Import Required Libraries

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

# Configure plots
%matplotlib inline
sns.set(style="whitegrid")
```

Upload and Load Dataset

```
# Upload the dataset file manually in Colab
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving Dataset .csv to Dataset .csv
```

```
# Load the CSV file (change the name if your file is different)
df = pd.read_csv('Dataset .csv')

# Preview the data
df.head()
```

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Currency	Has Table booking	Has Online delivery	Is delivering now	Switch to order menu	Price range	Aggregate rating	Rating color
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	...	Botswana Pula(P)	Yes	No	No	No	3	4.8	Dark Green
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	...	Botswana Pula(P)	Yes	No	No	No	3	4.5	Dark Green
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	14.581404	Seafood, Asian, Filipino, Indian	...	Botswana Pula(P)	Yes	No	No	No	4	4.4	Green
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	14.585318	Japanese, Sushi	...	Botswana Pula(P)	No	No	No	No	4	4.9	Dark Green
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	14.584450	Japanese, Korean	...	Botswana Pula(P)	Yes	No	No	No	4	4.8	Dark Green

5 rows × 21 columns

Basic Information and Structure

```
# Check the structure of the dataset
df.info()

# Check data types and null values
df.isnull().sum()

# Summary statistics
```

```
df.describe(include='all')
```

```
<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
```

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Currency	Has Table booking	Has Online delivery	Is delivering now	Switch to order menu	Price range	Aggregat ratin
count	9.551000e+03	9551	9551.000000	9551	9551	9551	9551	9551.000000	9551.000000	9542	...	9551	9551	9551	9551	9551	9551.000000	9551.00000
unique	NaN	7446	NaN	141	8918	1208	1265	NaN	NaN	1825	...	12	2	2	2	1	NaN	Nal
top	NaN	Cafe Coffee Day	NaN	New Delhi	Dilli Haat, INA, New Delhi	Connaught Place	Connaught Place, New Delhi	NaN	NaN	North Indian	...	Indian Rupees(Rs.)	No	No	No	No	NaN	Nal
freq	NaN	83	NaN	5473	11	122	122	NaN	NaN	936	...	8652	8393	7100	9517	9551	NaN	Nal
mean	9.051128e+06	NaN	18.365616	NaN	NaN	NaN	NaN	64.126574	25.854381	NaN	...	NaN	NaN	NaN	NaN	NaN	1.804837	2.66637
std	8.791521e+06	NaN	56.750546	NaN	NaN	NaN	NaN	41.467058	11.007935	NaN	...	NaN	NaN	NaN	NaN	NaN	0.905609	1.51637
min	5.300000e+01	NaN	1.000000	NaN	NaN	NaN	NaN	-157.948486	-41.330428	NaN	...	NaN	NaN	NaN	NaN	NaN	1.000000	0.00000
25%	3.019625e+05	NaN	1.000000	NaN	NaN	NaN	NaN	77.081343	28.478713	NaN	...	NaN	NaN	NaN	NaN	NaN	1.000000	2.50000
50%	6.004089e+06	NaN	1.000000	NaN	NaN	NaN	NaN	77.191964	28.570469	NaN	...	NaN	NaN	NaN	NaN	NaN	2.000000	3.20000
75%	1.835229e+07	NaN	1.000000	NaN	NaN	NaN	NaN	77.282006	28.642758	NaN	...	NaN	NaN	NaN	NaN	NaN	2.000000	3.70000
max	1.850065e+07	NaN	216.000000	NaN	NaN	NaN	NaN	174.832089	55.976980	NaN	...	NaN	NaN	NaN	NaN	NaN	4.000000	4.90000

11 rows × 21 columns

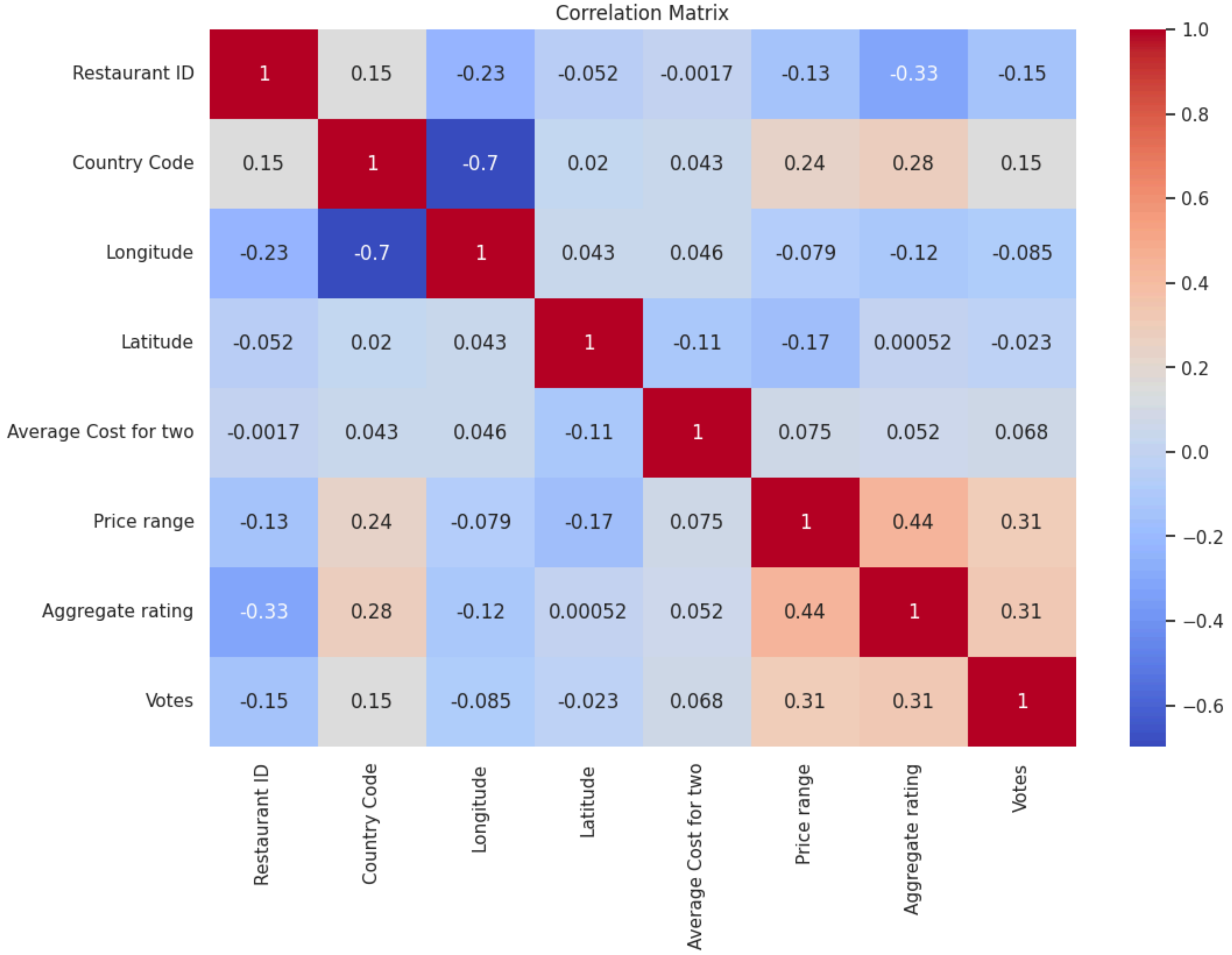
Data Cleaning

```
# Drop duplicate records
df.drop_duplicates(inplace=True)

# Handle missing values (example: fill numeric with median)
numeric_cols = df.select_dtypes(include=np.number).columns
df[numeric_cols] = df[numeric_cols].fillna(df[numeric_cols].median())

# Optional: Clean whitespace from column names
df.columns = df.columns.str.strip()
```

Exploratory Data Analysis (EDA)

> Correlation Heatmap (Numerical Features)

```
plt.figure(figsize=(12, 8))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap="coolwarm")
plt.title("Correlation Matrix")
plt.show()
```



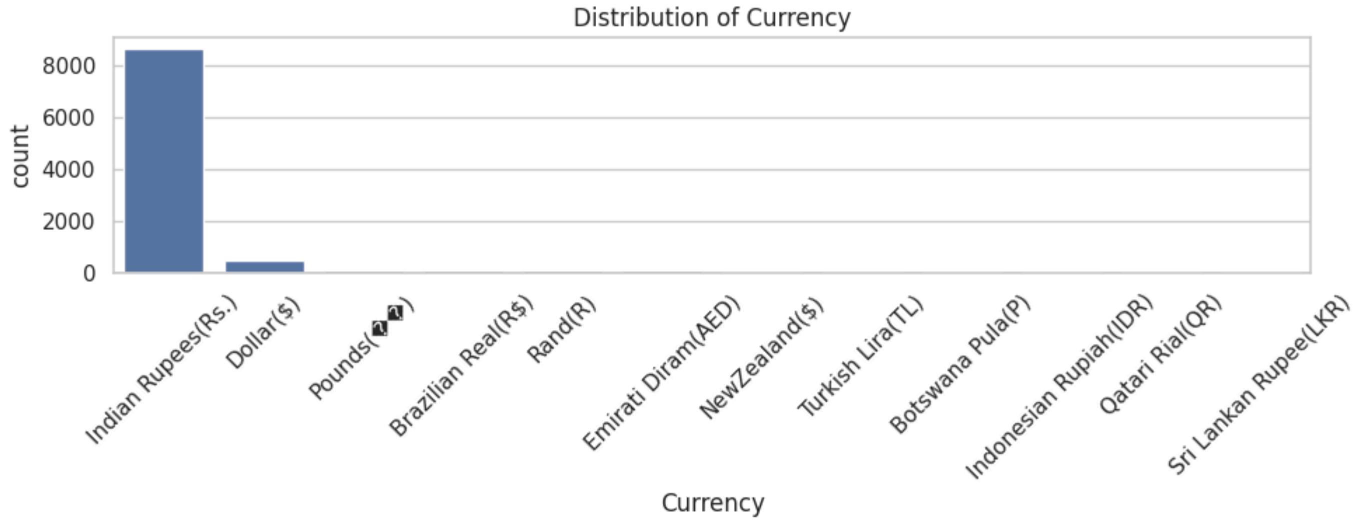
Distribution of Categorical Columns

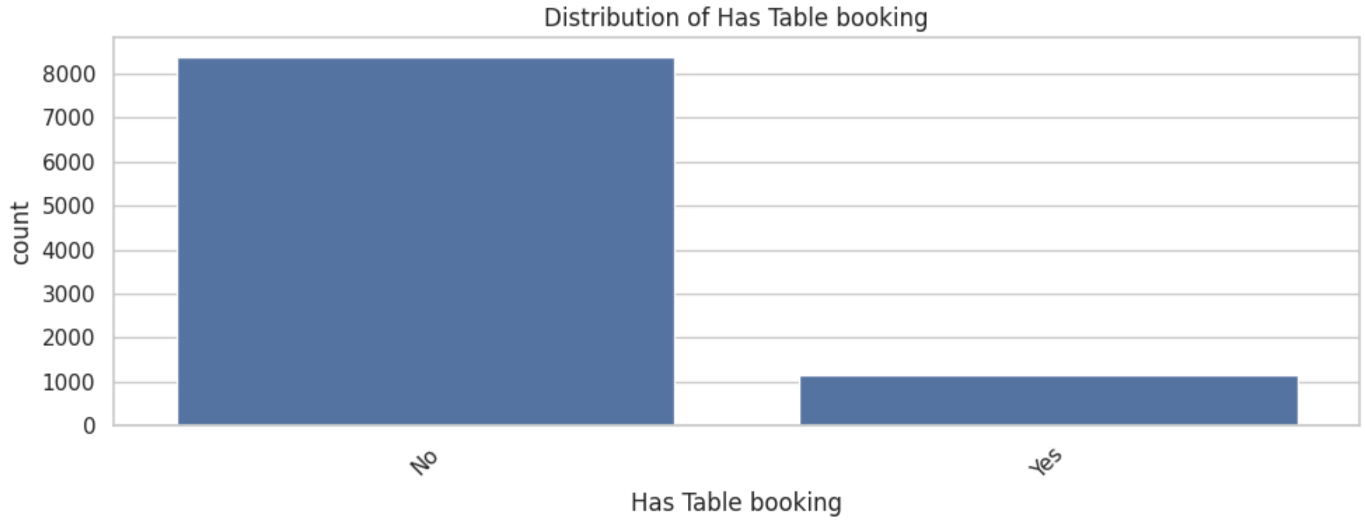
```
# Importing necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Configure plots
```

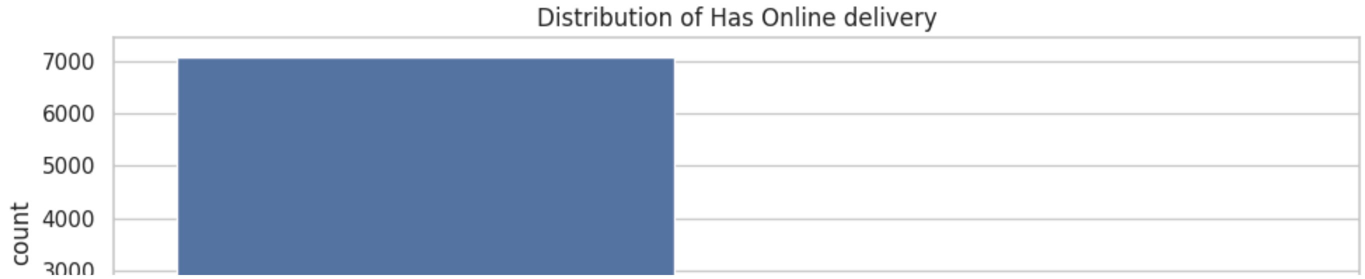
```
%matplotlib inline
sns.set(style="whitegrid")

# The following cell attempts to load the data using pd.read_csv
# Ensure the cell above with the import statement is run first.
# Load the CSV file (change the name if your file is different)
df = pd.read_csv('Dataset .csv')

# ... rest of the code
```

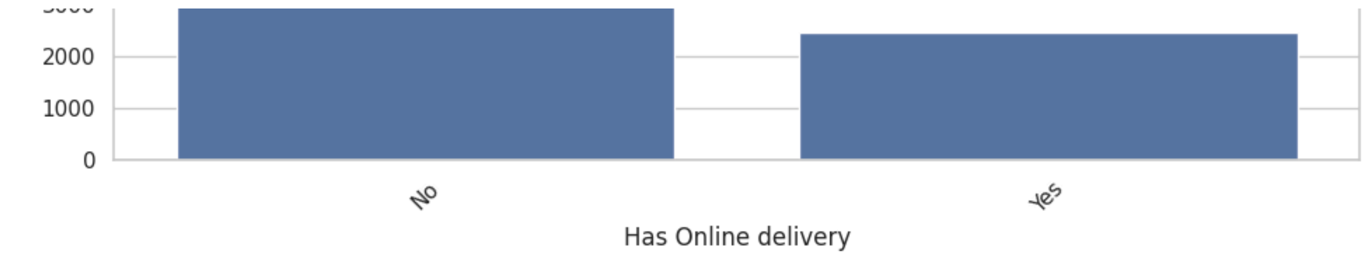
```
import matplotlib.pyplot as plt
import seaborn as sns

categorical_cols = df.select_dtypes(include='object').columns

for col in categorical_cols:
    if df[col].nunique() <= 15:  # limit to avoid overplotting
        plt.figure(figsize=(10, 4))
        sns.countplot(data=df, x=col, order=df[col].value_counts().index)
        plt.title(f"Distribution of {col}")
        plt.xticks(rotation=45)
        plt.tight_layout()
        plt.show()
    else:
        print(f"Skipped '{col}' due to too many unique categories: {df[col].nunique()}")
```

```
Skipped 'Restaurant Name' due to too many unique categories: 7446
Skipped 'City' due to too many unique categories: 141
Skipped 'Address' due to too many unique categories: 8918
Skipped 'Locality' due to too many unique categories: 1208
Skipped 'Locality Verbose' due to too many unique categories: 1265
Skipped 'Cuisines' due to too many unique categories: 1825
```

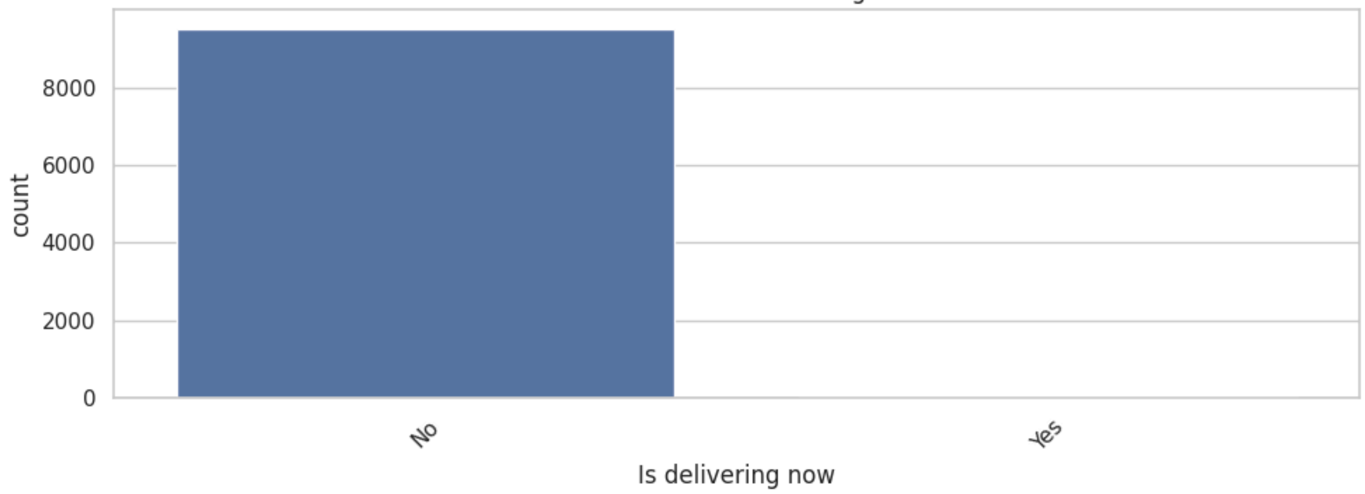




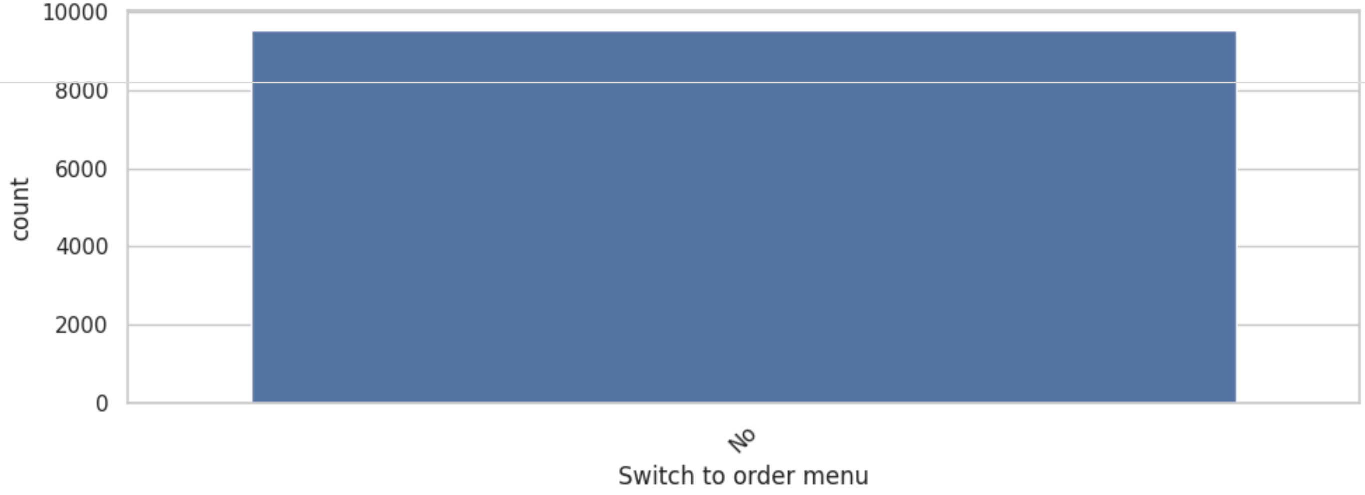




Distribution of Is delivering now

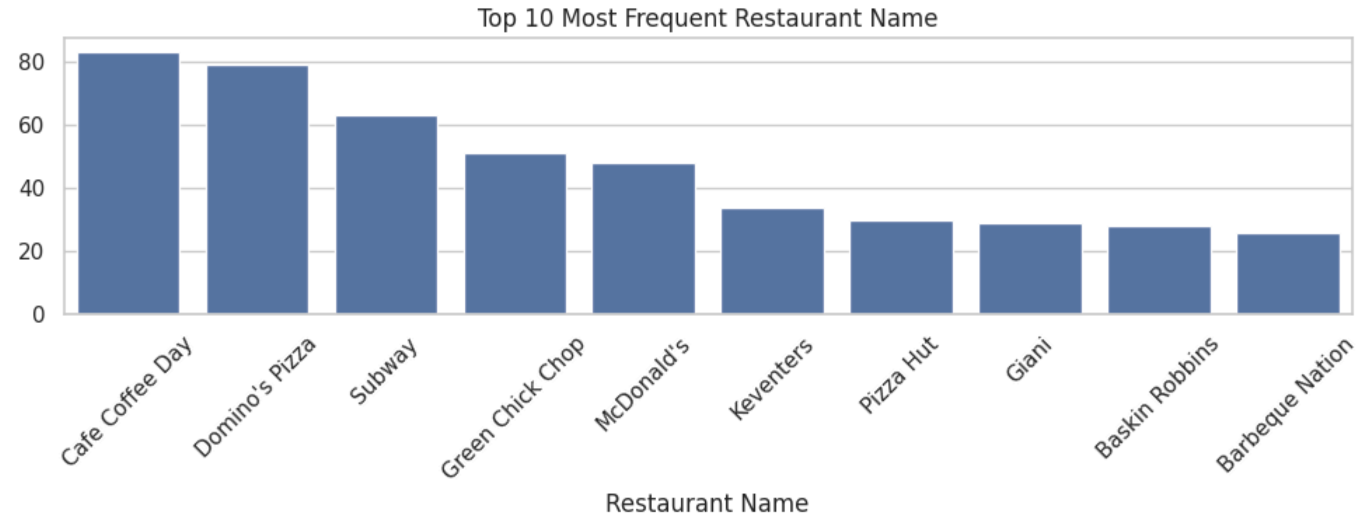


Distribution of Switch to order menu



Distribution of Rating color

```
top_n = 10
col = 'Restaurant Name'

plt.figure(figsize=(10, 4))
top_vals = df[col].value_counts().nlargest(top_n)
sns.barplot(x=top_vals.index, y=top_vals.values)
plt.title(f"Top {top_n} Most Frequent {col}")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



```
# If your dataset has a 'Date' column
# First, check the column names to ensure 'Date' exists
print(df.columns)

# If 'Date' exists and you want to proceed with datetime conversion and plotting:
# df['Date'] = pd.to_datetime(df['Date'], errors='coerce')

# plt.figure(figsize=(12,6))
# sns.lineplot(data=df, x='Date', y=df.columns[-1])  # Change y column accordingly
# plt.title("Trend Over Time")
# plt.xlabel("Date")
# plt.ylabel("Value")
# plt.tight_layout()
# 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')
```

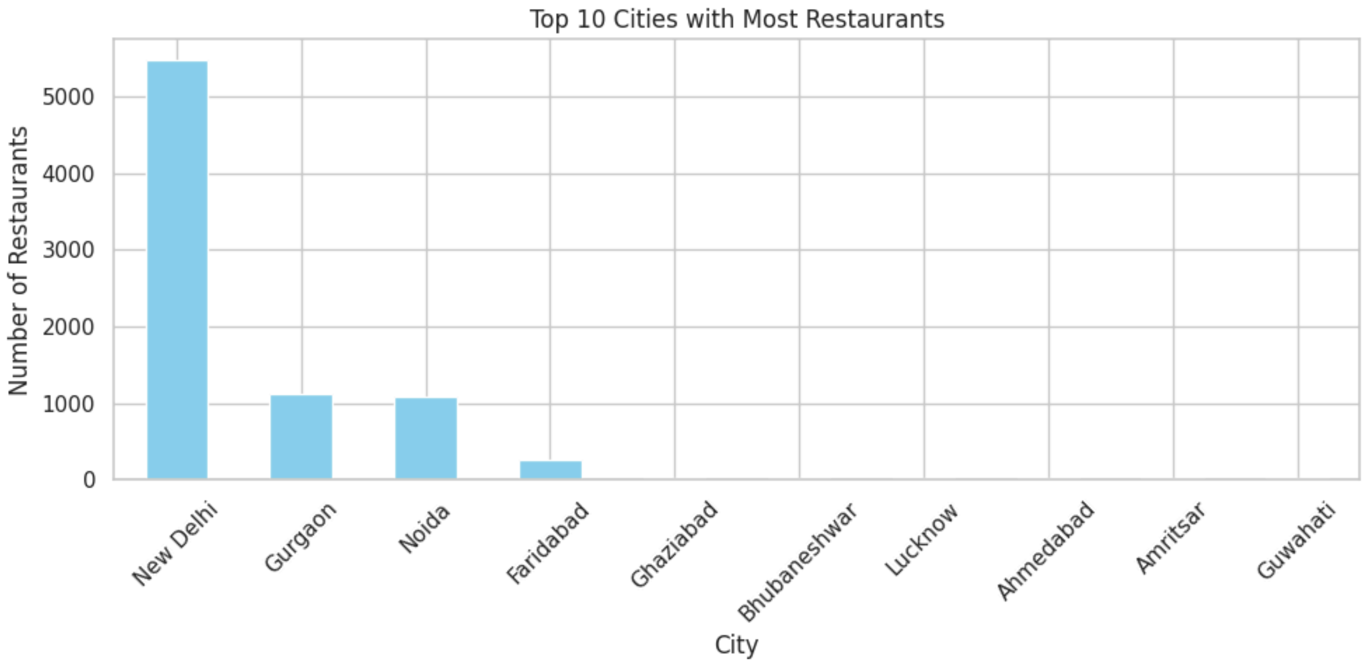
```
df.to_csv('Cleaned_Dataset.csv', index=False)
print("✅ Cleaned data saved as 'Cleaned_Dataset.csv'")
```

```
✅ Cleaned data saved as 'Cleaned_Dataset.csv'
```

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

%matplotlib inline
sns.set(style="whitegrid")
```

> Top 10 Cities with Most Restaurants

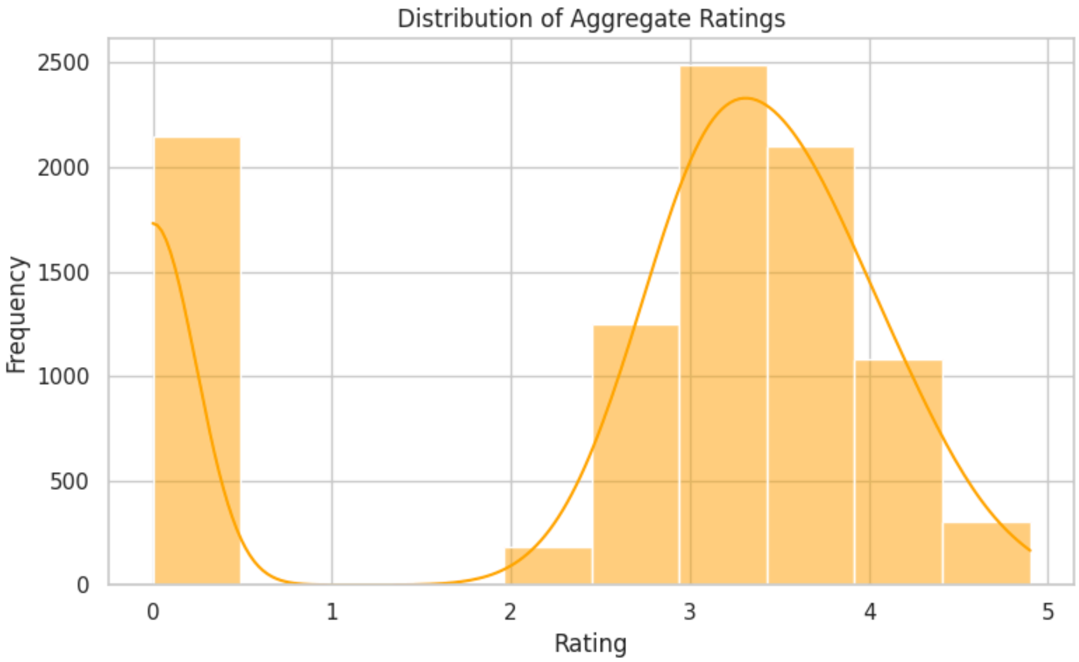
```
plt.figure(figsize=(10,5))
df['City'].value_counts().nlargest(10).plot(kind='bar', color='skyblue')
plt.title("Top 10 Cities with Most Restaurants")
plt.xlabel("City")
plt.ylabel("Number of Restaurants")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Top 10 Cities with Most Restaurants

Distribution of Aggregate Ratings

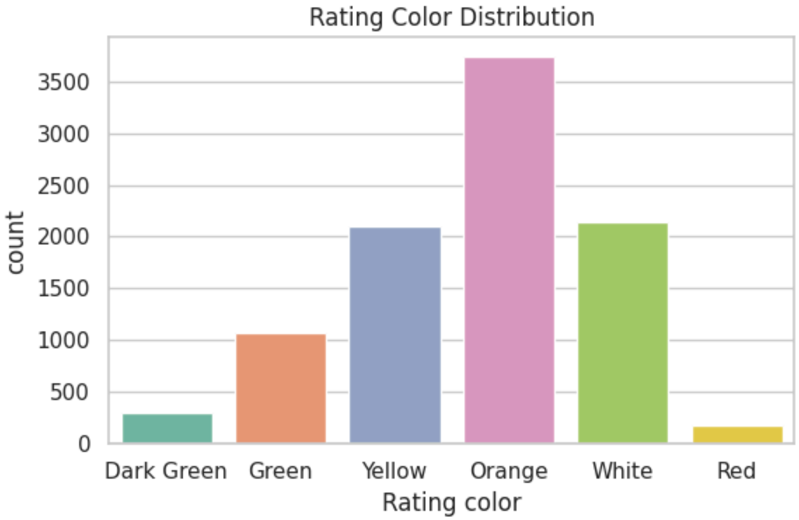
```
plt.figure(figsize=(8,5))
sns.histplot(df['Aggregate rating'], bins=10, kde=True, color='orange')
plt.title("Distribution of Aggregate Ratings")
plt.xlabel("Rating")
plt.ylabel("Frequency")
plt.tight_layout()
plt.show()
```



Rating Color Distribution

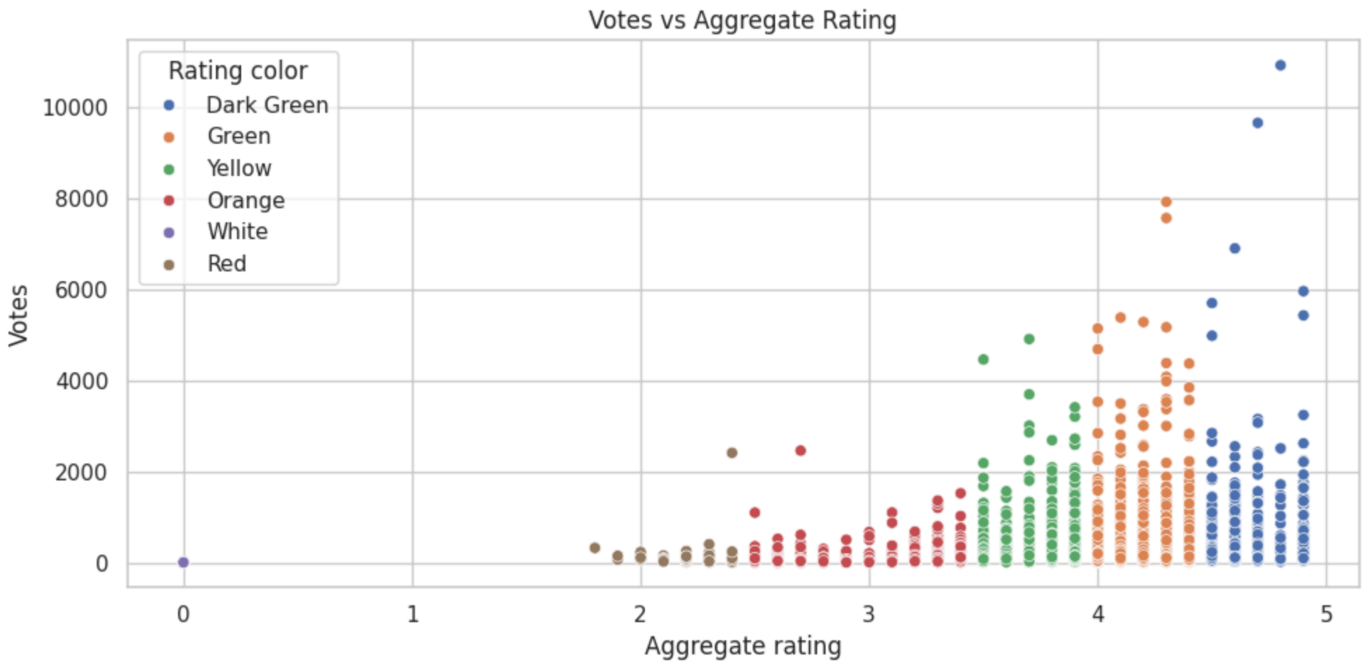
```
plt.figure(figsize=(6,4))
sns.countplot(data=df, x='Rating color', palette='Set2')
plt.title("Rating Color Distribution")
plt.tight_layout()
plt.show()
```

```
<ipython-input-16-377d9a1c264d>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(data=df, x='Rating color', palette='Set2')
```



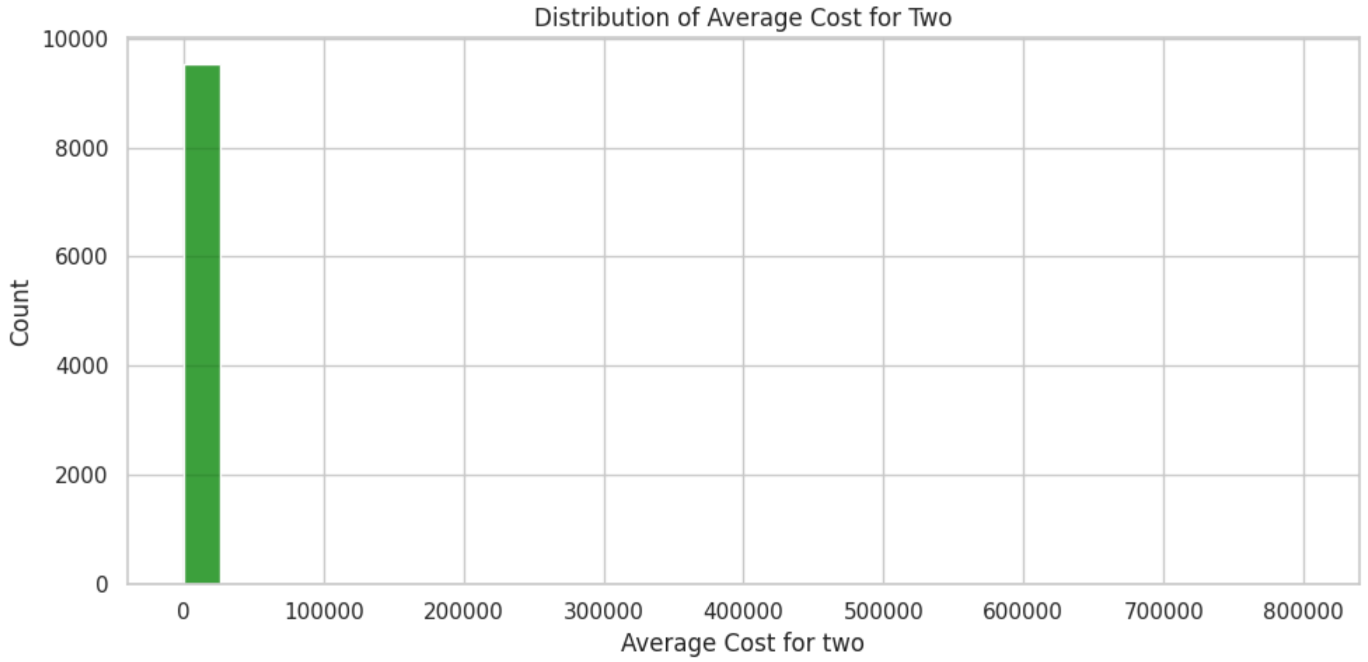
Votes vs Aggregate Rating

```
plt.figure(figsize=(10,5))
sns.scatterplot(data=df, x='Aggregate rating', y='Votes', hue='Rating color')
plt.title("Votes vs Aggregate Rating")
plt.tight_layout()
plt.show()
```



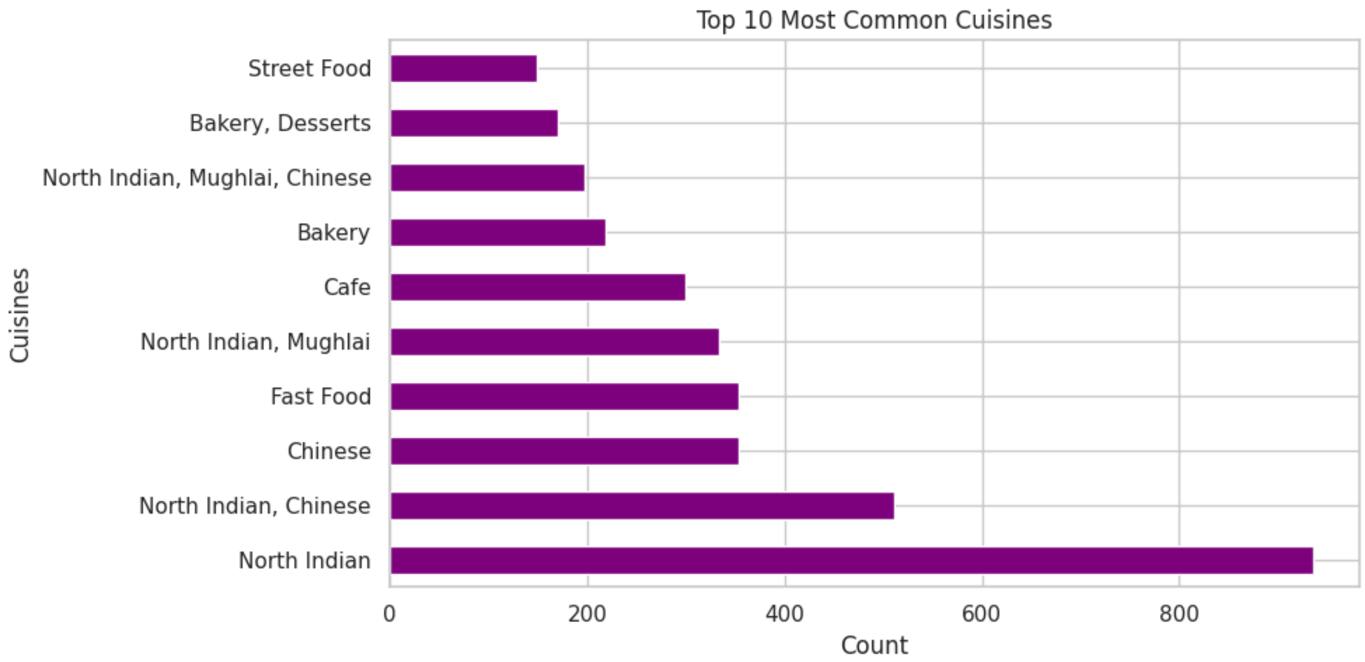
Average Cost for Two Distribution

```
plt.figure(figsize=(10,5))
sns.histplot(df['Average Cost for two'], bins=30, color='green')
plt.title("Distribution of Average Cost for Two")
plt.tight_layout()
plt.show()
```



Top 10 Most Common Cuisines

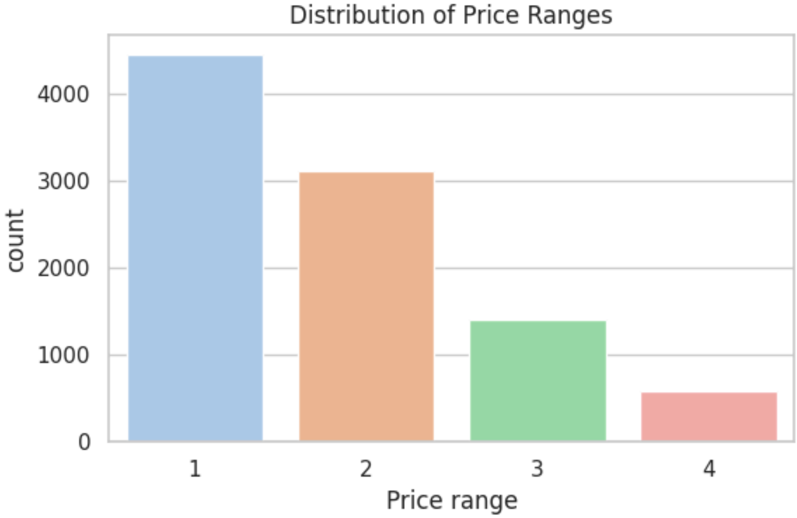
```
plt.figure(figsize=(10,5))
df['Cuisines'].value_counts().nlargest(10).plot(kind='barh', color='purple')
plt.title("Top 10 Most Common Cuisines")
plt.xlabel("Count")
plt.tight_layout()
plt.show()
```



Price Range Distribution

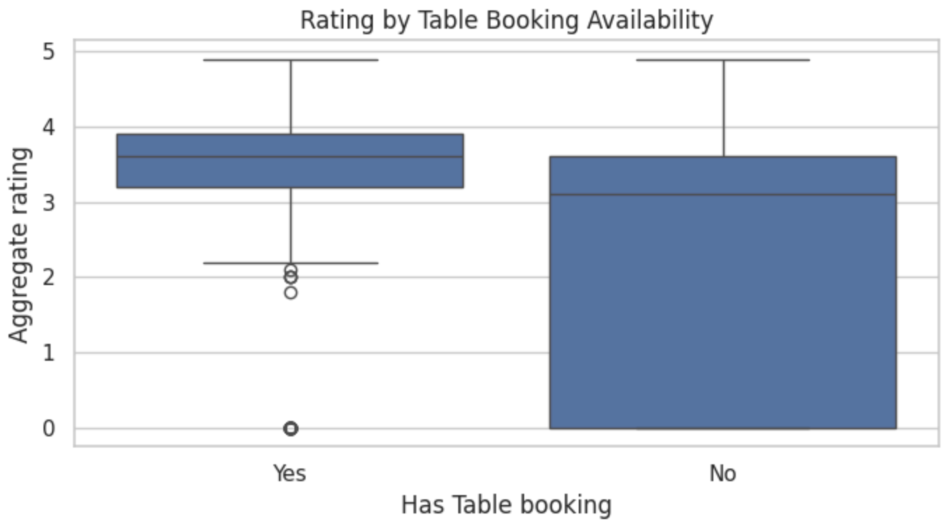
```
plt.figure(figsize=(6,4))
sns.countplot(data=df, x='Price range', palette='pastel')
plt.title("Distribution of Price Ranges")
plt.tight_layout()
plt.show()
```

```
<ipython-input-20-3ea1c3d33024>:2: FutureWarning: 

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(data=df, x='Price range', palette='pastel')
```



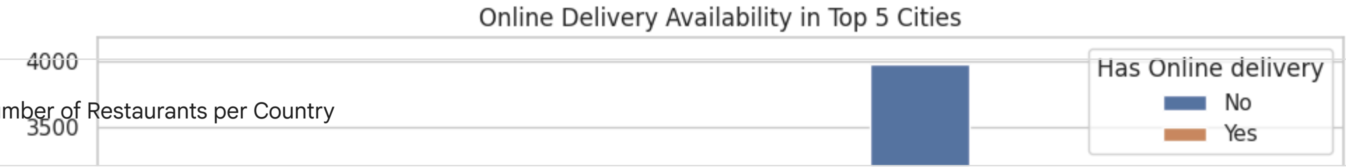
Has Table Booking vs Rating

```
plt.figure(figsize=(7,4))
sns.boxplot(data=df, x='Has Table booking', y='Aggregate rating')
plt.title("Rating by Table Booking Availability")
plt.tight_layout()
plt.show()
```



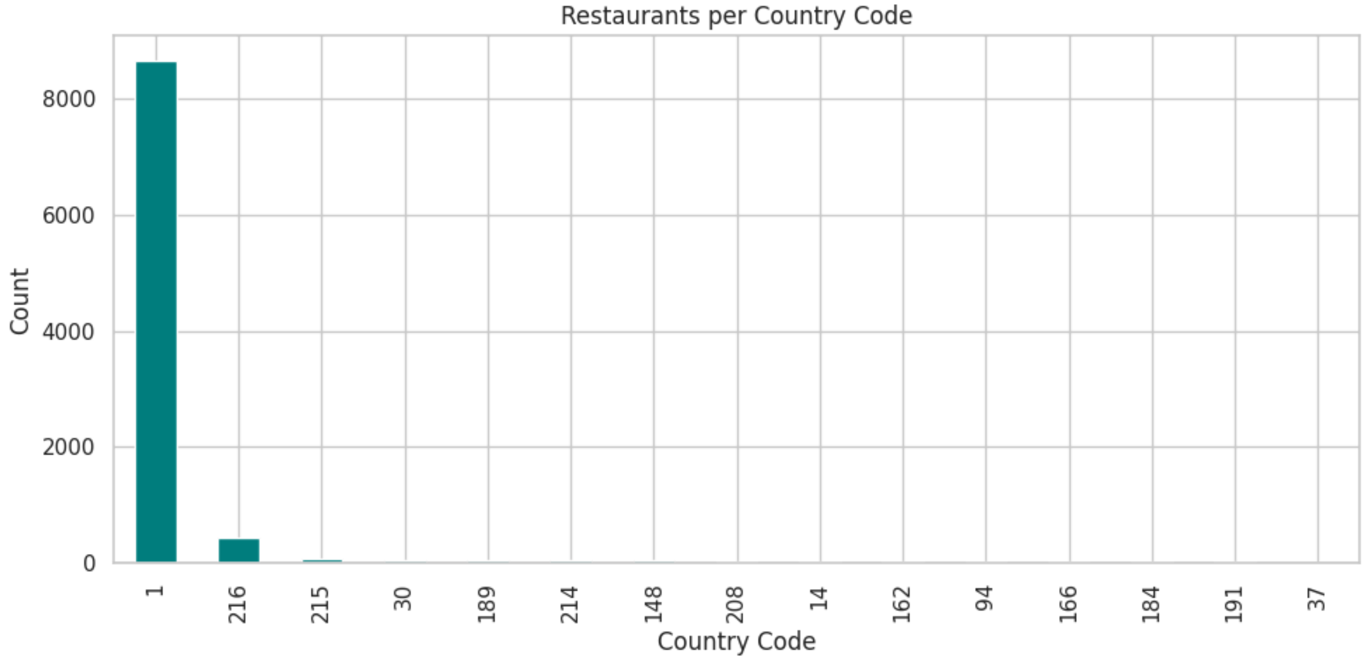
Online Delivery by City (Top 5 Cities)

```
top_cities = df['City'].value_counts().nlargest(5).index
plt.figure(figsize=(10,5))
sns.countplot(data=df[df['City'].isin(top_cities)], x='City', hue='Has Online delivery')
plt.title("Online Delivery Availability in Top 5 Cities")
plt.tight_layout()
plt.show()
```



Number of Restaurants per Country

```
plt.figure(figsize=(10,5))
df['Country Code'].value_counts().plot(kind='bar', color='teal')
plt.title("Restaurants per Country Code")
plt.xlabel("Country Code")
plt.ylabel("Count")
plt.tight_layout()
plt.show()
```



Switch to Order Menu Distribution

```
plt.figure(figsize=(6,4))
sns.countplot(data=df, x='Switch to order menu')
plt.title("Switch to Order Menu Count")
plt.tight_layout()
plt.show()
```