Shopify Data Science Intern Challenge *(Brianna Drew)*

Question #1

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.

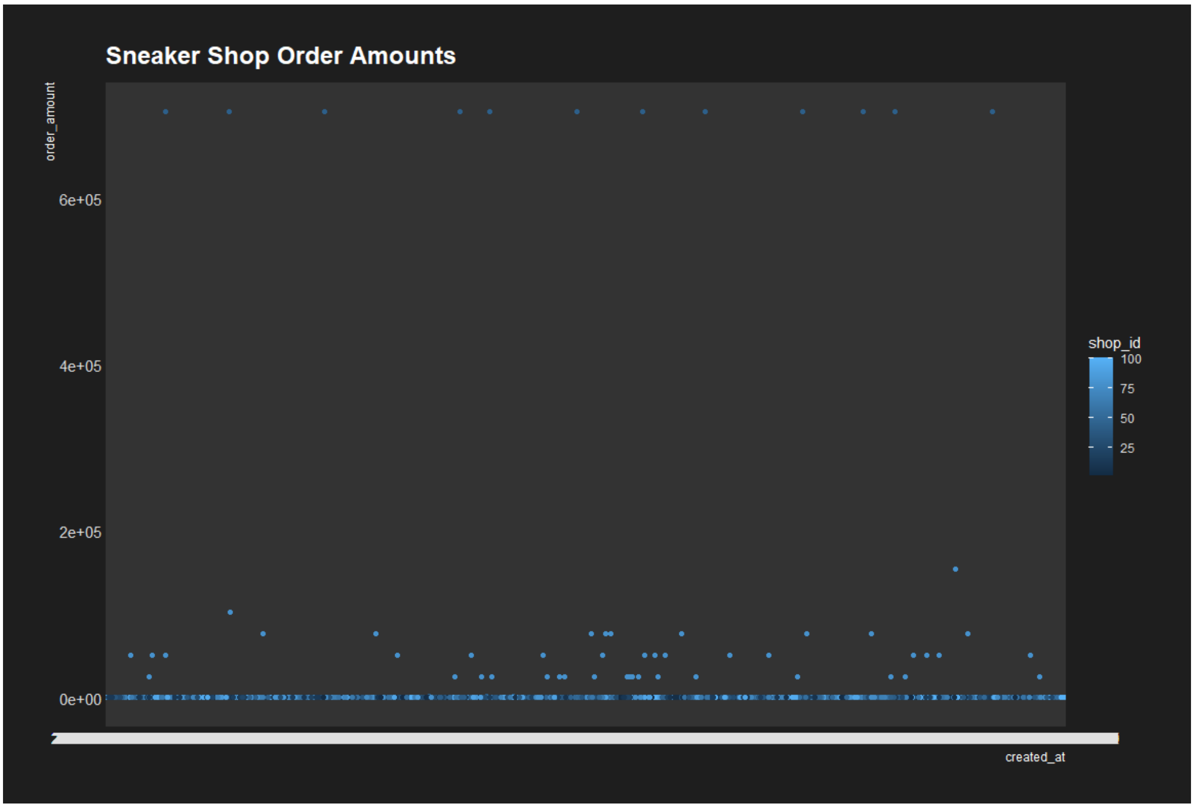
```
# Import and view CSV data file
sneaker_shops = read.csv("C:\\Users\\brian\\OneDrive\\Desktop\\Work\\Data
Science\\datasciencechallenge.csv", header = TRUE)
View(sneaker_shops)

# Calculate and print AOV
print(sum(sneaker_shops$order_amount)/nrow(sneaker_shops))
```

```
[1] 3145.128
```

As you can see, the AOV is indeed $3,145.13. But how is this possible when sneakers are relatively affordable? I will explore this further by first visualizing the data with a scatter plot, specifically looking at the total order amounts.

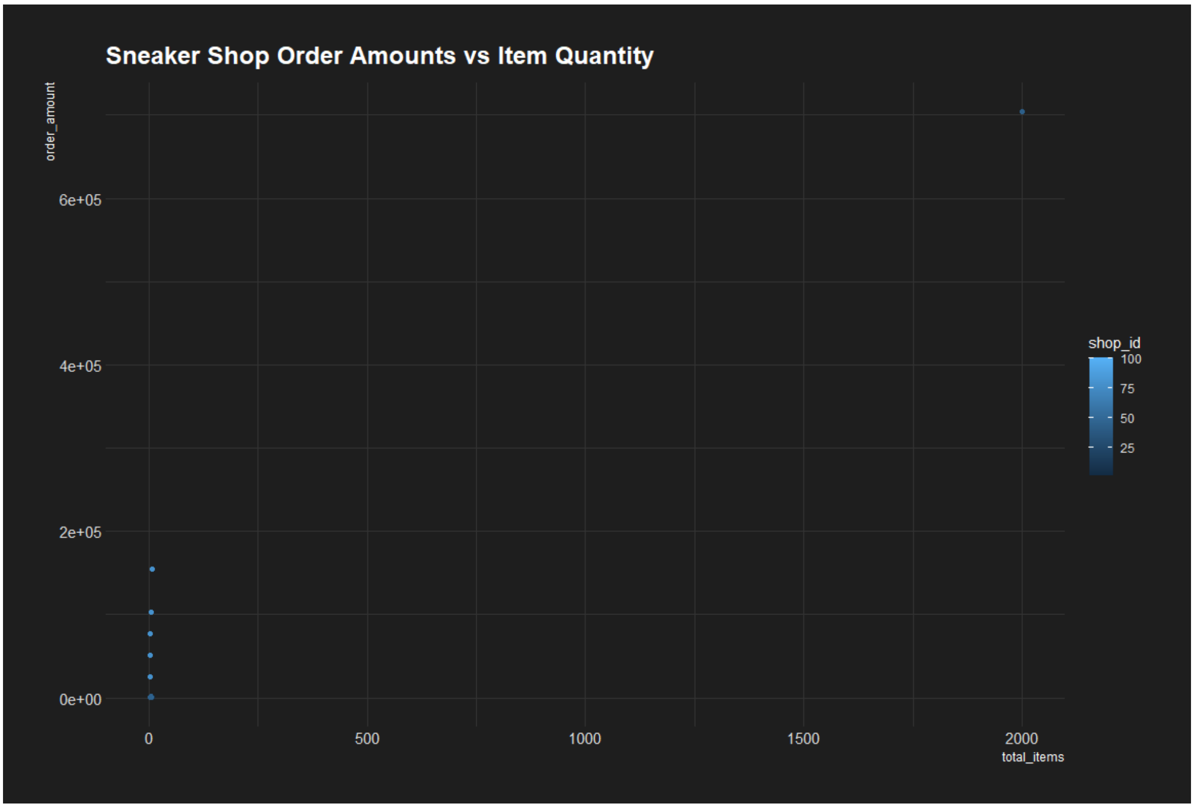
a. Think about what could be going wrong with our calculation. Think about a better way to evaluate this data.

```
# Create scatterplots
ggplot(sneaker_shops, aes(x=created_at, y=order_amount, color=shop_id)) +
geom_point() + theme_modern_rc() + ggtitle("Sneaker Shop Order Amounts")
```



As you can see from the above graph, we have have approximately 11 orders that were far pricier than the majority, indicating the presence of outliers that may be skewing the AOV. The next thing I need to figure out is whether this is due to a shop or two having much pricier sneakers, or a few orders where the quantity of items ordered were much greater. To investigate this, I created another scatter plot this time comparing the total order amounts to the total items ordered.

```
ggplot(sneaker_shops, aes(x=total_items, y=order_amount, color=shop_id)) +
geom_point() + theme_modern_rc() + ggtitle("Sneaker Shop Order Amounts vs Item
Quantity")
```



As we can see from this second graph, the orders that were around $700,000 appear to indeed be caused by a greater quantity of items ordered, but we can also see that there are a few models of shoes that seemed to be at quite a higher price point (approaching almost $150,000!). Therefore, these outliers are indeed what is most likely causing the AOV to be so high. Our options are either to handle these outliers in one way or another, or to use another metric entirely to analyze the data.

b. What metric would you report for this dataset? and c. What is its value?

The first option is to still calculate the AOV, but to handle some of the major outliers. First, I will determine the median order amount.

```
# Calculate median order amount
omedian <- median(sneaker_shops$order_amount)
print(omedian)
```

```
[1] 284
```

Then, I will replace all order amounts that are above a given threshold with the median. Let's set this threshold at 100,000. Some outliers will still be present, however, this should eliminate the ones that are affecting the AOV the most while still retaining some accuracy.

```
# Replace order amounts greater that $100,000 with the median order amount
sneaker_shops$order_amount[ sneaker_shops$order_amount>100000 ] <- omedian
```

Now, we can calculate the AOV again, this time with the major outliers handled.

```
# Calculate and print new AOV (with outliers handled)
print(sum(sneaker_shops$order_amount)/nrow(sneaker_shops))
```

```
[1] 701.1572
```

As you can see, now the AOV seems much more reasonable at $701.16. You could also set the threshold even lower (e.g. 5,000 as follows) and perhaps get an even more reasonable AOV.

```
# Replace order amounts greater that $100,000 with the median order amount
sneaker_shops$order_amount[ sneaker_shops$order_amount>5000 ] <- omedian

# Calculate and print new AOV (with outliers handled)
print(sum(sneaker_shops$order_amount)/nrow(sneaker_shops))
```

```
[1] 302.3464
```

Therefore, excluding orders above $5,000.00 would result in an AOV of $302.35.

Another option if you do not want to alter the existing data at all is to try to approximate the average cost per pair of sneakers, which would at least handle the outliers that are due to large quantities of items ordered in an order. We would accomplish this by adding a new column to the dataset containing the order amount divided my the total items per order.

```
# Replicate data set
sneaker_shops_new <- sneaker_shops

# Add new column to data set containing total order amount divided by item
quantity
sneaker_shops_new$per_item <- sneaker_shops_new$order_amount /
sneaker_shops_new$total_items
```

Then, simply calculate the mean of this new column.

```
# Calculate mean approximate item price
mean(sneaker_shops_new$per_item)
```

> [1] 387.7428

Therefore, the mean price per item would be $387.74.

Question #2

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

```
SELECT COUNT(*)
FROM Orders
INNER JOIN Shippers ON Orders.ShipperID = Shippers.ShipperID
WHERE ShipperName = "Speedy Express"
```

Results: There were 54 orders in total shipped by Speedy Express.

COUNT(*)
54

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

```
SELECT LastName, MAX(OrderCount)
FROM (SELECT LastName, COUNT(*) as OrderCount
      FROM Orders
      INNER JOIN Employees ON Orders.EmployeeID = Employees.EmployeeID
      GROUP BY Orders.EmployeeID
      ORDER BY COUNT(*) DESC)
```

Results: The last name of the employee with the most orders is Peacock.

LastName	MAX(OrderCount)
Peacock	40

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

```sql
SELECT ProductName, MAX(QTY)
FROM (
      SELECT Products.ProductName as ProductName, SUM(OrderDetails.Quantity) as QTY
      FROM Orders
      INNER JOIN Customers ON Orders.CustomerID=Customers.CustomerID
      INNER JOIN OrderDetails ON Orders.OrderID=OrderDetails.OrderID
      INNER JOIN Products ON Products.ProductID=OrderDetails.ProductID
      WHERE Country = "Germany"
      GROUP BY Products.ProductName
      ORDER BY SUM(OrderDetails.Quantity) DESC)
```

Results: The product that was ordered the most by customers in Germany is Boston Crab Meat.

ProductName	MAX(QTY)
Boston Crab Meat	160