BA Assignment 2

Pavan Chaitanya

2022-10-20

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse
1.3.2 ──
## ✓ ggplot2 3.3.6      ✓ purrr   0.3.4
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.0      ✓ stringr 1.4.1
## ✓ readr   2.1.2      ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
Online_Retail <- read_csv("C:/Users/Pavan
Chaitanya/Downloads/Online_Retail.csv")
```

```
## Rows: 541909 Columns: 8
## ── Column specification
──────────────────────────────────────────────────────
## Delimiter: ","
## chr (5): InvoiceNo, StockCode, Description, InvoiceDate, Country
## dbl (3): Quantity, UnitPrice, CustomerID
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

```
head(Online_Retail)
```

```
## # A tibble: 6 × 8
##   InvoiceNo StockCode Description        Quant…¹ Invoi…² UnitP…³ Custo…⁴
Country
##   <chr>     <chr>     <chr>                <dbl> <chr>     <dbl>   <dbl>
<chr>
## 1 536365    85123A    WHITE HANGING HEA…       6 12/1/2…    2.55   17850
United…
## 2 536365    71053     WHITE METAL LANTE…       6 12/1/2…    3.39   17850
United…
## 3 536365    84406B    CREAM CUPID HEART…       8 12/1/2…    2.75   17850
United…
## 4 536365    84029G    KNITTED UNION FLA…       6 12/1/2…    3.39   17850
United…
```

```
## 5 536365    84029E    RED WOOLLY HOTTIE…       6 12/1/2…    3.39   17850
United…
## 6 536365    22752     SET 7 BABUSHKA NE…       2 12/1/2…    7.65   17850
United…
## # … with abbreviated variable names ¹Quantity, ²InvoiceDate, ³UnitPrice,
## #   ⁴CustomerID
```

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 %>%
group_by(Country)  %>%
  tally(sort = TRUE) %>% summarise(Country, Counts = n, Percent =
n/sum(n)*100) %>% filter(Percent > 1)
```

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

UK, Germany, France, and EIRE account for more than 1% of the total transactions in this dataset.

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

```
Online_Retail <- mutate(Online_Retail, TransactionValue = Quantity *
UnitPrice)
head(Online_Retail[, 9])
```

```
## # A tibble: 6 × 1
##   TransactionValue
##              <dbl>
## 1             15.3
## 2             20.3
## 3             22
## 4             20.3
## 5             20.3
## 6             15.3
```

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.

```
Online_Retail %>%
group_by(Country) %>%
  summarise(TransValueSum = sum(TransactionValue)) %>% filter(TransValueSum >
130000) %>% arrange(desc(TransValueSum))
```

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

UK, Netherlands, EIRE, Germany, France, and Australia are the countries where their sum is greater than 130,000 British Pound.

4. 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.

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

```
head(Online_Retail)
```

```
## # A tibble: 6 × 9
##   InvoiceNo StockCode Descript…¹ Quant…² Invoi…³ UnitP…⁴ Custo…⁵ Country
Trans…⁶
##   <chr>     <chr>     <chr>         <dbl> <chr>     <dbl>   <dbl> <chr>
<dbl>
## 1 536365    85123A    WHITE HAN…        6 12/1/2…    2.55   17850 United…
15.3
## 2 536365    71053     WHITE MET…        6 12/1/2…    3.39   17850 United…
```

```
20.3
## 3 536365    84406B    CREAM CUP…       8 12/1/2…    2.75   17850 United…
22
## 4 536365    84029G    KNITTED U…       6 12/1/2…    3.39   17850 United…
20.3
## 5 536365    84029E    RED WOOLL…       6 12/1/2…    3.39   17850 United…
20.3
## 6 536365    22752     SET 7 BAB…       2 12/1/2…    7.65   17850 United…
15.3
## # … with abbreviated variable names ¹Description, ²Quantity, ³InvoiceDate,
## #   ⁴UnitPrice, ⁵CustomerID, ⁶TransactionValue
```

```
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"))
head(Online_Retail)
```

```
## # A tibble: 6 × 13
##   InvoiceNo StockCode Descript…¹ Quant…² Invoi…³ UnitP…⁴ Custo…⁵ Country
Trans…⁶
##   <chr>     <chr>     <chr>         <dbl> <chr>     <dbl>   <dbl> <chr>
<dbl>
## 1 536365    85123A    WHITE HAN…       6 12/1/2…    2.55   17850 United…
15.3
## 2 536365    71053     WHITE MET…       6 12/1/2…    3.39   17850 United…
20.3
## 3 536365    84406B    CREAM CUP…       8 12/1/2…    2.75   17850 United…
22
## 4 536365    84029G    KNITTED U…       6 12/1/2…    3.39   17850 United…
20.3
## 5 536365    84029E    RED WOOLL…       6 12/1/2…    3.39   17850 United…
20.3
## 6 536365    22752     SET 7 BAB…       2 12/1/2…    7.65   17850 United…
15.3
## # … with 4 more variables: New_Invoice_Date <date>, Invoice_Day_Week
<chr>,
##   New_Invoice_Hour <dbl>, New_Invoice_Month <dbl>, and abbreviated
variable
##   names ¹Description, ²Quantity, ³InvoiceDate, ⁴UnitPrice, ⁵CustomerID,
##   ⁶TransactionValue
```

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

```
Online_Retail %>%
  group_by(Invoice_Day_Week) %>%
  tally(sort = TRUE) %>%
  summarise(Invoice_Day_Week, TransactionCounts = n, Percent = n/sum(n)*100)
```

```
%>%
  arrange(desc(TransactionCounts))
```

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

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

```
Online_Retail %>%
  group_by(Invoice_Day_Week) %>%
  summarise(TransValueSum = sum(TransactionValue)) %>%
  mutate(TransValuePercent = TransValueSum/sum(TransValueSum)) %>%
  arrange(desc(TransValueSum))
```

```
## # A tibble: 6 × 3
##   Invoice_Day_Week TransValueSum TransValuePercent
##   <chr>                    <dbl>             <dbl>
## 1 Thursday              2112519             0.217
## 2 Tuesday               1966183.            0.202
## 3 Wednesday             1734147.            0.178
## 4 Monday                1588609.            0.163
## 5 Friday                1540611.            0.158
## 6 Sunday                 805679.            0.0827
```

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

```
Online_Retail %>%
  group_by(New_Invoice_Month) %>%
  summarise(TransValueSum = sum(TransactionValue)) %>%
  mutate(TransValuePercent = TransValueSum/sum(TransValueSum)) %>%
  arrange(desc(TransValuePercent))
```

```
## # A tibble: 12 × 3
##    New_Invoice_Month TransValueSum TransValuePercent
##                <dbl>         <dbl>             <dbl>
##  1                11      1461756.            0.150
##  2                12      1182625.            0.121
##  3                10      1070705.            0.110
##  4                 9      1019688.            0.105
##  5                 5       723334.            0.0742
```

```
##  6                 6       691123.            0.0709
##  7                 3       683267.            0.0701
##  8                 8       682681.            0.0700
##  9                 7       681300.            0.0699
## 10                 1       560000.            0.0574
## 11                 2       498063.            0.0511
## 12                 4       493207.            0.0506
```

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

```
Online_Retail %>%
  filter(Country == "Australia") %>%
  group_by(InvoiceDate) %>%
  tally(sort = TRUE) %>%
  filter(n == max(n))
```

```
## # A tibble: 1 × 2
##   InvoiceDate         n
##   <chr>           <int>
## 1 6/15/2011 13:37   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.

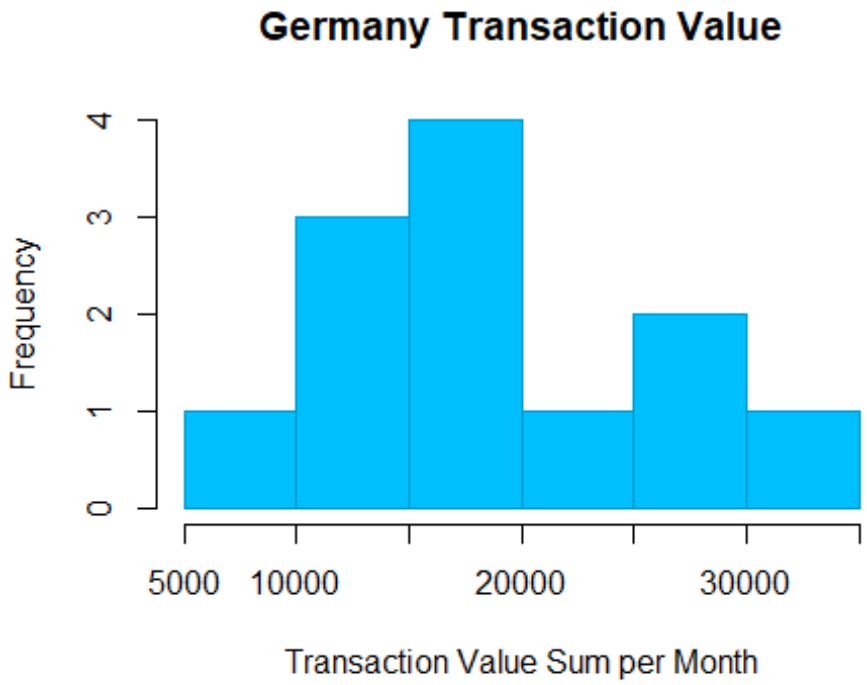
```
Online_Retail %>%
  group_by(New_Invoice_Hour) %>%
  tally(sort = TRUE) %>%
  filter(New_Invoice_Hour>= 7 & New_Invoice_Hour<=20) %>%
  arrange(n) %>%
  head(5)
```

```
## # A tibble: 5 × 2
##   New_Invoice_Hour     n
##              <dbl> <int>
## 1                7   383
## 2               20   871
## 3               19  3705
## 4               18  7974
## 5                8  8909
```

The answer is the 19th and 20th since they are the 2nd and 3rd lowest values and then combined would be the lowest sum of two consecutive hours.

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

```
Online_Retail %>%
  group_by(Country) %>%
  filter(Country == "Germany") %>%
  group_by(New_Invoice_Month) %>%
  summarise(TransValueSum = sum(TransactionValue)) -> Germany
hist(Germany$TransValueSum, border = "deepskyblue3", main = "Germany
Transaction Value", xlab = "Transaction Value Sum per Month", ylab =
"Frequency", col = "deepskyblue")
```



6.Which customer had the highest number of transactions? Which customer is most valuable

```
Online_Retail %>%
  group_by(CustomerID) %>%
  tally(sort = TRUE) %>%
  filter(!is.na(CustomerID)) %>%
  filter(n==max(n))
```

```
## # A tibble: 1 × 2
##   CustomerID     n
##        <dbl> <int>
## 1      17841  7983
```

```
Online_Retail %>%
  group_by(CustomerID) %>%
  summarise(Transvaluesum = sum(TransactionValue)) %>%
  filter(!is.na(CustomerID)) %>%
  filter(Transvaluesum == max(Transvaluesum))
```

```
## # A tibble: 1 × 2
##   CustomerID Transvaluesum
##        <dbl>         <dbl>
## 1      14646       279489.
```

Customer 17841 has the most transactions of 7,983 and customer 14646 is the most valuable spending 279,489 British Pound.

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

```
colMeans(is.na(Online_Retail))
```

```
##         InvoiceNo         StockCode       Description          Quantity
##       0.000000000       0.000000000       0.002683107       0.000000000
##       InvoiceDate         UnitPrice        CustomerID           Country
##       0.000000000       0.000000000       0.249266943       0.000000000
##  TransactionValue  New_Invoice_Date  Invoice_Day_Week  New_Invoice_Hour
##       0.000000000       0.000000000       0.000000000       0.000000000
## New_Invoice_Month
##       0.000000000
```

Only columns “Description” (.2% missing values) and “CustomerID” (24.9% missing values) have missing values.

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

```
Online_Retail %>%
  filter(is.na(CustomerID)) %>%
  group_by(Country) %>%
  summarise(CustomerID) %>%
  tally(sort = TRUE) # Total "NA" by country.
```

```
## `summarise()` has grouped output by 'Country'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 9 × 2
##   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
```

9. On average, how often the costumers comeback to the website for their next shopping?

```
Online_Retail %>% # Creating a variable for the number of days between
visits.
  select(CustomerID, New_Invoice_Date) %>%
  group_by(CustomerID) %>%
  distinct(New_Invoice_Date) %>%
  arrange(desc(CustomerID)) %>%
  mutate(DaysBetween = New_Invoice_Date - lag(New_Invoice_Date))->
CustDaysBtwVisit #Combined DaysBetween per CustomerID.

CustDaysBtwVisit %>%
   filter(!is.na(DaysBetween)) -> RetCustDaysBtwVisits # Filtered "NA" from
dataset.
mean(RetCustDaysBtwVisits$DaysBetween)

## Time difference of 38.4875 days
```

The customers who did return had an average of 38.5 days between visits.

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?

```
Online_Retail %>% # Found the returns from France.
  group_by(Country) %>%
  filter(Country == "France") %>%
  select(Country, Quantity) %>%
  filter(Quantity < 0) -> FrenchReturns
  Online_Retail %>%  # Found the purchases from France.
  group_by(Country) %>%
  filter(Country == "France") %>%
  select(Quantity, Country) %>%
  filter(Quantity > 0) -> FrenchPurchases
FRReturns <- sum(FrenchReturns$Quantity) # calculated the quantity of returns
```

```
from France.
FRTransactions <- sum(FrenchPurchases$Quantity) # calculated the quanity of
purchased from France.
FRReturns/FRTransactions *100 # Using the above two numbers, I then
calculated the return rate.

## [1] -1.448655
```

France has a 1.45% return rate.

11. What is the product that has generated the highest revenue for the retailer?

```
Online_Retail %>%
  group_by(StockCode) %>%
  summarise(TransactionValueTot = sum(TransactionValue)) %>%
  arrange(desc(TransactionValueTot)) %>%
  filter(StockCode != "DOT") %>%  # Looks like this is postage for delivering
products.
  filter(TransactionValueTot == max(TransactionValueTot))

## # A tibble: 1 × 2
##   StockCode TransactionValueTot
##   <chr>                   <dbl>
## 1 22423                 164762.

Online_Retail %>%
  group_by(StockCode) %>%
  filter(StockCode == "22423") %>%
  select(StockCode, Description) %>%
  distinct(StockCode, Description) %>%
  filter(Description == "REGENCY CAKESTAND 3 TIER")

## # A tibble: 1 × 2
## # Groups:   StockCode [1]
##   StockCode Description
##   <chr>     <chr>
## 1 22423     REGENCY CAKESTAND 3 TIER
```

Regency 3 tiered cakestand had the highest revenue.

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

```
Online_Retail %>%
  group_by(CustomerID) %>%
  distinct(CustomerID) -> UniqueCustomers
  length(UniqueCustomers$CustomerID)

## [1] 4373
```

There are 4373 unique customers in this dataset.