CUSTOMER SATISFACTION – STATISTICAL ANALYSIS

Abhilash Antony

We have a Restaurant Customer Satisfaction Dataset. This dataset, rich with comprehensive information on customer visits to restaurants, including demographic details, visit-specific metrics, and customer satisfaction ratings, holds immense potential for predictive modelling and analytics in the hospitality industry.

Our objective is to perform the basic statistical analysis such as:

- Tests for the difference between the means of two independent groups.
- Tests for the difference between the means of three or more groups.
- Compares the variances of two groups.
- Tests for relationships between categorical variables.
- Measures the strength and direction of the linear relationship between two continuous variables.

The features in the dataset are:

Demographic Information

- CustomerID: Unique identifier for each customer.
- Age: Age of the customer.
- Gender: Gender of the customer (Male/Female).
- Income: The customer's annual income is in USD.

Visit-specific Variables

- VisitFrequency: How often the customer visits the restaurant
- AverageSpend: Average amount spent by the customer per visit in USD.
- PreferredCuisine: The type of cuisine preferred by the customer
- TimeOfVisit: The time of day the customer usually visits (Breakfast, Lunch, Dinner).
- GroupSize: Number of people in the customer's group during the visit.
- DiningOccasion: The occasion for dining (Casual, Business, Celebration).
- MealType: Type of meal (Dine-in, Takeaway).
- OnlineReservation: Whether the customer made an online reservation (0: No, 1: Yes).
- DeliveryOrder: Whether the customer ordered delivery (0: No, 1: Yes).
- LoyaltyProgramMember: Whether the customer is a member of the restaurant's loyalty program (0: No, 1: Yes).
- WaitTime: Average wait time for the customer in minutes.

Satisfaction Ratings

- ServiceRating: Customer's rating of the service (1 to 5).
- FoodRating: Customer's rating of the food (1 to 5).
- AmbianceRating: Customer's rating of the restaurant ambiance (1 to 5).
- HighSatisfaction: Binary variable indicating whether the customer is highly satisfied (1) or not (0).

Exploratory Data Analysis

```
# importing necessary libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
import numpy as np

# load the dataset
df = pd.read_csv('/content/restaurant_customer_satisfaction.csv')
df.head()
```

CustomerID	Age	Gender	Income	VisitFrequency	AverageSpend	PreferredCuisine	TimeOfVisit	GroupSize	DiningOccasion	MealType
654	35	Male	83380	Weekly	27.829142	Chinese	Breakfast	3	Business	Takeaway
655	19	Male	43623	Rarely	115.408622	American	Dinner	1	Casual	Dine-in
656	41	Female	83737	Weekly	106.693771	American	Dinner	6	Celebration	Dine-in
657	43	Male	96768	Rarely	43.508508	Indian	Lunch	1	Celebration	Dine-in
658	55	Female	67937	Monthly	148.084627	Chinese	Breakfast	1	Business	Takeaway

OnlineReservation	DeliveryOrder	LoyaltyProgramMember	WaitTime	ServiceRating	FoodRating	AmbianceRating	HighSatisfaction
0	1	1	43.523929	2	5	4	0
0	0	0	57.524294	5	5	3	0
0	1	0	48.682623	3	4	5	0
0	0	0	7.552993	4	5	1	0
0	0	1	37.789041	2	3	5	0

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1500 entries, 0 to 1499
Data columns (total 19 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   CustomerID            1500 non-null   int64  
 1   Age                   1500 non-null   int64  
 2   Gender                1500 non-null   object 
 3   Income                1500 non-null   int64  
 4   VisitFrequency        1500 non-null   object 
 5   AverageSpend          1500 non-null   float64
 6   PreferredCuisine      1500 non-null   object 
 7   TimeOfVisit           1500 non-null   object 
 8   GroupSize             1500 non-null   int64  
 9   DiningOccasion        1500 non-null   object 
 10  MealType              1500 non-null   object 
 11  OnlineReservation     1500 non-null   int64  
 12  DeliveryOrder         1500 non-null   int64  
 13  LoyaltyProgramMember  1500 non-null   int64  
 14  WaitTime              1500 non-null   float64
 15  ServiceRating         1500 non-null   int64  
 16  FoodRating            1500 non-null   int64  
 17  AmbianceRating        1500 non-null   int64  
 18  HighSatisfaction      1500 non-null   int64  
dtypes: float64(2), int64(11), object(6)
memory usage: 222.8+ KB
```

```
df.describe()
```

	CustomerID	Age	Income	AverageSpend	GroupSize	OnlineReservation	DeliveryOrder	LoyaltyProgramMember	WaitTime	ServiceRating	FoodRating	AmbianceRating	HighSatisfaction
count	1500.000000	1500.000000	1500.000000	1500.000000	1500.000000	1500.000000	1500.000000	1500.000000	1500.000000	1500.000000	1500.000000	1500.000000	1500.000000
mean	1403.500000	43.832000	85921.890000	105.659004	5.035333	0.296667	0.405333	0.480000	30.163550	3.044000	2.997333	2.987333	0.134000
std	433.157015	14.967157	38183.051749	52.381849	2.558864	0.456941	0.491120	0.499766	17.214184	1.423405	1.418920	1.450716	0.340766
min	654.000000	18.000000	20012.000000	10.306127	1.000000	0.000000	0.000000	0.000000	0.001380	1.000000	1.000000	1.000000	0.000000
25%	1028.750000	31.750000	52444.000000	62.287907	3.000000	0.000000	0.000000	0.000000	15.235423	2.000000	2.000000	2.000000	0.000000
50%	1403.500000	44.000000	85811.000000	104.626408	5.000000	0.000000	0.000000	0.000000	30.044055	3.000000	3.000000	3.000000	0.000000
75%	1778.250000	57.000000	119159.250000	148.649330	7.000000	1.000000	1.000000	1.000000	45.285649	4.000000	4.000000	4.000000	0.000000
max	2153.000000	69.000000	149875.000000	199.973527	9.000000	1.000000	1.000000	1.000000	59.970762	5.000000	5.000000	5.000000	1.000000

```
# Check for missing values
df.isnull().sum()
```

```
CustomerID            0
Age                   0
Gender                0
Income                0
VisitFrequency        0
AverageSpend          0
PreferredCuisine      0
TimeOfVisit           0
GroupSize             0
DiningOccasion        0
MealType              0
OnlineReservation     0
DeliveryOrder         0
LoyaltyProgramMember  0
WaitTime              0
ServiceRating         0
FoodRating            0
AmbianceRating        0
HighSatisfaction      0
```

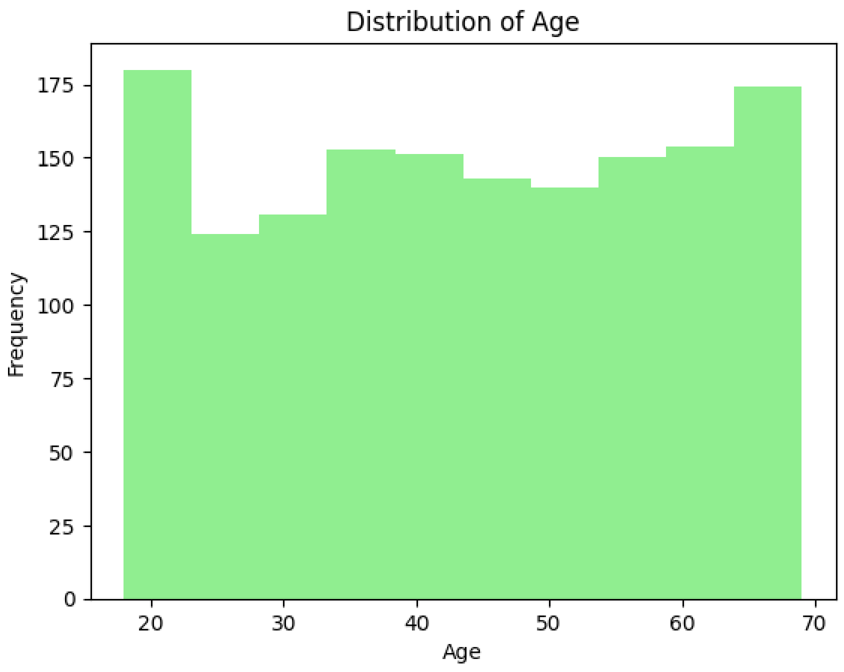
```
# Check for duplicates
df.duplicated().sum()
```

0

Thus, We see that there are no missing values or duplicate tuples present in the dataset

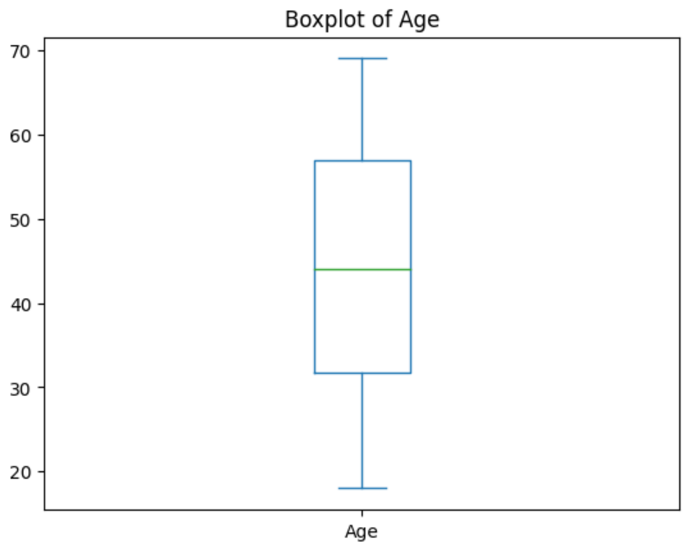
Data Visualisation

```
# Plot the distribution of age
df['Age'].plot.hist(bins=10, color='lightgreen')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.title('Distribution of Age')
plt.show()
```



It can be seen that the population in the dataset contains people of all age categories in a similar amount, except those at 20.

```
# Boxplot for Age
df['Age'].plot.box()
plt.title('Boxplot of Age')
plt.show()
```



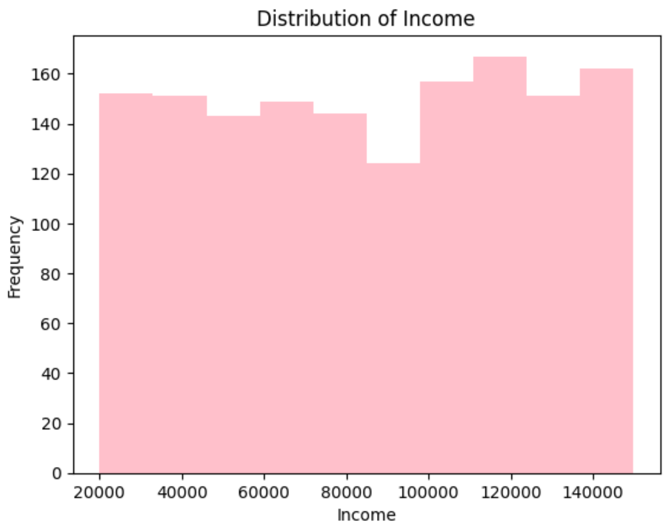
We see that there are no outliers in the variables Age..

```
df['Gender'].value_counts()
```

```
Gender
Female    759
Male      741
Name: count, dtype: int64
```

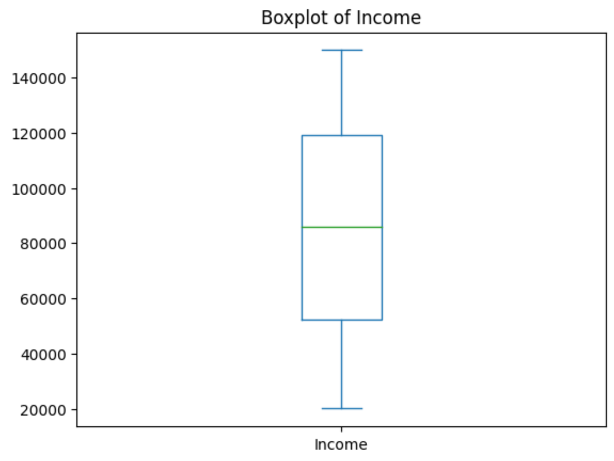
So, the data is balanced.

```
# Plot of the distribution of income
df['Income'].plot.hist(bins=10, color='pink')
plt.xlabel('Income')
plt.ylabel('Frequency')
plt.title('Distribution of Income')
plt.show()
```



Similarly, the people in the data has income from all ranges in a similar amount.

```
# Boxplot of income
df['Income'].plot.box()
plt.title('Boxplot of Income')
plt.show()
```

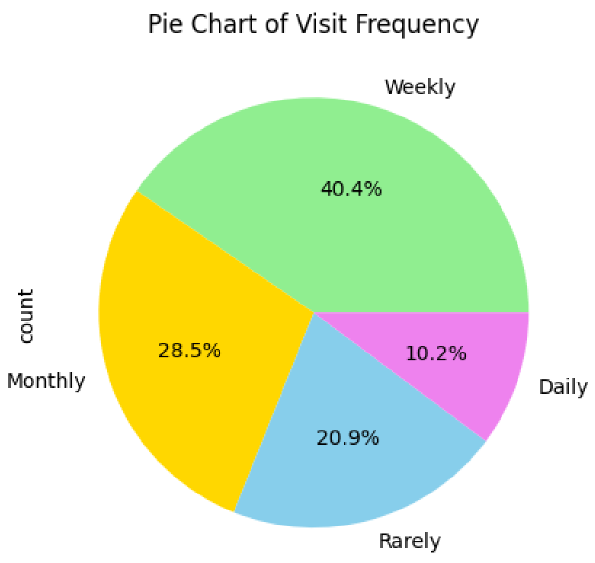


We see that there are no outliers in the variables Income.

```
df['VisitFrequency'].value_counts()
```

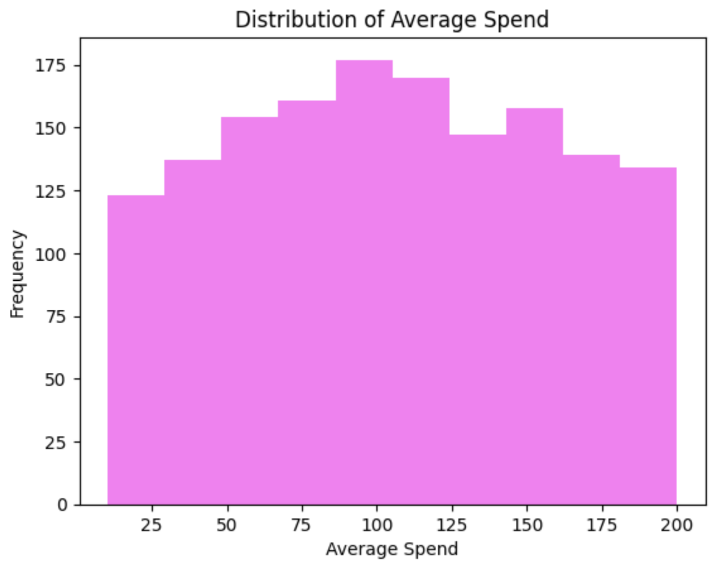
```
VisitFrequency
Weekly      606
Monthly     428
Rarely      313
Daily       153
Name: count, dtype: int64
```

```
# pie chart for VisitFrequency
import matplotlib.pyplot as plt
df['VisitFrequency'].value_counts().plot(kind='pie', autopct='%1.1f%%',
colors=['lightgreen', 'gold', 'skyblue', 'violet'])
plt.title('Pie Chart of Visit Frequency')
plt.show()
```



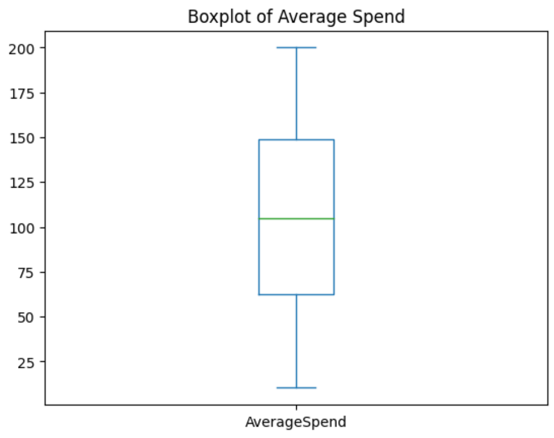
Thus we see that majority of the customers visit the restaurant weekly.

```
# plot AverageSpend
import matplotlib.pyplot as plt
df['AverageSpend'].plot.hist(bins=10, color='violet')
plt.xlabel('Average Spend')
plt.ylabel('Frequency')
plt.title('Distribution of Average Spend')
plt.show()
```



The amount spent by average seems to follow normal distribution with most of them clustered to the middle of the distribution.

```
# Boxplot for Average Spend
df['AverageSpend'].plot.box()
plt.title('Boxplot of Average Spend')
plt.show()
```

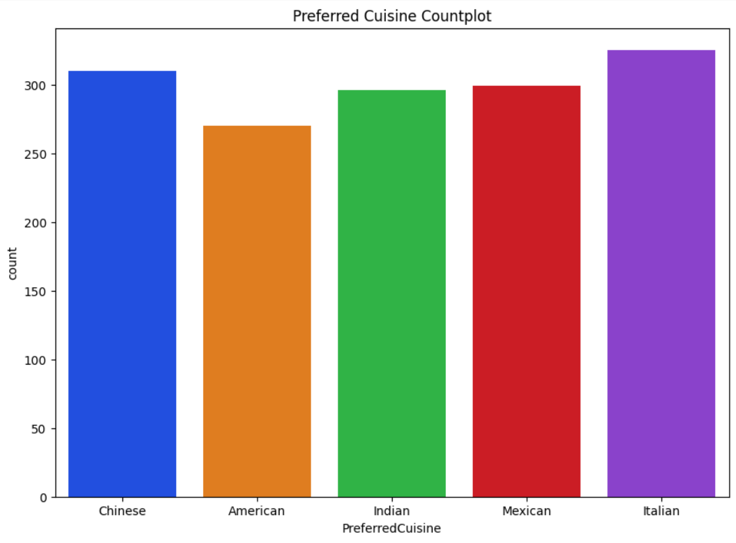


We see that there are no outliers in Average Spend.

```
df['PreferredCuisine'].value_counts()
```

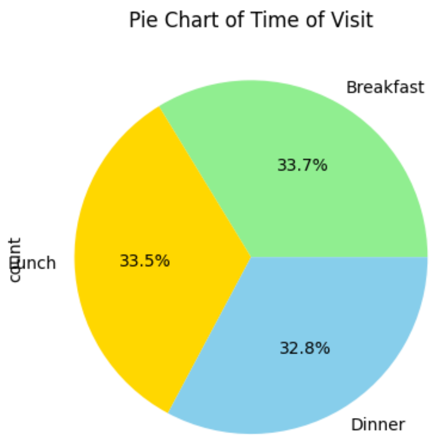
```
PreferredCuisine
Italian     325
Chinese     310
Mexican     299
Indian      296
American    270
Name: count, dtype: int64
```

```
# countplot for preferred Cuisine
cuisine_counts = df['PreferredCuisine'].value_counts()
plt.figure(figsize=(10, 7))

# Use the 'PreferredCuisine' column from df for the x-axis
sns.countplot(x='PreferredCuisine', data=df, palette='bright')
plt.title('Preferred Cuisine Countplot') # Changed title to reflect
plot type
plt.show()
```



So, we see that the majority of the customers seem to like Italian cuisine over other cuisines. However, all of the available cuisines are ordered in somewhat nearer quantities.

```
# plot time of visit
import matplotlib.pyplot as plt
# Plot the distribution of time of visit
df['TimeOfVisit'].value_counts().plot(kind='pie', autopct='%1.1f%%',
colors=['lightgreen', 'gold', 'skyblue', 'violet'])
plt.title('Pie Chart of Time of Visit')
plt.show()
```

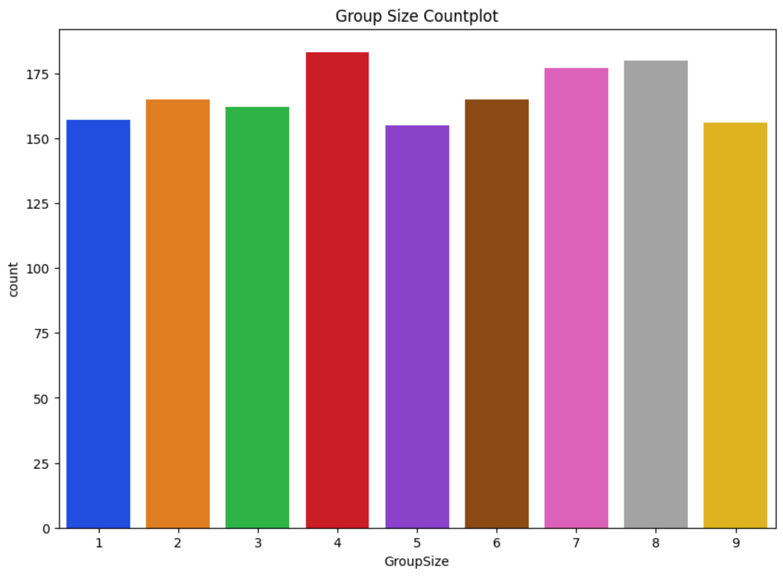


This also is unbalanced. this is very important for statistical analysis.

```
df['GroupSize'].value_counts()
```

```
GroupSize
4    183
8    180
7    177
6    165
2    165
3    162
1    157
9    156
5    155
Name: count, dtype: int64
```

```
# pie plot for GroupSize
import matplotlib.pyplot as plt
group_size_counts = df['GroupSize'].value_counts()
plt.figure(figsize=(10, 7))

# Use the 'GroupSize' column from df for the x-axis
sns.countplot(x='GroupSize', data=df, palette='bright')
plt.title('Group Size Countplot') # Changed title to reflect plot type
plt.show()
```



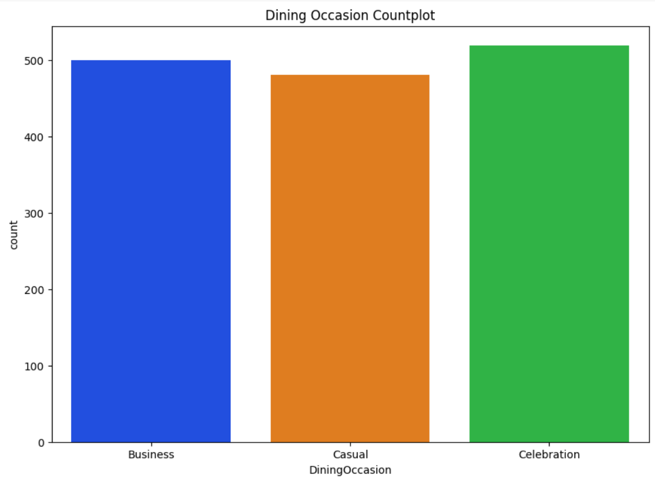
Most of the time, a group of 4 has come together to dine. Let's see in the further analysis if the number of members influences other variables.

```
df['DiningOccasion'].value_counts()
```

```
DiningOccasion
Celebration    519
Business       500
Casual         481
Name: count, dtype: int64
```

```
# ploting the DiningOccassion
import matplotlib.pyplot as plt
dining_occasion_counts = df['DiningOccasion'].value_counts()
plt.figure(figsize=(10, 7))

# Use the 'DiningOccasion' column from df for the x-axis
sns.countplot(x='DiningOccasion', data=df, palette='bright')
```

```
plt.title('Dining Occasion Countplot') # Changed title to reflect plot
type
plt.show()
```



Most of the customers choose the restaurant for celebrations. However, the other two dining occasions also bring together people of a similar number.

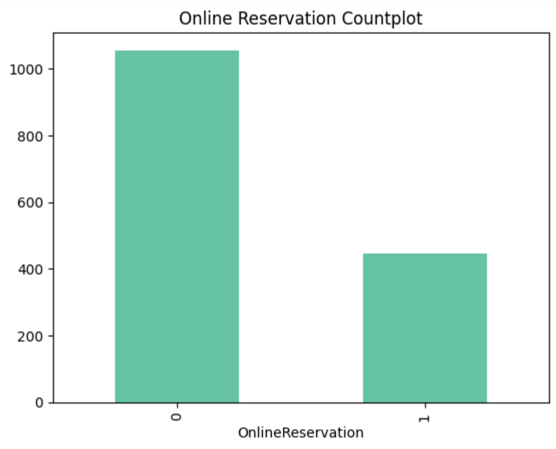
```
df['MealType'].value_counts()
```

```
MealType
Dine-in      751
Takeaway     749
Name: count, dtype: int64
```

```
df['OnlineReservation'].value_counts()
```

```
OnlineReservation
Onsite     1055
Online      445
Name: count, dtype: int64
```

```
# plotting the OnlineReservation
import matplotlib.pyplot as plt
df['OnlineReservation'].value_counts().plot(kind='bar')
plt.title('Online Reservation Countplot')
plt.show()
```

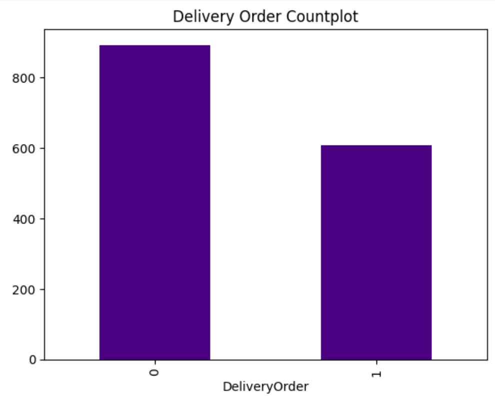


Not many people make online reservations compared to the rest, who come to the restaurant to dine in person.

```
df['DeliveryOrder'].value_counts()
```

```
DeliveryOrder
0    892
1    608
Name: count, dtype: int64
```

```
# plotting the DeliveryOrder
import matplotlib.pyplot as plt
df['DeliveryOrder'].value_counts().plot(kind='bar', color='indigo')
plt.title('Delivery Order Countplot')
plt.show()
```

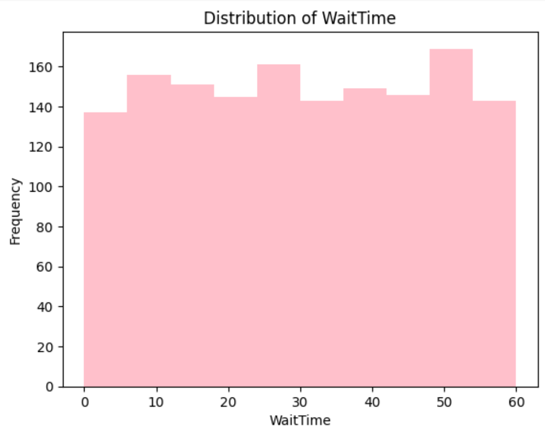


As most of the orders are placed at the restaurant, this has caused fewer orders to be delivered rather than eaten at the restaurant.

```
df['LoyaltyProgramMember'].value_counts()
```

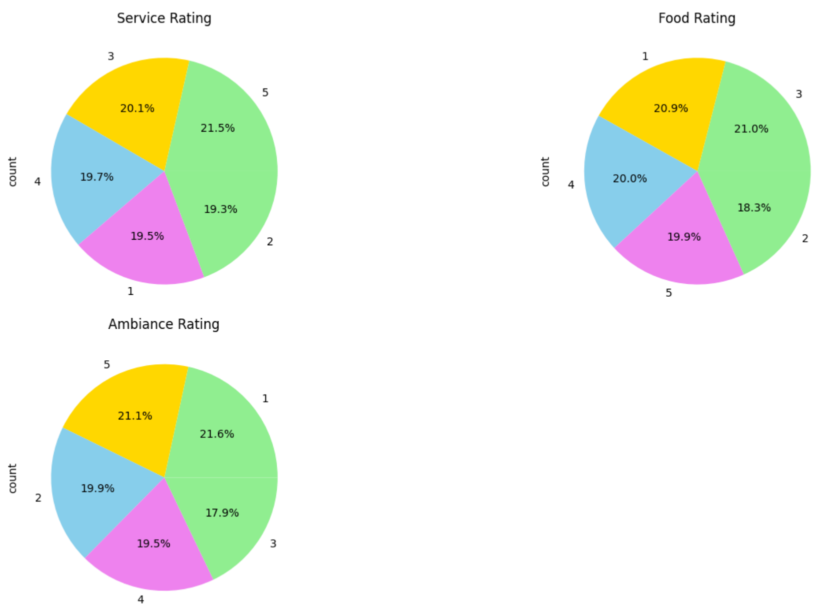
```
LoyaltyProgramMember
0    780
1    720
Name: count, dtype: int64
```

```
# plotting WaitTime
import matplotlib.pyplot as plt
df['WaitTime'].plot.hist(bins=10, color='pink')
plt.xlabel('WaitTime')
plt.ylabel('Frequency')
plt.title('Distribution of WaitTime')
plt.show()
```



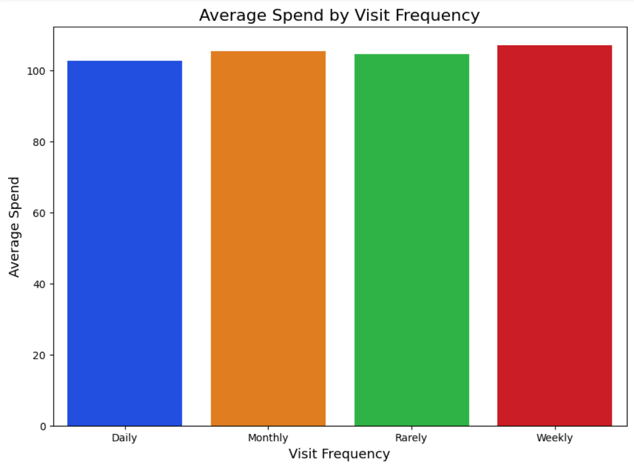
The wait time is also equally distributed.

```python
# Plot Various Ratings
# Create a 2x2 grid of subplots
fig, axes = plt.subplots(2, 2, figsize=(15, 8))

# Plot ServiceRating on the first subplot
df['ServiceRating'].value_counts().plot(ax=axes[0, 0], kind='pie',
autopct='%1.1f%%', colors=['lightgreen', 'gold', 'skyblue', 'violet'])
axes[0, 0].set_title('Service Rating')

# Plot FoodRating on the second subplot
df['FoodRating'].value_counts().plot(ax=axes[0, 1], kind='pie',
autopct='%1.1f%%', colors=['lightgreen', 'gold', 'skyblue', 'violet'])
axes[0, 1].set_title('Food Rating')

# Plot AmbianceRating on the third subplot
df['AmbianceRating'].value_counts().plot(ax=axes[1, 0], kind='pie',
autopct='%1.1f%%', colors=['lightgreen', 'gold', 'skyblue', 'violet'])
axes[1, 0].set_title('Ambiance Rating')

# Make axes[1, 1] blank
axes[1, 1].set_axis_off()

# Adjust spacing between subplots
plt.tight_layout(h_pad=0.5, w_pad=0.5)

# Show the plot
plt.show()
```



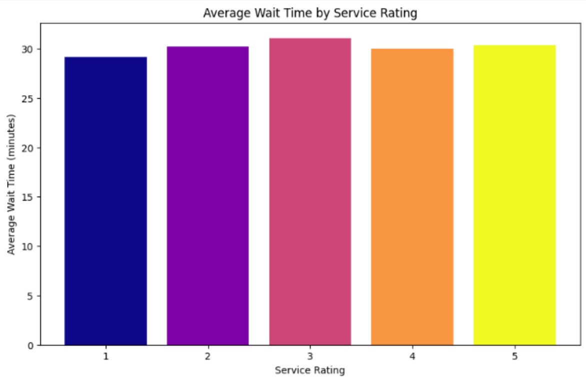
The rating is almost same for all ratings 1, 2, 3, 4 and 5. However, the restaurant has won the rating 5 slightly over the others.

```
# srelationship between VisitFrequency and AverageSpend
# Calculate the average spend for each visit frequency category
avg_spend_by_visit_freq =
df.groupby('VisitFrequency')['AverageSpend'].mean().reset_index()

# Create a bar chart
plt.figure(figsize=(10, 7))
sns.barplot(x='VisitFrequency', y='AverageSpend',
data=avg_spend_by_visit_freq, palette='bright')

# Add title and axis labels
plt.title('Average Spend by Visit Frequency', fontsize=16)
plt.xlabel('Visit Frequency', fontsize=13)
plt.ylabel('Average Spend', fontsize=13)
plt.show()
```



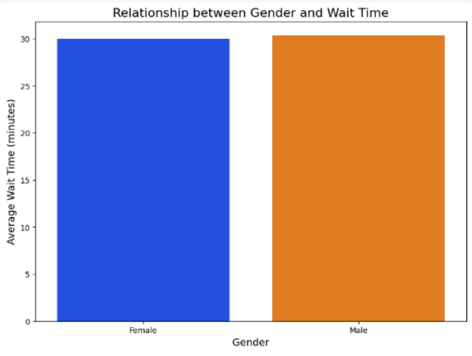
Those who visit during the weekends are slightly more likely to spend more on food.

```
# relationship between Average Wait Time by Service Rating
# Group by ServiceRating and calculate mean WaitTime
grouped_data =
df.groupby('ServiceRating')['WaitTime'].mean().reset_index()

# Define a color map
cmap = plt.cm.plasma

# Create a list of colors based on the color map
colors = [cmap(i) for i in np.linspace(0, 1,
len(grouped_data['ServiceRating']))]

# Plotting the bar chart
plt.figure(figsize=(10, 6))
plt.bar(grouped_data['ServiceRating'], grouped_data['WaitTime'],
color=colors)
plt.xlabel('Service Rating')
plt.ylabel('Average Wait Time (minutes)')
plt.title('Average Wait Time by Service Rating')
plt.xticks(grouped_data['ServiceRating'])
plt.show()
```



The average wait time seems to make no significant difference in the rating of the service. However, Those who gave a rating of 3 were the ones who had to wait the most.

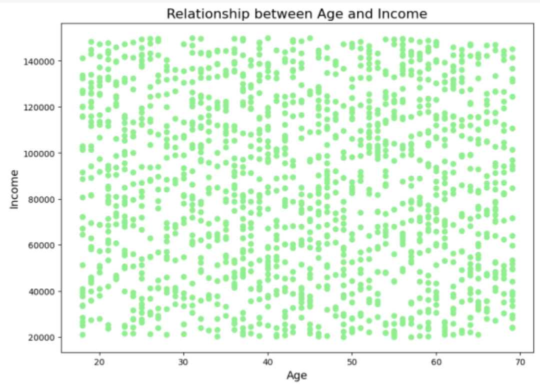
```
# relationship between Gender and WaitTime
# Calculate the average wait time for each gender
avg_wait_time_by_gender =
df.groupby('Gender')['WaitTime'].mean().reset_index()

# Create a bar chart
plt.figure(figsize=(10, 7))
sns.barplot(x='Gender', y='WaitTime', data=avg_wait_time_by_gender,
palette='bright')

# Add title and axis labels
plt.title('Relationship between Gender and Wait Time', fontsize=16)
plt.xlabel('Gender', fontsize=13)
plt.ylabel('Average Wait Time (minutes)', fontsize=13)
plt.show()
```



There is no difference among both male and female in given wait time as both of them have to wait a similar time.

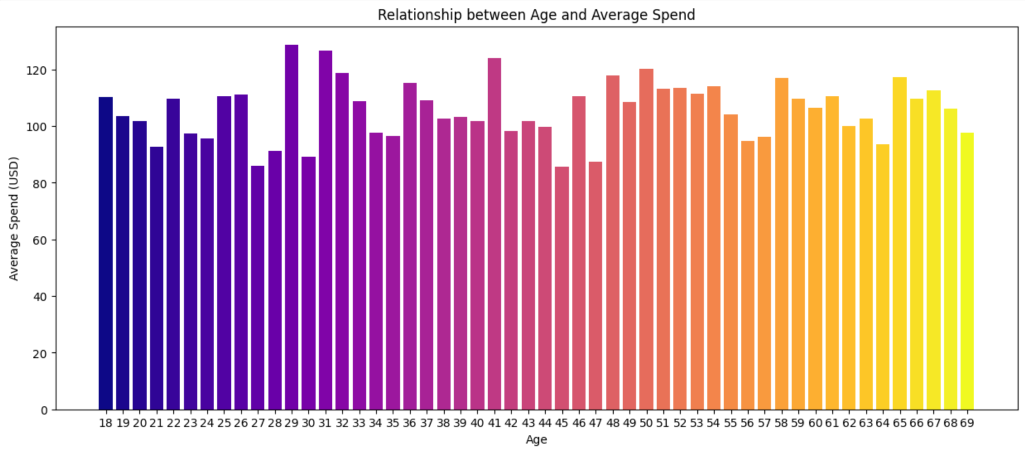
```
# relationship between Age and Income
# Create a scatter plot
plt.figure(figsize=(10, 7))
plt.scatter(df['Age'], df['Income'], color='lightgreen')

# Add title and axis labels
```

```
plt.title('Relationship between Age and Income', fontsize=16)
plt.xlabel('Age', fontsize=13)
plt.ylabel('Income', fontsize=13)
plt.show()
```



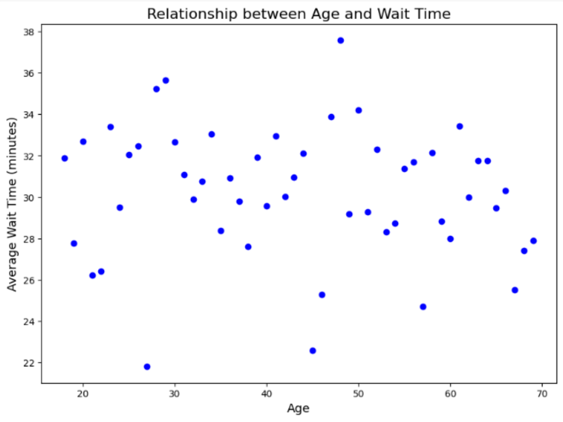
The relationship between the customers' age and their income seems completely random.

```
# relationship between Age and AverageSpend
# Group by age and calculate mean AverageSpend
grouped_data = df.groupby('Age')['AverageSpend'].mean().reset_index()

# Define a color map
cmap = plt.cm.plasma

# Create a list of colors based on the color map
colors = [cmap(i) for i in np.linspace(0, 1, len(grouped_data['Age']))]

# Plotting the bar chart
plt.figure(figsize=(15, 6))
plt.bar(grouped_data['Age'], grouped_data['AverageSpend'],
color=colors)
plt.xlabel('Age')
plt.ylabel('Average Spend (USD)')
plt.title('Relationship between Age and Average Spend')
plt.xticks(grouped_data['Age'])
plt.show()
```



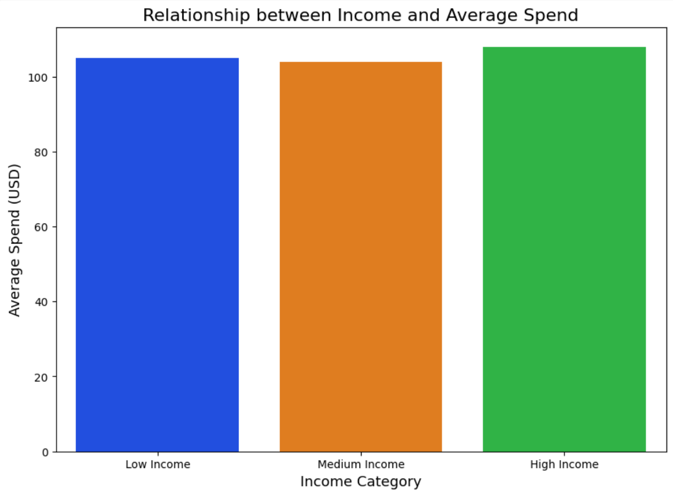
Above, middle-aged customers are more likely to spend more.

```
# relationship between Age and WaitTime
# Calculate the average wait time for each age group
avg_wait_time_by_age =
df.groupby('Age')['WaitTime'].mean().reset_index()

# Create a scatter plot
plt.figure(figsize=(10, 7))
plt.scatter(avg_wait_time_by_age['Age'],
avg_wait_time_by_age['WaitTime'], color='blue')

# Add title and axis labels
plt.title('Relationship between Age and Wait Time', fontsize=16)
plt.xlabel('Age', fontsize=13)
plt.ylabel('Average Wait Time (minutes)', fontsize=13)
plt.show()
```



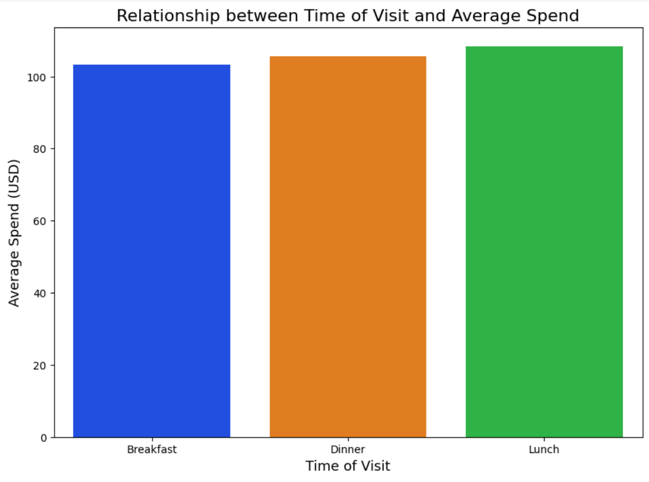
The average wait time seems to decrease with a slight increase in age.

```
# relationship between Income(as categories) and AverageSpend
# Define income categories
income_categories = ['Low Income', 'Medium Income', 'High Income']

# Create income category based on income range
df['Income_Category'] = pd.cut(df['Income'], 3,
labels=income_categories)

# Group by income category and calculate mean AverageSpend
grouped_data =
df.groupby('Income_Category')['AverageSpend'].mean().reset_index()

# Create a bar chart
plt.figure(figsize=(10, 7))
sns.barplot(x='Income_Category', y='AverageSpend', data=grouped_data,
palette='bright')

# Add title and axis labels
plt.title('Relationship between Income and Average Spend', fontsize=16)
plt.xlabel('Income Category', fontsize=13)
plt.ylabel('Average Spend (USD)', fontsize=13)
plt.show()
```



Obviously, the high-earner customers are the ones who spend more compared to the rest.

```
# relationship between TimeOfVisit and AverageSpend
# Group by TimeOfVisit and calculate mean AverageSpend
grouped_data =
df.groupby('TimeOfVisit')['AverageSpend'].mean().reset_index()

# Create a bar chart
plt.figure(figsize=(10, 7))
sns.barplot(x='TimeOfVisit', y='AverageSpend', data=grouped_data,
palette='bright')

# Add title and axis labels
plt.title('Relationship between Time of Visit and Average Spend',
fontsize=16)
plt.xlabel('Time of Visit', fontsize=13)
plt.ylabel('Average Spend (USD)', fontsize=13)
plt.show()
```



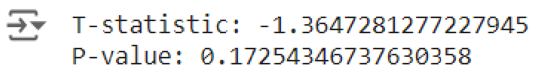
Lunchtime is the peak hour of the day in this restaurant. So, it can be very effective to implement customer satisfaction strategies at this time of the day.

STATISTICAL TESTS

1. Tests for the difference between the means of two independent groups.

Objective: Determine if there is a significant difference in average satisfaction ratings (e.g., ServiceRating, FoodRating, AmbianceRating) between two groups (e.g., gender, loyalty program members vs non-members).

```
from scipy import stats

# Separate the data into two groups based on gender
male_ratings = df[df['Gender'] == 'Male']['ServiceRating']
female_ratings = df[df['Gender'] == 'Female']['ServiceRating']

# Perform the independent samples t-test
t_statistic, p_value = stats.ttest_ind(male_ratings, female_ratings)

# Print the results
print("T-statistic:", t_statistic)
print("P-value:", p_value)
```



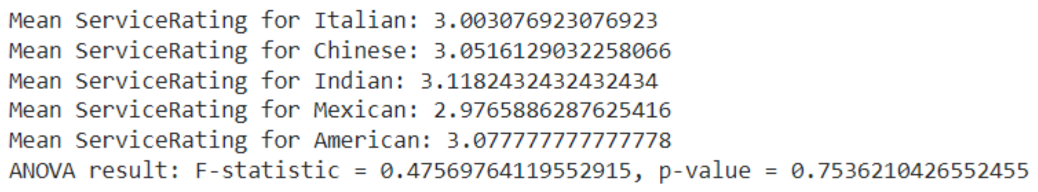
```
T-statistic: -1.3647281277227945
P-value: 0.17254346737630358
```

There is no significant difference in ServiceRating between genders.

2. Tests for the difference between the means of three or more groups.

Objective: Determine if there is a significant difference in satisfaction ratings across multiple groups (e.g., different cuisine preferences).

```
import pandas as pd
from scipy.stats import f_oneway, kruskal
import statsmodels.api as sm
from statsmodels.formula.api import ols
from statsmodels.stats.multicomp import pairwise_tukeyhsd

# Select the groups to compare - Different cuisine preferences
group1 = df[df['PreferredCuisine'] == 'Italian']['ServiceRating']
group2 = df[df['PreferredCuisine'] == 'Chinese']['ServiceRating']
group3 = df[df['PreferredCuisine'] == 'Indian']['ServiceRating']
group4 = df[df['PreferredCuisine'] == 'Mexican']['ServiceRating']
group5 = df[df['PreferredCuisine'] == 'American']['ServiceRating']

# Calculate the mean satisfaction ratings for each group
mean1, mean2, mean3, mean4, mean5 = group1.mean(), group2.mean(),
group3.mean(), group4.mean(), group5.mean()
print(f"Mean ServiceRating for Italian: {mean1}")
print(f"Mean ServiceRating for Chinese: {mean2}")
print(f"Mean ServiceRating for Indian: {mean3}")
print(f"Mean ServiceRating for Mexican: {mean4}")
print(f"Mean ServiceRating for American: {mean5}")

# Conduct one-way ANOVA
anova_result = f_oneway(group1, group2, group3, group4, group5)
print(f"ANOVA result: F-statistic = {anova_result.statistic}, p-value = 
{anova_result.pvalue}")
```



```
Mean ServiceRating for Italian: 3.003076923076923
Mean ServiceRating for Chinese: 3.0516129032258066
Mean ServiceRating for Indian: 3.1182432432432434
Mean ServiceRating for Mexican: 2.9765886287625416
Mean ServiceRating for American: 3.077777777777778
ANOVA result: F-statistic = 0.47569764119552915, p-value = 0.7536210426552455
```

These values represent the average ServiceRating given by customers for each type of cuisine. The means are quite close to each other, indicating that the average service ratings for different cuisines are similar.

Since the p-value is greater than 0.05, we fail to reject the null hypothesis. This means that there is no statistically significant difference in the average ServiceRating among the different cuisine groups (Italian, Chinese, Indian, Mexican, American).

3. Compares the variances of two groups.

Objective: Determine if the variance in satisfaction ratings differs significantly between two groups.

```
# Separate the data into two groups based on income
high_income = df[df['Income'] > df['Income'].mean()]['ServiceRating']
low_income = df[df['Income'] <= df['Income'].mean()]['ServiceRating']

# Perform Bartlett's test for equality of variances
bartlett_result = stats.bartlett(high_income, low_income)

# Print the results
print("Bartlett's test result: chi-squared statistic = {}, p-value = 
{}".format(bartlett_result.statistic, bartlett_result.pvalue))
```

```
Bartlett's test result: chi-squared statistic = 0.5989738233734577, p-
value = 0.43896982701133525
```

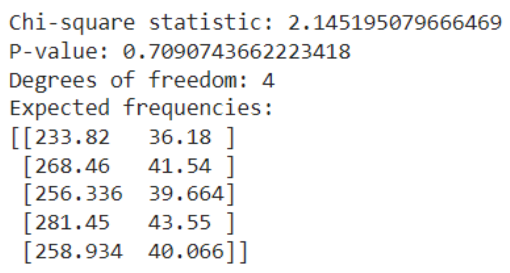
Since the p-value is greater than 0.05, we fail to reject the null hypothesis. This means that there is no statistically significant difference in the variances between the groups.

4. Tests for relationships between categorical variables.

Objective: Explore associations between categorical variables (e.g., preferred cuisine and high satisfaction).

```
from scipy.stats import chi2_contingency

# Create a contingency table
crosstab = pd.crosstab(df['PreferredCuisine'], df['HighSatisfaction'])
```

```
# Perform the chi-square test
chi2, p, dof, expected = chi2_contingency(crosstab)

# Print the results
print("Chi-square statistic:", chi2)
print("P-value:", p)
print("Degrees of freedom:", dof)
print("Expected frequencies:")
print(expected)
```



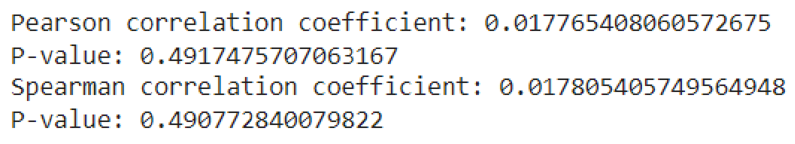
```
Chi-square statistic: 2.145195079666469
P-value: 0.7090743662223418
Degrees of freedom: 4
Expected frequencies:
[[233.82   36.18 ]
 [268.46   41.54 ]
 [256.336  39.664]
 [281.45   43.55 ]
 [258.934  40.066]]
```

Since the p-value is greater than 0.05, we fail to reject the null hypothesis. This means that there is no statistically significant association between PreferredCuisine and HighSatisfaction.

5. Measures the strength and direction of the linear relationship between two continuous variables.

Objective: Determine the correlation between continuous variables (e.g., Age and AverageSpend) and satisfaction ratings.

```
# Compute Pearson correlation coefficient between Age and AverageSpend
pearson_corr, p_value = stats.pearsonr(df['Age'], df['AverageSpend'])

# Print the results
print("Pearson correlation coefficient:", pearson_corr)
print("P-value:", p_value)

# Compute Spearman correlation coefficient between ServiceRating and
WaitTime
spearman_corr, p_value = stats.spearmanr(df['ServiceRating'],
df['WaitTime'])

# Print the results
print("Spearman correlation coefficient:", spearman_corr)
print("P-value:", p_value)
```



```
Pearson correlation coefficient: 0.017765408060572675
P-value: 0.4917475707063167
Spearman correlation coefficient: 0.017805405749564948
P-value: 0.490772840079822
```

Pearson correlation coefficient of 0.017765 suggests a very weak positive linear relationship between Age and AverageSpend. The p-value of 0.491748 is higher than the significance level of 0.05. This indicates that the observed correlation is not statistically significant. We fail to reject the null hypothesis, which means there is no evidence to suggest a significant linear relationship between Age and average speed.

Spearman correlation coefficient of 0.017805 suggests a very weak positive monotonic relationship between ServiceRating and WaitTime. The p-value of 0.490773 is higher than 0.05, indicating that the observed correlation is not statistically significant. We fail to reject the null hypothesis, meaning there is no evidence to suggest a significant monotonic relationship between ServiceRating and WaitTime.

Both the Pearson and Spearman correlation analyses indicate that there are no statistically significant relationships between the pairs of variables tested:

- Age and AverageSpend: No significant linear relationship.
- ServiceRating and WaitTime: No significant monotonic relationship.

Conclusions…

This analysis of the Restaurant Customer Satisfaction Dataset yielded the following key insights:

1. Differences Between Groups:
    - Service Ratings by Gender: No significant difference between male and female customers' service ratings.
    - Service Ratings by Preferred Cuisine: No significant difference in service ratings across various cuisine preferences.
2. Comparison of Variances:
    - Income Groups: No significant difference in service rating variances between high-income and low-income groups.
3. Relationships Between Categorical Variables:
    - Preferred Cuisine and High Satisfaction: No significant association between preferred cuisine and high satisfaction.
4. Correlation Between Continuous Variables:
    - Age and AverageSpend: Very weak and non-significant positive relationship.
    - ServiceRating and WaitTime:** Very weak and non-significant positive relationship.

Further, this analysis provided a foundational understanding of customer satisfaction drivers. However, further research can bring forth comprehensive insights.