Task: Table Booking and Online Delivery

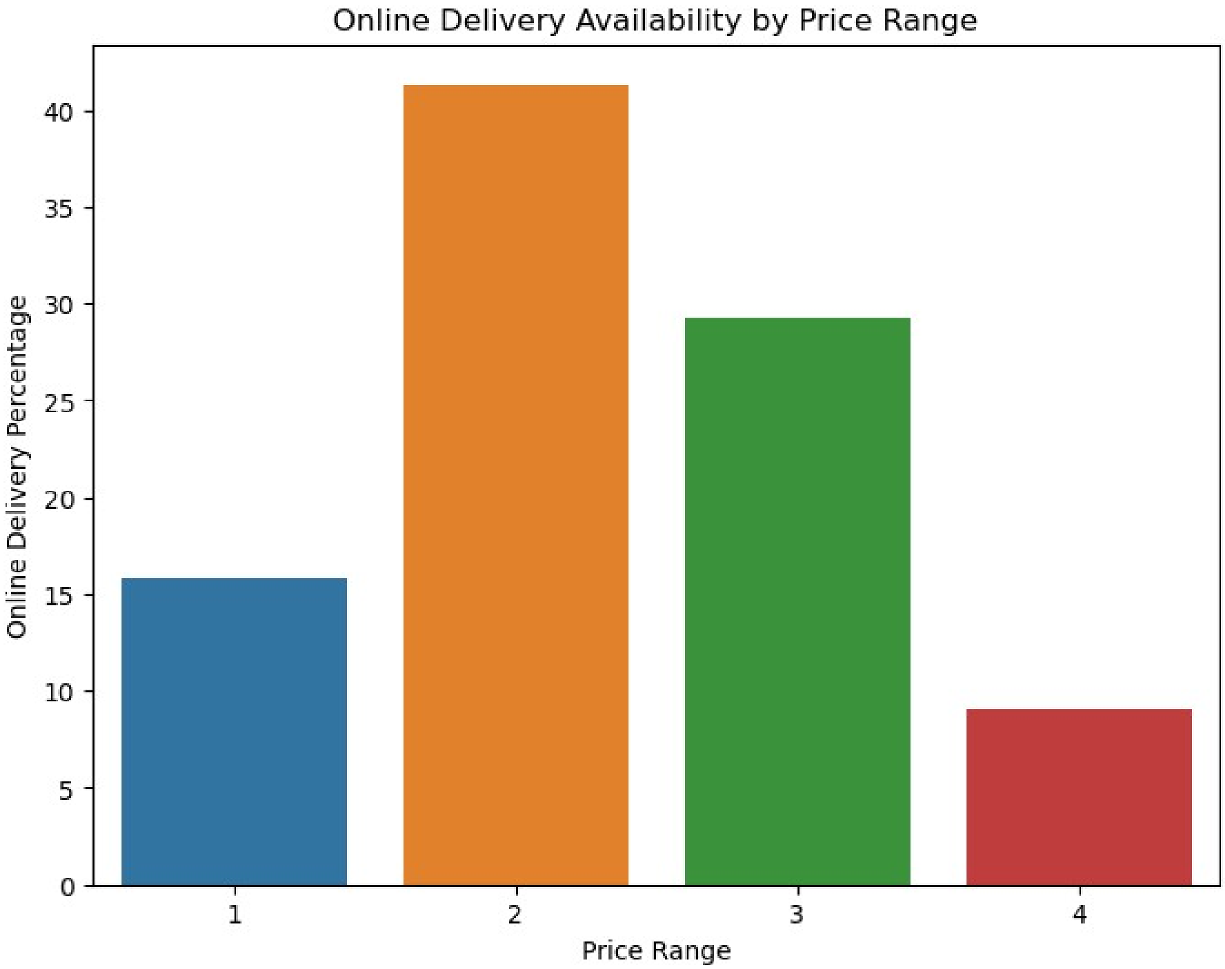
```python
#import necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import scipy.stats as stats

import ipywidgets as widgets
from bs4 import BeautifulSoup
import requests
import statsmodels.api as sm

#Load the dataset
df = pd.read_csv(r"C:\Users\hp\Desktop\EDA\Dataset.csv")

#Convert 'Has Table booking' and 'Has Online delivery' to numeric
df['Has Table booking'] = df['Has Table booking'].map({'Yes': 1, 'No': 0})
df['Has Online delivery'] = df['Has Online delivery'].map({'Yes': 1, 'No': 0})

#Calculate the percentage of restaurants offering table booking
table_booking_percentage = (df['Has Table booking'].sum() / len(df)) * 100

#Calculate the percentage of restaurants offering online delivery
online_delivery_percentage = (df['Has Online delivery'].sum() / len(df)) * 100

print(f"Percentage of restaurants with Table Booking: {table_booking_percentage}%")
print(f"Percentage of restaurants with Online Delivery: {online_delivery_percentage}%")
```

```
Percentage of restaurants with Table Booking: 12.135820582687067%
Percentage of restaurants with Online Delivery: 25.686438901697755%
```

```python
#Calculate the average rating for restaurants with and without table booking
avg_rating_with_table_booking = df[df['Has Table booking'] == 'Yes']['Aggregate rating'].mean()
avg_rating_without_table_booking = df[df['Has Table booking'] == 'No']['Aggregate rating'].mean()

print(f"Average rating with Table Booking: {avg_rating_with_table_booking}")
print(f"Average rating without Table Booking:
{avg_rating_without_table_booking}")
```

```
Average rating with Table Booking: nan
Average rating without Table Booking: nan
```

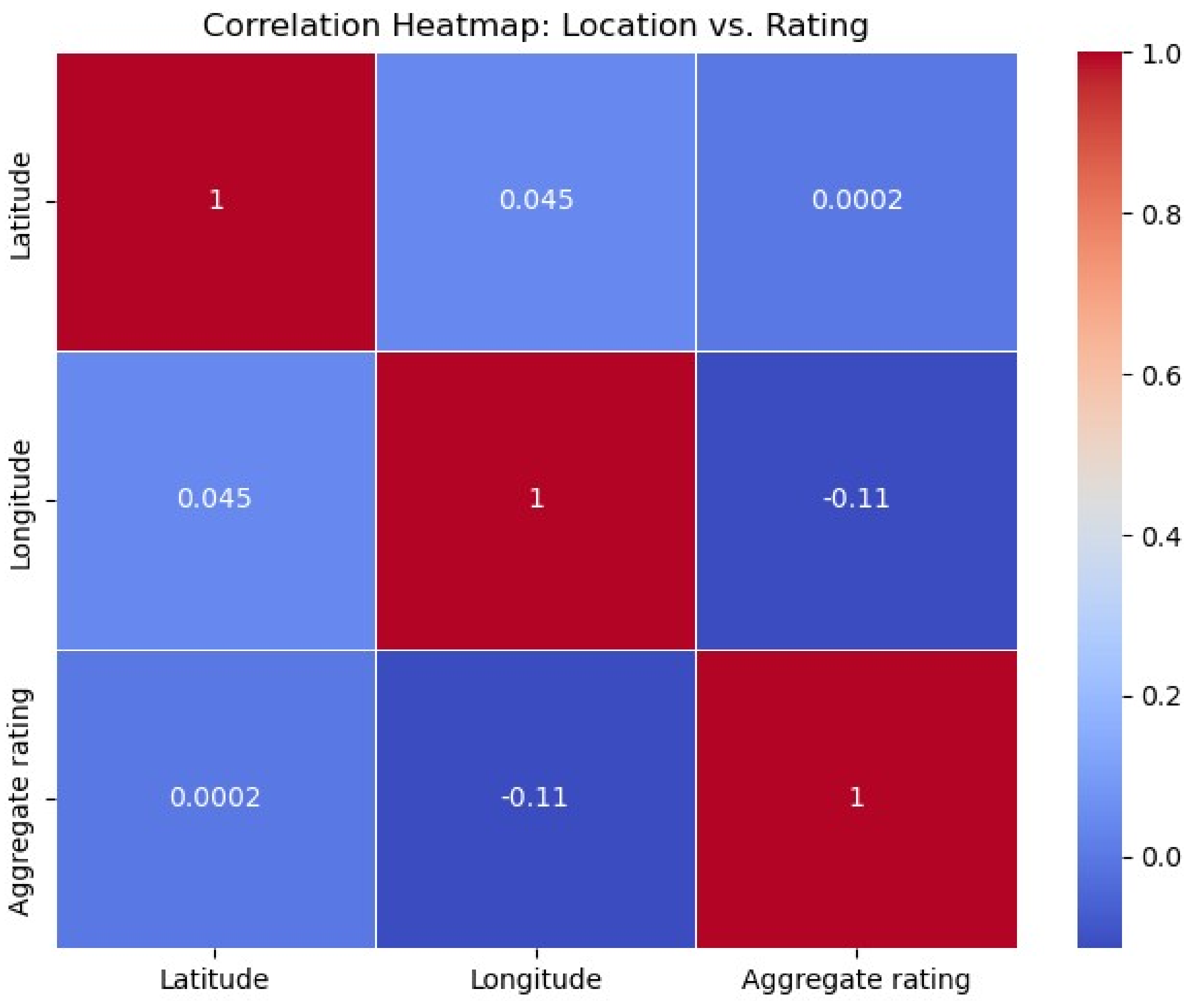
```
#Group restaurants by price range and calculate the percentage with
online delivery
price_ranges = df['Price range'].unique()
delivery_by_price_range = []

for price_range in price_ranges:
    subset = df[df['Price range'] == price_range]
    delivery_percentage = (subset['Has Online delivery'].sum() /
len(subset)) * 100
    delivery_by_price_range.append({'Price Range': price_range,
'Online Delivery Percentage': delivery_percentage})

# Create a Dataframe for Visualization
delivery_df = pd.DataFrame(delivery_by_price_range)

#Visualize online delivery availability by price range
plt.figure(figsize=(8, 6))
sns.barplot(data=delivery_df, x='Price Range', y='Online Delivery
Percentage')
plt.xlabel('Price Range')
plt.ylabel('Online Delivery Percentage')
plt.title('Online Delivery Availability by Price Range')
plt.show()
```



```
# Calculate the correlation between Latitude, Longitude, and Aggregate
rating
correlation = df[['Latitude', 'Longitude', 'Aggregate rating']].corr()
print("Correlation between Lacation and Rating:")
print(correlation)
```

```
Correlation between Lacation and Rating:
                  Latitude  Longitude  Aggregate rating
Latitude          1.000000   0.045415          0.000197
Longitude         0.045415   1.000000         -0.114733
Aggregate rating  0.000197  -0.114733          1.000000
```

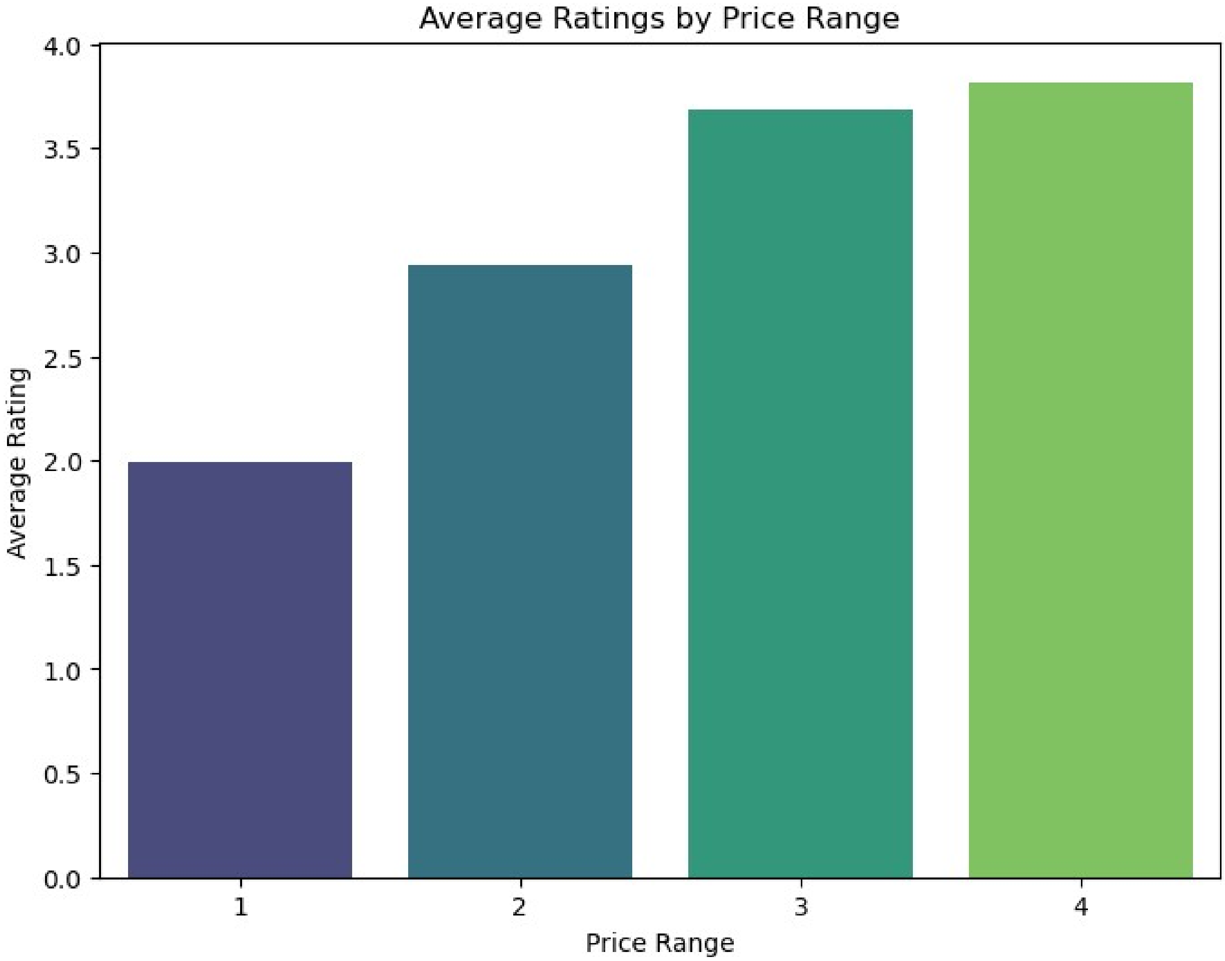
```
# Visualize the correlation
plt.figure(figsize=(8, 6))
sns.heatmap(correlation, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Heatmap: Location vs. Rating')
plt.show()
```



Task : Price Range Analysis

```
#Determining the Most Common Price range
most_common_price_range = df['Price range'].mode().values[0]
print(f"The most common price range among all restaurants is:
{most_common_price_range}")

The most common price range among all restaurants is: 1

#Calculate the Average Rating For Each Price Range
average_ratings_by_price = df.groupby('Price range')['Aggregate
rating'].mean().reset_index()
print("Average Ratings by Price Range:")
print(average_ratings_by_price)

Average Ratings by Price Range:
   Price range  Aggregate rating
0            1          1.997476
```

```
1             2           2.941054
2             3           3.682633
3             4           3.817918

#Find the price range with the highest average rating
highest_ratings_price_range =
average_ratings_by_price.loc[average_ratings_by_price['Aggregate
rating'].idxmax()]['Price range']

#Create a bar plot to visualize average ratings by price range
plt.figure(figsize=(8, 6))
sns.barplot(data=average_ratings_by_price, x='Price range',
y='Aggregate rating', palette='viridis')
plt.xlabel('Price Range')
plt.ylabel('Average Rating')
plt.title('Average Ratings by Price Range')
plt.xticks(rotation=0)
plt.show()

print(f"The color representing the highest average rating is
associated with the price range: {highest_ratings_price_range}")
```



```
The color representing the highest average rating is associated with
the price range: 4.0
```

Task : Feature Engineering

```
# Extract the length of the restaurant name
df['Restaurant Name Length'] = df['Restaurant Name'].apply(len)

#Extract the length of the restaurant address
df['Address Length'] = df['Address'].apply(len)

#Display the first few rows of the dataframe to see the new columns
print(df[['Restaurant Name', 'Restaurant Name Length', 'Address',
'Address Length']].head())
```

```
          Restaurant Name  Restaurant Name Length  \
0        Le Petit Souffle                      16
1        Izakaya Kikufuji                      16
```

```
2  Heat - Edsa Shangri-La                            22
3                    Ooma                             4
4             Sambo Kojin                            11
```

```
                                             Address  Address Length
0  Third Floor, Century City Mall, Kalayaan Avenu...              71
1  Little Tokyo, 2277 Chino Roces Avenue, Legaspi...              67
2  Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...              56
3  Third Floor, Mega Fashion Hall, SM Megamall, O...              70
4  Third Floor, Mega Atrium, SM Megamall, Ortigas...              64
```

```python
# Encode 'Has Table Booking' as 1 for 'Yes' and 0 for 'No'
df['Has Table Booking'] = df['Has Table booking'].map({'Yes':1,
'No':0})

# Encode 'Has Online Delivery' as 1 for 'Yes' and 0 for 'No'
df['Has Online Delivery'] = df['Has Online delivery'].map({'Yes':1,
'No':0})

print(df[['Has Table Booking', 'Has Online Delivery']].head())
```

```
   Has Table Booking  Has Online Delivery
0                NaN                  NaN
1                NaN                  NaN
2                NaN                  NaN
3                NaN                  NaN
4                NaN                  NaN
```