Descriptive Analytics

Melissa Paniagua

10/15/2020

Assignment 1 | Module 4

The purpose of this assignment is to develop descriptive statistics analysis utilizing data from an online retail company.

The data contains the following attributes:

- InvoiceNo: Invoice number has a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.
- StockCode: Product code. It is a 5-digit integral number uniquely assigned to each distinct product.
- Description: Product name.
- Quantity: The quantities of each product per transaction.
- InvoiceDate: It shows the day and time when each transaction was generated.
- UnitPrice: Product price per unit in sterling.
- CustomerID: It is a 5-digit integral number uniquely assigned to each customer.
- Country: The name of the country where each customer resides.

```
# Load the libraries needed for this assignment
library(readr)
library(dplyr)
library(tidyverse)

#Load the dataset
mydf <- read.csv("Online_Retail.csv")

#See the first 6 columns of the dataset
head(mydf)
```

```
  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
```

```
1 12/1/2010 8:26      2.55      17850 United Kingdom
2 12/1/2010 8:26      3.39      17850 United Kingdom
3 12/1/2010 8:26      2.75      17850 United Kingdom
4 12/1/2010 8:26      3.39      17850 United Kingdom
5 12/1/2010 8:26      3.39      17850 United Kingdom
6 12/1/2010 8:26      7.65      17850 United Kingdom
```

Data Exploration

```r
#To get descriptive statistics
summary(mydf)
```

```
   InvoiceNo        StockCode                               Description
 573585 :  1114   85123A :  2313   WHITE HANGING HEART T-LIGHT HOLDER:  2369
 581219 :   749   22423  :  2203   REGENCY CAKESTAND 3 TIER          :  2200
 581492 :   731   85099B :  2159   JUMBO BAG RED RETROSPOT           :  2159
 580729 :   721   47566  :  1727   PARTY BUNTING                     :  1727
 558475 :   705   20725  :  1639   LUNCH BAG RED RETROSPOT           :  1638
 579777 :   687   84879  :  1502   ASSORTED COLOUR BIRD ORNAMENT     :  1501
 (Other):537202   (Other):530366   (Other)                           :530315
    Quantity                   InvoiceDate       UnitPrice
 Min.   :-80995.00   10/31/2011 14:41:  1114   Min.   :-11062.06
 1st Qu.:     1.00   12/8/2011 9:28  :   749   1st Qu.:     1.25
 Median :     3.00   12/9/2011 10:03 :   731   Median :     2.08
 Mean   :     9.55   12/5/2011 17:24 :   721   Mean   :     4.61
 3rd Qu.:    10.00   6/29/2011 15:58 :   705   3rd Qu.:     4.13
 Max.   : 80995.00   11/30/2011 15:13:   687   Max.   : 38970.00
                     (Other)         :537202
   CustomerID               Country
 Min.   :12346    United Kingdom:495478
 1st Qu.:13953    Germany       :  9495
 Median :15152    France        :  8557
 Mean   :15288    EIRE          :  8196
 3rd Qu.:16791    Spain         :  2533
 Max.   :18287    Netherlands   :  2371
 NA's   :135080   (Other)       : 15279
```

Here me can see that CustomerID variable has 135,080 missing values.

```r
# To see the number of data points
nrow(mydf)
```

```
[1] 541909
```

```r
# Number of transaccions for each country
head(table(mydf$Country))
```

```
Australia    Austria    Bahrain    Belgium     Brazil     Canada
     1259        401         19       2069         32        151
```

Questions:

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.

```
#Total Number of transaccions for each country showing 1% of the total transactions
mypct <- mydf %>% group_by(Country) %>%
         summarise(Total_Trans = n(), Total_Perc = sum(n()/length(mydf$Country)*100)) %>%
         filter(Total_Perc > 1)

# See the dataframe
as.data.frame(mypct)
```

```
         Country Total_Trans Total_Perc
1           EIRE        8196   1.512431
2         France        8557   1.579047
3        Germany        9495   1.752139
4 United Kingdom      495478  91.431956
```

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

```
# Include the new variable
mydf <- mydf %>% mutate(TransactionValue = Quantity * UnitPrice)

# See the first 6 rows and last 6 columns of the dataframe
mydf[1:6, 4:9]
```

```
  Quantity    InvoiceDate UnitPrice CustomerID        Country TransactionValue
1        6 12/1/2010 8:26      2.55      17850 United Kingdom            15.30
2        6 12/1/2010 8:26      3.39      17850 United Kingdom            20.34
3        8 12/1/2010 8:26      2.75      17850 United Kingdom            22.00
4        6 12/1/2010 8:26      3.39      17850 United Kingdom            20.34
5        6 12/1/2010 8:26      3.39      17850 United Kingdom            20.34
6        2 12/1/2010 8:26      7.65      17850 United Kingdom            15.30
```

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.

```
# Show the countries and total transactions greater than 130,000 British Pound
greater_trans <- mydf %>% select(Country, TransactionValue) %>%
                          group_by(Country) %>%
                          summarise(Transactions = sum(TransactionValue)) %>%
                          filter(Transactions > 130000)

# See the values
as.data.frame(greater_trans)
```

```
          Country Transactions
1       Australia     137077.3
2            EIRE     263276.8
3          France     197403.9
4         Germany     221698.2
5     Netherlands     284661.5
6  United Kingdom    8187806.4
```

4. Convert 'InvoiceDate' from categorical into date variable:

```r
#First let's convert 'InvoiceDate' into a POSIXlt object
temp <- strptime(mydf$InvoiceDate,format='%m/%d/%Y %H:%M',tz='GMT')

# See the dataframe
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"
```

```r
#Now, separate date, day of the week, and hour components dataframe
mydf$New_Invoice_Date <- as.Date(temp)

# Know the difference between the two dates in terms of the number days.
mydf$New_Invoice_Date[20000] - mydf$New_Invoice_Date[10]
```

```
Time difference of 8 days
```

```r
# Define a new variable with the day name
mydf$Invoice_Day_Week = weekdays(mydf$New_Invoice_Date)

# Convert the hour into a normal numerical value
mydf$New_Invoice_Hour = as.numeric(format(temp, "%H"))

#Finally, define the month as a separate numeric variable.
mydf$New_Invoice_Month = as.numeric(format(temp, "%m"))

# To see the dataframe with the new columns
mydf[1:6, 10:13]
```

```
  New_Invoice_Date Invoice_Day_Week New_Invoice_Hour New_Invoice_Month
1       2010-12-01        Wednesday                8                12
2       2010-12-01        Wednesday                8                12
3       2010-12-01        Wednesday                8                12
4       2010-12-01        Wednesday                8                12
5       2010-12-01        Wednesday                8                12
6       2010-12-01        Wednesday                8                12
```

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

```r
# To get the total number of days transactions and its percentage
perc_day <- mydf %>% group_by(Invoice_Day_Week) %>%
        summarise(Num_Trans = n(), Percent = sum(n()/length(mydf$Invoice_Day_Week)*100))

# Show the dataframe
as.data.frame(perc_day)
```

```
  Invoice_Day_Week Num_Trans  Percent
1           Friday     82193 15.16731
2           Monday     95111 17.55110
3           Sunday     64375 11.87930
4         Thursday    103857 19.16503
5          Tuesday    101808 18.78692
6        Wednesday     94565 17.45035
```

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

```r
# To get the total volume of transactions and its percentage per week days
total_perc_day <- mydf %>% group_by(Invoice_Day_Week) %>%
                           summarise(Total_Trans = sum(TransactionValue)) %>%
                           mutate(Percent = Total_Trans/sum(Total_Trans)*100)

# Show the dataframe
as.data.frame(total_perc_day)
```

```
  Invoice_Day_Week Total_Trans   Percent
1           Friday   1540610.8 15.804787
2           Monday   1588609.4 16.297194
3           Sunday    805678.9  8.265282
4         Thursday   2112519.0 21.671867
5          Tuesday   1966182.8 20.170636
6        Wednesday   1734147.0 17.790232
```

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

```r
# To get the total volume of transactions by month
perc_month <- mydf %>% group_by(New_Invoice_Month) %>%
                       summarise(Total_Trans = sum(TransactionValue)) %>%
                       mutate(Percent = Total_Trans/sum(Total_Trans)*100)

# Show the dataframe
as.data.frame(perc_month)
```

```
   New_Invoice_Month Total_Trans   Percent
1                  1    560000.3  5.744919
2                  2    498062.6  5.109515
3                  3    683267.1  7.009487
4                  4    493207.1  5.059703
5                  5    723333.5  7.420519
```

```
6                   6    691123.1  7.090080
7                   7    681300.1  6.989308
8                   8    682680.5  7.003469
9                   9   1019687.6 10.460751
10                 10   1070704.7 10.984123
11                 11   1461756.2 14.995836
12                 12   1182625.0 12.132290
```

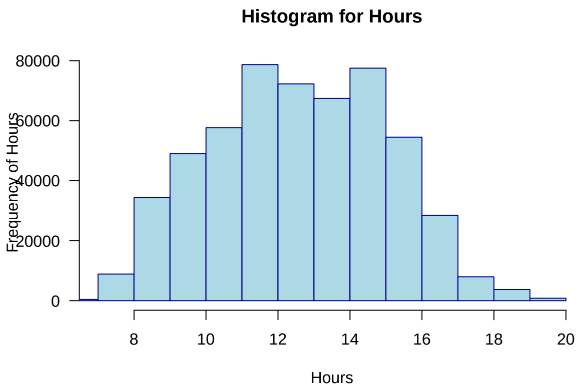
- **d) What was the date with the highest number of transactions from Australia?**

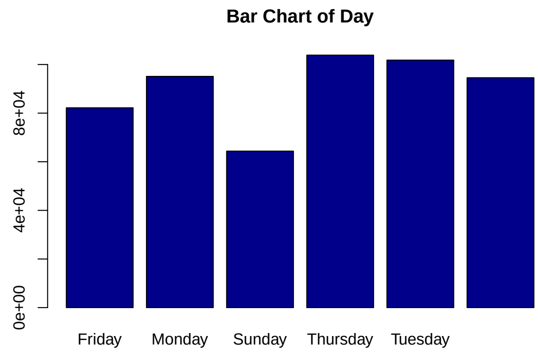
```r
# To select the highest number of transactions per country
highest_num1 <- mydf %>%
  filter(mydf$Country == "Australia") %>%
  group_by(New_Invoice_Date) %>%
  summarise(Num_TransactionValue = n()) %>%
  top_n(1, Num_TransactionValue)

# Show the dataframe
as.data.frame(highest_num1)
```

```
  New_Invoice_Date Num_TransactionValue
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.**

```r
# Histogram plot to show the less busy time of the day to develop the maintenance.
hist(mydf$New_Invoice_Hour,
     main="Histogram for Hours",
     xlab="Hours",
     border="darkblue",
     col="lightblue",
     xlim=c(7,20),
     ylab="Frequency of Hours",
     las=1,
     breaks=12)

# Bar plot to identify the best days to implement the maintenance
barplot(table(mydf$Invoice_Day_Week),
        main= "Bar Chart of Day",
        col= c("darkblue"))
```

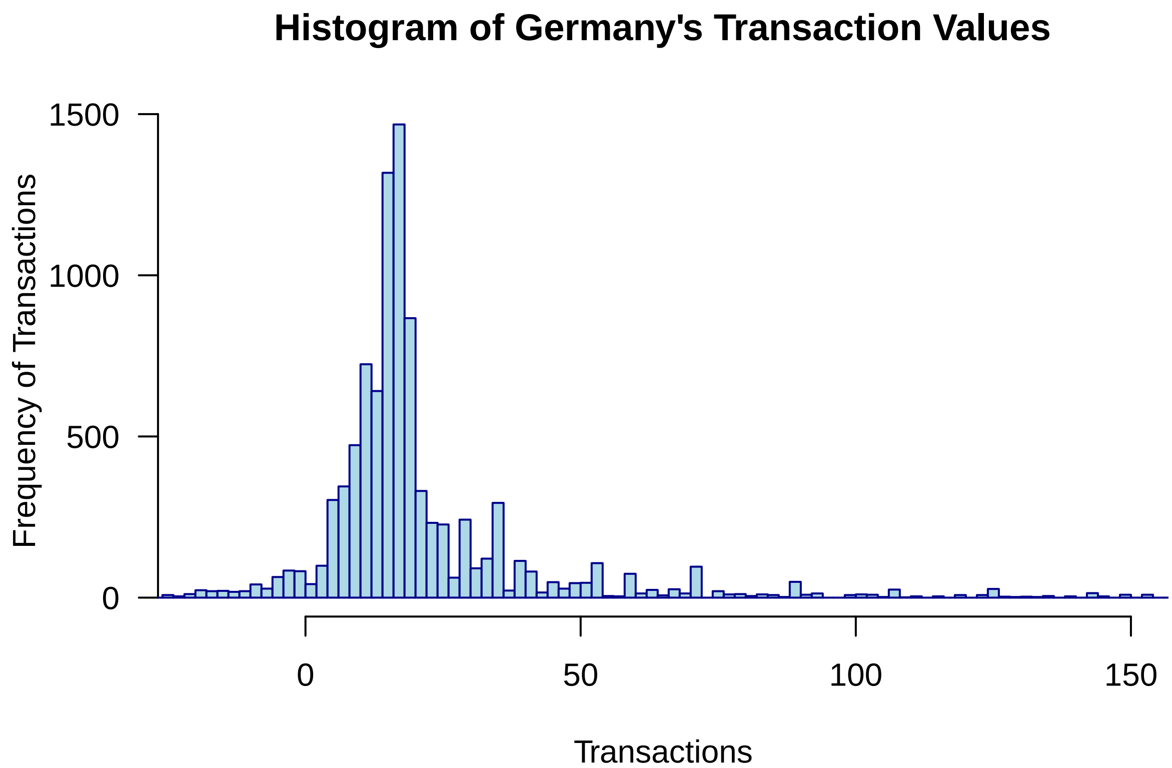




Here we can see on this plot, the best hours to do the maintanance of the company's website are between 18:00 and 20:00 pm. Additionally, Sundays night would be a good day to do the maintenance of the website.

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

```
# Transaction values from Germany
germany <- select(mydf, TransactionValue,Country) %>%
            filter(mydf$Country == "Germany")

# Histogram
hist(germany$TransactionValue,
     main="Histogram of Germany's Transaction Values",
     xlab="Transactions",
     border="darkblue",
     col="lightblue",
     xlim=c(-20, 150),
     ylab="Frequency of Transactions",
     las=1,
     breaks=600)
```



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

Part a)

```
# Most Valuable Customer with highest number of transactions
valuable_customer <- mydf %>% na.omit() %>%
                             group_by(CustomerID) %>%
                             summarise(Highest_Num = n()) %>%
                             top_n(1, Highest_Num)


# Show the dataframe
as.data.frame(valuable_customer)
```

```
  CustomerID Highest_Num
1      17841        7983
```

Part b)

```r
# Most Valuable Customer with highest volume of transactions
valuable_customer <- mydf %>% na.omit() %>%
                             group_by(CustomerID) %>%
                             summarise(Highest_Transaction = sum(TransactionValue)) %>%
                             top_n(1, Highest_Transaction)

# Show the dataframe
as.data.frame(valuable_customer)
```

```
  CustomerID Highest_Transaction
1      14646              279489
```

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

```r
# Percentage of missing values
perc_missing_val <- colMeans(is.na(mydf))

# Show the dataframe
as.data.frame(perc_missing_val)
```

```
                  perc_missing_val
InvoiceNo                0.0000000
StockCode                0.0000000
Description              0.0000000
Quantity                 0.0000000
InvoiceDate              0.0000000
UnitPrice                0.0000000
CustomerID               0.2492669
Country                  0.0000000
TransactionValue         0.0000000
New_Invoice_Date         0.0000000
Invoice_Day_Week         0.0000000
New_Invoice_Hour         0.0000000
New_Invoice_Month        0.0000000
```

The output shows that Description_name and CustomerID_name have missing values, with 0.27% and 24.93% of missing values respectively.

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

```r
# Total number of transactions with missing CustomerID by country
missing_ID <- mydf %>% group_by(Country, CustomerID) %>%
              filter(is.na(CustomerID)) %>%
              summarise(Num_Transactions = n())

# Show the dataframe
as.data.frame(missing_ID)
```

```
         Country CustomerID Num_Transactions
```

```
1        Bahrain         NA                2
2           EIRE         NA              711
3         France         NA               66
4      Hong Kong         NA              288
5         Israel         NA               47
6       Portugal         NA               39
7    Switzerland         NA              125
8 United Kingdom         NA           133600
9    Unspecified         NA              202
```

```r
# To make sure we are counting all the missing CustomerID values
sum(missing_ID$Num_Transactions)
```

```
[1] 135080
```

9. On average, how often the customers comeback to the website for their next shopping? (i.e. what is the average number of days between consecutive shopping) (Optional/Golden question: 18 additional marks!) Hint: 1. A close approximation is also acceptable and you may find diff() function useful.

```r
# Days average between consecutive shopping
often <- mydf %>% select(CustomerID, New_Invoice_Date) %>%
                  group_by(CustomerID) %>%
                  mutate(Diff_Days = as.numeric(c(diff(New_Invoice_Date),0))) %>%
                  summarise(Time_Days = sum(Diff_Days),
                            Days_Avg = sum(Diff_Days)/sum(n()))

# Show the dataframe
head(as.data.frame(often))
```

```
  CustomerID Time_Days Days_Avg
1      12346         0 0.000000
2      12347       365 2.005495
3      12348       283 9.129032
4      12349         0 0.000000
5      12350         0 0.000000
6      12352       260 2.736842
```

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

```r
# Ratio of devolutions from French customers
numenador <- mydf %>% select(Quantity, TransactionValue, Country) %>%
                   filter(Country == "France" & Quantity < 0)

denominador <- mydf %>% select(Quantity, TransactionValue, Country) %>%
                        filter(Country == "France")

my_ratio <- count(numenador) / count(denominador)

# Show the dataframe
as.data.frame(my_ratio)
```

```
           n
1 0.01741264
```

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

```
# Highest revenue for the retailer
highest_revenue <- mydf %>% group_by(Description) %>%
                           summarise(highest_transaction = sum(TransactionValue))  %>%
                           top_n(1)
```

```
Selecting by highest_transaction
```

```
# Show the dataframe
as.data.frame(highest_revenue)
```

```
     Description highest_transaction
1 DOTCOM POSTAGE            206245.5
```

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

```
# Show the number of unique customers
length(unique(mydf$CustomerID))
```

```
[1] 4373
```