pratheek sreerangam

2022-10-21

#import dataset

```
retail<-read.csv("C:/Users/prath/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.

```
summary(retail)
```

```
##    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
```

```
summary(retail$Country)
```

```
##    Length     Class      Mode
##    541909 character character
```

```
country_totalnumber<-table(retail$Country)
transaction_percent<-round(100*prop.table(country_totalnumber),digits = 2)
percentage<-cbind(country_totalnumber,transaction_percent)
total<-subset(percentage,transaction_percent>1)
total
```

```
##                country_totalnumber transaction_percent
## EIRE                          8196                1.51
## France                        8557                1.58
## Germany                       9495                1.75
## United Kingdom              495478               91.43
```

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

```
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
```

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

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

#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

```
data <- summarise(group_by(retail,Country),sum_1= sum(TransactionValue))
Transaction <- filter(data,sum_1 >130000)
Transaction
```

```
## # A tibble: 6 x 2
##   Country          sum_1
##   <chr>            <dbl>
## 1 Australia       137077.
## 2 EIRE            263277.
## 3 France          197404.
## 4 Germany         221698.
## 5 Netherlands     284662.
## 6 United Kingdom 8187806.
```

#4.he variable is read as a categorical when you read data from the file. Now we need to explicitly instruct R to interpret this as a Date variable."POSIXlt" and "POSIXct" are two powerful object classes in R to deal with date and time. irst let's convert 'InvoiceDate' into a POSIXlt object:

```
Temp=strptime(retail$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"
```

#Now, let's split the dataframe's components for the date, day of the week, and hour under the labels New Invoice Date, Invoice Day Week, and New Invoice Hour:

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

#There are many versatile functions available for Date objects. For instance, if you are aware of two date values, the object enables you to determine how many days separate the dates.

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

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

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

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

Ignoring the minute, let's just take the hour and convert it to a regular integer value:

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

#Create a distinct numerical variable for the month as well:

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

4.a) Show the percentage of transactions (by numbers) by days of the week.

```
a<-summarise(group_by(retail,Invoice_Day_Week),Transaction_Value=n_distinct(InvoiceNo))
a1<-mutate(a, transaction_percent=(Transaction_Value/sum(Transaction_Value))*100)
a1
```

```
## # A tibble: 6 x 3
##   Invoice_Day_Week Transaction_Value transaction_percent
##   <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
```

4. b)Show the percentage of transactions (by transaction volume) by days of the week

```
b1<-summarise(group_by(retail,Invoice_Day_Week),Transaction_Volume=sum(TransactionValue))
b2<-mutate(b1,percentage=(Transaction_Volume/sum(Transaction_Volume))*100)
b2
```

```
## # A tibble: 6 x 3
##   Invoice_Day_Week Transaction_Volume 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

```
c1<-summarise(group_by(retail,New_Invoice_Month),Transaction_Volume=sum(TransactionValue))
c1<-mutate(c1,percentage=(Transaction_Volume/sum(Transaction_Volume))*100)
c1
```

```
## # A tibble: 12 x 3
##    New_Invoice_Month Transaction_Volume 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
```

7. d)What was the date with the highest number of transactions from Australia?

```
retail <- retail %>% mutate(TransactionValue= Quantity * UnitPrice)
retail %>% filter(Country == 'Australia') %>% group_by(New_Invoice_Date) %>% summarise(max=max(Transact
```

```
## # 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
```

```
e1<-summarise(group_by(retail,New_Invoice_Hour),Transaction_min=n_distinct(InvoiceNo))
e1<-filter(e1,New_Invoice_Hour>=7&New_Invoice_Hour<=20)
e12<-rollapply(e1$Transaction_min,3,sum)
e123<-which.min(e12)
e123
```

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

starting the work at 12noon is correct for maintenance.

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

```
Germany_data <- subset(retail$TransactionValue, retail$Country == "Germany")
hist(Germany_data, xlim = c (-600, 900), breaks = 100 , xlab = "Transaction Values of Germany", main =
```

Germany

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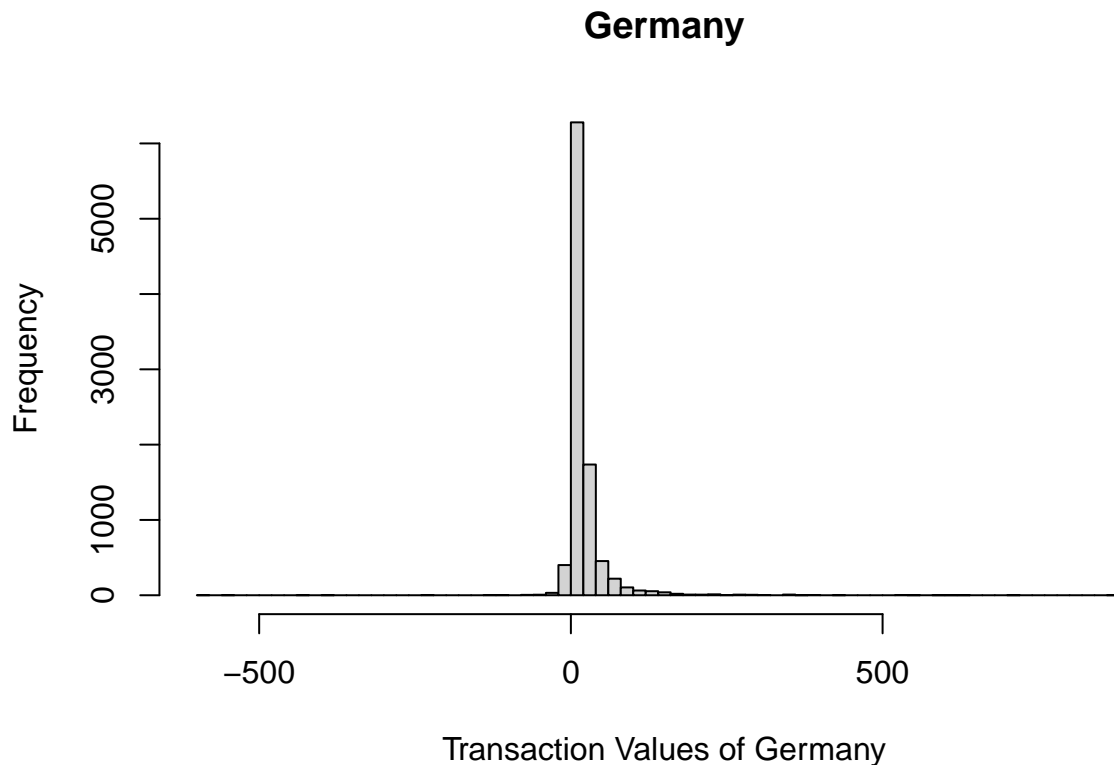
#6. Which customer had the highest number of transactions? Which customer is most valuable (i.e. highest total sum of transactions)?

```
retail1 <- na.omit(retail)
result1 <- summarise(group_by(retail1,CustomerID), sum2= sum(TransactionValue))
result1[which.max(result1$sum2),]
```

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

```
data2 <- table(retail$CustomerID)
data2 <- as.data.frame(data2)
result2 <- data2[which.max(data2$Freq),]
result2
```

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

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

```
missing_values <- colMeans(is.na(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
```

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

```
retail_2 <- retail %>% filter(is.na(CustomerID)) %>% group_by(Country)
summary(retail_2$Country)
```

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

#10.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 transaction with this definition, what is the return rate for the French customers?

```
retail_table <- filter(retail,Country=="France")
totalrow <- nrow(retail_table)
cancel <- nrow(subset(retail_table,TransactionValue<0))
cancel
```

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

```
notcancel <- totalrow-cancel
notcancel
```

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

```
TEST2=(cancel/8556)
TEST2
```

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

#11.What is the product that has generated the highest revenue for the retailer? (i.e. item with the highest total sum of TransactionValue').

```
Transaction_Value <- tapply(retail$TransactionValue, retail$StockCode  , sum)
Transaction_Value[which.max(Transaction_Value)]
```

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

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

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

[1] 4373