Fall 2022 Data Science Intern Challenge

The whole notebook is available here.

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

> At Shopify, we have precisely 100 sneaker shops, and each of these shops sells only one model of shoe. We want to analyze the average order value (AOV). When looking at order 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.**

Just reading the description of the assignment already states that the AOV number is ridiculously high. Indeed, sneakers are something that anyone could afford, but giving away over $3000 on average is just wrong.

After a careful analysis, it became evident what was causing the problem. First of all, AOV is a naive way of calculating the mean of the column, there is nothing fancy going on there. However, we should always be cautious about the "mean" behavior. It gets dragged around fiercely in case of outliers. A quick overview and descriptive statistics can give the first steps into the investigation:

```
shopify.describe()
```

	shop_id	user_id	order_amount	total_items
count	5000.000000	5000.000000	5000.000000	5000.00000
mean	50.078800	849.092400	3145.128000	8.78720
std	29.006118	87.798982	41282.539349	116.32032
min	1.000000	607.000000	90.000000	1.00000
25%	24.000000	775.000000	163.000000	1.00000
50%	50.000000	849.000000	284.000000	2.00000
75%	75.000000	925.000000	390.000000	3.00000
max	100.000000	999.000000	704000.000000	2000.00000

Fig 1. Descriptive statistics

Fig.1 shows the descriptive statistics for the numeric values of the dataset. Here we can see a couple of interesting points about the order_amount and total_items columns:

1. The average (mean) value of the order_amount column is $3,145.13 as given in the dataset description, so now we know where the number came from.
2. The standard deviation of the order_amount column is $41,282.54 which indicates that the dataset has a high variation. Looking at the quantiles shows that the main spread is within $90 to $390 which is more believable than the price given above.
3. The same behavior is observed for the total_items column. The average (mean) value is almost 9. This means that people buy almost 9 pairs of shoes in one go which is clearly wrong. Moreover, we can see the maximum value of 2,000 items per order. This could be the wholesale buyers which should be treated differently in a separate dataset.

Conclusion

The average (mean) value is not sufficient for the dataset. There are two solutions that can be employed. First, store the outliers separately and double-check with the shops (#42 and #78) for the correctness of data acquisition. Second, we can drop the values and move on with the analysis.

Why would we need to store the outliers separately? Fig 2. shows that there are 17 orders with 2,000 items that brought $704,000 per order in shop #42. Another shop #78 also made 19 successful transactions of $25,725. Since the numbers are not single-instance cases, we should definitely look closer into them!

	shop_id	order_amount	items_per_order	count
0	42	704000	2000	17
1	42	352	1	15
2	42	704	2	13
3	42	1056	3	3
4	42	1408	4	2
5	42	1760	5	1
6	78	25725	1	19
7	78	51450	2	16
8	78	77175	3	9
9	78	102900	4	1
10	78	154350	6	1

Fig 2. Closer look at outliers

b. What metric would you report for this dataset? &

c. What is its value?

If we store the outliers separately, we can use the same AOV formula which will result in a more-or-less adequate value of $300.16.

```
indices     = shopify['shop_id'].isin([42, 78])
outliers    = shopify[indices]
no_outliers = shopify[~indices]

print(f'AOV of the dataset with outliers: ${format(shopify.order_amount.mean(),
".2f")}')
# AOV of the dataset with outliers: $3,145.13

print(f'AOV of the dataset without outliers:
${format(no_outliers.order_amount.mean(), ".2f")}')
# AOV of the dataset without outliers: $300.16

print(f'AOV of outlier shops (#42 & #78): ${format(outliers.order_amount.mean(),
".2f")}')
# AOV of outlier shops (#42 & #78): $146,848.21
```

However, if we cannot separate the values, we can simply switch to the median value. It's less prone to getting dragged around unlike the mean. In both cases with and without outliers, the AOV is $284.

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. Paste your queries along with your final numerical answers below.

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

To get the total number of orders, we need to investigate the Shipper dataset and get the ShipperID for "Speedy Express". Then, we can simply use COUNT function to get the value.

```
SELECT COUNT(OrderID) FROM Orders
WHERE ShipperID = 1
```

Answer: 54

COUNT(OrderID)
54

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

This information can be easily found using GROUP BY and ORDER BY in descending order. The logic is the following: get the last name of each employee and count their occurrence in the Orders table while naming the new variable for further usage. For this purpose, it's named `num_orders`. Then, we order the occurrences in descending order and limit the results to show only one person.

```
SELECT e.LastName, COUNT(*) as num_orders FROM Employees as e
JOIN ORDERS as o ON e.EmployeeID = o.EmployeeID
GROUP BY e.LastName
ORDER BY num_orders DESC
LIMIT 1
```

Answer: Peacock

LastName	num_orders
Peacock	40

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

This question can be answered with two different approaches. **First**, if we select the most popular product by its occurrence without calculating the total items within the order. Here we would calculate the most popular product that customers order. To do that, we will need to find:

1. Customers from Germany only

```
SELECT CustomerID FROM Customers
WHERE Country = 'Germany'
```

2. Select orders from Germany

```
SELECT o.OrderID FROM Orders as o
JOIN Customers as c ON o.CustomerID = c.CustomerID
WHERE c.CustomerID IN (
  SELECT CustomerID FROM Customers
  WHERE Country = 'Germany'
)
```

3. The most popular product in the OrderDetails table

```
SELECT ProductID, COUNT(*) as total FROM OrderDetails
GROUP BY ProductID
ORDER BY total DESC
LIMIT 1
```

4. Name and description of the most popular product

```
SELECT DISTINCT p.ProductID, p.ProductName FROM Products as p
JOIN OrderDetails as o ON p.ProductID = o.ProductID
WHERE p.ProductID IN (SELECT ProductID FROM (
  SELECT ProductID, COUNT(*) as total FROM OrderDetails
  GROUP BY ProductID
  ORDER BY total DESC
  LIMIT 1
))
```

5. Combine everything together

```
SELECT DISTINCT p.ProductID, p.ProductName FROM Products as p
JOIN OrderDetails as o ON p.ProductId = o.ProductID
WHERE o.OrderID in (
      SELECT o.OrderID FROM Orders as o
    JOIN Customers as c ON o.CustomerID = c.CustomerID
    WHERE c.CustomerID IN (
      SELECT CustomerID FROM Customers
```

```
      WHERE Country = 'Germany'
    )
) AND p.ProductID IN (
     SELECT ProductID FROM (
      SELECT ProductID, COUNT(*) as total FROM OrderDetails
      GROUP BY ProductID
      ORDER BY total DESC
      LIMIT 1
      )
)
```

What happened here? I'm gathering the most popular product's id and name by filtering out only customers from Germany whose OrderIDs are present in OrderDetails, and selecting the most relevant product from there.
Answer: Mozzarella di Giovanni is the most popular item **by OrderID**

The **second** option is to look at the quantity-wise selling points. Thus, we will get the most popular items by their **quantity**.

```
SELECT p.ProductName, SUM(o.Quantity) AS total, c.Country
FROM Products as p
JOIN OrderDetails AS o ON p.ProductID = o.ProductID
JOIN Orders AS ord ON ord.OrderID = o.OrderID
JOIN Customers AS c ON c.CustomerID = ord.CustomerID
WHERE c.Country = "Germany"
GROUP BY p.ProductName
ORDER BY total DESC
LIMIT 1;
```

Answer: **Boston Crab Meat** is the most popular item **by quantity**

ProductName	total	Country
Boston Crab Meat	160	Germany