# PH125.9x - Capstone: Online Retail Project

Mohammed Al-Areqi

2023-12-01

## Contents

**1 Introduction** **2**

**2 Overview** **2**

**3 Data Structure** **3**

**4 Data Cleaning** **4**

**5 Analysis** **5**

5.1 Product Analysis . . . . . . . . 6

5.2 Country Analysis . . . . . . . . 8

5.3 Time Analysis . . . . . . . . 10

5.4 Customer Analysis . . . . . . . . 11

**6 RFM Analysis** **13**

**7 K-means Cluster** **14**

**8 Results** **19**

**9 Conclusion** **23**

**10 References** **23**

## 1 Introduction

Customer segmentation is a process in which customers are grouped with specific characteristics so company's sales, marking and services are focused on specific customer groups. This project will be focused on an online non-store retail business that is based in UK. The company sells gifts for occasions.

## 2 Overview

The data set available is a year worth of transaction data that occurred between `01-Dec-2010` and `09-Dec-2011`. The data was available in an excel format at UC Irvine Machine learning Repository under the name **Online Retail** (https://archive.ics.uci.edu/dataset/352/online+retail).

In the beginning, we will clean the data set. Then, we will analyze it. In this project we will be using two models. The first model is the RFM analysis and the second model is K-means clustering algorithm.

The RFM analysis is used to study data based on customer behaviors which helps in predicting future customers. RFM is short for the following:

1- Recency: how recent was a transaction made by a customer. In our case, it is the amount of days from the customer's last purchase.

2- Frequency: how frequent transactions were made by the customer. In our case, how many independent purchases were made in different times. Example, how many invoices were made by the customer.

3- Monetary: how much money the customer spent buying products. In our case, the sum of money the customer spent in buying products from the company.

This analysis helps in segmenting customers so you can build a strategy on how to deal with each segment. Example of segments; customers that shop frequently and make big purchases can be the best. While customers who shop less frequently and have small purchases are the least favorite.

K-means clustering model is a machine learning technique in which similar data or observations are clustered together to form a `K` cluster. The optimal number of clusters `K` can be found by some methods like elbow, silhouette, etc.

Since this is an unsupervised learning model and we will be using clustering, then we will not separate the data into training and testing sets. Because there is no final outcome that we can compare the data to. Also,we might be removing important *centroid* point when running K-means model after splitting the data which will increase our model error.

The data will be extracted as below.

```r
# Online Retail data set
#Extract the data from an excel format at UC Irvine Machine learning Repository below
# https://archive.ics.uci.edu/dataset/352/online+retail
# https://archive.ics.uci.edu/static/public/352/online+retail.zip

url <- "https://archive.ics.uci.edu/static/public/352/online+retail.zip"
temp_d <- tempdir()
temp_f <- tempfile(tmpdir = temp_d, fileext = ".zip")
if(!file.exists(temp_f))
  download.file(url, temp_f)

fname <- unzip(temp_f, list = TRUE)$Name[1]
if(!file.exists(fname))
  unzip(temp_f, files = fname, exdir = temp_d, overwrite = TRUE)
```

```r
fpath <- file.path(temp_d, fname)
retail <- read_excel(fpath)
```

## 3 Data Structure

Let us have a look at the structure of the data by executing the following codes.

```r
# A quick display of the data
head(retail)
```

```
## # A tibble: 6 x 8
##   InvoiceNo StockCode Description         Quantity InvoiceDate         UnitPrice
##   <chr>     <chr>     <chr>                  <dbl> <dttm>                  <dbl>
## 1 536365    85123A    WHITE HANGING HEAR~        6 2010-12-01 08:26:00      2.55
## 2 536365    71053     WHITE METAL LANTERN        6 2010-12-01 08:26:00      3.39
## 3 536365    84406B    CREAM CUPID HEARTS~        8 2010-12-01 08:26:00      2.75
## 4 536365    84029G    KNITTED UNION FLAG~        6 2010-12-01 08:26:00      3.39
## 5 536365    84029E    RED WOOLLY HOTTIE ~        6 2010-12-01 08:26:00      3.39
## 6 536365    22752     SET 7 BABUSHKA NES~        2 2010-12-01 08:26:00      7.65
## # i 2 more variables: CustomerID <dbl>, Country <chr>
```

```r
# Summary and structure of the data
str(retail, vec.len = 2)
```

```
## tibble [541,909 x 8] (S3: tbl_df/tbl/data.frame)
##  $ InvoiceNo  : chr [1:541909] "536365" "536365" ...
##  $ StockCode  : chr [1:541909] "85123A" "71053" ...
##  $ Description: chr [1:541909] "WHITE HANGING HEART T-LIGHT HOLDER" "WHITE METAL LANTERN" ...
##  $ Quantity   : num [1:541909] 6 6 8 6 6 ...
##  $ InvoiceDate: POSIXct[1:541909], format: "2010-12-01 08:26:00" "2010-12-01 08:26:00" ...
##  $ UnitPrice  : num [1:541909] 2.55 3.39 2.75 3.39 3.39 ...
##  $ CustomerID : num [1:541909] 17850 17850 ...
##  $ Country    : chr [1:541909] "United Kingdom" "United Kingdom" ...
```

The data contains 8 columns (InvoiceNo, StockCode, Description, Quantity, InvoiceDate, UnitePrice, CustomerID, and Country) and `541,909` rows. Each row represents a distinct transaction either a part of product purchase by a customer on a specific date and time or a return. One order (Invoice) can be either one row or multiple rows, depending on how many different products are added to the order.

The Attribute information below for each column was provided with the data at https://archive.ics.uci.edu/dataset/352/online+retail:

1- *InvoiceNo:* a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.

2- *StockCode:* a 5-digit integral number uniquely assigned to each distinct product.

3- *Description:* product name.

4- *Quantity:* the quantities of each product (item) per transaction.

5- *InvoiceDate:* the day and time when each transaction was generated.

6- *UnitPrice:* product price per unit.

7- *CustomerID:* a 5-digit integral number uniquely assigned to each customer.

8- *Country:* the name of the country where each customer resides.

## 4 Data Cleaning

Before cleaning the data we need to fix the formatting.

```r
#Convert the data into data frame
retail <- as.data.frame(retail)

#Covert the InvoiceDate to only a Date format
retail$InvoiceDate <- as.Date(retail$InvoiceDate)
```

We will start by checking if there are any records that are N/A in each column.

```r
#Check for NA cells
colSums(is.na(retail)) %>% kable(format = "pipe")
```

| | x |
|---|---|
| InvoiceNo | 0 |
| StockCode | 0 |
| Description | 1454 |
| Quantity | 0 |
| InvoiceDate | 0 |
| UnitPrice | 0 |
| CustomerID | 135080 |
| Country | 0 |

It seems that Description has `1,454` empty or N/A records while CustomerID has `135,080` empty or N/A records. Since our model will be focused on the Customers, we need to remove all the unknown customer records. Even though removing 135,080 rows means removing about 25% of the records.

```r
# Remove the rows that contain N/A
retail <- retail %>% drop_na()
```

When removing the unknown customer records, the unknown product description will be removed with it.

```r
# Check for NA cells if removed
colSums(is.na(retail)) %>% kable(format = "pipe")
```

| | x |
|---|---|
| InvoiceNo | 0 |
| StockCode | 0 |
| Description | 0 |
| Quantity | 0 |
| InvoiceDate | 0 |
| UnitPrice | 0 |
| CustomerID | 0 |
| Country | 0 |

Now let us check the data.

```r
# Summary and structure of the data
str(retail, vec.len = 2)
```

```
## 'data.frame':    406829 obs. of  8 variables:
##  $ InvoiceNo  : chr  "536365" "536365" ...
##  $ StockCode  : chr  "85123A" "71053" ...
##  $ Description: chr  "WHITE HANGING HEART T-LIGHT HOLDER" "WHITE METAL LANTERN" ...
##  $ Quantity   : num  6 6 8 6 6 ...
##  $ InvoiceDate: Date, format: "2010-12-01" "2010-12-01" ...
##  $ UnitPrice  : num  2.55 3.39 2.75 3.39 3.39 ...
##  $ CustomerID : num  17850 17850 ...
##  $ Country    : chr  "United Kingdom" "United Kingdom" ...
```

The data now has 406,829 rows which is 25% less after removing the unknown customers.

Now the data is ready for analysis.

## 5 Analysis

Before starting our analysis we want to add a new column which shows that total amount of purchases per row (per product). We can get that by multiplying the UnitPrice by the Quantity.

```r
# Adding the total amount of purchase per row
retail <- retail %>% mutate(Total = Quantity*UnitPrice)
head(retail)
```

```
##   InvoiceNo StockCode                         Description Quantity InvoiceDate
## 1    536365    85123A  WHITE HANGING HEART T-LIGHT HOLDER        6  2010-12-01
## 2    536365     71053                 WHITE METAL LANTERN        6  2010-12-01
## 3    536365    84406B      CREAM CUPID HEARTS COAT HANGER        8  2010-12-01
## 4    536365    84029G KNITTED UNION FLAG HOT WATER BOTTLE        6  2010-12-01
## 5    536365    84029E      RED WOOLLY HOTTIE WHITE HEART.        6  2010-12-01
## 6    536365     22752        SET 7 BABUSHKA NESTING BOXES        2  2010-12-01
##   UnitPrice CustomerID        Country Total
## 1      2.55      17850 United Kingdom 15.30
## 2      3.39      17850 United Kingdom 20.34
## 3      2.75      17850 United Kingdom 22.00
## 4      3.39      17850 United Kingdom 20.34
## 5      3.39      17850 United Kingdom 20.34
## 6      7.65      17850 United Kingdom 15.30
```

First let us check how many unique invoices, products, customers and different countries that data has.

```r
# Number of unique values for InvoiceNo, StockCode, CustomerID, and Country in retail data
retail %>% summarize(Invoices = n_distinct(InvoiceNo),
                StockCodes = n_distinct(StockCode),
                Customers = n_distinct(CustomerID),
                Countries = n_distinct(Country)) %>%
        kable( digits = 4, format = "pipe")
```

| Invoices | StockCodes | Customers | Countries |
|---:|---:|---:|---:|
| 22190 | 3684 | 4372 | 37 |

The table shows that between `01-Dec-2010` and `09-Dec-2011` there are 22,190 different invoices created, 3,684 unique products were purchased by 4,372 customers from 37 countries.

## 5.1 Product Analysis

Let us check the value of the top 10 product per unit.

```r
# Value of the top 10 products per unit
options(digits=3)
retail %>%
  group_by(Description) %>%
  filter (!(Description %in%
            c("Manual", "POSTAGE", "DOTCOM POSTAGE",
              "CRUK Commission", "Discount" ))) %>% #Removing unrelated transactions
  summarize(UnitPrice =  mean(UnitPrice)) %>%
  arrange(desc(UnitPrice)) %>%
  slice(1:10) %>%
  kable(format = "pipe")
```

| Description | UnitPrice |
|:---|---:|
| PICNIC BASKET WICKER 60 PIECES | 649.5 |
| REGENCY MIRROR WITH SHUTTERS | 156.4 |
| RUSTIC SEVENTEEN DRAWER SIDEBOARD | 156.0 |
| VINTAGE RED KITCHEN CABINET | 150.7 |
| VINTAGE BLUE KITCHEN CABINET | 143.7 |
| CHEST NATURAL WOOD 20 DRAWERS | 118.1 |
| LOVE SEAT ANTIQUE WHITE METAL | 115.4 |
| VINTAGE POST OFFICE CABINET | 66.4 |
| SCHOOL DESK AND CHAIR | 64.1 |
| DECORATIVE HANGING SHELVING UNIT | 60.0 |

Also check the 10 cheapest products per unit.

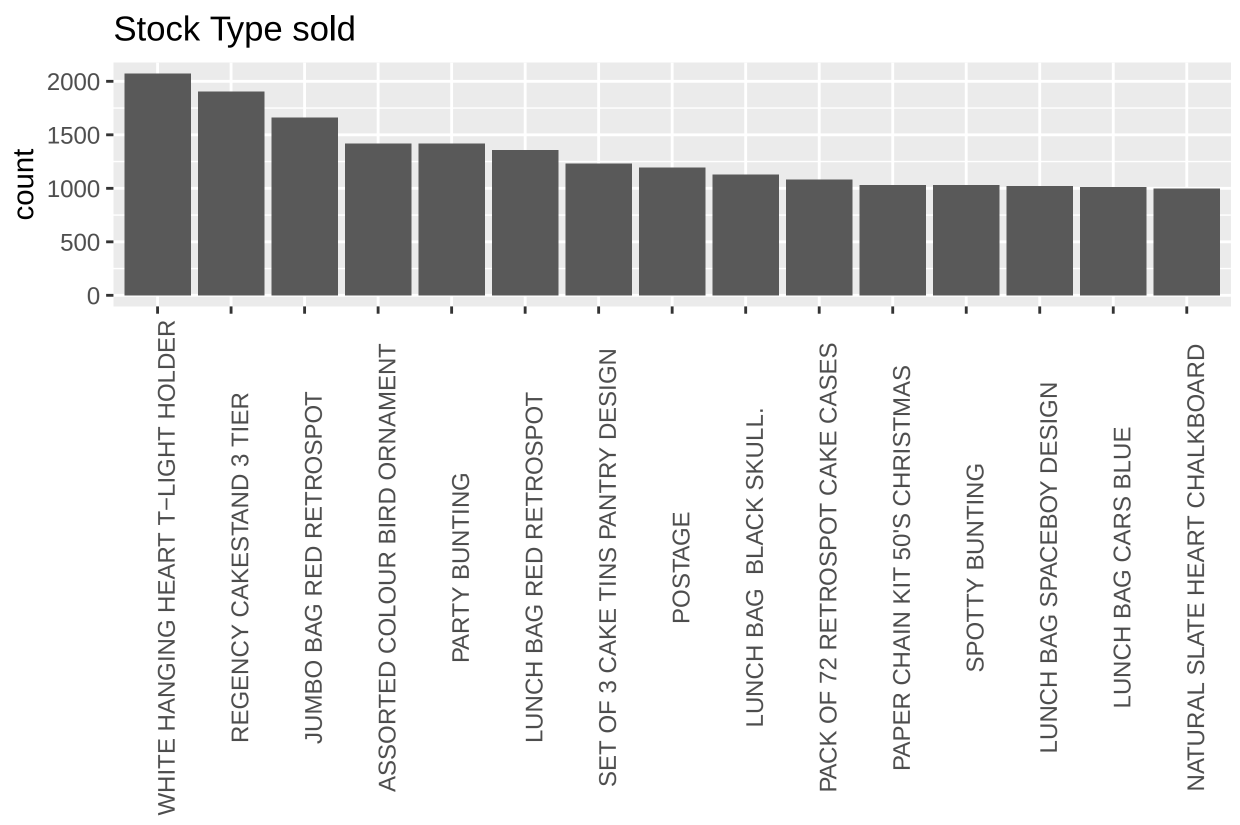
```r
# Value of the 10 cheapest products per unit
retail %>%
  group_by(Description) %>%
  filter (!(Description %in%
            c("Manual", "POSTAGE", "DOTCOM POSTAGE",
              "CRUK Commission", "Discount" ))) %>% #Removing unrelated transactions
  summarize(UnitPrice =  mean(UnitPrice)) %>%
  arrange(UnitPrice) %>%
  slice(1:10) %>%
  kable(format = "pipe")
```

| Description | UnitPrice |
|---|---|
| PADS TO MATCH ALL CUSHIONS | 0.001 |
| POPART WOODEN PENCILS ASST | 0.043 |
| CARTOON PENCIL SHARPENERS | 0.073 |
| HOUSE SHAPE PENCIL SHARPENER | 0.090 |
| LETTER SHAPE PENCIL SHARPENER | 0.095 |
| PORCELAIN BUDAH INCENSE HOLDER | 0.095 |
| PIECE OF CAMO STATIONERY SET | 0.096 |
| BLUE STONES ON WIRE FOR CANDLE | 0.100 |
| WRAP BAD HAIR DAY | 0.100 |
| DISCO BALL CHRISTMAS DECORATION | 0.119 |

It seems there is a big variation between the products from about `£600` to `£0.00075`.

Let us check the top 15 sold products.

```r
# Distribution of the top 15 Highest sold products per unit
retail %>% group_by(Description) %>%
  summarize(count =n()) %>%
  arrange(desc(count)) %>%
  slice(1:15) %>%
  ggplot(aes(fct_reorder(Description, desc(count)), count))+
  geom_col()+
  theme(axis.text.x = element_text(angle = 90))+
  labs(title = "Stock Type sold", x = "")
```



```r
# Display of the top 15 Highest sold products per unit
retail %>% group_by(Description) %>%
  summarize(count =n(), PriceUnit = mean(UnitPrice)) %>%
  arrange(desc(count)) %>%
  slice(1:15) %>%
  kable(format = "pipe")
```

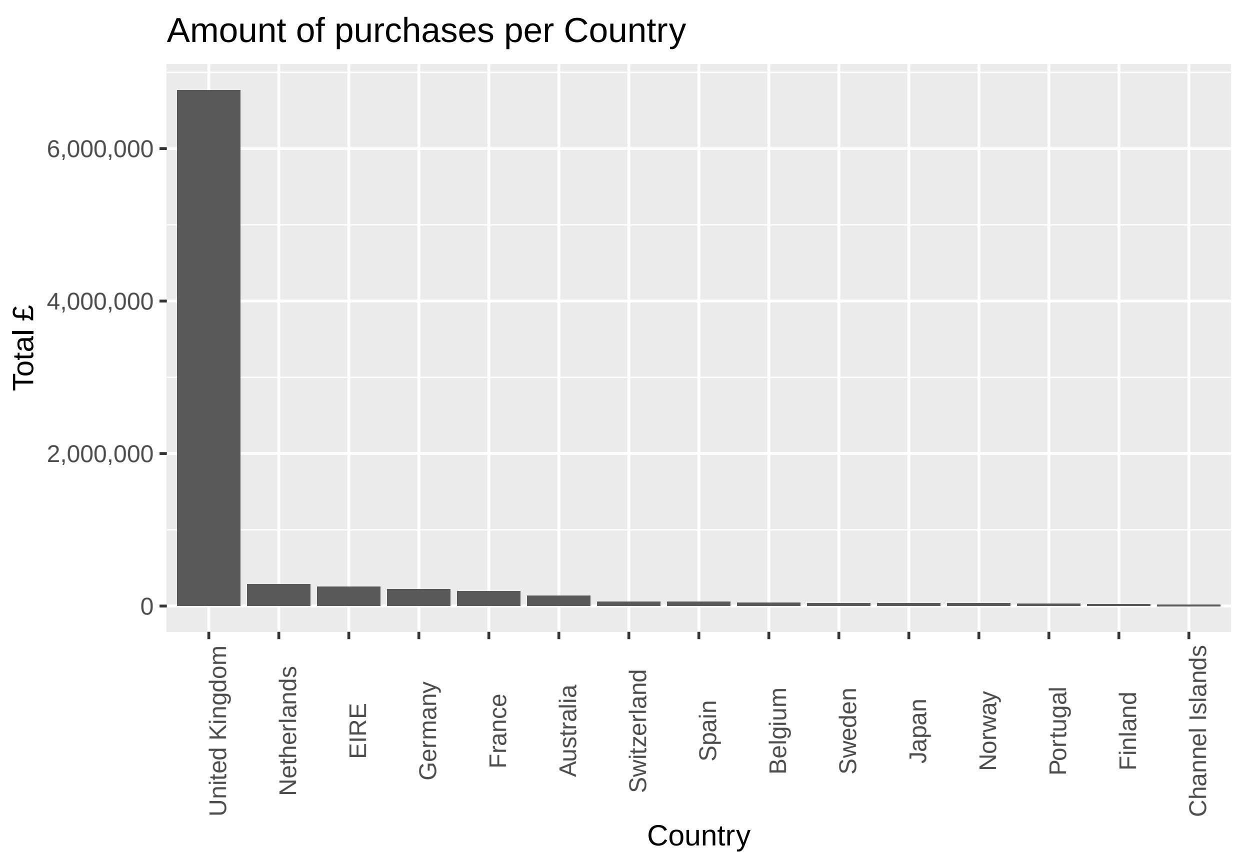
| Description | count | PriceUnit |
|:---|---:|---:|
| WHITE HANGING HEART T-LIGHT HOLDER | 2070 | 2.891 |
| REGENCY CAKESTAND 3 TIER | 1905 | 12.429 |
| JUMBO BAG RED RETROSPOT | 1662 | 2.013 |
| ASSORTED COLOUR BIRD ORNAMENT | 1418 | 1.681 |
| PARTY BUNTING | 1416 | 4.871 |
| LUNCH BAG RED RETROSPOT | 1358 | 1.657 |
| SET OF 3 CAKE TINS PANTRY DESIGN | 1232 | 4.951 |
| POSTAGE | 1196 | 37.889 |
| LUNCH BAG BLACK SKULL. | 1126 | 1.643 |
| PACK OF 72 RETROSPOT CAKE CASES | 1080 | 0.548 |
| PAPER CHAIN KIT 50'S CHRISTMAS | 1029 | 2.935 |
| SPOTTY BUNTING | 1029 | 4.916 |
| LUNCH BAG SPACEBOY DESIGN | 1021 | 1.655 |
| LUNCH BAG CARS BLUE | 1012 | 1.656 |
| NATURAL SLATE HEART CHALKBOARD | 997 | 2.974 |

This shows that the items that are sold the most are the low value items.

## 5.2 Country Analysis

Let us check the amount of purchases for the top 15 countries.

```r
# Distribution of the amount of purchases for the top 15 countries
retail %>% group_by(Country) %>%
  summarize(amount = sum(Total)) %>%
  arrange(desc(amount)) %>%
  slice(1:15) %>%
  ggplot(aes(fct_reorder(Country, desc(amount)), amount))+
  geom_col()+
  theme(axis.text.x = element_text(angle = 90)) +
  labs(title = "Amount of purchases per Country", x = "Country", y = "Total £") +
  scale_y_continuous(label=comma)
```



It seems that local customers in UK have the most purchases. Let us check the percentage.

```
# Display of the amount of purchases for the top 15 countries including the percentage
retail %>% group_by(Country) %>%
  summarize(amount = sum(Total)) %>%
  mutate(percent = percent(amount/sum(amount), accuracy = 0.01)) %>%
  arrange(desc(amount)) %>%
  kable(format = "pipe")
```

| Country | amount | percent |
|---|---:|---|
| United Kingdom | 6767873 | 81.54% |
| Netherlands | 284662 | 3.43% |
| EIRE | 250285 | 3.02% |
| Germany | 221698 | 2.67% |
| France | 196713 | 2.37% |
| Australia | 137077 | 1.65% |
| Switzerland | 55739 | 0.67% |
| Spain | 54775 | 0.66% |
| Belgium | 40911 | 0.49% |
| Sweden | 36596 | 0.44% |
| Japan | 35341 | 0.43% |
| Norway | 35163 | 0.42% |
| Portugal | 29060 | 0.35% |
| Finland | 22327 | 0.27% |
| Channel Islands | 20086 | 0.24% |

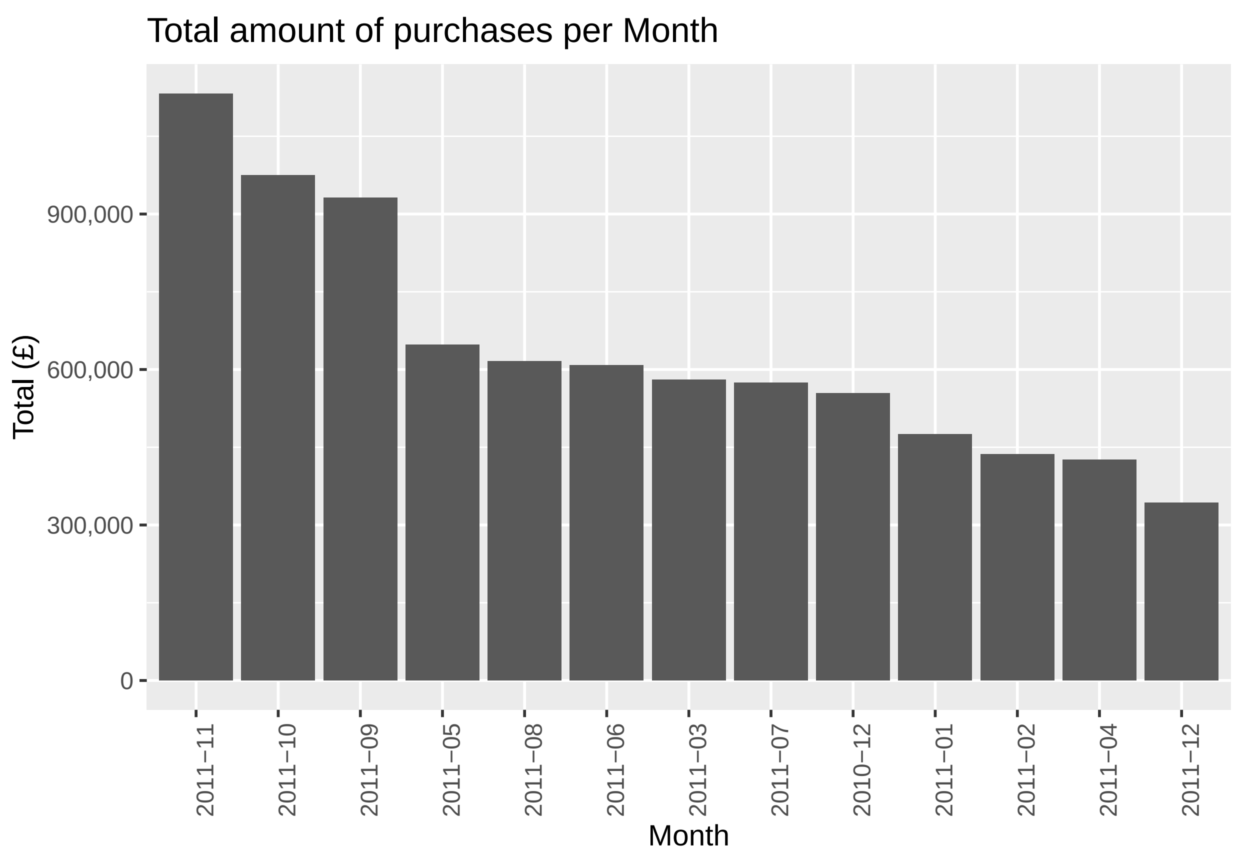
| Country | amount | percent |
|---|---|---|
| Denmark | 18768 | 0.23% |
| Italy | 16891 | 0.20% |
| Cyprus | 12946 | 0.16% |
| Austria | 10154 | 0.12% |
| Singapore | 9120 | 0.11% |
| Poland | 7213 | 0.09% |
| Israel | 6994 | 0.08% |
| Greece | 4711 | 0.06% |
| Iceland | 4310 | 0.05% |
| Canada | 3666 | 0.04% |
| Unspecified | 2667 | 0.03% |
| Malta | 2505 | 0.03% |
| United Arab Emirates | 1902 | 0.02% |
| USA | 1731 | 0.02% |
| Lebanon | 1694 | 0.02% |
| Lithuania | 1661 | 0.02% |
| European Community | 1292 | 0.02% |
| Brazil | 1144 | 0.01% |
| RSA | 1002 | 0.01% |
| Czech Republic | 708 | 0.01% |
| Bahrain | 548 | 0.01% |
| Saudi Arabia | 131 | 0.00% |

The table above shows that about 82% of the purchase value are from local customers from UK. Which makes sense since the delivery time and cost are low.

## 5.3 Time Analysis

Let us check the monthly purchases.

```r
# Distribution of the total amount of purchases per month
retail %>%
  mutate(month = format_ISO8601(retail$InvoiceDate, precision = "ym")) %>%
  group_by(month) %>%
  summarize(Total = sum(Total), products = sum(Quantity)) %>%
  ggplot(aes(x = reorder(month, -Total) , Total)) +
  geom_col() +
  theme(axis.text.x = element_text(angle = 90)) +
  scale_y_continuous(label=comma) +
  ggtitle("Total amount of purchases per Month") +
    xlab("Month") +
    ylab("Total (£)")
```



The data shows that most of the purchases are made around September, October and November. Since the online retail products are all-occasion gifts and most of the customers are whole sales, the most sales would be 3 months before Christmas to restock for holidays season.

## 5.4 Customer Analysis

```
# Display of the top 15 customers with the highest purchases
retail %>% group_by(CustomerID) %>%
  summarize(invoices = n_distinct(InvoiceNo), Country = Country[1],Total = sum(Total)) %>%
  arrange(desc(Total)) %>%
  slice(1:15)
```

```
## # A tibble: 15 x 4
##    CustomerID invoices Country          Total
##         <dbl>    <int> <chr>            <dbl>
##  1      14646       77 Netherlands    279489.
##  2      18102       62 United Kingdom 256438.
##  3      17450       55 United Kingdom 187482.
##  4      14911      248 EIRE           132573.
##  5      12415       26 Australia      123725.
##  6      14156       66 EIRE           113384.
##  7      17511       46 United Kingdom  88125.
##  8      16684       31 United Kingdom  65892.
##  9      13694       60 United Kingdom  62653.
## 10      15311      118 United Kingdom  59419.
## 11      13089      118 United Kingdom  57386.
## 12      14096       34 United Kingdom  57121.
## 13      15061       55 United Kingdom  54229.
## 14      17949       52 United Kingdom  52751.
## 15      15769       29 United Kingdom  51824.
```

The data above shows that customers who are local (UK) or neighboring countries like Ireland and Netherlands, tend to place smaller orders while a far customer based in Australia would order big quantities to save in transportation cost and time.

## 6 RFM Analysis

Before creating a RFM table we will assume that the analysis is done after the end year of 2011. First day on 2012.

```r
# Assign a_date variable a date value of first of January 2012.
a_date <- as.Date("2012-01-01")
```

Since we want to focus on customers and how to segment them we will create the RFM table which stands for Recency, Frequency and Monetary.

1- Recency would be created by counting the days from the last purchase for each customer. This would be the difference between `01-Jan-2012` and the last purchase.

2- Frequency would be calculate by adding all the unique invoice done from `01-Dec-2010` till `09-Dec-2011`.

3- Monetary represents all the sum of money spent by each customer buying products between `01-Dec-2010` and `09-Dec-2011`.

Note: To calculate Monetary we also added the cancelled and discounted purchases.

```r
# Calculate and create RFM table using the retail data
retail_RFM <- retail %>%
              group_by(CustomerID) %>%
              summarize(Recency = as.numeric(a_date - max(as.Date(InvoiceDate))),
              Frequency = n_distinct(InvoiceNo),
              Monetary = sum(Total))
head(retail_RFM) %>% kable(format = "pipe")
```

| CustomerID | Recency | Frequency | Monetary |
|---:|---:|---:|---:|
| 12346 | 348 | 2 | 0 |
| 12347 | 25 | 7 | 4310 |
| 12348 | 98 | 4 | 1797 |
| 12349 | 41 | 1 | 1758 |
| 12350 | 333 | 1 | 334 |
| 12352 | 59 | 11 | 1545 |

## 7 K-means Cluster

We will now use K-means cluster algorithm to the RFM analysis for each cusotmer.

Let us first remove the CustomerID for the data so we can have only the Recency, Frequency and Monetary columns.

```r
# Removing the CustomerID column from the retail_RFM table
RFM <- retail_RFM[2:4]
head(RFM) %>% kable(format = "pipe")
```

| Recency | Frequency | Monetary |
|---:|---:|---:|
| 348 | 2 | 0 |
| 25 | 7 | 4310 |
| 98 | 4 | 1797 |
| 41 | 1 | 1758 |
| 333 | 1 | 334 |
| 59 | 11 | 1545 |

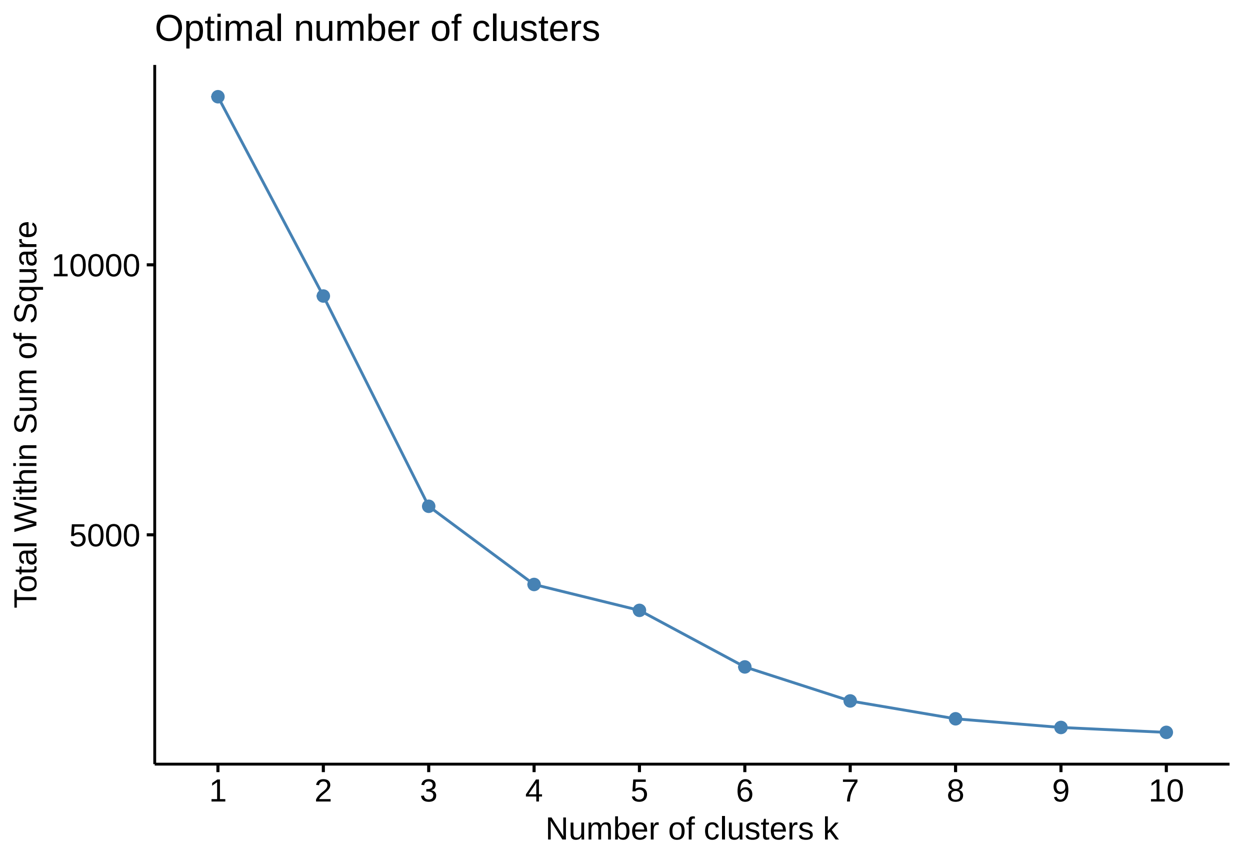
A very important step that should be done before running the clustering model is to scale the data. If you look at the table above the three columns are completely different from each other and we don't want the monetary values to have more influence on the other variables.

```r
# Scale the RFM data
RFM_scale <- scale(RFM)
head(RFM_scale) %>% kable(format = "pipe")
```

| Recency | Frequency | Monetary |
|---:|---:|---:|
| 2.316 | -0.329 | -0.231 |
| -0.889 | 0.206 | 0.293 |
| -0.165 | -0.115 | -0.012 |
| -0.730 | -0.436 | -0.017 |
| 2.167 | -0.436 | -0.190 |
| -0.552 | 0.634 | -0.043 |

Now to choose the optimal `K` value for our model using Within-Cluster-Sum of Squared Errors (wss) or called Elbow method. Elbow method uses the Sum of Squares to find the distances from a cluster mean.

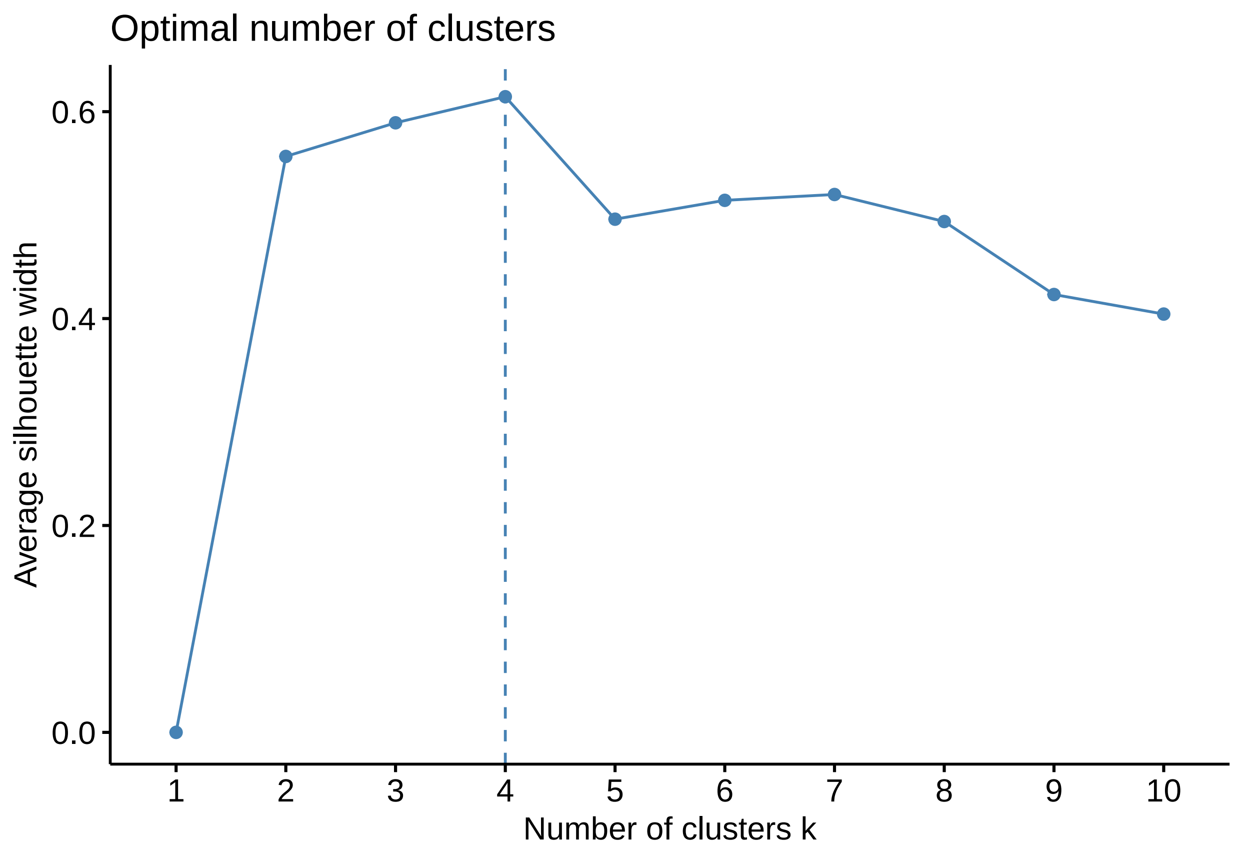
```r
# Display the elbow plot method for the scaled RFM data to find optimal K clusters
fviz_nbclust(RFM_scale, kmeans, method = "wss")
```



The elbow started at k=3, however, we can choose k=4. If we choose a bigger value than 4 the difference in the Sum of Squares wouldn't be big.

Now let us try the silhouette width method to find the optimal `K` value. This model measures how similar a value is to its clustered group and compares it with other clusters. The silhouette values will be between +1 and -1. The average cluster that has the highest silhouette range can represent the best `K` value.

```
# Display the silhouette plot method for the scaled RFM data to find optimal K clusters
fviz_nbclust(RFM_scale, kmeans, method = "silhouette")
```



The silhouette method also shows that when k=4 it will give the highest average silhouette width which is the optimal value for k.

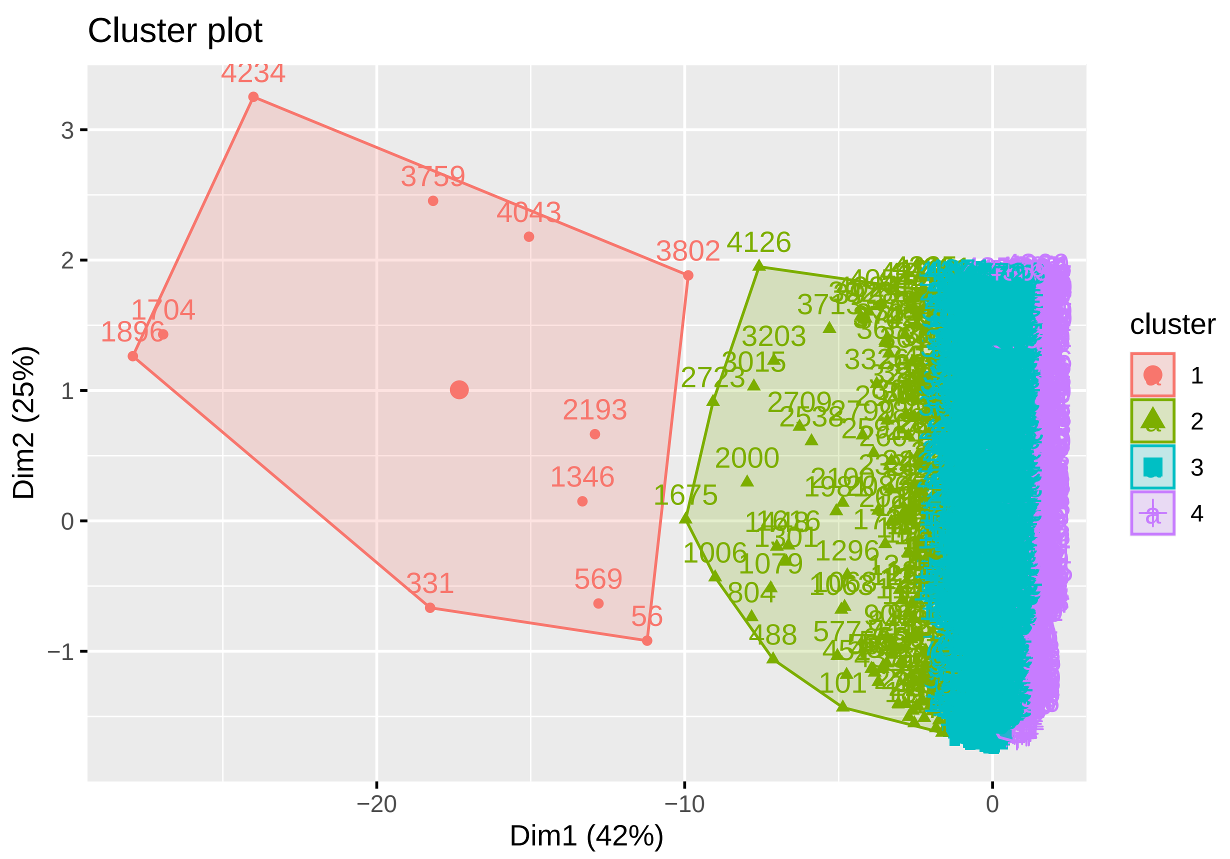
Both of these methods shows that 4 clusters (k=4) would give us the optimal results.

After choosing the number of clusters we will run K-means model

```
# set the seed and run K-means model
set.seed(14, sample.kind="Rounding")
```

```
## Warning in set.seed(14, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

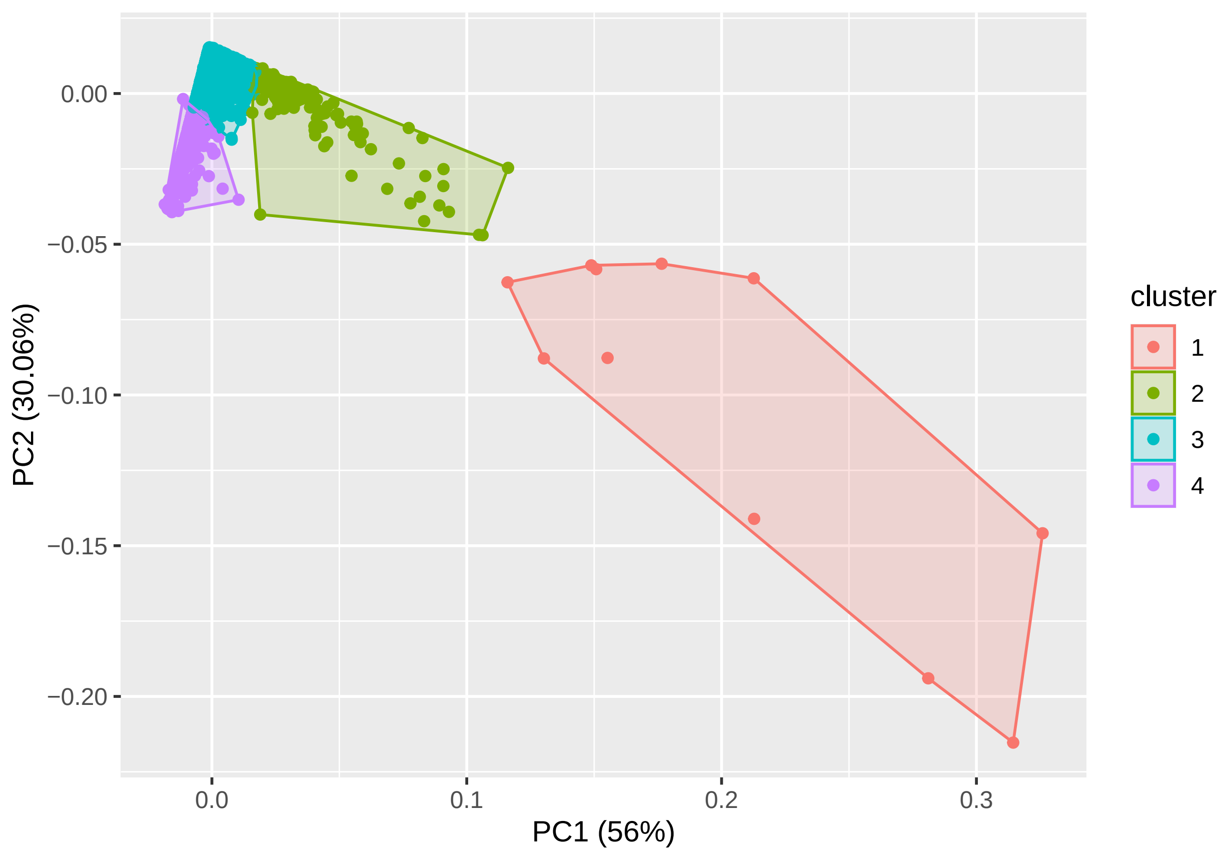
```
clusters <- kmeans(RFM_scale, centers = 4, iter.max = 100, nstart = 100)

#graphing the clusters
fviz_cluster(clusters, data = retail_RFM)
```



To evaluate our model first we will run a cluster plot. Then we will run the cluster centers.

To run the cluster plot see the following code.

```
# Cluster plot evaluation for the model
autoplot(clusters, RFM_scale, frame = TRUE)
```



All 4 clusters are not overlapping which shows that they are distinct.

To test the cluster centers see the following code.

```
# Cluster centers evaluation for the model
clusters$centers
```

```
##   Recency Frequency Monetary
## 1  -0.865   11.2257  14.8934
## 2  -0.809    2.5459   1.2803
## 3  -0.489   -0.0726  -0.0696
## 4   1.558   -0.3505  -0.1756
```

The centers in each column are not similar to each other which means they are not overlapping. Therefore the clusters are distinct.

## 8 Results

Now we will apply our model to the retail data and segment the customers based on the cluster.

```
# Set the seed and add the cluster model results column to the Customer data
set.seed(14, sample.kind="Rounding")
```

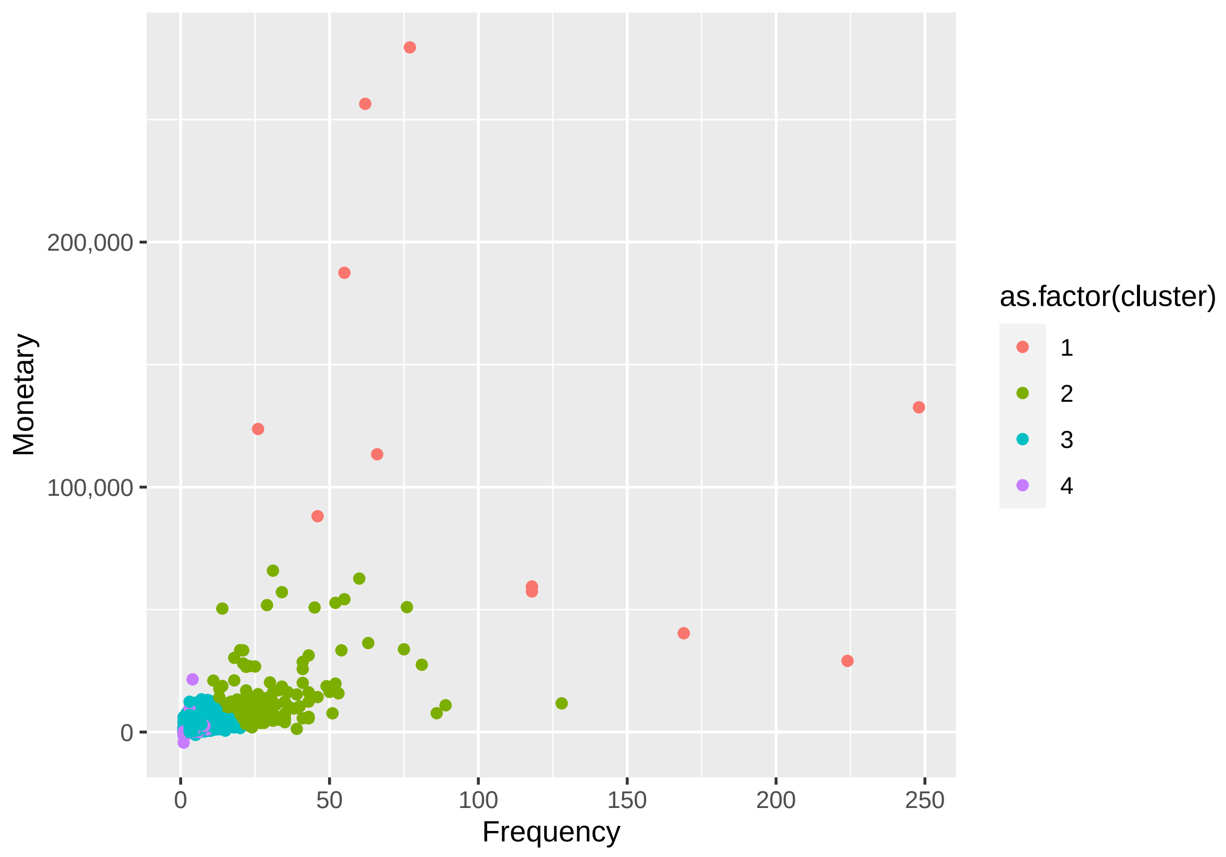
```
## Warning in set.seed(14, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```
retail_RFM<- retail_RFM %>% mutate(cluster = clusters$cluster)
head(retail_RFM) %>% kable(format = "pipe")
```

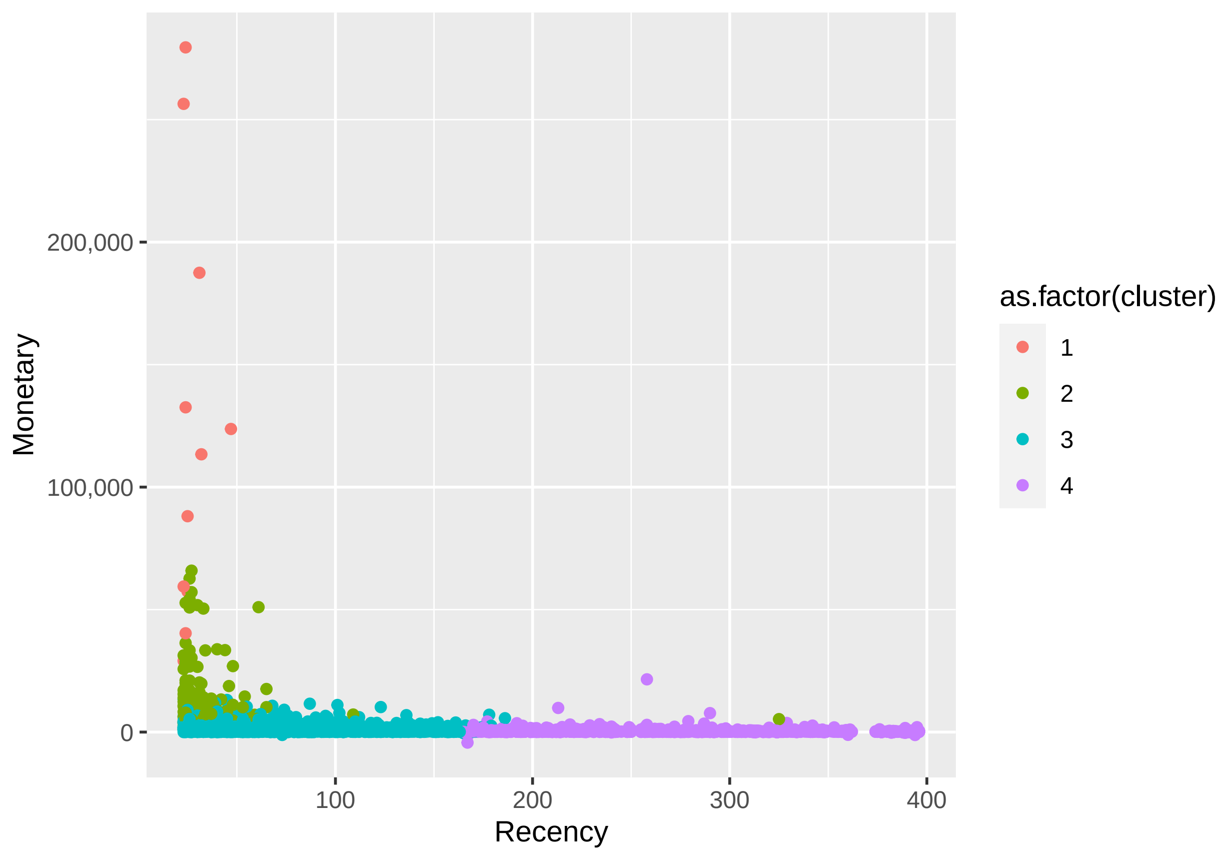
| CustomerID | Recency | Frequency | Monetary | cluster |
|---:|---:|---:|---:|---:|
| 12346 | 348 | 2 | 0 | 4 |
| 12347 | 25 | 7 | 4310 | 3 |
| 12348 | 98 | 4 | 1797 | 3 |
| 12349 | 41 | 1 | 1758 | 3 |
| 12350 | 333 | 1 | 334 | 4 |
| 12352 | 59 | 11 | 1545 | 3 |

If we graph the points based on the cluster that will give us the following:

```
# Display the cluster points on Monetary vs Frequency plot
retail_RFM %>%
  ggplot(aes(x = Frequency, y = Monetary, col = as.factor(cluster))) +
  geom_point() +
  scale_y_continuous(label=comma)
```



```r
# Display the cluster points on Monetary vs Recency plot
retail_RFM %>%
  ggplot(aes(x = Recency, y = Monetary, col = as.factor(cluster))) +
  geom_point() +
  scale_y_continuous(label=comma)
```



From the results on the graph the following can be concluded:

1- Cluster 1 contains customers that made the biggest purchase amount within a short time. They also purchased the most orders. However, this cluster has the smallest number of customers.

2- Cluster 2 contains customers that made good purchases within a slightly longer time than cluster 1. They placed a good amount of orders. This cluster has a higher number of customers compared to Cluster 1 but it is way smaller than clusters 3 and 4.

3- Cluster 3 and 4 contain the least amount of purchases and orders compared to clusters 1 and 2. Most of the customers fall in these two clusters.

The average values for each cluster can confirm the above results.

```r
# Group the clusters in to a table with Recency, Frequency, and Monetary averages.
retail_RFM %>%
  group_by(cluster) %>%
  summarize( count = n(),
             Recency_avg = mean(Recency),
             Frequency_avg = mean(Frequency),
             Monetary_avg = mean(Monetary)) %>%
  kable(format = "pipe")
```

| cluster | count | Recency_avg | Frequency_avg | Monetary_avg |
|---:|---:|---:|---:|---:|
| 1 | 11 | 27.4 | 109.9 | 124312 |
| 2 | 188 | 33.0 | 28.9 | 12422 |
| 3 | 3097 | 65.3 | 4.4 | 1326 |

| cluster | count | Recency_avg | Frequency_avg | Monetary_avg |
|---|---|---|---|---|
| 4 | 1076 | 271.6 | 1.8 | 455 |

The Online retail company can focus its efforts on the group of customers in clusters 1 and 2. They can reach out to them during the middle of the year to check their forecast for peak season. With that, it will help the company to replenish their stock based on these customers forecast.

However, for customers in cluster 3 it would be hard to keep in contact with over 3000 customers and the company shouldn't put any efforts for cluster 4.

## 9 Conclusion

We first gave a brief introduction to the project. Secondly, gave an overview for the source of the Online data set and what model we will be using through our project. Then, explained the data structure and cleaned the data by removing the unknown customers. After that, we analyzed the data and created a new RFM data analysis which was used later to create our K-means clustering model. The K-means clustering model helped us segment the customers based on the following:

1- Recency: Days passed since the last purchase was done.

2- Frequency: How many orders made.

3- Monetary: How much money was spent.

The model helped us divide the customers into 4 groups. Group 1 and 2 are the most important customers. They are making many and more frequent orders. While groups 3 and 4 consist of the majority of the customers, they are making little to less frequent orders.

There can be more room for improvement in our model if the data was available for the unknown customers. Also, an interesting project would be removing items that doesn't make much profit.

Also, to improve our model we could've included the amount of recalled (cancellation) or damaged products per customer.

## 10 References

Data Ninjas (2021) How to perform clustering in R with the K-means algorithm - R for data science, YouTube. Available at: https://www.youtube.com/watch?v=5mlth-yM2NE&t=587s

Delval, F. (2022) What is RFM analysis?, ActionIQ. Available at: https://www.actioniq.com/blog/what-is-rfm-analysis/

Irizarry, R.A. (2019) Introduction to data science, rafalab. Available at: https://rafalab.github.io/dsbook/

Online retail (2015) UCI Machine Learning Repository. Available at: https://doi.org/10.24432/C5BW33