ONLINE RETAIL ANALYTICS

Nikhil Reddy Addula

2022-10-29

```
#Importing library
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(magrittr)
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(readr)
#Importing online retail Data Set
OR <- read_csv("~/Documents/assignments/BUSINESS ANALYTICS/assignment 2/Online_Retail.csv")
```

```
## Rows: 541909 Columns: 8
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): InvoiceNo, StockCode, Description, InvoiceDate, Country
## dbl (3): Quantity, UnitPrice, CustomerID
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
View(OR)
```

```
#1. Show the breakdown of the number oftransactions by countries i.e. how many transactions are in the
set.seed(123)
OR %>% group_by(Country)%>% summarise(transactions = n())%>% mutate(percentage= (transactions/541909)*1
```

```
## # A tibble: 4 x 3
##   Country        transactions percentage
##   <chr>                 <int>      <dbl>
```

```
## 1 United Kingdom       495478      91.4
## 2 Germany                9495       1.75
## 3 France                 8557       1.58
## 4 EIRE                   8196       1.51
```

```
#2. Create a new variable 'T_value' that is the product of the exising 'Quantity' and 'UnitPrice' varia
OR<- mutate(OR, "TransactionValue"=TransactionValue<- OR$Quantity * OR$UnitPrice)
colnames(OR)
```

```
## [1] "InvoiceNo"        "StockCode"        "Description"      "Quantity"
## [5] "InvoiceDate"      "UnitPrice"        "CustomerID"       "Country"
## [9] "TransactionValue"
```

```
#3. Using the newly created variable,T_value, show the breakdown of T_valuesby countries i.e. how much
OR%>% group_by(Country)%>% summarise(total.sum.of.transaction.values = sum(TransactionValue))%>% arrang
```

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

```
#4.This is an optional question which carries additional marks (golden questions). In this question, we

#let's convert 'InvoiceDate' into a POSIXltobject:
Temp=strptime(OR$InvoiceDate,format='%m/%d/%Y %H:%M',tz='GMT')
#Now, let's separate date,  day  of  the  week  and  hour components dataframe with names as
#New_Invoice_Date,Invoice_Day_Weekand New_Invoice_Hour:
OR$New_Invoice_Date<-as.Date(Temp)
#knowing two date values,the object allows you to know the difference between the two dates in terms of
OR$New_Invoice_Date[20000]-OR$New_Invoice_Date[10]
```

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

```
## Time difference of 8 days
#Also we can convert dates to days of the week. Let's define a new variable for that
OR$Invoice_Day_Week=weekdays(OR$New_Invoice_Date)
#For the Hour, let's just take the hour (ignore the minute)  and  convert  into  a  normal  numerical v
OR$New_Invoice_Hour =as.numeric(format(Temp,"%H"))
#Finally, lets define the month as a separate numeric variable too:
OR$New_Invoice_Month = as.numeric(format(Temp, "%m"))
```

```
#4.A-Show the percentage of transactions (by numbers) by days of the week
OR%>% group_by(Invoice_Day_Week)%>% summarise(Number.of.transaction=(n()))%>% mutate(Number.of.transacti
```

```
## # A tibble: 6 x 3
##   Invoice_Day_Week Number.of.transaction percent
##   <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
OR%>% group_by(Invoice_Day_Week)%>% summarise(Volume.of.transaction=(sum(TransactionValue)))%>% mutate(
```

```
## # A tibble: 6 x 3
##   Invoice_Day_Week Volume.of.transaction percent
##   <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
OR%>% group_by(New_Invoice_Month)%>%
summarise(Volume.By.Month=sum(TransactionValue))%>% mutate(Volume.By.Month,'Percent'=(Volume.By.Month*1
```

```
## # A tibble: 12 x 3
##    New_Invoice_Month Volume.By.Month Percent
##                <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?

NR<-OR%>%
group_by(New_Invoice_Date,Country)%>%
filter(Country=='Australia')%>%
summarise(Number=sum(Quantity),amount=sum(TransactionValue))%>%
arrange(desc(Number))
```

```
## `summarise()` has grouped output by 'New_Invoice_Date'. You can override using
## the `.groups` argument.
```

```
NR<-NR[NR['Number']==max(NR['Number']),]
print(paste('The date with the highest number of transactions from Australia is', NR['New_Invoice_Date'
```

```
## [1] "The date with the highest number of transactions from Australia is 15140 which is 23426.81 $"
```

```
#4.E-The company needs to shut down the website for twoconsecutivehours for maintenance. What would be

G=OR%>% group_by(New_Invoice_Hour)%>% summarise(Total.transaction= n())
n<-rollapply(G['Total.transaction'],2,sum)
index(min(n))
```

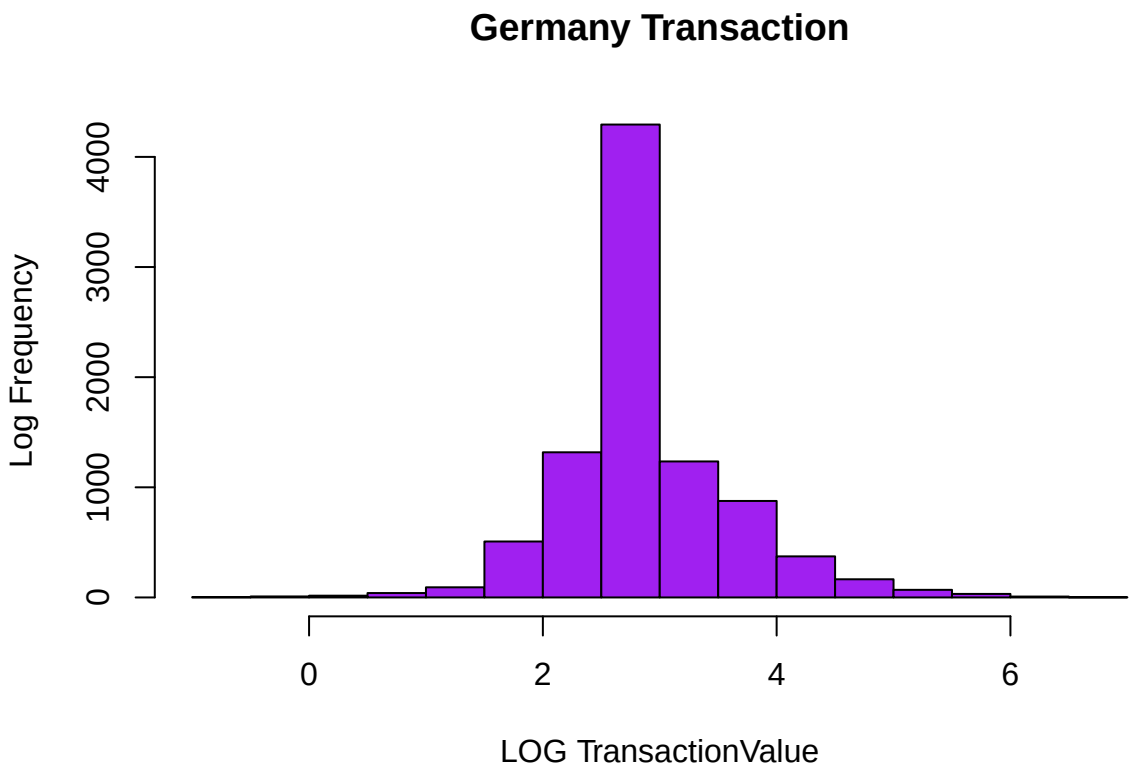
```
## [1] 1
```

```
print('According to the data, the ideal time to shut down a website for two hours straight for maintenar
```

```
## [1] "According to the data, the ideal time to shut down a website for two hours straight for maintena
```

```
#5.Plot the histogramof transaction values from Germany.Usethe hist() function to plot.
hist(x=log(OR$TransactionValue[OR$Country=="Germany"]),xlab = "LOG TransactionValue",col = 'Purple' ,ma
```

```
## Warning in log(OR$TransactionValue[OR$Country == "Germany"]): NaNs produced
```




```
#6.Which customer had the highest number of transactions? Which customer is most valuable (i.e. highest

Data123<- OR %>% group_by(CustomerID)%>%
summarise(CustomerTransaction = n())%>% filter(CustomerID != "NA")%>% filter(CustomerTransaction ==max(
print(paste('The customerID had the highest number of transactions is',Data123$CustomerID,'with max tra
```

```
## [1] "The customerID had the highest number of transactions is 17841 with max transaction of  7983"
```

```
Data234<- OR%>% group_by(CustomerID)%>%
summarise(total.transaction.by.each.customer = sum(TransactionValue))%>% arrange(desc(total.transaction
filter(CustomerID != "NA")%>% filter(total.transaction.by.each.customer ==max(total.transaction.by.each
print(paste('Most valuable customerID is',Data234$CustomerID,'with total transaction Amount $',Data234$
```

```
## [1] "Most valuable customerID is 14646 with total transaction Amount $ 279489.02"
```

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

null_v<-colMeans(is.na(OR))
print(paste('Online customerID column has missing values in dataset and  i.e.',null_v['CustomerID']*100
```

```
## [1] "Online customerID column has missing values in dataset and  i.e. 24.9266943342886 % of whole da
```

```
#8-What are the number of transactions with missing CustomerID records by countries
```

```
OR%>% group_by(Country)%>% filter(is.na(CustomerID))%>%
summarise(No.of.missing.CustomerID=n())
```

```
## # A tibble: 9 x 2
##   Country        No.of.missing.CustomerID
##   <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 Hint: 1. A close

Average<-OR%>% group_by(CustomerID)%>%
summarise(difference.in.consecutivedays= diff(New_Invoice_Date))%>%
filter(difference.in.consecutivedays>0)
```

```
## `summarise()` has grouped output by 'CustomerID'. You can override using the
## `.groups` argument.
```

```
print(paste('The average number of days between consecutive shopping is',mean(Average$difference.
```

```
## [1] "The average number of days between consecutive shopping is 38.4875000000001"
```

```
#10-In the retail sector, it is very important to understand the return rate of the goods purchased by

Return_value<-nrow(OR%>% group_by(CustomerID)%>% filter((Country=='France')&(TransactionValue<0)&(Custo
total_french_customer<-nrow(OR%>%
group_by(CustomerID)%>% filter((Country=='France')&(CustomerID != 'Na')))
print(paste('Return rate for french customer is given as',((Return_value)/(total_french_customer))*100,
```

```
## [1] "Return rate for french customer is given as 1.75479919915204 percent"
```

```
#11-What is the product that has generated the highest revenue for the retailer? (i.e. item with the hi
Total_customer1<-OR%>%
group_by(Description,StockCode)%>%
summarise(n=sum(TransactionValue))%>%
arrange(desc(n))
```

```
## `summarise()` has grouped output by 'Description'. You can override using the
## `.groups` argument.
```

```
rr<- Total_customer1[Total_customer1['n']==max(Total_customer1['n']),]
print(paste('The highest revenue generated product is', rr$Description,'with stock code',rr$StockCode))
```

```
## [1] "The highest revenue generated product is DOTCOM POSTAGE with stock code DOT"
```

```
#12-How many unique customers are represented in the dataset? You can use unique() and length() functio
print(paste('Total no. of customers with valid customer id are ',length(unique(OR$CustomerID))-1,'. This
```

```
## [1] "Total no. of customers with valid customer id are  4372 . This does not include null CustomerID
```