Zomato Restaurants Data

In [1]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib
%matplotlib inline
```

Importing and inspecting the dataset

In [3]:

```
df=pd.read_csv("zomato.csv",encoding='latin-1')
#encoding format
```

In [4]:

```
df.head(2)
```

Out[4]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Currency	Has Table booking	Has Online delivery	Is delivering now	Switc ord mer
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	...	Botswana Pula(P)	Yes	No	No	N
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	...	Botswana Pula(P)	Yes	No	No	N

2 rows × 21 columns

In [5]:

```
df.columns
```

Out[5]:

```
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')
```

In [6]:
```
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 [7]:
```
df.describe()
```

Out[7]:

	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range	Aggregate rating	Votes
count	9.551000e+03	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000
mean	9.051128e+06	18.365616	64.126574	25.854381	1199.210763	1.804837	2.666370	156.909748
std	8.791521e+06	56.750546	41.467058	11.007935	16121.183073	0.905609	1.516378	430.169145
min	5.300000e+01	1.000000	-157.948486	-41.330428	0.000000	1.000000	0.000000	0.000000
25%	3.019625e+05	1.000000	77.081343	28.478713	250.000000	1.000000	2.500000	5.000000
50%	6.004089e+06	1.000000	77.191964	28.570469	400.000000	2.000000	3.200000	31.000000
75%	1.835229e+07	1.000000	77.282006	28.642758	700.000000	2.000000	3.700000	131.000000
max	1.850065e+07	216.000000	174.832089	55.976980	800000.000000	4.000000	4.900000	10934.000000

In [8]:
```
df.describe(include='O')
```

Out[8]:

	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
count	9551	9551	9551	9551	9551	9542	9551	9551	9551	9551	9551	9551	9551
unique	7446	141	8918	1208	1265	1825	12	2	2	2	1	6	6
top	Cafe Coffee Day	New Delhi	Dilli Haat, INA, New Delhi	Connaught Place	Connaught Place, New Delhi	North Indian	Indian Rupees(Rs.)	No	No	No	No	Orange	Average
freq	83	5473	11	122	122	936	8652	8393	7100	9517	9551	3737	3737

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

Out[9]:
```
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
```

To get names of columns having null values

In [10]:
```
#get name of columns with null values
[features for features in df.columns if df[features].isnull().sum()>0]
```

Out[10]:
```
['Cuisines']
```

Merging another excel file containing country names with country codes in the dataset we have

In [5]:
```
df_c=pd.read_excel('Country-code.xlsx')
df_c.head()
```

Out[5]:

	Country Code	Country
0	1	India
1	14	Australia
2	30	Brazil
3	37	Canada
4	94	Indonesia

In [6]:
```
'Country Code' in df.columns
```

Out[6]:
```
True
```

In [7]:
```
final_df=pd.merge(df,df_c,on='Country Code',how='left')
```

In [14]:
```
final_df.head(2)
```

Out[14]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Has Table booking	Has Online delivery	Is delivering now	Switch to order menu	Price range
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	...	Yes	No	No	No	3
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	...	Yes	No	No	No	3

2 rows × 22 columns

In [15]:
```
final_df.columns
```

Out[15]:
```
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', 'Country'],
      dtype='object')
```

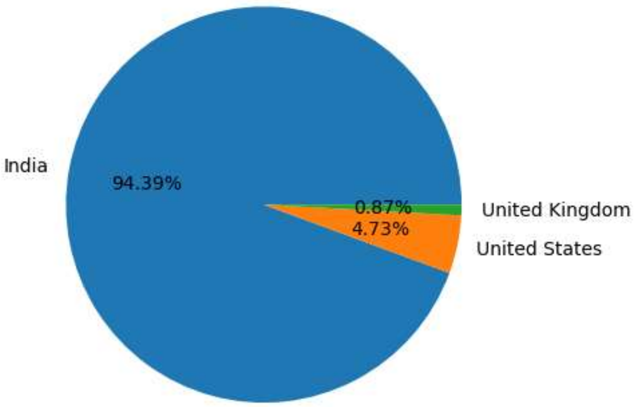
Plotting top 3 countries on basis of restaurants

In [16]:
```
country_names=final_df.Country.value_counts().index
```

In [17]:
```
country_val=final_df.Country.value_counts().values
```

In [21]:
```
#pie chart-top 3 countries that use zomato
plt.pie(country_val[:3],labels=country_names[:3],autopct='%1.2f%%')
```

Out[21]:
```
([<matplotlib.patches.Wedge at 0x1d6dfc0a210>,
  <matplotlib.patches.Wedge at 0x1d6dfc0b450>,
  <matplotlib.patches.Wedge at 0x1d6dfc1ca10>],
 [Text(-1.0829742700952103, 0.19278674827836725, 'India'),
  Text(1.077281715838356, -0.22240527134123297, 'United States'),
  Text(1.0995865153823035, -0.03015783794312073, 'United Kingdom')],
 [Text(-0.590713238233751, 0.10515640815183668, '94.39%'),
  Text(0.5876082086391032, -0.12131196618612707, '4.73%'),
  Text(0.5997744629358018, -0.01644972978715676, '0.87%')])
```



Observation

- Zomato's maximum transactions are in India followed by USA and UK

Analysing Restaurant Ratings

In [43]:
```
ratings=final_df.groupby(['Aggregate rating','Rating color','Rating text']).size().reset_index().rename(columns={0:'Rating Count'
ratings
```

Out[43]:

	Aggregate rating	Rating color	Rating text	Rating Count
0	0.0	White	Not rated	2148
1	1.8	Red	Poor	1
2	1.9	Red	Poor	2
3	2.0	Red	Poor	7
4	2.1	Red	Poor	15
5	2.2	Red	Poor	27
6	2.3	Red	Poor	47
7	2.4	Red	Poor	87
8	2.5	Orange	Average	110
9	2.6	Orange	Average	191
10	2.7	Orange	Average	250
11	2.8	Orange	Average	315
12	2.9	Orange	Average	381
13	3.0	Orange	Average	468
14	3.1	Orange	Average	519
15	3.2	Orange	Average	522
16	3.3	Orange	Average	483
17	3.4	Orange	Average	498
18	3.5	Yellow	Good	480
19	3.6	Yellow	Good	458
20	3.7	Yellow	Good	427
21	3.8	Yellow	Good	400
22	3.9	Yellow	Good	335
23	4.0	Green	Very Good	266
24	4.1	Green	Very Good	274
25	4.2	Green	Very Good	221
26	4.3	Green	Very Good	174
27	4.4	Green	Very Good	144
28	4.5	Dark Green	Excellent	95
29	4.6	Dark Green	Excellent	78
30	4.7	Dark Green	Excellent	42
31	4.8	Dark Green	Excellent	25
32	4.9	Dark Green	Excellent	61

Observation

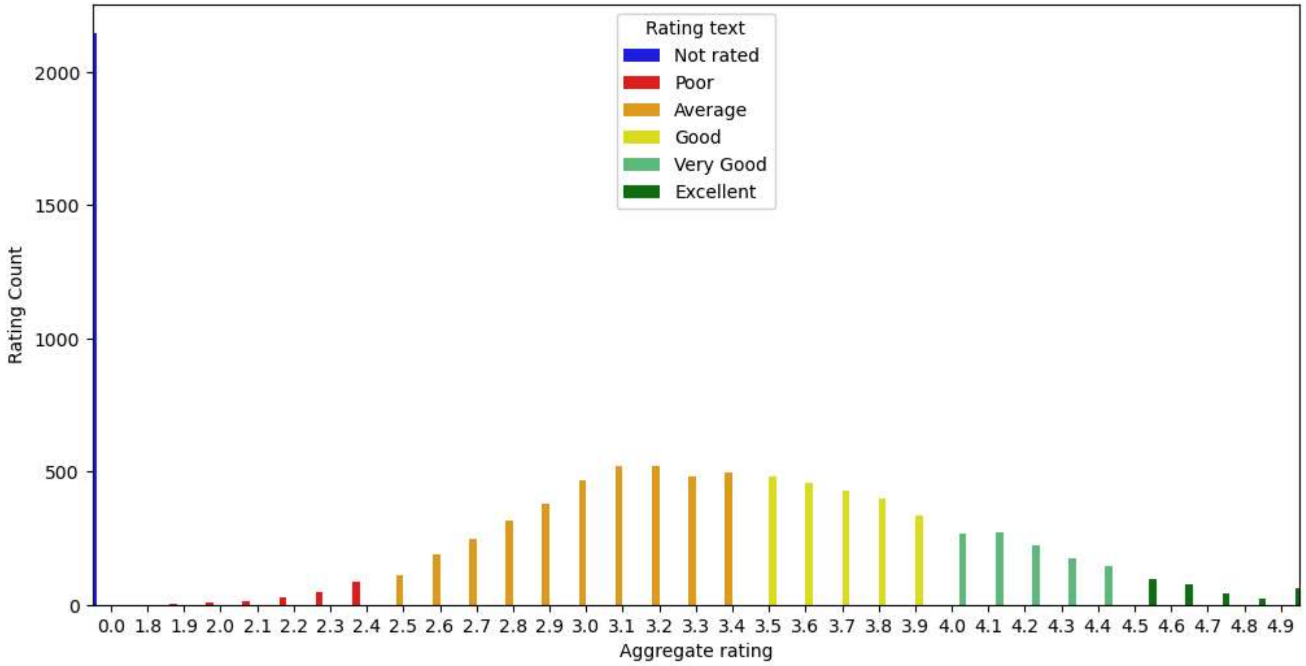
- 1.When rating between 4.5 to 4.9--->excellent
- 2.When rating between 4.0 to 4.4--->very good
- 3.When rating between 3.5 to 3.9--->good
- 4.When rating between 2.5 to 3.4--->average
- 5.When rating between 1.0 to 2.4--->poor

In [48]:

```
plt.figure(figsize=(12,6))
matplotlib.rcParams['figure.figsize']=(12,6)
sns.barplot(x='Aggregate rating',y="Rating Count",data=ratings,hue='Rating text',palette=['blue','red','orange','yellow','#50C878
```

Out[48]:

```
<Axes: xlabel='Aggregate rating', ylabel='Rating Count'>
```



Observation

- A very high number of restaurants are not rated
- Among rated restaurants, the average rating falls between 3.1 to 3.6

In [29]:

```
final_df[final_df['Aggregate rating']==0.0]['Country'].value_counts()
```

Out[29]:

```
India             2139
Brazil               5
United States        3
United Kingdom       1
Name: Country, dtype: int64
```

Observation

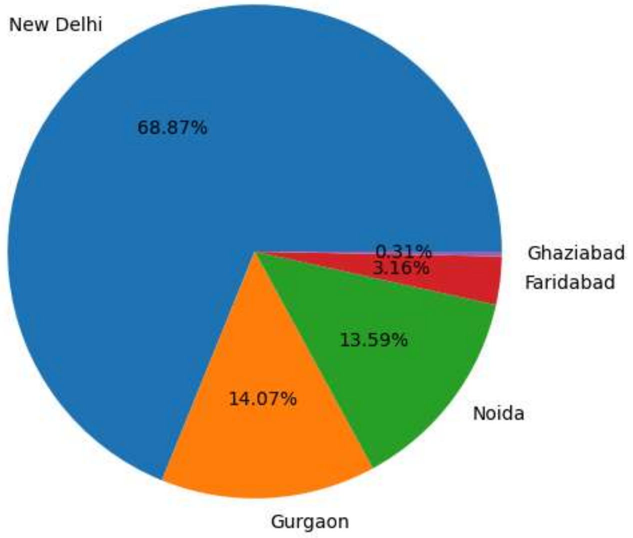
- Among 2148 non-rated restaurants, 2139 are in India.

In [69]:

```
non_r_labels=final_df[(final_df['Aggregate rating']==0.0) & (final_df['Country']=='India') ]['City'].value_counts().index
non_r_count=final_df[(final_df['Aggregate rating']==0.0) & (final_df['Country']=='India') ]['City'].value_counts().values
```

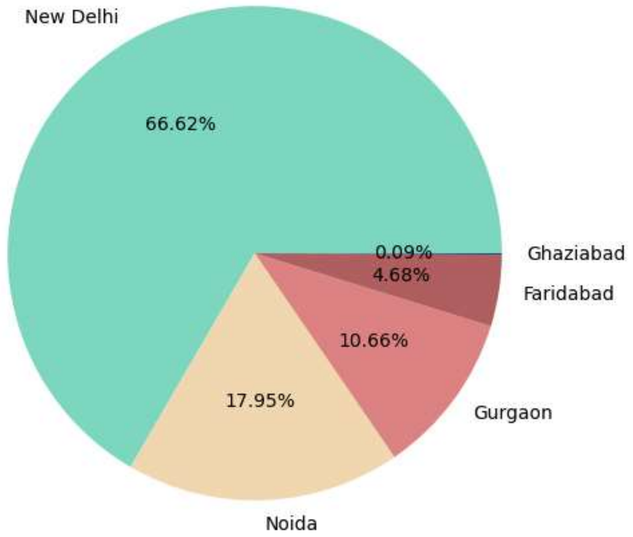
In [70]:

```
citylabels=final_df.City.value_counts().index
cityvalues=final_df.City.value_counts().values
plt.pie(x=cityvalues[:5],labels=citylabels[:5],autopct='%1.2f%%')
plt.show()
```



In [71]:

```
plt.pie(non_r_count[:5],labels=non_r_labels[:5],autopct='%1.2f%%',colors=['#7ED7C1','#F0DBAF','#DC8686','#B06161','#2C4E80'])
plt.show()
```



Observation

- Noida has lesser restaurants than Gurgaon, but more non-rated restaurants than gurgaon
- Zomato should focus on increasing the efficiency of rating system in Noida

Deeper analysis based on India,since the dataset mostly contains data from india

In [8]:

```
df_i=final_df[final_df['Country']=='India']
df_i.head(2)
```

Out[8]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Has Table booking	Has Online delivery	Is delivering now	Switch to order menu	Price range	Agg
624	3400025	Jahanpanah	1	Agra	E 23, Shopping Arcade, Sadar Bazaar, Agra Cant...	Agra Cantt	Agra Cantt, Agra	78.011544	27.161661	North Indian, Mughlai	...	No	No	No	No	3	
625	3400341	Rangrezz Restaurant	1	Agra	E-20, Shopping Arcade, Sadar Bazaar, Agra Cant...	Agra Cantt	Agra Cantt, Agra	0.000000	0.000000	North Indian, Mughlai	...	No	No	No	No	2	

2 rows × 22 columns

In [23]:

```
#Top Cities
df_i['City'].value_counts().head(5)
```

Out[23]:

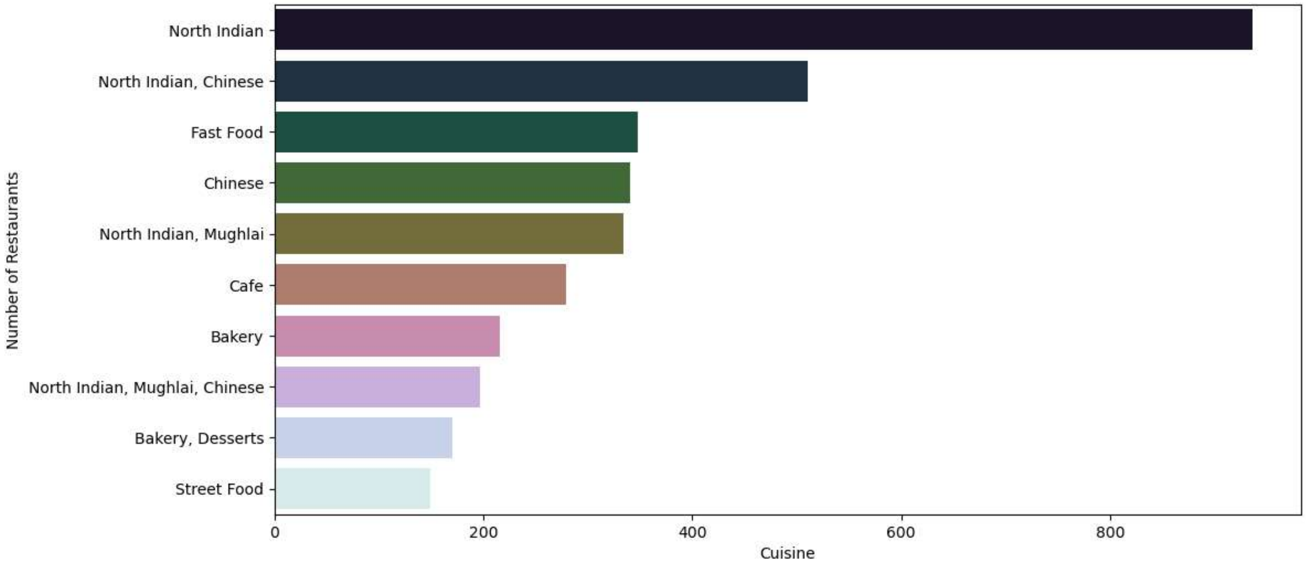
```
New Delhi     5473
Gurgaon       1118
Noida         1080
Faridabad      251
Ghaziabad       25
Name: City, dtype: int64
```

Observation

- New Delhi, Gurgaon and Noida constitutes the largest share of zomato's business

In [67]:

```
top_cuisines=df_i['Cuisines'].value_counts().reset_index().head(10)
sns.barplot(y='index',x='Cuisines',data=top_cuisines,palette='cubehelix')
plt.xlabel('Cuisine')
plt.ylabel('Number of Restaurants')
plt.show()
```

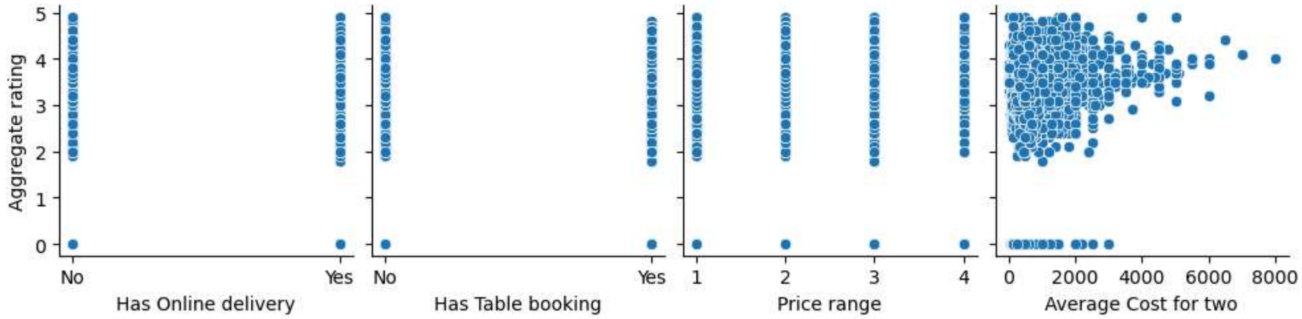


Observation

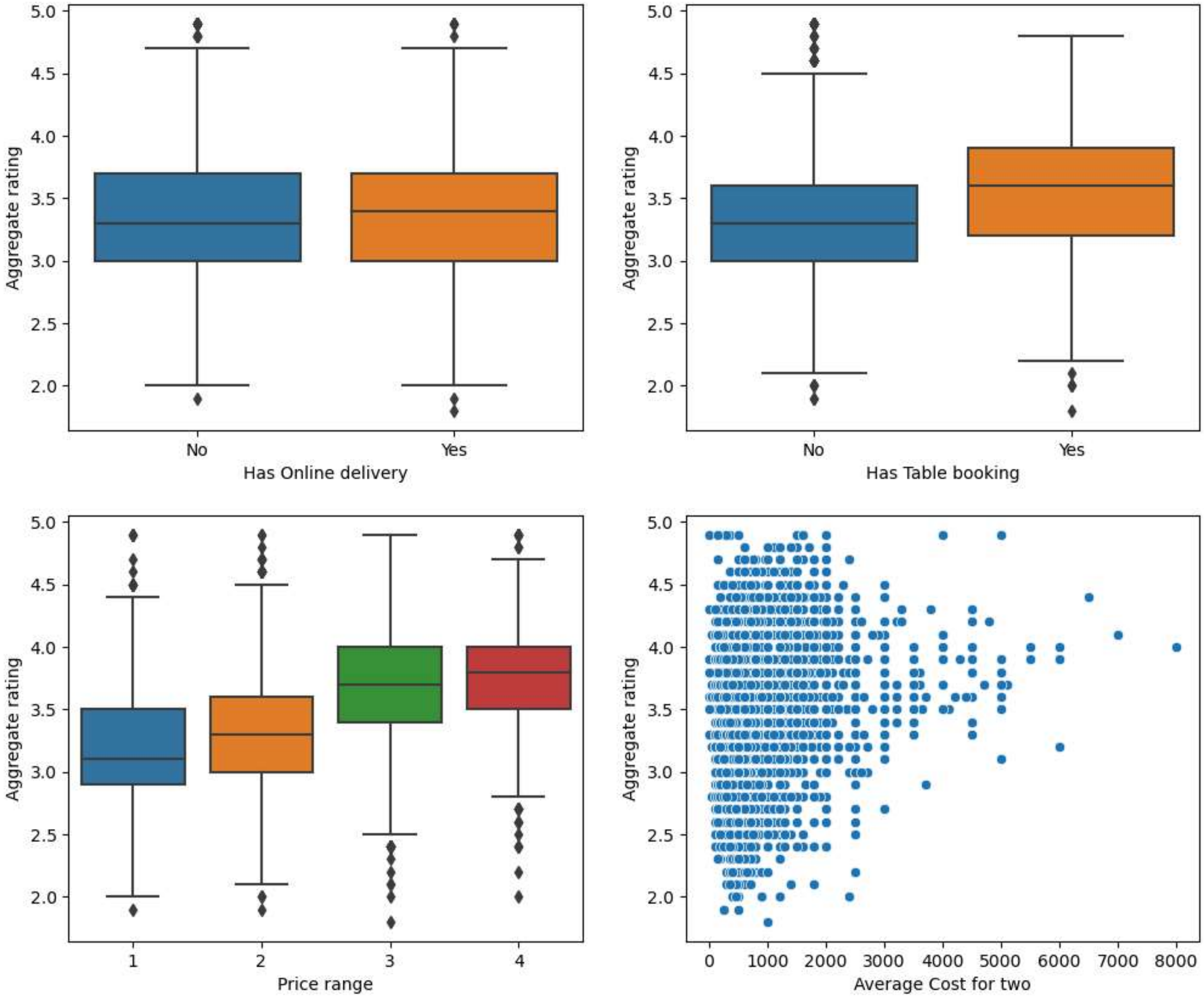
North Indian Cuisine is the most popular,in accordance with zomato's large business in Delhi NCR

Does Various factors Affect Rating?

```
In [66]: sns.pairplot(data=df_i,y_vars=('Aggregate rating'),x_vars=['Has Online delivery','Has Table booking','Price range', 'Average Cost
         plt.show()
```



In [65]:
```
plt.figure(figsize=(12,10))
plt.subplot(2,2,1)
sns.boxplot(x='Has Online delivery',y='Aggregate rating',data=df_i[df_i['Aggregate rating']!=0.0])

plt.subplot(2,2,2)
sns.boxplot(x='Has Table booking',y='Aggregate rating',data=df_i[df_i['Aggregate rating']!=0.0])

plt.subplot(2,2,3)
sns.boxplot(x='Price range',y='Aggregate rating',data=df_i[df_i['Aggregate rating']!=0.0])

plt.subplot(2,2,4)
sns.scatterplot(x='Average Cost for two',y='Aggregate rating',data=df_i[df_i['Aggregate rating']!=0.0])
plt.show()
```



Observations

- Restaurants with online delivery receive higher average rating than those who don't.
Also the former's rating falls in a higher range than the latter. This pattern has negligible outliers hence strongly correlated
- Restaurants with table booking also have higher rating range and average since they are most probably into high end dining category
- Price range also affects rating in similar pattern. Although there's no much difference in average rating for 3 and 4 price range category.
We can also observe more outliers here,indicating failure in maintaining the standard by some restaurants
- Average cost doesn't have a definite effect on rating,a large number of top rated restaurants have low average cost,but the numbers are only proportional to the number of restaurants zomato have in that cost range. high end restaurants do tend to recieve a rating higher than 3.0 but nothing here declares a definite pattern.

Analysing the top city-Delhi

In [9]:
```
df_dl=df_i[df_i['City']=='New Delhi']
df_dl.head(2)
```

Out[9]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Has Table booking	Has Online delivery	Is delivering now	Switch to order menu	Price range	A
2560	18287358	Food Cloud	1	New Delhi	Aaya Nagar, New Delhi	Aaya Nagar	Aaya Nagar, New Delhi	0.000000	0.000000	Cuisine Varies	...	No	No	No	No	2	
2561	18216944	Burger.in	1	New Delhi	84, Near Honda Showroom, Adchini, New Delhi	Adchini	Adchini, New Delhi	77.196923	28.535382	Fast Food	...	No	Yes	No	No	1	

2 rows × 22 columns

Top Rated Restaurants

In [11]:
```
l=df_dl[df_dl['Aggregate rating']>4.5][['Restaurant Name','Aggregate rating']].sort_values(by='Aggregate rating',ascending=False)
```

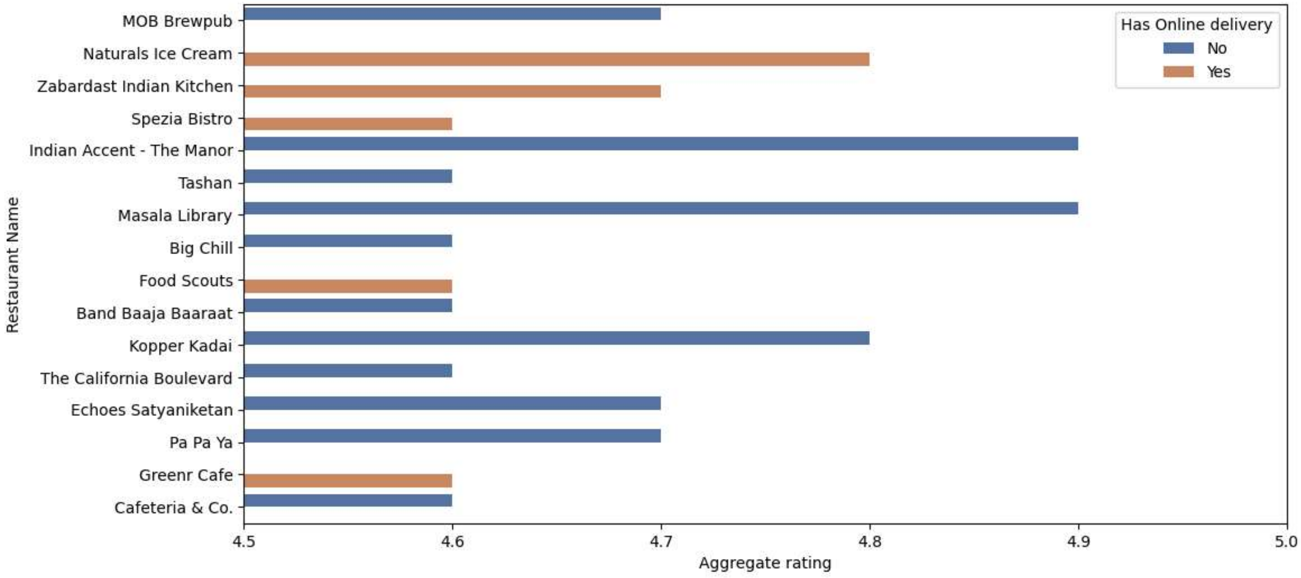
In [12]:
```
top_dl
```

Out[12]:

	Restaurant Name	Aggregate rating
3601	Indian Accent - The Manor	4.9
4298	Masala Library	4.9
3013	Naturals Ice Cream	4.9
6656	Kopper Kadai	4.8
3012	MOB Brewpub	4.7
6657	Naturals Ice Cream	4.7
3014	Zabardast Indian Kitchen	4.7
7067	Pa Pa Ya	4.7
6997	Echoes Satyaniketan	4.7
7180	Greenr Cafe	4.6
6659	The California Boulevard	4.6
6426	Food Scouts	4.6
6655	Band Baaja Baaraat	4.6
4639	Big Chill	4.6
3732	Tashan	4.6
3310	Spezia Bistro	4.6
7849	Cafeteria & Co.	4.6

In [76]:

```
sns.barplot(y='Restaurant Name',x='Aggregate rating',hue='Has Online delivery',data=top_dl,errorbar=None,palette='deep')
plt.xlim([4.5,5])
plt.show()
```



Observation

Only 5 of New Delhi's top rated restaurants has online delivery. Indicating the ratings are largely based on dining experience

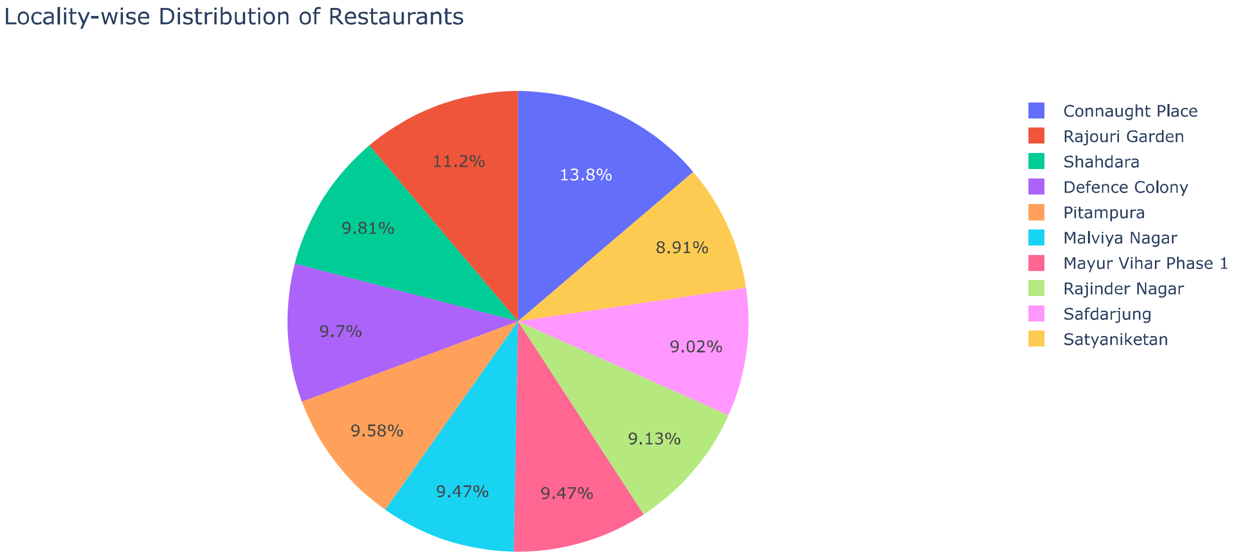
Top Localities

In [13]:

```
import plotly.express as px
```

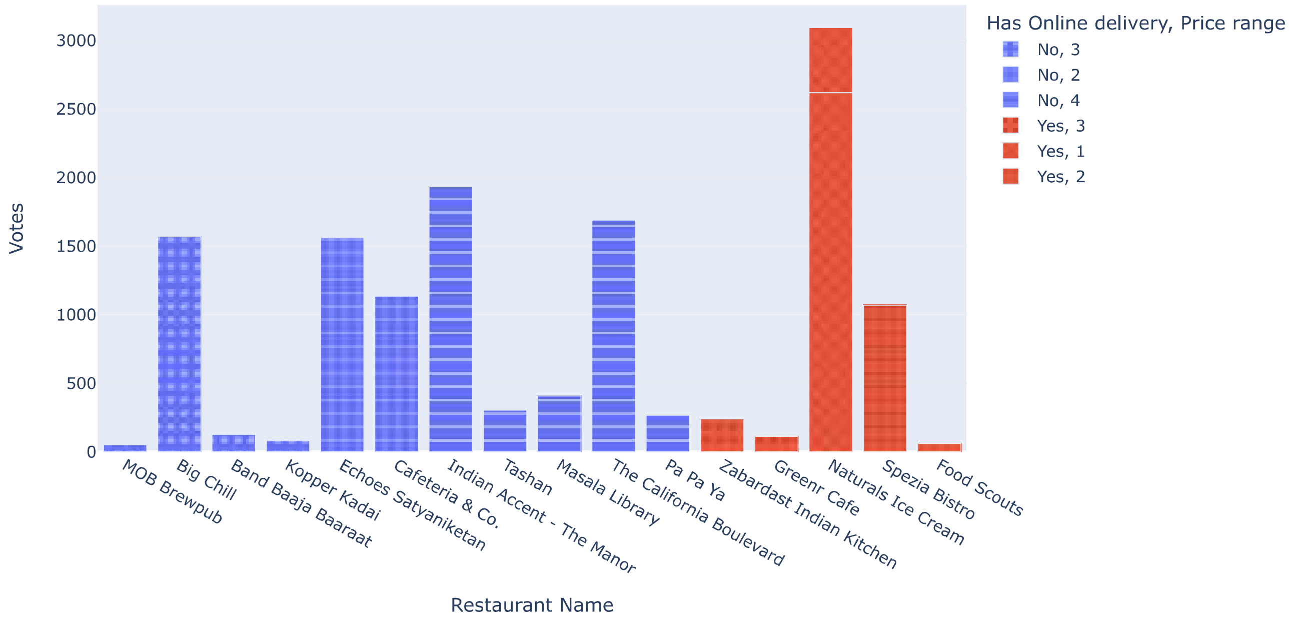
In [14]:

```

labels=df_dl['Locality'].value_counts().head(10).index
x=df_dl['Locality'].value_counts().head(10).values
fig = px.pie(values=x, names=labels, title='Locality-wise Distribution of Restaurants')
fig.show()

```



Comprehensive visualization for Top rated Restaurants

```
In [15]: fig=px.bar(df_dl[df_dl['Aggregate rating']>4.5], x='Restaurant Name', y='Votes', color="Has Online delivery",
                     pattern_shape="Price range", pattern_shape_sequence=[".", "x", "+",'-'])
         fig.show()
```



Observation

- Naturals Ice Cream is the most popular among top rated restaurants,it has online delivery and a low price range of 1
- No restaurant with online delivery and price range of 4 is rated top. Similarly, among offline, no restaurants with price range 1 is rated top
- Most of the top rated restaurants has a price range of 4 and no online delivery, that is, fine dining restaurants

```
In [ ]: 
```