Project 1

E-Commerce Dataset visualization

You are a data analyst at an e-commerce firm (just like Amazon) where you onboard

sellers/retailers to sell their products on your website.

After a customer purchases a product from your website, the seller gets notified to fulfill

that order. Once the customer receives the product, the customer gets a survey email

where he/she can share his/her shopping experience.

You have been asked to present a complete summary of the performance of the firm.

Include all possible metrics and graph visualizations from delivery, pricing, products,

payments, etc. to help the CEO make rational business decisions.

Tools / Skills Used

1. Python Programming
2. Jupyter Notebook
3. Pandas
4. Numpy
5. Matplotlib
6. Seaborn
7. Excel
8. Data Visualization
9. Tableau.

Workflow:

1.Importing all the library.

```
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
```

2.Uploading all the dataset we have .

```
orders = pd.read_csv('orders.csv')
customers = pd.read_csv('customers.csv')
order_items = pd.read_csv('order_items.csv')
payments = pd.read_csv('payments.csv')
products = pd.read_csv('products.csv')
sellers = pd.read_csv('sellers.csv')
user_feedback = pd.read_csv('user_feedback.csv')
```

3.Cleaning all the dataset and adding some new columns in orders dataset file and then merging all the dataset base on the primary key and foreign key.

4.After merging 6 files we have final dataset.

```
order_merge_order_item_products_customers_payments_seller
```

	seller_id	seller_state	order_id	payment_type	payment_value	customer_id	customer_unique_id	customer_state	product_id	product_photos_co
0	3442f8959a	SP	bc8a5de6ab	credit_card	158.0	1554ffe702	af0f26435f	SP	325a06bcce	
1	3442f8959a	SP	6d953888a9	credit_card	117.0	a973c4e3ad	b452742346	MG	f4621f8ad6	
2	3442f8959a	SP	4a90af3e85	credit_card	115.0	9d6837f970	f421a2a66b	SP	ffb64e34a3	
3	d1b65fc7de	SP	83f24bbc4f	credit_card	231.0	d49bac4ac7	51dc561233	RJ	765c417cdc	
4	d1b65fc7de	SP	4436eb8a73	credit_card	383.0	3ab417ebff	11426ed998	SP	51b10a13b4	
...	...	...	...	...	...	...	...	...	...	...
117605	525771517	NaN	4467611bf4	boleto	729.0	2b5292c3f7	abe9bf925c	RS	45bb83550f	
117606	525771517	NaN	a21218028d	credit_card	729.0	228874050	f475dc24f1	SP	45bb83550f	
117607	484380594	NaN	43da1954a3	credit_card	1474.0	4433a42a21	8865836804	SP	2676f64fcc	
117608	484380594	NaN	43da1954a3	credit_card	1474.0	4433a42a21	8865836804	SP	2676f64fcc	
117609	6.95E+303	NaN	9b7a838614	credit_card	182.0	61dd7b5fa2	70dad00a4a	SP	66a5ffd104	N

117610 rows × 29 columns

5.After doing some edit in csv file now we will again reupload this data in Jupyter notebook. Thin this edit we have added one columns on seller type and these columns have 2 value good sellers and bad sellers based on there performance .

```
My_data = pd.read_csv('MyData.csv')
```

```
My_data.head()
```

	Unnamed: 0	seller_id	seller_state	order_id	payment_type	payment_value	customer_id	customer_unique_id	customer_state	product_id	...	Days_Re
0	0	3442f8959a	SP	bc8a5de6ab	credit_card	158.0	1554ffe702	af0f26435f	SP	325a06bcce	...	
1	1	3442f8959a	SP	6d953888a9	credit_card	117.0	a973c4e3ad	b452742346	MG	f4621f8ad6	...	
2	2	3442f8959a	SP	4a90af3e85	credit_card	115.0	9d6837f970	f421a2a66b	SP	ffb64e34a3	...	
3	3	d1b65fc7de	SP	83f24bbc4f	credit_card	231.0	d49bac4ac7	51dc561233	RJ	765c417cdc	...	
4	4	d1b65fc7de	SP	4436eb8a73	credit_card	383.0	3ab417ebff	11426ed998	SP	51b10a13b4	...	

5 rows × 32 columns

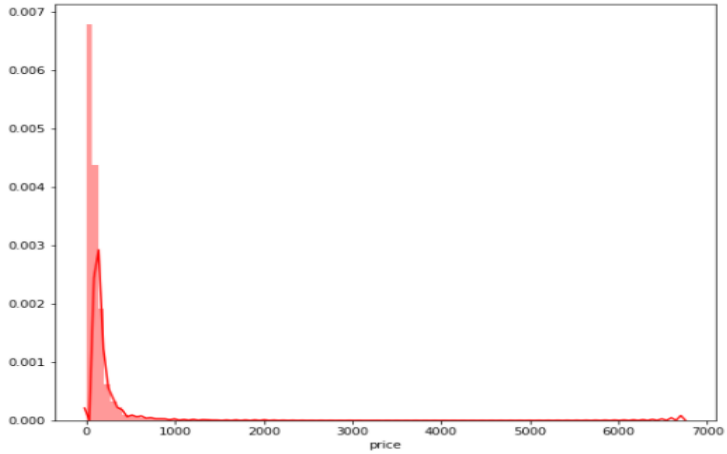
6.Now we will try to do some EDA and then we will move to Tableau visualization.

```
My_data.info()
```

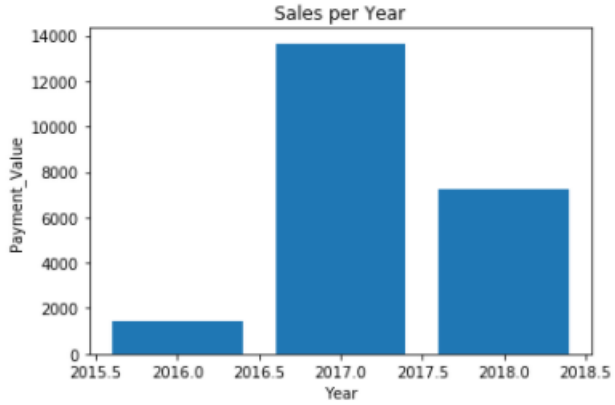
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 117610 entries, 0 to 117609
Data columns (total 32 columns):
Unnamed: 0                          117610 non-null int64
seller_id                           117610 non-null object
seller_state                        116446 non-null object
order_id                            117610 non-null object
payment_type                        117607 non-null object
payment_value                       117607 non-null float64
customer_id                         117610 non-null object
customer_unique_id                  117610 non-null object
customer_state                      117610 non-null object
product_id                          117610 non-null object
product_photos_count                115578 non-null float64
product_length_cm                   117255 non-null float64
product_height_cm                   117255 non-null float64
product_width_cm                    117255 non-null float64
product_category_name               117275 non-null object
product_weight_kg                   117255 non-null float64
order_status                        117610 non-null object
order_purchase_timestamp            117610 non-null object
order_approved_at                   117595 non-null object
order_delivered_to_carrier_date     116365 non-null object
order_delivered_to_customer_date    115043 non-null object
order_estimated_delivery_date       117610 non-null object
Days_Req_By_Sellers                 117610 non-null int64
Days_Req_By_Sellers_Positive        117610 non-null int64
Days_req_by_carrier                 117610 non-null int64
Days_req_by_carrier_Positive        117610 non-null int64
Total_Days_Req                      117610 non-null int64
Total_Days_Req_Positive             117610 non-null int64
Year_Of_Purchase                    117610 non-null int64
Month_Of_Purchase_Number            117610 non-null int64
Month Name                          117610 non-null object
price                               117610 non-null int64
dtypes: float64(6), int64(10), object(16)
memory usage: 28.7+ MB
```

```
print(My_Data['price'].describe())
plt.figure(figsize=(9, 8))
sns.distplot(My_Data['price'], color='r', bins=100, hist_kws={'alpha': 0.4});
```

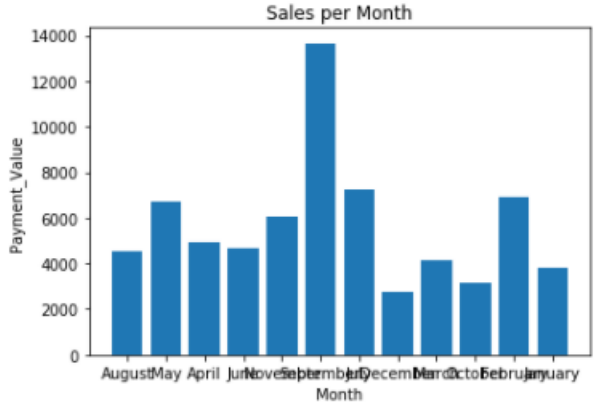
```
count    113393.000000
mean        119.596395
std         183.467199
min           0.000000
25%          39.000000
50%          74.000000
75%         133.000000
max        6735.000000
Name: price, dtype: float64
```




```
plt.bar(My_Data["Year_Of_Purchase"],My_Data["payment_value"])
plt.title("Sales per Year")
plt.xlabel("Year")
plt.ylabel("Payment_Value")
plt.show()
```

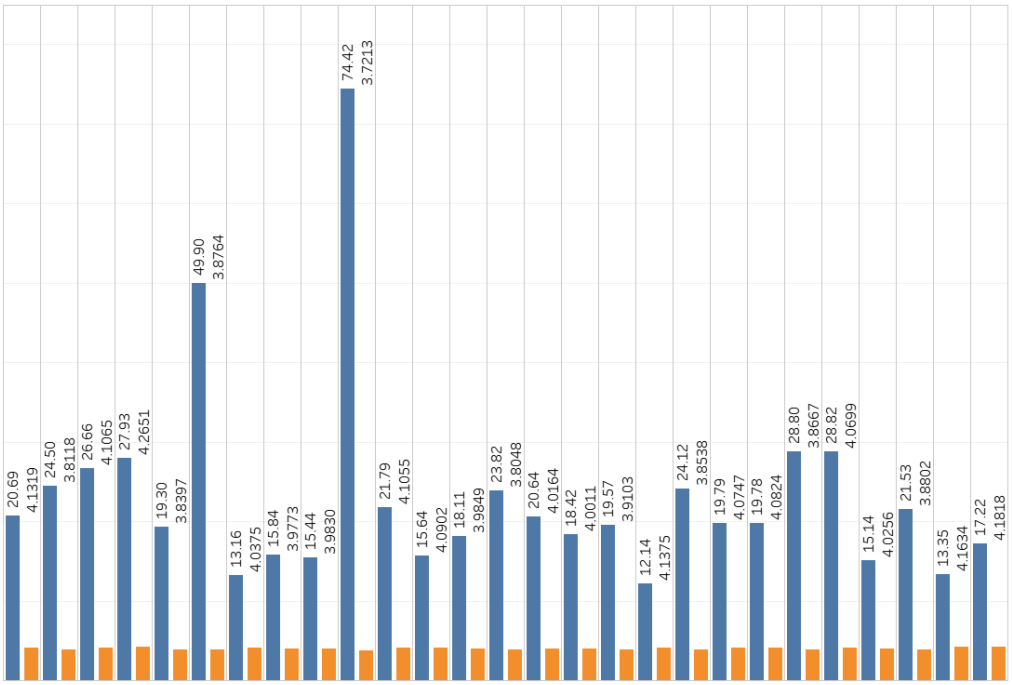



```
plt.bar(My_Data["Month Name"],My_Data["payment_value"])
plt.title("Sales per Month")
plt.xlabel("Month")
plt.ylabel("Payment_Value")
plt.show()
```

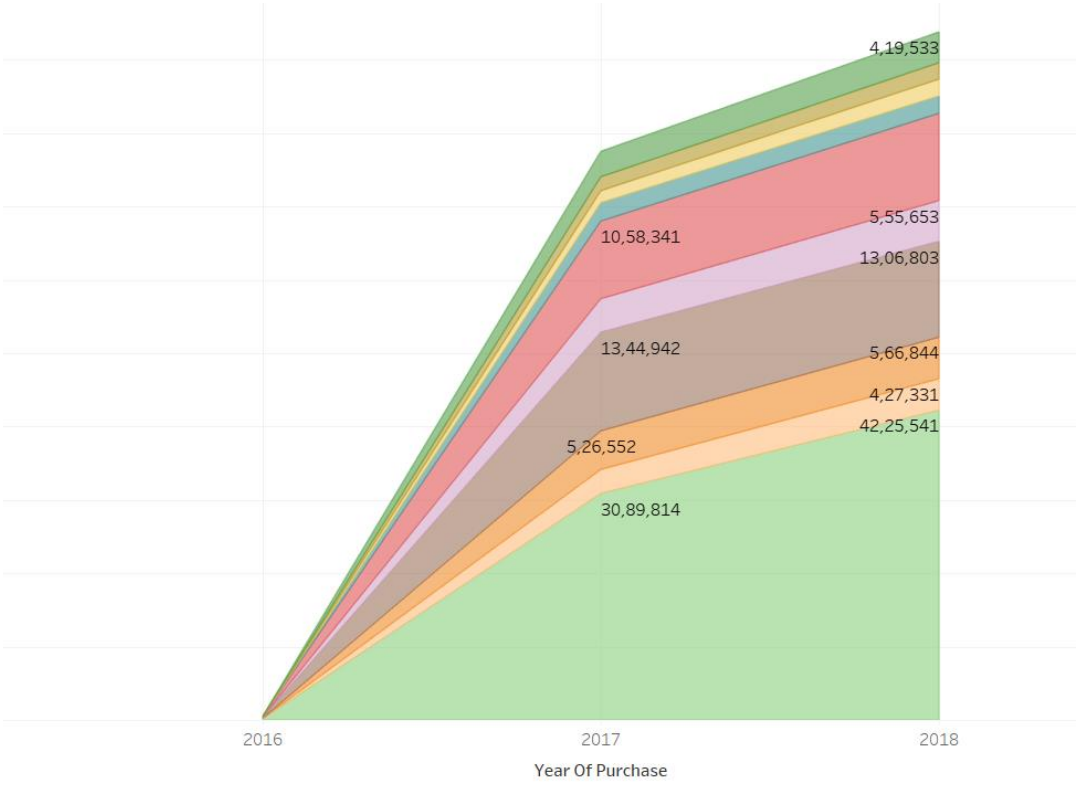


7.Tableau visualization.

Reviews Score State Wise And Day Required To Deliver

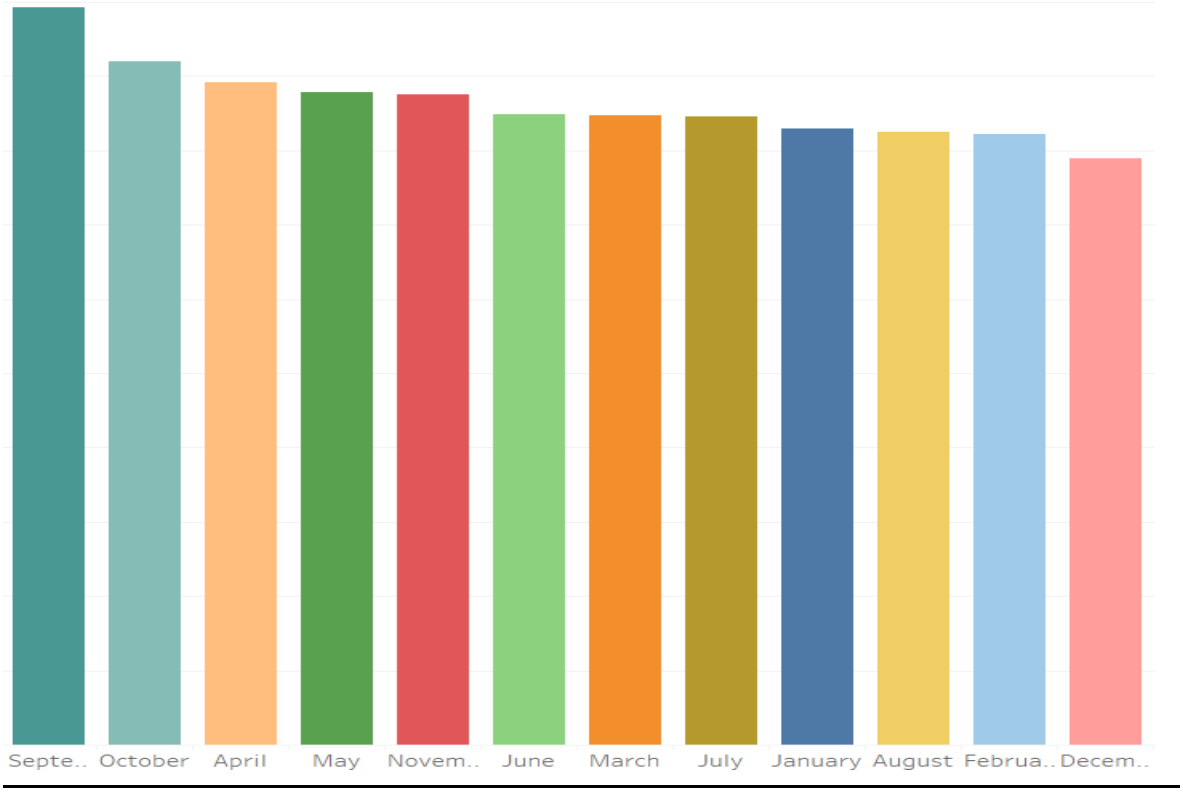


Total sales Year wise And State Wise

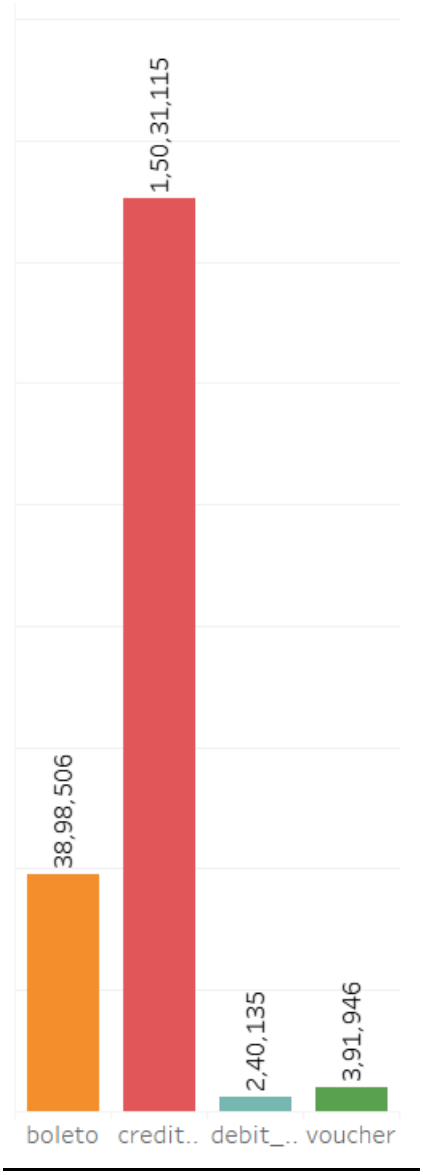


Reviews Score vs Type of Seller

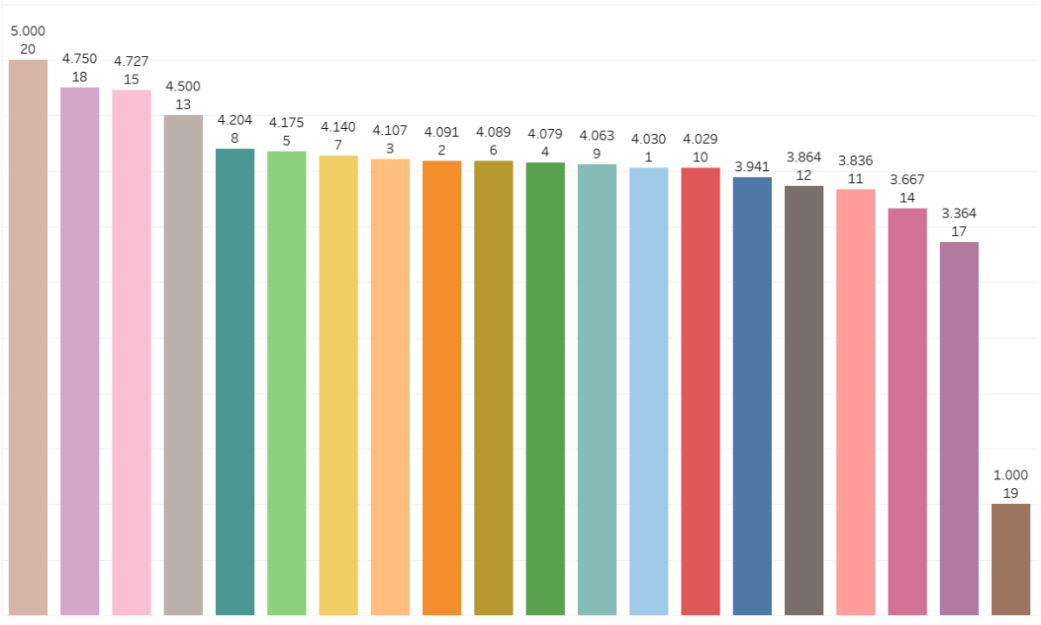
Average Payment Value Month Wise



Views on Payments type



RelationShip between reviews_score and Photo_count



Top 15 Selling item