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]:
```
df=pd.read_csv('zomato.csv',encoding='latin-1')
df.head()
```

Out[2]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longituc
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...	121.02753
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...	121.01410
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...	121.05683
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...	121.05647
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...	121.05750

5 rows × 21 columns

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

Out[3]:
```
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 [4]:
```
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 [5]:
```
df.describe()
```

Out[5]:

	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range	Aggre r
count	9.551000e+03	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551.00
mean	9.051128e+06	18.365616	64.126574	25.854381	1199.210763	1.804837	2.66
std	8.791521e+06	56.750546	41.467058	11.007935	16121.183073	0.905609	1.51
min	5.300000e+01	1.000000	-157.948486	-41.330428	0.000000	1.000000	0.00
25%	3.019625e+05	1.000000	77.081343	28.478713	250.000000	1.000000	2.50
50%	6.004089e+06	1.000000	77.191964	28.570469	400.000000	2.000000	3.20
75%	1.835229e+07	1.000000	77.282006	28.642758	700.000000	2.000000	3.70
max	1.850065e+07	216.000000	174.832089	55.976980	800000.000000	4.000000	4.90

In Data Analysis What All Things We Do

1. Missing Values
2. Explore About the Numerical Variables
3. Explore About categorical Variables
4. Finding Relationship between features

In [8]:
```
df.shape
```

Out[8]:
```
(9551, 21)
```

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
```

In [10]:
```
df.isnull().sum()
```

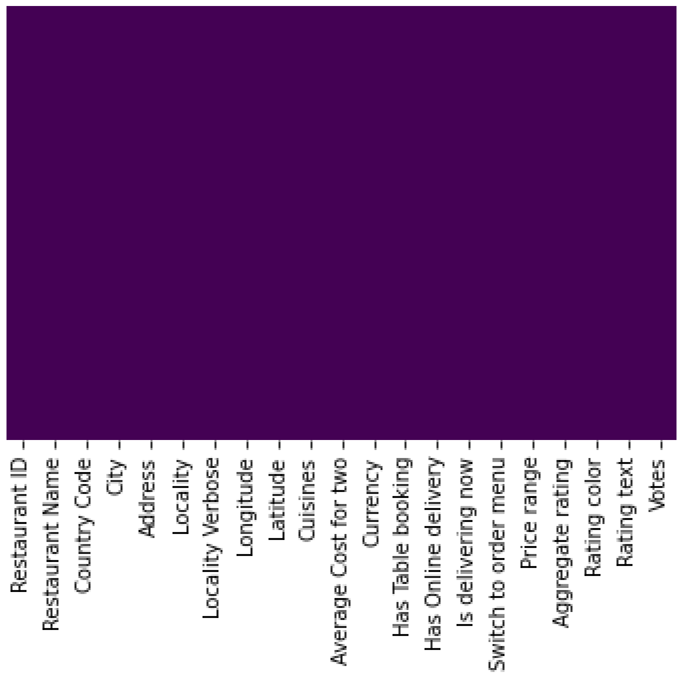
Out[10]:
```
Restaurant ID           0
Restaurant Name         0
Country Code            0
City                    0
Address                 0
Locality                0
Locality Verbose        0
Longitude               0
Latitude                0
Cuisines                9
Average Cost for two    0
Currency                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 [6]:
```
[features for features in df.columns if df[features].isnull().sum()>0]
```

Out[6]:
```
['Cuisines']
```

In [7]:
```
sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
```

Out[7]:
```
<AxesSubplot:>
```



In [8]:
```
df_country=pd.read_excel('Country-Code.xlsx')
df_country.head()
```

Out[8]:

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

In [9]:
```
df.columns
```

Out[9]:
```
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 [10]:
```
final_df=pd.merge(df,df_country,on='Country Code', how='left')
```

In [11]:
```
final_df.head(2)
```

Out[11]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitud
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	14.56544
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	14.55370

2 rows × 22 columns

In [12]:
```
##To check Data Types
final_df.dtypes
```

Out[12]:
```
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 [13]:
```
final_df.columns
```

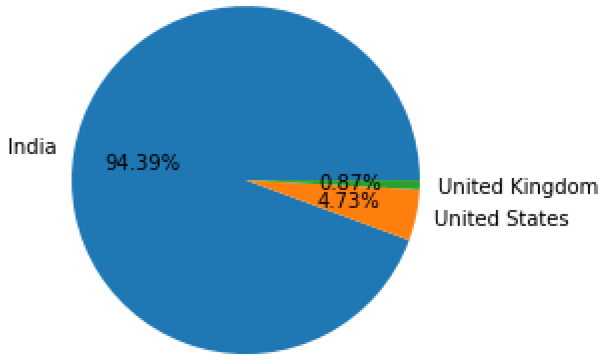
Out[13]:
```
Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City', 'Address',
       'Locality', 'Locality Verbose', 'Longitude', 'Latitude', 'Cuisines',
       'Average Cost for two', 'Currency', 'Has Table booking',
       'Has Online delivery', 'Is delivering now', 'Switch to order menu',
       'Price range', 'Aggregate rating', 'Rating color', 'Rating text',
       'Votes', 'Country'],
      dtype='object')
```

In [14]:
```
country_names=final_df.Country.value_counts().index
```

In [15]:
```
country_val=final_df.Country.value_counts().values
```

In [16]:
```
## Pie Chart- Top 3 countries that uses zomato
plt.pie(country_val[:3],labels=country_names[:3],autopct='%1.2f%%')
```

Out[16]:
```
([<matplotlib.patches.Wedge at 0x20c00cfd220>,
  <matplotlib.patches.Wedge at 0x20c00cf6a60>,
  <matplotlib.patches.Wedge at 0x20c00d59c10>],
 [Text(-1.0829742700952103, 0.19278674827836725, 'India'),
  Text(1.077281715838356, -0.22240527134123297, 'United States'),
  Text(1.0995865153823035, -0.030157837943120773, '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 maximum records or transaction are from India After that USA and then United Kingdoms

In [17]:
```
final_df.columns
```

Out[17]:
```
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')
```

In [18]:
```
ratings=final_df.groupby(['Aggregate rating','Rating text']).size().reset_index
```

In [19]:
```
ratings
```

Out[19]:

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

Observation

1. When Rating is between 4.5 to 4.9---> Excellent
2. When Rating are between 4.0 to 3.4--->very good
3. when Rating is between 3.5 to 3.9----> good
4. when Rating is between 3.0 to 3.4----> average
5. when Rating is between 2.5 to 2.9----> average
6. when Rating is between 2.0 to 2.4----> Poor

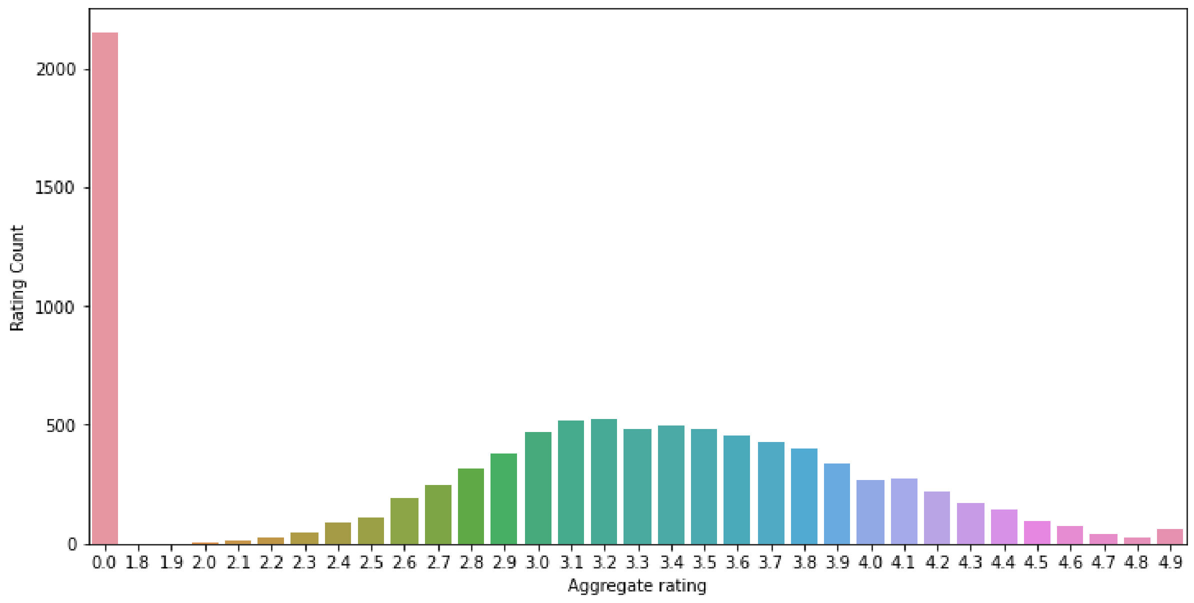
In [20]:
```
ratings.head()
```

Out[20]:

	Aggregate rating	Rating text	Rating Count
0	0.0	Not rated	2148
1	1.8	Poor	1
2	1.9	Poor	2
3	2.0	Poor	7
4	2.1	Poor	15

In [21]:
```
import matplotlib
matplotlib.rcParams['figure.figsize'] = (12, 6)
sns.barplot(x="Aggregate rating",y="Rating Count",data=ratings)
```

Out[21]:
```
<AxesSubplot:xlabel='Aggregate rating', ylabel='Rating Count'>
```



In [22]:
```
sns.barplot(x="Aggregate rating",y="Rating Count",hue='Rating color',data=ratir
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_12216/1943702941.py in <module>
----> 1 sns.barplot(x="Aggregate rating",y="Rating Count",hue='Rating color',
data=ratings,palette=['blue','red','orange','yellow','green','green'])

~\anaconda3\lib\site-packages\seaborn\_decorators.py in inner_f(*args, **kwar
gs)
     44             )
     45         kwargs.update({k: arg for k, arg in zip(sig.parameters, arg
s)})
---> 46         return f(**kwargs)
     47     return inner_f
     48

~\anaconda3\lib\site-packages\seaborn\categorical.py in barplot(x, y, hue, da
ta, order, hue_order, estimator, ci, n_boot, units, seed, orient, color, pale
tte, saturation, errcolor, errwidth, capsize, dodge, ax, **kwargs)
   3180 ):
   3181
-> 3182     plotter = _BarPlotter(x, y, hue, data, order, hue_order,
   3183                           estimator, ci, n_boot, units, seed,
   3184                           orient, color, palette, saturation,

~\anaconda3\lib\site-packages\seaborn\categorical.py in __init__(self, x, y,
hue, data, order, hue_order, estimator, ci, n_boot, units, seed, orient, colo
r, palette, saturation, errcolor, errwidth, capsize, dodge)
   1582                  errwidth, capsize, dodge):
   1583         """Initialize the plotter."""
-> 1584         self.establish_variables(x, y, hue, data, orient,
   1585                                  order, hue_order, units)
   1586         self.establish_colors(color, palette, saturation)

~\anaconda3\lib\site-packages\seaborn\categorical.py in establish_variables(s
elf, x, y, hue, data, orient, order, hue_order, units)
    151                 if isinstance(var, str):
    152                     err = "Could not interpret input '{}'".format(va
r)
--> 153                     raise ValueError(err)
    154
    155             # Figure out the plotting orientation

ValueError: Could not interpret input 'Rating color'
```

Observation:

1. Not Rated count is very high
2. Maximum number of rating are between 2.5 to 3.4

```
In [23]: ## Count plot
         sns.countplot(x="Rating color",data=ratings,palette=['blue','red','orange','yel
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_12216/2576527937.py in <module>
      1 ## Count plot
----> 2 sns.countplot(x="Rating color",data=ratings,palette=['blue','red','or
ange','yellow','green','green'])

~\anaconda3\lib\site-packages\seaborn\_decorators.py in inner_f(*args, **kwar
gs)
     44             )
     45         kwargs.update({k: arg for k, arg in zip(sig.parameters, arg
s)})
---> 46         return f(**kwargs)
     47     return inner_f
     48 

~\anaconda3\lib\site-packages\seaborn\categorical.py in countplot(x, y, hue,
data, order, hue_order, orient, color, palette, saturation, dodge, ax, **kwar
gs)
   3596         raise ValueError("Cannot pass values for both `x` and `y`")
   3597 
-> 3598     plotter = _CountPlotter(
   3599         x, y, hue, data, order, hue_order,
   3600         estimator, ci, n_boot, units, seed,

~\anaconda3\lib\site-packages\seaborn\categorical.py in __init__(self, x, y,
hue, data, order, hue_order, estimator, ci, n_boot, units, seed, orient, colo
r, palette, saturation, errcolor, errwidth, capsize, dodge)
   1582                  errwidth, capsize, dodge):
   1583         """Initialize the plotter."""
-> 1584         self.establish_variables(x, y, hue, data, orient,
   1585                                  order, hue_order, units)
   1586         self.establish_colors(color, palette, saturation)

~\anaconda3\lib\site-packages\seaborn\categorical.py in establish_variables(s
elf, x, y, hue, data, orient, order, hue_order, units)
    151                     if isinstance(var, str):
    152                         err = "Could not interpret input '{}'".format(va
r)
--> 153                         raise ValueError(err)
    154 
    155             # Figure out the plotting orientation

ValueError: Could not interpret input 'Rating color'
```

In [24]:
```
ratings
```

Out[24]:

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

In [25]:
```
### Find the countries name that has given 0 rating
final_df[final_df['Rating color']=='White'].groupby('Country').size().reset_inc
```

Out[25]:

	Country	0
0	Brazil	5
1	India	2139
2	United Kingdom	1
3	United States	3

In [26]:
```
final_df.groupby(['Aggregate rating','Country']).size().reset_index().head(5)
```

Out[26]:

	Aggregate rating	Country	0
0	0.0	Brazil	5
1	0.0	India	2139
2	0.0	United Kingdom	1
3	0.0	United States	3
4	1.8	India	1

Observations Maximum number of 0 ratings are from Indian customers

In [27]:
```
##find out which currency is used by which country?
final_df.columns
```

Out[27]:
```
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')
```

In [28]:
```
final_df[['Country','Currency']].groupby(['Country','Currency']).size().reset_i
```

Out[28]:

	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

In [29]:
```
## Which Countries do have online deliveries option
```

In [30]:
```
final_df[final_df['Has Online delivery'] =="Yes"].Country.value_counts()
```

Out[30]:
```
India    2423
UAE        28
Name: Country, dtype: int64
```

In [31]:
```
final_df[['Has Online delivery','Country']].groupby(['Has Online delivery','Cou
```

Out[31]:

	Has Online delivery	Country	0
0	No	Australia	24
1	No	Brazil	60
2	No	Canada	4
3	No	India	6229
4	No	Indonesia	21
5	No	New Zealand	40
6	No	Phillipines	22
7	No	Qatar	20
8	No	Singapore	20
9	No	South Africa	60
10	No	Sri Lanka	20
11	No	Turkey	34
12	No	UAE	32
13	No	United Kingdom	80
14	No	United States	434
15	Yes	India	2423
16	Yes	UAE	28

Observations:

1. Online Deliveries are available in India and UAE

In [32]:
```
final_df.columns
```

Out[32]:
```
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')
```

In [33]:
```
## Create a pie chart for top 5 cities distribution
```

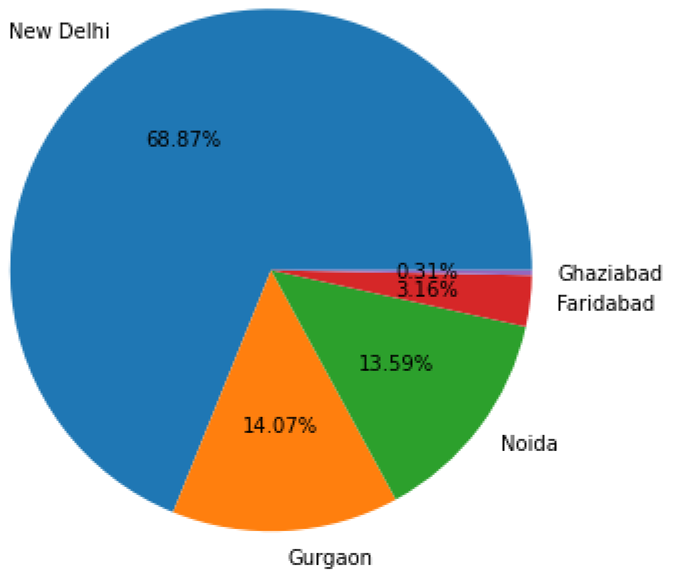
In [34]:
```
final_df.City.value_counts().index
```

Out[34]:
```
Index(['New Delhi', 'Gurgaon', 'Noida', 'Faridabad', 'Ghaziabad',
       'Bhubaneshwar', 'Amritsar', 'Ahmedabad', 'Lucknow', 'Guwahati',
       ...
       'Ojo Caliente', 'Montville', 'Monroe', 'Miller', 'Middleton Beach',
       'Panchkula', 'Mc Millan', 'Mayfield', 'Macedon', 'Vineland Station'],
      dtype='object', length=141)
```

In [35]:
```
city_values=final_df.City.value_counts().values
city_labels=final_df.City.value_counts().index
```

In [36]:
```
plt.pie(city_values[:5],labels=city_labels[:5],autopct='%1.2f%%')
```

Out[36]:
```
([<matplotlib.patches.Wedge at 0x20c010591f0>,
  <matplotlib.patches.Wedge at 0x20c01059970>,
  <matplotlib.patches.Wedge at 0x20c01041160>,
  <matplotlib.patches.Wedge at 0x20c01066760>,
  <matplotlib.patches.Wedge at 0x20c01066e80>],
 [Text(-0.6145352824185932, 0.9123301960708633, 'New Delhi'),
  Text(0.0623675251198054, -1.0982305276263407, 'Gurgaon'),
  Text(0.8789045225625368, -0.6614581167535246, 'Noida'),
  Text(1.0922218418223437, -0.13058119407559224, 'Faridabad'),
  Text(1.099946280005612, -0.010871113182029924, 'Ghaziabad')],
 [Text(-0.3352010631374145, 0.497634652402289, '68.87%'),
  Text(0.0340186500653484, -0.5990348332507311, '14.07%'),
  Text(0.47940246685229276, -0.36079533641101336, '13.59%'),
  Text(0.5957573682667329, -0.07122610585941394, '3.16%'),
  Text(0.5999706981848791, -0.005929698099289049, '0.31%')])
```


New Delhi
Ghaziabad
Faridabad
Noida
Gurgaon