BA - Assignment 2 – Online Retail Analytics

Gaurav Kudeshia

2023-10-15

Assignment Overview: This assignment focuses on analyzing the “Online_Retail.csv” dataset, involving various operations and visualizations.

Summary

• The total number of transactions for each country was computed, including canceled transactions. The analysis revealed that the United Kingdom had the highest number of transactions, totaling 495,478. A table was created, including only countries contributing more than 1% of the total transactions, which consisted of four countries.

• A new column named “TransactionValue” was added to the existing dataset “V,” representing the product of ‘Quantity’ and ‘UnitPrice.’ The total transaction values for each country were calculated, and countries with transaction values exceeding 130,000 British Pounds were filtered and displayed. Notably, the United Kingdom led with a transaction value of 8,187,806.

• The ‘InvoiceDate’ column was converted into a POSIXct object, allowing the analysis of transactions by days of the week. The percentage of transactions on different days was calculated: Friday (15.17%), Monday (17.55%), Sunday (11.88%), Thursday (19.17%), Tuesday (18.79%), and Wednesday (17.45%).

• The customer with CustomerID “17841” was identified as having the highest number of transactions. In contrast, CustomerID “14646” was the most valuable customer, contributing the highest total transaction value of 279,489.

• A histogram representing the distribution of transaction values from Germany was plotted, providing insights into the spending patterns of German customers.

• The dataset revealed that only the “CustomerID” variable had missing values. Additionally, Bahrain had the fewest missing transactions with an absent CustomerID, while the United Kingdom had the most.

• The analysis calculated the average number of days between consecutive shopping sessions for French customers, indicating a return rate of approximately 2,840,169.96 days and a rate of 1.74%.

• The product generating the highest revenue for the retailer was identified as “DOTCOM POSTAGE.” Additionally, the dataset contained 4,373 unique customers. This comprehensive analysis delved into customer behavior, transaction trends, and product performance, utilizing diverse techniques and visualizations to gain a comprehensive understanding of the retail dataset.

Problem

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. (10% of total points)
2. Create a new variable 'TransactionValue' that is the product of the exising 'Quantity' and 'UnitPrice' variables. Add this variable to the dataframe. (10% of total points)
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. (15% of total points)
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. Click here for more information. First let's convert 'InvoiceDate' into a POSIXlt object:

   Temp=strptime(Online_Retail$InvoiceDate,format='%m/%d/%Y %H:%M',tz='GMT') Check the variable using, head(Temp). Now, let's separate date, day of the week and hour components dataframe with names as New.I.Date, I.Day.Week and New.I.Hour:

   Online_Retail$New.I.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.I.Date [20000] - Online_Retail$New.I.Date[10] Also we can convert dates to days of the week. Let's define a new variable for that Online_Retail $ I.Day.Week = weekdays (Online_Retail$New.I.Date) For the Hour, let's just take the hour (ignore the minute) and convert into a normal numerical value:

   Online_Retail$New.I.Hour = as.numeric(format(Temp, "%H")) Finally, lets define the month as a separate numeric variable too:

   Online_Retail$New.I.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. (extra 4% of total points)

5. Plot the histogram of transaction values from Germany. Use the hist() function to plot. (5% of total points)
6. Which customer had the highest number of transactions? Which customer is most valuable (i.e. highest total sum of transactions)? (15% of total points)
7. Calculate the percentage of missing values for each variable in the dataset (5% of total points). Hint colMeans():
8. What are the number of transactions with missing CustomerID records by countries? (10 % of total points)
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) (5% of total points!) Hint: 1. A close approximation is also acceptable and you may find diff() function useful.
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? (10% of total points). Consider the cancelled transactions as those where the 'Quantity' variable has a negative value.
11. What is the product that has generated the highest revenue for the retailer? (i.e. item with the highest total sum of 'TransactionValue'). (10% of total points)
12. How many unique customers are represented in the dataset? You can use unique() and length() functions. (10% of total points)

Loaded datasets from the necessary libraries

```
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
```

Installed the dataset from "Online_Retail.csv"

```
V <- read.csv("/Users/gauravkudeshia/Desktop/Rhistory Business Analytics/Online_Re
tail.csv")
```

1. Show the breakdown of the number of transactions by countries i.e., how many transactions are in the data set 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.

```
#Determined the number of transactions conducted by each country
T_b_C <- table(V$Country)
T_b_C
```

```
##
##             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
```

```
#Added total number of transaction
Tot_Numberof <- sum(T_b_C)
Tot_Numberof
```

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

```
#calculated 1% of the total number of transactions
A1per <- (0.01*Tot_Numberof)

#"True" will be for those whose value are more then 1% of total number (A) of tran
sactions
Grater_then_one <- T_b_C >= A1per

#Calculated the countries which are more than 1% of total number (A) of transactio
ns
Filtered_T_b_C <- T_b_C[Grater_then_one]
Filtered_T_b_C
```

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

```
#Calculated each country percent wise out of total number of transaction
Percentage.more.than.one <- Filtered_T_b_C/Tot_Numberof*100
Percentage.more.than.one
```

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

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

Added "TransactionValue" which is the product of Quantity & Unit Price to the existing data frame

```
TransactionValue=(V$Quantity*V$UnitPrice)
V=data.frame(V,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.

```
#Analyzed the distribution of transaction values across different countries
Transaction.Values.by.Countries=V%>%
  group_by(Country)%>%
  summarize(total.t_by_each_country = sum(TransactionValue))


Transaction.Values.by.Countries
```

```
## # A tibble: 38 × 2
##    Country         total.t_by_each_country
##    <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
```

```
#Filtered countries with total transaction exceeding 130,000 British Pound
Filtered.Transaction.V.B.C= Transaction.Values.by.Countries[Transaction.Values.by.
Countries$total.t_by_each_country>130000, c("Country","total.t_by_each_country")]

Filtered.Transaction.V.B.C
```

```
## # A tibble: 6 × 2
##   Country        total.t_by_each_country
##   <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. Click here for more information. First let's convert 'InvoiceDate' into a POSIXlt object:

Temp=strptime(Online_Retail$InvoiceDate,format='%m/%d/%Y %H:%M',tz='GMT') Check the variable using, head(Temp). Now, let's separate date, day of the week and hour components dataframe with names as New.I.Date, I.Day.Week and New.I.Hour:

Online_Retail$New.I.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.I.Date [20000] - Online_Retail$New.I.Date[10] Also we can convert dates to days of the week. Let's define a new variable for that Online_Retail $ I.Day.Week = weekdays (Online_Retail$New.I.Date) For the Hour, let's just take the hour (ignore the minute) and convert into a

normal numerical value:

Online_Retail$New.I.Hour = as.numeric(format(Temp, “%H”)) Finally, lets define the month as a separate numeric variable too:

Online_Retail$New.I.Month = as.numeric(format(Temp, “%m”))

```
#Directly converted 'InvoiceDate' into a POSIXct object and extract its components
V$New.I.Date <- as.POSIXct(V$InvoiceDate, format='%m/%d/%Y %H:%M', tz='GMT')
V$I.Day.Week <- weekdays(V$New.I.Date)
V$New.I.Hour <- as.numeric(format(V$New.I.Date, "%H"))
V$New.I.Month <- as.numeric(format(V$New.I.Date, "%m"))
```

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

```
day.week.counts <- table(V$I.Day.Week)
day.week.P <- (day.week.counts / nrow(V)) * 100
day.week.P
```

```
##
##    Friday    Monday    Sunday  Thursday   Tuesday Wednesday
##  15.16731  17.55110  11.87930  19.16503  18.78692  17.45035
```

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

```
day_of_week_transaction_values <- tapply(V$TransactionValue, V$I.Day.Week, sum)
day_of_week_transaction_percentages <- (day_of_week_transaction_values / sum(V$Tra
nsactionValue)) * 100
day_of_week_transaction_percentages
```

```
##    Friday    Monday    Sunday  Thursday   Tuesday Wednesday
## 15.804787 16.297194  8.265282 21.671867 20.170636 17.790232
```

c) Show the percentage of transactions (by transaction volume) by month of the year

```
month_transaction_values <- tapply(V$TransactionValue, V$New.I.Month, sum)
month_transaction_percentages <- (month_transaction_values / sum(V$TransactionValu
e)) * 100
month_transaction_percentages
```

```
##         1         2         3         4         5         6         7         8
##  5.744919  5.109515  7.009487  5.059703  7.420519  7.090080  6.989308  7.003469
##         9        10        11        12
## 10.460751 10.984123 14.995836 12.132290
```

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

```
australia_dates <- V$New.I.Date[V$Country == "Australia"]
highest_transactions_date <- as.Date(names(sort(table(australia_dates), decreasing
= TRUE)[1]))
print(paste("date with the highest number of transactions from Australia:",highest
_transactions_date))
```

```
## [1] "date with the highest number of transactions from Australia: 2011-06-15"
```

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.

```
hourly_transaction_counts <- table(V$New.I.Hour)
optimal_maintenance_hour <- names(which.min(hourly_transaction_counts))
print(paste("Hour of the day to start:",optimal_maintenance_hour))
```

```
## [1] "Hour of the day to start: 6"
```

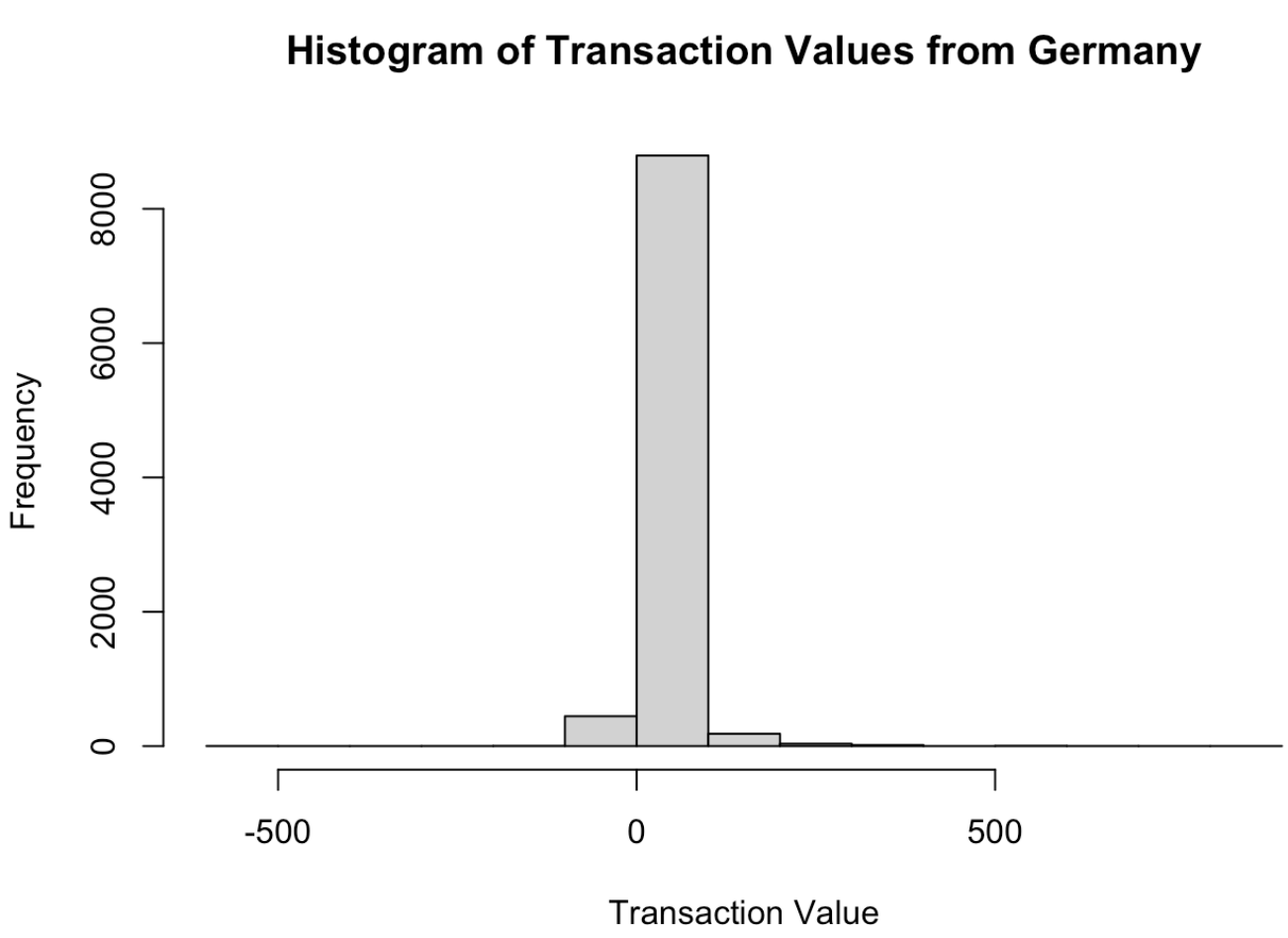
5. Plot the histogram of transaction values from Germany. Use the hist() function to plot.

Loaded required libraries

```
library(ggplot2)
```

```
# Filterd transactions from Germany
G.T = V[V$Country == "Germany", "TransactionValue"]

# Ploted histogram
hist(G.T, main = "Histogram of Transaction Values from Germany",
     xlab = "Transaction Value", ylab = "Frequency")
```



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

Identified the customer with the highest number of transactions

```
highest.trans.no <- names(sort(table(V$CustomerID), decreasing = TRUE)[1])
print(paste("Customer with highest number of transactions:",highest.trans.no))
```

```
## [1] "Customer with highest number of transactions: 17841"
```

Identified the customer who is most valued

```
#Computed the overall transaction value for each customer
Cust.Tran.V <- aggregate(TransactionValue ~ CustomerID, data = V, sum)

#Identified the customer with the highest total transaction amount
most.valuable.cust <- Cust.Tran.V[which.max(Cust.Tran.V$TransactionValue), ]

#Displayed the outcome
most.valuable.cust
```

```
##      CustomerID TransactionValue
## 1704      14646           279489
```

7. Calculate the percentage of missing values for each variable in the dataset (5% of total points). Hint colMeans():

```
#Computed the proportion of missing values for each variable as a percentage
Per_M <- colMeans(is.na(V))*100

#Displayed the percentage of missing values for each variable
Per_M
```

```
##        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.I.Date       I.Day.Week       New.I.Hour
##          0.00000          0.00000          0.00000          0.00000
##      New.I.Month
##          0.00000
```

8. What are the number of transactions with missing CustomerID records by countries? (10 % of total points)

```
#Filtered the dataframe to include only rows where the 'CustomerID' value is missi
ng
Numb.Trans.miss = V[is.na(V$CustomerID),]

#Tallied the count of missing CustomerID transactions based on countries.
Miss.Cust.Trans = table(Numb.Trans.miss$Country)

#Displayed the outcomes
Miss.Cust.Trans
```

```
##
##        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) (5% of total points!) Hint: 1. A close approximation is also acceptable and you may find diff() function useful.

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

# Sorted the dataframe by CustomerID and InvoiceDate
sorted.D <- V[order(V$CustomerID,V$InvoiceDate),]

# Calculated the time difference between consecutive transactions for each custome
r
diff.Time <- unlist(tapply(sorted.D$InvoiceDate, sorted.D$CustomerID, function(x)
c(0, diff(x))))

# Filtered out 0 time differences (transactions on the same day)
diff.Time <- diff.Time[diff.Time != 0]

# Calculated the average number of days between consecutive shopping sessions
average.days <- mean(diff.Time, na.rm = TRUE)


# Printed the result
print(paste("Average number of days between consecutive shopping :", round(average
.days, 2)))
```

```
## [1] "Average number of days between consecutive shopping : 2840169.96"
```

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? (10% of total points). Consider the cancelled transactions as those where the 'Quantity' variable has a negative value.

```
# Filtered the data to only include transactions from French customers
french.cust <- V[V$Country == "France",]

# Counted the number of cancelled transactions
cancelled.t <- french.cust[french.cust$Quantity < 0, ] %>%
nrow()

# Counted the total number of transactions
total.t <- french.cust %>% nrow()

# Calculated the return rate
r.rate <- (cancelled.t/total.t*100)

# Printed the result
print(paste("Return Rate:",r.rate))
```

```
## [1] "Return Rate: 1.7412644618441"
```

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

```
# Calculated total transaction values for each product
each.product.trans.values <- tapply(V$TransactionValue, V$Description, sum)

# Found the product with the highest total sum of 'TransactionValue'
which.product.high.revenue <- names(each.product.trans.values[which.max(each.produ
ct.trans.values)])

print(paste("product that has generated the highest revenue for the retailer:",whi
ch.product.high.revenue))
```

```
## [1] "product that has generated the highest revenue for the retailer: DOTCOM PO
STAGE"
```

12. How many unique customers are represented in the dataset? You can use unique() and length() functions. (10% of total points)

```
# Counted the unique customers in the dataset
uni.customers <- length(unique(V$CustomerID))
print(paste("Unique Customers:",uni.customers))
```

```
## [1] "Unique Customers: 4373"
```