BA_64036_Assignment_2

Yeswanth Siripurapu

2023-10-15

```
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
```

#Reading the CSV File

```
OnlineRetail<- read.csv("D:/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.

```
Countries_count <- OnlineRetail %>% group_by(Country) %>% count(Country)
Countries_pct <- OnlineRetail %>% group_by(Country) %>% summarise(percent = 100* n()/nrow(OnlineRetail))
Fltrd_Cntry_pct <- filter(Countries_pct, percent>1)

#Countries Count
Countries_count
```

```
## # A tibble: 38 x 2
## # Groups:   Country [38]
##    Country             n
##    <chr>           <int>
##  1 Australia        1259
##  2 Austria           401
##  3 Bahrain            19
##  4 Belgium          2069
##  5 Brazil             32
##  6 Canada            151
##  7 Channel Islands   758
```

```
##  8 Cyprus            622
##  9 Czech Republic     30
## 10 Denmark           389
## # i 28 more rows
```

#Percentage of transactions greater than 1

```
Fltrd_Cntry_pct
```

```
## # A tibble: 4 x 2
##   Country        percent
##   <chr>            <dbl>
## 1 EIRE              1.51
## 2 France            1.58
## 3 Germany           1.75
## 4 United Kingdom   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 = (OnlineRetail$Quantity * OnlineRetail$UnitPrice)

#Adding the TransactionValue column to the OnlineRetail table
Online_Retail = cbind(OnlineRetail,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.

```
Trans_sum = Online_Retail %>% group_by(Country) %>%
  summarise(sum=sum(TransactionValue))

Fltrd_Trans_sum = filter(Trans_sum,Trans_sum$sum>130000)

#Sum of TransactionValue for each countries
Trans_sum
```

```
## # A tibble: 38 x 2
##    Country             sum
##    <chr>             <dbl>
##  1 Australia       137077.
##  2 Austria          10154.
##  3 Bahrain            548.
##  4 Belgium          40911.
##  5 Brazil            1144.
##  6 Canada            3666.
##  7 Channel Islands  20086.
##  8 Cyprus           12946.
##  9 Czech Republic     708.
## 10 Denmark          18768.
## # i 28 more rows
```

#Filtering the transactions greater than 130000

```
Fltrd_Trans_sum
```

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

#4 This is an optional question which carries additional marks #(golden questions). In this question,we are dealing with the InvoiceDate #variable. The 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.

```
Temp=strptime(Online_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"
```

```
Online_Retail$New_Invoice_Date <- as.Date(Temp)
Online_Retail$Invoice_Day_Week= weekdays(Online_Retail$New_Invoice_Date)
Online_Retail$New_Invoice_Hour = as.numeric(format(Temp, "%H"))
Online_Retail$New_Invoice_Month = as.numeric(format(Temp, "%m"))
Online_Retail$New_Invoice_Date[20000]- Online_Retail$New_Invoice_Date[10]
```

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

#4(a)

```
#Percentage of number of transactions based on week days
Week_days_count = Online_Retail %>% group_by(Invoice_Day_Week) %>%
  summarise(percent = 100* n()/nrow(Online_Retail))
Week_days_count
```

```
## # A tibble: 6 x 2
##   Invoice_Day_Week percent
##   <chr>              <dbl>
## 1 Friday              15.2
## 2 Monday              17.6
## 3 Sunday              11.9
## 4 Thursday            19.2
## 5 Tuesday             18.8
## 6 Wednesday           17.5
```

#4(b)

```r
#percentage of TransactionsValue
Week_days_sum = Online_Retail %>% group_by(Invoice_Day_Week) %>% summarise(sum=sum(TransactionValue))
#Calculating the percentage for TransactionValue by week days
Week_quan_pct = 100*(Week_days_sum$sum)/sum(Week_days_sum$sum)
#replacing the sum with the percentage value
Week_days_sum$sum = Week_quan_pct
Week_days_sum
```

```
## # A tibble: 6 x 2
##   Invoice_Day_Week   sum
##   <chr>            <dbl>
## 1 Friday           15.8
## 2 Monday           16.3
## 3 Sunday            8.27
## 4 Thursday         21.7
## 5 Tuesday          20.2
## 6 Wednesday        17.8
```

#4(c)

```r
#Percentage of TransactionsValue by month of the year
Invoice_month_sum = Online_Retail %>% group_by(New_Invoice_Month) %>% summarise(sum=sum(TransactionValu
Month_quan_pct = 100*(Invoice_month_sum$sum)/sum(Invoice_month_sum$sum)
Invoice_month_sum$sum = Month_quan_pct
Invoice_month_sum
```

```
## # A tibble: 12 x 2
##    New_Invoice_Month   sum
##                <dbl> <dbl>
##  1                 1  5.74
##  2                 2  5.11
##  3                 3  7.01
##  4                 4  5.06
##  5                 5  7.42
##  6                 6  7.09
##  7                 7  6.99
##  8                 8  7.00
##  9                 9 10.5
## 10                10 11.0
## 11                11 15.0
## 12                12 12.1
```

#4(d)

```r
#Filtering the Australia's transactions based on New_Invoice_date
Australia_trans = Online_Retail %>% filter(Country == "Australia") %>% group_by(New_Invoice_Date) %>% su
#Finding the date which has maximum number of transactions
Max_trans_date = Australia_trans[which.max(Australia_trans$total),]
Max_trans_date
```

```
## # A tibble: 1 x 2
```

```
##   New_Invoice_Date total
##   <date>           <int>
## 1 2011-06-15         139
```

#4(e)

```
#Filtering the transactions for the hours between 7:00 to 20:00
Sum_quan = Online_Retail %>% filter( New_Invoice_Hour >=7) %>%
  group_by(New_Invoice_Hour) %>% summarise(sum_val= sum(Quantity))
#install.packages("zoo")
library(zoo)
```

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

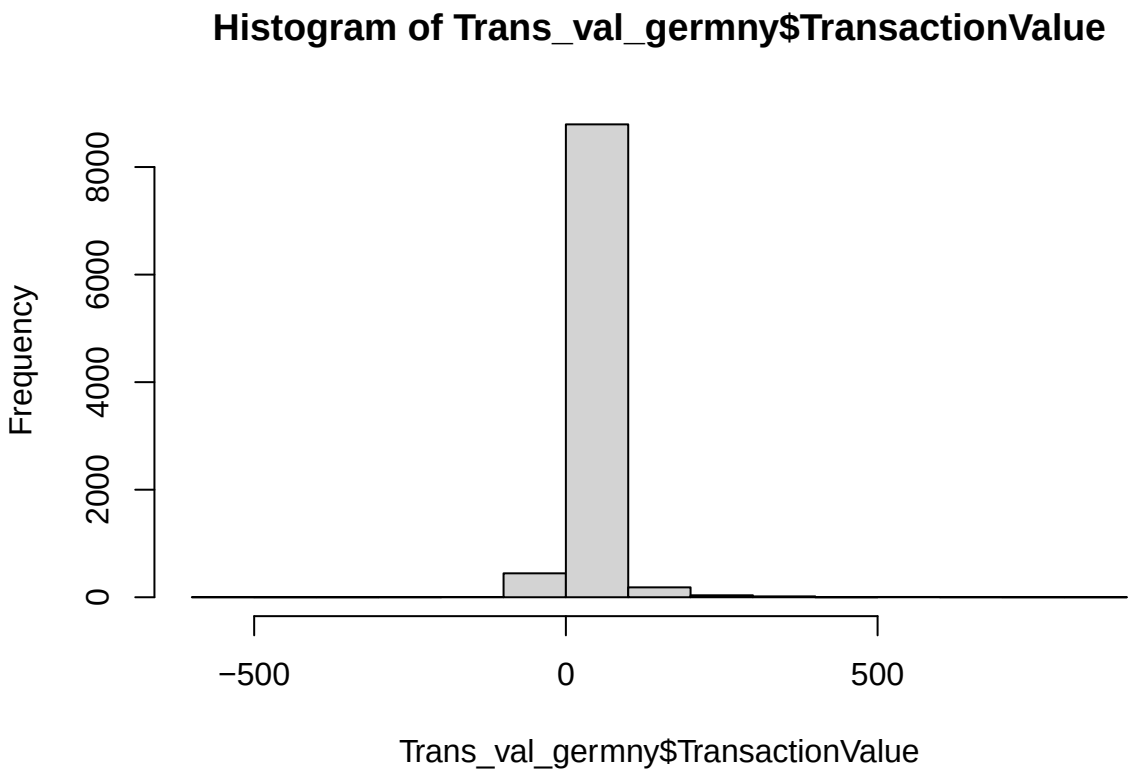
```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
#Adding the two consecutive rows
Consec_sum=rollapply(Sum_quan$sum_val,2,sum)
#Creating the maintainance column
maintainance=c(7:19)
#creating the dataframe for the maintainance and Consec_sum
Main_tab=data.frame(maintainance,Consec_sum)
#checking the minimum value of Consec_sum and the hour where they can start
#maintainance
maintainance_hour=Main_tab[which.min(Main_tab$Consec_sum),]
maintainance_hour
```

```
##    maintainance Consec_sum
## 13           19      40298
```

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

```
Trans_val_germny = filter(Online_Retail, Online_Retail$Country == "Germany")
#Plotting graph between transaction value with the frequency for Germany country
hist(Trans_val_germny$TransactionValue)
```



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

```
#Removing the NA values of CustomerID Column
NA_OnlineRetail=Online_Retail[!is.na(Online_Retail$CustomerID),]
#Number of transactions with respect to CustomerID
Count_transactions = NA_OnlineRetail %>% group_by(CustomerID) %>%
  summarise(count=n())
#printing the row which has max count of transactions
Max_Count_transactions= Count_transactions[which.max(Count_transactions$count),]
# Adding the transaction value with respect to Customer ID
Sum_transactions = NA_OnlineRetail %>% group_by(CustomerID) %>% summarise(Numoftransactions=(sum(Transa
#printing the row which has max sum of transaction value
Max_Sum_transactions= Sum_transactions[which.max
                                       (Sum_transactions$Numoftransactions),]
Max_Count_transactions
```

```
## # A tibble: 1 x 2
##   CustomerID count
##        <int> <int>
## 1      17841  7983
```

```
Max_Sum_transactions
```

```
## # A tibble: 1 x 2
```

```
##   CustomerID Numoftransactions
##        <int>             <dbl>
## 1      14646           279489.
```

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

```
#Percentage of NA's for each column
NA_per = colMeans(is.na(Online_Retail))*100
NA_per
```

```
##         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?

```
#Number of Transactions with missing customer ID
null_Customer = Online_Retail[is.na(Online_Retail$CustomerID),]
# Segregating the missing CustomerID based on countries
table(null_Customer$Country)
```

```
##
##        Bahrain           EIRE         France      Hong Kong         Israel
##              2            711             66            288             47
##       Portugal    Switzerland United Kingdom    Unspecified
##             39            125         133600            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)

```
# Check for missing values
if (any(is.na(OnlineRetail$InvoiceDate))) {
  # Handle missing values (e.g., remove or impute)
  OnlineRetail <- OnlineRetail[!is.na(OnlineRetail$InvoiceDate), ]
}

# Ensure the correct data type and format
OnlineRetail$InvoiceDate <- as.POSIXct(OnlineRetail$InvoiceDate,
                                       format = "%Y-%m-%d %H:%M:%S")

# Calculate the difference in days between consecutive purchases
days_between_purchases <- diff(OnlineRetail$InvoiceDate)

# Calculate the average of the differences in days
average_days_between_purchases <- mean(days_between_purchases, na.rm = TRUE)

# Print the result
cat("Average days between consecutive purchases:",
    average_days_between_purchases, "days\n")
```

```
## Average days between consecutive purchases: NaN 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?

```
# Filtering the dataset for french customers
French_cstmrs = filter(Online_Retail,Country=="France" )
#Returnrate for the french customers
Return_rate = nrow(filter(French_cstmrs,Quantity<1))/nrow(French_cstmrs)
Return_rate
```

```
## [1] 0.01741264
```

#11 What is the product that has generated the highest revenue for the retailer?

```
#revenue of each product
Prd_revenue= Online_Retail %>% group_by(StockCode) %>% summarise(Sum_trnsvalue = sum(TransactionValue))
#Selecting the product with highest revenue
Prd_revenue[which.max(Prd_revenue$Sum_trnsvalue),]
```

```
## # A tibble: 1 x 2
##   StockCode Sum_trnsvalue
##   <chr>             <dbl>
## 1 DOT             206245.
```

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

```
#Number of unique customers
length(unique(Online_Retail$CustomerID))
```

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

#Summary: * A dataset is loaded from “Online_Retail.csv” file in the beginning of the script.

- By grouping the data by ‘Country’, the script calculates the total number of transactions for each country and calculating its percentage of transactions out of the total. It also filters out countries with less than 1% of total transactions.
- The Transaction Value variable is created by multiplying the Quantity and the UnitPrice for each transaction. Each entry in this variable represents the total value of the transaction.
- An analysis of transaction value by country is performed by grouping the data by ‘Country’ and calculating the total transaction value per country. It filters and displays countries with transaction values exceeding $130000.
- An analysis of the invoiced date is performed by converting the variable ‘InvoiceDate’ to a date format for time and date analysis. The script considers various factors, such as weekdays, months, specific dates with high transaction numbers, and maintenance hours.

- An analysis of missing values is performed by the script which calculates the percentages of missing values for each variable.
- This report analyzes transactions with missing 'CustomerID' records by country, and determines the number of transactions with missing records.
- By calculating the average number of days between consecutive shopping visits, the script can provide insight into the frequency with which customers return to the website during their next shopping trip.