John Purcell

Question 1: Given some sample data, write a program to answer the following:

> *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.

Explanation:

The AOV calculation listed above is correct, as shown in the corresponding python program. So, we can rule out that it is not a technical error.

```
# Checking the AOV
AvgOrderValue = mean(df.order_amount)
print("Average Order Value")
print("--------------------")
print(AvgOrderValue)
```

```
Average Order Value
--------------------
3145.128
```

After sorting the data descending by order_value, we can see a better picture of what is taking place within the data.

```
# By sorting the list reversed we can get a look at the upper results of the sales data
df.sort_values(by="order_amount", ascending=False, inplace=True)
# Create a list and show the top results
OrderAmounts = list(df.order_amount)
print("List of 50 Highest Order Amounts")
print("--------------------------------")
count = 0
while count < len(OrderAmounts[:50]):
    print(OrderAmounts[count], OrderAmounts[count +1 ], OrderAmounts[count +2],
          OrderAmounts[count + 3], OrderAmounts[count + 4])
    count += 5
```

In this case, there are many orders that have an exceptionally high order_value. There is a gap from roughly 1000 dollars in value, to orders with roughly 25,000 dollars in value. Potentially these are corporate accounts ordering shoes for an event or activation. They do not seem to be representative of the “average” shoe store consumer / order.

```
List of 50 Highest Order Amounts
--------------------------------
704000 704000 704000 704000 704000
704000 704000 704000 704000 704000
704000 704000 704000 704000 704000
704000 704000 154350 102900 77175
77175 77175 77175 77175 77175
77175 77175 77175 51450 51450
51450 51450 51450 51450 51450
51450 51450 51450 51450 51450
51450 51450 51450 51450 25725
25725 25725 25725 25725 25725
```

b. What metric would you report for this dataset?

Explanation:

While the mean value of orders offers some insights into all orders at the upper and lower end of the distribution, adding metrics that investigate the most common order values, or an “approximate” average value.

To do this, we will look at mode (the most common order value) and median (the order value in the middle of a sorted list of all orders).

In this case, the order values are repetitive whole numbers, but, if we were looking at a dataset with more precise decimal values (i.e 121.87 vs. 122. 19) we would opt to round these numbers based on some criteria best suited for the dataset. As mentioned, we will forgo this process for this dataset.

Reference: https://www.shopify.ca/blog/average-order-value

c. What is its value?

Explanation:

```
# Part C: Calculate these values (Median and Mode)

ModeOrderValue = mode(OrderAmounts)
MedianOrderValue = median(OrderAmounts)
```

```
Mode Order Value
--------------------
153

Median Order Value
--------------------
284.0
```

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?

Explanation:
Using a simple CTE, we find the ID corresponding to Speedy Express.
Then, a query is generated to count the distinct instances of orders where the shipper was Speedy Express. The distinct key word is used as a safeguard if there are any accidental duplicate entries.

Query:

```
WITH SpeedyID AS (SELECT ShipperID FROM Shippers WHERE ShipperName="Speedy Express")

SELECT DISTINCT COUNT(*)
FROM Orders, SpeedyID
WHERE Orders.ShipperID = SpeedyID.ShipperID;
```

Result:
54

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

Explanation:
First, a CTE titled "Maximum Value" is done to return the highest order amount and the corresponding EmployeeID. By ordering by the "order_count" var descendingly, we get the highest value at the top of the list. With a limit of 1, we get the highest value.

Next we do a simple query to get the LastName from the Employees table where the corresponding ID is equal to our CTE result EmployeeID.

My first rendition used sub-queries, but after being displeased with the readability, I switched to CTE.

Query:

```
WITH MaximumValue AS (SELECT EmployeeID, COUNT(*) AS order_count
        FROM Orders
        GROUP BY EmployeeID
        ORDER BY order_count DESC
        LIMIT 1)

SELECT LastName
FROM MaximumValue, Employees
WHERE MaximumValue.EmployeeID = Employees.EmployeeID;
```

Result:
Peacock

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

Explanation:

Query:

```
WITH GermanyTopProduct AS (SELECT *, SUM(Quantity) as Totals FROM OrderDetails
        JOIN Orders on OrderDetails.OrderID = Orders.OrderID
        JOIN Customers ON Customers.CustomerID = Orders.CustomerID
        JOIN Products ON OrderDetails.ProductID = Products.ProductID
        WHERE Country = "Germany"
        GROUP BY Products.ProductID
        ORDER BY Totals DESC
        LIMIT 1
)

SELECT ProductName AS GermanyTopProduct FROM GermanyTopProduct;
```

Result:

Boston Crab Meat