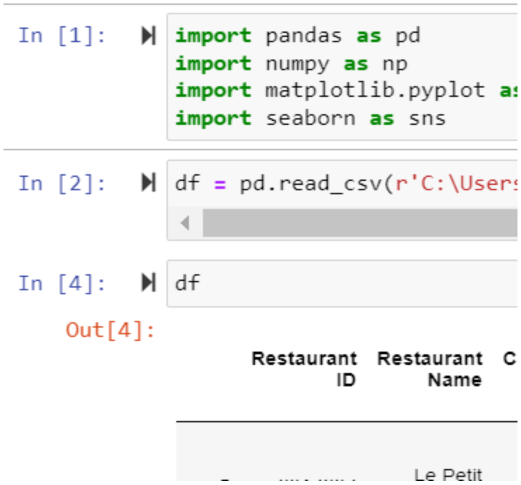
Zomato_Data_analysis(Python)

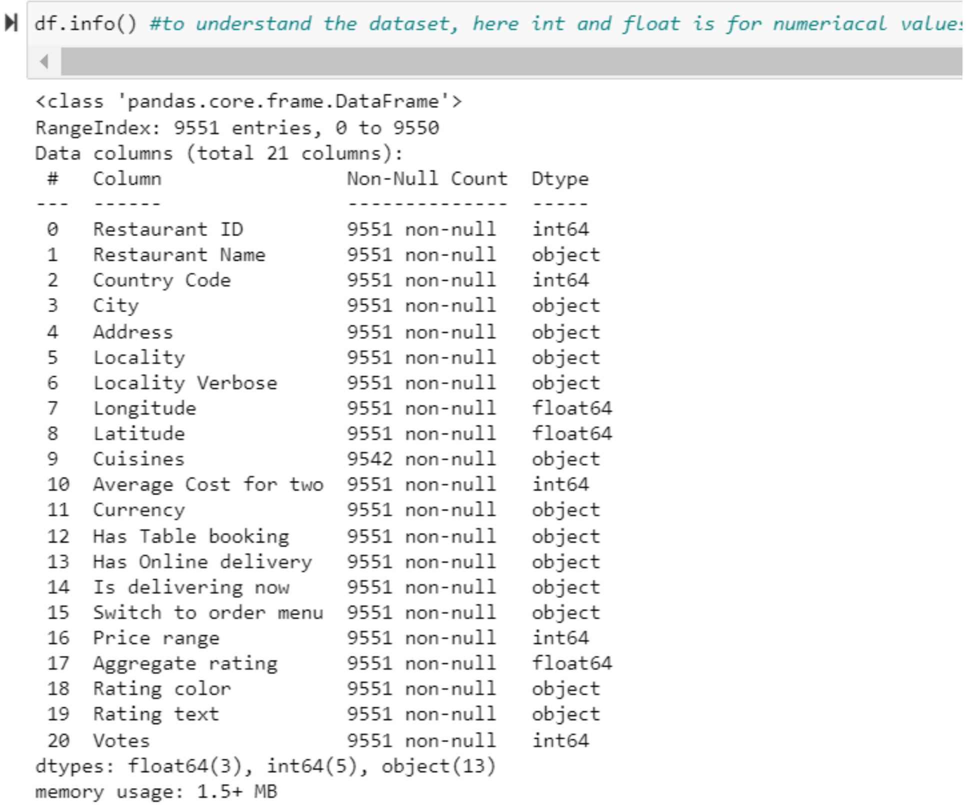
Conducted Exploratory Data Analysis (EDA) on the Zomato dataset encompassing sales data from the US, India, and the UK. The goal is to derive actionable insights into geographical sales patterns, user preferences, and restaurant characteristics, enabling data-driven recommendations for optimizing Zomato's services and market strategies in these regions.

These four libraries are important for Data analysis.

- import pandas as pd
- import numpy as np
- import matplotlib.pyplot as plt
- import seaborn as sns

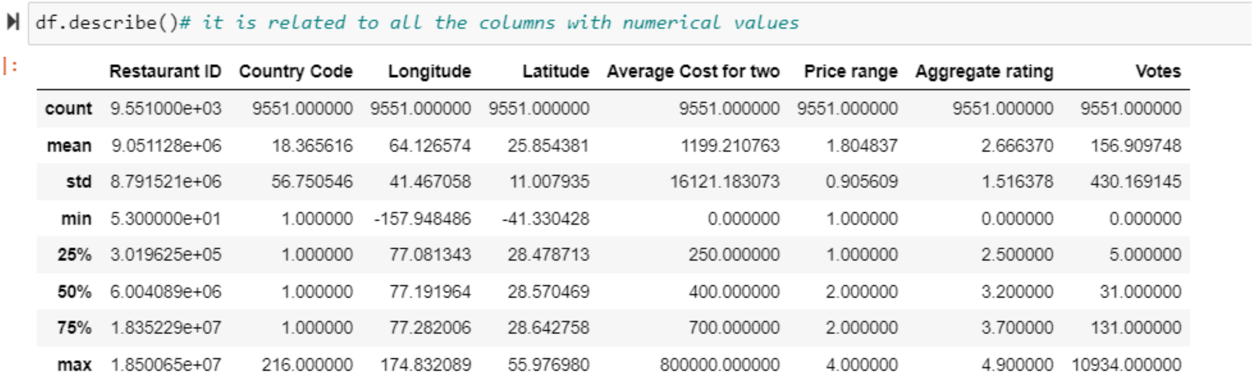


```
In [1]:   import pandas as pd
          import numpy as np
          import matplotlib.pyplot as
          import seaborn as sns

In [2]:   df = pd.read_csv(r'C:\Users

In [4]:   df

Out[4]:
           Restaurant   Restaurant   C
                   ID         Name

                          Le Petit
```



```
df.info() #to understand the dataset, here int and float is for numeriacal value:
```

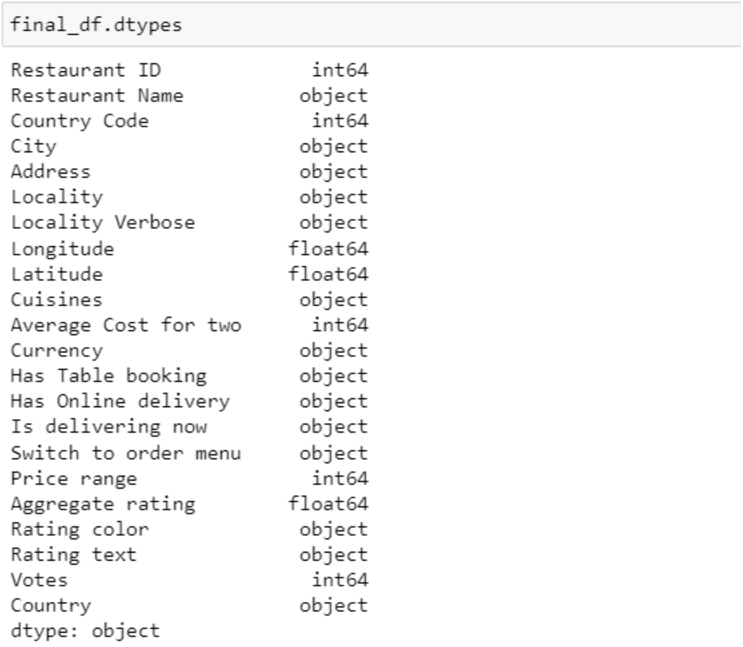
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9551 entries, 0 to 9550
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   Restaurant ID         9551 non-null   int64  
 1   Restaurant Name       9551 non-null   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()# it is related to all the columns with numerical values
```

]:

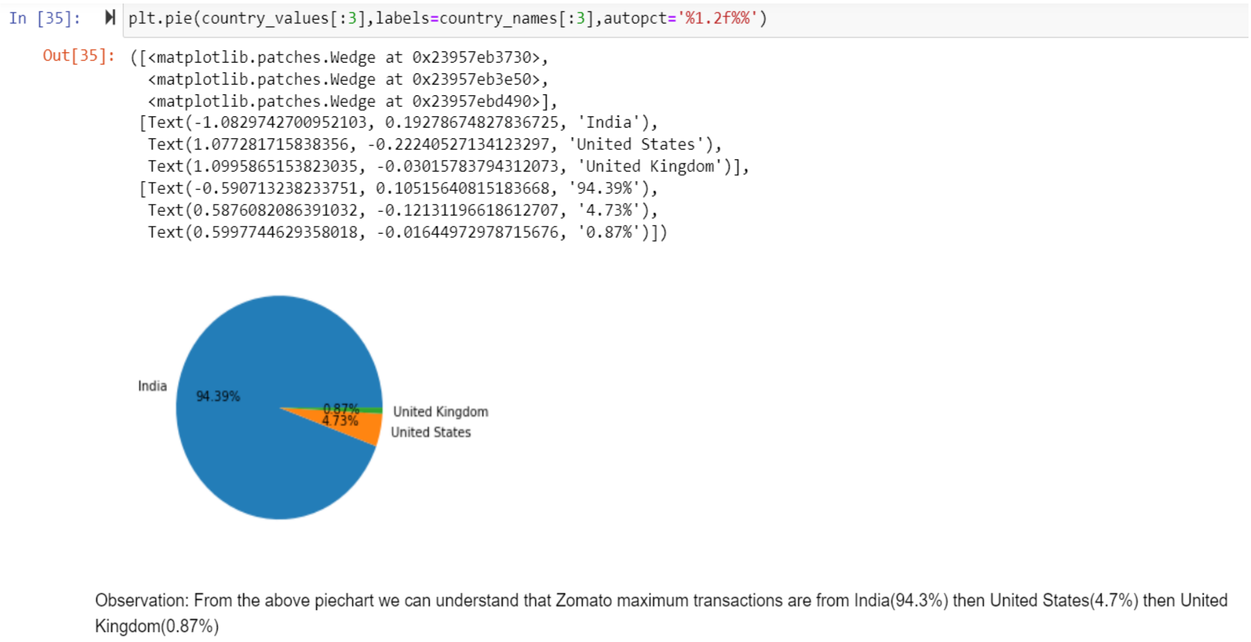
	Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range	Aggregate rating	Votes
count	9.551000e+03	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000	9551.000000
mean	9.051128e+06	18.365616	64.126574	25.854381	1199.210763	1.804837	2.666370	156.909748
std	8.791521e+06	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



```
final_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
Country                  object
dtype: object
```

Visualizations: -



```
In [35]:  plt.pie(country_values[:3],labels=country_names[:3],autopct='%1.2f%%')

Out[35]: ([<matplotlib.patches.Wedge at 0x23957eb3730>,
           <matplotlib.patches.Wedge at 0x23957eb3e50>,
           <matplotlib.patches.Wedge at 0x23957ebd490>],
          [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%')])
```

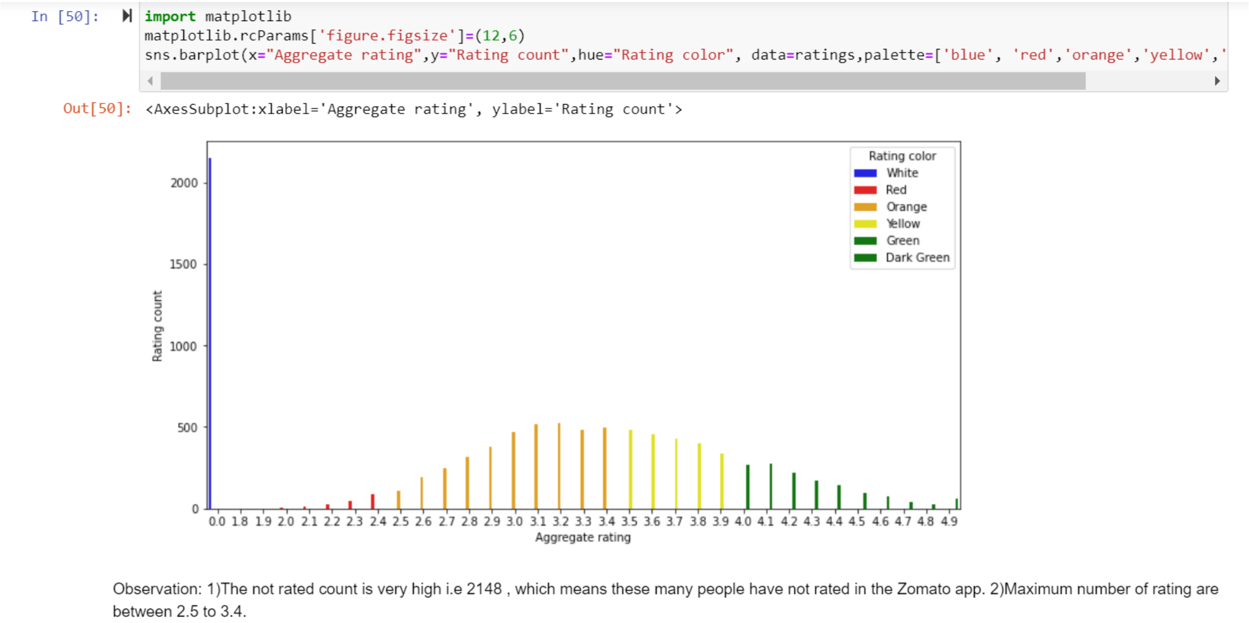
Observation: From the above piechart we can understand that Zomato maximum transactions are from India(94.3%) then United States(4.7%) then United Kingdom(0.87%)

Observation:

From the above pie chart we can understand that Zomato maximum transactions are from India (94.3%) then United States (4.7%) then United Kingdom (0.87%).

Observation:

1) When the rating is between 4.5 to 4.9, it is Excellent.
2) When the rating is between 4.0 to 4.4, it is Very Good.
3) When the rating is between 3.5 to 3.9, it is Good.
4) When the rating is between 2.5 to 3.4, it is Average.
5) when the rating is between 1.8 to 2.4, it is poor.



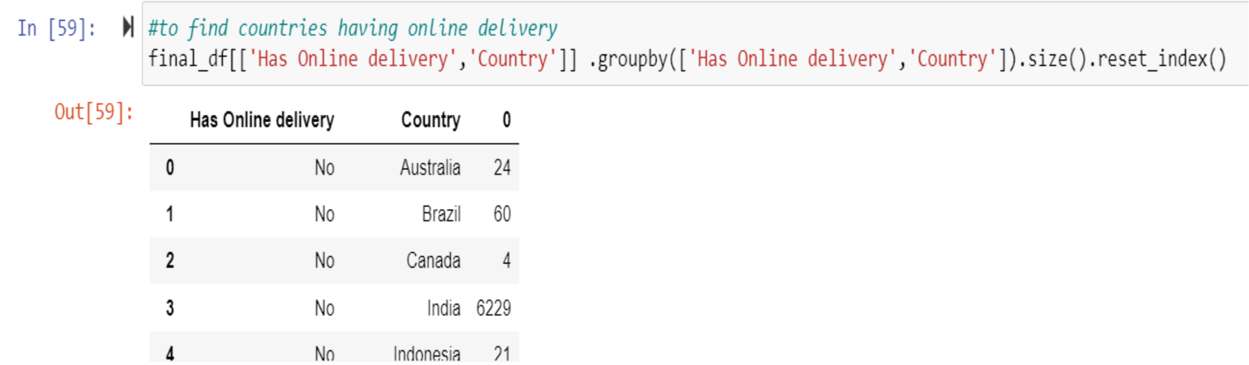
```
In [50]:  import matplotlib
          matplotlib.rcParams['figure.figsize']=(12,6)
          sns.barplot(x="Aggregate rating",y="Rating count",hue="Rating color", data=ratings,palette=['blue', 'red','orange','yellow','

Out[50]: <AxesSubplot:xlabel='Aggregate rating', ylabel='Rating count'>
```

Observation: 1)The not rated count is very high i.e 2148 , which means these many people have not rated in the Zomato app. 2)Maximum number of rating are between 2.5 to 3.4.

Observation:

1)The not rated count is very high i.e 2148, which means these many people have not rated in the Zomato app.

2)Maximum number of ratings are between 2.5 to 3.4.

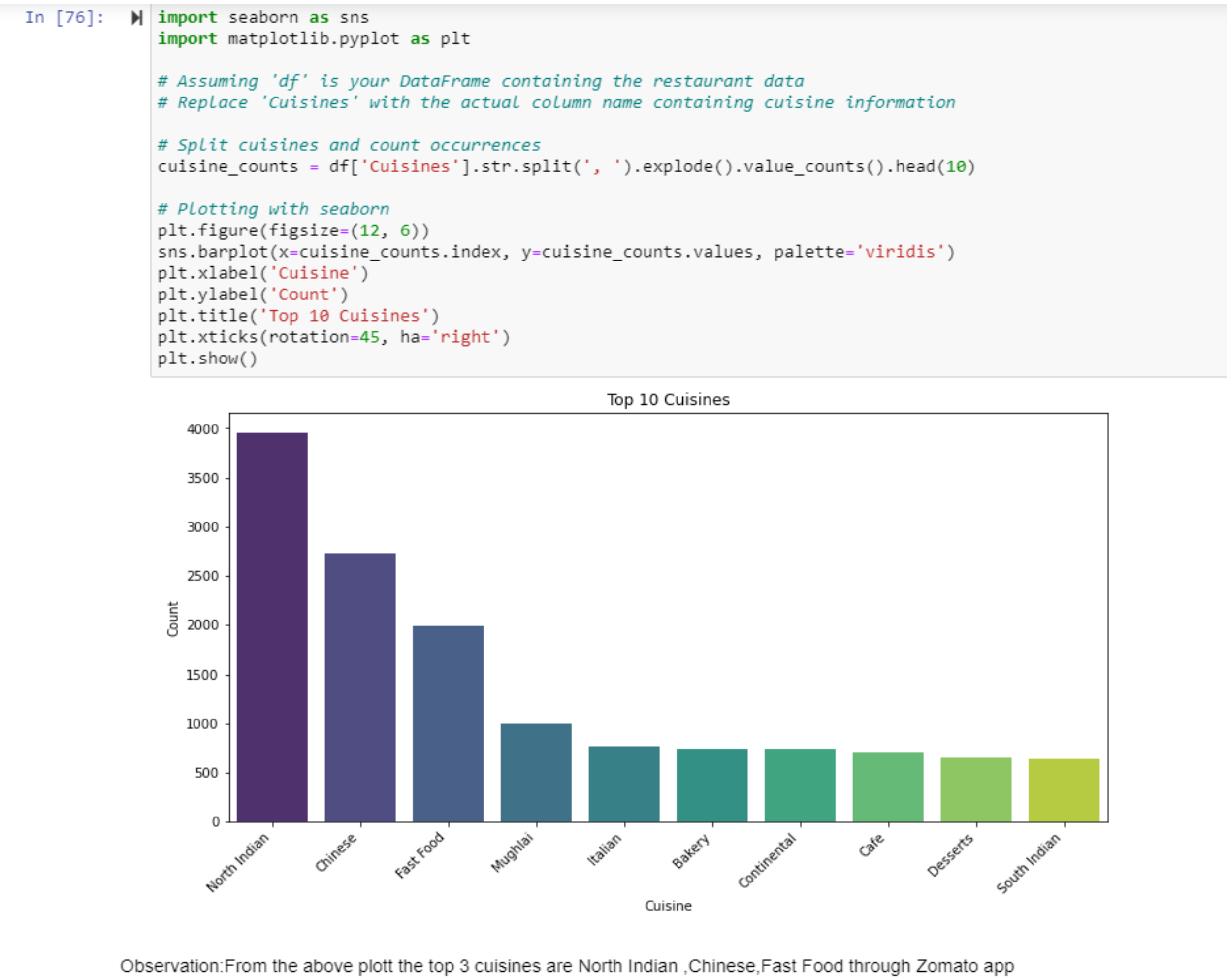


```
In [59]:  #to find countries having online delivery
          final_df[['Has Online delivery','Country']] .groupby(['Has Online delivery','Country']).size().reset_index()

Out[59]:
```

	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

Observation:

The online deliveries are in India and UAE.



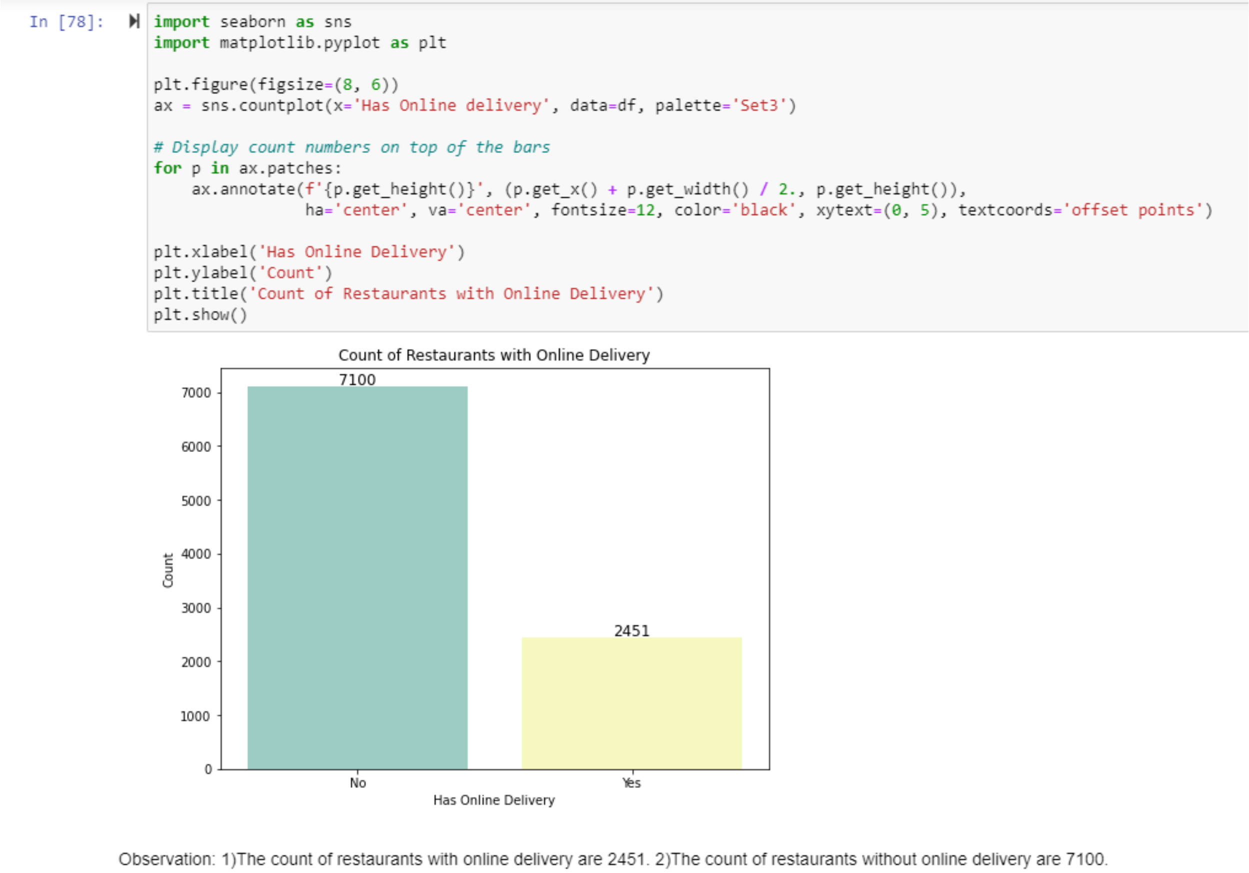
In [76]:

```
import seaborn as sns
import matplotlib.pyplot as plt

# Assuming 'df' is your DataFrame containing the restaurant data
# Replace 'Cuisines' with the actual column name containing cuisine information

# Split cuisines and count occurrences
cuisine_counts = df['Cuisines'].str.split(', ').explode().value_counts().head(10)

# Plotting with seaborn
plt.figure(figsize=(12, 6))
sns.barplot(x=cuisine_counts.index, y=cuisine_counts.values, palette='viridis')
plt.xlabel('Cuisine')
plt.ylabel('Count')
plt.title('Top 10 Cuisines')
plt.xticks(rotation=45, ha='right')
plt.show()
```

Observation:From the above plott the top 3 cuisines are North Indian ,Chinese,Fast Food through Zomato app

Observation:

From the above plot the top 3 cuisines are North Indian, Chinese, Fast Food through Zomato app.



In [78]:

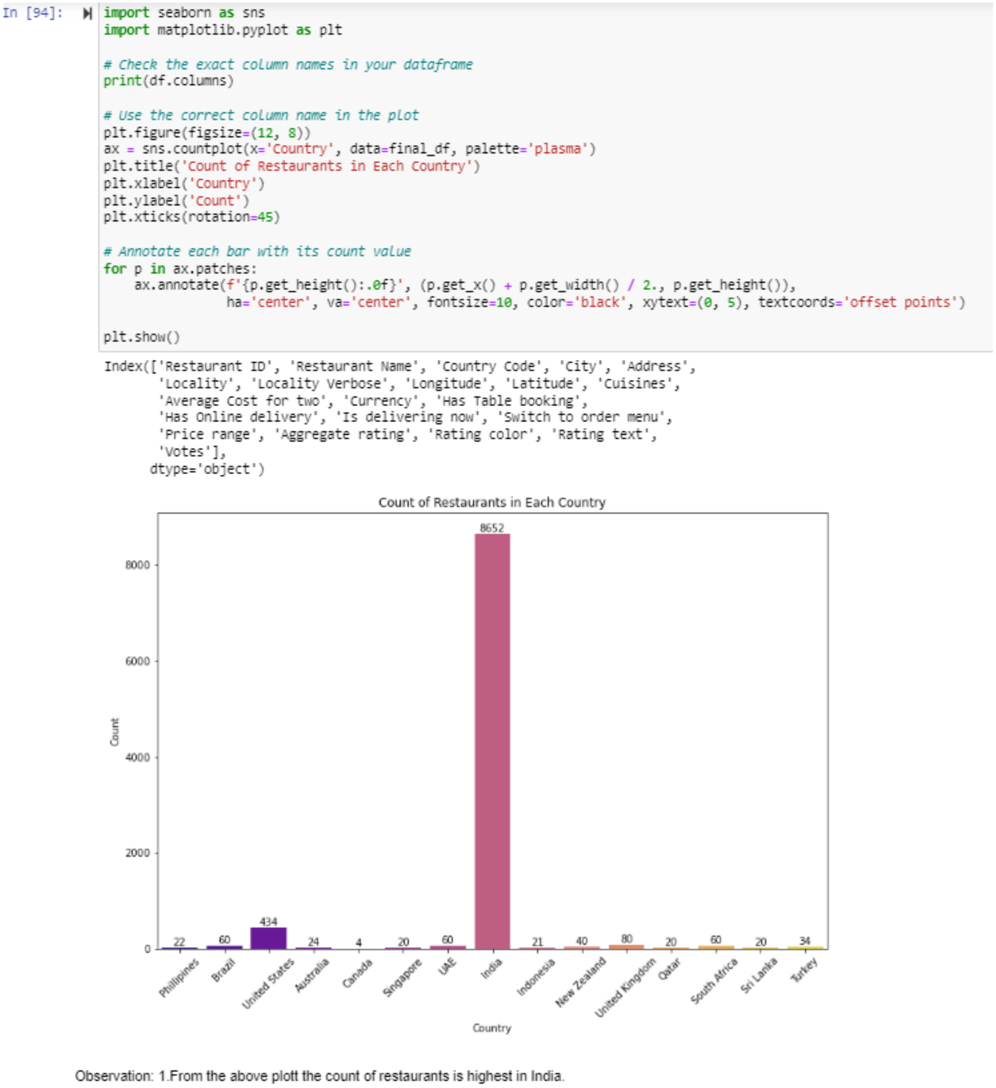
```
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 6))
ax = sns.countplot(x='Has Online delivery', data=df, palette='Set3')

# Display count numbers on top of the bars
for p in ax.patches:
    ax.annotate(f'{p.get_height()}', (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', fontsize=12, color='black', xytext=(0, 5), textcoords='offset points')

plt.xlabel('Has Online Delivery')
plt.ylabel('Count')
plt.title('Count of Restaurants with Online Delivery')
plt.show()
```

Observation: 1)The count of restaurants with online delivery are 2451. 2)The count of restaurants without online delivery are 7100.

Observation:

1)The number of restaurants with online delivery are 2451.

2)The number of restaurants without online delivery are 7100.



In [94]:

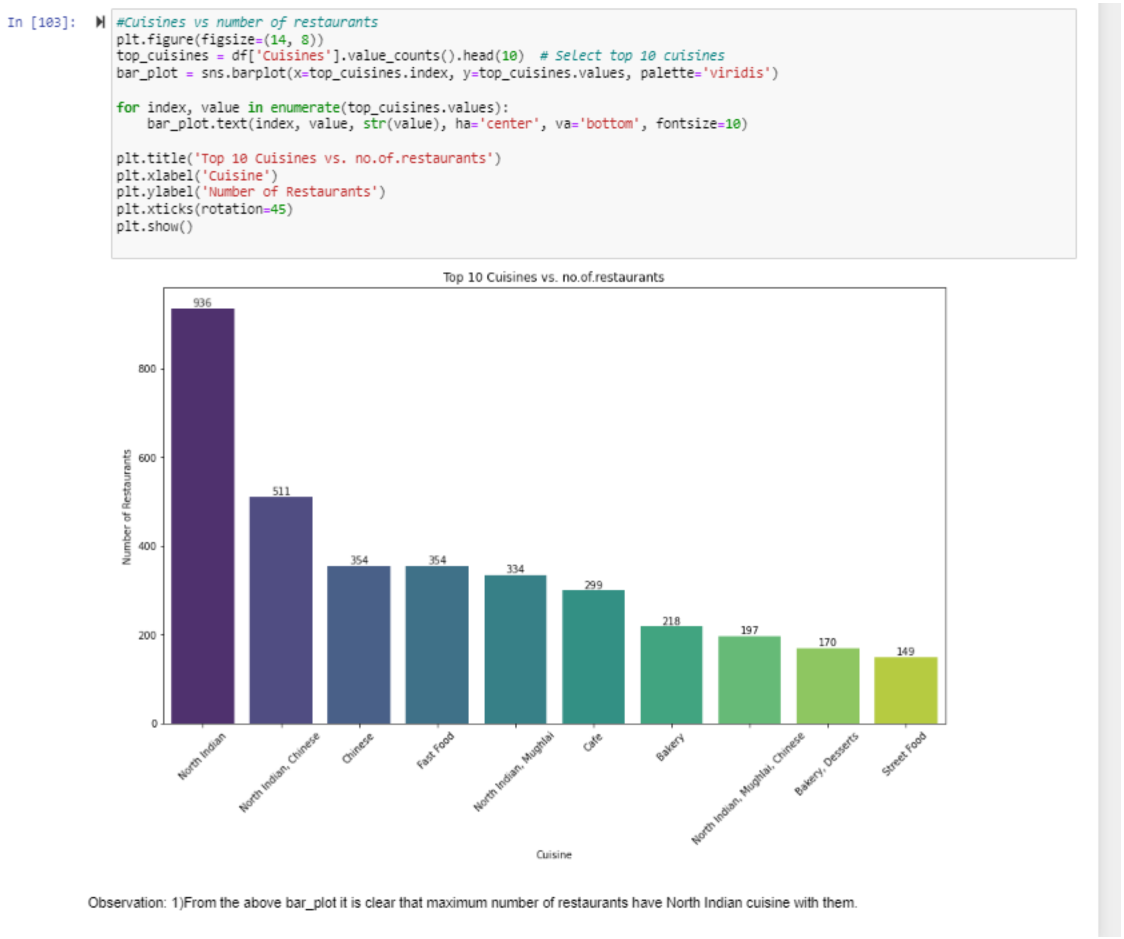
```
import seaborn as sns
import matplotlib.pyplot as plt

# Check the exact column names in your dataframe
print(df.columns)

# Use the correct column name in the plot
plt.figure(figsize=(12, 8))
ax = sns.countplot(x='Country', data=final_df, palette='plasma')
plt.title('Count of Restaurants in Each Country')
plt.xlabel('Country')
plt.ylabel('Count')
plt.xticks(rotation=45)

# Annotate each bar with its count value
for p in ax.patches:
    ax.annotate(f'{p.get_height():.0f}', (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', fontsize=10, color='black', xytext=(0, 5), textcoords='offset points')

plt.show()
```

```
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')
```

Observation: 1.From the above plott the count of restaurants is highest in India.

Observation:

From the above plot the count of restaurants is highest in India.



```
In [103]:  #Cuisines vs number of restaurants
           plt.figure(figsize=(14, 8))
           top_cuisines = df['Cuisines'].value_counts().head(10)  # Select top 10 cuisines
           bar_plot = sns.barplot(x=top_cuisines.index, y=top_cuisines.values, palette='viridis')

           for index, value in enumerate(top_cuisines.values):
               bar_plot.text(index, value, str(value), ha='center', va='bottom', fontsize=10)

           plt.title('Top 10 Cuisines vs. no.of.restaurants')
           plt.xlabel('Cuisine')
           plt.ylabel('Number of Restaurants')
           plt.xticks(rotation=45)
           plt.show()
```

Observation: 1)From the above bar_plot it is clear that maximum number of restaurants have North Indian cuisine with them.

Observation:

From the above bar plot it is clear that maximum number of restaurants have North Indian cuisine with them.

Insights: -

1) **Geographical Impact:**

   - Zomato dominates in India, with significant transactions, followed by the United States and the United Kingdom.

2) **User Ratings:**

   - The majority of user ratings fall between 2.5 to 3.4 i.e the Average and Very Good categories, indicating positive experiences.

   - Notably high instances of not-rated entries suggest users opting not to provide feedback.

3) **Top Cuisines:**

   - North Indian, Chinese, and Fast Food are the top three cuisines preferred by Zomato users.

4) **Online Delivery Significance:**

   - A substantial number of restaurants (2451) offer online delivery services, emphasizing its importance.

5) **Regional Restaurant number:**

   - India has the highest count of restaurants on Zomato, aligning with the platform's strong presence in the country.

Recommendations:

1. Strengthen Zomato's presence in India and tailor services to local preferences.

2. Improve overall user satisfaction by addressing issues in the 2.5 to 3.4 rating range. This can be achieved by encouraging detailed user feedback.

3. Partner with more restaurants offering North Indian, Chinese, and Fast Food. Introduce promotions for these popular cuisines.

4. Enhance the online delivery experience, expand partnerships with restaurants, and invest in delivery process efficiency.

5. Continue strengthening presence in India and explore growth opportunities in the United States and the United Kingdom. Tailor marketing to regional preferences.