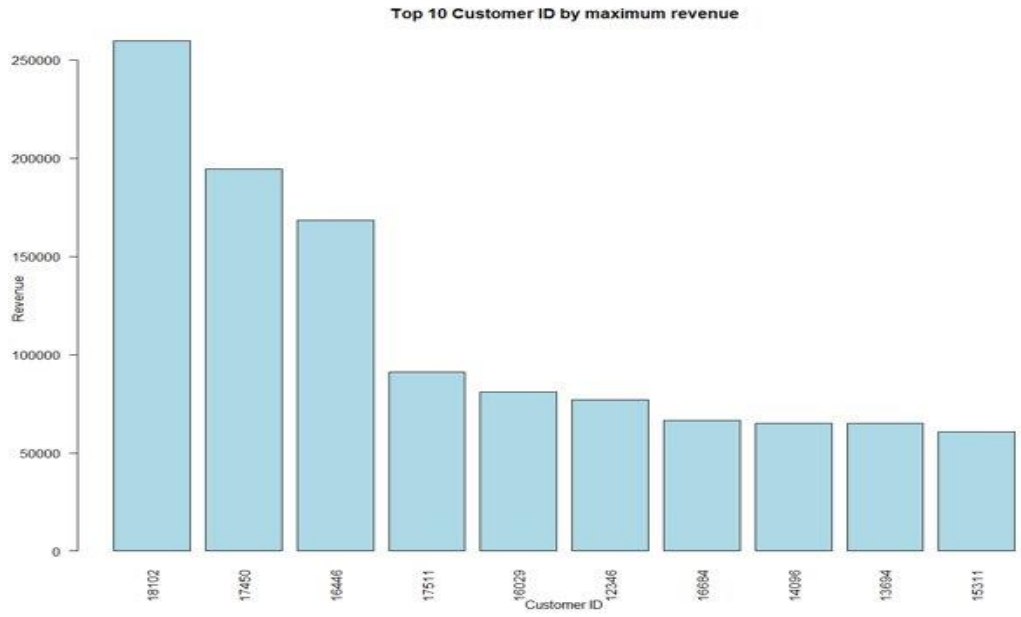
Number of daily transactions by each country



Note. The graph shows the total transactions on the y-axis with the corresponding country name on the x-axis. It shows the United Kingdom has the maximum number of transactions.

Figure 6

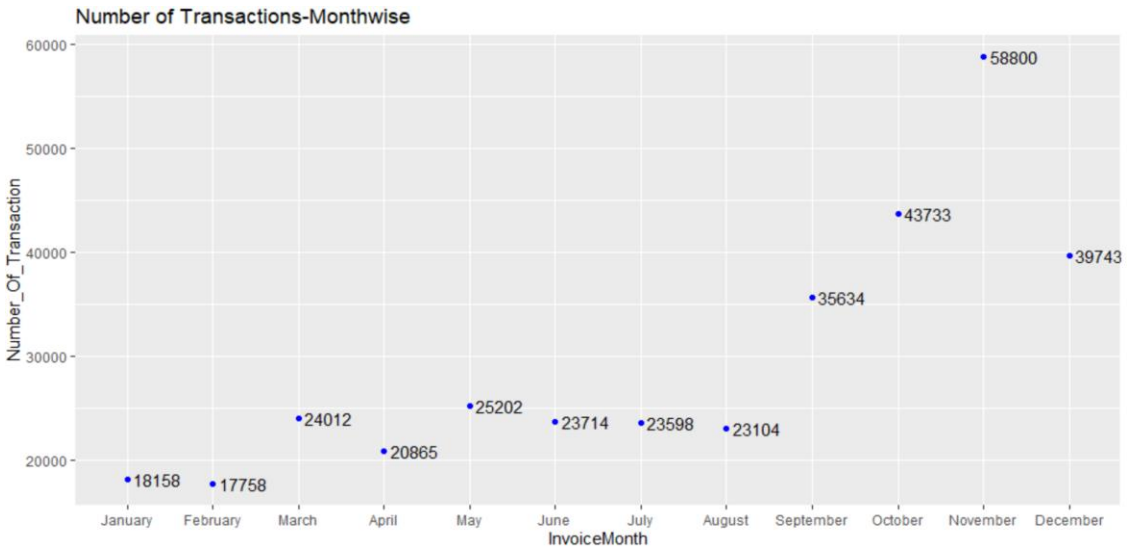
Top 10 customer IDs by the total Invoice price



Note. The graph shows the top 10 customer IDs for the customers accountable for maximum revenue shown on the y-axis with the corresponding IDs on the x-axis.

Figure 7

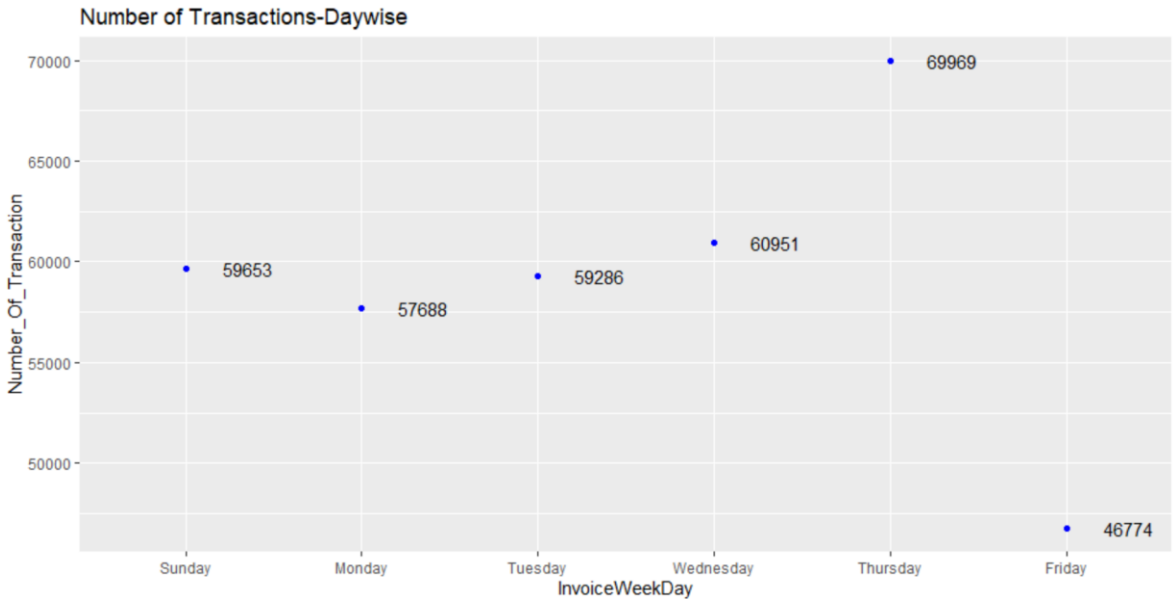
Number of daily transactions by month



Note. The graph shows the number of transactions on the y-axis for each month on the x-axis.

Figure 8

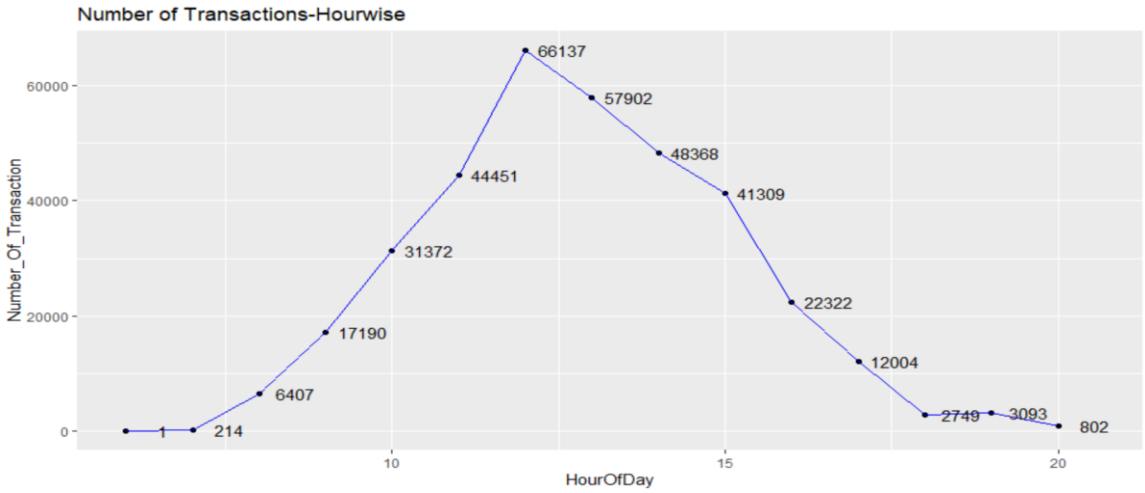
Number of daily transactions by days



Note. The graph shows the number of transactions on the y-axis for each weekday on the x-axis.

Figure 9

Number of daily transactions by the hour of the day



Note. The graph shows the number of transactions on the y-axis at different hours in a day on the x-axis.

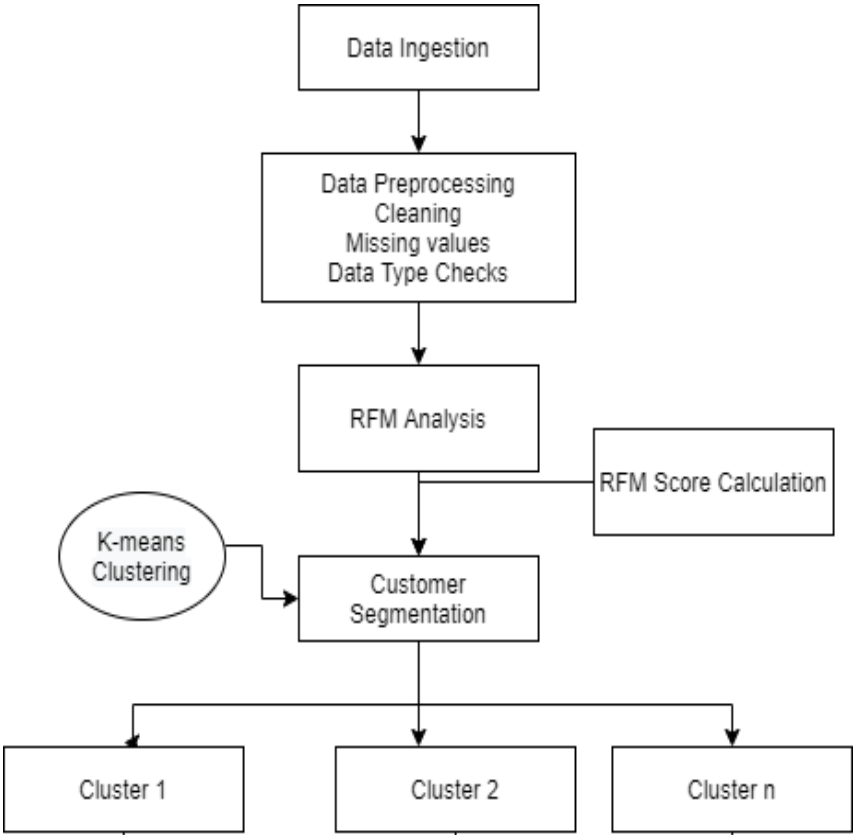
Project Implementation

We used RFM(Recency, Frequency, and Monetary) analysis and K-means clustering algorithm to perform customer segmentation. The objective of the implementation is to segment the customer by combining both the RFM and K-means techniques to get useful insights from their respective results. The implementation process is depicted in Figure 10. The project implementation steps are as follows:

- Read the dataset.
- Data Preprocessing and visualization(Appendices A&B).
- Calculate RFM scores, values for each customer, and manually segment the customers based on the RFM scores(Appendix C).
- Implement K-means clustering using RFM values as variables(Appendix D)
- Analyze and compare the K-means clusters and segments from RFM analysis.

Figure 10

Project architecture using RFM analysis and K-means clustering algorithm



Note. The figure shows the flow of the project architecture using RFM analysis and the K-means clustering algorithm.

Customer Segmentation

Customer segmentation is the process of distributing the business customers into segments based on similar characteristics such as interests, habits, personality, value-based, behavior, demographics, shopping patterns, etc. Dividing the customers into groups helps businesses to identify the needs for each segment and provide suitable marketing strategies for each segment. We have utilized the value-based segmentation for our project using the economic value of each customer in the dataset.

RFM Analysis

RFM analysis is a marketing framework used to analyze and segment customers based on three important factors: Recency, Frequency, and Monetary.

R(Recency) shows how recently a customer made a purchase.

F(Frequency) expresses the total number of transactions for each customer.

M(Monetary) shows how much is the total or average value of transactions for each customer.

With the help of RFM values, customers can be segmented into groups such as loyal customers (high paying), potential loyal, the customers who need more attention, customers who may churn out, etc. We used the function rfm_table_order() from R package "rfm" to implement RFM analysis (Blog, 2019). The rfm_table_order() function mainly accepts five inputs: the dataset, customer id, transaction date, revenue, and analysis date to calculate the RFM score. The output data frame will include recency, frequency, and monetary values as well as individual R, F, and M scores. The individual score is calculated based on the R, F, M values on a scale of 1 to 5. For example, if a customer has spent a lot in business, the corresponding monetary will be 5.

The total RFM score for each customer is calculated by adding their recency, frequency, and monetary scores. The RFM score represents the customer relationship with the companies. The RFM score lies between 3 to 15. We categorize our customers into four groups based on their RFM score as mentioned in Table 1. The cut() function is used to divide the RFM scores for each segment and added a new column in our dataset for customer segments as shown in Figure 11. The total count of customers in each segment is mentioned in Table 2. The RFM analysis implementation codebook is included in Appendix C.

Table 1

The range of combined RFM scores for the manual customer segmentation

Customer Segment	RFM score range
Platinum(Loyal or high paying)	13-15
Gold(Potential Loyal)	10-12
Silver(Promising)	7-9
Bronze(Needs more attention)	3-6

Note. This table shows the range of RFM scores used for manually dividing the customers based on their RFM scores.

Figure 11

Sample of the final dataset using RFM segmentation

X	customer_id	Recency	Frequency	Monetary	recency_score	frequency_score	monetary_score	rfm_group	rfm_score	Customer_Segment
1	12346	326	1	77183.6	1	1	5	115	7	Silver(Promising)
2	12747	3	103	4196.01	5	4	5	545	14	Platinum(Loyal)
3	12748	1	4595	33719.73	5	5	5	555	15	Platinum(Loyal)
4	12749	4	199	4090.88	5	5	5	555	15	Platinum(Loyal)
5	12820	4	59	942.34	5	4	4	544	13	Platinum(Loyal)
6	12821	215	6	92.72	1	1	1	111	3	Bronze(Need Attention)
7	12822	71	46	948.88	3	3	4	334	10	Gold(Potential loyal)
8	12823	75	5	1759.5	2	1	4	214	7	Silver(Promising)
9	12824	60	25	397.12	3	2	2	322	7	Silver(Promising)
10	12826	3	91	1474.72	5	4	4	544	13	Platinum(Loyal)
11	12827	6	25	430.15	5	2	2	522	9	Silver(Promising)
12	12828	3	56	1018.71	5	3	4	534	12	Gold(Potential loyal)

Note. This figure shows the newly added variable Customer_Segment with the segment information for each customer calculated by the RFM implementation.

Table 3

Customer count in each segment calculated by RFM technique

Customer Segment	Score(3-15)	Customer count per segment
Platinum(Loyal)	13-15	833
Gold(Potential Loyal)	10-12	911
Silver(Promising)	7-9	1012
Bronze(Needs Attention)	3-6	1164

Note. This table shows the count of customers in each RFM segment after assigning them a segment manually.

K-means Clustering Implementation

The K-means clustering algorithm is a popular unsupervised machine learning algorithm to group the data into clusters as well as to detect the patterns in the data. The algorithm consists of three steps: Recalculate centroids, reassign clusters, and repeat until it allocates the data point to the nearest centroid to form clusters.

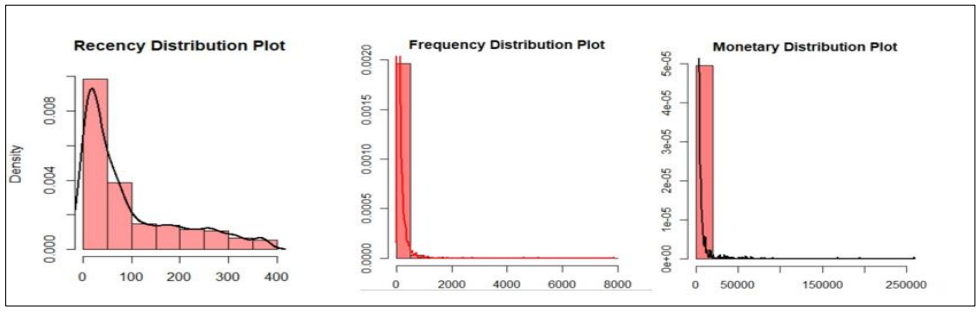
We used the K-means clustering algorithm to group the customers into segments using their recency, frequency, and monetary values. We started the implementation by examining the data distribution in recency, frequency, and monetary columns. We found that data is highly skewed right as shown in Figure 12. The log-transformation is used to remove the right skewness and scale() function to normalize the data as shown in Figure 13.

The Elbow method and Silhouette method are used to find the optimal number of clusters to implement K-means clustering. We picked the results from the elbow method and build the K-means model with three clusters using the recency, frequency, and monetary values in the dataset. The Elbow method and Silhouette method are shown in Figure 14.

After fitting our dataset to the model, we also calculated the total count of customers in each cluster mentioned in Table 3. We also plotted the clusters on a scatter plot using the Plotly package as shown in Figure 15. We analyzed each cluster to understand the criteria of segmentation. The K-means clustering implementation using the R, F, and M values codebook is included in Appendix D.

Figure 12

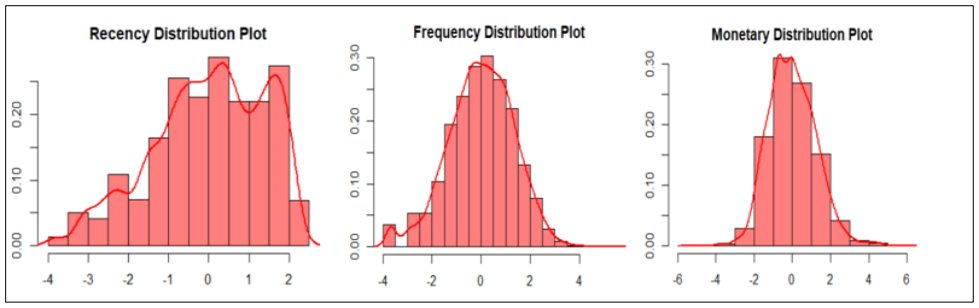
Recency, frequency, and monetary value distribution



Note. This figure shows the data distribution in R(Recency), F(Frequency), and M(Monetary) values.

Figure 13

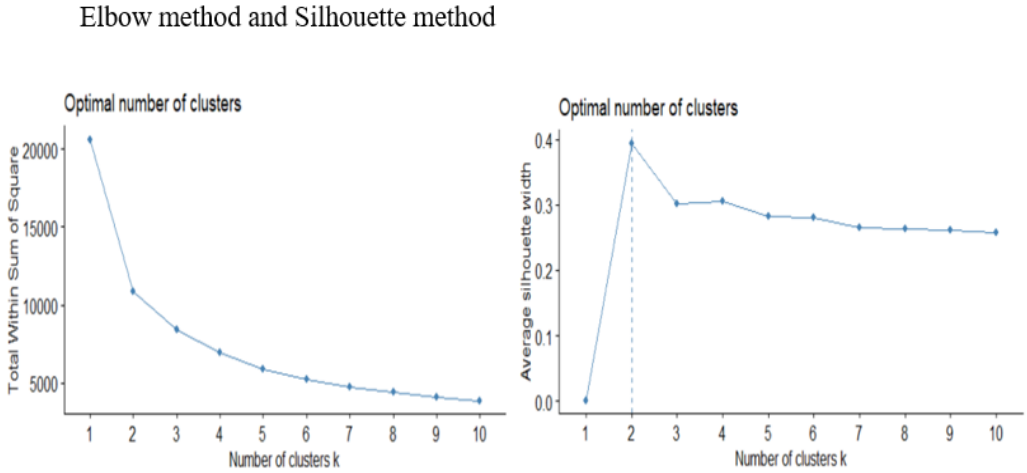
Recency, frequency, and monetary value distribution after scaling



Note. This figure shows the data distribution in R(Recency), F(Frequency), and M(Monetary) values after log-transformation and scaling.

Figure 14

Elbow method and Silhouette methods



Note. This figure shows the result after implementing the Elbow and Silhouette method to find the optimal number of clusters for the dataset.

Table 3

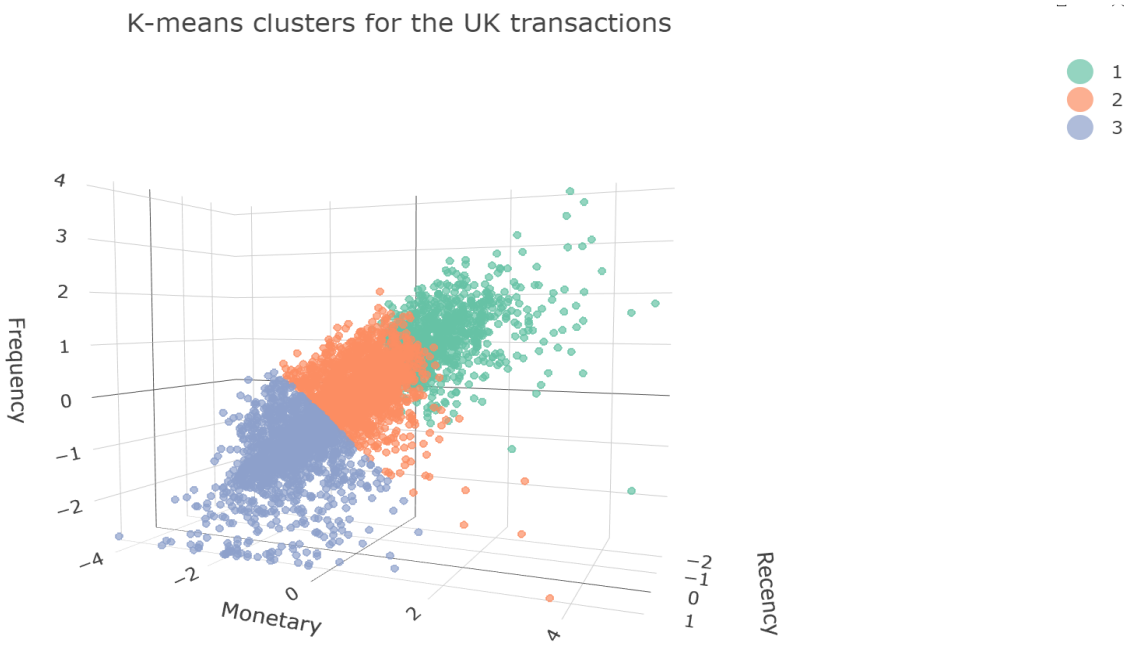
Customer count in three K-means clusters

Cluster	Customer count per cluster
1	845
2	1674
3	1401

Note. This table shows the count of customers in the K-means clusters 1, 2, and 3.

Figure 15

Graphical representation of three clusters



Note. This graph shows the distribution of the United Kingdom customers in three K-means clusters on a 3D scatter plot from Plotly library with Recency, Frequency, and Monetary values on each side.

Results & Evaluations

We merged the findings from the RFM analysis and K-means clustering algorithm to analyze the results which are also shown in Figure 16. The final CSV file contains the RFM segment as well as the K-means cluster column for each customer in the dataset. The elbow method was used to calculate the number of clusters to ensure the accurate performance of the K-means algorithm. K-means model used the variables Recency, Frequency, and Monetary after standardization and scaling.

Figure 16

Sample of the final dataset with each assigned K-means cluster and RFM segments

customer_id	Recency_value	Frequency_value	Monetary_value	recency_score	frequency_score	monetary_score	rfm_score	Customer_Segment	Recency	Frequency	Monetary	Cluster
12346	326	1	77183.6	1	1	5	7	Silver(Promising)	1.438	-2.736	3.777	2
12747	3	103	4196.01	5	4	5	14	Platinum(Loyal)	-1.953	0.727	1.442	1
12748	1	4595	33719.73	5	5	5	15	Platinum(Loyal)	-2.748	3.564	3.113	1
12749	4	199	4090.88	5	5	5	15	Platinum(Loyal)	-1.745	1.219	1.422	1
12820	4	59	942.34	5	4	4	13	Platinum(Loyal)	-1.745	0.31	0.245	1
12821	215	6	92.72	1	1	1	3	Bronze(Need Attention)	1.137	-1.397	-1.615	3
12822	71	46	948.88	3	3	4	10	Gold(Potential loyal)	0.336	0.124	0.25	2
12823	75	5	1759.5	2	1	4	7	Silver(Promising)	0.375	-1.534	0.745	3
12824	60	25	397.12	3	2	2	7	Silver(Promising)	0.214	-0.331	-0.448	2

Note. This figure shows the sample of the final dataset with segment information from RFM analysis as well as cluster information from K-means clustering for each customer.

After analyzing all three K-means clusters and customer segments from RFM analysis, we concluded that the K-means clustering algorithm grouped most of the silver and gold customers segment into one cluster which is cluster 2 as mentioned in Table 4 and Figure 17. These customers fall under the category of potentially loyal and promising. Hence, combining them could be beneficial for huge businesses to save marketing resources.

The RFM analysis and K-means clustering kept the customers from the platinum(loyal or high paying customer) and bronze(customers who need more attention or might churn out) categories into separate clusters as shown in Figure 16. It is ideal for situations when the marketing team needs to customize their marketing strategies targeting each group according to their needs.

Table 4 Customer segment analysis comparison and results for RFM and K-means clustering

RFM Analysis			K-Means Clustering		
RFM Score	Segment	Customer Count	Cluster	Segment	Customer Count
13-15	Platinum(Loyal)	833	1	Platinum	845
10-12	Gold(Potential Loyal)	911	2	Silver & Gold	1674
7-9	Silver(Promising)	1012			
3-6	Bronze(Needs Attention)	1164	3	Bronze	1401

Note. This table comprises the results from RFM analysis and K-means clustering and provides customer count in each RFM segment and K-means three clusters.

Figure 17

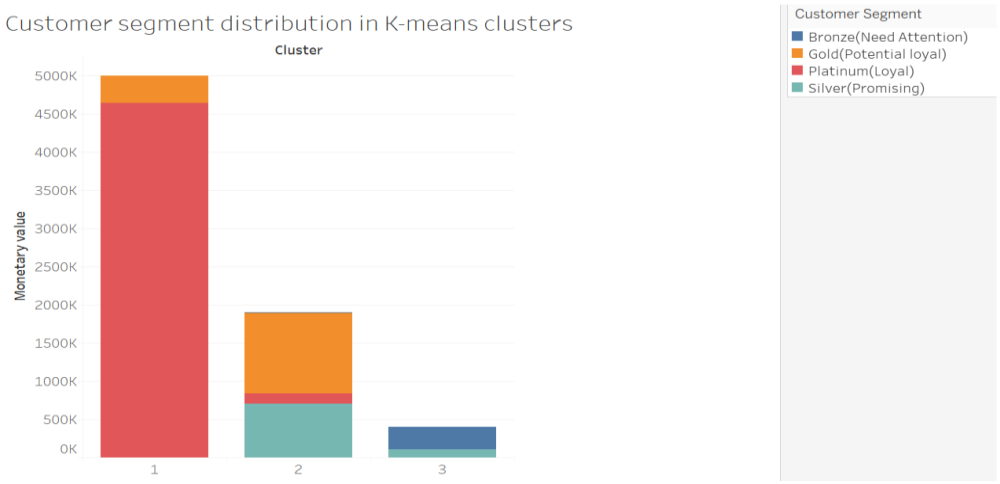
Customer segment analysis comparison and results for RFM and K-means clustering

Note. This figure is a graphical representation of the information available in Table 4 with cluster information on the x-axis and customer count on the y-axis. Cluster 1 contains platinum customers, cluster 2 combines the gold and silver customers, and cluster 3 mostly holds gold customers.

The use of Recency, Frequency, and Monetary variables can be flexible and industry-specific. For example, if a nonprofit organization wants to segment its customers to decide on prospective donors. They will consider the monetary value for each customer to decide to ask them for another donation as shown in Figure 18. We used economic values to segment the customer in the dataset. The descriptive columns such as product names and their description can be further used to perform product segmentation using natural language processing techniques.

Figure 18

Customer segmentation using only monetary values



Note. This figure shows the result of customer segmentation using the monetary values for each customer with cluster information on the x-axis and customer count on the y-axis. Cluster 1 mostly contains platinum customers who are responsible for maximum revenue.

Conclusion

After the comparative analysis, we concluded that the RFM segmentation and K-Means clustering will work great for customer segmentation. The combination of RFM analysis with K-means clustering has certainly improved the results for customer segmentation. The data stored in each cluster can lead to greater insights into each customer group. Our implementation is not only limited to use Recency, Frequency, and Monetary values. We can use different information as well such as age or income to segment the customers to derive insights. The marketing team can use derived insights from each cluster to develop improved marketing plans that can target a specific group of customers according to their needs. The improved marketing strategies target to increase profit and boost customer loyalty. Figure 17 and Figure 18 also show the support for the business paradigm that states that 80% of business comes from 20% of existing customers.

Our project emphasizes using recency, frequency, and monetary values for customer segmentation. The project can be extended to use other factors as well such as gender, age, behavior, etc. to implement customer segmentation. This project presents a wonderful example of prescriptive analysis. The implementation of RFM analysis requires a little amount of manual work for segmenting the customer however it provides the flexibility of choosing the desired parameters as well. Machine learning algorithms such as K-means clustering can automate the whole customer segmentation process with little effort.

The implementation of the RFM, K-means and their combination are solely based on the question we are trying to solve. Customer segmentation is vital for any business especially during COVID-19 as most of the people are shopping from their homes. The segmentation based on demographic location can be highly beneficial for businesses with delivery services.

References

Aditya. (2020, October 4). *A Simple Explanation of K-Means Clustering*. (Data Science Blogathon) Retrieved from https://www.analyticsvidhya.com/blog/2020/10/a-simple-explanation-of-k-means-clustering/

Blog, R. A. (2019, Feb 11). *RFM Analysis in R*. (R bloggers) Retrieved from https://www.r-bloggers.com/2019/02/rfm-analysis-in-r/

Brodbeck, T. (2020, April 13). *How to Evolve Your Marketing Strategy During COVID-19*. (FoundSM) Retrieved from https://www.foundsm.com/blog/marketing-covid19/

CDC. (2021, March 23). *About COVID-19*. (CDC) Retrieved from https://www.cdc.gov/coronavirus/2019-ncov/your-health/about-covid-19.html

Chen, D. D. (n.d.). *Online Retail Data Set*. (UCI- Center for Machine Learning and Intelligent Systems) Retrieved from https://archive.ics.uci.edu/ml/machine-learning-databases/00502/

Gavin, M. (2019, January 15). *4 EXAMPLES OF BUSINESS ANALYTICS IN ACTION*. (Harvard Business School Online) Retrieved from https://online.hbs.edu/blog/post/business-analytics-examples

Gavin, M. (2019, January 15). *4 EXAMPLES OF BUSINESS ANALYTICS IN ACTION*. (Harvard Business School Online) Retrieved from https://online.hbs.edu/blog/post/business-analytics-examples

Gersema, E. (2020, November 30). *Business closures and partial reopenings due to COVID-19 could cost the U.S. trillions*. Retrieved from https://news.usc.edu/178979/business-closures-covid-19-pandemic-united-states-gdp-losses/

Goyal, S. (2011). *The basis of market segmentation: : a critical review of literature*. (European Journal of Business and Management) Retrieved from https://core.ac.uk/download/pdf/234624114.pdf

Hannah Ritchie, E. O.-O.-G. (2020). *Coronavirus (COVID-19) Vaccinations*. (Our World in Data) Retrieved from https://ourworldindata.org/covid-vaccinations?country=OWID_WRL

Makhija, P. (2021, June 3). *RFM analysis for Customer Segmentation*. Retrieved from https://clevertap.com/blog/rfm-analysis/

Worldmeters. (2021). *COVID-19 CORONAVIRUS PANDEMIC*. (Worldmeters) Retrieved from https://www.worldometers.info/coronavirus/

Appendices

Appendix A – Data Preprocessing

```
# ANLY 506-51- B-2021/Summer-Exploratory Data Analysis
# Course Final Project
# Customer Segmentation using RFM analysis and K-means clustering
#Submitted to Dr. Doaa Taha
##############################

# Clear the environment
rm(list = ls())

# Setting work directory
setwd("C:\\Users\\itsni\\OneDrive - Harrisburg University\\Data
Analysis\\Project2")

# Install packages and libraries
install.packages("tidyverse")
library(tidyverse)
install.packages("lubridate")
library(lubridate)

# Read the CSV data file
eCommData <- read.csv("C:\\Users\\itsni\\OneDrive - Harrisburg
University\\Semester 5\\Data Analysis\\Project2\\e-commerce_data.csv")
dim(eCommData)

# Data pre-processing steps

# Check data set description
glimpse(eCommData)

##############################

# Find the total null values in the data set
table(is.na(eCommData))

# Find the total null values in each column
colSums(is.na(eCommData))

# Remove the null values
eCommData <- na.omit(eCommData)

# Keep unique rows
eCommData %>% distinct()

# Check the summary of the data set to analyze numerical columns
summary(eCommData)
```

```
# Remove observations with negative values in Quantity column
dim(eCommData[eCommData$Quantity < 0, ])
eCommData <- eCommData[eCommData$Quantity > 0, ]

# Remove observations with 0 values in UnitPrice column
dim(eCommData[eCommData$UnitPrice == 0, ])
eCommData <- eCommData[eCommData$UnitPrice > 0, ]
dim(eCommData)

# Add a new column "TotalInvoicePrice" to calculate the the total invoice
amount:  UnitePrice * Quantity
eCommData$TotalInvoicePrice <- eCommData$UnitPrice * eCommData$Quantity

# Convert the InvoiceDate column from character type to datetime type
eCommData$InvoiceDateTime = mdy_hm(eCommData$InvoiceDate)

# Extract date, month, year, weekday, and hour from invoice date
eCommData$InvoiceFullDate <- date(eCommData$InvoiceDateTime)
eCommData$InvoiceMonth <- as.factor(month(eCommData$InvoiceDateTime, label
= TRUE, abbr = FALSE))
eCommData$InvoiceWeekDay <- as.factor(wday(eCommData$InvoiceDateTime,
label = TRUE, abbr = FALSE))
eCommData$InvoiceYear <- as.factor(year(eCommData$InvoiceDateTime))
levels(eCommData$InvoiceYear) <- c(2010,2011)
eCommData$HourOfDay <- as.factor((hour(eCommData$InvoiceDateTime)))

# Delete the original InvoiceDate column
eCommData <- within(eCommData, rm(InvoiceDate))

# Change the Customer ID and Country column from character type to factor
eCommData$Country <- as.factor(eCommData$Country)
eCommData$CustomerID <- as.factor(eCommData$CustomerID)

# Write the data
write.csv(eCommData,
          "C:\\Users\\itsni\\OneDrive - Harrisburg University\\Data
Analysis\\Project2\\cleanedEcommData.csv")

# Filter the transactions for the United Kingdom customers
eCommDataUK <- eCommData %>% filter(eCommData$Country == "United Kingdom")
dim(eCommDataUK)
glimpse(eCommData)

# Write the UK data
write.csv(eCommDataUK,
          "C:\\Users\\itsni\\OneDrive - Harrisburg University\\Data
Analysis\\Project2\\eCommDataUK.csv")
```

Appendix B – Data Visualization

```
# Data Visualization
###############################################

install.packages("sqldf")
library(sqldf)
install.packages("ggplot2")
library(ggplot2)

################################################

# Total revenue for each country in 2010
ggplot(country.total10, aes(x= Countries, y=Revenue2010, color=
Countries)) +   geom_count() +   expand_limits( x = c(0,NA), y = c(0,NA))
+   scale_y_continuous(labels = scales::comma)
+   scale_size_area(max_size = 3) +   coord_cartesian(ylim = c(0, 700000),
expand = TRUE) +   labs(title = "Total Revenue by Countries (2010)"
,        subtitle = "The 7 Countries with most Revenue in 2010",        x
= "Countries",        y = "Revenue") +   theme_fivethirtyeight()
+   theme(axis.title = element_text())
2010

################################################

# Total revenue for each country in 2011
ggplot(country.total11, aes(x= Countries, y=Revenue2011, color=
Countries)) +   geom_count() +   expand_limits( x = c(0,NA), y = c(0,NA))
+   scale_y_continuous(labels = scales::comma)
+   scale_size_area(max_size = 3) +   labs(title = "Total Revenue by
Countries (2011)" ,        subtitle = "The 7 Countries with most Revenue
in 2011",        x = "Countries",        y = "Revenue")
+   theme_fivethirtyeight() +   theme(axis.title = element_text())
2011

################################################

#Top 10 customer IDs with maximum revenue
Top10_CustomerID <- sqldf("SELECT CustomerID, sum(TotalInvoicePrice) as
TotalInvoicePrice FROM eCommDataUK
      GROUP BY CustomerID
      ORDER BY sum(TotalInvoicePrice) DESC
      LIMIT 10")
Top10_CustomerID
barplot(Top10_CustomerID$TotalInvoicePrice,
        main = "Top 10 Customer ID by maximum revenue",
        xlab = "Customer ID",
        ylab = "Revenue",
        names.arg = Top10_CustomerID$CustomerID,
        col = "lightblue",
        las=2)
```

```
# Monthly distribution of the UK transaction
tranmo <- sqldf sqldf("SELECT InvoiceMonth,
                count(InvoiceNo) as Number_Of_Transaction
                FROM eCommDataUK
                GROUP BY InvoiceMonth
                ORDER BY Number_Of_Transaction DESC
                "))
tranmo
cleanup = theme(panel.grid.major = element_blank(),
                panel.grid.minor = element_blank(),
                panel.background = element_blank(),
                axis.line.x = element_line(color = 'black'),
                axis.line.y = element_line(color = 'black'),
                legend.key = element_rect(fill = 'white'),
                text = element_text(size = 15))
ggplot(tranmo, aes(x=InvoiceMonth, y=Number_Of_Transaction))+
  geom_point(color="blue")+geom_text(label=tranmo$Number_Of_Transaction,
nudge_x = .35, check_overlap = T)+
ggtitle("Number of transactions-Monthwise")+
  cleanup

#######################################################
# Day-wise distribution of the UK transactions
dayTransactionUK <- sqldf("SELECT InvoiceWeekDay, count(InvoiceNo) as
Number_Of_Transaction
                FROM eCommDataUK
                GROUP BY InvoiceWeekDay
                ORDER By Number_Of_Transaction DESC
                ")
ggplot(dayTransactionUK, aes(x=InvoiceWeekDay, y=Number_Of_Transaction))+
  geom_point(color="blue")+
  geom_text(label=dayTransactionUK$Number_Of_Transaction, nudge_x = .35,
check_overlap = TRUE)+
  ggtitle("Number of Transactions-Daywise")

#######################################################
# Hourly distribution of the UK transactions
hourTransactionUK <- sqldf("SELECT HourOfDay, count(InvoiceNo) as
Number_Of_Transaction
                FROM eCommDataUK
                GROUP BY HourOfDay
                ORDER By Number_Of_Transaction DESC
                ")
hourTransactionUK$HourOfDay <-
as.numeric(as.character(hourTransactionUK$HourOfDay))
ggplot(hourTransactionUK, aes(x=HourOfDay, y=Number_Of_Transaction))+
  geom_point(color="black")+
  geom_line(color="blue")+
  geom_text(label=hourTransactionUK$Number_Of_Transaction, nudge_x = .55,
check_overlap = TRUE)+
  ggtitle("Number of Transactions-Hourwise")
```

Appendix C – RFM analysis

```
# Customer segmentation through RFM analysis
##############################
install.packages("rfm")
library(rfm)

# Check the recent transaction date
max(eCommDataUK$InvoiceFullDate)

analysis_date <- as.Date("2011-12-10")

rfm_result <- rfm_table_order(eCommDataUK, CustomerID,
                              InvoiceFullDate, TotalInvoicePrice,
                              analysis_date)

# Check the class for rfm_result
class(rfm_result)

# Write rfm_result in a CSV file
write.csv(rfm_result$rfm, "C:\\Users\\itsni\\OneDrive - Harrisburg
University\\Data Analysis\\Project2\\rfm_results.csv")

# Read the rfm dataset
rfm_eCommDataUK <- read.csv("C:\\Users\\itsni\\OneDrive - Harrisburg
University\\Data Analysis\\Project2\\rfm_results.csv")
glimpse(rfm_eCommDataUK)

# Rename the recency_days,transaction_count, and amount columns to
Recency, Frequency, and Monetary values
rfm_eCommDataUK <-
  rename(
    rfm_eCommDataUK,
    Recency_value= recency_days,
    Frequency_value= transaction_count,
    Monetary_value=amount,
    )

# Change calculation for rfm_score
rfm_eCommDataUK$rfm_score =
rfm_eCommDataUK$recency_score+rfm_eCommDataUK$frequency_score+rfm_eCommDat
aUK$monetary_score

# Delete the date_most_recent column from the data set
rfm_eCommDataUK <- within(rfm_eCommDataUK, rm(date_most_recent))

glimpse(rfm_eCommDataUK)

# RFM score will be from 3 to 15
describe(rfm_eCommDataUK$rfm_score)
```

```
# Assign Segment to each customer
#Platinum(Loyal): 13-15, gold(potential loyal): 10-12, silver(Promising):
7-9, bronze(need attention):3-6
# Cut() function will start from 3
rfm_eCommDataUK$Customer_Segment <- cut(rfm_eCommDataUK$rfm_score,
                                        breaks = c(2, 6, 9, 12, Inf),
                                    labels = c("Bronze(Need Attention)",
                                               "Silver(Promising)",
                                               "Gold(Potential loyal)",
                                               "Platinum(Loyal)"))

write.csv(rfm_eCommDataUK, "C:\\Users\\itsni\\OneDrive - Harrisburg
University\\Data Analysis\\Project2\\rfmResultsCustomerSegments.csv")

# Plotting the RFM customer segment data set
install.packages("ggplot2")
library(ggplot2)
install.packages("plotly")
library(plotly)

# Check customers in each segment
table(rfm_eCommDataUK$Customer_Segment)

# Bar plot for RFM customer segmentation
png(file = "barchart_CS.png")
ggplot(rfm_eCommDataUK) +geom_bar(aes(x = Customer_Segment, fill =
Customer_Segment)) +theme(axis.text.x=element_text(angle=45,hjust=1))
+labs(title = "Customer Segmentation using RFM analysis")
dev.off()

# Scatter plot for RFM customer segmentation
p <- plot_ly(rfm_eCommDataUK, x=~Recency_value, y=~Monetary_value,
             z=~Frequency_value, color=~Customer_Segment) %>%
  add_markers(size=1.5)
p

############################################################
```

Appendix D – K-means Clustering

```
#Customer segmentation using K-means algorithm
##############################
# Create new columns Recency, Frequency, and Monetary from RFM values for
normalization and scaling purpose

rfm_eCommDataUK$Recency = rfm_eCommDataUK$Recency_value
rfm_eCommDataUK$Frequency = rfm_eCommDataUK$Frequency_value
rfm_eCommDataUK$Monetary = rfm_eCommDataUK$Monetary_value

# Check the distribution in Recency, Frequency, and Monetary value columns

# Install package for describe function
install.packages("psych")
library(psych)
par(mar=c(3,3,3,3))

# Recency Distribution Plot
describe(rfm_eCommDataUK$Recency)
hist(rfm_eCommDataUK$Recency,
     main = "Recency Distribution Plot",
     col = rgb(1, 0, 0, 0.4),
     freq = FALSE,
     xlab="Recency",
     ylab = "Density",
     border ="black")
lines(density(rfm_eCommDataUK$Recency), lwd = 2, col = 'black')

# Frequency Distribution Plot
describe(rfm_eCommDataUK$Frequency)
hist(rfm_eCommDataUK$Frequency,
     main = "Frequency Distribution Plot",
     col = rgb(1, 0, 0, 0.5),
     freq = FALSE,
     xlab="Frequency",
     ylab = "Density",
     border ="black")
lines(density(rfm_eCommDataUK$Frequency), lwd = 2, col = 'black')

# Monetary Distribution Plot
describe(rfm_eCommDataUK$Monetary)
hist(rfm_eCommDataUK$Monetary,
     main = "Monetary Distribution Plot",
     col = rgb(1, 0, 0, 0.5),
     freq = FALSE,
     xlab="Monetary",
     ylab = "Density",
  border ="black")

lines(density(rfm_eCommDataUK$Monetary), lwd = 2, col = 'black')
##############################################################

# Distribution shows that data is right skewed(positive skewed))
```

```
# Using log-transformation to remove the right-skewness in Recency,
Frequency, and Monetary columns

rfm_eCommDataUK[c("Recency","Frequency","Monetary")] <-
lapply(rfm_eCommDataUK[c("Recency","Frequency","Monetary")], log)

# rfm_eCommDataUK[11:13] <- lapply(rfm_eCommDataUK[11:13], log)

# Scaling the data to get R, F, and M on same level
rfm_eCommDataUK[c("Recency","Frequency","Monetary")] <-
as.data.frame(scale(rfm_eCommDataUK[c("Recency","Frequency","Monetary")]))
rfm_eCommDataUK[c("Recency","Frequency","Monetary")] <-
round(rfm_eCommDataUK[c("Recency","Frequency","Monetary")], 3)
glimpse(rfm_eCommDataUK)

# Check the distribution again for Recency, Frequency, Monetary
# Recency
hist(rfm_eCommDataUK$Recency,
     main = "Recency Distribution Plot",
     col = rgb(1, 0, 0, 0.5),
     freq = FALSE,
     xlab="Recency",
     ylab = "Density",
     border ="black")
lines(density(rfm_eCommDataUK$Recency), lwd = 2, col = 'red')

#Frequency
hist(rfm_eCommDataUK$Frequency,
  main = "Frequency Distribution Plot",
     col = rgb(1, 0, 0, 0.5),
     freq = FALSE,
     xlab="Frequency",
     ylab = "Density",
     border ="black")
lines(density(rfm_eCommDataUK$Frequency), lwd = 2, col = 'red')

# Monetary
hist(rfm_eCommDataUK$Monetary,
     main = "Monetary Distribution Plot",
     col = rgb(1, 0, 0, 0.5),
     freq = FALSE,
     xlab="Monetary",
     ylab = "Density",
     border ="black")
lines(density(rfm_eCommDataUK$Monetary), lwd = 2, col = 'red')

#####################################################################
```

```
# Find the optimal number of clusters for the data set: Elbow method and
Silhouette Method

install.packages("factoextra")
library(factoextra)

# Elbow method to find optimal number of clusters(k) k = 3"
fviz_nbclust(rfm_eCommDataUK[c("Recency","Frequency","Monetary")],
kmeans,method = "wss",nstart=50)

# Silhouette Method (gave k=2 as usual)
fviz_nbclust(rfm_eCommDataUK[c("Recency","Frequency","Monetary")], kmeans,
method = 'silhouette',nstart=50)

################################################

# Fit k-means clustering model on normalized R, F, and M values
set.seed(1234)
kMeanResult <-
kmeans(rfm_eCommDataUK[c("Recency","Frequency","Monetary")], centers = 3
,nstart = 50 ,iter.max = 1000)
rfm_eCommDataUK$Cluster <- kMeanResult$cluster
class(rfm_eCommDataUK$Cluster)
rfm_eCommDataUK$Cluster <- as.factor(rfm_eCommDataUK$Cluster)
glimpse(rfm_eCommDataUK)
table(rfm_eCommDataUK$Cluster)

# Plot the clusters on scatter plot
p <- plot_ly(rfm_eCommDataUK, x=~Recency, y=~Monetary,
             z=~Frequency, color=~Cluster)
p <- p %>% add_markers(size=1.5)
p <- p %>% layout(title="K-means clusters for the UK transactions")
p

# Write the final data set
write.csv(rfm_eCommDataUK, "C:\\Users\\itsni\\OneDrive - Harrisburg
University\\Data Analysis\\Project2\\final_data_rfm_kmeans.csv")

################################################
# Check Data point in each cluster
rfm_cluster1 <- rfm_eCommDataUK %>% filter(rfm_eCommDataUK$Cluster == 1)
rfm_cluster2 <- rfm_eCommDataUK %>% filter(rfm_eCommDataUK$Cluster == 2)
rfm_cluster3 <- rfm_eCommDataUK %>% filter(rfm_eCommDataUK$Cluster == 3)

write.csv(rfm_cluster1, "C:\\Users\\itsni\\OneDrive - Harrisburg
University\\Data Analysis\\Project2\\rfm_cluster1.csv")
write.csv(rfm_cluster2, "C:\\Users\\itsni\\OneDrive - Harrisburg
University\\Data Analysis\\Project2\\rfm_cluster2.csv")
write.csv(rfm_cluster3, "C:\\Users\\itsni\\OneDrive - Harrisburg
University\\Data Analysis\\Project2\\rfm_cluster3.csv")
################################################
```