Shopify Assessment1

Anil Kumar pathipati

08/09/2021

Introduction :

Question 1: Given some sample data, write a program to answer the following: click here to access the required data set

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

Lets go ahead and load the given data set

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

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

```
library(plotly)
```

```
##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

```
df<-read.csv("C://Users//maila//Desktop//Test REPL//REPL_ML_Exercise//2019 Winter Data Science Intern Ch
head(df)
```

```
##   order_id shop_id user_id order_amount total_items payment_method
## 1        1      53     746          224           2           cash
## 2        2      92     925           90           1           cash
## 3        3      44     861          144           1           cash
## 4        4      18     935          156           1    credit_card
## 5        5      18     883          156           1    credit_card
## 6        6      58     882          138           1    credit_card
##            created_at
## 1 2017-03-13 12:36:56
## 2 2017-03-03 17:38:52
## 3  2017-03-14 4:23:56
## 4 2017-03-26 12:43:37
## 5  2017-03-01 4:35:11
## 6 2017-03-14 15:25:01
```

It is said that the AOV found is #3145.13. Lets try to understand where this is coming from. For now lets calculate the same by taking straight average of the sale value

It looks like the above value is a straight average value taken from the order amount.

```
mean(df$order_amount)
```

```
## [1] 3145.128
```

```
summary(df)
```

```
##     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        Length:5000
##  1st Qu.:   1.000   Class :character   Class :character
##  Median :   2.000   Mode  :character   Mode  :character
##  Mean   :   8.787
##  3rd Qu.:   3.000
##  Max.   :2000.000
```

From above, we see that the order amount has a maximum value of 704000 which looks like an outlier. Now, lets look at what exactly is this transaction.

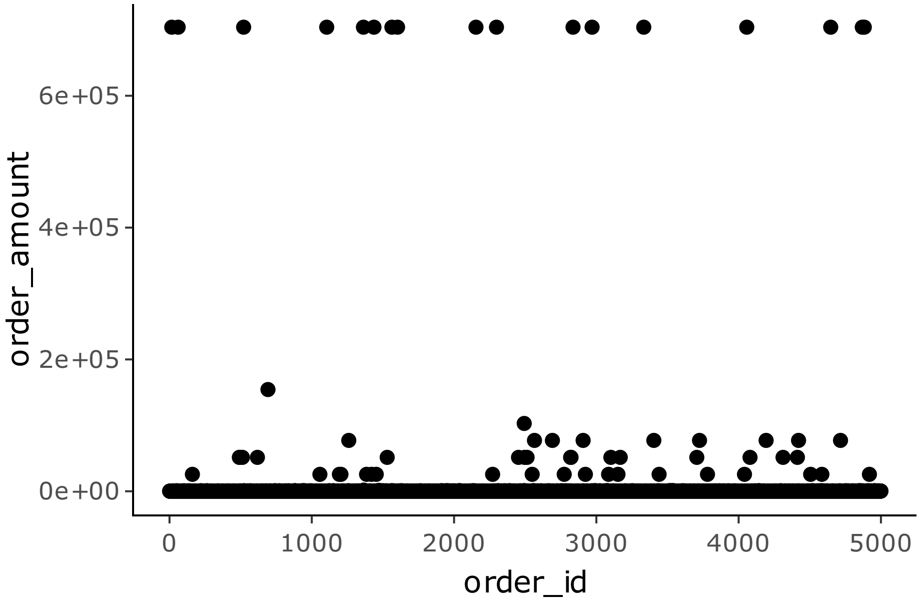
```
subset(df,df$order_amount==704000)
```

```
##    order_id shop_id user_id order_amount total_items payment_method
## 16       16      42     607       704000        2000    credit_card
## 61         61       42     607       704000        2000    credit_card
## 521       521       42     607       704000        2000    credit_card
## 1105     1105       42     607       704000        2000    credit_card
## 1363     1363       42     607       704000        2000    credit_card
## 1437     1437       42     607       704000        2000    credit_card
## 1563     1563       42     607       704000        2000    credit_card
## 1603     1603       42     607       704000        2000    credit_card
## 2154     2154       42     607       704000        2000    credit_card
## 2298     2298       42     607       704000        2000    credit_card
## 2836     2836       42     607       704000        2000    credit_card
## 2970     2970       42     607       704000        2000    credit_card
## 3333     3333       42     607       704000        2000    credit_card
## 4057     4057       42     607       704000        2000    credit_card
## 4647     4647       42     607       704000        2000    credit_card
## 4869     4869       42     607       704000        2000    credit_card
## 4883     4883       42     607       704000        2000    credit_card
##               created_at
## 16   2017-03-07 4:00:00
## 61   2017-03-04 4:00:00
## 521  2017-03-02 4:00:00
## 1105 2017-03-24 4:00:00
## 1363 2017-03-15 4:00:00
## 1437 2017-03-11 4:00:00
## 1563 2017-03-19 4:00:00
## 1603 2017-03-17 4:00:00
## 2154 2017-03-12 4:00:00
## 2298 2017-03-07 4:00:00
## 2836 2017-03-28 4:00:00
## 2970 2017-03-28 4:00:00
## 3333 2017-03-24 4:00:00
## 4057 2017-03-28 4:00:00
## 4647 2017-03-02 4:00:00
## 4869 2017-03-22 4:00:00
## 4883 2017-03-25 4:00:00
```

From above, we see that there are many transactions with an order amount 70400$ which is done using credit card by same user id 607 with a same shop id 42 and purchased same items which are 2000. This is so weird. It looks like the user is purchasing every 3 days at one particular point same items and in some days the data is duplicated especially on 2017-03-28.

Lets try to see if there are any other transactions like this in our data set. This can be found by visualising the given data set.

```
a=ggplot(df, aes(x=order_id, y=order_amount))+geom_point()+theme_classic()
b=ggplotly(a)
b
```



From above it shows that these transactions with 70400$ are the big outlier when calculating the AOV. It is caused due to the fact that the rows are duplicated and also the there is something not right with this transaction which is done every 3 days. It may be possible only if Shopify was running a sale and there is a limit in the purchase quantity per user/day or 3 days. This has lead for the buyer to accumulate stock at a cheaper price from shopify and he may be planning to sell it high post shopify sale or in his retail.

Answers - Question1 :

One way to look at this may be using a Median value because mean is not always reliable in this skewed data sets. When we look at median we will get an AOV of 284$ which is close to actuals.

I would report a median value for this data set if asked and highlight the transaction which looks like an outlier and clean the raw data set to prevent duplicate transactions.

The median value as said above would be 284$

```
median(df$order_amount)
```

```
## [1] 284
```

Question 2

1. SELECT count(*)FROM Orders AS A, Shippers AS B WHERE A.ShipperId = B.ShipperId AND ShipperName = “Speedy Express”;

Total orders shipped via speedy express are 54

2. SELECT E.LastName FROM Employees AS E, Orders AS A WHERE E.EmployeeID = A.EmployeeID GROUP BY E.EmployeeID ORDER BY count(OrderID) DESC Limit 1;

The Last Name is Peacock

3.SELECT Customers.Country, OrderDetails.ProductID,OrderDetails.Quantity,Products.ProductID,Products.ProductName FROM Customers INNER JOIN Orders on Customers.CustomerID=Orders.CustomerID INNER JOIN OrderDetails ON Orders.OrderID=OrderDetails.OrderID INNER JOIN Products ON OrderDetails.ProductID=Products.ProductID where Country==“Germany” Group By ProductName Order By Quantity Desc Limit 1

The top selling product in Germany is “Steeleye Stout”