Business Analytics Assignment

Ankith Dasu

R Markdown

##Read the Csv and create a data frame

```
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(e1071)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ISLR)
library(caret)


getwd()
```

```
## [1] "/Users/ankithdasu/Desktop/Spring 2022/Business Analytics/Assignment 1"
```

```
setwd("/Users/ankithdasu/Desktop/Spring 2022/Business Analytics/Assignment 1")
Online_Retail <- read.csv("Online_Retail.csv")

Online_Retail$Country   = as.factor(Online_Retail$Country)
Online_Retail$Quantity = as.numeric(Online_Retail$Quantity)

summary(Online_Retail)
```

```
##   InvoiceNo          StockCode         Description          Quantity
##  Length:541909      Length:541909      Length:541909      Min.   :-80995.00
##  Class :character   Class :character   Class :character   1st Qu.:     1.00
##  Mode  :character   Mode  :character   Mode  :character   Median :     3.00
##                                                           Mean   :     9.55
##                                                           3rd Qu.:    10.00
##                                                           Max.   : 80995.00
##
##  InvoiceDate          UnitPrice           CustomerID
##  Length:541909      Min.   :-11062.06   Min.   :12346
##  Class :character   1st Qu.:     1.25   1st Qu.:13953
##  Mode  :character   Median :     2.08   Median :15152
##                     Mean   :     4.61   Mean   :15288
##                     3rd Qu.:     4.13   3rd Qu.:16791
##                     Max.   : 38970.00   Max.   :18287
##                                         NA's   :135080
##           Country
##  United Kingdom:495478
##  Germany       :  9495
##  France        :  8557
##  EIRE          :  8196
##  Spain         :  2533
##  Netherlands   :  2371
##  (Other)       : 15279
```

1. Show the breakdown of the number of transactions by countries i.e. how many transactions are in the dataset for each country (consider all records including cancelled transactions). Show this in total number and also in percentage. Show only countries accounting for more than 1% of the total transactions.

```
summary(Online_Retail$Country)
```

```
##            Australia              Austria              Bahrain
##                 1259                  401                   19
##              Belgium               Brazil               Canada
##                 2069                   32                  151
##      Channel Islands               Cyprus       Czech Republic
##                  758                  622                   30
##              Denmark                 EIRE   European Community
##                  389                 8196                   61
##              Finland               France              Germany
##                  695                 8557                 9495
##               Greece            Hong Kong              Iceland
##                  146                  288                  182
```

```
##               Israel                Italy                Japan
##                  297                  803                  358
##              Lebanon            Lithuania               Malta
##                   45                   35                  127
##          Netherlands               Norway               Poland
##                 2371                 1086                  341
##             Portugal                  RSA         Saudi Arabia
##                 1519                   58                   10
##            Singapore                Spain               Sweden
##                  229                 2533                  462
##          Switzerland United Arab Emirates       United Kingdom
##                 2002                   68               495478
##          Unspecified                  USA
##                  446                  291
```

```
Countries_Count <- table(Online_Retail$Country)
prop.table(Countries_Count) # We need to know the country values as a whole of countries
```

```
##
##            Australia              Austria              Bahrain
##         2.323268e-03         7.399766e-04         3.506124e-05
##              Belgium               Brazil               Canada
##         3.817984e-03         5.905050e-05         2.786446e-04
##      Channel Islands               Cyprus       Czech Republic
##         1.398759e-03         1.147794e-03         5.535985e-05
##              Denmark                 EIRE   European Community
##         7.178327e-04         1.512431e-02         1.125650e-04
##              Finland               France              Germany
##         1.282503e-03         1.579047e-02         1.752139e-02
##               Greece            Hong Kong              Iceland
##         2.694179e-04         5.314545e-04         3.358497e-04
##               Israel                Italy                Japan
##         5.480625e-04         1.481799e-03         6.606275e-04
##              Lebanon            Lithuania               Malta
##         8.303977e-05         6.458649e-05         2.343567e-04
##          Netherlands               Norway               Poland
##         4.375273e-03         2.004027e-03         6.292569e-04
##             Portugal                  RSA         Saudi Arabia
##         2.803054e-03         1.070290e-04         1.845328e-05
##            Singapore                Spain               Sweden
##         4.225802e-04         4.674217e-03         8.525417e-04
##          Switzerland United Arab Emirates       United Kingdom
##         3.694347e-03         1.254823e-04         9.143196e-01
##          Unspecified                  USA
##         8.230164e-04         5.369905e-04
```

```
Percentage_Transaction <- round(100*prop.table(Countries_Count),digits = 3) #prop.table is used to roun
Percent_Table <- cbind(Countries_Count,Percentage_Transaction) # We get the transaction value of each c
Value <- subset(Percent_Table,Percentage_Transaction>1)
```

2. Create a new variable 'TransactionValue' that is the product of the exising 'Quantity' and 'UnitPrice' variables. Add this variable to the dataframe. (5 marks)

```
#TransactionValue <- Quantity * UnitPrice
#creating new variable. Variable here refers to the column
library(dplyr)
TransactionValue = Online_Retail$Quantity * Online_Retail$UnitPrice
Online_Retail$TransactionValue <- TransactionValue #Assigning it to the dataframe as Transactionvalue i
summary(TransactionValue)
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -168469.60       3.40       9.75      17.99      17.40  168469.60
```

3. Using the newly created variable, TransactionValue, show the breakdown of transaction values by countries i.e. how much money in total has been spent each country. Show this in total sum of transaction values. Show only countries with total transaction exceeding 130,000 British Pound. (10 marks)

```
EachCountryTransaction <-
  group_by(Online_Retail,Country)%>%summarize(transactionvalue=sum(TransactionValue))
EachCountryTransaction
```

```
## # A tibble: 38 x 2
##    Country         transactionvalue
##    <fct>                      <dbl>
##  1 Australia                137077.
##  2 Austria                   10154.
##  3 Bahrain                     548.
##  4 Belgium                   40911.
##  5 Brazil                     1144.
##  6 Canada                     3666.
##  7 Channel Islands           20086.
##  8 Cyprus                    12946.
##  9 Czech Republic              708.
## 10 Denmark                   18768.
## # ... with 28 more rows
```

```
TransactionAbove130 <- filter(EachCountryTransaction,transactionvalue >130000)
TransactionAbove130
```

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

4. This is an optional question which carries additional marks (golden questions). In this question, we are dealing with the InvoiceDate variable. The variable is read as a categorical when you read data from the file. Now we need to explicitly instruct R to interpret this as a Date variable. "POSIXlt" and "POSIXct" are two powerful object classes in R to deal with date and time. Click here for more information. First let's convert 'InvoiceDate' into a POSIXlt object:

```
Temp=strptime(Online_Retail$InvoiceDate,format='%m/%d/%Y %H:%M',tz='GMT')

Online_Retail$New_Invoice_Date <- as.Date(Temp)
Online_Retail$New_Invoice_Date[20000]- Online_Retail$New_Invoice_Date[10]
```

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

```
Online_Retail$Invoice_Day_Week= weekdays(Online_Retail$New_Invoice_Date)
Online_Retail$New_Invoice_Hour = as.numeric(format(Temp, "%H"))
Online_Retail$New_Invoice_Month = as.numeric(format(Temp, "%m"))
Online_Retail$New_Invoice_Year = as.numeric(format(Temp,"%y"))
```

a) Show the percentage of transactions (by numbers) by days of the week (extra 2 marks)

```
# For transactions of days of the weeks, frequencyof the days is calculated and divide by Quantity. Per

Total_Transaction = length(Online_Retail$TransactionValue)
Total_Transaction # Total of 541909 Transaction
```

```
## [1] 541909
```

```
Days <- table(Online_Retail$Invoice_Day_Week)
Days
```

```
##
##    Friday    Monday    Sunday  Thursday   Tuesday Wednesday
##     82193     95111     64375    103857    101808     94565
```

```
sum(Days)
```

```
## [1] 541909
```

```
#Sunday=64375
Sunday = 64375
Sunday_Percent= Sunday / Total_Transaction

#Monday=95111
Monday = 95111
Monday_Percent = Monday / Total_Transaction

#Tuesday=101808
Tuesday = 101808
Tuesday_Percent = Tuesday / Total_Transaction

#Wednesday=94565
Wednesday = 94565
Wednesday_Percent = Wednesday / Total_Transaction

#Thursday = 103857
```

```
Thursday = 103857
Thursday_Percent = Thursday / Total_Transaction

#Friday = 82193
Friday = 82193
Friday_Percent = Friday / Total_Transaction


Days_Percent <- data.frame(Sunday_Percent,Monday_Percent,Tuesday_Percent,Wednesday_Percent,Thursday_Perc
Days_Percent
```

```
##   Sunday_Percent Monday_Percent Tuesday_Percent Wednesday_Percent
## 1       0.118793       0.175511       0.1878692         0.1745035
##   Thursday_Percent Friday_Percent
## 1        0.1916503      0.1516731
```

b) Show the percentage of transactions (by transaction volume) by days of the week

```
# The Transaction volume ( Products per order ) is calculated individually for every day.

TransVol <- Online_Retail %>% select(Invoice_Day_Week,Quantity) %>% filter(Invoice_Day_Week == "Sunday"
Sunday_sum <- sum(TransVol$Quantity)


TransVol <- Online_Retail %>% select(Invoice_Day_Week,Quantity) %>% filter(Invoice_Day_Week == "Monday"
Monday_sum <- sum(TransVol$Quantity)


TransVol <- Online_Retail %>% select(Invoice_Day_Week,Quantity) %>% filter(Invoice_Day_Week == "Tuesday
Tuesday_sum <- sum(TransVol$Quantity)


TransVol <- Online_Retail %>% select(Invoice_Day_Week,Quantity) %>% filter(Invoice_Day_Week == "Wednesda
Wednesday_sum <- sum(TransVol$Quantity)


TransVol <- Online_Retail %>% select(Invoice_Day_Week,Quantity) %>% filter(Invoice_Day_Week == "Thursday
Thursday_sum <- sum(TransVol$Quantity)


TransVol <- Online_Retail %>% select(Invoice_Day_Week,Quantity) %>% filter(Invoice_Day_Week == "Friday"
Friday_sum <- sum(TransVol$Quantity)

data.frame(Sunday_sum,Monday_sum,Tuesday_sum,Wednesday_sum,Thursday_sum,Friday_sum)  # we can get a com
```

```
##   Sunday_sum Monday_sum Tuesday_sum Wednesday_sum Thursday_sum Friday_sum
## 1     467732     815354      961543        969558      1167823     794440
```

```
# Percentage of transaction, dividing the day sum by the total transaction per week, we get individual

Transaction_percent_week = sum(Online_Retail$Quantity)
```

```
Sunday_Vol = Sunday_sum / Transaction_percent_week
Monday_Vol = Monday_sum / Transaction_percent_week
Tuesday_Vol = Tuesday_sum / Transaction_percent_week
Wednesday_Vol = Wednesday_sum / Transaction_percent_week
Thursday_Vol = Thursday_sum / Transaction_percent_week
Friday_Vol = Friday_sum / Transaction_percent_week

data.frame(Transaction_percent_week,Sunday_Vol,Monday_Vol,Tuesday_Vol,Wednesday_Vol,Thursday_Vol,Friday
```

```
##   Transaction_percent_week Sunday_Vol Monday_Vol Tuesday_Vol Wednesday_Vol
## 1                  5176450 0.09035768  0.1575122   0.1857534     0.1873017
##   Thursday_Vol Friday_Vol
## 1    0.2256031   0.153472
```

c) Show the percentage of transactions (by transaction volume) by month of the year

```
#Monthly transactions can be achieved by taking Invoice month, Year and also Quantity
Transaction_Month <- Online_Retail %>% select(New_Invoice_Month,Quantity,New_Invoice_Year) %>% filter(N
count(New_Invoice_Month)

data.frame(Transaction_Month)
```

```
##    New_Invoice_Month     n
## 1                  1 35147
## 2                  2 27707
## 3                  3 36748
## 4                  4 29916
## 5                  5 37030
## 6                  6 36874
## 7                  7 39518
## 8                  8 35284
## 9                  9 50226
## 10                10 60742
## 11                11 84711
## 12                12 68006
```

d) What was the date with the highest number of transactions from Australia? (3 marks)

```
# by using the pipeline function we can select the required coulumn which are invoice date as date is r

Australia = Online_Retail%>%select(Quantity,Country,TransactionValue,InvoiceDate)%>% filter(Country ==

Australia
```

```
##         InvoiceDate   n
## 1    1/10/2011 9:58   1
## 2    1/11/2011 9:47  19
## 3   1/14/2011 11:36   3
## 4   1/17/2011 11:12  19
## 5    1/19/2011 9:13  13
## 6   1/20/2011 12:11   4
```

```
## 7   1/28/2011 14:37  20
## 8    1/6/2011 11:12  46
## 9    1/6/2011 12:37   2
## 10  10/5/2011 12:35   1
## 11  10/5/2011 12:44  81
## 12   10/6/2011 9:31  27
## 13   10/6/2011 9:32   5
## 14 11/15/2011 10:32  26
## 15 11/15/2011 14:22   1
## 16  11/2/2011 12:03   1
## 17  11/2/2011 12:05   1
## 18 11/24/2011 12:30   8
## 19  11/3/2011 11:26   5
## 20  11/4/2011 10:18   1
## 21  11/4/2011 11:55   2
## 22  12/1/2010 10:03  14
## 23 12/14/2010 11:12   3
## 24 12/17/2010 14:10  10
## 25   12/8/2010 9:53   8
## 26   2/15/2011 9:52  69
## 27  2/27/2011 14:43  10
## 28   2/7/2011 13:59   6
## 29   2/7/2011 15:01   2
## 30   2/7/2011 15:09   2
## 31   2/7/2011 15:10   2
## 32  3/24/2011 13:05  16
## 33   3/3/2011 10:59  82
## 34   3/3/2011 13:11   2
## 35   3/9/2011 15:47  10
## 36   4/1/2011 14:28   1
## 37   4/28/2011 9:49   1
## 38    4/4/2011 9:57   1
## 39    4/8/2011 9:45  17
## 40  5/12/2011 12:34  23
## 41  5/17/2011 15:42  73
## 42  5/20/2011 14:13   4
## 43   5/23/2011 9:14  17
## 44  5/31/2011 11:29   1
## 45  6/15/2011 13:37 139
## 46    6/2/2011 9:57   2
## 47  6/30/2011 12:06  30
## 48  7/13/2011 15:30  22
## 49  7/13/2011 15:31   1
## 50  7/14/2011 13:28  35
## 51  7/19/2011 10:42  23
## 52  7/19/2011 10:51  57
## 53  7/19/2011 12:26  57
## 54  7/24/2011 12:05  20
## 55  7/26/2011 10:15   1
## 56  7/26/2011 10:16   1
## 57  8/12/2011 14:19  10
## 58   8/18/2011 8:51  97
## 59   9/1/2011 13:50   8
## 60   9/1/2011 13:51   1
```

```
## 61  9/16/2011 12:38  34
## 62  9/25/2011 11:30  22
## 63  9/28/2011 14:26   2
## 64  9/28/2011 14:55   4
## 65  9/28/2011 15:41  25
## 66    9/5/2011 9:48   8
```

```
max(Australia$n) #139 is the highest
```

```
## [1] 139
```

```
which.max(Australia$n)
```

```
## [1] 45
```

```
Final_Value<- Australia[45,] # The location of 139 is 45 from the list below, by using the index value
data.frame(Final_Value)
```

```
##        InvoiceDate   n
## 45 6/15/2011 13:37 139
```

e) The company needs to shut down the website for two consecutive hours for maintenance. What would be the hour of the day to start this so that the distribution is at minimum for the customers? The responsible IT team is available from 7:00 to 20:00 every day. (3 marks)

```
Maintainence_Time <-Online_Retail %>% select(Quantity,New_Invoice_Hour,New_Invoice_Date) %>% filter(New_
```

```
which.min(Maintainence_Time$Quantity)
```

```
## [1] 131
```

```
which.min(Maintainence_Time$n)
```

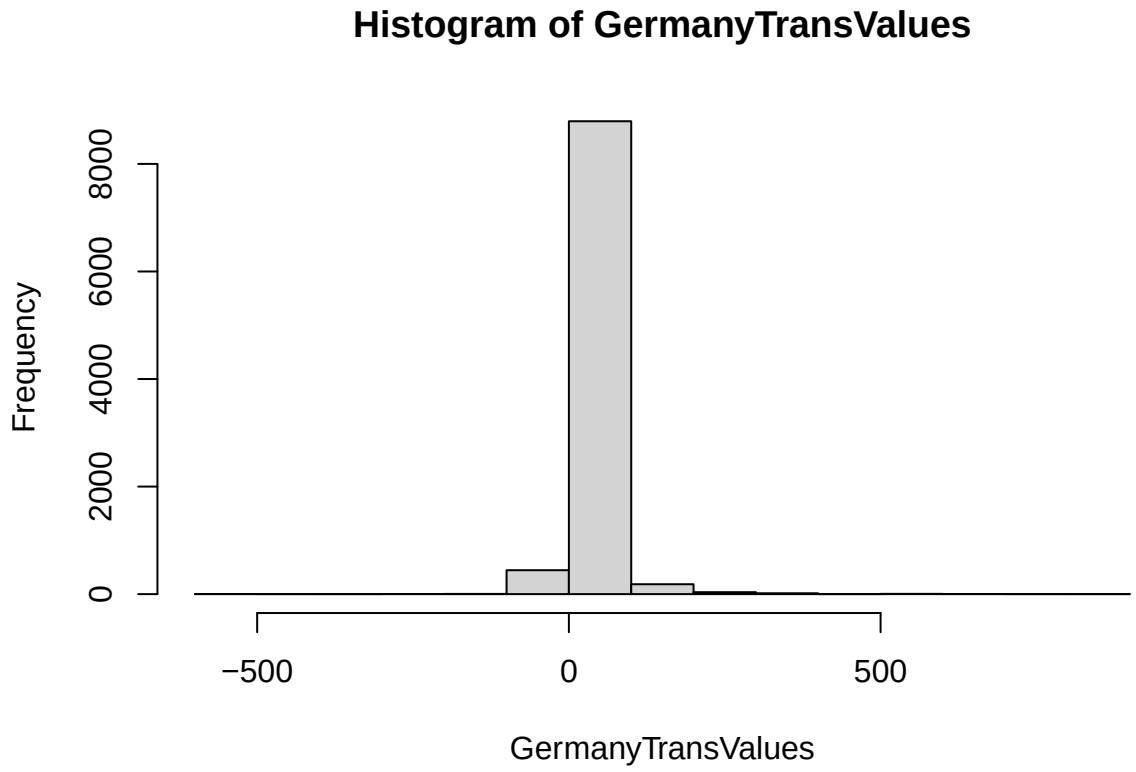
```
## [1] 1
```

```
Minimum_Quantity<- Maintainence_Time["131",]

Minimum_Ret<-Maintainence_Time["1",]

data.frame(Minimum_Quantity,Minimum_Ret)
```

```
##     New_Invoice_Hour Quantity n New_Invoice_Hour.1 Quantity.1 n.1
## 131                9   -80995 1                 7         -4   1
```

5. Plot the histogram of transaction values from Germany. Use the hist() function to plot. (5 marks)

```
GermanyTransValues <- subset(TransactionValue,Online_Retail$Country == 'Germany')
#GermanyTransValues

hist(GermanyTransValues)
```



6. Which customer had the highest number of transactions? Which customer is most valuable (i.e. highest total sum of transactions)? (10 marks)

```
Cust_Transaction <-  Online_Retail %>% select(CustomerID,Quantity,TransactionValue) %>% count(CustomerI

which.max(Cust_Transaction$n)
```

```
## [1] 4373
```

```
Cust_Transaction["4373",] #Here we are getting a value which is NA. So it is a missing value
```

```
##      CustomerID      n
## 4373         NA 135080
```

```
#Valuable customer in this case is as follows:
Most_Valuable_CustomerNA <- group_by(Online_Retail,CustomerID) %>% summarize(CustomerValNA = sum(Transa

which.max(Most_Valuable_CustomerNA$CustomerValNA)
```

```
## [1] 4373
```

```
Most_Valuable_CustomerNA["4373",]
```

```
## # A tibble: 1 x 2
##   CustomerID CustomerValNA
##        <int>         <dbl>
## 1         NA      1447682.
```

```
Missing_Value_Removal <- na.omit(Online_Retail %>% select(CustomerID, Quantity, TransactionValue)%>% cou
which.max(Missing_Value_Removal$n)
```

```
## [1] 4043
```

```
Missing_Value_Removal["4043",]
```

```
##      CustomerID    n
## 4043      17841 7983
```

```
#Valuable customer in this case wheere we have removed the missing cases is as follows:

Most_Val_Customer <- na.omit(group_by(Online_Retail,CustomerID) %>% summarize(Customer_Value = sum(Trans

which.max(Most_Val_Customer$Customer_Value)
```

```
## [1] 1704
```

```
Most_Val_Customer["1704",]
```

```
## # A tibble: 1 x 2
##   CustomerID Customer_Value
##        <int>          <dbl>
## 1      14646        279489.
```

7. Calculate the percentage of missing values for each variable in the dataset (5 marks). Hint colMeans():

```
# For the missing values, is.na is used here
colMeans(is.na(Online_Retail))
```

```
##         InvoiceNo         StockCode       Description          Quantity
##         0.0000000         0.0000000         0.0000000         0.0000000
##       InvoiceDate         UnitPrice        CustomerID           Country
##         0.0000000         0.0000000         0.2492669         0.0000000
##  TransactionValue  New_Invoice_Date  Invoice_Day_Week  New_Invoice_Hour
##         0.0000000         0.0000000         0.0000000         0.0000000
## New_Invoice_Month  New_Invoice_Year
##         0.0000000         0.0000000
```

8. What are the number of transactions with missing CustomerID records by countries? (10 marks)

```
#   is.na funtion is used here to find missing customer ID records by countries
Online_Retail %>%select(Country,CustomerID) %>% filter(is.na(Online_Retail$CustomerID)) %>% count (Coun
```

```
##          Country      n
## 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
```

9. On average, how often the costumers comeback to the website for their next shopping? (i.e. what is the average number of days between consecutive shopping) (Optional/Golden question: 18 additional marks!) Hint: 1. A close approximation is also acceptable and you may find diff() function useful.

```
Comeback_df <- table (Online_Retail$Invoice_Day_Week,Online_Retail$New_Invoice_Date)
Updated_Comeback_df<- diff(Comeback_df)
mean(Updated_Comeback_df)
```

```
## [1] 8.112787
```

10. In the retail sector, it is very important to understand the return rate of the goods purchased by customers. In this example, we can define this quantity, simply, as the ratio of the number of transactions cancelled (regardless of the transaction value) over the total number of transactions. With this definition, what is the return rate for the French customers? (10 marks). Consider the cancelled transactions as those where the 'Quantity' variable has a negative value.

```
 #We need to calculate the percentage of cancelled order with reference to the total orders in france.

France_Transaction <- Online_Retail%>% select(Quantity,Country) %>% filter (Country == "France") # Fren

Length_French_Orders <- length(France_Transaction$Quantity)

 #If the quantity value is less than 0, then we can consider it as a cancelled transaction
Cancelled_Transactions <- Online_Retail%>% select(Quantity,Country) %>% filter (Country == "France",Quan

French_Cancelled <-length(Cancelled_Transactions$Quantity)

#We perform cancelled order divided by total orders for France
Percentage_France <-  French_Cancelled / Length_French_Orders
Percentage_France
```

```
## [1] 0.01741264
```

```
data.frame(Length_French_Orders,French_Cancelled,Percentage_France)
```

```
##   Length_French_Orders French_Cancelled Percentage_France
## 1                 8557              149        0.01741264
```

11. What is the product that has generated the highest revenue for the retailer? (i.e. item with the highest total sum of 'TransactionValue'). (10 marks)

```
Highest_Revenue <- group_by(Online_Retail,Description) %>% summarize(Item = sum(TransactionValue))
which.max(Highest_Revenue$Item)
```

```
## [1] 1140
```

```
# Index is 1140

Highest_Revenue["1140",]
```

```
## # A tibble: 1 x 2
##   Description       Item
##   <chr>            <dbl>
## 1 DOTCOM POSTAGE 206245.
```

12. How many unique customers are represented in the dataset? You can use unique() and length() functions. (5 marks)

```
# ssapply() function in R Language takes list, vector or data frame as input and gives output in vector

Unique_Customer <- sapply(Online_Retail, function(Online_Retail) length(unique(Online_Retail)))
Unique_Customer
```

```
##         InvoiceNo         StockCode       Description          Quantity
##             25900              4070              4224               722
##       InvoiceDate         UnitPrice        CustomerID           Country
##             23260              1630              4373                38
##  TransactionValue  New_Invoice_Date  Invoice_Day_Week  New_Invoice_Hour
##              6204               305                 6                15
## New_Invoice_Month  New_Invoice_Year
##                12                 2
```

```
Unique_ID <- length(unique(Online_Retail$CustomerID))
Unique_ID
```

```
## [1] 4373
```