# KMeans Clustering and Time Series Forecasting on Online retail dataset of UCI Machine Learning Repository

**ABSTRACT**

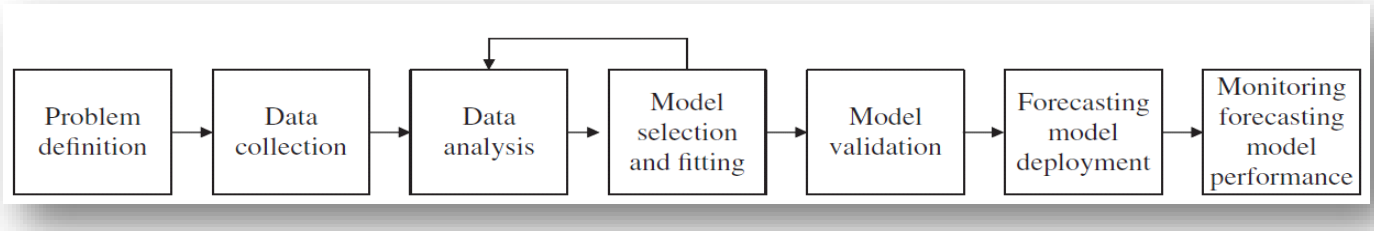
The purpose of this research is to assist the business with improved methodology and understand their clients for better sales and marketing of products. Machine learning methodology is used to predict the sales pattern using Time Series analysis, whereas, K means Clustering is used for and targeting the class and cluster of Clients for predicting the recency, Frequency and Monetary values of the Clients / customers from the dataset of Online retail. Time Series Analysis of Online Retail have become important to predict about the future sales & marketing of products. Whereas, using K Mean Clustering we have divided the Clients into clusters based on the Recency, Frequency and Monetary Score. If the store employees are able to communicate to those clients who are recent and frequent then they will be able to advertise more about the other different products and benefits of different discounts or it may be some reward cards which can be used in future purchases. With the help of time series predictions, the store will be able to manage the sales and employability and using exploratory data analysis we will be able to find out the peak time of the day where the transactions are occurring. The data is structured and is having a lot of missing values and cancelled orders which are to be taken care of while working with the data. Under the Exploratory Data Analysis, we have found that most of the client's transactions are from United Kingdom even if it is an online store. By using the time series analysis and the Kmean Clustering algorithm we were able to predict the future sales and the have pointed out the most recent frequent and monetary scores of clients from different countries.

## ** INTRODUCTION **

Forecast is prognosis of some imminent case/cases or episode/episodes. Anticipating is an imperative issue that traverses numerous fields including business and industry, government, financial matters, ecological sciences, medication, sociology and governmental issues. Neuroscientists have even studied how our brains make decisions about how much we're willing to pay for a product (Lindström 2010).

Time series is broadly classified into qualitative modelling and quantitative modelling. Qualitative forecasting techniques inhabits biased essence and require experience on the part of professionals. Qualitative forecasts are tested on insufficient or no historical data. Quantitative forecasting techniques are applied to historical data and an estimated model. The model properly encloses arrangements in the data and transmits a statistical relationship between earlier and prevailing values of the variable. Then that forecasted model is used to predict the future pattern of the data. The Quantitative forecasting operates using hypothesize past and prevailing behaviour into the future.

A process can be defined as transformation of connected activities from particular inputs into single or multiple outputs. The activities in the forecasting process are:




Time Series Forecasting Approach

A day to day business problem is to project impending sales and future income. Based on data from previous transactions, data science can aid to enhance forecasts and develop models that describe the main factors of influence. Online e-commerce is a structure of commercial business which allows customers to directly buy services and products from traders over the internet through a web browser. This paper shows a machine learning prediction of the further stages of sales and improvement using time series forecasting method. Looking at the data set from machine learning repository for Online retail it is evident that there are conclusions that can be drawn and predictions can be made on the future sales of the products for the next year. The bases of the advanced methods are usually some of the structural and intensity-based approaches. (Daróczi 2013)

**Critical Summary of Methodical Approach:**

Data Science uses Statics and Machine Informative Models to Visualize data. A **time series defines** time-oriented or historical sequence of investigation on interest variables. Time Series forecasting signifies a sequence of data set or combination of packages with arbitrary errors. Forecasting is the extraordinary form of data science and is underestimated with respect to frilly modelling techniques, converting the data into Information that can be further refined to extract relevant features is believed as the most essential part of any analysis. In this article there will be a discussion on the data cleaning, data extraction, time series forecasting and will be making a prediction on the dataset of online retail. **The *online forecasting process* follows an iterative concept, where an initial, already accurate forecast is provided as a first solution that is afterwards iteratively refined over time (Dannecker, 2015).**

The central package used in this modelling is tidyverse for modelling and tidyquant is used for additional design functionalities.

Data Set: 'UC Irvines ML repository (converted from xlsx to csv).
The dataset is a transnational dataset which consist of all transactions between 01-12-2010 and 09-12-2011 for a registered online store based in UK.

**Data Analysis: -**

To interpret data from a csv format file we use read.csv() function and data is containing a time/date column, so we will be converting the same data for reading it correctly by function col_datetime().
The data consist the below features: -

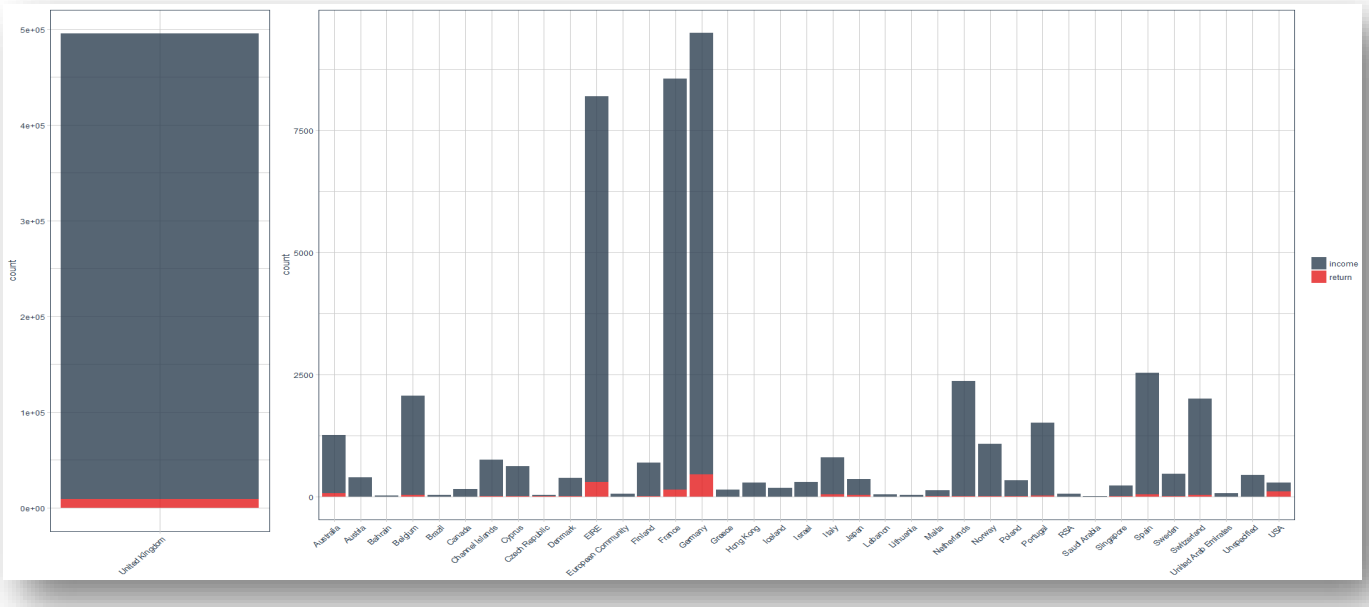
InvoiceNo: - Each transaction is assigned with an exclusive Invoice number. Ordered cancellation is indicated by a letter “c” followed by the invoice number.
StockCode: - Product/Item code authorized to each specific product.
Description: - Product/Item name.
Quantity: - Per transaction batch of individual product/item.
InvoiceDate: - per transaction date/time.
UnitPrice: - per unit price of product in sterling.
CustomerID: - Unique customer number for each client.
Country: - residential country of each customer.

As read.csv() function generates a tibble and is included in tidyverse, we added more columns to the csv file as:
day: Invoice date of transaction created
time: Invoice time of transaction created
month: month of transaction created
income: Income generated for each transaction
income_return: description of transaction in relation with income or loss.

**Exploratory Data Analysis: -**

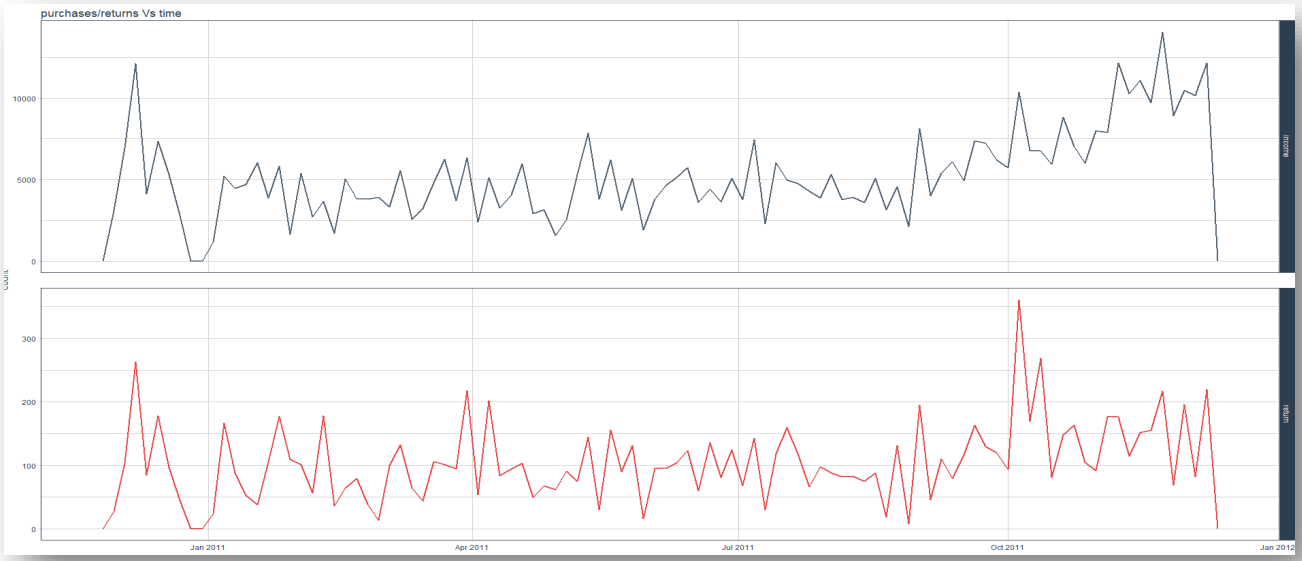
We will be creating different visualizations for different features of our data set so as to decide the variables used in final dataset. These data visualized model will help us to assess the final data modelling and features.

Country Based Transaction: -
The registered online store is Virtually based in UK, but the customers are trading from 37 different countries.
The bar plot below shows the distribution of customer from various regions.



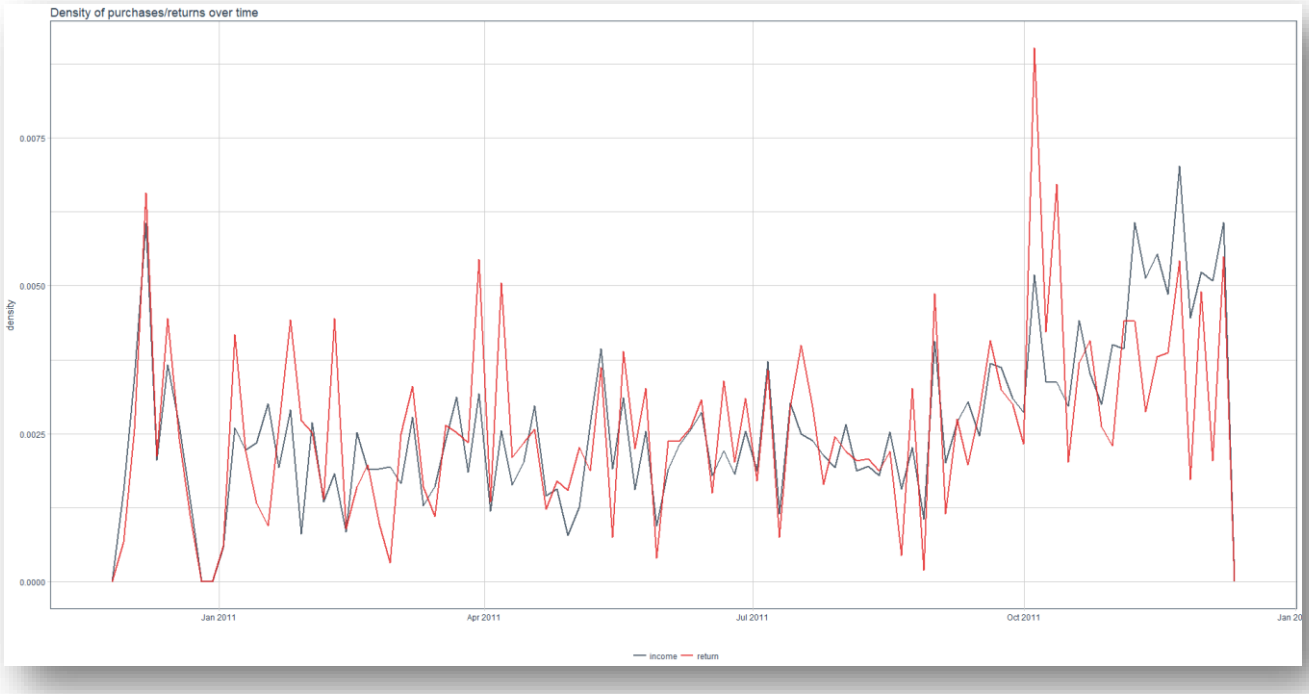

Transaction based on Country

Transaction based on Time: -
We are using a frequency polygon to get the transaction based on time. From the figure its visible that the product purchases by customers have marginally heightened in the last few months of the dataset, whereas the number of item returned remain comparatively stable.




Purchase/Return over Time

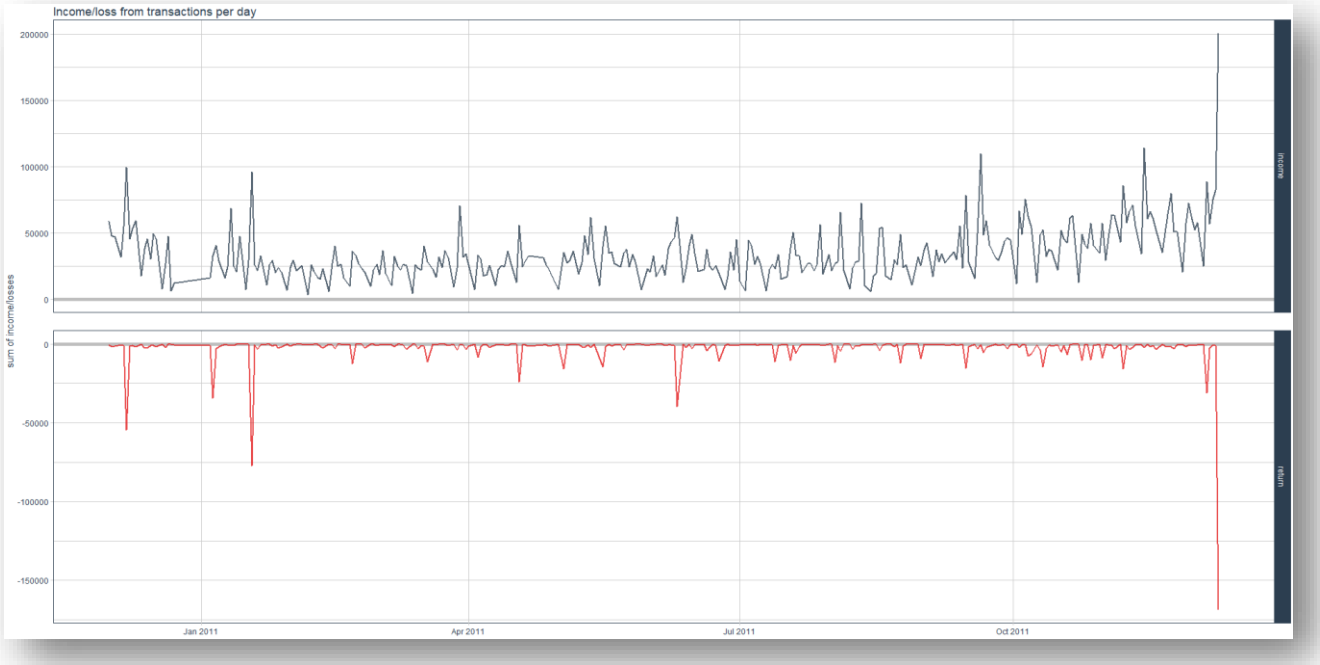
It is a bit problematic to compare and visualize both the above features in the same frequency polygon as the statics of purchase is much bigger than that of returns. From the plot of purchase/return over the relation between purchases and return based on time can be easily identified. **Traditional statistical techniques or regression-based models are not appropriate for prediction of economic time-series as the external parameters influencing the series in most circumstances are not clearly known (Konar, 2017).**




Density over time (Purchase/Return)

**Income gain and loss from the transactions: -**

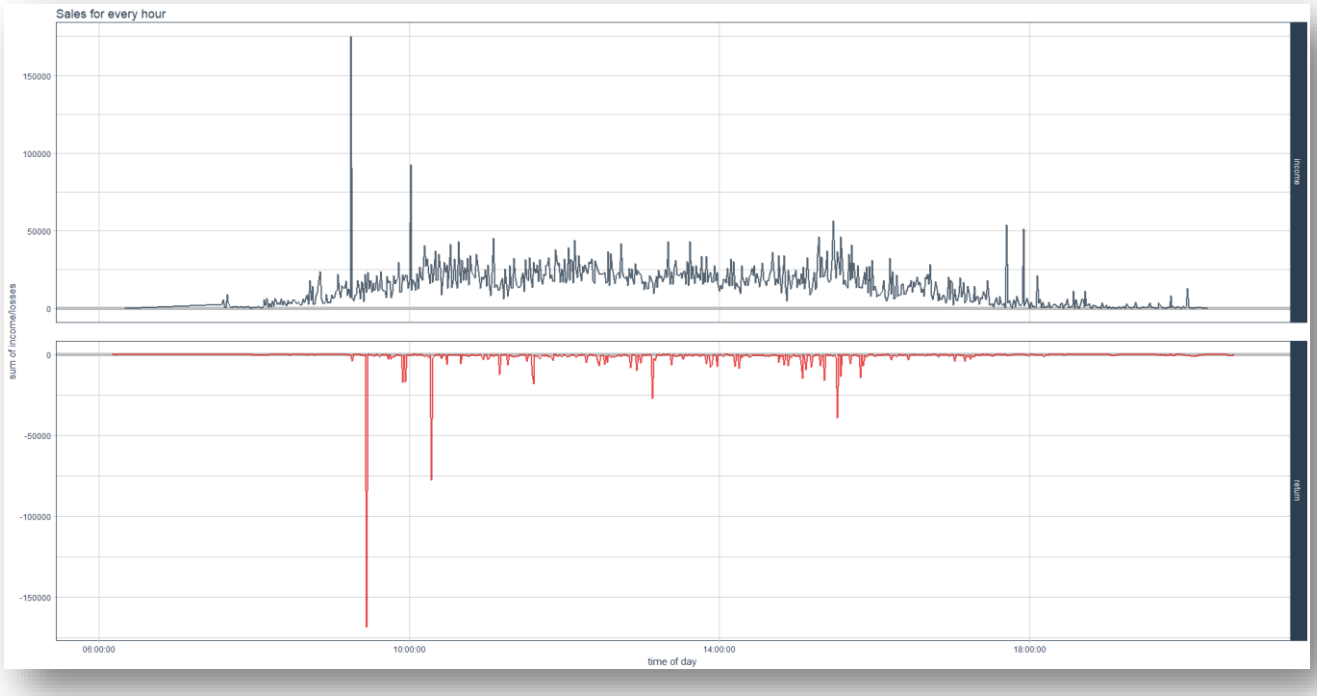
The loss has been stable throughout the dataset but the income has slightly increased for the last month. The only severe outliner is the last day.




Income/loss from transaction per day

Sales per hour from the Online retail store: -

Below graph represents the sales (sum on income and losses) as per the day time of transactions. top most transactions were made during the day time (Business hours).




Sales/hour of the day

Extreme Outliers: -
From the above frequency polygonal it is evident that there exist two extreme outliers in the dataset. According to the dataset its can be concluded that the papercraft birdies transaction might be a blunder as the person who purchased them at 9:15 cancelled the order after 12 minutes and did not order any product after that.

```
  InvoiceNo StockCode Description                      Quantity InvoiceDate         UnitPrice CustomerID Country        day        day_of_week time   month   income income_return
  <chr>     <chr>     <chr>                                 <int> <dttm>                  <dbl>      <int> <chr>          <date>     <ord>       <time> <chr>   <dbl> <chr>
1 581483    23843     PAPER CRAFT , LITTLE BIRDIE           80995 2011-12-09 09:15:00     2.08      16446 United Kingdom 2011-12-09 Fri         09:15  12    168470. income
2 581476    16008     SMALL FOLDING SCISSOR(POINTED EDGE)     240 2011-12-09 08:48:00     0.120     12433 Norway         2011-12-09 Fri         08:48  12        28.8 income
3 581476    22693     GROW A FLYTRAP OR SUNFLOWER IN TIN      192 2011-12-09 08:48:00     1.06      12433 Norway         2011-12-09 Fri         08:48  12       204.  income
> retail %>% filter(day == "2011-12-09") %>% arrange(quantity) %>% .[1:3, ]
# A tibble: 3 x 14
  InvoiceNo StockCode Description                    Quantity InvoiceDate         UnitPrice CustomerID Country        day        day_of_week time   month     income income_return
  <chr>     <chr>     <chr>                               <int> <dttm>                  <dbl>      <int> <chr>          <date>     <ord>       <time> <chr>     <dbl> <chr>
1 C581484   23843     PAPER CRAFT , LITTLE BIRDIE       -80995 2011-12-09 09:27:00     2.08      16446 United Kingdom 2011-12-09 Fri         09:27  12    -168470.   return
2 C581490   22178     VICTORIAN GLASS HANGING T-LIGHT       -12 2011-12-09 09:57:00     1.95      14397 United Kingdom 2011-12-09 Fri         09:57  12        -23.4  return
3 C581490   23144     ZINC T-LIGHT HOLDER STARS SMALL       -11 2011-12-09 09:57:00     0.830     14397 United Kingdom 2011-12-09 Fri         09:57  12         -9.13 return
> retail %>% filter(CustomerID == 16446)
# A tibble: 4 x 14
  InvoiceNo StockCode Description                    Quantity InvoiceDate         UnitPrice CustomerID Country        day        day_of_week time   month     income income_return
  <chr>     <chr>     <chr>                               <int> <dttm>                  <dbl>      <int> <chr>          <date>     <ord>       <time> <chr>     <dbl> <chr>
1 553573    22980     PANTRY SCRUBBING BRUSH                  1 2011-05-18 09:52:00     1.65      16446 United Kingdom 2011-05-18 Wed         09:52  05          1.65 income
2 553573    22982     PANTRY PASTRY BRUSH                     1 2011-05-18 09:52:00     1.25      16446 United Kingdom 2011-05-18 Wed         09:52  05          1.25 income
3 581483    23843     PAPER CRAFT , LITTLE BIRDIE         80995 2011-12-09 09:15:00     2.08      16446 United Kingdom 2011-12-09 Fri         09:15  12     168470.   income
4 C581484   23843     PAPER CRAFT , LITTLE BIRDIE        -80995 2011-12-09 09:27:00     2.08      16446 United Kingdom 2011-12-09 Fri         09:27  12    -168470.   return
>
```

Transaction for Customer ID 16446 (Two Extreme Outliers)

The Customer with Customer ID 16446 purchased papercraft birdies at 9:15 and cancelled the same order after 12 minutes at 9:27.
We can also calculate the Items that are returned or purchased using the R summarize function. There are items that are bought on regular basis and majority of products are being shopped occasionally.

| | StockCode | Description | sum |
|---|---|---|---|
| | <chr> | <chr> | <int> |
| 1 | 84077 | WORLD WAR 2 GLIDERS ASSTD DESIGNS | 53847 |
| 2 | 85099B | JUMBO BAG RED RETROSPOT | 47363 |
| 3 | 84879 | ASSORTED COLOUR BIRD ORNAMENT | 36381 |
| 4 | 22197 | POPCORN HOLDER | 36334 |
| 5 | 21212 | PACK OF 72 RETROSPOT CAKE CASES | 36039 |
| 6 | 85123A | WHITE HANGING HEART T-LIGHT HOLDER | 35025 |
| 7 | 23084 | RABBIT NIGHT LIGHT | 30680 |
| 8 | 22492 | MINI PAINT SET VINTAGE | 26437 |
| 9 | 22616 | PACK OF 12 LONDON TISSUES | 26315 |
| 10 | 21977 | PACK OF 60 PINK PAISLEY CAKE CASES | 24753 |

# ... with 5,739 more rows

> 

Net quantity of each item

We have also calculated the Most purchased item and the sum of days, that item has being bought.

| | StockCode | Description | n |
|---|---|---|---|
| | <chr> | <chr> | <int> |
| 1 | 85123A | WHITE HANGING HEART T-LIGHT HOLDER | 304 |
| 2 | 85099B | JUMBO BAG RED RETROSPOT | 302 |
| 3 | 22423 | REGENCY CAKESTAND 3 TIER | 301 |
| 4 | 84879 | ASSORTED COLOUR BIRD ORNAMENT | 300 |
| 5 | 20725 | LUNCH BAG RED RETROSPOT | 299 |
| 6 | 21212 | PACK OF 72 RETROSPOT CAKE CASES | 299 |

> 

Product bought every day

Preparing Data for Modelling by Day: -

There are various ways to get the visualize output data but for now I am sure I have all the data that was required to prepare a dataset for modelling. As we have dataset for only one year we might not get a precise forecast or model time dependent trends. **By using these trends, we can construct a table of aspects per day. Forecasting techniques and information technology that convert the unpredictable into the predictable are an essential and integral part of a production system (Chatterjee & Samuelson, 2014).**

Repeated Customer: -

The Customers who had made multiple transactions during the year are acknowledged as repeated customers. From the above data we can figure out the repetitive customers who made transactions every day. There are 2992 customers who are purchasing the products everyday for the given dataset.

consumer and day per item, quantities and transactions: -

We have calculated the following aspects as per day and customers:

number: item transaction per day per consumer

sum_product: total sum of item transaction per day per consumer

sum_income: mean total income per day per consumer

from the above details, I can calculate mean statics per day:

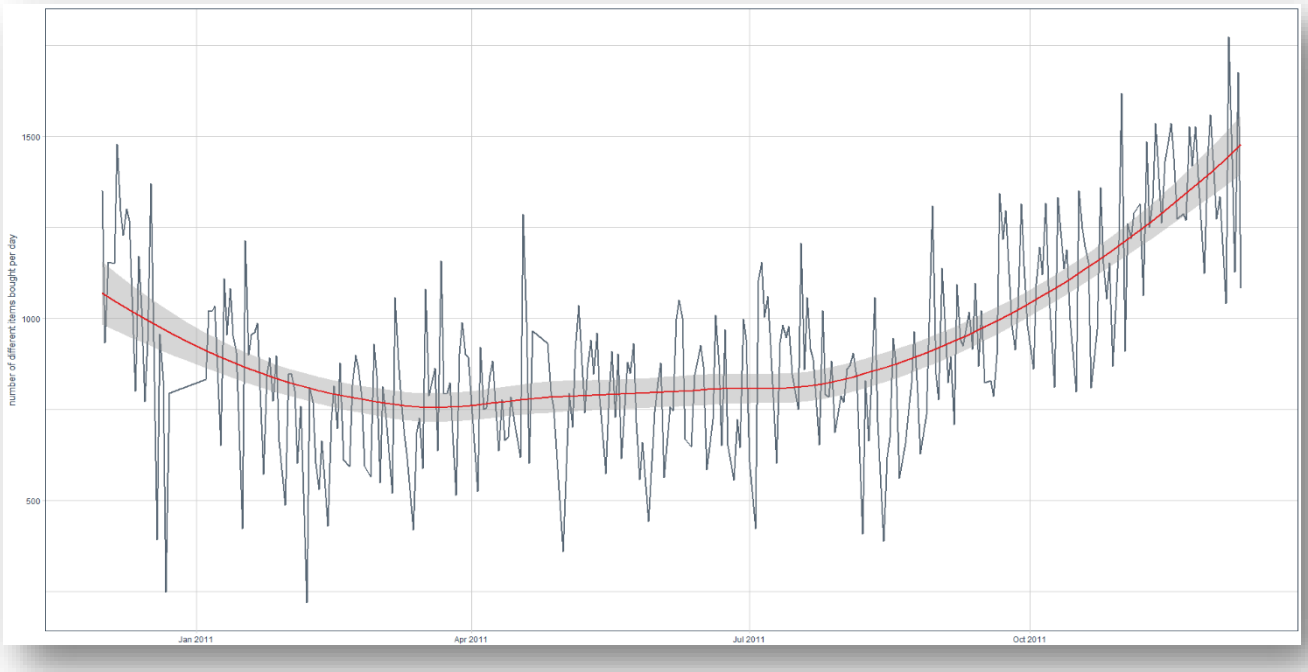
mean_income_customer: total income per day from all consumers

mean_quantity_customer: mean total quantity of items per day for all consumers

mean_product_customer: mean number of product per day from all consumers

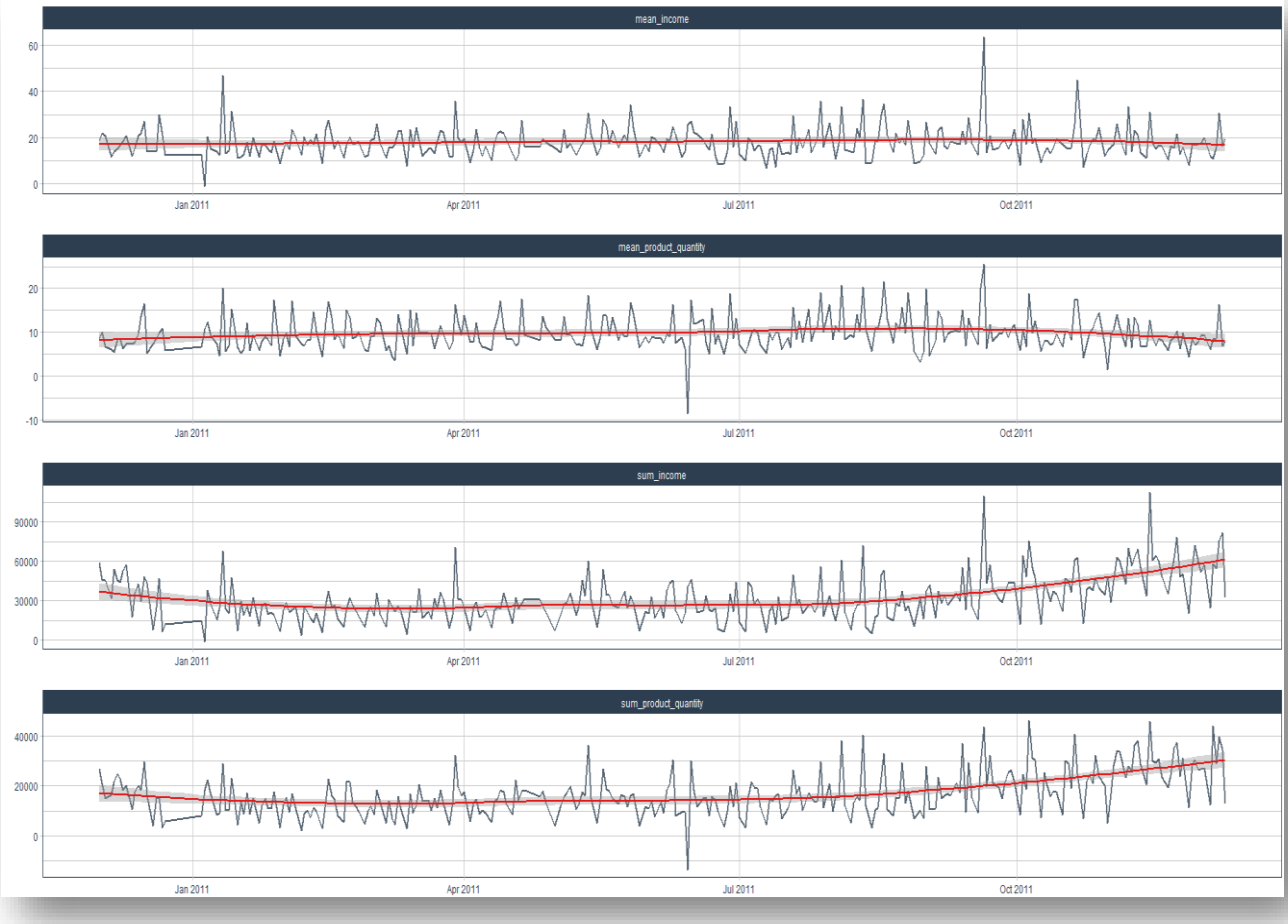

consumer and day per item, quantities and transactions

Transaction of Products per day: -
We have also plotted the transactions for different items. we see a comparable example where the quantity of various things every day expanded amid the most recent two months.



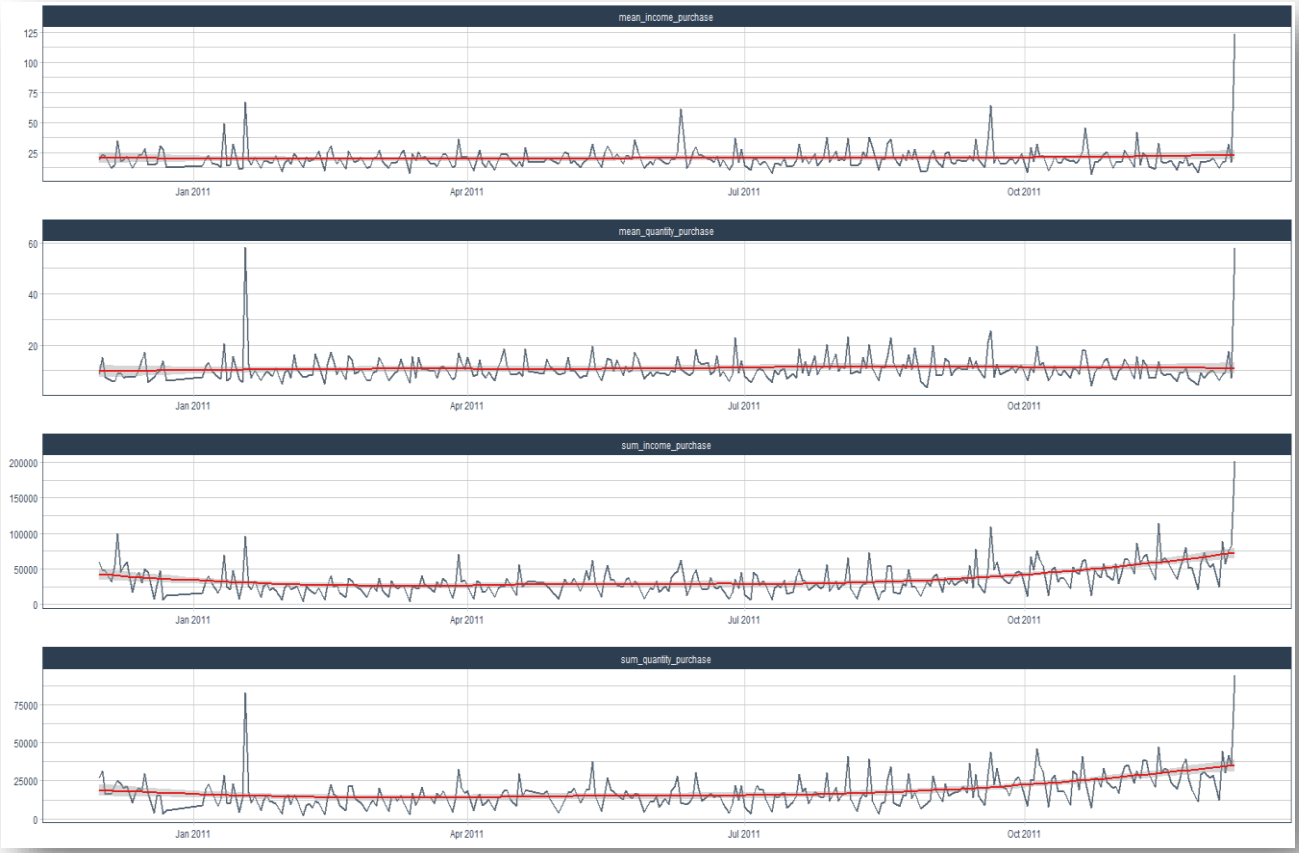

Transaction of Products per day

Total Income and Product Quantities Summarized: -
We will be calculating the sum and mean of total income and the product quantity sold per day.

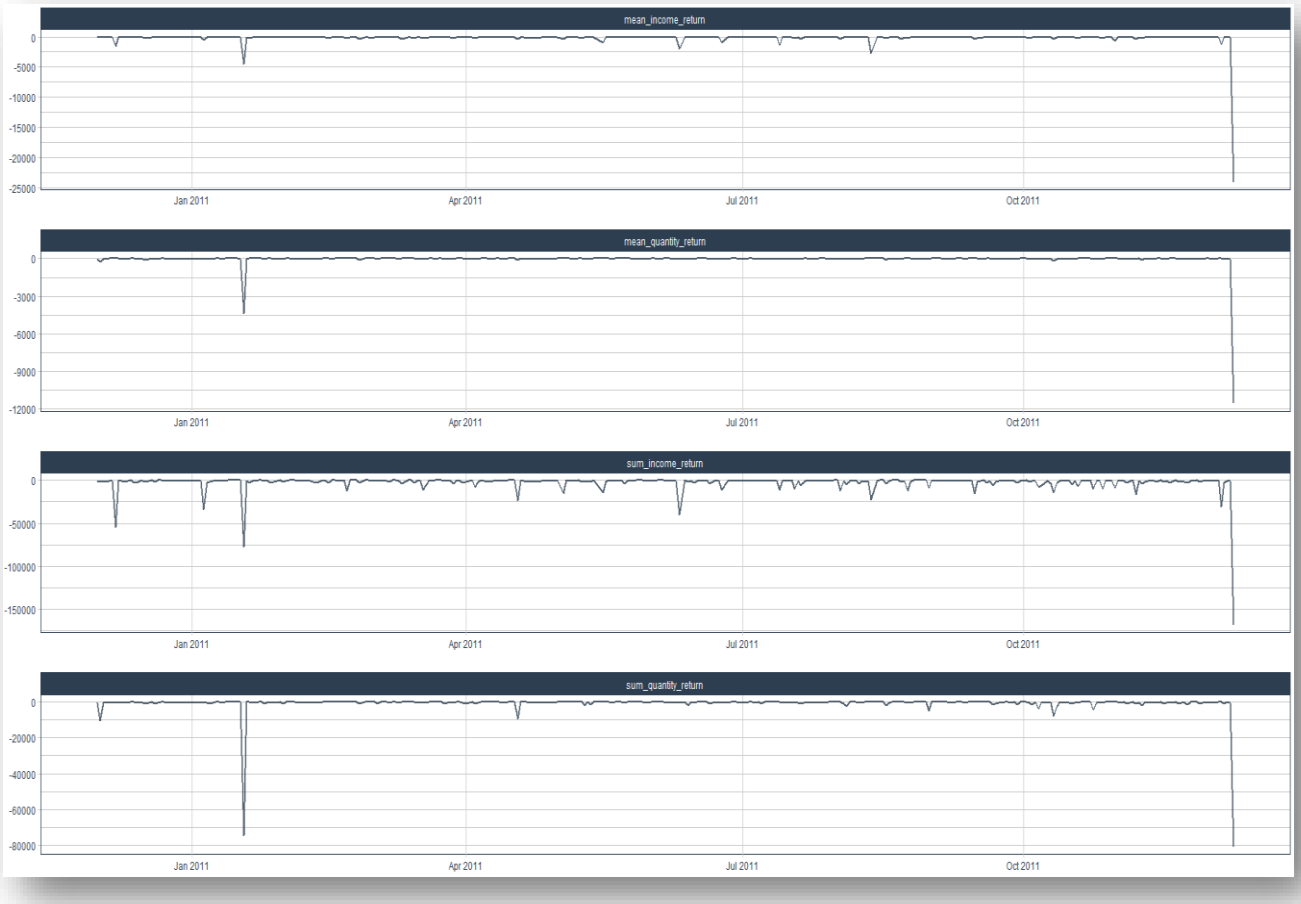



Total Income and Product Quantities Summarized

Income from transactions (purchase/returns):
The same is calculated for the transactions of returns and purchases independently.



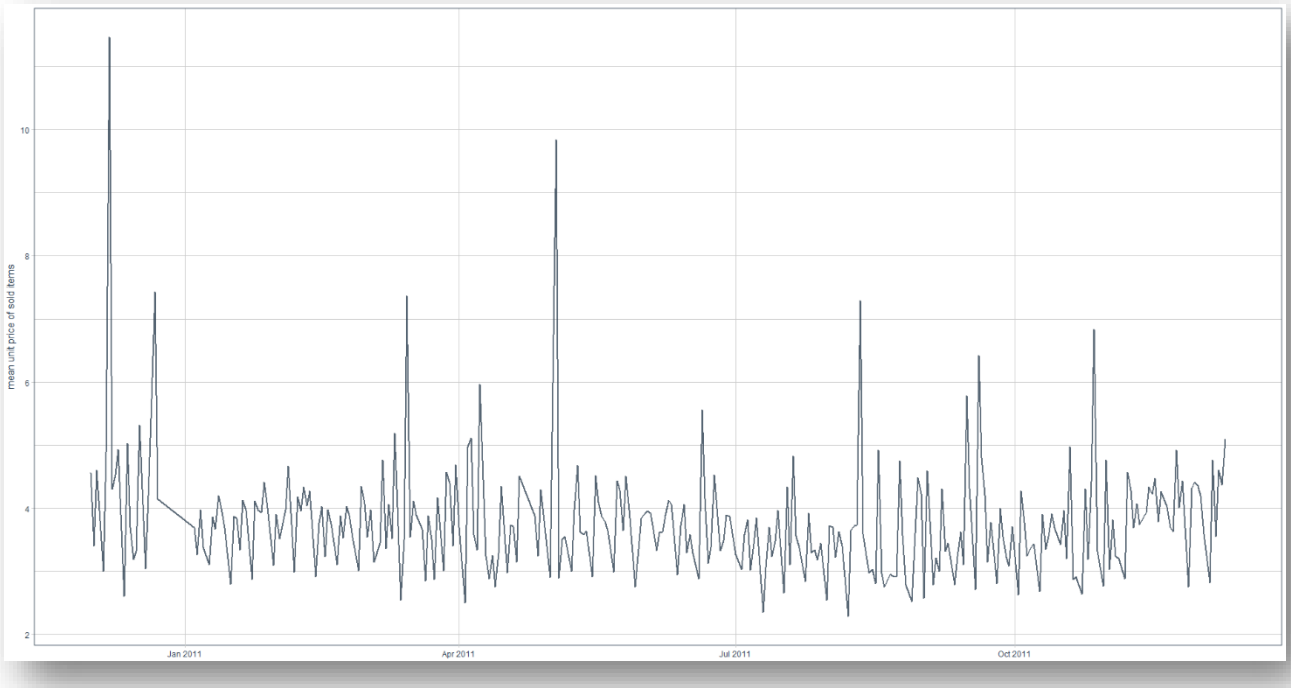

Income from Purchases




Income from Return

Mean Price per day per unit sold: -

we have created a provisional data frame for associate day and stockcode to overcome the product sale multiple times per day and unit price of same product changed between purchase/return and days. We will combine this temporary table with a different table of day and items per day transactions. Following which I can calculate the mean unit price of an item and mean unit price of all item per day.




Mean Price per day per unit sold

Combining Data: -

Now, we will be combining all the data that we have got into a single data frame. I will be joining all the outputs from the previous methods and will be creating a new column names as season. We will be using a lag() function as it can take negative combination of values to work with the series of data. Using the lag() function we can determine the change between the total sum of income with the previous days.

Taken altogether of the above outputs we are having all the data to start with final visualization and that will help us to describe patterns and possible response variables which will be suitable for modelling. We will be adding more points that are colored by weekdays so we as to inspect potential biases in the days of week.

From the above observations we can anticipate that there is a trivial amount of increase in product sales during the months of November and December. we don't have information from extra years to judge whether that needs to do with Christmas transactions or whether it is denoting a general increment in deals that will proceed into January and February. Furthermore, undoubtedly, more number of transactions are made during Monday to Friday as compared to Sunday (no transactions on Saturday recorded).

**Forecasting using facebook's Prophet package: (https://research.fb.com/prophet-forecasting-at-scale/)**

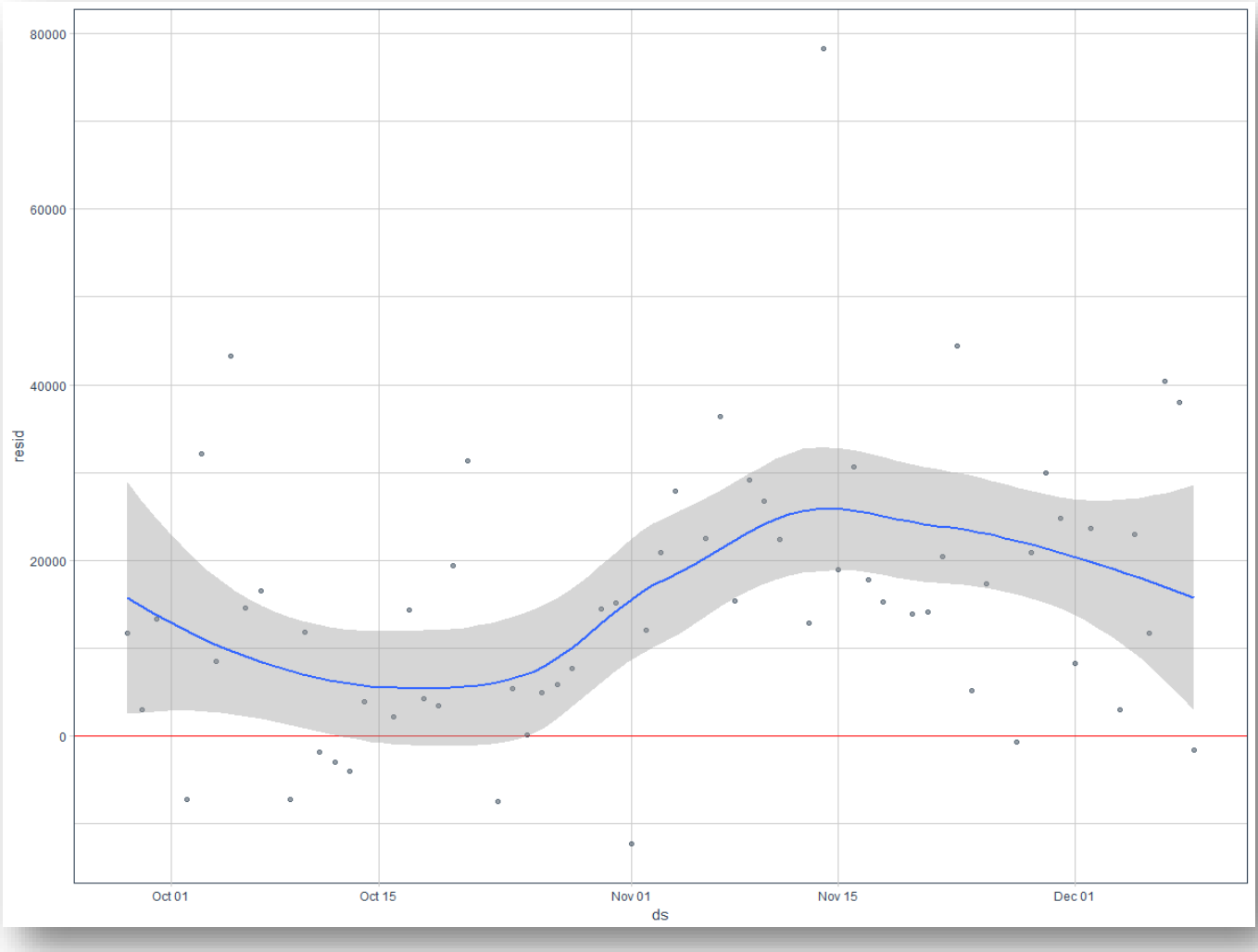
Different conclusions and algorithms can be used to predict the futuristic events such as sales and transactions. We will be using the facebook Prophet package for forecasting this dataset due to the following advantages: -

- Prophet produces an unequivocal, justifiable and accurate forecast for the set of Inputs.
- Every forecast package consists of different techniques such as smoothing, ARIMA, exponential, etc. each technique of forecast is having its own tuning parameters strength and weakness. An Incorrect technique of forecasting can produce wrong results.

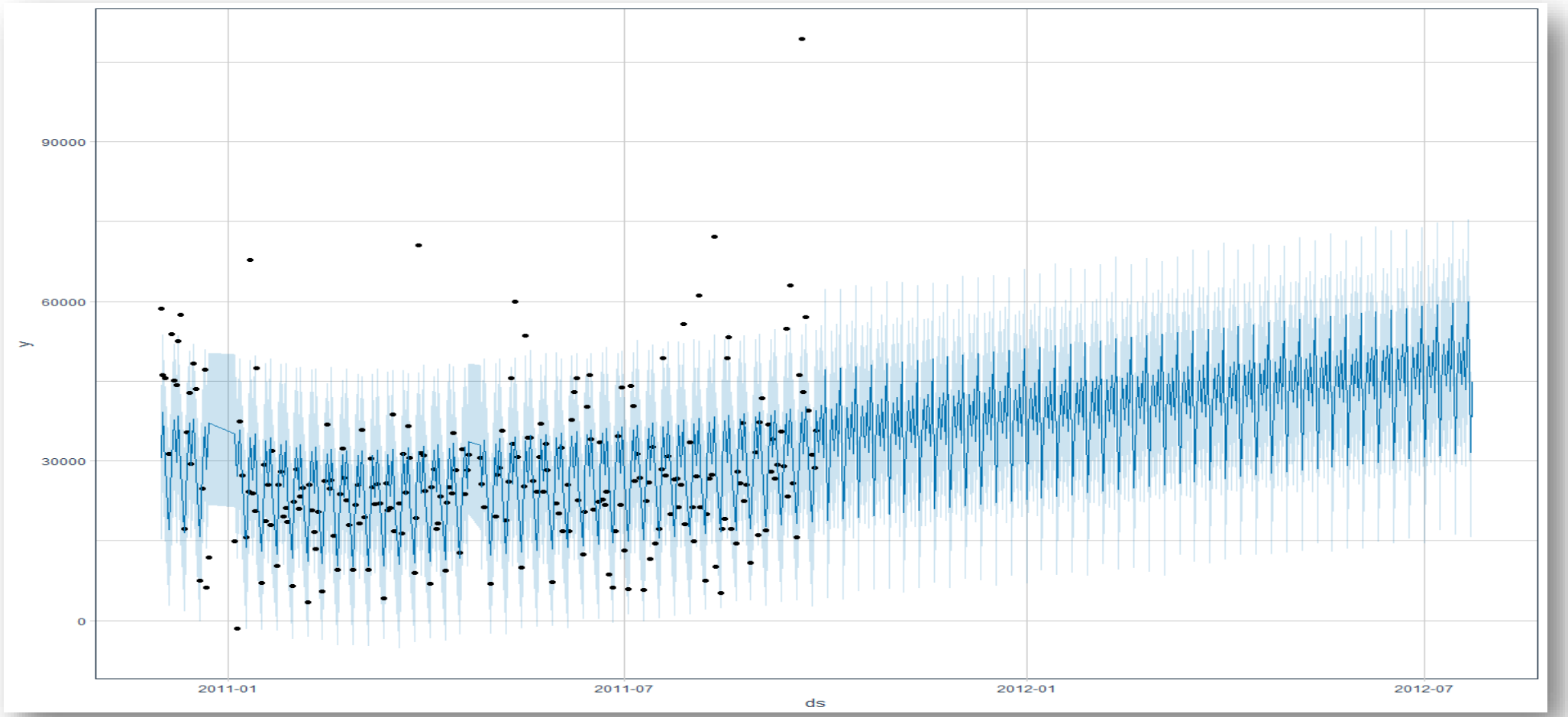
- The prophet forecasting technique is customizable and is easy to understand even by a beginner in machine learning.
- We can specify the dates and holidays like black Friday, Christmas, thanksgiving and many more.
- For growth curves you can manually define “capacities” of an incremental curve. This allow us to input our own inputs data/information about how our model grows or decline.
- Prophet works on daily, hourly or weekly observations.
- A legitimate number of missing data or Information and vast range of outliers.

Prophet generate an Additive Regression Design with the following main peripherals
Prophet selects the change points automatically within the data and recognize variations in the trend.
This particular method uses Fourier series to model yearlong seasonal composition.
Weekly seasonal component using dummy variables.
A user provided list of important holidays.

Creating Training and Test Dataset: -
We will be separating the dataset into training and test dataset.
Train dataset: - 01/12/2010 – 27/09/2011 => 300 days (80%)
Test dataset: - 27/09/2011 – 09/12/2011 => 74 days (20%)



Predicted model: -



**Future Sales Prediction: -**

From the above graph it is evident that the sale starts to increase in the future. By using the data from the online retail, we are able to study the different layer of seasonality and were able to regulate the missing data. During the end of year there was a considerable amount of increase in sales but as the data is insufficient we are not able to judge whether it was due to the Christmas and new year holidays or something else. There is no data present for every Saturday so that means the store is not operational. The store is operational from Sunday to Friday and the sales are high on every Thursday. Between 9 am to 12 noon the sales are at peak.

## Critical Summary of Methodical Approach:

In this method of approach, we are going to apply K mean Clustering method to the dataset of Online retail. We are going to separate out many different things as follows: -

1. How frequent the customers bought products from the store?
2. How recent the product was bought?
3. What was the amount spend by the customer on the products?
4. The online store can figure out such customers and can provide good services for the marketing.
5. The store will be able to classify the customers who are frequent buyers and can provide them with good offers for more profit.
6. Store will be able to gain customer loyalty compared to those who buy product from the store on rare occasions and spend less or are not frequent customers.

For a Marketing analyst there are only 3 aspects that can improve the sales, those are as follows: -

1. Recency: date on which the customer bought a recent product.
2. Frequency: the number of times the customer made the purchase.
3. Money: the money spent by the customer on the transactions.

Depending on the above 3 characteristics we can distribute the classes of the customers.

Recency is the most critical aspect because the more often a client buys a product the expectancy rate goes high for revisit.

## K Mean Clustering: -

**Unsupervised learning algorithms process the input data without any prior labels or training to come up with patterns and relationships (Bali & Sarkar, 2016).** K-means clustering is one of the method of unsupervised learning used to characterize the data which is without categories, classes or groups. K-means algorithm assign each and every datapoint to a single K group based on the similarity aspects of the data. **The basic concepts on which the social exchange theory relies are people's costs, rewards, relationship outcomes and equity perceptions with regard to the relationship they engage in (Nacif, 2003).**
K-mean clustering gives the following output: -

1. The Centroid, used to label new data
2. Classify the data to a single cluster

## Exploratory Data Analysis: -

We will be creating different visualizations for different features of our data set so as to decide the variables used in final dataset. These data visualized model will help us to assess the final data modelling and features.

In this analysis, the function read.csv() enables us to read the csv file in the comma separated format and with the help of 'stringsAsFactors = FALSE' we avoid complications such as re-encoding of strings.

We have used the packages such as

Readr: - readr provides the fastest way to read csv tsv and fwf like files into R studio.

Dplyr: - allows data frame objects to be read in and out of memory.

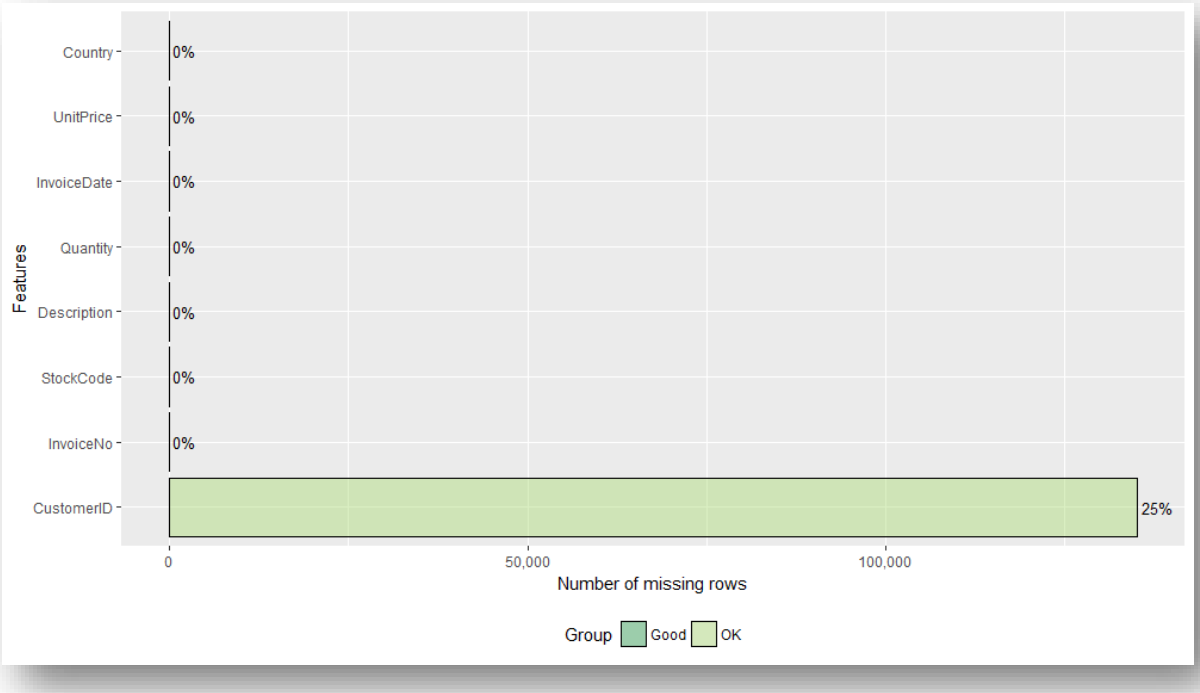
Ggplot2: - Grammar of Graphics reads the data from the R studio and allow us to sketch variables to aesthetics and explains the details with graphical information.

DataExplorer: - this package allows the user to analyze the data and build the required models for better understanding and data extraction. This package automatically examines the variables of the data and does data distribution. **The ability to detect the uniform effect of K-means is highlighted as an important criterion for measure selection (Wu, 2014).**

Lubridate: - this package allows us to use functions and convert data into different format of time.

Finding the missing values using the function plot_missing():

Using the plot_missing() function we are able to find that the data is missing 25% of the values. So, we have decided to remove the 135080 values from the dataset so that we can get a complete data frame. We will be working with 406829 which completes the data set which completes the dataset for the Online retail**. A dimensionality reduction technique, such as principal component analysis, can be used to separate groups of patterns in data (Trevino, 2018).**




**Plotting missing values**

we use na.omit() function to remove the missing data from the dataset.

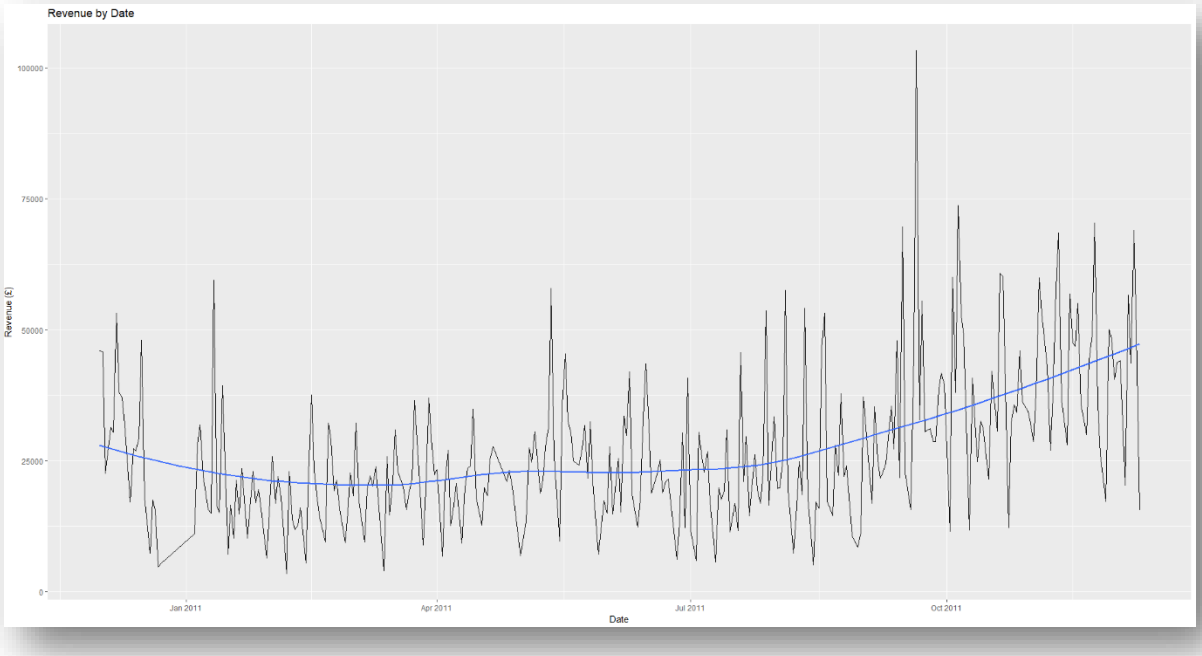
## Separating Date and Time Components of dataset: -

We have used the sapply() function which comes with R basic package. Sapply() function applies to all the elements that are given as inputs in the the form of list, vector or data frame and it gives the output as vector or matrix. Sapply() function is used to perform iteration operation on various components of a dataset.

The sapply() function performs iterative action on the InvoiceDate and create a new variable date which consist of the separate details about the date only. Same function is applied to the InvoiceDate to separate

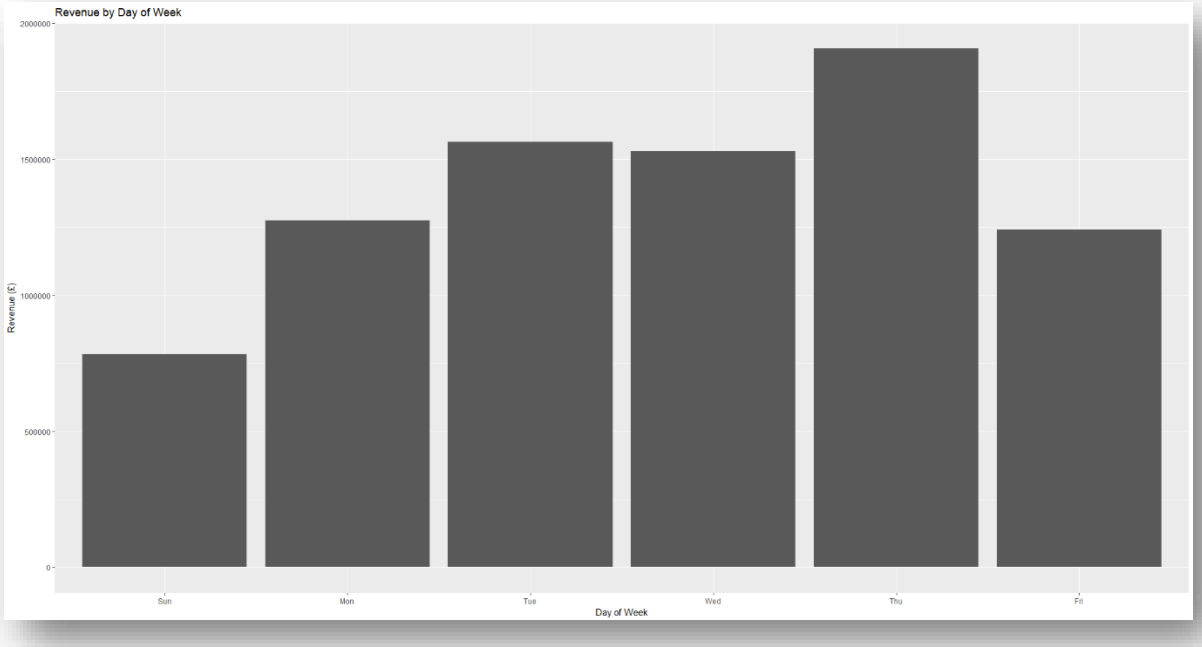
out the hourofday time, month and year components, creating columns individually for each of the above for further process.

Rough Time Series Representation (Date Vs Revenue): -




We have created a time series graph of the date and revenue using the ggplot() function in which the total revenue is represented by sum of all the income per day (Netincome = Quantity * UnitPrice). On the X axis we have represented the date component created from the dataset and on Y axis it's the total revenue.

Revenue by Day of the week: -



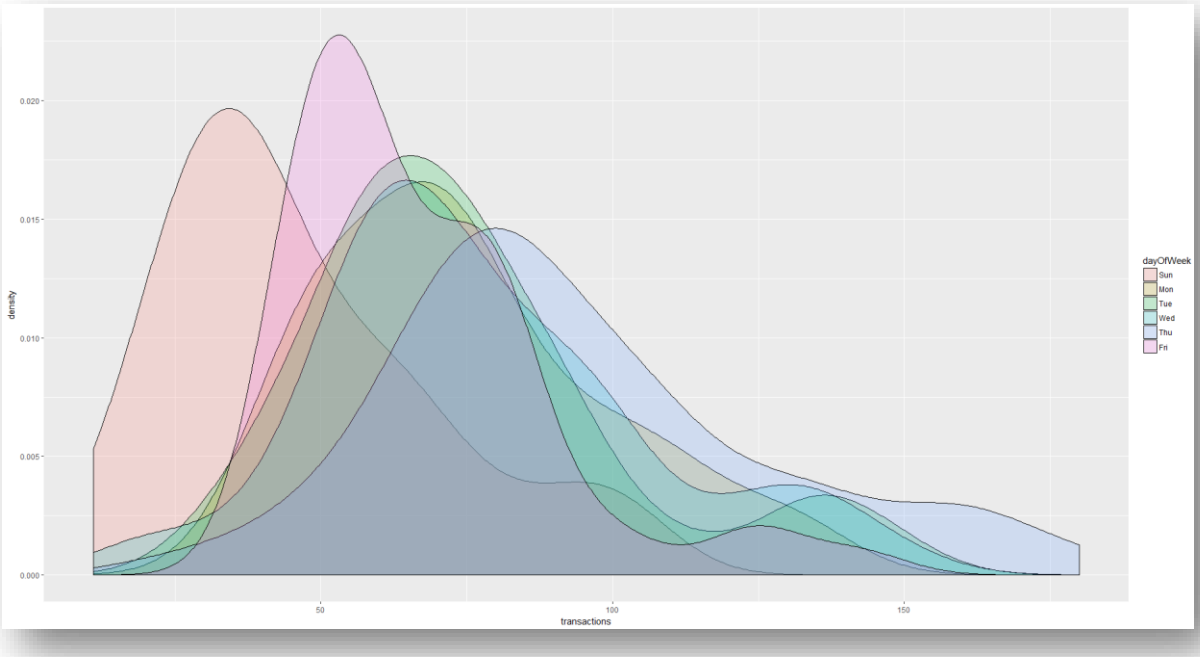
The above graph represents the revenue generated everyday by the online retail store. The online store is closed on Sundays and the highest revenue generated is on Thursdays.

We have also generated the data for everyday transaction from 01-12-2010 to 09-12-2011 which is shown below (only the first five entries of the data)

```
> head(weekdaySummary, n = 10)
# A tibble: 10 x 5
   date       dayOfWeek revenue transactions aveOrdVal
   <date>     <ord>       <dbl>        <int>     <dbl>
 1 2010-12-01 Wed        46051.          127      363.
 2 2010-12-02 Thu        45775.          160      286.
 3 2010-12-03 Fri        22598.           64      353.
 4 2010-12-05 Sun        31381.           94      334.
 5 2010-12-06 Mon        30465.          111      274.
 6 2010-12-07 Tue        53126.           79      672.
 7 2010-12-08 Wed        38049.          134      284.
 8 2010-12-09 Thu        37178.          132      282.
 9 2010-12-10 Fri        32005.           78      410.
10 2010-12-12 Sun        17218.           50      344.
>
```

The above data consist of the everyday transaction with the average value of order and the revenue earned everyday by the store.

Weekday Summary: -

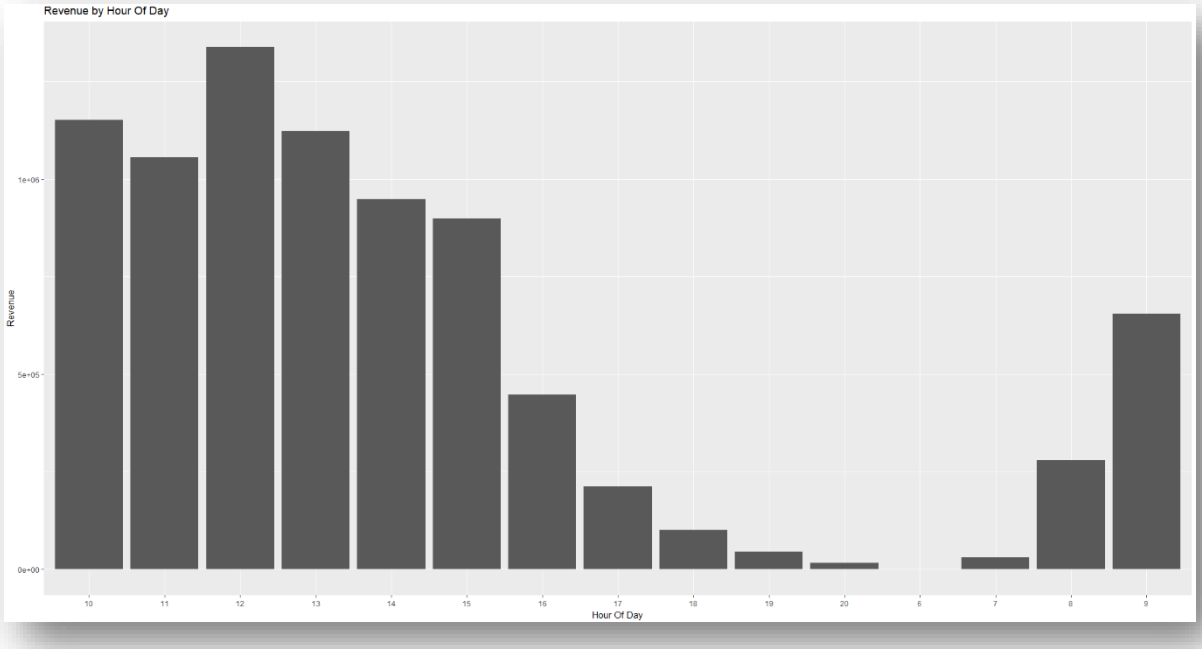



Weekday Summary

The summary of day of the weeks compared with the transactions is plotted above.

Hour of the Day Data: -

Further we need to see what are the hours in which the most revenue was generated by the store for the given time period.




Summary of Hourly Data

From morning 10 am to afternoon 3pm the sales for the company is high and the revenue is higher as compared to other time in the day. The 12$^{th}$ hour of the day is where the highest order are placed for the retail store and the revenue is highly generated at that time.

Country Summary: -

We will be summarizing the data with respect to every country with the revenue generated and the transactions from every country.

```
> head(countrySummary, n = 10)
# A tibble: 10 x 4
   Country        revenue transactions aveOrdVal
   <fct>            <dbl>        <int>     <dbl>
 1 United Kingdom 6767873.        19857      341.
 2 Netherlands     284662.          101     2818.
 3 EIRE            250285.          319      785.
 4 Germany         221698.          603      368.
 5 France          196713.          458      430.
 6 Australia       137077.           69     1987.
 7 Switzerland      55739.           71      785.
 8 Spain            54775.          105      522.
 9 Belgium          40911.          119      344.
10 Sweden           36596.           46      796.
> unique(countrySummary$Country)
 [1] United Kingdom       Netherlands          EIRE                 Germany              France               Australia
 [7] Switzerland          Spain                Belgium              Sweden               Japan                Norway
[13] Portugal             Finland              Channel Islands      Denmark              Italy                Cyprus
[19] Austria              Singapore            Poland               Israel               Greece               Iceland
[25] Canada               Unspecified          Malta                United Arab Emirates USA                  Lebanon
[31] Lithuania            European Community    Brazil               RSA                  Czech Republic       Bahrain
[37] Saudi Arabia
37 Levels: Australia Austria Bahrain Belgium Brazil Canada Channel Islands Cyprus Czech Republic Denmark EIRE ... USA
>
```

Country Summary

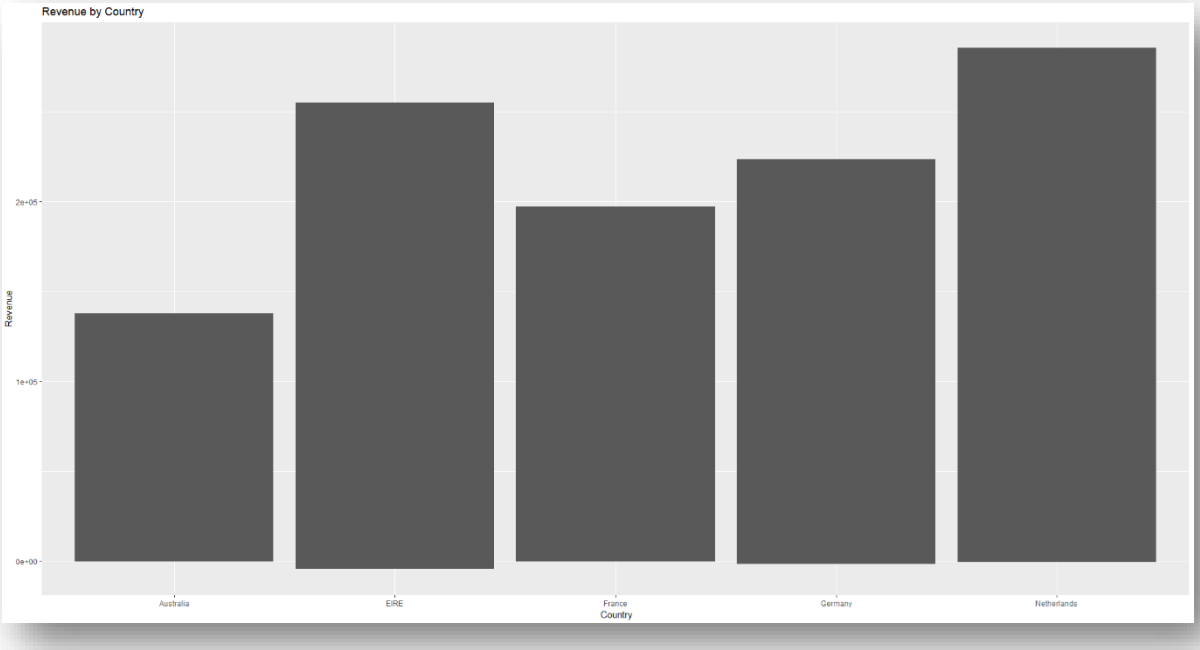
From the data it is evident that United Kingdom tops the list with the overall transactions and the revenue. There are 37 countries worldwide from where the customers are making the transactions with the online store. We will now look into a transaction where apart from United Kingdom we want to know what the top five countries are where the transaction and income is highest and target out the customers.

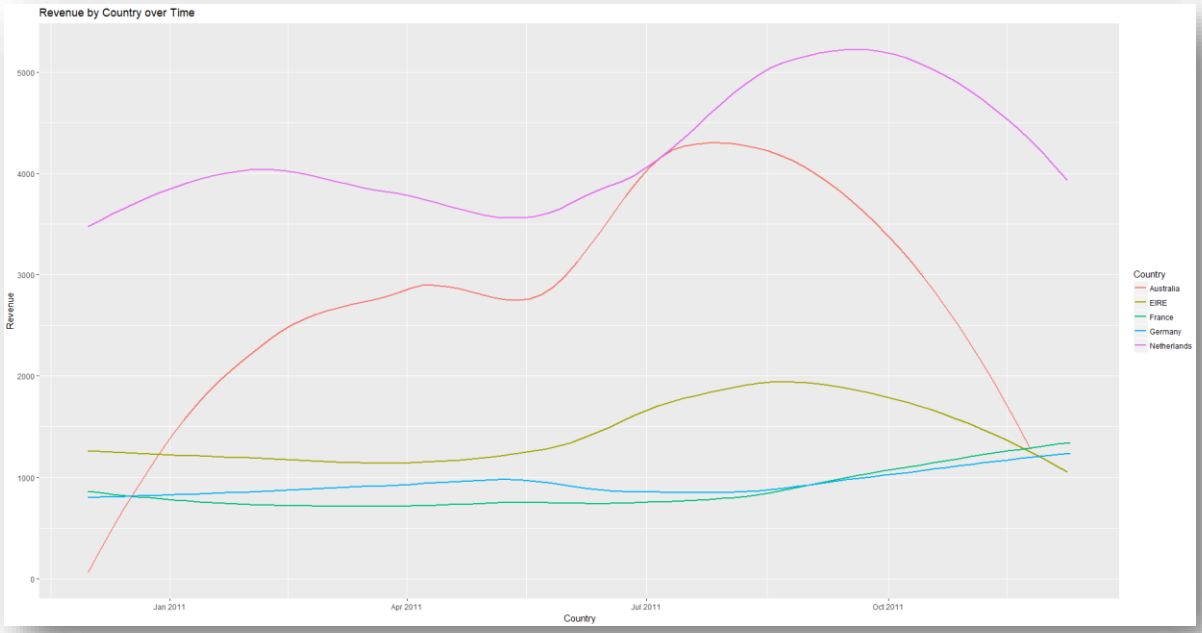
| | Country | date | revenue | transactions | customers | aveOrdVal |
|---|---|---|---|---|---|---|
| | <fct> | <date> | <dbl> | <int> | <int> | <dbl> |
| 1 | Netherlands | 2011-10-20 | 25834. | 3 | 1 | 8611. |
| 2 | Australia | 2011-06-15 | 23427. | 2 | 1 | 11713. |
| 3 | Australia | 2011-08-18 | 21880. | 1 | 1 | 21880. |
| 4 | Netherlands | 2011-08-11 | 19151. | 1 | 1 | 19151. |
| 5 | Netherlands | 2011-02-21 | 18279. | 2 | 1 | 9140. |
| 6 | Netherlands | 2011-03-29 | 18248. | 2 | 1 | 9124. |
| 7 | EIRE | 2011-01-14 | 16775. | 1 | 1 | 16775. |
| 8 | Australia | 2011-03-03 | 16558. | 2 | 1 | 8279. |
| 9 | Netherlands | 2011-05-12 | 16478. | 3 | 1 | 5493. |
| 10 | Australia | 2011-10-05 | 16472. | 2 | 1 | 8236. |

# ... with 729 more rows

Top 5 Countries

The above data results into the top five countries as from country summary we found that the five countries, i.e., Netherland, EIRE, Germany, France and Australia are the most indulging countries for higher transactions with the online retail store.






Top 5 countries revenue over time

Customer Summary: -

Now we will take a look at the data for customer ID who have the highest number of revenue and the number of transactions by those customers with respect to the online retail dataset.

```
# A tibble: 10 x 4
   CustomerID revenue transactions aveOrdVal
        <int>   <dbl>        <int>     <dbl>
 1      14646 279489.           77     3630.
 2      18102 256438.           62     4136.
 3      17450 187482.           55     3409.
 4      14911 132573.          248      535.
 5      12415 123725.           26     4759.
 6      14156 113384.           66     1718.
 7      17511  88125.           46     1916.
 8      16684  65892.           31     2126.
 9      13694  62653.           60     1044.
10      15311  59419.          118      504.
> |
```

Customer Summary

Cancelled transactions: -

There are few of the customers who have bought the product and later have cancelled the order. We have pointed out those customers which are marked by “C” in the invoice number. Along with that, one customer order some products worth 80995 maybe it’s a mistake but later the customer have never purchased as per the history of data of customer ID 16446. The transaction 581483 was done on 09-12-2011 at 9:15 am and twelve minutes later the transaction was cancelled with invoice number C581484 at 9:27 am.

```
# A tibble: 10 x 6
   CustomerID InvoiceNo  revenue transactions aveOrdVal   cumsum
        <int> <chr>        <dbl>        <int>     <dbl>    <dbl>
 1      16446 C581484   -168470.            1  -168470. -168470.
 2      12346 C541433    -77184.            1   -77184. -245653.
 3      15098 C556445    -38970.            1   -38970. -284623.
 4      15749 C550456    -22998.            1   -22998. -307622.
 5      16029 C570556    -11817.            1   -11817. -319438.
 6      12536 C573079     -8322.            1    -8322. -327760.
 7      16029 C551685     -8143.            1    -8143. -335903.
 8      16029 C551699     -6930.            1    -6930. -342833.
 9      12744 C571750     -6068.            1    -6068. -348901.
10      14911 C562375     -4345.            1    -4345. -353246.
>
```

Cancelled order

```
> customerdata %>% filter(CustomerID == 16446)
  InvoiceNo StockCode                Description Quantity    InvoiceDate UnitPrice CustomerID        Country       date time month year hourOfDay dayOfWeek  lineTotal
1    553573     22980      PANTRY SCRUBBING BRUSH        1 5/18/2011 9:52      1.65      16446 United Kingdom 2011-05-18 9:52     5 2011         9       Wed       1.65
2    553573     22982        PANTRY PASTRY BRUSH        1 5/18/2011 9:52      1.25      16446 United Kingdom 2011-05-18 9:52     5 2011         9       Wed       1.25
3    581483     23843 PAPER CRAFT , LITTLE BIRDIE    80995 12/9/2011 9:15      2.08      16446 United Kingdom 2011-12-09 9:15    12 2011         9       Fri  168469.60
4  C581484     23843 PAPER CRAFT , LITTLE BIRDIE   -80995 12/9/2011 9:27      2.08      16446 United Kingdom 2011-12-09 9:27    12 2011         9       Fri -168469.60
>
```

Customer ID 16446

Recency Monitoring: -

We have to calculate the recency rate of the customer. So, there is a function Sys.Date() under the package lubridate which take the present date of the computer as input and from the system date we will subtract the date from the online retail dataset to find out about the last transaction made by each customer. Using the Sapply() function we will be able to achieve our task of creating a new column for recent days. Basically sapply() function work as a repetitive loop so we have to write one line of code which will allow us to create a new column of data with the recency rate.

```
# A tibble: 5 x 11
  InvoiceNo CustomerID Country        date       month year  hourOfDay dayOfWeek orderVal recent recentDays
  <chr>          <int> <fct>          <date>     <fct> <fct> <fct>     <ord>        <dbl> <chr>        <int>
1 536365         17850 United Kingdom 2010-12-01 12    2010  8         Wed          139.  2726          2726
2 536366         17850 United Kingdom 2010-12-01 12    2010  8         Wed           22.2 2726          2726
3 536367         13047 United Kingdom 2010-12-01 12    2010  8         Wed          279.  2726          2726
4 536368         13047 United Kingdom 2010-12-01 12    2010  8         Wed           70.1 2726          2726
5 536369         13047 United Kingdom 2010-12-01 12    2010  8         Wed           17.8 2726          2726
>
```

Recency Rate

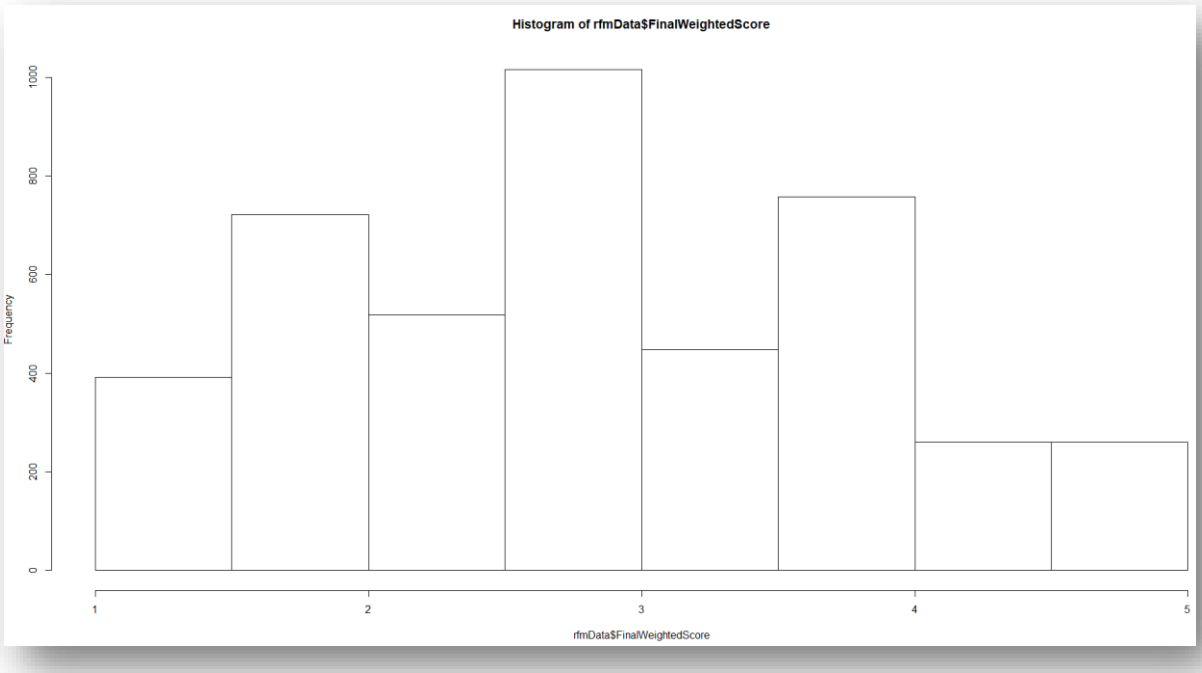
**Applying K Mean Clustering to the Dataset: -**

We are using the package didrooRFM which allow us to calculate the recency frequency and monetary values for the customer dataset. Under didrooRFM, there is a function called findRFM score which allow us to rate the data on the scale of 1-5 for the transaction data. The input to the function is customerID, transaction number and date of transaction (date format i.e., 01-01-2018) and the net income as the other attributes. The findRFM function will generate a dataframe as recency frequency and monetary values for the marketing and sales information.

From the online retail dataset, we will create a new object which will hold the columns of InvoiceNo, customer ID and InvoiceDate to separate out the required components. We will define one more object for invoice number with the total amount (net income). Later we will merge these objects and will get a new dataset with InvoiceNo CustomerID InvoiceDate and Amount.

We will have to change the date format from 12/01/2010 to 12-01-2010 as the function findRFM only works with the date format 12-01-2010. We will use the function as.date() provided by the lubridate package to convert the date to the required format of findRFM() function. Further, using the findRFM function we will we will find out the recency score, monetary score and frequency score.

Function findRFM() generates a histogram that shows final weighted scores distribution.




Result of FindRFM() function

When we apply the head() function to our final dataset for the column 1 to 4 and that leads to the RFM score. Figure below shows the output.

```
# A tibble: 6 x 4
  CustomerID MeanValue LastTransaction NoTransaction
  <chr>          <dbl> <date>                   <int>
1 12346             0. 2011-01-18                   2
2 12347           616. 2011-12-07                   7
3 12348           449. 2011-09-25                   4
4 12349          1758. 2011-11-21                   1
5 12350           334. 2011-02-02                   1
6 12352           140. 2011-11-03                  11
```

Head(finaldata[ , c(1:4)])

These data will result into the recency, monetary and frequency score.

```
# A tibble: 6 x 5
  CustomerID MonetoryScore FrequencyScore RecencyScore FinalCustomerClass
  <chr>              <dbl>          <dbl>        <dbl> <chr>
1 12346                 1.             2.           1. Class-1
2 12347                 5.             4.           5. Class-4
3 12348                 5.             3.           2. Class-3
4 12349                 5.             1.           4. Class-3
5 12350                 4.             1.           1. Class-2
6 12352                 2.             5.           3. Class-3
```

Head(finaldata[ ,c(1,8:10,16)])

for findRFM() function the class is average score of the Recency, frequency and monetary.

The FinalCustomerClass is representation of the most beneficial (customer ID 12347) to least beneficial (customer ID 12346). We will tabulate the recurrences in each class to visualize the distribution using the table table() function.

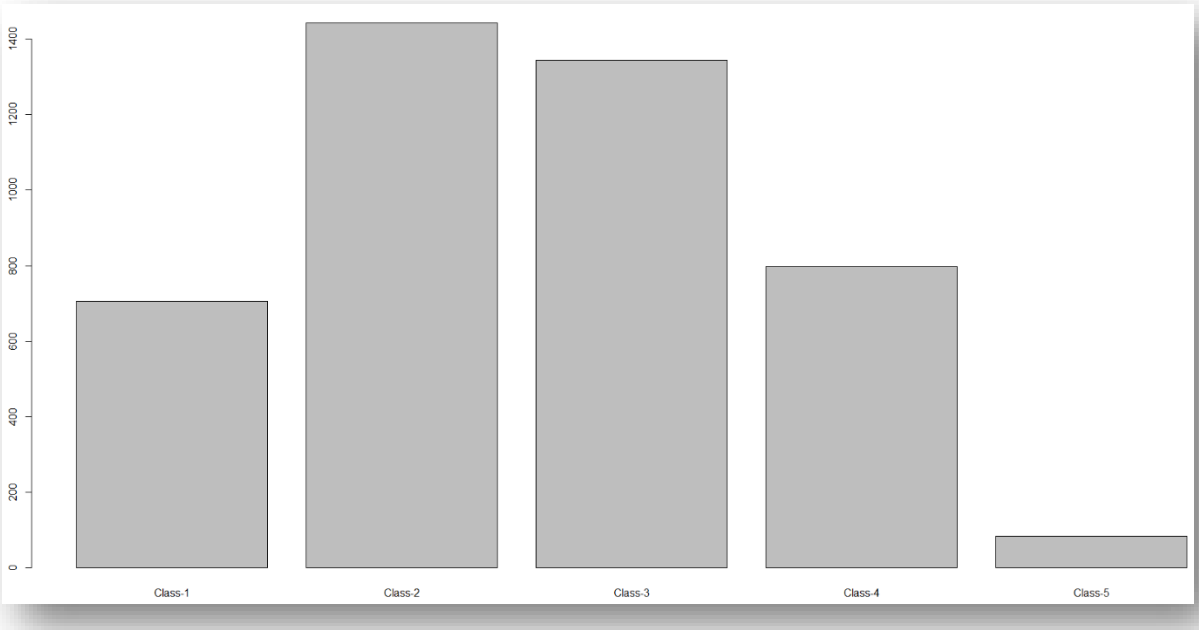
The table data is

```
> tabledata <- table(finaldata$FinalCustomerClass)
> tabledata

Class-1 Class-2 Class-3 Class-4 Class-5
    705    1442    1344     797      84
```

Table data

We will barplot the classes for a clear view of the RFM customer segmentation.




Barplot of the classes

For robust marketing, we will use customer country as the demographic data to see how the clients are distributed in different countries. **Influencers are active in shaping the opinions of others by sharing their experiences online amongst often wide networks of contact groups (Treadgold, 2016).**

Adding the country column from original dataset into the finaldata data set. To achieve this we will use !duplicated() function which will allow us to create a unique customerID with a unique country for merging with the finaldata data frame.

```
> barplot(tabledata)
> custduplicate <- customerdata[!duplicated(customerdata$CustomerID),c(7,8)]
> head(custduplicate)
   CustomerID        Country
1       17850 United Kingdom
10      13047 United Kingdom
27      12583         France
47      13748 United Kingdom
66      15100 United Kingdom
83      15291 United Kingdom
> Countrydata <-merge(finaldata[,c(1,8:10,16)],custduplicate, by="CustomerID")
> colnames(Countrydata) <- c("ID","Monetary","Frequency","Recency","Class","Country")
> head(Countrydata)
     ID Monetary Frequency Recency   Class        Country
1 12346        1         2       1 Class-1 United Kingdom
2 12347        5         4       5 Class-4        Iceland
3 12348        5         3       2 Class-3        Finland
4 12349        5         1       4 Class-3          Italy
5 12350        4         1       1 Class-2         Norway
6 12352        2         5       3 Class-3         Norway
>
```

Merging country to the finaldata

Distribution of country into different classes depending on the findRFM function. From this we were able to decide the customer belonging to different classes as per the recency, frequency and monetary score.

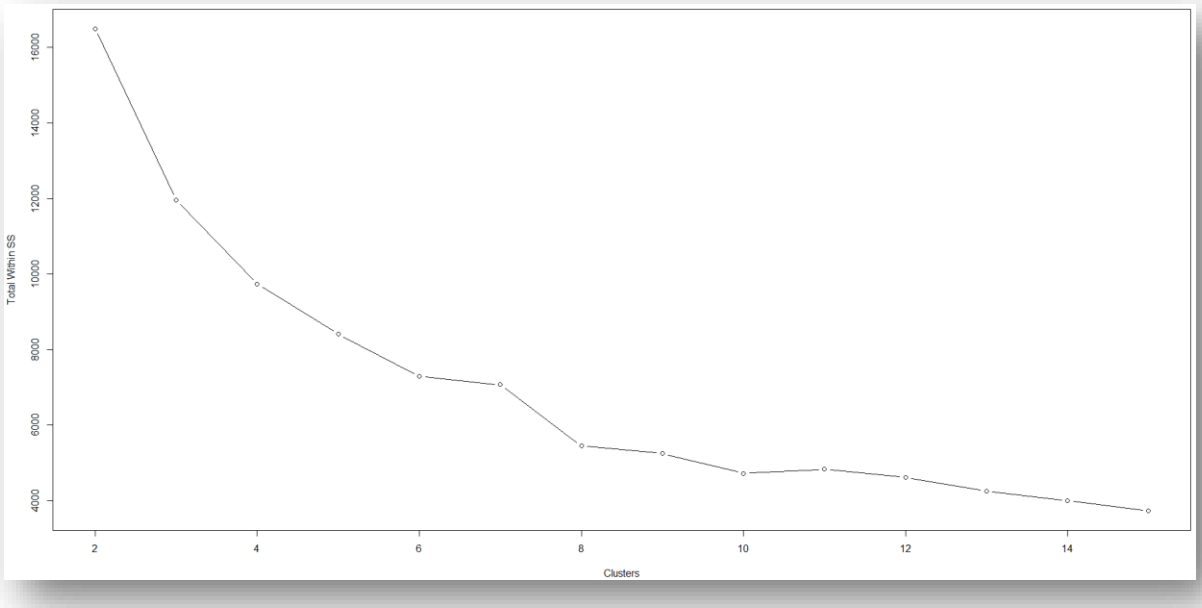
```
> table(Countrydata$Country,Countrydata$class)

                     Class-1 Class-2 Class-3 Class-4 Class-5
  Australia                0       4       2       3       0
  Austria                  1       4       2       2       0
  Bahrain                  1       1       0       0       0
  Belgium                  3       5       8       8       0
  Brazil                   0       1       0       0       0
  Canada                   2       1       1       0       0
  Channel Islands          0       3       4       1       1
  Cyprus                   2       0       4       1       0
  Czech Republic           0       0       1       0       0
  Denmark                  0       3       3       2       0
  EIRE                     0       0       1       0       2
  European Community       0       0       1       0       0
  Finland                  0       4       4       3       1
  France                  12      23      25      25       2
  Germany                  5      32      27      29       2
  Greece                   0       2       2       0       0
  Iceland                  0       0       0       1       0
  Israel                   1       2       1       0       0
  Italy                    1       7       4       3       0
  Japan                    1       3       2       1       1
  Lebanon                  0       1       0       0       0
  Lithuania                0       0       1       0       0
  Malta                    0       1       0       1       0
  Netherlands              2       2       4       0       1
  Norway                   0       4       2       4       0
  Poland                   1       2       2       1       0
  Portugal                 1       6       6       6       0
  RSA                      0       0       1       0       0
  Saudi Arabia             1       0       0       0       0
  Singapore                0       0       0       1       0
  Spain                    5       6      13       5       0
  Sweden                   1       2       4       0       1
  Switzerland              0       7      10       3       0
  United Arab Emirates     0       1       1       0       0
  United Kingdom         665    1309    1207     696      73
  Unspecified              0       3       1       0       0
  USA                      0       3       0       1       0
>
```

table(Countrydata$Country,Countrydata$Class)

**Process for K mean Clustering: -**

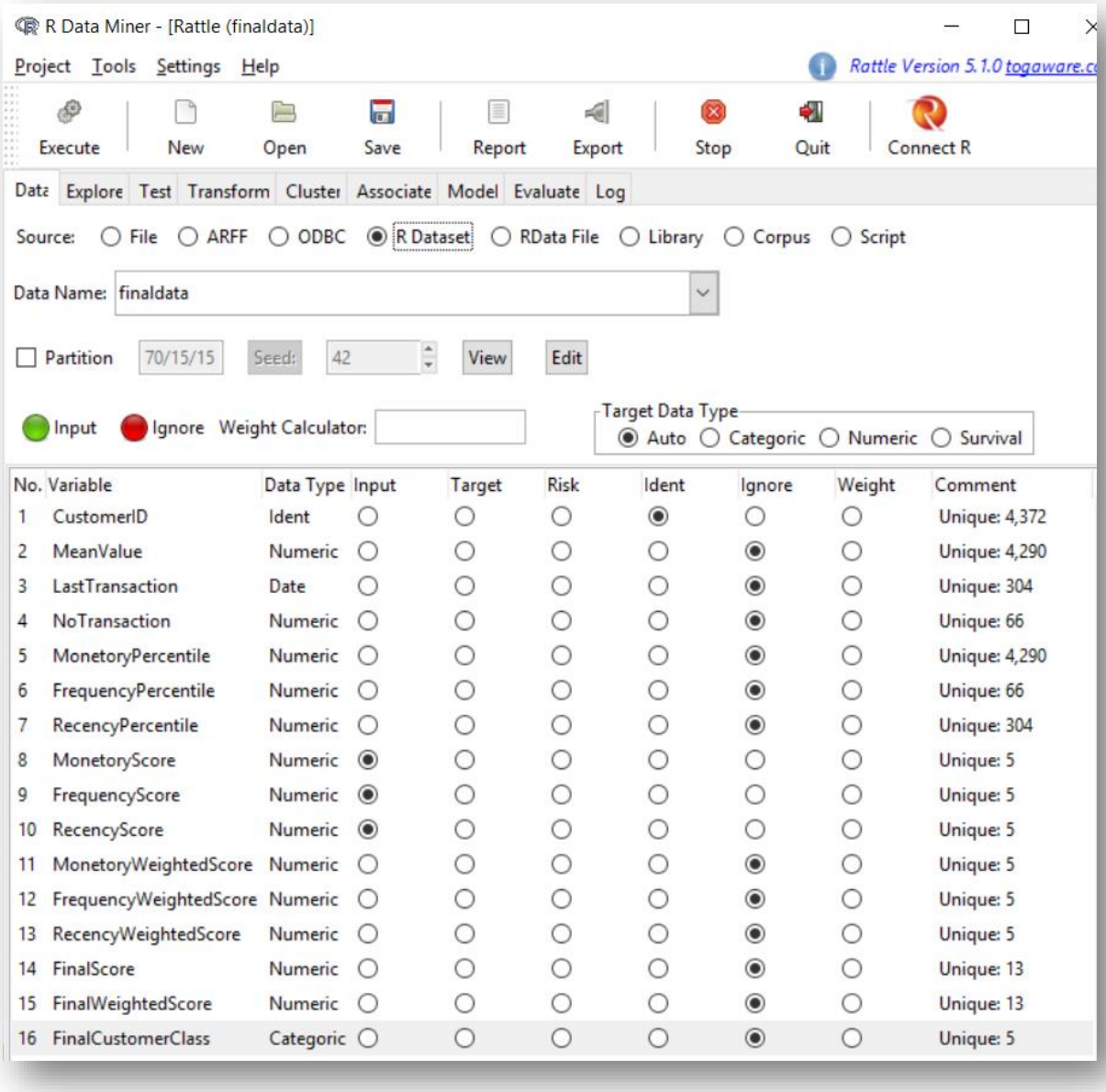
Created a vector as total_with_sums_quare and initial value is given as “NULL”. For loop will iteratively process cluster values from 2-15, and using the function append() store the result in total_with_sums_square. Kmean() function will help us to create the cluster. We will plot the total_with_sums_square vs the cluster number.




Plot total_with_sums_square vs the cluster number

From the above plot total_with_sums_square is somewhat stable after eleven clusters, implying 11 to be a good cluster for the data set.

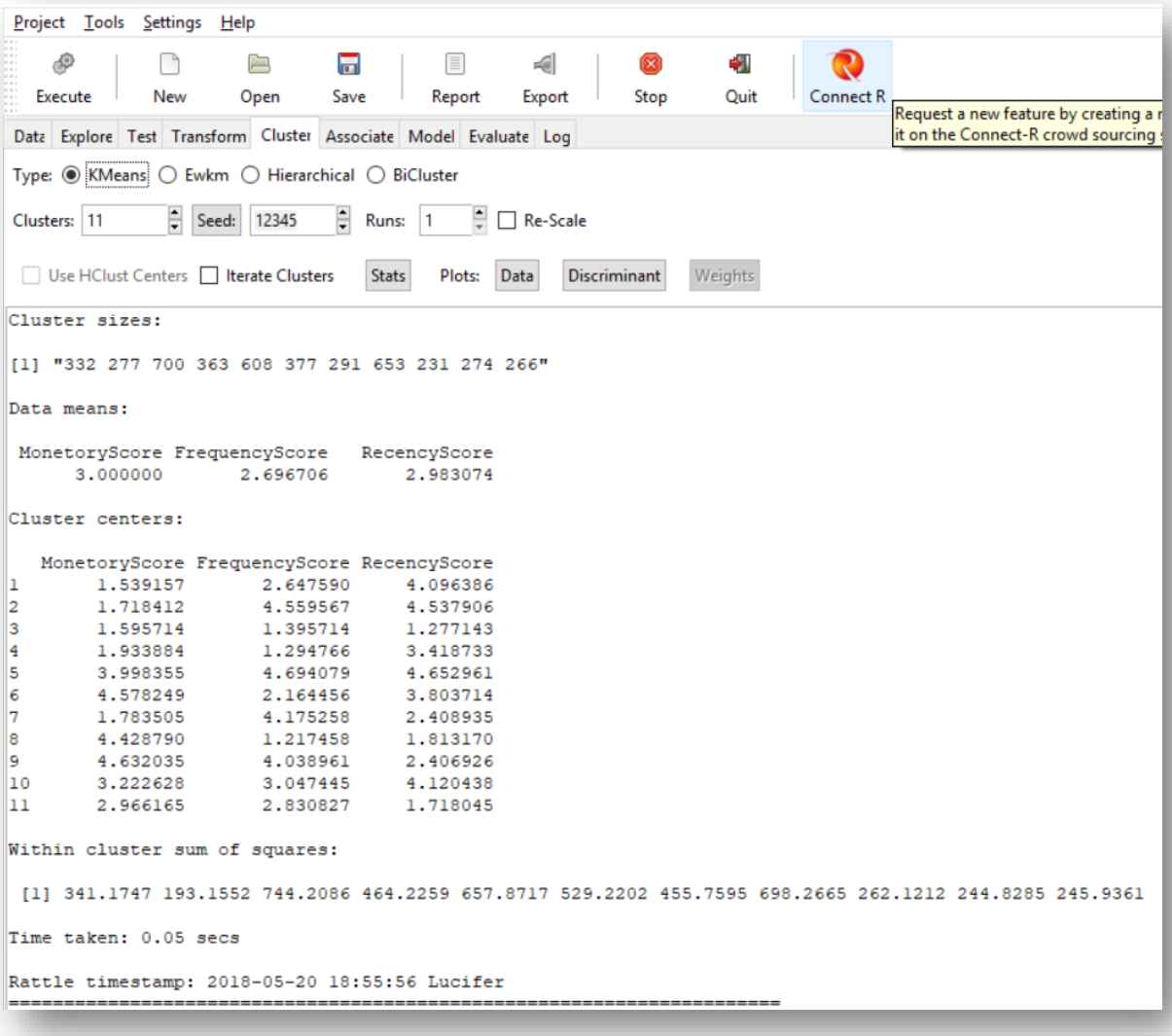
Function Kmeans() have a Graphic user interface which can be used by rattle() function. Using the GUI of rattle() function we will select the data from finaldata dataset. We will select the variables Recency score, Monetary score and the Frequency score from the data tab and will ignore all other variables. Select Customer ID as Ident radiobutton and click execute.




| No. | Variable | Data Type | Input | Target | Risk | Ident | Ignore | Weight | Comment |
|---|---|---|---|---|---|---|---|---|---|
| 1 | CustomerID | Ident | ○ | ○ | ○ | ◉ | ○ | ○ | Unique: 4,372 |
| 2 | MeanValue | Numeric | ○ | ○ | ○ | ○ | ◉ | ○ | Unique: 4,290 |
| 3 | LastTransaction | Date | ○ | ○ | ○ | ○ | ◉ | ○ | Unique: 304 |
| 4 | NoTransaction | Numeric | ○ | ○ | ○ | ○ | ◉ | ○ | Unique: 66 |
| 5 | MonetoryPercentile | Numeric | ○ | ○ | ○ | ○ | ◉ | ○ | Unique: 4,290 |
| 6 | FrequencyPercentile | Numeric | ○ | ○ | ○ | ○ | ◉ | ○ | Unique: 66 |
| 7 | RecencyPercentile | Numeric | ○ | ○ | ○ | ○ | ◉ | ○ | Unique: 304 |
| 8 | MonetoryScore | Numeric | ◉ | ○ | ○ | ○ | ○ | ○ | Unique: 5 |
| 9 | FrequencyScore | Numeric | ◉ | ○ | ○ | ○ | ○ | ○ | Unique: 5 |
| 10 | RecencyScore | Numeric | ◉ | ○ | ○ | ○ | ○ | ○ | Unique: 5 |
| 11 | MonetoryWeightedScore | Numeric | ○ | ○ | ○ | ○ | ◉ | ○ | Unique: 5 |
| 12 | FrequencyWeightedScore | Numeric | ○ | ○ | ○ | ○ | ◉ | ○ | Unique: 5 |
| 13 | RecencyWeightedScore | Numeric | ○ | ○ | ○ | ○ | ◉ | ○ | Unique: 5 |
| 14 | FinalScore | Numeric | ○ | ○ | ○ | ○ | ◉ | ○ | Unique: 13 |
| 15 | FinalWeightedScore | Numeric | ○ | ○ | ○ | ○ | ◉ | ○ | Unique: 13 |
| 16 | FinalCustomerClass | Categoric | ○ | ○ | ○ | ○ | ◉ | ○ | Unique: 5 |

Rattle GUI

Now under cluster tab with k Means selected we set the seed and the number of clusters as 11 which best suits to our data and then we click execute to create the below output.




Selecting Cluster and execution

The result is stored in CRS$KMeans variable which can be found in the log tab. The Centroid table can be represented as Recency, Frequency and monetary value of every cluster. We will convert the table crs$kmeans$centers into a dataframe. Before doing this, we will convert convert table to matrix and then to a data frame using the below command

normalisedcluster <- as.data.frame.matrix(round(crs$kmeans$centers))

| | MonetoryScore | FrequencyScore | RecencyScore |
|---|---|---|---|
| 1 | 1 | 3 | 4 |
| 2 | 2 | 3 | 2 |
| 3 | 1 | 1 | 3 |
| 4 | 3 | 4 | 5 |
| 5 | 1 | 1 | 1 |
| 6 | 3 | 2 | 4 |
| 7 | 4 | 1 | 1 |
| 8 | 5 | 1 | 3 |
| 9 | 3 | 5 | 3 |
| 10 | 5 | 3 | 3 |
| 11 | 4 | 4 | 5 |

Normalized cluster

For marketing and sales purpose, the customer who is recent will more likely comeback which will increase his frequency count and that will increase the monetary value. So as per the data the recency score of a customer is most important for the purpose of more sales and marketing of products.

We will order the function by a single command as mentioned below: -

with(normalisedcluster, normalisedcluster[order(-RecencyScore,-FrequencyScore,-MonetoryScore),])

the "-" sign mean to arrange the data in descending format.

| | MonetoryScore | FrequencyScore | RecencyScore |
|---|---|---|---|
| 11 | 4 | 4 | 5 |
| 4 | 3 | 4 | 5 |
| 1 | 1 | 3 | 4 |
| 6 | 3 | 2 | 4 |
| 9 | 3 | 5 | 3 |
| 10 | 5 | 3 | 3 |
| 8 | 5 | 1 | 3 |
| 3 | 1 | 1 | 3 |
| 2 | 2 | 3 | 2 |
| 7 | 4 | 1 | 1 |
| 5 | 1 | 1 | 1 |

With() function output

It is evident that the most beneficial clients are in the cluster 11 and the least beneficial customer are in cluster 5. The clients from cluster 11 are the most recent and frequent consumers and are at the 2 most level on the spending (monetary value). Likewise, every cluster can be ranked accordingly.

We will now convert this data to table to understand how many clients belong to which cluster and class.

| | Class | | | | |
|---|---|---|---|---|---|
| Cluster | Class-1 | Class-2 | Class-3 | Class-4 | Class-5 |
| 1 | 0 | 184 | 201 | 0 | 0 |
| 2 | 41 | 256 | 20 | 0 | 0 |
| 3 | 159 | 119 | 0 | 0 | 0 |
| 4 | 0 | 0 | 153 | 201 | 0 |
| 5 | 412 | 40 | 0 | 0 | 0 |
| 6 | 0 | 228 | 181 | 0 | 0 |
| 7 | 93 | 474 | 0 | 0 | 0 |
| 8 | 0 | 141 | 300 | 22 | 0 |
| 9 | 0 | 0 | 194 | 88 | 0 |
| 10 | 0 | 0 | 295 | 104 | 0 |
| 11 | 0 | 0 | 0 | 382 | 84 |

K mean Clustering output for the dataset

The class 5 clients who are recent and frequent and are on the top second of spending (monetary value) are in cluster 11. Whereas, the Cluster 5 is at the lowest rank entirely consisting of class 1 & 2 Clients.

**CITATIONS: -**


- Anon, Available CRAN Packages By Name. *README*. Available at: https://cran.r-project.org/web/packages/available_packages_by_name.html [Accessed May 20, 2018].
- Bali, R. & Sarkar, D., 2016. *R machine learning by example: understand the fundamentals of machine learning with R and build your own dynamic algorithms to tackle complicated real-world problems successfully*, Birmingham: Packt Publishing.
- Box, G.E.P., Jenkins, G.M. & Reinsel, G.C., 2015. *Time series analysis: forecasting and control*, New Jersey: John Wiley & Sons, Inc.
- Chatterjee, K. & Samuelson, W., 2014. *Game theory and business applications*, New York: Springer.
- Chatterjee, K. & Samuelson, W., 2014. *Game theory and business applications*, New York: Springer.
- Dannecker, L., 2015. *Energy time series forecasting efficient and accurate forecasting of evolving time series from the energy domain*, Wiesbaden: Springer Vieweg.
- Daróczi Gergely, 2013. *Introduction to R for quantitative finance: solve a diverse range of problems with R, one of the most powerful tools for quantitative finance*, Birmingham, UK: Packt Publishing.
- KONAR, A.M.I.T., 2017. *TIME-SERIES PREDICTION AND APPLICATIONS*, S.l.: SPRINGER INTERNATIONAL PU.
- Kramer, O., 2016. *Machine learning for evolution strategies*, Switzerland: Springer.
- Lewis, R. & Dart, M., 2014. *The new rules of retail: competing in the world's toughest marketplace*, New York: Palgrave Macmillan.
- Lindström Martin, 2010. *Buy·ology: truth and lies about why we buy*, New York: Broadway Books.
- Nacif, R.C., 2003. *Online customer loyalty: forecasting the repatronage behavior of online retail customers*, Wiesbaden: Dt. Univ.-Verl.
- Treadgold, A.D. & Reynolds, J., 2016. *Navigating the new retail landscape: a guide for business leaders*, Oxford: Oxford University Press.

- Trevino, A., Introduction to K-means Clustering. *DataScience.com*. Available at: https://www.datascience.com/blog/k-means-clustering [Accessed May 20, 2018].
- Wu, J., 2014. *Advances in k-means clustering: a data mining thinking*, Place of publication not identified: Springer.
- Zhao, Y., 2013. *R and data mining: examples and case studies*, Amsterdam: Academic Press, an imprint of Elsevier.