```
import pandas as pd
import numpy as np
import seaborn as sn
import matplotlib.pyplot as plt


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

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitu
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	14.5654
					Little				

```
df.columns
```

```
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')
```

					Shangri-La		City		

```
df.shape
```

```
(9551, 21)
```

							Floor	SM	

```
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
```

```
df.describe()
```

	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range	Aggregate rating	
count	9.551000e+03	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	955
mean	9.051128e+06	18.365616	64.126574	25.854381	1199.210763	1.804837	2.666370	15
std	8.791521e+06	56.750546	41.467058	11.007935	16121.183073	0.905609	1.516378	43
min	5.300000e+01	1.000000	-157.948486	-41.330428	0.000000	1.000000	0.000000	
25%	3.019625e+05	1.000000	77.081343	28.478713	250.000000	1.000000	2.500000	
50%	6.004089e+06	1.000000	77.191964	28.570469	400.000000	2.000000	3.200000	3
75%	1.835229e+07	1.000000	77.282006	28.642758	700.000000	2.000000	3.700000	13
max	1.850065e+07	216.000000	174.832089	55.976980	800000.000000	4.000000	4.900000	1093

```
#Data Analysis
#1.Missing Values
#2.Explore numerical Categorical Description
```

```
#3.Explore categorical variables
#4.Relationship between two variables


df.isnull().sum()
```

```
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
```

```
[features for features in df.columns if df[features].isnull().sum() > 0]
```

```
['Cuisines']
```

```
df_country = pd.read_excel('/content/Country-Code.xlsx')
df_country.head()
```

	Country Code	Country
0	1	India
1	14	Australia

```
df.columns
```

```
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')
```

```
df_final = pd.merge(df,df_country,on='Country Code',how='left')
df_final.head(2)
```

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisine
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	Frenc Japanes Desser
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	Japanes

2 rows × 22 columns

```
df.dtypes
```

```
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
dtype: object
```

```
df_final.columns
```

```
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')
```

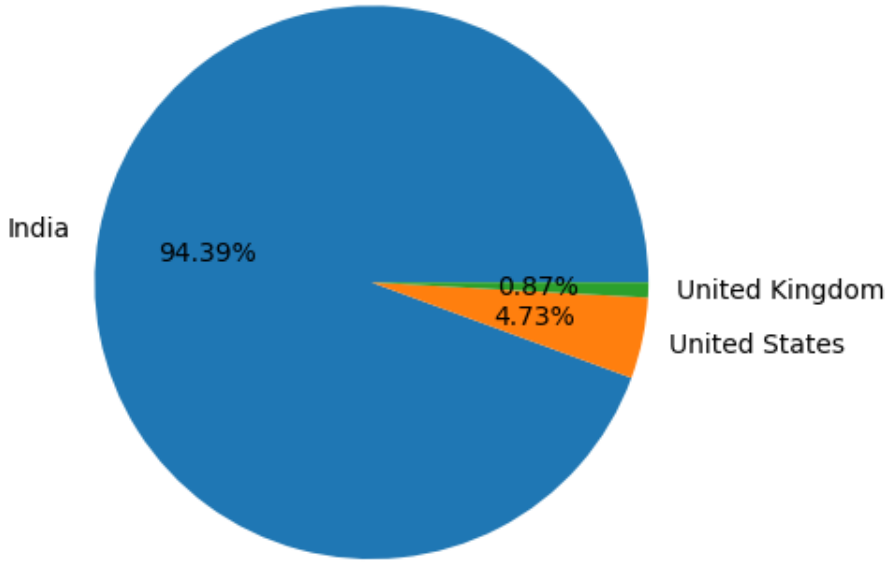
```
country_names = df_final.Country.value_counts().index
```

```
country_values = df_final.Country.value_counts().values
```

```
#Pie Chart- top 3 countries that uses Zomato


plt.pie(country_values[:3],labels=country_names[:3],autopct='%1.2f%%')
```

```
([<matplotlib.patches.Wedge at 0x799170ab4fa0>,
  <matplotlib.patches.Wedge at 0x799170ab71f0>,
  <matplotlib.patches.Wedge at 0x799170ab7910>],
 [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%')])
```



Zomato maximum record or transactions are from India then USA then UK

```
##Numerical variables
```

```
df_final.columns
```

```
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')
```

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

```
ratings
```

	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

```
#Observations
#1.When rating is between 4.5 to 4.9 -> Excellent
#2.When rating is between 4.0 to 3.9 -> 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 2.0 to 2.4 -> Poor
```

22	3.9	Yellow	Good	335

```
ratings.head()
```

	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

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

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


```
#Observation:
#1.Not Rated count is very high
#2. Max no. of ratings are between 2.5 to 3.4
```

```
sn.barplot(x="Aggregate rating",
y="Rating Count",hue='Rating color',data=ratings,
palette=['blue','red','orange','yellow',
'green','green'])
```

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


```
#Count Plot
sn.countplot(x='Rating color',data=ratings,palette=['blue','red','orange','yellow',
'green','green'])
```

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

```
#find country names that has given 0 rating
```

```
df_final.groupby(['Aggregate rating','Country']).size().reset_index().head(5)
```

	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: Max no. of 0 ratings are from Indian customers

```
df_final[['Country','Currency']].groupby(['Country','Currency']).size().reset_index()
```

	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

```
#Which countries have online delivery option
```

```
df_final[['Has Online delivery','Country']].groupby(['Has Online delivery','Country']).size().reset_index()
```

	Has Online delivery	Country	0
0	No	Australia	24
1	No	Brazil	60

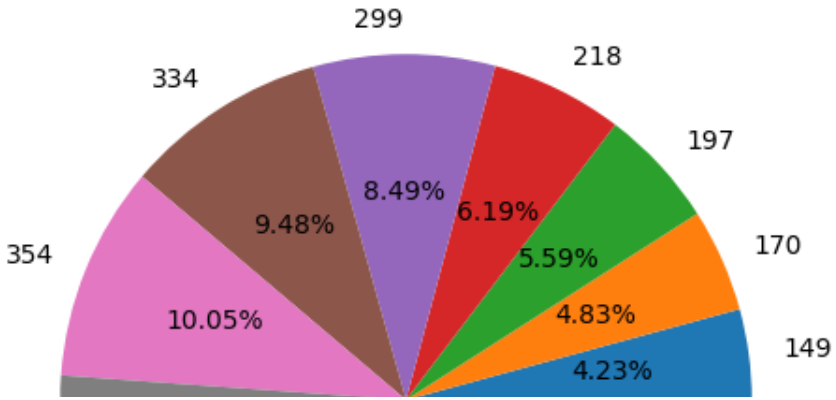
```
#Find top 10 cuisines
top10_cuisines = df_final['Cuisines'].value_counts()[:10].sort_values(ascending=True)
top10_cuisines
```

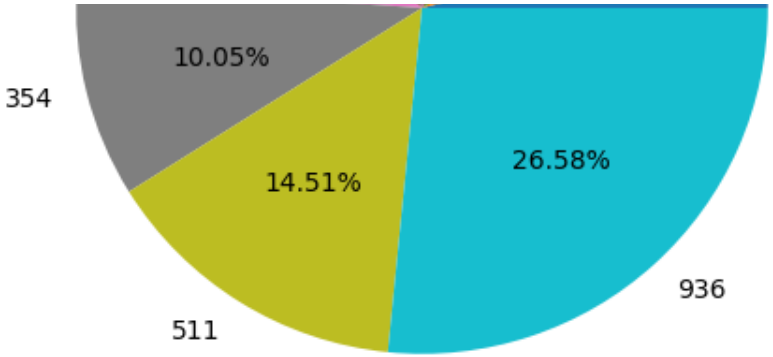
```
Street Food                       149
Bakery, Desserts                  170
North Indian, Mughlai, Chinese    197
Bakery                            218
Cafe                              299
North Indian, Mughlai             334
Chinese                           354
Fast Food                         354
North Indian, Chinese             511
North Indian                      936
Name: Cuisines, dtype: int64
```

11	No	Turkey	34

```
plt.pie(top10_cuisines[:10],labels=top10_cuisines[:10],autopct='%1.2f%%')
```

```
([<matplotlib.patches.Wedge at 0x799169ab1930>,
  <matplotlib.patches.Wedge at 0x799169ab26b0>,
  <matplotlib.patches.Wedge at 0x799169ab3e80>,
  <matplotlib.patches.Wedge at 0x799169ae4550>,
  <matplotlib.patches.Wedge at 0x799169ae4be0>,
  <matplotlib.patches.Wedge at 0x799169ae5270>,
  <matplotlib.patches.Wedge at 0x799169ae5900>,
  <matplotlib.patches.Wedge at 0x799169ae5f90>,
  <matplotlib.patches.Wedge at 0x799169ae6620>,
  <matplotlib.patches.Wedge at 0x799169ae6cb0>],
 [Text(1.0902989921485133, 0.14576730675956207, '149'),
  Text(1.0055375189814202, 0.4459756696510365, '170'),
  Text(0.8087364597137011, 0.7456174211549441, '197'),
  Text(0.48420622833859467, 0.9876964758659984, '218'),
  Text(-0.00588710534690472, 1.0999842462465699, '299'),
  Text(-0.5935788671579647, 0.9261015756727052, '334'),
  Text(-1.0185984597005273, 0.41527963819059754, '354'),
  Text(-1.0673094733853672, -0.26617754980811276, '354'),
  Text(-0.5794679095388423, -0.9349956908000615, '511'),
  Text(0.738374003780563, -0.8153550334308739, '936')],
 [Text(0.5947085411719163, 0.07950944005067022, '4.23%'),
  Text(0.5484750103535018, 0.24325945617329262, '4.83%'),
  Text(0.4411289780256551, 0.4067004115390604, '5.59%'),
  Text(0.26411248818468797, 0.5387435322905445, '6.19%'),
  Text(-0.003211148371038938, 0.5999914070435836, '8.49%'),
  Text(-0.3237702911770716, 0.5051463140032937, '9.48%'),
  Text(-0.5555991598366512, 0.22651616628578045, '10.05%'),
  Text(-0.5821688036647457, -0.14518775444078877, '10.05%'),
  Text(-0.31607340520300486, -0.5099976495273062, '14.51%'),
  Text(0.4027494566075797, -0.44473910914411297, '26.58%')])
```




10.05%
354
14.51%
26.58%
936
511