BA_Assignment_2

Balamanoj Reddy Kommareddy

2023-10-13

##Installing and loading the dplyr package

#install.packages("dplyr")

```
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
```

#loading the dataset

#Setwd("G:\64036_BA_Assignment_2_Retail.csv")

```
Store_data <- read.csv("Online_Retail.csv")
head(Store_data)
```

```
##   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
```

#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.

```
store_data.df <- as.data.frame(table(Store_data$Country))
head(store_data.df)
```

```
##        Var1 Freq
## 1 Australia 1259
## 2   Austria  401
## 3   Bahrain   19
## 4   Belgium 2069
## 5    Brazil   32
## 6    Canada  151
```

#1.(1)Show only countries accounting for more than 1% of the total transactions.

```
store_data.df$Percentage <- store_data.df$Freq/nrow(Store_data)*100
colnames(store_data.df) <- c("Country", "Count", "Percentage")
store_data.df[store_data.df$Percentage>1,]
```

```
##           Country  Count Percentage
## 11           EIRE   8196   1.512431
## 14         France   8557   1.579047
## 15        Germany   9495   1.752139
## 36 United Kingdom 495478  91.431956
```

```
##EIRE, France, Germany and United Kingdom are the Countries accounting for more than 1% of the total transactions
```

#2.Creating a new variable ‘TransactionValue’ that is the product of the exising ‘Quantity’ and ‘UnitPrice’ variables and adding this variable to the dataframe.

```
Store_data$TransactionValue <- Store_data$Quantity * Store_data$UnitPrice
colnames(Store_data)
```

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

```
Transaction_data <- Store_data %>% group_by(Country) %>% summarise(Total= sum(TransactionValue))
Transaction_data
```

```
## # A tibble: 38 × 2
##    Country           Total
##    <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
```

```
#United Kingdom, Netherlands, EIRE, Germany, France & Australia are the countries where the transaction value exceeds 130,00
0 British Pound
```

#3(2).Show only countries with total transaction exceeding 130,000 British Pound.

```
Transaction_data %>% filter(Total>=130000) %>% arrange(desc(Total))
```

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

#4.Converting Invoice Date into a POSIXlt object.

```
Temp_data=strptime(Store_data$InvoiceDate,format='%m/%d/%Y %H:%M',tz='GMT')
head(Temp_data)
```

```
## [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"
```

```
Store_data$New_Invoice_Date <- as.Date(Temp_data)
Store_data$New_Invoice_Date[20000]-Store_data$New_Invoice_Date[10]
```

```
## Time difference of 8 days
```

```
Store_data$Invoice_Day_Week= weekdays(Store_data$New_Invoice_Date)

Store_data$New_Invoice_Hour= as.numeric(format(Temp_data, "%H"))

Store_data$New_Invoice_Month= as.numeric(format(Temp_data, "%m"))
```

#4(a).Percentage of transactions (by numbers) by days of the week.

```
Percentage_by_days <- Store_data %>% group_by(Invoice_Day_Week) %>% summarise(count=n()) %>% mutate(Percentage=count/nrow(Store_data)*100)
Percentage_by_days
```

```
## # A tibble: 6 × 3
##   Invoice_Day_Week  count Percentage
##   <chr>             <int>      <dbl>
## 1 Friday            82193       15.2
## 2 Monday            95111       17.6
## 3 Sunday            64375       11.9
## 4 Thursday         103857       19.2
## 5 Tuesday          101808       18.8
## 6 Wednesday         94565       17.5
```

#4(b).Percentage of transactions (by transaction volume) by days of the week.

```
Percentage_by_week <- Store_data %>% group_by(Invoice_Day_Week) %>% summarise(Total=sum(TransactionValue)) %>% mutate(Percentage=Total/sum(Total)*100)
Percentage_by_week
```

```
## # A tibble: 6 × 3
##   Invoice_Day_Week    Total Percentage
##   <chr>               <dbl>      <dbl>
## 1 Friday           1540611.      15.8
## 2 Monday           1588609.      16.3
## 3 Sunday            805679.       8.27
## 4 Thursday         2112519       21.7
## 5 Tuesday          1966183.      20.2
## 6 Wednesday        1734147.      17.8
```

#4(c).Percentage of transactions (by transaction volume) by month of the year

```
Percentage_by_month <- Store_data %>% group_by(New_Invoice_Month) %>% summarise(Total = sum(TransactionValue)) %>% mutate(Percentage = Total/sum(Total) * 100)
Percentage_by_month
```

```
## # A tibble: 12 × 3
##    New_Invoice_Month    Total Percentage
##                <dbl>    <dbl>      <dbl>
##  1                 1  560000.       5.74
##  2                 2  498063.       5.11
##  3                 3  683267.       7.01
##  4                 4  493207.       5.06
##  5                 5  723334.       7.42
##  6                 6  691123.       7.09
##  7                 7  681300.       6.99
##  8                 8  682681.       7.00
##  9                 9 1019688.      10.5
## 10                10 1070705.      11.0
## 11                11 1461756.      15.0
## 12                12 1182625.      12.1
```

#4(d).The date with the highest number of transactions from Australia.

```
Store_data %>% filter(Country=="Australia") %>% group_by(New_Invoice_Date) %>% summarise(Total_Count=n()) %>% arrange(desc(T
otal_Count))
```

```
## # A tibble: 49 × 2
##    New_Invoice_Date Total_Count
##    <date>                 <int>
##  1 2011-06-15               139
##  2 2011-07-19               137
##  3 2011-08-18                97
##  4 2011-03-03                84
##  5 2011-10-05                82
##  6 2011-05-17                73
##  7 2011-02-15                69
##  8 2011-01-06                48
##  9 2011-07-14                35
## 10 2011-09-16                34
## # ℹ 39 more rows
```

```
#Australia has recorded the highest number of transactions with 139 Transactions on 2011-06-15.
```

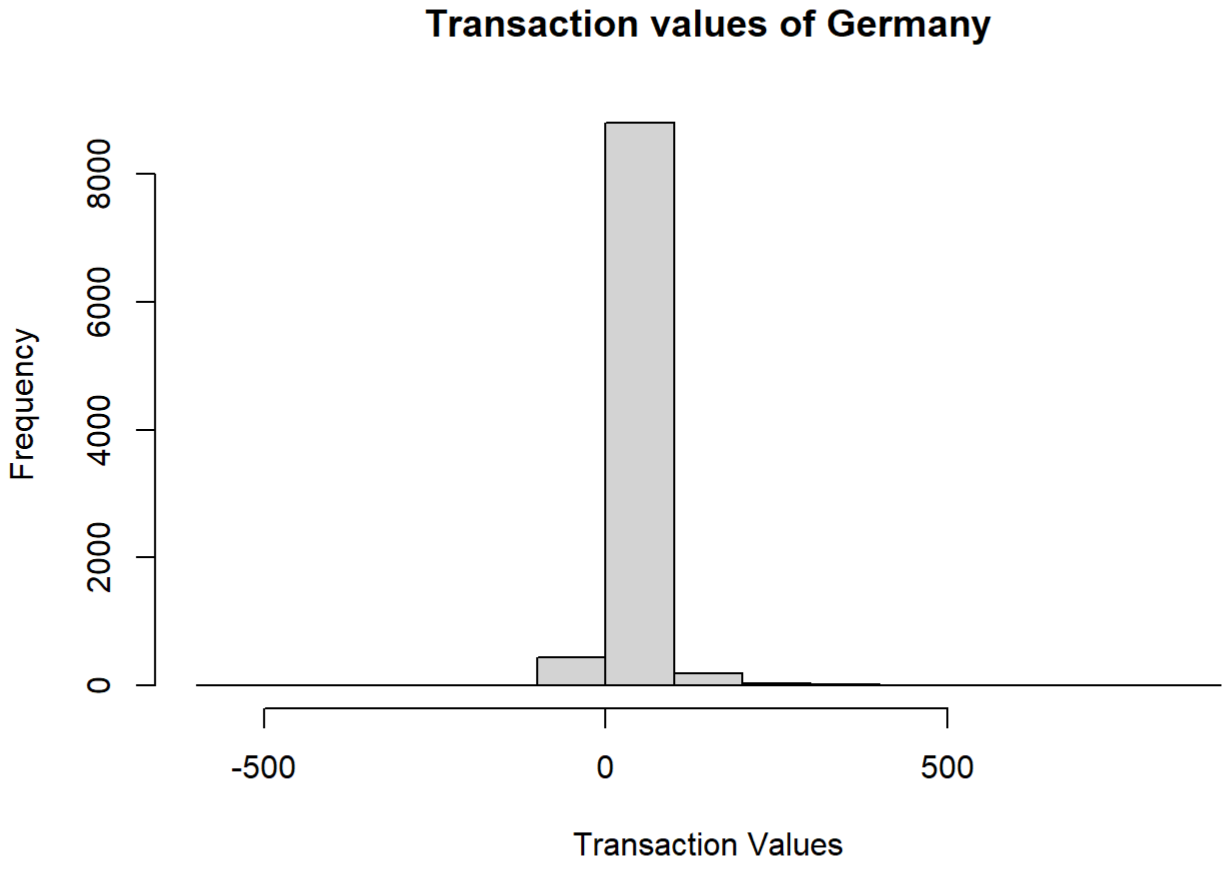
#4(e).The company needs to shut down the website for two consecutive hours for maintenance. What would be the hour of the day to start this so that the distribution is at minimum for the customers? The responsible IT team is available from 7:00 to 20:00 every day.

```
m=distribution <- Store_data %>%group_by(New_Invoice_Hour) %>%summarize(count = n()) %>%arrange(count) %>%filter(New_Invoice
_Hour %in% 7:20)

# Calculate the average number of transactions per hour
hourly_transaction_counts <- table(m$New_Invoice_Hour)

# Find the hour with the lowest average transaction rate
optimal_hour <- which.min(hourly_transaction_counts)

# Convert the hour back to 24-hour format
optimal_hour <- ifelse(optimal_hour == 1, 7, optimal_hour + 6)

# Display the optimal hour
print(paste("Optimal Hour for Maintenance:",optimal_hour))
```

```
## [1] "Optimal Hour for Maintenance: 7"
```

#5.Plot the histogram of transaction values from Germany.

```
Transactions_Germany <- subset(Store_data, Country=="Germany")
hist(Transactions_Germany$TransactionValue, main = "Transaction values of Germany", xlab = "Transaction Values", ylab = "Fre
quency")
```



#6.Which customer had the highest number of transactions?

```
Store_data %>% group_by(CustomerID) %>% filter(!is.na(CustomerID)) %>% summarise(n_count=n()) %>% arrange(desc(n_count))
```

```
## # A tibble: 4,372 × 2
##    CustomerID n_count
##         <int>   <int>
##  1      17841    7983
##  2      14911    5903
##  3      14096    5128
##  4      12748    4642
##  5      14606    2782
##  6      15311    2491
##  7      14646    2085
##  8      13089    1857
##  9      13263    1677
## 10      14298    1640
## # ℹ 4,362 more rows
```

```
# 17841 customer has the highest number of transactions of 7983.
```

#6(2). Most valuable customer with the highest total sum of transactions.

```
Store_data %>% group_by(CustomerID) %>% filter(!is.na(CustomerID)) %>% summarise(max_spending = sum(TransactionValue)) %>% a
rrange(desc(max_spending))
```

```
## # A tibble: 4,372 × 2
##    CustomerID max_spending
##         <int>        <dbl>
##  1      14646      279489.
##  2      18102      256438.
##  3      17450      187482.
##  4      14911      132573.
##  5      12415      123725.
##  6      14156      113384.
##  7      17511       88125.
##  8      16684       65892.
##  9      13694       62653.
## 10      15311       59419.
## # ℹ 4,362 more rows
```

```
#Most valuable customer with the highest total sum of transactions was with CustomerID 14646.
```

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

```
colMeans(is.na(Store_data)*100)
```

```
##         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           0.00000           0.00000           0.00000
## New_Invoice_Month
##           0.00000
```

```
#the percentage of missing values for each variable in the dataset was 24.92669
```

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

```
Store_data %>% filter(is.na(CustomerID)) %>% group_by(Country) %>% count() %>% arrange(desc(n))
```

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

```
#There are in total 9 countries with missing CustomerID.
```

#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).

```
Avg_days <- Store_data %>% group_by(CustomerID) %>% distinct(New_Invoice_Date) %>% arrange(desc(CustomerID)) %>% mutate(come
back=New_Invoice_Date-lag(New_Invoice_Date)) %>% filter(!is.na(comeback))
Avg_days
```

```
## # A tibble: 15,200 × 3
## # Groups:   CustomerID [2,992]
##    CustomerID New_Invoice_Date comeback
##         <int> <date>           <drtn>
##  1      18287 2011-10-12       143 days
##  2      18287 2011-10-28        16 days
##  3      18283 2011-01-23        17 days
##  4      18283 2011-02-28        36 days
##  5      18283 2011-04-21        52 days
##  6      18283 2011-05-23        32 days
##  7      18283 2011-06-14        22 days
##  8      18283 2011-06-23         9 days
##  9      18283 2011-07-14        21 days
## 10      18283 2011-09-05        53 days
## # ℹ 15,190 more rows
```

```
mean(Avg_days$comeback)
```

```
## Time difference of 38.4875 days
```

```
#On an average of approximately the costumers comeback to the website for their next shopping for every 38 days.
```

#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?

```
France_Trans_Cancelled <- Store_data %>% filter(Country=="France",Quantity<0) %>% count()
France_Trans <- Store_data %>% filter(Country=="France") %>% count()
Return_Percentage_France <- France_Trans_Cancelled/France_Trans*100
Return_Percentage_France
```

```
##          n
## 1 1.741264
```

```
#The return rate of customers who made purchases in France is 1.741264%.
```

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

```
Store_data %>% group_by(StockCode) %>% summarise(Total=sum(TransactionValue)) %>% arrange(desc(Total))
```

```
## # A tibble: 4,070 × 2
##    StockCode   Total
##    <chr>       <dbl>
##  1 DOT       206245.
##  2 22423     164762.
##  3 47566      98303.
##  4 85123A     97894.
##  5 85099B     92356.
##  6 23084      66757.
##  7 POST       66231.
##  8 22086      63792.
##  9 84879      58960.
## 10 79321      53768.
## # ℹ 4,060 more rows
```

```
#The product DOT that has generated the highest revenue of 206245 for the retailer.
```

#12. How many unique customers are represented in the dataset?

```
Store_data %>% group_by(CustomerID) %>% unique() %>% count()
```

```
## # A tibble: 4,373 × 2
## # Groups:   CustomerID [4,373]
##    CustomerID     n
##         <int> <int>
##  1      12346     2
##  2      12347   182
##  3      12348    31
##  4      12349    73
##  5      12350    17
##  6      12352    95
##  7      12353     4
##  8      12354    58
##  9      12355    13
## 10      12356    59
## # ℹ 4,363 more rows
```

```
#There are total 4,373 unique customers in the dataset.
```