# Summer 2022 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!

Submission for Alvin Yap

**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.
2. What metric would you report for this dataset?
3. What is its value?

**R Solution:**

**Environment Used:**

**Anaconda Navigator (RStudio 1.1.456)  
  
Libraries Used:  
Tidyverse, GGPlot**

**Link to RMarkdown GitHub Page:**

**Python Solution:**

**Environment Used:**

**Anaconda Navigator (Spyder Version 4.2.5) for the Python File  
Anaconda Navigator (Jupyter Notebook 6.3.0) for the ipyb File**  
**Packages Used:**

**Pandas, Seaborn, Matplotlib**

**Github Link to Python Solution, Jupyter Notebook and R Source Files:** <https://github.com/AlvinSJYap/ShopifyDataScienceInternChallenge>

**Answers:**

1. What went wrong with the AOV calculation was the usage of Mean as the metric. Mean is sensitive to outliers and it can cause the data average to skew in ways that cause errors in analysis. We can evaluate the AOV in better ways by either cleaning the data first and removing the outliers before using the mean, or we can use the median value if we do not have time to clean the data..
2. The metric I would report for this dataset would be a mean that is based on the cleaned data where the initial outliers are removed. This would give us a more accurate depiction of the AOV for our 100 sneaker shops. If data points that would normally be considered statistical outliers must be kept in, the median would be a valid metric to report for this data set.
3. The metric to calculate the AOV is the mean of the data after removing outliers: $293.72.Alternatively, if the data is to not be modified in any manner, then we will use the median of the original data set thus the AOV would be: $284.0.

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

**Answers:**

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

**Query (Using Joins):** SELECT count(\*) As OrderCount, ShipperName FROM Orders Ord inner join Shippers Ship on (Ord.ShipperID = Ship.ShipperID) group by ShipperName;  
  
**Query (Using Subqueries):**

SELECT Count(\*) as SpeedyExpressOrders FROM Orders where ShipperID = (Select ShipperID FROM Shippers where ShipperName = 'Speedy Express') ;

**Query breakdown:** Using a subquery to first select the shipperID that belongs Speedy Express to then use to select the counter of orders from the Orders table that match that ShipperID.

**Result:** 54 Orders were shipped by Speedy Express.

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

**Query (Using Joins):**SELECT TOP 1 LastName, count(LastName) as OrderCount

FROM Employees Emp inner join Orders Ord on (Emp.EmployeeID = Ord.EmployeeID) group by LastName order by count(LastName) DESC

**Query (Using Subqueries):**

Select LastName from Employees where EmployeeID =

(Select EmployeeID from

(Select TOP 1 COUNT(EmployeeID) as OrderCount, EmployeeID from Orders

group by EmployeeID

order by COUNT(EmployeeID) DESC))

**Query Breakdown**:

Get the Employee ID with the highest count in the Orders Table ->(Select EmployeeID from (Select TOP 1 COUNT(EmployeeID) as OrderCount, EmployeeID from Orders group by EmployeeID order by COUNT(EmployeeID) DESC))

Get the LastName of the the employee that matches that employeeID -> (Select EmployeeID from (Select TOP 1 COUNT(EmployeeID) as OrderCount, EmployeeID from Orders group by EmployeeID order by COUNT(EmployeeID) DESC))

**Result:**  Peacock is the last name of the employee with the most orders at 40 orders.

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

**Query (Using Joins) for largest # of Orders per Product:**

SELECT Count(\*) as OrdersFromGermany , ProductName FROM Products Prod inner join OrderDetails OrdDet on (Prod.ProductID = OrdDet.ProductID) where OrderID in

(SELECT OrderID FROM Orders where CustomerID in

(SELECT CustomerID FROM Customers where Country = "Germany")) group By ProductName Order by Count(\*) DESC;

**Query (using Subqueries) for largest Quantity of Product Ordered:**

SELECT ProductName FROM Products where ProductID =

(Select ProductID from

(Select TOP 1 ProductID, SUM(Quantity) as Quantity from OrderDetails where OrderID in

(SELECT OrderID FROM Orders where CustomerID in

(SELECT CustomerID FROM Customers where Country = "Germany")) group by ProductID order by SUM(Quantity) DESC)

)

**Query Breakdown:**

Gets me the Customer ID in germany -> (SELECT CustomerID FROM Customers where Country = "Germany")

Gets me all the orders that haver been ordered in germany by their ID-> SELECT OrderID FROM Orders where CustomerID in (SELECT CustomerID FROM Customers where Country = "Germany");

Get the ProductID with the highest summed Quantity across all orders -> Select TOP 1 ProductID, SUM(Quantity) as Quantity from OrderDetails where OrderID in

(SELECT OrderID FROM Orders where CustomerID in

(SELECT CustomerID FROM Customers where Country = "Germany")) group by ProductID order by SUM(Quantity) DESC

Get the Product Name by the Product ID -> SELECT ProductName FROM Products where ProductID =

(Select ProductID from

(Select TOP 1 ProductID, SUM(Quantity) as Quantity from OrderDetails where OrderID in

(SELECT OrderID FROM Orders where CustomerID in

(SELECT CustomerID FROM Customers where Country = "Germany")) group by ProductID order by SUM(Quantity) DESC)

**)**

**Result:** Boston Crab Meat is the Germany’s most ordered in terms of sheer amount of Product ordered at 160 Individual Crab Meats in 4 Orders. However, Gorgonzola Telino can also be the most ordered with 5 Orders total. So, the most ordered product by German Customers can be either of these two products depending on what metric we use to measure “Most Ordered” by (EG. Raw Total # of Order Units, or Raw # of individual units of product per order). By convention, the **answer** should be **Gorgonzola Telina** at 5 total orders.