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

In [2]:
```
df = pd.read_csv("Dataset .csv",encoding = 'UTF-8-SIG')
```

In [3]:
```
df.head(10) #using head(10) Top 10 rows are visible,head() by default 5 can be seer
```

Out[3]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longit
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...	121.027
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...	121.014
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...	121.056
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...	121.056
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...	121.057
5	18189371	Din Tai Fung	162	Mandaluyong City	Ground Floor, Mega Fashion Hall, SM Megamall, ...	SM Megamall, Ortigas, Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City, Mandal...	121.056
6	6300781	Buffet 101	162	Pasay City	Building K, SM By The Bay, Sunset Boulevard, M...	SM by the Bay, Mall of Asia Complex, Pasay City	SM by the Bay, Mall of Asia Complex, Pasay Cit...	120.979
7	6301290	Vikings	162	Pasay City	Building B, By The Bay, Seaside Boulevard, Mal...	SM by the Bay, Mall of Asia Complex, Pasay City	SM by the Bay, Mall of Asia Complex, Pasay Cit...	120.979
8	6300010	Spiral - Sofitel Philippine Plaza Manila	162	Pasay City	Plaza Level, Sofitel Philippine Plaza Manila, ...	Sofitel Philippine Plaza Manila, Pasay City	Sofitel Philippine Plaza Manila, Pasay City, P...	120.980
9	6314987	Locavore	162	Pasig City	Brixton Technology	Kapitolyo	Kapitolyo, Pasig City	121.056

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longit
					Center, 10 Brixton Street, ...			

10 rows × 21 columns

Level 2 (Task-1) Restaurant rating

1. Analyze the distribution of aggregate ratings and determine the most common rating range.

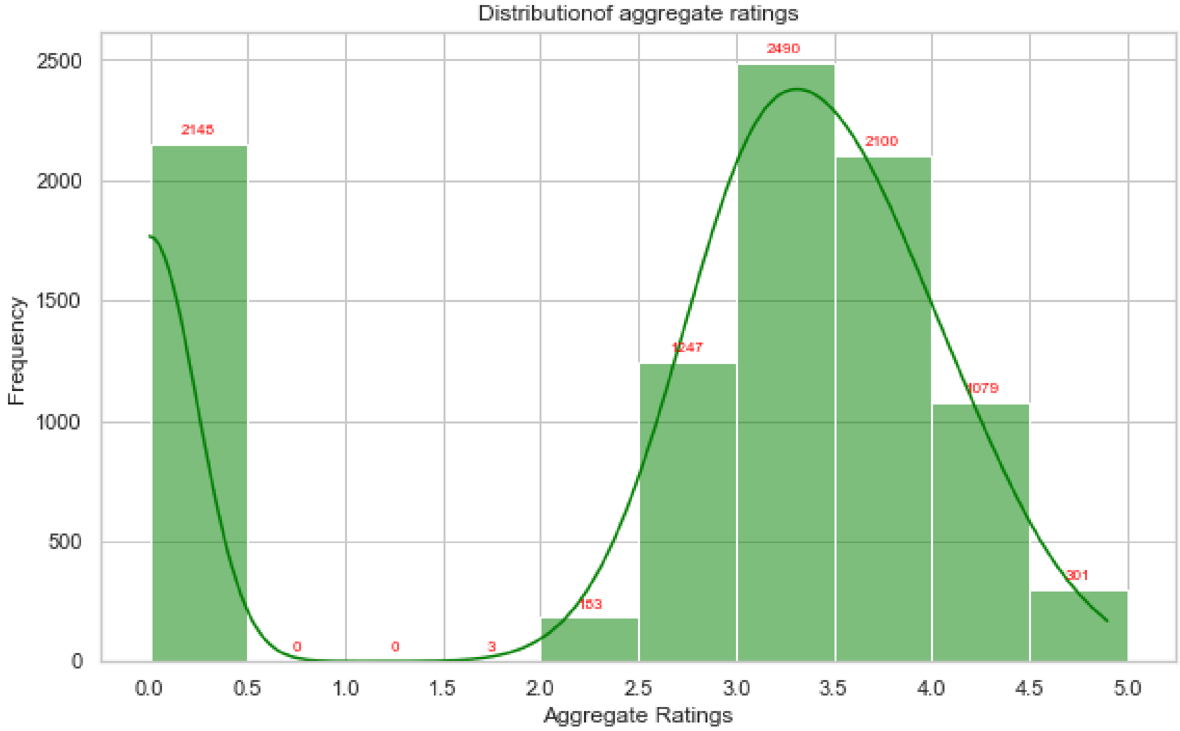
In [4]:
```
Rating_distribution = df['Aggregate rating'].value_counts().sort_index()
```

In [5]:
```
Rating_distribution.head()
```

Out[5]:
```
0.0    2148
1.8       1
1.9       2
2.0       7
2.1      15
Name: Aggregate rating, dtype: int64
```

Creating histogram of aggregate rating

In [6]:
```
ratings = df['Aggregate rating']

sns.set(style ="whitegrid")

plt.figure(figsize=(10,6))


ax = sns.histplot(ratings,binwidth = 0.5,kde = True,color = 'green')


plt.title('Distributionof aggregate ratings')
plt.xlabel('Aggregate Ratings')
plt.ylabel('Frequency')

for p in ax.patches:
    ax.annotate(f'{int(p.get_height())}',(p.get_x()+p.get_width()/2,p.get_height())
                ha = 'center',va = 'baseline',fontsize = 8,color= 'red',xytext=(0,5

plt.xticks(np.arange(min(ratings),max(ratings)+0.5,0.5))

plt.show()
```



We can see that most common rating in 3.0-3.5

2.Calculate the average number of votes received by restaurants.

In [7]:
```
average_vote = df['Votes'].mean()
round(average_vote )
```

Out[7]:
```
157
```

Task-2 Cousine combination

3.Identify most common combination of cuisines in the data set

In [8]:
```
cuisine_counts = df['Cuisines'].value_counts().sort_values()
```

In [9]:
```
cuisine_counts.tail(10)
```

Out[9]:
```
Street Food                       149
Bakery, Desserts                  170
North Indian, Mughlai, Chinese    197
Bakery                            218
Cafe                              299
North Indian, Mughlai             334
Fast Food                         354
Chinese                           354
North Indian, Chinese             511
North Indian                      936
Name: Cuisines, dtype: int64
```

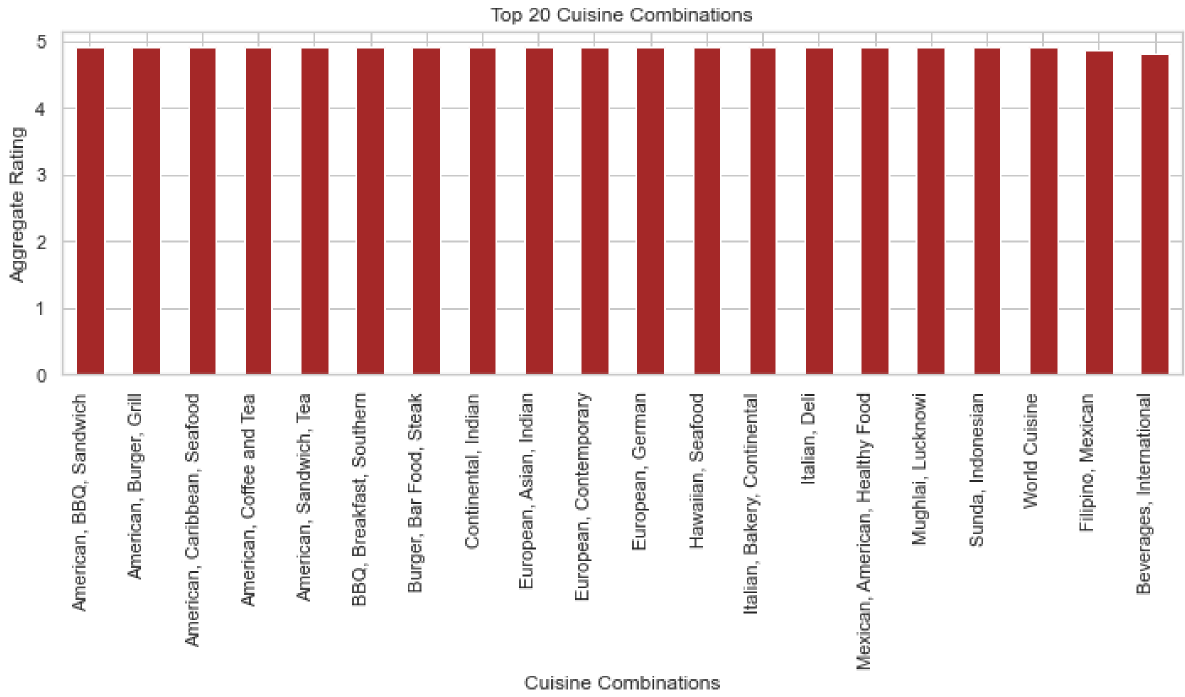
Most common cuisine combinations are Nort Indian , North Indian Chinese,Chinese.

4. Determin if certain cuisine combination tend to have higher rating.

In [10]:
```
cuisine_comb_avg_rating = df.groupby("Cuisines")['Aggregate rating'].mean()
higher_rated_cuisines_combinations = cuisine_comb_avg_rating.nlargest(20)
higher_rated_cuisines_combinations
```

Out[10]:
```
Cuisines
American, BBQ, Sandwich            4.90
American, Burger, Grill            4.90
American, Caribbean, Seafood       4.90
American, Coffee and Tea           4.90
American, Sandwich, Tea            4.90
BBQ, Breakfast, Southern           4.90
Burger, Bar Food, Steak            4.90
Continental, Indian                4.90
European, Asian, Indian            4.90
European, Contemporary             4.90
European, German                   4.90
Hawaiian, Seafood                  4.90
Italian, Bakery, Continental       4.90
Italian, Deli                      4.90
Mexican, American, Healthy Food    4.90
Mughlai, Lucknowi                  4.90
Sunda, Indonesian                  4.90
World Cuisine                      4.90
Filipino, Mexican                  4.85
Beverages, International           4.80
Name: Aggregate rating, dtype: float64
```

In [11]:
```
ax = higher_rated_cuisines_combinations.head(20).plot(kind = 'bar',color = 'brown',
plt.xlabel('Cuisine Combinations')
plt.ylabel('Aggregate Rating')
plt.title('Top 20 Cuisine Combinations')
plt.xticks(rotation = 90, ha='right')
plt.tight_layout(rect = [0,0,1,1.20])
```



In [12]:
```
df['Country Code'].value_counts()
```

Out[12]:
```
1      8652
216     434
215      80
30       60
214      60
189      60
148      40
208      34
14       24
162      22
94       21
184      20
166      20
191      20
37        4
Name: Country Code, dtype: int64
```

In [13]:
```
restaurant_df = pd.DataFrame({'Longitude':df['Longitude'],'Latitude':df['Latitude']
```

In [14]:
```
sns.set(style="darkgrid")

unique_country_codes = restaurant_df['Country_Code'].unique()
color_palette = sns.color_palette("colorblind", n_colors=len(unique_country_codes))

plt.figure(figsize=(8, 6))

scatter_all = sns.scatterplot(x='Longitude', y='Latitude', data=restaurant_df, hue=

plt.title('Restaurant Locations')
plt.xlabel('Longitude')
plt.ylabel('Latitude')

# Retrieve the legend from the current axes and set font size for its title
plt.legend(title='Country Code', fontsize=10, title_fontsize='10')

plt.grid(True)
```

From above chart we can easily find clusters of country code 1,14 and 216

Task 4 of level 2 Restaurant Chains

5. Identify whether there are anly restaurant chains present in the dataset(more that 2 outlet)

In [15]:
```
restaurant_counts = df['Restaurant Name'].value_counts()

chain_restaurants = df[df['Restaurant Name'].isin(restaurant_counts[restaurant_cour

chain_restaurants['Restaurant Name'].value_counts()
```

Out[15]:
```
Cafe Coffee Day              83
Domino's Pizza               79
Subway                       63
Green Chick Chop             51
McDonald's                   48
                             ..
D�_ner Grill                  3
Cafe Delhi Heights           3
Chawla's Tandoori Junction   3
Changezi Chicken             3
Bikkgane Biryani             3
Name: Restaurant Name, Length: 266, dtype: int64
```

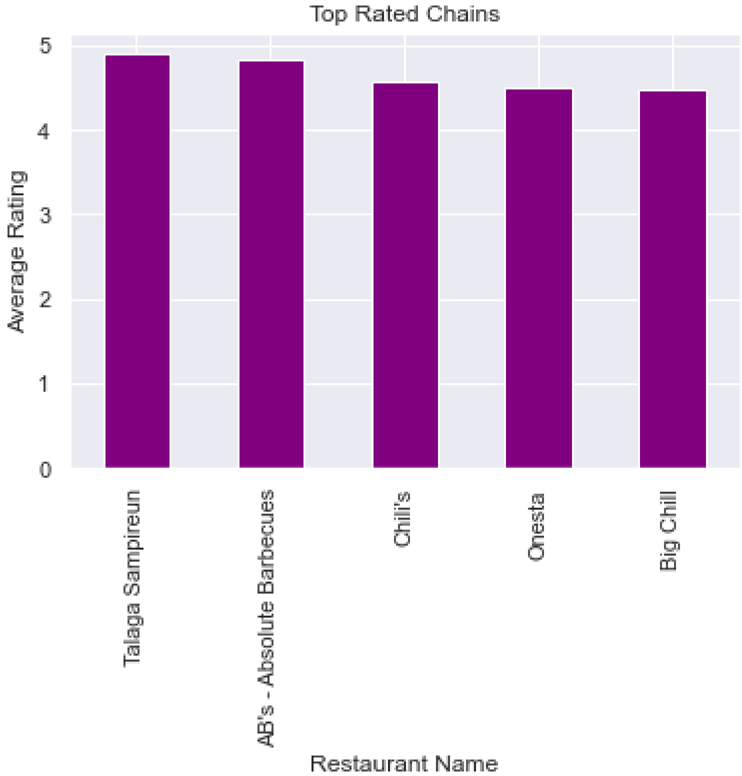
6. Analyse rating and popularity of cahin restaurant

In [16]:
```
chain_ratings = chain_restaurants.groupby('Restaurant Name')['Aggregate rating'].me
```

In [17]:
```
top_rated_chains = chain_ratings.nlargest(5)
print(round(top_rated_chains,2))
```

```
Restaurant Name
Talaga Sampireun           4.90
AB's - Absolute Barbecues  4.82
Chili's                    4.58
Onesta                     4.50
Big Chill                  4.47
Name: Aggregate rating, dtype: float64
```

In [18]:
```
top_rated_chains.plot(kind = 'bar',color = 'purple',title = 'Top Rated Chains')
plt.xlabel('Restaurant Name')
plt.ylabel('Average Rating')
plt.show()
```



In [19]:
```
chain_popularity = chain_restaurants.groupby('Restaurant Name')['Votes'].mean()
```

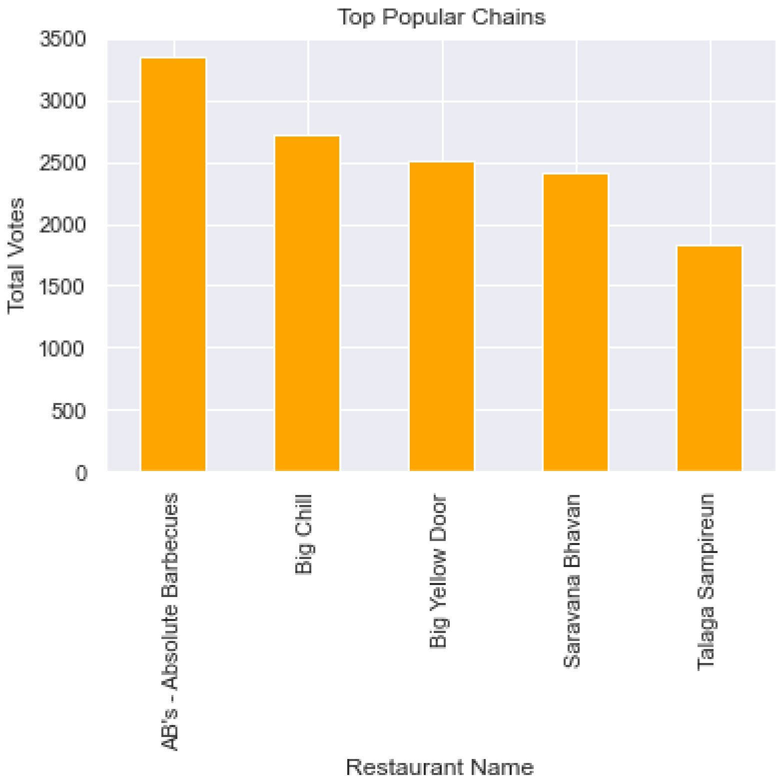
In [20]:
```
#top popular chains based on votes


top_popular_chains =chain_popularity.nlargest(5)
```

In [21]:
```
top_popular_chains
```

Out[21]:
```
Restaurant Name
AB's - Absolute Barbecues  3350.000000
Big Chill                  2713.250000
Big Yellow Door            2503.666667
Saravana Bhavan            2412.666667
Talaga Sampireun           1838.000000
Name: Votes, dtype: float64
```

In [22]:
```
top_popular_chains.plot(kind = 'bar',color = 'orange',title = 'Top Popular Chains')
plt.xlabel('Restaurant Name')
```

```
plt.ylabel('Total Votes')
plt.show()
```



LEVEL 3

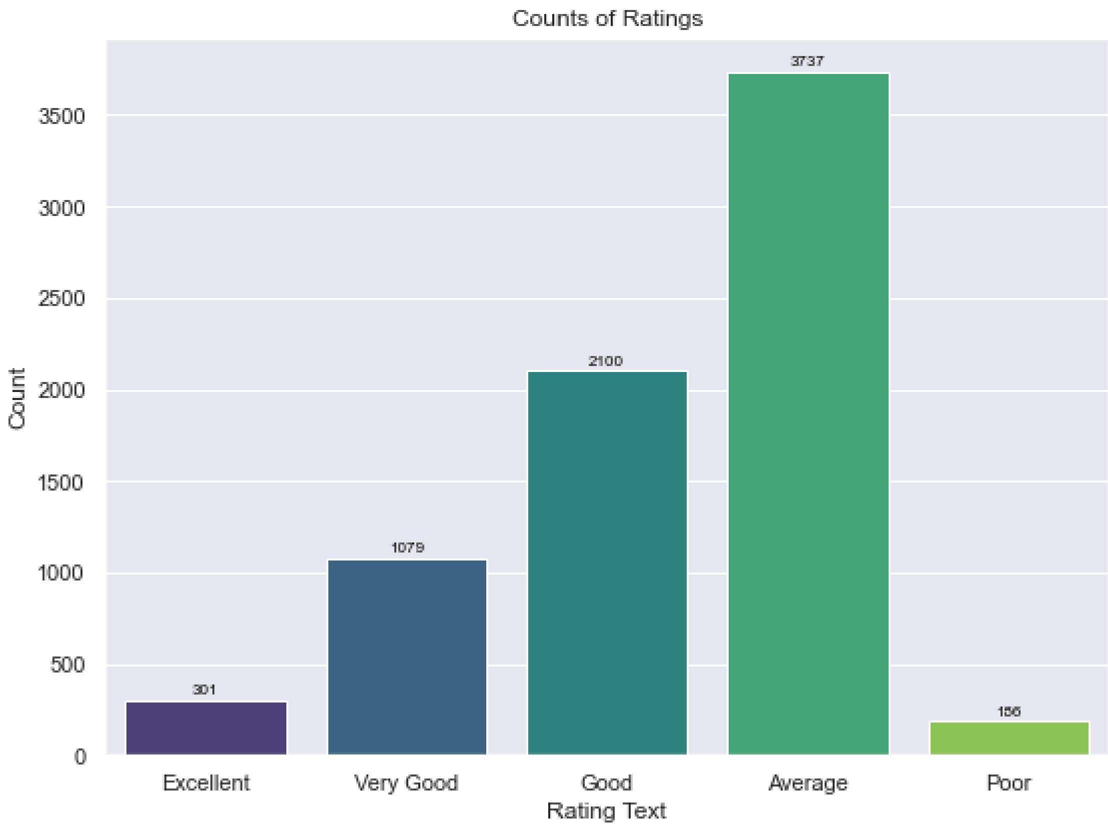
1. To identify the most common positive and negative keywords

In [23]:
```
df_modified = df[df['Rating text'] != 'Not rated']
```

In [24]:
```
df_modified['Rating text'].value_counts()
```

Out[24]:
```
Average      3737
Good         2100
Very Good    1079
Excellent     301
Poor          186
Name: Rating text, dtype: int64
```

In [25]:
```
plt.figure(figsize = (8,6))
sns.countplot(x = 'Rating text',data = df_modified, palette = 'viridis')
plt.title('Counts of Ratings')
plt.xlabel('Rating Text')
plt.ylabel('Count')

for p in plt.gca().patches:
    height = p.get_height()
    plt.gca().annotate(f'{int(height)}',(p.get_x()+p.get_width()/2,height),ha = 'ce
                       textcoords = 'offset points',fontsize = 8,color = 'black')
plt.tight_layout()
```



Conclusion: Most Common positive keywords is "Good" and Most common negative Keyword is Poor

2. To calculate the average length of reviews

In [26]:
```
df_modified['Review_Length'] = df["Rating text"].apply(len)
average_review_length = df_modified['Review_Length'].mean()

round(average_review_length)
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_5452\440985195.py:1: SettingWithCopyWa
rning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  df_modified['Review_Length'] = df["Rating text"].apply(len)
```

Out[26]: 6

3. Explore if there is a relationship between review lengthand rating

In [27]:
```
print(df['Aggregate rating'].isnull().sum())
```

```
0
```

In [28]:
```
max_votes_index = df['Votes'].idxmax()
restaurant_with_highest_votes = df.loc[max_votes_index, 'Restaurant Name']
```

In [29]:
```
restaurant_with_highest_votes
```
Out[29]:
```
'Toit'
```

In [30]:
```
highest_votes_counts = df.loc[max_votes_index,'Votes']
```

In [31]:
```
highest_votes_counts
```
Out[31]:
```
10934
```

In [33]:
```
 min_votes_index = df['Votes'].idxmin()
restaurant_with_lowest_votes = df.loc[min_votes_index,'Restaurant Name']
```

In [34]:
```
restaurant_with_lowest_votes
```
Out[34]:
```
'Cantinho da Gula'
```

In [35]:
```
lowest_votes_count = df.loc[min_votes_index,'Votes']
```

In [36]:
```
lowest_votes_count
```
Out[36]:
```
0
```

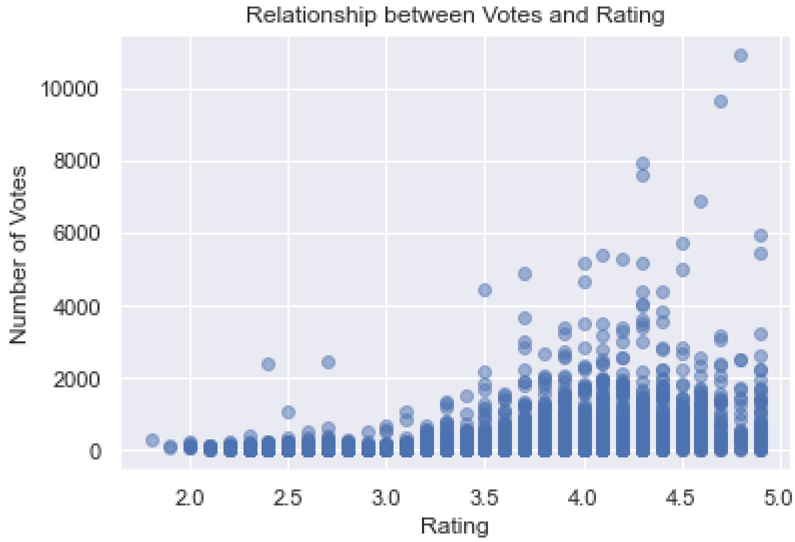
4. Analyze if there is a correlation between the number of votes and the rating of a restaurant

In [37]:
```
df_filtered = df[(df['Votes'] != 0) & (df['Aggregate rating'] != 0)]
```

In [38]:
```
from scipy.stats import pearsonr

correlation_coefficient, _ = pearsonr(df_filtered['Votes'], df_filtered['Aggregate
```

In [39]:
```
print(f"Correlatioon Coefficient between Votes and Rating:{correlation_coefficient:
```
```
Correlatioon Coefficient between Votes and Rating:0.409018
```

In [40]:
```
plt.scatter(df_filtered['Aggregate rating'],df_filtered['Votes'],alpha=0.5)
plt.title('Relationship between Votes and Rating')
plt.xlabel('Rating')
plt.ylabel('Number of Votes')
plt.show()
```



Conclusion: a correlation coefficient of 0.41 indicated a moderate correlationbetween the numbr of votes and the rating of the restaurants.on average, restaurants with more votes to have slightly higher ratings and vice versa,

Task 3: Price vs Online Delivery and Table booking

1. Relationship betwen the price range and the availability of online delivery and table booking

In [41]:
```
df.head(3)
```

Out[41]:

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitud
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...	121.02753
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...	121.01410
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...	121.05683

3 rows × 21 columns

In [43]:

```
result = df.groupby('Price range').agg(percentage_online_delivery = ('Has Table boo
```

In [44]:

```
result
```

Out[44]:

	Price range	percentage_online_delivery
0	1	0.02
1	2	7.68
2	3	45.74
3	4	46.76

In [45]:

```
melted_result = pd.melt(result,id_vars = 'Price range',value_name = 'percentage')
```

In [46]:

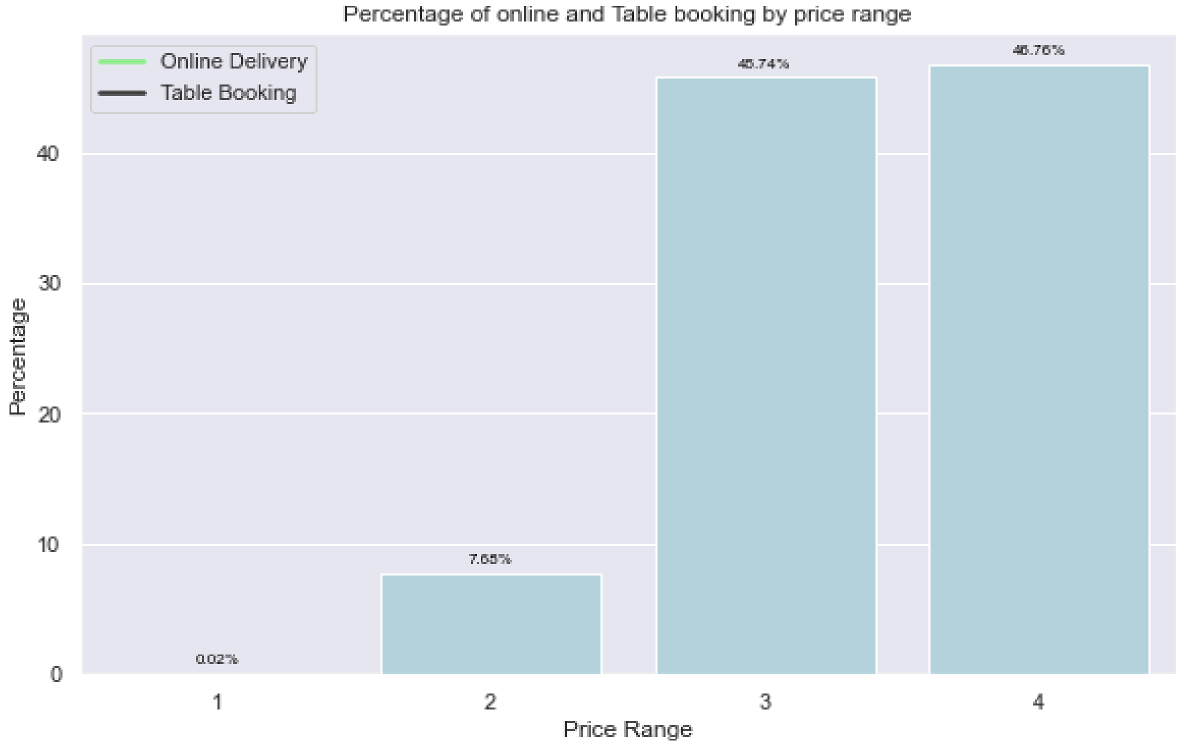
```
custom_palette = {'percentage_online_delivery':'lightblue','percentage_table_bookin
```

In [47]:

```
plt.figure(figsize=(10,6))
ax = sns.barplot(x = 'Price range',y = 'percentage',hue = 'variable',data = melted_
plt.title('Percentage of online and Table booking by price range')
plt.xlabel('Price Range')
plt.ylabel('Percentage')
handles,labels = ax.get_legend_handles_labels()
custom_legend = plt.legend(handles,labels,title = None, labels = ['Online Delivery'
custom_legend.legendHandles[0].set_color('lightblue')
custom_legend.legendHandles[0].set_color('lightgreen')


for p in ax.patches:
```

```
    height = p.get_height()
    ax.annotate(f'{height:.2f}%',(p.get_x()+p.get_width()/2,height),ha = 'center',v
```

```
C:\Users\Admin\AppData\Local\Temp\ipykernel_5452\4088803761.py:7: UserWarning: You
have mixed positional and keyword arguments, some input may be discarded.
  custom_legend = plt.legend(handles,labels,title = None, labels = ['Online Delive
ry','Table Booking'])
```



Online Delivery:

The percentage of restaurants offering online offering online delivery decreases as the price range increases. The highest percentage of online delivery is observed in Price Range 2(41.31%), which is relatively lower-prices, while the lowest percentage is in Price Range 4(9.04%), which is the highest - periods.

Table Booking

The percentage of restaurants ordering table booking generally increases as the price range increases. The highest percentage of table booking is obsrved in Price Range(46.76%), Which is the highest-price, while the lowest percentage is in Price Range 1(0.02%), which is the lowest-priced.

Conclusion:

There is an inverse relationship between the price and the availibility of online delivery indicating that lower-priced restaurants are more likely to offer this service.Convesely, there is a positive relationship between the price range and the availibility of table booking, suggesting that highest-priced restaurant are more likely to provide reservation options

In [ ]: