# Fall 2021 Data Science Intern Challenge

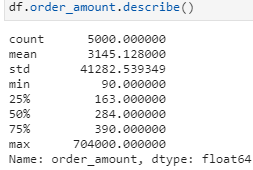
Please complete the following questions, and provide your thought process/work. You can attach your work in a text file, link, etc. on the application page. Please ensure answers are easily visible for reviewers!

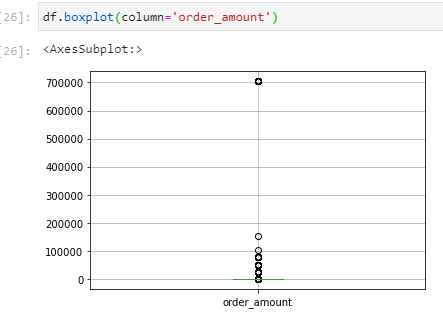
**Question 1:** Given some sample data, write a program to answer the following: [click here to access the required data set](https://docs.google.com/spreadsheets/d/16i38oonuX1y1g7C_UAmiK9GkY7cS-64DfiDMNiR41LM/edit#gid=0)

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.

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

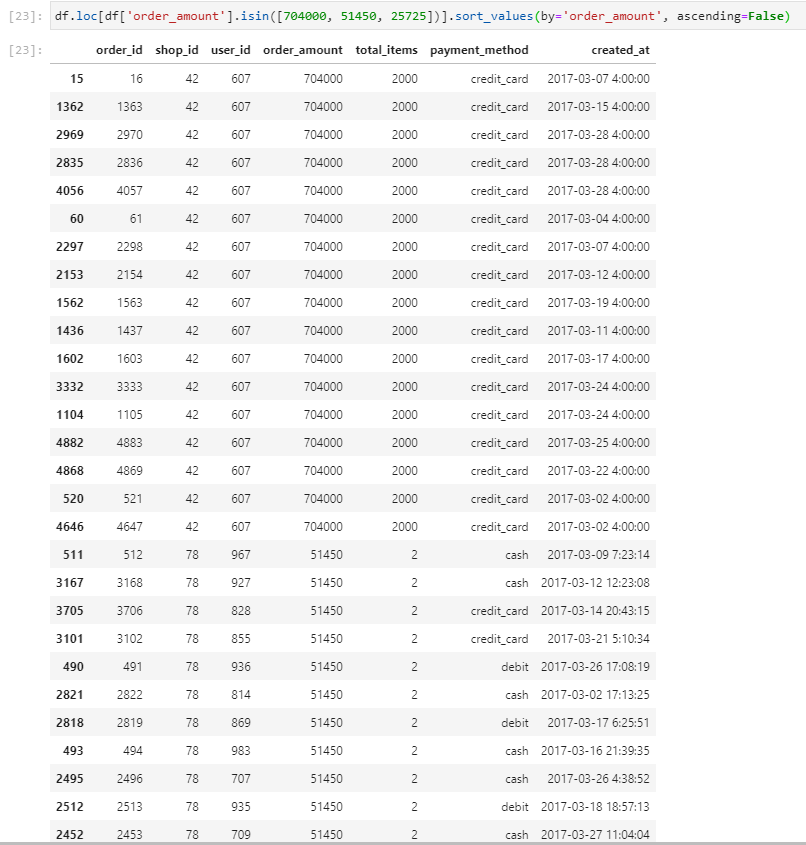
With the code df.order\_amount.describe() we can get the count, mean, min, max,

std, 25%, 50%, 75% and column names. 

The mean appears to indeed be $3145.13 however, the standard deviation is too large with a value of $41282.54. This means that, on average, values vary by $41282.54, which means that $3145.13 is not a reliable AOV. We can also see that, compared to all values presented, the max value $704,000 seems abnormally high, indicating that there may be one or more outliers. If we use df.boxplot(column='order\_amount') we see that the boxplot is compressed into a line along 0. 

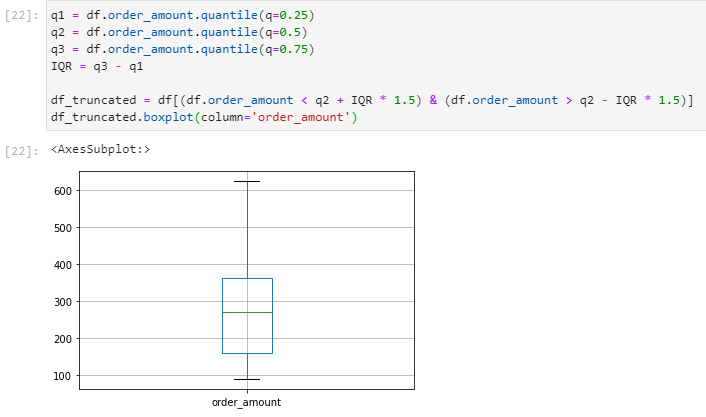
In order to identify the outliers the data needs to be grouped by order amount, count the number of entries for each group, and sort it in descending order.

This tells us that there are a few outliers. Which shows us that there are three outliers: $704000, $51450 and $25725. Using df.loc[df['order\_amount'].isin([704000, 51450, 25725])].sort\_values(by='order\_amount', ascending=False) we can investigate the data further. As each shop sells one item, the transactions $51450 and $25725 are made from the same store id, this means that the $51450 transactions are just two $25725 transactions. As $704000 transactions are all made at the same time and same amounts (2000) each time this maybe suppliers making scheduled bulk purchases.





To properly evaluate this data we could try cleaning it by displaying all values that are median +/- 1.5 times the interquartile range.

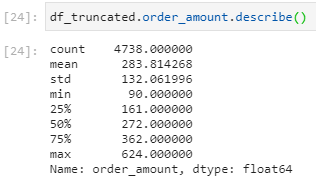


1. What metric would you report for this dataset?

I would report the median of the truncated dataset as the data seems to be skewed towards lower values and using the higher values will make the data unreliable.

1. What is its value?

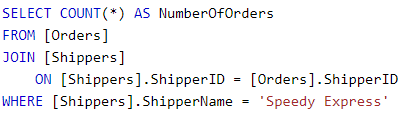
By using df\_truncated.order\_amount.describe() we get a median of $272. The std has also gone down to 132.06 which is a more trustworthy value.



**Question 2:** For this question you’ll need to use SQL. [Follow this link](https://www.w3schools.com/SQL/TRYSQL.ASP?FILENAME=TRYSQL_SELECT_ALL) 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.

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

First we need to join Orders and Shippers tables on ShipperID, then filter the table by orders performed by Speedy Express and count the entries. Using



We get a value of 54.

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

This time we need to merge orders and employees tables by employee ids then group by last names and count order amounts. Using

SELECT [Employees].LastName, COUNT(\*) AS NumberOfOrders

FROM [Orders]

JOIN [Employees]

ON [Orders].EmployeeID = [Employees].EmployeeID

GROUP BY [Employees].LastName

ORDER BY NumberOfOrders DESC

LIMIT 1

We get the last name Peacock with a max order amount of 40.

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

For this we need to join order details on order id column then sum the quantity column and group by product id. To get the product name we join the products on the product ids column.

SELECT [Products].ProductName,

SUM([OrderDetails].Quantity) AS "TotalOrdered"

FROM [Orders]

JOIN [Customers]

ON [Customers].CustomerID = [Orders].CustomerID

JOIN [OrderDetails]

ON [OrderDetails].OrderID = [Orders].OrderID

JOIN [Products]

ON [Products].ProductID = [OrderDetails].ProductID

WHERE [Customers].Country = 'Germany'

GROUP BY [OrderDetails].ProductID

ORDER BY TotalOrdered DESC -- Show most ordered item at the top.

Which tells us that Boston Crab Meat was ordered most in Germany with an order amount of 160.