Marketing

Description

Data Analysis is the process of creating a story using the data for easy and effective communication. It mostly utilizes visualization methods like plots, charts and tables to convey what the data holds beyond the formal modelling or hypothesis testing task.

Read the information given below and also refer to the data dictionary provided separately in an excel file to build your understanding.

Problem Statement

A restaurant consolidator is looking to revamp its B-to-C portal using intelligent automation tech. It is in search of different matrix to identify and recommend restaurants. To make sure an effective model can be achieved it is important to understand the behaviour of the data in hand.

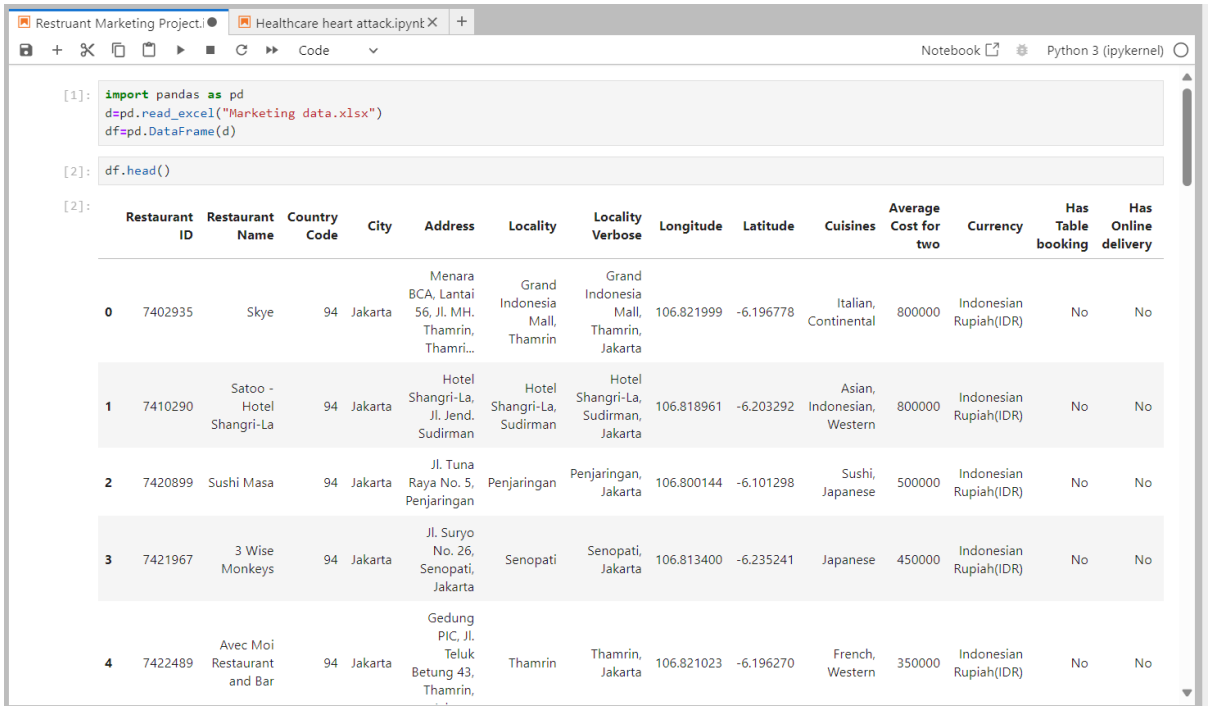
Approach:

1. Data Preliminary analysis:
    1. Perform preliminary data inspection and report the findings as the structure of the data, missing values, duplicates cleaning variable names etc.
    2. Based on the findings from the previous questions identify duplicates and remove them.

1. Prepare a preliminary report of the given data by answering following questions. Expressing the results using graphs and plot will make it more appealing.
    1. Explore the geographical distribution of the restaurants, finding out the cities with maximum / minimum number of restaurants.
    2. Explore how ratings are distributed overall.
    3. Restaurant franchise is a thriving venture. So, it becomes very important to explore the franchise with most national presence.
    4. What is the ratio between restaurants that allow table booking vs that do not allow table booking?
    5. What is the percentage of restaurants providing online delivery?
    6. Is there a difference in no. of votes for the restaurants that deliver and the restaurant that don't?
    7. What are the top 10 cuisines served across cities?
    8. What is the maximum and minimum no. of cuisines that a restaurant serves? Also, what is the relationship between No. of cuisines served and Ratings
    9. Discuss the cost vs the other variables.
    10. Explain the factors in the data that may have an effect on ratings e.g. No. of cuisines, cost, delivery option etc.

2. Data Preliminary analysis

1. Retrieve Data from Excel to Notebook using “pandas”



```
[1]: import pandas as pd
     d=pd.read_excel("Marketing data.xlsx")
     df=pd.DataFrame(d)

[2]: df.head()
```

[2]:

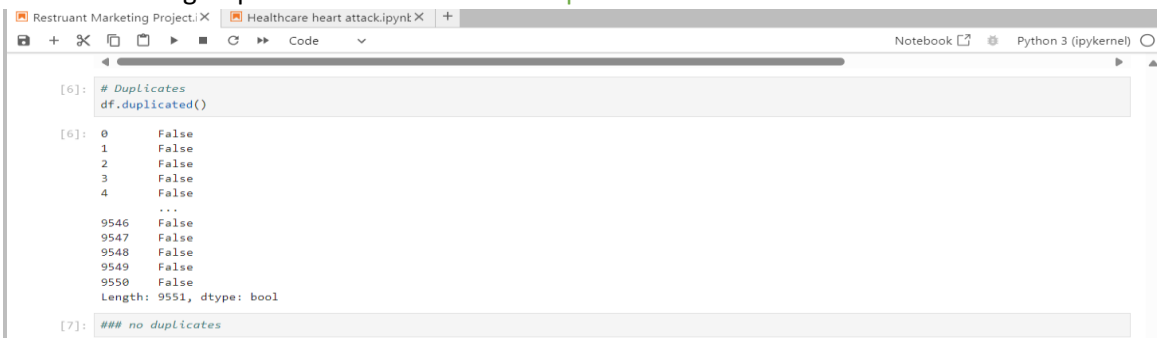
	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	Average Cost for two	Currency	Has Table booking	Has Online delivery
0	7402935	Skye	94	Jakarta	Menara BCA, Lantai 56, Jl. MH. Thamrin, Thamri...	Grand Indonesia Mall, Thamrin	Grand Indonesia Mall, Thamrin, Jakarta	106.821999	-6.196778	Italian, Continental	800000	Indonesian Rupiah(IDR)	No	No
1	7410290	Satoo - Hotel Shangri-La	94	Jakarta	Hotel Shangri-La, Jl. Jend. Sudirman	Hotel Shangri-La, Sudirman	Hotel Shangri-La, Sudirman, Jakarta	106.818961	-6.203292	Asian, Indonesian, Western	800000	Indonesian Rupiah(IDR)	No	No
2	7420899	Sushi Masa	94	Jakarta	Jl. Tuna Raya No. 5, Penjaringan	Penjaringan	Penjaringan, Jakarta	106.800144	-6.101298	Sushi, Japanese	500000	Indonesian Rupiah(IDR)	No	No
3	7421967	3 Wise Monkeys	94	Jakarta	Jl. Suryo No. 26, Senopati, Jakarta	Senopati	Senopati, Jakarta	106.813400	-6.235241	Japanese	450000	Indonesian Rupiah(IDR)	No	No
4	7422489	Avec Moi Restaurant and Bar	94	Jakarta	Gedung PIC, Jl. Teluk Betung 43, Thamrin,	Thamrin	Thamrin, Jakarta	106.821023	-6.196270	French, Western	350000	Indonesian Rupiah(IDR)	No	No

2. Searching for Missing values: No missing vales Found

```
[4]: # 2. Missing Values
     print("\nMissing Values:")
     print(df.isnull().sum())
```

```
Missing Values:
Restaurant ID         0
Restaurant Name       1
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
Price range           0
Aggregate rating      0
Rating color          0
Rating text           0
Votes                 0
dtype: int64
```

3. Finding Duplicates in data set . : No duplicates Found

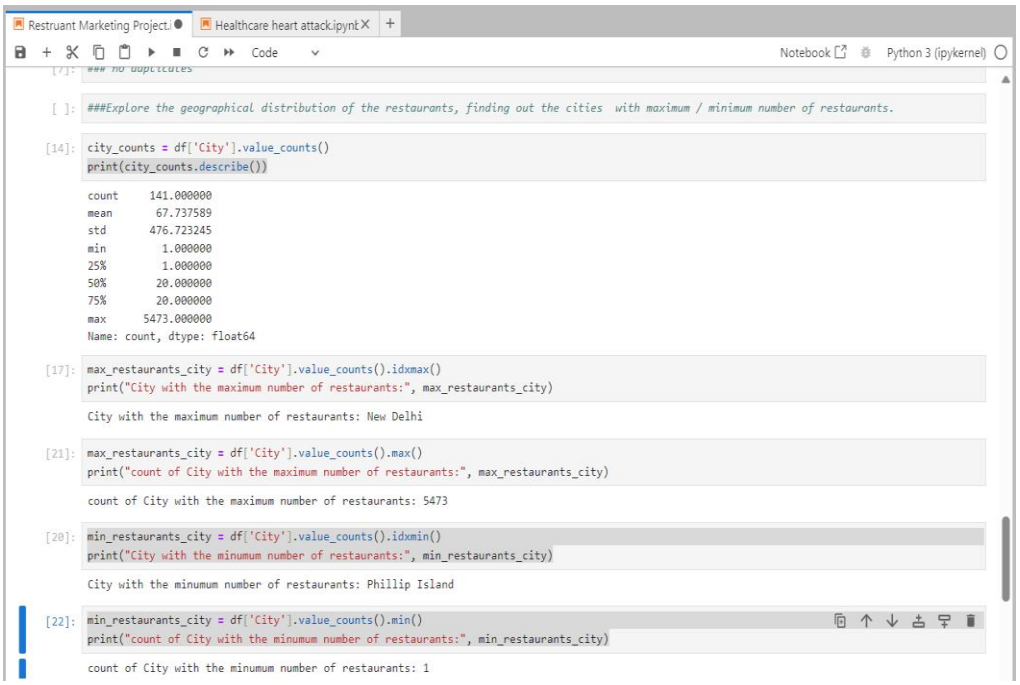


```
[6]: # Duplicates
     df.duplicated()

[6]: 0       False
     1       False
     2       False
     3       False
     4       False
             ...
     9546    False
     9547    False
     9548    False
     9549    False
     9550    False
     Length: 9551, dtype: bool

[7]: ### no duplicates
```

3. Prepare a preliminary report of the given data by answering following questions. Expressing the results using graphs and plot will make it more appealing
    1. Explore the geographical distribution of the restaurants, finding out the cities with maximum / minimum number of restaurants.



```
###Explore the geographical distribution of the restaurants, finding out the cities  with maximum / minimum number of restaurants.

city_counts = df['City'].value_counts()
print(city_counts.describe())
```

```
count     141.000000
mean       67.737589
std       476.723245
min         1.000000
25%         1.000000
50%        20.000000
75%        20.000000
max      5473.000000
Name: count, dtype: float64
```

```
max_restaurants_city = df['City'].value_counts().idxmax()
print("City with the maximum number of restaurants:", max_restaurants_city)
```

```
City with the maximum number of restaurants: New Delhi
```

```
max_restaurants_city = df['City'].value_counts().max()
print("count of City with the maximum number of restaurants:", max_restaurants_city)
```

```
count of City with the maximum number of restaurants: 5473
```

```
min_restaurants_city = df['City'].value_counts().idxmin()
print("City with the minumum number of restaurants:", min_restaurants_city)
```

```
City with the minumum number of restaurants: Phillip Island
```

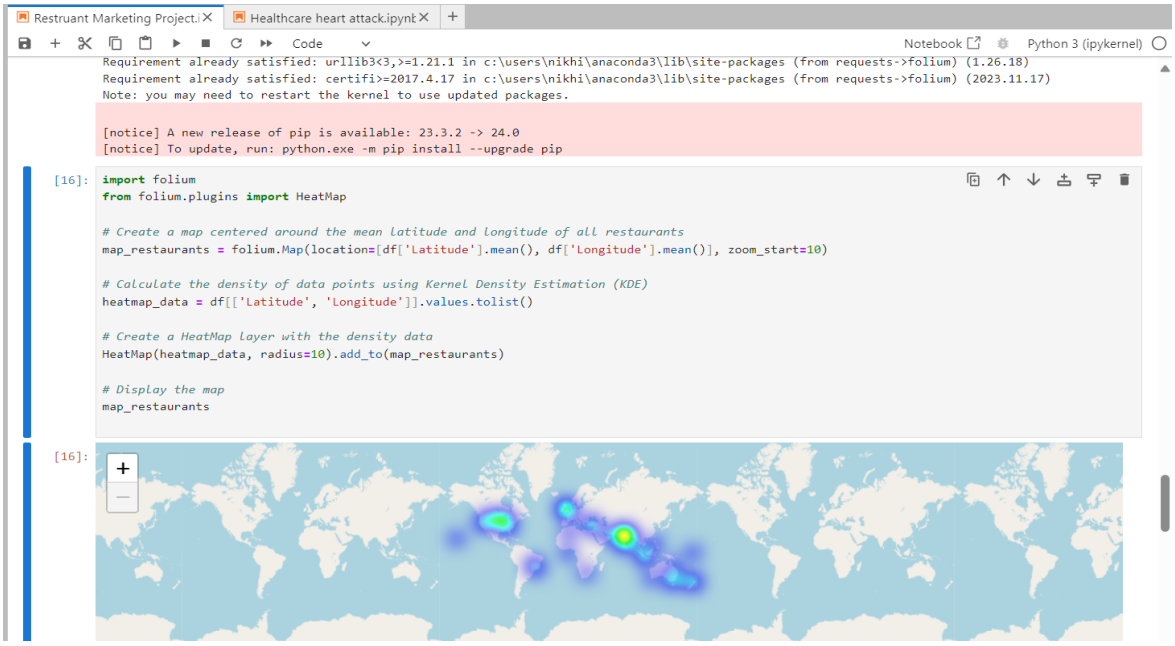
```
min_restaurants_city = df['City'].value_counts().min()
print("count of City with the minumum number of restaurants:", min_restaurants_city)
```

```
count of City with the minumum number of restaurants: 1
```

MAX: New Delhi= 5473 no's

MIN: Phillip Island = 1 no's

Heat map to plot data



```
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\nikhi\anaconda3\lib\site-packages (from requests->folium) (1.26.18)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\nikhi\anaconda3\lib\site-packages (from requests->folium) (2023.11.17)
Note: you may need to restart the kernel to use updated packages.

[notice] A new release of pip is available: 23.3.2 -> 24.0
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
import folium
from folium.plugins import HeatMap

# Create a map centered around the mean latitude and longitude of all restaurants
map_restaurants = folium.Map(location=[df['Latitude'].mean(), df['Longitude'].mean()], zoom_start=10)

# Calculate the density of data points using Kernel Density Estimation (KDE)
heatmap_data = df[['Latitude', 'Longitude']].values.tolist()

# Create a HeatMap layer with the density data
HeatMap(heatmap_data, radius=10).add_to(map_restaurants)

# Display the map
map_restaurants
```

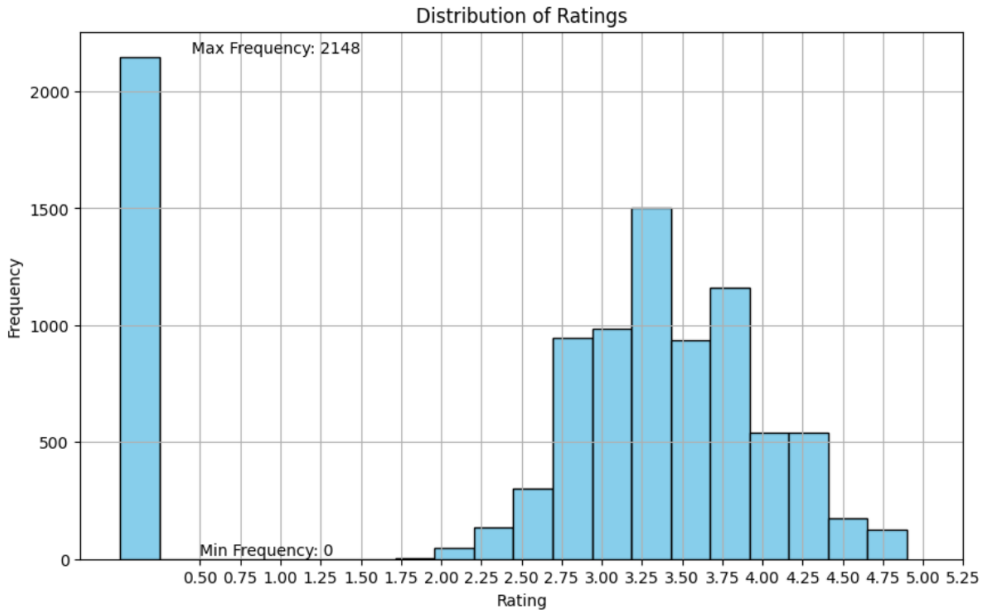
2. Explore how ratings are distributed overall: Max frequency of rating is 0 = 2148 no of restaurants

Restruant Marketing Project.i | Healthcare heart attack.ipynb | Code | Notebook | Python 3 (ipykernel)

Leaflet | © OpenStreetMap contributors

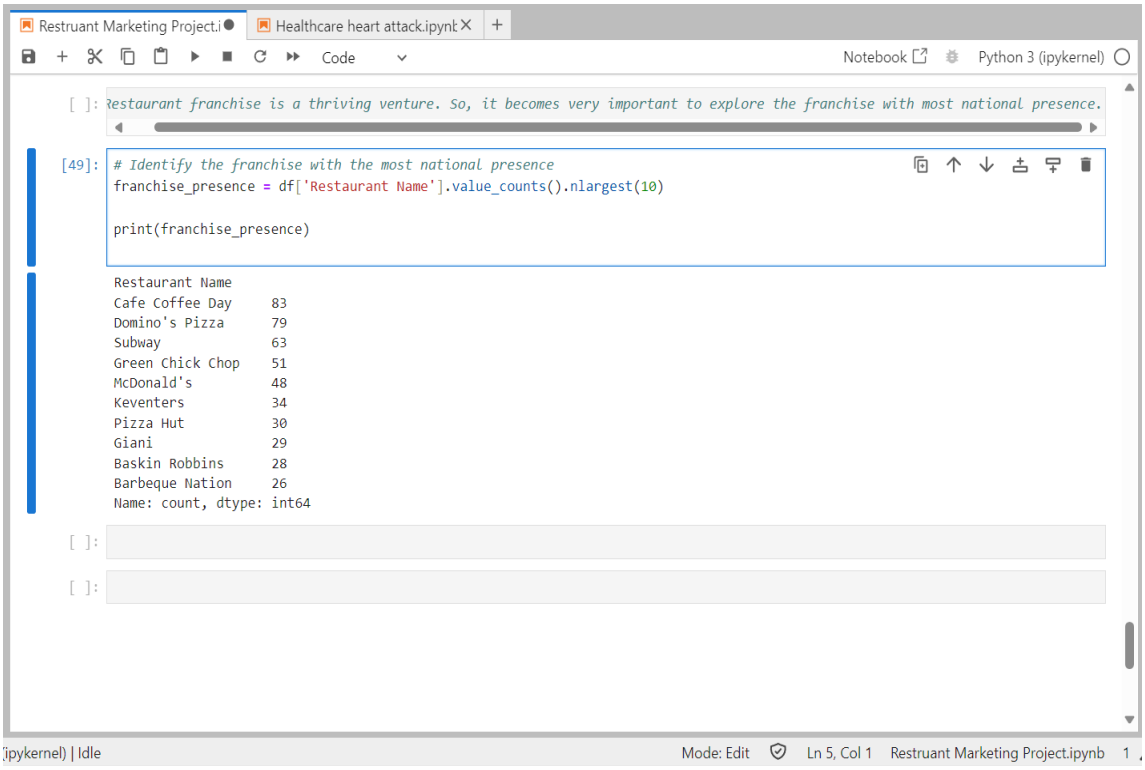
```
[ ]: #Explore how ratings are distributed overall.

•[34]:
import numpy as np
import matplotlib.pyplot as plt
# Visualize the distribution of ratings
plt.figure(figsize=(10, 6))
n, bins, patches = plt.hist(df['Aggregate rating'], bins=20, color='skyblue', edgecolor='black')
plt.xlabel('Rating')
plt.ylabel('Frequency')
plt.title('Distribution of Ratings')
plt.grid(True)
# Add sub-numbers in the x-axis
plt.xticks(np.arange(0.5, 5.5, 0.25))
# Identify max and min frequency values
max_freq = max(n)
min_freq = min(n)
# Annotate max and min frequency values
plt.text(1.5, max_freq, f'Max Frequency: {int(max_freq)}', fontsize=10, va='bottom', ha='right')
plt.text(0.5, min_freq, f'Min Frequency: {int(min_freq)}', fontsize=10, va='bottom', ha='left')
plt.show()
```



3. Identify the franchise with the most national presence: Café coffee Day has most presence so it would be a good franchise to take-up

:



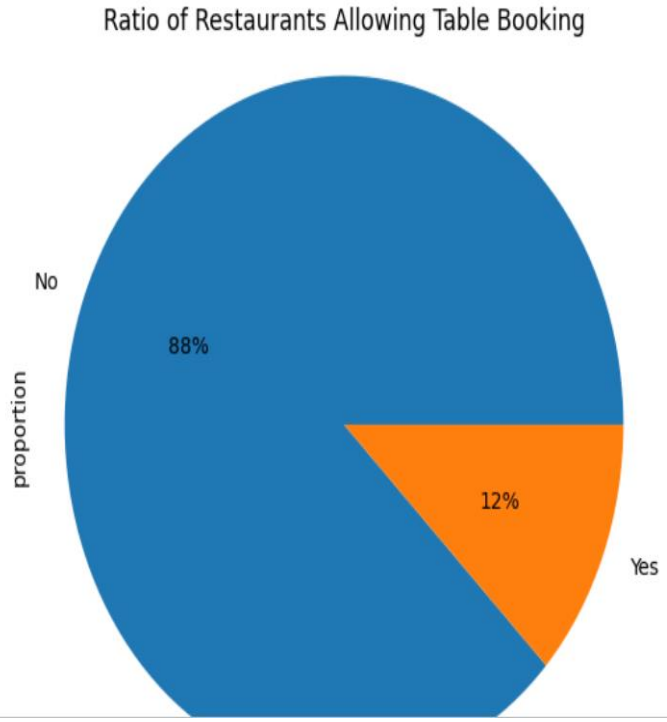
4. Calculate the ratio of restaurants allowing table booking vs those that do not:
88% people are not doing table booking and 12% people are doing table booking

[85]:
```
# Calculate the ratio of restaurants allowing table booking vs those that do not
table_booking_ratio = df['Has Table booking'].value_counts(normalize=True)
table_booking_percentage = table_booking_ratio * 100
print("table_booking_percentage:")
print("table_booking_percentage:")
print(table_booking_percentage.round(2).astype(str) + '%')


# Visualize the ratio
plt.figure(figsize=(6, 6))
table_booking_ratio.plot(kind='pie', autopct='%1.0f%%')
plt.title('Ratio of Restaurants Allowing Table Booking')
plt.axis('equal')

plt.show()
```

```
table_booking_percentage:
table_booking_percentage:
Has Table booking
No     87.88%
Yes    12.12%
Name: proportion, dtype: object
```



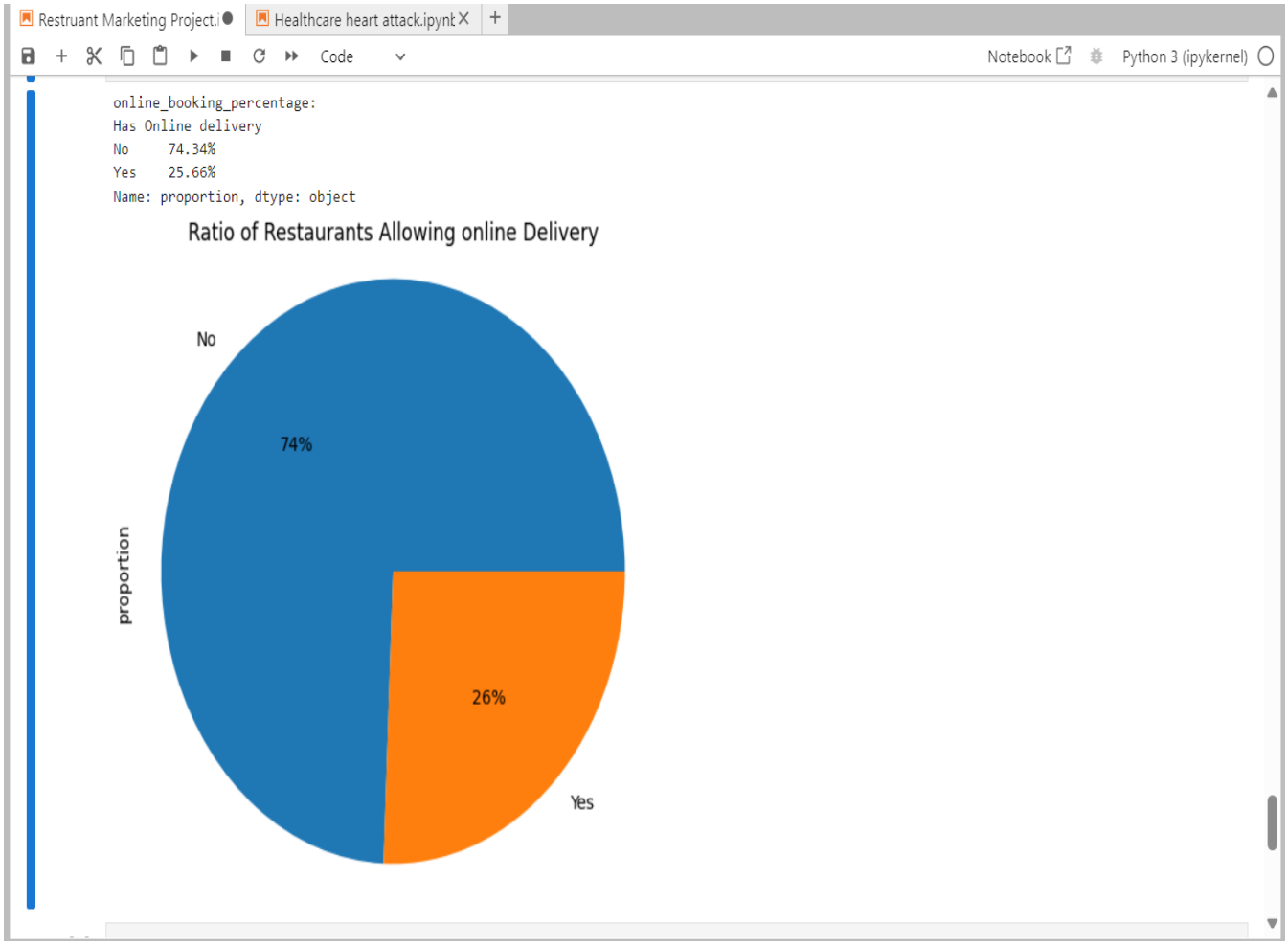
4. What is the percentage of restaurants providing online delivery?: 26% people are doing online booking

```
[88]: # Calculate the ratio of restaurants allowing table booking vs those that do not
online_booking_ratio = df['Has Online delivery'].value_counts(normalize=True)
online_booking_percentage = online_booking_ratio * 100
print("online_booking_percentage:")

print(online_booking_percentage.round(2).astype(str) + '%')


# Visualize the ratio
plt.figure(figsize=(6, 6))
online_booking_ratio.plot(kind='pie', autopct='%1.0f%%')
plt.title('Ratio of Restaurants Allowing online Delivery')
plt.axis('equal')

plt.show()
```



5. Is there a difference in no. of votes for the restaurants that deliver and the restaurant that don't? : online delivery has mean votes of 211 no;'s and no online delivery has 138 no's votes
6. What are the top 10 cuisines served across cities? : North Indian cuisines Ranks the top

[89]:
```
# Compare the average number of votes for restaurants providing delivery vs those that don't
delivery_votes_comparison = df.groupby('Has Online delivery')['Votes'].mean()
print(delivery_votes_comparison)
```

```
Has Online delivery
No     138.131127
Yes    211.307222
Name: Votes, dtype: float64
```

[91]:
```
#What are the top 10 cuisines served across cities?
top_cuisines = df['Cuisines'].value_counts().nlargest(10)
print(top_cuisines)
```

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

7. What is the maximum and minimum no. of cuisines that a restaurant serves? Also, what is the relationship between No. of cuisines served and Ratings: Max- North Indian cuisines, and Min is other

[25]:
```
##Find the maximum and minimum number of cuisines served by any single restaurant

# Find the maximum and minimum number of cuisines served by any single restaurant
max_cuisines_served = df['Cuisines'].str.split(',').str.len().max()
min_cuisines_served = df['Cuisines'].str.split(',').str.len().min()

print("Maximum number of cuisines served:", max_cuisines_served)
print("Minimum number of cuisines served:", min_cuisines_served)
```

```
Maximum number of cuisines served: 8.0
Minimum number of cuisines served: 1.0
```

[26]:
```
# Split the cuisine column by comma and space, and then stack to create a Series
cuisine_series = df['Cuisines'].str.split(', ', expand=True).stack()

# Count the occurrences of each cuisine
cuisine_counts = cuisine_series.value_counts()

# Find the cuisine with the maximum and minimum frequency
max_freq_cuisine = cuisine_counts.idxmax()
min_freq_cuisine = cuisine_counts.idxmin()

print("Cuisine with maximum frequency:", max_freq_cuisine)
print("Cuisine with minimum frequency:", min_freq_cuisine)
```

```
Cuisine with maximum frequency: North Indian
Cuisine with minimum frequency: Cuisine Varies
```

8. Discuss the cost vs the other variables.

[31]:
```
import seaborn as sns

# Create a boxplot to visualize the distribution of ratings for different cost ranges
plt.figure(figsize=(8, 6))
sns.boxplot(x='Price range', y='Aggregate rating', data=df)
plt.xlabel('Price Range')
plt.ylabel('Aggregate Rating')
plt.title('Distribution of Ratings by Price Range')
plt.show()
```

```
C:\Users\nikhi\AppData\Roaming\Python\Python311\site-packages\seaborn\categorical.py:640: FutureWarning: SeriesGroupBy.grouper is deprecated and will be removed in a f
uture version of pandas.
  positions = grouped.grouper.result_index.to_numpy(dtype=float)
```


```
# Create a bar plot to visualize the mean rating for each price range
plt.figure(figsize=(8, 6))
mean_rating_by_price.plot(kind='bar', color='skyblue')
plt.xlabel('Price Range')
plt.ylabel('Mean Aggregate Rating')
plt.title('Mean Rating by Price Range')
plt.xticks(rotation=0)
plt.show()
```

9. Explain the factors in the data that may have an effect on ratings e.g. No. of cuisines, cost, delivery option etc.

[34]:
```
# Replace missing values with 0
df['Has Online delivery'].fillna(0, inplace=True)

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

# Explore factors affecting ratings
factors = ['Average Cost for two', 'Has Online delivery']
for factor in factors:
    print(df[[factor, 'Aggregate rating']].corr())
```

```
                    Average Cost for two  Aggregate rating
Average Cost for two            1.000000          0.051792
Aggregate rating                0.051792          1.000000
                    Has Online delivery  Aggregate rating
Has Online delivery            1.000000          0.225699
Aggregate rating               0.225699          1.000000
```

Let's interpret the correlation results:

Average Cost for Two and Aggregate Rating: The correlation coefficient between 'Average Cost for Two' and 'Aggregate Rating' is approximately 0.0518. This value is close to zero, indicating a very weak positive correlation between the average cost for two people dining at a restaurant and the aggregate rating. In other words, there is almost no linear relationship between the average cost and the rating.

Has Online Delivery and Aggregate Rating: The correlation coefficient between 'Has Online Delivery' and 'Aggregate Rating' is approximately 0.2257. This value indicates a weak positive correlation between whether a restaurant offers online delivery and its aggregate rating. However, the correlation is still relatively low, suggesting that the presence of online delivery services is only mildly associated with higher ratings.