Data Science Intern Challenge

January 9, 2022

0.1 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.

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

Short Answer: There are some outlier data in the data set which makes the** mean **misleading.

b. What metric would you report for this dataset?

Short Answer: Median

c. What is its value?

Short Answer: 284

Answer 1 Process/Work:

This part is elaborating more on the process of getting the results: *What is more Probable is the existence of outlier data. These data must be very large and make the distribution of the Order Value right-skewed. So let us explore the data:*

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import altair as alt

%matplotlib inline
```

[2]:
```
data = pd.read_csv("Sneakers.csv", usecols=[1, 2, 3, 4, 5, 6])
```

[3]:
```
data.info()
```

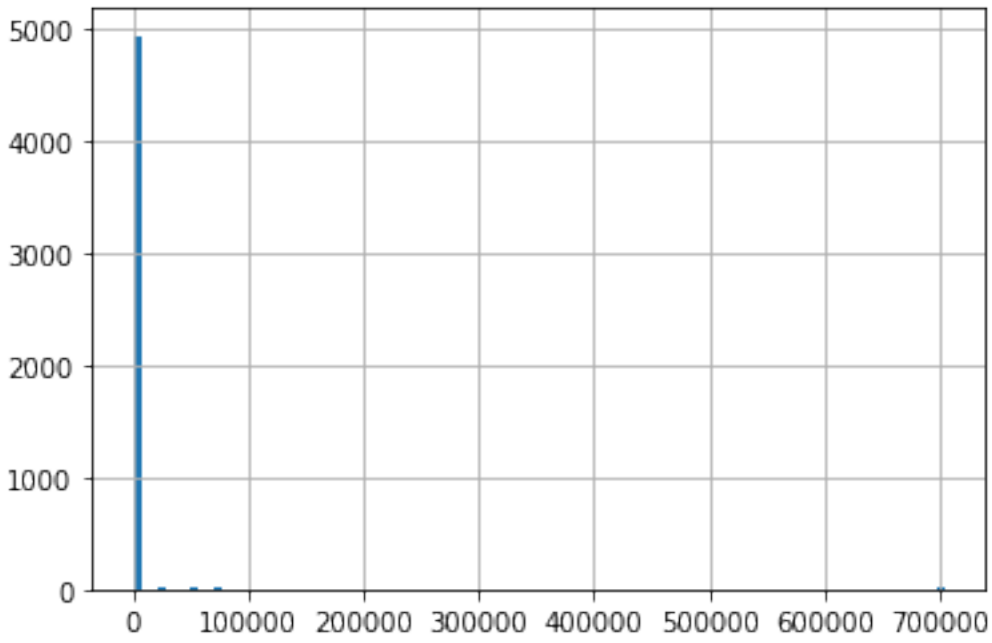
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   shop_id         5000 non-null   int64
 1   user_id         5000 non-null   int64
 2   order_amount    5000 non-null   int64
 3   total_items     5000 non-null   int64
 4   payment_method  5000 non-null   object
 5   created_at      5000 non-null   object
dtypes: int64(4), object(2)
memory usage: 234.5+ KB
```

[4]:
```
data.order_amount.describe()
```

[4]:
```
count      5000.000000
mean       3145.128000
std       41282.539349
min          90.000000
25%         163.000000
50%         284.000000
75%         390.000000
max      704000.000000
Name: order_amount, dtype: float64
```

[5]:
```
data['order_amount'].hist(bins=100,)
```

[5]:
```
<AxesSubplot:>
```



As it is expected, the reason for the high average of orders value is because of outlier data which makes the data distribution right-skewed. In other words, the range of data is between 90 and 704000, while 75% of data is less than 390. Therefore, in this dataset, "Mean" is not a good metric to describe data, and "Median" should be selected as the metric.

But before selecting median as our metric, let us plot the data and check whether our data can be categorized in different groups based on payment methods or not. For instance, the large number of payments may be by credit card, and therefore, for reporting the metrics, we can mention this point. Going to the details, here is the box plot of data, based on payment method:

[6]:
```
alt.Chart(data).mark_boxplot(size=50, extent=0.5).encode(
    y=alt.Y("payment_method", title="Payment Method"),
    x=alt.X(
        "order_amount",
        title="Order Amounts",
    ),
).properties(width=300)
```

[6]:
```
alt.Chart(…)
```

Interesting! The box plot contains only circle marks instead of the box! Since most data is under 400. Those points are outliers. let us try dropping data larger than 700 and explore the data again to find whether there is a difference in the distribution of data based on payment method:_

[7]:
```
data_new = data[data["order_amount"] < 750]
```

[8]:
```
alt.Chart(data_new).mark_boxplot(size=50, extent=0.5).encode(
    y=alt.Y("payment_method", title="Payment Method"),
    x=alt.X(
        "order_amount",
        title="Order Amounts",
    ),
).properties(width=300, height=250)
```

[8]: alt.Chart(…)

Although It is not statistically accurate and we should use hypothesis testing to prove whether these distribution metrics are different among payment methods or not, for simplicity by observing the box plots, we can see no significant difference among payment methods, so we can report Median as the metric

[9]:
```
print(f"median= { data['order_amount'].median()}")
```

```
median= 284.0
```

0.2 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?

Short Answer : 54 SQL code:

```
SELECT
    count(*)
FROM
    Orders,
    Shippers
WHERE
    Orders.ShipperID = Shippers.ShipperID
    AND
    ShipperName= "Speedy Express"
;
```

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

Short Answer : "Peacock" SQL code:

```
WITH temp AS(
    SELECT
        count(orderID) as order_count,
        EmployeeID
    FROM
```

```
        Orders
    GROUP BY
        EmployeeID)

SELECT
    LastName
FROM
    temp
JOIN
    Employees
ON
    temp.employeeID = Employees.EmployeeID
WHERE
    temp.order_count=(SELECT max(order_count) FROM temp)
;
```

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

Short Answer : "Boston Crab Meat" (the quantity is 160 units) SQL code:

```
WITH temp AS(
  SELECT
      ProductName,
      ttl_qty
  FROM
      (SELECT
          SUM(Quantity) as ttl_qty,
          ProductName,
          OrderDetails.ProductID
      FROM
          OrderDetails
      JOIN
          Products
      ON
          OrderDetails.ProductID=Products.ProductID
      WHERE
          OrderID in (SELECT OrderID FROM (
                                          SELECT
                                              Customers.*,
                                              Orders.OrderID
                                          FROM
                                              Orders
                                          JOIN
                                              Customers
                                          ON
                                              Customers.CustomerID=Orders.CustomerID
                                          WHERE
                                              customers.Country = 'Germany'))
```

```
      Group BY
          OrderDetails.ProductID,ProductName) as t)

SELECT
    ProductName
FROM
    temp
WHERE
    ttl_qty = (SELECT MAX(ttl_qty) FROM temp)
;
```