Business Analytics Assignment 1

Sai Supriya Vengala

10/23/2022

```
#Required libraries are to be loaded first
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.1.2
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
##
## 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
```

```
Online_Retail <- read_csv("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.
```

```
#task1: calculating the countries accounting for more than 1% of the total transactions.
trans_country <-table(Online_Retail$Country)
transaction_percent<- round(100*prop.table(trans_country))
percentage <- cbind(trans_country, transaction_percent)
account <-subset(percentage, transaction_percent >1)
account
```

```
##                trans_country transaction_percent
## EIRE                    8196                   2
## France                  8557                   2
## Germany                 9495                   2
## United Kingdom        495478                  91
```

#task2:Creating a new variable 'TransactionValue' that is the product of the exising #'Quantity' and 'Unit-Price' variables.

```
TransactionValue <-Online_Retail$Quantity * Online_Retail$UnitPrice
Online_Retail <- Online_Retail %>% mutate(TransactionValue)
summary(Online_Retail$TransactionValue)
```

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

#task3:Calculating the countries with total transaction exceeding 130,000 British Pound

```
Sum_trans <- sum(TransactionValue)
store<-summarise(group_by(Online_Retail, Country), Sum_trans)
Total <- filter(store, Sum_trans >130000)
Total
```

```
## # A tibble: 38 x 2
##    Country         Sum_trans
##    <chr>               <dbl>
##  1 Australia        9747748.
##  2 Austria          9747748.
##  3 Bahrain          9747748.
##  4 Belgium          9747748.
##  5 Brazil           9747748.
##  6 Canada           9747748.
##  7 Channel Islands  9747748.
##  8 Cyprus           9747748.
##  9 Czech Republic   9747748.
## 10 Denmark          9747748.
## # ... with 28 more rows
```

#task4

```
Temp=strptime(Online_Retail$InvoiceDate,format='%m/%d/%Y %H:%M',tz='GMT')
Online_Retail$New_Invoice_Date <- as.Date(Temp)
Diff <- Online_Retail$New_Invoice_Date[20000]- Online_Retail$New_Invoice_Date[10]
Diff
```

```
## 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"))
```

#a)Showing the percentage of transactions (by numbers) by #days of the week

```
country_totaltran1<-summarise(group_by(Online_Retail,Invoice_Day_Week) ,Trans_val=n_distinct(InvoiceNo))
percentage1<-mutate(country_totaltran1,
                    Trans_perc=(Trans_val/sum(Trans_val))*100)
percentage1
```

```
## # A tibble: 6 x 3
##   Invoice_Day_Week Trans_val Trans_perc
##   <chr>                <int>      <dbl>
## 1 Friday                4184      16.2
## 2 Monday                4138      16.0
## 3 Sunday                2381       9.19
## 4 Thursday              5660      21.9
## 5 Tuesday               4722      18.2
## 6 Wednesday             4815      18.6
```

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

```
country_tran1<-summarise(group_by(Online_Retail,Invoice_Day_Week),Trans_val1=sum(TransactionValue))
percen1<-mutate(country_tran1,Trans_perc1=(Trans_val1/sum(Trans_val1))*100)
percen1
```

```
## # A tibble: 6 x 3
##   Invoice_Day_Week Trans_val1 Trans_perc1
##   <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
```

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

```
country_totaltran2<-summarise(group_by(Online_Retail,New_Invoice_Month),Trans_val2=sum(TransactionValue
percentage2<-mutate(country_totaltran2,Trans_perc2=(Trans_val2/sum(Trans_val2))*100)
percentage2
```

```
## # A tibble: 12 x 3
##    New_Invoice_Month Trans_val2 Trans_perc2
##                <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
```

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

```
Online_Retail %>% filter(Country == 'Australia') %>% group_by(New_Invoice_Date) %>%
  summarise(max=max(TransactionValue))
```

```
## # A tibble: 49 x 2
##    New_Invoice_Date     max
##    <date>             <dbl>
##  1 2010-12-01         51
##  2 2010-12-08         71.4
##  3 2010-12-14         -6.25
##  4 2010-12-17        148.
##  5 2011-01-06       1020
##  6 2011-01-10         81.6
##  7 2011-01-11         35.4
##  8 2011-01-14        142.
##  9 2011-01-17         47.4
## 10 2011-01-19         38.2
## # ... with 39 more rows
```

#e)Calculating the hour of the day to start this so that the distribution is at #minimum for the customers?

```
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.1.2
```

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

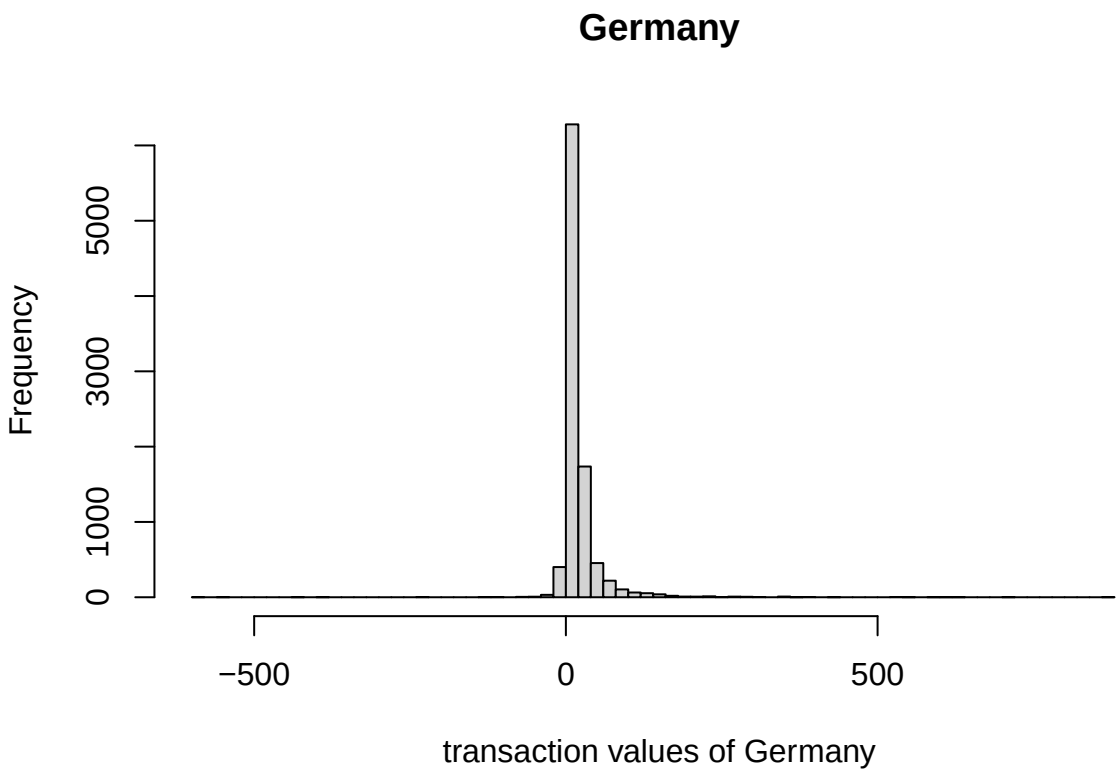
```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
start1<-summarise(group_by(Online_Retail,New_Invoice_Hour),
                  Tran_mini=n_distinct(InvoiceNo))
start1<-filter(start1,New_Invoice_Hour>=7&New_Invoice_Hour<=20)
start2<-rollapply(start1$Tran_mini,3,sum)
start3<-which.min(start2)
start3
```

```
## [1] 12
```

#task5:Plotting the histogram of transaction values from Germany. Using the hist() function to plot.

```
German_Trans <- subset(Online_Retail$TransactionValue, Online_Retail$Country =="Germany")
hist(German_Trans, xlim = c (-600, 900), breaks = 100, xlab = "transaction values of Germany",
     main = "Germany")
```



#task6:Calculating the customer with highest number of transactions most valuable customer. #(i.e. highest total sum of transactions)

```
High_Trans <- na.omit(Online_Retail)
High_Trans <- summarise(group_by(Online_Retail, CustomerID), sum2 = sum(TransactionValue))
High_Trans[which.max(High_Trans$sum2),]
```

```
## # A tibble: 1 x 2
##   CustomerID     sum2
##        <dbl>    <dbl>
## 1         NA 1447682.
```

```
store1 <- table(High_Trans$CustomerID)
store1 <- as.data.frame(store1)
Val_Cust <- store1[which.max(store1$Freq),]
Val_Cust
```

```
##    Var1 Freq
## 1 12346    1
```

#task7:Calculating the percentage of missing values for each variable in the dataset

```
Miss_Val <-colMeans(is.na(Online_Retail)*100)
Miss_Val
```

```
##         InvoiceNo         StockCode       Description          Quantity
##         0.0000000         0.0000000         0.2683107         0.0000000
##       InvoiceDate         UnitPrice        CustomerID           Country
##         0.0000000         0.0000000        24.9266943         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
```

#task8:Calculating the number of transactions with missing CustomerID records by countries?

```
Val_Cust <- Online_Retail %>% filter(is.na(CustomerID)) %>% group_by(Country)
summary(Val_Cust$Country)
```

```
##    Length     Class      Mode
##    135080 character character
```

#task9:Calculating how often the costumers #comeback to the website for their next shopping

```
Freaq <- Online_Retail %>%
  group_by(InvoiceNo, CustomerID, Country, New_Invoice_Date, New_Invoice_Month, New_Invoice_Hour,
           Invoice_Day_Week) %>%
  summarise(Trans6 = sum(TransactionValue)) %>%
  mutate(Freq1 = Sys.Date() - New_Invoice_Date) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'InvoiceNo', 'CustomerID', 'Country',
## 'New_Invoice_Date', 'New_Invoice_Month', 'New_Invoice_Hour'. You can override
## using the `.groups` argument.
```

```
Freaq$Freq1 <- as.character(Freaq$Freq1)
Freaq$Freq2 <- sapply(Freaq$Freq1,
  FUN = function(x) {strsplit(x, split = '[ ]')[[1]][1]})
Freaq$Freq2 <- as.integer(Freaq$Freq2)
head(Freaq, n = 5)
```

```
## # A tibble: 5 x 10
##   Invoic~1 Custo~2 Country New_Invo~3 New_I~4 New_I~5 Invoi~6 Trans6 Freq1 Freq2
##   <chr>      <dbl> <chr>   <date>       <dbl>   <dbl> <chr>    <dbl> <chr> <int>
## 1 536365     17850 United~ 2010-12-01      12       8 Wednes~  139.  4345   4345
## 2 536366     17850 United~ 2010-12-01      12       8 Wednes~   22.2 4345   4345
## 3 536367     13047 United~ 2010-12-01      12       8 Wednes~  279.  4345   4345
## 4 536368     13047 United~ 2010-12-01      12       8 Wednes~   70.1 4345   4345
## 5 536369     13047 United~ 2010-12-01      12       8 Wednes~   17.8 4345   4345
## # ... with abbreviated variable names 1: InvoiceNo, 2: CustomerID,
## #   3: New_Invoice_Date, 4: New_Invoice_Month, 5: New_Invoice_Hour,
## #   6: Invoice_Day_Week
```

```
attach(Freaq)
FreaqCust <- Online_Retail %>%
  group_by(CustomerID, Country) %>%
```

```
  summarise(Cust_order = n_distinct(InvoiceNo),
            Trans7 = sum(TransactionValue),
  PerDay = names(which.max(table(Invoice_Day_Week))),
PerHour=names(which.max(table(New_Invoice_Hour))),
            Frequency = min(Freaq$Freq2))%>%
  ungroup()
```

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

```
head(FreaqCust)
```

```
## # A tibble: 6 x 7
##   CustomerID Country        Cust_order Trans7 PerDay    PerHour Frequency
##        <dbl> <chr>               <int>  <dbl> <chr>     <chr>       <int>
## 1      12346 United Kingdom          2      0  Tuesday   10           3972
## 2      12347 Iceland                 7   4310  Tuesday   14           3972
## 3      12348 Finland                 4   1797. Thursday  19           3972
## 4      12349 Italy                   1   1758. Monday    9            3972
## 5      12350 Norway                  1    334. Wednesday 16           3972
## 6      12352 Norway                 11   1545. Tuesday   14           3972
```

#task10:Calculating the return rate for the French customers.

```
France_Trans <- filter(Online_Retail, Country=="France")
Trow <- nrow(France_Trans)
Cancel_Trans <- nrow(subset(France_Trans,TransactionValue<0))
Cancel_Trans
```

```
## [1] 149
```

```
No_Cancel<- Trow-Cancel_Trans
No_Cancel
```

```
## [1] 8408
```

```
Return=(Cancel_Trans/8556)
Return
```

```
## [1] 0.01741468
```

#task11:Calculating the product that has generated the highest revenue for the retailer. #(i.e. item with the highest total sum of 'TransactionValue').

```
TransactionValue <- tapply(Online_Retail$TransactionValue, Online_Retail$StockCode, sum)
TransactionValue[which.max(TransactionValue)] # to find highest value
```

```
##      DOT
## 206245.5
```

#task12:Finding the unique customers that are represented in the dataset using unique() function

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

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