# Summer 2022 Data Science Intern Challenge

Zirui Li

January 11, 2022

## Question 1

Please check the Python program "shopify challenge.ipynb" for details.

**(a) Think about what could be going wrong with our calculation. Think about a better way to evaluate this data**

To find out what goes wrong of the dataset, I first analyzed the relationship between AOV and the known features. The features of 'order_amount' and 'total_items' have close relationship with AOV calculation. So I checked the details of 'order_amount' and 'total_items'. In Figure 1, we can tell there are some outliers in the data, e.g. the max number of total items (2000).

```
df.order_amount.describe()  # AOV = 3145.128
```

```
count      5000.000000
mean       3145.128000
std       41282.539349
min          90.000000
25%         163.000000
50%         284.000000
75%         390.000000
max      704000.000000
Name: order_amount, dtype: float64
```

```
df.total_items.describe()
```

```
count    5000.00000
mean        8.78720
std       116.32032
min         1.00000
25%         1.00000
50%         2.00000
75%         3.00000
max      2000.00000
Name: total_items, dtype: float64
```

Figure 1: Details of order_amount

In Figure 2 and Figure 3, there are some points far away from the majorities which can be defined as outliers, so we need to find the details of those outliers.

After the analysis, the store 78 sells a sneaker at 25725 per item, which is unreasonable high, this data may have been entered incorrectly. All transactions at store 78 in March 2017 had a really high order_amount value. It made a huge impact to the AOV calculation so should be treated as

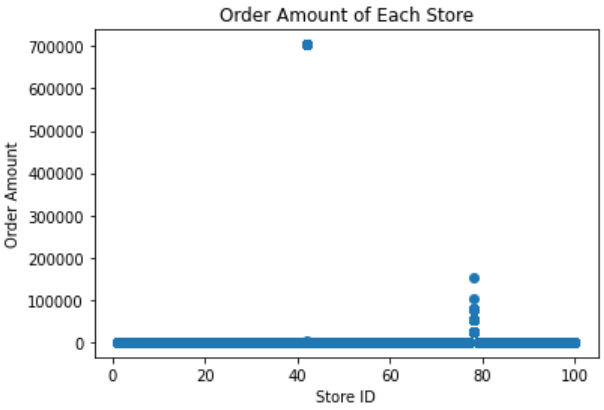


Figure 2: Plot of order_amount by shop_id

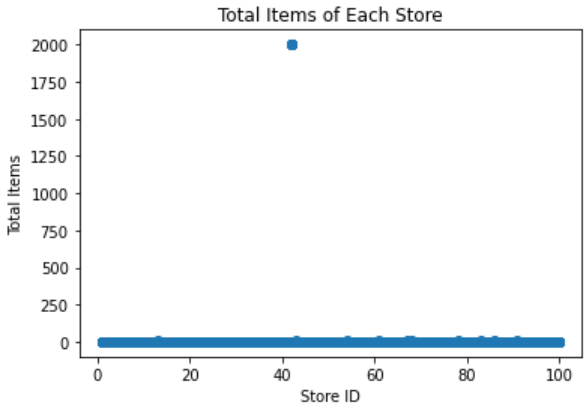


Figure 3: Plot of total_items by shop_id

an outlier.

Not all transactions at store 42 have a really large total_items value. But the transactions with total_items = 2000 can misleading the AOV calculation.

Consider from the time period, this sample data contains the transaction of one month (March) which does not include discount day such as Black Friday, so we can ignore the impact of the discount day on AOV calculation. The sample data contains 100 different stores results, so we need to consider the location/environment factor. The order amount of stores in shopping malls is generally higher than those of stores in small towns. Combining the order amount of all stores and calculating the AOV can make the AOV result inaccurate.

**(b) What metric would you report for this dataset?**
Metric: Calculate the AOV and median values separately by store. Because the order amount of store 78 and store 42 are special cases, calculating the order amount by store will not influence other stores' AOV and median results

**(c) What is its value?**

Figure 4 shows part of the AOV and median results. The details of the final result is saved in "final data report.csv"

| | shop_id | store_amount | store_AOV | store_median |
|---|---|---|---|---|
| **0** | 1 | 13588 | 308.818182 | 316.0 |
| **1** | 2 | 9588 | 174.327273 | 188.0 |
| **2** | 3 | 14652 | 305.250000 | 296.0 |
| 3 | 4 | 13184 | 258.509804 | 256.0 |
| **4** | 5 | 13064 | 290.311111 | 284.0 |
| ... | ... | ... | ... | ... |
| **95** | 96 | 16830 | 330.000000 | 306.0 |
| **96** | 97 | 15552 | 324.000000 | 324.0 |
| **97** | 98 | 14231 | 245.362069 | 266.0 |
| **98** | 99 | 18330 | 339.444444 | 390.0 |
| **99** | 100 | 8547 | 213.675000 | 222.0 |

100 rows × 4 columns

Figure 4: Final Report of the Data

## Question 2

**(a) How many orders were shipped by Speedy Express in total?**

```
SELECT COUNT(Orders.ShipperID) FROM Orders JOIN shippers ON
Orders.ShipperID=Shippers.ShipperID WHERE ShipperName="Speedy␣Express";
```

**The answer is 54.**

**(b) What is the last name of the employee with the most orders?**

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

**The answer is Peacock.**

**(c) What product was ordered the most by customers in Germany?**

```
SELECT ProductName 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 = "Germany"
```

```
GROUP BY ProductName
ORDER BY COUNT(ProductName) DESC LIMIT 1;
```

**The answer is Gorgonzola Telino.**