ZOMATO DATA ANALYSIS USING PYTHON

This project analyzes Zomato's customer and restaurant data to uncover insights about ordering patterns, customer preferences, and restaurant performance. Using Python for data analysis and visualization, the project answers key business questions that can help Zomato enhance its services and strategize more effectively .

1. **What type of restaurant do the majority of customers order from?**
2. **How many votes has each type of restaurant received from customers?**
3. **What are the ratings that the majority of restaurants have received?**
4. **Zomato has observed that most couples order most of their food online. What is their average spending on each order?**
5. **Which mode (online or offline) has received the maximum rating?**
6. **Which type of restaurant received more offline orders, so that Zomato can provide those customers with some good offers**?

Data Analyst Project

Import Libraries

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Create data Frame

[2]:
```
dataframe = pd.read_csv("zomato data .csv")
print(dataframe)
```

```
                      name online_order book_table   rate  votes  \
0                    Jalsa          Yes        Yes  4.1/5    775
1           Spice Elephant          Yes         No  4.1/5    787
2          San Churro Cafe          Yes         No  3.8/5    918
3    Addhuri Udupi Bhojana           No         No  3.7/5     88
4            Grand Village           No         No  3.8/5    166
..                     ...          ...        ...    ...    ...
143       Melting Melodies           No         No  3.3/5      0
144        New Indraprasta           No         No  3.3/5      0
145           Anna Kuteera          Yes         No  4.0/5    771
146                 Darbar           No         No  3.0/5     98
147          Vijayalakshmi          Yes         No  3.9/5     47

     approx_cost(for two people) listed_in(type)
0                            800          Buffet
1                            800          Buffet
2                            800          Buffet
3                            300          Buffet
4                            600          Buffet
..                           ...             ...
143                          100          Dining
144                          150          Dining
145                          450          Dining
146                          800          Dining
147                          200          Dining

[148 rows x 7 columns]
```

[3]:
```
dataframe
```

[3]:

	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
...	...	...	...	...	...	...	...
143	Melting Melodies	No	No	3.3/5	0	100	Dining
144	New Indraprasta	No	No	3.3/5	0	150	Dining
145	Anna Kuteera	Yes	No	4.0/5	771	450	Dining
146	Darbar	No	No	3.0/5	98	800	Dining
147	Vijayalakshmi	Yes	No	3.9/5	47	200	Dining

148 rows × 7 columns

convert the data type of "rate" column to float and remove the denominator

[4]:
```
def handleRate(value):
    value = str(value).split('/')
    value = value[0];
    return float(value)
dataframe['rate']=dataframe['rate'].apply(handleRate)
print(dataframe.head())
```

```
                  name online_order book_table  rate  votes  \
0                Jalsa          Yes         Yes   4.1    775
1       Spice Elephant          Yes          No   4.1    787
2      San Churro Cafe          Yes          No   3.8    918
3  Addhuri Udupi Bhojana          No          No   3.7     88
4          Grand Village          No          No   3.8    166

   approx_cost(for two people) listed_in(type)
0                          800          Buffet
1                          800          Buffet
2                          800          Buffet
3                          300          Buffet
4                          600          Buffet
```

Summary of the data frame

[5]:
```
dataframe.info()
```

```
<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    float64
 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: float64(1), int64(2), object(4)
memory usage: 8.2+ KB
```

Conclusion: there is no NULL value in the data frame

Type of Resturant

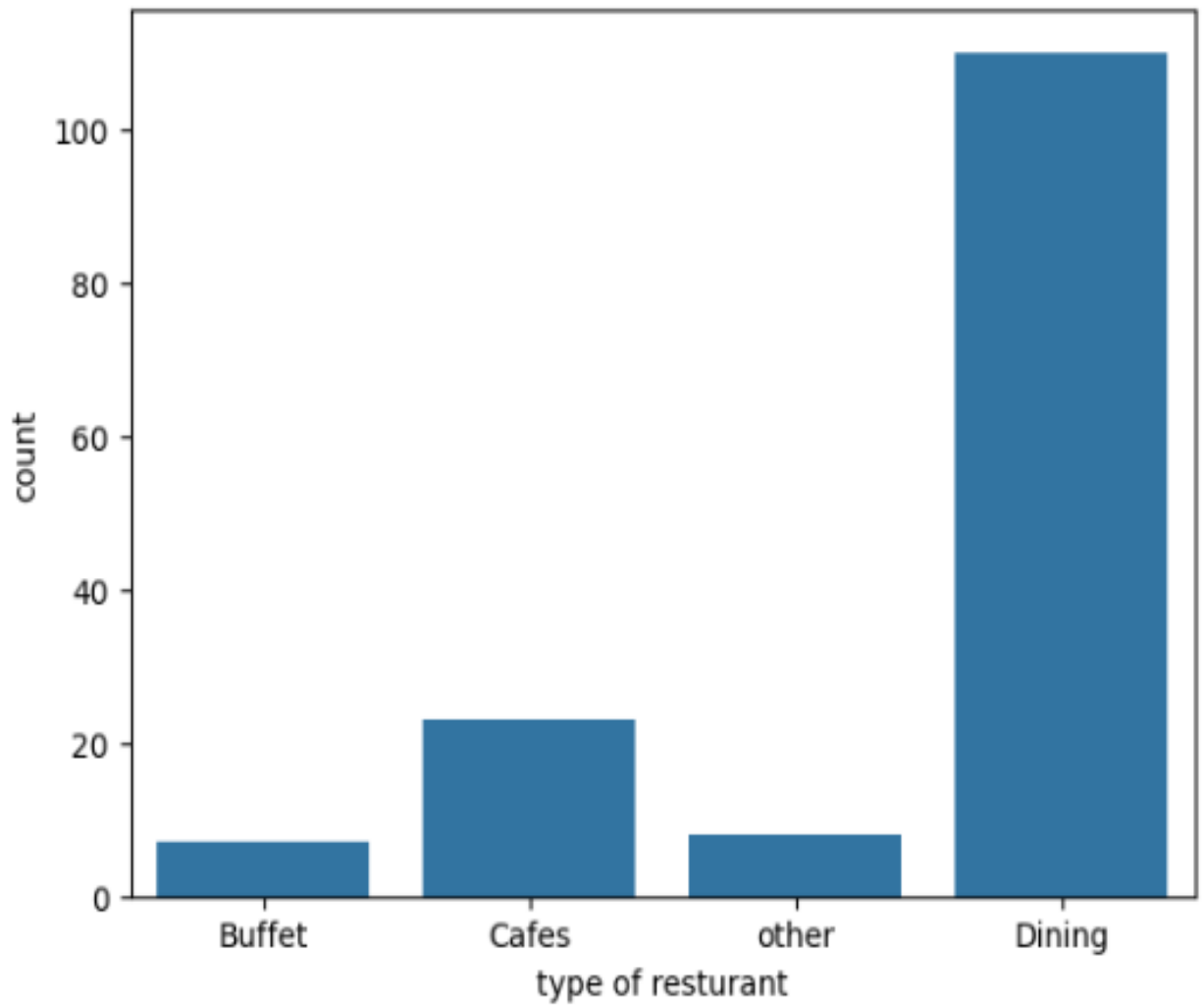
[6]:
```
dataframe.head()
```

[6]:

	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

[7]:
```
sns.countplot(x=dataframe['listed_in(type)'])
plt.xlabel("type of resturant")
```

[7]:
```
Text(0.5, 0, 'type of resturant')
```



Conclusion - Majority of the resturant falls in dining category

[8]:
```
dataframe.head()
```

[8]:

	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

[11]:
```
grouped_data = dataframe.groupby('listed_in(type)')['votes'].sum()
result = pd.DataFrame({'votes': grouped_data})
plt.plot(result, c= "green",marker="o")
plt.xlabel("type os restaurant",c= "red", size = 20)
plt.ylabel("votes",c="red", size = 20)
```

[11]:
```
Text(0, 0.5, 'votes')
```

Conclusion:- Dining resaurant has recieved maximum votes

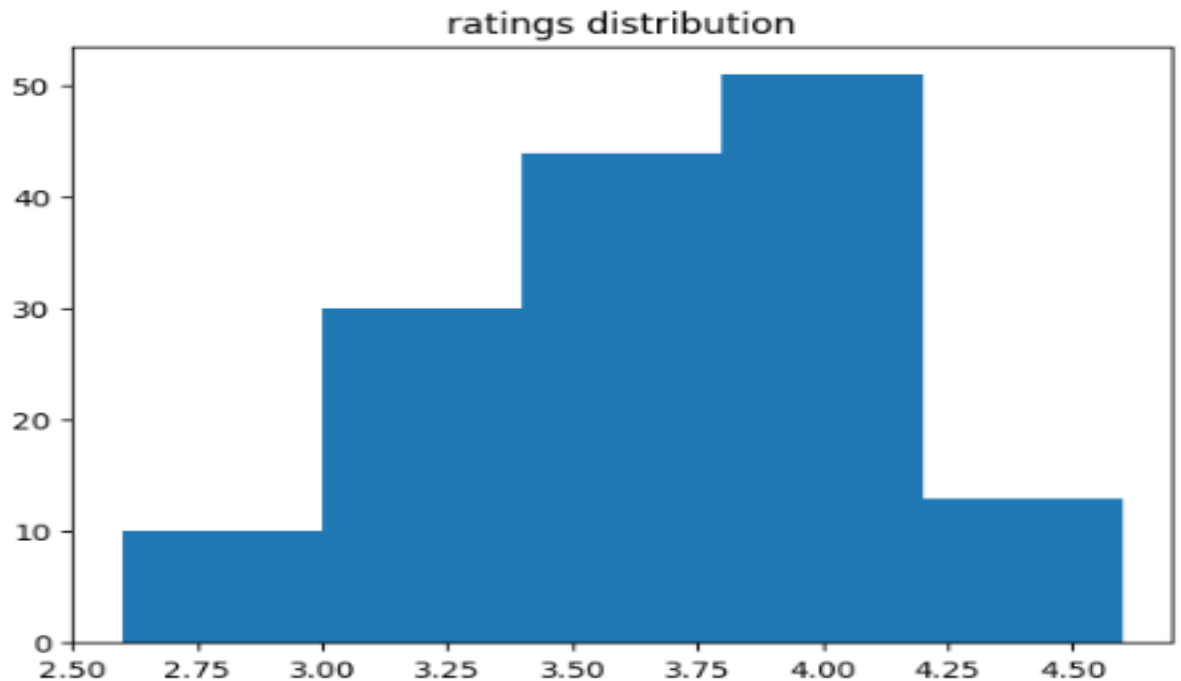
What aee the rating that majority restaurant received ?

[12]:
```
dataframe.head()
```

[12]:

	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

[15]:
```
plt.hist(dataframe['rate'],bins = 5)
plt.title("ratings distribution")
plt.show()
```



conclusion- the majority restaurant received rating from 3.5 to 4

[16]:
```
dataframe.head()
```

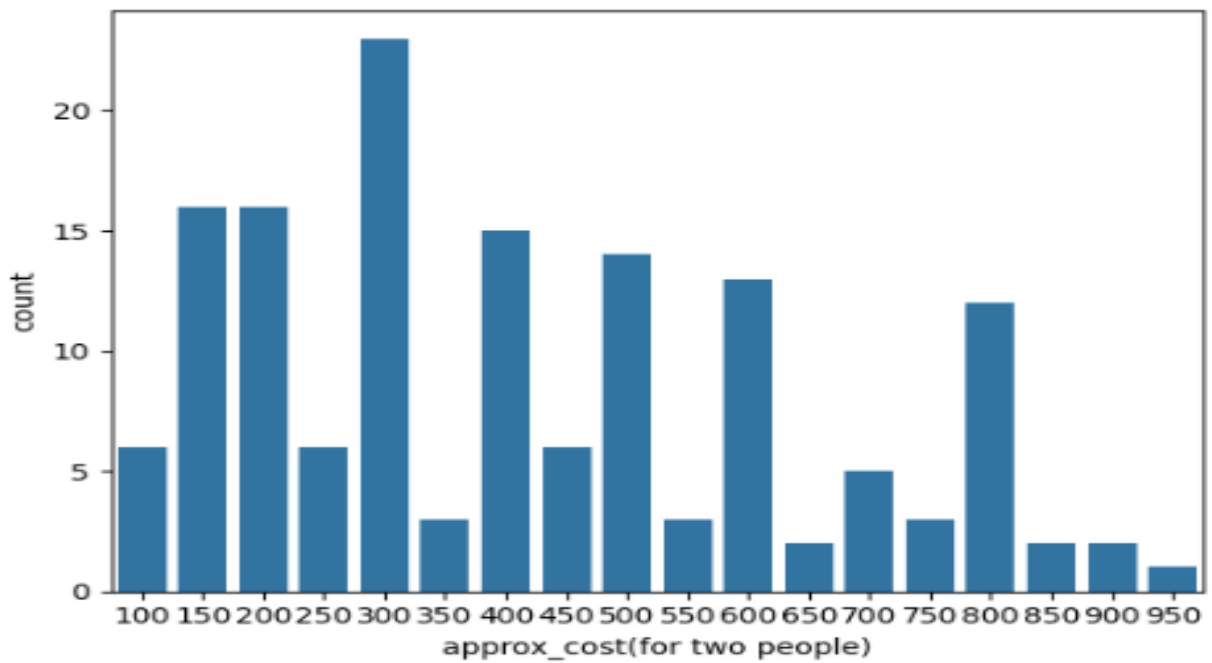
[16]:

	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

average spending of couples on each order

[19]:
```
couple_data = dataframe['approx_cost(for two people)']
sns.countplot(x= couple_data)
```

[19]:
```
<Axes: xlabel='approx_cost(for two people)', ylabel='count'>
```



conclusion- the majority of couple prefer restaurant with an approximate cost of 300 rupees

Which mode(online or offline) receives maximum rating online or offline

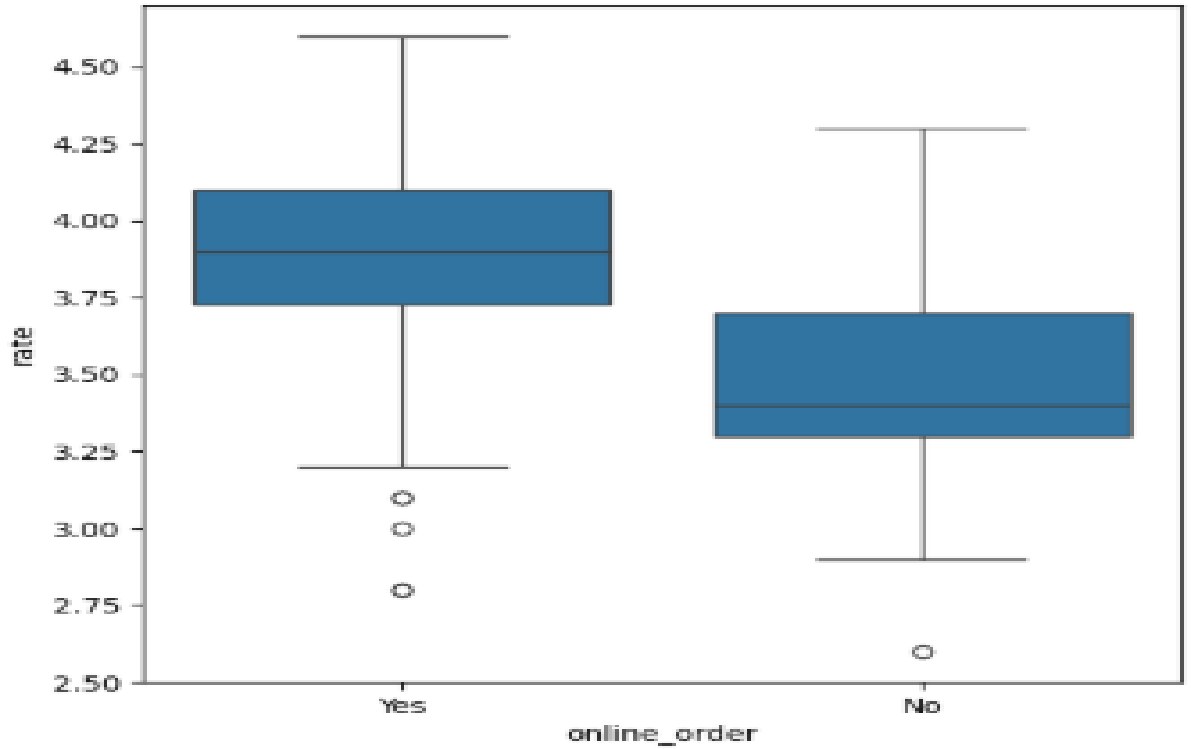
[20]:
```
dataframe.head()
```

[20]:

	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

[21]:
```
plt.figure(figsize = (6,6))
sns.boxplot(x='online_order', y= 'rate', data= dataframe)
```

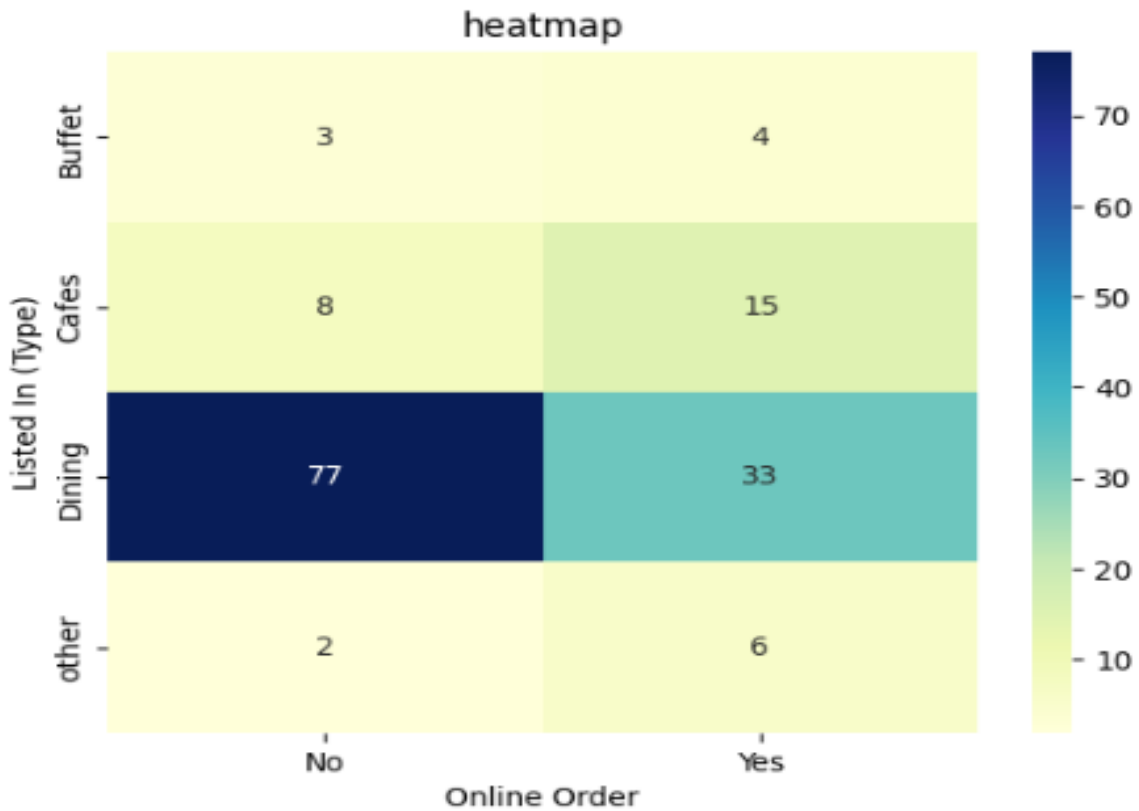
[21]:
```
<Axes: xlabel='online_order', ylabel='rate'>
```



conclusion- offline order receives lower rating in comparision of online order

Which types of restaurant receives more no, of offline order

[29]:
```
pivot_table = dataframe.pivot_table(index='listed_in(type)', columns='online_order', aggfunc='size', fill_value=0)
sns.heatmap(pivot_table, annot=True, cmap="YlGnBu", fmt='d')
plt.title("heatmap")
plt.xlabel("Online Order")
plt.ylabel("Listed In (Type)")
plt.show()
```



CONCLUSION: Dining restaurants primarily accept offline orders, whereas cafes primarily receive online orders. This suggests that clients prefer to place orders in person at restaurants, but prefer online ordering at cafes.