Shopify application

Russell Parco

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("2019 Winter Data Science Intern Challenge Data Set - 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 can look at the most expensive sales excluding sales from store 42.

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

```
##       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
```

```
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. Therefore store 42 is an outlier in terms of sales volume. However, now we see that store 78 dominates the order amount. To investigate this we will look at the stores with the highest order amount when removing 42 and 78 and look at the stores with the highest price per unit.

```
sales[~sales["shop_id"].isin([42, 78])].nlargest(n=10, columns="order_amount")[["shop_id","order_amount
```

```
##       shop_id  order_amount  total_items
## 3538       43          1086            6
## 4141       54          1064            8
## 3077       89           980            5
## 2494       50           965            5
## 1563       91           960            6
## 4847       13           960            6
## 2307       61           948            6
## 1256        6           935            5
## 2560        6           935            5
## 3532       51           935            5
```

```
unit_prices = pd.DataFrame({"shop_id" : sales["shop_id"], "unit_price": sales["order_amount"]/sales["to
unit_prices.nlargest(n=10, columns="unit_price")
```

```
##      shop_id  unit_price
## 160       78     25725.0
## 15        42       352.0
## 107       12       201.0
## 205       89       196.0
## 44        99       195.0
## 90        50       193.0
## 242       38       190.0
## 55        51       187.0
## 116        6       187.0
## 70        11       184.0
```

From this it is clear that store 78 is an outlier in unit price. Also after removing these two outer stores we see a variety of stores in the top order amount rankings. This indicates that there aren't anymore obvious outliers that will skew our AOV to be too larger than it should be.

It is now clear what went wrong with our calculation. The mean metric is very sensitive to outliers, like the outlier orders from store 42 and 78. Therefore these orders have more influence on the mean compared to the majority of smaller orders, resulting in the large and misleading AOV. This forces us to ask the question what is the differnce between these two stores and the other stores. Is store 42 also a manufacturer while the others are not? Does 42 store supply other stores with shoes? These may be reasons why the volume of the orders is larger. Why store 78 charges so much more per unit is a harder question to answer. Maybe store 78 ships there shoes internationally to remote locations which would increase costs. We should even ask the question if these stores 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 outliers.

By calculation the mean of order amounts excluding stores 42 and 78, and the median of all order amounts will help us decide which to use.

```
round(sales[~sales["shop_id"].isin([42, 78])]["order_amount"].mean(), 2)
```

```
## 300.16
```

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

```
## 284.0
```

We see that the mean excluding stores 42 and 78 is similar to the median of all orders. Since I would like to avoid excluding data from our data set and the values are similar, 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 median 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 store 42 which has the ability to sell greater volumes of shoes in one order, and store 78 which has much higher unit price.

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);
```