BA ASSIGNMENT 2

Keerthi Priya Nallamekala

2023-10-15

#Question 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.

```
online_retail <- read.csv("Online_Retail.csv")
transaction_counts <- table(online_retail$Country)
Total_Transactions <- sum(transaction_counts)
percentages <- (transaction_counts / Total_Transactions) * 100
countries_over_1_percent <- names(percentages[percentages > 1])
filtered_counts <- transaction_counts[countries_over_1_percent]
print(filtered_counts)
```

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

```
print(percentages[countries_over_1_percent])
```

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

#Question 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
```

```
online_retail <- online_retail %>% mutate(TransactionValue=Quantity*UnitPrice)
head(online_retail)
```

```
##   InvoiceNo StockCode                         Description Quantity
## 1    536365    85123A  WHITE HANGING HEART T-LIGHT HOLDER        6
## 2    536365     71053                 WHITE METAL LANTERN        6
## 3    536365    84406B      CREAM CUPID HEARTS COAT HANGER        8
## 4    536365    84029G KNITTED UNION FLAG HOT WATER BOTTLE        6
## 5    536365    84029E      RED WOOLLY HOTTIE WHITE HEART.        6
## 6    536365     22752        SET 7 BABUSHKA NESTING BOXES        2
##      InvoiceDate UnitPrice CustomerID        Country TransactionValue
## 1 12/1/2010 8:26      2.55      17850 United Kingdom            15.30
## 2 12/1/2010 8:26      3.39      17850 United Kingdom            20.34
## 3 12/1/2010 8:26      2.75      17850 United Kingdom            22.00
## 4 12/1/2010 8:26      3.39      17850 United Kingdom            20.34
## 5 12/1/2010 8:26      3.39      17850 United Kingdom            20.34
## 6 12/1/2010 8:26      7.65      17850 United Kingdom            15.30
```

#Question 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.

```
transactionvalues_by_country <- tapply(online_retail$TransactionValue, online_retail$Country, sum)
filtered_countries <- transactionvalues_by_country[transactionvalues_by_country > 130000]
print(filtered_countries)
```

```
##      Australia           EIRE         France        Germany    Netherlands
##       137077.3       263276.8       197403.9       221698.2       284661.5
## United Kingdom
##      8187806.4
```

#Question 4

```
Online_Retail <- read.csv("Online_Retail.csv")
Temp <- strptime(Online_Retail$InvoiceDate, format = '%m/%d/%Y %H:%M', tz = 'GMT')
Online_Retail$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"))
```

#Question 4(a)-Show the percentage of transactions (by numbers) by days of the week

```
day_counts <- table(Online_Retail$Invoice_Day_Week)
day_percentage <- (day_counts / sum(day_counts)) * 100
print(day_percentage)
```

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

#Question 4(b)-Show the percentage of transactions (by transaction volume) by days of the week

```
day_volume <- tapply(Online_Retail$Quantity, Online_Retail$Invoice_Day_Week, sum)
day_volume_percentage <- (day_volume / sum(day_volume)) * 100
print(day_volume_percentage)
```

```
##    Friday    Monday    Sunday  Thursday   Tuesday Wednesday
## 15.347197 15.751219  9.035768 22.560307 18.575336 18.730172
```

#Question 4(c)-Show the percentage of transactions (by transaction volume) by month of the year

```
month_volume <- tapply(Online_Retail$Quantity, Online_Retail$New_Invoice_Month, sum)
month_volume_percentage <- (month_volume / sum(month_volume)) * 100
print(month_volume_percentage)
```

```
##         1         2         3         4         5         6         7         8
##  5.968685  5.370263  6.797554  5.584870  7.348492  6.599561  7.555680  7.847057
##         9        10        11        12
## 10.621507 11.021685 14.301036 10.983608
```

#Question 4(d)-Date with the highest number of transactions from Australia

```
max_transactions_date <- subset(Online_Retail, Country == "Australia")$
New_Invoice_Date[which.max(table(subset(Online_Retail, Country == "Australia")$New_Invoice_Date))]
print(max_transactions_date)
```

```
## [1] "2010-12-17"
```

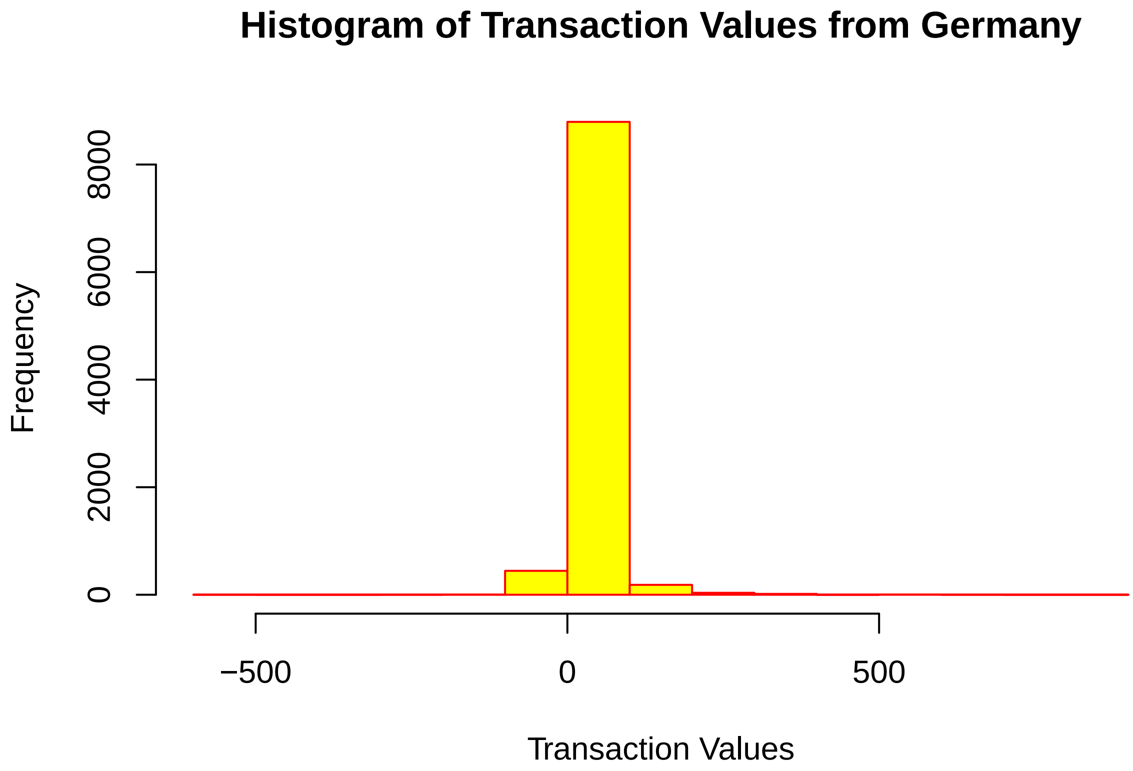
#Question 4(e)-Find the hour of the day to minimize customer impact during maintenance

```
hourly_counts <- table(Online_Retail$New_Invoice_Hour)
hours_available <- 7:20
customer_impact <- sapply(hours_available, function(hour) sum(hourly_counts[hour:(hour + 1)]))
optimal_hour <- hours_available[which.min(customer_impact)]
print(optimal_hour)
```

```
## [1] 14
```

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

```
germany_transactions <- online_retail[online_retail$Country == "Germany", ]
hist(germany_transactions$TransactionValue,
main = "Histogram of Transaction Values from Germany",
xlab = "Transaction Values",
ylab = "Frequency",
col = "yellow",
border = "red")
```



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

```
transactions_per_customer <- table(online_retail$CustomerID)
customer_with_most_transactions <- names(transactions_per_customer[transactions_per_customer ==
max(transactions_per_customer)])
total_transaction_values <- tapply(online_retail$TransactionValue, online_retail$CustomerID, sum)
most_valuable_customer <- names(total_transaction_values[total_transaction_values ==
max(total_transaction_values)])
print(paste("Customer with the highest number of transactions:", customer_with_most_transactions))
```

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

```
print(paste("Most valuable customer (highest total sum of transactions):", most_valuable_customer))
```

```
## [1] "Most valuable customer (highest total sum of transactions): 14646"
```

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

```
missing_percentage <- colMeans(is.na(online_retail)) * 100
print("Percentage of missing values for each variable:")
```

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

```r
print(missing_percentage)
```

```
##        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
##          0.00000
```

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

```r
missing_customer_transactions <- online_retail[is.na(online_retail$CustomerID), ]
missing_customer_transactions_by_country <- table(missing_customer_transactions$Country)
print("Number of transactions with missing CustomerID records by countries:")
```

```
## [1] "Number of transactions with missing CustomerID records by countries:"
```

```r
print(missing_customer_transactions_by_country)
```

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

#Question 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)

```r
online_retail$InvoiceDate <- as.POSIXlt(online_retail$InvoiceDate, format="%m/%d/%Y %H:%M", tz="GMT")
sorted_data <- online_retail[order(online_retail$CustomerID, online_retail$InvoiceDate), ]
time_diff <- unlist(tapply(sorted_data$InvoiceDate, sorted_data$CustomerID, function(x) c(0, diff(x))))
time_diff <- time_diff[time_diff != 0]
average_days_between_shopping <- mean(time_diff, na.rm = TRUE)
print(paste("Average number of days between consecutive shopping sessions:",
round(average_days_between_shopping, 2)))
```

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

#Question 10-n 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?

```r
total_transactions <- nrow(online_retail)
cancelled_transactions_france <- online_retail[online_retail$Country ==
"France" & grepl("^C", online_retail$InvoiceNo), ]
cancelled_transactions_count <- nrow(cancelled_transactions_france)
return_rate_france <- cancelled_transactions_count / total_transactions * 100
print(paste("Return rate for French customers:", round(return_rate_france, 2), "%"))
```

```
## [1] "Return rate for French customers: 0 %"
```

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

```r
Total_Transaction_values <- tapply(online_retail$TransactionValue, online_retail$Description, sum)
highest_revenue_product <- names(Total_Transaction_values[Total_Transaction_values ==
max(Total_Transaction_values)])
print(paste("Product with the highest revenue:", highest_revenue_product))
```

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

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

```r
Unique_Customers <- unique(online_retail$CustomerID)
number_of_unique_customers <- length(Unique_Customers)
print(paste("Number of unique customers:", number_of_unique_customers))
```

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