LEVEL 3 : TASK 1 : RESTAURANT REVIEWS

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

In [2]:
```
df=pd.read_csv('Dataset .csv',encoding='unicode_escape')
```

In [3]:
```
df.head()
```

Out[3]:

	ï»¿Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	C
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	...	B
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	...	B
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	...	B
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	...	B
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	...	B

5 rows × 21 columns

In [4]:
```
df.shape
```

Out[4]:
```
(9551, 21)
```

In [5]:
```
pd.isnull(df).sum()
```

Out[5]:
```
ï»¿Restaurant ID        0
Restaurant Name         0
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
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
```

In [6]:
```
df.dropna(inplace=True)
```

In [7]:
```
df.shape
```

Out[7]:
```
(9542, 21)
```

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 9542 entries, 0 to 9550
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   ï»¿Restaurant ID      9542 non-null   int64  
 1   Restaurant Name       9542 non-null   object 
 2   Country Code          9542 non-null   int64  
 3   City                  9542 non-null   object 
 4   Address               9542 non-null   object 
 5   Locality              9542 non-null   object 
 6   Locality Verbose      9542 non-null   object 
 7   Longitude             9542 non-null   float64
 8   Latitude              9542 non-null   float64
 9   Cuisines              9542 non-null   object 
 10  Average Cost for two  9542 non-null   int64  
 11  Currency              9542 non-null   object 
 12  Has Table booking     9542 non-null   object 
 13  Has Online delivery   9542 non-null   object 
 14  Is delivering now     9542 non-null   object 
 15  Switch to order menu  9542 non-null   object 
 16  Price range           9542 non-null   int64  
 17  Aggregate rating      9542 non-null   float64
 18  Rating color          9542 non-null   object 
 19  Rating text           9542 non-null   object 
 20  Votes                 9542 non-null   int64  
dtypes: float64(3), int64(5), object(13)
memory usage: 1.6+ MB
```

In [9]:
```
df.describe()
```

Out[9]:

	ï»¿Restaurant ID	Country Code	Longitude	Latitude	Average Cost for two	Price range	Aggregate rating	Votes
count	9.542000e+03	9542.000000	9542.000000	9542.000000	9542.000000	9542.000000	9542.000000	9542.000000
mean	9.043301e+06	18.179208	64.274997	25.848532	1200.326137	1.804968	2.665238	156.772060
std	8.791967e+06	56.451600	41.197602	11.010094	16128.743876	0.905563	1.516588	430.203324
min	5.300000e+01	1.000000	-157.948486	-41.330428	0.000000	1.000000	0.000000	0.000000
25%	3.019312e+05	1.000000	77.081565	28.478658	250.000000	1.000000	2.500000	5.000000
50%	6.002726e+06	1.000000	77.192031	28.570444	400.000000	2.000000	3.200000	31.000000
75%	1.835260e+07	1.000000	77.282043	28.642711	700.000000	2.000000	3.700000	130.000000
max	1.850065e+07	216.000000	174.832089	55.976980	800000.000000	4.000000	4.900000	10934.000000

ANALYZE THE TEXT REVIEWS TO IDENTIFY THE MOST COMMON POSITIVE AND NEGATIVE KEYWORDS.

In [4]:
```
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
df=pd.read_csv('Dataset .csv',encoding='unicode_escape')
```

In [10]:
```
df.columns
```

Out[10]:
```
Index(['ï»¿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')
```

In [5]:
```
df['Rating text'].value_counts()
```

Out[5]:
```
Rating text
Average      3737
Not rated    2148
Good         2100
Very Good    1079
Excellent     301
Poor          186
Name: count, dtype: int64
```

In [6]:
```
ax=sns.countplot(x='Rating text',data=df)

for bars in ax.containers:
    ax.bar_label(bars)
```

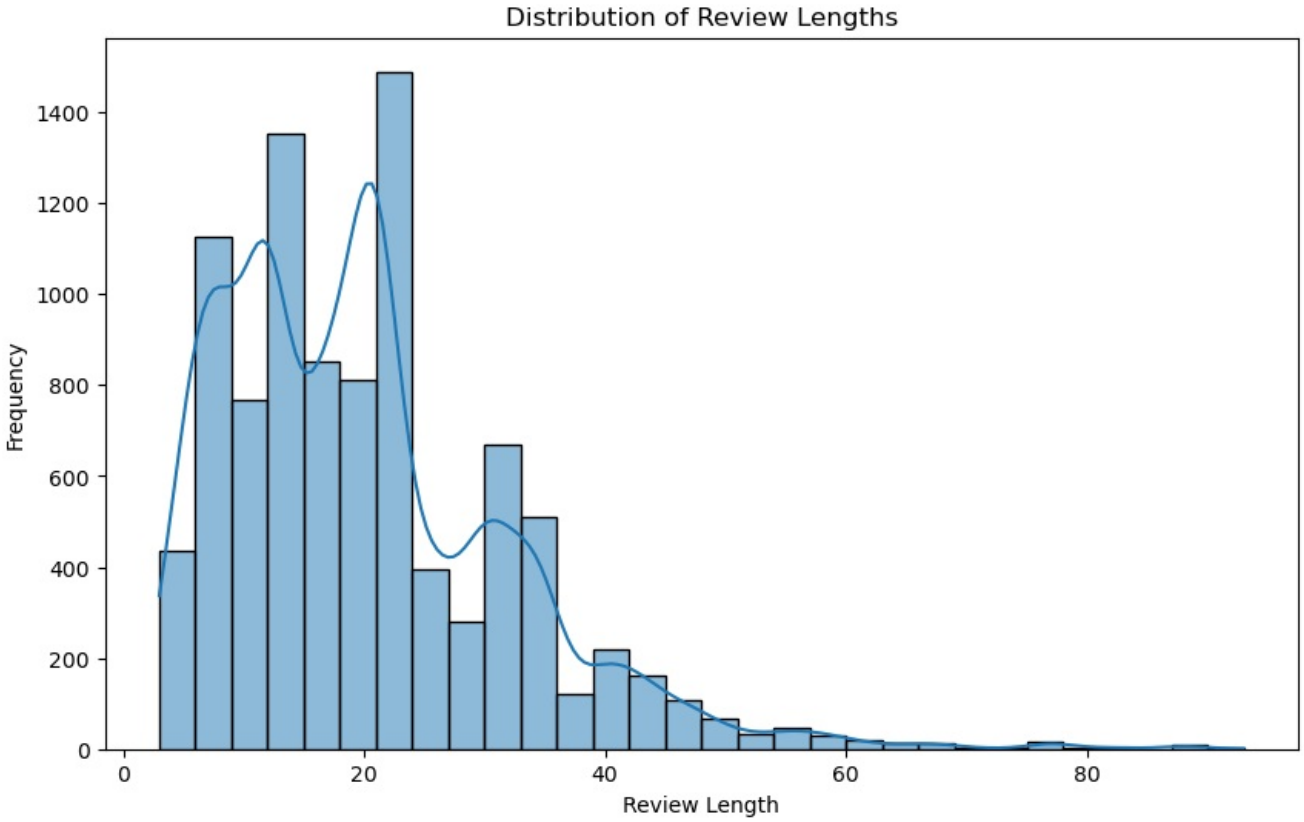
CALCULATE THE AVERAGE LENGTH OF REVIEWS AND EXPLORE IF THERE IS A RELATIONSHIP BETWEEN REVIEW LENGTH AND RATING.

In [8]:
```
v=df['Aggregate rating'].value_counts().mean()
v
```

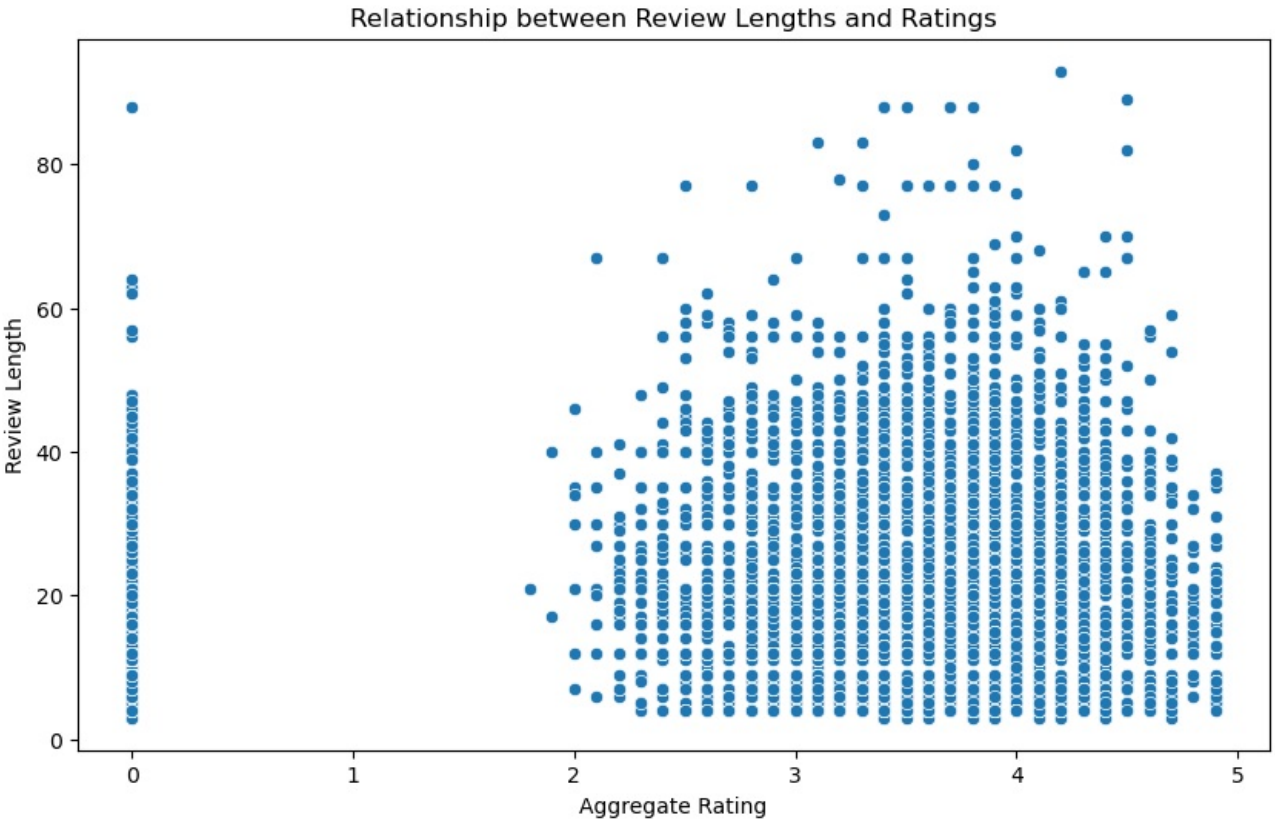
Out[8]:
```
289.42424242424244
```

In [12]:
```
#Step 1: Calculate review length distribution (using 'Cuisines' as a proxy)
df['Review Length']=df['Cuisines'].apply(lambda x:len(str(x)))
#Step 2: Explore Review length distribution
plt.figure(figsize=(10,6))
sns.histplot(data=df,x='Review Length',bins=30,kde=True)
plt.title('Distribution of Review Lengths')
plt.xlabel('Review Length')
plt.ylabel('Frequency')
plt.show()
#Step 3: Calculate average review length
average_review_length=df['Review Length'].mean()
print(f'Average Review Length: {average_review_length:.2f}characters')
#Step 4: Explore Relationship with Ratings
plt.figure(figsize=(10,6))
sns.scatterplot(data=df,x='Aggregate rating',y='Review Length')
plt.title('Relationship between Review Lengths and Ratings')
```

```
plt.xlabel('Aggregate Rating')
plt.ylabel('Review Length')
plt.show()
#Step 5: Statistical analysis
correlation_coefficient=df[['Aggregate rating','Review Length']].corr().iloc[0,1]
print(f'Correlation coefficient: {correlation_coefficient:.2f}')
#Step 6: Statistical analysis (optional)
from scipy.stats import pearsonr
correlation, p_value=pearsonr(df['Aggregate rating'],df['Review Length'])
print(f'correlation:{correlation:.2f},p_value:{p_value:.4f}')
```




```
Average Review Length: 19.91characters
```




```
Correlation coefficient: 0.19
correlation:0.19,p_value:0.0000
```

LEVEL 3 : TASK 2 : VOTES ANALYSIS

IDENTIFY THE RESTAURANTS WITH THE HIGHEST AND LOWEST NUMBER OF VOTES.

In [15]:
```
Restaurant_votes=df.groupby(['Restaurant Name'])['Votes'].mean()
print(Restaurant_votes.max())
print(Restaurant_votes.min())
```

```
10934.0
0.0
```

In [16]:
```
Restaurant_highest_votes=df.loc[df['Votes'].idxmax()]
print('Restaurant with highest vote number is:')
print(Restaurant_highest_votes[['Restaurant Name','Votes']])
print('\n')
Restaurant_lowest_votes=df.loc[df['Votes'].idxmin()]
print('Restaurant with lowest vote number is:')
print(Restaurant_lowest_votes[['Restaurant Name','Votes']])
```

```
Restaurant with highest vote number is:
Restaurant Name     Toit
Votes              10934
Name: 728, dtype: object


Restaurant with lowest vote number is:
Restaurant Name    Cantinho da Gula
Votes                             0
Name: 69, dtype: object
```

ANALYZE IF THERE IS A CORRELATION BETWEEN THE NUMBER OF VOTES AND THE RATING OF A RESTAURANT.

In [17]:
```
df[['Votes','Aggregate rating']].head()
```

Out[17]:

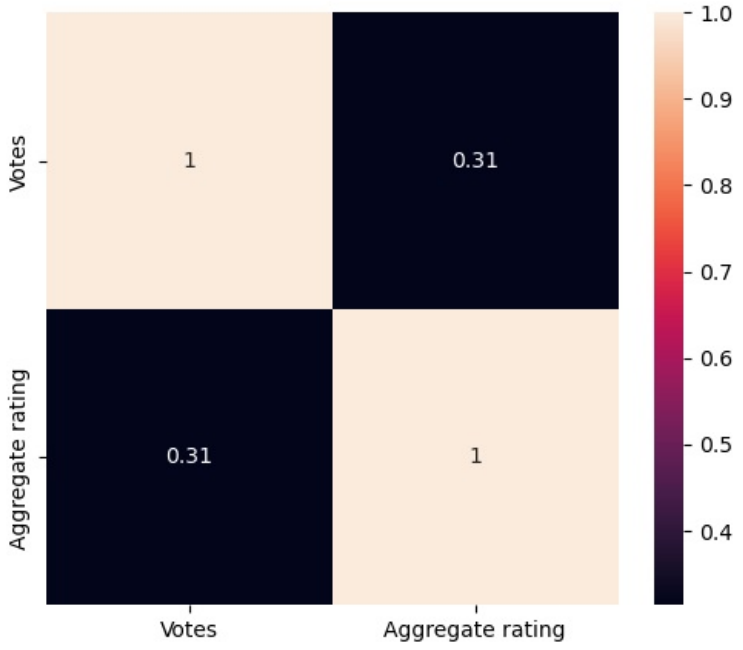
	Votes	Aggregate rating
0	314	4.8
1	591	4.5
2	270	4.4
3	365	4.9
4	229	4.8

In [18]:
```
correlation=df[['Votes','Aggregate rating']].corr()
correlation
```

Out[18]:

	Votes	Aggregate rating
Votes	1.000000	0.313691
Aggregate rating	0.313691	1.000000

In [21]:
```
plt.figure(figsize=(6,5))
sns.heatmap(correlation,annot=True)
plt.show()
```



LEVEL 3 : TASK 3 : PRICE RANGE VS ONLINE DELIVERY AND TABLE BOOKING.

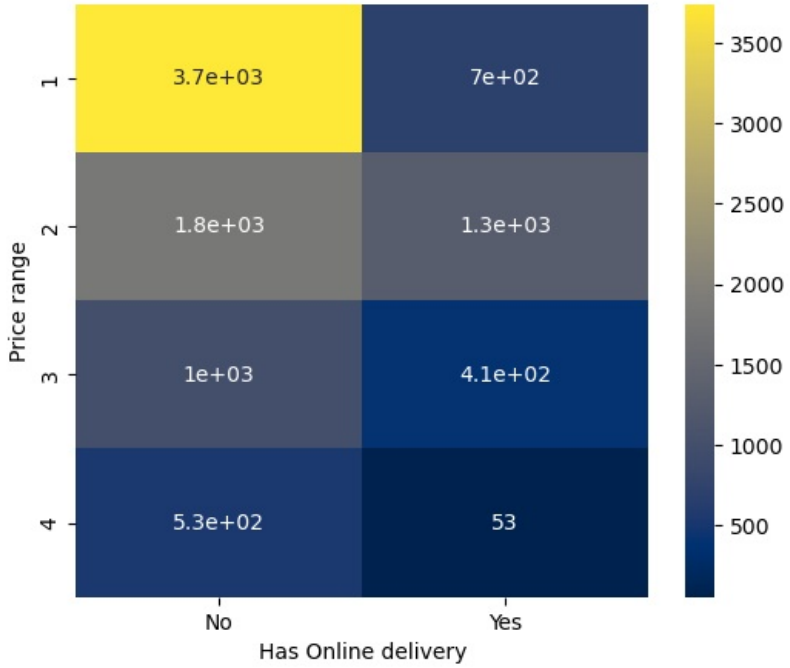
ANALYZE IF THERE IS A RELATIONSHIP BETWEEN THE PRICE RANGE AND THE AVAILABILITY OF ONLINE DELIVERY AND TABLE BOOKING.

In [22]:
```
cross_tab=pd.crosstab(df['Price range'],df['Has Online delivery'])
cross_tab.head()
```

Out[22]:

Has Online delivery	No	Yes
Price range		
1	3743	701
2	1827	1286
3	997	411
4	533	53

In [23]:
```
plt.figure(figsize=(6,5))
sns.heatmap(cross_tab,annot=True,cmap='cividis')
plt.show()
```



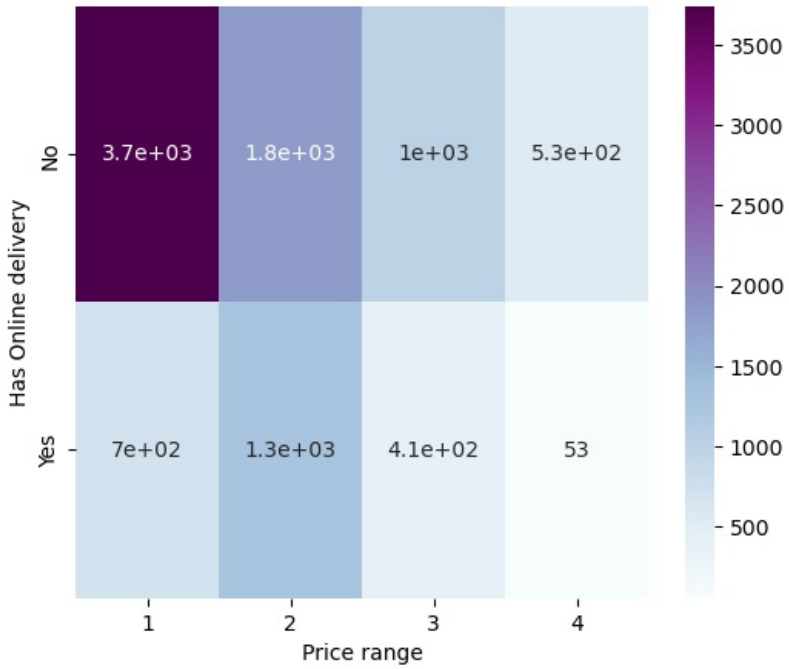
In [25]:
```
df['Has Table booking'].unique()
```

Out[25]:
```
array(['Yes', 'No'], dtype=object)
```

In [26]:
```
df['Has Table booking'].unique()
df.groupby('Has Online delivery')['Price range'].mean()
```

Out[26]:
```
Has Online delivery
No     1.763380
Yes    1.924929
Name: Price range, dtype: float64
```

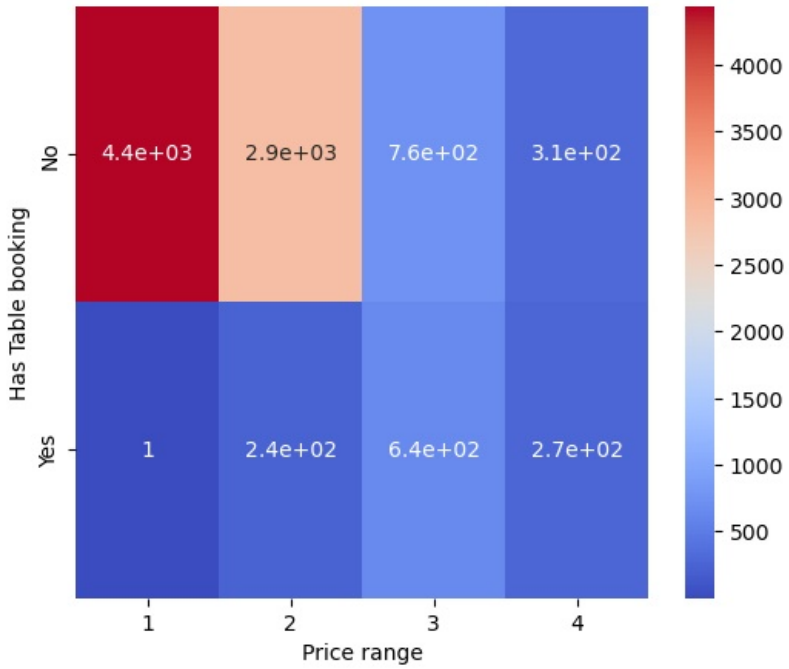
In [28]:
```
crosstab=pd.crosstab(df['Has Online delivery'],df['Price range'])
plt.figure(figsize=(6,5))
sns.heatmap(crosstab,annot=True,cmap='BuPu')
plt.show()
```



In [29]:
```
df.groupby('Has Table booking')['Price range'].mean()
```

Out[29]:
```
Has Table booking
No     1.636006
Yes    3.028497
Name: Price range, dtype: float64
```

In [31]:
```
crosstab=pd.crosstab(df['Has Table booking'],df['Price range'])
plt.figure(figsize=(6,5))
sns.heatmap(crosstab,annot=True,cmap='coolwarm')
plt.show()
```



DETERMINE IF HIGHER- PRICED RESTAURANTS ARE MORE LIKELY TO OFFER THESE SERVICES.

In [32]:
```
df['Price range'].max()
```

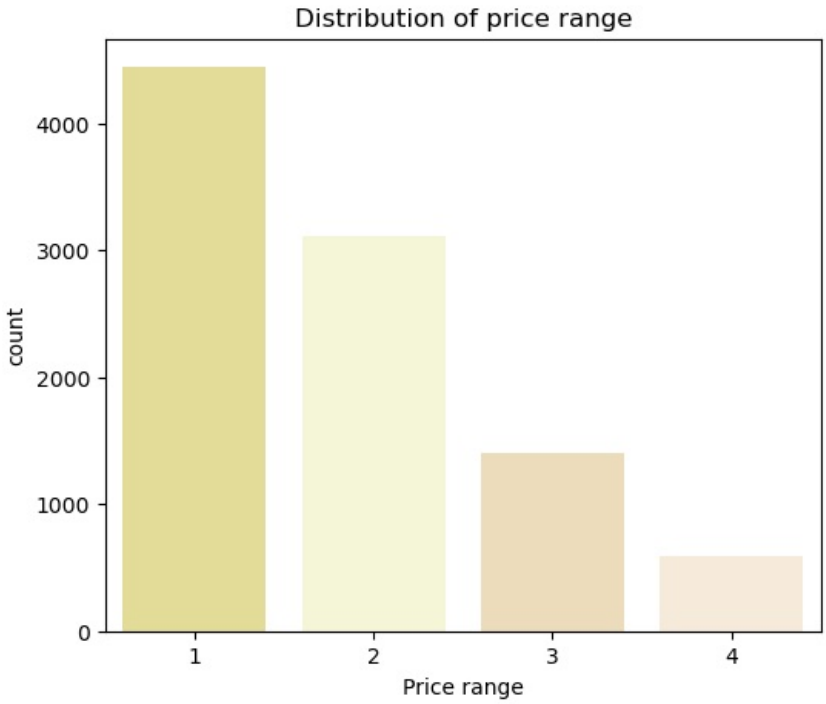
Out[32]:
```
4
```

In [33]:
```
df['Has Online delivery'].value_counts()
```

Out[33]:
```
Has Online delivery
No     7100
Yes    2451
Name: count, dtype: int64
```

In [35]:
```
plt.figure(figsize=(6,5))
custom_palette=['khaki','lightgoldenrodyellow','wheat','antiquewhite']
sns.countplot(x='Price range',data=df,palette=custom_palette)
```

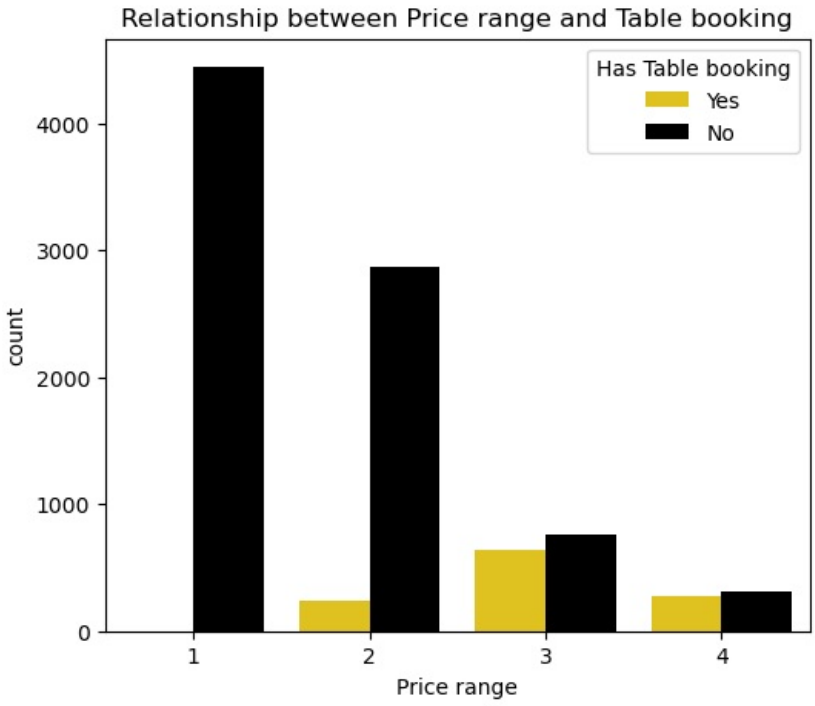
```
plt.title('Distribution of price range')
plt.xlabel('Price range')
plt.ylabel('count')
plt.show()
```



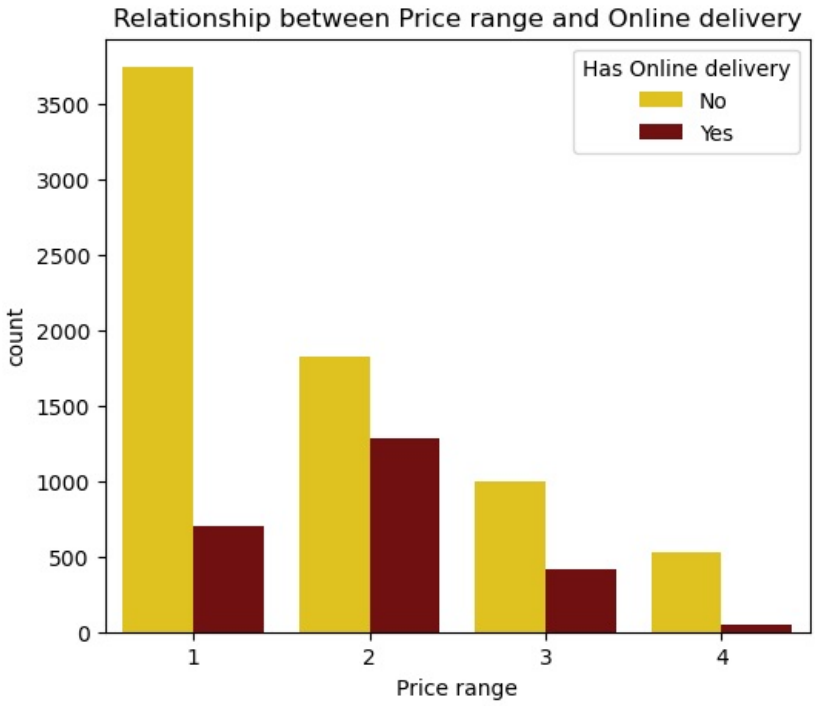
In [36]:
```
table_booking_proportion=df['Has Table booking'].value_counts(normalize=True)
online_delivery_proportion=df['Has Online delivery'].value_counts(normalize=True)
print('Proportion of restaurants offerring table booking:')
print(table_booking_proportion)
print('Proportion of restaurants offerring online delivery:')
print(online_delivery_proportion)
```

```
Proportion of restaurants offerring table booking:
Has Table booking
No     0.878756
Yes    0.121244
Name: proportion, dtype: float64
Proportion of restaurants offerring online delivery:
Has Online delivery
No     0.743378
Yes    0.256622
Name: proportion, dtype: float64
```

In [39]:
```
plt.figure(figsize=(13,5))
plt.subplot(1,2,1)
custom_palette=['gold','black']
sns.countplot(x='Price range',hue='Has Table booking' ,data=df,palette=custom_palette)
plt.title('Relationship between Price range and Table booking')
plt.show()
```



In [40]:
```
plt.figure(figsize=(13,5))
plt.subplot(1,2,1)
custom_palette=['gold','maroon']
sns.countplot(x='Price range',hue='Has Online delivery' ,data=df,palette=custom_palette)
plt.title('Relationship between Price range and Online delivery')
plt.show()
```



THANKYOU!

CONNECT WITH ME:

LinkedIn: https://www.linkedin.com/in/harshita-sharma-b68154220/

GitHub: https://github.com/DATAPREDICTS

Instagram: https://www.instagram.com/datapredicts?utm_source=qr&igsh=czVzc2k5c3oxOWQ4

YouTube: https://youtube.com/@Datapredicts?si=eDKAqVciVxg23zab