In [5]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

data processing

In [7]:
```
df= pd.read_csv("Dataset .csv")
df
```

Out[7]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	
0	6317637	Le Petit Souffle	162	Makati City	Third Floor, Century City Mall, Kalayaan Avenu...	Century City Mall, Poblacion, Makati City	Ma
1	6304287	Izakaya Kikufuji	162	Makati City	Little Tokyo, 2277 Chino Roces Avenue, Legaspi...	Little Tokyo, Legaspi Village, Makati City	Le
2	6300002	Heat - Edsa Shangri-La	162	Mandaluyong City	Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...	Edsa Shangri-La, Ortigas, Mandaluyong City	Eds N
3	6318506	Ooma	162	Mandaluyong City	Third Floor, Mega Fashion Hall, SM Megamall, O...	SM Megamall, Ortigas, Mandaluyong City	S N C
4	6314302	Sambo Kojin	162	Mandaluyong City	Third Floor, Mega Atrium, SM Megamall, Ortigas...	SM Megamall, Ortigas, Mandaluyong City	S N C
...	...	...	...	...	...	...	
9546	5915730	Naml\ Gurme	208	××stanbul	Kemanke×× Karamustafa Pa××a Mahallesi, R\ht\m ...	Karak×_y	
9547	5908749	Ceviz A××ac\	208	××stanbul	Ko××uyolu Mahallesi, Muhittin ××st×_nda×× Cadd...	Ko××uyolu	
9548	5915807	Huqqa	208	××stanbul	Kuru×_e××me Mahallesi, Muallim Naci Caddesi, N...	Kuru×_e××me	Kur
9549	5916112	A×××k Kahve	208	××stanbul	Kuru×_e××me Mahallesi, Muallim Naci Caddesi, N...	Kuru×_e××me	Kur
9550	5927402	Walter's Coffee Roastery	208	××stanbul	Cafea××a Mahallesi, Bademalt\	Moda	

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality
					Sokak, No 21/B, ...	

9551 rows × 21 columns

In [8]:
```
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 [12]:
```
# while looking at the cusine column we see that there is multiple cusisines in sam
```

In [14]:
```
df['Cuisines'] = df['Cuisines'].str.split(', ')
df=df.explode('Cuisines').reset_index(drop=True)
df.head()
```

Out[14]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	L
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...	1
1	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...	1
2	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...	1
3	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...	1
4	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...	1

5 rows × 21 columns

In [16]:

```
df.info()
#here we can see that our number of rows got increased
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19719 entries, 0 to 19718
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Restaurant ID         19719 non-null  int64
 1   Restaurant Name       19719 non-null  object
 2   Country Code          19719 non-null  int64
 3   City                  19719 non-null  object
 4   Address               19719 non-null  object
 5   Locality              19719 non-null  object
 6   Locality Verbose      19719 non-null  object
 7   Longitude             19719 non-null  float64
 8   Latitude              19719 non-null  float64
 9   Cuisines              19710 non-null  object
 10  Average Cost for two  19719 non-null  int64
 11  Currency              19719 non-null  object
 12  Has Table booking     19719 non-null  object
 13  Has Online delivery   19719 non-null  object
 14  Is delivering now     19719 non-null  object
 15  Switch to order menu  19719 non-null  object
 16  Price range           19719 non-null  int64
 17  Aggregate rating      19719 non-null  float64
 18  Rating color          19719 non-null  object
 19  Rating text           19719 non-null  object
 20  Votes                 19719 non-null  int64
dtypes: float64(3), int64(5), object(13)
memory usage: 3.2+ MB
```

Level (1)

task 1 :Top Cuisines

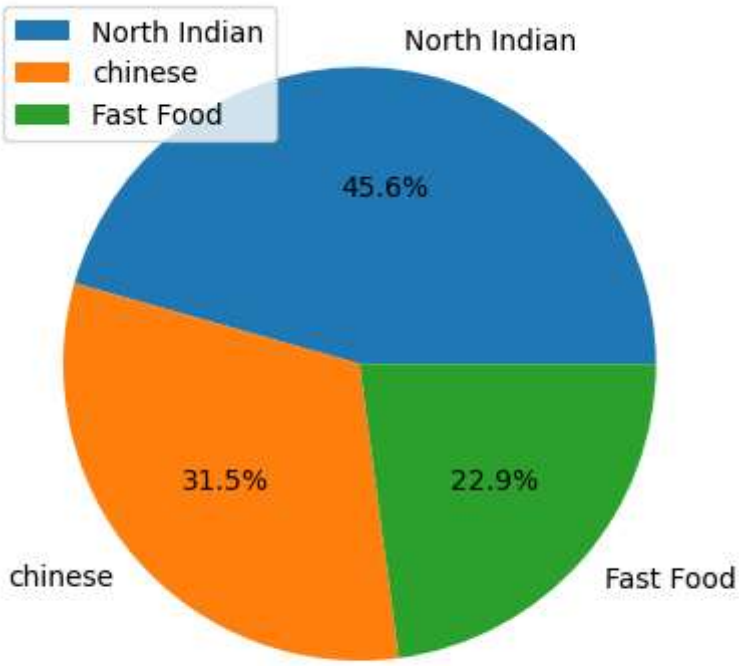
Determine the top three most common cuisines in the dataset.

In [21]:
```
top_3_cuisines= df['Cuisines'].value_counts().head(3)
print(f'Top three  {top_3_cuisines}')
```

```
Top three  Cuisines
North Indian    3960
Chinese         2735
Fast Food       1986
Name: count, dtype: int64
```

In [23]:
```
lb=['North Indian','chinese','Fast Food']
plt.pie(top_3_cuisines,labels=lb,autopct='%0.1f%%')
plt.legend()
```

Out[23]: <matplotlib.legend.Legend at 0x1ad04cff560>



Calculate the percentage of restaurants that serve each of the top cuisines.

In [26]:
```
total_restaurants = df["Restaurant Name"].nunique()
print(f'There are {total_restaurants} restaurants')
```

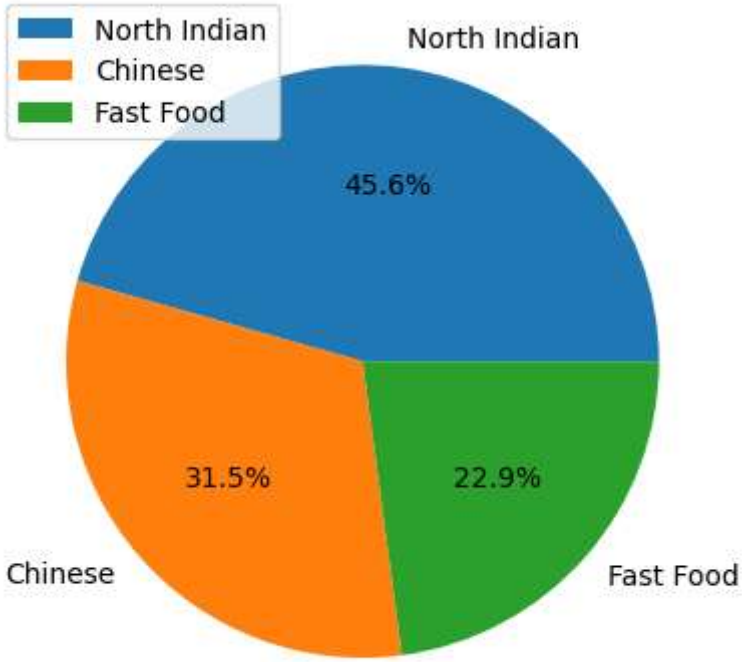
```
There are 7446 restaurants
```

In [28]:
```
top_cuisines_percentage=(top_3_cuisines/total_restaurants)*100
print( top_cuisines_percentage)
```

```
Cuisines
North Indian    53.182917
Chinese         36.731131
Fast Food       26.672039
Name: count, dtype: float64
```

In [30]:
```
lb=['North Indian','Chinese','Fast Food']
plt.pie(top_cuisines_percentage, labels=lb,autopct='%0.1f%%')
plt.legend()
 # ni dalna hai isko
```

Out[30]: <matplotlib.legend.Legend at 0x1ad04a35fa0>



task 2: City Analysis

In [33]:
```
print(df.columns)
```

```
Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Address',
       'Locality', 'Locality Verbose', 'Longitude', 'Latitude', 'Cuisines',
       'Average Cost for two', 'Currency', 'Has Table booking',
       'Has Online delivery', 'Is delivering now', 'Switch to order menu',
       'Price range', 'Aggregate rating', 'Rating color', 'Rating text',
       'Votes'],
      dtype='object')
```

Identify the city with the highest number of restaurants in the dataset.

In [36]:
```
city_count=df.groupby('City')['Restaurant ID'].nunique().sort_values(ascending=Fals
print(city_count)
```

```
City
New Delhi      5473
Gurgaon        1118
Noida          1080
Faridabad       251
Ghaziabad        25
               ...
Dicky Beach       1
Lorn              1
Quezon City       1
Lincoln           1
Lakeview          1
Name: Restaurant ID, Length: 141, dtype: int64
```

In [38]:
```
most_restaurants_city = city_count.idxmax()  # Get the city with the highest count
most_restaurants_count = city_count.max()  # Get the highest count

print(f"City with the most restaurants: {most_restaurants_city} ({most_restaurants_
```

```
City with the most restaurants: New Delhi (5473 restaurants)
```

Calculate the average rating for restaurants in each city.

In [41]:
```
avg_rating = df.groupby('City')['Aggregate rating'].mean().sort_values(ascending =
print("Average ratings for each city:\n", avg_rating)
```

```
Average ratings for each city:
 City
Inner City          4.900000
Quezon City         4.800000
Makati City         4.725000
Mandaluyong City    4.600000
Beechworth          4.600000
                      ...
Paynesville         2.600000
Mc Millan           2.400000
Montville           2.400000
Noida               2.231468
Faridabad           2.021443
Name: Aggregate rating, Length: 141, dtype: float64
```

Find the City with the Highest Average Rating

In [44]:
```
best_avg_rating_city = avg_rating.idxmax()
best_avg_rating_value =avg_rating.max()
print(f"City with the highest average rating: {best_avg_rating_city} ({best_avg_rat
```

```
City with the highest average rating: Inner City (4.90 stars)
```

In [ ]:

task 3 : Price Range Distribution

Create a histogram or bar chart to visualize the distribution of price ranges among the restaurants.

In [49]:
```
# Count the number of restaurants in each price range
no_of_restro_in_range = df["Price range"].value_counts().sort_index()

print("Number of restaurants in each price range:\n", no_of_restro_in_range)
```

```
Number of restaurants in each price range:
 Price range
1    7434
2    7133
3    3761
4    1391
Name: count, dtype: int64
```

In [51]:
```
plt.bar(no_of_restro_in_range.index, no_of_restro_in_range.values, color="skyblue")

# Labels and title
plt.xlabel("Price Range")
plt.ylabel("Number of Restaurants")
plt.title("Distribution of Price Ranges")
plt.xticks(no_of_restro_in_range.index)  # Ensure correct x-axis labels

# Show the plot
plt.show()
```

Calculate the percentage of restaurants in each price range category.

In [54]:
```
# Total unique restaurants
total_restaurants = df["Restaurant ID"].nunique()

# Calculate percentage for each price range
price_percentage = (no_of_restro_in_range / total_restaurants) * 100

print("Percentage of restaurants in each price range:\n", price_percentage)
```

```
Percentage of restaurants in each price range:
 Price range
1    77.834782
2    74.683279
3    39.378076
4    14.563920
Name: count, dtype: float64
```

Task 4: Online Delivery

Determine the percentage of restaurants that offer online delivery.

In [58]:
```
total_restaurants = df["Restaurant ID"].nunique()
restro_online_delivery = df[df["Has Online delivery"] == "Yes"]["Restaurant ID"].nu
online_delivery_percentage = (restro_online_delivery / total_restaurants) * 100
print('percentage of restaurants that offer online delivery',online_delivery_percen
```

```
percentage of restaurants that offer online delivery 25.662234321013504
```

Compare the average ratings of restaurants with and without online delivery.

In [61]:
```
# Calculate average rating for restaurants with online delivery
avg_rating_with_delivery = df[df["Has Online delivery"] == "Yes"]["Aggregate rating

# Calculate average rating for restaurants without online delivery
avg_rating_without_delivery = df[df["Has Online delivery"] == "No"]["Aggregate rati

print(f"Average rating of restaurants with online delivery: {avg_rating_with_delive
print(f"Average rating of restaurants without online delivery: {avg_rating_without_
```

```
Average rating of restaurants with online delivery: 3.31
Average rating of restaurants without online delivery: 2.68
```

In [ ]:

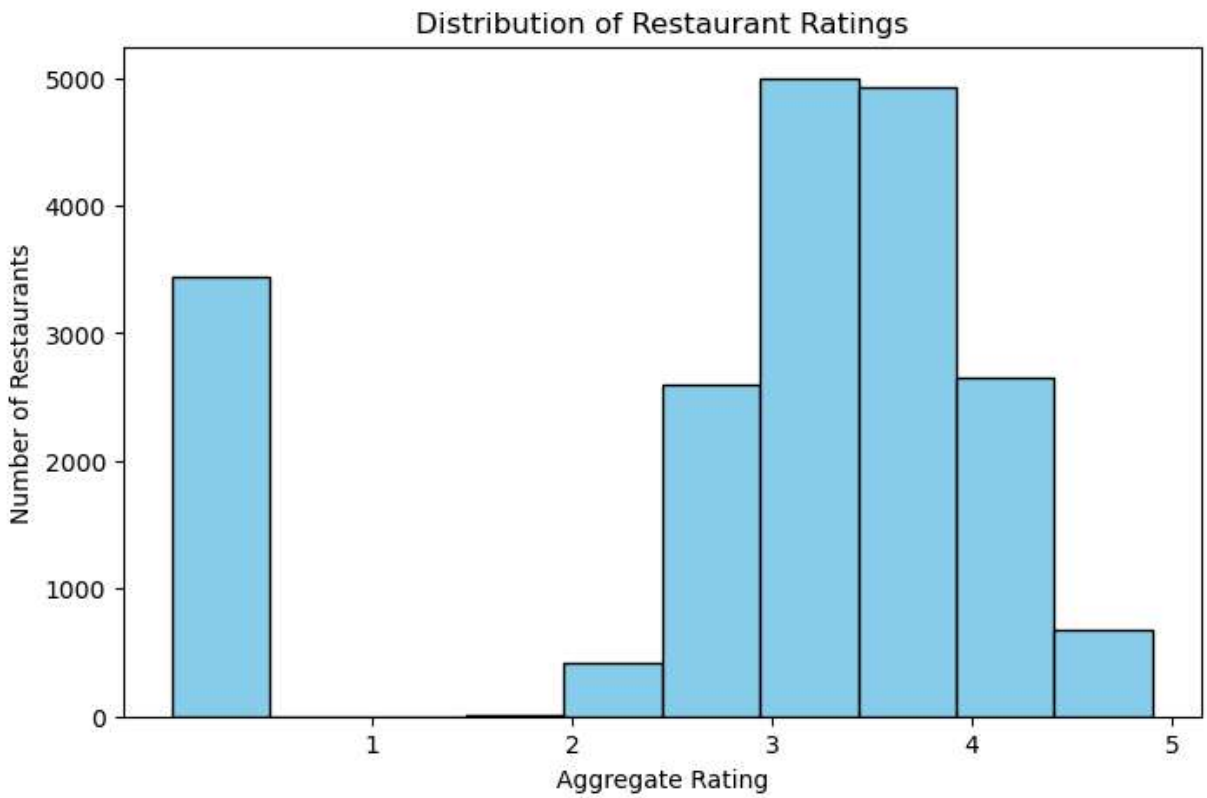
level 2

task 1 :Restaurant Ratings

Analyze the distribution of aggregate ratings and determine the most common rating range.

In [67]:
```
plt.figure(figsize=(8, 5))
plt.hist(df["Aggregate rating"].dropna(), bins=10, color="skyblue", edgecolor="blac

# Labels and title
plt.xlabel("Aggregate Rating")
plt.ylabel("Number of Restaurants")
plt.title("Distribution of Restaurant Ratings")
plt.xticks(range(1, 6))  # Assuming ratings range from 1 to 5
```

```
# Show plot
plt.show()
```



In [69]:
```
# Define rating bins
rating_bins = [0, 1, 2, 3, 4, 5]
rating_labels = ["0-1", "1-2", "2-3", "3-4", "4-5"]

# Categorize aggregate ratings into bins
df["Rating Range"] = pd.cut(df["Aggregate rating"], bins=rating_bins, labels=rating

# Count occurrences of each rating range
rating_counts = df["Rating Range"].value_counts()

# Find the most common rating range
most_common_range = rating_counts.idxmax()

print("Most common rating range:", most_common_range)
```

```
Most common rating range: 3-4
```

Calculate the Average Number of Votes

In [72]:
```
# Calculate the average votes per restaurant
average_votes = df["Votes"].mean()

print(f"Average number of votes per restaurant: {average_votes:.2f}")
```

```
Average number of votes per restaurant: 198.67
```

Task 2: Cuisine Combination

Identify the most common combinations of cuisines in the dataset.

In [76]:
```
from collections import Counter

# Ensure 'Cuisines' is treated as string and fill NaN values
df["Cuisines"] = df["Cuisines"].fillna("").astype(str)

# Group cuisines by restaurant ID (assuming there's a unique ID column like 'Restau
df_cuisine_grouped = df.groupby("Restaurant ID")["Cuisines"].apply(lambda x: ", ".j

# Count occurrences of each cuisine combination
cuisine_counts = Counter(df_cuisine_grouped["Cuisines"])

# Get the top 5 most common cuisine combinations
top_cuisines = cuisine_counts.most_common(5)

# Display results
print("Most common cuisine combinations:")
for combo, count in top_cuisines:
    print(f"{combo}: {count} restaurants")
```

```
Most common cuisine combinations:
North Indian: 936 restaurants
Chinese, North Indian: 616 restaurants
Mughlai, North Indian: 395 restaurants
Chinese: 354 restaurants
Fast Food: 354 restaurants
```

Determine if certain cuisine combinations tend to have higher ratings.

In [79]:
```
# Ensure 'Cuisines' column exists and fill missing values
df["Cuisines"] = df["Cuisines"].fillna("").astype(str)

# Group cuisines per restaurant by joining them into a single string
df_cuisine_grouped = df.groupby("Restaurant ID")["Cuisines"].apply(lambda x: ", ".j
```

In [81]:
```
# Ensure "Aggregate rating" is numeric
df["Aggregate rating"] = pd.to_numeric(df["Aggregate rating"], errors="coerce")

# Drop rows where rating is NaN
df = df.dropna(subset=["Aggregate rating"])

# Merge grouped cuisines back with the main dataset
df_merged = df.merge(df_cuisine_grouped, on="Restaurant ID", how="left")
```

```
# Check if the merge was successful
print(df_merged.head())
```

```
   Restaurant ID         Restaurant Name  Country Code              City  \
0        6317637        Le Petit Souffle           162       Makati City
1        6317637        Le Petit Souffle           162       Makati City
2        6317637        Le Petit Souffle           162       Makati City
3        6304287        Izakaya Kikufuji           162       Makati City
4        6300002  Heat - Edsa Shangri-La           162  Mandaluyong City

                                            Address  \
0  Third Floor, Century City Mall, Kalayaan Avenu...
1  Third Floor, Century City Mall, Kalayaan Avenu...
2  Third Floor, Century City Mall, Kalayaan Avenu...
3  Little Tokyo, 2277 Chino Roces Avenue, Legaspi...
4  Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...

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

                                  Locality Verbose   Longitude   Latitude  \
0  Century City Mall, Poblacion, Makati City, Mak...  121.027535  14.565443
1  Century City Mall, Poblacion, Makati City, Mak...  121.027535  14.565443
2  Century City Mall, Poblacion, Makati City, Mak...  121.027535  14.565443
3  Little Tokyo, Legaspi Village, Makati City, Ma...  121.014101  14.553708
4  Edsa Shangri-La, Ortigas, Mandaluyong City, Ma...  121.056831  14.581404

  Cuisines_x  ...  Has Online delivery Is delivering now Switch to order menu  \
0     French  ...                   No                No                   No
1   Japanese  ...                   No                No                   No
2   Desserts  ...                   No                No                   No
3   Japanese  ...                   No                No                   No
4    Seafood  ...                   No                No                   No

  Price range Aggregate rating Rating color  Rating text  Votes Rating Range  \
0           3              4.8   Dark Green    Excellent    314          4-5
1           3              4.8   Dark Green    Excellent    314          4-5
2           3              4.8   Dark Green    Excellent    314          4-5
3           3              4.5   Dark Green    Excellent    591          4-5
4           4              4.4        Green    Very Good    270          4-5

                       Cuisines_y
0        Desserts, French, Japanese
1        Desserts, French, Japanese
2        Desserts, French, Japanese
3                         Japanese
4  Asian, Filipino, Indian, Seafood

[5 rows x 23 columns]
```

In [85]:
```
print(df_merged.columns)
```

```
Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Address',
       'Locality', 'Locality Verbose', 'Longitude', 'Latitude', 'Cuisines_x',
       'Average Cost for two', 'Currency', 'Has Table booking',
       'Has Online delivery', 'Is delivering now', 'Switch to order menu',
       'Price range', 'Aggregate rating', 'Rating color', 'Rating text',
       'Votes', 'Rating Range', 'Cuisines_y'],
      dtype='object')
```

In [87]:
```
print(df_merged.head())  # Check if "Cuisines" exists
```

```
   Restaurant ID         Restaurant Name  Country Code              City  \
0        6317637        Le Petit Souffle           162       Makati City
1        6317637        Le Petit Souffle           162       Makati City
2        6317637        Le Petit Souffle           162       Makati City
3        6304287        Izakaya Kikufuji           162       Makati City
4        6300002  Heat - Edsa Shangri-La           162  Mandaluyong City

                                            Address  \
0  Third Floor, Century City Mall, Kalayaan Avenu...
1  Third Floor, Century City Mall, Kalayaan Avenu...
2  Third Floor, Century City Mall, Kalayaan Avenu...
3  Little Tokyo, 2277 Chino Roces Avenue, Legaspi...
4  Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...

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

                                  Locality Verbose   Longitude   Latitude  \
0  Century City Mall, Poblacion, Makati City, Mak...  121.027535  14.565443
1  Century City Mall, Poblacion, Makati City, Mak...  121.027535  14.565443
2  Century City Mall, Poblacion, Makati City, Mak...  121.027535  14.565443
3  Little Tokyo, Legaspi Village, Makati City, Ma...  121.014101  14.553708
4  Edsa Shangri-La, Ortigas, Mandaluyong City, Ma...  121.056831  14.581404

  Cuisines_x  ...  Has Online delivery Is delivering now Switch to order menu  \
0     French  ...                   No                No                   No
1   Japanese  ...                   No                No                   No
2   Desserts  ...                   No                No                   No
3   Japanese  ...                   No                No                   No
4    Seafood  ...                   No                No                   No

  Price range Aggregate rating Rating color  Rating text  Votes Rating Range  \
0           3              4.8   Dark Green    Excellent    314          4-5
1           3              4.8   Dark Green    Excellent    314          4-5
2           3              4.8   Dark Green    Excellent    314          4-5
3           3              4.5   Dark Green    Excellent    591          4-5
4           4              4.4        Green    Very Good    270          4-5

                        Cuisines_y
0        Desserts, French, Japanese
1        Desserts, French, Japanese
2        Desserts, French, Japanese
3                          Japanese
4  Asian, Filipino, Indian, Seafood

[5 rows x 23 columns]
```

In [95]:

```
# Group by cuisine combination and calculate average rating
cuisine_ratings = df_merged.groupby("Cuisines_x")["Aggregate rating"].mean().sort_v

# Display the top-rated cuisine combinations
```

```
print("Top-Rated Cuisine Combinations:")
print(cuisine_ratings.head(10))
```

```
Top-Rated Cuisine Combinations:
Cuisines_x
Sunda         4.900000
B_rek         4.700000
Taiwanese     4.650000
Ramen         4.500000
Dim Sum       4.466667
Hawaiian      4.412500
D_ner         4.400000
Bubble Tea    4.400000
Curry         4.400000
Kebab         4.380000
Name: Aggregate rating, dtype: float64
```

task 3 :Geographic Analysis

Plot the locations of restaurants on a map using longitude and latitude coordinates.

In [140…

```
df["Latitude"] = pd.to_numeric(df["Latitude"], errors="coerce")
df["Longitude"] = pd.to_numeric(df["Longitude"], errors="coerce")
df = df.dropna(subset=["Latitude", "Longitude"])  # Remove invalid rows
```

In [141…

```
from IPython.display import display
display(restaurant_map)
```

Make this Notebook Trusted to load map: File -> Trust Notebook

In [142…

```
import folium

# Define the center of the map (mean latitude & longitude)
map_center = [df["Latitude"].mean(), df["Longitude"].mean()]

# Create a map
restaurant_map = folium.Map(location=map_center, zoom_start=12)

# Add restaurant markers
for _, row in df.iterrows():
    folium.Marker(
        location=[row["Latitude"], row["Longitude"]],
        popup=row["Restaurant Name"],  # Show restaurant name on click
        icon=folium.Icon(color="blue", icon="cutlery")
    ).add_to(restaurant_map)

# Display map
restaurant_map
```

Out[142…

Make this Notebook Trusted to load map: File -> Trust Notebook

In [143…

```
restaurant_map.save("restaurant_map.html")
```

Identify any patterns or clusters of restaurants in specific areas.

In [109…

```
from folium.plugins import MarkerCluster

# Create a map
cluster_map = folium.Map(location=map_center, zoom_start=12)

# Create a marker cluster
marker_cluster = MarkerCluster().add_to(cluster_map)
```

```
# Add restaurant markers to the cluster
for _, row in df.iterrows():
    folium.Marker(
        location=[row["Latitude"], row["Longitude"]],
        popup=row["Restaurant Name"]
    ).add_to(marker_cluster)

# Display map
cluster_map
```

Out[109…

Make this Notebook Trusted to load map: File -> Trust Notebook

task 4: Restaurant Chains

Identify if there are any restaurant chains present in the dataset.

In [119…

```
# Count the number of restaurants with the same name
chain_counts = df["Restaurant Name"].value_counts()

# Filter out restaurant chains (those appearing more than once)
restaurant_chains = chain_counts[chain_counts > 1]

# Display top 10 most common restaurant chains
print("Top 10 Restaurant Chains:")
print(restaurant_chains.head(10))
```

```
Top 10 Restaurant Chains:
Restaurant Name
Subway              252
Domino's Pizza      157
Green Chick Chop    150
Bikanervala         103
Haldiram's          102
McDonald's           93
Cafe Coffee Day      83
Pizza Hut            76
Barbeque Nation      67
Dunkin' Donuts       66
Name: count, dtype: int64
```

Analyze Ratings of Restaurant Chains

In [122…

```
# Filter original dataset to include only restaurant chains
df_chains = df[df["Restaurant Name"].isin(restaurant_chains.index)]

# Calculate average rating for each chain
chain_ratings = df_chains.groupby("Restaurant Name")["Aggregate rating"].mean().sor

# Display top-rated restaurant chains
print("Top-Rated Restaurant Chains:")
print(chain_ratings.head(10))
```

```
Top-Rated Restaurant Chains:
Restaurant Name
Talaga Sampireun                        4.9
Oakwood Cafe                            4.9
Pom Pom's Teahouse and Sandwicheria     4.9
Grandson of Tunday Kababi               4.9
Yellow Dog Eats                         4.9
Ingleside Village Pizza                 4.9
Cube - Tasting Kitchen                  4.9
Ooma                                    4.9
Sheroes Hangout                         4.9
Bao                                     4.9
Name: Aggregate rating, dtype: float64
```

Analyze Popularity of Restaurant Chains

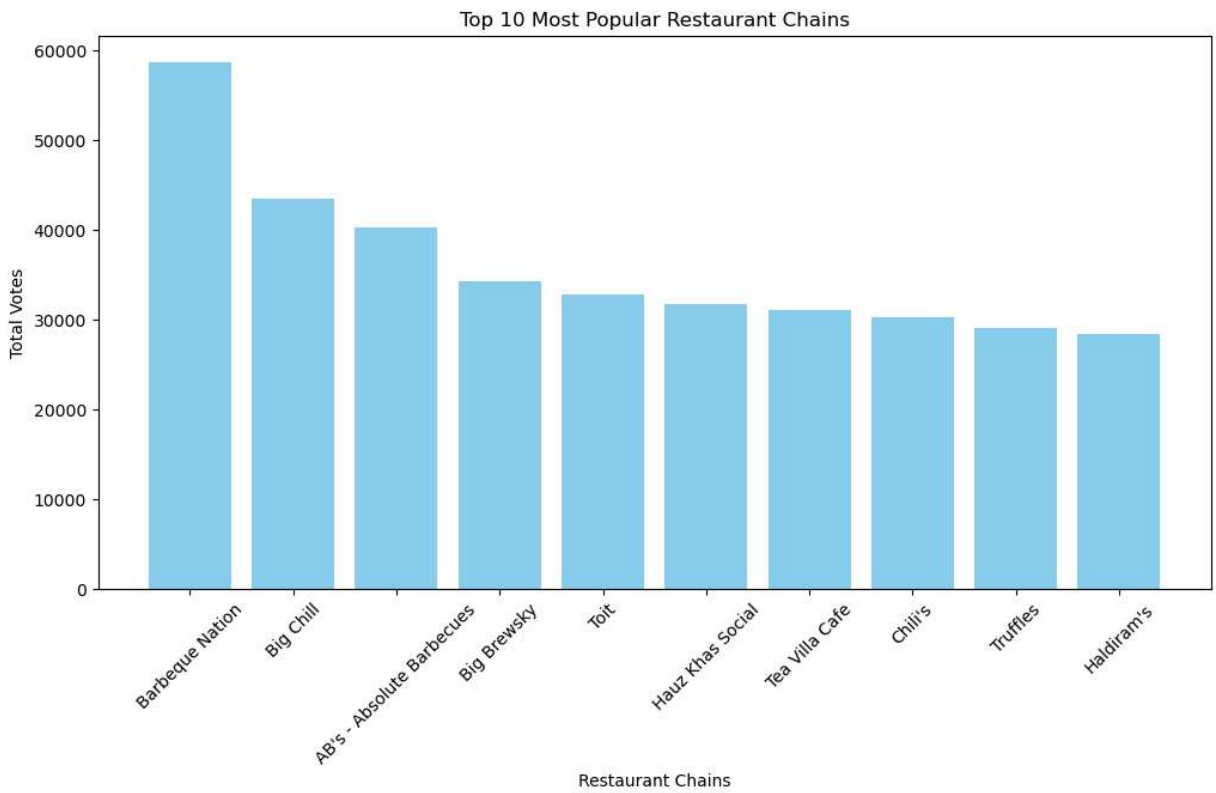
In [125…

```
                              # Calculate total votes for each restaurant chain
chain_votes = df_chains.groupby("Restaurant Name")["Votes"].sum().sort_values(ascen

# Display top 10 most popular restaurant chains
print("Most Popular Restaurant Chains (Based on Votes):")
print(chain_votes.head(10))
```

```
Most Popular Restaurant Chains (Based on Votes):
Restaurant Name
Barbeque Nation              58631
Big Chill                    43412
AB's - Absolute Barbecues    40200
Big Brewsky                  34230
Toit                         32802
Hauz Khas Social             31724
Tea Villa Cafe               31002
Chili's                      30215
Truffles                     29016
Haldiram's                   28445
Name: Votes, dtype: int64
```

In [127…

```
top_chains = chain_votes.head(10)

# Plot bar chart
plt.figure(figsize=(12, 6))
plt.bar(top_chains.index, top_chains.values, color="skyblue")
plt.xlabel("Restaurant Chains")
plt.ylabel("Total Votes")
plt.title("Top 10 Most Popular Restaurant Chains")
plt.xticks(rotation=45)
plt.show()
```



In [ ]: