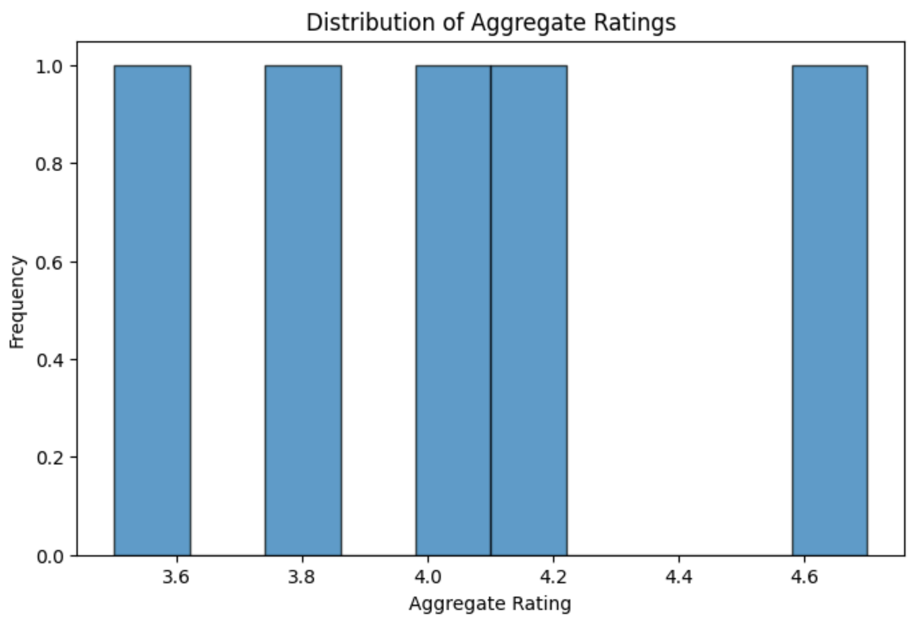
Level 2 (Task1) -: Analyze the distribution of aggregate rating and determine the most common rating range

Calculate the average number of votes received by restaurants

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
data = {
    'restaurant_name': ['Restaurant A', 'Restaurant B', 'Restaurant C', 'Restaurant D', 'Restaurant E'],
    'aggregate_rating': [4.2, 3.8, 4.7, 4.0, 3.5],
    'num_votes': [250, 150, 300, 200, 180]
}
df = pd.DataFrame(data)
rating_range = pd.cut(df['aggregate_rating'], bins=np.arange(0, 5.5, 0.5), right=False)
most_common_rating_range = rating_range.mode()[0]
plt.figure(figsize=(8, 5))
df['aggregate_rating'].plot(kind='hist', bins=10, edgecolor='black', alpha=0.7)
plt.title('Distribution of Aggregate Ratings')
plt.xlabel('Aggregate Rating')
plt.ylabel('Frequency')
plt.show()
average_votes = df['num_votes'].mean()
print(f"Most common rating range: {most_common_rating_range}")
print(f"Average number of votes received by restaurants: {average_votes:.2f}")
```



```
Most common rating range: [3.5, 4.0)
Average number of votes received by restaurants: 216.00
```

Level 2 (Task2) -:Identify the most common combinations of cuisines in the dataset

Determine if certain cuisine combinations tend to have higher ratings

```
import pandas as pd
import itertools
import matplotlib.pyplot as plt
data = {
    'restaurant_name': ['Restaurant A', 'Restaurant B', 'Restaurant C', 'Restaurant D', 'Restaurant E'],
    'cuisines': ['Indian, Chinese', 'Italian, Mexican', 'Indian, Chinese', 'Italian', 'Chinese'],
    'aggregate_rating': [4.2, 3.8, 4.7, 4.0, 3.5]
}
df = pd.DataFrame(data)
df['cuisines_split'] = df['cuisines'].apply(lambda x: x.split(', '))
cuisine_combinations = df['cuisines_split'].apply(lambda x: list(itertools.combinations(sorted(x), 2)))
flat_combinations = [item for sublist in cuisine_combinations for item in sublist]
combinations_df = pd.DataFrame(flat_combinations, columns=['cuisine_1', 'cuisine_2'])
common_combinations = combinations_df.groupby(['cuisine_1', 'cuisine_2']).size().reset_index(name='count')
common_combinations = common_combinations.sort_values(by='count', ascending=False)
print("Most common combinations of cuisines:")
print(common_combinations)
df['cuisines_combination'] = df['cuisines_split'].apply(lambda x: list(itertools.combinations(sorted(x), 2)))
exploded_df = df.explode('cuisines_combination')
```

```
Most common combinations of cuisines:
  cuisine_1 cuisine_2  count
0   Chinese    Indian      2
1   Italian   Mexican      1
```

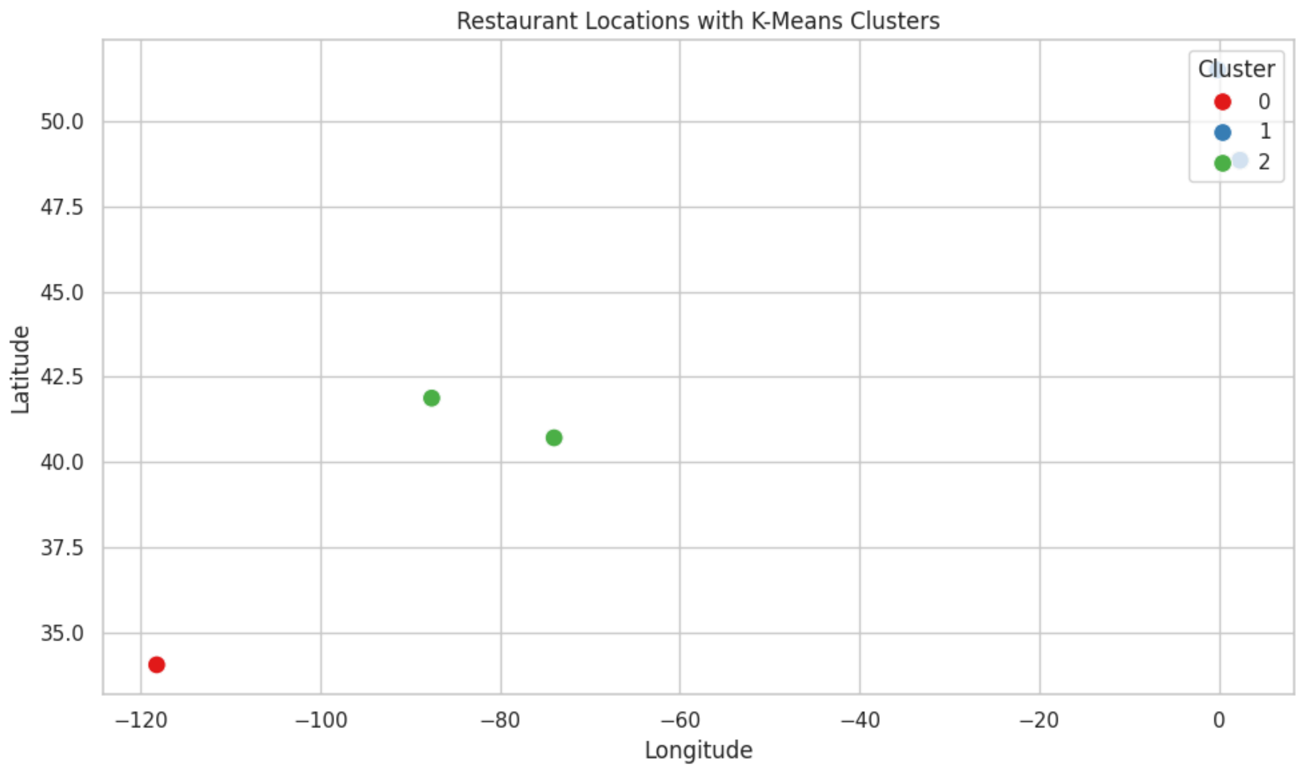
Level 2 (Task3) -:Plot the locations of restaurants on a map using longitude and latitude coordinates

Identify any patterns or clusters of restaurants in specific areas

```
import pandas as pd
import folium
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import seaborn as sns
data = {
    'restaurant_name': ['Restaurant A', 'Restaurant B', 'Restaurant C', 'Restaurant D', 'Restaurant E'],
    'latitude': [40.7128, 34.0522, 51.5074, 48.8566, 41.8781],
    'longitude': [-74.0060, -118.2437, -0.1278, 2.3522, -87.6298]
}
df = pd.DataFrame(data)
map_center = [df['latitude'].mean(), df['longitude'].mean()]
restaurant_map = folium.Map(location=map_center, zoom_start=2)
for _, row in df.iterrows():
    folium.Marker([row['latitude'], row['longitude']], popup=row['restaurant_name']).add_to(restaurant_map)
restaurant_map.save("restaurants_map.html")
print("Interactive map saved as 'restaurants_map.html'.")
coords = df[['latitude', 'longitude']].values
kmeans = KMeans(n_clusters=3, random_state=42)
df['cluster'] = kmeans.fit_predict(coords)
sns.set(style="whitegrid")
plt.figure(figsize=(10, 6))
sns.scatterplot(x='longitude', y='latitude', hue='cluster', data=df, palette='Set1', s=100, marker='o')

plt.title('Restaurant Locations with K-Means Clusters')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.legend(title='Cluster', loc='upper right')
plt.tight_layout()
plt.show()
print("Cluster centers (latitude, longitude):")
print(kmeans.cluster_centers_)
```

```
Interactive map saved as 'restaurants_map.html'.
```



```
Cluster centers (latitude, longitude):
[[  34.0522  -118.2437 ]
 [  50.182      1.1122 ]
 [  41.29545  -80.8179 ]]
```

Level 2 (Task 4) -:Identify if there are any restaurant chains present in a dataset

Analyze the ratings and ppopularity of different restaurant chains

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
data = {
    'restaurant_name': ['McDonald\'s A', 'McDonald\'s B', 'Starbucks A', 'Subway C', 'Subway A', 'Starbucks B', 'Local Diner'],
    'chain_name': ['McDonald\'s', 'McDonald\'s', 'Starbucks', 'Subway', 'Subway', 'Starbucks', None],  # Chain names
    'aggregate_rating': [4.2, 3.9, 4.5, 4.0, 4.2, 4.4, 3.5],
    'num_votes': [1000, 800, 1200, 600, 500, 900, 150]
}
df = pd.DataFrame(data)
chains_df = df[df['chain_name'].notna()]
average_ratings = chains_df.groupby('chain_name')['aggregate_rating'].mean().reset_index()
popularity = chains_df.groupby('chain_name')['num_votes'].sum().reset_index()
chain_analysis = pd.merge(average_ratings, popularity, on='chain_name')
chain_analysis = chain_analysis.sort_values(by='num_votes', ascending=False)
print("Restaurant Chain Analysis (Average Rating and Popularity):")
print(chain_analysis)
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
sns.barplot(x='aggregate_rating', y='chain_name', data=chain_analysis, palette='Blues_d')
plt.title('Average Ratings by Restaurant Chain')
plt.subplot(1, 2, 2)
sns.barplot(x='num_votes', y='chain_name', data=chain_analysis, palette='Blues_d')
plt.title('Popularity by Restaurant Chain (Total Votes)')

plt.tight_layout()
plt.show()
```

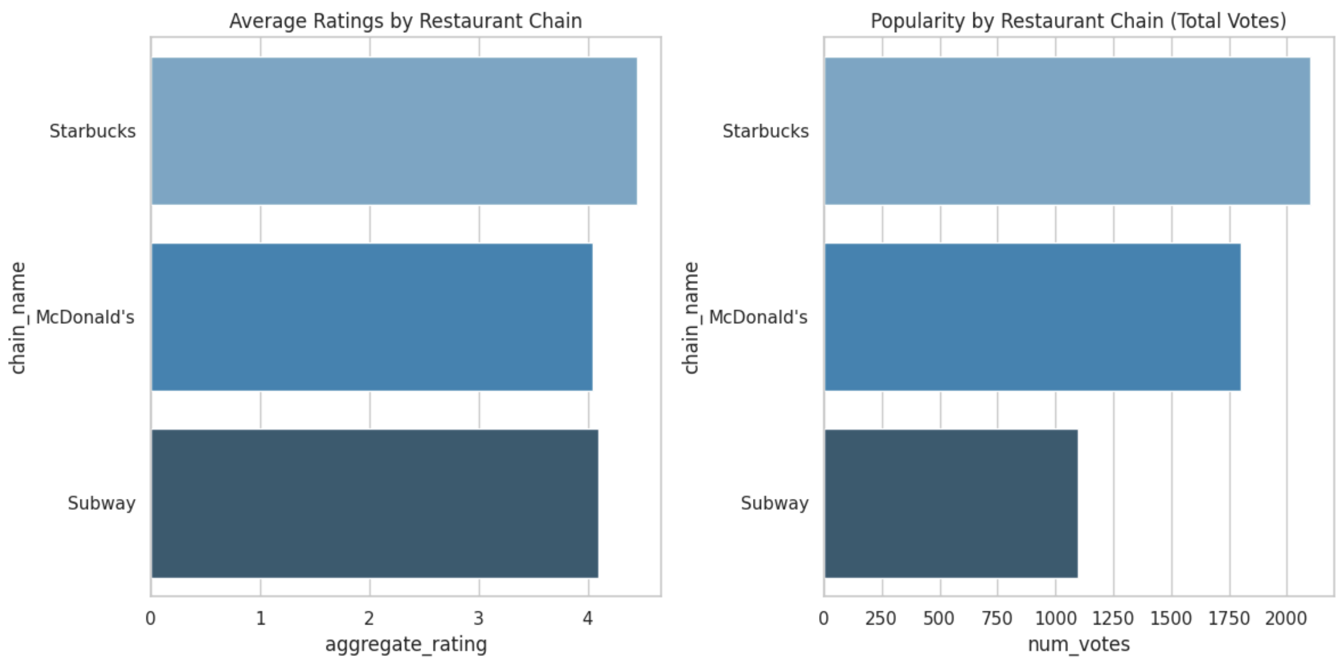
```
Restaurant Chain Analysis (Average Rating and Popularity):
   chain_name  aggregate_rating  num_votes
1   Starbucks              4.45       2100
0  McDonald's              4.05       1800
2      Subway              4.10       1100
<ipython-input-8-f372083bbc7d>:20: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `le

  sns.barplot(x='aggregate_rating', y='chain_name', data=chain_analysis, palette='Blues_d')
<ipython-input-8-f372083bbc7d>:23: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `le

  sns.barplot(x='num_votes', y='chain_name', data=chain_analysis, palette='Blues_d')
```


Average Ratings by Restaurant Chain
Popularity by Restaurant Chain (Total Votes)
Starbucks
McDonald's
Subway
chain_name
aggregate_rating
num_votes