Assignment\_2

Rupesh Suragani

#Installing the 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

Download the dataset, and use the read.csv() command to load the file into a R dataframe

Online\_Retail <- read.csv("C:/Users/rupes/OneDrive/Desktop/Kent State University/BA/Assignment-2/Online\_Retail.csv")  
  
dim(Online\_Retail)

## [1] 541909 8

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.

# Country wise transactions and its percentage which is more than 1%.  
# Here the dataframe has to be grouped by country and to be summarized and filtered to more than 1%  
  
Country\_wise\_Transactions <- Online\_Retail %>%  
 group\_by(Country) %>%  
 summarise(Transactions = n(), percentage = (n()/nrow(Online\_Retail))\*100) %>%  
 filter(percentage > 1.0)  
  
cat("Number of Transactions by Countries and their percentages (greater than 1%)", "\n")

## Number of Transactions by Countries and their percentages (greater than 1%)

Country\_wise\_Transactions

## # A tibble: 4 × 3  
## Country Transactions percentage  
## <chr> <int> <dbl>  
## 1 EIRE 8196 1.51  
## 2 France 8557 1.58  
## 3 Germany 9495 1.75  
## 4 United Kingdom 495478 91.4

1. Create a new variable ‘TransactionValue’ that is the product of the existing ‘Quantity’ and ‘UnitPrice’ variables. Add this variable to the dataframe.

# Creating the new variable "TransactionValue" to the dataframe (using Mutate function)  
  
Online\_Retail <- Online\_Retail %>%  
 mutate(TransactionValue = Quantity\*UnitPrice)  
  
dim(Online\_Retail) #Total dimensions

## [1] 541909 9

colnames(Online\_Retail)

## [1] "InvoiceNo" "StockCode" "Description" "Quantity"   
## [5] "InvoiceDate" "UnitPrice" "CustomerID" "Country"   
## [9] "TransactionValue"

1. Using the newly created variable, TransactionValue, show the breakdown of transaction values by countries i.e. how much money in total has been spent by each country. Show this in total sum of transaction values. Show only countries with total transaction exceeding 130,000 British Pound.

# For showing only the countries with Total Transactions exceeding 130,000 pounds the data needs to be grouped country wise totaling each of their transaction values and filtering the values greater than 130,000.  
  
Total\_transaction\_values <- Online\_Retail %>% group\_by(Country)%>%  
 summarise(Total\_TransValue = sum(TransactionValue)) %>%  
 filter(Total\_TransValue > 130000)  
  
cat("Country wise total transaction values exceeding 130,000 pounds ", "\n")

## Country wise total transaction values exceeding 130,000 pounds

Total\_transaction\_values

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

1. 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:

Check the variable using, head(Temp).

# Converting the 'InvoiceDate' to a POSIXlt object  
Temp <- strptime(Online\_Retail$InvoiceDate,format='%m/%d/%Y %H:%M',tz='GMT')  
  
#This code uses the strptime function to convert the 'InvoiceDate' column in the 'Online\_Retail' data frame to a POSIXlt object. It specifies the format of the date and time in your data, which is 'month/day/year hour:minute', and sets the time zone to '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"

#Extracting the data component in a new column called New\_Invoice\_Date  
Online\_Retail$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.  
  
# Calculating the difference between two dates  
Online\_Retail$New\_Invoice\_Date[20000] - Online\_Retail$New\_Invoice\_Date[10]

## Time difference of 8 days

#This line calculates the difference in days between the 20000th and 10th entries in the 'New\_Invoice\_Date' column.  
  
#For the Hour, let’s just take the hour (ignore the minute) and convert into a normal numerical value:  
  
# Extracting the hour component(Using as.numeric)  
Online\_Retail$New\_Invoice\_Hour <- as.numeric(format(Temp, "%H"))  
  
# Extracting the hour component(Using weekdays)  
Online\_Retail$New\_Invoice\_Week <- weekdays(Online\_Retail$New\_Invoice\_Date)  
  
# Extracting the Month component(Using as.numeric)  
Online\_Retail$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  
# For this, the data needs to be grouped by New\_Invoice\_week column.  
Online\_Retail %>%  
 group\_by(New\_Invoice\_Week) %>%  
 summarise(Trans\_percentage = (n()/nrow(Online\_Retail))\*100)

## # A tibble: 6 × 2  
## New\_Invoice\_Week Trans\_percentage  
## <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

#b) Show the percentage of transactions (by transaction volume) by days of the week  
Online\_Retail %>%  
 group\_by(New\_Invoice\_Week) %>%  
 summarise(percent\_by\_transactions\_volume = (sum(TransactionValue)/sum(Online\_Retail$TransactionValue))\*100)

## # A tibble: 6 × 2  
## New\_Invoice\_Week percent\_by\_transactions\_volume  
## <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

#c) Show the percentage of transactions (by transaction volume) by month of the year  
Online\_Retail %>%  
 group\_by(New\_Invoice\_Month) %>%  
 summarise(percent\_by\_transactions\_volume = (sum(TransactionValue)/sum(Online\_Retail$TransactionValue))\*100)

## # A tibble: 12 × 2  
## New\_Invoice\_Month percent\_by\_transactions\_volume  
## <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

#d) What was the date with the highest number of transactions from Australia?  
subset(Online\_Retail, Country == "Australia") %>%  
 group\_by(New\_Invoice\_Date) %>%  
 summarise(Transactions = n()) %>%  
 arrange(desc(Transactions)) %>%  
 head(1)

## # A tibble: 1 × 2  
## New\_Invoice\_Date Transactions  
## <date> <int>  
## 1 2011-06-15 139

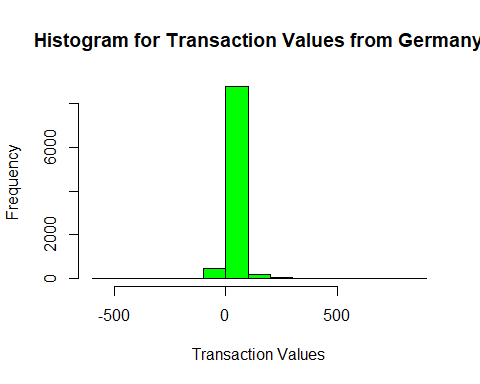
#e) The company needs to shut down the website for two consecutive hours for maintenance. What would be the hour of the day to start this so that the distribution is at minimum for the customers? The responsible IT team is available from 7:00 to 20:00 every day.  
Online\_Retail %>%  
 group\_by(New\_Invoice\_Hour) %>%  
 summarise(percent\_of\_transactions = (n()/nrow(Online\_Retail))\*100) %>%  
 arrange(percent\_of\_transactions)

## # A tibble: 15 × 2  
## New\_Invoice\_Hour percent\_of\_transactions  
## <dbl> <dbl>  
## 1 6 0.00757  
## 2 7 0.0707   
## 3 20 0.161   
## 4 19 0.684   
## 5 18 1.47   
## 6 8 1.64   
## 7 17 5.26   
## 8 9 6.34   
## 9 10 9.05   
## 10 16 10.1   
## 11 11 10.6   
## 12 14 12.5   
## 13 13 13.3   
## 14 15 14.3   
## 15 12 14.5

# Therefore, from running the above code, the best possible hour would be at 7.00 every day as the distribution (percent\_of\_transactions) at that time is minimum.

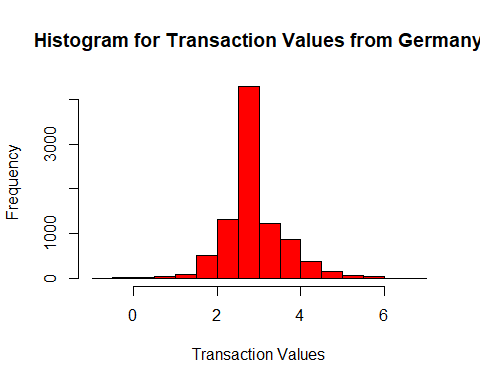
1. Plot the histogram of transaction values from Germany. Use the hist() function to plot.

# For plotting only the transaction values from Germany, the germany Transaction values data needs to be taken out using the subset function to the dataframe.  
  
#Filtering the data for germany  
Germany\_TransValues <- subset(Online\_Retail, Country == "Germany")   
  
# Plotting the data without scaling  
hist(Germany\_TransValues$TransactionValue,   
 main = "Histogram for Transaction Values from Germany",   
 xlab = "Transaction Values",   
 ylab = "Frequency",   
 col = "green")



#plotting the data with scaling using log  
hist(log(Germany\_TransValues$TransactionValue),   
 main = "Histogram for Transaction Values from Germany",   
 xlab = "Transaction Values",   
 ylab = "Frequency",   
 col = "red")

## Warning in log(Germany\_TransValues$TransactionValue): NaNs produced



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

# Grouping the data by customerID to find the highest number of transactions  
customers <- Online\_Retail %>%   
 na.omit %>% group\_by(CustomerID) %>%  
 summarise(Transactions = n()) %>%  
 arrange(desc(Transactions))  
  
  
# Grouping the data by customerID and adding there transaction values  
Highest\_customers <- Online\_Retail %>%   
 na.omit %>% group\_by(CustomerID) %>%  
 summarise(total = sum(TransactionValue))%>%  
 arrange(desc(total))  
  
# print the data  
cat("The customer with the highest number of transactions was:", as.numeric(customers[1,1]), "\n")

## The customer with the highest number of transactions was: 17841

cat("The most valuble customer is:", as.numeric(Highest\_customers[1,1]), "with the total transaction value of", as.numeric(Highest\_customers[1,2]))

## The most valuble customer is: 14646 with the total transaction value of 279489

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

# get the count of missing values using is.na and Calculating the percentage of missing values for CustomerId as it only has missing values  
missingValues <- colMeans(is.na(Online\_Retail))  
  
# Calculating the percentage of missing values for CustomerId as it only has missing values  
Missing\_percent <- missingValues["CustomerID"]\*100  
  
# print the data  
cat("The Online\_Retail dataframe consists of only CustomerID with missing values. The percentage of data missing is:", Missing\_percent)

## The Online\_Retail dataframe consists of only CustomerID with missing values. The percentage of data missing is: 24.92669

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

Online\_Retail %>%  
 filter(is.na(CustomerID)) %>% # Filter transactions with missing CustomerID records  
 group\_by(Country) %>%  
 summarise(Missing\_CustomerID = n())

## # A tibble: 9 × 2  
## Country Missing\_CustomerID  
## <chr> <int>  
## 1 Bahrain 2  
## 2 EIRE 711  
## 3 France 66  
## 4 Hong Kong 288  
## 5 Israel 47  
## 6 Portugal 39  
## 7 Switzerland 125  
## 8 United Kingdom 133600  
## 9 Unspecified 202

1. 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)

# Sorting the data frame by CustomerID and New\_Invoice\_Date  
Online\_Retail <- Online\_Retail[order(Online\_Retail$CustomerID, Online\_Retail$New\_Invoice\_Date), ]  
  
# Calculating the average time difference in days  
return\_time <- mean(diff(Online\_Retail$New\_Invoice\_Date), na.rm = TRUE)  
  
# Display the average return time  
cat("The average number of days that the customer comeback to their next shopping is ", as.numeric(return\_time), "days")

## The average number of days that the customer comeback to their next shopping is 0.0005997328 days

1. 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?

FrenchOrders <- subset(Online\_Retail, Country == "France")   
FrenchOrders\_cancelled <- subset(Online\_Retail, Country == "France" & Quantity <= 0)  
France\_ReturnRate <- (nrow(FrenchOrders\_cancelled)/nrow(FrenchOrders))\*100  
  
cat("The return rate for the french Customers is",France\_ReturnRate)

## The return rate for the french Customers is 1.741264

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

# Here, for generating the highest revenue by product the data needs to be grouped using stock code.  
  
Highest\_revenue\_product <- Online\_Retail %>%  
 group\_by(Description) %>%  
 summarise(TotalRevenue = sum(TransactionValue)) %>%  
 arrange(desc(TotalRevenue))  
  
cat("The product which has generated the highest revenue is ", as.character(Highest\_revenue\_product[1,1]), "and the total revenue is ", as.numeric(Highest\_revenue\_product[1,2]) )

## The product which has generated the highest revenue is DOTCOM POSTAGE and the total revenue is 206245.5

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

Unique\_customer <- length(unique(Online\_Retail$CustomerID))  
  
cat("The total no.of unique customers in the dataset is",Unique\_customer )

## The total no.of unique customers in the dataset is 4373