Zomato Data Analysis by Python

Project By Ayush Kumar

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

Start coding or generate with AI.

```
dataframe = pd.read_csv("/content/Zomato data .csv")
```

```
dataframe.iloc[10:101, 0:4]
```

	name	online_order	book_table	rate
10	Village Café	Yes	No	4.1/5
11	Cafe Shuffle	Yes	Yes	4.2/5
12	The Coffee Shack	Yes	Yes	4.2/5
13	Caf-Eleven	No	No	4.0/5
14	San Churro Cafe	Yes	No	3.8/5
...	...	...	...	...
96	Kaggis	No	No	3.8/5
97	Ayda Persian Kitchen	No	No	3.7/5
98	Chatar Patar	No	No	3.7/5
99	Polar Bear	Yes	No	3.8/5
100	Kidambi's Kitchen	No	No	3.5/5

91 rows × 4 columns

Start coding or generate with AI.

Create the dataframe

```
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]
```

dataframe

	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

Next steps: Generate code with dataframe | View recommended plots | New interactive sheet

Start coding or generate with AI.

convert the data type of column-rate

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

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

```
                     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
..                     ...          ...        ...   ...    ...
143       Melting Melodies           No         No   3.3      0
144        New Indraprasta           No         No   3.3      0
145           Anna Kuteera          Yes         No   4.0    771
146                 Darbar           No         No   3.0     98
147          Vijayalakshmi          Yes         No   3.9     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]
```

```
# This is formatted as code
```

```
dataframe.head()
```

	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

Next steps: Generate code with dataframe | View recommended plots | New interactive sheet

```
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
```

Start coding or generate with AI.

Type of Resturant

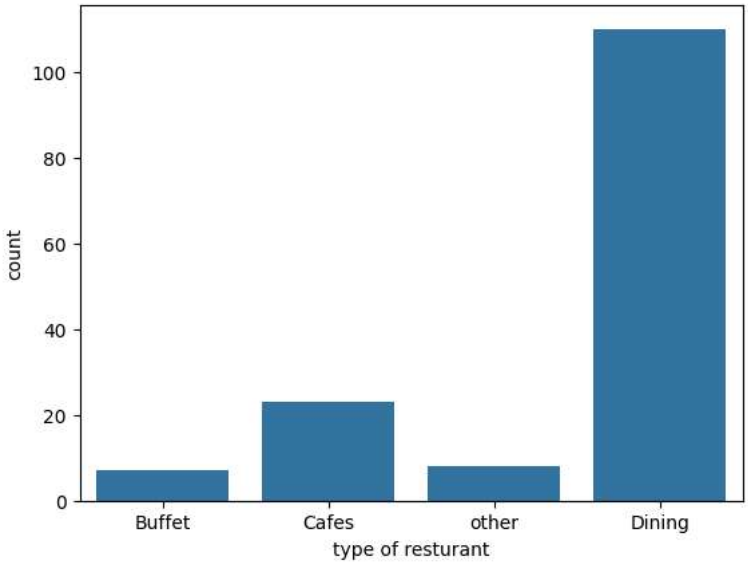
```
dataframe.head()
```

	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

Next steps: Generate code with dataframe | View recommended plots | New interactive sheet

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

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



Start coding or generate with AI.

Conclusion: Majority of the resturant falls in dinning Category.

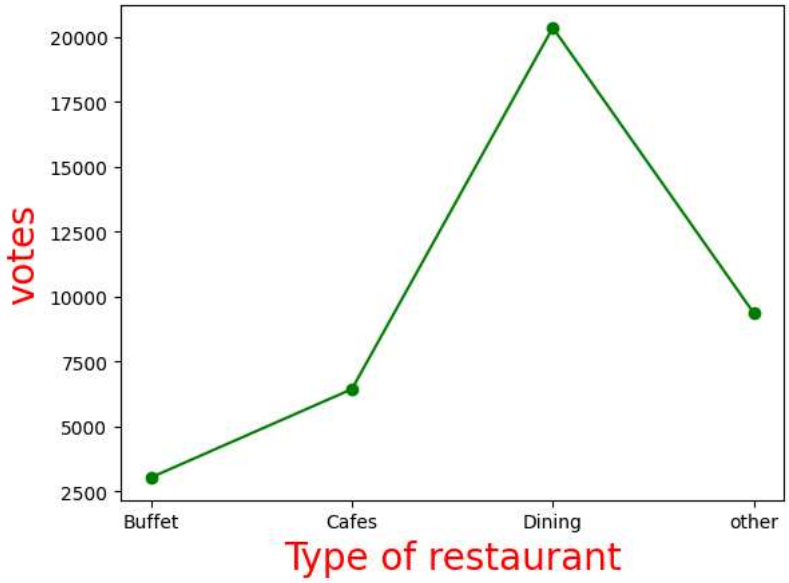
```
dataframe.head()
```

	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

Next steps: Generate code with dataframe | View recommended plots | New interactive sheet

```
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)
```

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



Start coding or generate with AI.

Conclusion: Dinning restaurants has recieved maximum votes.

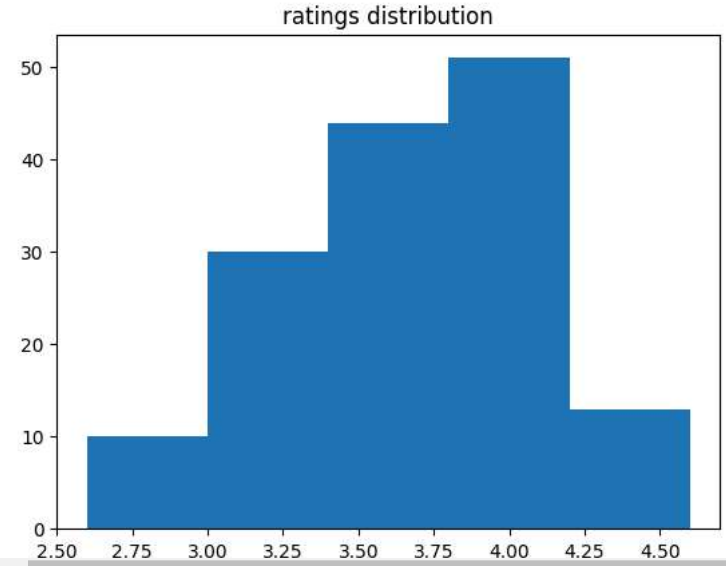
```
dataframe.head()
```

	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

Next steps: Generate code with dataframe | View recommended plots | New interactive sheet

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

```
Text(0.5, 1.0, 'ratings distribution')
```



Start coding or generate with AI.

Conclusion: The majority restaurant recieved ratings from 3.6 to 6.

```
dataframe.head()
```

	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

Next steps: Generate code with dataframe | View recommended plots | New interactive sheet

Start coding or generate with AI.

Average Order Spending by couples

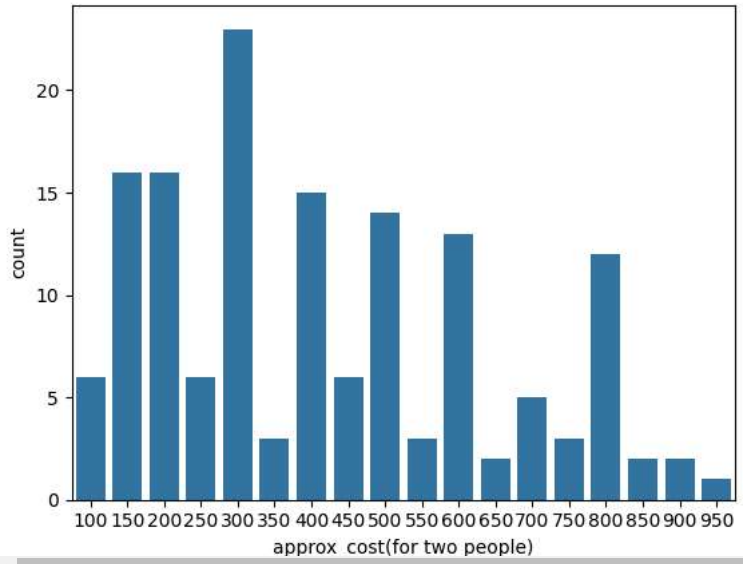
```
dataframe.head()
```

	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

Next steps: Generate code with dataframe | View recommended plots | New interactive sheet

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

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



Start coding or generate with AI.

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

Start coding or generate with AI.

Which mode recieves maximim rating

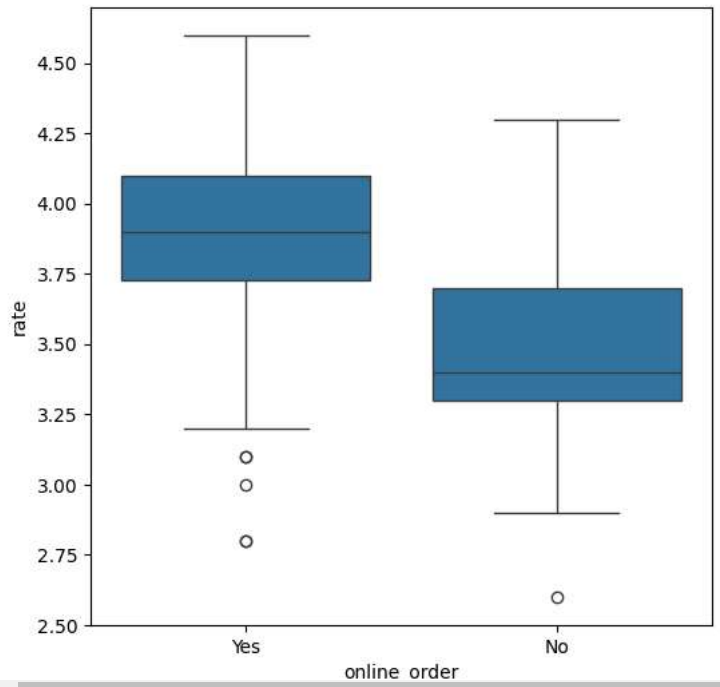
```
dataframe.head()
```

	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

Next steps: Generate code with dataframe | View recommended plots | New interactive sheet

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

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



Start coding or generate with AI.

Conclusion: Offline order recieved lower rating in comparision to online order.

```
dataframe.head()
```

	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

Next steps: Generate code with dataframe | View recommended plots | New interactive sheet

```
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()
```

+ Code + Text

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.

Start coding or generate with AI.