ONLINE_RETAIL_ASSIGNMENT_BUSINESS_ANALYTICS

Anurodh Singh

2023-09-25

```
knitr::opts_chunk$set(echo = TRUE)
```

#SUMMARY With the help of packages "dplyr" and "ggplot2:- 1. constructed a code to load a CSV file and calculate the number of transactions by country, showing both the total number and percentage of transactions. It filters for countries with transactions exceeding 1% of the total.

2.Now created a new variable, 'TransactionValue,' by multiplying 'Quantity' and 'UnitPrice,' adding it to the dataframe.

3. Using 'TransactionValue,' now calculated the total transaction values by country, displaying countries with total transactions exceeding £130,000.

4. Hereafter converted 'InvoiceDate' into a 'POSIXlt' object, separated date, day of the week, hour, and month components, and answered several questions related to them, including percentages of transactions by days of the week and months, and the date with the highest transactions from Australia.

5.Then plotted a histogram of transaction values from Germany using the 'hist()' function.

6.Now created a code that identifies the customer with the highest number of transactions and the most valuable customer in terms of the total sum of transactions whic is the customer with customer ID= "17841".

7.Now calculated the percentage of missing values for each variable in the dataset using the 'colMeans()' function.

8.Here constructed a code that determines the number of transactions with missing 'CustomerID' records by countries.

9.Now calculated the average number of days between consecutive shopping sessions for customers using 'CustomerID' and 'New_Invoice_Date.'

10.The code written now calculates the return rate for French customers, defined as the ratio of canceled transactions to the total number of transactions, considering 'Quantity' with a negative value.The return rate for french customers is = 1.741264.

Q11.Here found out the product that has generated the highest revenue for the retailer based on the total sum of 'TransactionValue.' which is= "DOTCOM POSTAGE".

12. Finally with the code here calculated the number of unique customers in the dataset using 'unique()' and 'length()' functions. and the result comes out to be= "4373"

#PROBLEM STATEMENT AND CODE:-

Q1.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.

```
#install.packages("dplyr")
#install.packages("ggplot2")
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(ggplot2)
```

```
# Loading the CSV file into a data frame
xyz <- read.csv("C:/Users/ASUS/Desktop/Online_Retail.csv")
#Q1. Specifying the name of the column I want to count entries in
column_name <- "InvoiceNo"

# Counting the total number of entries in the specified column
num_entries <- length(xyz$InvoiceNo)
num_entries
```

```
## [1] 541909
```

```
#defining a variable transactions_by_country to display country's
transactions_by_country <- table(xyz$Country)
transactions_by_country
```

```
##
##           Australia             Austria             Bahrain
##                1259                 401                  19
##             Belgium              Brazil              Canada
##                2069                  32                 151
##      Channel Islands              Cyprus      Czech Republic
##                 758                 622                  30
##             Denmark                EIRE European Community
##                 389                8196                  61
##             Finland              France             Germany
##                 695                8557                9495
```

```
##                Greece             Hong Kong               Iceland
##                   146                   288                   182
##                Israel                 Italy                 Japan
##                   297                   803                   358
##               Lebanon             Lithuania                 Malta
##                    45                    35                   127
##           Netherlands                Norway                Poland
##                  2371                  1086                   341
##              Portugal                   RSA          Saudi Arabia
##                  1519                    58                    10
##             Singapore                 Spain                Sweden
##                   229                  2533                   462
##           Switzerland United Arab Emirates        United Kingdom
##                  2002                    68                495478
##           Unspecified                   USA
##                   446                   291
```

```
#calculating one percent of total transactions
one_percent <- (0.01*num_entries)
one_percent
```

```
## [1] 5419.09
```

#counting countries with transactions more than one percentage of total transactions

```
more_than_one <- transactions_by_country >= one_percent
more_than_one
```

```
##
##             Australia               Austria               Bahrain
##                 FALSE                 FALSE                 FALSE
##               Belgium                Brazil                Canada
##                 FALSE                 FALSE                 FALSE
##       Channel Islands                Cyprus        Czech Republic
##                 FALSE                 FALSE                 FALSE
##               Denmark                  EIRE    European Community
##                 FALSE                  TRUE                 FALSE
##               Finland                France               Germany
##                 FALSE                  TRUE                  TRUE
##                Greece             Hong Kong               Iceland
##                 FALSE                 FALSE                 FALSE
##                Israel                 Italy                 Japan
##                 FALSE                 FALSE                 FALSE
##               Lebanon             Lithuania                 Malta
##                 FALSE                 FALSE                 FALSE
##           Netherlands                Norway                Poland
##                 FALSE                 FALSE                 FALSE
##              Portugal                   RSA          Saudi Arabia
##                 FALSE                 FALSE                 FALSE
##             Singapore                 Spain                Sweden
##                 FALSE                 FALSE                 FALSE
##           Switzerland United Arab Emirates        United Kingdom
##                 FALSE                 FALSE                  TRUE
```

```
##          Unspecified                 USA
##                FALSE               FALSE
```

```
refined_data_frame <- transactions_by_country[more_than_one]
refined_data_frame
```

```
##
##           EIRE         France        Germany United Kingdom
##           8196           8557           9495         495478
```

#calculating percentage for countries with more than one percent of total countries

```
percent_country_transactions <- refined_data_frame/num_entries*100
percent_country_transactions
```

```
##
##           EIRE         France        Germany United Kingdom
##       1.512431       1.579047       1.752139      91.431956
```

Q2.Create a new variable 'TransactionValue' that is the product of the exising 'Quantity' and 'UnitPrice' variables. Add this variable to the dataframe.

```
TransactionValue <- (xyz$Quantity*xyz$UnitPrice)
B <- data.frame(xyz,TransactionValue)
```

Q3.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

```
Countrywise_total_transactions <- B %>%
  group_by(Country) %>%
  summarize(Countrywise_total_transactions = sum(TransactionValue))

Countrywise_total_transactions
```

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

Q4.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. Click here for more information. First let's convert 'InvoiceDate' into a POSIXlt object: Temp=strptime(Online_Retail$InvoiceDate, format$ ='New_Invoice_Date <- as.Date(Temp) The Date objects have a lot of flexible functions. For example knowing two date values, the object allows you to know the difference between the two dates in terms of the number days. Try this: Online_Retail$New_Invoice_Date[20000] - Online_Retail$New_Invoice_Date[10] Also we can convert dates to days of the week. Let's define a new variable for that Online_Retail$Invoice_Day_Week = weekdays(Online_Retail$New_Invoice_Date) For the Hour, let's just take the hour (ignore the minute) and convert into a normal numerical value: Online_Retail$New_Invoice_Hour = as.numeric(format(Temp, "New_Invoice_Month = as.numeric(format(Temp, "%m")) Now answer the flowing questions. a) Show the percentage of transactions (by numbers) by days of the week (extra 1% of total points) b) Show the percentage of transactions (by transaction volume) by days of the week (extra 1% of total points) c) Show the percentage of transactions (by transaction volume) by month of the year (extra 2% of total points) d) What was the date with the highest number of transactions from Australia? (extra 2% of total points) 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.

```
#Firstly let's Convert 'InvoiceDate' to POSIXlt object
Temp <- strptime(B$InvoiceDate, format='%m/%d/%Y %H:%M', tz='GMT')

#Now Let's separate date, day of the week, hour, and month components
B$New_Invoice_Date <- as.Date(Temp)
B$Invoice_Day_Week <- weekdays(B$New_Invoice_Date)
B$New_Invoice_Hour <- as.numeric(format(Temp, "%H"))
B$New_Invoice_Month <- as.numeric(format(Temp, "%m"))

# a) Show the percentage of transactions (by numbers) by days of the week
day_of_week_counts <- table(B$Invoice_Day_Week)
day_of_week_percent <- (day_of_week_counts / sum(day_of_week_counts)) * 100

# b) Show the percentage of transactions (by transaction volume) by days of the week
day_of_week_volume <- tapply(B$Quantity, B$Invoice_Day_Week, sum)
day_of_week_volume_percent <- (day_of_week_volume / sum(day_of_week_volume)) * 100

# c) Show the percentage of transactions (by transaction volume) by month of the year
month_counts <- table(B$New_Invoice_Month)
month_percent <- (month_counts / sum(month_counts)) * 100

# d) Date with the highest number of transactions from Australia
australia_dates <- B$New_Invoice_Date[B$Country == "Australia"]
highest_transactions_date <- as.Date(names(sort(table(australia_dates), decreasing = TRUE)[1]))
highest_transactions_date
```

```
## [1] "2011-06-15"
```

```
# e) Calculate the optimal hour to start maintenance with minimal impact on customers

# Defining a function to calculate the distribution of transactions by hour
calculate_distribution <- function(hour) {
  # Filtering data for the specified hour
```

```
  hour_data <- subset(B, New_Invoice_Hour == hour)

  # Calculating the number of transactions for that hour
  transactions_count <- nrow(hour_data)

  return(transactions_count)
}

# starting Loop through the hours from 7 to 20 and find the optimal start time
optimal_start_hour <- 7:20
distribution <- sapply(optimal_start_hour, calculate_distribution)
optimal_start_hour <- optimal_start_hour[which.min(distribution)]

# Printing the results
cat("a) Percentage of transactions (by numbers) by days of the week:\n")
```

```
## a) Percentage of transactions (by numbers) by days of the week:
```

```
print(day_of_week_percent)
```

```
##
##    Friday    Monday    Sunday  Thursday   Tuesday Wednesday
##  13.35200  18.58359  12.35151  18.86778  19.46399  17.38113
```

```
cat("b) Percentage of transactions (by transaction volume) by days of the week:\n")
```

```
## b) Percentage of transactions (by transaction volume) by days of the week:
```

```
print(day_of_week_volume_percent)
```

```
##    Friday    Monday    Sunday  Thursday   Tuesday Wednesday
## 14.525431 16.676909  9.418968 21.071105 19.593294 18.714294
```

```
cat("c) Percentage of transactions (by transaction volume) by month of the year:\n")
```

```
## c) Percentage of transactions (by transaction volume) by month of the year:
```

```
print(month_percent)
```

```
##
##         1         2         3         4         5         6         7         8
##  7.197928  5.538760  7.824567  5.684739  6.653180  7.100502  7.800291  6.547661
##         9        10        11        12
## 10.727626 11.967632 17.859524  5.097589
```

```
cat("d) Date with the highest number of transactions from Australia:\n")
```

```
## d) Date with the highest number of transactions from Australia:
```

```r
print(highest_transactions_date)
```

```
## [1] "2011-06-15"
```

```r
cat("e) Optimal start hour for maintenance to minimize customer impact:\n")
```

```
## e) Optimal start hour for maintenance to minimize customer impact:
```

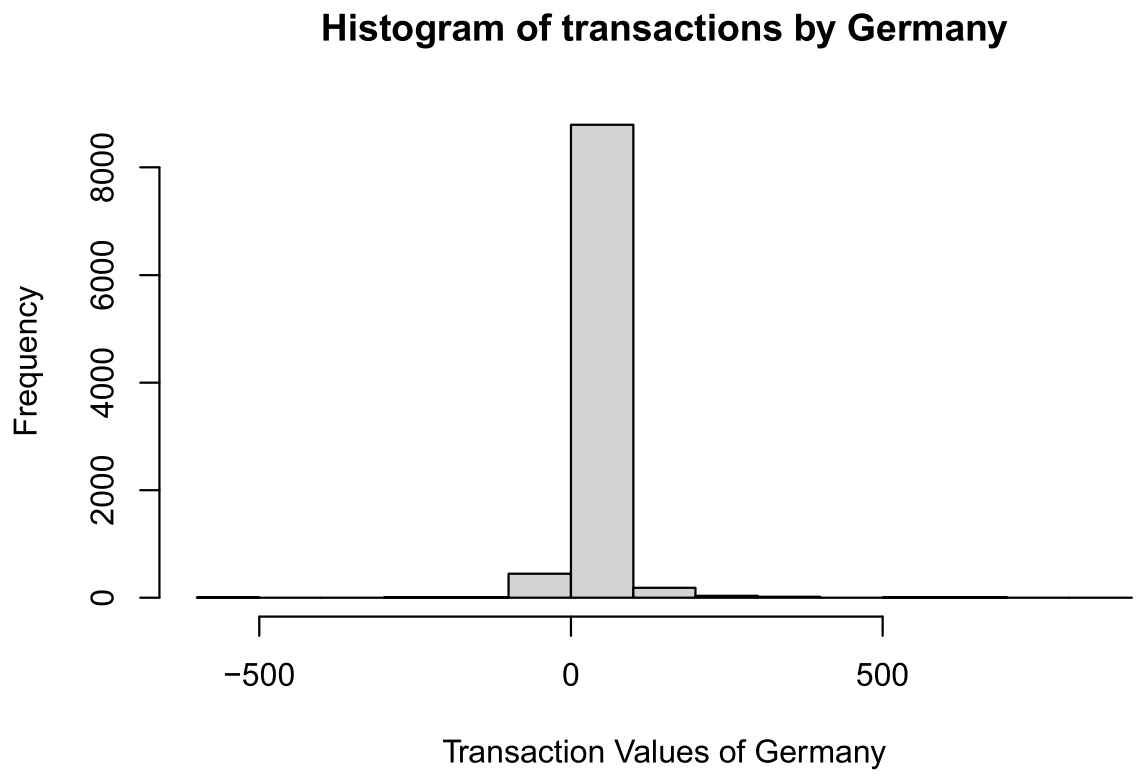
```r
print(optimal_start_hour)
```

```
## [1] 7
```

Q5.Plot the histogram of transaction values from Germany. Use the hist() function to plot.

```r
transactions_of_germany <- B$TransactionValue[xyz$Country=="Germany"]
hist(transactions_of_germany,
     main = "Histogram of transactions by Germany",

     xlab = "Transaction Values of Germany",
     ylab = "Frequency")
```



Q6Which customer had the highest number of transactions? Which customer is most valuable (i.e. highest total sum of transactions)

```r
sst <- c("CustomerID")
tbl_cst_id <- xyz[sst]
MX <- names(sort(table(tbl_cst_id$CustomerID), decreasing = TRUE)[1])
MX
```

```
## [1] "17841"
```

```r
ctv <- B[, c("CustomerID", "TransactionValue")]
Valuable_cust<- ctv %>%
  group_by(CustomerID) %>%
  summarize(TotalTransactionValue = sum(TransactionValue))
Valuable_cust
```

```
## # A tibble: 4,373 x 2
##    CustomerID TotalTransactionValue
##         <int>                 <dbl>
##  1      12346                    0
##  2      12347                 4310
##  3      12348                 1797.
##  4      12349                 1758.
##  5      12350                  334.
##  6      12352                 1545.
##  7      12353                   89
##  8      12354                 1079.
##  9      12355                  459.
## 10      12356                 2811.
## # i 4,363 more rows
```

Q7.Calculate the percentage of missing values for each variable in the dataset. Hint colMeans()

```r
#assuming B is the name of my dataframe
#calculating the percentage of missing values for each variable
mis_per <- colMeans(is.na(B))*100
print("Percentage of missing values for each variable")
```

```
## [1] "Percentage of missing values for each variable"
```

```r
print(mis_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          42.98858          42.98858          42.98858
## New_Invoice_Month
##          42.98858
```

Q8.What are the number of transactions with missing CustomerID records by countries?

```r
missing_ct <- B[is.na(B$CustomerID),]
mis_ctcr <- table(missing_ct$Country)
print(mis_ctcr)
```

```
##
##        Bahrain           EIRE         France      Hong Kong         Israel
##              2            711             66            288             47
##       Portugal    Switzerland United Kingdom    Unspecified
##             39            125         133600            202
```

Q9.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) Hint: 1. A close approximation is also acceptable and you may find diff() function useful

```r
# Convert InvoiceDate to POSIXlt object for date calculations
B$InvoiceDate <- as.POSIXlt(B$InvoiceDate, format="%m/%d/%Y %H:%M", tz="GMT")

# Sort the dataframe by CustomerID and InvoiceDate
sorted_data <- B[order(B$CustomerID,B$InvoiceDate),]

# Calculate the time difference between consecutive transactions for each customer
time_diff <- unlist(tapply(sorted_data$InvoiceDate, sorted_data$CustomerID, function(x) c(0, diff(x))))

# Filter out 0 time differences (transactions on the same day)
time_diff <- time_diff[time_diff != 0]

# Calculate the average number of days between consecutive shopping sessions
average_days_between_shopping <- mean(time_diff, na.rm = TRUE)


# Print the result
print(paste("Average number of days between consecutive shopping sessions:", round(average_days_between
```

```
## [1] "Average number of days between consecutive shopping sessions: 3450692.52"
```

Q10.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? Consider the cancelled transactions as those where the 'Quantity' variable has a negative value.

```r
# Filter the data to only include transactions from French customers
french_customers <- B[B$Country == "France",]

# Count the number of cancelled transactions
cancelled_transactions <- french_customers[french_customers$Quantity < 0, ] %>%
nrow()

# Count the total number of transactions
total_transactions <- french_customers %>% nrow()

# Calculate the return rate
```

```r
return_rate <- (cancelled_transactions/total_transactions*100)

# Print the result
print(return_rate)
```

```
## [1] 1.741264
```

Q11.What is the product that has generated the highest revenue for the retailer? (i.e. item with the highest total sum of 'TransactionValue')

```r
# Calculating total transaction values for each product
total_transaction_values <- tapply(B$TransactionValue, B$Description, sum)

# Finding the product with the highest total sum of 'TransactionValue'
highest_revenue_product <- names(total_transaction_values[total_transaction_values == max(total_transac

# Printing the result
print(paste("Product with the highest revenue:", highest_revenue_product))
```

```
## [1] "Product with the highest revenue: DOTCOM POSTAGE"
```

Q12.How many unique customers are represented in the dataset? You can use unique() and length()functions

```r
# Getting unique CustomerIDs
unique_customers <- unique(B$CustomerID)

# Calculating the number of unique customers
number_of_unique_customers <- length(unique_customers)

# Printing the result
print(paste("Number of unique customers:", number_of_unique_customers))
```

```
## [1] "Number of unique customers: 4373"
```