Zomato Dataset Exploratory Data Analysis

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:
```
zomato=pd.read_csv(r"C:\Users\hemil\OneDrive\Desktop\Data Analyst\EDA PYTHON\K
zomato.head()
```

Out[2]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitu
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...	121.027
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...	121.014
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...	121.056
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...	121.056
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...	121.057

5 rows × 21 columns

In [3]:
```
zomato.columns
```

Out[3]:
```
Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Addres
s',
       '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 [4]:
```
zomato.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 [5]:
```
zomato.describe()
```

Out[5]:

	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range	Agg
count	9.551000e+03	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551.
mean	9.051128e+06	18.365616	64.126574	25.854381	1199.210763	1.804837	2.
std	8.791521e+06	56.750546	41.467058	11.007935	16121.183073	0.905609	1.
min	5.300000e+01	1.000000	-157.948486	-41.330428	0.000000	1.000000	0.
25%	3.019625e+05	1.000000	77.081343	28.478713	250.000000	1.000000	2.
50%	6.004089e+06	1.000000	77.191964	28.570469	400.000000	2.000000	3.
75%	1.835229e+07	1.000000	77.282006	28.642758	700.000000	2.000000	3.
max	1.850065e+07	216.000000	174.832089	55.976980	800000.000000	4.000000	4.

In [6]:
```
## 1:finding missing values
## 2:Explore About the Numerical Variables
## 3:Explore about the Categorical Variables
## 4:Finding Relationships between Features
```

In [7]:
```
zomato.isnull().sum()
```

Out[7]:
```
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 [8]:

```
"""What does the heatmap show?

The heatmap visually represents where missing data (NaN values) are located ir
Each cell in the heatmap corresponds to a cell in the DataFrame: cells with mi
Since the yticklabels is set to False, you won't see the row labels, making th
The cbar=False means there's no color bar showing what the colors represent, b
"""
plt.figure(figsize=(15, 10))  # Increase the figure size
sns.heatmap(zomato.isnull(),yticklabels= False,cbar= False,cmap='viridis')
plt.show()
```

In [9]:

```
country=pd.read_excel(r"C:\Users\hemil\OneDrive\Desktop\Data Analyst\EDA PYTHC
country.head()
```

Out[9]:

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

In [10]:

```
##combining both tables(country_code and Zomato as they have country_code colu
final_zomato=pd.merge(zomato,country,on='Country Code',how='left')
final_zomato
```

Out[10]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose
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	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...
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...
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...
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...
...	...	...	...	...	...	...	...
9546	5915730	NamlÛ± Gurme	208	ÛÁstanbul	Kemanke□ô Karamustafa Pa□ôa Mahallesi, RÛ±htÛ±...	Karakí_y	Karakí_y, ÛÁstanbul
9547	5908749	Ceviz AÛôacÛ±	208	ÛÁstanbul	Ko□ôuyolu Mahallesi, Muhittin íìstí_ndaÛô Cadd...	Ko□ôuyolu	Ko□ôuyolu, ÛÁstanbul
9548	5915807	Huqqa	208	ÛÁstanbul	Kuruí_e□ôme Mahallesi, Muallim Naci Caddesi, N...	Kuruí_e□ôme	Kuruí_e□ôme, ÛÁstanbul
9549	5916112	A□ô□ôk Kahve	208	ÛÁstanbul	Kuruí_e□ôme Mahallesi, Muallim Naci Caddesi, N...	Kuruí_e□ôme	Kuruí_e□ôme, ÛÁstanbul
9550	5927402	Walter's Coffee Roastery	208	ÛÁstanbul	CafeaÛôa Mahallesi, BademaltÛ± Sokak, No 21/B,...	Moda	Moda, ÛÁstanbul

9551 rows × 22 columns

In [11]:
```
## To check datatypes of columns
final_zomato.dtypes
```

Out[11]:
```
Restaurant ID             int64
Restaurant Name          object
Country Code              int64
City                     object
Address                  object
Locality                 object
Locality Verbose         object
Longitude               float64
Latitude                float64
Cuisines                 object
Average Cost for two      int64
Currency                 object
Has Table booking        object
Has Online delivery      object
Is delivering now        object
Switch to order menu     object
Price range               int64
Aggregate rating        float64
Rating color             object
Rating text              object
Votes                     int64
Country                  object
dtype: object
```

In [12]:
```
final_zomato.columns
```

Out[12]:
```
Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Addres
s',
       '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')
```

In [13]:

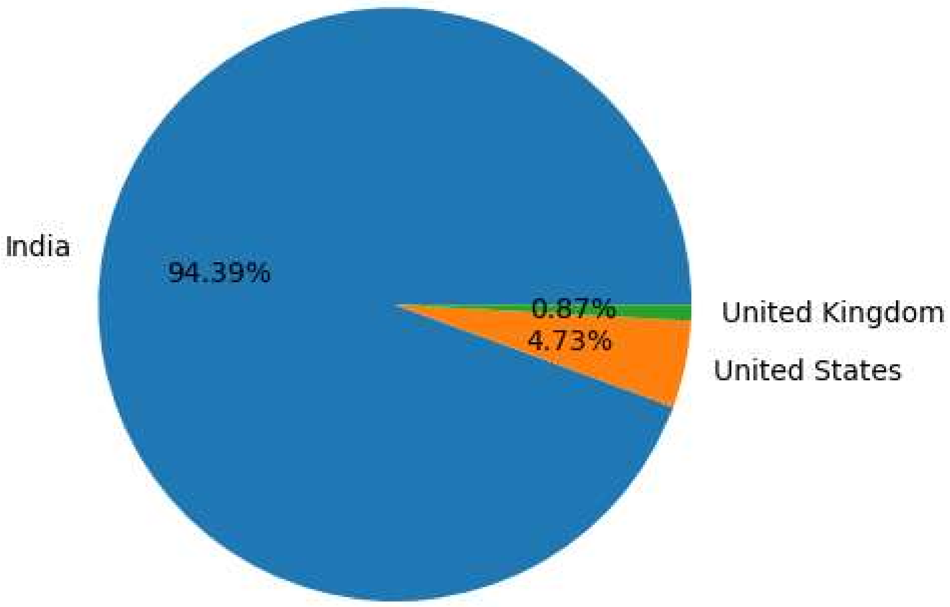
```
## How many records belong to which country(segregation of records based on Co

country_names=final_zomato.Country.value_counts().index ## this is for develop
country_values=final_zomato.Country.value_counts().values ## this is for devel
## top 3 country where zomato sells
plt.pie(country_values[:3],labels=country_names[:3],autopct='%1.2f%%') ##Makin

##Answer:Zomato maximum bussiness comes from India
```

Out[13]:

```
([<matplotlib.patches.Wedge at 0x17ba09dd790>,
  <matplotlib.patches.Wedge at 0x17ba16ac690>,
  <matplotlib.patches.Wedge at 0x17ba16ae3d0>],
 [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%')])
```



In [14]:

```
final_zomato.columns
```

Out[14]:

```
Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Addres
s',
       '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')
```

In [15]:

```
## Relationship between ratings,colors,text using group by function
final_zomato.groupby(['Aggregate rating','Rating color','Rating text']).size()
```

Out[15]:

```
Aggregate rating  Rating color  Rating text
0.0               White         Not rated      2148
1.8               Red           Poor              1
1.9               Red           Poor              2
2.0               Red           Poor              7
2.1               Red           Poor             15
2.2               Red           Poor             27
2.3               Red           Poor             47
2.4               Red           Poor             87
2.5               Orange        Average         110
2.6               Orange        Average         191
2.7               Orange        Average         250
2.8               Orange        Average         315
2.9               Orange        Average         381
3.0               Orange        Average         468
3.1               Orange        Average         519
3.2               Orange        Average         522
3.3               Orange        Average         483
3.4               Orange        Average         498
3.5               Yellow        Good            480
3.6               Yellow        Good            458
3.7               Yellow        Good            427
3.8               Yellow        Good            400
3.9               Yellow        Good            335
4.0               Green         Very Good       266
4.1               Green         Very Good       274
4.2               Green         Very Good       221
4.3               Green         Very Good       174
4.4               Green         Very Good       144
4.5               Dark Green    Excellent        95
4.6               Dark Green    Excellent        78
4.7               Dark Green    Excellent        42
4.8               Dark Green    Excellent        25
4.9               Dark Green    Excellent        61
dtype: int64
```

In [16]:

```
"""Answer:
    rating = 0.0-white-not rated
    rating = 1.8 to 2.4-red-poor
    rating = 2.5-3.4-orange-average
    rating = 3.5-3.9-yellow-good
    rating = 4-4.4-green-very good
    rating = 4.4-4.9-Dark-green-excellent
Rating is between 0 to 5
Many guys have not rated the zomato application
You can also derive many more things from the data"""
```

Out[16]:

```
'Answer:\n    rating = 0.0-white-not rated\n    rating = 1.8 to 2.4-red-poor
\n    rating = 2.5-3.4-orange-average\n    rating = 3.5-3.9-yellow-good\n
rating = 4-4.4-green-very good\n    rating = 4.4-4.9-Dark-green-excellent\nR
ating is between 0 to 5\nMany guys have not rated the zomato application\nYo
u can also derive many more things from the data'
```

In [17]:

```
##This is necessary for making charts out of it
Ratings=final_zomato.groupby(['Aggregate rating','Rating color','Rating text']
Ratings
```

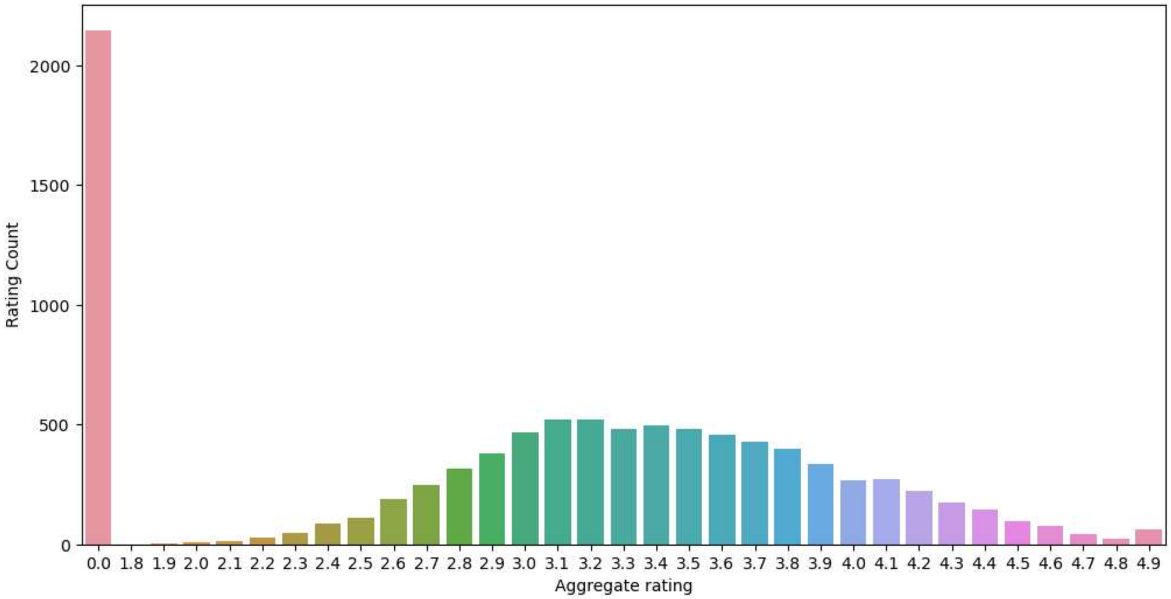
Out[17]:

	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

In [18]:
```
"""
3 reset_index():
Flattening the Result: After using groupby and size, the result is a Series wi
4 rename(columns={0: 'Rating Count'}):
Renaming the Column: The .size() function creates a new column in the DataFran
"""
```

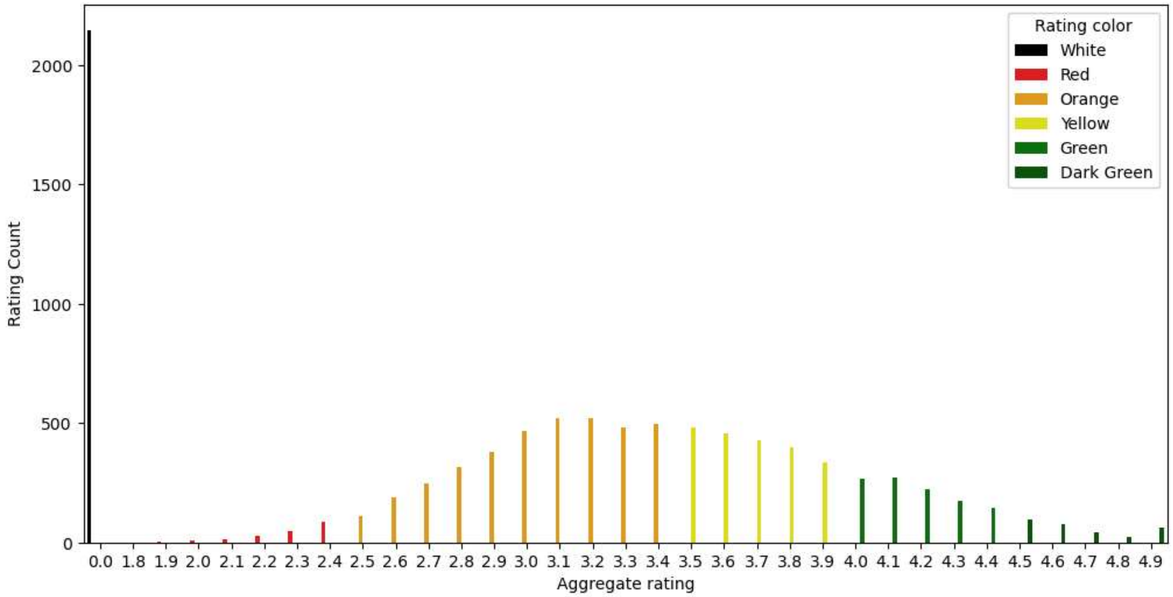
Out[18]:
```
"\n3 reset_index():\nFlattening the Result: After using groupby and size, th
e result is a Series with a multi-index (one index for each of the group-by
columns). The .reset_index() function converts this Series back into a DataF
rame by moving the indices (which are the grouping columns) back into regula
r columns.\n4 rename(columns={0: 'Rating Count'}):\nRenaming the Column: The
.size() function creates a new column in the DataFrame that holds the count
of each group. By default, this column is unnamed (indexed as 0). The .renam
e(columns={0: 'Rating Count'}) part renames this column to 'Rating Count' fo
r clarity.\n"
```

In [19]:
```
# Assuming Ratings is your DataFrame and it has already been created
plt.figure(figsize=(12, 6))
sns.barplot(x='Aggregate rating', y='Rating Count', data=Ratings)
# Show the plot
plt.show()
```



In [20]:

```
# More sophisticated chart(details are same as above)
# Plot the barplot with the filtered data
plt.figure(figsize=(12, 6))
sns.barplot(x='Aggregate rating', y='Rating Count',hue='Rating color', data=Ra
plt.show()
```



In [21]:

```
"""
Answer:
1:Not Rated counts are very high
2:Maximum number of rating are between 2.5 to 3.4
"""
```

Out[21]:

```
'\nAnswer:\n1:Not Rated counts are very high\n2:Maximum number of rating are
between 2.5 to 3.4\n'
```

In [22]:

```
### find the countries name that has given 0 ratings

countries_with_zero_ratings=final_zomato[final_zomato['Aggregate rating']==0]
countries_with_zero_ratings
```

Out[22]:

	Country	No of Zero Ratings
0	Brazil	5
1	India	2139
2	United Kingdom	1
3	United States	3

""" Answer: Indians have given most number of zero Ratings, followed by Brazillians and United States people """

In [23]:
```
###Find out which currency is used by which country
final_zomato[['Country','Currency']].groupby(['Country','Currency']).size().re
```

Out[23]:

	Country	Currency	0
0	Australia	Dollar($)	24
1	Brazil	Brazilian Real(R$)	60
2	Canada	Dollar($)	4
3	India	Indian Rupees(Rs.)	8652
4	Indonesia	Indonesian Rupiah(IDR)	21
5	New Zealand	NewZealand($)	40
6	Phillipines	Botswana Pula(P)	22
7	Qatar	Qatari Rial(QR)	20
8	Singapore	Dollar($)	20
9	South Africa	Rand(R)	60
10	Sri Lanka	Sri Lankan Rupee(LKR)	20
11	Turkey	Turkish Lira(TL)	34
12	UAE	Emirati Diram(AED)	60
13	United Kingdom	Pounds(□£)	80
14	United States	Dollar($)	434

""" Answer: Insights from the data is Zomato sells in 14 Countries,Most of its sells comes from India and USA """

In [24]:
```
### Which countries do have online deliveries option
Countries_with_online_deliveries=final_zomato[final_zomato['Has Online deliver
Countries_with_online_deliveries
```

Out[24]:

	Country	Have Online Delivery
0	India	2423
1	UAE	28

In [25]:
```
"""
Answer:It shows that online deliveries option is only available In India and U
Majorly in India
"""
```

Out[25]:
```
'\nAnswer:It shows that online deliveries option is only available In India
and UAE.\nMajorly in India\n'
```

In [26]:

```
 ### Which countries do not have online deliveries option
Countries_with_online_deliveries=final_zomato[final_zomato['Has Online delive
Countries_with_online_deliveries
```

Out[26]:

	Country	No Online Delivery
0	Australia	24
1	Brazil	60
2	Canada	4
3	India	6229
4	Indonesia	21
5	New Zealand	40
6	Phillipines	22
7	Qatar	20
8	Singapore	20
9	South Africa	60
10	Sri Lanka	20
11	Turkey	34
12	UAE	32
13	United Kingdom	80
14	United States	434

""" Answer: From this you can infer that India has Online delivery in some part only, while most of the countries does not have online deliveries """

In [27]:

```
### Create a pie chart for cities distribution(Cities where most orders are pl
top_cities=final_zomato[['City']].groupby(['City']).size().sort_values(ascendi
top_cities
```

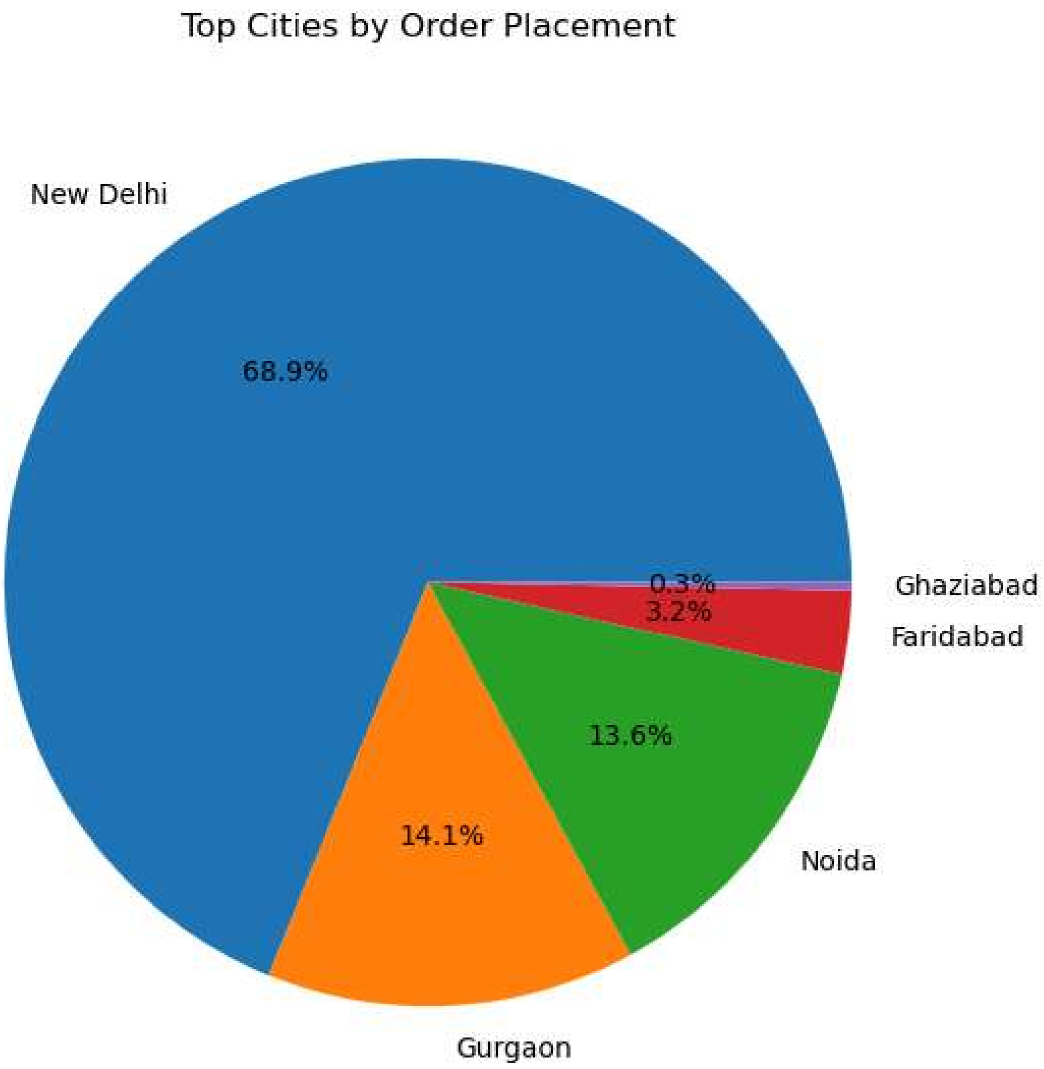
Out[27]:

	City	Orders_Placed
0	New Delhi	5473
1	Gurgaon	1118
2	Noida	1080
3	Faridabad	251
4	Ghaziabad	25

In [28]:

```
plt.figure(figsize=(10, 7))
plt.pie(top_cities['Orders_Placed'], labels=top_cities['City'], autopct='%1.1

# Add title
plt.title('Top Cities by Order Placement')
plt.show()
```



In [29]:

```
## find the Top 10 Cuisines
Top_Cuisines=final_zomato[['Cuisines']].groupby(['Cuisines']).size().sort_valu
Top_Cuisines
```

Out[29]:

	Cuisines	Most loved Cuisines
0	North Indian	936
1	North Indian, Chinese	511
2	Chinese	354
3	Fast Food	354
4	North Indian, Mughlai	334
5	Cafe	299
6	Bakery	218
7	North Indian, Mughlai, Chinese	197
8	Bakery, Desserts	170
9	Street Food	149

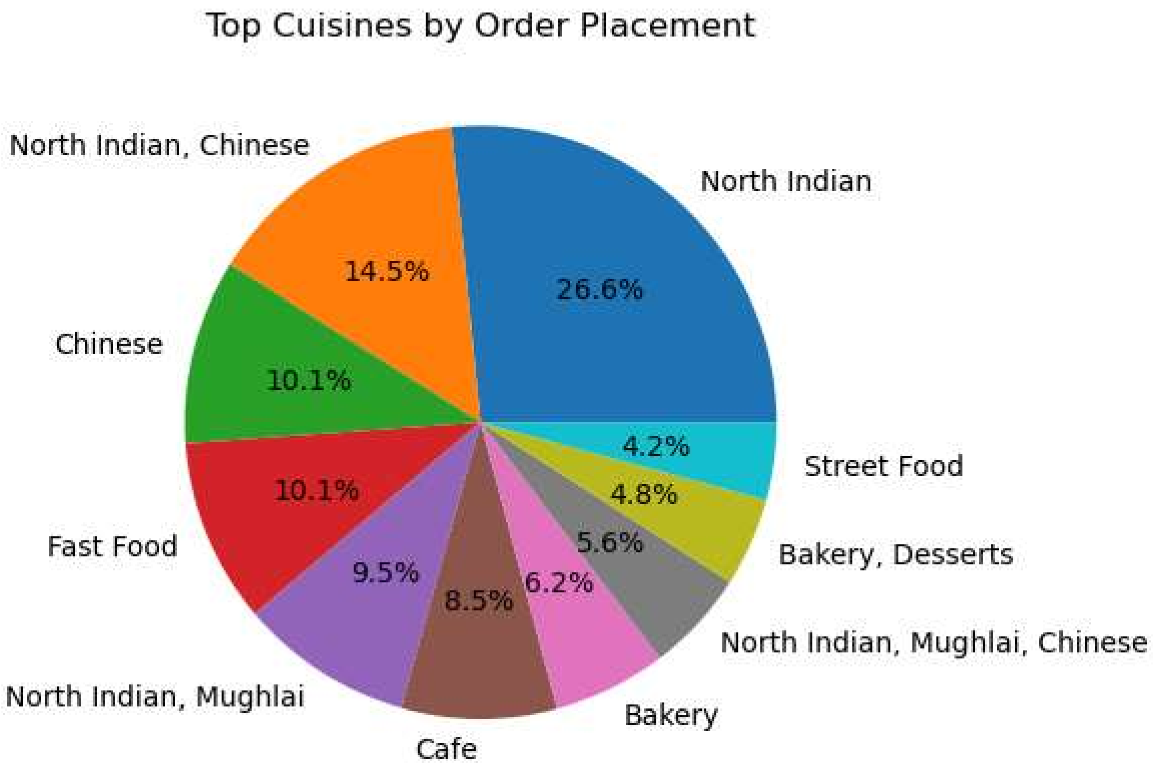
In [30]:

```
"""
Answer:Most loved cuisines are North Indian,Chinese,Mughlai
"""
```

Out[30]:

```
'\nAnswer:Most loved cuisines are North Indian,Chinese,Mughlai\n'
```

```
In [31]: plt.pie(Top_Cuisines['Most loved Cuisines'],labels=Top_Cuisines['Cuisines'],au
         # Add title
         plt.title('Top Cuisines by Order Placement')
         plt.show()
```



```
In [ ]:
```