Online Retail Analytics

Khushboo Yadav

10/18/2020

## Loading Data and Importing libraries for Online Retail dataset

Online\_Retail <- read.csv("~/Desktop/MSBA/Business Analytics/Data/Online\_Retail.csv")  
View(Online\_Retail)  
  
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(caret)

## Loading required package: lattice

## Loading required package: ggplot2

library(tidyselect)  
library(lubridate)

##   
## Attaching package: 'lubridate'

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

library(tidyverse)

## ── Attaching packages ───────────────────────────────────────────────────────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ tibble 3.0.3 ✓ purrr 0.3.4  
## ✓ tidyr 1.1.2 ✓ stringr 1.4.0  
## ✓ readr 1.3.1 ✓ forcats 0.5.0

## ── Conflicts ──────────────────────────────────────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## x lubridate::as.difftime() masks base::as.difftime()  
## x lubridate::date() masks base::date()  
## x dplyr::filter() masks stats::filter()  
## x lubridate::intersect() masks base::intersect()  
## x dplyr::lag() masks stats::lag()  
## x purrr::lift() masks caret::lift()  
## x lubridate::setdiff() masks base::setdiff()  
## x lubridate::union() masks base::union()

library(Lahman)  
  
str(Online\_Retail) #Analyzing the structure of the dataset

## 'data.frame': 541909 obs. of 8 variables:  
## $ InvoiceNo : chr "536365" "536365" "536365" "536365" ...  
## $ StockCode : chr "85123A" "71053" "84406B" "84029G" ...  
## $ Description: chr "WHITE HANGING HEART T-LIGHT HOLDER" "WHITE METAL LANTERN" "CREAM CUPID HEARTS COAT HANGER" "KNITTED UNION FLAG HOT WATER BOTTLE" ...  
## $ Quantity : int 6 6 8 6 6 2 6 6 6 32 ...  
## $ InvoiceDate: chr "12/1/2010 8:26" "12/1/2010 8:26" "12/1/2010 8:26" "12/1/2010 8:26" ...  
## $ UnitPrice : num 2.55 3.39 2.75 3.39 3.39 7.65 4.25 1.85 1.85 1.69 ...  
## $ CustomerID : int 17850 17850 17850 17850 17850 17850 17850 17850 17850 13047 ...  
## $ Country : chr "United Kingdom" "United Kingdom" "United Kingdom" "United Kingdom" ...

nrow(Online\_Retail) #Determining the row count of the dataset

## [1] 541909

## 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. (5 marks)

#Breakdown of the no. of transactions in numbers and in percentage by Countries  
  
TransactByCountry<-Online\_Retail %>%   
 group\_by( Country ) %>%   
 summarise( Percentage = 100 \* (n() / nrow( Online\_Retail ) ),Number\_of\_Transactions=n() ) %>%  
   
#Printing the complete results  
print(TransactByCountry)

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 38 x 3  
## Country Percentage Number\_of\_Transactions  
## <chr> <dbl> <int>  
## 1 Australia 0.232 1259  
## 2 Austria 0.0740 401  
## 3 Bahrain 0.00351 19  
## 4 Belgium 0.382 2069  
## 5 Brazil 0.00591 32  
## 6 Canada 0.0279 151  
## 7 Channel Islands 0.140 758  
## 8 Cyprus 0.115 622  
## 9 Czech Republic 0.00554 30  
## 10 Denmark 0.0718 389  
## # … with 28 more rows

# Output : Countries accounting for for more than 1 percentage  
  
filter(TransactByCountry,TransactByCountry$Percentage>=1)

## # A tibble: 4 x 3  
## Country Percentage Number\_of\_Transactions  
## <chr> <dbl> <int>  
## 1 EIRE 1.51 8196  
## 2 France 1.58 8557  
## 3 Germany 1.75 9495  
## 4 United Kingdom 91.4 495478

## 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  
TransactionValue<-Online\_Retail$Quantity\*Online\_Retail$UnitPrice  
  
#Binding TransactionValue in Online Retail Dataframe  
Online\_Retail<-cbind(Online\_Retail,TransactionValue)  
  
#Using str function to validate the new addition in the Online Retail dataframe  
str(Online\_Retail)

## 'data.frame': 541909 obs. of 9 variables:  
## $ InvoiceNo : chr "536365" "536365" "536365" "536365" ...  
## $ StockCode : chr "85123A" "71053" "84406B" "84029G" ...  
## $ Description : chr "WHITE HANGING HEART T-LIGHT HOLDER" "WHITE METAL LANTERN" "CREAM CUPID HEARTS COAT HANGER" "KNITTED UNION FLAG HOT WATER BOTTLE" ...  
## $ Quantity : int 6 6 8 6 6 2 6 6 6 32 ...  
## $ InvoiceDate : chr "12/1/2010 8:26" "12/1/2010 8:26" "12/1/2010 8:26" "12/1/2010 8:26" ...  
## $ UnitPrice : num 2.55 3.39 2.75 3.39 3.39 7.65 4.25 1.85 1.85 1.69 ...  
## $ CustomerID : int 17850 17850 17850 17850 17850 17850 17850 17850 17850 13047 ...  
## $ Country : chr "United Kingdom" "United Kingdom" "United Kingdom" "United Kingdom" ...  
## $ TransactionValue: num 15.3 20.3 22 20.3 20.3 ...

## 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)

# Aggregating the results of Transaction Values based on different countries in the Online Retail Dataframe  
Sum\_TransactByCountry <- Online\_Retail%>%  
 group\_by(Country)%>%  
 summarise(Sum\_Transaction\_Value=sum(TransactionValue))

## `summarise()` ungrouping output (override with `.groups` argument)

# Output:Filtering the results based on the 130000 British Pound  
filter(Sum\_TransactByCountry,Sum\_Transaction\_Value>130000)

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

## Q4.This is an optional question which carries additional marks (golden questions). In this question, we are dealing with the InvoiceDate variable.

#explicitly instructing R to interpret InvoiceDate as a Date variable  
Temp=strptime(Online\_Retail$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"

Online\_Retail$New\_Invoice\_Date <- as.Date(Temp)  
#convert dates to days of the week  
Online\_Retail$Invoice\_Day\_Week= weekdays(Online\_Retail$New\_Invoice\_Date)  
#Taking the hour from Temp   
Online\_Retail$New\_Invoice\_Hour = as.numeric(format(Temp, "%H"))  
#Taking months from Temp  
Online\_Retail$New\_Invoice\_Month = as.numeric(format(Temp, "%m"))

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

# Aggregating and determining percentage of Transaction Values by number based on different countries  
 Prcntg\_N\_DOW <- Online\_Retail%>%  
 group\_by(Online\_Retail$Invoice\_Day\_Week)%>%  
 summarise(Transaction\_DOW = (n()),PercentageTransaction = (n()/nrow(Online\_Retail))\*100)

## `summarise()` ungrouping output (override with `.groups` argument)

# Output:To print the results  
print(Prcntg\_N\_DOW)

## # A tibble: 6 x 3  
## `Online\_Retail$Invoice\_Day\_Week` Transaction\_DOW PercentageTransaction  
## <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) Show the percentage of transactions (by transaction volume) by days of the week (extra 1 marks)

# Aggregating and determining percentage of Transaction Values by volume based on different countries  
 Prcntg\_V\_DOW<- Online\_Retail%>%  
 group\_by(Online\_Retail$Invoice\_Day\_Week)%>%  
 summarise(Sum\_TransactDOW = (sum(TransactionValue)))

## `summarise()` ungrouping output (override with `.groups` argument)

#Calculating the percentage   
 Prcntg\_V\_DOW$PercentageTransaction<- (Prcntg\_V\_DOW$Sum\_TransactDOW/sum(Prcntg\_V\_DOW$Sum\_TransactDOW))\*100   
   
# Output:Print the result  
 print(Prcntg\_V\_DOW)

## # A tibble: 6 x 3  
## `Online\_Retail$Invoice\_Day\_Week` Sum\_TransactDOW PercentageTransaction  
## <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) Show the percentage of transactions (by transaction volume) by month of the year (extra 1 marks)

# Aggregating and determining percentage of Transaction Values by volume based on different countries  
Prcntg\_MOY <- Online\_Retail%>%  
 group\_by(Online\_Retail$New\_Invoice\_Month)%>%  
 summarise(Sum\_TransactionValue = (sum(TransactionValue)))

## `summarise()` ungrouping output (override with `.groups` argument)

#Calculating the percentage   
 Prcntg\_MOY$Percentage\_TransactionValue<- (Prcntg\_MOY$Sum\_TransactionValue/sum(Prcntg\_MOY$Sum\_TransactionValue))\*100   
   
 # Output:Print the result  
 print(Prcntg\_MOY)

## # A tibble: 12 x 3  
## `Online\_Retail$New\_Invoice\_Mon… Sum\_TransactionVal… Percentage\_TransactionVa…  
## <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) What was the date with the highest number of transactions from Australia? (3 marks)

#Filtering the data based on Country = "Australia"  
 Australia\_Tran<- filter(Online\_Retail,Online\_Retail$Country=="Australia")  
  
#Aggregating the transactions from Australia based on Invoice Dates  
 Australia\_Tran\_bydate <- Australia\_Tran%>%  
 group\_by(Australia\_Tran$New\_Invoice\_Date)%>%  
 summarise(TranCount\_Aust = n())

## `summarise()` ungrouping output (override with `.groups` argument)

#Arranging the results in order to see the date with highest number of transactions   
 arrange(Australia\_Tran\_bydate,desc(Australia\_Tran\_bydate$TranCount\_Aust))

## # A tibble: 49 x 2  
## `Australia\_Tran$New\_Invoice\_Date` TranCount\_Aust  
## <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

#Output:Print the date with highest number of transactions   
 top\_n(Australia\_Tran\_bydate,1)

## Selecting by TranCount\_Aust

## # A tibble: 1 x 2  
## `Australia\_Tran$New\_Invoice\_Date` TranCount\_Aust  
## <date> <int>  
## 1 2011-06-15 139

## 4.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)

#Aggregating the no.of transactionValues by hours  
 Tran\_byhour <- Online\_Retail%>%  
 group\_by(Online\_Retail$New\_Invoice\_Hour)%>%  
 summarise(TranCount = n())

## `summarise()` ungrouping output (override with `.groups` argument)

#Creating a variable with numeric data   
 smtrn <-numeric()  
   
 #Using for loop for adding consecutive hours together:-  
   
 for (i in 1:dim(Tran\_byhour)[1])  
 {   
 if (between(Tran\_byhour$`Online\_Retail$New\_Invoice\_Hour`[i],6,20))   
 {   
 #Summation of the row [i] with row[i+1]  
 smtrn<- c(smtrn,sum((Tran\_byhour$TranCount[i]),(Tran\_byhour$TranCount[i+1])))   
   
 }  
 }  
 #Adding column  
 Tran\_byhour$NextHour<-Tran\_byhour[,2]  
 Tran\_byhour$NextHour<-c(7:20,NA)  
   
 #Combining the data into a single data frame  
 Tran\_byhour$ConsecutiveHours\_CountTransaction<-smtrn  
   
 #Printing the output  
 print(Tran\_byhour)

## # A tibble: 15 x 4  
## `Online\_Retail$New\_Invoice\_H… TranCount NextHour ConsecutiveHours\_CountTrans…  
## <dbl> <int> <int> <dbl>  
## 1 6 41 7 424  
## 2 7 383 8 9292  
## 3 8 8909 9 43241  
## 4 9 34332 10 83369  
## 5 10 49037 11 106711  
## 6 11 57674 12 136383  
## 7 12 78709 13 150968  
## 8 13 72259 14 139730  
## 9 14 67471 15 144990  
## 10 15 77519 16 132035  
## 11 16 54516 17 83025  
## 12 17 28509 18 36483  
## 13 18 7974 19 11679  
## 14 19 3705 20 4576  
## 15 20 871 NA NA

##Filtering 6 hours , due to unavailability of responsible IT team  
Tran\_byhour<-filter(Tran\_byhour,Tran\_byhour$`Online\_Retail$New\_Invoice\_Hour`>6)  
   
  
 #Output:determining the smallest(sum of transactions for2 consecutive hours) number   
top\_n(Tran\_byhour,-1)

## Selecting by ConsecutiveHours\_CountTransaction

## # A tibble: 1 x 4  
## `Online\_Retail$New\_Invoice\_H… TranCount NextHour ConsecutiveHours\_CountTransa…  
## <dbl> <int> <int> <dbl>  
## 1 19 3705 20 4576

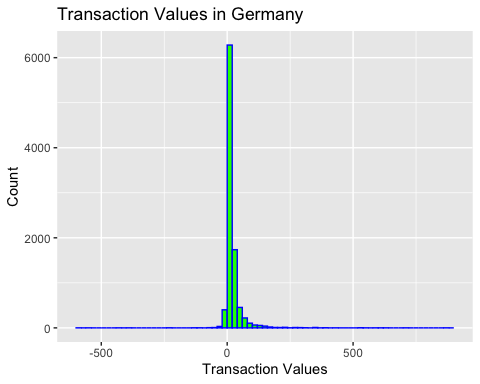
Answer: Therefore the we can see the least no. of consecutive transactions are happening in the 19-20 hrs.Hence 19hrs and 20 hrs are best for doing the maintainence.

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

#Making a subset for Germany Transactions  
Germany\_TransactionValue<-subset(Online\_Retail,Country=='Germany')  
  
#To understand the data frame for a better Histogram  
summary(Germany\_TransactionValue)

## InvoiceNo StockCode Description Quantity   
## Length:9495 Length:9495 Length:9495 Min. :-288.00   
## Class :character Class :character Class :character 1st Qu.: 5.00   
## Mode :character Mode :character Mode :character Median : 10.00   
## Mean : 12.37   
## 3rd Qu.: 12.00   
## Max. : 600.00   
## InvoiceDate UnitPrice CustomerID Country   
## Length:9495 Min. : 0.000 Min. :12426 Length:9495   
## Class :character 1st Qu.: 1.250 1st Qu.:12480 Class :character   
## Mode :character Median : 1.950 Median :12592 Mode :character   
## Mean : 3.967 Mean :12646   
## 3rd Qu.: 3.750 3rd Qu.:12662   
## Max. :599.500 Max. :14335   
## TransactionValue New\_Invoice\_Date Invoice\_Day\_Week New\_Invoice\_Hour  
## Min. :-599.50 Min. :2010-12-01 Length:9495 Min. : 8.00   
## 1st Qu.: 11.10 1st Qu.:2011-04-04 Class :character 1st Qu.:10.00   
## Median : 16.60 Median :2011-07-19 Mode :character Median :12.00   
## Mean : 23.35 Mean :2011-07-03 Mean :12.11   
## 3rd Qu.: 23.40 3rd Qu.:2011-10-11 3rd Qu.:14.00   
## Max. : 876.00 Max. :2011-12-09 Max. :19.00   
## New\_Invoice\_Month  
## Min. : 1.00   
## 1st Qu.: 5.00   
## Median : 8.00   
## Mean : 7.29   
## 3rd Qu.:10.00   
## Max. :12.00

ggplot(Germany\_TransactionValue, aes(x=TransactionValue)) +   
 geom\_histogram(breaks=seq(-600,900,by=20),col="Blue",fill="Green")+  
 labs(title="Transaction Values in Germany",x="Transaction Values",y="Count")



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

## 6.1 Which customer had the highest number of transactions

# Customers id with the number of transactions each  
Customer\_TranNo <- Online\_Retail%>%  
 group\_by(CustomerID)%>%  
 summarise( TransactionNo= n())

## `summarise()` ungrouping output (override with `.groups` argument)

#arranging data in descending format to find the customer with maximum no. of transactions  
 arrange(Customer\_TranNo,desc(TransactionNo))

## # A tibble: 4,373 x 2  
## CustomerID TransactionNo  
## <int> <int>  
## 1 NA 135080  
## 2 17841 7983  
## 3 14911 5903  
## 4 14096 5128  
## 5 12748 4642  
## 6 14606 2782  
## 7 15311 2491  
## 8 14646 2085  
## 9 13089 1857  
## 10 13263 1677  
## # … with 4,363 more rows

#removing na values from the dataset( highest no. of transactions are coming for the Customer 'IDs as "NA")  
Customer\_TranNo<-na.omit(Customer\_TranNo)  
  
#Output:populating the customer with highest no. of transaction after removing NA customer ID's  
top\_n(Customer\_TranNo,1)

## Selecting by TransactionNo

## # A tibble: 1 x 2  
## CustomerID TransactionNo  
## <int> <int>  
## 1 17841 7983

## 6.2 Which customer is most valuable (i.e. highest total sum of transactions)?

#Valuable Customer based on the highest total sum of transaction  
Customer\_TranVol <- Online\_Retail%>%  
 group\_by(CustomerID)%>%  
 summarise(sum\_Transaction = sum(TransactionValue))

## `summarise()` ungrouping output (override with `.groups` argument)

#arranging data in descending format to find the customer ID with highest total sum of transactions   
arrange(Customer\_TranVol,desc(sum\_Transaction))

## # A tibble: 4,373 x 2  
## CustomerID sum\_Transaction  
## <int> <dbl>  
## 1 NA 1447682.  
## 2 14646 279489.  
## 3 18102 256438.  
## 4 17450 187482.  
## 5 14911 132573.  
## 6 12415 123725.  
## 7 14156 113384.  
## 8 17511 88125.  
## 9 16684 65892.  
## 10 13694 62653.  
## # … with 4,363 more rows

#removing na values from the dataset(Since custmer id=NA has the highest total sum of transactions)  
Customer\_TranVol<-na.omit(Customer\_TranVol)  
  
#Output:populating most valuable customer  
top\_n(Customer\_TranVol,1)

## Selecting by sum\_Transaction

## # A tibble: 1 x 2  
## CustomerID sum\_Transaction  
## <int> <dbl>  
## 1 14646 279489.

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

#Determining the Percentage of missing values for all the variables   
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   
## 0.0000000

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

# Aggregating the date based on the countries and validating the customerID in terms of TRUE/FALSE by countries  
 misCust <- Online\_Retail%>%  
 group\_by(Country)%>%  
 summarize(miss\_val= is.na(CustomerID))

## `summarise()` regrouping output by 'Country' (override with `.groups` argument)

#Filtering the records based on missing values  
mis\_cust\_cntry <- filter(misCust,miss\_val=="TRUE")  
  
#Output:Result for no. of transactions with missing CustomerID records by countries  
count(mis\_cust\_cntry)

## # 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

## 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.

#Creating subset with only required columns  
  
Online\_Retail\_2<-subset(Online\_Retail[,c(1,4,5,7,10)])  
Online\_Retail\_2 <- na.omit(Online\_Retail\_2)   
  
 #Aggregating the data based on Customer ID and determining the distinct InvoiceNo with the Invoice dates for all non-canceled transactions   
  
InvoiceDt\_byCustomerID <- Online\_Retail\_2 %>%   
 filter(Quantity>1) %>%  
 group\_by(CustomerID) %>%  
 distinct(InvoiceNo,.keep\_all = TRUE)%>%  
 arrange(CustomerID)  
  
#Determining the difference of Invoice Dates  
 Difference\_InvoiceDt<-difftime(InvoiceDt\_byCustomerID$New\_Invoice\_Date,lag(InvoiceDt\_byCustomerID$New\_Invoice\_Date),units="days")  
   
  
# Changing the type   
Difference\_InvoiceDt<-as.data.frame((Difference\_InvoiceDt))  
  
Difference\_InvoiceDt$Dif\_InvoiceDT<-Difference\_InvoiceDt  
  
#Merging the Invoice date difference with the data frame using cbind  
 InvoiceDt\_byCustomerID<-data.frame(cbind(InvoiceDt\_byCustomerID,Difference\_InvoiceDt$Dif\_InvoiceDT))  
  
  
   
 ## To determine the consumer's come back, we don't need the customer's with no recurring transactions.Therefore we should remove them in order to get the right result.  
   
# Updating the first record of difference in dates for each customer as NA , as it was difference of last transaction of one customer id with first transaction date of another.Also, this will help to remove the records where customer didn't come for the second purchase.  
   
 InvoiceDt\_byCustomerID <- InvoiceDt\_byCustomerID %>%  
 group\_by(CustomerID) %>%  
 mutate(X.Difference\_InvoiceDt. = replace ( X.Difference\_InvoiceDt. , row\_number(CustomerID) == 1, NA))  
  
#Removing the NA values   
InvoiceDt\_byCustomerID<- na.omit(InvoiceDt\_byCustomerID)  
  
 #Output  
 mean(InvoiceDt\_byCustomerID$X.Difference\_InvoiceDt.)

## Time difference of 40.44346 days

## 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.

#Filtering the records based on French customer with canceled transactions  
France\_cancelled<-filter(Online\_Retail,Country=="France"&Quantity<0)  
  
  
# Filtering the French Customer transactions  
France\_Total\_Transactions<-filter(Online\_Retail,Country=="France")  
  
  
#Determining the rate of the canceled transactions by French Customer  
rate<- (nrow(France\_cancelled)/nrow(France\_Total\_Transactions))\*100  
  
#Output  
print(rate)

## [1] 1.741264

## 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)

#Aggregating the Transaction Value based on Product   
high\_Revenue\_Product <- Online\_Retail%>%  
 group\_by(Online\_Retail$Description)%>%  
 summarise(highRev = sum(TransactionValue))

## `summarise()` ungrouping output (override with `.groups` argument)

# Output:To determine the product with highest revenue  
top\_n(high\_Revenue\_Product,1)

## Selecting by highRev

## # A tibble: 1 x 2  
## `Online\_Retail$Description` highRev  
## <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)

#Storing the unique customer ids in a new data frame  
unique\_customers<-data.frame(unique(Online\_Retail$CustomerID,incomparables =FALSE,nmax=NA))  
  
# Output:To check the count of unique customers  
count(unique\_customers)

## n  
## 1 4373