

```
# Import Libraries
import pandas as pd
import matplotlib.pyplot as plt
import folium
```

```
#!pip install folium
```

```
# Reading the DataFrame
df = pd.read_csv('/content/Dataset .csv')
```

```
df.head(5)
```

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Current
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	14.565443	French, Japanese, Desserts	...	Botswai Pula(
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	14.553708	Japanese	...	Botswai Pula(
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	14.581404	Seafood, Asian, Filipino, Indian	...	Botswai Pula(
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	14.585318	Japanese, Sushi	...	Botswai Pula(
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	14.584450	Japanese, Korean	...	Botswai Pula(

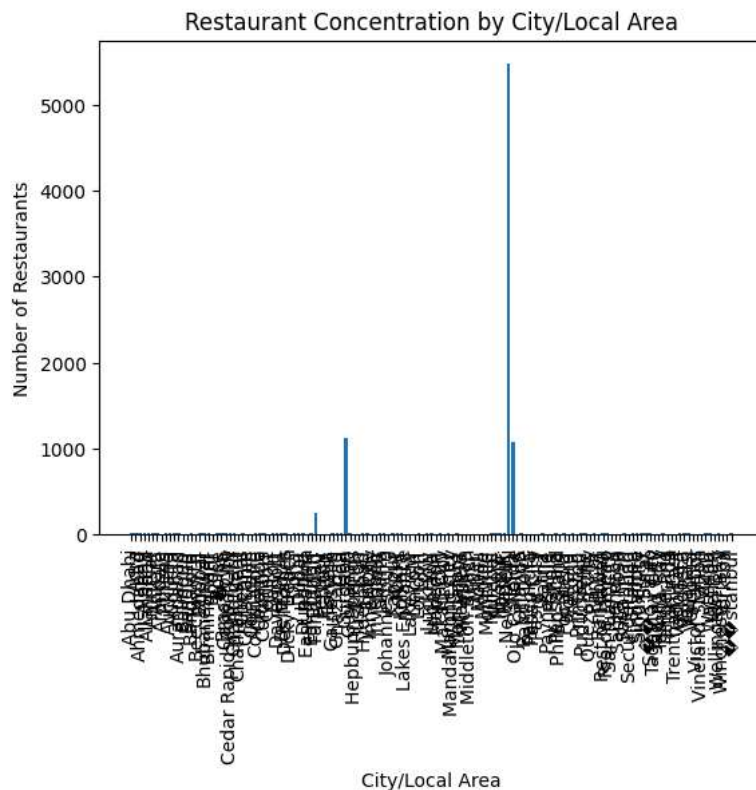
5 rows × 21 columns

```
# Using Group by function, grouping the City Column
grouped_by_city = df.groupby('City')
```

```
# Count of Restaurant
restaurant_count = grouped_by_city['Restaurant Name'].count()
```

```
import matplotlib.pyplot as plt

plt.bar(restaurant_count.index, restaurant_count.values)
plt.xlabel('City/Local Area')
plt.ylabel('Number of Restaurants')
plt.title('Restaurant Concentration by City/Local Area')
plt.xticks(rotation=90)
plt.show()
```



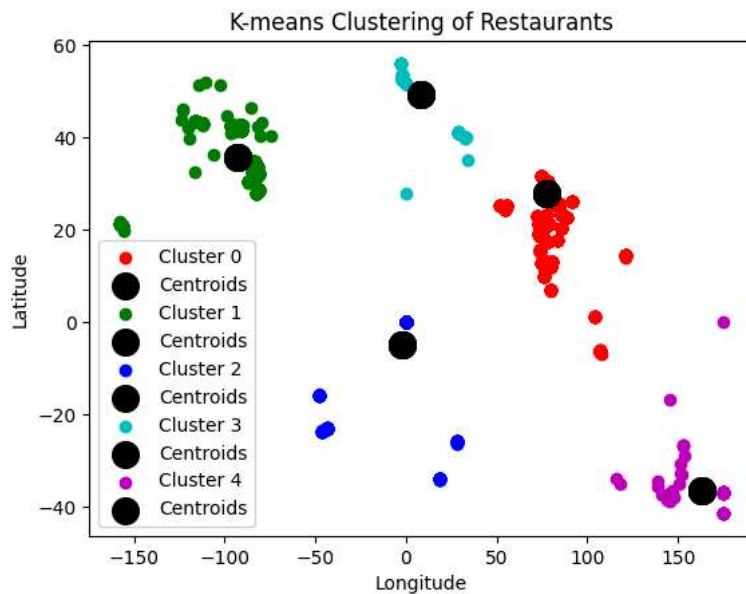
```
# Forming into Clusters
from sklearn.cluster import KMeans

X = df[['Latitude', 'Longitude']]
k = 5
kmeans = KMeans(n_clusters=k, random_state=0)
cluster_labels = kmeans.fit_predict(X)
df['Cluster'] = cluster_labels
```

```
colors = ['r', 'g', 'b', 'c', 'm', 'y', 'k']

for cluster_num in range(k):
    cluster_data = df[df['Cluster'] == cluster_num]
    plt.scatter(cluster_data['Longitude'], cluster_data['Latitude'],
                c=colors[cluster_num], label=f'Cluster {cluster_num}')
    plt.scatter(kmeans.cluster_centers[:, 1], kmeans.cluster_centers[:, 0],
                s=200, c='black', label='Centroids')
    plt.xlabel('Longitude')
    plt.ylabel('Latitude')
    plt.title('K-means Clustering of Restaurants')
    plt.legend()

plt.show()
```

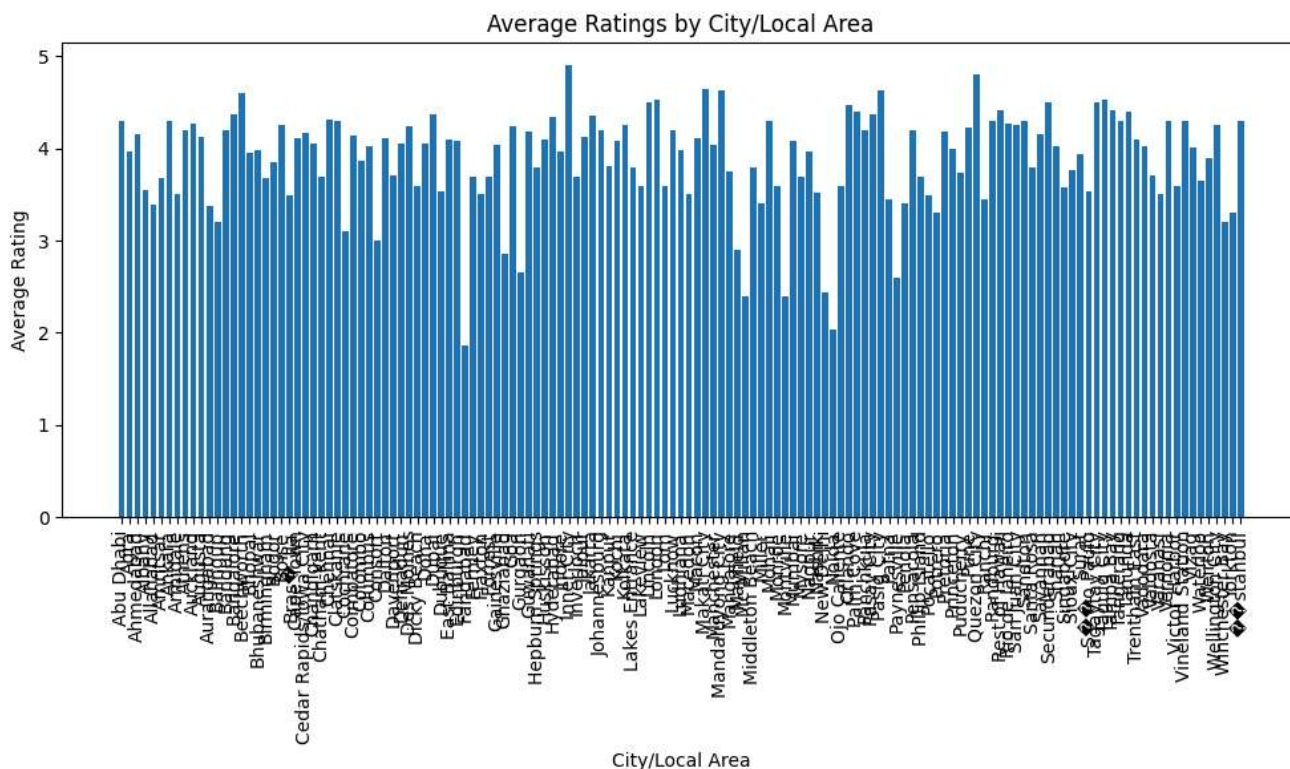


```
average_ratings_by_city = grouped_by_city['Aggregate rating'].mean()
print(average_ratings_by_city)
```

```
City
Abu Dhabi      4.300000
Agra           3.965000
Ahmedabad      4.161905
Albany         3.555000
Allahabad      3.395000
...
Weirton        3.900000
Wellington City 4.250000
Winchester Bay 3.200000
Yorkton        3.300000
📍📍istanbul    4.292857
Name: Aggregate rating, Length: 141, dtype: float64
```

```
plt.figure(figsize=(10, 6))

plt.bar(average_ratings_by_city.index, average_ratings_by_city.values)
plt.xlabel('City/Local Area')
plt.ylabel('Average Rating')
plt.title('Average Ratings by City/Local Area')
plt.xticks(rotation=90) # Rotate x-axis labels for better readability
plt.tight_layout()
plt.show()
```



```
popular_cuisines_by_city = grouped_by_city['Cuisines'].agg(lambda x: x.mode().tolist())
print(popular cuisines by city)
```

```
City
Abu Dhabi                [American, Indian, Italian, Pizza]
Agra                      [North Indian, Mughlai]
Ahmedabad                [Cafe, American, Continental, Armenian, Fast F...
Albany                   [Japanese, Steak, Sushi]
Allahabad                [North Indian, Chinese]
...
Weirton                  [Burger, Greek, Sandwich]
Wellington City          [Cafe]
Winchester Bay           [Burger, Seafood, Steak]
Yorkton                  [Asian]
🇹🇹stanbul                [Cafe]
Name: Cuisines, Length: 141, dtype: object
```

```
plt.figure(figsize=(10, 6))

for city, cuisines in popular_cuisines_by_city.items():
    plt.bar(city, ', '.join(cuisines))
    plt.xlabel('City/Local Area')
    plt.ylabel('Popular Cuisines')
    plt.title('Popular Cuisines by City/Local Area')
    plt.xticks(rotation=90) # Rotate x-axis labels for better readability
    plt.tight_layout()

plt.show()
```

```
/tmp/ipython-input-3718216243.py:9: UserWarning: Tight layout not applied. The left and right margins cannot be made large enough
plt.tight_layout()
```



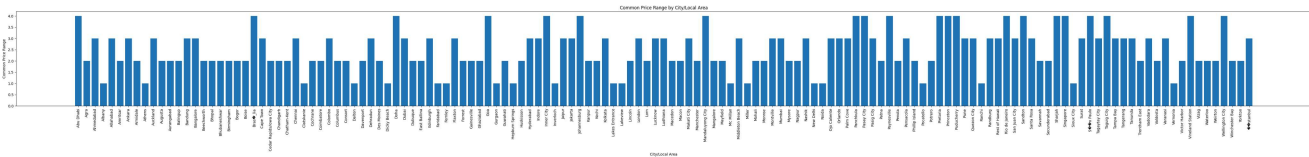
```
common_price_range_by_city = grouped_by_city['Price range'].agg(lambda x: x.mode().iloc[0])
print(common_price_range_by_city)
```

City	
Abu Dhabi	4

```
Agra                2
Ahmedabad           3
Albany              1
Allahabad           3
..
Weirton             2
Wellington City     4
Winchester Bay      2
Yorkton             2
🇹🇷istanbul          3
Name: Price range, Length: 141, dtype: int64
```

```
plt.figure(figsize=(50, 6))

plt.bar(common_price_range_by_city.index, common_price_range_by_city.values)
plt.xlabel('City/Local Area')
plt.ylabel('Common Price Range')
plt.title('Common Price Range by City/Local Area')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```



```
grouped_by_city = df.groupby('City')
cuisine_diversity = grouped_by_city['Cuisines'].apply(lambda x: len(set(x)))
print(cuisine_diversity)
```

```
City
Abu Dhabi          17
Agra                15
Ahmedabad           21
Albany              17
Allahabad           18
..
Weirton             1
Wellington City     17
Winchester Bay      1
Yorkton             1
🇹🇷istanbul          11
Name: Cuisines, Length: 141, dtype: int64
```

```
plt.figure(figsize=(10, 6))

plt.bar(cuisine_diversity.index, cuisine_diversity.values)
plt.xlabel('City/Local Area')
plt.ylabel('Cuisine Diversity (Number of Unique Cuisines)')
plt.title('Cuisine Diversity by City/Local Area')
plt.xticks(rotation=90)
plt.tight_layout()
plt.show()
```

