Online Retail Analytics

Setting default values to get a clean output

```
knitr::opts_chunk$set(message = FALSE)
knitr::opts_chunk$set(warning = FALSE)
```

Loading all the required packages

```
library("VIM")
library("ISLR")
library("caret")
library("class")
library("e1071")
library("ggplot2")
library("corrplot")
library("dplyr")
```

Setting working directory and loading data

```
setwd("/Users/sampathnikhilkumar/Desktop")
data.df <- read.csv("Online_Retail.csv")
```

1.Show the breakdown of the number of transactions by countries

```
data_country <- as.data.frame(table(data.df$Country))

data_country$Percentage <- data_country$Freq/nrow(data.df) * 100

colnames(data_country) <- c("Country", "Count", "Percentage")

data_country[data_country$Percentage > 1,]
```

```
##           Country  Count Percentage
## 11           EIRE   8196   1.512431
## 14         France   8557   1.579047
## 15        Germany   9495   1.752139
## 36 United Kingdom 495478  91.431956
```

Countries accounting for more than 1% of the total transactions are EIRE, France, Germany and United Kingdom.

2. Adding new attribute "TransactionValue" which is the product of Quantity and UnitPrice

```r
data.df$TransactionValue <- data.df$Quantity * data.df$UnitPrice
```

By adding this new attribute we can now calculate the value of the transactions based on our requirement.

3. Using the newly created variable, TransactionValue, showing the breakdown of transaction values by countries with total transaction exceeding 130,000 British Pound

```r
data.df %>% select(TransactionValue,Country) %>% group_by(Country) %>% summarise(Total = sum(Transactio
```

```
## # A tibble: 6 x 2
##   Country          Total
##   <chr>            <dbl>
## 1 United Kingdom 8187806.
## 2 Netherlands     284662.
## 3 EIRE            263277.
## 4 Germany         221698.
## 5 France          197404.
## 6 Australia       137077.
```

There are total 6 countries where the transaction value exceeds 130,000 British Pound and the highest among them is "United Kingdom".

4. Converting Invoice Date into a POSIXlt object

```r
Temp=strptime(data.df$InvoiceDate,format='%m/%d/%Y %H:%M',tz='GMT')
head(Temp)
```

```
## [1] "2010-12-01 08:26:00 GMT" "2010-12-01 08:26:00 GMT"
## [3] "2010-12-01 08:26:00 GMT" "2010-12-01 08:26:00 GMT"
## [5] "2010-12-01 08:26:00 GMT" "2010-12-01 08:26:00 GMT"
```

```r
#New_Invoice_Date
data.df$New_Invoice_Date <- as.Date(Temp)

data.df$New_Invoice_Date[20000]- data.df$New_Invoice_Date[10]
```

```
## Time difference of 8 days
```

```r
#Invoice_Day_Week
data.df$Invoice_Day_Week= weekdays(data.df$New_Invoice_Date)

#New_Invoice_Hour
data.df$New_Invoice_Hour = as.numeric(format(Temp, "%H"))

#New_Invoice_Month
data.df$New_Invoice_Month = as.numeric(format(Temp, "%m"))
```

4(a). Percentage of transactions (by numbers) by days of the week

```r
data.df %>% group_by(Invoice_Day_Week) %>% summarise(count=n()) %>% mutate(percentage=count/nrow(data.d
```

```
## # A tibble: 6 x 3
##   Invoice_Day_Week  count percentage
##   <chr>             <int>      <dbl>
## 1 Friday            82193       15.2
## 2 Monday            95111       17.6
## 3 Sunday            64375       11.9
## 4 Thursday         103857       19.2
## 5 Tuesday          101808       18.8
## 6 Wednesday         94565       17.5
```

4(b). Percentage of transactions (by transaction volume) by days of the week

```r
data.df %>% group_by(Invoice_Day_Week) %>% summarise(Total = sum(TransactionValue)) %>% mutate(Percenta
```

```
## # A tibble: 6 x 3
##   Invoice_Day_Week    Total Percentage
##   <chr>               <dbl>      <dbl>
## 1 Friday           1540611.      15.8
## 2 Monday           1588609.      16.3
## 3 Sunday            805679.       8.27
## 4 Thursday         2112519.      21.7
## 5 Tuesday          1966183.      20.2
## 6 Wednesday        1734147.      17.8
```

4(c). Percentage of transactions (by transaction volume) by month of the year

```r
data.df %>% group_by(New_Invoice_Month) %>% summarise(Total = sum(TransactionValue)) %>% mutate(Percent
```

```
## # A tibble: 12 x 3
##    New_Invoice_Month    Total Percentage
##                <dbl>    <dbl>      <dbl>
##  1                 1  560000.       5.74
##  2                 2  498063.       5.11
##  3                 3  683267.       7.01
##  4                 4  493207.       5.06
##  5                 5  723334.       7.42
##  6                 6  691123.       7.09
##  7                 7  681300.       6.99
##  8                 8  682681.       7.00
##  9                 9 1019688.      10.5
## 10                10 1070705.      11.0
## 11                11 1461756.      15.0
## 12                12 1182625.      12.1
```

4(d). The date with the highest number of transactions from Australia

```r
data.df %>% filter(Country =="Australia") %>% group_by(New_Invoice_Date) %>% summarise(Total_Count = n(
```

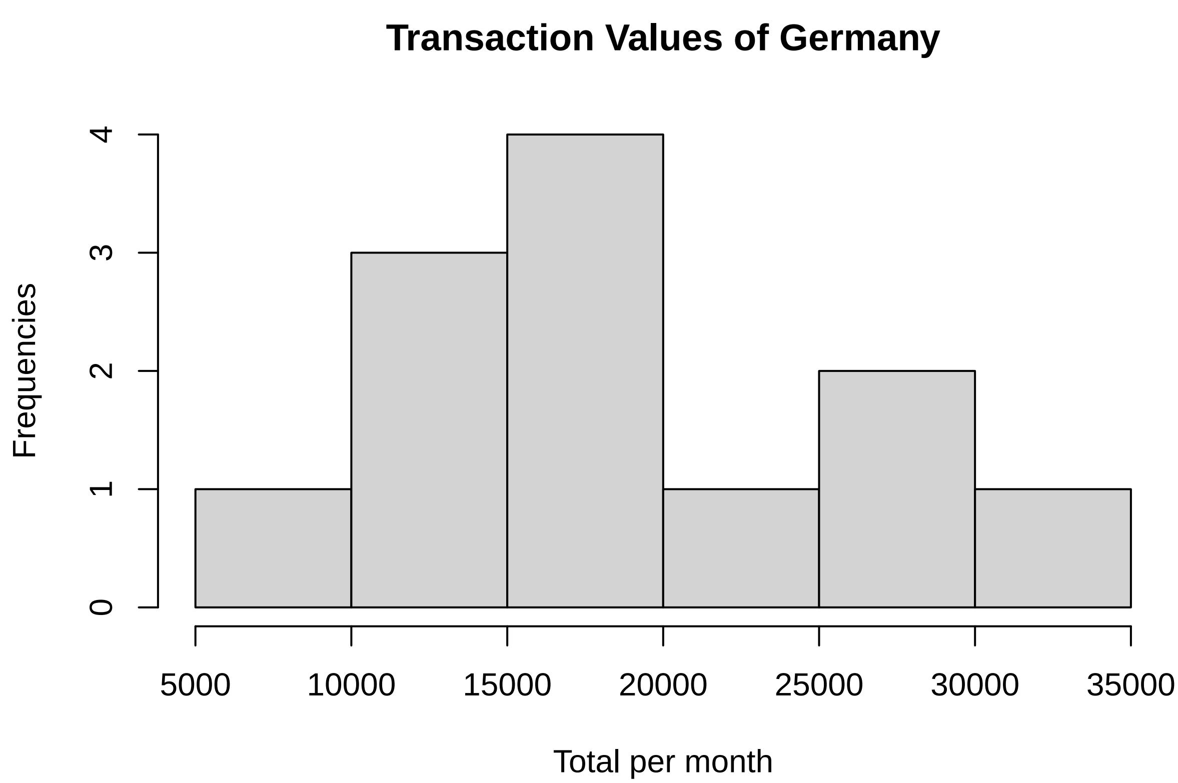
```
## # A tibble: 49 x 2
##    New_Invoice_Date Total_Count
##    <date>                 <int>
##  1 2011-06-15               139
##  2 2011-07-19               137
##  3 2011-08-18                97
##  4 2011-03-03                84
##  5 2011-10-05                82
##  6 2011-05-17                73
##  7 2011-02-15                69
##  8 2011-01-06                48
##  9 2011-07-14                35
## 10 2011-09-16                34
## # ... with 39 more rows
```

As we can see from above on 2011-06-15 Australia has recorded the highest number of transactions i.e. 139 Transactions.

5. Plot the histogram of transaction values from Germany

```
Germany <- data.df %>% filter(Country == "Germany") %>% group_by(New_Invoice_Month) %>% summarise(Total=

hist(Germany$Total, main = "Transaction Values of Germany", xlab="Total per month", ylab="Frequencies")
```



6(a). Customer who had highest number of transactions

```
data.df %>% group_by(CustomerID) %>% select(CustomerID) %>% filter(!is.na(CustomerID)) %>% summarise(n_
```

```
## # A tibble: 4,372 x 2
##    CustomerID n_count
##         <int>   <int>
##  1      17841    7983
##  2      14911    5903
##  3      14096    5128
##  4      12748    4642
##  5      14606    2782
##  6      15311    2491
##  7      14646    2085
##  8      13089    1857
##  9      13263    1677
## 10      14298    1640
## # ... with 4,362 more rows
```

The CustomerID 17841 had the highest number of transactions amongst others with a total of 7983 transactions.

6(b). Most valuable customer with the highest total sum of transactions

```
data.df %>% group_by(CustomerID) %>% select(CustomerID, TransactionValue) %>% filter(!is.na(CustomerID))
```

```
## # A tibble: 4,372 x 2
##    CustomerID Spending_max
##         <int>        <dbl>
##  1      14646      279489.
##  2      18102      256438.
##  3      17450      187482.
##  4      14911      132573.
##  5      12415      123725.
##  6      14156      113384.
##  7      17511       88125.
##  8      16684       65892.
##  9      13694       62653.
## 10      15311       59419.
## # ... with 4,362 more rows
```

The CustomerID 14646 is the most valuable customer with the highest spending sum of 279,489.020 British Sterling Pound.

7. Percentage of missing values for each variable in the dataset

```
colMeans(is.na(data.df)*100)
```

```
##          InvoiceNo          StockCode        Description           Quantity
##            0.00000            0.00000            0.00000            0.00000
##        InvoiceDate          UnitPrice         CustomerID            Country
##            0.00000            0.00000           24.92669            0.00000
```

```
##  TransactionValue  New_Invoice_Date  Invoice_Day_Week  New_Invoice_Hour
##           0.00000           0.00000           0.00000           0.00000
## New_Invoice_Month
##           0.00000
```

We can observe that CustomerID is the only attribute with 24.9266% of NAs in the entire dataset.

8. The number of transactions with missing CustomerID records by Countries

```r
data.df %>% filter(is.na(CustomerID)) %>% group_by(Country) %>% count()
```

```
## # A tibble: 9 x 2
## # Groups:   Country [9]
##   Country              n
##   <chr>            <int>
## 1 Bahrain              2
## 2 EIRE               711
## 3 France              66
## 4 Hong Kong          288
## 5 Israel              47
## 6 Portugal            39
## 7 Switzerland        125
## 8 United Kingdom 133600
## 9 Unspecified        202
```

There are in total 8 countries and 1 unspecified country in the entire dataset which has NA values in them amongst these United Kingdom is the country with highest NA records of 133,600 rows.

9. On average, how often the costumers comeback to the website for their next shopping?

```r
Diff_Days <- data.df %>% select(CustomerID,New_Invoice_Date) %>% group_by(CustomerID) %>% distinct(New_I

Diff_Days
```

```
## # A tibble: 15,200 x 3
## # Groups:   CustomerID [2,992]
##    CustomerID New_Invoice_Date Days_Between
##         <int> <date>           <drtn>
##  1      18287 2011-10-12       143 days
##  2      18287 2011-10-28        16 days
##  3      18283 2011-01-23        17 days
##  4      18283 2011-02-28        36 days
##  5      18283 2011-04-21        52 days
##  6      18283 2011-05-23        32 days
##  7      18283 2011-06-14        22 days
##  8      18283 2011-06-23         9 days
##  9      18283 2011-07-14        21 days
## 10      18283 2011-09-05        53 days
## # ... with 15,190 more rows
```

```
mean(Diff_Days$Days_Between)
```

```
## Time difference of 38.4875 days
```

On an average approximately for every 38 days customers come back to the website for their next shopping.

10. Return rate of goods purchased by the customers from France

```
France_Cancel <- data.df %>% filter(Country=="France",Quantity<0) %>% count()

France_Total <- data.df %>% filter(Country=="France") %>% count()

Return_Percentage_France <- France_Cancel/France_Total*100
Return_Percentage_France
```

```
##          n
## 1 1.741264
```

The return rate of customers who made purchases in France is 1.741264%.

11. The product that has generated the highest revenue for the retailer

```
data.df %>% select(StockCode,TransactionValue) %>% group_by(StockCode) %>%  summarise(Total=sum(Transac
```

```
## # A tibble: 4,070 x 2
##    StockCode   Total
##    <chr>       <dbl>
##  1 DOT       206245.
##  2 22423     164762.
##  3 47566      98303.
##  4 85123A     97894.
##  5 85099B     92356.
##  6 23084      66757.
##  7 POST       66231.
##  8 22086      63792.
##  9 84879      58960.
## 10 79321      53768.
## # ... with 4,060 more rows
```

The product with the StockCode as "DOT" is the one which has generated highest revenue to the retailer i.e. 206,245.48 British Sterling Pound.

12. Unique Customers in the dataset

```
data.df %>% select(CustomerID) %>% unique() %>% count()
```

```
##      n
## 1 4373
```

In total there are 4,373 unique customers in the dataset.