ZOMATO DATA ANALYSIS PROJECT USING PYTHON

Description:

This project focuses on analyzing customer behavior and restaurant performance on Zomato, a leading food delivery and review platform. Using Python and data analysis libraries like Pandas, NumPy, and Matplotlib, the project explores key business questions to uncover insights about customer preferences, restaurant popularity, and order patterns.

The analysis covers:

- Identifying the most preferred restaurant types by customers
- Aggregating customer votes across restaurant categories
- Understanding common rating trends among restaurants
- Estimating average spending per order by couples who order online
- Comparing ratings between online and offline ordering modes
- Detecting restaurant types with high offline order volumes to target promotional offers

This project demonstrates practical skills in data cleaning, grouping, aggregation, and visualization, and provides actionable insights that can help Zomato optimize its marketing and customer engagement strategie

Data Analysis Project Using Python — Zomato Case Study Problems:

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?

Zomato data analysis project

Importing libraries

In [2]:

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

Read data

In [4]:

```
df = pd.read_csv("Zomato data .csv")
print(df)
```

```
                      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]
```

Show data

In [12]:

```
df
```

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

148 rows × 7 columns

Convert the datatype of rate by removing denominator

In [10]:

```
def handleRate(value):
    value = str(value).split('/')
    value = value[0];
    return float(value)

df['rate']=df['rate'].apply(handleRate)
print(df.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  
```

Checking null and info

In [11]:

```
df.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
```

Type of restaurant

In [14]:

```
df.head()
```

Out[14]:

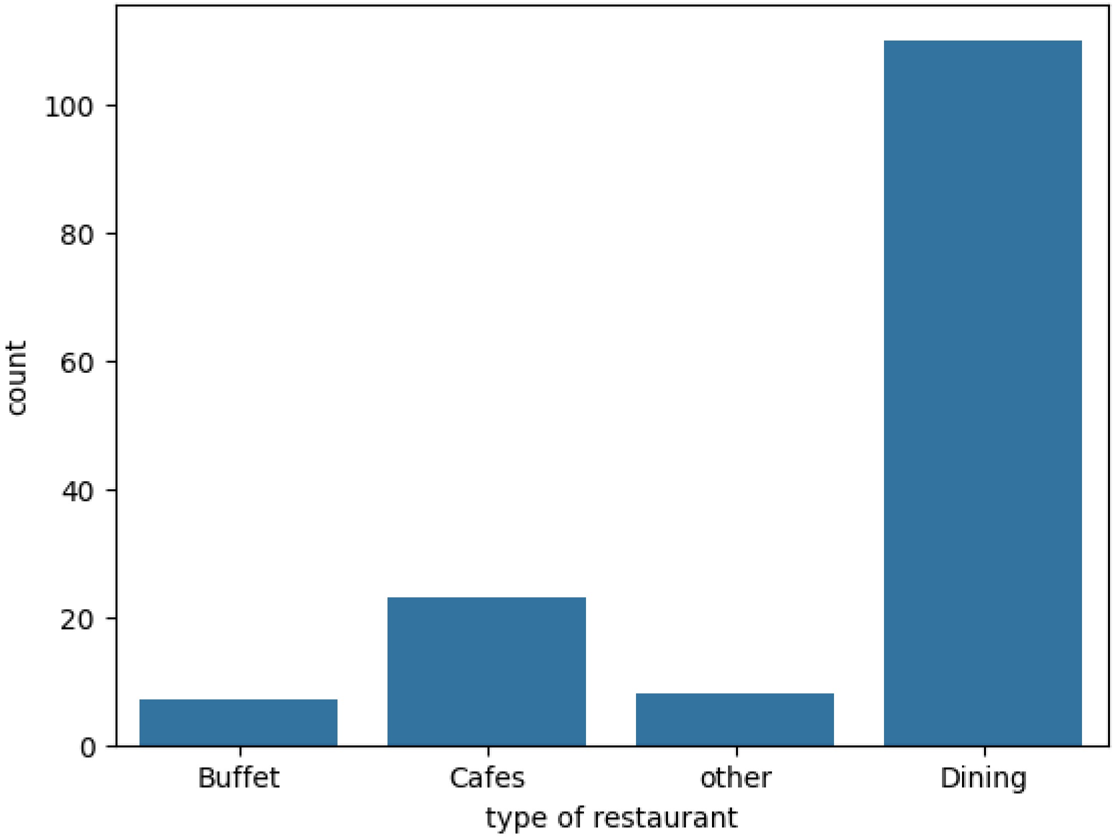
	name	online_order	book_table	rate	votes	approx_cost(for two people)	listed_in(type)
0	Jalsa	Yes	Yes	4.1	775	800	Buffet
1	Spice Elephant	Yes	No	4.1	787	800	Buffet
2	San Churro Cafe	Yes	No	3.8	918	800	Buffet
3	Addhuri Udupi Bhojana	No	No	3.7	88	300	Buffet
4	Grand Village	No	No	3.8	166	600	Buffet

In [15]:

```
sns.countplot(x=df['listed_in(type)'])
plt.xlabel("type of restaurant")
```

Out[15]:

```
Text(0.5, 0, 'type of restaurant')
```



Conclusion: majority of the restaurant falls in dinning category

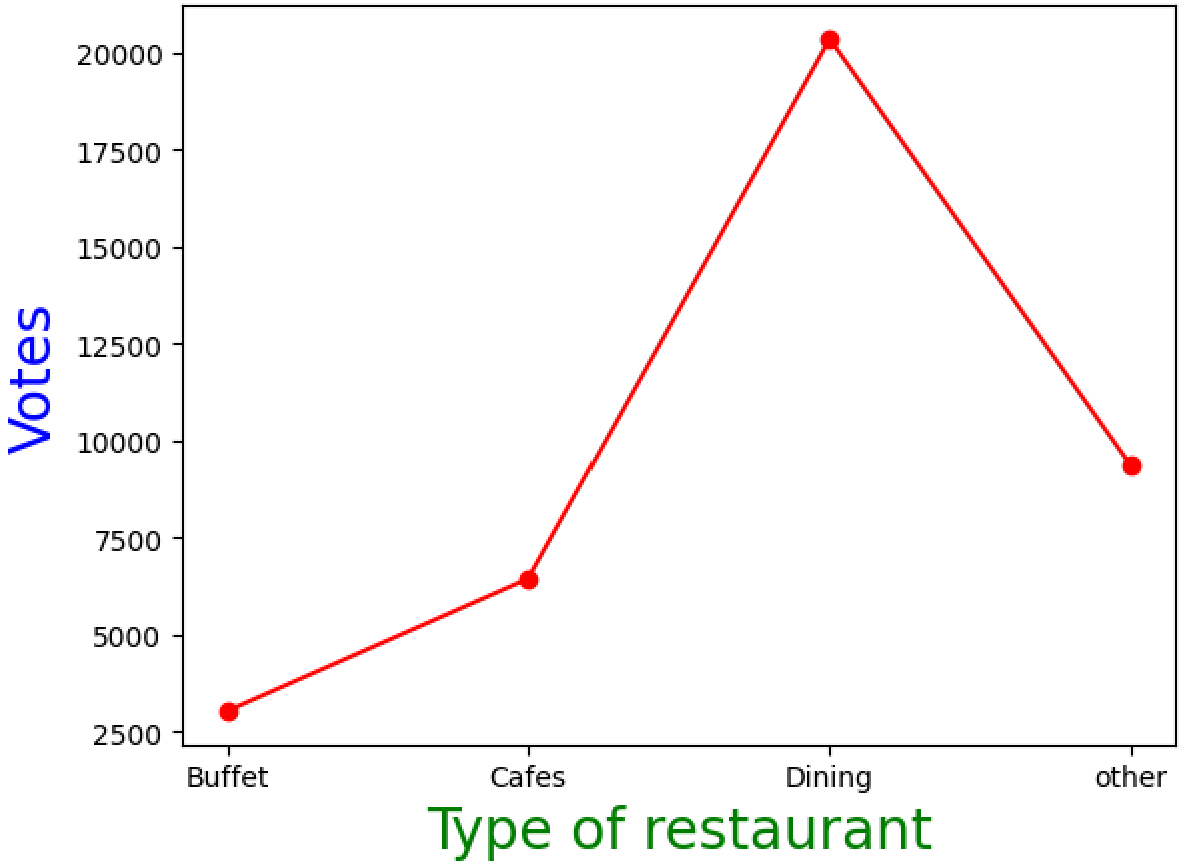
Group by restaurant type and sum of votes(line graph)

In [20]:

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

Out[20]:

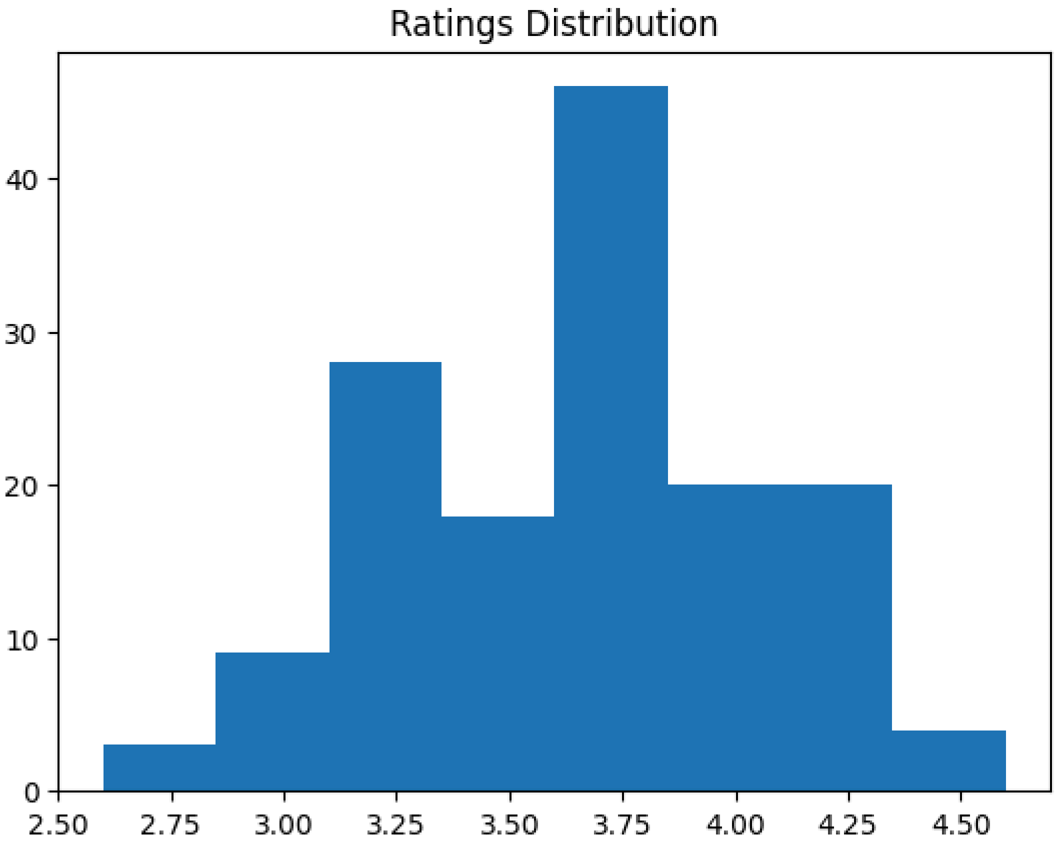
```
Text(0, 0.5, 'Votes')
```



Conclusion: dinning restaurants has received most votes

Histogram showing ratings distributions

In [29]:
```
plt.hist(df['rate'], bins=8)
plt.title("Ratings Distribution")
plt.show()
```



Conclusion: majority rating is 3.75

Average spending in each order

In [30]:

```
df.head()
```

Out[30]:

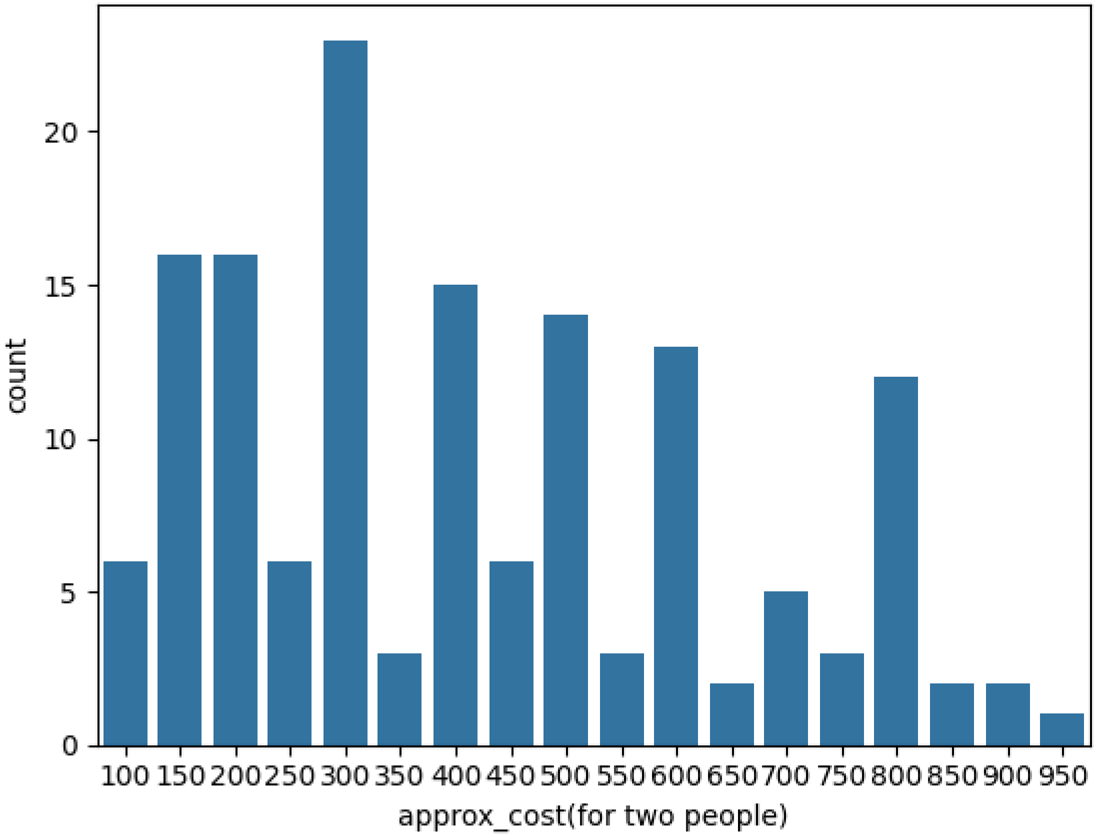
	name	online_order	book_table	rate	votes	approx_cost(for two people)	listed_in(type)
0	Jalsa	Yes	Yes	4.1	775	800	Buffet
1	Spice Elephant	Yes	No	4.1	787	800	Buffet
2	San Churro Cafe	Yes	No	3.8	918	800	Buffet
3	Addhuri Udupi Bhojana	No	No	3.7	88	300	Buffet
4	Grand Village	No	No	3.8	166	600	Buffet

In [31]:

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

Out[31]:

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



Conclusion: the majority of people prefers restaurants with an approx cost of 300 rupees.

Highest rating between online and offline mode

In [32]:

```
df.head()
```

Out[32]:

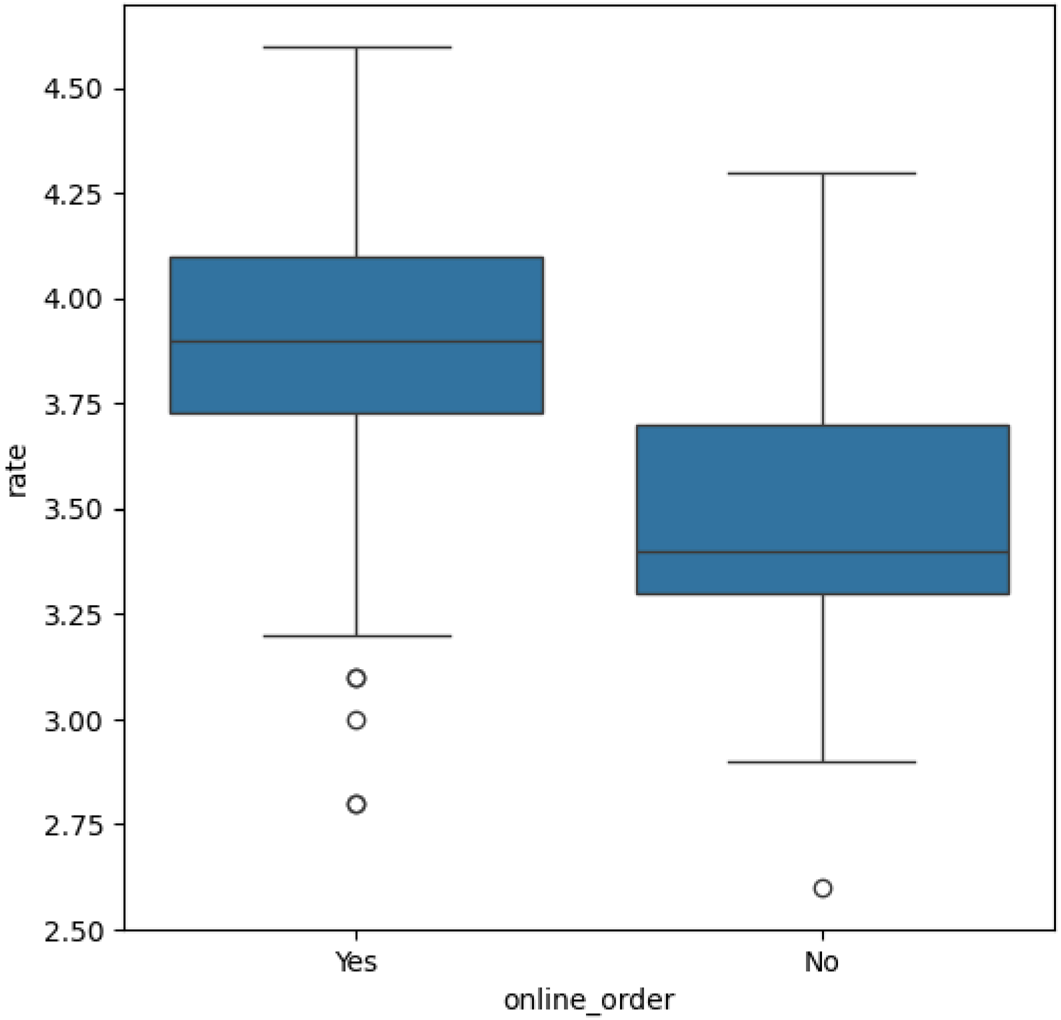
	name	online_order	book_table	rate	votes	approx_cost(for two people)	listed_in(type)
0	Jalsa	Yes	Yes	4.1	775	800	Buffet
1	Spice Elephant	Yes	No	4.1	787	800	Buffet
2	San Churro Cafe	Yes	No	3.8	918	800	Buffet
3	Addhuri Udupi Bhojana	No	No	3.7	88	300	Buffet
4	Grand Village	No	No	3.8	166	600	Buffet

In [37]:

```
plt.figure(figsize=(6,6))
sns.boxplot(x='online_order', y='rate', data=df)
```

Out[37]:

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



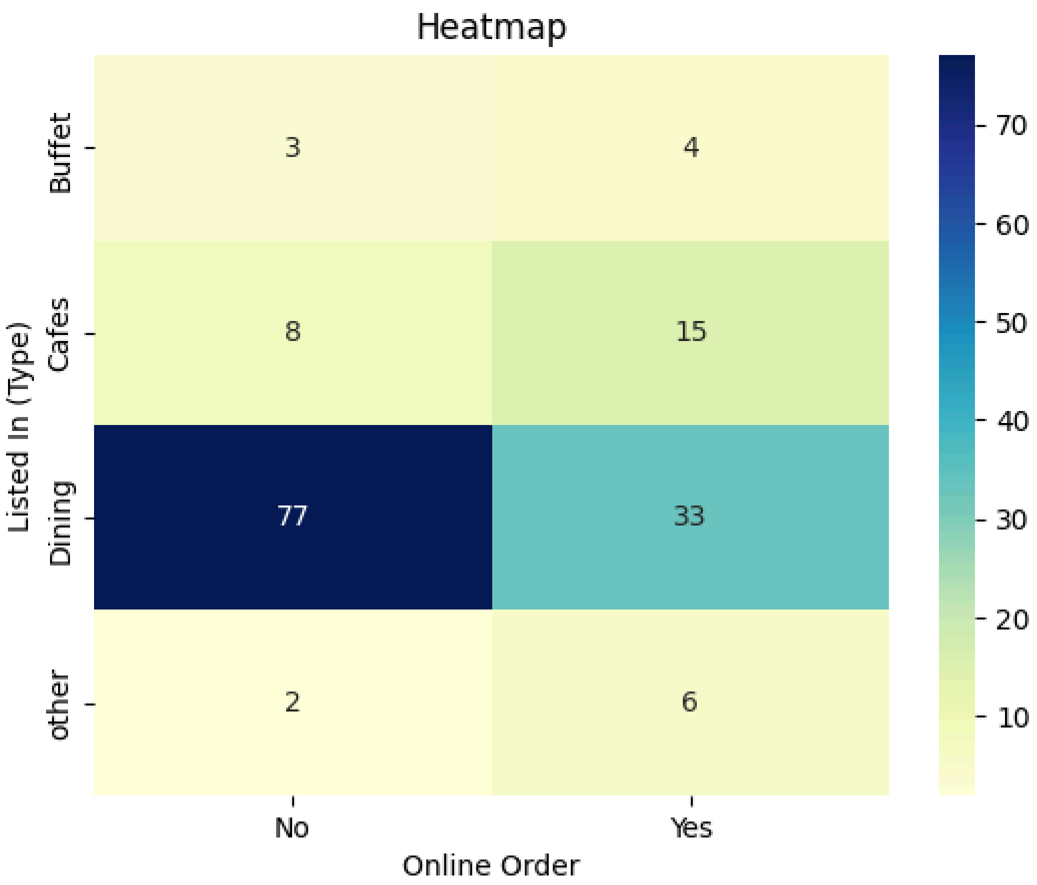
Conclusion: Offline orders received low rating in comparison to online.

Heatmap

In [38]:

```
pivot_table = df.pivot_table(index="listed_in(type)", columns="online_order", aggfunc='size')

# Heatmap plot karte hain
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.

In [ ]: