ognifyz-internship-tast-level-1-1

May 4, 2024

TASK 1 : DATA EXPLORATION AND PREPROCESSING

```
[1]: import warnings
     warnings.filterwarnings("ignore")
```

```
[2]: import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     %matplotlib inline
     import seaborn as sns
```

```
[3]: df = pd.read_csv("C:/Users/rishi/OneDrive/Documents/cognifyz internship/Dataset␣
       ↪.csv")
```

```
[4]: df.head()
```

```
[4]:    Restaurant ID         Restaurant Name  Country Code              City  \
     0        6317637        Le Petit Souffle           162       Makati City
     1        6304287        Izakaya Kikufuji           162       Makati City
     2        6300002  Heat - Edsa Shangri-La           162  Mandaluyong City
     3        6318506                    Ooma           162  Mandaluyong City
     4        6314302             Sambo Kojin           162  Mandaluyong City

                                                   Address  \
     0  Third Floor, Century City Mall, Kalayaan Avenu…
     1  Little Tokyo, 2277 Chino Roces Avenue, Legaspi…
     2  Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal…
     3  Third Floor, Mega Fashion Hall, SM Megamall, O…
     4  Third Floor, Mega Atrium, SM Megamall, Ortigas…

                                         Locality  \
     0   Century City Mall, Poblacion, Makati City
     1  Little Tokyo, Legaspi Village, Makati City
     2  Edsa Shangri-La, Ortigas, Mandaluyong City
     3      SM Megamall, Ortigas, Mandaluyong City
     4      SM Megamall, Ortigas, Mandaluyong City

                                         Locality Verbose   Longitude   Latitude  \
     0  Century City Mall, Poblacion, Makati City, Mak…  121.027535  14.565443
     1  Little Tokyo, Legaspi Village, Makati City, Ma…  121.014101  14.553708
     2  Edsa Shangri-La, Ortigas, Mandaluyong City, Ma…  121.056831  14.581404
     3  SM Megamall, Ortigas, Mandaluyong City, Mandal…  121.056475  14.585318
     4  SM Megamall, Ortigas, Mandaluyong City, Mandal…  121.057508  14.584450

                                Cuisines  …         Currency Has Table booking  \
     0       French, Japanese, Desserts  …  Botswana Pula(P)               Yes
     1                         Japanese  …  Botswana Pula(P)               Yes
     2  Seafood, Asian, Filipino, Indian  …  Botswana Pula(P)               Yes
     3                  Japanese, Sushi  …  Botswana Pula(P)                No
     4                 Japanese, Korean  …  Botswana Pula(P)               Yes

       Has Online delivery Is delivering now Switch to order menu Price range  \
     0                  No                No                   No           3
     1                  No                No                   No           3
     2                  No                No                   No           4
     3                  No                No                   No           4
     4                  No                No                   No           4

        Aggregate rating  Rating color Rating text Votes
     0               4.8    Dark Green   Excellent   314
     1               4.5    Dark Green   Excellent   591
     2               4.4         Green   Very Good   270
     3               4.9    Dark Green   Excellent   365
     4               4.8    Dark Green   Excellent   229

     [5 rows x 21 columns]
```

[5]:
```
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
```

[6]:
```
df.shape
```

[6]:
```
(9551, 21)
```

[7]:
```
df.isnull().sum()
```

[7]:
```
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
```

[8]:
```
df['Cuisines'].fillna('Not Specified', inplace=True)
```

[9]:
```
df.isnull().sum()
```

[9]:

```
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
```

[10]:

```
dup= df.duplicated().sum()
print(f'number of Duplicated Rows are {dup}')
```

```
number of Duplicated Rows are 0
```

[11]:

```
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
```

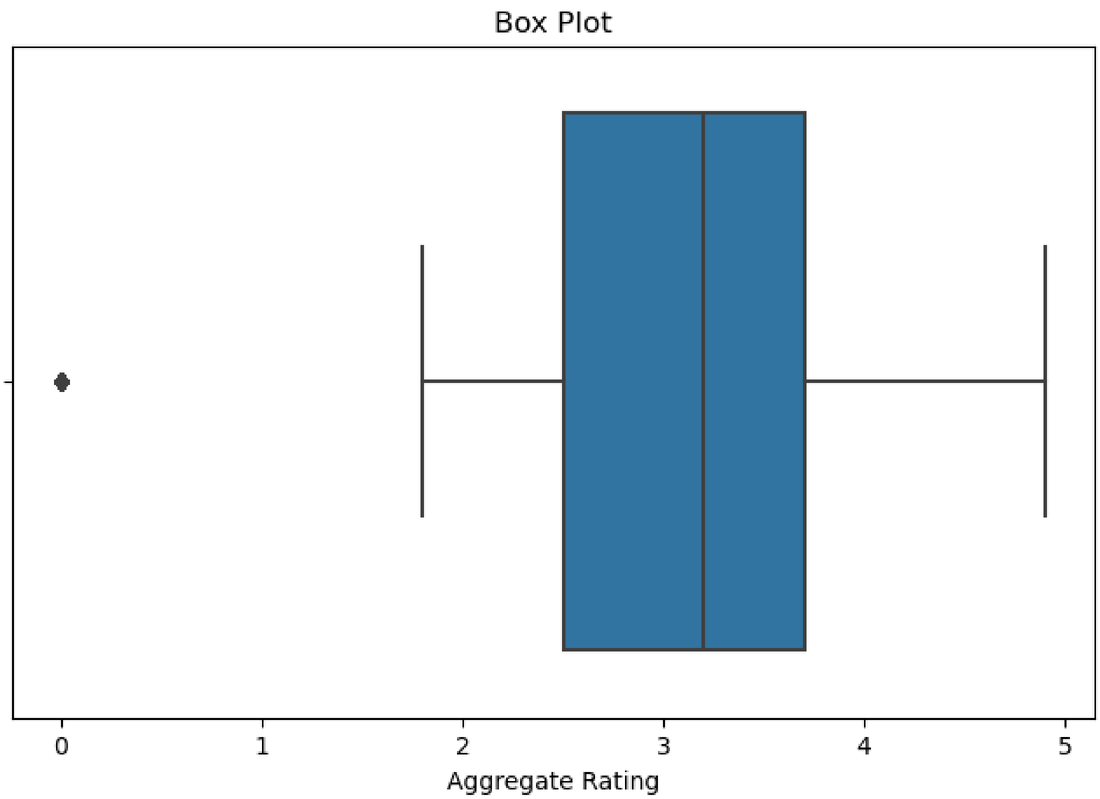
[12]:

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

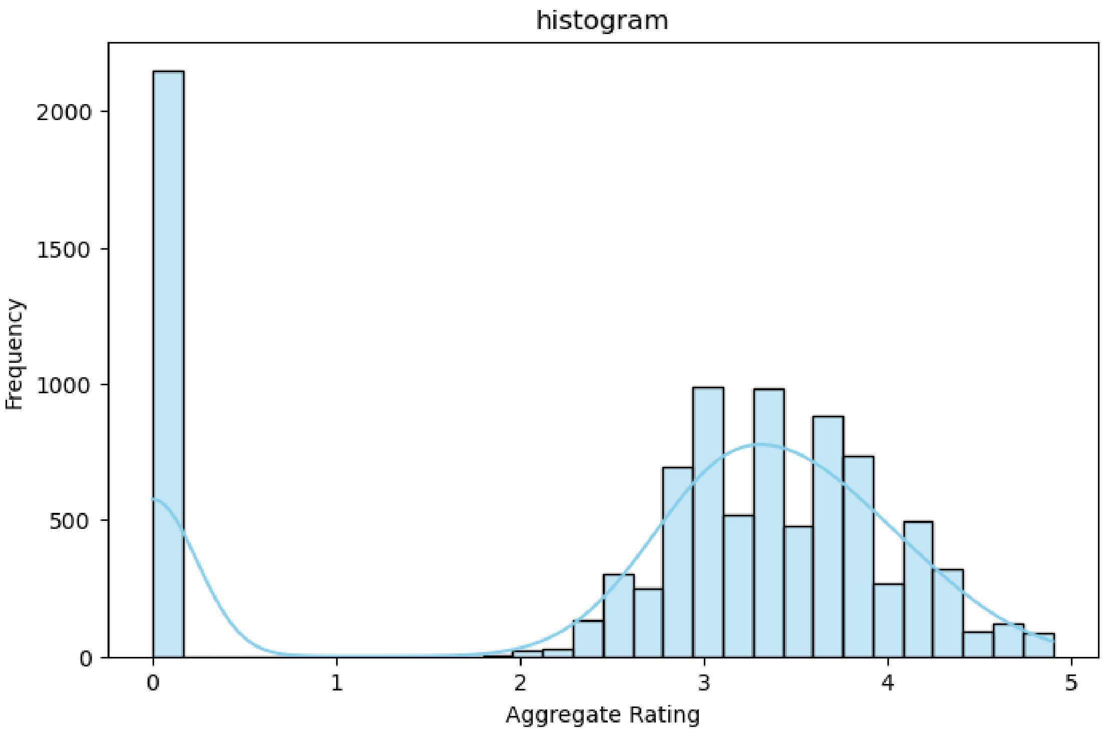
```
count    9551.000000
mean        2.666370
std         1.516378
min         0.000000
25%         2.500000
50%         3.200000
75%         3.700000
max         4.900000
Name: Aggregate rating, dtype: float64
```

[13]:

```
plt.figure(figsize=(8,5))
sns.boxplot(x=df[target])
plt.title('Box Plot')
plt.xlabel('Aggregate Rating')
plt.show()
```



[14]:
```
plt.figure(figsize=(8,5))
sns.histplot(df[target], bins=30, kde=True, color='skyblue')
plt.title('histogram')
plt.xlabel('Aggregate Rating')
plt.ylabel('Frequency')
plt.show()
```



TASK 2 : DESCRIPTIVE ANALYSIS

[15]:
```
df.describe()
```

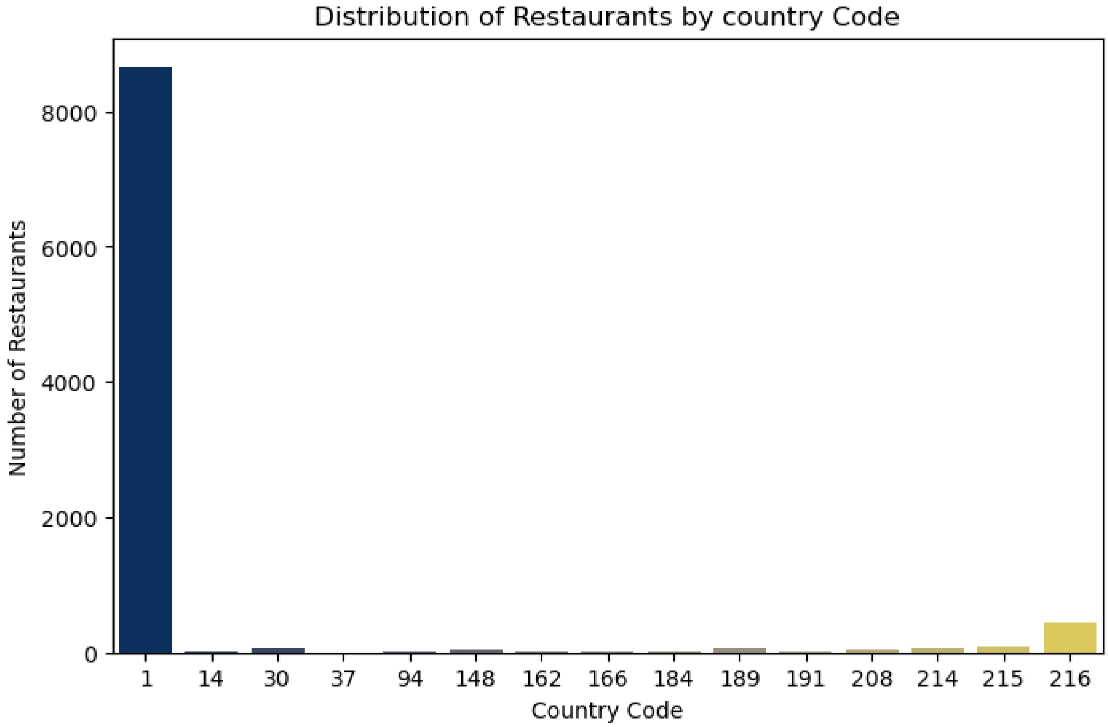
[15]:
```
       Restaurant ID  Country Code   Longitude    Latitude  \
count   9.551000e+03   9551.000000 9551.000000 9551.000000
mean    9.051128e+06     18.365616   64.126574   25.854381
std     8.791521e+06     56.750546   41.467058   11.007935
min     5.300000e+01      1.000000 -157.948486  -41.330428
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
```

```
[16]: df[["Average Cost for two", "Price range", "Aggregate rating", "Votes"]].
       ↪describe()
```

```
[16]:        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
```

```
[17]: plt.figure(figsize=(8,5))
      sns.countplot(x='Country Code' , data=df, palette='cividis')
      plt.title('Distribution of Restaurants by country Code')
      plt.xlabel('Country Code')
      plt.ylabel('Number of Restaurants')
      plt.show()
```

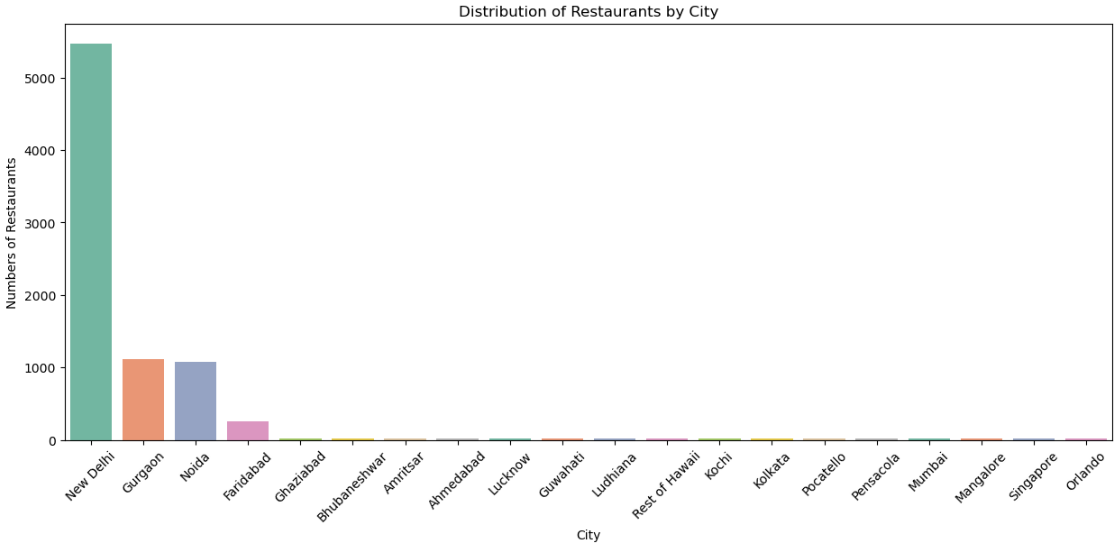


```
[18]: top_countries = df["Country Code"].value_counts().head()
      print('Top 5 Countries with the Highest Numbers of Restaurants:')
      print(top_countries)
```

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

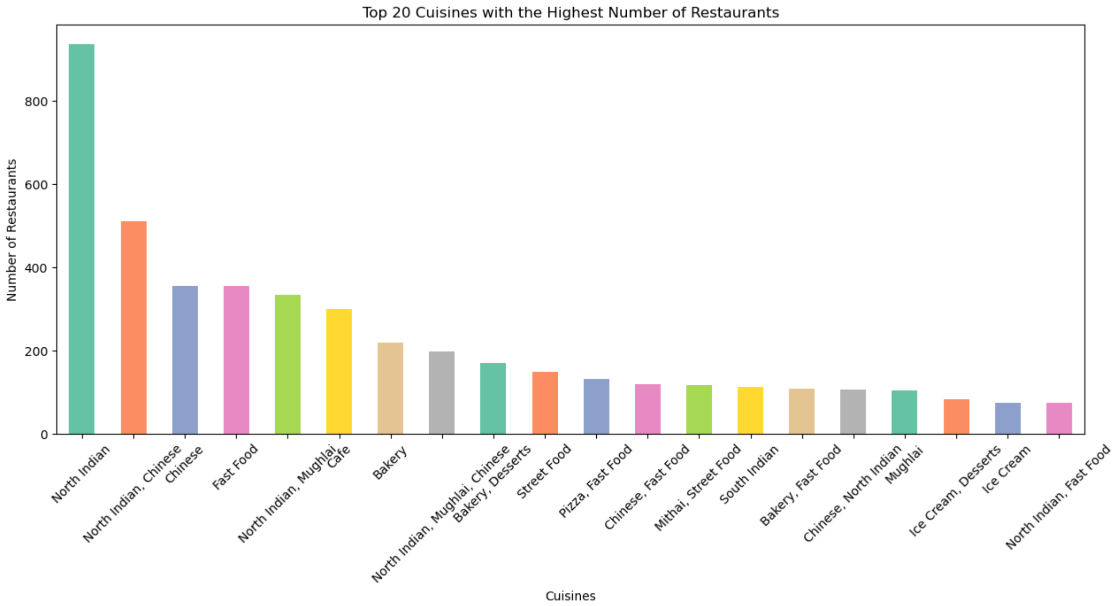
[19]:

```
plt.figure(figsize=(15,6))
sns.countplot(x='City', data=df, order=df['City'].value_counts().head(20).
 ↪index, palette='Set2')
plt.title('Distribution of Restaurants by City')
plt.xlabel('City')
plt.ylabel('Numbers of Restaurants')
plt.xticks(rotation=45)
plt.show()
```



[20]:

```
plt.figure(figsize=(15,6))
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 Restaurants')
plt.xlabel('Cuisines')
plt.ylabel('Number of Restaurants')
plt.xticks(rotation=45)
plt.show()
```



[21]:
```
top_cities = df['City'].value_counts().head(10)
print('Top 10 Cities with the Highest Number of Restaurants:')
print(top_cities)
```

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

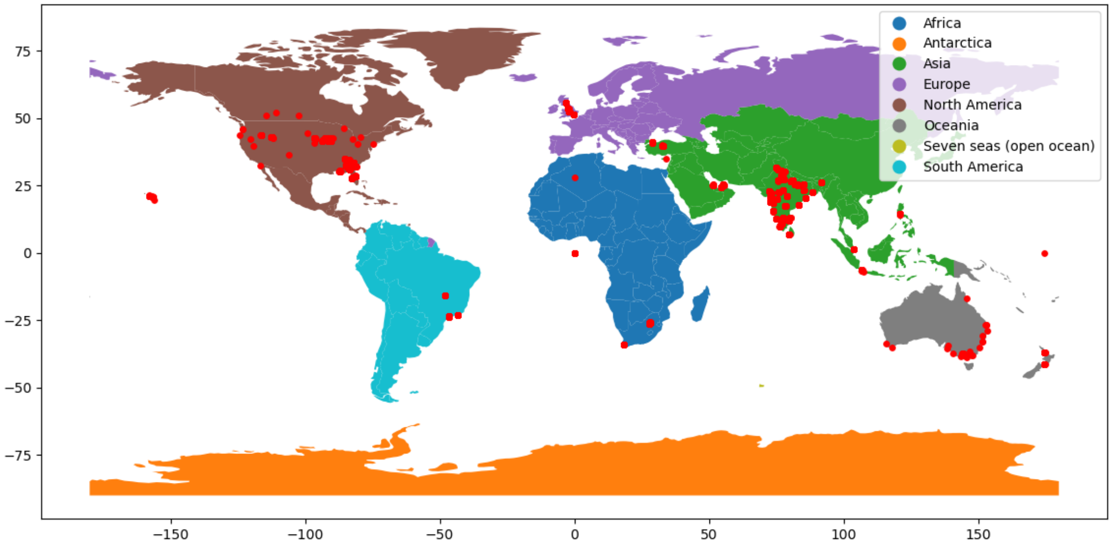
[22]:
```
top_cuisines = cuisines_count.head(10)
print('Top 10 Cuisines with the Highest Number of Restaurants:')
print(top_cuisines)
```

```
Top 10 Cuisines with the Highest Number of Restaurants:
Cuisines
North Indian                          936
North Indian, Chinese                 511
Chinese                               354
Fast Food                             354
```

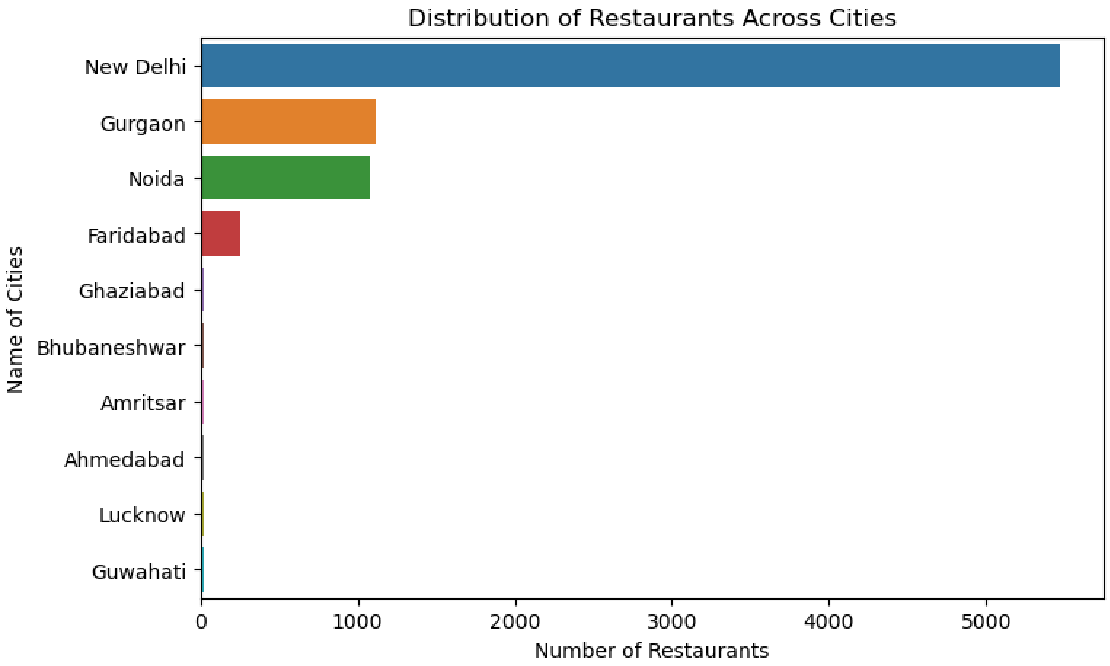
```
North Indian, Mughlai              334
Cafe                               299
Bakery                             218
North Indian, Mughlai, Chinese     197
Bakery, Desserts                   170
Street Food                        149
Name: count, dtype: int64
```

TASK 3 : GEOSPATIAL ANALYSIS

[23]:
```
import shapely.geometry
from shapely.geometry import Point
import geopandas as gpd
from geopandas import GeoDataFrame
gdf = gpd.GeoDataFrame(df,geometry=gpd.points_from_xy(df.Longitude, df.
 ↪Latitude))
world = gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
gdf.plot(ax=world.plot("continent", legend = True, figsize=(14, 12)),␣
 ↪marker='o', color='red', markersize=15)
plt.show()
```



[24]:
```
plt.figure(figsize=(8, 5))
sns.countplot(y = df['City'], order=df.City.value_counts().iloc[:10].index)
plt.xlabel('Number of Restaurants')
plt.ylabel('Name of Cities')
plt.title('Distribution of Restaurants Across Cities')
plt.show()
```



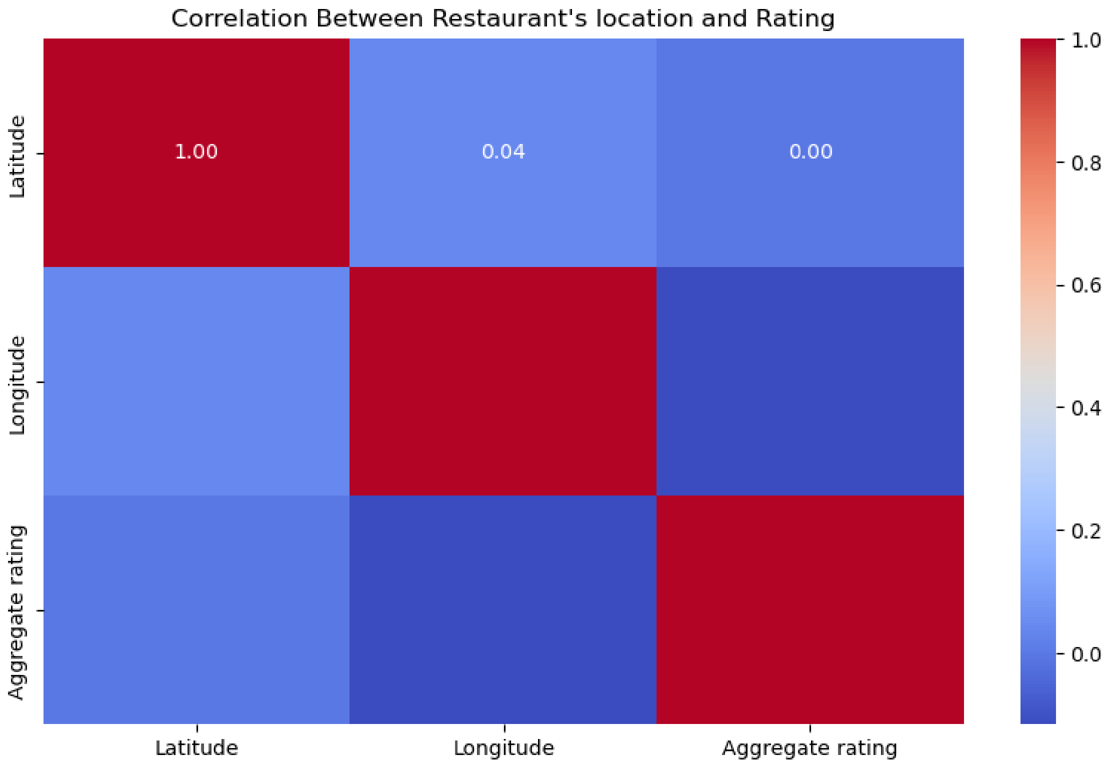
[30]:
```
# Checking correlation between the restaurant's location and its rating
# Set plot size
plt.figure(figsize=(10, 6))

# Calculate the correlation between latitude, longitude, and ratings
correlation_matrix = df[['Latitude', 'Longitude', 'Aggregate rating']].corr()

# Create a heatmap to visualize the correlation
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")

# Set Title
plt.title("Correlation Between Restaurant's location and Rating")

# Display Chart
plt.show()
```


Correlation Between Restaurant's location and Rating
Latitude
Longitude
Aggregate rating
1.00
0.04
0.00
Latitude
Longitude
Aggregate rating
1.0
0.8
0.6
0.4
0.2
0.0

```
[ ]:
```