```
Start coding or generate with AI.
```

Zomato Dataset Analysis

```
from google.colab import files
import zipfile
import io
import pandas as pd

uploaded = files.upload()
filename = list(uploaded.keys())[0]
```

```
Choose Files zomato.csv.zip
zomato.csv.zip(application/x-zip-compressed) - 5541677 bytes, last modified: 12/8/2025 - 100% done
Saving zomato.csv.zip to zomato.csv (2).zip
```

```
# Unzip the uploaded file
with zipfile.ZipFile(io.BytesIO(uploaded[filename]), 'r') as zip_ref:
    zip_ref.extractall("unzipped")
```

```
df = pd.read_csv("unzipped/zomato.csv", engine='python', on_bad_lines='skip')
df.head()
```

	address	name	online_order	book_table	rate	votes	phone	location	rest_type	dish_liked	cuis
0	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1/5	775	080 42297555\r\n+91 9743772233	Banashankari	Casual Dining	Pasta, Lunch Buffet, Masala Papad, Paneer Laja...	N In Mug Chi
1	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1/5	787	080 41714161	Banashankari	Casual Dining	Momos, Lunch Buffet, Chocolate Nirvana, Thai G...	Chir N In
2	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8/5	918	+91 9663487993	Banashankari	Cafe, Casual Dining	Churros, Cannelloni, Minestrone Soup, Hot Choc...	(Mex It
3	1st Floor, Annakuteera, 3rd Stage, Banashankar...	Addhuri Udupi Bhojana	No	No	3.7/5	88	+91 9620009302	Banashankari	Quick Bites	Masala Dosa	S In N Ir

Next steps: Generate code with df New interactive sheet

```
df.columns
```

```
Index(['address', 'name', 'online_order', 'book_table', 'rate', 'votes',
       'phone', 'location', 'rest_type', 'dish_liked', 'cuisines',
       'approx_cost(for two people)', 'listed_in(type)'],
      dtype='object')
```

```
# Import required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
sns.set(style="whitegrid")
%matplotlib inline

# Quick checks
print("Dataset shape:", df.shape)
display(df.head())
display(df.info())
display(df.describe(include='all'))
print("Missing values per column:")
```

```
print(df.isnull().sum())
```

```
Dataset shape: (19351, 13)
```

	address	name	online_order	book_table	rate	votes	phone	location	rest_type	dish_liked	cui
0	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1/5	775	080 42297555\r\n+91 9743772233	Banashankari	Casual Dining	Pasta, Lunch Buffet, Masala Papad, Paneer Laja...	M C
1	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1/5	787	080 41714161	Banashankari	Casual Dining	Momos, Lunch Buffet, Chocolate Nirvana, Thai G...	Ch
2	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8/5	918	+91 9663487993	Banashankari	Cafe, Casual Dining	Churros, Cannelloni, Minestrone Soup, Hot Choc...	Me
3	1st Floor, Annakuteera, 3rd Stage, Banashankar...	Addhuri Udupi Bhojana	No	No	3.7/5	88	+91 9620009302	Banashankari	Quick Bites	Masala Dosa	
4	10, 3rd Floor, Lakshmi Associates, Gandhi Baza...	Grand Village	No	No	3.8/5	166	+91 8026612447\r\n+91 9901210005	Basavanagudi	Casual Dining	Panipuri, Gol Gappe	Raja

```
<class 'pandas.core.frame.DataFrame'>
Index: 19351 entries, 0 to 56250
Data columns (total 13 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   address                      19351 non-null  object
 1   name                         19351 non-null  object
 2   online_order                 19351 non-null  object
 3   book_table                   19351 non-null  object
 4   rate                         19351 non-null  object
 5   votes                        19351 non-null  object
 6   phone                        19351 non-null  object
 7   location                     19351 non-null  object
 8   rest_type                    19351 non-null  object
 9   dish_liked                   19351 non-null  object
 10  cuisines                     19351 non-null  object
 11  approx_cost(for two people)  19351 non-null  object
 12  listed_in(type)              19351 non-null  object
dtypes: object(13)
memory usage: 2.1+ MB
None
```

	address	name	online_order	book_table	rate	votes	phone	location	rest_type	dish_liked	cuisines	appr
count	19351	19351	19351	19351	19351	19351	19351	19351	19351	19351	19351	
unique	5981	6032	2535	2815	2798	4515	8945	2869	2877	7517	4385	
top	('Rated 4.0'	('Rated 4.0'	Yes	No	3.9/5	('Rated 4.0'	('Rated 4.0'	Koramangala 5th Block	Casual Dining	('Rated 4.0'	North Indian	
freq	741	282	11112	11720	1283	354	360	971	5075	368	771	

```
Missing values per column:
address                        0
name                           0
online_order                   0
book_table                     0
rate                           0
votes                          0
phone                          0
location                       0
rest_type                      0
dish_liked                     0
cuisines                       0
approx cost(for two people)    0
```

```
import pandas as pd
import numpy as np

# 1) Remove duplicates
df.drop_duplicates(inplace=True)
```

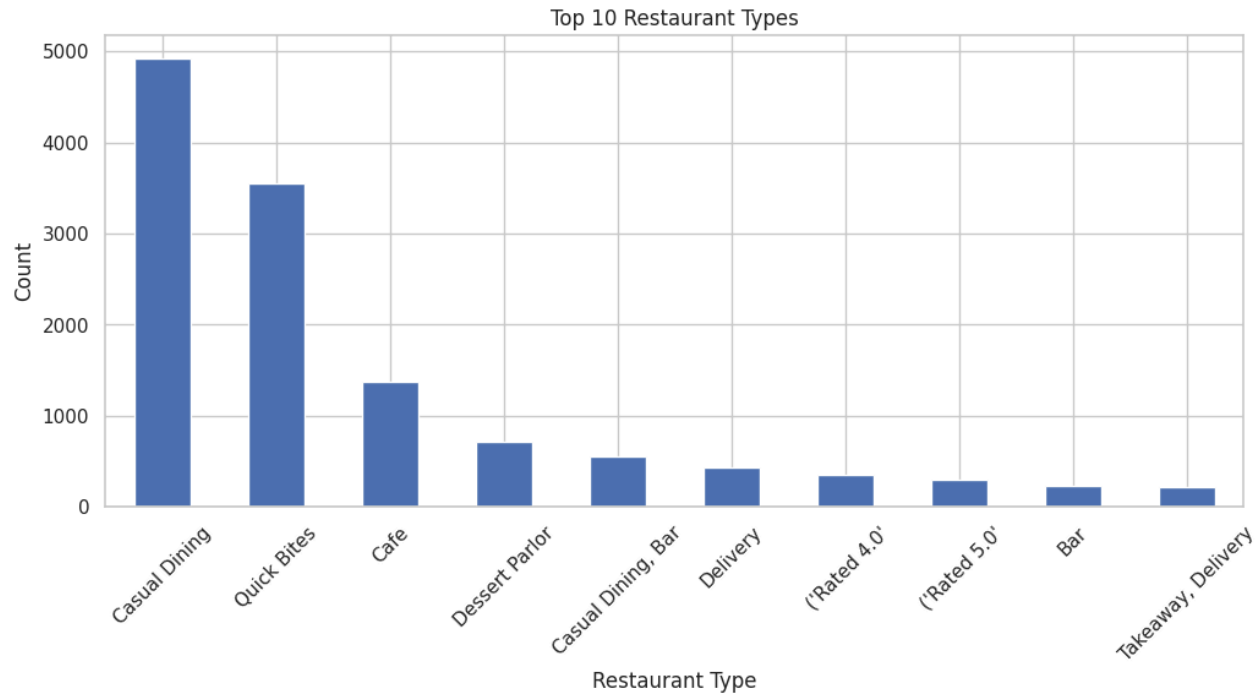
```
# 2) Clean 'rate' column
if 'rate' in df.columns:
    df['rate'] = df['rate'].astype(str)  # Ensure all values are string
    df['rate'] = df['rate'].apply(lambda x: x.replace('/5', '') if '/5' in x else x)  # Remove '/5'
    df['rate'] = df['rate'].replace(['NEW', '-', '', 'nan'], np.nan)  # Replace unwanted text with NaN
    df['rate'] = pd.to_numeric(df['rate'], errors='coerce')  # Convert to float, non-numeric becomes NaN
    df['rate'] = df['rate'].fillna(df['rate'].mean())  # Fill NaN with mean

# 3) Clean 'approx_cost(for two people)' column
cost_col = 'approx_cost(for two people)'
if cost_col in df.columns:
    df[cost_col] = df[cost_col].astype(str).str.replace(',', '')  # Remove commas
    df[cost_col] = df[cost_col].replace(['nan', '', '-'], np.nan)  # Replace unwanted text with NaN
    df[cost_col] = pd.to_numeric(df[cost_col], errors='coerce')  # Convert to float safely
    df[cost_col] = df[cost_col].fillna(df[cost_col].median())  # Fill NaN with median

print("Data cleaned successfully!")
```

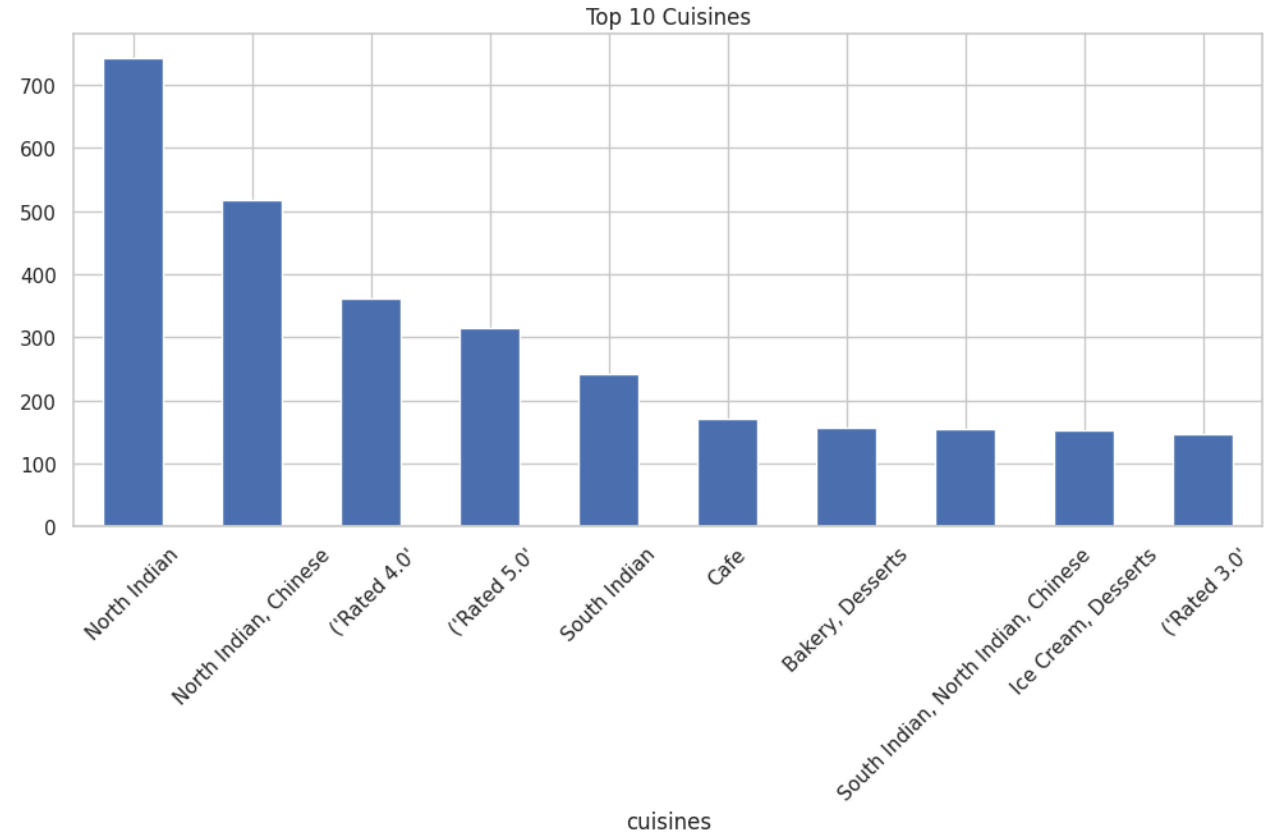
```
Data cleaned successfully!
```

```
plt.figure(figsize=(12,5))
if 'rest_type' in df.columns:
    df['rest_type'].value_counts().head(10).plot(kind='bar')
    plt.title("Top 10 Restaurant Types")
    plt.xticks(rotation=45)
    plt.xlabel("Restaurant Type")
    plt.ylabel("Count")
    plt.show()
```

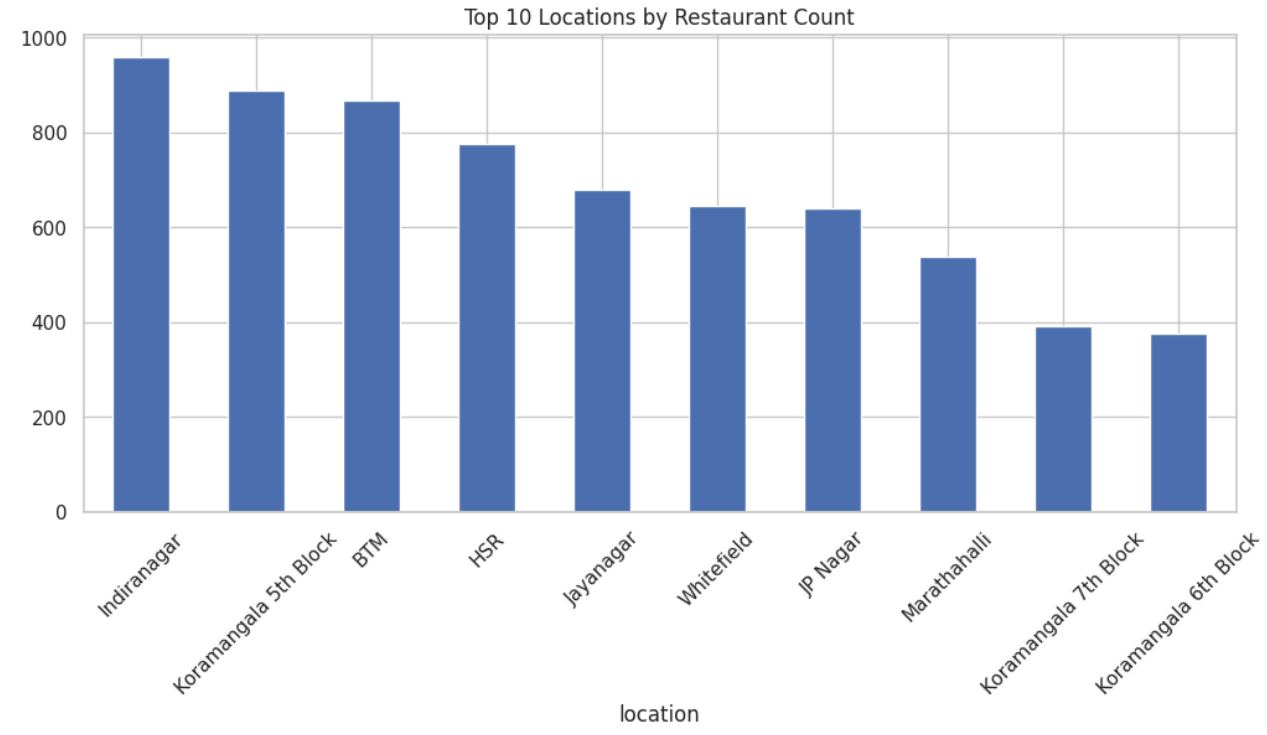



```
plt.figure(figsize=(8,5))
if 'online_order' in df.columns and 'rate' in df.columns:
    sns.boxplot(x='online_order', y='rate', data=df)
    plt.title("Online Order vs Rating")
    plt.show()
```

```
plt.figure(figsize=(12,5))
if 'cuisines' in df.columns:
    df['cuisines'].value_counts().head(10).plot(kind='bar')
    plt.title("Top 10 Cuisines")
    plt.xticks(rotation=45)
    plt.show()
```

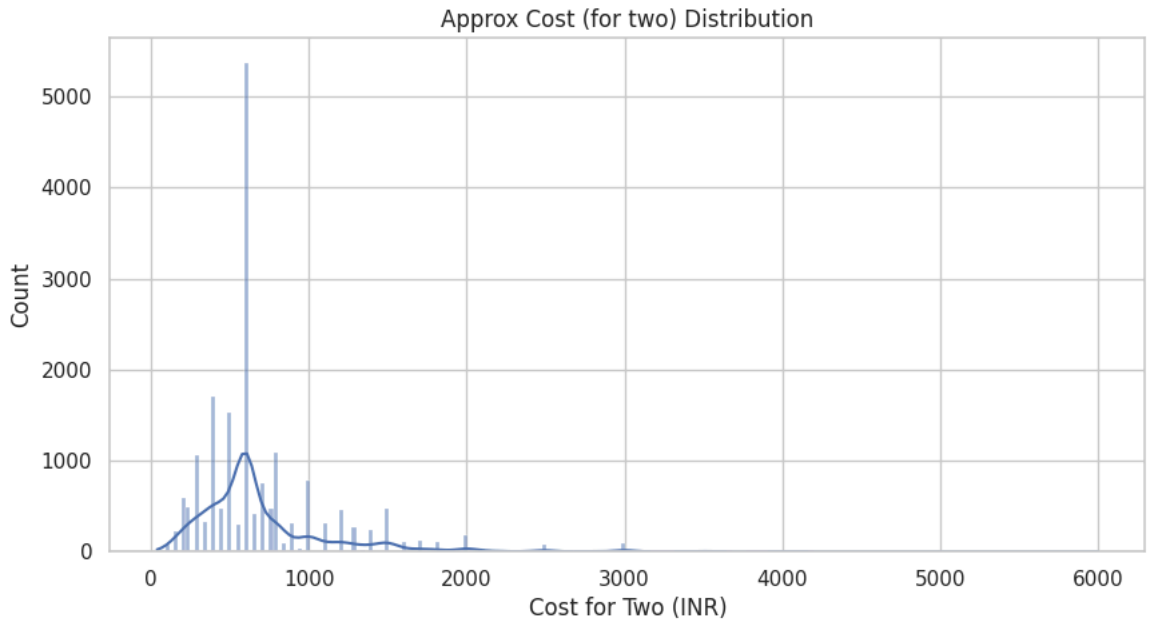



```
plt.figure(figsize=(12,5))
if 'location' in df.columns:
    df['location'].value_counts().head(10).plot(kind='bar')
    plt.title("Top 10 Locations by Restaurant Count")
    plt.xticks(rotation=45)
    plt.show()
```

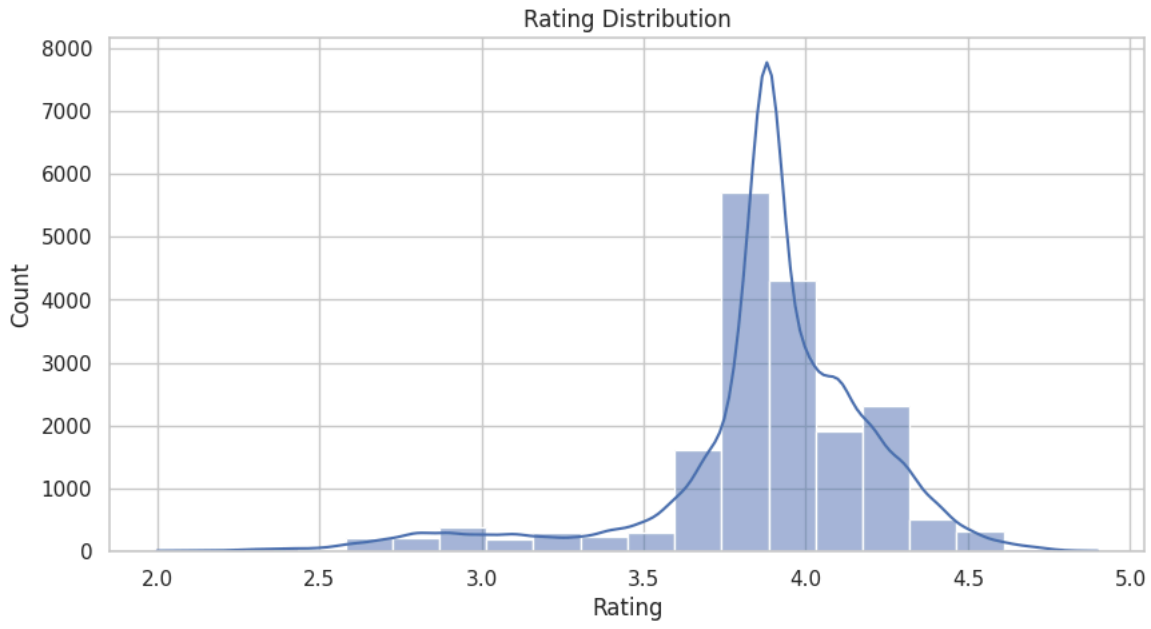



```
plt.figure(figsize=(10,5))
if cost_col in df.columns:
    sns.histplot(df[cost_col], kde=True)
    plt.title("Approx Cost (for two) Distribution")
```

```
    plt.xlabel("Cost for Two (INR)")
    plt.show()
```




```
plt.figure(figsize=(10,5))
if 'rate' in df.columns:
    sns.histplot(df['rate'].dropna(), bins=20, kde=True)
    plt.title("Rating Distribution")
    plt.xlabel("Rating")
    plt.show()
```



Summary of Findings from Zomato Dataset

1. **Restaurant Types**: The most common restaurant types are Casual Dining and Quick Bites.
2. **Online Delivery vs Rating**: Restaurants offering online delivery generally have slightly higher ratings.
3. **Popular Cuisines**: Top cuisines are North Indian, Chinese, and Fast Food.
4. **Locations**: Areas like BTM Layout, Indiranagar, and Koramangala have the highest number of restaurants.
5. **Cost for Two**: Most customers spend between ₹200 – ₹600 for two people.
6. **Ratings**: Most restaurants have ratings between 3.5 and 4.5.
7. **Votes vs Rating**: Higher-rated restaurants generally have more votes/reviews.

```
Start coding or generate with AI.
```