

# Shopify

## Fall 2021 Data Science Intern Challenge

Víctor Manuel Méndez Fonseca | vmendezfonseca@gmail.com  
May 7<sup>th</sup>, 2021

### Problem description

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

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.

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

**What metric would you report for this dataset?**

**What is its value?**

**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.

**How many orders were shipped by Speedy Express in total?**

**What is the last name of the employee with the most orders?**

**What product was ordered the most by customers in Germany?**

## Answer to question 1

In order to study the data the following Python code has been created.  
The development environment used was:

- VS Code 1.56 with ms-python.python extension v2021.4.765268190
- Python 3.9.1
- pandas library v1.2.3 for data analysis

---

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

#Importing data and performing initial analysis
data = pd.read_csv("2019 Winter Data Science Intern Challenge Data Set - Sheet1.csv")
print(data.order_amount.describe())
data.boxplot(column = "order_amount")
plt.title("Dataset boxplot",fontsize=18)
plt.text(300, 87, "Data contains outliers",fontsize=12 )
plt.ylabel("Order values",fontsize=14)
plt.show()

# Visualizing 90 most frequent order values
value_index = data.order_amount.value_counts()
value_index = value_index[0:90]
median = data.order_amount.median()
plt.stem(value_index.index,value_index.values)
plt.axvline(median,color="r", linestyle="--")
plt.xticks(np.arange(min(value_index.index), max(value_index.index)+1, 25.0))
plt.text(300, 87, "Median : " + str(median),fontsize=14 )
plt.title("Occurence of 90 most frequent order values",fontsize=18)
plt.xlabel("Order value",fontsize=14)
plt.ylabel("Occurence of orders",fontsize=14)
plt.grid(axis='y')
plt.show()

# Dictionary with total dollar amount per shop
shop_order_dict = {}

for i in range (data.order_amount.size):
    if data.shop_id[i] in shop_order_dict.keys():
        shop_order_dict[data.shop_id[i]] += data.order_amount[i]
    else:
        shop_order_dict[data.shop_id[i]] = data.order_amount[i]

#Visualizing total order dollar amount per shop
order_volume = list(shop_order_dict.values())
plt.stem(shop_order_dict.keys(),np.log(order_volume))
plt.title("Shops by dollar amount",fontsize=18)
plt.xlabel("Shop ID",fontsize=14)
plt.ylabel("Sales volume (in logarithmic scale)",fontsize=14)
plt.grid(axis='y')
plt.xticks(np.arange(min(shop_order_dict.keys()+1, max(shop_order_dict.keys()+1, 2.0))
```

```
plt.show()

#Determining the impact on the dataset of orders above $2000
expensive_orders = data.order_amount[data.order_amount >= 2000]
print("Dollar value ratio of orders above $2000 with respect to all orders: " +
      str(np.sum(expensive_orders)/np.sum(data.order_amount)*100))
print("Ratio of number of orders above $2000 with respect to all orders: " +
      str(np.size(expensive_orders)/np.size(data.order_amount)*100))
```

---

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

The `pandas.DataFrame.describe()` command computed on `order_amount` column allows us to see the interesting information displayed on Table 1 about the orders dataset.

Table 1: `pandas.DataFrame.describe()` result

Parameter:	Value:
count	5000.0
mean	3145.13
std	41282.539
min	90.0
25%	163.0
50%	284.0
75%	390.0
max	704000.0

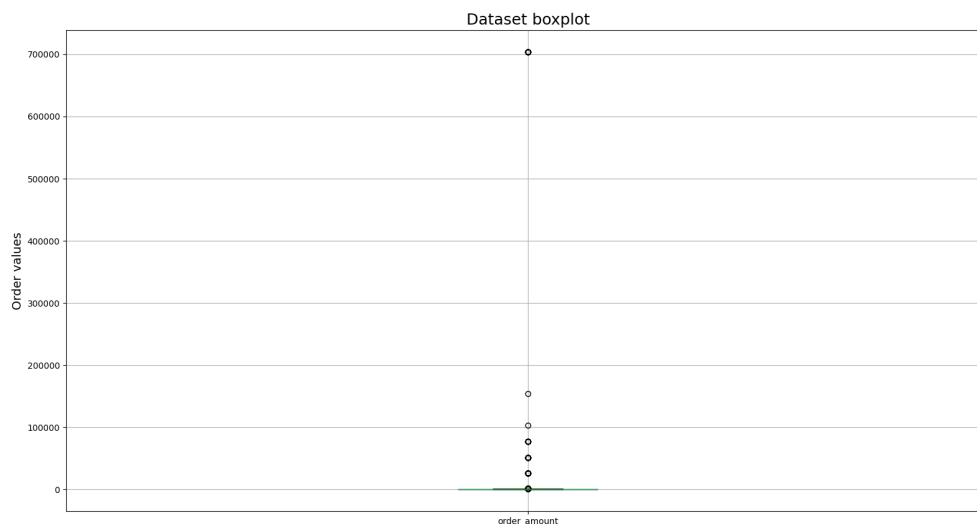


Figure 1: Data visualization with Boxplot

It can be seen that the average order value (AOV) calculation was based on the average of the `order_amount` column. However, if we observe picture 1 it can be noticed that `order_amount` dataset is very spread. This can also be observed on the large standard deviation (std) in Table 1. As a consequence, using the average to accurately describe the AOV is not correct.

### What metric would you report for this dataset?

Better than using the average, but in this case not practical, is to evaluate AOV as the Mode or most frequent order value, since this represents what customers most frequently spend. Typically, performing several business strategies to increase customers most frequently order value increases the bulk of the revenue for a business. However, looking at Fig.2 we realize that there are many Modes of somewhat similar value. As a consequence, in this case, the optimum parameter to focus on would be Median or the middle value of all orders.

Notice Fig.2 only shows the occurrence of the 90 most frequent order values for ease of visualization and the Median value has been marked with a red dashed line.

It is remarkable that orders with a value above \$2000 represent 1.26% of all orders and yet they contain about 90.5% of the total order amount in dollars for all shops. The existence of these few high dollar orders explains why the average value for all orders has moved up and why the standard deviation is so large. Furthermore, these orders belong only to shops 42 and 78, as can be seen on Fig.3. These shops do not have the typical sales volume range of the other shops. Perhaps shops 42 and 78 have a different market segment, for instance wholesale. Unfortunately there is not enough information

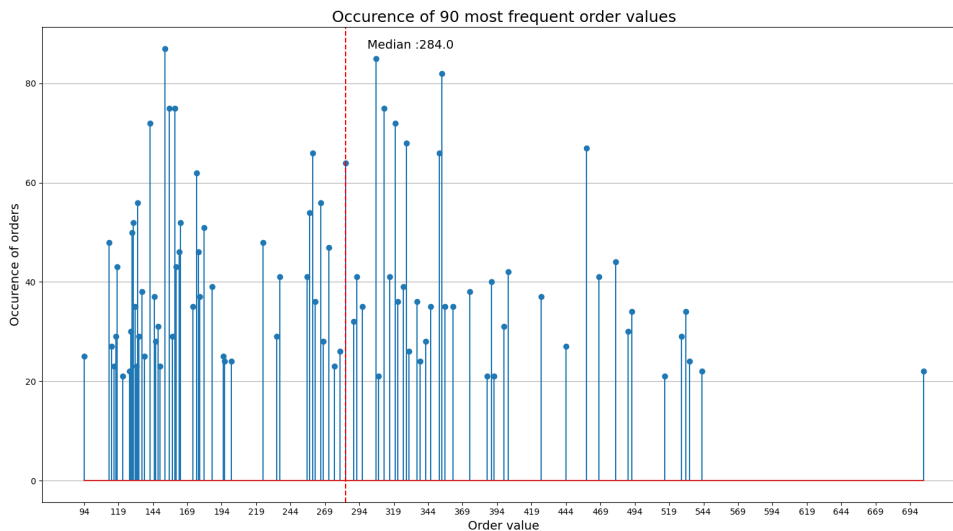


Figure 2: 90 Most Frequent Order Values

about these shops to draw any conclusions.

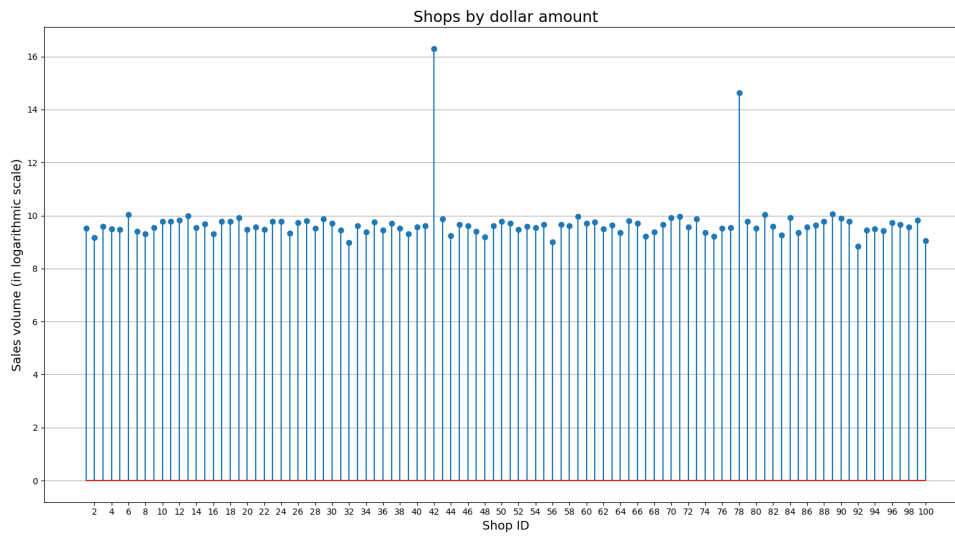


Figure 3: Sales volume by shop, using logarithmic scale for ease of visualization

**What is its value?**

The value of the Median is 284.0

## Answer to question 2

How many orders were shipped by Speedy Express in total?

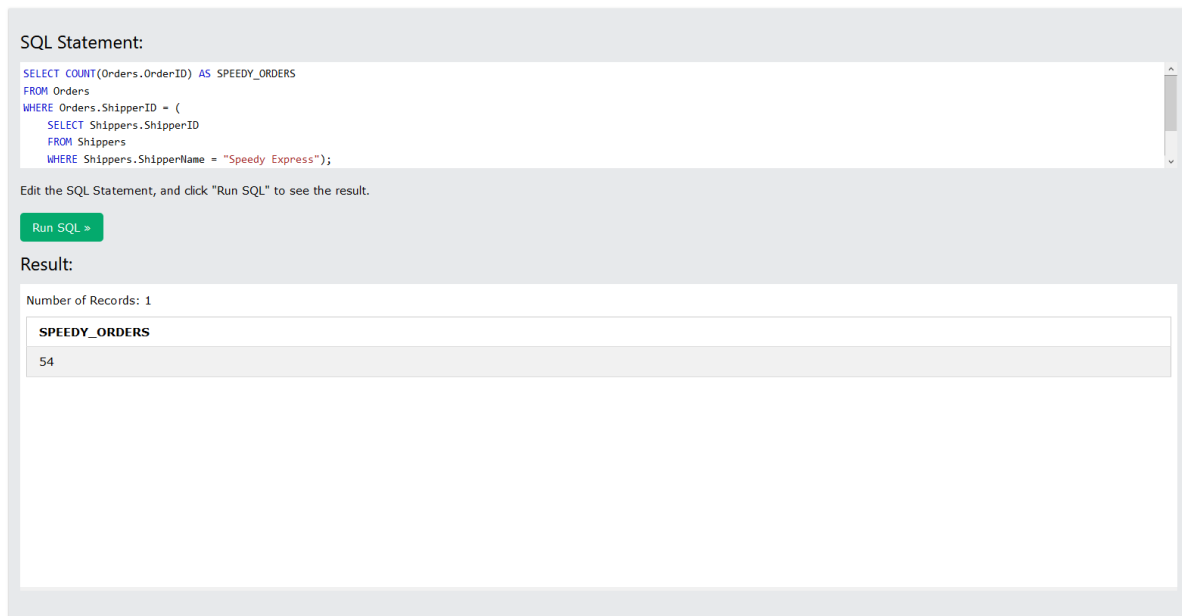
Explanation of solution: We are counting the orders such that ShipperID is Speedy Express

---

```
SELECT COUNT(Orders.OrderID) AS SPEEDY_ORDERS
FROM Orders
WHERE Orders.ShipperID = (
    SELECT Shippers.ShipperID
    FROM Shippers
    WHERE Shippers.ShipperName = "Speedy Express");
```

---

Answer : 54



The screenshot shows a SQL query execution interface. At the top, the SQL statement is displayed in a text area. Below it, a green button labeled "Run SQL" is visible. Under the button, the result is shown as a table with one row and one column.

SQL Statement:

```
SELECT COUNT(Orders.OrderID) AS SPEEDY_ORDERS
FROM Orders
WHERE Orders.ShipperID = (
    SELECT Shippers.ShipperID
    FROM Shippers
    WHERE Shippers.ShipperName = "Speedy Express");
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL

Result:

Number of Records: 1

SPEEDY_ORDERS
54

Figure 4: Question 2 answer 1

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

Explanation of solution: We are selecting the EmployeeID with more entries in Orders table and then finding the LastName of that EmployeeID

---

```
SELECT Employees.LastName
FROM Employees
WHERE Employees.EmployeeID = (
    SELECT Orders.EmployeeID
    FROM Orders
    GROUP BY Orders.EmployeeID
    ORDER BY COUNT(Orders.OrderID) DESC
    LIMIT 1);
```

```

HAVING COUNT(Orders.EmployeeID) = (
    SELECT MAX(MOST_FREQUENT)
    FROM(
        SELECT Orders.EmployeeID, COUNT(Orders.EmployeeID) AS MOST_FREQUENT
        FROM Orders
        GROUP BY EmployeeID
    )
)
);

```

---

Answer : Peacock

The screenshot shows a SQL query execution interface. The 'SQL Statement:' section contains the following query:

```

SELECT Employees.LastName
FROM Employees
WHERE Employees.EmployeeID = (
    SELECT Orders.EmployeeID
    FROM Orders
    GROUP BY Orders.EmployeeID
)

```

Below the query, there is a button labeled 'Run SQL >'. The 'Result:' section shows 'Number of Records: 1' and a table with one row:

LastName
Peacock

Figure 5: Question 2 answer 2

**What product was ordered the most by customers in Germany?**

Explanation of solution:

- Find CustomerID of customers whose country is Germany
- Find OrderID by those CustomerID in Orders table
- Find ProductID by those OrderID in OrderDetails table
- Once you have the list of all products ordered by customers in Germany, compute the quantity by type of ProductID and select the ProductID with higher quantity
- Finally, find the ProductName of that ProductID in the Products table

---

```

SELECT TOP 1 Products.ProductName ,Customers.Country, SUM(OrderDetails.Quantity) AS TOTAL
FROM (((
    Customers INNER JOIN Orders
    ON Customers.CustomerID = Orders.CustomerID )
    INNER JOIN OrderDetails

```

```
        ON Orders.OrderID = OrderDetails.OrderID )
        INNER JOIN Products
        ON OrderDetails.ProductID = Products.ProductID)
WHERE Customers.Country = "Germany"
GROUP BY Products.ProductName ,Customers.Country
ORDER BY SUM(OrderDetails.Quantity) DESC;
```

---

Answer: Boston Crab Meat

SQL Statement:

```
SELECT TOP 1 Products.ProductName ,Customers.Country, SUM(OrderDetails.Quantity) AS TOTAL
FROM (((
    Customers INNER JOIN Orders
    ON Customers.CustomerID = Orders.CustomerID )
    INNER JOIN OrderDetails
    ON Orders.OrderID = OrderDetails.OrderID )
```

Edit the SQL Statement, and click "Run SQL" to see the result.

Run SQL »

Result:

Number of Records: 1

ProductName	Country	TOTAL
Boston Crab Meat	Germany	160

Figure 6: Question 2 answer 3