The Summer Data Science Intern Challenge Data Set

By Chinedu H Obetta

Load Packages

```
library(readxl)  # To read files
library(ggplot2) # To plot graphs etc
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.
3.1 --
```

```
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflict
s() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

Environment Set up and Data Import

```
setwd("C:/Users/Kenechi/Documents/SHOPIFY")
shopify_data = read_xlsx("Winter Data Science Intern Challenge Data Set.xlsx"
)
```

Overview of the dataset

```
dim(shopify_data)
```

```
## [1] 5000    7
```

Observation:

- The dataset contains seven(7) variables and 5000 observations.

Sanity Checks

```
# To confirm that the data is read in properly

head(shopify_data)

## # A tibble: 6 x 7
##   order_id shop_id user_id order_amount total_items payment_method
##      <dbl>   <dbl>   <dbl>        <dbl>       <dbl> <chr>
## 1       16      42     607       704000        2000 credit_card
## 2       61      42     607       704000        2000 credit_card
## 3      521      42     607       704000        2000 credit_card
## 4     1105      42     607       704000        2000 credit_card
## 5     1363      42     607       704000        2000 credit_card
## 6     1437      42     607       704000        2000 credit_card
## # ... with 1 more variable: created_at <dttm>

tail(shopify_data)

## # A tibble: 6 x 7
##   order_id shop_id user_id order_amount total_items payment_method
##      <dbl>   <dbl>   <dbl>        <dbl>       <dbl> <chr>
## 1     4184      92     844           90           1 debit
## 2     4220      92     747           90           1 credit_card
## 3     4415      92     927           90           1 credit_card
## 4     4761      92     937           90           1 debit
## 5     4924      92     965           90           1 credit_card
## 6     4933      92     823           90           1 credit_card
## # ... with 1 more variable: created_at <dttm>

names(shopify_data)

## [1] "order_id"       "shop_id"        "user_id"        "order_amount"
## [5] "total_items"    "payment_method" "created_at"
```

Observations:

- The values of the variable seems consistent.

An Overview of the dataset

```
summary(shopify_data)

##     order_id       shop_id          user_id       order_amount
##  Min.   :   1   Min.   :  1.00   Min.   :607.0   Min.   :    90
##  1st Qu.:1251   1st Qu.: 24.00   1st Qu.:775.0   1st Qu.:   163
##  Median :2500   Median : 50.00   Median :849.0   Median :   284
##  Mean   :2500   Mean   : 50.08   Mean   :849.1   Mean   :  3145
##  3rd Qu.:3750   3rd Qu.: 75.00   3rd Qu.:925.0   3rd Qu.:   390
##  Max.   :5000   Max.   :100.00   Max.   :999.0   Max.   :704000
##   total_items       payment_method       created_at
##  Min.   :   1.000   Length:5000        Min.   :2017-03-01 00:08:09
##  1st Qu.:   1.000   Class :character   1st Qu.:2017-03-08 07:08:04
```

```
##  Median :   2.000   Mode  :character   Median :2017-03-16 00:21:20
##  Mean   :   8.787                      Mean   :2017-03-15 22:20:37
##  3rd Qu.:   3.000                      3rd Qu.:2017-03-23 10:39:57
##  Max.   :2000.000                      Max.   :2017-03-30 23:55:35
```

```
str(shopify_data)
```

```
## tibble [5,000 x 7] (S3: tbl_df/tbl/data.frame)
##  $ order_id      : num [1:5000] 16 61 521 1105 1363 ...
##  $ shop_id       : num [1:5000] 42 42 42 42 42 42 42 42 42 42 ...
##  $ user_id       : num [1:5000] 607 607 607 607 607 607 607 607 607 607 ..
.
##  $ order_amount  : num [1:5000] 704000 704000 704000 704000 704000 704000
704000 704000 704000 704000 704000 ...
##  $ total_items   : num [1:5000] 2000 2000 2000 2000 2000 2000 2000 2000 20
00 2000 ...
##  $ payment_method: chr [1:5000] "credit_card" "credit_card" "credit_card"
"credit_card" ...
##  $ created_at    : POSIXct[1:5000], format: "2017-03-07 04:00:00" "2017-03
-04 04:00:00" ...
```

#Observations: * There are marked differences between the maximum value and the third quartiles in the variable “order_amount” and “total_items”. The marked differences indicate the existence of outliers on the variables.

- There are no missing values in the dataset.
- The variables “Shop_Id”, “order_id”, and “user_ID” were classified as numeric variables instead of factor variables.
- The variable “payment_method’ needs to be changed to a factor variable. Observed that the variable”created_at” is not formatted into date format.

#Modification of the identified variables

```
shopify_data$order_id = as.factor(shopify_data$order_id)
shopify_data$shop_id = as.factor(shopify_data$shop_id)
shopify_data$user_id = as.factor(shopify_data$user_id)
shopify_data$payment_method = as.factor(shopify_data$payment_method)
```

##Summer 2022 Data Science Intern Challenge

Question: On Shopify, we have exactly 100 sneaker shops, and each of these shops sells only one model of shoe. We want to do some analysis of the average order value (AOV). When we look at orders data over a 30 day window, we naively calculate an AOV of $3145.13. Given that we know these shops are selling sneakers, a relatively affordable item, something seems wrong with our analysis.

#Solutions

```
summary(shopify_data$order_amount)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      90     163     284    3145     390  704000
```

1a.i. Think about what could be going wrong with our calculation.

Solutions:

The issue with the calculation could be the wrong interpretation of the variable " the order-amount". The variable "order_amount" is the sales realized for the selling of shoes across the 100 shops. Therefore, the calculated sum of $3,145.13, is the average sales volume per shop over a 30-day window.

#1bi: Think about a better way to evaluate this data.

Average Order Value(AOV) is the ratio of total order_amount to total_items.

This is calculated as follows;

```
data_treated=shopify_data

data_treated$aov_shop = data_treated$order_amount/data_treated$total_items
head(data_treated[,c(2,8)], 10)
```

```
## # A tibble: 10 x 2
##    shop_id aov_shop
##    <fct>      <dbl>
##  1 42           352
##  2 42           352
##  3 42           352
##  4 42           352
##  5 42           352
##  6 42           352
##  7 42           352
##  8 42           352
##  9 42           352
## 10 42           352
```

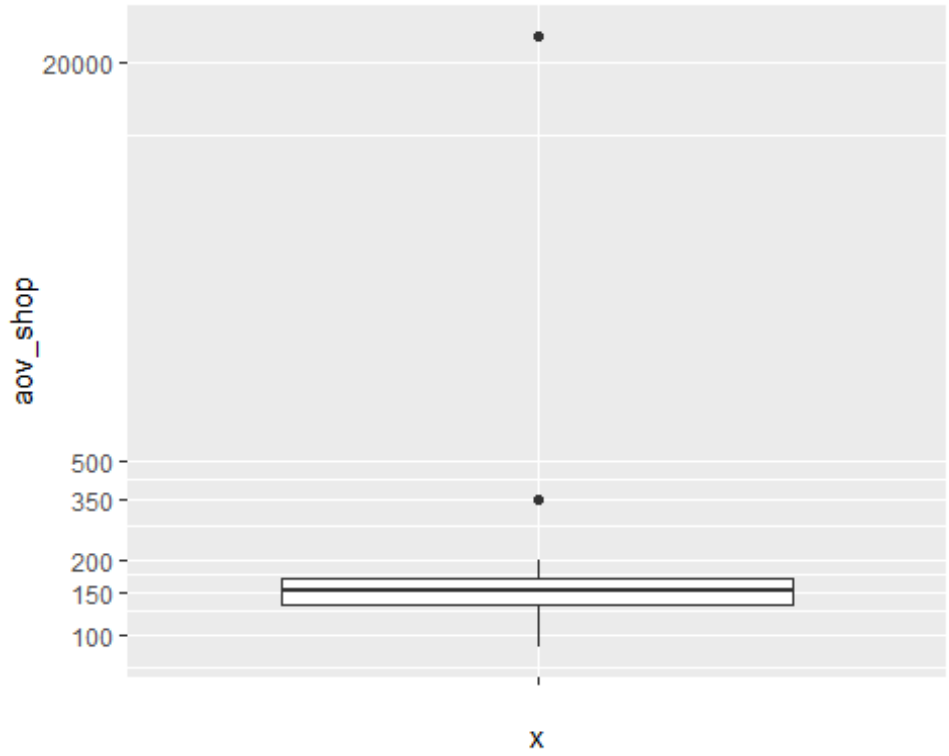
```
AOV = mean(data_treated$aov_shop)
AOV

## [1] 387.7428

summary(data_treated$aov_shop)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    90.0   133.0   153.0   387.7   169.0 25725.0
```

#Observation:

- The average order value for Shopify using the mean is $387.7. However, a further review of the summarized values shows the existence of outliers. These call for further review.

Further analysis of the average order value across the 100 shops

```
ggplot(data = data_treated, aes('', aov_shop )) +
  geom_boxplot() + coord_trans(y = "log10") +
   scale_y_continuous(breaks = c(50, 100,150,200,350,500,20000)) +
  theme()
```



#Observations:

The review of the boxplot above confirms the existence of outliers that must have influenced the mean of $387.7. It is not accurate to use the mean or median to determine the average order value for Shopify.

Calculation of Mode

```
b = data_treated$aov_shop

Mode_aov <- function(b) {
 a <- table(b)
 as.numeric(names(a)[a == max(a)])
}

Mode = Mode_aov(b)

Mode

## [1] 153
```

Question 1b: What metric would you report for this dataset?

I would report MODE as the metric for the dataset. Mode is another measure of central tendencies that can determine the average order value(AOV). The Mode, as a determinant, is not influenced by outliers like mean and median.

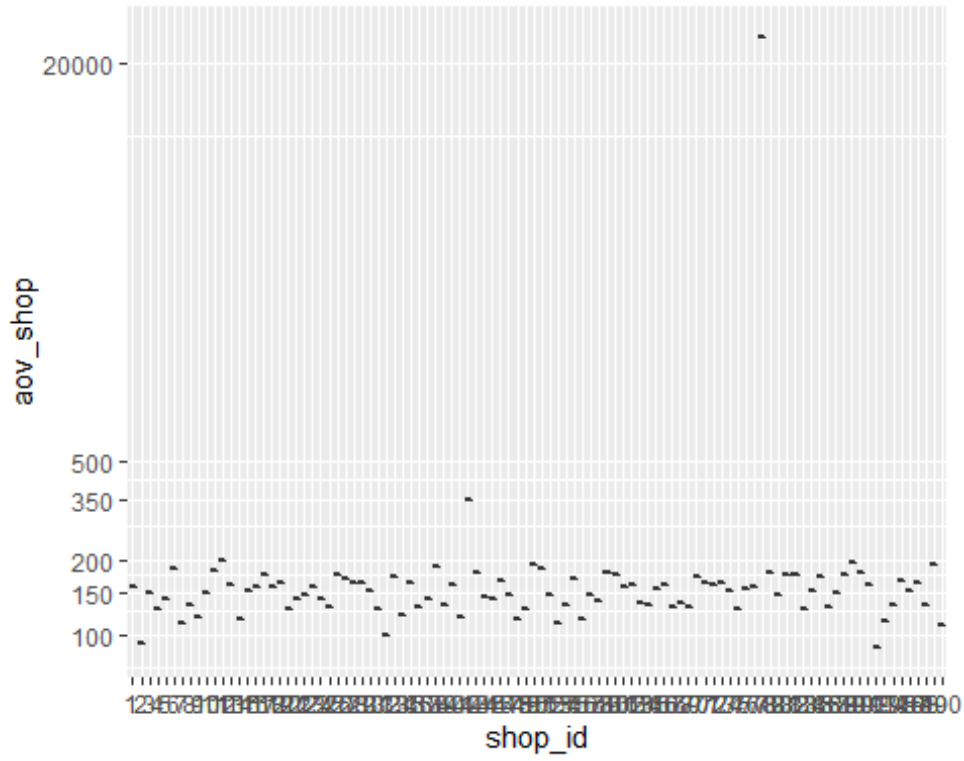
#Question 1c: c. What is its value?

The value is $153 as calculated above.

*******FURTHER ANALYSIS*******

#Analysis of the outliers

```
ggplot(data = data_treated , aes( x = shop_id, y = aov_shop )) +
  geom_boxplot() + coord_trans(y = "log10") +
   scale_y_continuous(breaks = c(50, 100,150,200,350,500,20000)) +
  theme()
```



#Observations

*Over 90% of the AOV is less than $360. Hence, a further review of the dataset to determine the specific shops responsible for the outliers.

Review of shops vis a vis AOV

```
shops_with_outliers = data_treated %>%

select(shop_id, aov_shop) %>%
  filter(aov_shop > 360)

unique(shops_with_outliers)

## # A tibble: 1 x 2
##   shop_id aov_shop
##   <fct>      <dbl>
## 1 78         25725
```

#Observations:

- The AOV for the shop with ID 78 is $25,725 that is 168 times that of the reported AOV for Shopify. A further review of the dataset submitted by SHOPiD 78 could not reveal any notable trend. It could be that the dataset for the shop was not captured correctly. Kindly see below the analysis.

```
data_treated$hour= hour(data_treated$created_at)
data_treated$day = day(data_treated$created_at)
data_treated$weekdays = weekdays(data_treated$created_at)
```

```r
case = data_treated %>%
  select(shop_id, aov_shop, hour, day, weekdays) %>%
  filter(shop_id ==78)

table(case$day, case$weekdays)
```

```
##
##       Friday Monday Saturday Sunday Thursday Tuesday Wednesday
##   1        0      0        0      0        0       0         1
##   2        0      0        0      0        3       0         0
##   4        0      0        1      0        0       0         0
##   5        0      0        0      1        0       0         0
##   9        0      0        0      0        2       0         0
##   11       0      0        1      0        0       0         0
##   12       0      0        0      3        0       0         0
##   14       0      0        0      0        0       2         0
##   15       0      0        0      0        0       0         2
##   16       0      0        0      0        5       0         0
##   17       5      0        0      0        0       0         0
##   18       0      0        4      0        0       0         0
##   19       0      0        0      1        0       0         0
##   20       0      1        0      0        0       0         0
##   21       0      0        0      0        0       1         0
##   22       0      0        0      0        0       0         2
##   25       0      0        2      0        0       0         0
##   26       0      0        0      4        0       0         0
##   27       0      3        0      0        0       0         0
##   29       0      0        0      0        0       0         1
##   30       0      0        0      0        1       0         0
```

Question 2:

For this question you'll need to use SQL. Follow this link to access the data set required for the challenge. Please use queries to answer the following questions. Paste your queries along with your final numerical answers below.

#a. How many orders were shipped by Speedy Express in total?

```
#Solution: Speedy Express shipped 54 orders

SELECT COUNT(*)
FROM Orders
WHERE ShipperID == 1;
```

#b. What is the last name of the employee with the most orders?

```
Solution: The Last Name of the employee with most orders is Peacock and the c
ount is 40

SELECT LastName, COUNT(*)
FROM Employees
    JOIN Orders
        ON Employees.EmployeeID = Orders.EmployeeID
GROUP BY LastName
ORDER BY COUNT(Orders.EmployeeID) DESC
LIMIT 1
```

#c. What product was ordered the most by customers in Germany?

```
Soluton: The product that was ordered most by customer in Germany is Steeleye
Stout.

SELECT ProductName, Country, Quantity
FROM Products
```

```
    JOIN OrderDetails
        ON Products.ProductID = OrderDetails.ProductID
    JOIN Orders
        ON Orders.OrderID =OrderDetails.OrderID
    JOIN Customers
        ON Customers.CustomerID = Orders.CustomerID
        WHERE Country IS 'Germany'
        GROUP BY ProductName
        ORDER BY Quantity DESC
```