The Zomato Data Analytics Project aims to analyse restaurant and customer data from Zomato to identify key performance indicators (KPIs) and derive actionable insights. The project focuses on examining dining patterns, votes, and ratings provided by users. By utilizing data analysis and visualization tool like Python, the project aims to draw inferences and conclusions from the data.

Zomato Data Analysis

Utkarsh Sharma
sharma.utkarsh2402@gmail.com

Zomato Sales Analysis

All the related steps to analyze the data.

Step-1. Import all the libraries.

Code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

[2] ✓ 0.0s

Step-2. Read the dataset

Code: df = pd.read_csv("Zomato data .csv")

print(df.head(5))

Output:

	name	online_order	book_table	rate	votes	approx_cost(for two people)	listed_in(type)
0	Jalsa	Yes	Yes	4.1/5	775	800	Buffet
1	Spice Elephant	Yes	No	4.1/5	787	800	Buffet
2	San Churro Cafe	Yes	No	3.8/5	918	800	Buffet
3	Addhuri Udupi Bhojana	No	No	3.7/5	88	300	Buffet
4	Grand Village	No	No	3.8/5	166	600	Buffet

Step3. Check all the info related to the dataset.

Code: df.info()

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 148 entries, 0 to 147
Data columns (total 7 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   name                         148 non-null    object
 1   online_order                 148 non-null    object
 2   book_table                   148 non-null    object
 3   rate                         148 non-null    object
 4   votes                        148 non-null    int64 
 5   approx_cost(for two people)  148 non-null    int64 
 6   listed_in(type)              148 non-null    object
dtypes: int64(2), object(5)
memory usage: 8.2+ KB
```

Step-4. Check all the null values in the dataset.

Code : df.isna().sum()

Output:

```
name                           0
online_order                   0
book_table                     0
rate                           0
votes                          0
approx_cost(for two people)    0
listed_in(type)                0
dtype: int64
```

Step-5. Change the rating datatype to float datatype.

Code:

```
def handle(a):
    a=str(a).split('/')
    a = a[0]
    return float(a)
df['rate']=df['rate'].apply(handle)
df.head()
```

Output:

	name	online_order	book_table	rate	votes	approx_cost(for two people)	listed_in(type)
0	Jalsa	Yes	Yes	4.1	775	800	Buffet
1	Spice Elephant	Yes	No	4.1	787	800	Buffet
2	San Churro Cafe	Yes	No	3.8	918	800	Buffet
3	Addhuri Udupi Bhojana	No	No	3.7	88	300	Buffet
4	Grand Village	No	No	3.8	166	600	Buffet

+ Code + Markdown

Cleaning part is done now

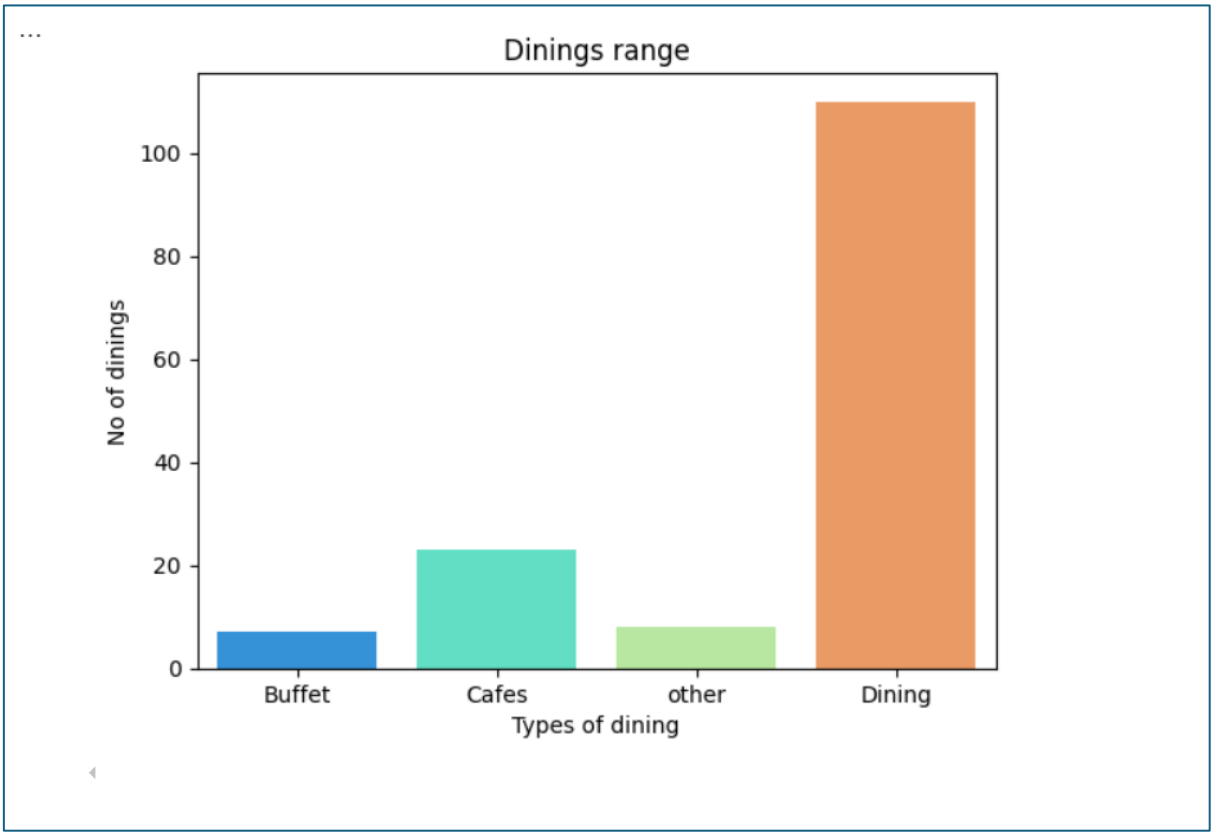
Data Visualization

1. No of Dining

Code:

```
sns.countplot(x=df['listed_in(type)'],palette='rainbow')
plt.title("Dinings range")
plt.xlabel("Types of dining")
plt.ylabel("No of dinings")
plt.show()
```

Output:

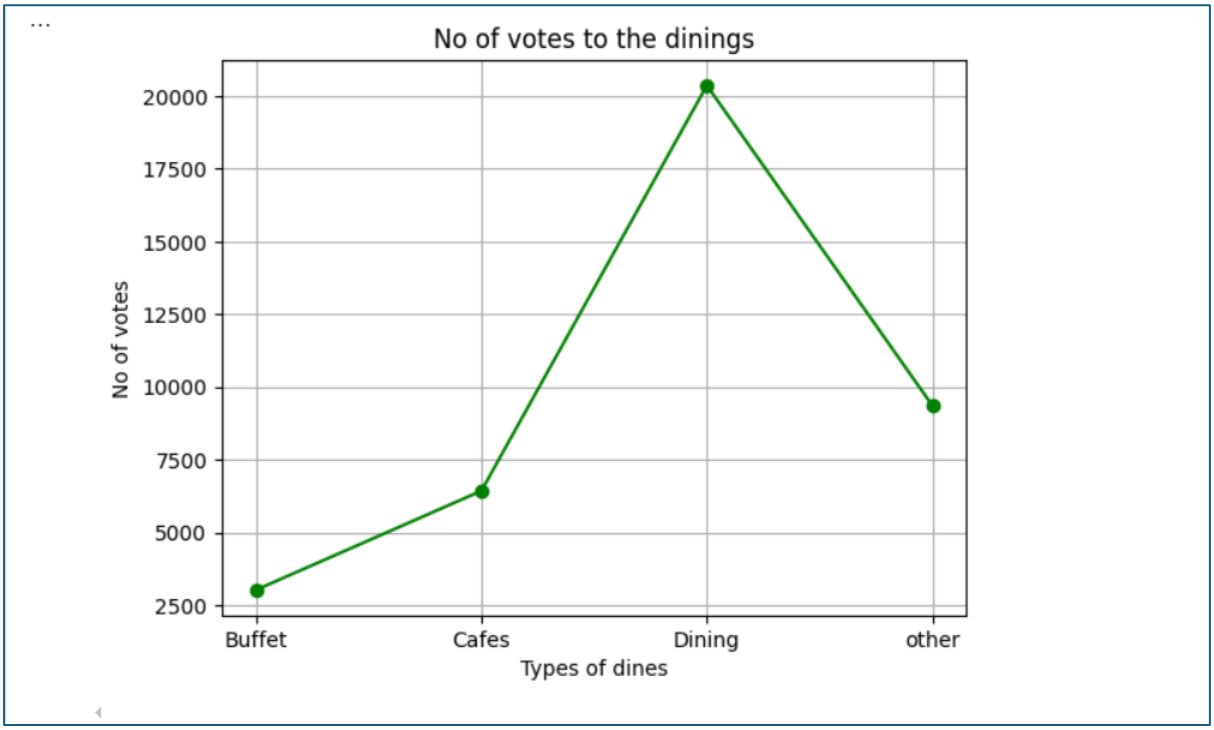


2. Vote wise Dining

Code:

```
group = df.groupby('listed_in(type)')['votes'].sum()
result = pd.DataFrame({'votes':group})
plt.plot(result,color='green',marker='o')
plt.title("No of votes to the dinings")
plt.ylabel('No of votes')
plt.xlabel("Types of dines")
plt.grid()
plt.show()
```

Output:



3. Restaurants who got maximum votes

Code:

```
max =df['votes'].max()

df.loc[df['votes']==max]
```

Output:

	name	online_order	book_table	rate	votes	approx_cost(for two people)	listed_in(type)
38	Empire Restaurant	Yes	No	4.4	4884	750	other

4. Online Booking Status

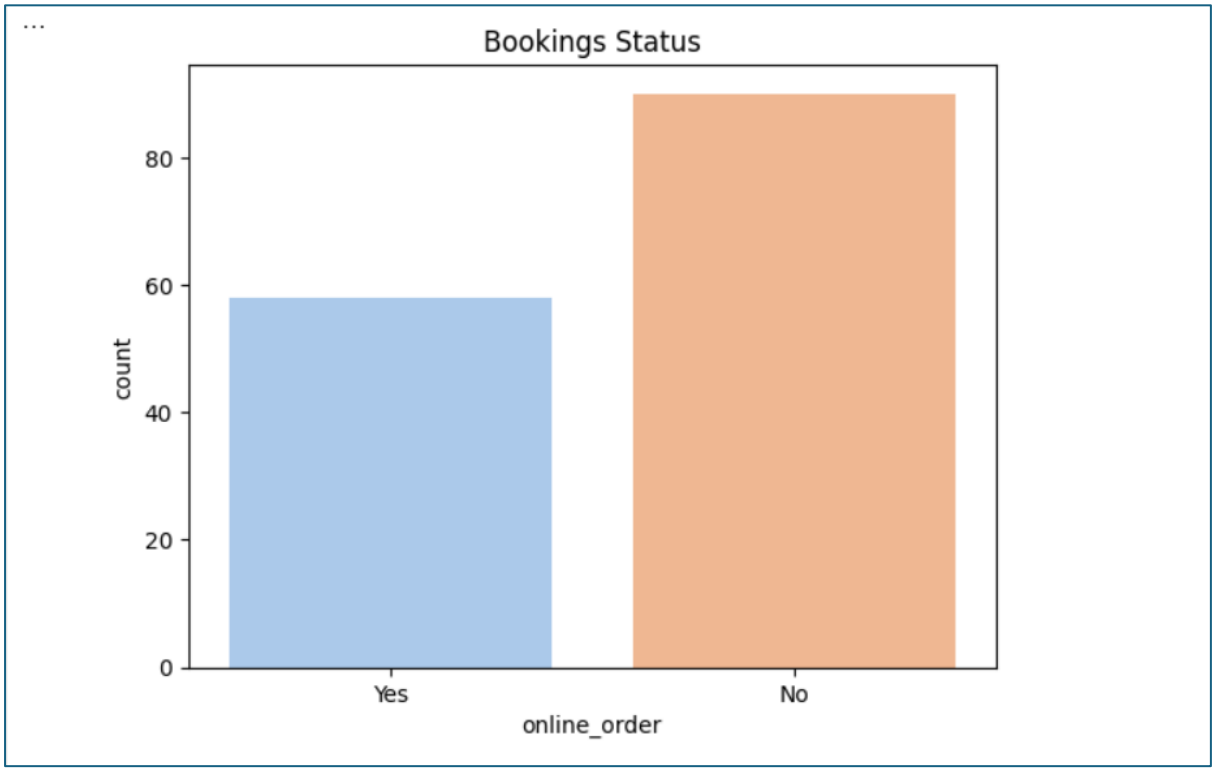
Code:

```
sns.countplot(x=df['online_order'],palette='pastel')

plt.title("Bookings Status")

plt.show()
```

Output:

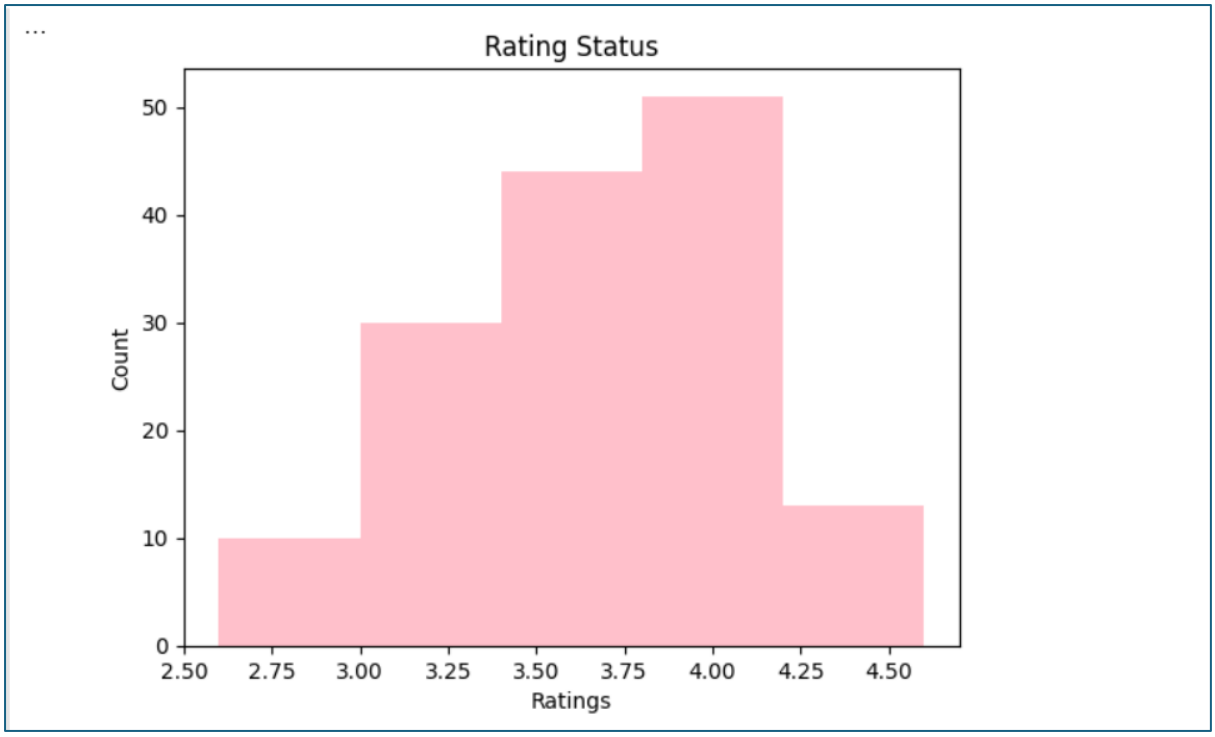


5. Rating Status

Code:

```
plt.hist(x=df['rate'],bins=5,color='pink')
plt.title('Rating Status')
plt.xlabel('Ratings')
plt.ylabel('Count')
plt.show()
```

Output:

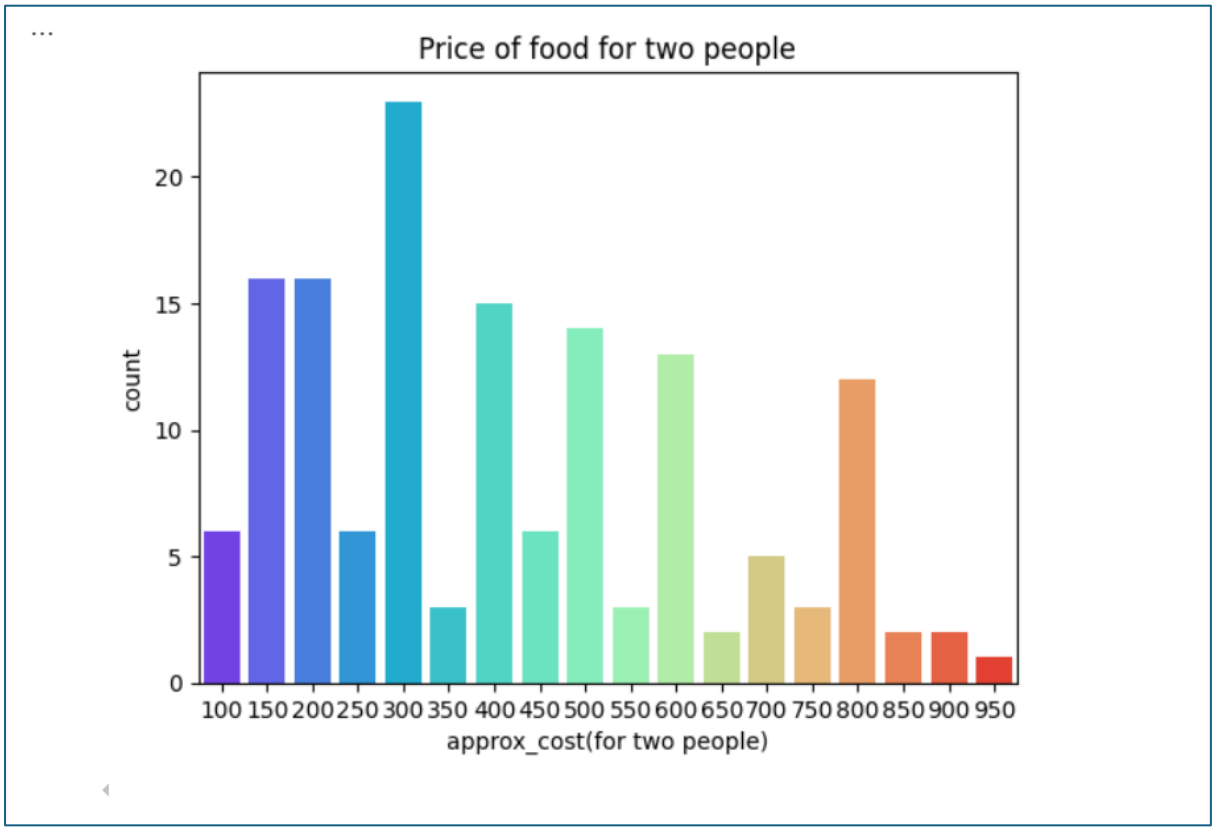


6. **Price of food for two people**

Code:

```
sns.countplot(x=df['approx_cost(for two people)'],palette='rainbow')
plt.title('Price of food for two people')
plt.show()
```

Output:

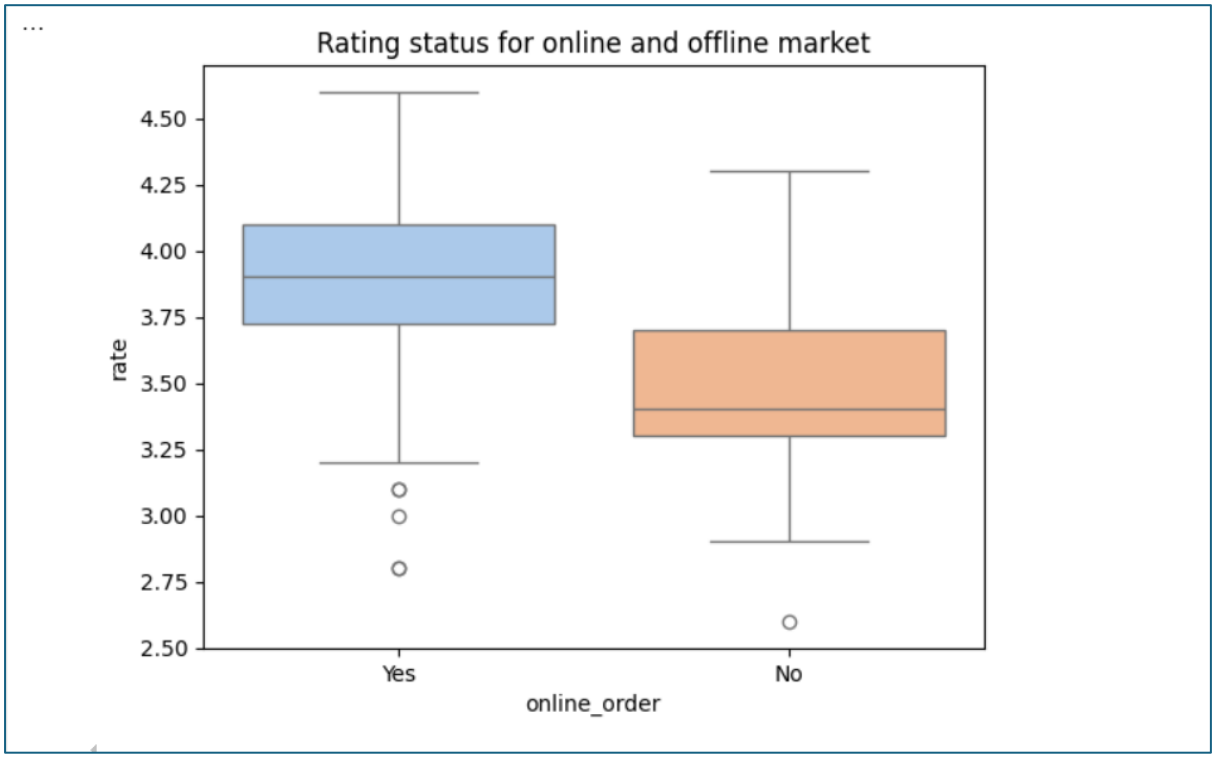


7. **Offline Vs Online Market**

Code:

```
sns.boxplot(x=df['online_order'],y=df['rate'],palette='pastel')
plt.title("Rating status for online and offline market")
plt.show()
```

Output:



Dashboard Preview

Code:

```
plt.figure(figsize=(18,12))
plt.subplot(2,3,1)
sns.countplot(x=df['approx_cost(for two people)'],palette='rainbow')
plt.title('Price of food for two people')

plt.subplot(2,3,2)
sns.boxplot(x=df['online_order'],y=df['rate'],palette='pastel')
plt.title("Rating status for online and offline market")

plt.subplot(2,3,3)
plt.hist(x=df['rate'],bins=5,color='pink')
plt.title('Rating Status')
plt.xlabel('Ratings')
plt.ylabel('Count')

plt.subplot(2,3,4)
group = df.groupby('listed_in(type)')['votes'].sum()
result = pd.DataFrame({'votes':group})
plt.plot(result,color='green',marker='o')
plt.title("No of votes to the dinings")
plt.ylabel('No of votes')
plt.xlabel("Types of dines")
plt.grid()

plt.subplot(2,3,5)
sns.countplot(x=df['online_order'],palette='pastel')
plt.title("Bookings Status")
```

```
plt.subplot(2,3,6)
sns.countplot(x=df['listed_in(type)'],palette='rainbow')
plt.title("Dinings range")
plt.xlabel("Types of dining")
plt.ylabel("No of dinings")

plt.suptitle("Zomato Data Analysis",fontsize=40)
plt.show()
```

Output: