Task 1: Data Exploration and Preprocessing

In [1]:

```
import pandas as pd
```

In [3]:

```
df = pd.read_csv("Dataset .csv")
```

In [4]:

```
print("Missing values in each column:")
print(df.isnull().sum())
```

```
Missing values in each column:
Restaurant ID           0
Restaurant Name         0
Country Code            0
City                    0
Address                 0
Locality                0
Locality Verbose        0
Longitude               0
Latitude                0
Cuisines                9
Average Cost for two    0
Currency                0
Has Table booking       0
Has Online delivery     0
Is delivering now       0
Switch to order menu    0
Price range             0
Aggregate rating        0
Rating color            0
Rating text             0
Votes                   0
dtype: int64
```

In [6]:

```
df['Restaurant ID'] = pd.to_numeric(df['Restaurant ID'])
df['Longitude'] = pd.to_numeric(df['Longitude'])
df['Latitude'] = pd.to_numeric(df['Latitude'])
df['Average Price'] = pd.to_numeric(df['Average Cost for two'])
df['Votes'] = pd.to_numeric(df['Votes'])
```

In [8]:

```
print("\nDistribution of Aggregate rating:")
print(df['Aggregate rating'].value_counts())
```

```
Distribution of Aggregate rating:
0.0    2148
3.2     522
3.1     519
3.4     498
3.3     483
3.5     480
3.0     468
3.6     458
3.7     427
3.8     400
2.9     381
3.9     335
2.8     315
4.1     274
4.0     266
2.7     250
4.2     221
2.6     191
4.3     174
4.4     144
2.5     110
4.5      95
2.4      87
4.6      78
4.9      61
2.3      47
4.7      42
2.2      27
4.8      25
2.1      15
2.0       7
1.9       2
1.8       1
Name: Aggregate rating, dtype: int64
```

In [10]:

```
print("\nDescriptive statistics:")
print(df.describe())
```

```
Descriptive statistics:
       Restaurant ID  Country Code     Longitude      Latitude  \
count   9.551000e+03   9551.000000   9551.000000   9551.000000   
mean    9.051128e+06     18.365616     64.126574     25.854381   
std     8.791521e+06     56.750546     41.467058     11.007935   
min     5.300000e+01      1.000000   -157.948486    -41.330428   
25%     3.019625e+05      1.000000     77.081343     28.478713   
50%     6.004089e+06      1.000000     77.191964     28.570469   
75%     1.835229e+07      1.000000     77.282006     28.642758   
max     1.850065e+07    216.000000    174.832089     55.976980   

       Average Cost for two  Price range  Aggregate rating         Votes  \
count           9551.000000  9551.000000       9551.000000   9551.000000   
mean            1199.210763     1.804837          2.666370    156.909748   
std            16121.183073     0.905609          1.516378    430.169145   
min                0.000000     1.000000          0.000000      0.000000   
25%              250.000000     1.000000          2.500000      5.000000   
50%              400.000000     2.000000          3.200000     31.000000   
75%              700.000000     2.000000          3.700000    131.000000   
max           800000.000000     4.000000          4.900000  10934.000000   

       Average Price  
count    9551.000000  
mean     1199.210763  
std     16121.183073  
min         0.000000  
25%       250.000000  
50%       400.000000  
75%       700.000000  
max    800000.000000  
```

In [11]:

```
print("\nCorrelation matrix:")
print(df.corr())
```

```
Correlation matrix:
                    Restaurant ID  Country Code  Longitude  Latitude  \
Restaurant ID            1.000000      0.148471  -0.226081 -0.052081   
Country Code             0.148471      1.000000  -0.698299  0.019792   
Longitude               -0.226081     -0.698299   1.000000  0.043207   
Latitude                -0.052081      0.019792   0.043207  1.000000   
Average Cost for two    -0.001693      0.043225   0.045891 -0.111088   
Price range             -0.134540      0.243327  -0.078939 -0.166688   
Aggregate rating        -0.326212      0.282189  -0.116818  0.000516   
Votes                   -0.147023      0.154530  -0.085101 -0.022962   
Average Price           -0.001693      0.043225   0.045891 -0.111088   

                    Average Cost for two  Price range  Aggregate rating  \
Restaurant ID                  -0.001693    -0.134540         -0.326212   
Country Code                    0.043225     0.243327          0.282189   
Longitude                       0.045891    -0.078939         -0.116818   
Latitude                       -0.111088    -0.166688          0.000516   
Average Cost for two            1.000000     0.075083          0.051792   
Price range                     0.075083     1.000000          0.437944   
Aggregate rating                0.051792     0.437944          1.000000   
Votes                           0.067783     0.309444          0.313691   
Average Price                   1.000000     0.075083          0.051792   

                         Votes  Average Price  
Restaurant ID        -0.147023      -0.001693  
Country Code          0.154530       0.043225  
Longitude            -0.085101       0.045891  
Latitude             -0.022962      -0.111088  
Average Cost for two  0.067783       1.000000  
Price range           0.309444       0.075083  
Aggregate rating      0.313691       0.051792  
Votes                 1.000000       0.067783  
Average Price         0.067783       1.000000  
```

```
C:\Users\Jananisha\AppData\Local\Temp\ipykernel_2972\4132872281.py:2: FutureWarnin
g: The default value of numeric_only in DataFrame.corr is deprecated. In a future
version, it will default to False. Select only valid columns or specify the value
of numeric_only to silence this warning.
  print(df.corr())
```

Task 2: Descriptive Analysis

In [12]:

```
import pandas as pd
```

In [13]:

```
df = pd.read_csv('Dataset .csv')
```

In [15]:

```
print("Mean Average Cost for two:", df['Average Cost for two'].mean())
print("Median Average Cost for two:", df['Average Cost for two'].median())
print("Standard Deviation of Average Cost for two:", df['Average Cost for two'].st
```

```
Mean Average Cost for two: 1199.2107632708617
Median Average Cost for two: 400.0
Standard Deviation of Average Cost for two: 16121.18307349965
```

In [16]:

```
print("Frequency of Cities:", df['City'].value_counts())
print("Frequency of Cuisines:", df['Cuisines'].value_counts())
```

```
Frequency of Cities: New Delhi           5473
Gurgaon             1118
Noida               1080
Faridabad            251
Ghaziabad             25
                    ...
Panchkula              1
Mc Millan              1
Mayfield               1
Macedon                1
Vineland Station       1
Name: City, Length: 141, dtype: int64
Frequency of Cuisines: North Indian                                             93
6
North Indian, Chinese                                    511
Chinese                                                  354
Fast Food                                                354
North Indian, Mughlai                                    334
                                                        ...
Bengali, Fast Food                                         1
North Indian, Rajasthani, Asian                            1
Chinese, Thai, Malaysian, Indonesian                       1
Bakery, Desserts, North Indian, Bengali, South Indian      1
Italian, World Cuisine                                     1
Name: Cuisines, Length: 1825, dtype: int64
```

In [17]:

```
top_cuisines = df['Cuisines'].value_counts().head(5)
top_cities = df['City'].value_counts().head(5)
```

In [18]:

```
print("Top 5 Cuisines:")
print(top_cuisines)
```

```
Top 5 Cuisines:
North Indian             936
North Indian, Chinese    511
Chinese                  354
Fast Food                354
North Indian, Mughlai    334
Name: Cuisines, dtype: int64
```

In [19]:

```
print("Top 5 Cities:")
print(top_cities)
```

```
Top 5 Cities:
New Delhi    5473
Gurgaon      1118
Noida        1080
Faridabad     251
Ghaziabad      25
Name: City, dtype: int64
```

Task 3: Geospatial Analysis

In [21]:

```
pip install folium
```

```
Defaulting to user installation because normal site-packages is not writeable
Collecting folium
  Downloading folium-0.17.0-py2.py3-none-any.whl (108 kB)
     -------------------------------------- 108.4/108.4 kB 3.2 MB/s eta 0:00:00
Requirement already satisfied: numpy in c:\programdata\anaconda3\lib\site-packages
(from folium) (1.23.5)
Requirement already satisfied: requests in c:\programdata\anaconda3\lib\site-packa
ges (from folium) (2.28.1)
Requirement already satisfied: jinja2>=2.9 in c:\programdata\anaconda3\lib\site-pa
ckages (from folium) (3.1.2)
Collecting xyzservices
  Downloading xyzservices-2024.9.0-py3-none-any.whl (85 kB)
     ---------------------------------------- 85.1/85.1 kB 5.0 MB/s eta 0:00:00
Collecting branca>=0.6.0
  Downloading branca-0.7.2-py3-none-any.whl (25 kB)
Requirement already satisfied: MarkupSafe>=2.0 in c:\programdata\anaconda3\lib\sit
e-packages (from jinja2>=2.9->folium) (2.1.1)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-p
ackages (from requests->folium) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\programdata\anaconda3\l
ib\site-packages (from requests->folium) (1.26.14)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\programdata\anaconda
3\lib\site-packages (from requests->folium) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib
\site-packages (from requests->folium) (2022.12.7)
Installing collected packages: xyzservices, branca, folium
Successfully installed branca-0.7.2 folium-0.17.0 xyzservices-2024.9.0
Note: you may need to restart the kernel to use updated packages.
```

In [36]:

```python
import pandas as pd
import folium
import matplotlib.pyplot as plt
```

In [37]:

```python
df = pd.read_csv('Dataset .csv')
```

In [38]:

```python
m = folium.Map(location=[df['Latitude'].mean(), df['Longitude'].mean()], zoom_start
```

In [39]:

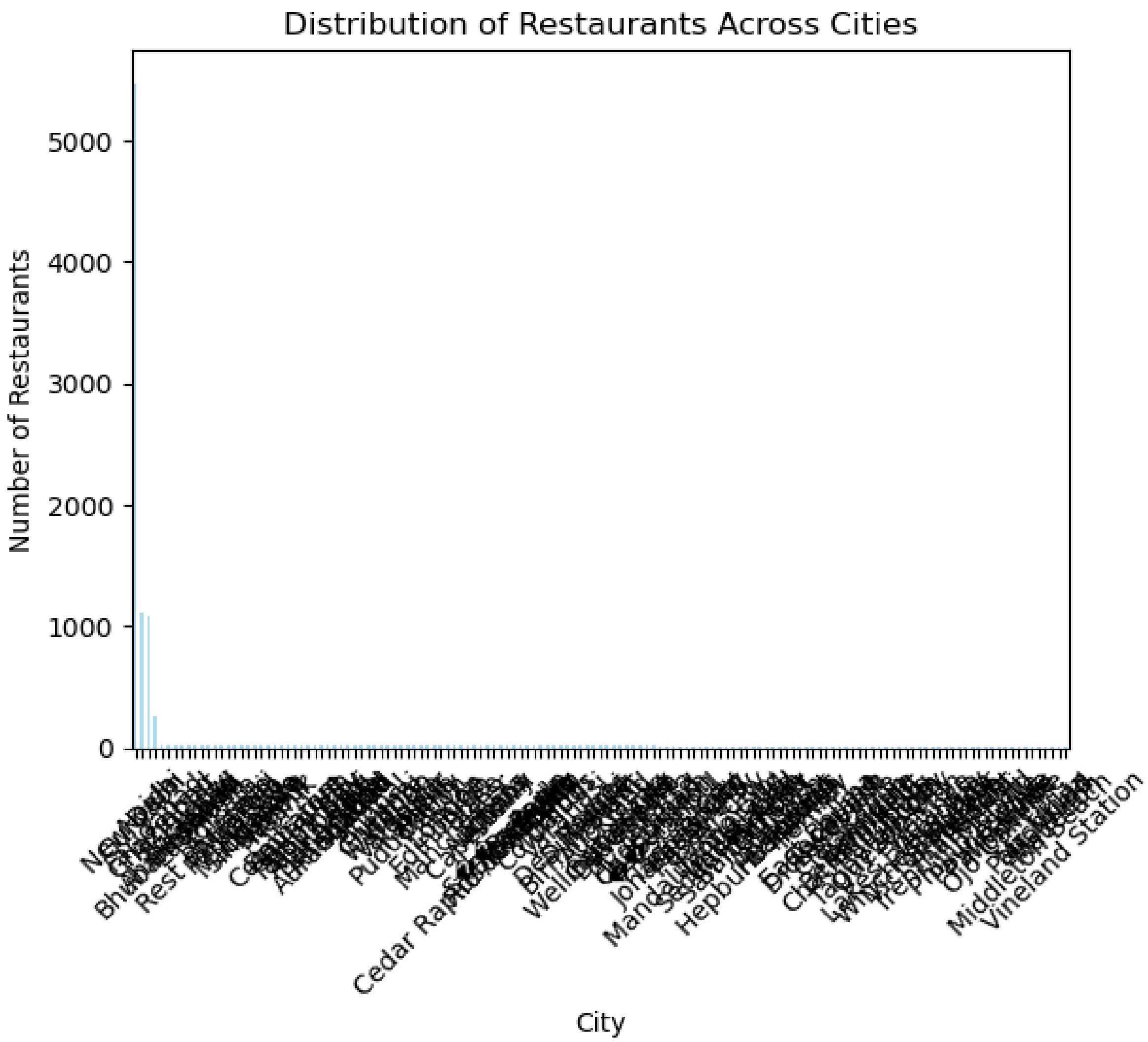
```python
for index, row in df.iterrows():
    popup = f"*Restaurant:* {row['Restaurant Name']}<br>*City:* {row['City']}<br>*F
    folium.Marker([row['Latitude'], row['Longitude']], popup=popup).add_to(m)
```

In [40]:

```python
m.save('restaurants_map.html')
```

In [41]:

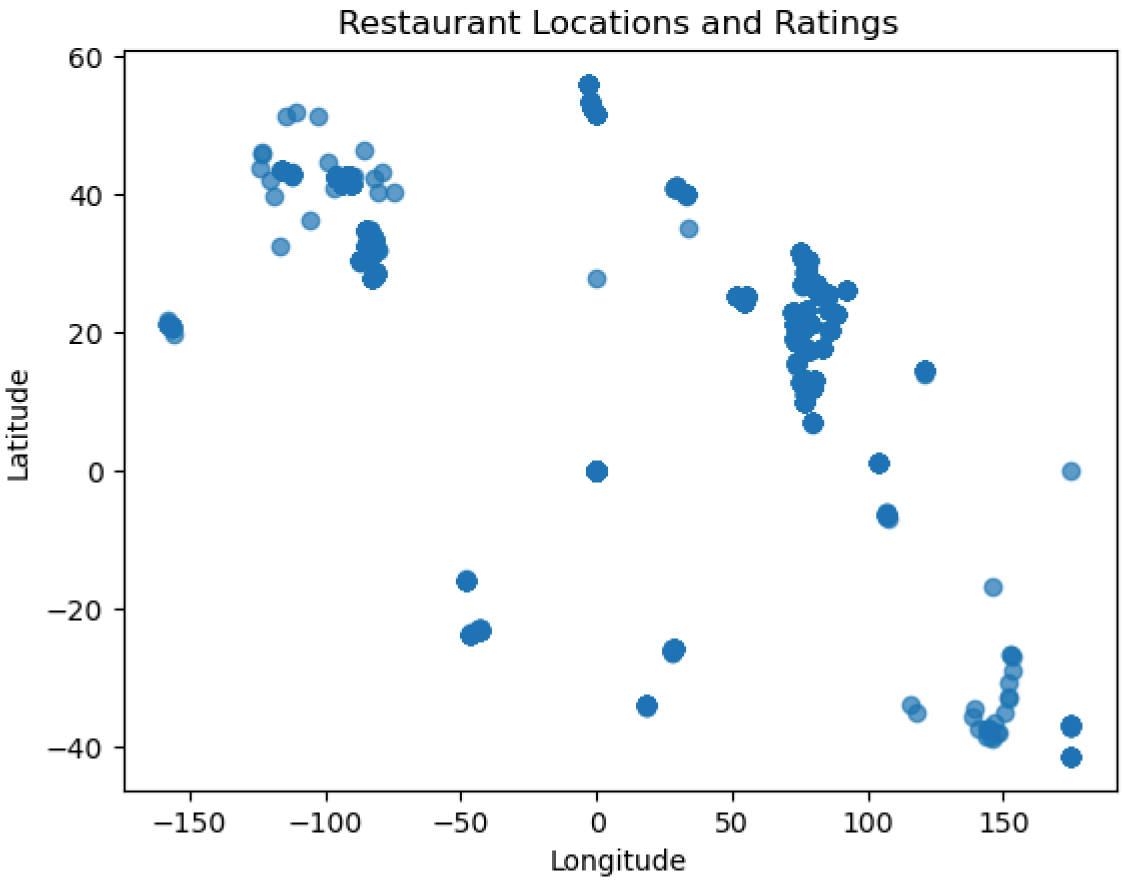
```python
city_counts = df['City'].value_counts()
city_counts.plot(kind='bar', color='skyblue', alpha=0.7)
plt.xlabel('City')
plt.ylabel('Number of Restaurants')
plt.title('Distribution of Restaurants Across Cities')
plt.xticks(rotation=45)
plt.show()
```



In [45]:

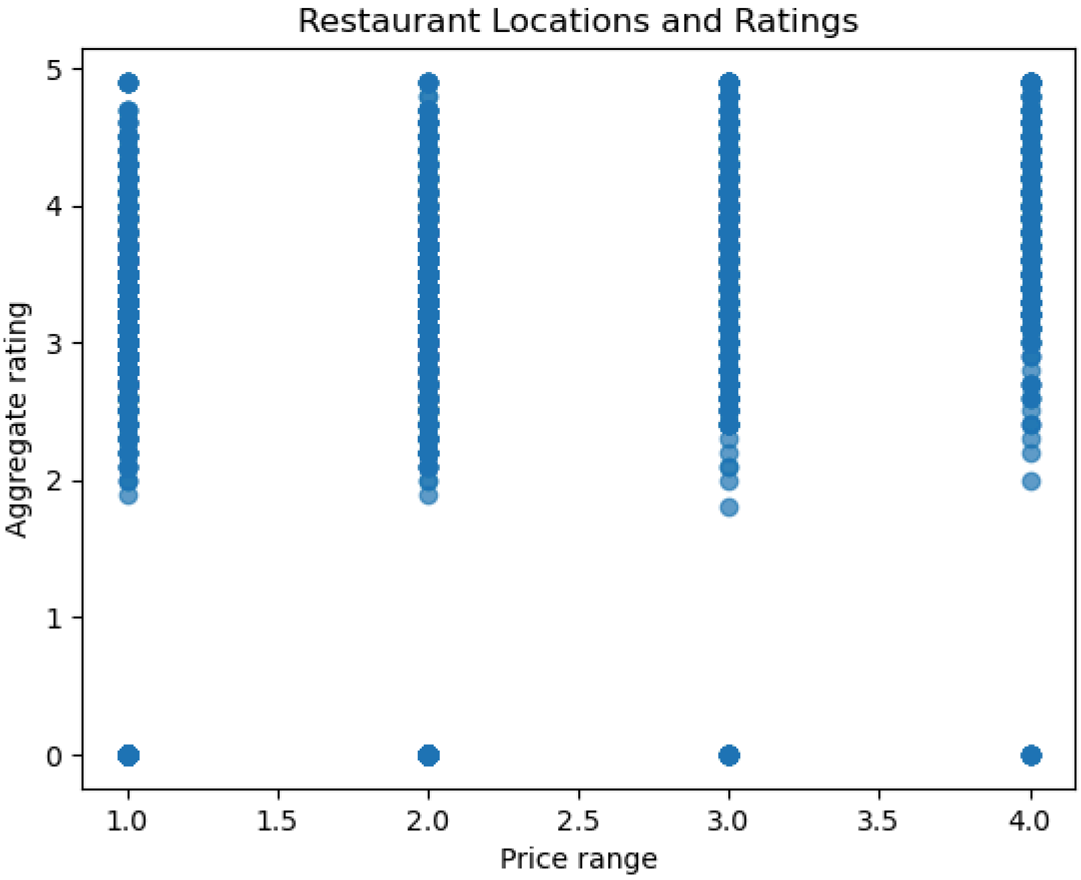
```
plt.scatter(df['Longitude'], df['Latitude'], cmap='viridis', alpha=0.7)
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.title('Restaurant Locations and Ratings')
plt.show()
```

```
C:\Users\Jananisha\AppData\Local\Temp\ipykernel_2972\2068308286.py:1: UserWarning:
No data for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  plt.scatter(df['Longitude'], df['Latitude'], cmap='viridis', alpha=0.7)
```



In [46]:
```
plt.scatter(df['Price range'], df['Aggregate rating'], cmap='viridis', alpha=0.7)
plt.xlabel('Price range')
plt.ylabel('Aggregate rating')
plt.title('Restaurant Locations and Ratings')
plt.show()
```

```
C:\Users\Jananisha\AppData\Local\Temp\ipykernel_2972\677575574.py:1: UserWarning:
No data for colormapping provided via 'c'. Parameters 'cmap' will be ignored
  plt.scatter(df['Price range'], df['Aggregate rating'], cmap='viridis', alpha=0.
7)
```


Restaurant Locations and Ratings
Aggregate rating
0
1
2
3
4
5
Price range
1.0
1.5
2.0
2.5
3.0
3.5
4.0