# Fall 2022 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.
b. What metric would you report for this dataset?
c. What is its value?

**Solution 1:**

**a.** The problem with this calculation is the following:

i. The average is not taking into account the quantity of shoes that are being sold on a particular day. For example in the below shown orders -

| order_id | shop_id | user_id | order_amount | total_items | yment_meth | created_at | Date |
|---|---|---|---|---|---|---|---|
| 16 | 42 | 607 | 704000 | 2000 | credit_card | 2017-03-07 4:00:00 | 2017-03-07 |
| 61 | 42 | 607 | 704000 | 2000 | credit_card | 2017-03-04 4:00:00 | 2017-03-04 |

The order amount is quite high and the reason is higher quantity being sold. Using these values directly can skew the average of the product

ii. Average as a statistic is inherently prone to outliers. Extremely high or low values could skew the average towards their direction. For example in the orders by store 78 as shown below sneakers were sold at a much higher price compared to other stores (Store 92 is selling the same product in 90 dollars v/s 25,725)

| order_id | shop_id | user_id | order_amount | total_items | yment_meth | created_at | Date |
|---|---|---|---|---|---|---|---|
| 161 | 78 | 990 | 25725 | 1 | credit_card | 2017-03-12 5:56:57 | 2017-03-12 |
| 229 | 92 | 757 | 90 | 1 | debit | 2017-03-13 23:57:51 | 2017-03-13 |

**b.** In order to evaluate the data correctly such that the metric reports the right AOV that could be useful for making useful decisions we need to use a statistic that is not prone to outliers like Median. However, in this case we need to consider the quantity sold as well. Thus we can do the following:
   i. Find price of one pair of sneakers by dividing the order_amount by total_items sold in each individual orders
   ii. Find the median across this newly created metric. The average would still sway since there is a store that is selling one pair of shoes at 25k.

**c.** The median value of price per shoe across all the orders comes out to be **153**. This value could help in understanding which stores are selling the shoes at very low or higher prices.

**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?
b. What is the last name of the employee with the most orders?
c. What product was ordered the most by customers in Germany?

**Solution 2:**

**a. SQL Query:**

```
select count(*) as Number_of_Orders
from Orders A
left join Shippers B
on A.ShipperID = B.ShipperID
where ShipperName = 'Speedy Express';
```

**Output:**

Result:

Number of Records: 1

| Number_of_Orders |
| --- |
| 54 |

**The total number of orders shipped by Speedy Express is 54**

b. **SQL Query:**

```
select B.LastName
from
(select EmployeeID,count(OrderID) as Number_of_Orders
from Orders
group by EmployeeID
order by Number_of_Orders desc
limit 1) A
left join
Employees B
on A.EmployeeID = B.EmployeeID;
```

**Output:**

Result:

Number of Records: 1

| LastName |
| --- |
| Peacock |

**The last name of the employee who has the maximum number of orders (40) is Peacock**

c. **SQL Query**

```
SELECT ProductName
FROM Orders
left join Customers
on Orders.CustomerID = Customers.CustomerID
left join OrderDetails
on Orders.OrderID = OrderDetails.OrderID
left join Products
on OrderDetails.ProductID = Products.ProductID
where Country = 'Germany'
group by ProductName
order by sum(Quantity) desc
limit 1;
```

**Output:**

Result:

Number of Records: 1

| ProductName |
| --- |
| Boston Crab Meat |

**The product that was ordered the most by customers in Germany was Boston Crab Meat (this is based on the quantity ordered by the customers)**