In [43]:
```python
#import necessary library
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

In [44]:
```python
#load and read dataset
df=pd.read_csv('Dataset.csv')
df
```

Out[44]:

	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	.
...	...	...	...	...	...	...	...	...	...	...	.
9546	5915730	Namlı Gurme	208	��stanbul	Kemanke�� Karamustafa Pa��a Mahallesi, Rıhtım ...	Karak�_y	Karak�_y, ��stanbul	28.977392	41.022793	Turkish	.
9547	5908749	Ceviz A��acı	208	��stanbul	Ko��uyolu Mahallesi, Muhittin ��st�_nda�� Cadd...	Ko��uyolu	Ko��uyolu, ��stanbul	29.041297	41.009847	World Cuisine, Patisserie, Cafe	.
9548	5915807	Huqqa	208	��stanbul	Kuru�_e��me Mahallesi, Muallim Naci Caddesi, N...	Kuru�_e��me	Kuru�_e��me, ��stanbul	29.034640	41.055817	Italian, World Cuisine	.
9549	5916112	A���k Kahve	208	��stanbul	Kuru�_e��me Mahallesi, Muallim Naci Caddesi, N...	Kuru�_e��me	Kuru�_e��me, ��stanbul	29.036019	41.057979	Restaurant Cafe	.
9550	5927402	Walter's Coffee Roastery	208	��stanbul	Cafea��a Mahallesi, Bademaltı Sokak, No 21/B, ...	Moda	Moda, ��stanbul	29.026016	40.984776	Cafe	.

9551 rows × 21 columns

In [45]:
```python
#Check no of rows and column
print("Number of rows =",df.shape[0])
print("Number of column =",df.shape[1])
```

```
Number of rows = 9551
Number of column = 21
```

In [46]:
```python
#Checking infomation of all data
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 [47]:
```
#Check null value present in our datasets
df.isnull().sum()
```

Out[47]:
```
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 [48]:
```
#Drop null value in our datasets
df.dropna(inplace=True)
```

In [49]:
```
#Again check null value present in our datasets
pd.isnull(df).sum()
```

Out[49]:
```
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 [50]:
```
#descriptive stat
df.describe()
```

Out[50]:

	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range	Aggregate rating	Votes
count	9.542000e+03	9542.000000	9542.000000	9542.000000	9542.000000	9542.000000	9542.000000	9542.000000
mean	9.043301e+06	18.179208	64.274997	25.848532	1200.326137	1.804968	2.665238	156.772060
std	8.791967e+06	56.451600	41.197602	11.010094	16128.743876	0.905563	1.516588	430.203324
min	5.300000e+01	1.000000	-157.948486	-41.330428	0.000000	1.000000	0.000000	0.000000
25%	3.019312e+05	1.000000	77.081565	28.478658	250.000000	1.000000	2.500000	5.000000
50%	6.002726e+06	1.000000	77.192031	28.570444	400.000000	2.000000	3.200000	31.000000
75%	1.835260e+07	1.000000	77.282043	28.642711	700.000000	2.000000	3.700000	130.000000
max	1.850065e+07	216.000000	174.832089	55.976980	800000.000000	4.000000	4.900000	10934.000000

In [51]:

```
#Number of unique value in our datasets
df.nunique()
```

Out[51]:

```
Restaurant ID           9542
Restaurant Name         7437
Country Code              15
City                     140
Address                 8910
Locality                1206
Locality Verbose        1263
Longitude               8111
Latitude                8668
Cuisines                1825
Average Cost for two     140
Currency                  12
Has Table booking          2
Has Online delivery        2
Is delivering now          2
Switch to order menu       1
Price range                4
Aggregate rating          33
Rating color               6
Rating text                6
Votes                   1012
dtype: int64
```

In [52]:

```
#check duplicate value of sum in our datasets
print(df.duplicated().sum())
```

```
0
```

In [53]:

```
df.columns
```

Out[53]:

```
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 Categorical & Numerical feature present in our datasets.

In [54]:

```
#These are our categorical feature
[feature for feature in df.columns if df[feature].dtype=='O']
```

Out[54]:

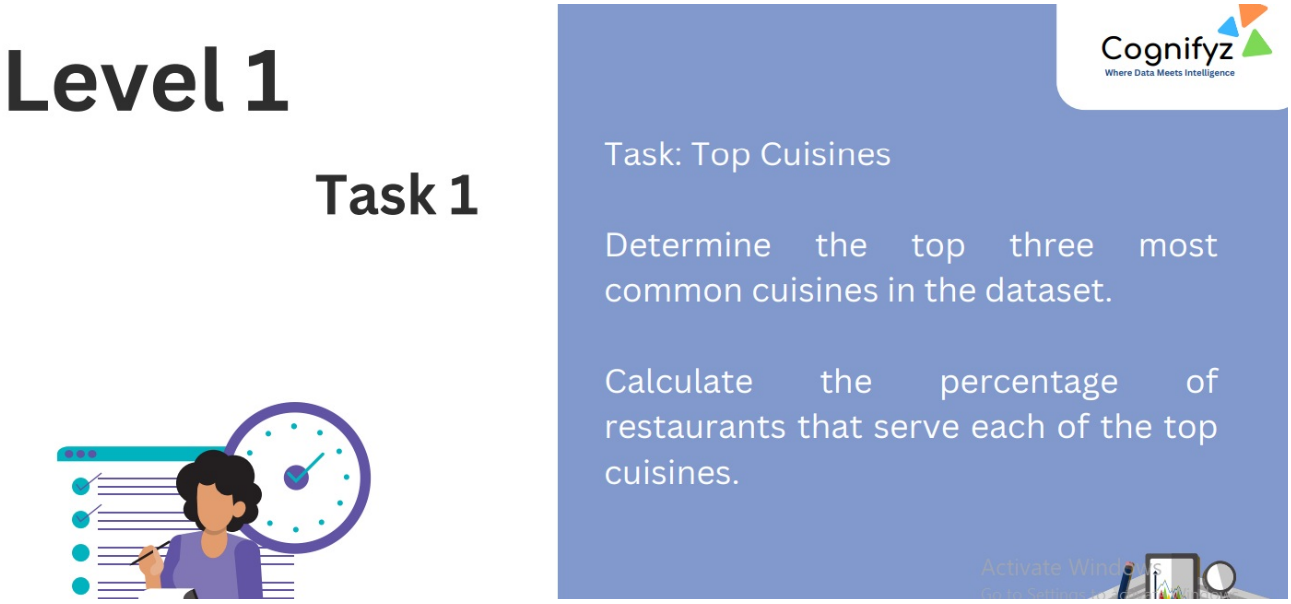
```
['Restaurant Name',
 'City',
 'Address',
 'Locality',
 'Locality Verbose',
 'Cuisines',
 'Currency',
 'Has Table booking',
 'Has Online delivery',
 'Is delivering now',
 'Switch to order menu',
 'Rating color',
 'Rating text']
```

In [55]:

```
#These are our Numerical feature
[feature for feature in df.columns if df[feature].dtype!='O']
```

Out[55]:

```
['Restaurant ID',
 'Country Code',
 'Longitude',
 'Latitude',
 'Average Cost for two',
 'Price range',
 'Aggregate rating',
 'Votes']
```



Task: Top Cuisines

1. Determine the top three most common cuisines in the dataset.
2. Calculate the percentage of restaurants that serve each of the top cuisines.

Task

1. Determine the top three most common cuisines in the dataset

In [56]:
```
df_cuisines=df['Cuisines'].str.split(', ').explode().value_counts()
```

In [57]:
```
top_cuisin=df_cuisines.head(3)
top_cuisin
```

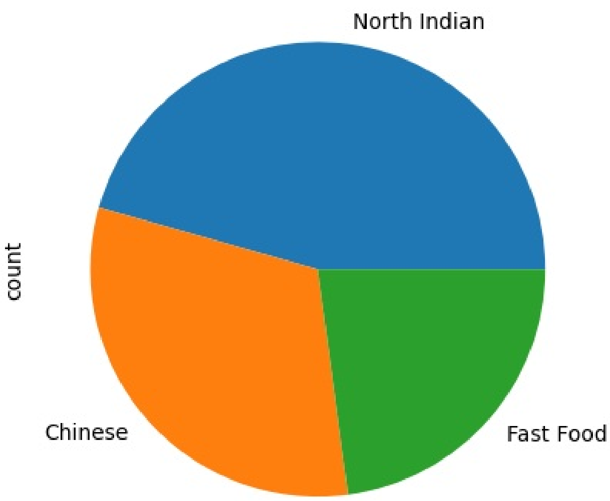
Out[57]:
```
Cuisines
North Indian    3960
Chinese         2735
Fast Food       1986
Name: count, dtype: int64
```

Insights:-

Dominant Cuisines: North Indian cuisine is the most popular in the dataset, followed by Chinese and Fast Food.

In [58]:
```
#visualize by using pie chart
top_cuisin.plot(kind='pie')
```

Out[58]:
```
<Axes: ylabel='count'>
```



Task

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

In [59]:
```
total_restaurant = len(df)
total_percentage=(top_cuisin/total_restaurant)*100
total_percentage
```

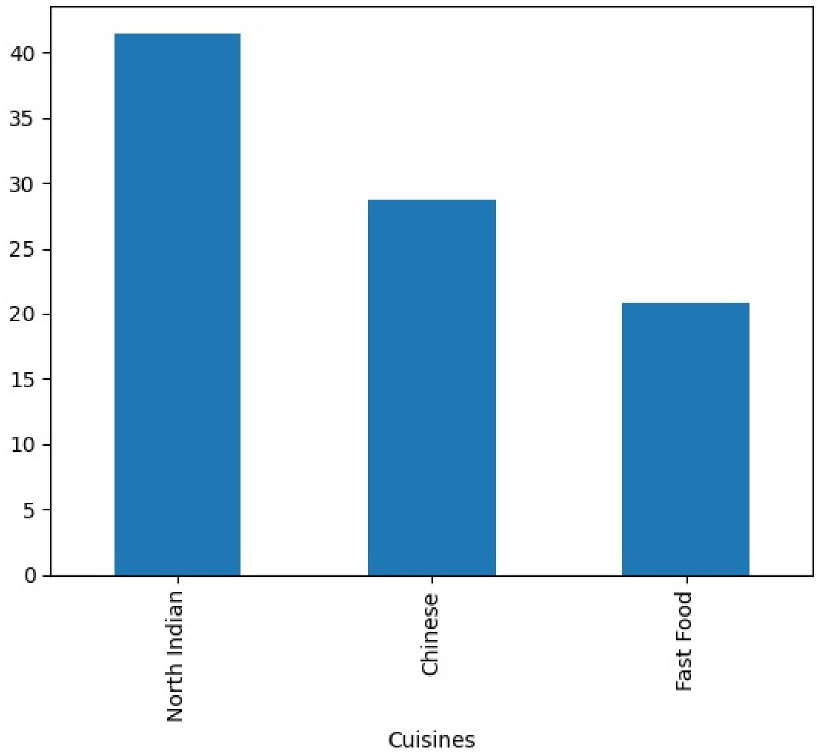
Out[59]:
```
Cuisines
North Indian    41.500734
Chinese         28.662754
Fast Food       20.813247
Name: count, dtype: float64
```

Insights:-

1. Dominance of North Indian Cuisine is a significant portion (41.46%) of restaurants offer North Indian cuisine, indicating its popularity.
2. Chinese and Fast Food cuisines are also widely available, but with lower percentages compared to North Indian.

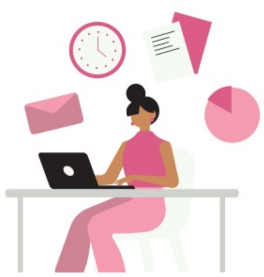
In [60]:
```
total_percentage.plot(kind='bar')
```

Out[60]:
```
<Axes: xlabel='Cuisines'>
```



Level 1

Task 2



Task: City Analysis

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

Calculate the average rating for restaurants in each city.

Determine the city with the highest average rating.

Task

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

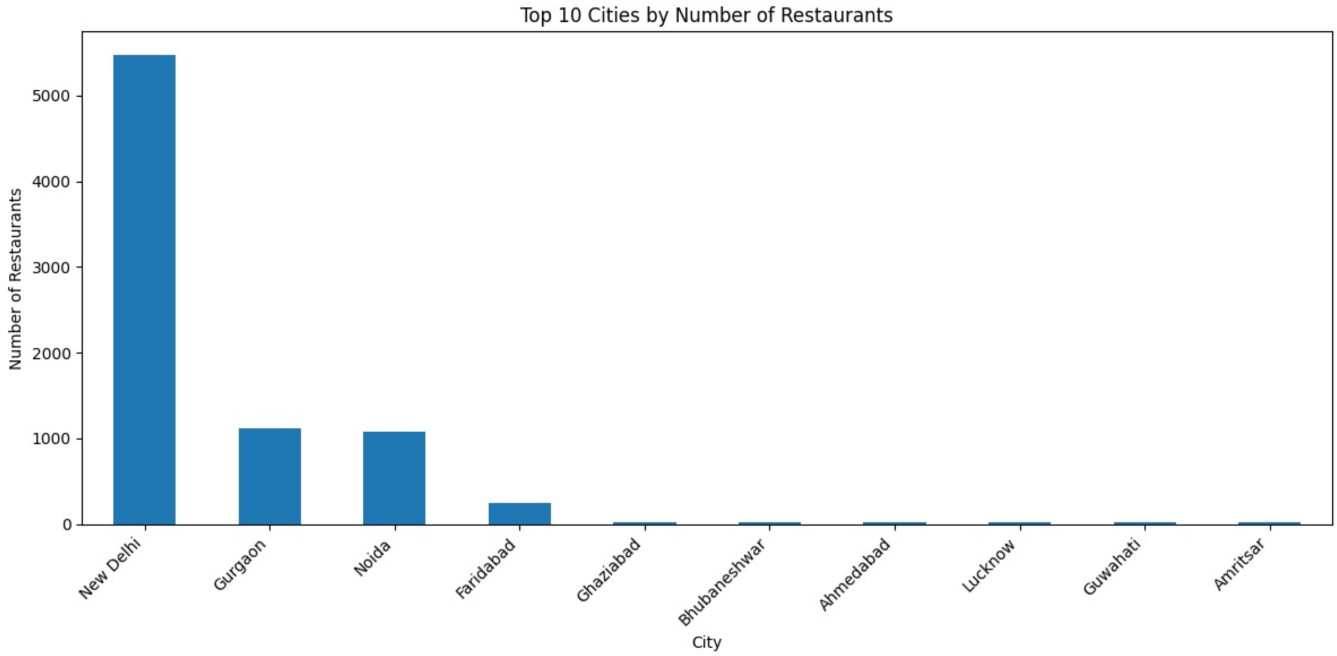
In [61]:
```
#count the number of restaurant percity
city_counts = df['City'].value_counts()
city_counts
```

Out[61]:
```
City
New Delhi    5473
Gurgaon      1118
Noida        1080
Faridabad     251
Ghaziabad      25
             ... 
Inverloch       1
Mohali          1
Panchkula       1
Bandung         1
Randburg        1
Name: count, Length: 140, dtype: int64
```

In [62]:
```
top_city = city_counts.idxmax()
top_city_count = city_counts.max()
print(f"1. The city with the highest number of restaurants is {top_city} with {top_city_count} restaurants.")
```

```
1. The city with the highest number of restaurants is New Delhi with 5473 restaurants.
```

In [63]:
```
# Visualize the top 10 cities by number of restaurants
plt.figure(figsize=(12, 6))
city_counts.head(10).plot(kind='bar')
plt.title('Top 10 Cities by Number of Restaurants')
plt.xlabel('City')
plt.ylabel('Number of Restaurants')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()
plt.show()
```



Insights:-

1. New Delhi has a significantly higher number of restaurants compared to other cities in the dataset.
2. There's a considerable difference between New Delhi and the other cities in terms of restaurant count.

Task

1. Calculate the average rating for restaurants in each city.

In [64]:
```
avg_city_ratting=df.groupby('City')['Aggregate rating'].mean().sort_values(ascending=False)
avg_city_ratting
```

Out[64]:

```
City
Inner City          4.900000
Quezon City         4.800000
Makati City         4.650000
Pasig City          4.633333
Mandaluyong City    4.625000
                      ...
New Delhi           2.438845
Montville           2.400000
Mc Millan           2.400000
Noida               2.036204
Faridabad           1.866932
Name: Aggregate rating, Length: 140, dtype: float64
```

In [65]:

```
print("Average ratings for restaurants in each city:")
print(avg_city_ratting.head())
```

```
Average ratings for restaurants in each city:
City
Inner City          4.900000
Quezon City         4.800000
Makati City         4.650000
Pasig City          4.633333
Mandaluyong City    4.625000
Name: Aggregate rating, dtype: float64
```

Insights:

1. City-wise Rating Variation:- There's a significant difference in average ratings across cities, indicating varying restaurant quality standards.
2. Top-rated Cities:- Cities like Inner City, Quezon City, and Makati City have exceptionally high average ratings, suggesting a concentration of high-quality restaurants.
3. Low-rated Cities:- Cities like Noida and Faridabad have relatively lower average ratings, which might indicate areas for improvement in restaurant quality or service.

Task

1. Determine the city with the highest average rating.

In [66]:

```
hight_city_avg_rating = avg_city_ratting.index[0]
hight_avg_rating = avg_city_ratting.iloc[0]
```

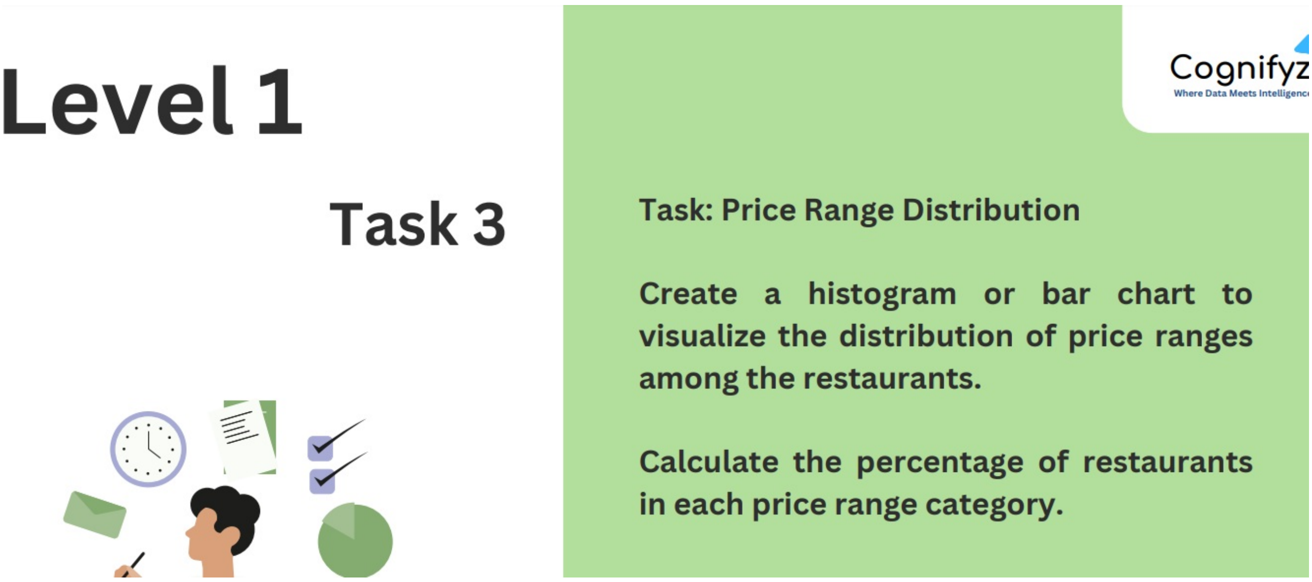
In [67]:

```
print(f"The city with the highest average rating is {hight_city_avg_rating} with an average rating of {hight_av
```

```
The city with the highest average rating is Inner City with an average rating of 4.90
```

Insights:-

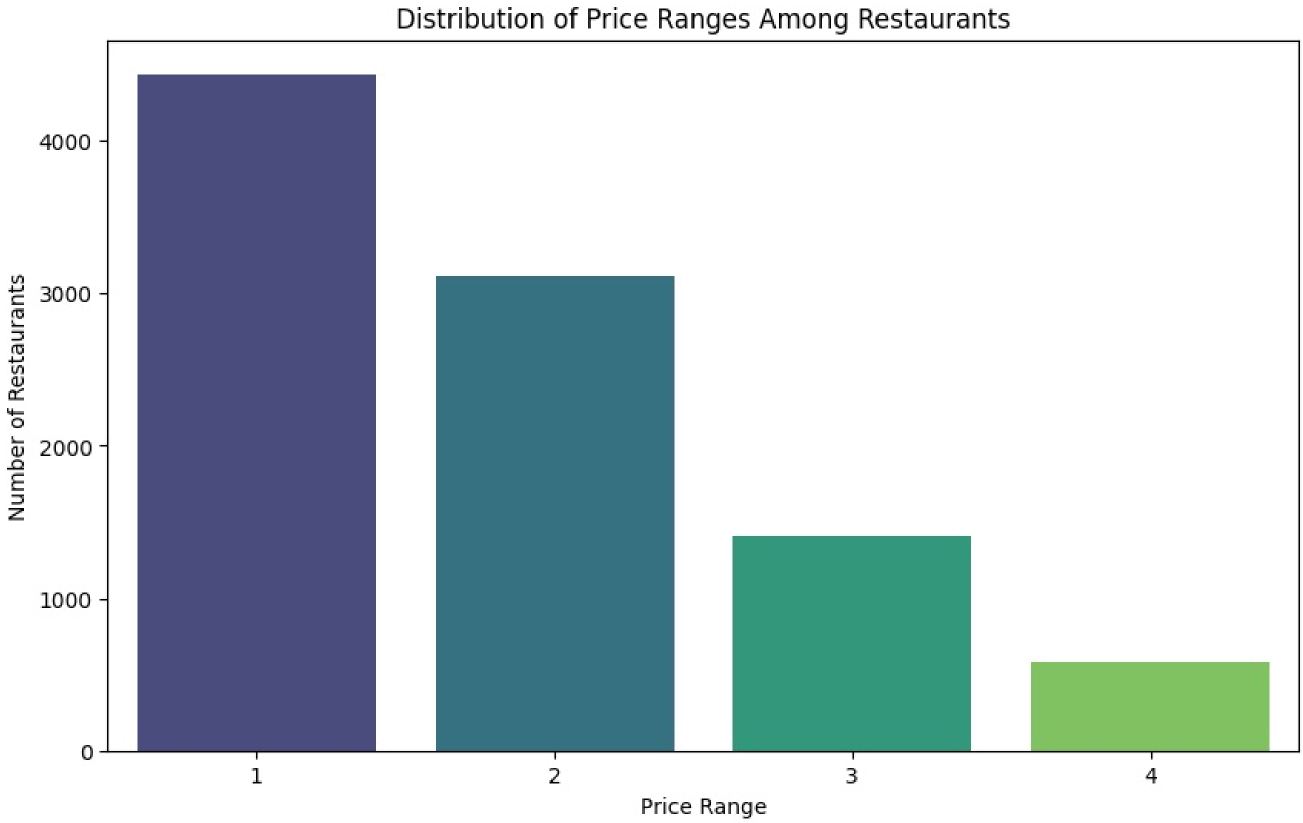
Inner City has the highest average restaurant rating among all cities in the dataset.



Task: Price Range Distribution

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

In [68]:
```
# Create a bar chart for price range distribution
plt.figure(figsize=(10, 6))
sns.countplot(x='Price range', data=df, palette='viridis')
plt.title('Distribution of Price Ranges Among Restaurants')
plt.xlabel('Price Range')
plt.ylabel('Number of Restaurants')
plt.show()
```



In [69]:
```
df['Price range'].value_counts()
```

Out[69]:
```
Price range
1    4438
2    3113
3    1405
4     586
Name: count, dtype: int64
```

Observations:

1. Price Range Distribution: Most restaurants fall in lower price ranges, with fewer in higher tiers.
2. Dominance of Lower Prices: Budget-friendly options dominate the market.

Task

1. Calculate the percentage of restaurants in each price range category

In [70]:
```
# Calculate the percentage of restaurants in each price range category
price_range_counts = df['Price range'].value_counts(normalize=True) * 100
```

In [71]:
```
# Convert to a DataFrame for better readability
price_range_percentages = price_range_counts.sort_index().reset_index()
price_range_percentages.columns = ['Price Range', 'Percentage']
price_range_percentages
```

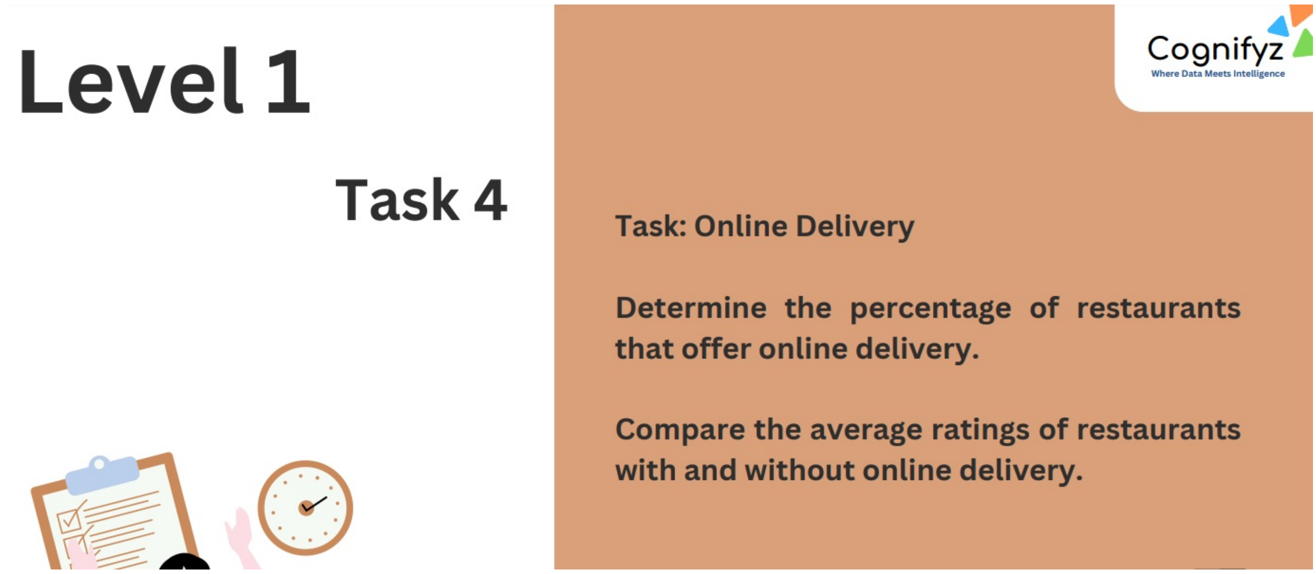
Out[71]:

	Price Range	Percentage
0	1	46.510166
1	2	32.624188
2	3	14.724376
3	4	6.141270

Insight:

Insight:

1. The majority of restaurants fall into price range 1, followed by price range 2.
2. As the price range increases, the percentage of restaurants decreases significantly.



Task

1. Determine the percentage of restaurants that offer online delivery.

In [72]:
```
total_count=df[['Restaurant ID','Restaurant Name','Has Online delivery']].groupby(['Restaurant ID','Restaurant
total_count
```

Out[72]:

	Restaurant ID	Restaurant Name	Has Online delivery	total
0	53	Amber	Yes	1
1	55	Berco's	Yes	1
2	60	Colonel's Kababz	No	1
3	64	Diva - The Italian Restaurant	Yes	1
4	65	Drums of Heaven	Yes	1
...	...	...	...	...
9537	18499493	Zombiez	No	1
9538	18500618	Veg. Darbar	No	1
9539	18500628	Grill & Cafe	No	1
9540	18500639	Chandni Chowk 2 China	No	1
9541	18500652	Mahek By Greenz	No	1

9542 rows × 4 columns

In [73]:
```
delivery_online_percentage=(df['Has Online delivery']=="Yes").mean()*100
print(f"Percentage of restaurants offering online delivery: {delivery_online_percentage:.2f}%")
```

```
Percentage of restaurants offering online delivery: 25.69%
```

Ovservation

Percentage of restaurants offering online delivery: 25.66%.This means that about a quarter of the restaurants in the dataset offer online delivery services

Task

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

In [74]:
```
#Compare rating
avg_rating_with_delivery = df[df['Has Online delivery']=="Yes"]['Aggregate rating'].mean()
avg_rating_without_delivery = df[df['Has Online delivery']=="No"]['Aggregate rating'].mean()
print("Average rating of restaurants with online delivery =",avg_rating_with_delivery)
print("Average rating of restaurants without online delivery =",avg_rating_without_delivery)
```

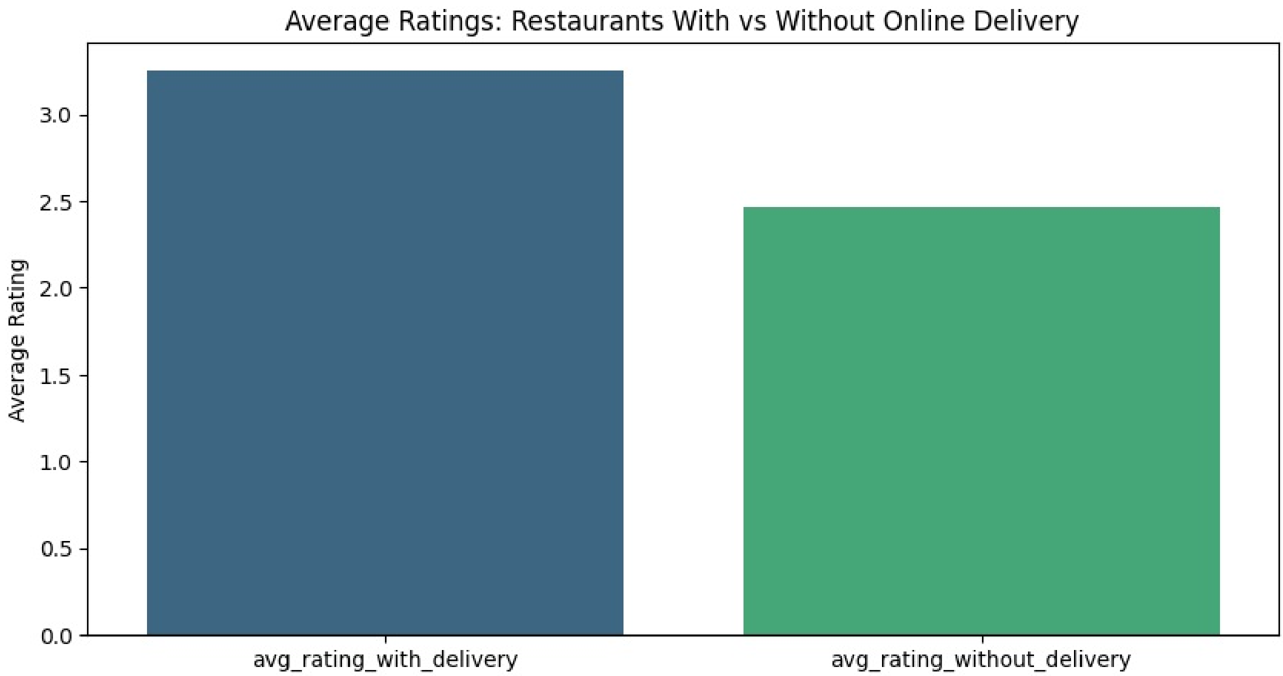
```
Average rating of restaurants with online delivery = 3.2488372093023257
Average rating of restaurants without online delivery = 2.4635171343957127
```

Ovservation

1. Average rating of restaurants with online delivery: 3.24
2. Average rating of restaurants without online delivery: 2.46

In [75]:

```
# Create a bar plot to visualize the comparison
plt.figure(figsize=(10,5))
sns.barplot(x=['avg_rating_with_delivery','avg_rating_without_delivery'],y=[avg_rating_with_delivery,avg_rating_
plt.title('Average Ratings: Restaurants With vs Without Online Delivery')
plt.ylabel('Average Rating')
plt.show()
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js