Descriptive Analytics

Jayasri

10/21/2021

Online Retail Analysis

For this, we need to use the 'Online Retail' dataset which can be downloaded in CSV format from the Dataset folder.This is a transnational data set which contains all the transactions occurring between 01 Dec 2010 and 09 Dec 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. The data contains the following attributes:

The data contains the following attributes: InvoiceNo: Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.

StockCode: Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.

Description: Product (item) name. Nominal.

Quantity: The quantities of each product (item) per transaction. Numeric.

InvoiceDate: Invoice Date and time. Numeric, the day and time when each transaction was generated.

UnitPrice: Unit price. Numeric, Product price per unit in sterling.

CustomerID: Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.

Country: Country name. Nominal, the name of the country where each customer resides.

```
chooseCRANmirror(graphics = getOption("menu.graphics"), ind = 79,
                 local.only = FALSE)
#Load the libraries
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
```

```r
#Import the dataset
Online_data<- read.csv("Online_Retail.csv")
#View(Online_data)
#See the first 6 rows of the dataset
head(as.data.frame(Online_data))
```

```
##   InvoiceNo StockCode                         Description Quantity
## 1    536365    85123A  WHITE HANGING HEART T-LIGHT HOLDER        6
## 2    536365     71053                 WHITE METAL LANTERN        6
## 3    536365    84406B      CREAM CUPID HEARTS COAT HANGER        8
## 4    536365    84029G KNITTED UNION FLAG HOT WATER BOTTLE        6
## 5    536365    84029E      RED WOOLLY HOTTIE WHITE HEART.        6
## 6    536365     22752        SET 7 BABUSHKA NESTING BOXES        2
##      InvoiceDate UnitPrice CustomerID        Country
## 1 12/1/2010 8:26      2.55      17850 United Kingdom
## 2 12/1/2010 8:26      3.39      17850 United Kingdom
## 3 12/1/2010 8:26      2.75      17850 United Kingdom
## 4 12/1/2010 8:26      3.39      17850 United Kingdom
## 5 12/1/2010 8:26      3.39      17850 United Kingdom
## 6 12/1/2010 8:26      7.65      17850 United Kingdom
```

Data Exploration

```r
#Descrptive statistics
summary(Online_data)
```

```
##   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       Country
##  Length:541909      Min.   :-11062.06   Min.   :12346    Length:541909
##  Class :character   1st Qu.:     1.25   1st Qu.:13953    Class :character
##  Mode  :character   Median :     2.08   Median :15152    Mode  :character
##                     Mean   :     4.61   Mean   :15288
##                     3rd Qu.:     4.13   3rd Qu.:16791
##                     Max.   : 38970.00   Max.   :18287
##                                         NA's   :135080
```

Questions

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.

```
#Total number of transactions by each country accounting more than 1% of total transactions
Country_data <- Online_data %>% group_by(Country) %>%
  summarise(n_Transactions=n(),percentage=100*(n()/nrow(Online_data))) %>%
  filter(percentage > 0.1) %>% arrange(desc(percentage))
as.data.frame(Country_data)
```

```
##            Country n_Transactions percentage
## 1   United Kingdom         495478 91.4319563
## 2          Germany           9495  1.7521392
## 3           France           8557  1.5790474
## 4             EIRE           8196  1.5124311
## 5            Spain           2533  0.4674217
## 6      Netherlands           2371  0.4375273
## 7          Belgium           2069  0.3817984
## 8      Switzerland           2002  0.3694347
## 9         Portugal           1519  0.2803054
## 10       Australia           1259  0.2323268
## 11          Norway           1086  0.2004027
## 12           Italy            803  0.1481799
## 13 Channel Islands            758  0.1398759
## 14         Finland            695  0.1282503
## 15          Cyprus            622  0.1147794
```

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

```
# Included New variable 'TransactionValue'
Online_data<- Online_data %>% mutate(TransactionValue= Quantity*UnitPrice)
#see the first 6 rows of the dataset
head(Online_data)
```

```
##   InvoiceNo StockCode                          Description Quantity
## 1    536365    85123A  WHITE HANGING HEART T-LIGHT HOLDER        6
## 2    536365     71053                 WHITE METAL LANTERN        6
## 3    536365    84406B      CREAM CUPID HEARTS COAT HANGER        8
## 4    536365    84029G KNITTED UNION FLAG HOT WATER BOTTLE        6
```

```
## 5    536365    84029E      RED WOOLLY HOTTIE WHITE HEART.        6
## 6    536365     22752        SET 7 BABUSHKA NESTING BOXES        2
##      InvoiceDate UnitPrice CustomerID        Country TransactionValue
## 1 12/1/2010 8:26      2.55      17850 United Kingdom            15.30
## 2 12/1/2010 8:26      3.39      17850 United Kingdom            20.34
## 3 12/1/2010 8:26      2.75      17850 United Kingdom            22.00
## 4 12/1/2010 8:26      3.39      17850 United Kingdom            20.34
## 5 12/1/2010 8:26      3.39      17850 United Kingdom            20.34
## 6 12/1/2010 8:26      7.65      17850 United Kingdom            15.30
```

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.

```
# List of countries with total transaction exceeding 130,000 British Pounds
Total_Transaction <- Online_data %>% group_by(Country) %>%
  summarise(Total_Sum_of_Transactions=sum(TransactionValue)) %>%
  filter(Total_Sum_of_Transactions >130000)
(as.data.frame(Total_Transaction))
```

```
##          Country Total_Sum_of_Transactions
## 1      Australia                  137077.3
## 2           EIRE                  263276.8
## 3         France                  197403.9
## 4        Germany                  221698.2
## 5    Netherlands                  284661.5
## 6 United Kingdom                 8187806.4
```

4.Conversion of categorical variable 'InvoiceDate' into POSIXIt Object

```
Temp=strptime(Online_data$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"
```

```
#let's separate date,  day  of  the  week  and  hour components dataframe with names as New_Invoice_Dat
Online_data$New_Invoice_Date <- as.Date(Temp)
#The difference between the two dates in terms of the number days
Online_data$New_Invoice_Date[20000]- Online_data$New_Invoice_Date[10]
```

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

```
#Convert dates to days of the week
Online_data$Invoice_Day_Week= weekdays(Online_data$New_Invoice_Date)
# Now Consider the hour and convert into the normal numerical value
Online_data$New_Invoice_Hour = as.numeric(format(Temp, "%H"))
# Now Consider the month and convert into the normal numerical value
Online_data$New_Invoice_Month = as.numeric(format(Temp, "%m"))

#Lets see the few values of the dataset with new columns
head(Online_data)
```

```
##   InvoiceNo StockCode                         Description Quantity
## 1    536365    85123A  WHITE HANGING HEART T-LIGHT HOLDER        6
## 2    536365     71053                 WHITE METAL LANTERN        6
## 3    536365    84406B      CREAM CUPID HEARTS COAT HANGER        8
## 4    536365    84029G KNITTED UNION FLAG HOT WATER BOTTLE        6
## 5    536365    84029E      RED WOOLLY HOTTIE WHITE HEART.        6
## 6    536365     22752        SET 7 BABUSHKA NESTING BOXES        2
##      InvoiceDate UnitPrice CustomerID        Country TransactionValue
## 1 12/1/2010 8:26      2.55      17850 United Kingdom            15.30
## 2 12/1/2010 8:26      3.39      17850 United Kingdom            20.34
## 3 12/1/2010 8:26      2.75      17850 United Kingdom            22.00
## 4 12/1/2010 8:26      3.39      17850 United Kingdom            20.34
## 5 12/1/2010 8:26      3.39      17850 United Kingdom            20.34
## 6 12/1/2010 8:26      7.65      17850 United Kingdom            15.30
##   New_Invoice_Date Invoice_Day_Week New_Invoice_Hour New_Invoice_Month
## 1       2010-12-01        Wednesday                8                12
## 2       2010-12-01        Wednesday                8                12
## 3       2010-12-01        Wednesday                8                12
## 4       2010-12-01        Wednesday                8                12
## 5       2010-12-01        Wednesday                8                12
## 6       2010-12-01        Wednesday                8                12
```

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

```
Trans_num_by_week<-Online_data %>% group_by(Invoice_Day_Week) %>%
  summarise(Percent_of_Trans_Num_by_week = 100*(n()/nrow(Online_data)))
as.data.frame(Trans_num_by_week)
```

```
##   Invoice_Day_Week Percent_of_Trans_Num_by_week
## 1           Friday                     15.16731
## 2           Monday                     17.55110
## 3           Sunday                     11.87930
## 4         Thursday                     19.16503
## 5          Tuesday                     18.78692
## 6        Wednesday                     17.45035
```

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

```
Trans_Vol_by_week<-Online_data %>% group_by(Invoice_Day_Week) %>%
  summarise(Percent_of_Trans_Vol_by_week=100*(sum(TransactionValue)/sum(Online_data$TransactionValue)))
as.data.frame(Trans_Vol_by_week)
```

```
##   Invoice_Day_Week Percent_of_Trans_Vol_by_week
## 1           Friday                    15.804787
## 2           Monday                    16.297194
## 3           Sunday                     8.265282
## 4         Thursday                    21.671867
## 5          Tuesday                    20.170636
## 6        Wednesday                    17.790232
```

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

```
Percent_Trans_by_Month<-Online_data %>% group_by(New_Invoice_Month) %>%
  summarise(Percent_of_Trans_by_month=100*(sum(TransactionValue)/sum(Online_data$TransactionValue)))
as.data.frame(Percent_Trans_by_Month)
```

```
##    New_Invoice_Month Percent_of_Trans_by_month
## 1                  1                  5.744919
## 2                  2                  5.109515
## 3                  3                  7.009487
## 4                  4                  5.059703
## 5                  5                  7.420519
## 6                  6                  7.090080
## 7                  7                  6.989308
## 8                  8                  7.003469
## 9                  9                 10.460751
## 10                10                 10.984123
## 11                11                 14.995836
## 12                12                 12.132290
```

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

```
s1<-filter(Online_data,Country=="Australia") %>% group_by(InvoiceDate) %>%
  summarise(Australia_highest_no_transactions=n())
as.data.frame(s1[which.max(s1$Australia_highest_no_transactions),])
```

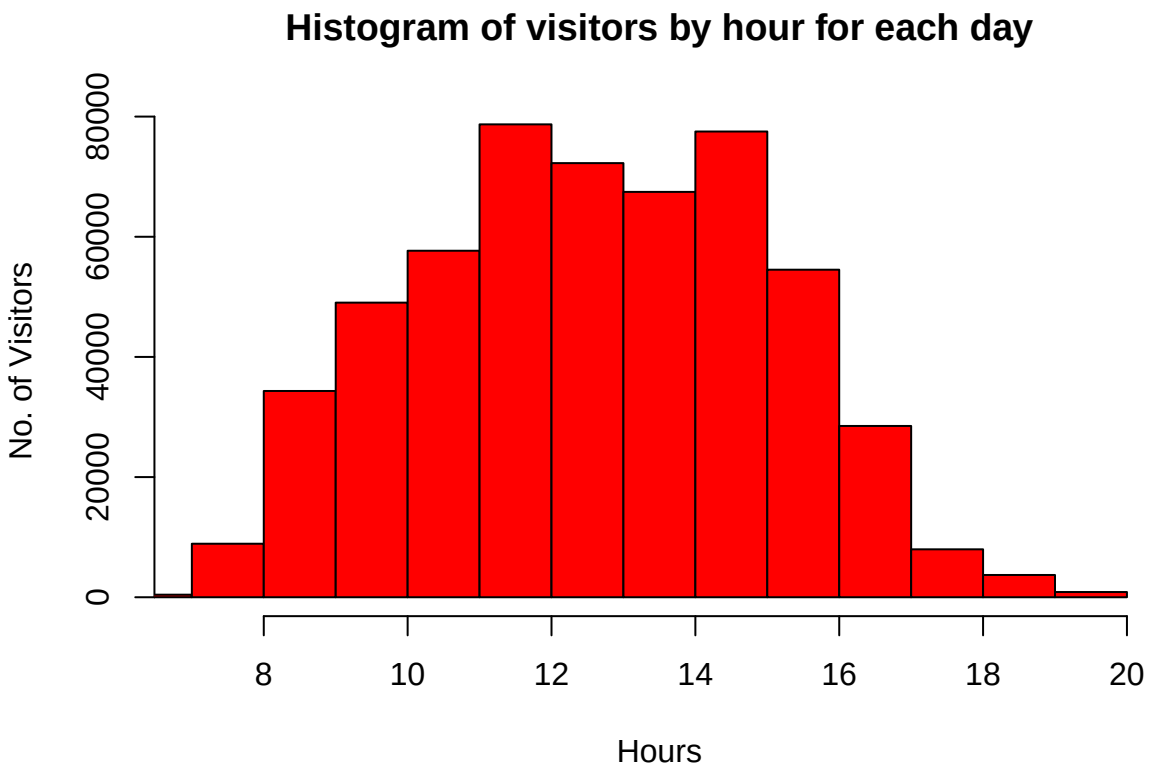
```
##       InvoiceDate Australia_highest_no_transactions
## 1 6/15/2011 13:37                              139
```

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

```
# Dataframe with Hour and its corresponding no of transactions per day
distribution<-Online_data %>% group_by(New_Invoice_Hour)%>%
  summarise(No_Of_Transactions=n(),Percentage=100*(n()/nrow(Online_data))) %>%
  filter(New_Invoice_Hour >=7 & New_Invoice_Hour <= 20)
as.data.frame(distribution)
```

```
##    New_Invoice_Hour No_Of_Transactions  Percentage
## 1                 7                383  0.07067607
## 2                 8               8909  1.64400296
## 3                 9              34332  6.33538103
## 4                10              49037  9.04893626
## 5                11              57674 10.64274629
## 6                12              78709 14.52439432
## 7                13              72259 13.33415758
## 8                14              67471 12.45061440
## 9                15              77519 14.30480025
## 10               16              54516 10.05999162
## 11               17              28509  5.26084638
## 12               18               7974  1.47146477
## 13               19               3705  0.68369413
## 14               20                871  0.16072809
```

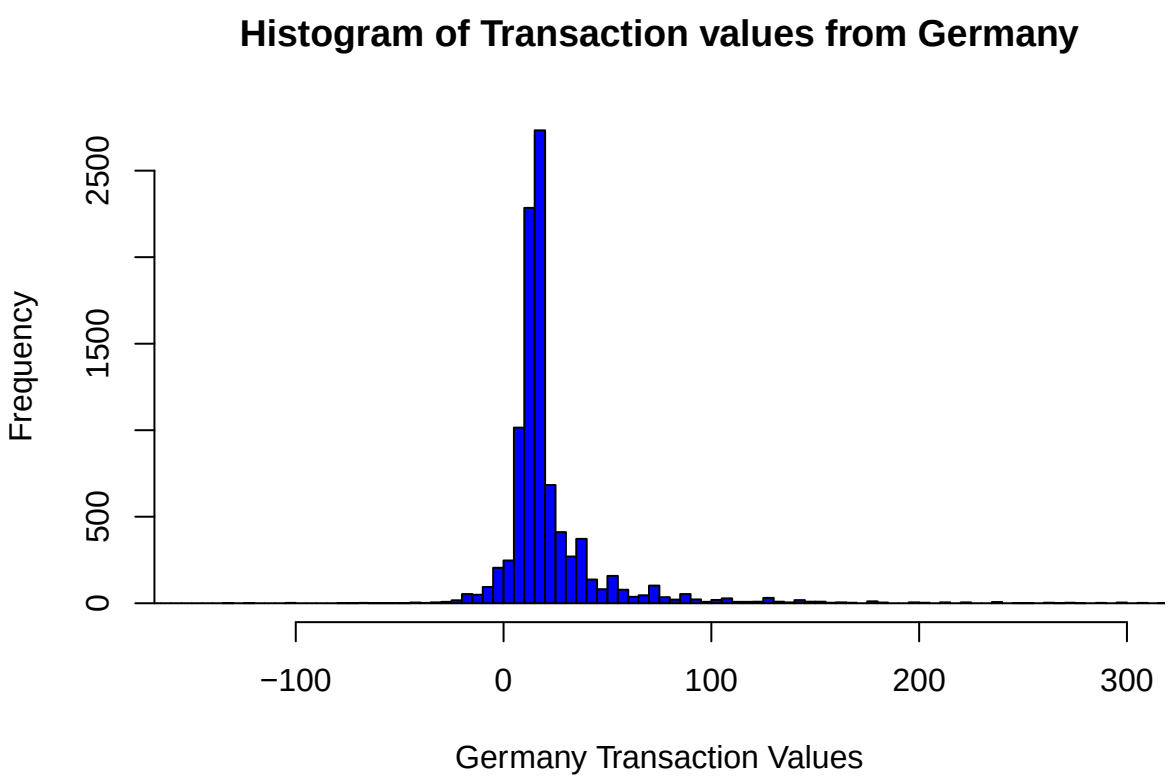
```
#Plotting a graph to show the website visitors for transactions per hour

hist(Online_data$New_Invoice_Hour,
     main="Histogram of visitors by hour for each day",
     xlim= c(7,20),
     col = "Red",
     xlab = "Hours",
     ylab= "No. of Visitors",
     breaks = 12
     )
```



It is clearly evident that the good time for maintenance shutdown would be 6.00 am and 20:00 pm. As it is mentioned in the question that responsible IT team would be available from 7.00am to 20:00 pm, the best time would be 7.00 am and 20:00 pm as the distribution would be minimum at these hours.

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

```
Germany_Transactions<-filter(Online_data,Country=="Germany")
hist(Germany_Transactions$TransactionValue,
     main = "Histogram of Transaction values from Germany",
     col = 'Blue',
     xlab = "Germany Transaction Values",
     ylab="Frequency",
     xlim = c(-150,300),
     breaks=500)
```



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

```
# Assumption 1: Considering the no. of transactions to calculate highest No. of transactions(valuable c

Cust_high_trans_withNA<-Online_data %>%  group_by(CustomerID) %>%
  summarise(Highest_no_of_Trans_with_NAValues=n()) %>% arrange(desc(Highest_no_of_Trans_with_NAValues))
  top_n(3)
```

```
## Selecting by Highest_no_of_Trans_with_NAValues
```

```
as.data.frame(Cust_high_trans_withNA)
```

```
##   CustomerID Highest_no_of_Trans_with_NAValues
## 1         NA                            135080
## 2      17841                              7983
## 3      14911                              5903
```

```
# Assumption 2 : Omitted NA Values and checked for the valuable customer
Cust_high_trans_without_NA<-Online_data %>% na.omit() %>%
  group_by(CustomerID) %>% summarise(Highest_no_of_Trans=n()) %>% arrange(desc(Highest_no_of_Trans)) %>%
  top_n(1)
```

```
## Selecting by Highest_no_of_Trans
```

```
as.data.frame(Cust_high_trans_without_NA)
```

```
##   CustomerID Highest_no_of_Trans
## 1      17841                7983
```

```
# Assumption 3: Considering the total sum of transactions(Transaction Volume) to calculate
# highest number of transactions(Valuable Customer)
Cust_high_TransVol_withNA<-Online_data %>% group_by(CustomerID) %>%
  summarise(Highest_Trans_Volume_with_NAValues=sum(TransactionValue)) %>%
  arrange(desc(Highest_Trans_Volume_with_NAValues)) %>% top_n(3)
```

```
## Selecting by Highest_Trans_Volume_with_NAValues
```

```
as.data.frame(Cust_high_TransVol_withNA)
```

```
##   CustomerID Highest_Trans_Volume_with_NAValues
## 1         NA                          1447682.1
## 2      14646                           279489.0
## 3      18102                           256438.5
```

```
# Assumption 4: Omitted NA Values and checked for the valuable customer
Cust_high_TransVol_without_NA <- Online_data %>% na.omit() %>% group_by(CustomerID) %>%
  summarise(Highest_Trans_Volume=sum(TransactionValue)) %>% arrange(desc(Highest_Trans_Volume)) %>% top_
```

```
## Selecting by Highest_Trans_Volume
```

```
as.data.frame(Cust_high_TransVol_without_NA)
```

```
##   CustomerID Highest_Trans_Volume
## 1      14646               279489
```

7. Calculate the percentage of missing values for each variable in the dataset

```
#Percentage of missing values in the dataset
Missing_Values_Percent<-colMeans(is.na(Online_data))
as.data.frame(Missing_Values_Percent)
```

```
##                   Missing_Values_Percent
## InvoiceNo                      0.0000000
## StockCode                      0.0000000
## Description                    0.0000000
## Quantity                       0.0000000
## InvoiceDate                    0.0000000
```

```
## UnitPrice                    0.0000000
## CustomerID                   0.2492669
## Country                      0.0000000
## TransactionValue             0.0000000
## New_Invoice_Date             0.0000000
## Invoice_Day_Week             0.0000000
## New_Invoice_Hour             0.0000000
## New_Invoice_Month            0.0000000
```

The output data frame shows that CustomerID column has 24.92% of missing values.

8. What are the number of transactions with missing CustomerID records by countries?

```
#No. of transactions with missing CustomerID records by countries
Online_data%>%filter(is.na(Online_data$CustomerID)) %>% group_by(Country) %>%
  summarise(No_of_missing_ID=n()) %>% arrange(desc(No_of_missing_ID))
```

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

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

```
# The average number of days between consecutive shopping per customer with all the transactions
#(Including Cancelled Transactions )

Online_data_without_NA<- Online_data %>% na.omit()

Avg_days_Per_Customer<- select(Online_data_without_NA,CustomerID,New_Invoice_Date) %>% distinct(Customer

#Lets see few rows of the customers with their Avg number of days
head(as.data.frame(Avg_days_Per_Customer))
```

```
##   CustomerID            avg
## 1      12347  60.83333 days
## 2      12348  94.33333 days
## 3      12352  43.33333 days
## 4      12356 151.50000 days
## 5      12358 149.00000 days
## 6      12359  64.80000 days
```

The average number of days between consecutive shopping per customer with out cancelled transactions.

```
#The average number of days between shopping per customer with out cancelled transactions.
Avg_days_Per_Cust_without_Cancelled_trans<- select(Online_data_without_NA,CustomerID,New_Invoice_Date) %
  filter(Online_data_without_NA$Quantity>0) %>% distinct(CustomerID,New_Invoice_Date) %>%
  group_by(CustomerID) %>% arrange(New_Invoice_Date) %>% summarise(avg=mean(diff(New_Invoice_Date))) %>%
  na.omit()

Avg_days_Per_Cust_without_Cancelled_trans
```

```
## # A tibble: 2,790 x 2
##    CustomerID avg
##         <int> <drtn>
##  1      12347  60.83333 days
##  2      12348  94.33333 days
##  3      12352  43.33333 days
##  4      12356 151.50000 days
##  5      12358 149.00000 days
##  6      12359  91.33333 days
##  7      12360  74.00000 days
##  8      12362  32.44444 days
##  9      12363 133.00000 days
## 10      12364  35.00000 days
## # ... with 2,780 more rows
```

```
#Average number of days between consecutive shopping for all the customers
Avg_days_Per_Cust_without_Cancelled_trans%>% summarise(avg_days_between_shopping = mean(avg))
```

```
## # A tibble: 1 x 1
##   avg_days_between_shopping
##   <drtn>
## 1 78.42025 days
```

10.n 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? Consider the cancelled transactions as those where the 'Quantity' variable has a negative value.

```
#Calculation of return rate for the french customers
France_Transactions<-filter(Online_data,Country=='France')

France_Cancelled_Transactions<-filter(Online_data,Country=='France'& Quantity<0)

Return_rate_France<- (nrow(France_Cancelled_Transactions)/nrow(France_Transactions))*100
print(paste("The return rate for the french customers is :",Return_rate_France))
```

```
## [1] "The return rate for the french customers is : 1.7412644618441"
```

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

```
#Highest revenue generated by the product for the retailer
High_Revenue<-Online_data %>% group_by(Description) %>% summarise(High_Revenue=sum(TransactionValue)) %
```

```
## Selecting by High_Revenue
```

```
as.data.frame(High_Revenue)
```

```
##      Description High_Revenue
## 1 DOTCOM POSTAGE     206245.5
```

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

```
Unique_Customers<-length(unique(Online_data$CustomerID))
print(paste("The number of Unique Customers in the dataset are:",Unique_Customers))
```

```
## [1] "The number of Unique Customers in the dataset are: 4373"
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.