Fall 2021 Data Science Intern Challenge

Note: The notebook for question 1 is attached. It is also available on my Github: https://github.com/nazirimu/Shopify-Data-Science-Intern-Challenge

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

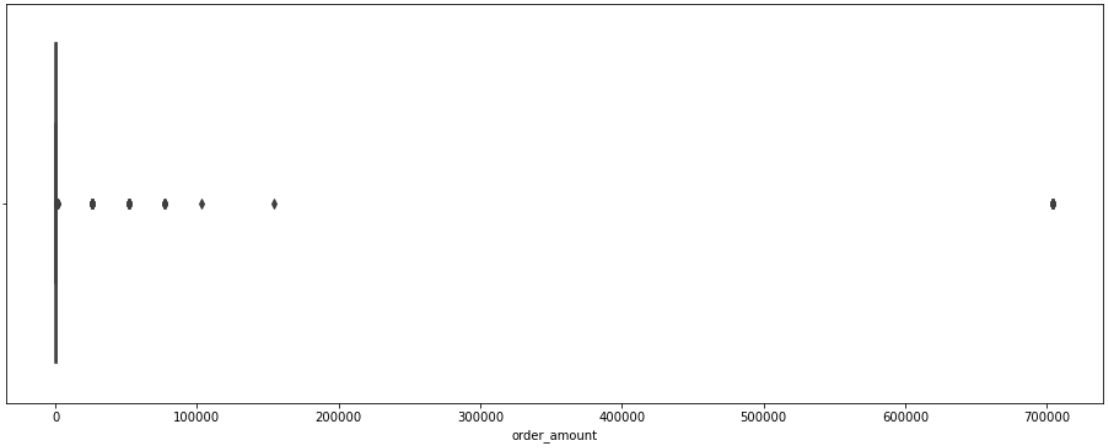
On Shopify, we have exactly 100 sneaker shops, and each of these shops sells only one model of shoe. We want to do some analysis of the average order value (AOV). When we look at orders data over a 30 day window, we naively calculate an AOV of $3145.13. Given that we know these shops are selling sneakers, a relatively affordable item, something seems wrong with our analysis.

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

There are two possible scenarios. Firstly, there could be outliers in the data. Central tendency calculations, such as means (AOV) are sensitive to outliers and the results can be skewed by them. Instead of the mean, the median can be used as it is less sensitive to outliers. Another scenario could be mistakes in the data input. Since I cannot investigate the accuracy of the data, I will assume it to be the first case for my analysis.

After some investigation of the data, there are outliers present in the data. To understand the data better, I will be cleaning and removing the outliers using the interquartile method. The limitation of this method is that it removes 25% range of the data.

Before cleaning the data, the box plot showed a lot of order amounts that were much higher than the rest of the data points. The image below shows the large spread of the original data.



After using the IQR method, order values higher than $730.5 are removed as outliers. This creates a much more reasonable spread. However, the limitation of this method is that it removes 25% range of the data.

Alternative evaluations:
An alternative method could be to use the 95% technique (keeping only values within 2 standard deviations). Instead of removing outliers, another possible way of evaluating this data could be using categorization. After looking at the data, there are some orders with high total order items. As this data is for sneaker shops, it is not likely that it is retailers that are purchasing high amounts of the same shoe. There could be wholesalers buying as well. A reasonable assumption could be using total order items higher than 10 to be wholesalers. Using this classification, every order can be categorized as 'retail' or 'wholesale'. Then, we can group by these categories to get means for both types of order.

Among all three of these techniques, the biggest outliers come from a repeating order pattern of $704000. This order is made on multiple days at the same time (4:00 PM) with the same order quantity. There is a likelihood that this could be a wholesaler or it could also be a bot (since the purchases are so systematic).

I have attached a PDF of the jupyter notebook where all of these suggestions were applied. Along with this, trends and other graphs were analyzed.

b. What metric would you report for this dataset?

With the presence of outliers in this dataset, using a median instead would be a better approach. The cleaned data has a mean close to the median value of the original data.

c. *What is its value?*

The median value of this data is $284. This is a much more reasonable amount for a sneaker store.

Alternative analysis:
According to the categorization method mentioned above, the retail mean was found to be $754. The wholesale mean was found to be $704000

Question 2: For this question you'll need to use SQL. Follow this link to access the data set required for the challenge. Please use queries to answer the following questions.

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

Answer: There are **54** orders shipped by Speedy Express in total.

```
SELECT
  Shippers.ShipperName,
  count(Orders.OrderID) AS number_of_orders
FROM
  Orders
INNER JOIN Shippers ON Shippers.ShipperID=Orders.ShipperID
WHERE
  Shippers.ShipperName="Speedy Express"
GROUP BY
Shippers.ShipperName
```

Thought Process: Selected two column names to visualize the table. Joined Orders to shippers on the column for ShipperID that was common to both. Then filtered the results for orders shipped through speedy express.

ShipperName	number_of_orders
Speedy Express	54

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

Answers: **Peacock** is the name of the employee with the most orders

```
SELECT
   Employees.LastName,
   count(Employees.LastName) As number_of_employees
FROM
   Orders
INNER JOIN Employees ON Employees.EmployeeID=Orders.EmployeeID
GROUP by
   Employees.LastName
ORDER by count(Employees.LastName) DESC
LIMIT 1;
```

Thought Process: Selected two column names to visualize the table. Joined Orders to Employees on the column for Employer ID that was common to both. Then group the results by last name and display the count for each along side.

LastName	number_of_employees
Peacock	40

c. What product was ordered the most by customers in Germany?
Answer: **Boston Crab Meat**

```
SELECT
   Products.ProductName,
   sum(OrderDetails.Quantity)
FROM
   Customers
INNER JOIN Orders ON Orders.CustomerID=Customers.CustomerID
INNER JOIN OrderDetails ON Orders.OrderID=OrderDetails.OrderID
INNER JOIN Products ON OrderDetails.ProductID=Products.ProductID
Where
   Customers.Country="Germany";
```

Thought Process: Selected two column names to visualize the table. First, I joined the *Customers* table to the *Orders* table on the column **customer id** that was common to both to get the **order id**. After, I joined those to the *OrderDetails* table on the column **order id** to get the **product id.** Lastly, I used the **product id to** join the *Products* table to get the names of each product and its sum. The table was filtered for Germany as the country.

Number of Records: 1

ProductName	sum(OrderDetails.Quantity)
Boston Crab Meat	2015