>importing necessary python libraries.

In [14]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

> Creating the data frame.

In [16]:
```
dataframe = pd.read_csv("zomato_data.csv")
print(dataframe.head())
```

```
                  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

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

>Data Cleaning and Preparation

In [18]:
```
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
```

>Getting summary of the dataframe use df.info().

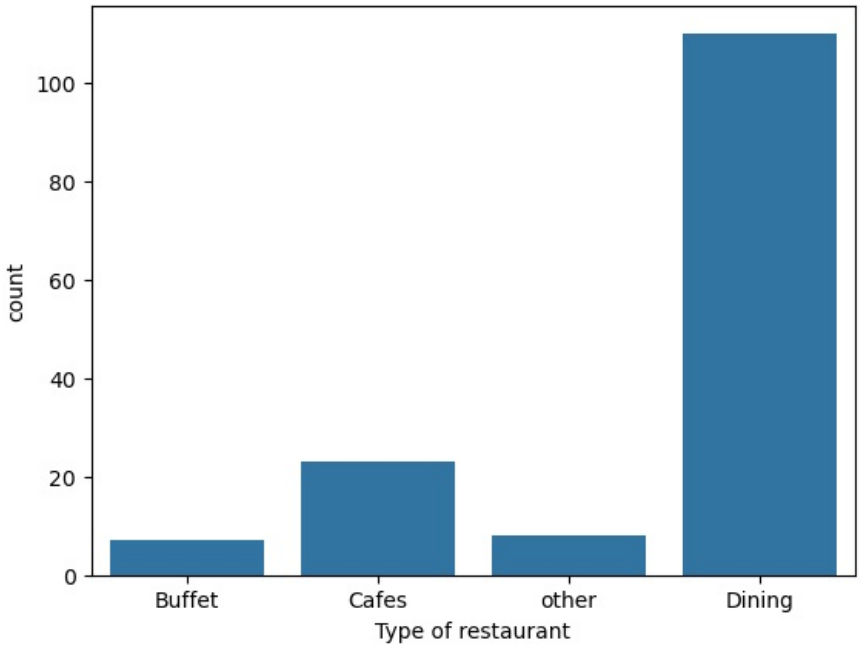
In [20]:
```
dataframe.info()
```

```
<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
```

>Exploring Restaurant Types

In [27]:
```
sns.countplot(x=dataframe['listed_in(type)'])
plt.xlabel("Type of restaurant")
```

Out[27]: Text(0.5, 0, 'Type of restaurant')

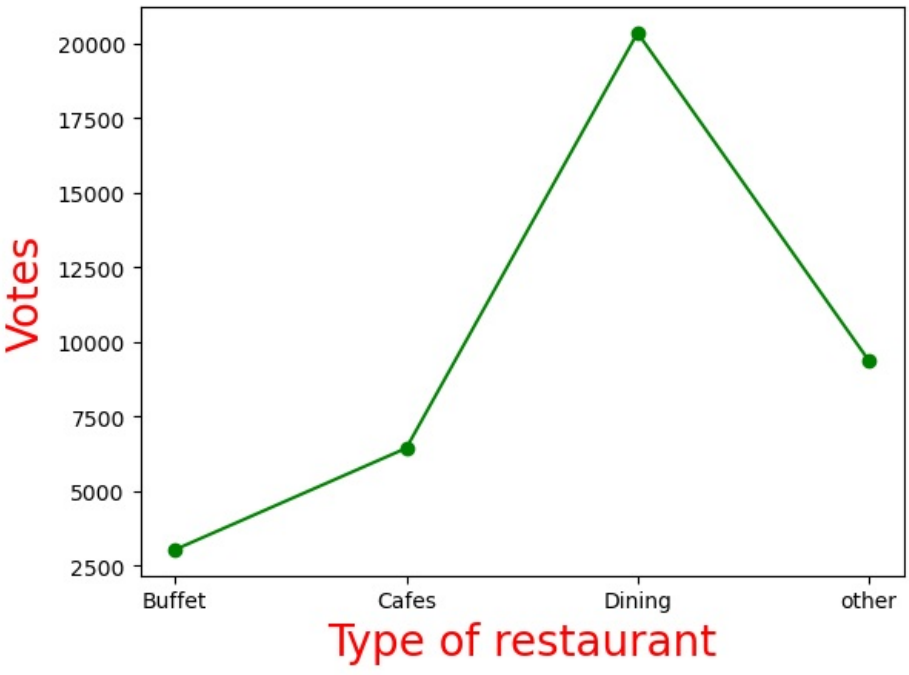


>Votes by Restaurant Type

Here we get the count of votes for each category.

In [30]:
```
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 of restaurant', c='red', size=20)
plt.ylabel('Votes', c='red', size=20)
```

Out[30]: Text(0, 0.5, 'Votes')



>Identifing the Most Voted Restaurant

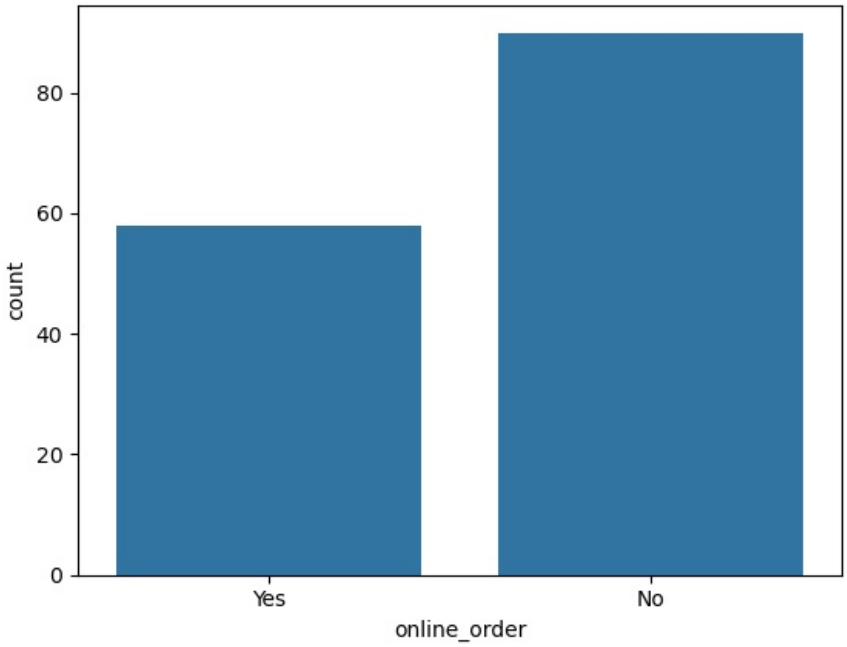
In [34]:
```
max_votes = dataframe['votes'].max()
restaurant_with_max_votes = dataframe.loc[dataframe['votes'] == max_votes, 'name']

print('Restaurant(s) with the maximum votes:')
print(restaurant_with_max_votes)
```

```
Restaurant(s) with the maximum votes:
38    Empire Restaurant
Name: name, dtype: object
```

>Online Order Availability

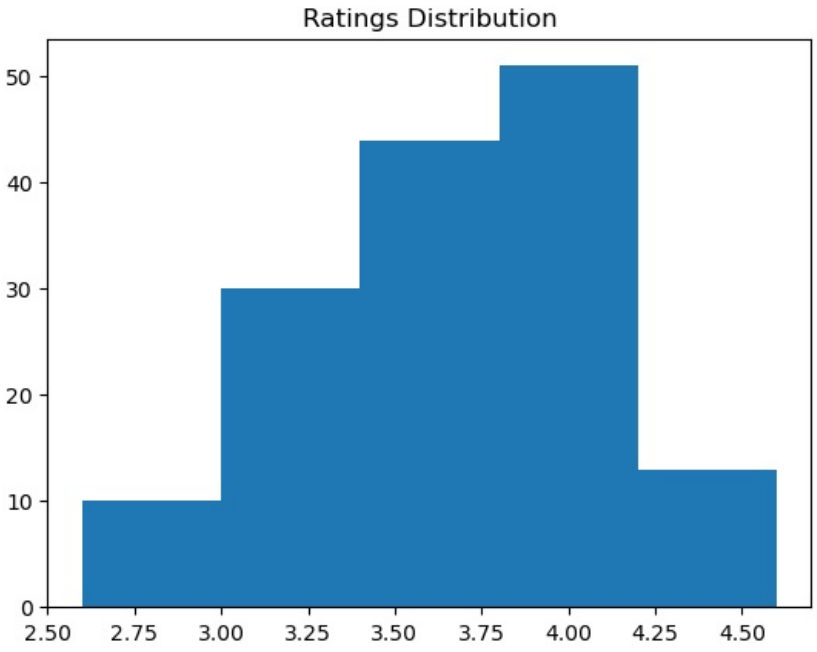
In [37]:
```
sns.countplot(x=dataframe['online_order'])
```

Out[37]:
```
<Axes: xlabel='online_order', ylabel='count'>
```



>Analyzing the Ratings

In [40]:
```
plt.hist(dataframe['rate'],bins=5)
plt.title('Ratings Distribution')
plt.show()
```

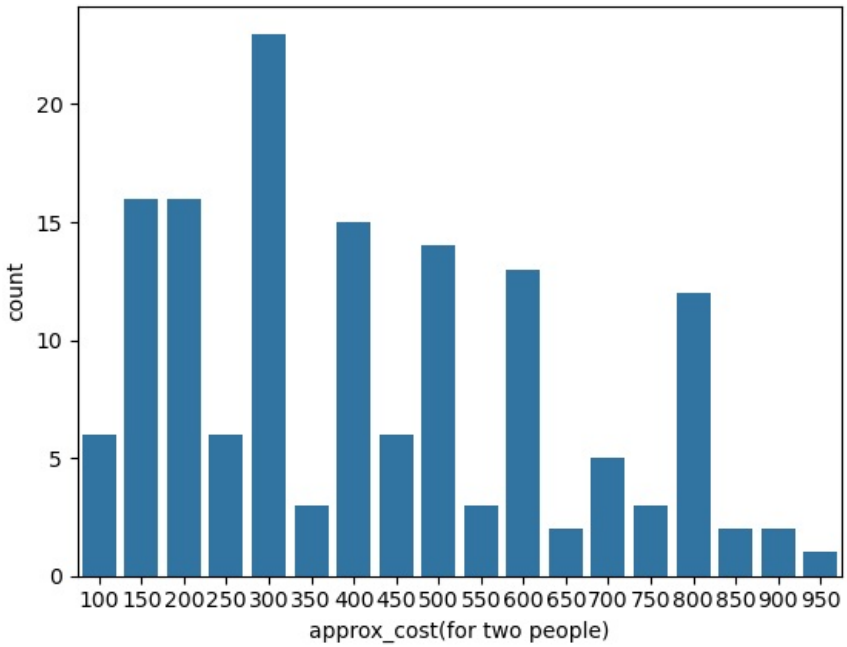


here we can observe that the majority of restaurants received ratings ranging from 3.5 to 4.

>Approximate Cost for Couples

In [44]:
```
couple_data=dataframe['approx_cost(for two people)']
sns.countplot(x=couple_data)
```

Out[44]:
```
<Axes: xlabel='approx_cost(for two people)', ylabel='count'>
```

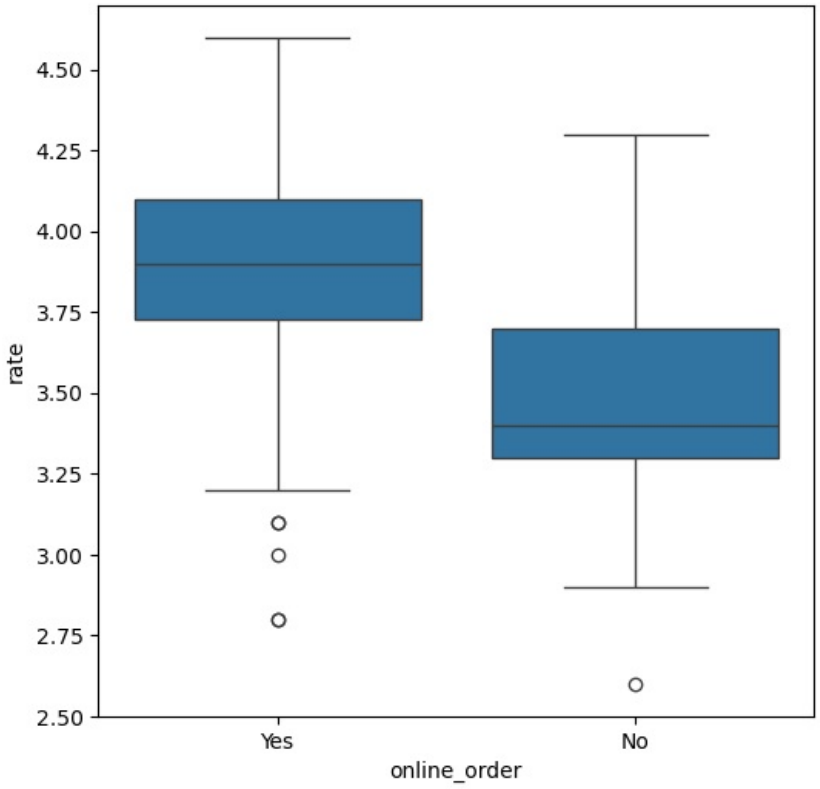


As per calculations Average cost for couple (two people) is Rs.300

>Ratings Comparison - Online vs Offline Orders

In [48]:
```
plt.figure(figsize = (6,6))
sns.boxplot(x = 'online_order', y = 'rate', data = dataframe)
```

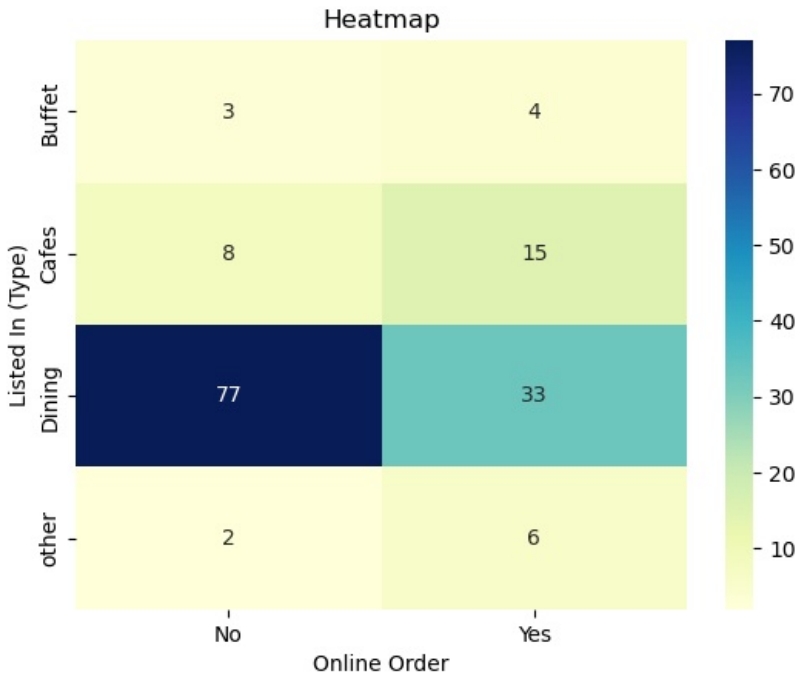
Out[48]:
```
<Axes: xlabel='online_order', ylabel='rate'>
```



from the above figure we conclude that Offline orders received lower ratings in comparison to online orders which obtained excellent ratings.

>Order Mode Preferences by Restaurant Type

In [52]:
```
pivot_table = dataframe.pivot_table(index='listed_in(type)', columns='online_order', aggfunc='size', fill_value=
sns.heatmap(pivot_table, annot=True, cmap='YlGnBu', fmt='d')
plt.title('Heatmap')
plt.xlabel('Online Order')
plt.ylabel('Listed In (Type)')
plt.show()
```



the final conclusion is 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 [ ]: