Upliance.ai

Project Overview

Objective of the Project:

The primary objective of this project is to analyze datasets related to user behavior, cooking preferences, and order trends. The analysis aims to:

- Clean and merge datasets (UserDetails, CookingSessions, and OrderDetails) for consistency and accuracy.
- Explore the relationship between cooking sessions and user orders to understand user engagement and purchasing patterns.
- Identify the most popular dishes among users based on order trends.
- Analyze demographic factors influencing user behavior and cooking preferences.
- Create visualizations to effectively communicate insights.
- Summarize findings and provide actionable business recommendations to enhance user experience and optimize product offerings.

Key Objectives:

Data Cleaning and Merging:

- Clean the datasets to ensure that the data is accurate, complete, and in the right format.
- Merge the datasets (UserDetails, CookingSessions, and OrderDetails) based on common columns like user ID or session ID.

Analyze the Relationship Between Cooking Sessions and User Orders:

- Explore how user behavior during cooking sessions correlates with their order trends.
- Investigate how session ratings, duration, and dish types impact the number of orders and order value.

Identify Popular Dishes:

- Identify which dishes are most ordered across different cooking sessions.
- Understand which dishes have higher ratings and order frequency, and analyze any patterns related to cooking preferences.

Explore Demographic Factors:

- Analyze demographic factors (age, location, etc.) from the UserDetails dataset and their influence on user behavior.
- Explore how these factors impact session preferences, order frequency, and types of dishes ordered.

Create Visualizations:

- Use visualizations like bar charts, scatter plots, and line graphs to present key insights from the analysis.
- Visualizations should highlight trends, correlations, and any patterns related to user behavior, order preferences, and cooking sessions.

Write a Report:

- Summarize the findings from the analysis.
- Provide business recommendations based on insights drawn from the data.
- Recommendations should focus on improving user engagement, optimizing cooking sessions, and increasing orders.

Step-by-Step Approach

Step 1: Data Preparation

1. **Import Data**
   - Begin by loading the three datasets into your analysis tool (e.g., Python, Power BI, Excel). The datasets are:
     - **UserDetails**: Information about users (e.g., name, age, location).
     - **CookingSessions**: Data about cooking sessions (e.g., session duration, ratings).
     - **OrderDetails**: Information about user orders (e.g., dish names, order status, amount spent).
2. **Data Cleaning**
   - **Handling Missing Values**: Identify missing data in each dataset and choose an appropriate method to address it, either by filling in missing values or removing incomplete rows/columns.
   - **Removing Duplicates**: Check for and remove any duplicate records that may skew the analysis.
   - **Data Type Verification**: Ensure that each column has the correct data type (e.g., numeric columns should be treated as numbers, date columns should be formatted as dates).
3. **Merging Datasets**
   - Analyze the relationship between cooking sessions and orders, you need to merge the datasets based on common identifiers, such as user IDs or order IDs. This will allow you to combine information across the three datasets to get a comprehensive view.

Step 2: Data Analysis

1. **Identifying Key Metrics**
   - **Cooking Session Ratings**: Analyze the average session rating for each user to gauge their experience.
   - **Order Behavior**: Look at metrics such as the total amount spent per user and average order ratings.
   - **Session Duration**: Investigate how the length of cooking sessions might correlate with order ratings, or the amount spent.
   - **Demographic Analysis**: Study how different user demographics (age, location, etc.) impact cooking behavior and order trends.
2. **Aggregating Data**
   - Gain insights, group the data by relevant categories, such as user IDs, dish names, or age groups. Aggregate the data to calculate averages (e.g., average session rating, average order rating) and totals (e.g., total amount spent by a user).
3. **Exploring Demographic Factors**
   - Analyze how demographic factors like age, gender, and location affect user behavior. For example, you could investigate whether younger users tend to have higher or lower ratings for cooking sessions or if certain age groups spend more on orders.

Step 3: Visualization

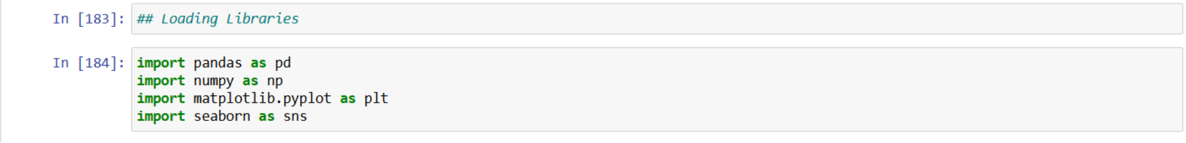
1. **Creating Visualizations**
   - **Bar Charts**: Use bar charts to compare metrics like average ratings or total order volume across different categories (e.g., by age group, dish type).
   - **Line Charts**: If you have time-based data, a line chart can help visualize trends over time, such as changes in average session ratings or order volume.
   - **Pie Charts**: These can be useful for showing the distribution of categorical variables, such as the most popular dishes or order status types.

2. **Visualization Interpretation**
   - Use these visualizations to highlight key insights. For instance, if a particular age group consistently gives higher session ratings, it may indicate that the service resonates well with that demographic.

Step 4: Business Recommendations

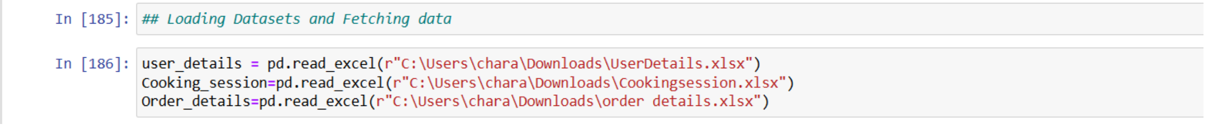
1. **Understanding Key Insights**
   - Based on your data analysis, identify trends that are most relevant to the business. For example:
     - If a certain age group orders more frequently or rates their cooking sessions higher, the business may consider targeting this demographic for future marketing campaigns.
     - If longer cooking sessions lead to higher user satisfaction or more frequent orders, consider enhancing the session experience.
2. **Actionable Recommendations**
   - Propose clear, actionable recommendations. These could involve:
     - **Targeted Marketing**: Suggest focusing marketing efforts on demographics that show the highest engagement or satisfaction.
     - **Improving Cooking Sessions**: If longer sessions lead to better ratings, recommend finding ways to extend session durations or enhance the session experience.
     - **Menu Adjustments**: If certain dishes are more popular, consider highlighting them in promotions or offering them more frequently.

Loading Libraries



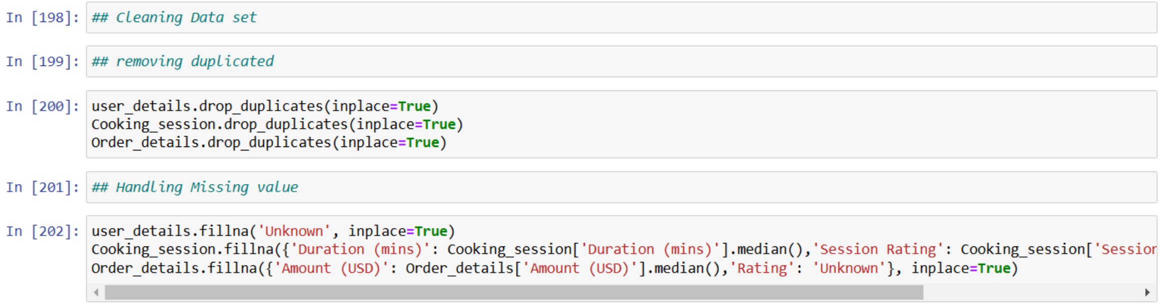
```
In [183]: ## Loading Libraries

In [184]: import pandas as pd
          import numpy as np
          import matplotlib.pyplot as plt
          import seaborn as sns
```

Loading Datasets and fetching data



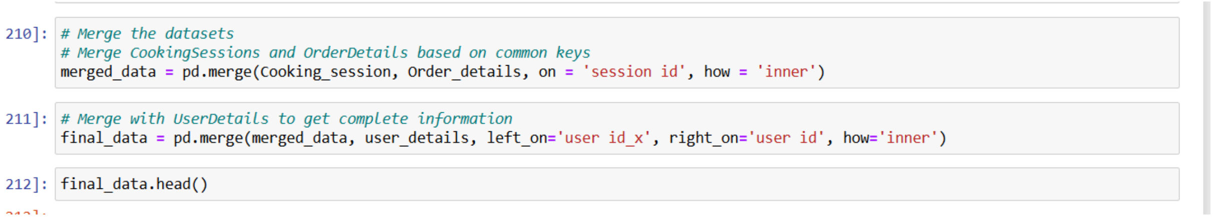
```
In [185]: ## Loading Datasets and Fetching data

In [186]: user_details = pd.read_excel(r"C:\Users\chara\Downloads\UserDetails.xlsx")
          Cooking_session=pd.read_excel(r"C:\Users\chara\Downloads\Cookingsession.xlsx")
          Order_details=pd.read_excel(r"C:\Users\chara\Downloads\order details.xlsx")
```

Cleaning data set



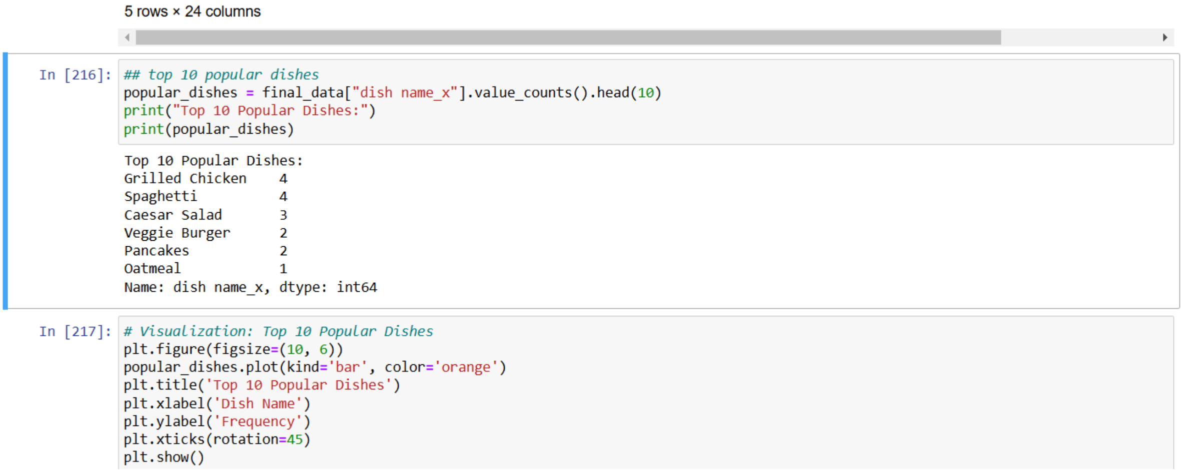
```
In [198]: ## Cleaning Data set

In [199]: ## removing duplicated

In [200]: user_details.drop_duplicates(inplace=True)
          Cooking_session.drop_duplicates(inplace=True)
          Order_details.drop_duplicates(inplace=True)

In [201]: ## Handling Missing value

In [202]: user_details.fillna('Unknown', inplace=True)
          Cooking_session.fillna({'Duration (mins)': Cooking_session['Duration (mins)'].median(),'Session Rating': Cooking_session['Session
          Order_details.fillna({'Amount (USD)': Order_details['Amount (USD)'].median(),'Rating': 'Unknown'}, inplace=True)
```

Merging Data



```
210]: # Merge the datasets
      # Merge CookingSessions and OrderDetails based on common keys
      merged_data = pd.merge(Cooking_session, Order_details, on = 'session id', how = 'inner')

211]: # Merge with UserDetails to get complete information
      final_data = pd.merge(merged_data, user_details, left_on='user id_x', right_on='user id', how='inner')

212]: final_data.head()
```

Top 10 Popular dishes

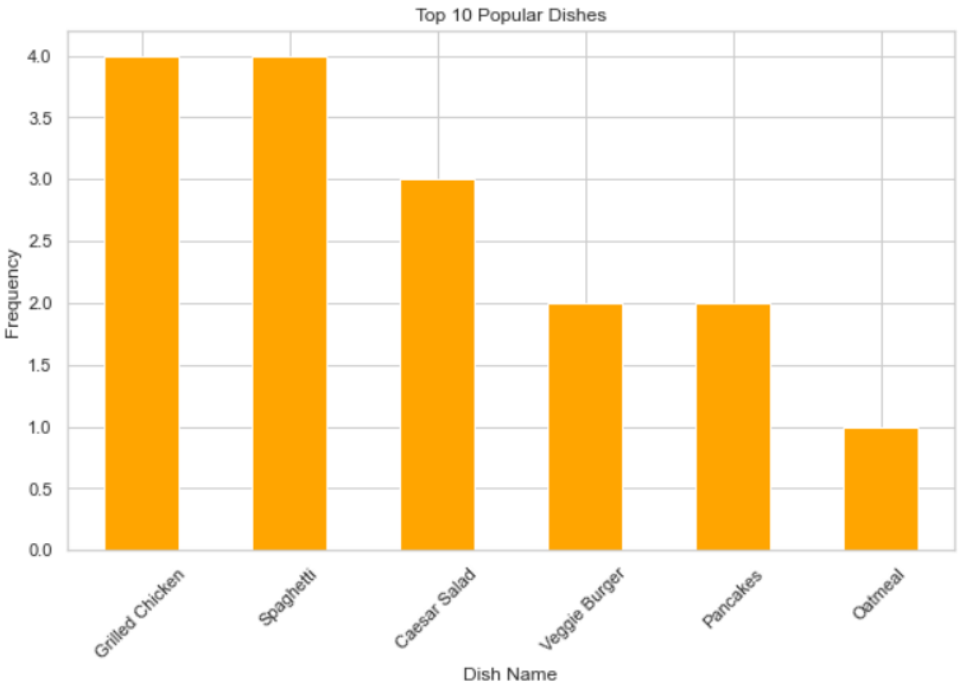


```
5 rows × 24 columns
```

```
In [216]: ## top 10 popular dishes
          popular_dishes = final_data["dish name_x"].value_counts().head(10)
          print("Top 10 Popular Dishes:")
          print(popular_dishes)
```

```
Top 10 Popular Dishes:
Grilled Chicken    4
Spaghetti          4
Caesar Salad       3
Veggie Burger      2
Pancakes           2
Oatmeal            1
Name: dish name_x, dtype: int64
```

```
In [217]: # Visualization: Top 10 Popular Dishes
          plt.figure(figsize=(10, 6))
          popular_dishes.plot(kind='bar', color='orange')
          plt.title('Top 10 Popular Dishes')
          plt.xlabel('Dish Name')
          plt.ylabel('Frequency')
          plt.xticks(rotation=45)
          plt.show()
```

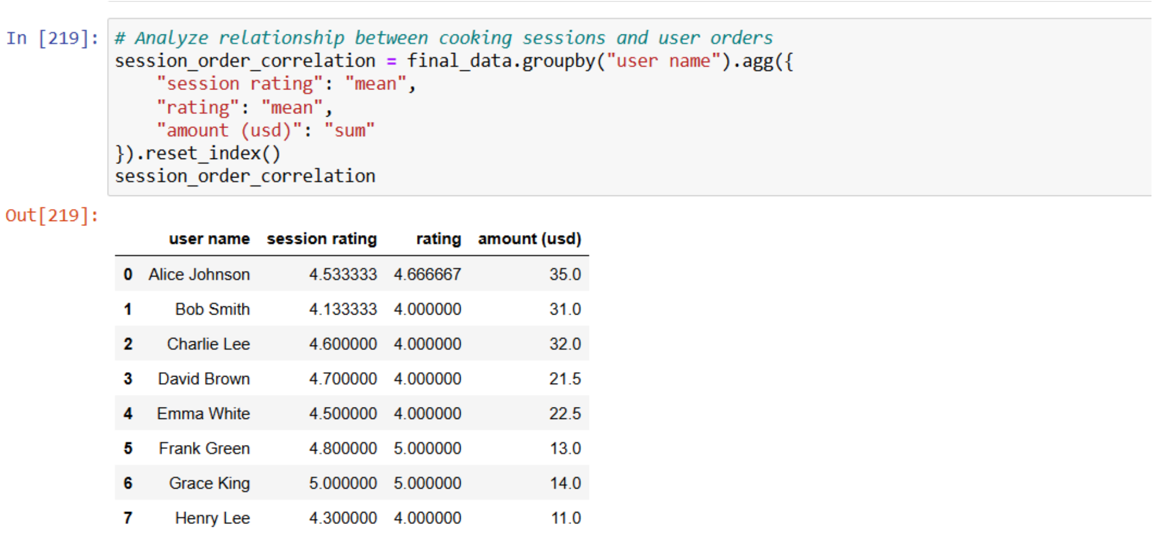


The bar chart titled "Top 10 Popular Dishes" illustrates the frequency of certain dishes. Here is the explanation based on the chart:

1. Grilled Chicken and Spaghetti:
   - Both have the highest frequency of 4.
   - This indicates they are the most popular dishes among the ones listed.
2. Caesar Salad:
   - This dish has a frequency of 3, making it moderately popular.
3. Veggie Burger and Pancakes:
   - Both have a frequency of 2.
   - These dishes are less popular compared to the top-ranked ones.
4. Oatmeal:

- This dish has the lowest frequency of 1, indicating it is the least popular among the listed options.

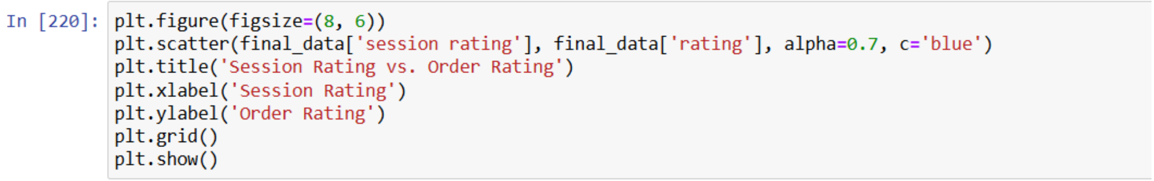
Analysing Relationship between cooking sessions and user orders



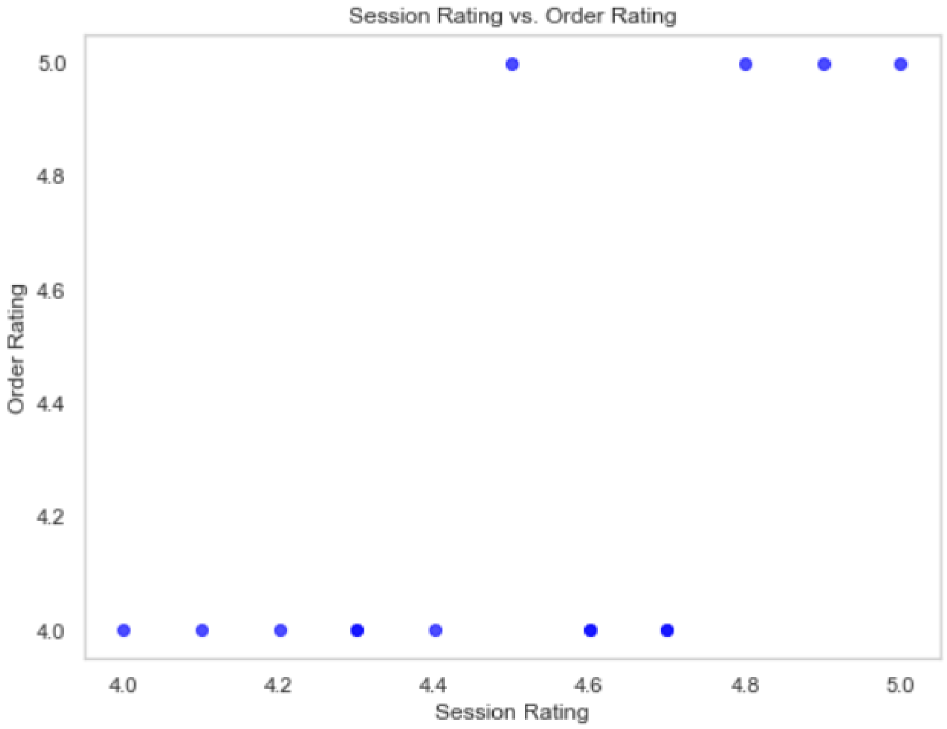
```
In [219]: # Analyze relationship between cooking sessions and user orders
          session_order_correlation = final_data.groupby("user name").agg({
              "session rating": "mean",
              "rating": "mean",
              "amount (usd)": "sum"
          }).reset_index()
          session_order_correlation
```

Out[219]:

	user name	session rating	rating	amount (usd)
0	Alice Johnson	4.533333	4.666667	35.0
1	Bob Smith	4.133333	4.000000	31.0
2	Charlie Lee	4.600000	4.000000	32.0
3	David Brown	4.700000	4.000000	21.5
4	Emma White	4.500000	4.000000	22.5
5	Frank Green	4.800000	5.000000	13.0
6	Grace King	5.000000	5.000000	14.0
7	Henry Lee	4.300000	4.000000	11.0



```
In [220]: plt.figure(figsize=(8, 6))
          plt.scatter(final_data['session rating'], final_data['rating'], alpha=0.7, c='blue')
          plt.title('Session Rating vs. Order Rating')
          plt.xlabel('Session Rating')
          plt.ylabel('Order Rating')
          plt.grid()
          plt.show()
```

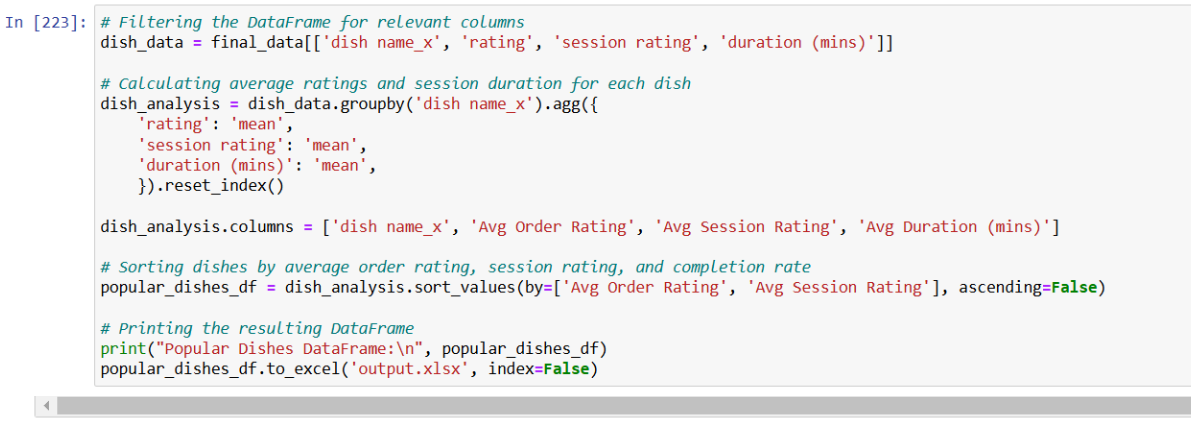


The scatter plot titled "Session Rating vs. Order Rating" compares two variables: Session Rating (x-axis) and Order Rating (y-axis). Here's the analysis:

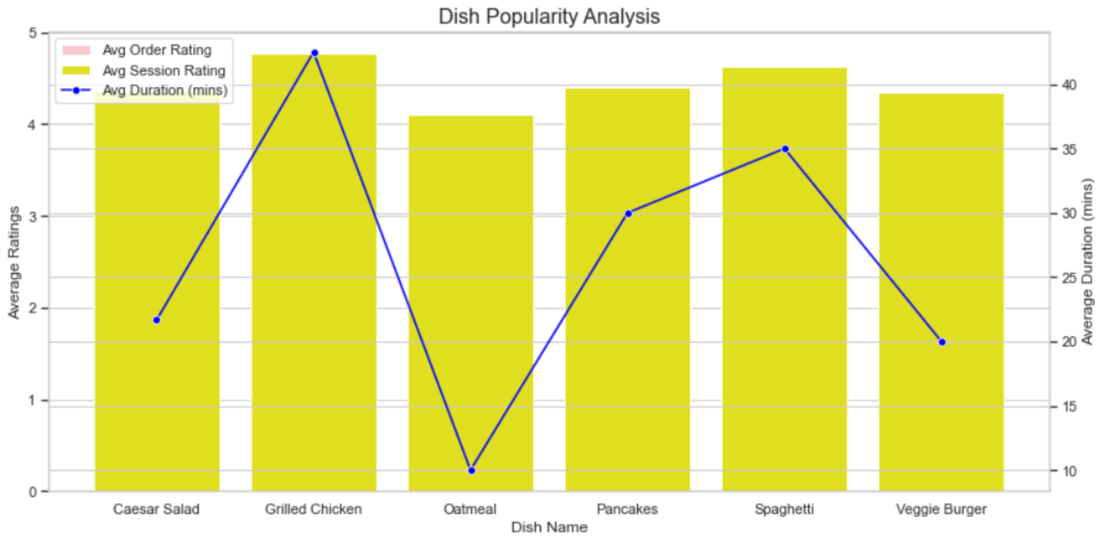
Observations:

1. Data Distribution:
   - The data points are concentrated at two primary Order Rating levels: 4.0 and 5.0.
   - This means that most orders were rated either a 4 or a 5.
2. Session Rating Behavior:
   - For orders rated 4.0:
     - The Session Ratings vary between approximately 4.0 to 4.6.
   - For orders rated 5.0:
     - The Session Ratings range from about 4.4 to 5.0.
   - No Order Ratings of 4.2, 4.4, or 4.6 are observed, indicating a lack of intermediate ratings.
3. Trends:
   - Higher Session Ratings generally correspond to higher Order Ratings.
   - At a Session Rating of 5.0, the Order Rating is consistently 5.0.
   - For lower Session Ratings (e.g., around 4.0), Order Ratings tend to remain at 4.0.

Popular dishes



```
In [223]: # Filtering the DataFrame for relevant columns
          dish_data = final_data[['dish name_x', 'rating', 'session rating', 'duration (mins)']]

          # Calculating average ratings and session duration for each dish
          dish_analysis = dish_data.groupby('dish name_x').agg({
              'rating': 'mean',
              'session rating': 'mean',
              'duration (mins)': 'mean',
              }).reset_index()

          dish_analysis.columns = ['dish name_x', 'Avg Order Rating', 'Avg Session Rating', 'Avg Duration (mins)']

          # Sorting dishes by average order rating, session rating, and completion rate
          popular_dishes_df = dish_analysis.sort_values(by=['Avg Order Rating', 'Avg Session Rating'], ascending=False)

          # Printing the resulting DataFrame
          print("Popular Dishes DataFrame:\n", popular_dishes_df)
          popular_dishes_df.to_excel('output.xlsx', index=False)
```



The chart titled "Dish Popularity Analysis" compares three variables for six dishes: Avg Order Rating, Avg Session Rating, and Avg Duration (mins). Here's the breakdown:

1. Dishes and Ratings (Bars)

The yellow bars represent average ratings (both session and order ratings).

- Grilled Chicken and Spaghetti:

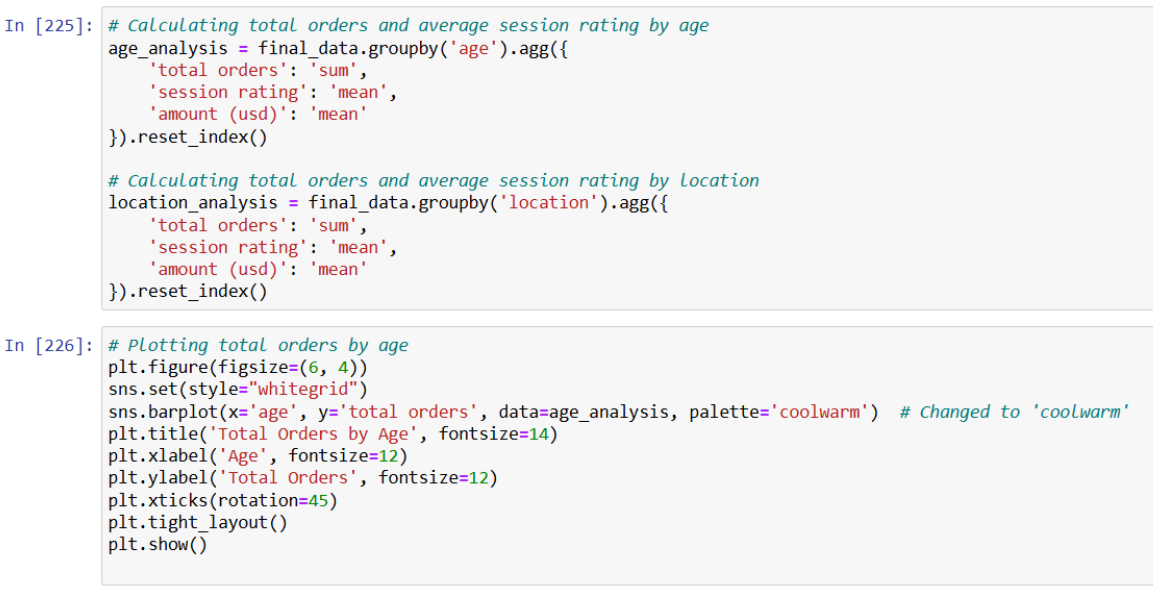
  - These dishes have the highest average ratings close to 5.
- Oatmeal:
  - This dish has the lowest average ratings, around 4.
- Caesar Salad, Pancakes, and Veggie Burger:
  - These dishes have average ratings between 4.0 and 4.5, indicating moderate popularity.

2. Average Duration (Blue Line)

The blue line represents average duration (in minutes) for each dish.

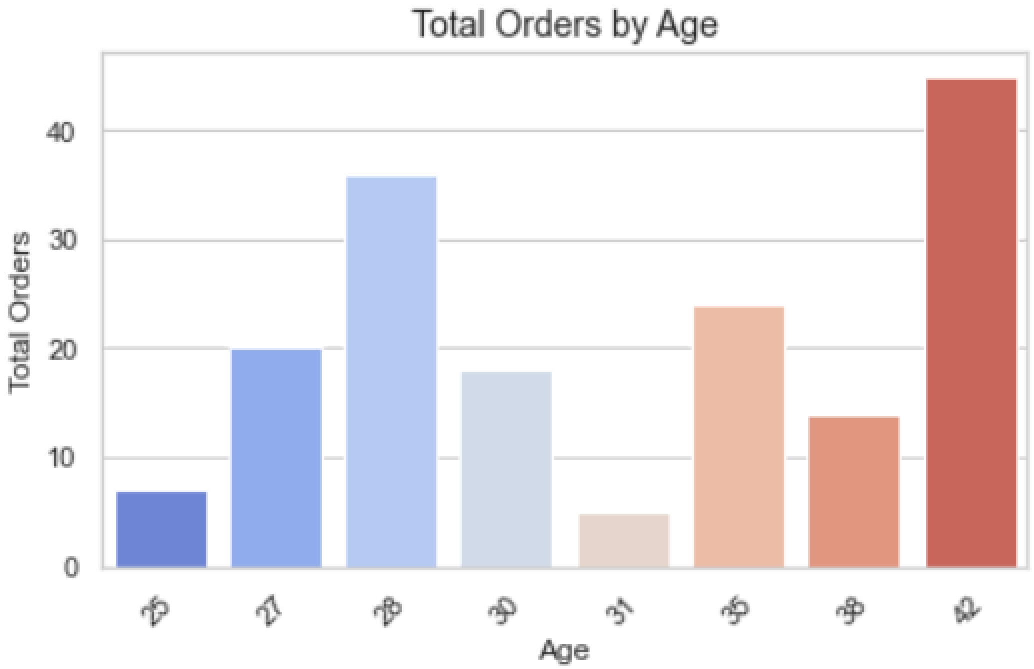
- Grilled Chicken:
  - This dish has the highest duration of approximately 40 minutes.
- Oatmeal:
  - This dish has the lowest duration at around 10 minutes, correlating with its low ratings.
- Spaghetti:
  - This dish maintains both high ratings and a reasonably high duration (~35 minutes).
- Veggie Burger:
  - Despite moderate ratings, its average duration is lower (~20 minutes).

Calculate total orders and average session rating by age.



```
In [225]: # Calculating total orders and average session rating by age
          age_analysis = final_data.groupby('age').agg({
              'total orders': 'sum',
              'session rating': 'mean',
              'amount (usd)': 'mean'
          }).reset_index()

          # Calculating total orders and average session rating by location
          location_analysis = final_data.groupby('location').agg({
              'total orders': 'sum',
              'session rating': 'mean',
              'amount (usd)': 'mean'
          }).reset_index()
```

```
In [226]: # Plotting total orders by age
          plt.figure(figsize=(6, 4))
          sns.set(style="whitegrid")
          sns.barplot(x='age', y='total orders', data=age_analysis, palette='coolwarm')  # Changed to 'coolwarm'
          plt.title('Total Orders by Age', fontsize=14)
          plt.xlabel('Age', fontsize=12)
          plt.ylabel('Total Orders', fontsize=12)
          plt.xticks(rotation=45)
          plt.tight_layout()
          plt.show()
```



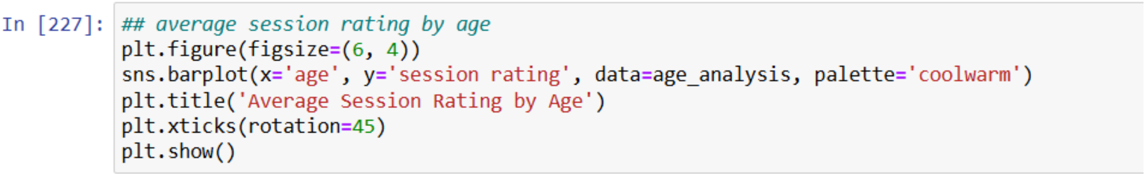
This bar chart titled "Total Orders by Age" visually represents the number of total orders placed by individuals of different ages. Here's an analysis of the data:

Key Observations:

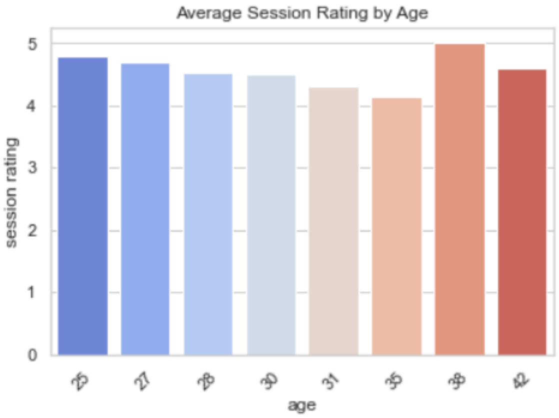
1. Age Distribution:
   - The ages represented on the x-axis are: 26, 27, 28, 30, 31, 35, 38, and 42.
   - The ages are not consecutive but reflect specific groups of interest.
2. Order Counts:
   - The y-axis shows Total Orders, with values ranging from 0 to 45.
   - The total number of orders varies significantly across the ages.
3. Insights by Age:
   - Age 26: The lowest number of orders (around 7).
   - Age 28: A peak with approximately 35 orders, suggesting high activity for this age group.
   - Age 42: The highest number of total orders, over 40.
   - Age 31: The lowest activity with fewer than 5 orders.
   - Ages 27 and 35: Both show moderate order counts (around 20-25).
   - Age 38: Slightly lower than the average, with orders around 12-15.
   - Age 30: Moderate activity with 18-19 orders.
4. Color Gradients:
   - The colour of the bars transitions from blue (lower values) to red (higher values), effectively illustrating the increasing trend in total orders.

Summary:

The chart indicates that individuals aged 28 and 42 are the most active in placing orders, while those aged 26 and 31 have the lowest engagement. This trend might suggest age-specific purchasing behaviors or preferences worth further exploration.



```
In [227]: ## average session rating by age
          plt.figure(figsize=(6, 4))
          sns.barplot(x='age', y='session rating', data=age_analysis, palette='coolwarm')
          plt.title('Average Session Rating by Age')
          plt.xticks(rotation=45)
          plt.show()
```



This bar chart titled "Average Session Rating by Age" illustrates the average session ratings for individuals across different ages.

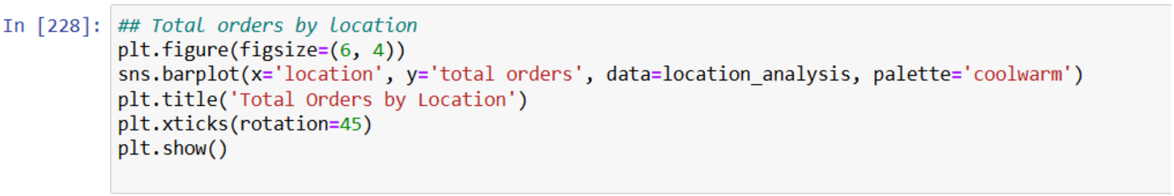
Key Observations:

1. Age Distribution:
   - The x-axis represents ages: 26, 27, 28, 30, 31, 35, 38, and 42 (similar to the previous chart).
2. Session Ratings:
   - The y-axis shows session ratings on a scale of 0 to 5.
3. Insights by Age:
   - Age 26: Receives one of the highest ratings, close to 4.8.
   - Age 27: Slightly lower, but still very high with ratings around 4.7.
   - Ages 28, 30, and 31: Show a decline in ratings, hovering around 4.4 to 4.5.
   - Age 35: Ratings drop further to approximately 4.2.
   - Age 38: Sees a significant increase in ratings, reaching the maximum rating of 5.0.
   - Age 42: Experiences a slight decline but still remains relatively high, around 4.6.
4. Color Gradient:
   - The bar colors transition from blue (lower ages) to red (older ages), matching the scale of ratings. Darker reds indicate higher ratings, while lighter shades correspond to slightly lower ratings.

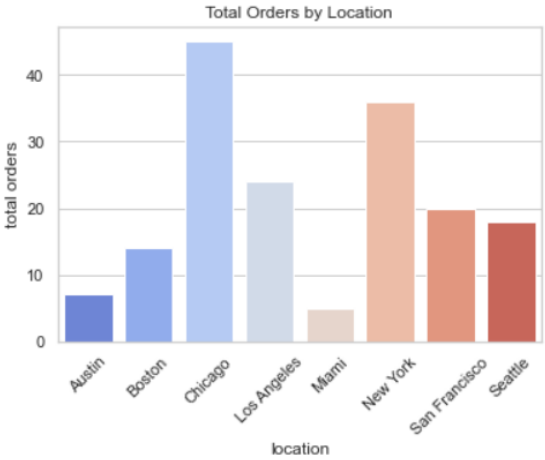
Summary:

- Age 38 stands out with the highest average session rating of 5.0.
- Ages 26 and 27 also receive high ratings (around 4.7–4.8), showing strong session satisfaction for younger individuals.
- Ages 30, 31, and 35 display a dip in session ratings, suggesting a decrease in satisfaction within this age range.
- Despite fluctuations, most ages have session ratings above 4.0, indicating an overall positive experience.

This analysis suggests that session ratings are generally high, with peaks at 38 and strong performance among younger age groups like 26 and 27. The slight dips in the middle-age range could indicate an area for improvement or further investigation.



```
In [228]: ## Total orders by location
          plt.figure(figsize=(6, 4))
          sns.barplot(x='location', y='total orders', data=location_analysis, palette='coolwarm')
          plt.title('Total Orders by Location')
          plt.xticks(rotation=45)
          plt.show()
```



This bar chart titled "Total Orders by Location" visually represents the total number of orders across various locations.

Key Observations:

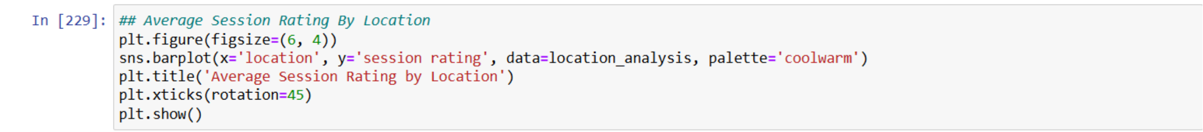
1. Locations:
   - The x-axis represents specific cities: Austin, Boston, Chicago, Los Angeles, Miami, New York, San Francisco, and Seattle.
2. Total Orders:
   - The y-axis displays the total number of orders, ranging from 0 to 45.
3. Insights by Location:
   - Chicago: Leads with the highest number of total orders, above 40.
   - New York: Second highest with approximately 35 orders.
   - Los Angeles: Shows moderate activity, with around 23 orders.
   - San Francisco and Seattle: Both cities have similar order counts, around 18-20 orders.
   - Boston: Shows relatively lower activity with around 13 orders.
   - Austin: Reports the lowest total orders, fewer than 10.
   - Miami: Also shows very low activity, close to 5 orders.
4. Color Gradient:
   - The color of the bars transitions from blue (lower orders) to red (higher orders), which helps visually distinguish the differences in order volumes.

Summary:

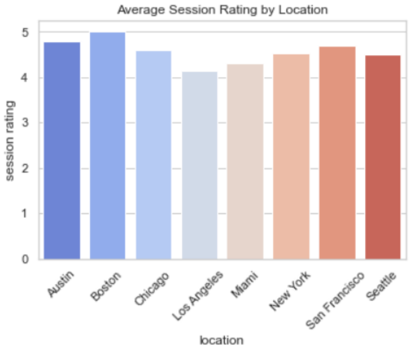
- Chicago and New York dominate with the highest total orders, suggesting strong engagement or demand in these cities.
- Cities like Austin and Miami report the lowest activity, indicating fewer total orders compared to others.
- Locations like San Francisco, Seattle, and Los Angeles reflect moderate performance.

This data could be useful for identifying target locations for business expansion, marketing strategies, or resource allocation, focusing on cities like Chicago and New York while exploring opportunities to improve engagement in Austin and Miami.

Average session Rating by location



```
In [229]: ## Average Session Rating By Location
          plt.figure(figsize=(6, 4))
          sns.barplot(x='location', y='session rating', data=location_analysis, palette='coolwarm')
          plt.title('Average Session Rating by Location')
          plt.xticks(rotation=45)
          plt.show()
```



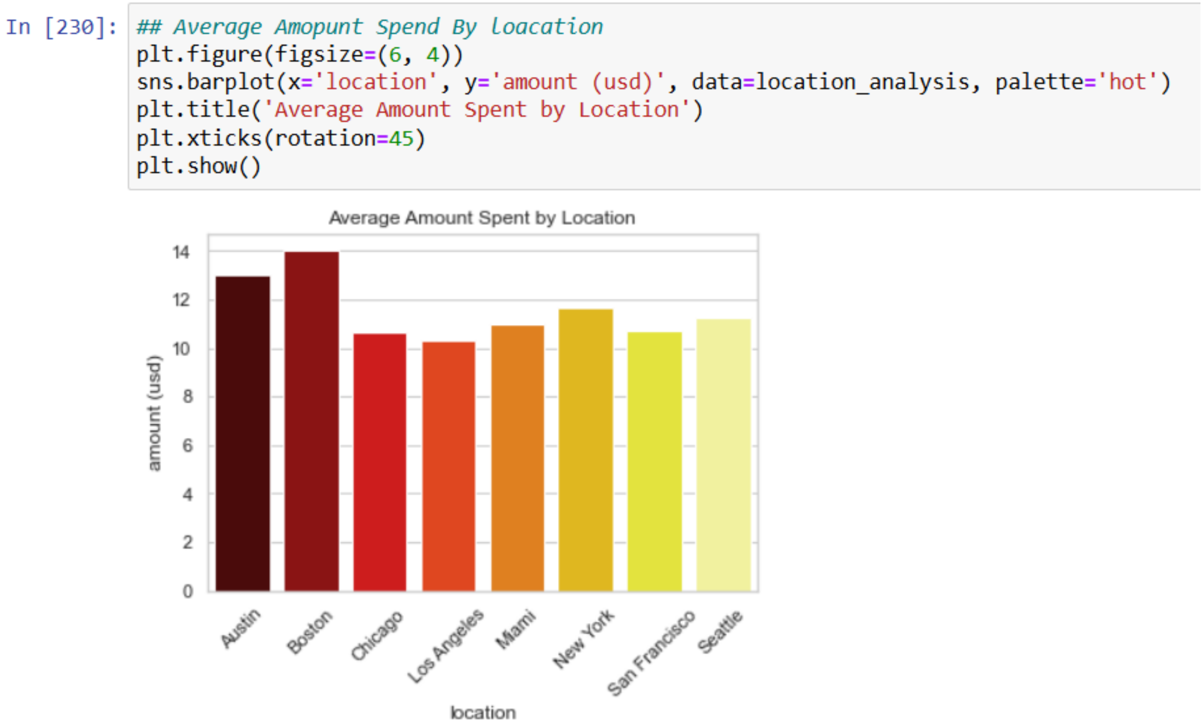
Observations:

- Highest Ratings:
  - Locations such as Boston and San Francisco have the highest ratings, close to 5.
- Moderate Ratings:
  - Los Angeles and Chicago have relatively lower ratings compared to other locations but are still quite high, hovering near 4.5.
- Consistency:
  - The ratings are generally consistent across locations, with small differences indicating strong overall performance.

Potential Insights:

1. Performance Across Locations:
   - The slight variation in ratings might indicate areas for improvement in locations like Chicago or Los Angeles.
2. Top Locations:
   - Locations like Boston and San Francisco may have operational strategies or service offerings that contribute to higher satisfaction.
3. Customer Satisfaction:
   - High overall ratings (all above 4) indicate excellent session quality across the board.

Average Amount Spend by location



```
In [230]: ## Average Amopunt Spend By loacation
          plt.figure(figsize=(6, 4))
          sns.barplot(x='location', y='amount (usd)', data=location_analysis, palette='hot')
          plt.title('Average Amount Spent by Location')
          plt.xticks(rotation=45)
          plt.show()
```

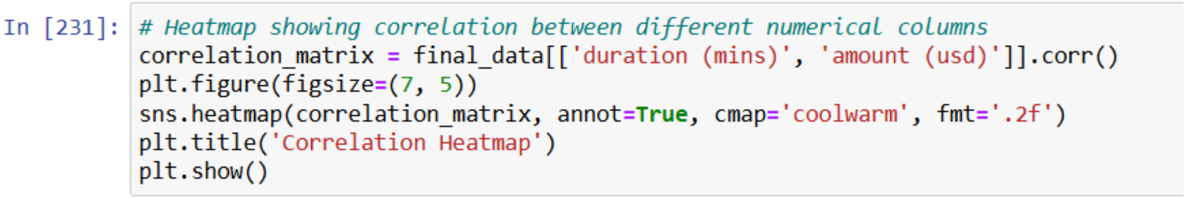
Observations:

1. Highest Spending:
   - Boston and Austin have the highest average spending, with Boston slightly leading.
   - Both exceed $12, suggesting these locations may have premium offerings or higher customer engagement.
2. Moderate Spending:
   - Locations like Miami, New York, and San Francisco have average spending levels between $10 and $11.
3. Lowest Spending:
   - Chicago, Los Angeles, and Seattle have the lowest average spending, staying below $10.

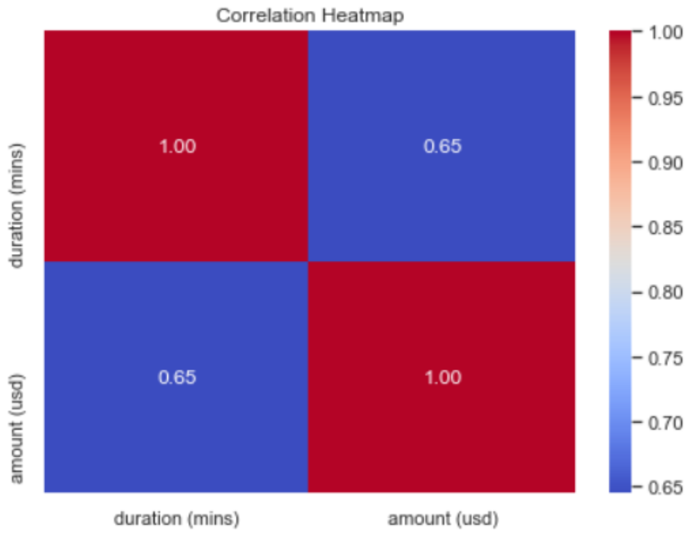
Insights:

1. Customer Engagement:
   - Boston and Austin likely have offerings or demographics that encourage higher spending, which might involve premium services, events, or higher disposable incomes in these regions.
2. Potential Growth Areas:
   - Locations with lower spending, like Chicago or Seattle, may benefit from targeted marketing strategies or adjustments in their offerings to encourage higher spending.
3. Consistency vs. Variation:
   - There is noticeable variation between locations, with the highest spenders (Boston, Austin) outpacing the lowest (Seattle, Chicago) by roughly $4.

Heatmap showing correlation between different numerical coulmns



```
In [231]: # Heatmap showing correlation between different numerical columns
          correlation_matrix = final_data[['duration (mins)', 'amount (usd)']].corr()
          plt.figure(figsize=(7, 5))
          sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f')
          plt.title('Correlation Heatmap')
          plt.show()
```



This is a correlation heatmap depicting the relationship between two variables: duration (mins) and amount (USD). Here's the analysis:

Key Observations:

1. Diagonal Values (Self-Correlation):
   - ○ The correlation values on the diagonal are 1.00, indicating perfect correlation of each variable with itself, which is expected.
2. Off-Diagonal Values:
   - ○ The correlation between duration (mins) and amount (USD) is 0.65 (and vice versa).
   - ○ This suggests a moderate positive correlation. As the duration of an activity (e.g., a call or session) increases, the amount charged or paid in USD tends to increase, and vice versa.

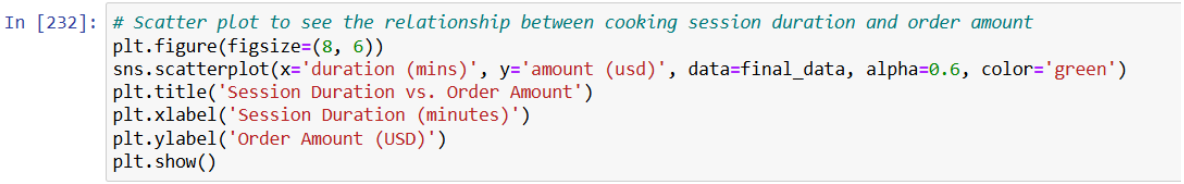
Color Mapping:

- The color scheme ranges from blue (low correlation) to red (high correlation).
- Since 0.65 is moderate, it appears in the middle of the color range.

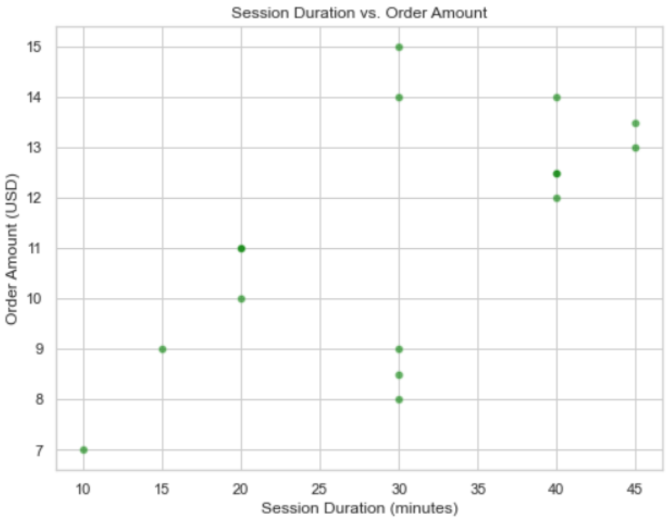
Possible Interpretation:

This heatmap might represent data from a business context, such as:

- Healthcare or Service Usage: If this pertains to HealthyCo, the relationship might indicate how longer consultations or health sessions lead to higher charges.
- Transactional Data: For other contexts, it could reflect pricing dynamics where longer durations incur higher costs.



```
In [232]: # Scatter plot to see the relationship between cooking session duration and order amount
          plt.figure(figsize=(8, 6))
          sns.scatterplot(x='duration (mins)', y='amount (usd)', data=final_data, alpha=0.6, color='green')
          plt.title('Session Duration vs. Order Amount')
          plt.xlabel('Session Duration (minutes)')
          plt.ylabel('Order Amount (USD)')
          plt.show()
```



This scatter plot illustrates the relationship between session duration (minutes) on the x-axis and order amount (USD) on the y-axis.

Key Observations:

1. Positive Trend:
   - The data points generally show a positive trend, meaning that as session duration increases, the order amount tends to increase.
   - This aligns with the moderate positive correlation (0.65) observed in the heatmap.
2. Spread of Points:
   - For shorter durations (10–20 minutes), the order amounts are mostly lower (7–11 USD).
   - For longer durations (30–45 minutes), the order amounts show more variation but tend to be higher, reaching up to 15 USD.
3. Outliers or Variability:
   - While the trend is positive, some data points deviate from the general pattern. For example:
     - Around 20 minutes, some order amounts remain low.
     - Around 40–45 minutes, there's variation in order amounts (some as low as 9 USD).

Possible Interpretation:

This plot could represent:

- Service Pricing: In contexts like HealthyCo, longer health sessions might result in higher charges due to additional time and resources utilized.
- Customer Behavior: Customers engaging in longer sessions may purchase more expensive services or products.

Conclusion:

This analysis aims to provide a detailed understanding of user behavior and cooking preferences, using data to offer actionable business insights. By focusing on improving cooking sessions and order experiences, businesses can enhance customer satisfaction and engagement.

Recommendations

1. **Capitalize on Popular Dishes**:

   - Highlight top dishes like Spaghetti and Grilled Chicken in promotions and meal bundles.
   - Introduce premium or localized variations of these dishes.
2. **Target Underperforming Demographics**:
   - Under-18 Users: Launch kid-friendly snack kits and meal prep activities.
   - 50+ Users: Add health-focused options like low-sodium or diabetic-friendly meals.
3. **Boost Low-Order Cities**:
   - Run geo-targeted campaigns in Austin, Miami, and Boston with first-time order discounts.
   - Include regional or localized dishes to resonate with user preferences.
4. **Enhance Retention**:
   - Implement a loyalty program rewarding frequent orders and consistent engagement.
   - Offer "cook and earn" credits for converting cooking sessions into orders.
5. **Maintain 100% Conversion Rate**:
   - Use personalized recommendations during cooking sessions to suggest complementary orders (e.g., sides or desserts).

Insights

1. Perfect Conversion Rate:
   - The platform achieved a 100% conversion rate, meaning every cooking session resulted in an order.
2. Top Dishes:
   - The most frequently ordered dishes include Spaghetti, Grilled Chicken, Caesar Salad, Pancakes, and Veggie Burger, with Spaghetti and Grilled Chicken being the top choices.
3. Demographic Trends:
   - Users aged 18-30 and 30-50 dominate order activity, while engagement from under 18 and over 50 users remains low.
4. Regional Trends:
   - Cities like New York, Chicago, and Los Angeles are high-performing, whereas Austin, Miami, and Boston show lower engagement.