

**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?

**Answer:** For this question, I used Matlab.

The naively calculated AOV is \$3145.13, and it's got by simply just averaging all the order\_amount. This means that there may be more than one items in one order, or some items may have much higher prices than others.

**% This is the AOV = \$3145.13 from given analysis**

**AOV = mean(order\_amount);**

Name ▲	Value
 AOV	3.1451e+03

So it seems more reasonable to calculate the mean value of each item, dividing the order\_amount by total\_items.

**% A better way to do this:**

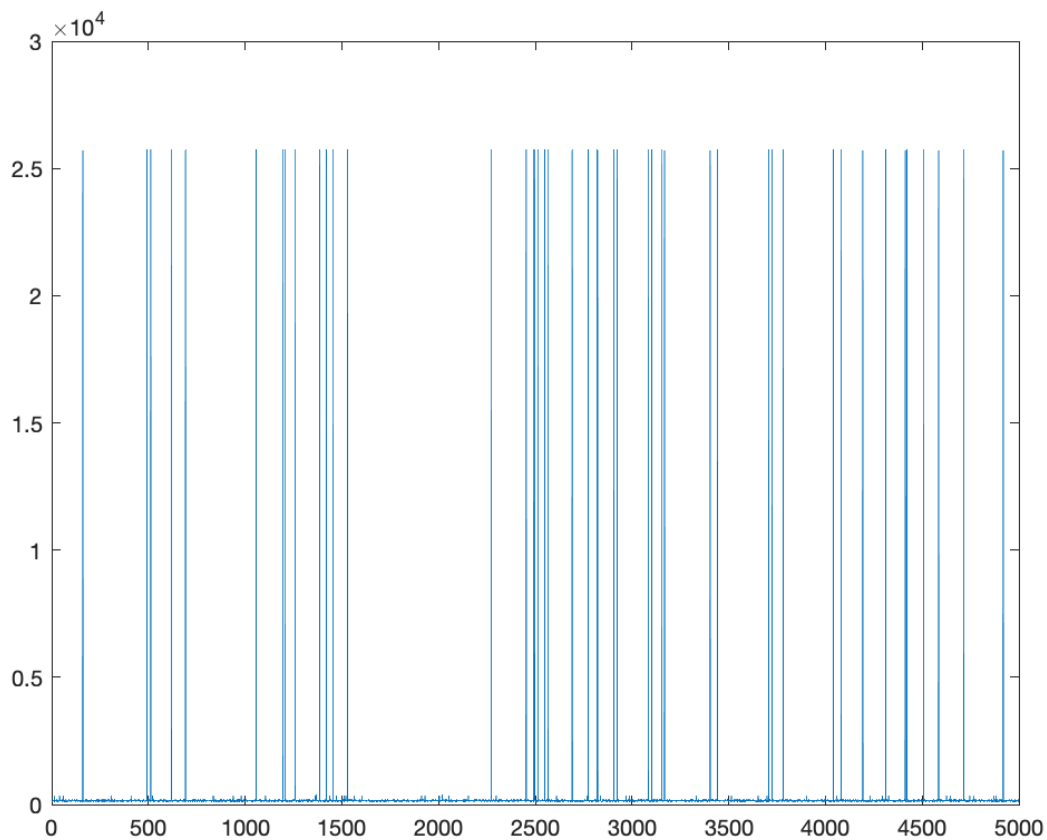
**item\_price = zeros(5000,1);**

```
for i = 1:5000
    item_price(i) = order_amount(i)/total_items(i);
end
```

**AOV\_new = mean(item\_price); % AOV\_new = 387.7;**

 AOV_new	387.7428
---	----------

Here we got the value of \$387.7, which is still a bit expensive for sneaker, in order to figure out what's going on, I plotted the price for each item. We can see that most item prices are very low (under \$200), several prices reach more than \$25,000.



So for next step, I filtered out those high prices by setting \$500 threshold, and calculated a new average item price, which is \$152.47. Also, the average value for high price sneaker is \$25725.

```
% revise a little bit
plot(item_price);
new_item_price = [];
expensive_item_price = [];
for i = 1:5000
    % for each item price, if it's more than $500 then it will be
    % considerate as expensive item
    item = order_amount(i)/total_items(i);
    if item < 500
        new_item_price(end+1) = item;
    else
        expensive_item_price(end+1) = item;
    end
end

AOV_revised = mean(new_item_price);
AOV_expensive = mean(expensive_item_price);
AOV_expensive    25725
AOV_new          387.7428
AOV_revised      152.4756
```

Finally, I want to see who is selling these expensive sneakers, so I listed the shop\_id of those shops, and it turns out its only shop\_id 78!

```
%extra analysis: how many stores are selling expensive sneakers?
shop = [];
for i = 1:5000
    % put the shop_id of shops which are selling expensive sneakers into
    % array "shop"
    item = order_amount(i)/total_items(i);
    if item > 500
        shop_counted = ismember(shop_id(i),shop);
        if shop_counted == 0
            shop(end+1) = shop_id(i);
        end
    end
end
shop
```

78

**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? ----54
- What is the last name of the employee with the most orders?----Peacock
- What product was ordered the most by customers in Germany?----Boston Crab Meat

a)

SQL Statement:

```
SELECT COUNT (*) AS order_numbers
FROM Orders
JOIN Shippers
    ON Shippers.ShipperID = Orders.ShipperID
WHERE Shippers.ShipperName = 'Speedy Express'
```

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

Run SQL »

Result:

Number of Records: 1

order\_numbers

54

b)

#### SQL Statement:

```
SELECT Employees.LastName, COUNT (*) AS order_number
FROM Orders JOIN Employees
ON Orders.EmployeeID = Employees.EmployeeID
GROUP BY Employees.LastName
ORDER BY order_number DESC
LIMIT 1
```

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

Run SQL »

#### Result:

Number of Records: 1

LastName	order_number
Peacock	40

c)

#### SQL Statement:

```
SELECT Products.ProductName, SUM(Quantity) AS Total_Quantity
FROM Orders, OrderDetails, Customers, Products
WHERE Customers.Country = "Germany" AND Customers.CustomerID = Orders.CustomerID AND OrderDetails.OrderID = Orders.OrderID AND
OrderDetails.ProductID = Products.ProductID
GROUP BY Products.ProductID
ORDER BY Total_Quantity DESC
```

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

Run SQL »

#### Result:

Number of Records: 1

ProductName	Total_Quantity
Boston Crab Meat	160