BA Assignment-2

Sai Kiran

2023-03-11

```
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(zoo)
```

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

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

```
set.seed(158)
library(readxl)
```

```
Online_Retail<-read.csv("C:/Users/panug/Downloads/Online_Retail.csv")
```

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.

```
Online_Retail %>%group_by(Country)%>%summarise(transactions=n())%>%mutate(percentage=(transactions/5419
```

```
## # 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 'TransactionValue' that is the product of the exising 'Quantity' and 'UnitPrice' variables. Add this variable to the dataframe

```
Online_Retail<- mutate(Online_Retail, "TransactionValue"=TransactionValue<- Online_Retail$Quantity * On
colnames(Online_Retail)
```

```
## [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.

```
Online_Retail%>%group_by(Country)%>%summarise(total.sum.of.transaction.values=sum(TransactionValue))%>%
```

```
## # 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 #Converting the "InvoiceDate" column into a POSIXlt object:

```
Temp=strptime(Online_Retail$InvoiceDate,format='%m/%d/%Y %H:%M',tz='GMT')
```

#dividing the components of the dataframe into three separate categories, namely date, day of the week, and hour. These categories are labeled as New Invoice Date, Invoice Day Week, and New Invoice Hour:

```
Online_Retail$New_Invoice_Date<-as.Date(Temp)
```

#Having knowledge of two date values enables you to calculate the duration between them in terms of the number of days:

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

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

#Dates can be converted to weekdays. For that, let's create a new variable.

```
Online_Retail$Invoice_Day_Week=weekdays(Online_Retail$New_Invoice_Date)
```

#Let's just turn the hour into a standard numerical value for the hour (ignore the minute):

```
Online_Retail$New_Invoice_Hour =as.numeric(format(Temp,"%H"))
```

#defining the month as a separate numeric variable too:

```
Online_Retail$New_Invoice_Month = as.numeric(format(Temp, "%m"))
```

Now answer the flowing questions 4.a) Show the percentage of transactions (by numbers) by days of the week (extra 1% of total points)

```
Online_Retail%>%
  group_by(Invoice_Day_Week)%>%
  summarise(Number.of.transaction=(n()))%>%
  mutate(Number.of.transaction,'percent'=(Number.of.transaction*100)/sum(Number.of.transaction))
```

```
## # 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) bydays of the week

```
Online_Retail%>%
  group_by(Invoice_Day_Week)%>%
  summarise(Volume.of.transaction=(sum(TransactionValue)))%>%
  mutate(Volume.of.transaction,'percent'=(Volume.of.transaction*100)/sum(Volume.of.transaction))
```

```
## # 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

```
Online_Retail%>%group_by(New_Invoice_Month)%>%summarise(Volume.By.Month=sum(TransactionValue))%>%mutate
```

```
## # 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?

```
Online_Retail <- Online_Retail %>% mutate(Transactionvalue= Quantity * UnitPrice)
Online_Retail %>% filter(Country == 'Australia') %>% group_by(New_Invoice_Date) %>% summarise(max=max(T
```

```
## # 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
```

4.e)The company needs to shut down the website for twovconsecutive 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

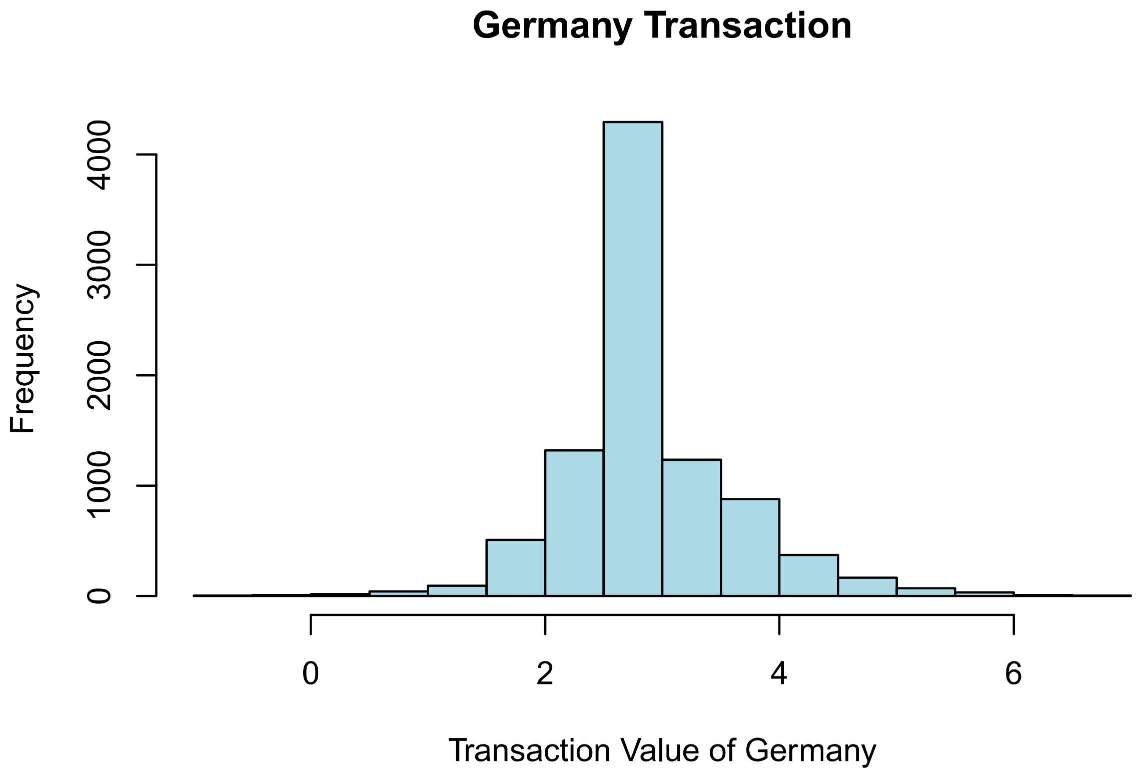
```
Hr1<-summarise(group_by(Online_Retail,New_Invoice_Hour),Transaction_min=n_distinct(InvoiceNo))
Hr1<-filter(Hr1,New_Invoice_Hour>=7&New_Invoice_Hour<=20)
Hr2<-rollapply(Hr1$Transaction_min,2,sum)
Hr3<-which.min(Hr2)
Hr3
```

```
## [1] 13
```

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

```
hist(x=log(Online_Retail$TransactionValue[Online_Retail$Country=="Germany"]),xlab = "Transaction Value
```

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



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

```
Online_Customer <- na.omit(Online_Retail)
result_data <- summarise(group_by(Online_Customer,CustomerID), sum.data= sum(Transactionvalue))
result_data[which.max(result_data$sum.data),]
```

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

```
Cust_data <- table(Online_Retail$CustomerID)
Cust_data <- as.data.frame(Cust_data)
result_data_2 <- Cust_data[which.max(Cust_data$Freq),]
result_data_2
```

```
##       Var1 Freq
## 4043 17841 7983
```

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

```
missing_values<-colMeans(is.na(Online_Retail))
print(paste('Online customerID column in dataset lacks values  i.e.',missing_values['CustomerID']*100,'
## [1] "Online customerID column in dataset lacks values  i.e. 24.9266943342886 % of whole data"
```

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

#Out of the total number of eight nations and one unnamed country that had missing values in the dataset, the United Kingdom has the highest number of such records, with 133,600 rows.

```
Online_Retail%>%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? (i.e. what is the average number of days between consecutive shopping)

```
avg<-Online_Retail%>%group_by(CustomerID)%>%summarise(difference.in.consecutivedays=diff(New_Invoice_Da
```

```
## Warning: Returning more (or less) than 1 row per ‘summarise()‘ group was deprecated in
## dplyr 1.1.0.
## i Please use ‘reframe()‘ instead.
## i When switching from ‘summarise()‘ to ‘reframe()‘, remember that ‘reframe()‘
##   always returns an ungrouped data frame and adjust accordingly.
```

```
## ‘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(avg$difference.in.c
```

```
## [1] "the average  number  of  days  between  consecutive  shopping is 38.4875"
```

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?

```
returnvalue<-nrow(Online_Retail%>%group_by(CustomerID)%>%filter((Country=='France')&(TransactionValue<0
total.fcustomer<-nrow(Online_Retail%>%group_by(CustomerID)%>%filter((Country=='France')&(CustomerID !=
print(paste('For French customers, the return rate is provided as',((returnvalue)/(total.fcustomer))*10
```

```
## [1] "For French customers, the return rate is provided as 1.75479919915204 Percent"
```

11. What is 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)]
```

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

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

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

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