Exploratory data analysis on Zomato Dataset

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

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

In [ ]:
```
df.head()
```

Out[ ]:

	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	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	14.58140
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	14.58531
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	14.58445

5 rows × 21 columns

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

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

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

Out[ ]:
```
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 [ ]:
```
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 [ ]:
```
df["Restaurant Name"].value_counts()
```

Out[ ]:
```
Cafe Coffee Day             83
Domino's Pizza              79
Subway                      63
Green Chick Chop            51
McDonald's                  48
                            ..
Odeon Social                 1
Johnny Rockets               1
House of Commons             1
HotMess                      1
Walter's Coffee Roastery     1
Name: Restaurant Name, Length: 7446, dtype: int64
```

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

Out[ ]:

	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range	Aggregate rating	V
count	9.551000e+03	9551.000000	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	2.666370	156.90
std	8.791521e+06	56.750546	41.467058	11.007935	16121.183073	0.905609	1.516378	430.16
min	5.300000e+01	1.000000	-157.948486	-41.330428	0.000000	1.000000	0.000000	0.00
25%	3.019625e+05	1.000000	77.081343	28.478713	250.000000	1.000000	2.500000	5.00
50%	6.004089e+06	1.000000	77.191964	28.570469	400.000000	2.000000	3.200000	31.00
75%	1.835229e+07	1.000000	77.282006	28.642758	700.000000	2.000000	3.700000	131.00
max	1.850065e+07	216.000000	174.832089	55.976980	800000.000000	4.000000	4.900000	10934.00

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

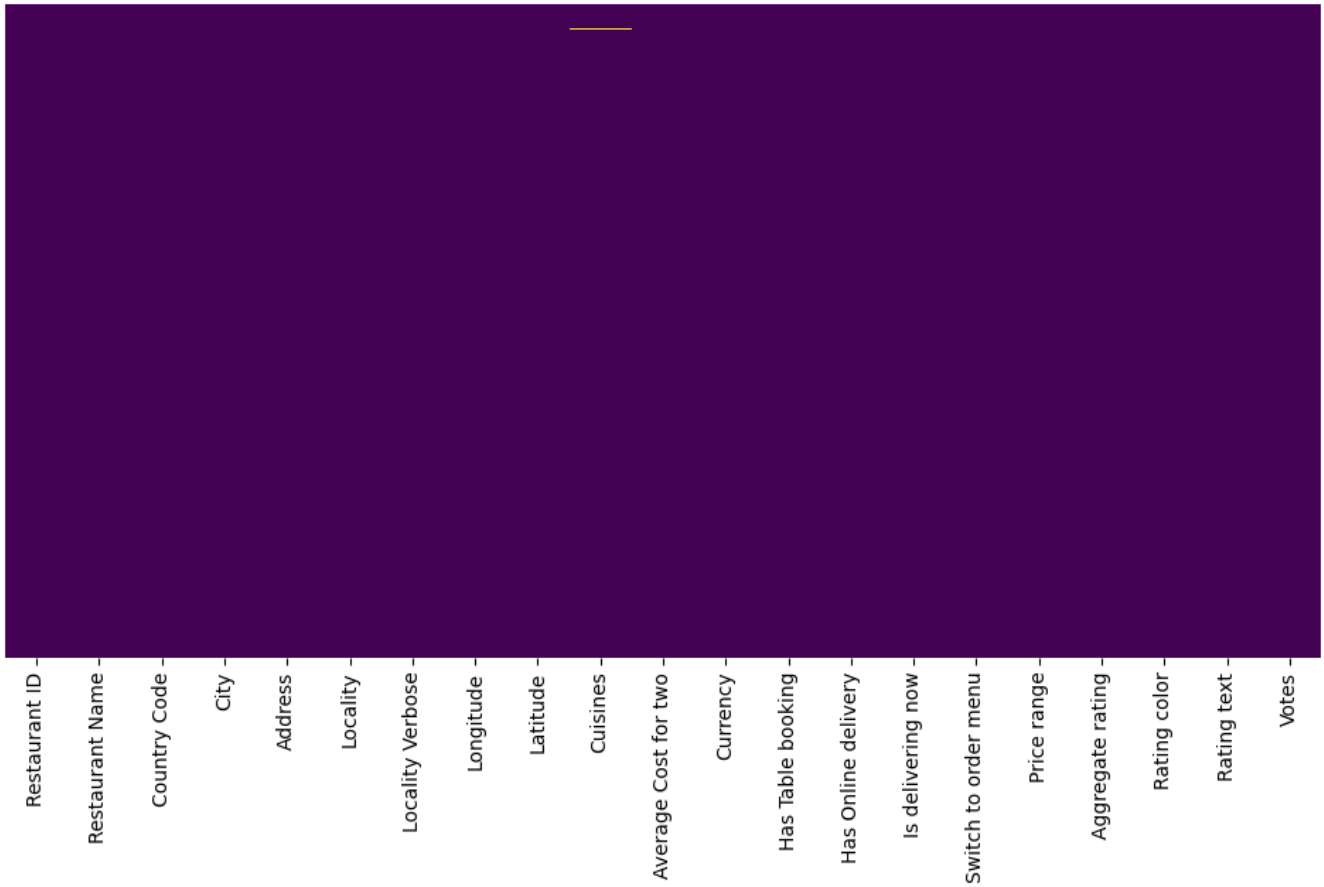
Out[ ]:
```
Restaurant ID         0
Restaurant Name       0
Country Code          0
City                  0
Address               0
Locality              0
Locality Verbose      0
Longitude             0
Latitude              0
Cuisines              9
Average Cost for two  0
Currency              0
Has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

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

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

Out[ ]:
```
<Axes: >
```



In [ ]:
```
! pip install openpyxl
```

```
Requirement already satisfied: openpyxl in c:\python 3.11.12\lib\site-packages (3.1.2)
Requirement already satisfied: et-xmlfile in c:\python 3.11.12\lib\site-packages (from openpy
xl) (1.1.0)
```

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

In [ ]:
```
df_country.head()
```

Out[ ]:

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

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

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

In [ ]:
```
new_df.head()
```

Out[ ]:

	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	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	14.58140
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	14.58531
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	14.58445

5 rows × 22 columns

In [ ]:
```
new_df.dtypes
```

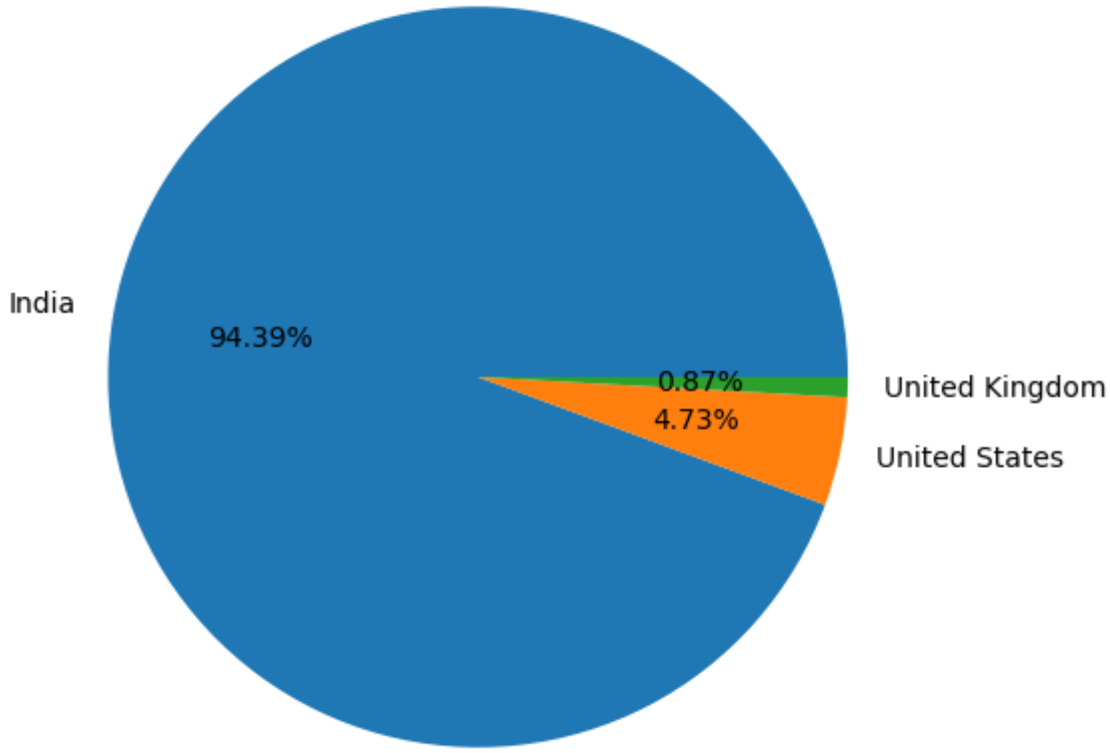
Out[ ]:
```
Restaurant ID             int64
Restaurant Name          object
Country Code              int64
City                     object
Address                  object
Locality                 object
Locality Verbose         object
Longitude               float64
Latitude                float64
Cuisines                 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 [ ]:
```
country_names = new_df['Country'].value_counts().index
```

In [ ]:
```
country_values = new_df['Country'].value_counts().values
```

In [ ]:
```
### pie chart, top 3 countries from where the zomato is making  most of its business
plt.pie(country_values[:3], labels = country_names[:3], autopct="%1.2f%%")
```

Out[ ]:
```
([<matplotlib.patches.Wedge at 0x21032c2da50>,
  <matplotlib.patches.Wedge at 0x21032c38e50>,
  <matplotlib.patches.Wedge at 0x21032c3a690>],
 [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%')])
```



Top three countries that uses zomato: Maximum transaction happens in India followed by United States and then United Kingdom

In [ ]:
```
new_df.head()
```

Out[ ]:

	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	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	14.58140
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	14.58531
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	14.58445

5 rows × 22 columns

In [ ]:
```
new_df.columns
```

Out[ ]:
```
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 [ ]:
```
new_df['Aggregate rating'].value_counts().head()
```

Out[ ]:
```
0.0    2148
3.2     522
3.1     519
3.4     498
3.3     483
Name: Aggregate rating, dtype: int64
```

In [ ]:
```
new_df['Rating color'].value_counts()
```

Out[ ]:
```
Orange         3737
White          2148
Yellow         2100
Green          1079
Dark Green      301
Red             186
Name: Rating color, dtype: int64
```

In [ ]:
```
new_df['Rating text'].value_counts()
```

Out[ ]:
```
Average      3737
Not rated    2148
Good         2100
Very Good    1079
Excellent     301
Poor          186
Name: Rating text, dtype: int64
```

In [ ]:
```
new_df.groupby(['Aggregate rating', 'Rating color', 'Rating text']).size().reset_index().rena
```

Out[ ]:

	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 [ ]:
```
ratings = new_df.groupby(['Aggregate rating', 'Rating color', 'Rating text']).size().reset_in
```

In [ ]:
```
ratings
```

Out[ ]:

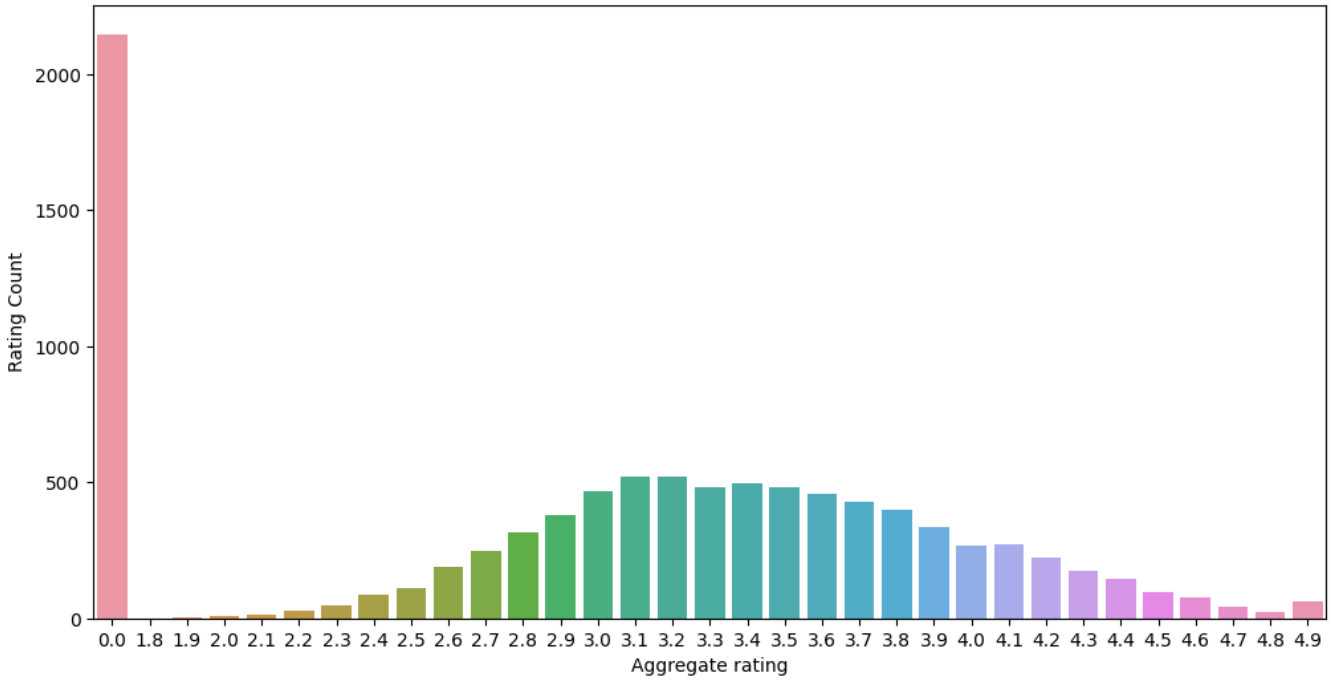
	Aggregate rating	Rating color	Rating text	Rating Count
0	0.0	White	Not rated	2148
1	1.8	Red	Poor	1
2	1.9	Red	Poor	2
3	2.0	Red	Poor	7
4	2.1	Red	Poor	15
5	2.2	Red	Poor	27
6	2.3	Red	Poor	47
7	2.4	Red	Poor	87
8	2.5	Orange	Average	110
9	2.6	Orange	Average	191
10	2.7	Orange	Average	250
11	2.8	Orange	Average	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 [ ]:

```
### Observations  based on aggregate ratings
### when Rating is between 1.8 to 2.4----poor
### when Rating is between 2.5 to 3.4----Average
### when Rating is between 3.5 to 3.9----Good
### when Rating is between 4.0 to 4.4----Very Good
### when Rating is between 4.5 to 4.9----Excellent
```

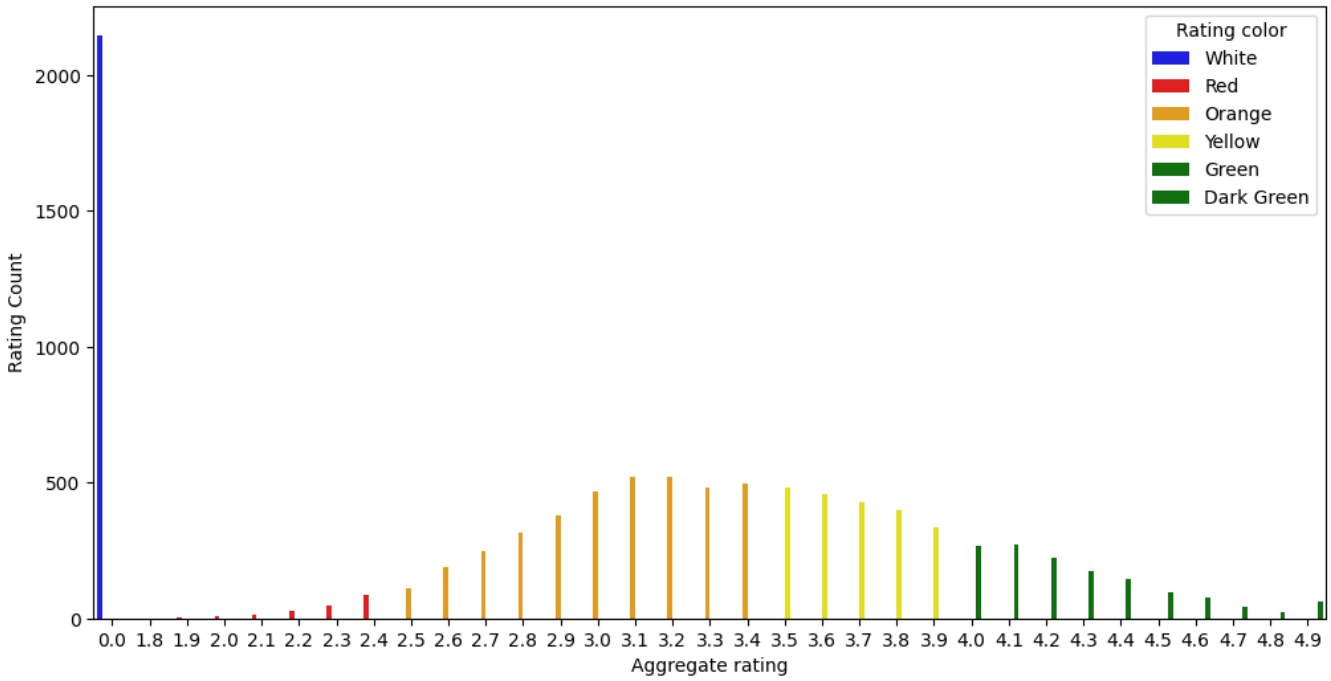
In [ ]:
```
import matplotlib
matplotlib.rcParams['figure.figsize'] =(12,6)
sns.barplot(x="Aggregate rating", y="Rating Count", data = ratings)
```

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



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

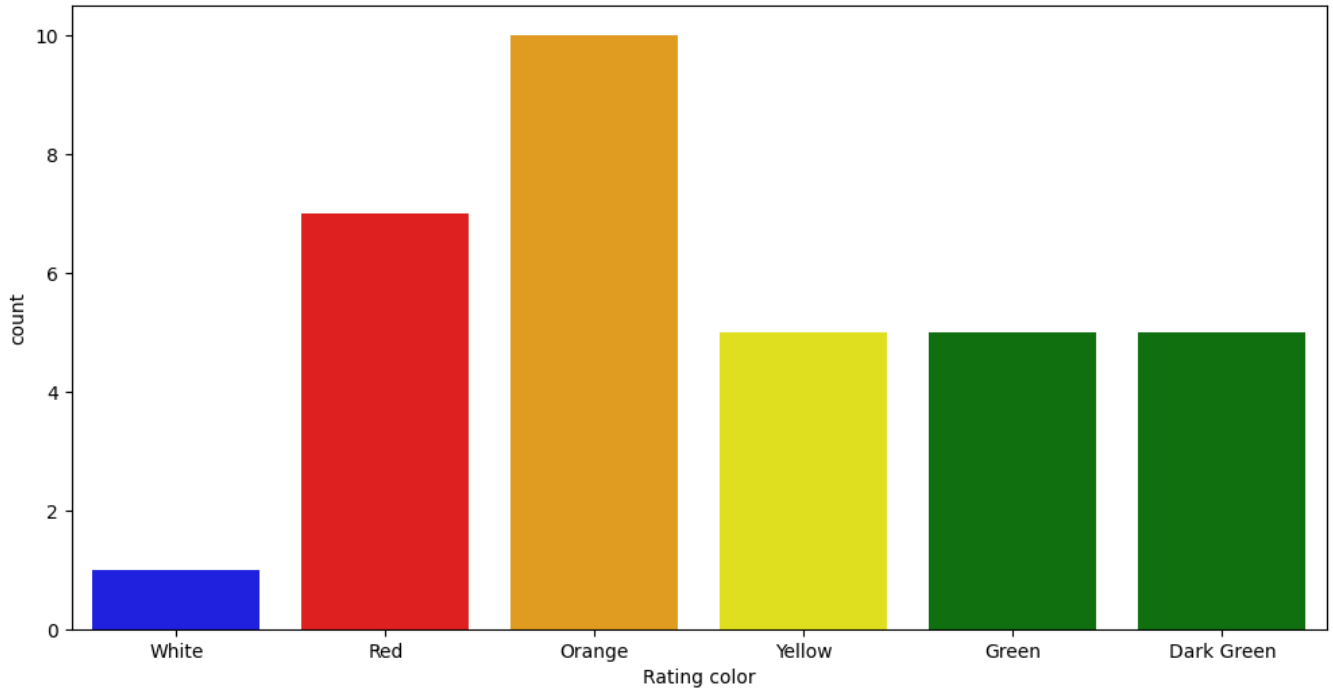
Out[ ]:
```
<Axes: xlabel='Aggregate rating', ylabel='Rating Count'>
```



Maximum number of people have not given the ratings , and average number of rating liea between 2.5 to 4.5

In [ ]:
```
sns.countplot(x='Rating color', data = ratings, palette=["blue", "red", "orange", "yellow", "
```

Out[ ]:
```
<Axes: xlabel='Rating color', ylabel='count'>
```



In [ ]:
```
new_df.head(2)
```

Out[ ]:

	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 rows × 22 columns

In [ ]:
```
### find the countries name that has given 0 rating
new_df[new_df["Rating color"] == "White"].groupby('Country').size().reset_index()
```

Out[ ]:

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

In [ ]:
```
###  Maximum number of zero ratings are from Indian customers
```

In [ ]:
```
new_df[new_df["Aggregate rating"] == 0.0].groupby('Country').size().reset_index()
```

Out[ ]:

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

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

Out[ ]:
```
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 [ ]:
```
new_df[["Country", "Currency"]].groupby(["Country", "Currency"]).size().reset_index().sort_va
```

Out[ ]:

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

In [ ]:
```
new_df[new_df['Has Online delivery'] =='Yes'].groupby("Country").size().reset_index()
```

Out[ ]:

	Country	0
0	India	2423
1	UAE	28

In [ ]:
```
### online deliveries are available only in India and UAE
```

In [ ]:
```
new_df['Has Online delivery'].value_counts()
```

Out[ ]:
```
No     7100
Yes    2451
Name: Has Online delivery, dtype: int64
```

In [ ]:
```
new_df[["Country", "Has Online delivery"]].groupby(["Country", "Has Online delivery"]).size()
```

Out[ ]:

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

In [ ]:
```
### Online delivery facility is available in India and UAE
```

In [ ]:
```
new_df['Country'].value_counts().head()
```

Out[ ]:
```
India             8652
United States      434
United Kingdom      80
Brazil              60
UAE                 60
Name: Country, dtype: int64
```

In [ ]:
```
new_df[new_df['Has Online delivery'] =='Yes'].groupby("Country").size().reset_index()
```

Out[ ]:

	Country	0
0	India	2423
1	UAE	28

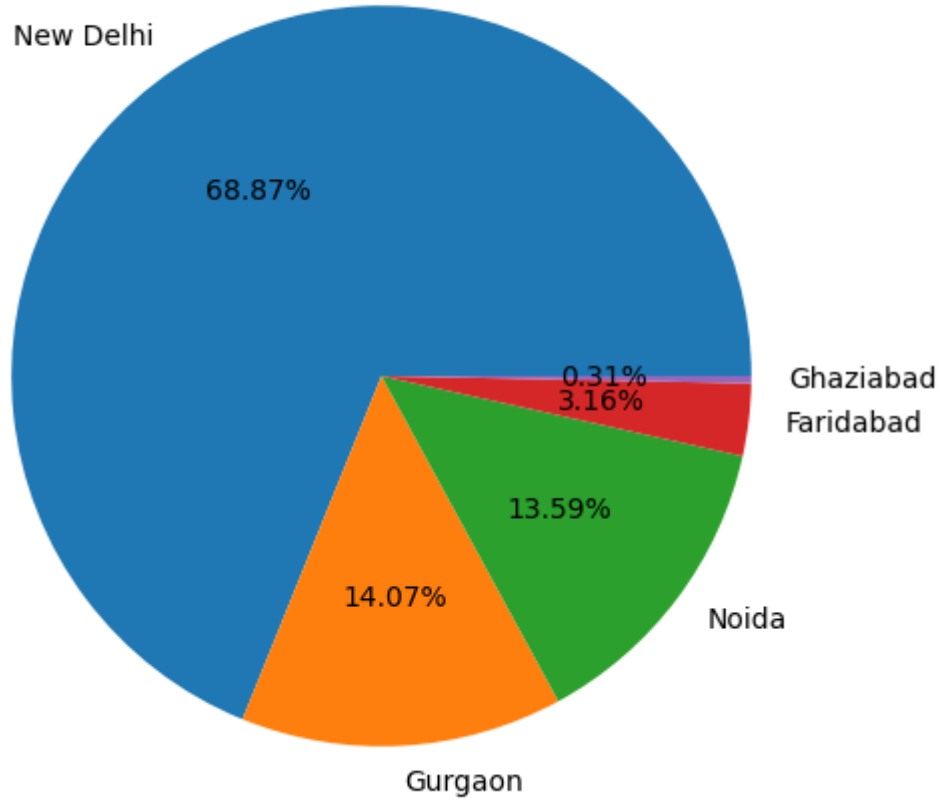
In [ ]:
```
##### Draw a pie chart to show the distributions in cities
```

In [ ]:
```
cities_values = new_df["City"].value_counts().values
```

In [ ]:
```
cities_name = new_df["City"].value_counts().index
```

In [ ]:
```
plt.pie(cities_values[:5],labels=cities_name[:5],autopct='%1.2f%%')
```

Out[ ]:
```
([<matplotlib.patches.Wedge at 0x21033b64450>,
  <matplotlib.patches.Wedge at 0x21033b141d0>,
  <matplotlib.patches.Wedge at 0x21033b66490>,
  <matplotlib.patches.Wedge at 0x21033b67b50>,
  <matplotlib.patches.Wedge at 0x21033b712d0>],
 [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%')])
```



In [ ]:
```
## Top five cities where the distribution of Zomato is most are : New Delhi, Gurgaon, Noida,
```

In [ ]:
```
cities= new_df[new_df['Country'] =='India'].groupby(["City"]).size().reset_index().sort_value
```

In [ ]:
```
cities.shape
```

Out[ ]:
```
(43, 2)
```

In [ ]:
```
cities.columns
```

Out[ ]:
```
Index(['City', 0], dtype='object')
```

In [ ]:
```
cities.rename(columns = {0:'cities_count'}, inplace = True)
```

In [ ]:
```
cities.head()
```

Out[ ]:

	City	cities_count
31	New Delhi	5473
15	Gurgaon	1118
32	Noida	1080
12	Faridabad	251
13	Ghaziabad	25

In [ ]:
```
## Top 10 cuisines
Cuisines_values = new_df["Cuisines"].value_counts().values
```

In [ ]:
```
Cuisines_values
```

Out[ ]:
```
array([936, 511, 354, ...,   1,   1,   1], dtype=int64)
```

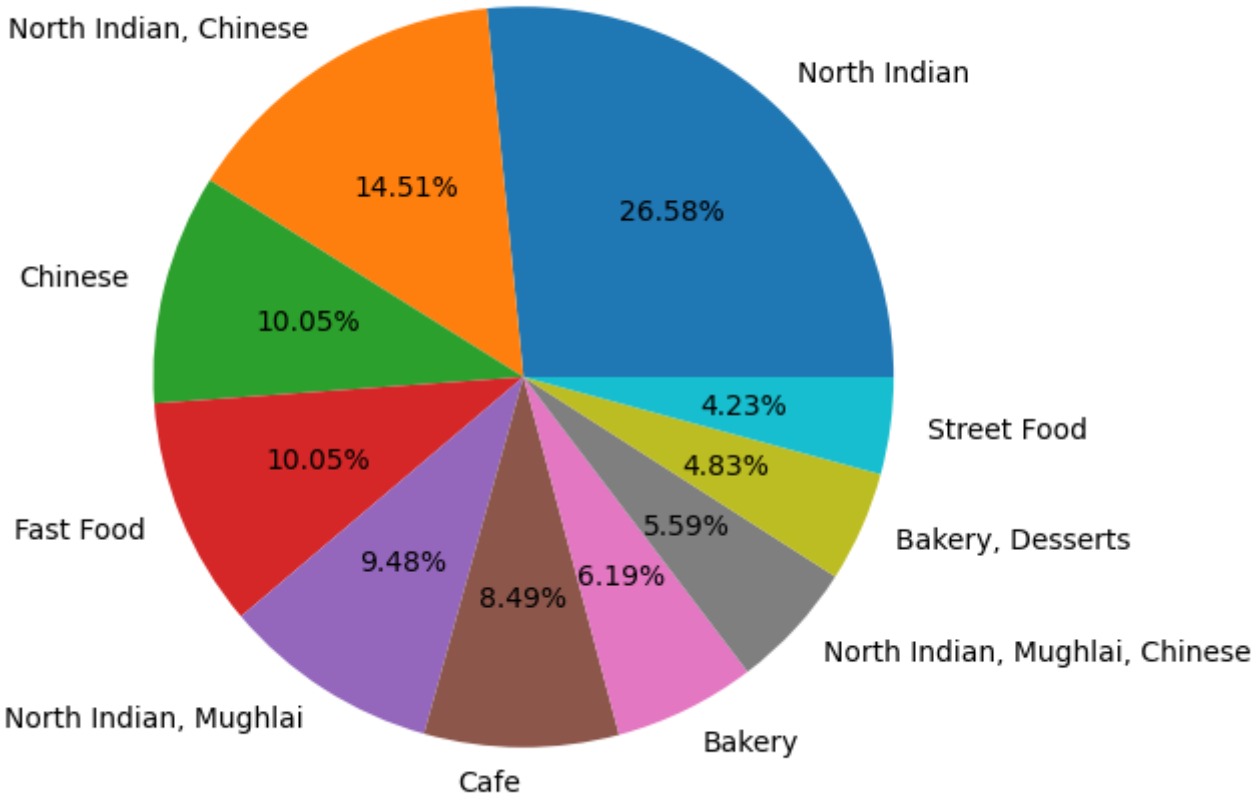
In [ ]:
```
Cuisines_names = new_df["Cuisines"].value_counts().index
```

In [ ]:
```
Cuisines_names
```

Out[ ]:
```
Index(['North Indian', 'North Indian, Chinese', 'Chinese', 'Fast Food',
       'North Indian, Mughlai', 'Cafe', 'Bakery',
       'North Indian, Mughlai, Chinese', 'Bakery, Desserts', 'Street Food',
       ...
       'Cafe, Pizza, Burger',
       'Healthy Food, Continental, Juices, Beverages, Italian, Salad, Lebanese',
       'Goan, American, Portuguese', 'South Indian, Desserts, Beverages',
       'Healthy Food, North Indian, Italian, Salad', 'Bengali, Fast Food',
       'North Indian, Rajasthani, Asian',
       'Chinese, Thai, Malaysian, Indonesian',
       'Bakery, Desserts, North Indian, Bengali, South Indian',
       'Italian, World Cuisine'],
      dtype='object', length=1825)
```

In [ ]:
```
plt.pie(Cuisines_values[:10],labels=Cuisines_names[:10],autopct='%1.2f%%')
```

Out[ ]:
```
([<matplotlib.patches.Wedge at 0x21033bc0e90>,
  <matplotlib.patches.Wedge at 0x21033bc1fd0>,
  <matplotlib.patches.Wedge at 0x21033bc3710>,
  <matplotlib.patches.Wedge at 0x21033bd4d90>,
  <matplotlib.patches.Wedge at 0x21033bd6490>,
  <matplotlib.patches.Wedge at 0x21033bd7bd0>,
  <matplotlib.patches.Wedge at 0x21033be9350>,
  <matplotlib.patches.Wedge at 0x21033beaa10>,
  <matplotlib.patches.Wedge at 0x21033bd6590>,
  <matplotlib.patches.Wedge at 0x21033bf55d0>],
 [Text(0.7383739846958008, 0.8153550507137645, 'North Indian'),
  Text(-0.5794679314239953, 0.9349956772366362, 'North Indian, Chinese'),
  Text(-1.067309479615702, 0.26617752482593154, 'Chinese'),
  Text(-1.0185984499802057, -0.4152796620326146, 'Fast Food'),
  Text(-0.5935788454809928, -0.9261015895664211, 'North Indian, Mughlai'),
  Text(-0.005887079599915552, -1.0999842463843672, 'Cafe'),
  Text(0.4842062514572988, -0.9876964645323336, 'Bakery'),
  Text(0.808736477166136, -0.7456174022251013, 'North Indian, Mughlai, Chinese'),
  Text(1.0055375294202338, -0.44597564611473206, 'Bakery, Desserts'),
  Text(1.090298995560443, -0.14576728123927227, 'Street Food')],
 [Text(0.4027494461977095, 0.4447391185711442, '26.58%'),
  Text(-0.316073417140361, 0.5099976421290743, '14.51%'),
  Text(-0.5821688070631101, 0.14518774081414446, '10.05%'),
  Text(-0.5555991545346576, -0.22651617929051704, '10.05%'),
  Text(-0.32377027935326874, -0.5051463215816842, '9.48%'),
  Text(-0.003211134327226664, -0.5999914071187457, '8.49%'),
  Text(0.26411250079489024, -0.5387435261085456, '6.19%'),
  Text(0.441128987545165, -0.40670040121369155, '5.59%'),
  Text(0.5484750160474001, -0.24325944333530836, '4.83%'),
  Text(0.5947085430329688, -0.07950942613051214, '4.23%')])
```



In [ ]:
```
# Top 10 cuisines are:
## North Indian,
## North Indian, Chinese,
## Chinese,
## Fast Food,
## North Indian, Mughlai,
## Cafe,
## Bakery,
```

```
## North Indian, Mughlai, Chinese,
## Bakery, Desserts,
## Street Food
```

OBSERVATIONS

In [ ]:
```
### Top three countries that uses zomato are India,
   ###  United States and  United Kingdom

### Observations  based on aggregate ratings
            ### when Rating is between 1.8 to 2.4----poor rating
            ### when Rating is between 2.5 to 3.4----Average rating
            ### when Rating is between 3.5 to 3.9----Good rating
            ### when Rating is between 4.0 to 4.4----Very Good rating
            ### when Rating is between 4.5 to 4.9----Excellent rating


### Maximum number of people have not given the ratings ,
#####   and average number of rating liea between 2.5 to 4.5

###  Maximum number of zero ratings are from Indian customers

### online deliveries are available only in India and UAE

## Top five cities where the distribution of Zomato is most are : New Delhi, Gurgaon,
###   Noida, Faridabad, Ghaziabad.

# Top 10 cuisines are:
            ## North Indian,
            ## North Indian, Chinese,
            ## Chinese,
            ## Fast Food,
            ## North Indian, Mughlai,
            ## Cafe,
            ## Bakery,
            ## North Indian, Mughlai, Chinese,
            ## Bakery, Desserts,
            ## Street Food
```