615 Final Report

Ecommerce Data Analysis

Qianhui Rong

12/15/2018

Data Import

This data is downloaded from Github:
https://github.com/rpomponio/ecommerce_analytics.

It’s a transnational data set which contains all the transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail. The company mainly sells unique all-occasion gifts. Many customers of the company are wholesalers.

```
data_retail <- read.csv("data.csv")
```

Data Cleaning

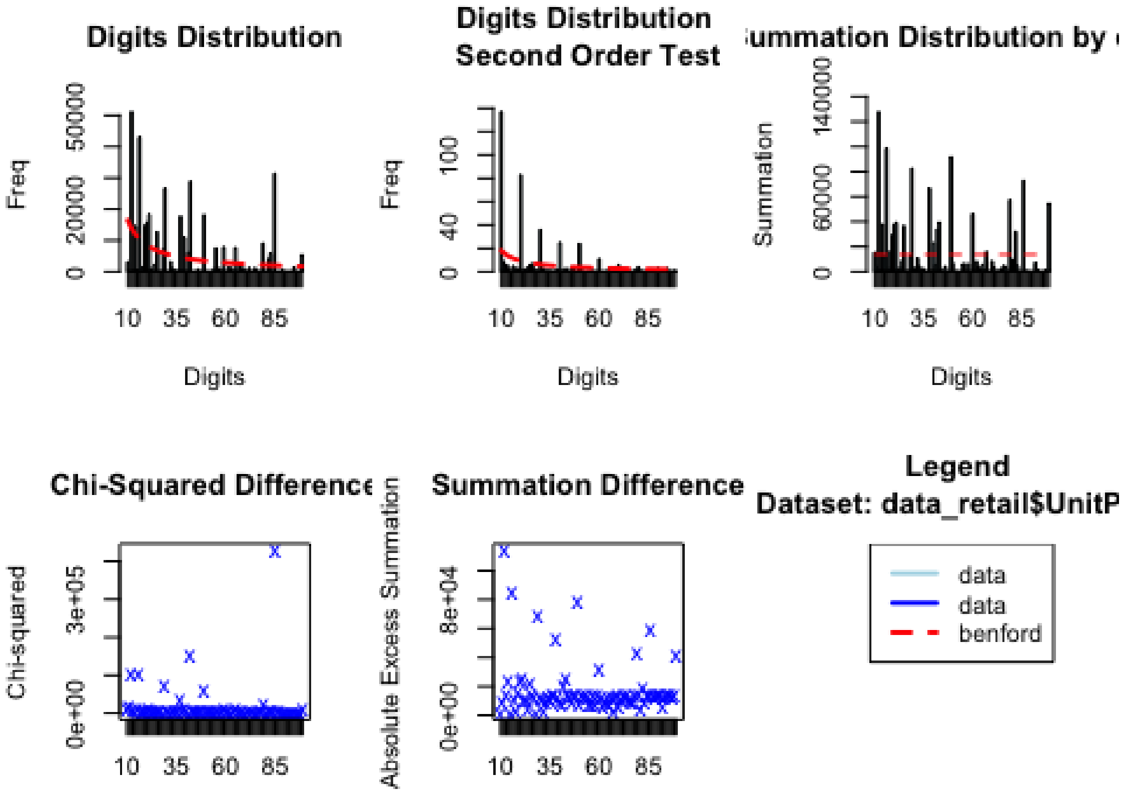
```
#Upon observation, there are negative value in quantity and unit price column
s, which doesn't make sense in normal sense.


#(maybe that's because item return or item damaged goods, but to simplify the
data I decided to remove them).


data_retail %<>% filter(UnitPrice>0 & Quantity>0)
data_retail <- na.omit(data_retail)

#Add a column of total amount spent
data_retail$amount_spend <- data_retail$Quantity*data_retail$UnitPrice


#Add Date,Day,Month,Year,Hour columns
#Split InvoiceDate into date and time
data_retail$Date <- sapply(data_retail$InvoiceDate, FUN = function(x){str_spl
it(x, ' ')[[1]][1]})
data_retail$Time <- sapply(data_retail$InvoiceDate, FUN=  function(x){str_spl
it(x, ' ')[[1]][2]})


#Create Month,Year,Hour
```

```
data_retail$Month <- sapply(data_retail$Date, FUN = function(x) {str_split(x,
'/')[[1]][1]})
data_retail$Year <- sapply(data_retail$Date, FUN = function(x) {str_split(x,
'/')[[1]][3]})
data_retail$Hour <- sapply(data_retail$Time, FUN = function(x) {str_split(x,
':')[[1]][1]})


#Convert to Date format
data_retail$Date <- as.Date(data_retail$Date,"%m/%d/%Y")
data_retail$DateofMonth <- as.factor(day(data_retail$Date))

#Add a column of day of the week
data_retail$Day <- wday(data_retail$Date,label=TRUE) #Day in the week

#Format conversion for other variables
data_retail$Country <- as.factor(data_retail$Country)
data_retail$Month <- as.factor(data_retail$Month)
data_retail$Year <- as.factor(data_retail$Year)
data_retail$Hour <- as.factor(data_retail$Hour)
data_retail$Day <- as.factor(data_retail$Day)
levels(data_retail$Year) <- c(2010,2011)
```

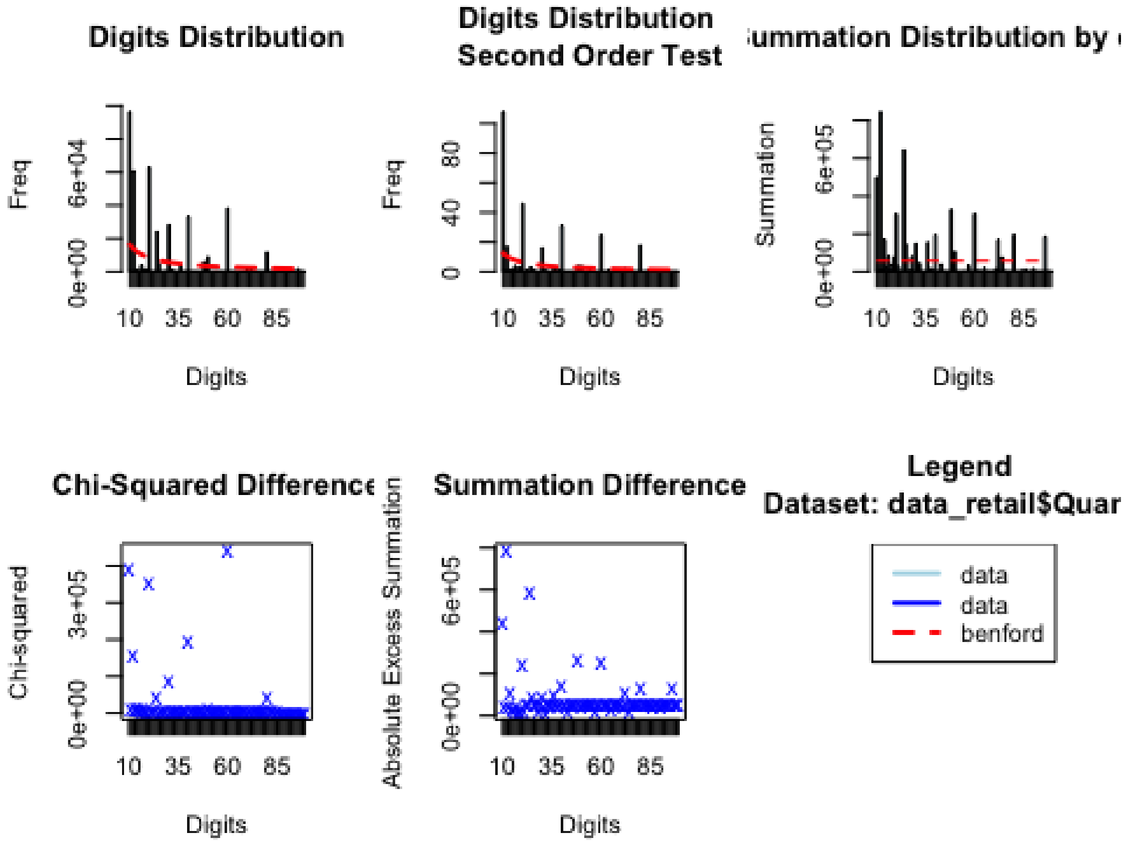
Data Validity Analysis

This section consists of proving data’s distribution.

```
#Benford Law Test
bfd.Price <- benford(data_retail$UnitPrice)
bfd.Quant <- benford(data_retail$Quantity)
plot(bfd.Price)
```



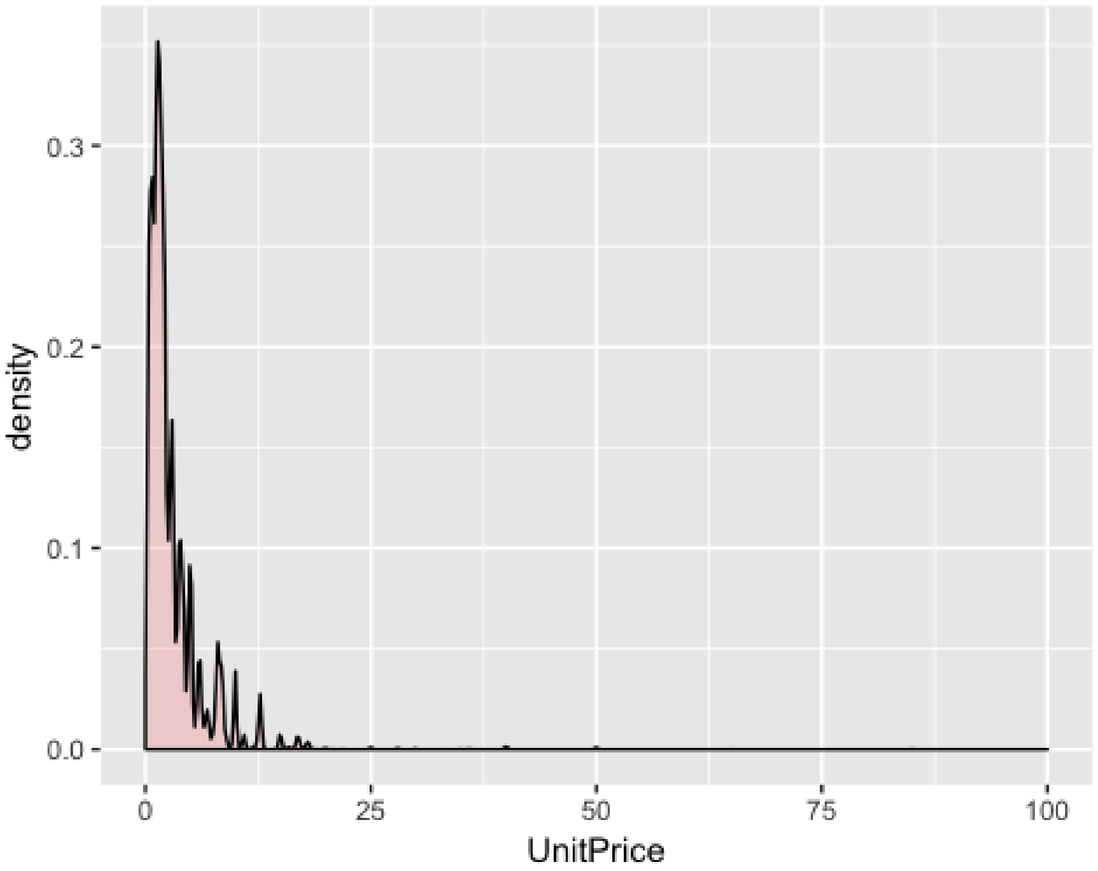
```
plot(bfd.Quant)
```



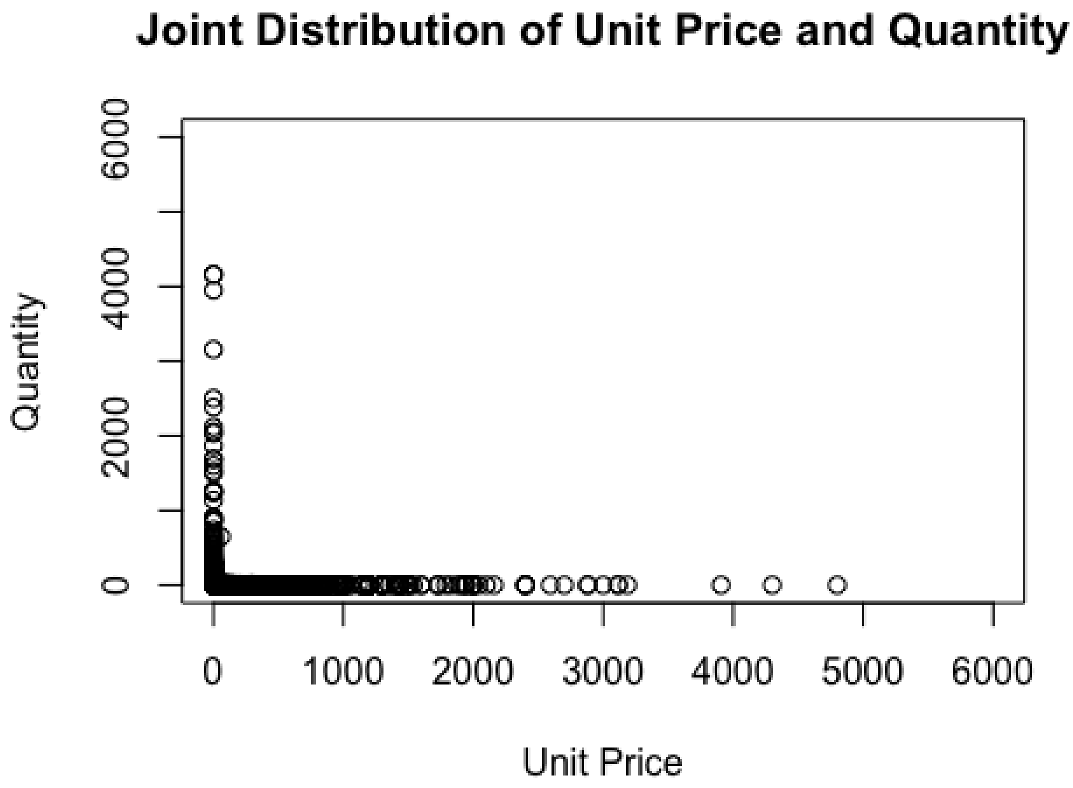
From both benford plots and chi-square test, we can see that unit price and quantity don't follow Benford Distribution and the difference is large.

Besides Benford Law check: I'll check the distributions of Unit Price and Quantity to see if they follow the ones I've expected.

```
#For Unit Price: I expect a larger amount at lower price (ex. 0 ~ 10 dollars) 
and a smaller amount at higher price;


ggplot(data_retail, aes(x=UnitPrice)) +
    geom_histogram(aes(y=..density..),binwidth=100,
                   colour="black", fill="white") +
    geom_density(alpha=.2, fill="#FF6666")+
    xlim(c(0,100))
```



```
#Almost all data are located between 0~25 dollar. This distribution plot seem
s like a Laplace Distribution's right side with a drastical decay rate.

#For Quantity: I expect a larger amount for cheaper price items and a smaller
amount for more expensive items.


plot(x=data_retail$Quantity,y=data_retail$UnitPrice,ylim = c(0,6000),xlim=c(0
,6000),
     ylab="Quantity",xlab="Unit Price",main = "Joint Distribution of Unit Pri
ce and Quantity")
```

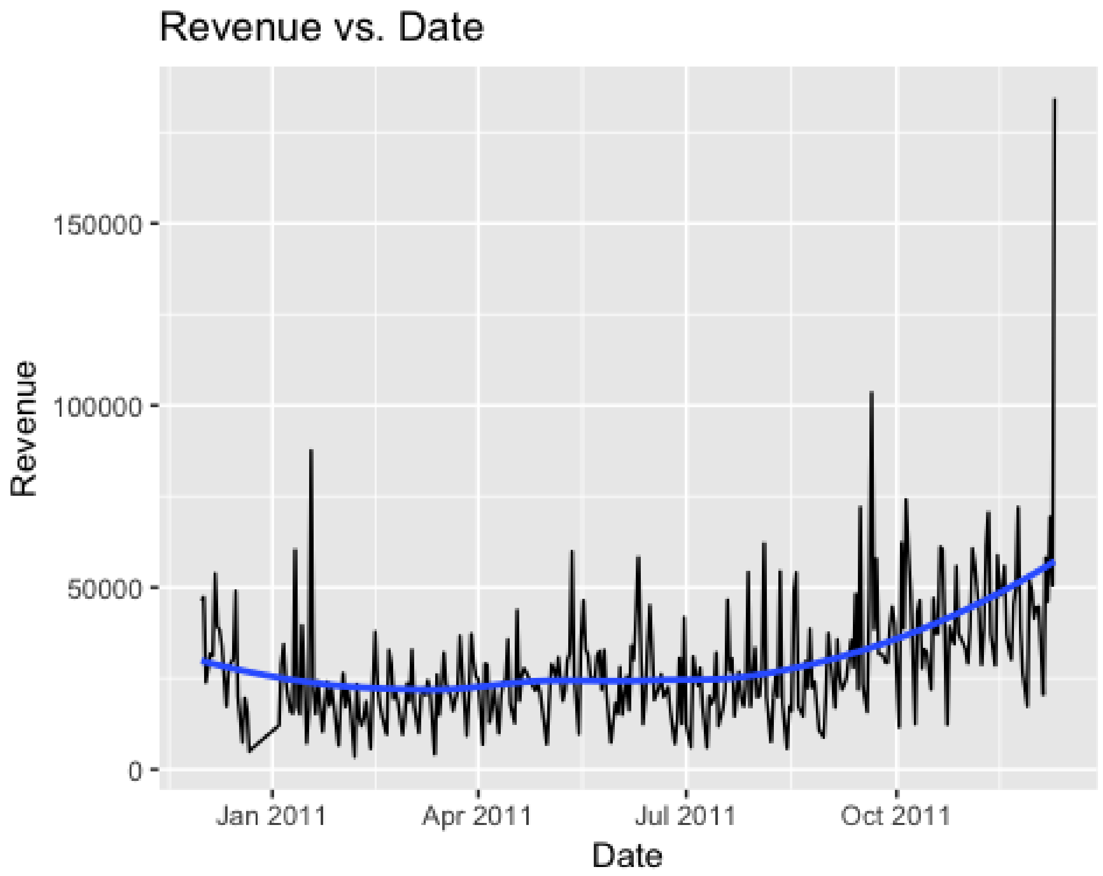


```
#The plot confirms my guess: around zero the purchase quantity is really high
, and as the price increases, the quantity decrease(sharply). We can say that
the purchase quantity of items priced around zero can not be compared with th
ose higher than zero to the same scale.
```

Exploratory Data Analysis

General Trend

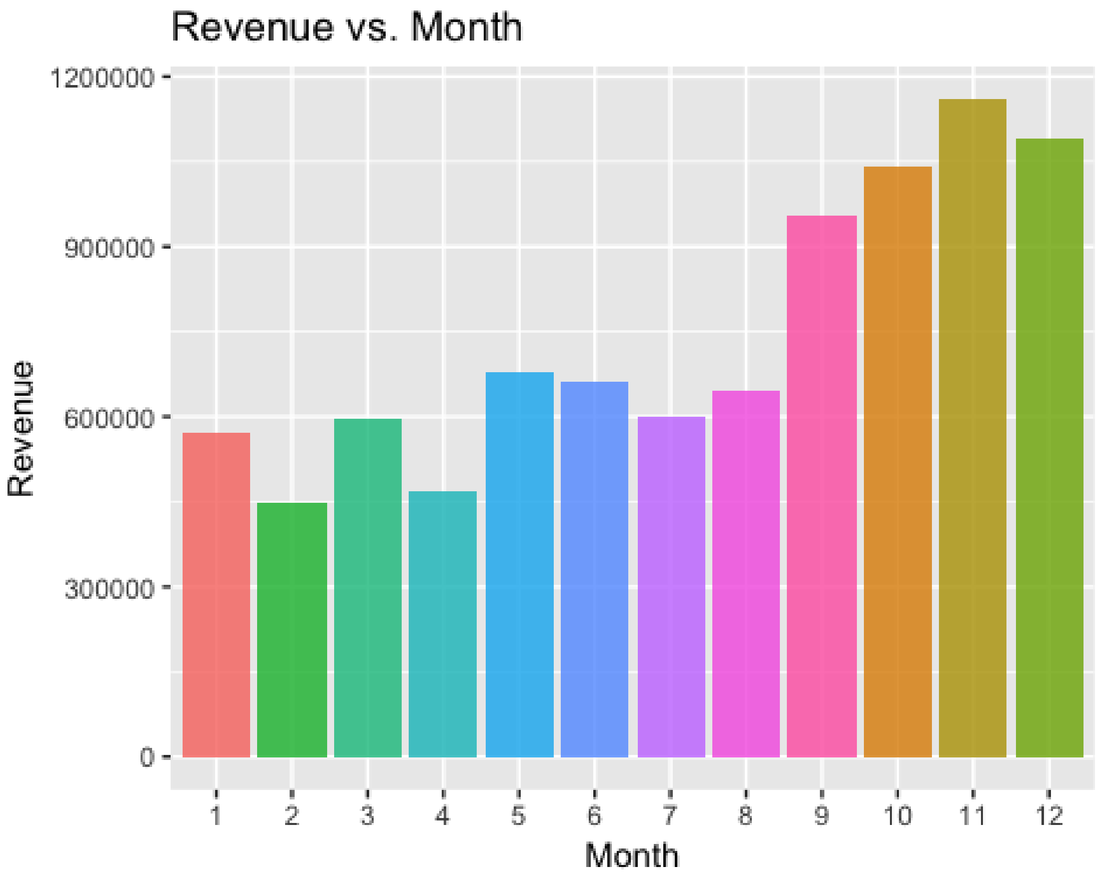
```
#1.Revenue by Date
data_retail %>%
  group_by(Date) %>%
  summarise(revenue = sum(amount_spend)) %>%
  ggplot(aes(x=Date,y=revenue))+geom_line()+
  geom_smooth(method = 'auto', se = FALSE)+
  labs(x = 'Date', y = 'Revenue', title = 'Revenue vs. Date')

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



This plot shows an increasing trend starting from Sept.2011.

Month Trend

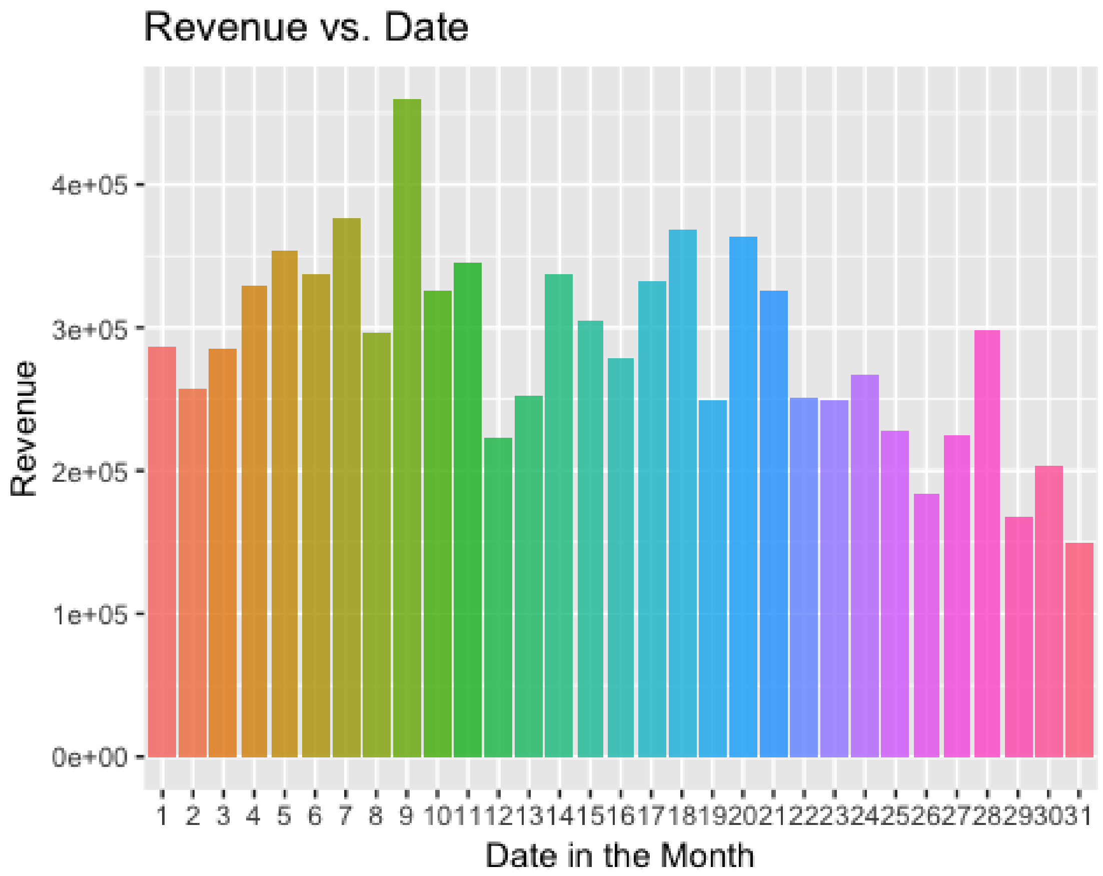
```
#2.Revenue by Month
data_retail %>%
  group_by(Month) %>%
  summarise(revenue = sum(amount_spend)) %>%
  ggplot(aes(x=Month,y=revenue,fill=Month))+geom_col(alpha=0.8)+
  labs(x = 'Month', y = 'Revenue', title = 'Revenue vs. Month')+
  scale_x_discrete(limits=c("1","2","3","4","5","6","7","8","9","10","11","12
"))+
  guides(fill=FALSE)
```



This plot shows a relatively constant revenue during Jan~Aug, and a boost from Sept to Dec with a peak in Nov. It is reasonable in normal sense, due to Black Friday and Christmas Shopping.

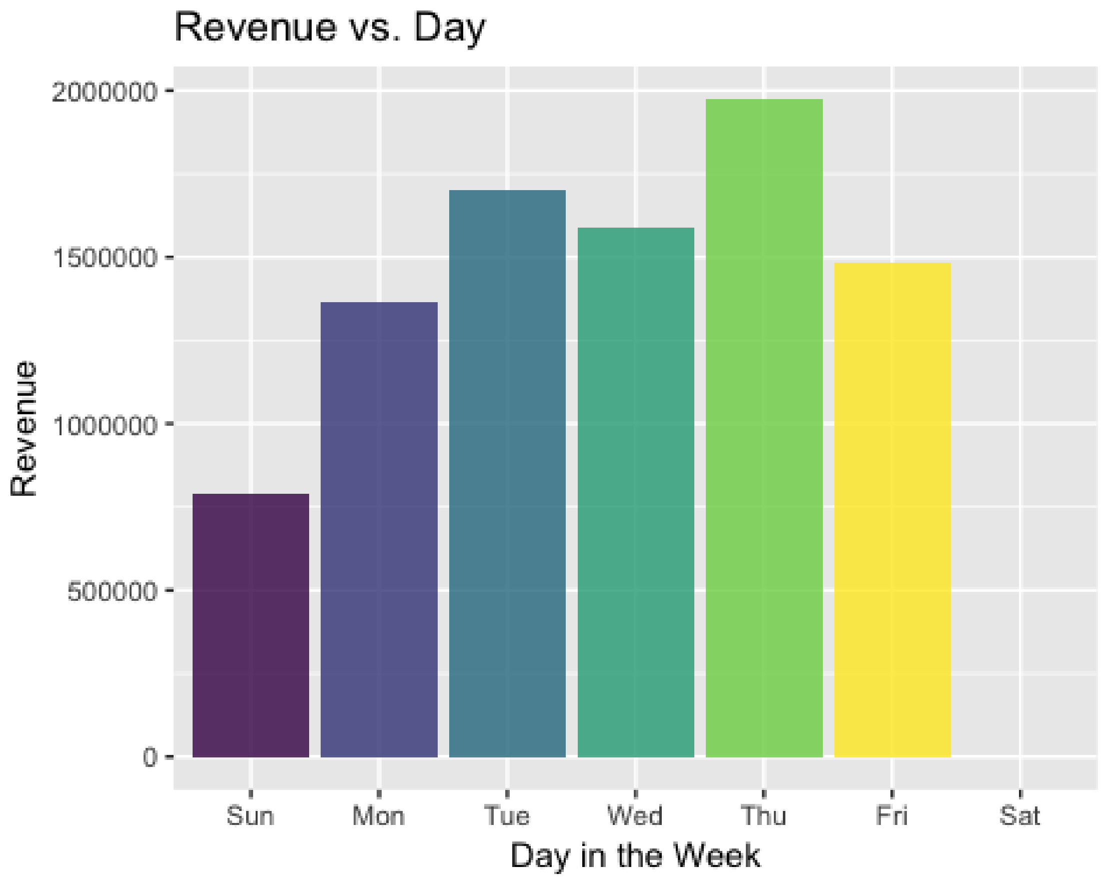
Day and Date Trend

```
#3.Revenue by Date
data_retail %>%
  group_by(DateofMonth) %>%
  summarise(revenue = sum(amount_spend)) %>%
  ggplot(aes(x=DateofMonth,y=revenue,fill=DateofMonth))+geom_col(alpha=0.8)+
  labs(x = 'Date in the Month', y = 'Revenue', title = 'Revenue vs. Date')+
  guides(fill=FALSE)
```



The trend is relatively constant during the month. There are some higher and lower dates but not too insightful.

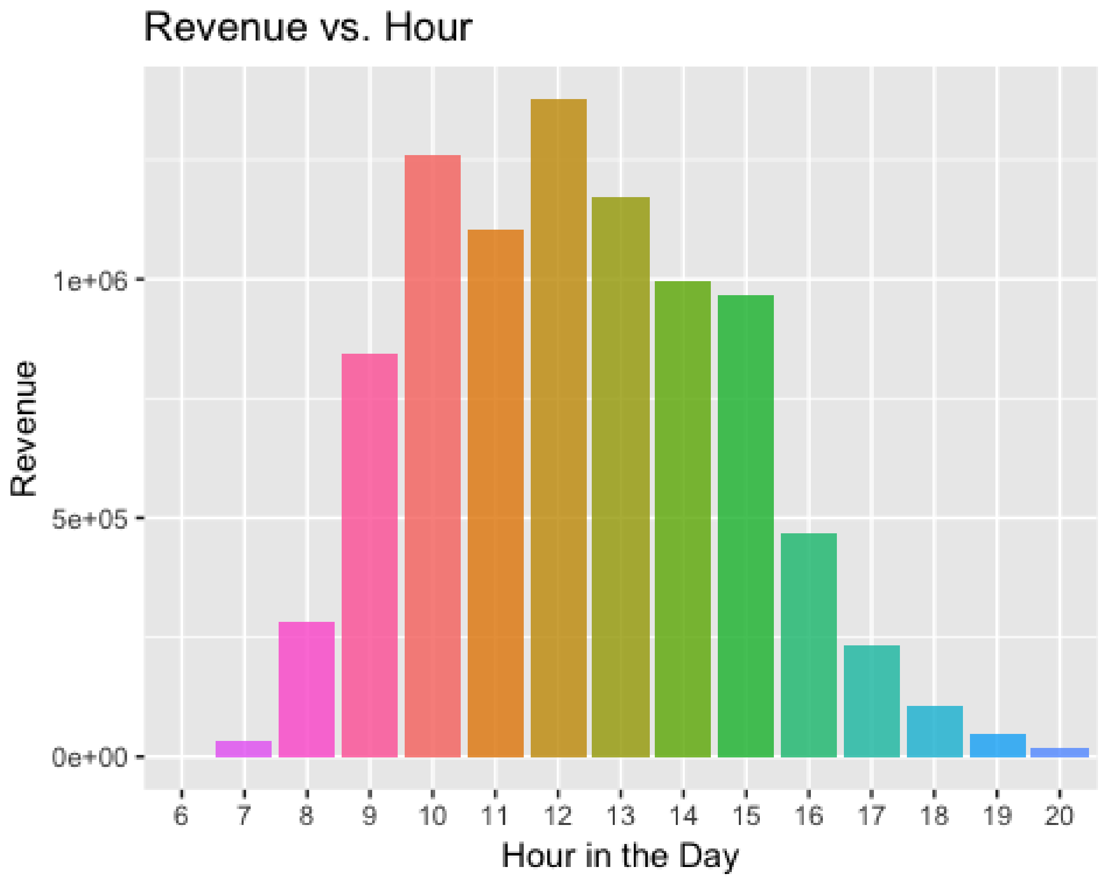
```
#4.Revenue by Day
data_retail %>%
  group_by(Day) %>%
  summarise(revenue = sum(amount_spend)) %>%
  ggplot(aes(x=Day,y=revenue,fill=Day))+geom_col(alpha=0.8)+
  labs(x = 'Day in the Week', y = 'Revenue', title = 'Revenue vs. Day')+
  scale_x_discrete(limits=c("Sun","Mon","Tue","Wed","Thu","Fri","Sat"))+
  guides(fill=FALSE)
```



First thing to observe is that no transaction on Saturday. Sunday, Monday, Tuesday and Wednesday have similar amount of revenue; Thursday is the peak, bout 25% higher than those four days; and Friday has seen lower revenue, about 25% lower than those four days.

Hour Trend

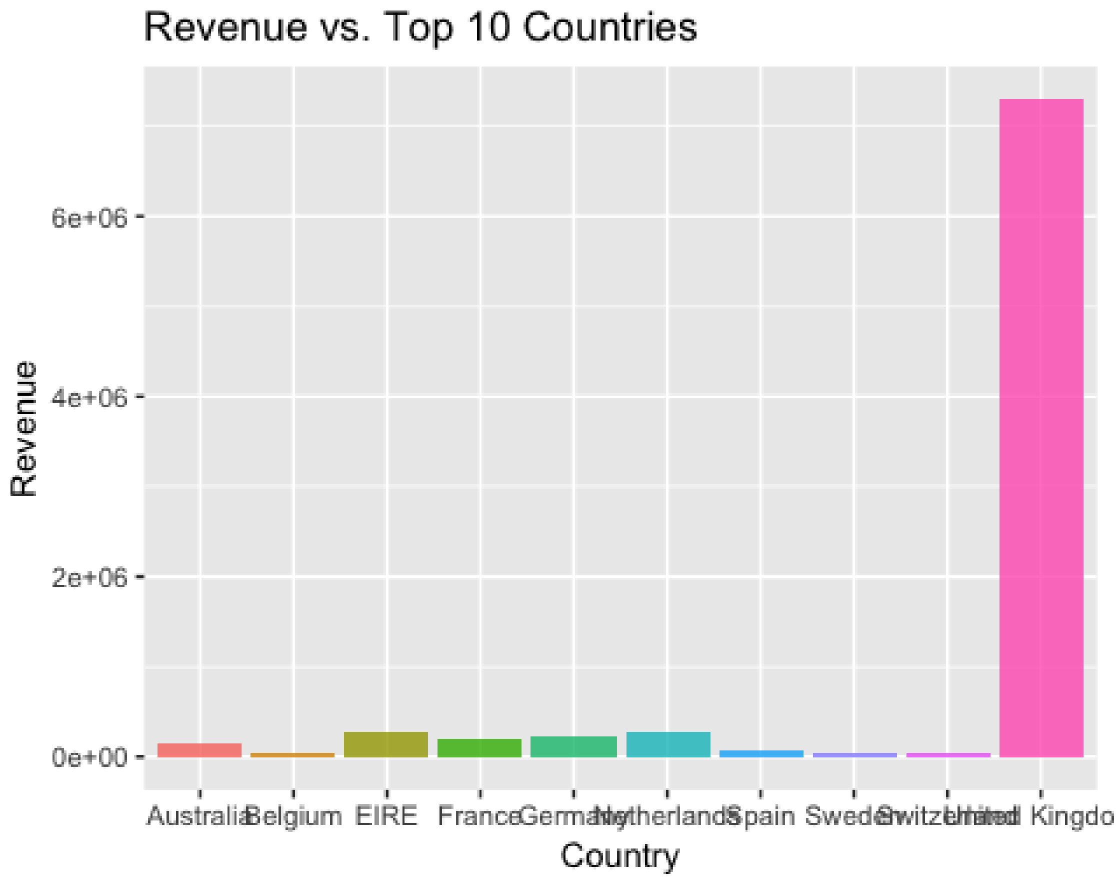
```
#5.Revenue by Hour
data_retail %>%
  group_by(Hour) %>%
  summarise(revenue = sum(amount_spend)) %>%
  ggplot(aes(x=Hour,y=revenue,fill=Hour))+geom_col(alpha=0.8)+
  labs(x = 'Hour in the Day', y = 'Revenue', title = 'Revenue vs. Hour')+
  scale_x_discrete(limits=c("6","7","8","9","10","11","12","13","14","15","16
","17","18","19","20"))+
  guides(fill=FALSE)
```



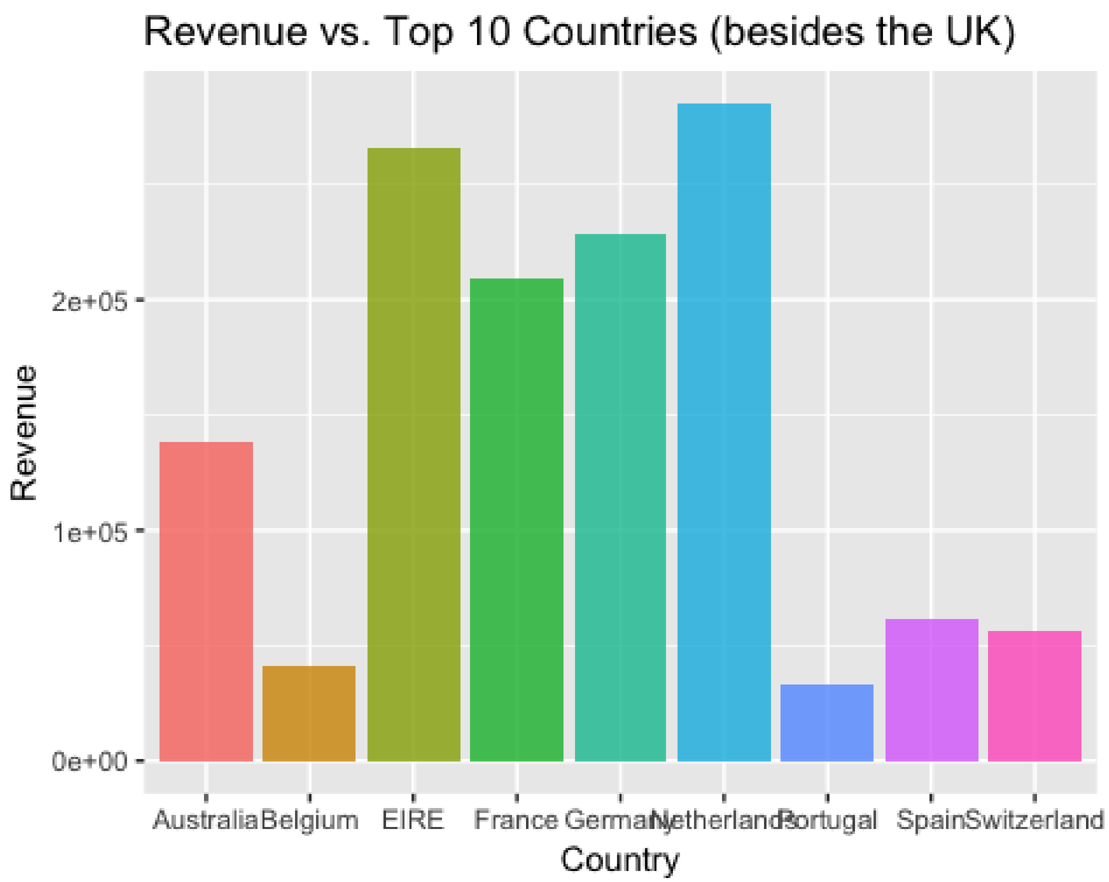
We have more transactions and more revenue in the morning to mid-afternoon. There are some hours missing, especially hours in the evening, which is weird because this is an online retailer.

Country Analysis

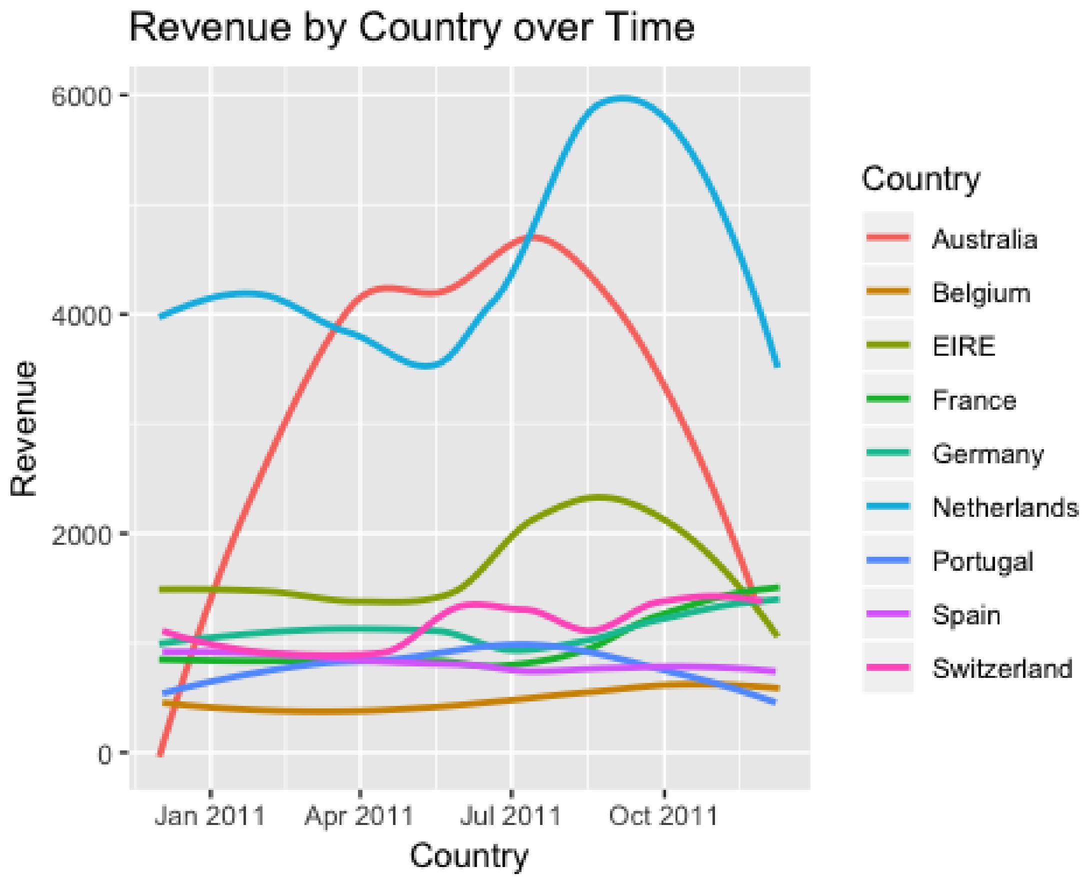
```
length(levels(data_retail$Country)) #38 countries in the history

## [1] 38

#1.Revenue by Country
data_retail %>%
  group_by(Country) %>%
  summarise(revenue = sum(amount_spend)) %>%
  arrange(revenue) %>%
  top_n(10) %>%
  ggplot(aes(x=Country,y=revenue,fill=Country))+geom_col(alpha=0.8)+
  labs(x = 'Country', y = 'Revenue', title = 'Revenue vs. Top 10 Countries')+
  guides(fill=FALSE)

## Selecting by revenue
```



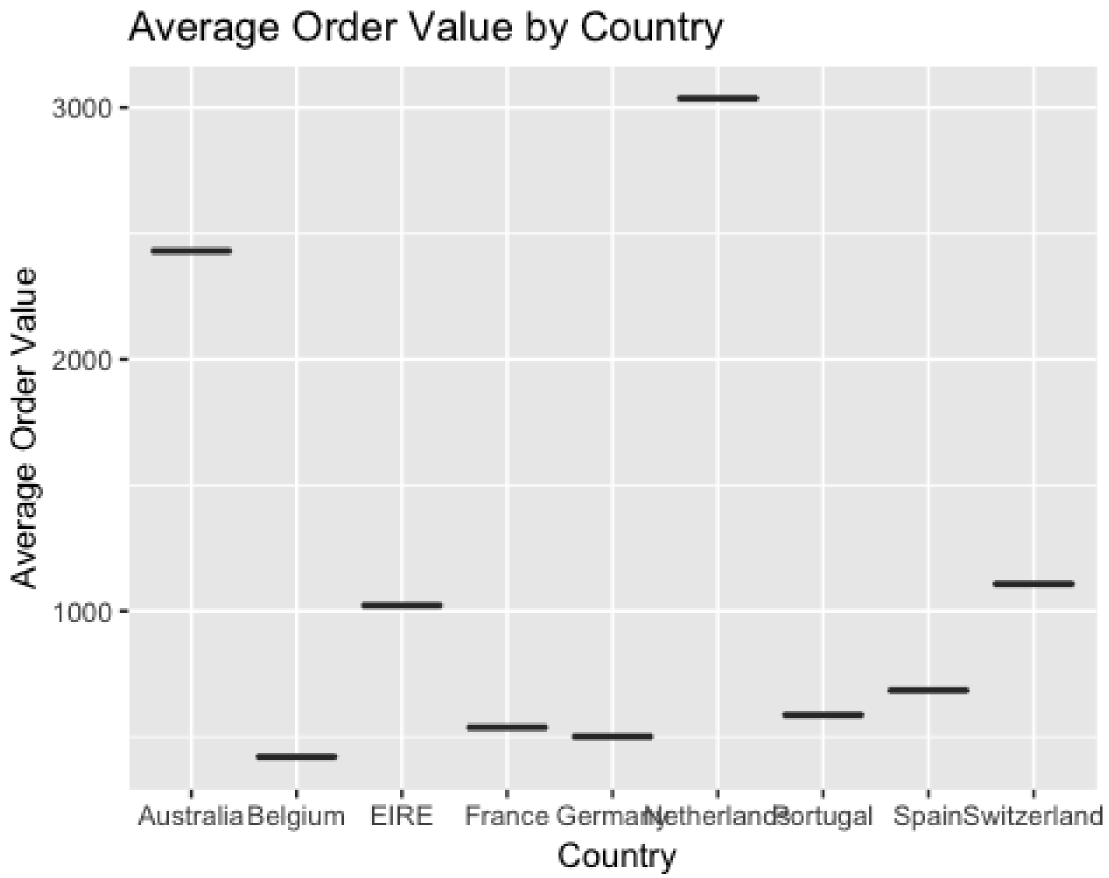
```
data_retail %>%
  filter(Country=="Australia"|Country=="Portugal"|Country=="Switzerland"|Coun
try=="Belgium"|Country=="Netherlands"|Country=="Netherlands"|
           Country=="Spain"|Country=="EIRE"|Country=="France"|Country=="Germa
ny") -> top_country

top_country %>%
  group_by(Country) %>%
  summarise(revenue = sum(amount_spend)) %>%
  ggplot(aes(x=Country,y=revenue,fill=Country))+geom_col(alpha=0.8)+
  labs(x = 'Country', y = 'Revenue', title = 'Revenue vs. Top 10 Countries (b
esides the UK)')+
  guides(fill=FALSE)
```



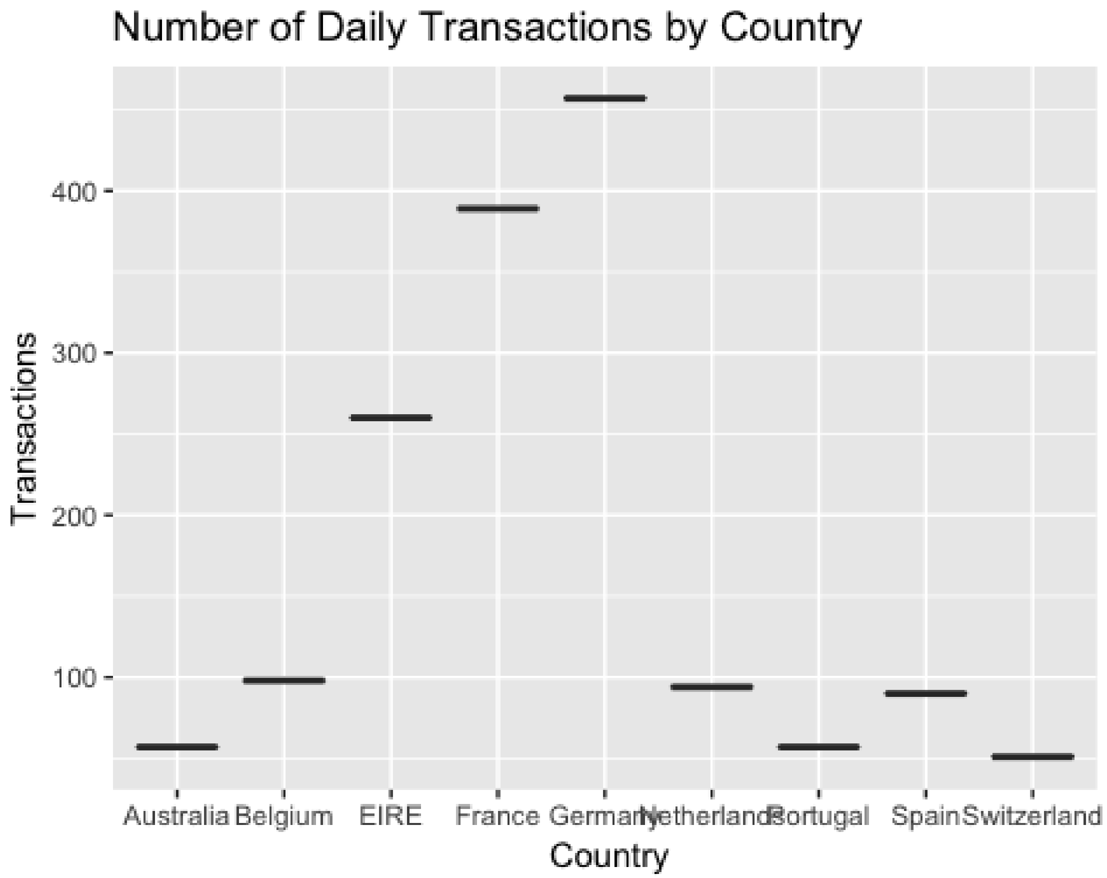
```
#2.Revenue by Country over time
top_country %>%
  group_by(Country,Date) %>%
  summarise(revenue = sum(amount_spend)) %>%
  ggplot(aes(x = Date, y = revenue, colour = Country)) +
  geom_smooth(method = 'auto', se = FALSE) +
  labs(x = 'Country', y = 'Revenue', title = 'Revenue by Country over Time')

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



```
#3.Average Order Value by Country
top_country %>%
  group_by(Country) %>%
  summarise(revenue = sum(amount_spend), transactions = n_distinct(InvoiceNo)
,customer=n_distinct(CustomerID)) %>%
  mutate(aveOrdVal = (round((revenue / transactions),2))) %>%
  ggplot(aes(x = Country, y = aveOrdVal)) +
  geom_boxplot() +
  labs(x = ' Country', y = 'Average Order Value', title = 'Average Order Valu
e by Country')
```



```
#4.Number of Daily Transaction by Country
top_country %>%
  group_by(Country) %>%
  summarise(revenue = sum(amount_spend), transactions = n_distinct(InvoiceNo)
,customer=n_distinct(CustomerID)) %>%
  ggplot(aes(x = Country, y = transactions)) +
  geom_boxplot() +
  labs(x = ' Country', y = 'Transactions', title = 'Number of Daily Transacti
ons by Country')
```



From these country analysis plots(respectively):

–United Kingdom has a dominant place. So the following plot analysis will be conducted without UK to explore more on the other countries, which may be more beneficial to make some conclusion on next marketing actions.

–EIRE, Germany, France,Belgium show an increasing trend during time.

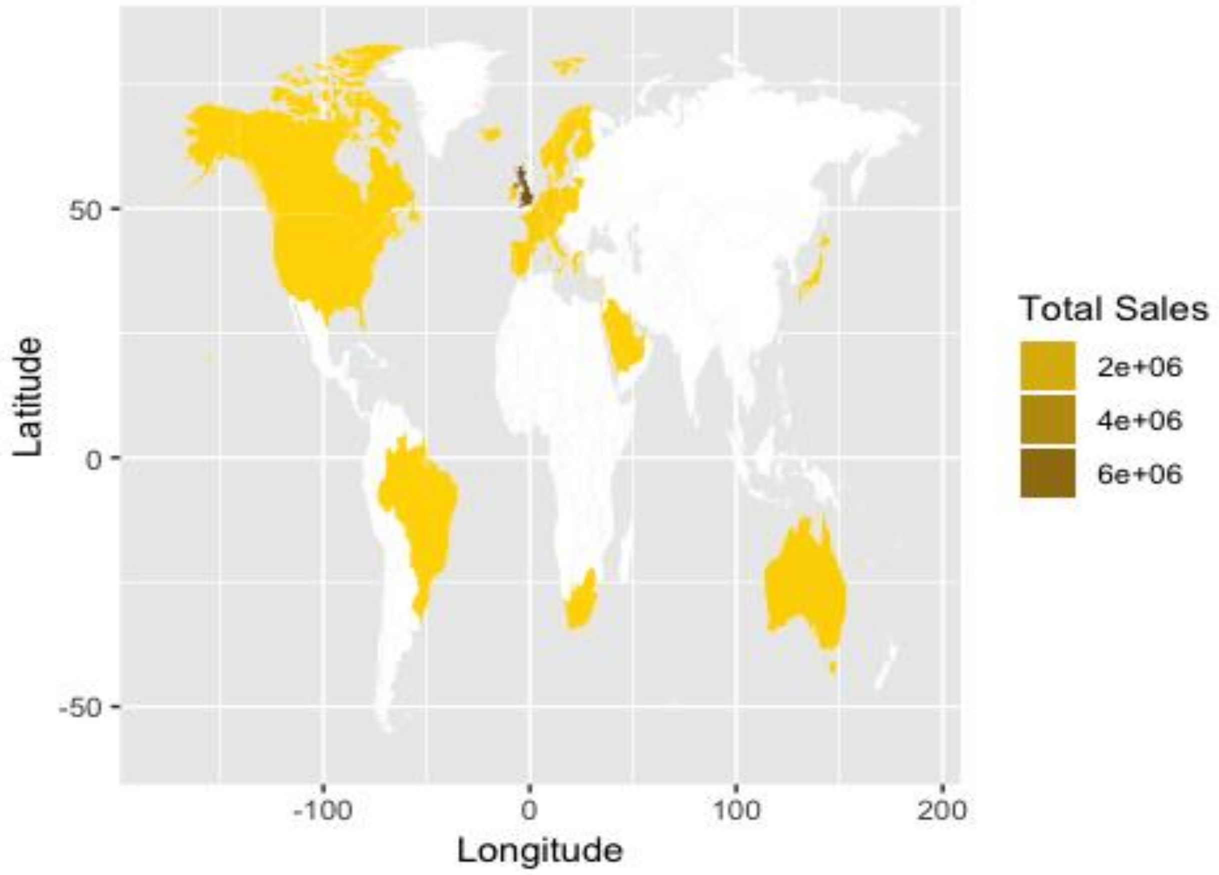
–Switzerland owns the first place in average order value compared to EIRE who owns the second place. But EIRE's second place is supported only by three customers while Switzerland has 51 customers. France and Germany have larger transactions number but lowest aveарage order value.

–Germany and France are the top two countries in transcation number, and EIRE holds the third place. The other top countries are similar in this aspect, all been placed below 100 times of transactions.

Map of amount and number of customer

```
WorldData <- map_data('world')
WorldData %>% filter(region != "Antarctica") -> WorldData
WorldData <- fortify(WorldData)
```

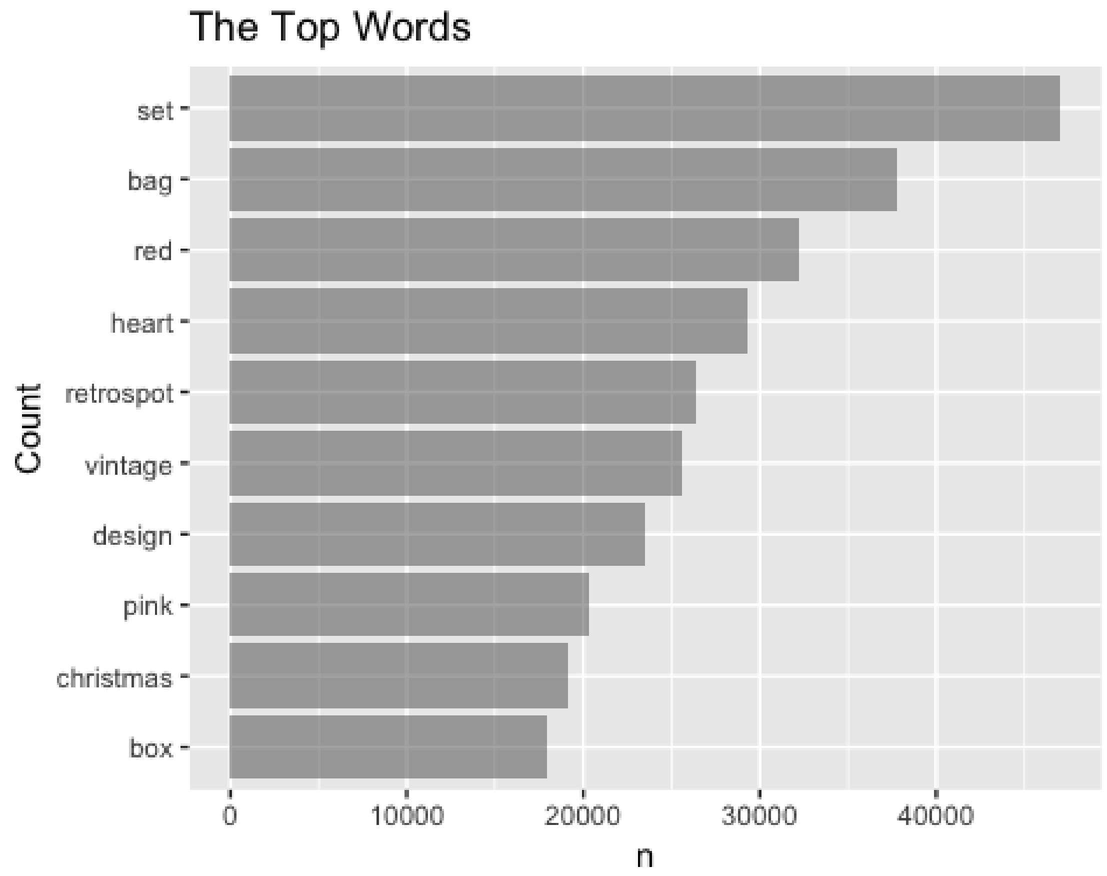
```
#Change EIRE to Ireland, RSA to South Africa, and United Kingdom to UK
countries <- data_retail %>% select(Country) %>% unique()
levels(data_retail$Country)[levels(data_retail$Country)=="United Kingdom"] <-
"UK"
levels(data_retail$Country)[levels(data_retail$Country)=="EIRE"] <- "Ireland"
levels(data_retail$Country)[levels(data_retail$Country)=="RSA"] <- "South Afr
ica"

data_retail %>% group_by(Country) %>% summarise(sum=sum(amount_spend)) %>%
  ggplot() + geom_polygon(data = WorldData,
                          aes(x = long, y = lat,
                              group=group),fill="white")+
  geom_map(map=WorldData,
           aes(fill=sum,
               map_id=Country))+
  scale_fill_gradient(low="gold",high = "goldenrod4")+
  xlab("Longitude")+ylab("Latitude")+
  guides(fill=guide_legend(title="Total Sales"))
```



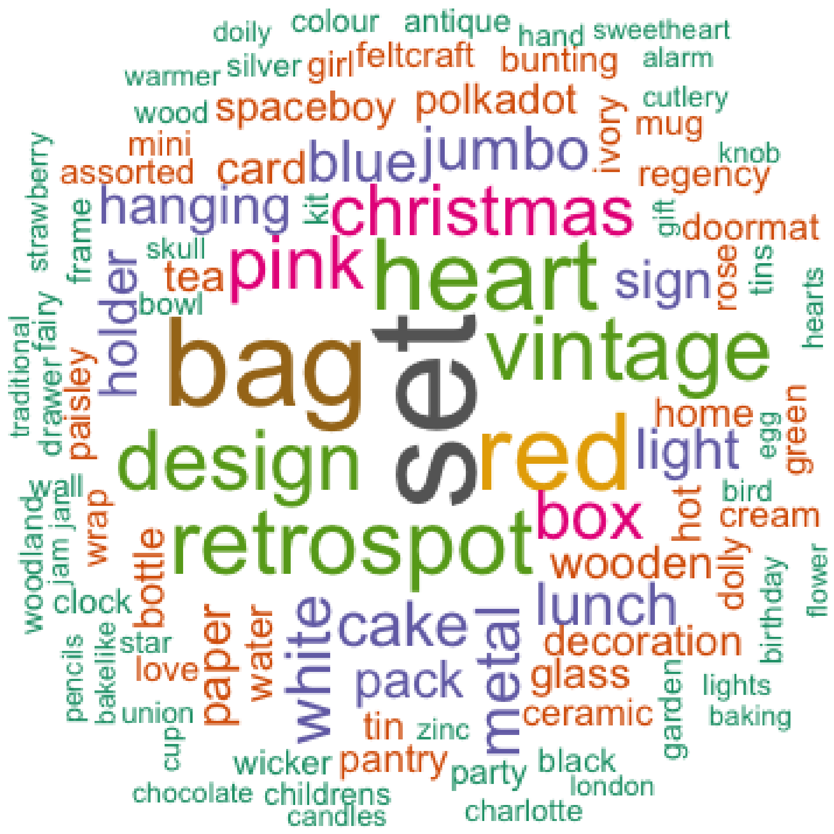
From the map, we can confirm the UK’s dominant place in total sales amount and the other countries’ amounts are far from the UK’s. This company has become worldwide, with sales history in every single continent.

Text Mining on Items Description

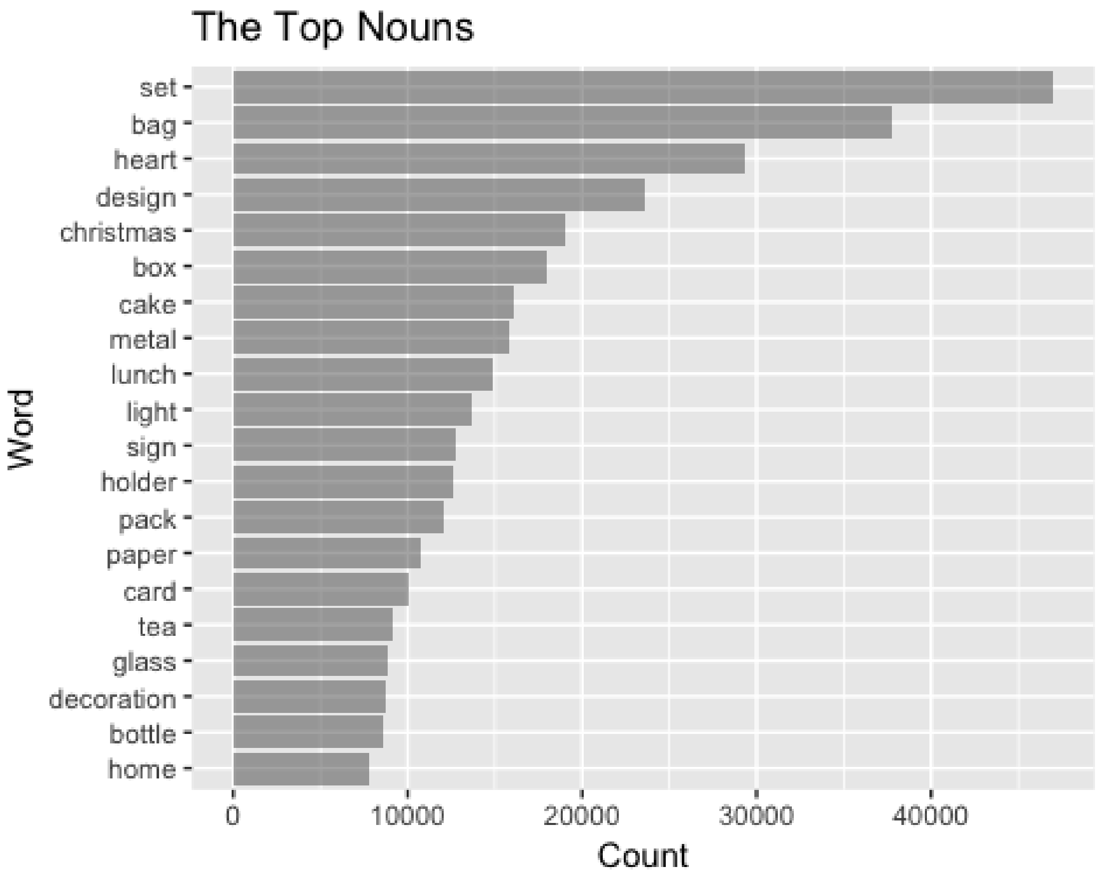
```
#Untoken descriptions
item_desc <- data_retail[,3]
names(item_desc) <- c("text")
item_desc <- as.data.frame(item_desc)
item_desc$text <- as.character(item_desc$item_desc)
item_desc$text <- removeNumbers(item_desc$text) #remove numbers
item_desc <- unnest_tokens(tbl = item_desc, input = text,output = word)

#Remove Stop words
data("stop_words")
item_desc %<>% anti_join(stop_words)

## Joining, by = "word"

#The top words
item_desc %>%
  count(word, sort = TRUE) %>%
  top_n(10) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_col(alpha=0.5) +
  xlab("Count") +
  coord_flip()+
  ggtitle("The Top Words")

## Selecting by n
```



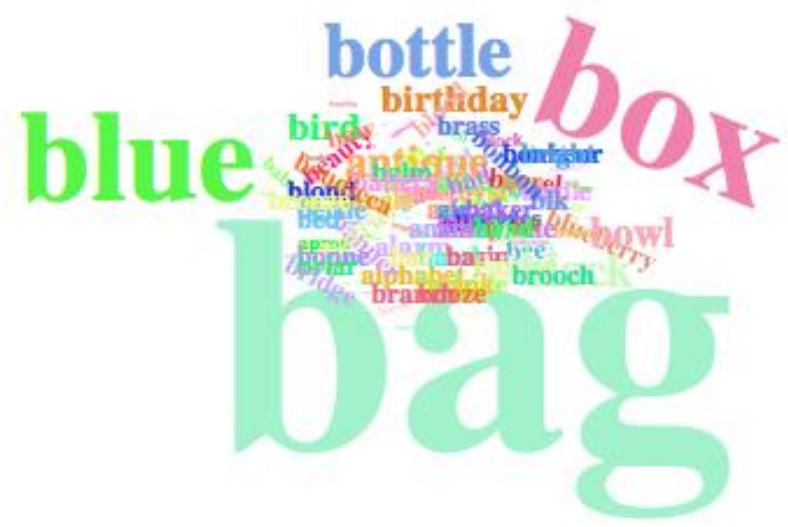
```
#WordCloud for all kinds of words
item_desc %>%
  anti_join(stop_words) %>%
  count(word) %>%
  with(wordcloud(word, n, max.words = 100,
                 random.order=FALSE, rot.per=0.35,
                 colors=brewer.pal(8, "Dark2")))

## Joining, by = "word"
```



```
#Inner join with parts-of-speech to get the nouns in the descriptions
item_desc %>%
  inner_join(parts_of_speech) %>%
  filter(pos=="Noun") %>%
  filter(word!="pink" & word!="white" & word!="red" & word!="vintage" & word!
="jumbo" & word!="blue"& word!="wooden") %>% #delete some top words that are
not nouns
  count(word) %>%
  top_n(20) %>%
  ggplot()+
  aes(x=reorder(word,n),y=n) +
  geom_col(alpha=0.5)+
  coord_flip()+
  ylab("Count")+
  xlab("Word")+
  ggtitle("The Top Nouns")

## Joining, by = "word"

## Selecting by n
```



```
#WordCloud for nouns: I want to see what are the most popular items sold?
word_count <- item_desc %>%
  inner_join(parts_of_speech) %>%
  filter(pos=="Noun") %>%
  count(word)

## Joining, by = "word"

wordcloud2(word_count,color="random-light",rotateRatio = 0.3)
```



The wordclouds show some top-sellers: bag, box, bottle, alarm, bowl, etc. I join speech dictionary to filter out nouns, in order to see the most popular sold items.

Marketing Analysis

I then will proceed customer clustering based on recency-frequency-monetary analysis.

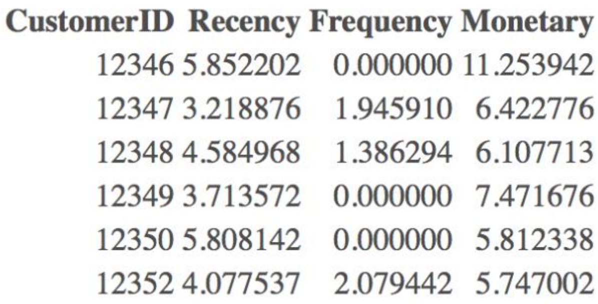
Recency: how recent a customer has purchased

Frequency: how often they purchase

Monetary: how much the customer spends

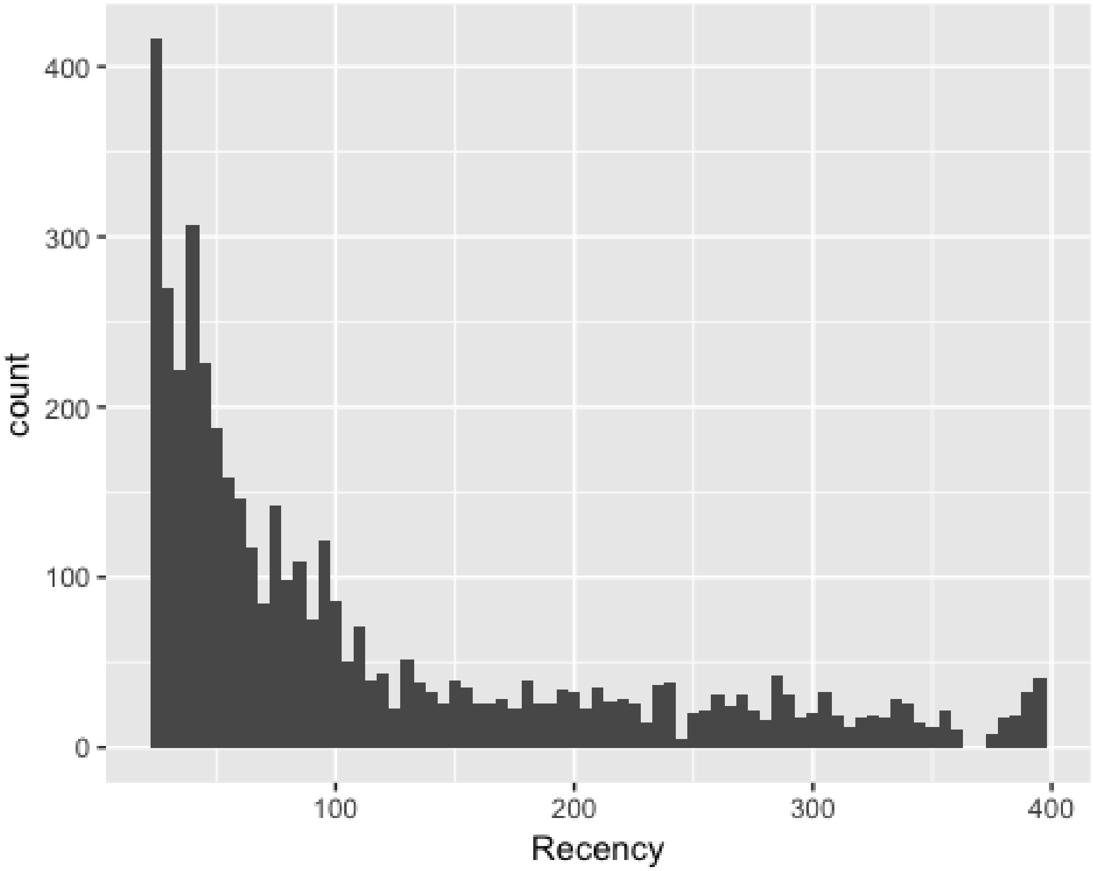
RFM(Recency-Frequency-Monetary) Analysis

```
RFM <- data_retail %>%
  group_by(CustomerID) %>%
  summarise(Recency=as.numeric(as.Date("2012-01-01")-max(Date)),
            Frequency=n_distinct(InvoiceNo),
            Monetary= round(sum(amount_spend)/n_distinct(InvoiceNo),2))
kable(head(RFM))
```

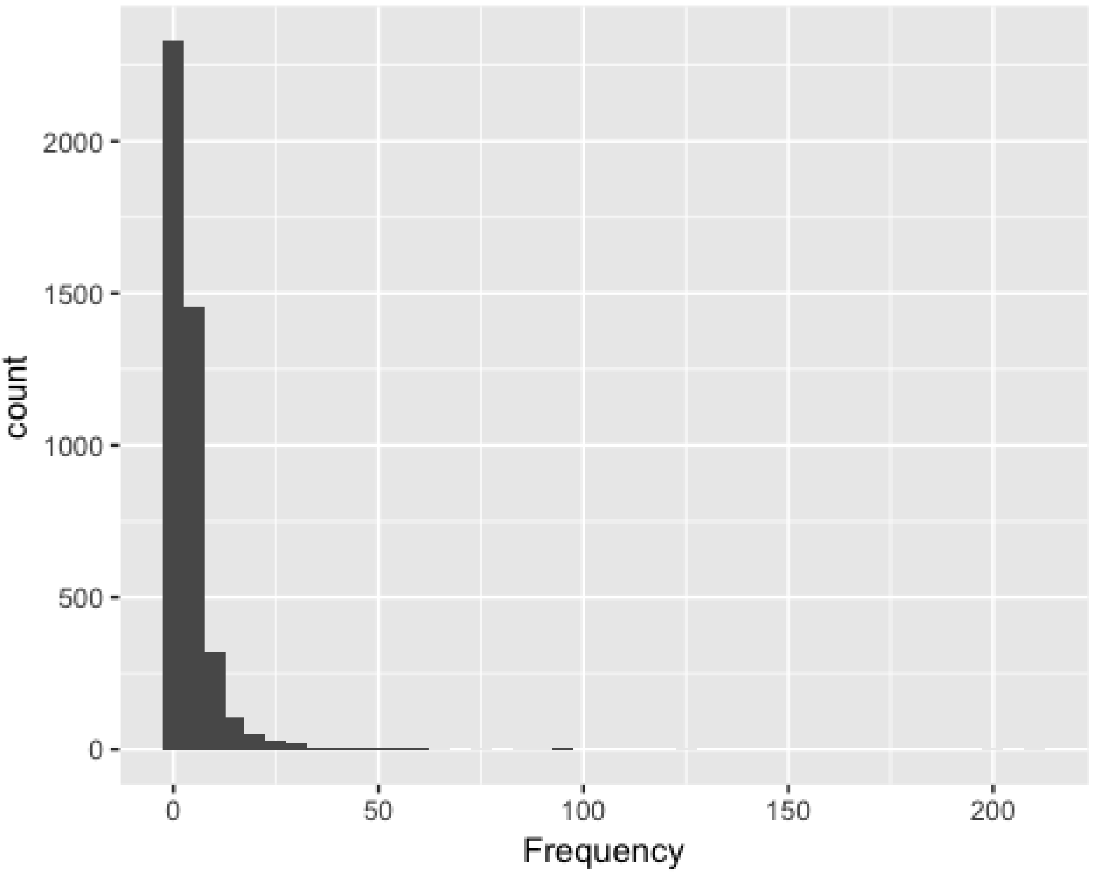


CustomerID	Recency	Frequency	Monetary
12346	5.852202	0.000000	11.253942
12347	3.218876	1.945910	6.422776
12348	4.584968	1.386294	6.107713
12349	3.713572	0.000000	7.471676
12350	5.808142	0.000000	5.812338
12352	4.077537	2.079442	5.747002

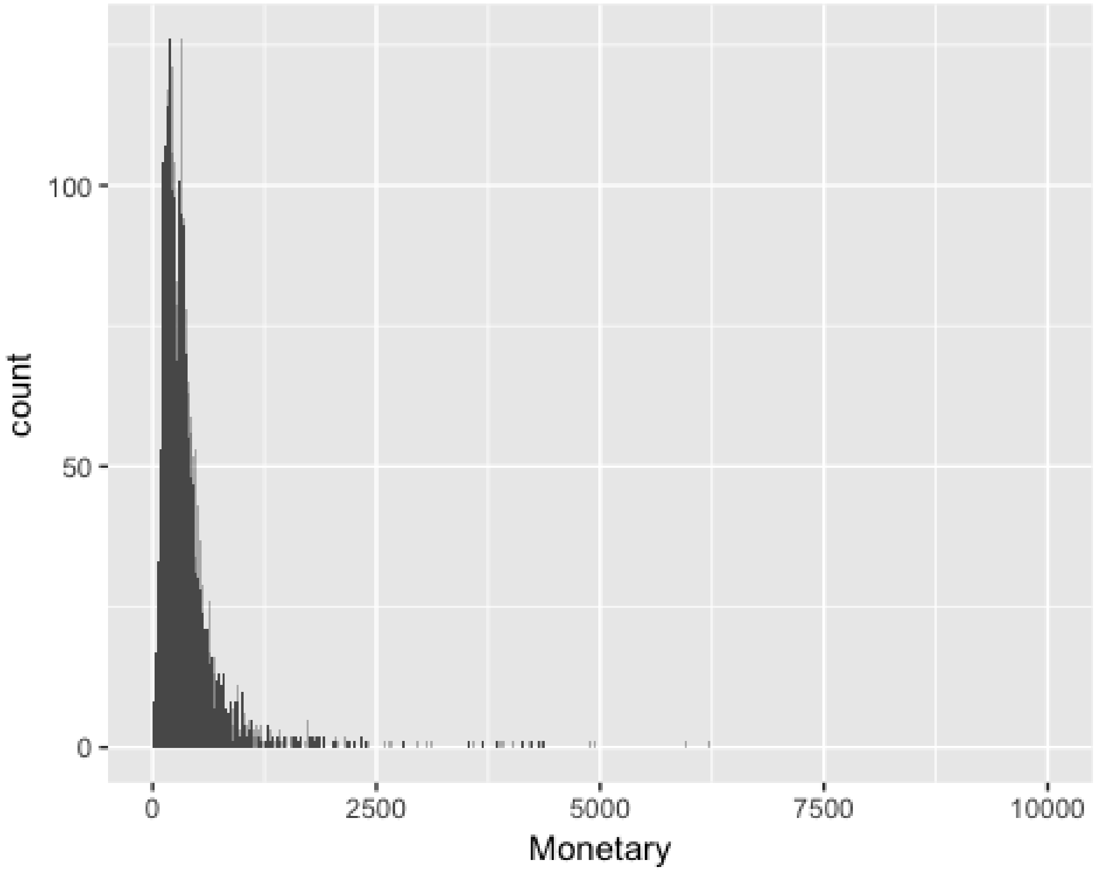
```
#Visualize R/F/M distribution
ggplot(RFM)+aes(x=Recency)+geom_histogram(binwidth = 5)
```



```
ggplot(RFM)+aes(x=Frequency)+geom_histogram(binwidth = 5)
```



```r
ggplot(RFM)+aes(x=Monetary)+geom_histogram(binwidth = 10)+coord_cartesian(xlim = c(0,10000)) #Zoom into 0~10000 to have a closer look.
```

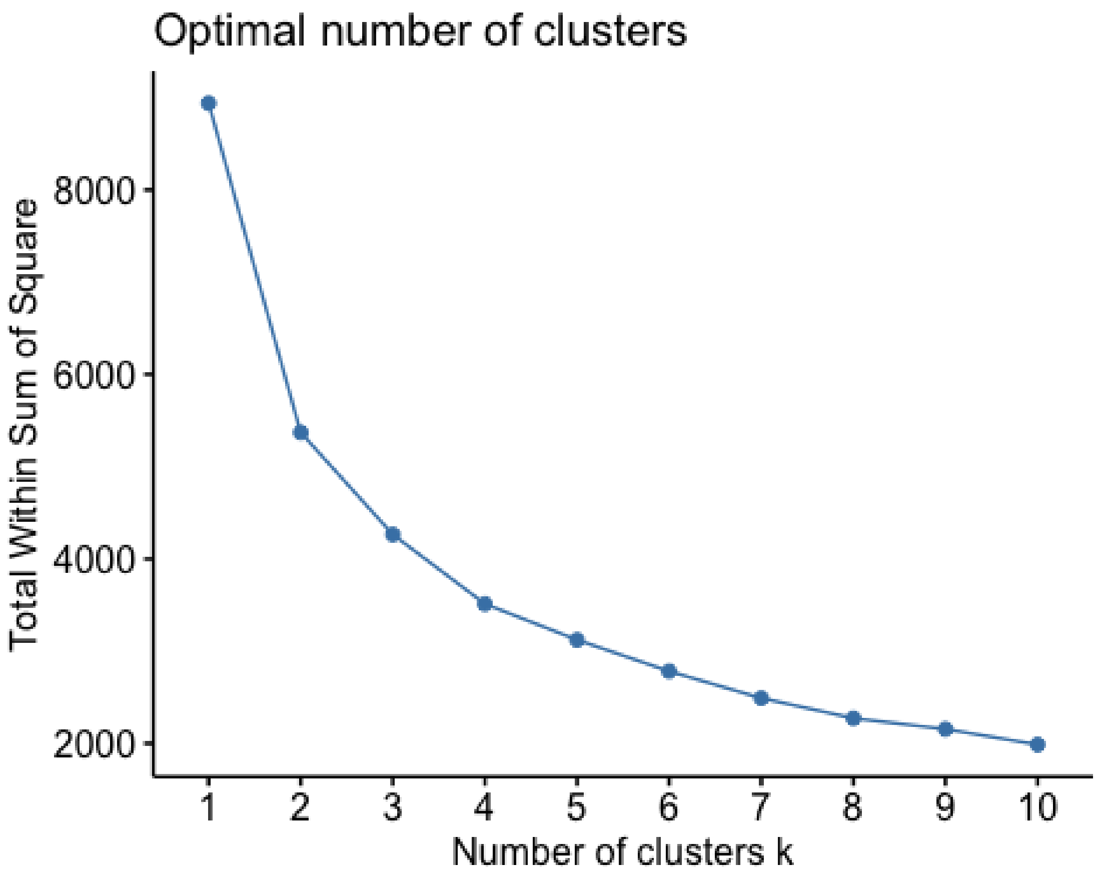


```
#These three amounts' distribution seems like Laplace Distribution's right si
de and they have similar distributions. Each of them decays fast as the value
increases.
```

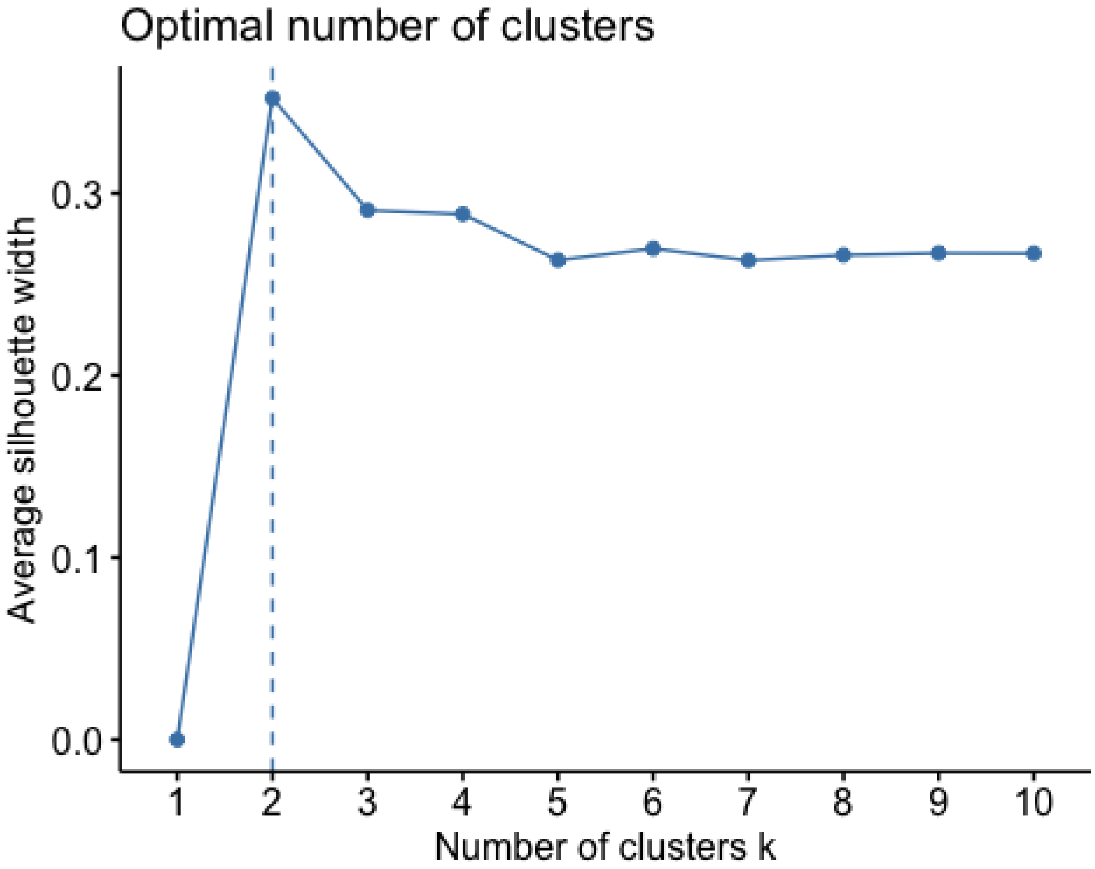
Customer Segmentation based on RFM

With RFM dataset, segment the customers into groups (number of groups t.b.d) using KMeans.

```
set.seed(2018)
#To have a better cluster result, first transform RFM values into log scale.
RFM$Recency <- log(RFM$Recency)
RFM$Frequency <- log(RFM$Frequency)
RFM$Monetary <- log(RFM$Monetary)
#Determining Optimal Clusters: Elbow Method
fviz_nbclust(RFM[,2:4], kmeans, method = "wss")
```

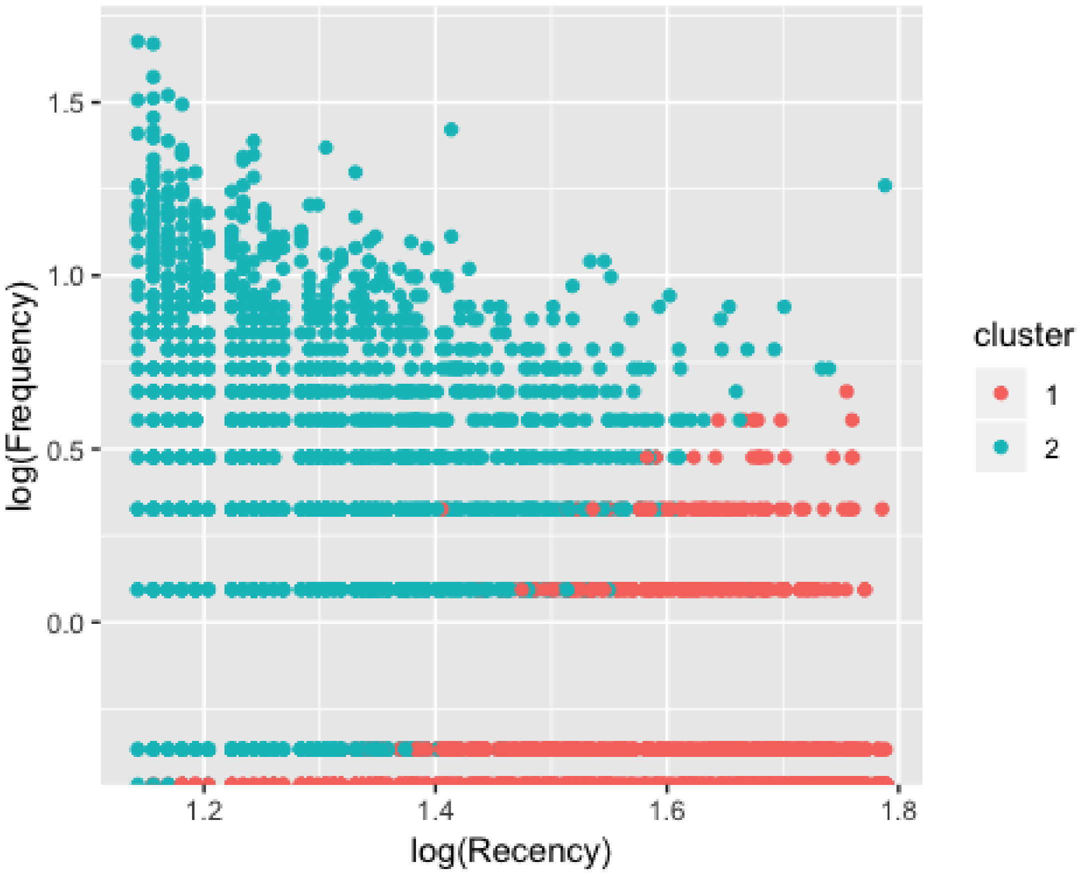


```
fviz_nbclust(RFM[,2:4], kmeans, method = "silhouette")
```

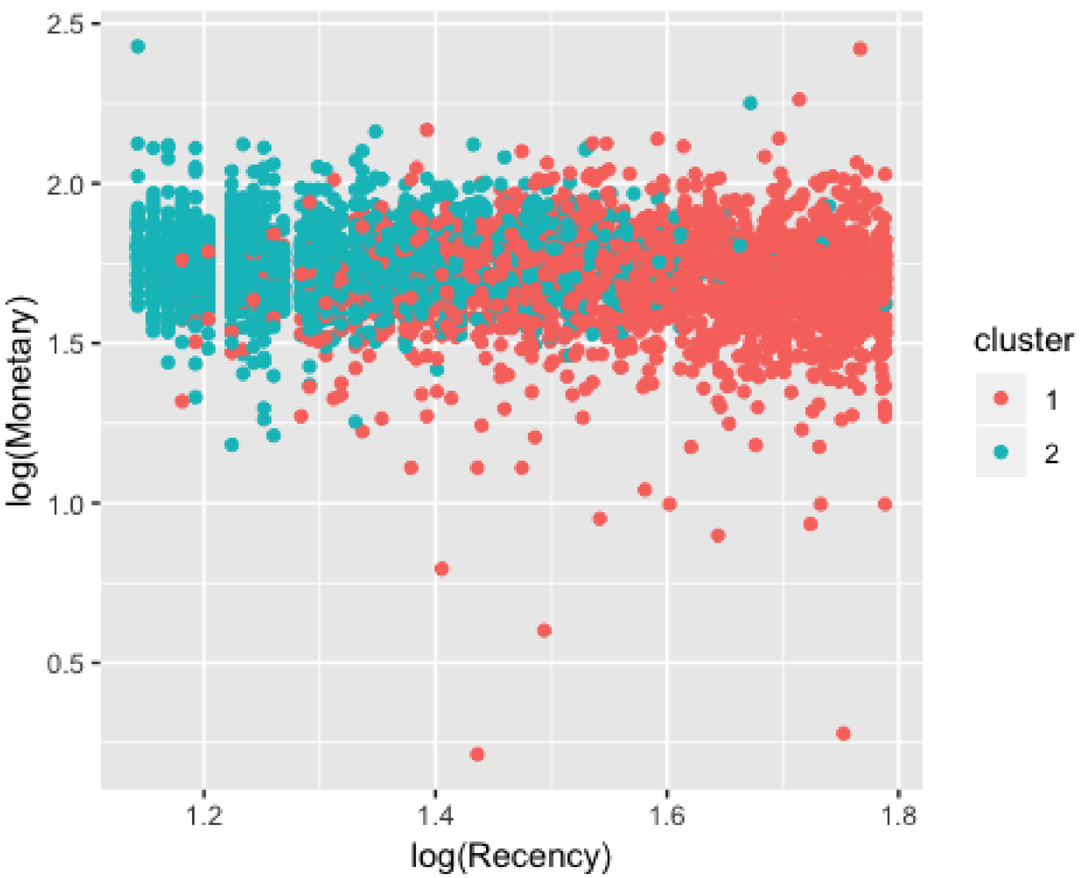


The wss method indicates that 2 is a changing point which determine the optimal k for Kmeans. So I've implemented 2Means for the RFM analysis.

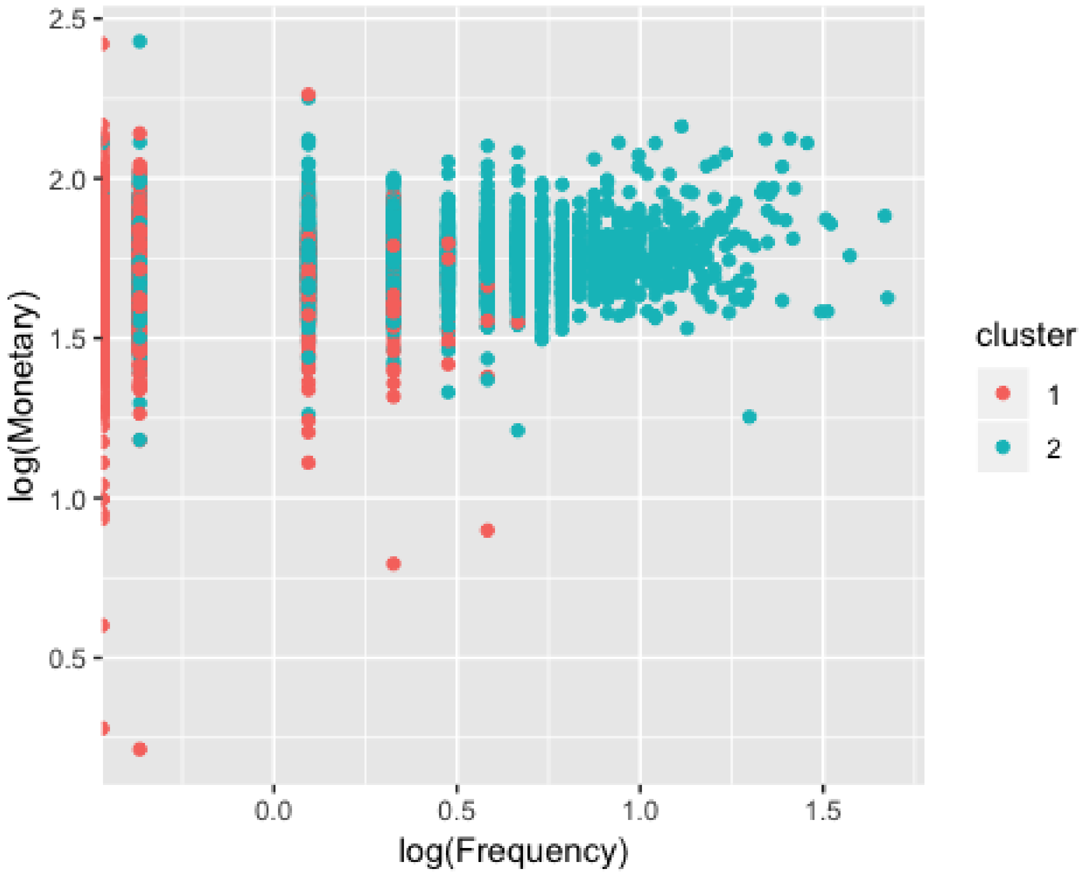
```
#Taking number of group equals 2
km <- kmeans(select(RFM,Recency,Frequency,Monetary),2,nstart=10)
RFM %<>% mutate(cluster=as.factor(km$cluster))

ggplot(RFM)+aes(x=log(Recency),y=log(Frequency),color=cluster)+geom_point()
```



```
ggplot(RFM)+aes(x=log(Recency),y=log(Monetary),color=cluster)+geom_point()
```



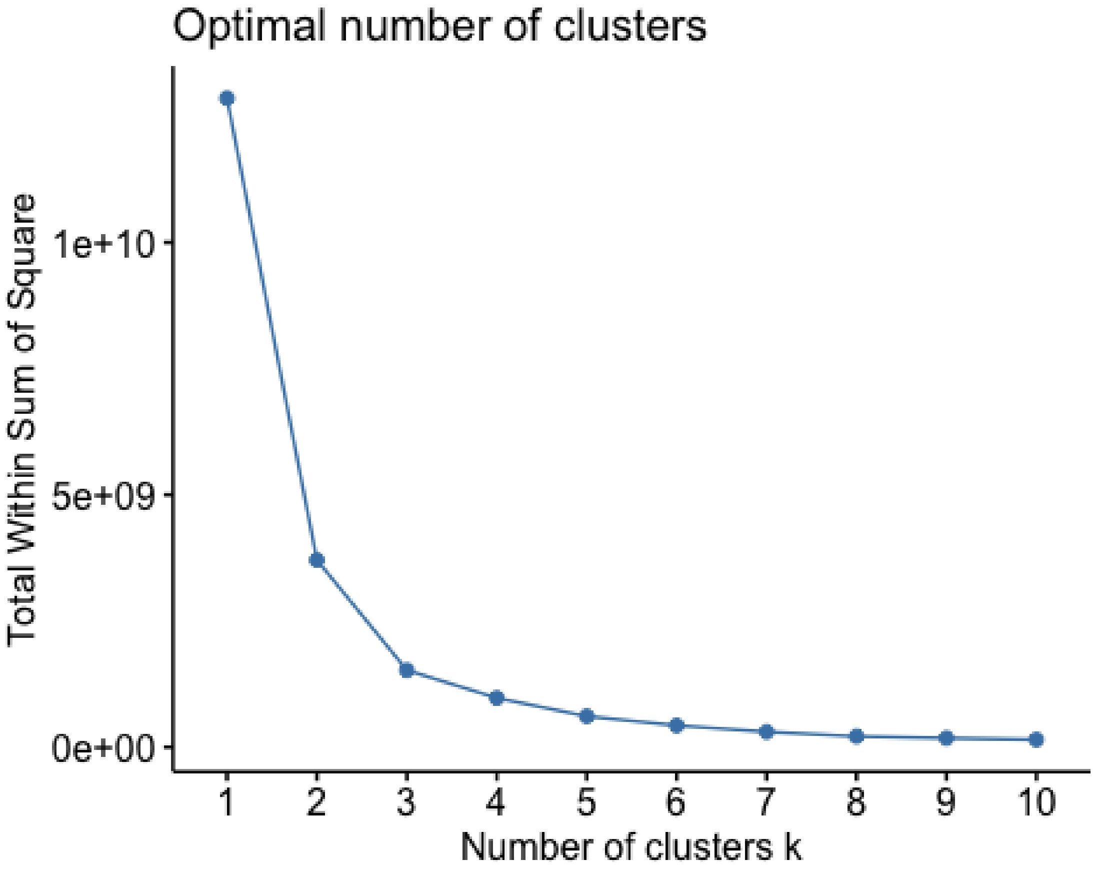
```
ggplot(RFM)+aes(x=log(Frequency),y=log(Monetary),color=cluster)+geom_point()
```



```
#This KMeans result is weird, with only two observations in the second group.
#So I will try another method of segmentation.
```

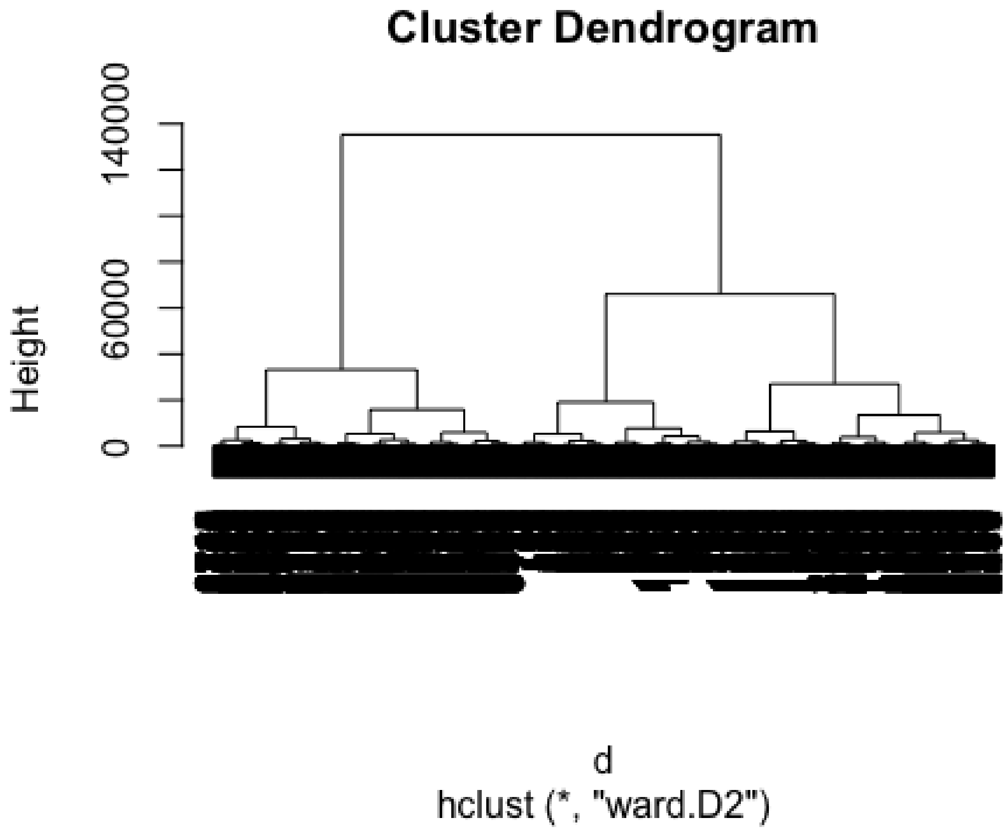
With RFM dataset, segment the customers into groups (number of groups t.b.d) using Hierarchical Cluster.

```
set.seed(2018)
#Create a new clustering table
RFM2 <- RFM
#Determining Optimal Clusters: Elbow Method
fviz_nbclust(RFM2, FUN = hcut, method = "wss")
```



The wss method indicates that 2 is a changing point which determine the optimal k for HCluster.

```
d <- dist(RFM2)
c <- hclust(d, method = 'ward.D2')
plot(c)
```



```
#Aggregate the information into a table
members <- cutree(c,k = 2)
members[1:5]

## [1] 1 1 1 1 1

table(members)

## members
##    1    2 
## 2652 1686
```

Grouping the data into 2 clusters. We have approximately 3:1 in two member groups.

Modeling

I've divided the model data into train set and test set.

```
data_model <- data_retail %>%
  group_by(CustomerID,amount_spend) %>%
  summarise(Recency=as.numeric(as.Date("2012-01-01")-max(Date)),
            Frequency=n_distinct(InvoiceNo),
            Monetary= round(sum(amount_spend)/n_distinct(InvoiceNo),2))
```

```r
#Divide the data
set.seed(2018)
test_index <- sort(sample(nrow(data_model), nrow(data_model)*.2))
data_test <- data_model[test_index,]
data_train <- data_model[-test_index,]

#Fit a linear model
model <- lm(log(amount_spend+1)~log(Recency+1)+log(Frequency+1)+log(Monetary+
1),data=data_train)
summary(model)

##
## Call:
## lm(formula = log(amount_spend + 1) ~ log(Recency + 1) + log(Frequency +
##     1) + log(Monetary + 1), data = data_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8365 -0.2035  0.1558  0.2429  1.0865
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.105208   0.008930   11.78   <2e-16 ***
## log(Recency + 1)      0.018156   0.001509   12.03   <2e-16 ***
## log(Frequency + 1) -0.205897   0.003314  -62.12   <2e-16 ***
## log(Monetary + 1)    0.910812   0.001071  850.29   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.388 on 114064 degrees of freedom
## Multiple R-squared:  0.8648, Adjusted R-squared:  0.8648
## F-statistic: 2.433e+05 on 3 and 114064 DF,  p-value: < 2.2e-16

#plot(x=fitted(model),y=model$residuals)
```

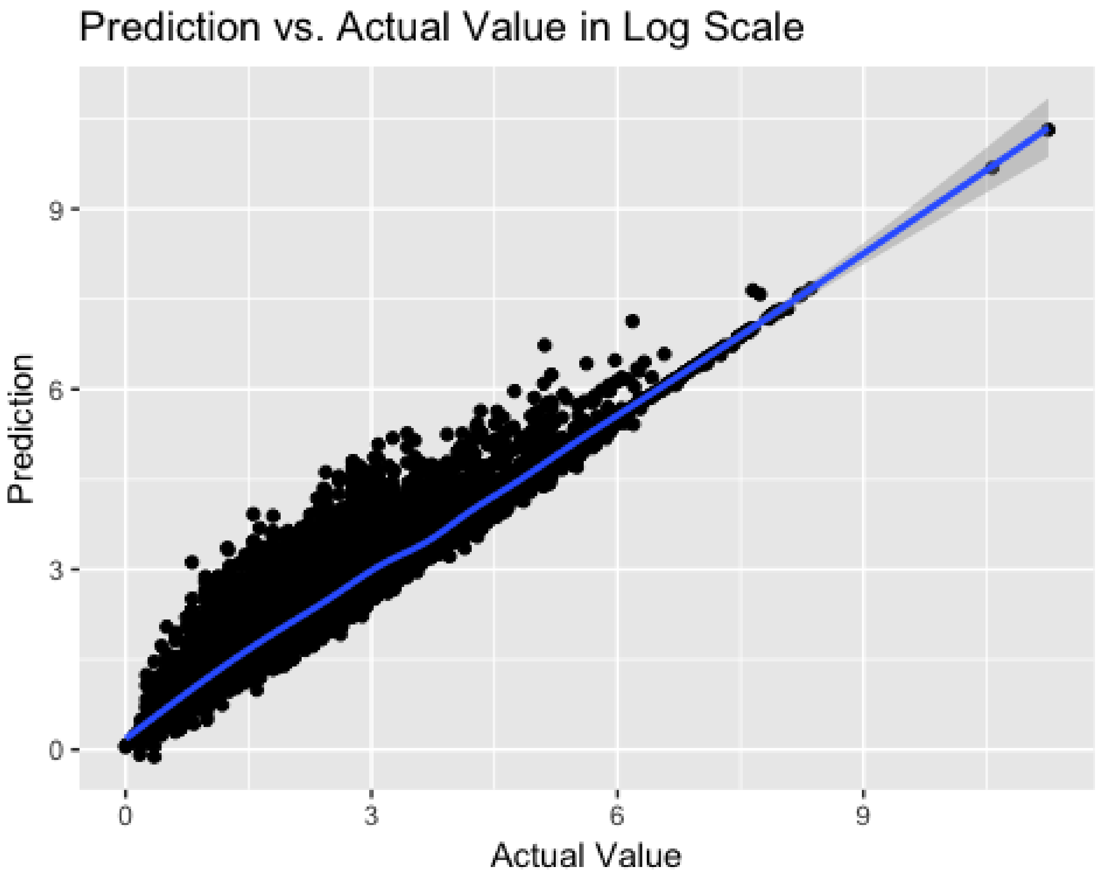
The linear model is written as follow:

$$log(Amount + 1) = 0.105 + 0.018log(Recency + 1) - 0.206log(Frequency + 1) + 0.911log(Monetary + 1)$$

I’ve implemented a classic linear model to explain the total spend for each customer, with their RFM values. I’ve done log transformations for each of them in order to adjust the right-skewness. This model is fairly good at first look, with a high R-square value. But as for residual plot, a clear pattern is shown.

```r
#Check the model's accuracy with test set
prediction <- predict(model,newdata = data_test[,3:5])
ggplot(data_test)+aes(x=log(amount_spend+1),y=prediction)+
  ggtitle("Prediction vs. Actual Value in Log Scale")+
  xlab("Actual Value")+ylab("Prediction")+
  geom_point()+geom_smooth(se=T,formula = y~x)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



It seems that a large amount of points from 0 to 6 are not well predicted, and the points above 6 are relatively well predicted.