#Restaurant Data Analysis In this project, we will be doing data analysis and data visualisation.

```
from google.colab import files

# Upload the file
uploaded = files.upload()

# Read the uploaded file into a DataFrame
import pandas as pd
df = pd.read_csv(io.BytesIO(uploaded['restaurant_dataset.csv']))

# Display the first few rows
print(df.head())
```

```
<IPython.core.display.HTML object>

Saving restaurant_dataset.csv to restaurant_dataset.csv
   Restaurant ID         Restaurant Name  Country Code
City  \
0             53        Le Petit Souffle           162       Makati
City
1             55        Izakaya Kikufuji           162       Makati
City
2             60  Heat - Edsa Shangri-La           162  Mandaluyong
City
3             64                   Ooma           162  Mandaluyong
City
4             65             Sambo Kojin           162  Mandaluyong
City

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

                                    Locality  Unnamed: 6
Longitude  \
0   Century City Mall, Poblacion, Makati City         NaN -98.989100

1  Little Tokyo, Legaspi Village, Makati City         NaN -84.154000

2  Edsa Shangri-La, Ortigas, Mandaluyong City         NaN  80.354002

3      SM Megamall, Ortigas, Mandaluyong City         NaN -84.175900

4      SM Megamall, Ortigas, Mandaluyong City         NaN -84.219400
```

```
    Latitude                      Cuisines  ...            Currency  \
0  44.515800                           NaN  ...           Dollar($)
1  31.577200                           NaN  ...           Dollar($)
2  26.472001  Indian, Chinese, Continental  ...  Indian Rupees(Rs.)
3  31.588200                           NaN  ...           Dollar($)
4  31.615800                       Mexican  ...           Dollar($)

  Has Table booking Has Online delivery Is delivering now  \
0                No                  No                No
1                No                  No                No
2                No                  No                No
3                No                  No                No
4                No                  No                No

  Switch to order menu Price range  Aggregate rating  Rating color  \
0                   No           1               3.4        Orange
1                   No           1               3.4        Orange
2                   No           1               3.6        Yellow
3                   No           1               3.4        Orange
4                   No           1               3.4        Orange

  Rating text Votes
0     Average    11
1     Average    34
2        Good    34
3     Average    36
4     Average    45

[5 rows x 21 columns]
```

```
df.describe()
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 8,\n  \"fields\": [\n
{\n      \"column\": \"Restaurant ID\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 7645150.642496774,\n
\"min\": 53.0,\n        \"max\": 18500652.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n
9051128.349178096,\n          6004089.0,\n          9551.0\
n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Country Code\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 3362.6853318315943,\n        \"min\":
1.0,\n        \"max\": 9551.0,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          18.365616165846507,\n          216.0,\n
56.75054560094657\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Unnamed: 6\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": null,\n        \"min\": 0.0,\n
\"max\": 0.0,\n        \"num_unique_values\": 1,\n        \"samples\":
[\n          0.0\n        ],\n        \"semantic_type\": \"\",\n
```

\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Longitude\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 3360.2106786503405,\n   \"min\": -
157.948486,\n   \"max\": 9551.0,\n   \"num_unique_values\":
8,\n   \"samples\": [\n   64.12657446168704\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Latitude\",\n   \"properties\":
{\n   \"dtype\": \"number\",\n   \"std\":
3369.9729885680754,\n   \"min\": -41.330428,\n   \"max\":
9551.0,\n   \"num_unique_values\": 8,\n   \"samples\": [\n
25.854380700074756\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Average Cost for two\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 281478.0961029089,\n   \"min\":
0.0,\n   \"max\": 800000.0,\n   \"num_unique_values\": 8,\n
\"samples\": [\n   1199.2107632708617\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Price range\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
3376.146607146113,\n   \"min\": 0.905608847397614,\n
\"max\": 9551.0,\n   \"num_unique_values\": 6,\n
\"samples\": [\n   9551.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"Aggregate rating\",\n
\"properties\": {\n   \"dtype\": \"number\",\n   \"std\":
3375.855226922187,\n   \"min\": 0.0,\n   \"max\": 9551.0,\n
\"num_unique_values\": 8,\n   \"samples\": [\n
2.666370013611141\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"Votes\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 4699.7638410944965,\n   \"min\": 0.0,\n   \"max\":
10934.0,\n   \"num_unique_values\": 8,\n   \"samples\": [\n
156.909747670401\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}

```
df.isnull()
```

{"type":"dataframe"}

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

```
Restaurant ID            0
Restaurant Name          0
Country Code             0
City                     0
Address                  0
Locality                 0
Unnamed: 6            9551
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
```

Handling Null Values: The columns with null values are: 'Unnamed: 6', 'Cuisines'. For the 'Unnamed: 6' column, since it has 9551 null values, it might be best to drop this column altogether, as it doesn't seem to contain useful information. For the 'Cuisines' column, you can either drop the rows with null values or fill the null values with a suitable value (e.g., 'Unknown' or the most common cuisine).

```
# Drop the 'Unnamed: 6' column
df = df.drop('Unnamed: 6', axis=1)

# Fill null values in 'Cuisines' column
df['Cuisines'] = df['Cuisines'].fillna('Unknown')
```

Handle Categorical Variables: The following columns are likely categorical: 'Restaurant Name', 'Country Code', 'City', 'Locality', 'Cuisines', 'Currency', 'Has Table booking', 'Has Online delivery', 'Is delivering now', 'Switch to order menu', 'Price range', 'Rating color', 'Rating text'.

You can encode these columns using techniques like one-hot encoding or label encoding, depending on the cardinality of the categories.

```
# Example of one-hot encoding for 'Cuisines' column
from sklearn.preprocessing import OneHotEncoder
ohe = OneHotEncoder()
cuisines_ohe = ohe.fit_transform(df[['Cuisines']]).toarray()
df = df.join(pd.DataFrame(cuisines_ohe,
columns=ohe.get_feature_names_out()))
df = df.drop('Cuisines', axis=1)
```

Feature Engineering: Based on the columns available, you could create new features like: Distance from restaurant (using the 'Longitude' and 'Latitude' columns) Average rating per cuisine

Price range category (e.g., low, medium, high)

Scaling and Normalization: The numeric columns like 'Average Cost for two', 'Aggregate rating', and 'Votes' might benefit from scaling or normalization.

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
df[['Average Cost for two', 'Aggregate rating', 'Votes']] =
scaler.fit_transform(df[['Average Cost for two', 'Aggregate rating',
'Votes']])

df
```

{"type":"dataframe","variable_name":"df"}

Handling Outliers: You can check for outliers in the numeric columns and handle them

```
from scipy.stats import zscore

z_scores = zscore(df['Average Cost for two'])
df = df[(z_scores < 3) & (z_scores > -3)]

df.isna()
```

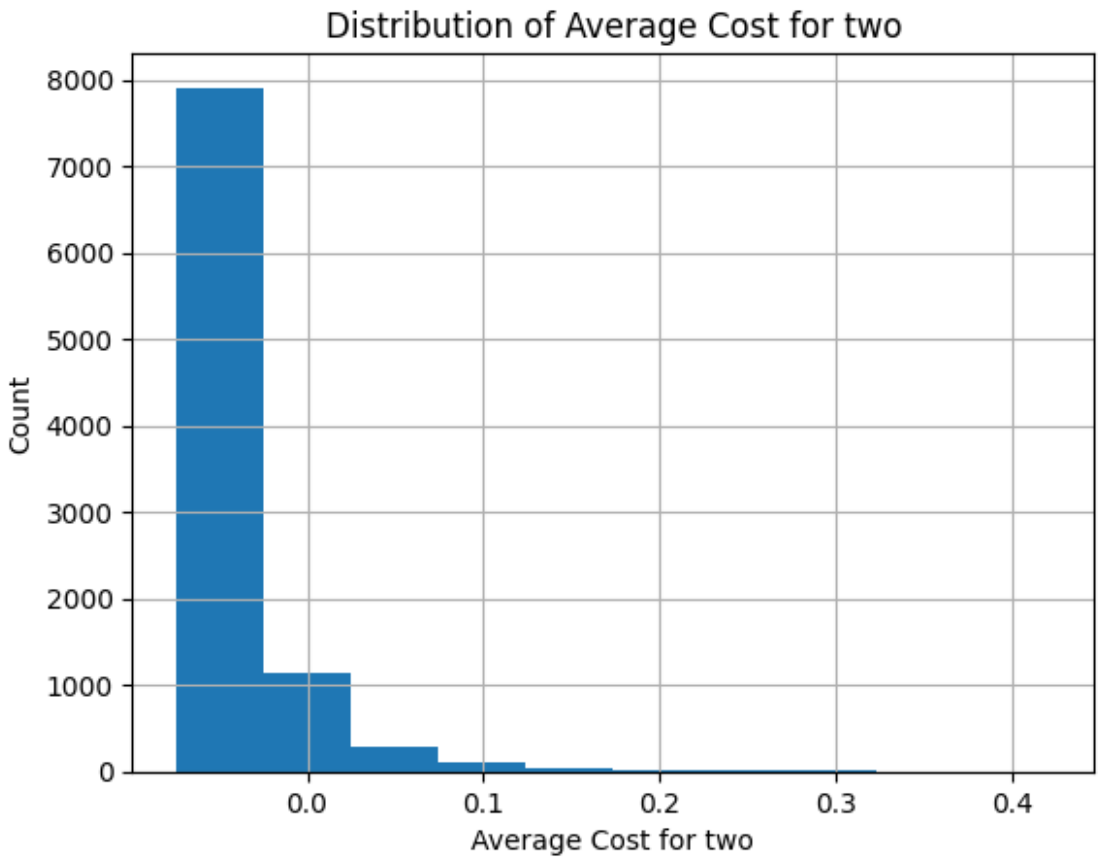
{"type":"dataframe"}

#Data Visualisation

#Histogram

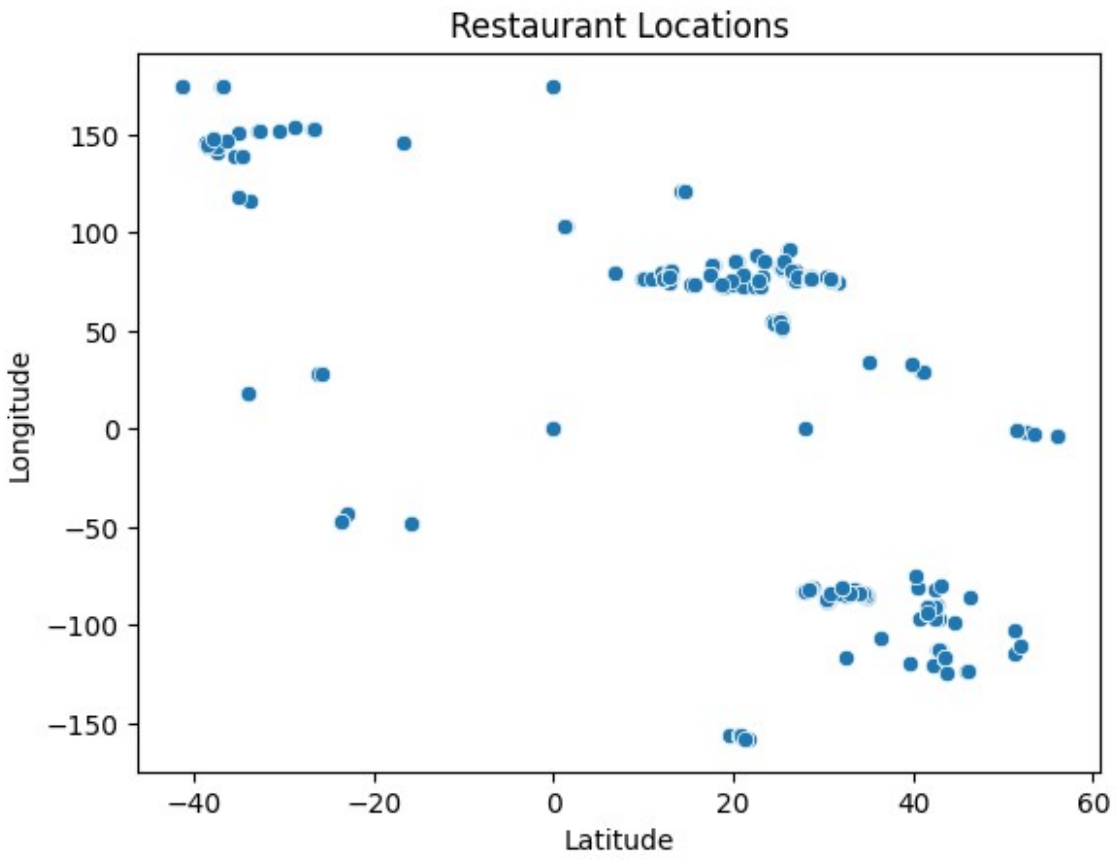
```
import matplotlib.pyplot as plt

# Plot a histogram of the 'Average Cost for two' column
df['Average Cost for two'].hist()
plt.xlabel('Average Cost for two')
plt.ylabel('Count')
plt.title('Distribution of Average Cost for two')
plt.show()
```



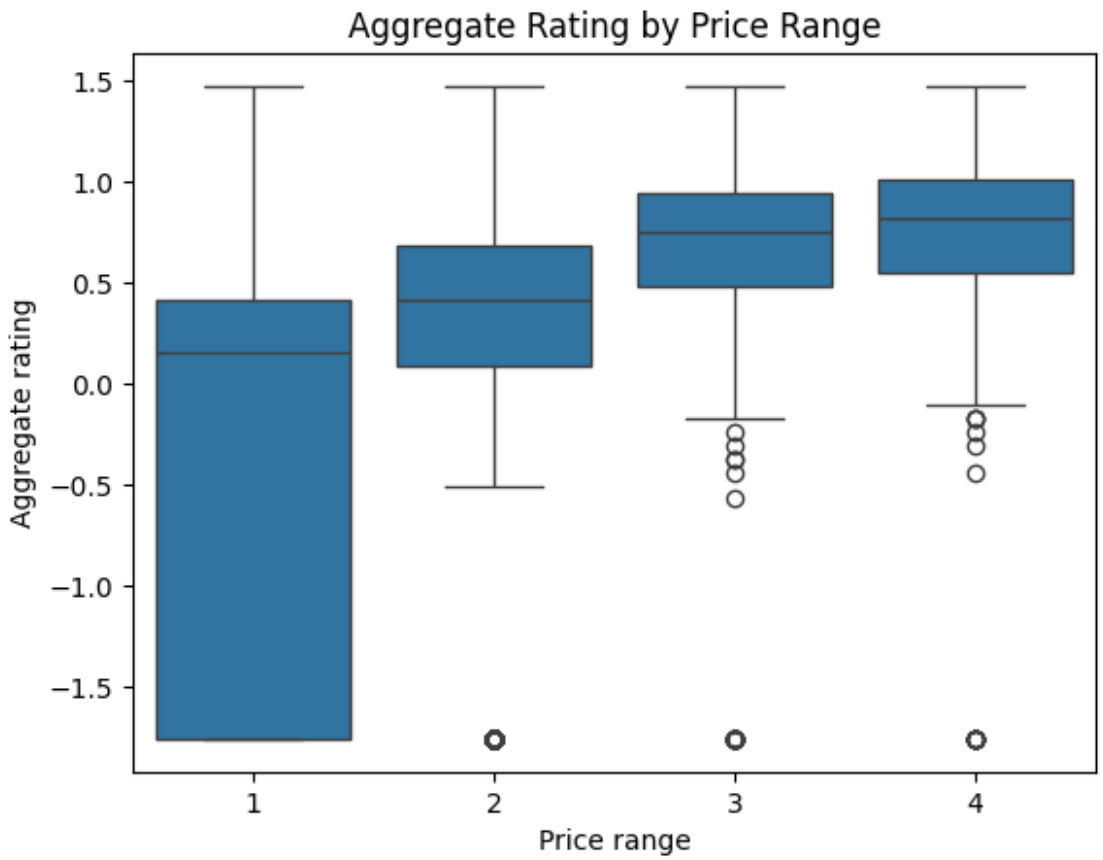
#Scatter Plot

```
# Plot a scatter plot of 'Latitude' vs 'Longitude'
sns.scatterplot(x='Latitude', y='Longitude', data=df)
plt.xlabel('Latitude')
plt.ylabel('Longitude')
plt.title('Restaurant Locations')
plt.show()
```



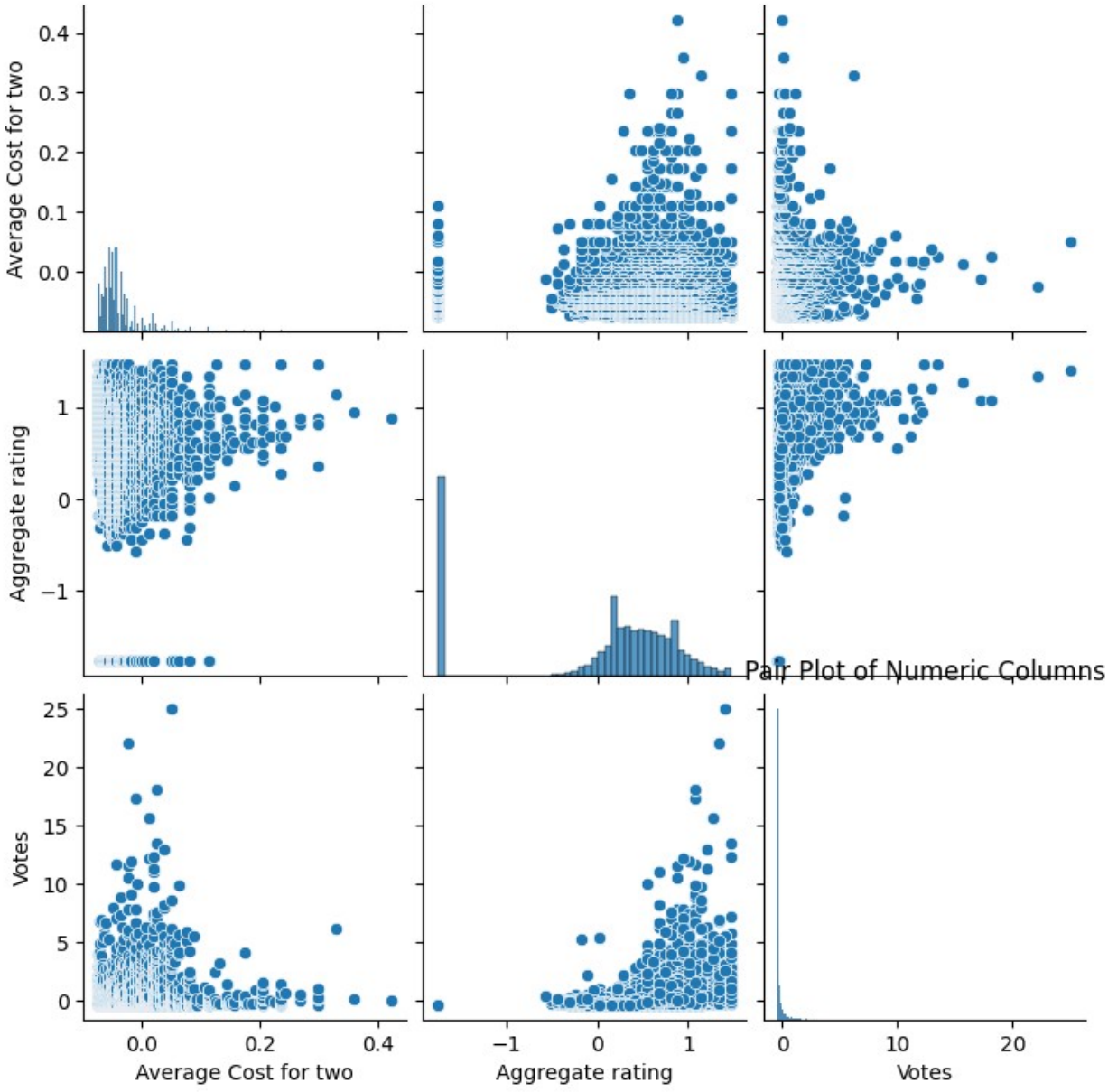
#Box PLot

```
# Plot a box plot of 'Aggregate rating' grouped by 'Price range'
sns.boxplot(x='Price range', y='Aggregate rating', data=df)
plt.xlabel('Price range')
plt.ylabel('Aggregate rating')
plt.title('Aggregate Rating by Price Range')
plt.show()
```



#Pair Plot

```
# Create a pair plot of numeric columns
sns.pairplot(df[['Average Cost for two', 'Aggregate rating',
'Votes']])
plt.title('Pair Plot of Numeric Columns')
plt.show()
```


Pair Plot of Numeric Columns
Average Cost for two
Aggregate rating
Votes