Zomato Data Analysis Using Python

Step-1: Importing necessary Python libraries

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

pandas is used for data manipulation and analysis. numpy is used for numerical operations. matplotlib.pyplot and seaborn are used for data visualization.

step-2:Loading the dataset

In [5]:
```python
dataframe = pd.read_csv ("C://Users//kbabu//Downloads//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
```

Step-3: Creating the data frame

In [6]:
```python
dataframe
```

Out[6]:

	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

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

In [13]:
```python
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

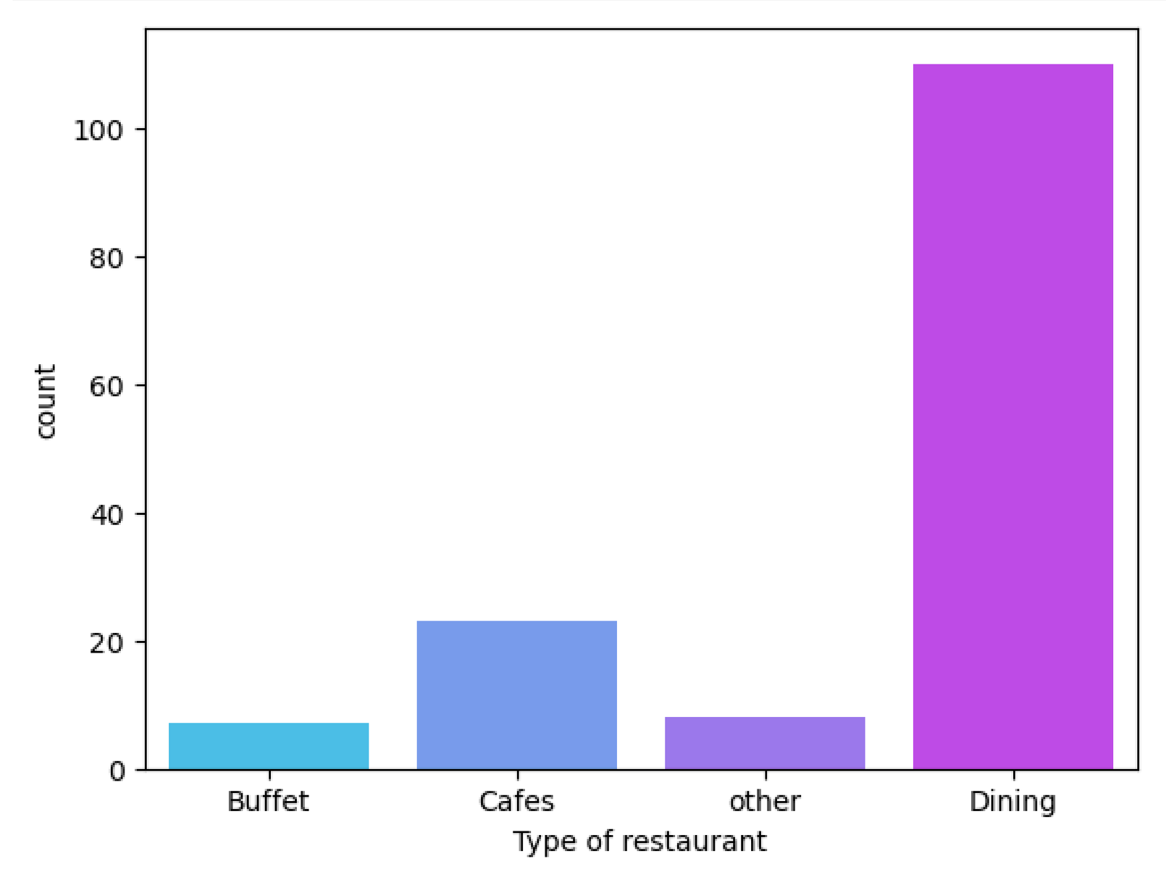
In [14]:
```python
dataframe.info()
```

```
<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 dataframe.

Type of Resturant

In [65]:
```python
sns.countplot(x=dataframe['listed_in(type)'], palette='cool')
plt.xlabel("Type of restaurant")
plt.show()
```

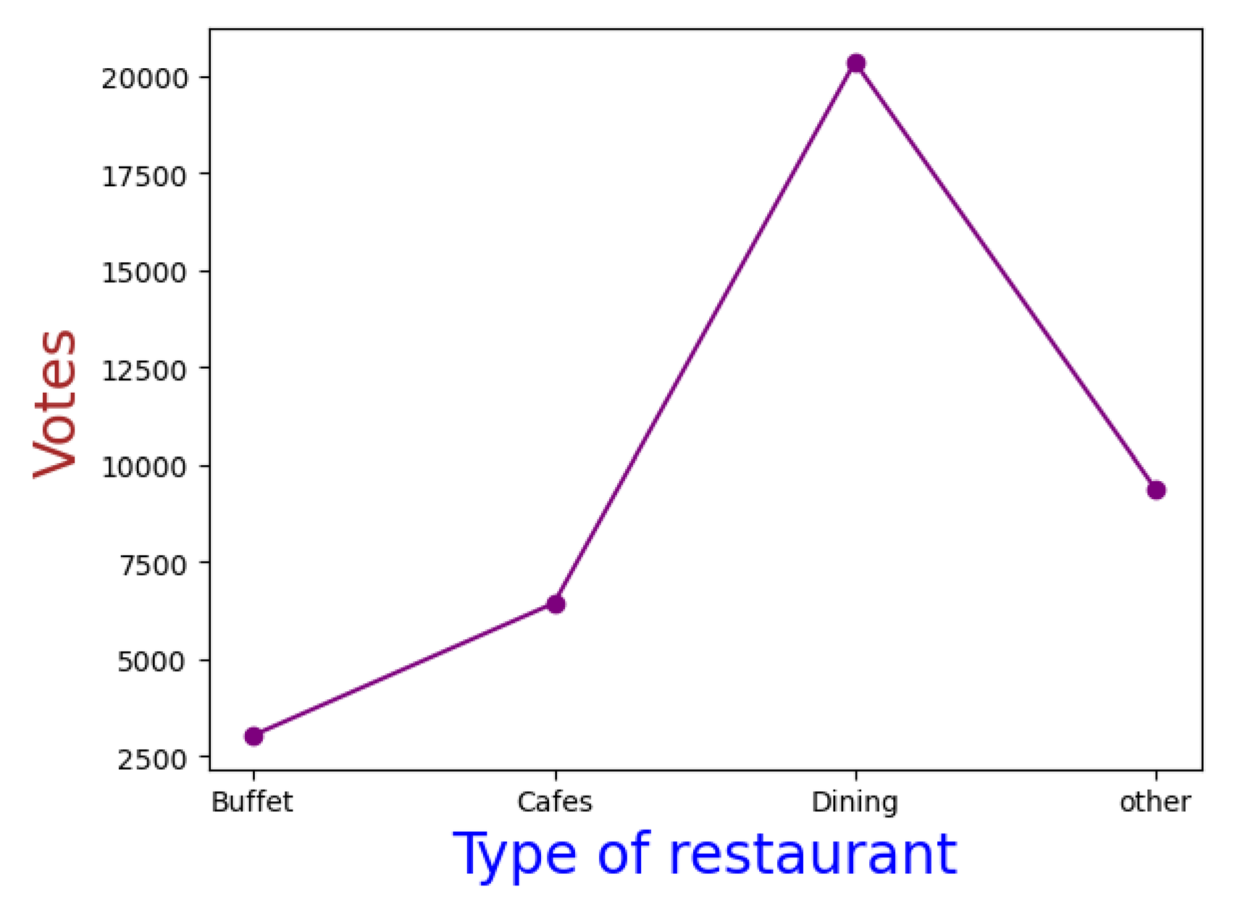


1. What are the ratings that the majority of restaurants have received?

Conclusion: The majority of the restaurants fall into the dining category.

In [57]:
```python
grouped_data= dataframe.groupby('listed_in(type)')['votes'].sum()
result=pd.DataFrame({'votes':grouped_data})
plt. plot(result, c="purple", marker="o")
plt.xlabel("Type of restaurant", c="blue",size=20)
plt.ylabel("Votes", c="brown", size=20)
```

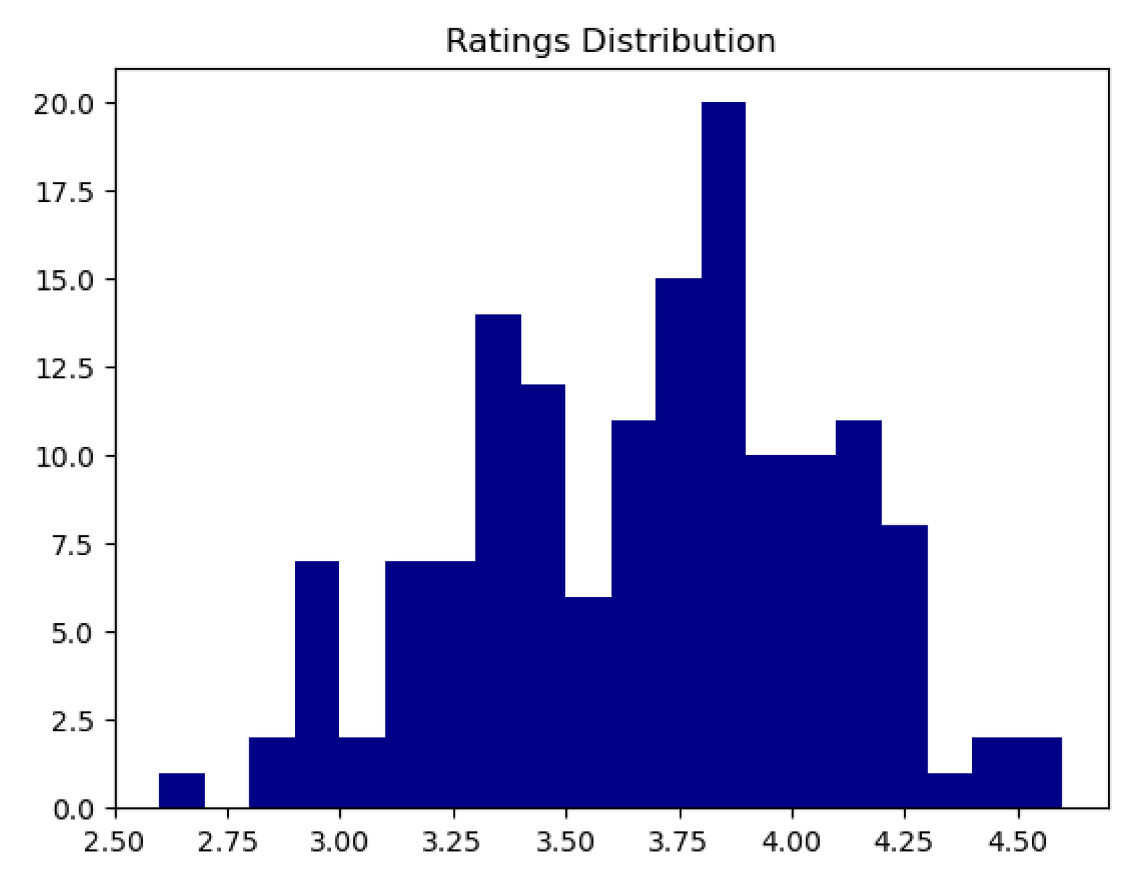
Out[57]:
```
Text(0, 0.5, 'Votes')
```



2 How many votes has each type of restaurant received from customers?

Conclusion: Dining restaurants are preferred by a larger number of individuals

In [56]:
```python
plt.hist(dataframe['rate'] , bins=20, color= 'darkblue')
plt.title("Ratings Distribution")
plt.show()
```

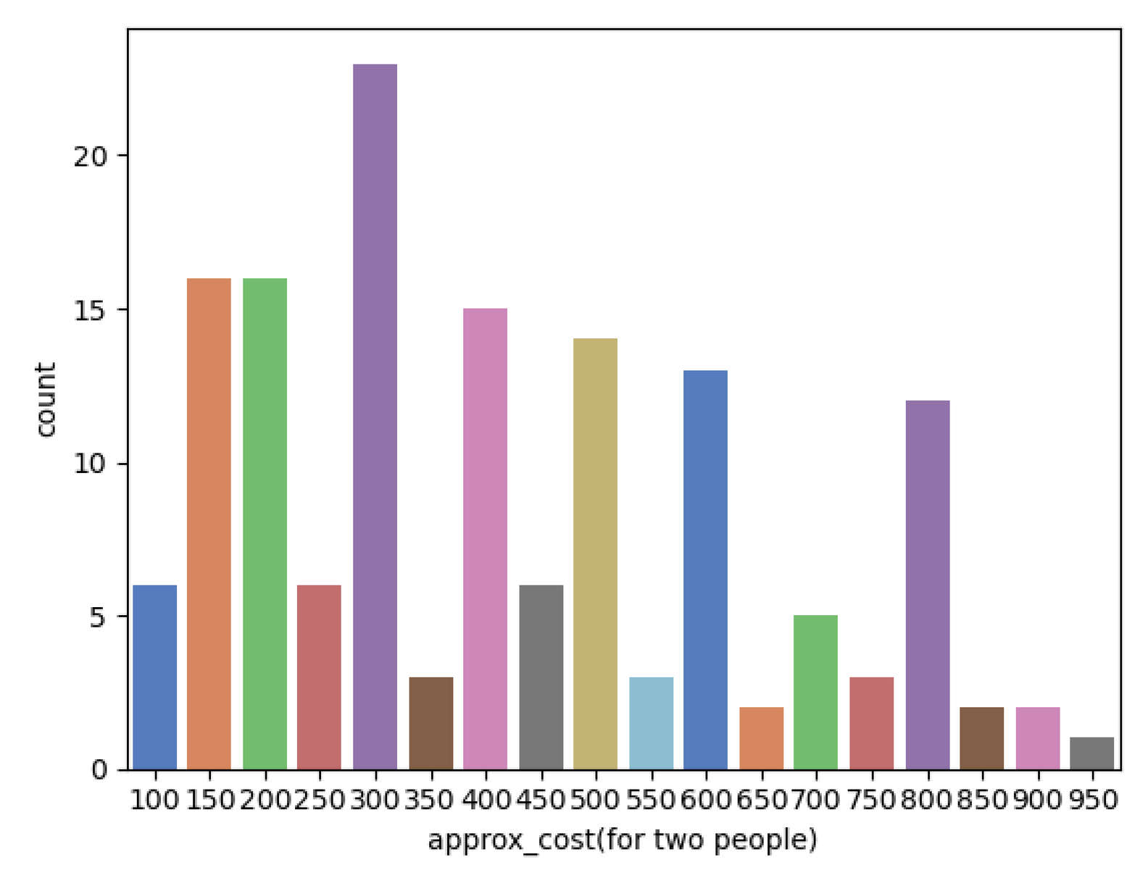


3) What are the ratings that the majority of restaurants have received?

Conclusion: The majority of restaurants received ratings ranging from 3.5 to 4

In [33]:
```python
couple_data=dataframe['approx_cost(for two people)']
sns.countplot(x=couple_data, palette='muted')
```

Out[33]:
```
<Axes: xlabel='approx_cost(for two people)', ylabel='count'>
```

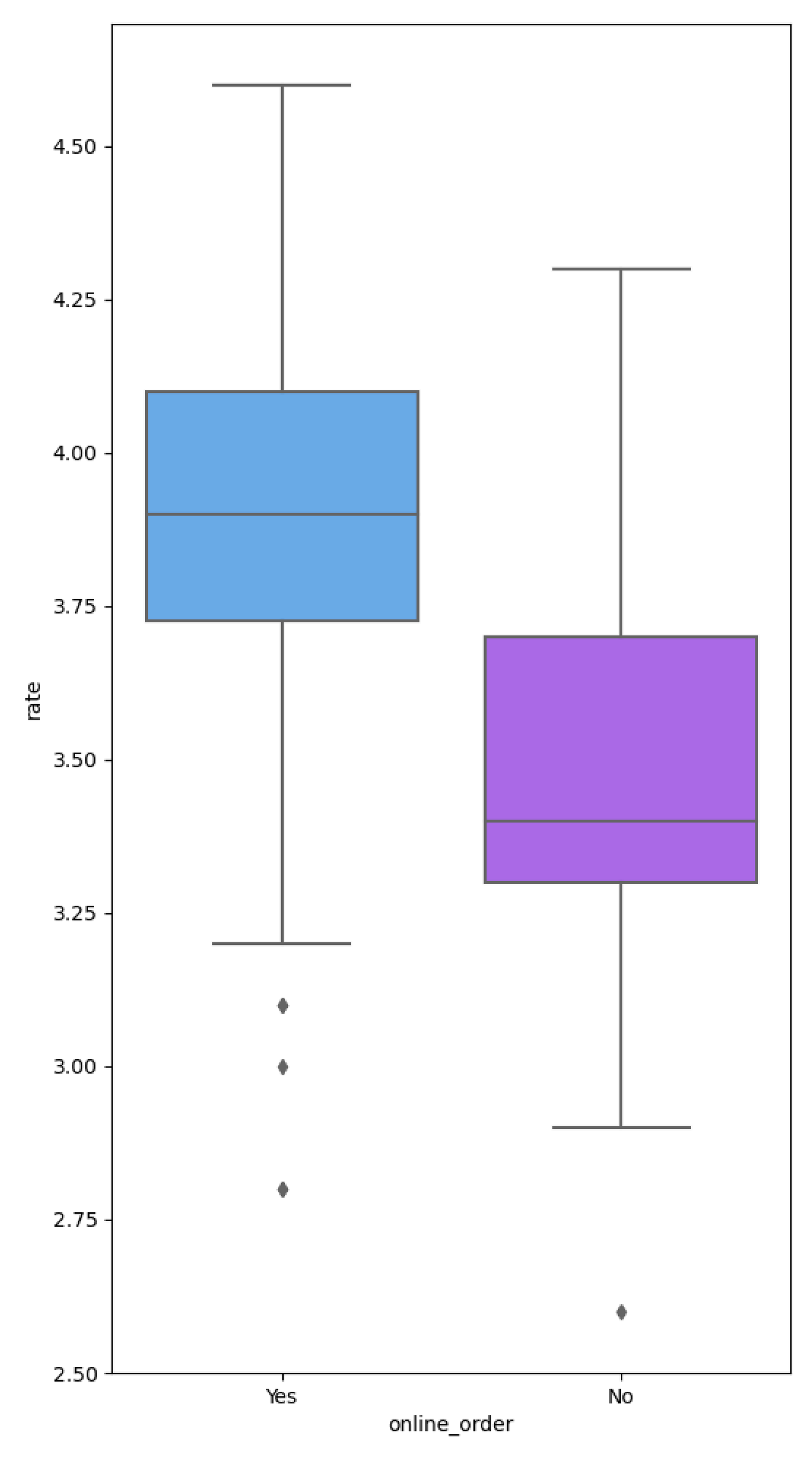


4)Zomato has observed that most couples order most of their food online. What is their average spending on each order?

The majority of couples prefer restaurants with an approximate cost of 300 rupees

In [39]:
```python
plt.figure(figsize = (6,12))
sns.boxplot (x='online_order', y ='rate',palette='cool', data = dataframe)
```

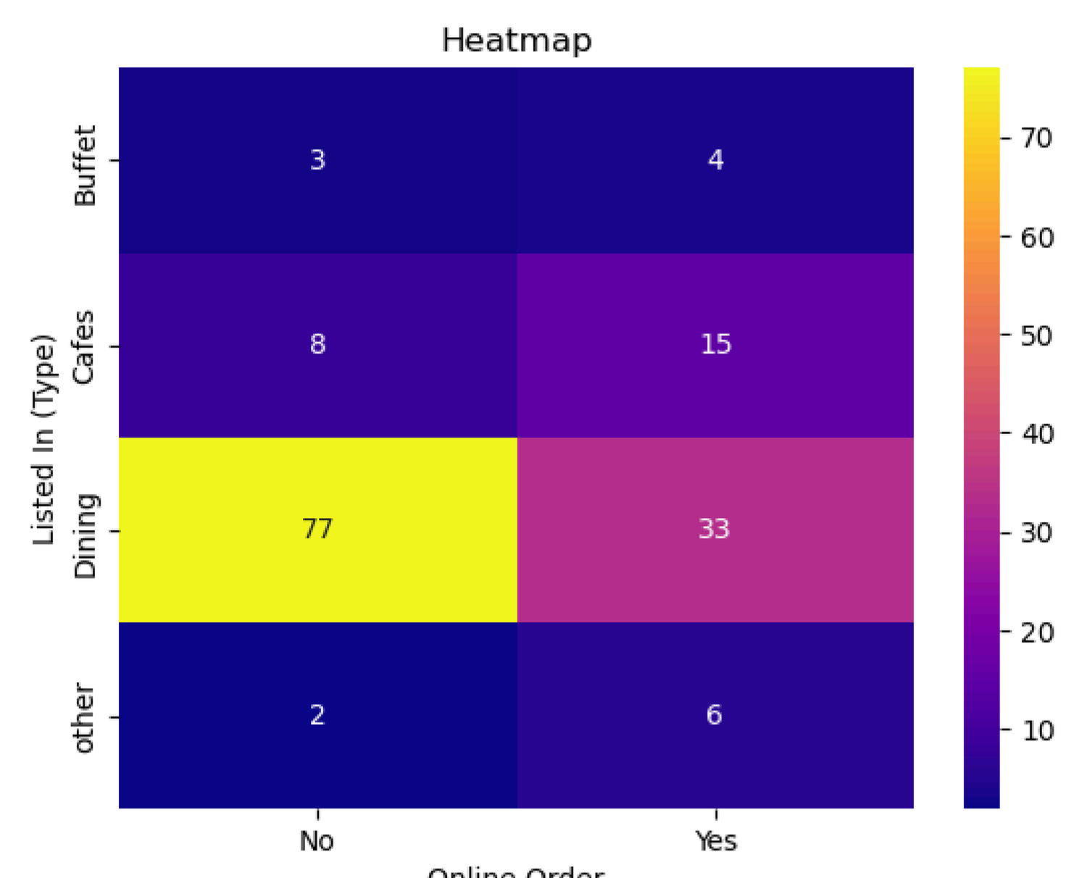
Out[39]:
```
<Axes: xlabel='online_order', ylabel='rate'>
```



5) Which mode (online or offline) has received the maximum rating?

CONCLUSION: Offline orders received lower ratings in comparison to online orders, which obtained excellent ratings

In [50]:
```python
pivot_table = dataframe.pivot_table(index='listed_in(type)', columns='online_order', aggfunc='size', fill_value=0)
sns.heatmap(pivot_table, annot=True, cmap="plasma", fmt='d')
plt.title( "Heatmap")
plt.xlabel("Online Order")
plt.ylabel("Listed In (Type)")
plt.show( )
```



6) Which type of restaurant received more offline orders, so that Zomato can provide customers with some good offers?

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 [ ]: