Oliver Pan Shopify Summer 2021 Data Science Challenge

Question 1:

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

I have a couple observations about the data that I would like to highlight.

- The reason that our analysis seems wrong is because we are analysing the affordability of the shoe with the wrong dataset and metrics. There are certain characteristics about the dataset that are skewing the observation, in which there are various resolutions we can take.

    1. **Scale orders to 1 item purchased and engineer new feature (Jupyter Section 1)**
        a. As we can observe, each order does not only have 1 item purchased. Our issue is that we are judging sneaker affordability based off of total order amount, which is incorrect. Hence, we can scale order data to 1 item per order; we cannot compute affordability or average sale cost of the shoe in this dataset without feature engineering, given some orders have more than 1 item purchased.

    2. **Remove anomalies and re-evaluate average order value (Jupyter Section 2)**
        a. For instance, we see 17 instances of 2000 items purchased, which of course will skew the average; we should take out shop 42.
        b. As well, the average order will also be skewed because shop 78 is selling each shoe for about $27,725; sneakers should not be this high!
        c. Therefore after taking out the anomalous shops, shop 42 and 78, we get an average order value of $300, which makes sense as some customers purchase more than 1 item during their visit!
            ■ We also see that the average number of items purchased (after removing outliers) is about 2 items; this seems correct!

3. **Use new metrics (Jupyter Section 3)**
    a. Assuming we leave the anomalies in for data reasons, we can use a new metric to track shoe affordability. As seen previously, the average is $3145.13 because of anomalies.
    b. Hence, we can use the describe() function to assist us with a new metric.
    c. After seeing the metrics, we can also implement z-score to see how each transaction differs from the norm. Then, we can filter out anomalies and see what went wrong with our analysis!

b. What metric would you report for this dataset?

Following up with observation 3, it is often incorrect to judge and analyze the dataset based off of the average. For instance, any majorly large number will skew the average. Hence, we can use percentiles (25%, median, 75%) to judge the entire dataset.

Adding on to our .describe() function, instead of looking at affordability through the average order amount of the **total** dataset, we can look at the z-score of **each order**, to determine if they are within the limits of a typical order. The z-score tells us the number of standard deviations away from the average, in which this feature would capture irregular values (being outside of multiple standard deviations).

Therefore, we now have a metric based off of the entire dataset and a metric to analyze each independent order!

a. What is its value?

The value of using median is that we disregard anomalies, contrary to using average. Even so, we were able to get 25%, median, and 75% using the describe() function. We actually found that the 75% percentile is similar to our average. Hence, median (and percentiles) are now a better judgement of the entire dataset.

In addition, rather than looking at the whole dataset, we can now see each individual transaction and analyze whether it is a valid measure of affordability using z-score. After feature engineering the standard deviation of each transaction, we can analyze when orders are anomalous, whether it be from shop 78 that is overpricing the product, or shop 42 that is selling 2000 items per transaction. Knowing that Shopify works with various businesses, we have pointed out that shop 78 and shop 42 were the reasons why our average order value is skewed, in which there is value to using z-score.

To summarize, one investigation turned into two metrics, using percentiles for judgement of the entire dataset, and z-score as a judgement for each independent order.

Question 2:

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

```
SELECT ShipperName, COUNT(ShipperName) OrdersShipped FROM(
SELECT *
FROM Orders o
LEFT JOIN Shippers s
ON s.ShipperID = o.ShipperID)
GROUP BY ShipperName
HAVING ShipperName = 'Speedy Express'
```

Answer: Speedy Express has shipped 54 orders

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

```
SELECT * FROM(
(SELECT EmployeeID, COUNT(*) NumOrders FROM [Orders]
GROUP BY EmployeeID) e
LEFT JOIN (SELECT EmployeeID, LastName FROM Employees) a
ON e.EmployeeID = a.EmployeeID)
ORDER BY NumOrders DESC
LIMIT 1
```

Answer: Peacock is the last name of the employee with the most orders, with 40 orders

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

Note: This assumes that we sum quantity of product, not number of purchases

```
SELECT ProductName, Country, TotalQuantity FROM (
SELECT ProductId, Country, SUM(Quantity) TotalQuantity FROM OrderDetails od
LEFT JOIN (
SELECT o.OrderID, o.CustomerID, c.Country FROM Orders o
LEFT JOIN (SELECT CustomerId, Country FROM Customers) c
ON o.CustomerID = c.CustomerID) custom
ON od.OrderID = custom.OrderID
GROUP BY Country, ProductID
HAVING Country = 'Germany') germany
LEFT JOIN (SELECT ProductId, ProductName FROM Products) p
```

```
ON germany.ProductId = p.ProductId
ORDER BY TotalQuantity DESC
LIMIT 1
```

Answer: Boston Crab Meat is the product that was ordered most by customers in Germany, with 160 ordered.