Assignment_2

Krishna Kumar Tavva - 811283461

2023-03-09

install functions and call libraries needed

```
library(ISLR)
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
```

Load the Online Retail data into R

```
OR <- read.csv("E:/Business Analyst/Module 4/Online_Retail.csv")
summary(OR)
```

```
##   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
```

1.Show the breakdown of the number of transactions by countries i.e., how many transactions are in the dataset for each country. Show this in total number and also in percentage. Show only countries accounting for more than 1% of the total transactions.

```
OR %>% group_by(Country) %>% summarise(Total_Trans=n(), Total_Perc = sum(n()/nrow(OR))*100) %>%
  filter(Total_Perc>1)
```

```
## # A tibble: 4 x 3
##   Country        Total_Trans Total_Perc
##   <chr>                <int>      <dbl>
## 1 EIRE                  8196       1.51
## 2 France                8557       1.58
## 3 Germany               9495       1.75
## 4 United Kingdom      495478      91.4
```

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

```
TransactionValue <- OR$Quantity*OR$UnitPrice
OR <- cbind(OR, TransactionValue)
colnames(OR)
```

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

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.

```
OR %>% group_by(Country) %>% summarise(Total = sum(TransactionValue)) %>%
  filter(Total >= 130000) %>% arrange(desc(Total))
```

```
## # A tibble: 6 x 2
##   Country           Total
##   <chr>             <dbl>
## 1 United Kingdom 8187806.
```

```
## 2 Netherlands     284662.
## 3 EIRE            263277.
## 4 Germany         221698.
## 5 France          197404.
## 6 Australia       137077.
```

4.Converting Invoice Date into a POSIXlt object

```
Temp=strptime(OR$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"
```

```
#New_Invoice_Date
OR$New_Invoice_Date <- as.Date(Temp)

OR$New_Invoice_Date[20000]- OR$New_Invoice_Date[10]
```

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

```
#Invoice_Day_Week
OR$Invoice_Day_Week= weekdays(OR$New_Invoice_Date)

#New_Invoice_Hour
OR$New_Invoice_Hour = as.numeric(format(Temp, "%H"))

#New_Invoice_Month
OR$New_Invoice_Month = as.numeric(format(Temp, "%m"))

#4(a).Percentage of transactions (by numbers) by days of the week

OR %>% group_by(Invoice_Day_Week) %>% summarise(count=n()) %>% mutate(Percentage=count/nrow(OR)*100)
```

```
## # A tibble: 6 x 3
##   Invoice_Day_Week  count Percentage
##   <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).Percentage of transactions (by transaction volume) by days of the week

OR %>% group_by(Invoice_Day_Week) %>% summarise(Total= sum(TransactionValue)) %>%
  mutate(Percentage = Total/sum(Total)*100)
```

```
## # A tibble: 6 x 3
##   Invoice_Day_Week    Total Percentage
##   <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).Percentage of transactions (by transaction volume) by month of the year

OR %>% group_by(New_Invoice_Month) %>% summarise(Total = sum(TransactionValue))%>%
  mutate(Percentage = Total/sum(Total)*100)
```

```
## # A tibble: 12 x 3
##    New_Invoice_Month    Total Percentage
##                <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).The date with the highest number of transactions from Australia

OR %>% filter(Country =="Australia") %>% group_by(New_Invoice_Date) %>%
  summarise(Total_Count = n()) %>% arrange((desc(Total_Count)))
```

```
## # A tibble: 49 x 2
##    New_Invoice_Date Total_Count
##    <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
```

```
#4(e).The company needs to shut down the website for two consecutive hours for maintenance. What would
```

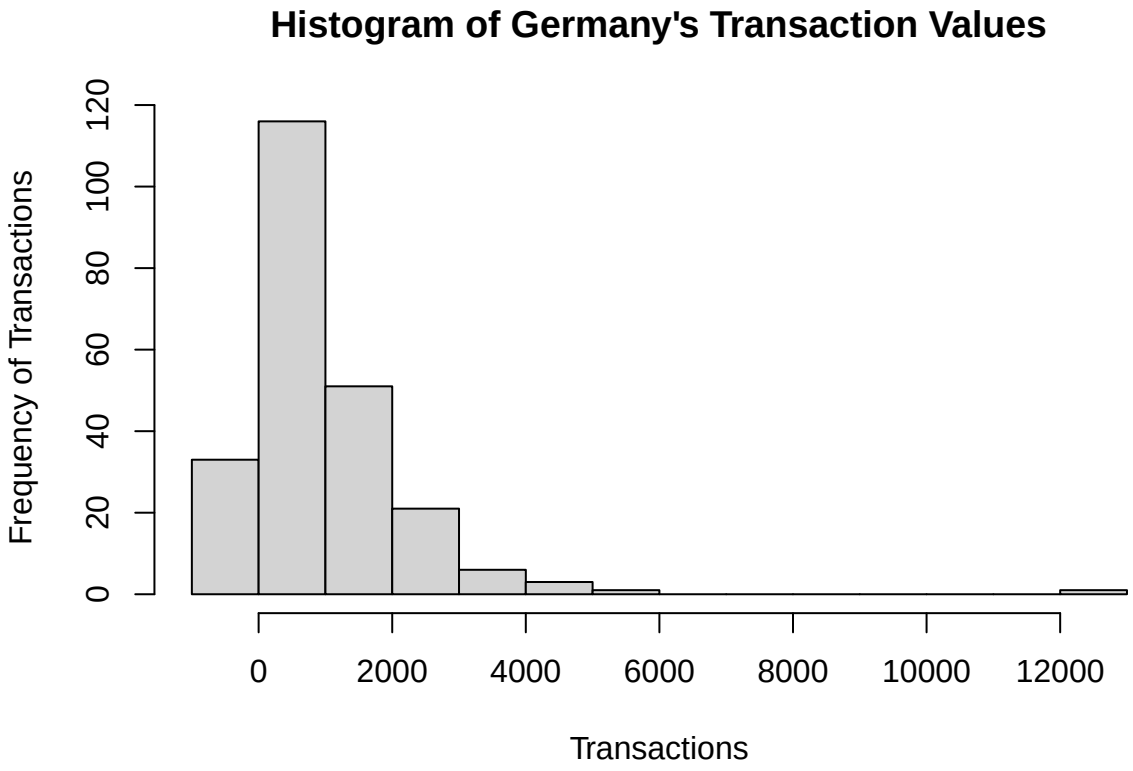
```
transactVolumeByHour = OR %>%
  group_by(New_Invoice_Hour) %>% summarise(TransactionVolume =n())

transactVolumeByHour = transactVolumeByHour %>%
  arrange(New_Invoice_Hour) %>%
  mutate(TwoConsecutiveHourTraffic=TransactionVolume+lead(TransactionVolume),NextHour=lead(New_Invoice_
  filter(New_Invoice_Hour>=7,New_Invoice_Hour<19)

transactVolumeByHour %>%
  filter(TwoConsecutiveHourTraffic== min(transactVolumeByHour$TwoConsecutiveHourTraffic))
```

```
## # A tibble: 1 x 4
##   New_Invoice_Hour TransactionVolume TwoConsecutiveHourTraffic NextHour
##              <dbl>             <int>                     <int>    <dbl>
## 1                7               383                      9292        8
```

```
#transactVolumeByHour = OR %>% group_by(New_Invoice_Hour) %>% summarise(TransactionVolume =n()) %>% arr
```

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

```
Germany <- OR %>% filter(Country == "Germany") %>%
  group_by(New_Invoice_Date) %>%
summarise(Total=sum(TransactionValue))
hist(Germany$Total, main = "Histogram of Germany's Transaction Values",
     xlab="Transactions", ylab="Frequency of Transactions")
```



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

```
OR %>% group_by(CustomerID) %>% summarise(Total_Transactions = n()) %>%
  arrange((desc(Total_Transactions))) %>% filter(!is.na(CustomerID))
```

```
## # A tibble: 4,372 x 2
##    CustomerID Total_Transactions
##         <int>              <int>
##  1      17841               7983
##  2      14911               5903
##  3      14096               5128
##  4      12748               4642
##  5      14606               2782
##  6      15311               2491
##  7      14646               2085
##  8      13089               1857
##  9      13263               1677
## 10      14298               1640
## # ... with 4,362 more rows
```

#Customer ID 17841 is having highest number of transactions (excluding NA)

```
OR %>% group_by(CustomerID) %>% summarise(Spending_max = sum(TransactionValue)) %>%
  arrange((desc(Spending_max))) %>% filter(!is.na(CustomerID))
```

```
## # A tibble: 4,372 x 2
##    CustomerID Spending_max
##         <int>        <dbl>
##  1      14646      279489.
##  2      18102      256438.
##  3      17450      187482.
##  4      14911      132573.
##  5      12415      123725.
##  6      14156      113384.
##  7      17511       88125.
##  8      16684       65892.
##  9      13694       62653.
## 10      15311       59419.
## # ... with 4,362 more rows
```

#Customer ID 14646 is having highest total sum of transactions (excluding NA)

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

```
colMeans(is.na(OR))
```

```
##         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
```

#For the Customer ID, 24.92669% variables are missing.

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

```
OR %>% filter(is.na(CustomerID)) %>% group_by(Country) %>% count()
```

```
## # 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)

```
Days_Gap <- OR %>% group_by(CustomerID) %>% distinct(New_Invoice_Date) %>%
  arrange(desc(CustomerID)) %>%
  mutate(Past_Date=lag(New_Invoice_Date), Days_Between = New_Invoice_Date-lag(New_Invoice_Date)) %>%
  filter(!is.na(Days_Between))

Days_Gap
```

```
## # A tibble: 15,200 x 4
## # Groups:   CustomerID [2,992]
##    CustomerID New_Invoice_Date Past_Date  Days_Between
##         <int> <date>           <date>     <drtn>
##  1      18287 2011-10-12       2011-05-22 143 days
##  2      18287 2011-10-28       2011-10-12  16 days
##  3      18283 2011-01-23       2011-01-06  17 days
##  4      18283 2011-02-28       2011-01-23  36 days
##  5      18283 2011-04-21       2011-02-28  52 days
##  6      18283 2011-05-23       2011-04-21  32 days
##  7      18283 2011-06-14       2011-05-23  22 days
##  8      18283 2011-06-23       2011-06-14   9 days
##  9      18283 2011-07-14       2011-06-23  21 days
## 10      18283 2011-09-05       2011-07-14  53 days
## # ... with 15,190 more rows
```

```
mean(Days_Gap$Days_Between)
```

```
## Time difference of 38.4875 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?

```
France_Cancel <- OR %>% filter(Country=="France",Quantity<0) %>% count()

France_Total <- OR %>% filter(Country=="France") %>% count()

Return_Rate_of_France <- France_Cancel/France_Total*100
Return_Rate_of_France
```

```
##          n
## 1 1.741264
```

#The return rate for the French customers is 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').

```
OR %>% group_by(Description) %>%summarise(Total=sum(TransactionValue)) %>%
  arrange(desc(Total))
```

```
## # A tibble: 4,224 x 2
##    Description                          Total
##    <chr>                                <dbl>
##  1 "DOTCOM POSTAGE"                    206245.
##  2 "REGENCY CAKESTAND 3 TIER"          164762.
##  3 "WHITE HANGING HEART T-LIGHT HOLDER"  99668.
##  4 "PARTY BUNTING"                      98303.
##  5 "JUMBO BAG RED RETROSPOT"            92356.
##  6 "RABBIT NIGHT LIGHT"                 66757.
##  7 "POSTAGE"                            66231.
##  8 "PAPER CHAIN KIT 50'S CHRISTMAS "    63792.
##  9 "ASSORTED COLOUR BIRD ORNAMENT"      58960.
## 10 "CHILLI LIGHTS"                      53768.
## # ... with 4,214 more rows
```

#DOTCOM POSTAGE is the highest revenue for the retailer

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

```
OR %>% select(CustomerID) %>% unique() %>% count()
```

```
##      n
## 1 4373
```

#4373 unique customers are represeted in the dataset