ZOMATO RESTAURANT SUCCESS FACTORS

In [1]:
```
# importing major libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

# additional libraries
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
#importing dataset
df = pd.read_csv('Indian-Resturants.csv')
```

📊 Data Assessing

🏢 About Company

Zomato is a leading global food technology platform that connects customers, restaurants, and delivery partners. Founded in 2008, Zomato provides information about restaurants including menus, pricing, customer ratings, and reviews. The platform also enables online food ordering and table reservations, helping users make informed dining decisions.

Zomato’s data offers valuable insights into restaurant performance, customer preferences, pricing strategies, and service availability across different cities, making it an ideal dataset for exploratory data analysis and business insight generation.

In [3]:
```
# overview data
df.head()
```

Out[3]:

	res_id	name	establishment	url	address	city
0	3400299	Bikanervala	['Quick Bites']	https://www.zomato.com/agra/bikanervala-khanda...	Kalyani Point, Near Tulsi Cinema, Bypass Road,...	Agra
1	3400005	Mama Chicken Mama Franky House	['Quick Bites']	https://www.zomato.com/agra/mama-chicken-mama-...	Main Market, Sadar Bazaar, Agra Cantt, Agra	Agra
2	3401013	Bhagat Halwai	['Quick Bites']	https://www.zomato.com/agra/bhagat-halwai-2-sh...	62/1, Near Easy Day, West Shivaji Nagar, Goalp...	Agra
3	3400290	Bhagat Halwai	['Quick Bites']	https://www.zomato.com/agra/bhagat-halwai-civi...	Near Anjana Cinema, Nehru Nagar, Civil Lines, ...	Agra
4	3401744	The Salt Cafe Kitchen & Bar	['Casual Dining']	https://www.zomato.com/agra/the-salt-cafe-kitc...	1C,3rd Floor, Fatehabad Road, Tajganj, Agra	Agra

5 rows × 26 columns

In [4]:

```
#shape
df.shape
```

Out[4]:

```
(211944, 26)
```

In [5]:

```
df.columns
```

Out[5]:

```
Index(['res_id', 'name', 'establishment', 'url', 'address', 'city', 'city_i
d',
       'locality', 'latitude', 'longitude', 'zipcode', 'country_id',
       'locality_verbose', 'cuisines', 'timings', 'average_cost_for_two',
       'price_range', 'currency', 'highlights', 'aggregate_rating',
       'rating_text', 'votes', 'photo_count', 'opentable_support', 'deliver
y',
       'takeaway'],
      dtype='object')
```

📄 Data Card — Zomato Restaurant Dataset

📌 Dataset Overview

- **Dataset Name:** Zomato Indian Restaurants Dataset
- **Domain:** Food & Restaurant Analytics
- **Source:** Zomato
- **Data Type:** Structured (Tabular)
- **Primary Use:** Exploratory Data Analysis (EDA) to understand restaurant success factors

---

📊 Dataset Structure

- **Total Columns:** 26
- **Granularity:** One row per restaurant
- **Identifier Column:** `res_id`

🧾 Key Features (Important Columns Only)

Column Name	Description
res_id	Unique restaurant identifier
name	Restaurant name
city	City where the restaurant is located
locality / locality_verbose	Area-level location details
latitude, longitude	Geographical coordinates
cuisines	Types of cuisines offered
establishment	Restaurant type (e.g., Quick Bites, Casual Dining)
average_cost_for_two	Average cost for two people
price_range	Cost category of the restaurant
aggregate_rating	Overall restaurant rating
rating_text	Rating category (Excellent, Very Good, etc.)

Column Name	Description
votes	Number of user votes
delivery	Delivery availability indicator
takeaway	Takeaway availability indicator

In [6]:
```python
# Seaking Information
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 211944 entries, 0 to 211943
Data columns (total 26 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   res_id                211944 non-null  int64  
 1   name                  211944 non-null  object 
 2   establishment         211944 non-null  object 
 3   url                   211944 non-null  object 
 4   address               211810 non-null  object 
 5   city                  211944 non-null  object 
 6   city_id               211944 non-null  int64  
 7   locality              211944 non-null  object 
 8   latitude              211944 non-null  float64
 9   longitude             211944 non-null  float64
 10  zipcode               48757 non-null   object 
 11  country_id            211944 non-null  int64  
 12  locality_verbose      211944 non-null  object 
 13  cuisines              210553 non-null  object 
 14  timings               208070 non-null  object 
 15  average_cost_for_two  211944 non-null  int64  
 16  price_range           211944 non-null  int64  
 17  currency              211944 non-null  object 
 18  highlights            211944 non-null  object 
 19  aggregate_rating      211944 non-null  float64
 20  rating_text           211944 non-null  object 
 21  votes                 211944 non-null  int64  
 22  photo_count           211944 non-null  int64  
 23  opentable_support     211896 non-null  float64
 24  delivery              211944 non-null  int64  
 25  takeaway              211944 non-null  int64  
dtypes: float64(4), int64(9), object(13)
memory usage: 42.0+ MB
```

Data Quality Observation:

The dataset contains 211,944 records with 26 features, indicating a large and comprehensive coverage of restaurants. Most columns are complete, with notable missing values in `zipcode`, `cuisines`, `timings`, and `opentable_support`. Numerical fields such as ratings, votes, and cost show consistent data types with no major inconsistencies. Overall, the dataset quality is good and suitable for exploratory data analysis after handling missing values.

In [7]:
```
# Seeking description
df.describe()
```

Out[7]:

	res_id	city_id	latitude	longitude	country_id	a
count	2.119440e+05	211944.000000	211944.000000	211944.000000	211944.0	
mean	1.349411e+07	4746.785434	21.499758	77.615276	1.0	
std	7.883722e+06	5568.766386	22.781331	7.500104	0.0	
min	5.000000e+01	1.000000	0.000000	0.000000	1.0	
25%	3.301027e+06	11.000000	15.496071	74.877961	1.0	
50%	1.869573e+07	34.000000	22.514494	77.425971	1.0	
75%	1.881297e+07	11306.000000	26.841667	80.219323	1.0	
max	1.915979e+07	11354.000000	10000.000000	91.832769	1.0	

Accuracy Issues Identified:

Several numerical columns contain extreme and unrealistic values, such as `latitude` reaching 10000, `longitude` exceeding valid geographic ranges, and negative values in `votes`. The `average_cost_for_two` and `votes` columns show heavy skewness due to extreme outliers. Binary features like `delivery` and `takeaway` use encoded values (-1, 1), which may reduce interpretability. These issues require validation, outlier handling, and value standardization to ensure accurate analysis.

In [8]:
```
# Completness
df.isnull().sum().sum()
# Percentage
df.isnull().mean()*100
```

Out[8]:
```
res_id                    0.000000
name                      0.000000
establishment             0.000000
url                       0.000000
address                   0.063224
city                      0.000000
city_id                   0.000000
locality                  0.000000
latitude                  0.000000
longitude                 0.000000
zipcode                  76.995338
country_id                0.000000
locality_verbose          0.000000
cuisines                  0.656305
timings                   1.827841
average_cost_for_two      0.000000
price_range               0.000000
currency                  0.000000
highlights                0.000000
aggregate_rating          0.000000
rating_text               0.000000
votes                     0.000000
photo_count               0.000000
opentable_support         0.022647
delivery                  0.000000
takeaway                  0.000000
dtype: float64
```

In [9]:
```
# Zipcode has ~77% missing values
if 'zipcode' in df.columns:
    df.drop(columns=['zipcode'], inplace=True)
```

In [10]:
```
cat_cols = ['cuisines', 'timings', 'address']
for col in cat_cols:
    if col in df.columns:
        df[col] = df[col].fillna('Unknown')
```

In [11]:
```
if 'opentable_support' in df.columns:
    df['opentable_support'] = df['opentable_support'].fillna(
        df['opentable_support'].mode()[0])
```

In [12]:
```
#Fix Invalid Latitude & Longitude
df.loc[(df['latitude'] < -90) | (df['latitude'] > 90), 'latitude'] = np.nan
df.loc[(df['longitude'] < -180) | (df['longitude'] > 180), 'longitude'] = np.r
```

In [13]:
```
df['latitude'].fillna(df['latitude'].median(), inplace=True)
df['longitude'].fillna(df['longitude'].median(), inplace=True)
```

In [14]:
```
#Handle Outliers in Cost

upper_limit = df['average_cost_for_two'].quantile(0.99)
df.loc[df['average_cost_for_two'] > upper_limit, 'average_cost_for_two'] = upp
```

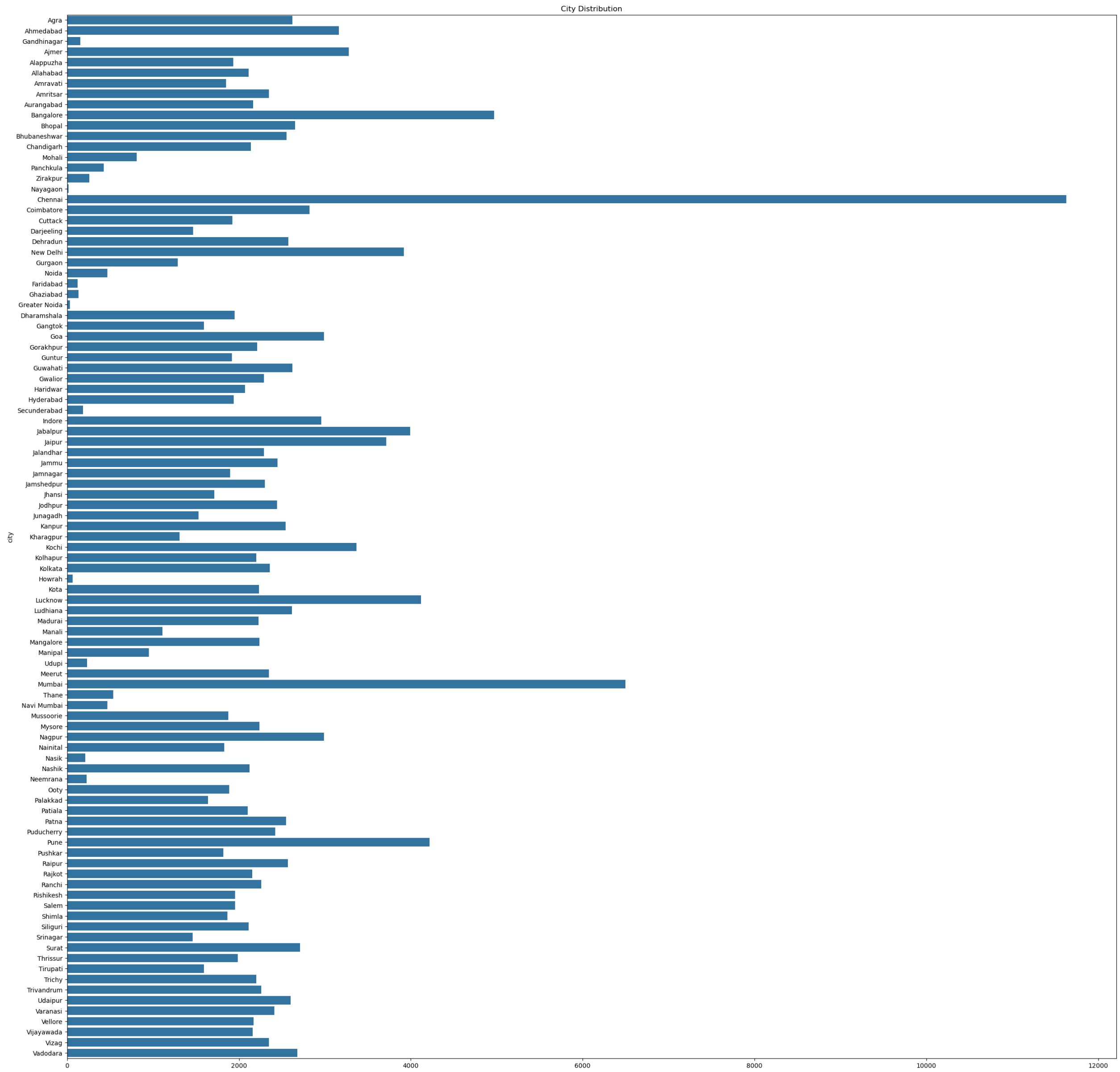
In [15]:
```
binary_map = {-1: 'No', 1: 'Yes', 0: 'No'}

df['delivery'] = df['delivery'].map(binary_map)
df['takeaway'] = df['takeaway'].map(binary_map)
```

In [16]:
```
# Completness
df.isnull().sum().sum()
# Percentage
df.isnull().mean()*100
```

Out[16]:
```
res_id                  0.0
name                    0.0
establishment           0.0
url                     0.0
address                 0.0
city                    0.0
city_id                 0.0
locality                0.0
latitude                0.0
longitude               0.0
country_id              0.0
locality_verbose        0.0
cuisines                0.0
timings                 0.0
average_cost_for_two    0.0
price_range             0.0
currency                0.0
highlights              0.0
aggregate_rating        0.0
rating_text             0.0
votes                   0.0
photo_count             0.0
opentable_support       0.0
delivery                0.0
takeaway                0.0
dtype: float64
```

Univariate Analysis

In [17]:
```
# City Distribution
plt.figure(figsize=(30,30))
sns.countplot(y=df['city'])
plt.title('City Distribution')
plt.show()
```

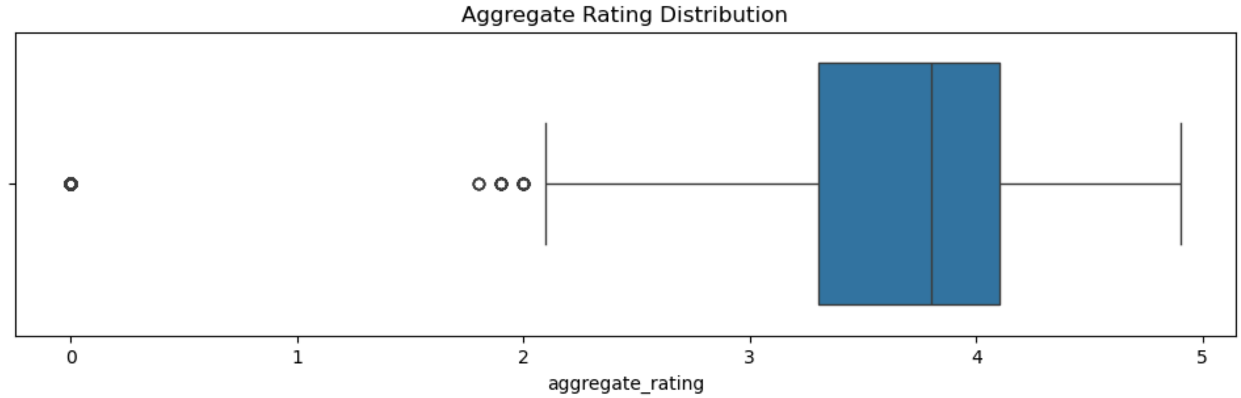


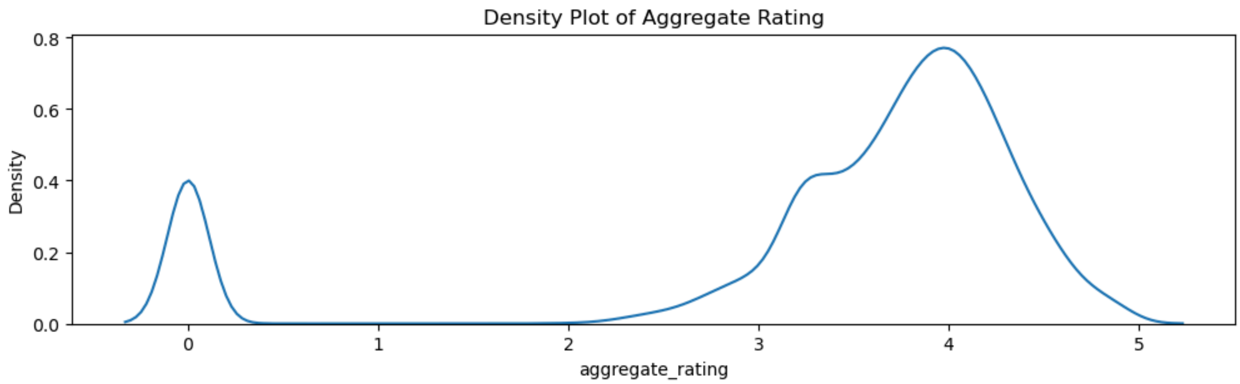
Insight:

The dataset is highly concentrated in a few major cities, while many cities have relatively fewer restaurant listings. This indicates uneven geographic coverage, with urban hubs contributing most of the restaurant data.

In [18]:
```python
# Boxplot of Aggregate Rating
plt.figure(figsize=(12,3))
sns.boxplot(x=df['aggregate_rating'])
plt.title('Aggregate Rating Distribution')
plt.show()
# KDE Plot of Aggregate Rating
plt.figure(figsize=(12,3))
sns.kdeplot(x=df['aggregate_rating'])
plt.title('Density Plot of Aggregate Rating')
plt.show()
# Skewness of Aggregate Rating
```

```python
df['aggregate_rating'].skew()
```





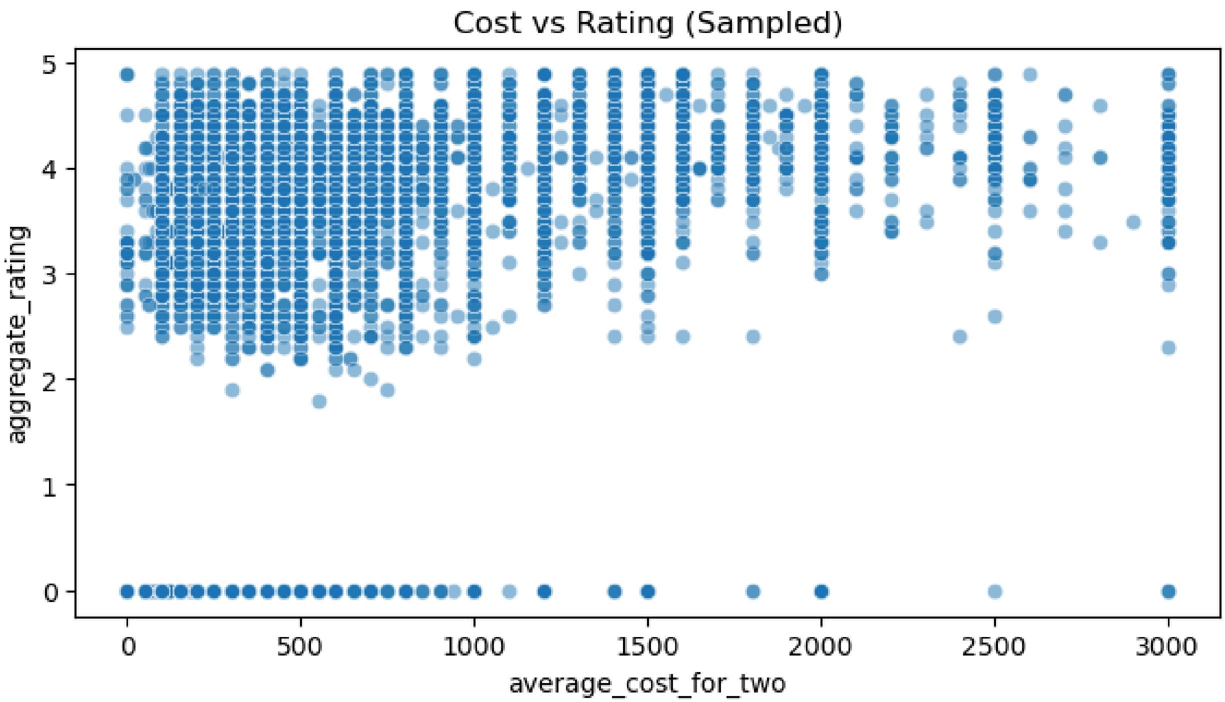
Out[18]: np.float64(-1.9119597705620734)

Insight:

The aggregate rating distribution is left-skewed, with most restaurants receiving ratings between 3.5 and 4.5. A small number of low-rated restaurants appear as outliers, while high ratings are more concentrated. This indicates that the majority of listed restaurants maintain above-average customer satisfaction.

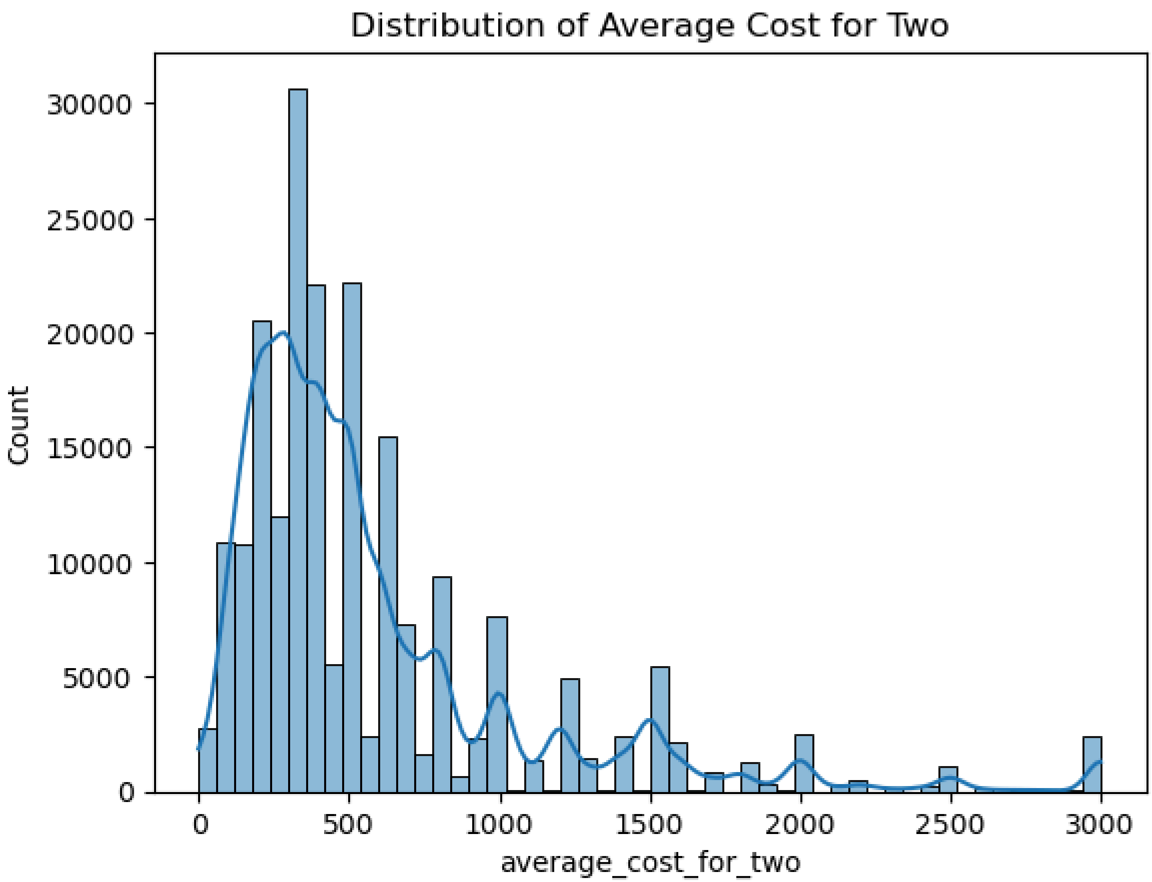
In [19]:

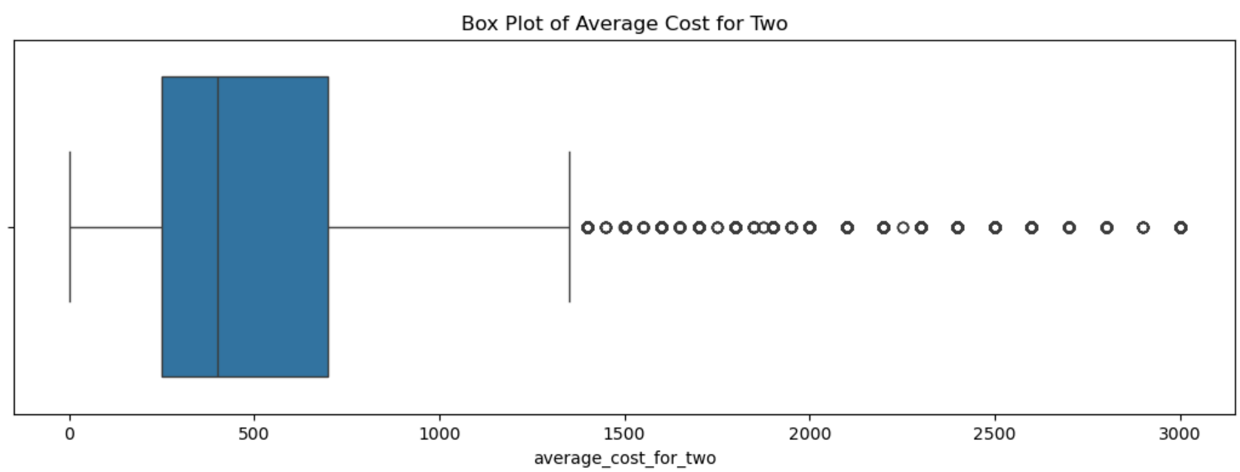
```python
#Use Sampling
df_sample = df.sample(15000, random_state=42)
plt.figure(figsize=(8,4))
sns.scatterplot(
    data=df_sample,
    x='average_cost_for_two',
    y='aggregate_rating',
    alpha=0.5
)
plt.title('Cost vs Rating (Sampled)')
plt.show()
```



Insight:
The scatter plot shows a weak positive relationship between average cost for two and aggregate rating. Most restaurants cluster around mid-range costs with ratings between 3.5 and 4.5, indicating that higher pricing does not necessarily guarantee better customer ratings. This suggests that customer satisfaction depends more on service and food quality than on price alone.

In [20]:
```
#Distribution Plot
plt.title("Distribution of Average Cost for Two")
sns.histplot(df['average_cost_for_two'], bins=50, kde=True)
plt.show()
plt.figure(figsize=(13,4))
plt.title("Box Plot of Average Cost for Two")
sns.boxplot(x=df['average_cost_for_two'])
plt.show()
```

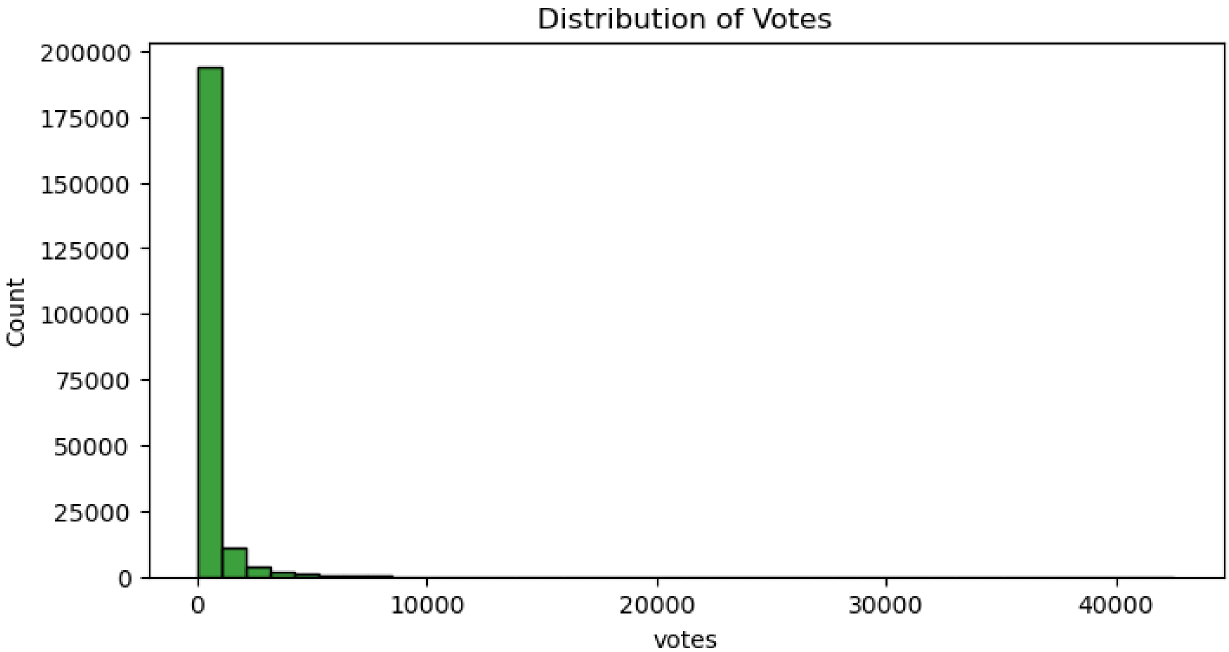




Insight:
The distribution of average cost for two is right-skewed, indicating that most restaurants are priced in the lower to mid-cost range. A small number of restaurants have very high prices, appearing as outliers in the box plot. This suggests that affordable dining options dominate the dataset, while premium restaurants form a smaller segment.

In [21]:
```
#Votes – Histogram
plt.figure(figsize=(8,4))
sns.histplot(df['votes'], bins=40, color='green')
```

```
plt.title('Distribution of Votes')
plt.show()
```

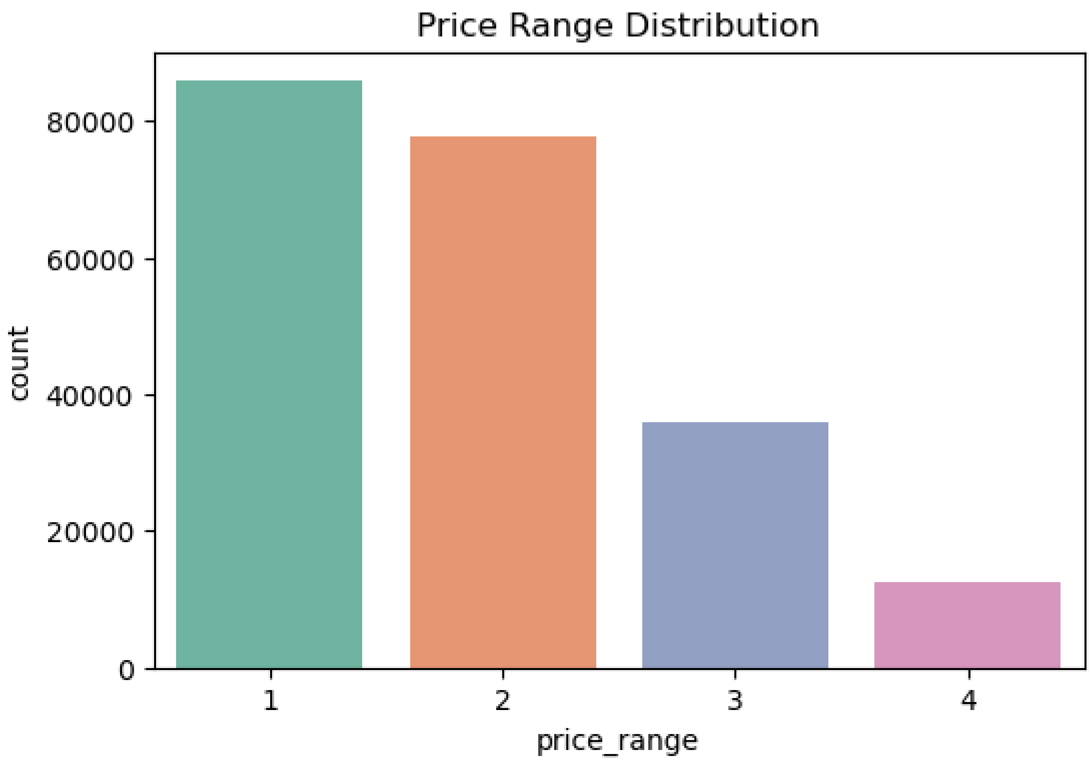


Insight:

The votes distribution is highly right-skewed, where most restaurants receive a relatively low number of votes, while a small number of popular restaurants attract very high engagement. This suggests that customer attention is concentrated on a limited set of well-known or highly rated restaurants.

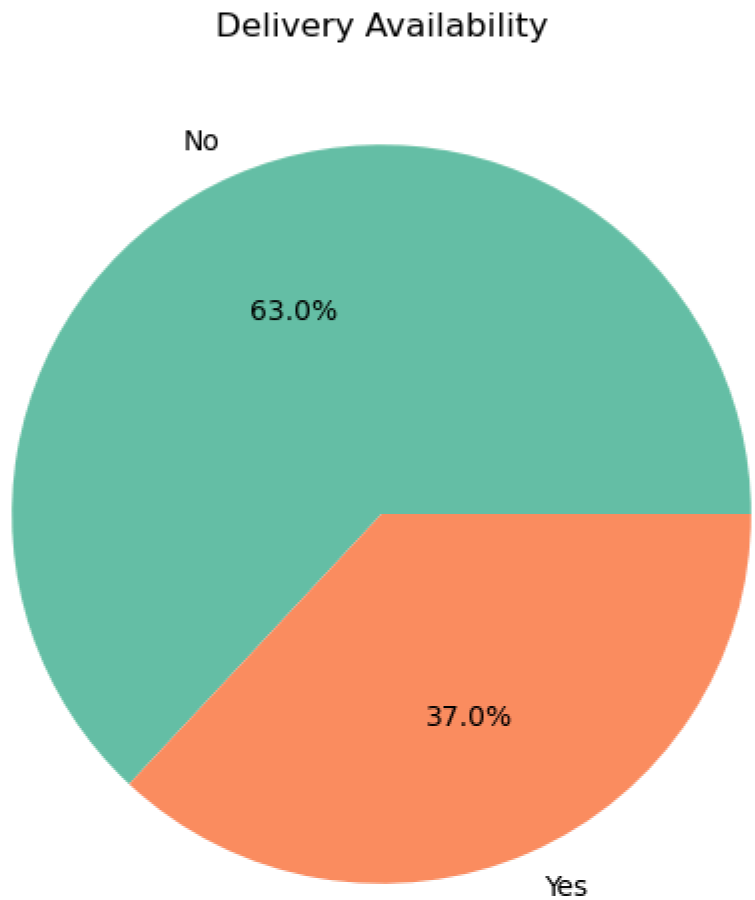
In [22]:

```
#Price Range – Countplot
plt.figure(figsize=(6,4))
sns.countplot(x=df['price_range'], palette='Set2')
plt.title('Price Range Distribution')
plt.show()
```



Insight:
Most restaurants fall within the lower to mid price ranges, indicating a strong focus on affordable dining options. Higher price range restaurants are comparatively fewer, suggesting limited premium dining availability in the dataset.

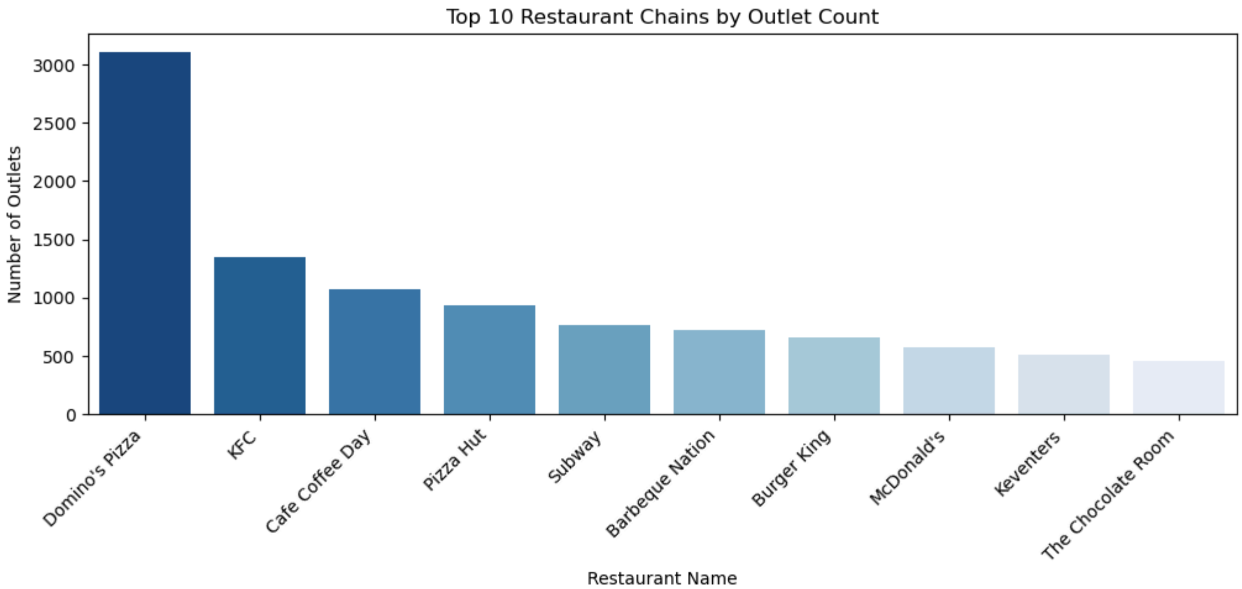
In [23]:
```
#Delivery Availability – Pie Chart
delivery_counts = df['delivery'].value_counts()

plt.figure(figsize=(6,6))
plt.pie(
    delivery_counts,
    labels=delivery_counts.index,
    autopct='%1.1f%%',
    colors=['#66c2a5', '#fc8d62']
)
plt.title('Delivery Availability')
plt.show()
```

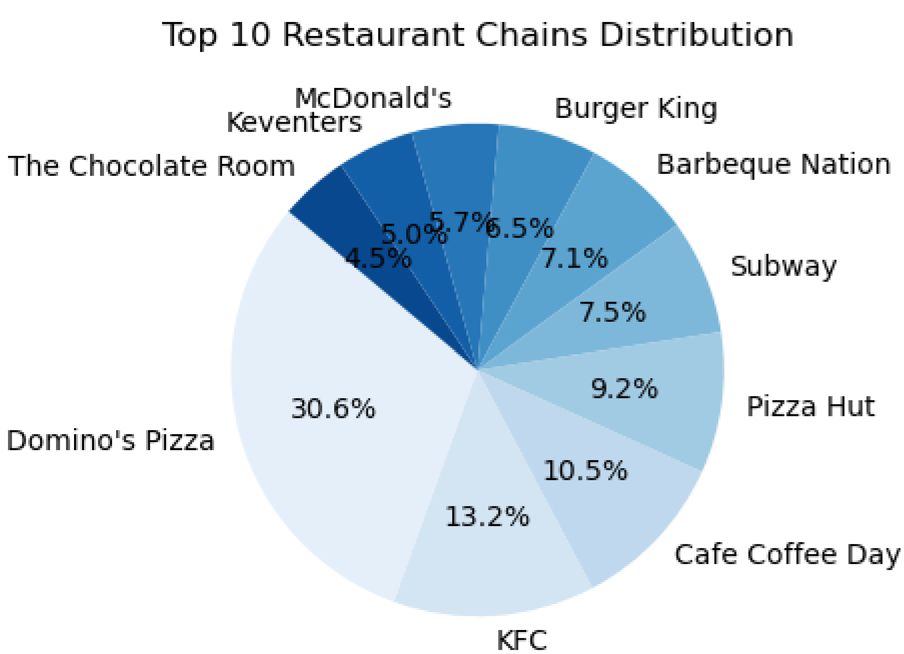


Insight:
A large proportion of restaurants offer delivery services, indicating that online food delivery is a widely adopted feature on the Zomato platform. This reflects strong customer demand for convenience and highlights delivery as a key factor in restaurant visibility and reach.

In [24]:
```
#Countplot (Top 10 Restaurant Names)
plt.figure(figsize=(12,4))
sns.countplot(
    data=df,
    x='name',
    order=df['name'].value_counts().head(10).index,
    palette='Blues_r')
plt.xticks(rotation=45, ha='right')
plt.title("Top 10 Restaurant Chains by Outlet Count")
plt.xlabel("Restaurant Name")
plt.ylabel("Number of Outlets")
plt.show()
#Pie Chart (Top 10 Restaurant Chains)
temp = df['name'].value_counts().head(10).reset_index()
temp.columns = ['name', 'count']
```

```
plt.figure(figsize=(12,4))
plt.pie(
    temp['count'],
    labels=temp['name'],
    autopct='%1.1f%%',
    startangle=140,
    colors=sns.color_palette('Blues', len(temp)))
plt.title('Top 10 Restaurant Chains Distribution')
plt.show()
```





Insight:

A small number of restaurant chains dominate the dataset with multiple outlets, while most restaurants appear only once or a few times. This indicates a fragmented restaurant market where popular chains have higher visibility, but independent restaurants still form a large portion of listings.

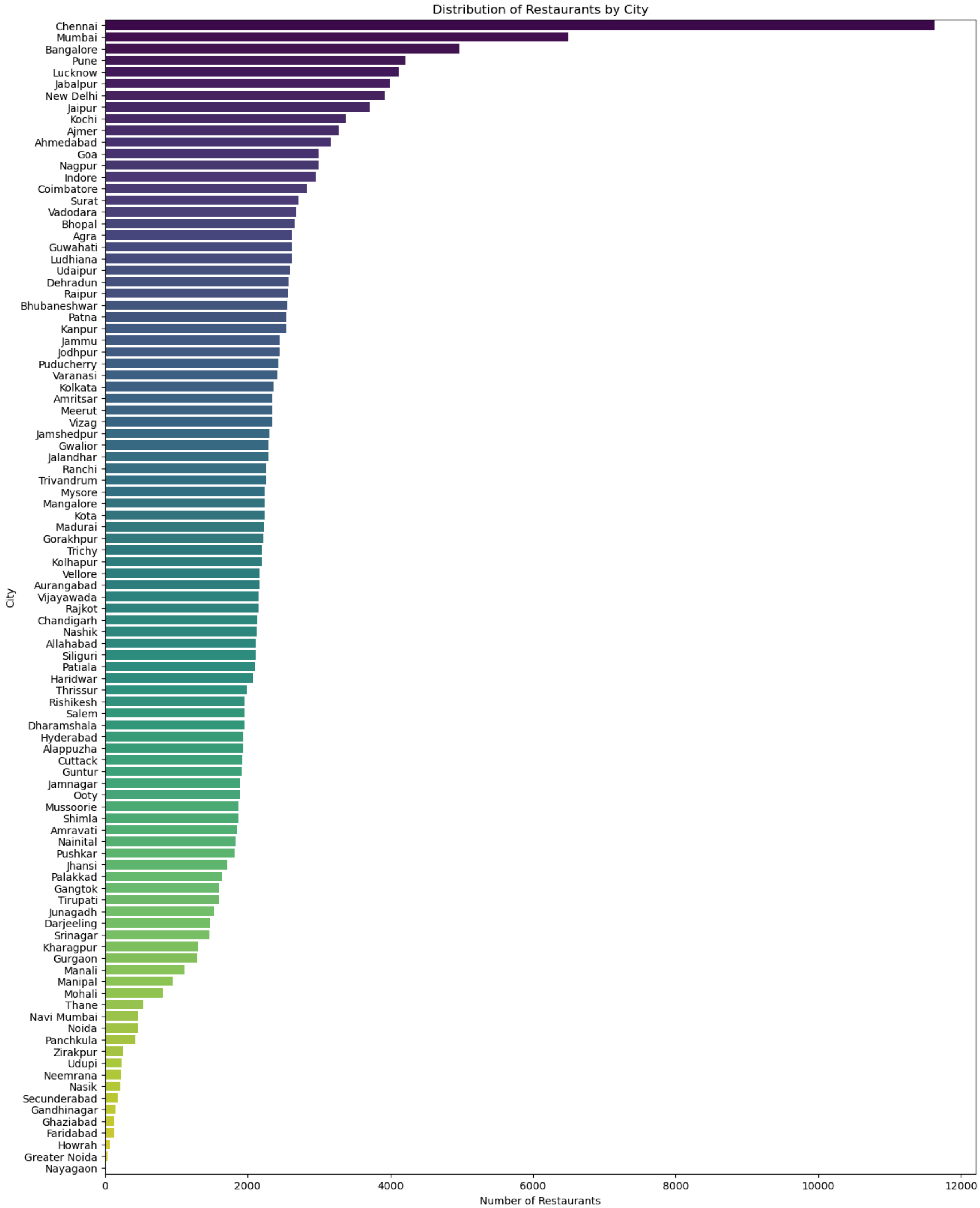
Bivariate Analysis

In [25]:
```
px.scatter(
    df,
    x='average_cost_for_two',
    y='aggregate_rating',
    color='votes',
    hover_data=['name', 'average_cost_for_two', 'aggregate_rating', 'votes'],
    height=400,
    title='Average Cost for Two vs Aggregate Rating with Popularity Intensity'
).show()
```

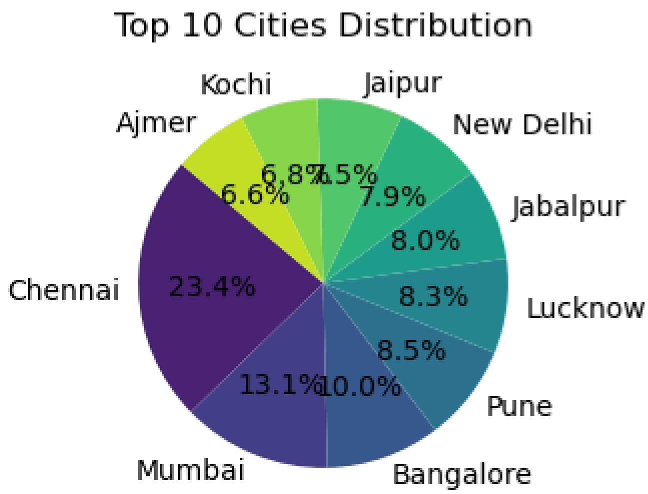
Insight:
The scatter plot shows that restaurants with moderate pricing receive consistently higher ratings, while very high-cost restaurants do not always achieve better customer ratings. Higher vote intensity is concentrated around mid-priced restaurants, indicating that popularity and customer engagement are driven more by value for money than by premium pricing.

In [26]:
```
#Countplot (City Distribution)
plt.figure(figsize=(15,20))
```

```python
sns.countplot(
    data=df,
    y='city',
    order=df['city'].value_counts().index,
    palette='viridis')
plt.title('Distribution of Restaurants by City')
plt.xlabel('Number of Restaurants')
plt.ylabel('City')
plt.show()
#Pie Chart
# Top 10 cities for pie chart
temp = df['city'].value_counts().head(10)

plt.figure(figsize=(12,3))
plt.pie(
    temp.values,
    labels=temp.index,
    autopct='%1.1f%%',
    startangle=140,
    colors=sns.color_palette('viridis', len(temp)))
plt.title('Top 10 Cities Distribution')
plt.show()
```



Distribution of Restaurants by City
City
Number of Restaurants
0
2000
4000
6000
8000
10000
12000
Chennai
Mumbai
Bangalore
Pune
Lucknow
Jabalpur
New Delhi
Jaipur
Kochi
Ajmer
Ahmedabad
Goa
Nagpur
Indore
Coimbatore
Surat
Vadodara
Bhopal
Agra
Guwahati
Ludhiana
Udaipur
Dehradun
Raipur
Bhubaneshwar
Patna
Kanpur
Jammu
Jodhpur
Puducherry
Varanasi
Kolkata
Amritsar
Meerut
Vizag
Jamshedpur
Gwalior
Jalandhar
Ranchi
Trivandrum
Mysore
Mangalore
Kota
Madurai
Gorakhpur
Trichy
Kolhapur
Vellore
Aurangabad
Vijayawada
Rajkot
Chandigarh
Nashik
Allahabad
Siliguri
Patiala
Haridwar
Thrissur
Rishikesh
Salem
Dharamshala
Hyderabad
Alappuzha
Cuttack
Guntur
Jamnagar
Ooty
Mussoorie
Shimla
Amravati
Nainital
Pushkar
Jhansi
Palakkad
Gangtok
Tirupati
Junagadh
Darjeeling
Srinagar
Kharagpur
Gurgaon
Manali
Manipal
Mohali
Thane
Navi Mumbai
Noida
Panchkula
Zirakpur
Udupi
Neemrana
Nasik
Secunderabad
Gandhinagar
Ghaziabad
Faridabad
Howrah
Greater Noida
Nayagaon



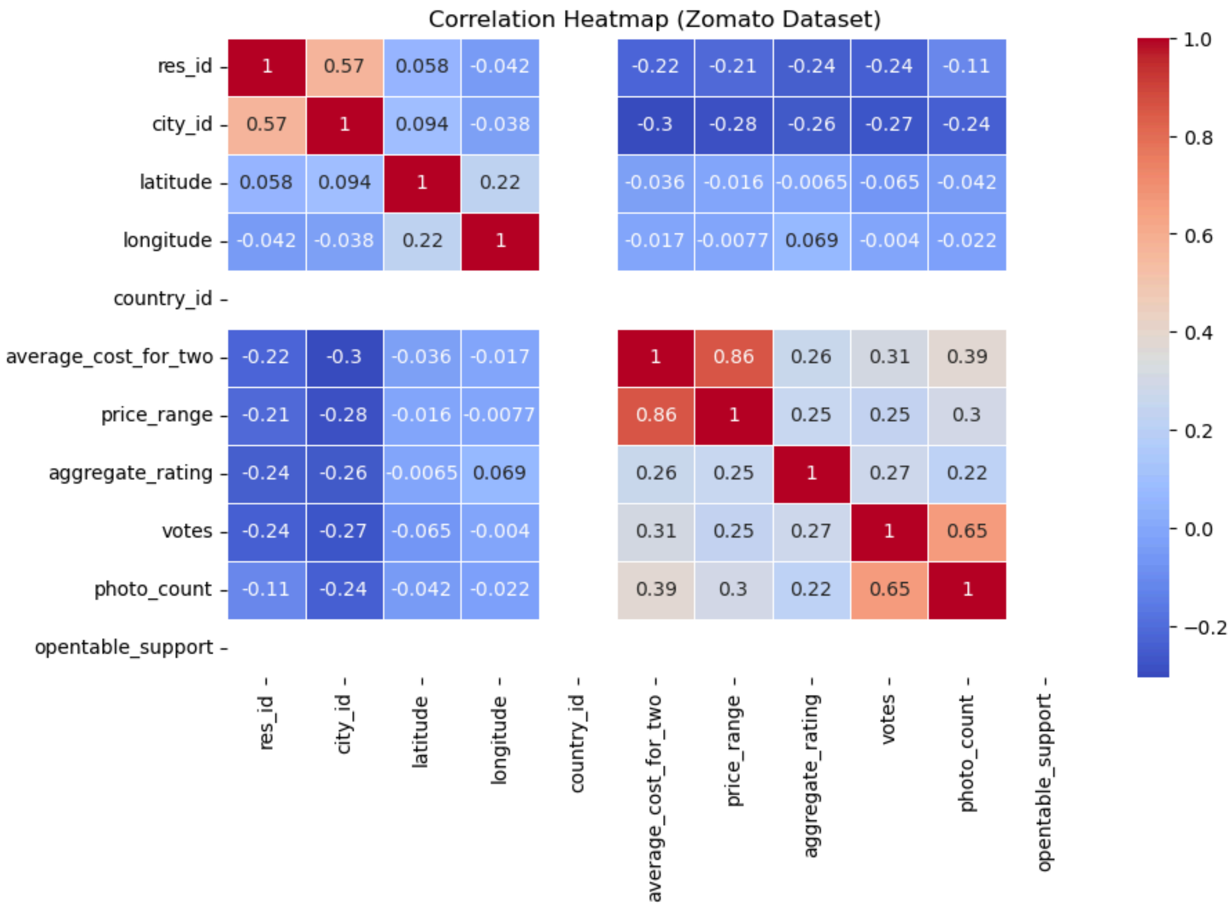
Insight:

Restaurant listings are heavily concentrated in a few major cities, while many cities have comparatively fewer restaurants. This indicates that urban hubs dominate the Zomato platform, reflecting higher restaurant density and customer demand in metropolitan areas.

Multivariate Analysis

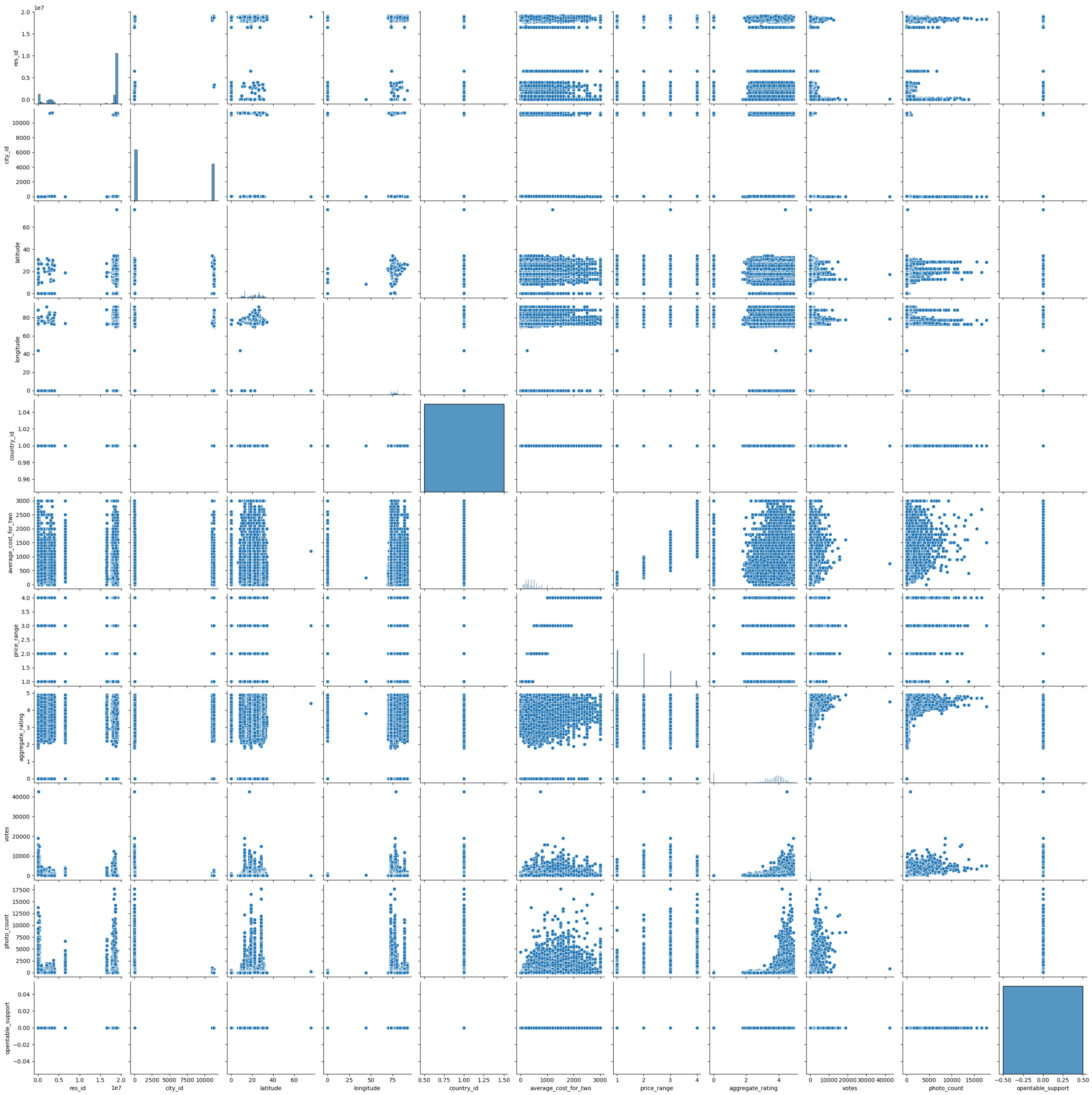
In [27]:

```
num_df = df.select_dtypes(include='number')

plt.figure(figsize=(10,6))
sns.heatmap(
    num_df.corr(),
    annot=True,
    cmap='coolwarm',
    linewidths=0.5
)
plt.title('Correlation Heatmap (Zomato Dataset)')
plt.show()
```

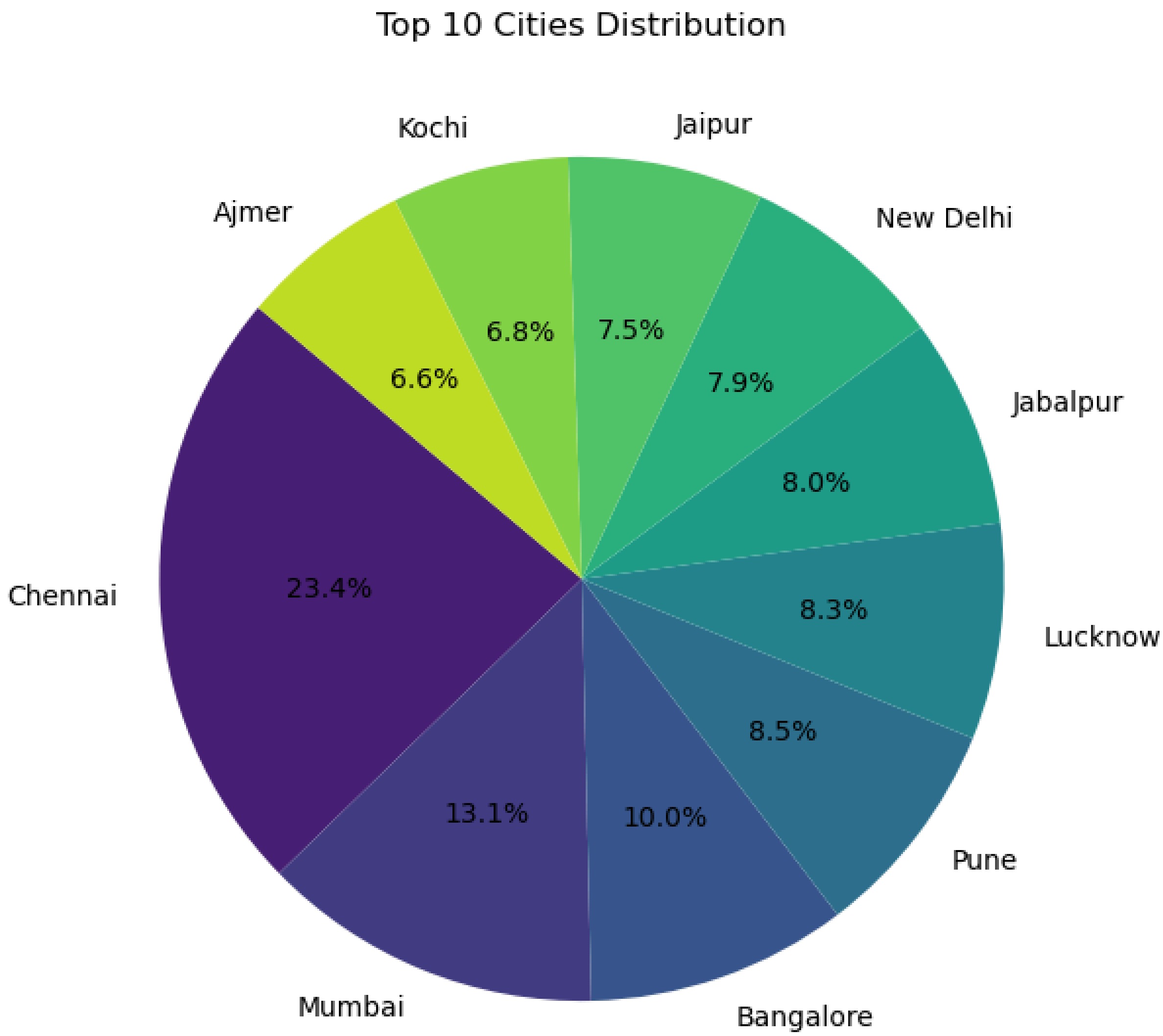


Insight:
The heatmap indicates a strong positive correlation between `votes` and `photo_count`, showing that restaurants with higher customer engagement tend to have greater visibility. Pricing variables such as `average_cost_for_two` and `price_range` are positively correlated with each other but show weak correlation with `aggregate_rating`. Overall, customer engagement metrics influence restaurant popularity more strongly than pricing.

In [28]:
```
# Pair Plot (Multiple Variables)
sns.pairplot(df)
plt.show()
```



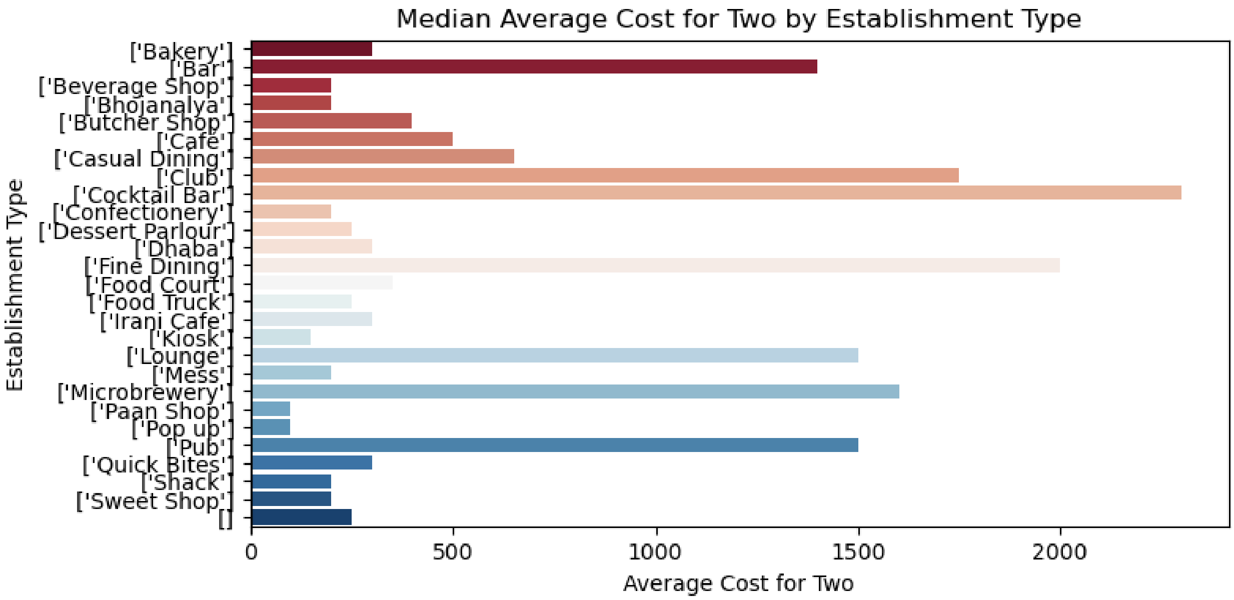
In [29]:
```python
# Directly get top 10 city counts
city_counts = df['city'].value_counts().head(10)

plt.figure(figsize=(7,7))
plt.pie(
    city_counts.values,
    labels=city_counts.index,
    autopct='%1.1f%%',
    startangle=140,
    colors=sns.color_palette('viridis', len(city_counts))
)
plt.title('Top 10 Cities Distribution')
plt.show()
```



Insight:
The pie chart shows that restaurant listings are heavily concentrated in a few major cities. These top cities account for a large share of Zomato's restaurants, indicating higher market activity and customer demand in metropolitan regions.

In [30]:
```
temp2 = df.groupby('establishment')['average_cost_for_two'].median().reset_inc

plt.figure(figsize=(8,4))
sns.barplot(
    data=temp2,
    x='average_cost_for_two',
    y='establishment',
    palette='RdBu'
)
plt.title('Median Average Cost for Two by Establishment Type')
plt.xlabel('Average Cost for Two')
plt.ylabel('Establishment Type')
```

```
plt.show()
```



Insight:
Quick Bites and Casual Dining dominate the restaurant distribution, indicating a strong preference for affordable and fast dining options. Fine Dining and premium establishments show a higher median cost for two, reflecting their positioning toward high-spending customers. This highlights a clear segmentation in Zomato's restaurant ecosystem based on dining style and pricing.

🔚 Final Conclusion — Zomato Restaurant Data Analysis

The analysis reveals that Zomato's restaurant ecosystem is highly concentrated in major urban cities, indicating stronger market presence and customer demand in metropolitan areas. Smaller cities contribute comparatively fewer restaurant listings, highlighting an uneven geographic distribution.

Quick Bites and Casual Dining dominate the platform, reflecting customer preference for affordable, fast, and convenient dining options. Premium categories such as Fine Dining are fewer in number but show a higher average cost for two, clearly positioning them toward a niche, high-spending customer segment.

Customer engagement metrics like votes and photo count show a strong positive relationship, suggesting that visibility and popularity play a crucial role in restaurant performance. In contrast, pricing factors have a weak relationship with customer ratings, indicating that higher cost does not necessarily guarantee better customer satisfaction.

Overall, the dataset highlights that **engagement, accessibility, and location** are more influential drivers of restaurant success on Zomato than pricing alone. These insights can help restaurants and stakeholders optimize pricing strategies, improve visibility, and focus on high-demand urban markets.

In [ ]:

```
```