Shopify application

Russell Parco

05/09/2020

1. a) Looking at the the AOV I first suspect that the metric used was the mean of the order amount and we can confirm this using Python

```
import pandas as pd
sales = pd.read_csv("Sheet1.csv")
sales["order_amount"].mean()
```

```
## 3145.128
```

Now we will look at what went wrong. First I will look at the top ten sales by order amount to see what might have gone wrong.

```
sales.nlargest(n=10, columns="order_amount")[["shop_id","order_amount", "total_items"]]
```

```
##       shop_id  order_amount  total_items
## 15         42        704000         2000
## 60         42        704000         2000
## 520        42        704000         2000
## 1104       42        704000         2000
## 1362       42        704000         2000
## 1436       42        704000         2000
## 1562       42        704000         2000
## 1602       42        704000         2000
## 2153       42        704000         2000
## 2297       42        704000         2000
```

I see that all of the most expensive sales come from the store with store id 42. Now we look at the most expensive sales excluding sales from stor 42.

```
sales[sales["shop_id"] != 42].nlargest(n=10, columns="order_amount")[["shop_id","order_amount", "total_
```

```
##       shop_id  order_amount  total_items
## 691        78        154350           6
## 2492       78        102900           4
## 1259       78         77175           3
## 2564       78         77175           3
## 2690       78         77175           3
## 2906       78         77175           3
## 3403       78         77175           3
## 3724       78         77175           3
## 4192       78         77175           3
## 4420       78         77175           3
```

We see that the next highest sale is nowhere near the order amount of the top sales from store 42. The other stores also do not sell the same amount of volume of shoes in a single order as store 42 does. It is now clear what went wrong with our calculation. The mean metric is very sensitive to outliers, like the outlier orders of 2000 shoes from store 42, therefore these large orders have more influence on the mean rather then the

majority of smaller orders, resulting in the large AOV which is misleading of what the average order for shoes costs. This forces us to ask the question what is the differnce between store 42 and the other stores. Is this store also a manufacturer while the others are not? Does this store supply other stores with shoes? We should even ask the question if this store should be included in our analysis of AOV. However, assuming that we wish to include this store in our analysis we can pick a measure of central tendency that is less sensitive to the large orders from store 42. By calculation the mean of order amounts excluding store 42 and the median of all order amounts will help us decide which to use.

```
round(sales[sales["shop_id"] != 42]["order_amount"].mean(), 2)
```

```
## 754.79
```

```
round(sales["order_amount"].median(), 2)
```

```
## 284.0
```

We see that the mean excluding store 42 is still much greater than the median. The discrepancy may come from the fact that a single pair shoes from store 78 costs $25,725 which is a very large amount for a pair of shoes and so orders from store 78 will have greater influence on the mean. We could also remove the orders from store 78 and calculate the mean, but I would like to avoid going down this dangerous rabbit hole of just removing all outliers from our data set. Therefore, I would suggest using the median of all orders as an alternative for a more reasonable AOV.

b) The metric that I choose to calculate the AOV is the media of all order amounts. The median metric allows us to keep all orders in our calculation and avoid the sensitivity to the outlier orders from stores 42, which has the ability to sell greater volumes of shoes in one order, and store 78 which sells very expensive shoes.

c) the value of the median AOV would be $284

2. a) 54 total orders were shipped by Speedy Express

```
SELECT COUNT(*) FROM Orders
 WHERE ShipperID ==
  (SELECT ShipperID FROM Shippers
        WHERE ShipperName == 'Speedy Express');
```

b) The last name of the employee with the most shipments is Peacock

```
SELECT LastName FROM Employees
 WHERE EmployeeId ==
    (SELECT EmployeeID FROM Orders
    GROUP BY EmployeeID
    ORDER BY COUNT(EmployeeID) DESC
    LIMIT 1);
```

c) The product that was ordered most by Customers in Germany is Boston Crab Meat

```
SELECT ProductName FROM Products
 WHERE ProductID ==
 (SELECT ProductID FROM OrderDetails
 WHERE OrderId in
     (SELECT OrderID FROM Orders
     WHERE CustomerID in
         (SELECT CustomerID FROM Customers
         WHERE Country == 'Germany'))
 GROUP BY ProductID
 ORDER BY SUM(Quantity) DESC Limit 1);
```