Business Analytics- Assignment-2

Elmy Luka

```
Online_Retail <- read.csv("~/Desktop/MS BA/Business Analytics/Assignment-2/Online_Retail.csv")

#TASK-1

#Showing the number of transaction by countries i.e
#the number of transactions in the dataset for each country.
#(Considering all records including cancelled transactions)
total_transactions.by.country <- table(Online_Retail$Country)
#Showing the number of transactions in the dataset for
#each country in total number and also in percentage.
transaction_percent<-round(100*prop.table(total_transactions.by.country))
percentage <- cbind(total_transactions.by.country, transaction_percent)
#Countries accunting more than 1%
result <- subset(percentage, transaction_percent >1)
result
```

```
##                total_transactions.by.country transaction_percent
## EIRE                                    8196                   2
## France                                  8557                   2
## Germany                                 9495                   2
## United Kingdom                        495478                  91
```

```
#TASK-2

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
```

```
#Creating a new variable 'TransactionValue' that is
#the product of the existing 'Quantity' and 'UnitPrice'
#variables and adding this variable to the dataframe.

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
```

```
#TASK-3

#total sum of transaction i.e money in total that has been
#spent by each country
sum <- sum(TransactionValue)
data <- summarise(group_by(Online_Retail, Country), sum)

#Countries with transaction exceeding 130,000 British Pound
exceed_transaction <- filter(data, sum > 130000)
exceed_transaction
```

```
## # A tibble: 38 x 2
##    Country              sum
##    <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
```

```
#TASK-4
#Creating a POSIXlt to object from "InvoiceDate":
Temp=strptime(Online_Retail$InvoiceDate,format='%m/%d/%Y %H:%M',tz='GMT')

#spliting the dataframe components for the date,
#day of the week, and hour under the labels New_Invoice_Date, Invoice_Day_Week, and New_Invoice_ Hour

Online_Retail$New_Invoice_Date<-as.Date(Temp)

#determining two date values gives the ability to
#analyse how many days are between the two dates.
Online_Retail$New_Invoice_Date[20000]-Online_Retail$New_Invoice_Date[10]
```

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

```
#Creating a new variable to convert dates to weekdays.
Online_Retail$Invoice_Day_Week=weekdays(Online_Retail$New_Invoice_Date)

#turning 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
```

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

#Answering the following 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_transactions=(n()))%>%
  mutate(Number_of_transactions,'Percentage'=(Number_of_transactions*100)/sum(Number_of_transactions))
```

```
## # A tibble: 6 x 3
##   Invoice_Day_Week Number_of_transactions 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)Show the percentage of transactions
#(by transaction volume) bydays of the week
Online_Retail%>%
  group_by(Invoice_Day_Week)%>%
  summarise(Volume_of_transactions=(sum(TransactionValue)))%>%
  mutate(Volume_of_transactions,'Percentage'=(Volume_of_transactions*100)/sum(Volume_of_transactions))
```

```
## # A tibble: 6 x 3
##   Invoice_Day_Week Volume_of_transactions 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)Show the percentage of transactions
#(by transaction volume) by month of the year
Online_Retail%>%group_by(New_Invoice_Month)%>%
  summarise(Volume_Transaction_By_Month=sum(TransactionValue))%>%
  mutate(Volume_Transaction_By_Month,        'Percentage'=(Volume_Transaction_By_Month*100)/
           sum(Volume_Transaction_By_Month))
```

```
## # A tibble: 12 x 3
##    New_Invoice_Month Volume_Transaction_By_Month 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)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(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
```

```
#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.
library(zoo)
```

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

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

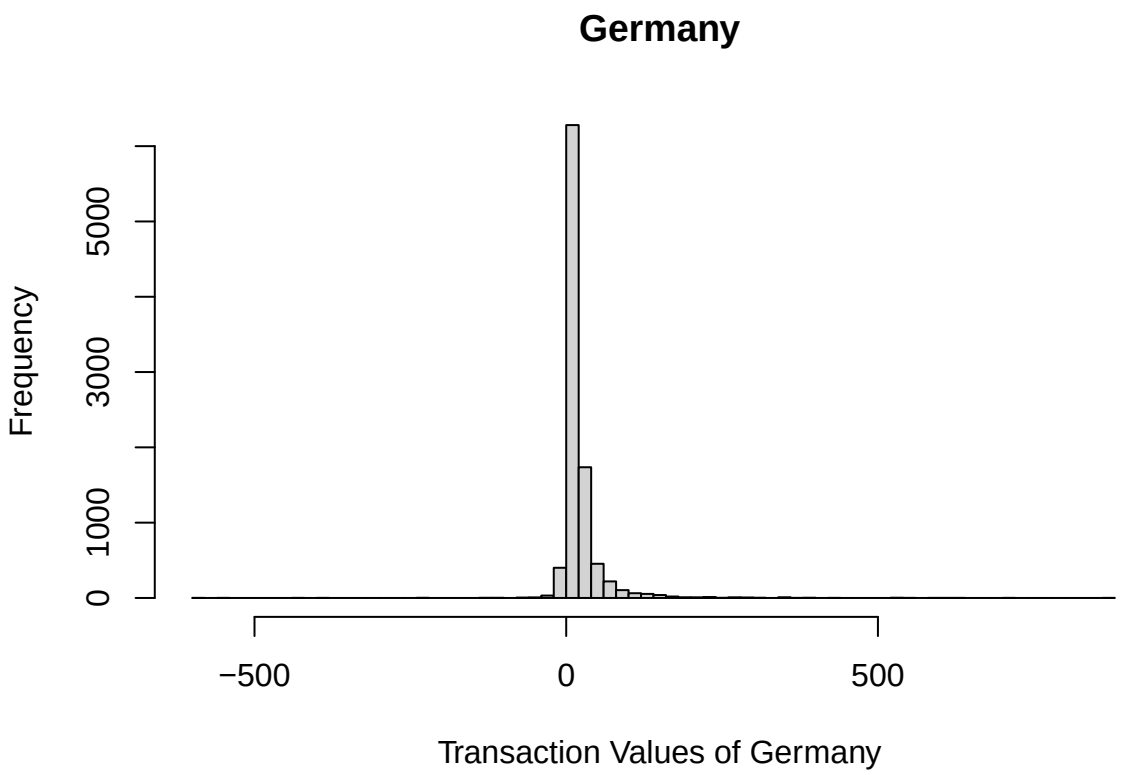
```
hour<-summarise(group_by(Online_Retail,New_Invoice_Hour),Transaction_min=n_distinct(InvoiceNo))
hour<-filter(hour,New_Invoice_Hour>=7&New_Invoice_Hour<=20)
hour_2<-rollapply(hour$Transaction_min,2,sum)
hour_3<-which.min(hour_2)
hour_3
```

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

```
#TASK-5

#Plotting the histogram of transaction values from Germany.
#(Using the hist() function to plot)
Germany_transaction_value <- subset(Online_Retail$TransactionValue, Online_Retail$Country == "Germany")

hist(Germany_transaction_value, xlim = c (-600, 900),
     breaks = 100 , xlab = "Transaction Values of Germany", main = "Germany")
```




```
#TASK-6

#Finding the customer who had the highest number of
#transactions and also who is the most valuable
#(i.e. the customer with highest total sum of transactions)
retail_1 <- na.omit(Online_Retail)
result_1 <- summarise(group_by(retail_1,CustomerID), sum2= sum(TransactionValue))
result_1[which.max(result_1$sum2),]
```

```
## # A tibble: 1 x 2
##   CustomerID    sum2
##        <int>   <dbl>
## 1      14646 279489.
```

```
data_1 <- table(Online_Retail$CustomerID)
data_1 <- as.data.frame(data_1)
result_2 <- data_1[which.max(data_1$Freq),]
result_2
```

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

```
#TASK-7

#Calculating the percentage of missing
#values for each variable in the dataset.
missing_values <- colMeans(is.na(Online_Retail)*100)
missing_values
```

```
##         InvoiceNo         StockCode       Description          Quantity
##           0.00000           0.00000           0.00000           0.00000
##       InvoiceDate         UnitPrice        CustomerID           Country
##           0.00000           0.00000          24.92669           0.00000
##  TransactionValue  New_Invoice_Date  Invoice_Day_Week  New_Invoice_Hour
##           0.00000           0.00000           0.00000           0.00000
## New_Invoice_Month
##           0.00000
```

```
#TASK-8

#Finding the number of transactions with
#missing CustomerID records by countries.
retail_2 <- Online_Retail %>% filter(is.na(CustomerID)) %>% group_by(Country)
summary(retail_2$Country)
```

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

```
#TASK-9

#Finding on an average how often the costumers
#comeback to the website for their next shopping
#(i.e. finding the average number of days
#between consecutive shopping)

average_1<-Online_Retail%>%group_by(CustomerID)%>%
  summarise(difference_in_consecutive_days=diff(New_Invoice_Date))%>%
  filter(difference_in_consecutive_days>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_1$difference_in_consecutive_days)))
```

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

```
#TASK-10

#With this definition, what is the return
```

```r
#rate for the French customers?
retail_table <- filter(Online_Retail,Country=="France")
total_row <- nrow(retail_table)

#(10 marks). Consider the cancelled transactions
#as those where the 'Quantity'
#variable has a negative value.
cancel <- nrow(subset(retail_table,TransactionValue<0))
cancel
```

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

```r
non_cancelled <- total_row-cancel
non_cancelled
```

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

```r
test_1=(cancel/8556)
test_1
```

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

```r
#TASK-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
```

```r
#TASK-12
#Finding the number of unique customers who
#are represented in the dataset using unique() and
#length() functions.
unique_customers <- unique(Online_Retail$CustomerID)
length(unique_customers)
```

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