Challenger Notebook

January 19, 2022

Story of Sneaker Shoes at Shopify

image source - istockphoto

1 Summary of Analysis Results

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

After analyzing the data, I found that the data distribution is right-skewed. Due to the presence of some large order values, data distribution is no longer normal. This results in a significant difference in mean, median, and mode values. In addition, the mean value is not a true representative of the central tendency of data. Therefore, the metric of average order value (AOV) is misleading as it uses the mean value to measure the average dollar amount spent each time a customer places an order.

2. What metric would you report for this dataset?

I would use the median metric for this dataset.

3. What is its value?

284.0

2 My thought process during this investigation

2.0.1 I followed the below steps programmatically to analyze the data.

1. Reading data from the given CSV file and calculating the preliminary AOV value.
2. Understanding the data structure by observing features (columns) and corresponding data types.
3. Evaluating if the data has some missing or invalid data points.
4. Understanding the quartiles of data distribution and observing mean, median, minimum, and maximum values.
5. Visualizing the data distribution using histogram and violin plots.
6. As I observe that the data distribution is right-skewed from step 3, I performed a log transformation of the orders data for better visualization of data distribution and the values of mean, median, and mode.
7. After performing step 4, it is clearly evident that the median is a better metric than the mean (AOV) value. Therefore, I decide to use the median metric for the given dataset and compute its value.

3 Investigation process using programming

[1]:
```
# Importing required libraries and packages

# For basic operations
import pandas as pd
import numpy as np
from pandas import plotting
from scipy import stats

# For visualizations
import matplotlib.pyplot as plt
import seaborn as sbn

# To avoid warnings while plottting results
import warnings
warnings.filterwarnings('ignore')
```

4 Step 1: Reading data and calculating preliminary AOV value

[2]:
```
# Reading data from the provided CSV file
DF_data = pd.read_csv("./2019 Winter Data Science Intern Challenge Data Set -␣
 ↪Sheet1.csv")

# Calculating the average order value (AOV) naively
print("\n********************* AOV value *********************\n",␣
 ↪round(sum(DF_data["order_amount"])/len(DF_data["order_amount"]),2))
```

```
******************** AOV value ********************
 3145.13
```

5 Step 2: Understanding features and corresponding data types

[3]:
```
# Printing first 5 rows of the given dataset
print("\n******************** Data Structure ********************\n",DF_data.
 ↪head())

# Printing data types of columns
print("\n******************** Data Types ********************\n")
DF_data.dtypes
```

```
******************** Data Structure ********************
    order_id  shop_id  user_id  order_amount  total_items payment_method  \
0         1       53      746           224            2           cash
1         2       92      925            90            1           cash
2         3       44      861           144            1           cash
3         4       18      935           156            1    credit_card
4         5       18      883           156            1    credit_card

            created_at
0  2017-03-13 12:36:56
1  2017-03-03 17:38:52
2   2017-03-14 4:23:56
3  2017-03-26 12:43:37
4   2017-03-01 4:35:11

******************** Data Types ********************
```

[3]:
```
order_id           int64
shop_id            int64
user_id            int64
order_amount       int64
total_items        int64
payment_method    object
created_at        object
dtype: object
```

6 Step 3: Evaluating if the data has some missing or invalid data points

[4]:
```
# Checking for NaN values
print("******************** Does the data has any NaN values?␣
 ↪********************\n", DF_data.isnull().any().any())
```

```
******************** Does the data has any NaN values? ********************
 False
```

7 Step 4: Understanding the quartiles of data distribution and other metrics

[5]:
```
# Obtaining the mean, count, max, min, and quartiles of the given data set
DF_data.describe()
```

[5]:

	order_id	shop_id	user_id	order_amount	total_items
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.00000
mean	2500.500000	50.078800	849.092400	3145.128000	8.78720
std	1443.520003	29.006118	87.798982	41282.539349	116.32032
min	1.000000	1.000000	607.000000	90.000000	1.00000
25%	1250.750000	24.000000	775.000000	163.000000	1.00000
50%	2500.500000	50.000000	849.000000	284.000000	2.00000
75%	3750.250000	75.000000	925.000000	390.000000	3.00000
max	5000.000000	100.000000	999.000000	704000.000000	2000.00000

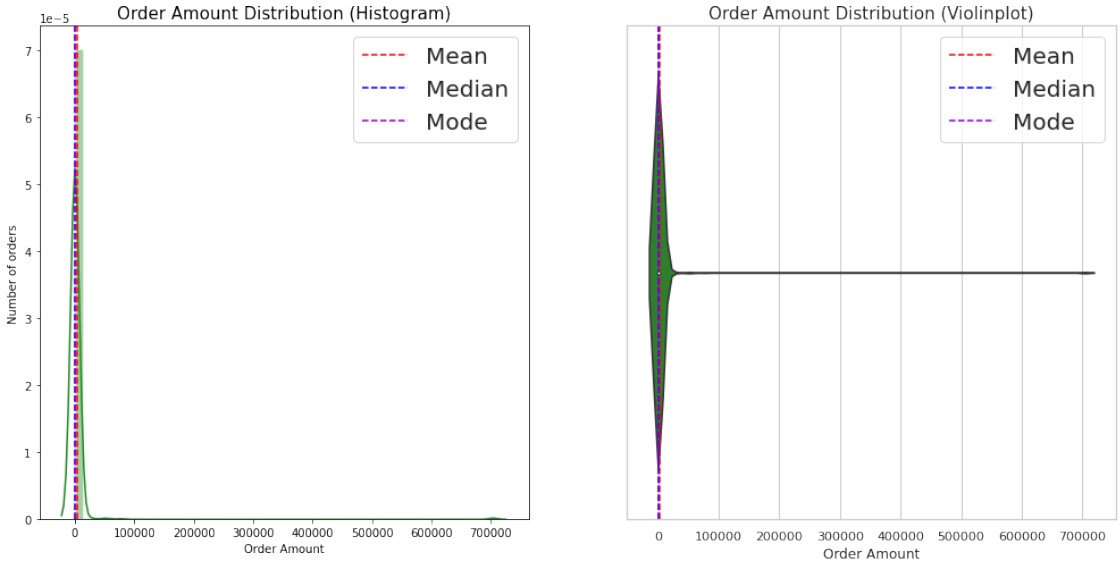
7.0.1 We observe from the above results that there is a significant difference between the mean and median (2nd quartile - 50%). This results in the right-skewed data distribution. We validate the same by visualizing the data.

8 Step 5: Visualizing the data distribution using histogram and violin plots

[6]:
```
# Setting plot dimensions
plt.rcParams['figure.figsize'] = (17,8)

plt.subplot(1,2,1)
sbn.set(style='whitegrid')
sbn.distplot(DF_data['order_amount'], color='forestgreen')
plt.title("Order Amount Distribution (Histogram)", fontsize = 15)

# Marking mean explicitly
plt.axvline(x=np.mean(DF_data['order_amount']), color='red', ls="--",␣
 ↪label="Mean")
```

```
# Marking median explicitly
plt.axvline(x=np.median(DF_data['order_amount']), color='blue', ls="--",␣
 ↪label="Median")

# Marking mode explicitly
plt.axvline(x=stats.mode(DF_data['order_amount'])[0], color='darkviolet',␣
 ↪ls="--", label="Mode")

# For labels and legend
plt.xlabel("Order Amount")
plt.ylabel("Number of orders")
plt.legend(loc=1, prop={'size': 20})

plt.subplot(1,2,2)
sbn.set(style='whitegrid')
sbn.violinplot(DF_data['order_amount'],color='forestgreen')
plt.title("Order Amount Distribution (Violinplot)", fontsize = 15)

# Marking mean explicitly
plt.axvline(x=np.mean(DF_data['order_amount']), color='red', ls="--",␣
 ↪label="Mean")

# Marking median explicitly
plt.axvline(x=np.median(DF_data['order_amount']), color='blue', ls="--",␣
 ↪label="Median")

# Marking mode explicitly
plt.axvline(x=stats.mode(DF_data['order_amount'])[0], color='darkviolet',␣
 ↪ls="--", label="Mode")
plt.xlabel("Order Amount")
plt.legend(loc=1, prop={'size': 20})
```

[6]: <matplotlib.legend.Legend at 0x7f25696d55c0>



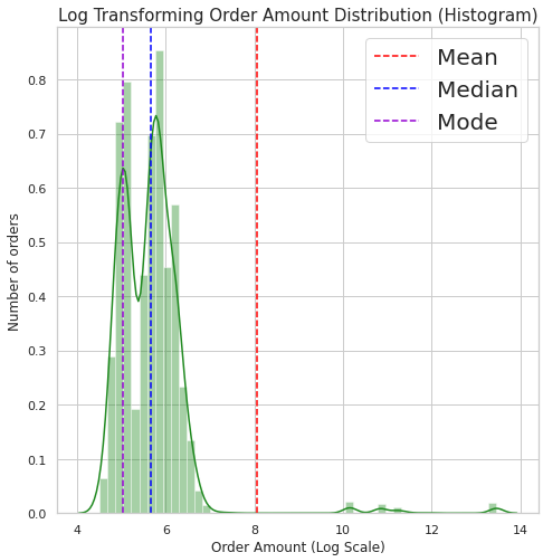
8.0.1 Following my intuition, the data distribution is skewed. To better visualize this skewed data, I will perform the log transformation of the data.

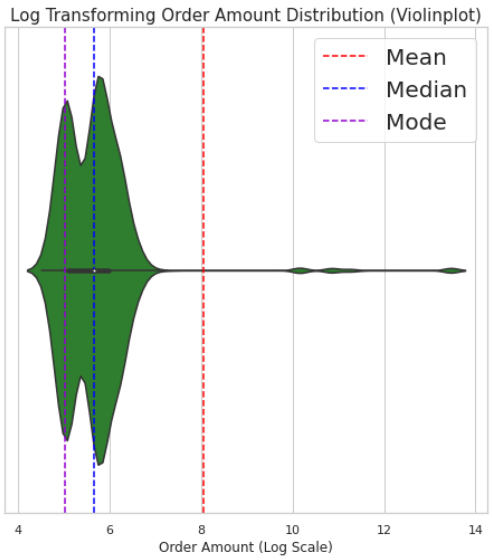
9 Step 6: Performing log transformation for visualizing the skewed data

[7]:
```
plt.subplot(1,2,1)
sbn.set(style='whitegrid')
sbn.distplot(np.log(DF_data['order_amount']), color='forestgreen')
plt.title("Log Transforming Order Amount Distribution (Histogram)", fontsize =␣
↪15)

# Marking Mean (log scale) explicitly
plt.axvline(x=np.log(np.mean(DF_data['order_amount'])), color='red', ls="--",␣
↪label="Mean")

# Marking Median (log scale) explicitly
plt.axvline(x=np.log(np.median(DF_data['order_amount'])), color='blue',␣
↪ls="--", label="Median")

# Marking Mode (log scale) explicitly
plt.axvline(x=np.log(stats.mode(DF_data['order_amount'])[0]),␣
↪color='darkviolet', ls="--", label="Mode")
plt.xlabel("Order Amount (Log Scale)")
plt.ylabel("Number of orders")
plt.legend(loc=1, prop={'size': 20})
```

```
plt.subplot(1,2,2)
sbn.set(style='whitegrid')
sbn.violinplot(x=np.log(DF_data['order_amount']),color='forestgreen')
plt.title("Log Transforming Order Amount Distribution (Violinplot)", fontsize =␣
↪15)

# Marking Mean (log scale) explicitly
plt.axvline(x=np.log(np.mean(DF_data['order_amount'])), color='red', ls="--",␣
↪label="Mean")

# Marking Median (log scale) explicitly
plt.axvline(x=np.log(np.median(DF_data['order_amount'])), color='blue',␣
↪ls="--", label="Median")

# Marking Mode (log scale) explicitly
plt.axvline(x=np.log(stats.mode(DF_data['order_amount'])[0]),␣
↪color='darkviolet', ls="--", label="Mode")
plt.xlabel("Order Amount (Log Scale)")
plt.legend(loc=1, prop={'size': 20})
```

[7]: <matplotlib.legend.Legend at 0x7f25695a0358>





9.0.1 We observe from the above results that the data distribution is right-skewed. Moreover, there is a significant difference in the values of mean, median, and mode.

10 Step 7: Finally, I decide to use median metric and compute its value.

[8]:
```
print("I will use the {} metric for the  given dataset and its value is {}".
 ↪format("median", np.median(DF_data['order_amount'])))
```

```
I will use the median metric for the  given dataset and its value is 284.0
```