Level 1

Task 1 :- Data Exploration and Preprocessing

◉Explore the dataset and identify the number of rows and columns.

◉Check for missing values in each column and handle them accordingly.

◉Perform data type conversion if necessary. Analyze the distribution of the target variable ("Aggregate rating") and identify any class imbalances.

In [1]:

```
import warnings
warnings.filterwarnings("ignore")
```

In [5]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns; sns.set(color_codes=True)
%matplotlib inline
```

In [6]:

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

Out[6]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latit
0	6317637	Le Petit Souffle	162	Makati City	Third Floor, Century City Mall, Kalayaan Avenu...	Century City Mall, Poblacion, Makati City	Century City Mall, Poblacion, Makati City, Mak...	121.027535	14.565
1	6304287	Izakaya Kikufuji	162	Makati City	Little Tokyo, 2277 Chino Roces Avenue, Legaspi...	Little Tokyo, Legaspi Village, Makati City	Little Tokyo, Legaspi Village, Makati City, Ma...	121.014101	14.553
2	6300002	Heat - Edsa Shangri-La	162	Mandaluyong City	Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...	Edsa Shangri-La, Ortigas, Mandaluyong City	Edsa Shangri-La, Ortigas, Mandaluyong City, Ma...	121.056831	14.581
3	6318506	Ooma	162	Mandaluyong City	Third Floor, Mega Fashion Hall, SM Megamall, O...	SM Megamall, Ortigas, Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City, Mandal...	121.056475	14.585
4	6314302	Sambo Kojin	162	Mandaluyong City	Third Floor, Mega Atrium, SM Megamall, Ortigas...	SM Megamall, Ortigas, Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City, Mandal...	121.057508	14.584

5 rows × 21 columns

In [9]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9551 entries, 0 to 9550
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Restaurant ID         9551 non-null   int64  
 1   Restaurant Name       9551 non-null   object 
 2   Country Code          9551 non-null   int64  
 3   City                  9551 non-null   object 
 4   Address               9551 non-null   object 
 5   Locality              9551 non-null   object 
 6   Locality Verbose      9551 non-null   object 
 7   Longitude             9551 non-null   float64
 8   Latitude              9551 non-null   float64
 9   Cuisines              9542 non-null   object 
 10  Average Cost for two  9551 non-null   int64  
 11  Currency              9551 non-null   object 
 12  Has Table booking     9551 non-null   object 
 13  Has Online delivery   9551 non-null   object 
 14  Is delivering now     9551 non-null   object 
 15  Switch to order menu  9551 non-null   object 
 16  Price range           9551 non-null   int64  
 17  Aggregate rating      9551 non-null   float64
 18  Rating color          9551 non-null   object 
 19  Rating text           9551 non-null   object 
 20  Votes                 9551 non-null   int64  
dtypes: float64(3), int64(5), object(13)
memory usage: 1.5+ MB
```

In [11]:
```
df.shape
```

Out[11]:
```
(9551, 21)
```

In [13]:
```
df.isnull().sum()
```

Out[13]:
```
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 [20]:
```
df['Cuisines'].fillna('Not Specified', inplace=True)
```

In [19]:
```
df.isnull().sum()
```

Out[19]:

```
Restaurant ID           0
Restaurant Name         0
Country Code            0
City                    0
Address                 0
Locality                0
Locality Verbose        0
Longitude               0
Latitude                0
Cuisines                0
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 [24]:

```
dupli = df.duplicated().sum()
print(f'Number of duplicate Rows are', (dupli))
```

```
Number of duplicate Rows are 0
```

In [25]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9551 entries, 0 to 9550
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Restaurant ID         9551 non-null   int64  
 1   Restaurant Name       9551 non-null   object 
 2   Country Code          9551 non-null   int64  
 3   City                  9551 non-null   object 
 4   Address               9551 non-null   object 
 5   Locality              9551 non-null   object 
 6   Locality Verbose      9551 non-null   object 
 7   Longitude             9551 non-null   float64
 8   Latitude              9551 non-null   float64
 9   Cuisines              9551 non-null   object 
 10  Average Cost for two  9551 non-null   int64  
 11  Currency              9551 non-null   object 
 12  Has Table booking     9551 non-null   object 
 13  Has Online delivery   9551 non-null   object 
 14  Is delivering now     9551 non-null   object 
 15  Switch to order menu  9551 non-null   object 
 16  Price range           9551 non-null   int64  
 17  Aggregate rating      9551 non-null   float64
 18  Rating color          9551 non-null   object 
 19  Rating text           9551 non-null   object 
 20  Votes                 9551 non-null   int64  
dtypes: float64(3), int64(5), object(13)
memory usage: 1.5+ MB
```

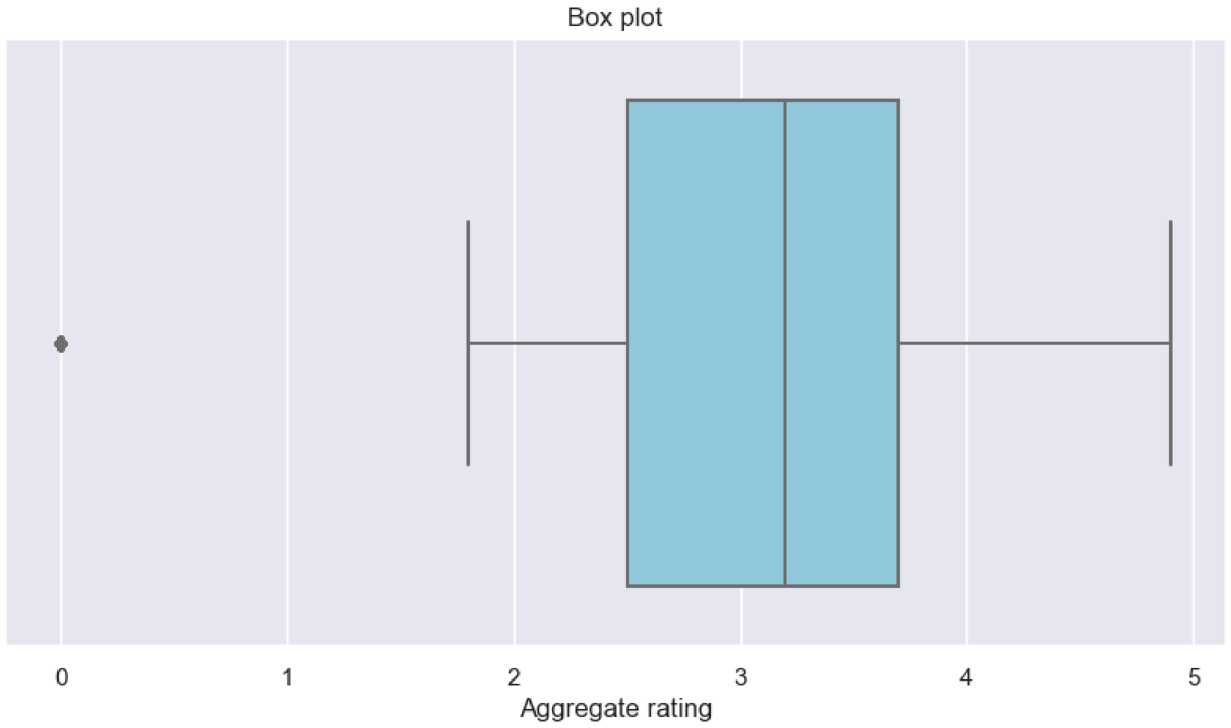
In [27]:

```
target = "Aggregate rating"
print(df[target].describe())
```

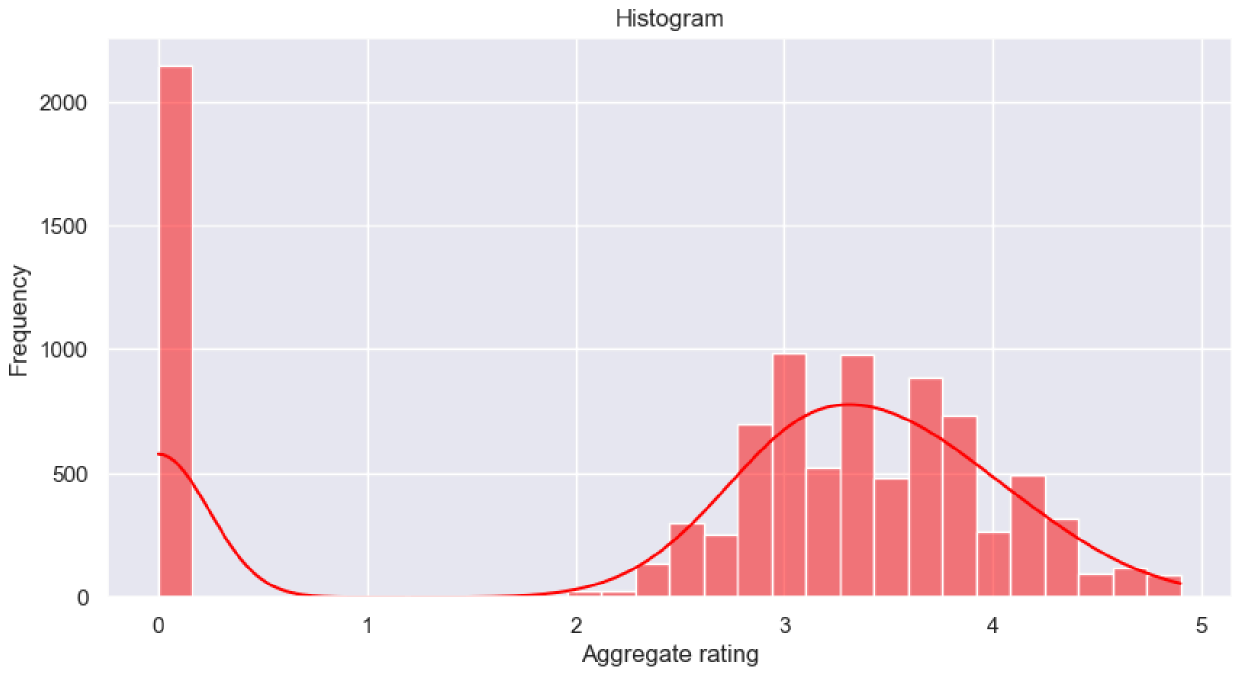
```
count    9551.000000
mean        2.666370
std         1.516378
min         0.000000
25%         2.500000
50%         3.200000
75%         3.700000
max         4.900000
Name: Aggregate rating, dtype: float64
```

In [47]:

```
plt.figure(figsize=(10,5))
sns.boxplot(x=df[target],color='skyblue')
plt.title('Box plot')
plt.xlabel('Aggregate rating')
plt.show()
```



In [44]:
```
plt.figure(figsize=(10,5))
sns.histplot(x=df[target],bins=30, kde=True, color='red')
plt.title('Histogram')
plt.xlabel('Aggregate rating')
plt.ylabel('Frequency')
plt.show()
```



Level 1

Task 2 :- Descriptive Analysis

◉Calculate basic statistical measures (mean,median, standard deviation, etc.) for numericalcolumns.

◉Explore the distribution of categoricalvariables like "Country Code," "City," and"Cuisines."

◉Identify the top cuisines and cities with thehighest number of restaurants.

In [40]:
```
df.describe()
```

Out[40]:

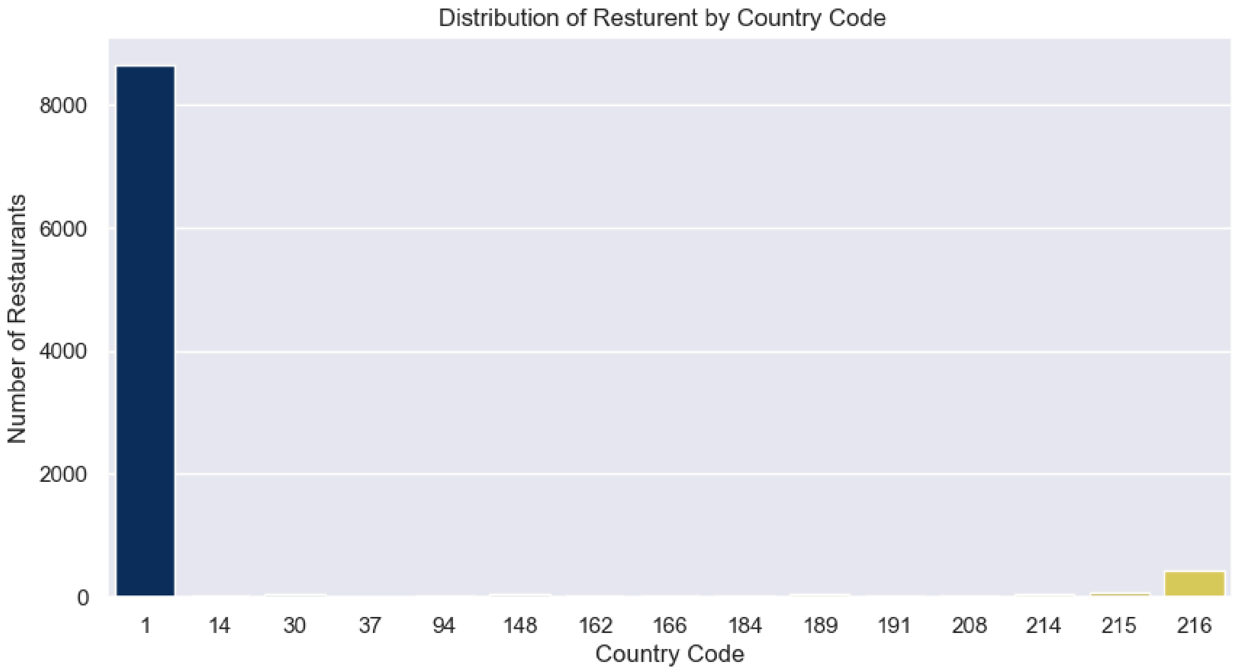
	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range	Aggregate rating	
count	9.551000e+03	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551
mean	9.051128e+06	18.365616	64.126574	25.854381	1199.210763	1.804837	2.666370	156
std	8.791521e+06	56.750546	41.467058	11.007935	16121.183073	0.905609	1.516378	430
min	5.300000e+01	1.000000	-157.948486	-41.330428	0.000000	1.000000	0.000000	0
25%	3.019625e+05	1.000000	77.081343	28.478713	250.000000	1.000000	2.500000	5
50%	6.004089e+06	1.000000	77.191964	28.570469	400.000000	2.000000	3.200000	31
75%	1.835229e+07	1.000000	77.282006	28.642758	700.000000	2.000000	3.700000	131
max	1.850065e+07	216.000000	174.832089	55.976980	800000.000000	4.000000	4.900000	10934

In [41]:
```
df[['Average Cost for two','Price range','Aggregate rating','Votes']].describe()
```

Out[41]:

	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

In [56]:
```
plt.figure(figsize=(10,5))
sns.countplot(x='Country Code',data=df,palette='cividis')
plt.title('Distribution of Resturent by Country Code  ')
plt.xlabel('Country Code')
plt.ylabel('Number of Restaurants')
plt.show()
```

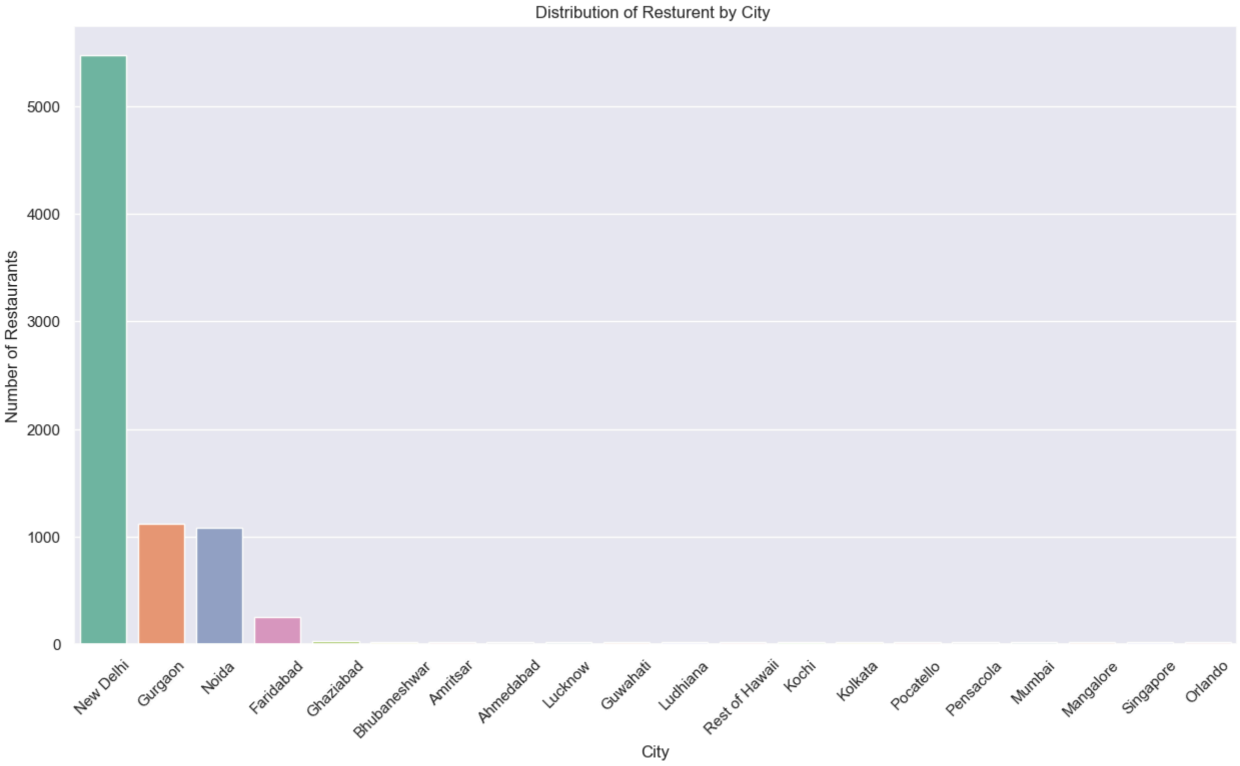


In [48]:
```
top_countries = df['Country Code'].value_counts().head()
print('Top 5 Countries with the Highest Number of Resturents:')
print(top_countries)
```

```
Top 5 Countries with the Highest Number of Resturents:
1      8652
216     434
215      80
30       60
214      60
Name: Country Code, dtype: int64
```

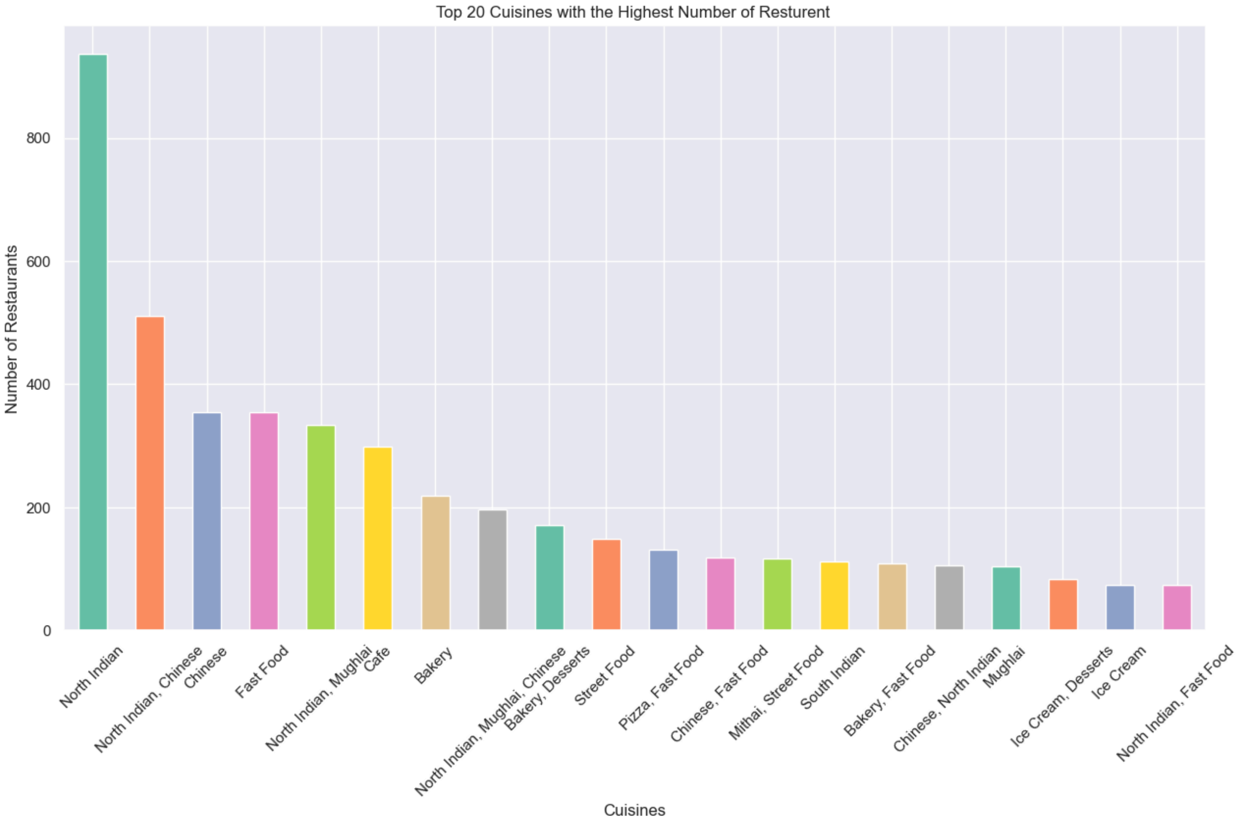
In [57]:

```
plt.figure(figsize=(15,8))
sns.countplot(x='City',data=df,order=df['City'].value_counts().head(20).index,palette='Set2
plt.title('Distribution of Resturent by City ')
plt.xlabel('City')
plt.ylabel('Number of Restaurants')
plt.xticks(rotation=45)
plt.show()
```



In [60]:

```
plt.figure(figsize=(15,8))
cuisines_count = df['Cuisines'].value_counts()
cuisines_count.head(20).plot(kind='bar',color=sns.color_palette('Set2'))
plt.title('Top 20 Cuisines with the Highest Number of Resturent ')
plt.xlabel('Cuisines')
plt.ylabel('Number of Restaurants')
plt.xticks(rotation=45)
plt.show()
```



In [61]:

```
top_cities = df['City'].value_counts().head(10)
print('Top 10 City with the Highest Number of Resturents:')
print(top_cities)
```

```
Top 10 City with the Highest Number of Resturents:
New Delhi       5473
Gurgaon         1118
Noida           1080
Faridabad        251
Ghaziabad         25
Bhubaneshwar      21
Amritsar          21
Ahmedabad         21
Lucknow           21
Guwahati          21
Name: City, dtype: int64
```

In [62]:

```
top_cuisines = df['Cuisines'].value_counts().head(10)
print('Top 10 Cuisines with the Highest Number of Resturents:')
print(top_cuisines)
```

```
Top 10 Cuisines with the Highest Number of Resturents:
North Indian                      936
North Indian, Chinese             511
Chinese                           354
Fast Food                         354
North Indian, Mughlai             334
Cafe                              299
Bakery                            218
North Indian, Mughlai, Chinese    197
Bakery, Desserts                  170
Street Food                       149
Name: Cuisines, dtype: int64
```

Level 1

Task 3 :- Geospatial Analysis

◉Visualize the locations of restaurants on amap using latitude and longitudeinformation.

◉Analyze the distribution of restaurantsacross different cities or countries.

◉Determine if there is any correlationbetween the restaurant's location and itsrating.

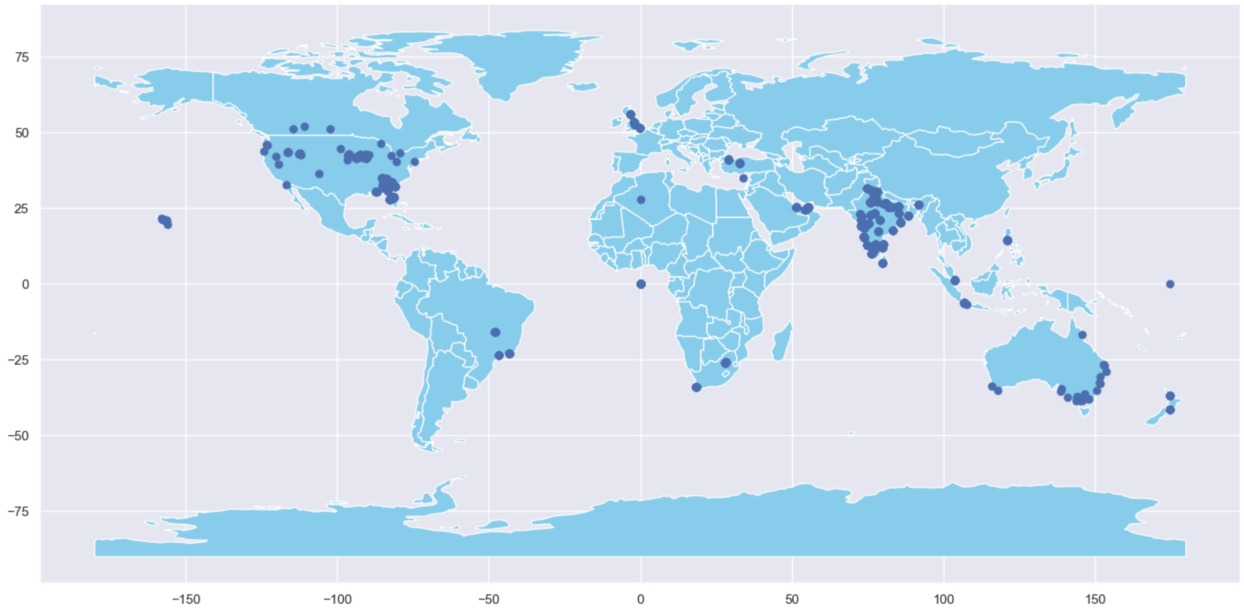
In [2]:

```
from shapely.geometry import point
import geopandas as gpd
from geopandas import GeoDataFrame
```

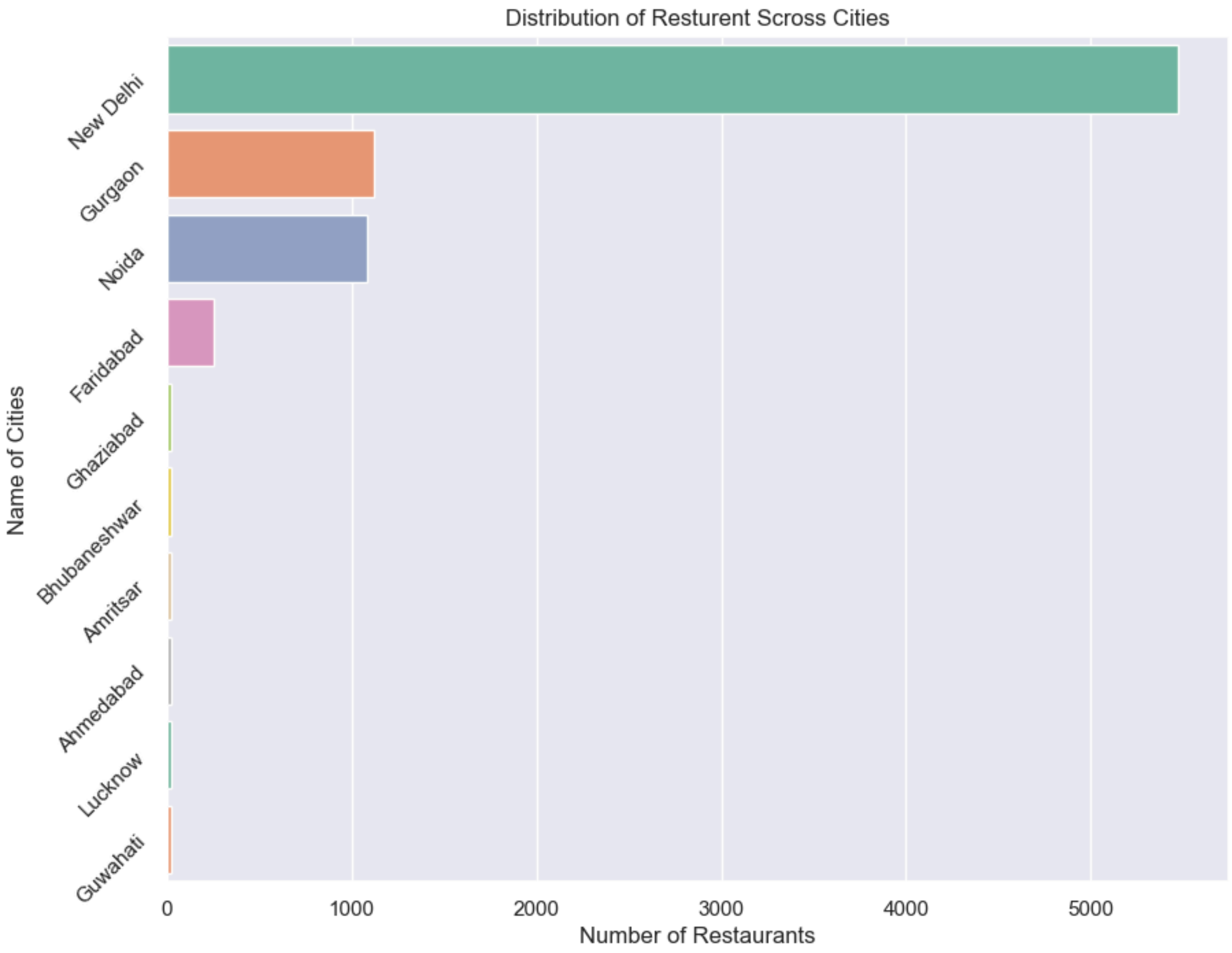
In [11]:

```
gf = gpd.GeoDataFrame(df,geometry=gpd.points_from_xy(df.Longitude, df.Latitude))

world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))

gf.plot(ax=world.plot('continent', legend="True", figsize=(18,15),marker='0', color='skyblu
plt.show()
```

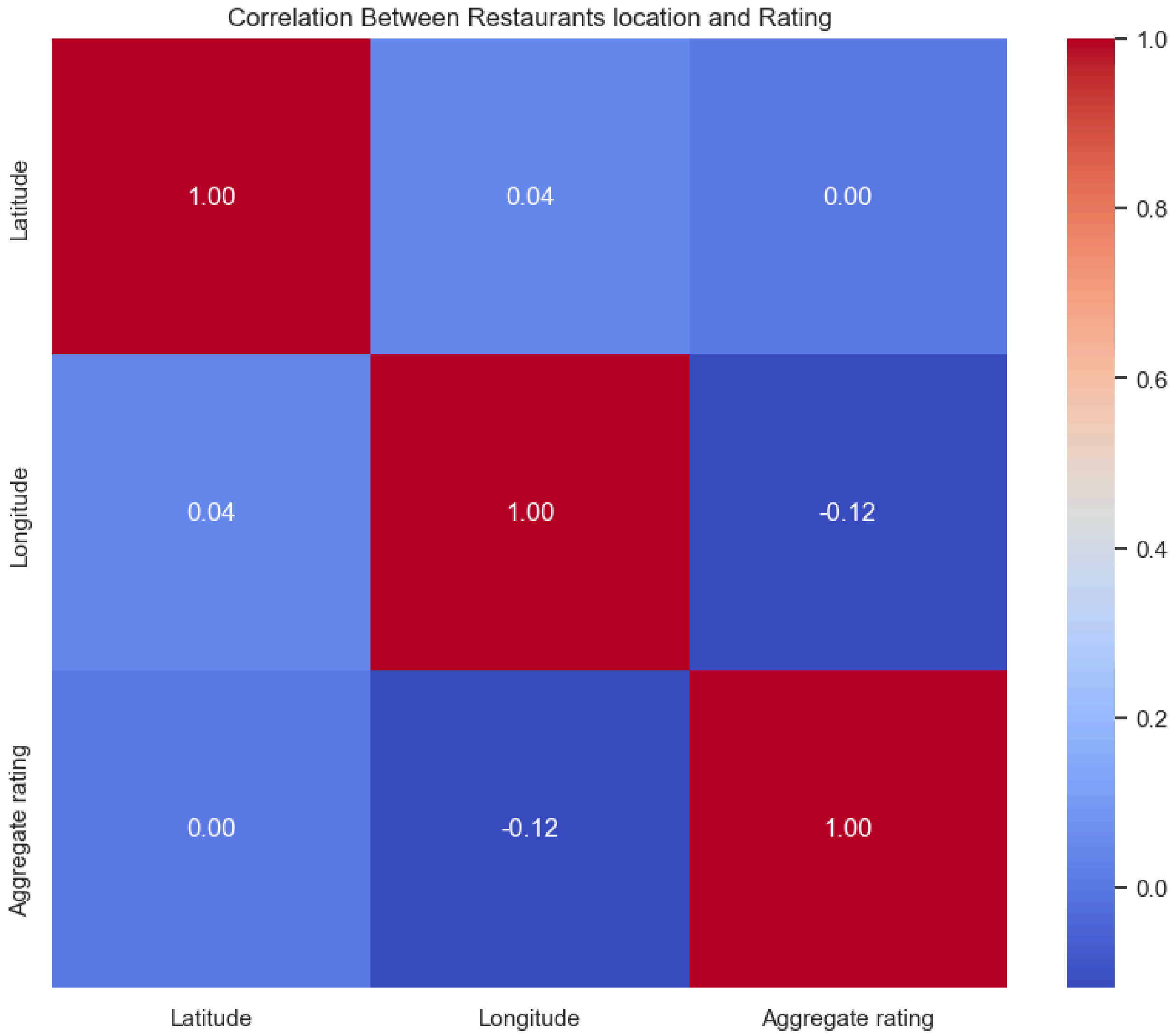
```
C:\Users\Ayush Pallaw\AppData\Local\Temp\ipykernel_14296\3039822377.py:3: FutureWarning: Th
e geopandas.dataset module is deprecated and will be removed in GeoPandas 1.0. You can get
the original 'naturalearth_lowres' data from https://www.naturalearthdata.com/downloads/110
m-cultural-vectors/.
  world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
C:\Users\Ayush Pallaw\AppData\Local\Temp\ipykernel_14296\3039822377.py:5: UserWarning: Only
specify one of 'column' or 'color'. Using 'color'.
  gf.plot(ax=world.plot('continent', legend="True", figsize=(18,15),marker='0', color='skyb
lue',markersize=10))
```



In [13]:
```
plt.figure(figsize=(10,8))
sns.countplot(y = df['City'],order=df.City.value_counts().head(10).index,palette='Set2')
plt.title('Distribution of Resturent Scross Cities')
plt.xlabel('Number of Restaurants')
plt.ylabel('Name of Cities')
plt.yticks(rotation=45)
plt.show()
```



In [15]:
```
plt.figure(figsize=(10,8))
corelatio_matrix = df[['Latitude','Longitude','Aggregate rating']].corr()
plt.title('Correlation Between Restaurants location and Rating')
plt.xlabel('Number of Restaurants')
sns.heatmap(corelatio_matrix, annot = True,cmap= 'coolwarm',fmt=".2f")
plt.show()
```



OBSERVATION :-

◉The restaurant dataset includes information such as restaurant IDs, names, cities, countries, and types of cuisines.

◉The dataset has 9561 rows and 21 columns.

◉There are 9 missing values in the "Cuisines" column, which can be replaced with "Not Specified."

◉There are no duplicates in the dataset.

◉No data type conversion or class balancing is needed.

◉Most restaurants are in Country Code 1, with the next highest number in Country Code 216. Specifically, there are 5473 restaurants in Delhi, 1118 in Gurgaon, and 1080 in Noida.

◉The most common cuisines are "North Indian," "Chinese," and "Fast Food."

◉The USA and India have the most restaurants in this dataset.