Sejal Sanas

Perform a geographical analysis of the restaurants in the dataset.

In [3]:
```
# Import Libraries
import pandas as pd
import matplotlib.pyplot as plt
import folium
```

In [2]:
```
pip install folium
```

```
Collecting folium
  Obtaining dependency information for folium from https://files.pythonhosted.org/packages/87/1f/b053
4aa3c537bc4477f371de1966b92c041816e5f943dcf73298c1f1b8a5/folium-0.19.3-py2.py3-none-any.whl.metadata
(https://files.pythonhosted.org/packages/87/1f/b0534aa3c537bc4477f371de1966b92c041816e5f943dcf73298c1
f1b8a5/folium-0.19.3-py2.py3-none-any.whl.metadata)
  Downloading folium-0.19.3-py2.py3-none-any.whl.metadata (3.8 kB)
Collecting branca>=0.6.0 (from folium)
  Obtaining dependency information for branca>=0.6.0 from https://files.pythonhosted.org/packages/f
8/9d/91cddd38bd00170aad1a4b198c47b4ed716be45c234e09b835af41f4e717/branca-0.8.1-py3-none-any.whl.metad
ata (https://files.pythonhosted.org/packages/f8/9d/91cddd38bd00170aad1a4b198c47b4ed716be45c234e09b835
af41f4e717/branca-0.8.1-py3-none-any.whl.metadata)
  Downloading branca-0.8.1-py3-none-any.whl.metadata (1.5 kB)
Requirement already satisfied: jinja2>=2.9 in c:\users\shejal sanas\anaconda3\lib\site-packages (from
folium) (3.1.2)
Requirement already satisfied: numpy in c:\users\shejal sanas\anaconda3\lib\site-packages (from foliu
m) (1.24.3)
Requirement already satisfied: requests in c:\users\shejal sanas\anaconda3\lib\site-packages (from fo
lium) (2.31.0)
Requirement already satisfied: xyzservices in c:\users\shejal sanas\anaconda3\lib\site-packages (from
folium) (2022.9.0)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\shejal sanas\anaconda3\lib\site-packages
(from jinja2>=2.9->folium) (2.1.1)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\shejal sanas\anaconda3\lib\site-p
ackages (from requests->folium) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\shejal sanas\anaconda3\lib\site-packages (fro
m requests->folium) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\shejal sanas\anaconda3\lib\site-package
s (from requests->folium) (1.26.16)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\shejal sanas\anaconda3\lib\site-package
s (from requests->folium) (2023.7.22)
Downloading folium-0.19.3-py2.py3-none-any.whl (110 kB)
   ---------------------------------------- 0.0/110.5 kB ? eta -:--:--
   ---------------------------------------- 0.0/110.5 kB ? eta -:--:--
   ---------------------------------------- 0.0/110.5 kB ? eta -:--:--
   ---------------------------------------- 0.0/110.5 kB ? eta -:--:--
   ------- -------------------------------- 20.5/110.5 kB ? eta -:--:--
   ------- -------------------------------- 20.5/110.5 kB ? eta -:--:--
   ------- -------------------------------- 20.5/110.5 kB ? eta -:--:--
   -------------- ------------------------ 41.0/110.5 kB 217.9 kB/s eta 0:00:01
   -------------- ------------------------ 41.0/110.5 kB 217.9 kB/s eta 0:00:01
   --------------------- ----------------- 61.4/110.5 kB 218.8 kB/s eta 0:00:01
   --------------------- ----------------- 61.4/110.5 kB 218.8 kB/s eta 0:00:01
   -------------------------------- ------ 92.2/110.5 kB 275.8 kB/s eta 0:00:01
   -------------------------------- ------ 92.2/110.5 kB 275.8 kB/s eta 0:00:01
   -------------------------------- ------ 92.2/110.5 kB 275.8 kB/s eta 0:00:01
   -------------------------------- ------ 92.2/110.5 kB 275.8 kB/s eta 0:00:01
   -------------------------------- ------ 92.2/110.5 kB 275.8 kB/s eta 0:00:01
   -------------------------------- ------ 92.2/110.5 kB 275.8 kB/s eta 0:00:01
   -------------------------------- ------ 92.2/110.5 kB 275.8 kB/s eta 0:00:01
   -------------------------------- ------ 92.2/110.5 kB 275.8 kB/s eta 0:00:01
   -------------------------------- ------ 92.2/110.5 kB 275.8 kB/s eta 0:00:01
   -------------------------------- ------ 92.2/110.5 kB 275.8 kB/s eta 0:00:01
   -------------------------------- ------ 92.2/110.5 kB 275.8 kB/s eta 0:00:01
   -------------------------------- ------ 92.2/110.5 kB 275.8 kB/s eta 0:00:01
   -------------------------------- ------ 92.2/110.5 kB 275.8 kB/s eta 0:00:01
   -------------------------------- ------ 92.2/110.5 kB 275.8 kB/s eta 0:00:01
   -------------------------------- ------ 92.2/110.5 kB 275.8 kB/s eta 0:00:01
   ------------------------------------ -- 102.4/110.5 kB 88.0 kB/s eta 0:00:01
   --------------------------------------- 110.5/110.5 kB 93.0 kB/s eta 0:00:00
Downloading branca-0.8.1-py3-none-any.whl (26 kB)
Installing collected packages: branca, folium
Successfully installed branca-0.8.1 folium-0.19.3
Note: you may need to restart the kernel to use updated packages.
```

In [4]:
```
# Reading the DataFrame
df = pd.read_csv(r'C:/Users/Shejal Sanas/Downloads/Dataset.csv')
```

In [5]:
```
df.head(5)
```

Out[5]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...
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	...
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	...
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	...
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	...
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	...

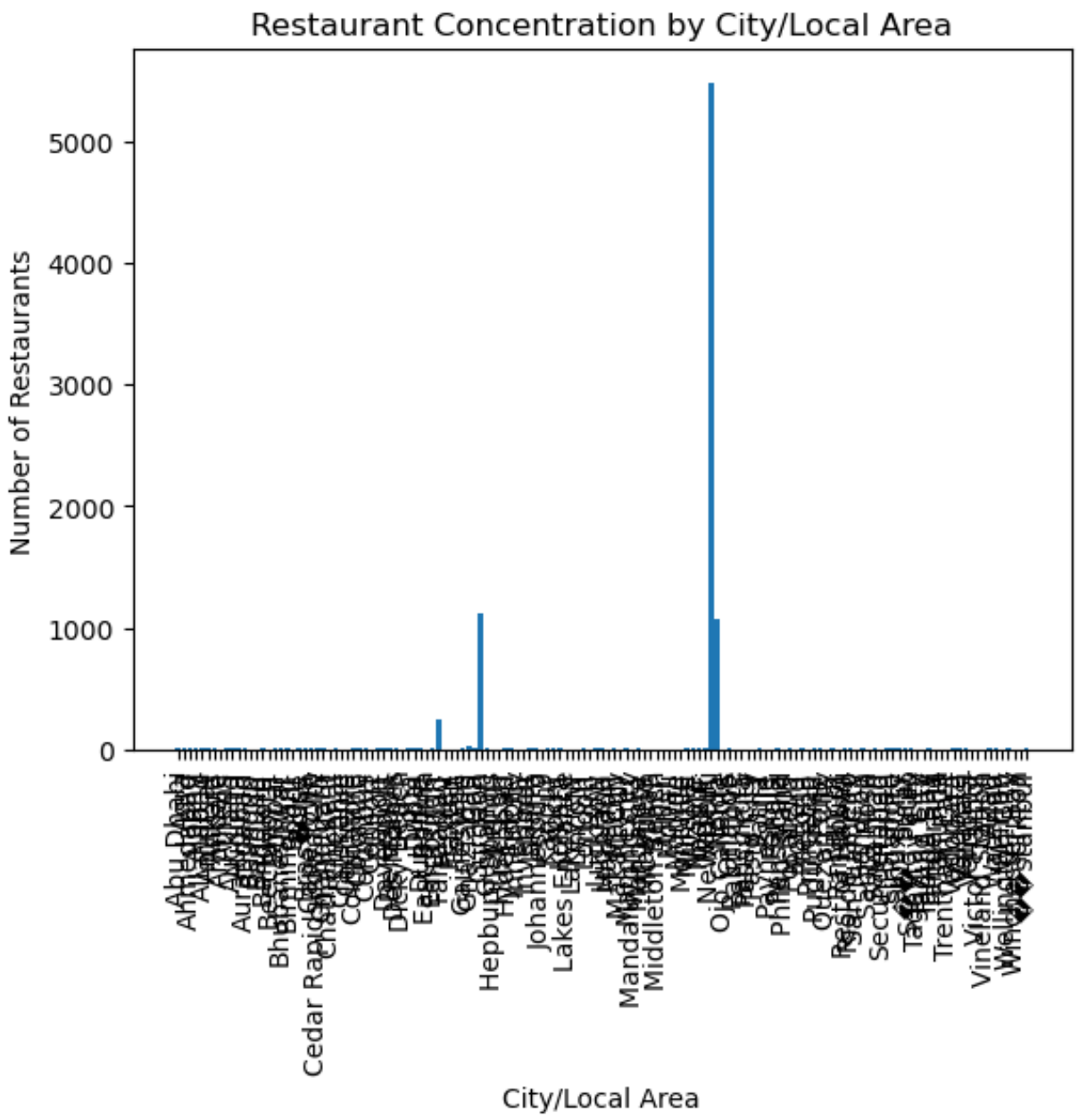
5 rows × 21 columns

In [6]:
```
# Using Group by function, grouping the City Column
grouped_by_city = df.groupby('City')
```

In [7]:
```
# Count of Restaurant
restaurant_count = grouped_by_city['Restaurant Name'].count()
```

In [8]:
```
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()
```



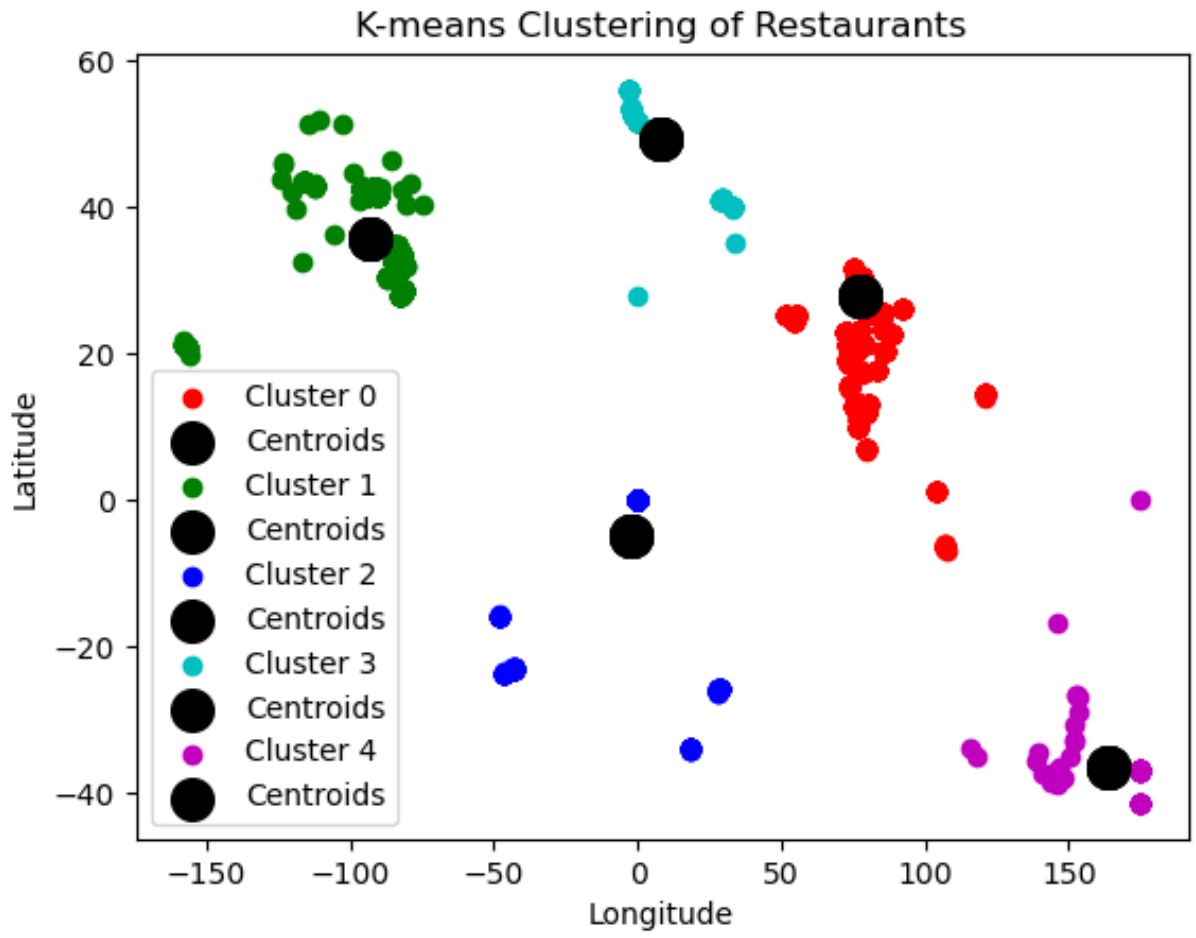
In [9]:
```
# Forming into Clusters
from sklearn.cluster import KMeans

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

```
C:\Users\Shejal Sanas\anaconda3\Lib\site-packages\sklearn\cluster\_kmeans.py:1412: FutureWarning: The
default value of `n_init` will change from 10 to 'auto' in 1.4. Set the value of `n_init` explicitly
to suppress the warning
  super()._check_params_vs_input(X, default_n_init=10)
```

In [10]:
```
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()
```



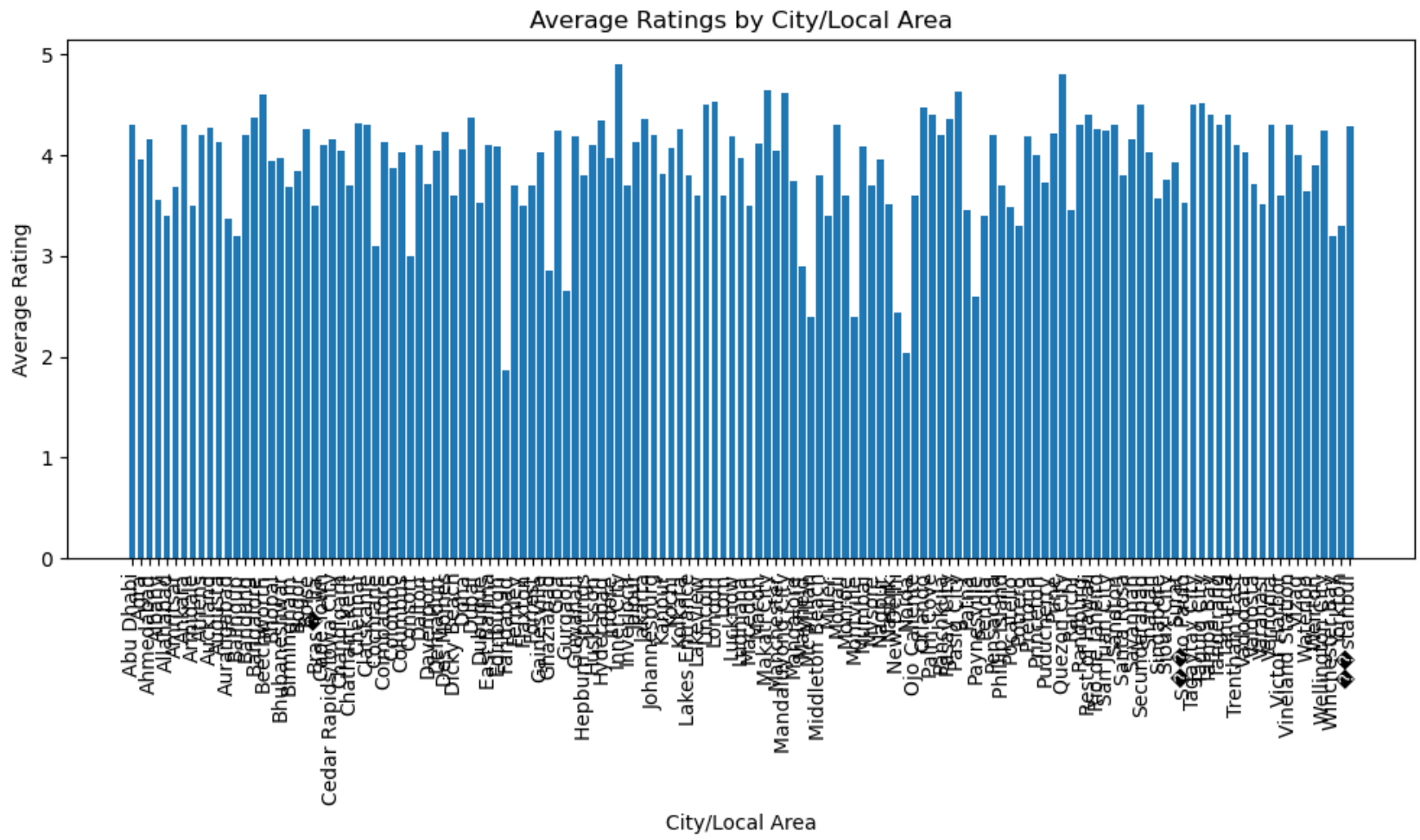
Average Ratings by City/Locality

In [11]:
```
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
��stanbul          4.292857
Name: Aggregate rating, Length: 141, dtype: float64
```

In [12]:
```
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/Locality

In [13]:
```
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]
ÔÔstanbul                                                  [Cafe]
Name: Cuisines, Length: 141, dtype: object
```

In [14]:
```
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()
```

```
C:\Users\Shejal Sanas\AppData\Local\Temp\ipykernel_27988\3718216243.py:9: UserWarning: Tight layout n
ot applied. The left and right margins cannot be made large enough to accommodate all axes decoration
s.
  plt.tight_layout()
```

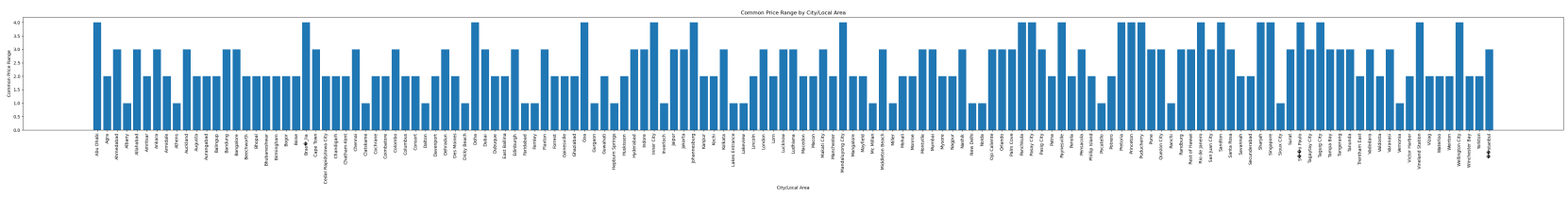
Price Range Analysis by City/Locality

In [15]:
```
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
��stanbul          3
Name: Price range, Length: 141, dtype: int64
```

In [16]:
```
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()
```



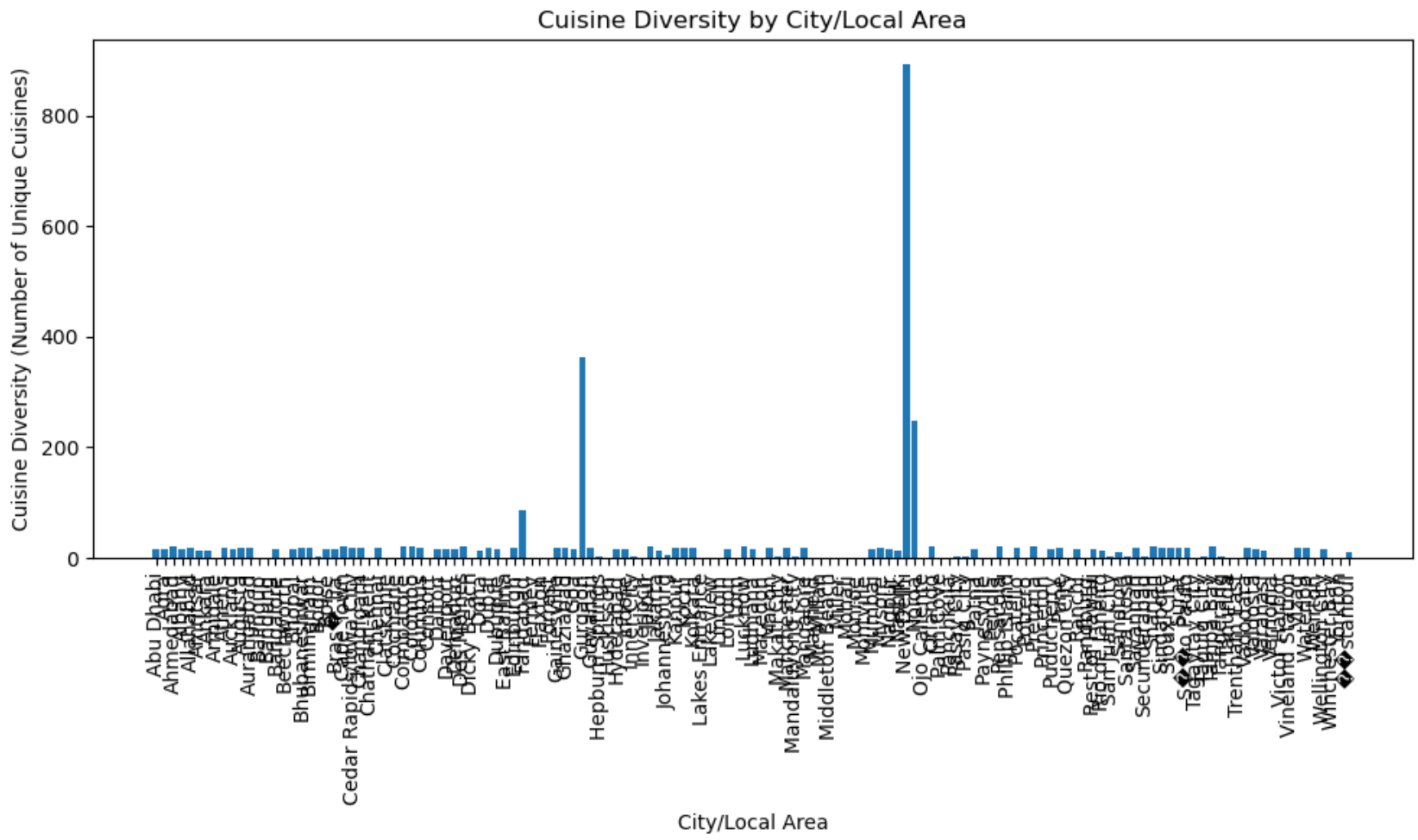
Cuisine Diversity

In [17]:
```
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
��stanbul          11
Name: Cuisines, Length: 141, dtype: int64
```

In [18]:
```
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()
```



Conclusion: Performed a geographical analysis of the restaurants based on Average Ratings, Cuisine Ratings and Cuisines Diversity by City/Locality.