FoodHub Data Analysis

Project Python Foundations

Yair Brama - June 2024

Contents / Agenda

- Executive Summary
- Business Problem Overview and Solution Approach
- Data Overview
- EDA - Univariate Analysis
- EDA - Multivariate Analysis
- Appendix

Executive Summary

Observation from the analysis -

- 40% of the orders are not rated
- 1200 customers, 178 restaurants
- 4 cuisine types are almost 60% of the orders - American, Japanese, Italian, Chinese
- The mean cost id $16.5, 75% under $23
- Weekends are much more popular for ordering through the app
- No correlation between the cost of the order and the time it takes to deliver or prepare the food
- No correlation between the prep time and the rating
- Lower rating was given to longer delivery time
- Shorter delivery time on the weekend - makes sense considering traffic
- Popular cuisine types are similar in cost and in prep time
- Higher rating was given to the more expensive orders
- There are more ‘cheap’ orders purchases through the app than expensive ones.

Executive Summary

Actionable Insights -

- *The top successful restaurants offer the most popular cuisines - American, Japanese, Italian and Chinese*
- *Delivery time is shorter on the weekends, probably due to traffic (less traffic during the weekends)*
- *Ratings data is not sufficient (40% of the orders are not rated)*
- *The cost of order data shows that the app is selling a lot more 'cheap' orders than higher priced ones.*

Recommendations -

- *Focus the service on the 4 leading cuisine types - American, Japanese, Italian and Chinese*
- *On weekdays, limit the delivery range, to allow better user experience*
- *Encourage the users to rate the experience to get more useful insights*
- *Encourage users to purchase the higher priced orders, by offering discounts and more marketing. In the long run, this can contribute to higher revenue for FoodHub.*

Business Problem Overview and Solution Approach

Problem Overview

FoodHub offers access to multiple restaurants through a single smartphone app. FoodHub earns money by collecting a fixed margin of the delivery order from the restaurants.

Our analysis answers the questions:

Which restaurants drive the most revenue in FoodHub?

What makes these restaurants so popular?

What other parameters drive more orders/revenue?

What can we do to generate more revenue?

Solution approach / methodology

FoodHub stores the data of the different orders made by the registered customers in their online portal. By running data analysis on these orders, we can do statistical analysis on variables such as time (preparation, delivery), cost and ratings of the orders. We can also find the correlation between some of these variables and draw some relevant conclusions

Data Overview

Question 1: How many rows and columns are present in the data?

Answer:: 1898 rows, 9 columns

Question 2: What are the datatypes of the different columns in the dataset?

Answer:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1898 entries, 0 to 1897
Data columns (total 9 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   order_id               1898 non-null   int64
 1   customer_id            1898 non-null   int64
 2   restaurant_name        1898 non-null   object
 3   cuisine_type           1898 non-null   object
 4   cost_of_the_order      1898 non-null   float64
 5   day_of_the_week        1898 non-null   object
 6   rating                 1898 non-null   object
 7   food_preparation_time  1898 non-null   int64
 8   delivery_time          1898 non-null   int64
dtypes: float64(1), int64(4), object(4)
memory usage: 133.6+ KB
```

Question 3:Are there any missing values in the data?

Answer: No. All rows/columns are filled

Question 4: What is the minimum, average, and maximum time it takes for food to be prepared once an order is placed?

Answer: minimum: 20 minutes
average: 27.37 minutes
maximum: 35 minutes

Question 5: How many orders are not rated?

Answer: 736 orders are not rated (59%)

Univariate Analysis

Question 6: Explore all the variables and provide observations on their distribution

No. of unique Order IDs:

1898

No. of Cuisine types

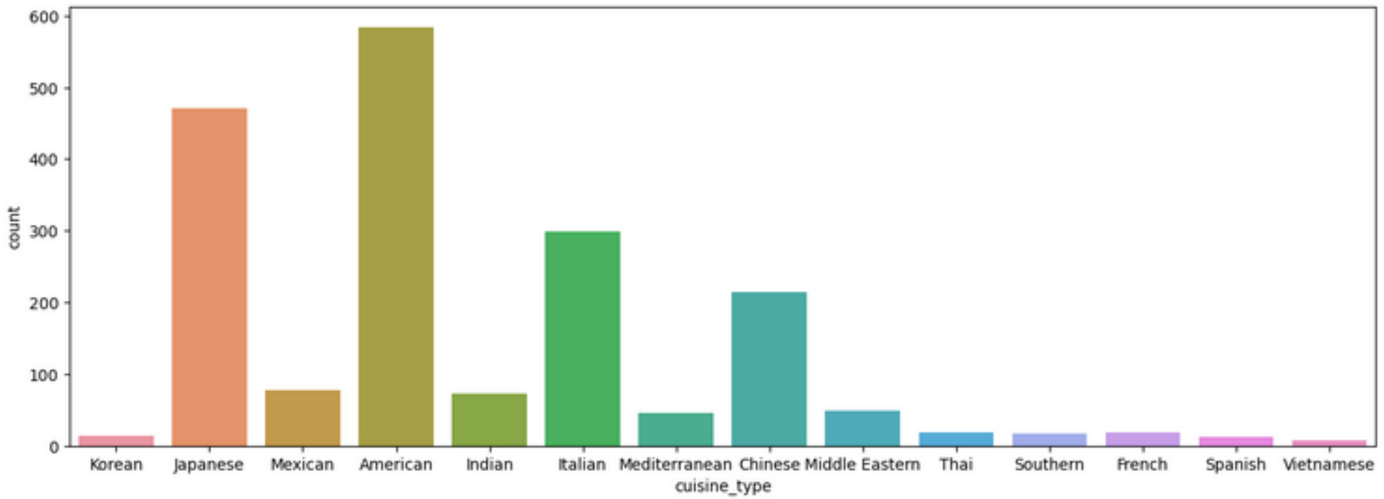
14

Unique Customer IDs

1200

Unique Restaurant names

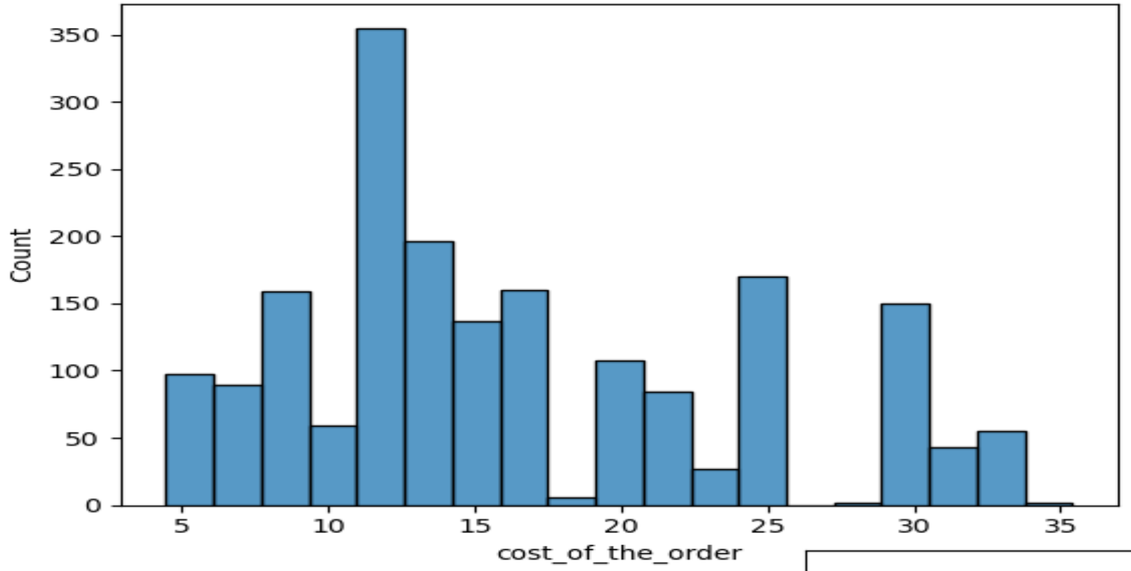
178



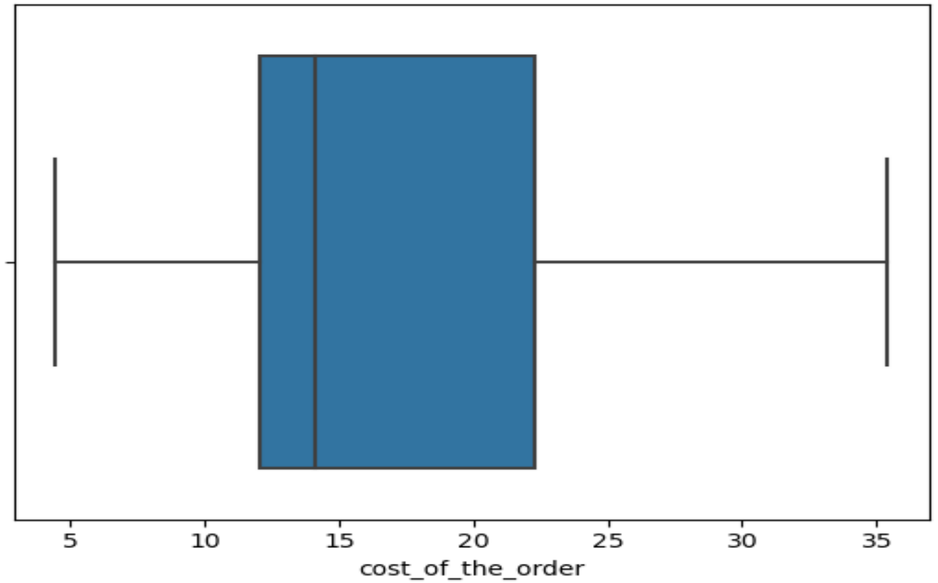
Observation – The leading cuisine types are - American, Japanese, Italian and Chinese

Univariate Analysis

Cost of the order

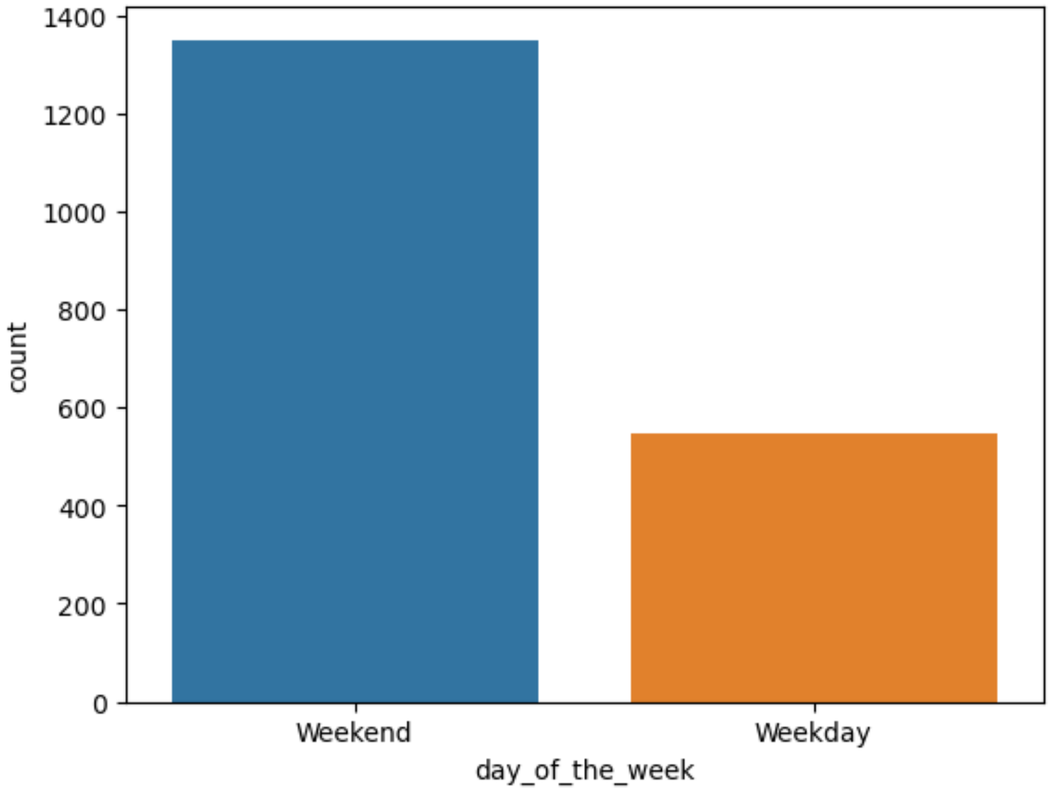


Observation – The average cost of order is $16.5, the median cost is below $15
The cost of order is right-skewed, meaning there are more cheap orders purchased through the app.

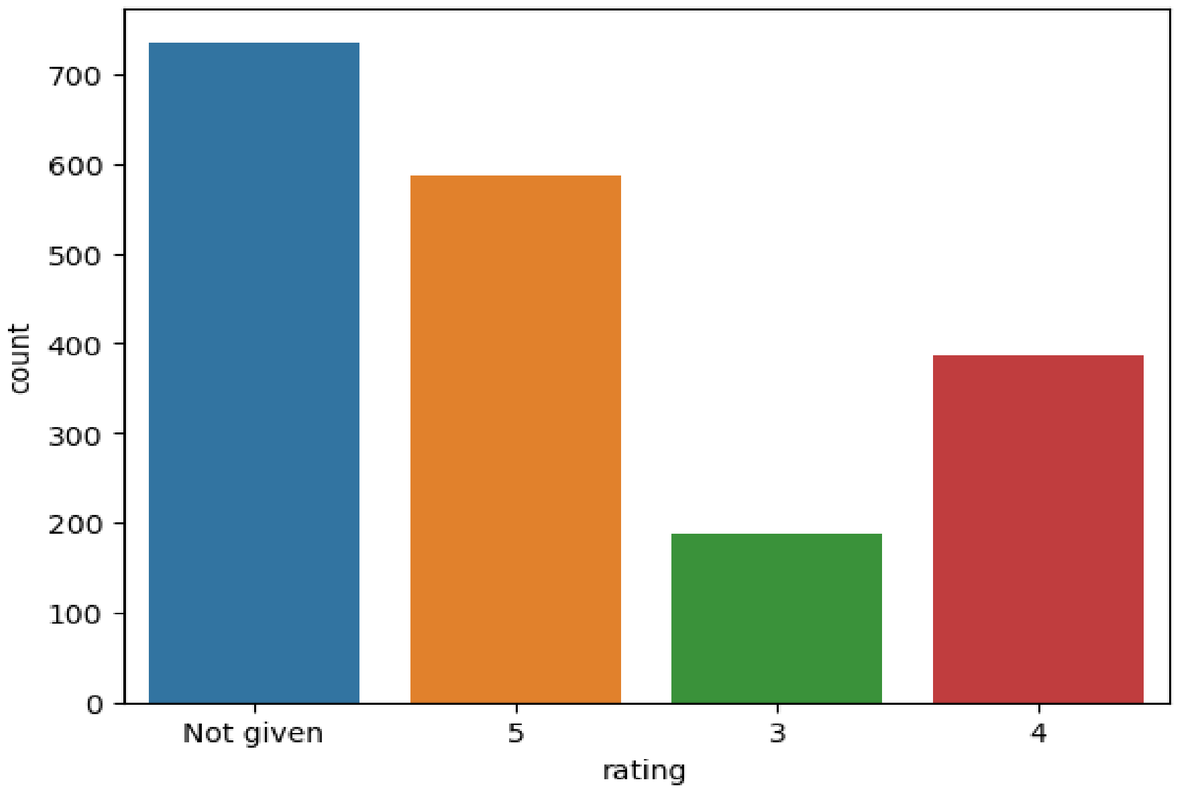


Univariate Analysis - Cont.

Days of the week

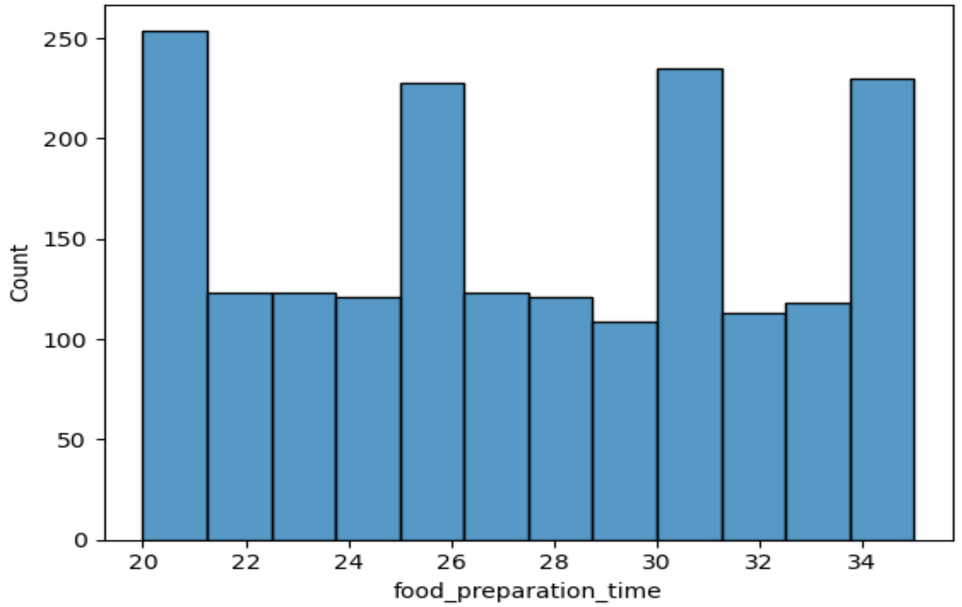


Rating

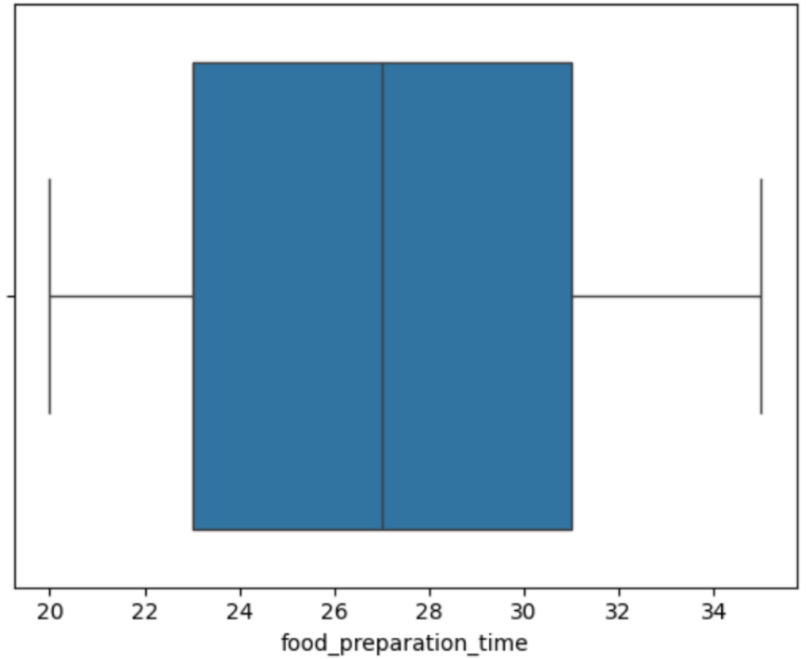


Observation – The weekends are much more popular than weekdays in term of ordering through the app. Since people need to eat during the week as well, it is worth putting more effort in enhancing the usage of the app during the week.
The rating data shows that ~40% of the users don't bother to rate the service, and that says that rating cannot be a reliable factor in deriving conclusions. FoodHub should find out a way to encourage the users to rate the service, so it can be used to improve service to the customers and revenue to the company.

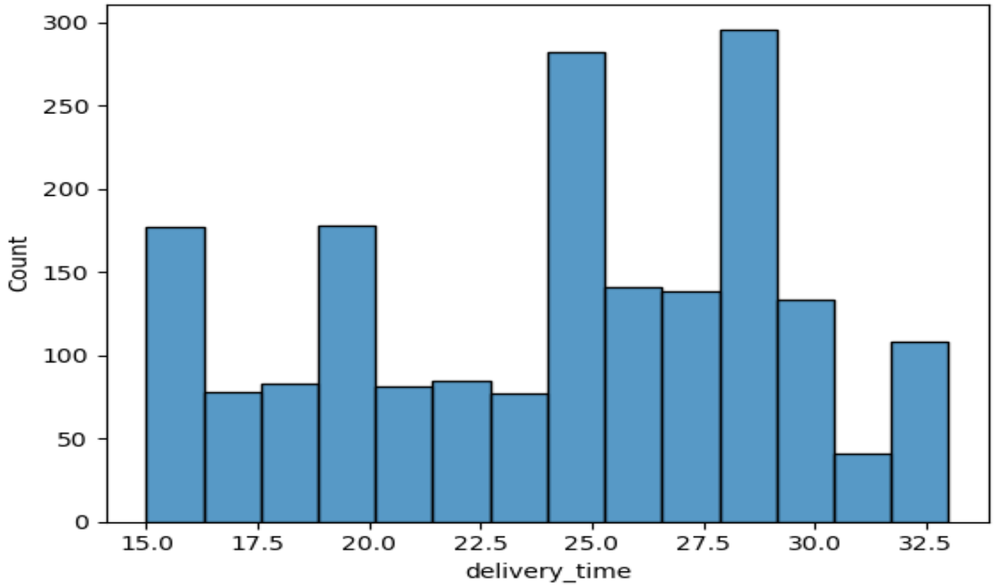
Univariate Analysis - Cont.



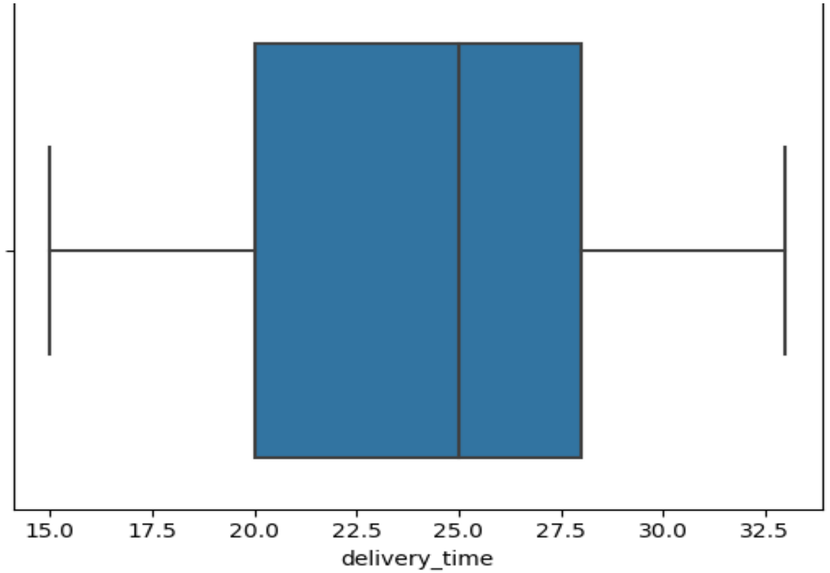
Food Preparation time



Observation – Preparation time is normally distributed around 27 minutes, while delivery time is left-skewed. 50% of the orders are delivered within 25 minutes



Delivery time



Univariate Analysis - Cont.

Question7: Which are the top 5 restaurants in terms of the number of orders received?

ShakeShack	**219**
The Meat ball Shop	**132**
Blue Ribbon Sushi	**119**
Blue Ribbon Fried Chicken	**96**
Parm	**68**

Question 8: Which is the most popular cuisine on weekends?

Answer: American, 415 orders

Question 9: What percentage of the orders cost more than 20 dollars?

Answer: The number of total orders that cost above 20 dollars is: **555**

Percentage of orders above 20 dollars: **29.24 %**

Question 10: What is the mean order delivery time?

Answer: The mean delivery time for this dataset is **24.16** minutes

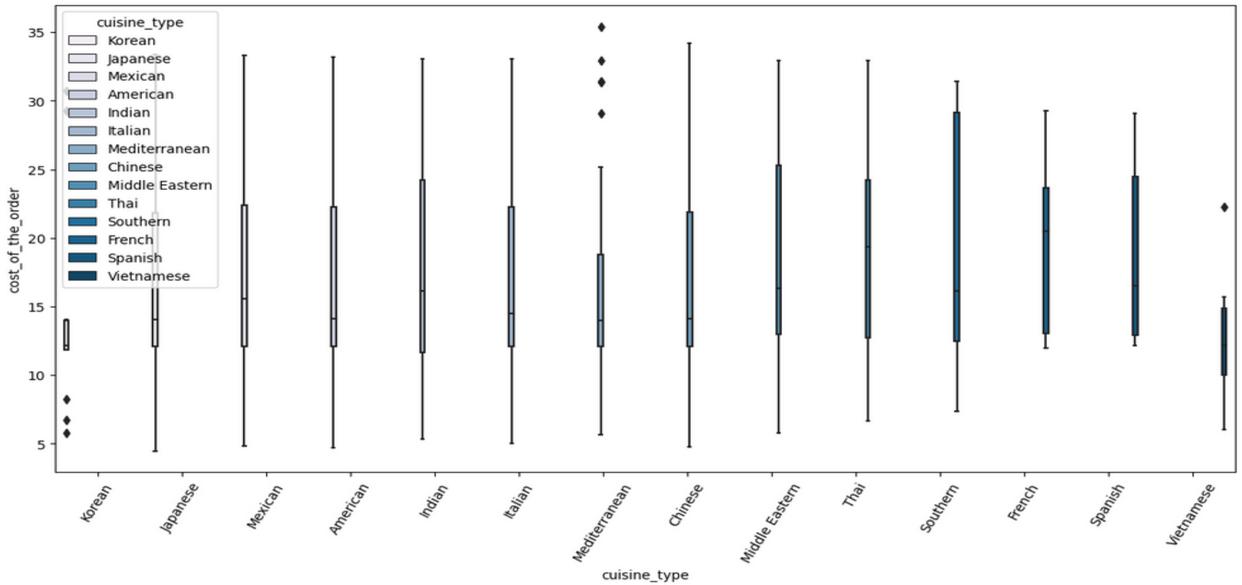
Question 11: The company has decided to give 20% discount vouchers to the top 3 most frequent customers. Find the IDs of these customers and the number of orders they placed.

Customer ID	**No. of Orders**
52832	**13**
47440	**10**
83287	**9**

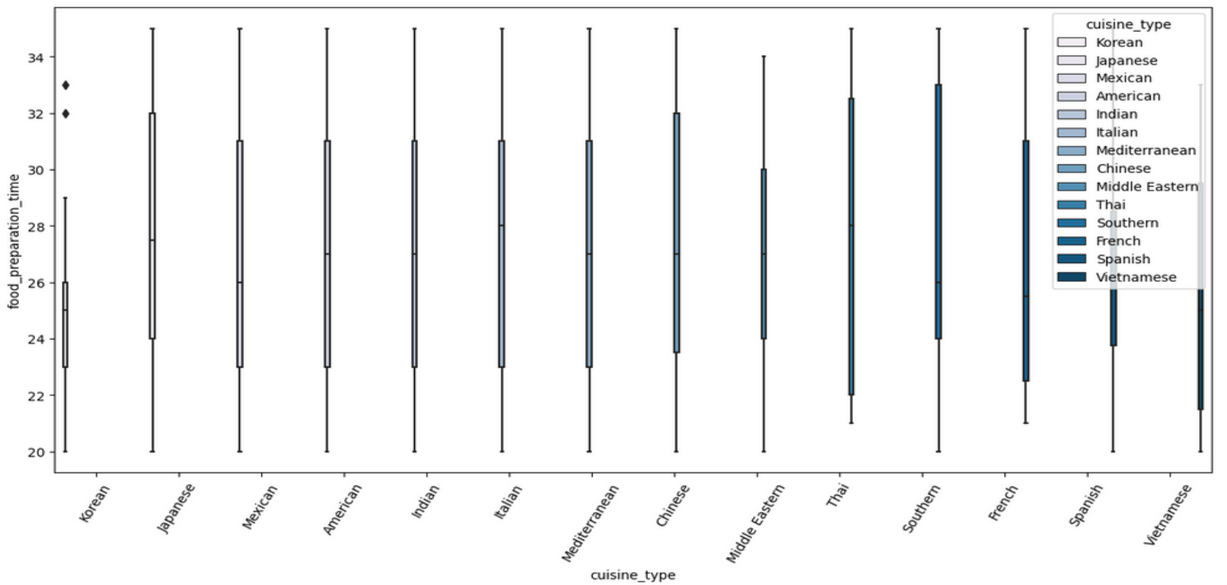
Multivariate Analysis

Question 12: Perform a multivariate analysis to explore relationships between the important variables in the dataset.

Cuisine vs Cost of the order



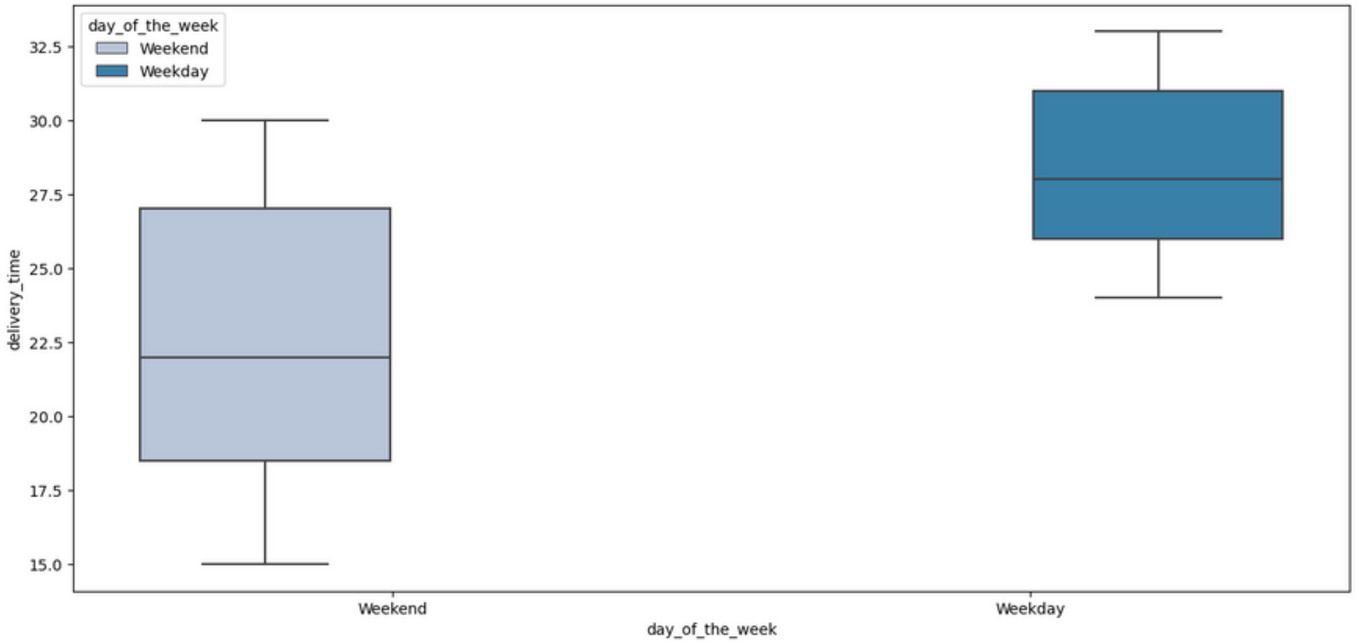
Cuisine vs Food Preparation time



Observation – The leading cuisine types (American, Japanese, Italian and Chinese) are not significantly different in terms of cost and preparation times

Multivariate Analysis - Cont.

Day of the Week vs Delivery time



Observation – Delivery time is significantly lower on weekend. That leads to higher rating and more business that can be done through the app on the weekends.

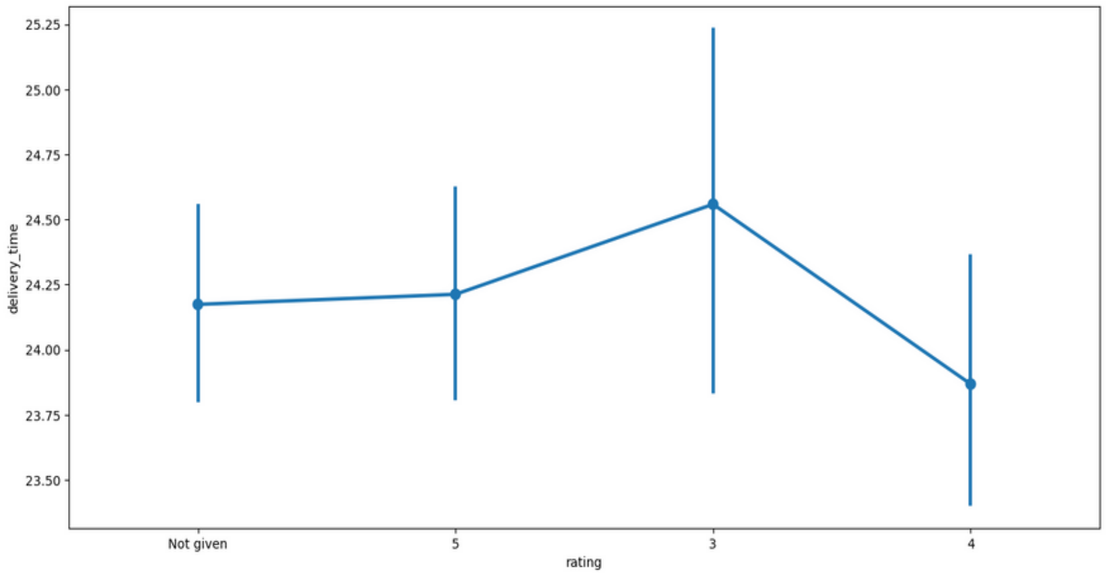
Revenue generated by the restaurants:

restaurant_name	
Shake Shack	3579.53
The Meatball Shop	2145.21
Blue Ribbon Sushi	1903.95
Blue Ribbon Fried Chicken	1662.29
Parm	1112.76
RedFarm Broadway	965.13
RedFarm Hudson	921.21
TAO	834.50
Han Dynasty	755.29
Blue Ribbon Sushi Bar & Grill	666.62
Rubirosa	660.45
Sushi of Gari 46	640.87
Nobu Next Door	623.67
Five Guys Burgers and Fries	506.47

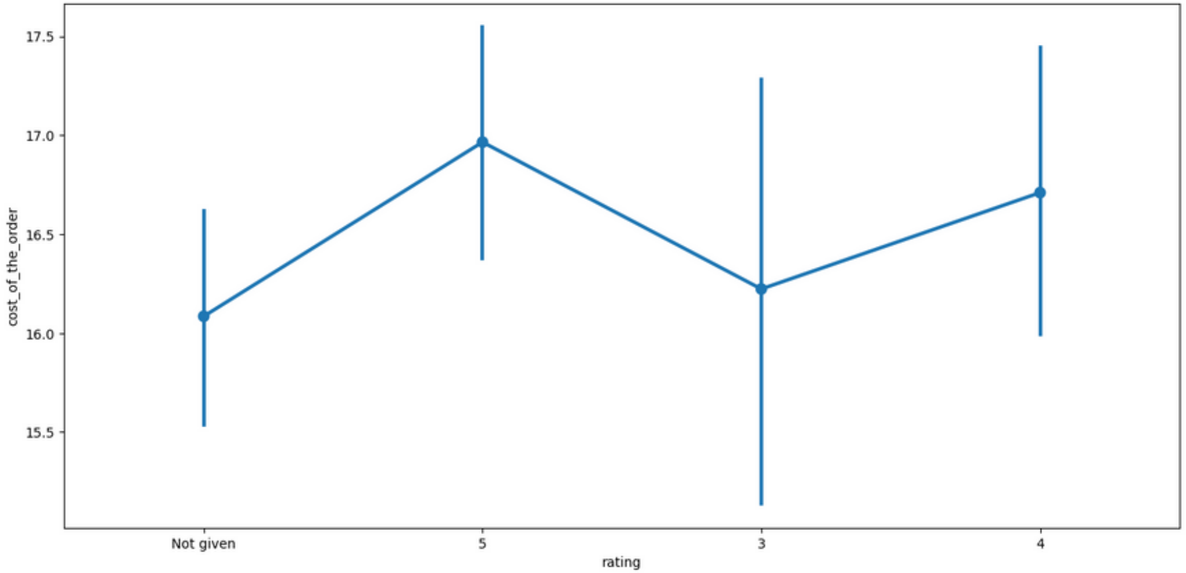
Observation – The leading revenue- generating restaurants are all serving the popular cuisine types

Multivariate Analysis - Cont.

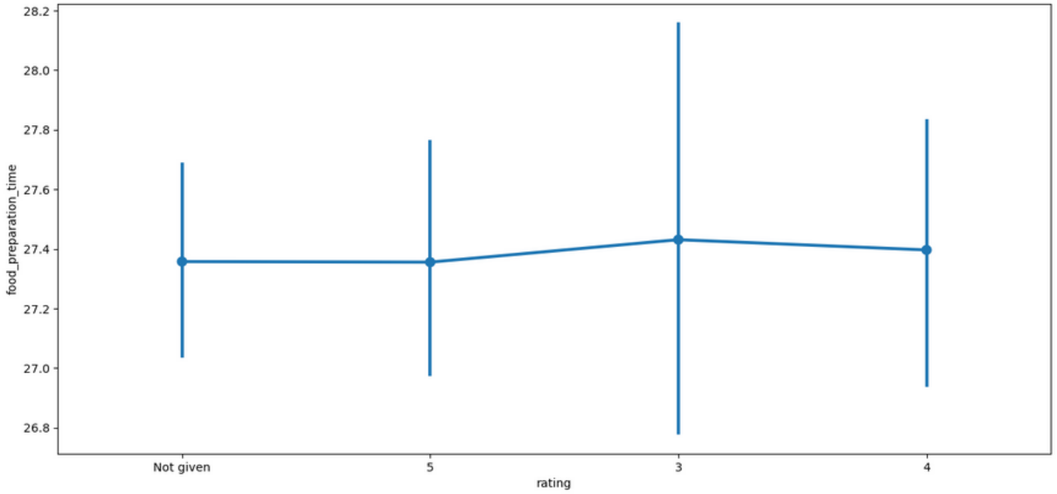
Rating vs Delivery time



Rating vs Cost of the order



Rating vs Food preparation time



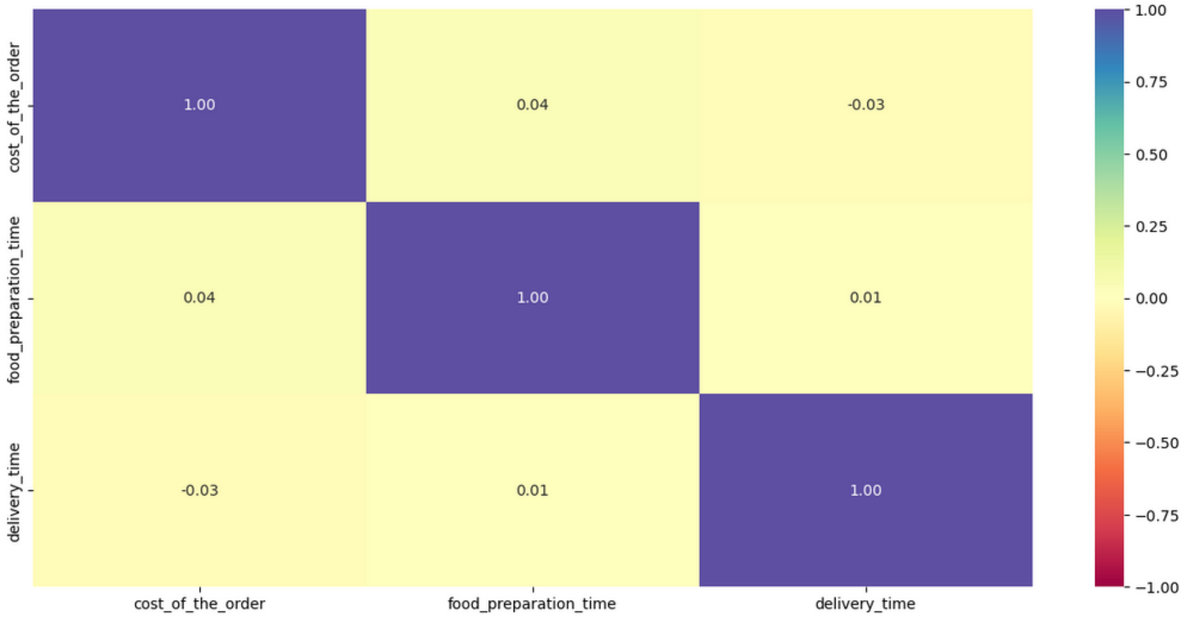
Observation – The rating is NOT impacted by the food preparation time. Better ratings are given to the higher cost orders, which might be due to the quality of the orders, but it can also be due to the human tendency to ‘justify’ the higher expense.

The delivery time has a relation to the rating in the following way – Longer delivery times cause lower rating. Shorter delivery times are expected, so users don’t give a rating, or give it a higher rating.

Note - More analysis is needed to understand what generates higher rating and how the rating impacts the customers’ will to use the app more often, because so many users chose not to rate the service.

Multivariate Analysis - Cont.

Correlation among variables



Observation – There is a negligent correlation between the cost of the order and the time it takes to prepare it or deliver it.

Question 13: The company wants to provide a promotional offer in the advertisement of the restaurants. The condition to get the offer is that the restaurants must have a rating count of more than 50 and the average rating should be greater than 4. Find the restaurants fulfilling the criteria to get the promotional offer.

[125]:

	restaurant_name	rating
0	Shake Shack	133
1	The Meatball Shop	84
2	Blue Ribbon Sushi	73
3	Blue Ribbon Fried Chicken	64
4	RedFarm Broadway	41

This is the list of the restaurants with the highest average ratings:

	restaurant_name	rating
0	The Meatball Shop	4.511905
1	Blue Ribbon Fried Chicken	4.328125
2	Shake Shack	4.278195
3	Blue Ribbon Sushi	4.219178

Observation – The restaurants that get the highest ratings and get the most engagement from customers are all serving the leading 4 cuisines.
That gives another reason why FoodHub should focus on these 4 cuisine types in their offering.

Multivariate Analysis - Cont.

Question 14: The company charges the restaurant 25% on the orders having cost greater than 20 dollars and 15% on the orders having cost greater than 5 dollars. Find the net revenue generated by the company across all orders.

Answer : The net revenue is around 6166.3 dollars

Question 15: The company wants to analyze the total time required to deliver the food. What percentage of orders take more than 60 minutes to get de- livered from the time the order is placed? (The food has to be prepared and then delivered.)

Answer:

The number of total orders that take more than 60 minutes: 200

Percentage of orders above 60 minutes: 10.54 %

Question 16: The company wants to analyze the delivery time of the orders on weekdays and weekends. How does the mean delivery time vary during weekdays and weekends?

Answer:

The mean delivery time on weekdays is around 28 minutes

The mean delivery time on weekdays is around 22 minutes

Observation – Relatively low number of orders take more than an hour to prepare and deliver. This is good, since it means the user experience is better when they don't need to wait too long for their order through the app, and we might want to investigate if these orders should even be offered on the app. (if their revenue doesn't justify keeping them)

APPENDIX

The appendix includes the code that used to generate the plots above

Data Overview

Question 1: How many rows and columns are present in the data?

```
[11]: df.shape
```

```
[11]: (1898, 9)
```

Question 2: What are the datatypes of the different columns in the dataset?

```
[13]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1898 entries, 0 to 1897
Data columns (total 9 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   order_id               1898 non-null   int64
 1   customer_id            1898 non-null   int64
 2   restaurant_name        1898 non-null   object
 3   cuisine_type           1898 non-null   object
 4   cost_of_the_order      1898 non-null   float64
 5   day_of_the_week        1898 non-null   object
 6   rating                 1898 non-null   object
 7   food_preparation_time  1898 non-null   int64
 8   delivery_time          1898 non-null   int64
dtypes: float64(1), int64(4), object(4)
memory usage: 133.6+ KB
```

Question 3:Are there any missing values in the data? If yes, treat them using an appropriate method.

```
[15]: # Checking for missing values in the data
      df.isnull().sum()
```

```
[15]:order_id                0
      customer_id            0
      restaurant_name        0
      cuisine_type           0
      cost_of_the_order      0
      day_of_the_week        0
      rating                 0
      food_preparation_time  0
      delivery_time          0
      dtype: int64
```

Question 4: Check the statistical summary of the data. What is the minimum, average, and maximum time it takes for food to be prepared once an order is placed?

```
[17]: # Get the summary statistics of the numerical data
      df.describe()
```

```
[17]:            order_id    customer_id  cost_of_the_order  food_preparation_time  \
      count  1.898000e+03    1898.000000        1898.000000            1898.000000
      mean   1.477496e+06  171168.478398          16.498851              27.371970
      std    5.480497e+02  113698.139743           7.483812               4.632481
      min    1.476547e+06    1311.000000           4.470000              20.000000
      25%    1.477021e+06   77787.750000          12.080000              23.000000
      50%    1.477496e+06  128600.000000          14.140000              27.000000
      75%    1.477970e+06  270525.000000          22.297500              31.000000
      max    1.478444e+06  405334.000000          35.410000              35.000000

             delivery_time
      count    1898.000000
      mean       24.161749
      std         4.972637
      min        15.000000
      25%        20.000000
      50%        25.000000
      75%        28.000000
      max        33.000000
```

Question 5: How many orders are not rated?

```
[19]: df['rating'].value_counts()
```

```
[19]: rating
      Not given    736
      5            588
      4            386
      3            188
      Name: count, dtype: int64
```

Univariate Analysis

Question 6: Explore all the variables and provide observations on their distribution

Order ID

```
[21]: # check unique order ID
      df['order_id'].nunique()
```

[21]: 1898

Customer ID

```
[23]: # check unique customer ID
      df['customer_id'].nunique()
```

[23]: 1200

Restaurant name

```
[27]: # check unique Restaurant Name
      df['restaurant_name'].nunique()
```

[27]: 178

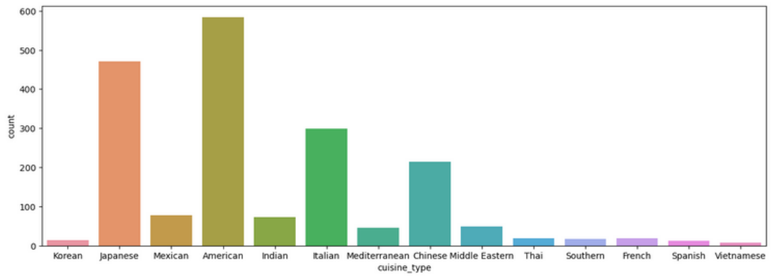
Cuisine type

```
[25]: # Check unique cuisine type
      df['cuisine_type'].nunique()
```

[25]: 14

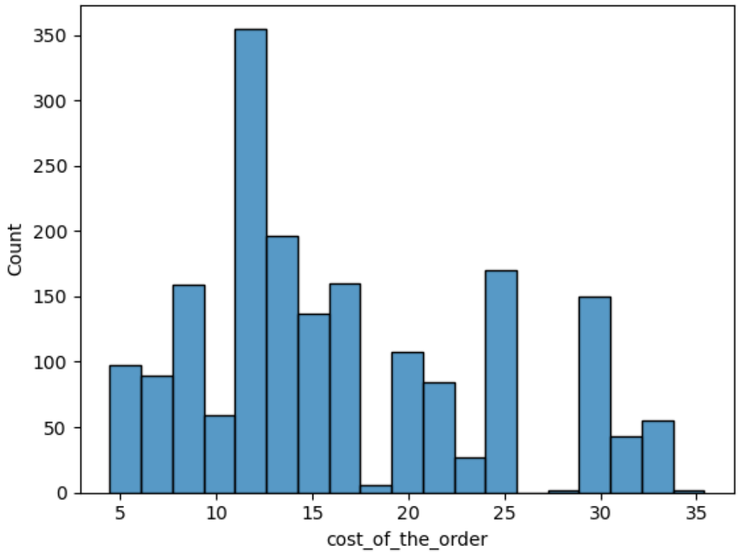
```
[29]: plt.figure(figsize = (15,5))
      sns.countplot(data = df, x = 'cuisine_type')
```

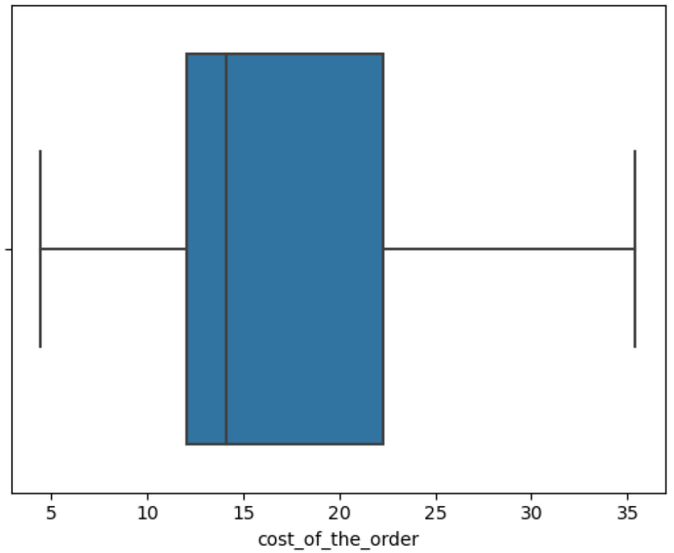
[29]: <Axes: xlabel='cuisine_type', ylabel='count'>



Cost of the order

```
[31]: sns.histplot(data=df,x='cost_of_the_order') ## Histogram for the cost of order
      plt.show()
      sns.boxplot(data=df,x='cost_of_the_order') ## Boxplot for the cost of order
      plt.show()
```





Univariate Analysis - Cont.

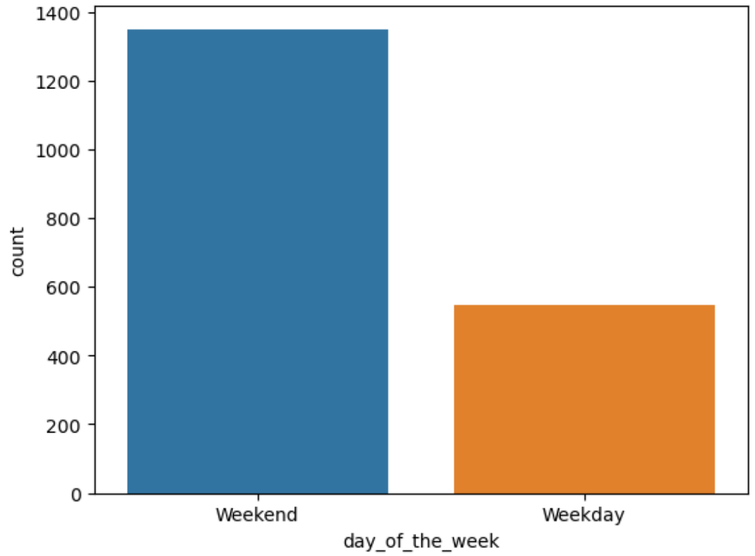
Day of the week

[33]:
```
# # Check the unique values
df['day_of_the_week'].nunique()
```

[33]: 2

[35]:
```
sns.countplot(data = df, x = 'day_of_the_week')
```

[35]: <Axes: xlabel='day_of_the_week', ylabel='count'>



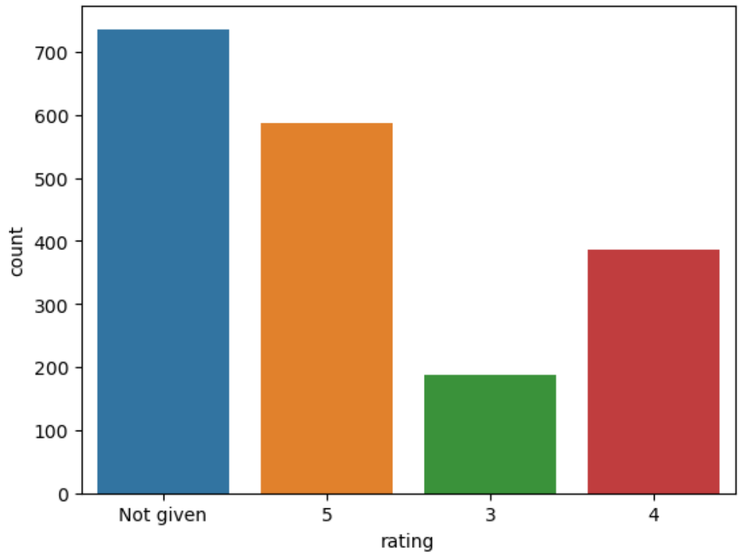
Rating

[37]:
```
# Check the unique values
df['rating'].nunique()
```

[37]: 4

[39]:
```
sns.countplot(data = df, x = 'rating')
```

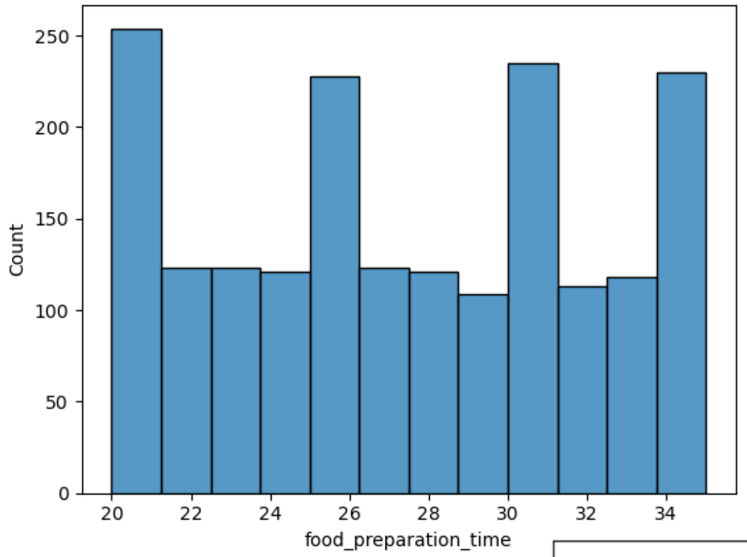
[39]: <Axes: xlabel='rating', ylabel='count'>

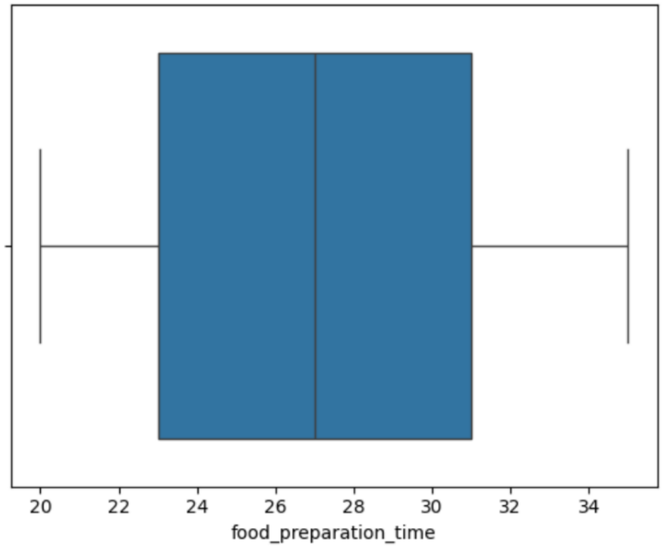


Univariate Analysis - Cont.

Food Preparation time

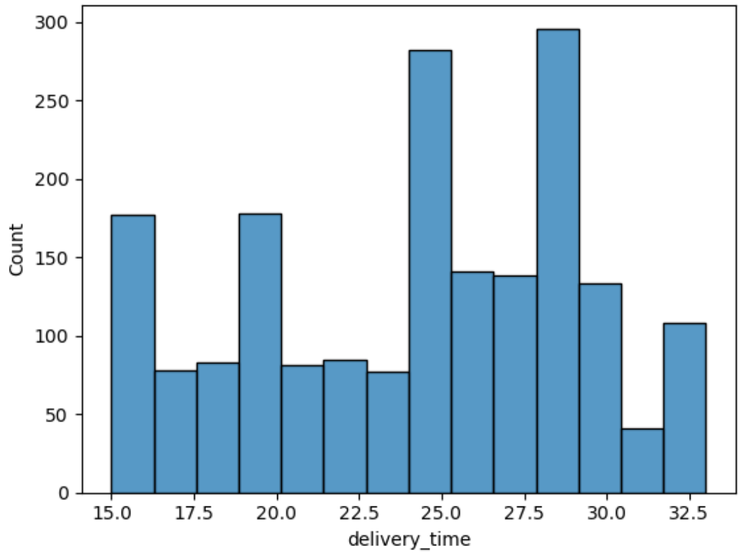
```
[41]: sns.histplot(data=df,x='food_preparation_time')
      plt.show()
      sns.boxplot(data=df,x=food_preparation_time')
      plt.show()
```

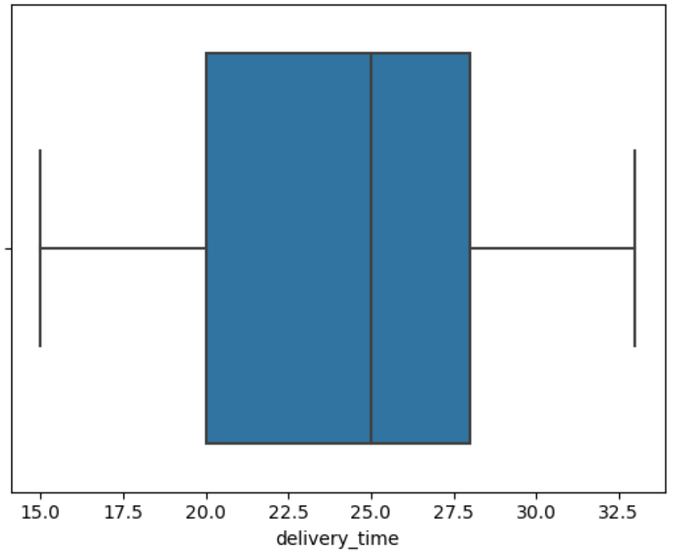




Delivery time

```
[43]: sns.histplot(data=df,x='delivery_time')
      plt.show()
      sns.boxplot(data=df,x='delivery_time')
      plt.show()
```





Univariate Analysis - Cont.

Question7: Which are the top 5 restaurants in terms of the number of orders received?

[93]:
```
# Get top 5 restaurants with highest number of orders
df['restaurant_name'].value_counts().head(5
)
```

[93]: restaurant_name

ShakeShack	219
TheMeatballShop	132
BlueRibbonSushi	119
BlueRibbonFriedChicken	96
Parm	68

Name: count, dtype: int64

Question 8: Which is the most popular cuisine on weekends?

[81]:
```
# Get most popular cuisine on weekends
df_weekend = df[df['day_of_the_week'] == 'Weekend']
df_weekend['cuisine_type'].value_counts()
```

[81]: cuisine_type

American	415
Japanese	335
Italian	207
Chinese	163
Mexican	53
Indian	49
Mediterranean	32
MiddleEastern	32
Thai	15
French	13
Korean	11
Southern	11
Spanish	11
Vietnamese	4

Name: count, dtype: int64

Question 9: What percentage of the orders cost more than 20 dollars?

[83]:
```
# Get orders that cost above 20 dollars
df_greater_than_20 = df[df['cost_of_the_order']>20]

# Calculate the number of total orders where the cost is above 20 dollars
print('The number of total orders that cost above 20 dollars is:',␣
  ↪df_greater_than_20.shape[

  0])
# Calculate percentage of such orders in the dataset
percentage = (df_greater_than_20.shape[0] / df.shape[0]) * 100

print("Percentage of orders above 20 dollars:", round(percentage, 2), '%')
```

The number of total orders that cost above 20 dollars is: 555
Percentage of orders above 20 dollars: 29.24 %

Question 10: What is the mean order delivery time?

[87]:
```
# Get the mean delivery time
 mean_del_time =
 df['delivery_time'].mean()
 print('The mean delivery time for this dataset is', round(mean_del_time, 2),␣
```

The mean delivery time for this dataset is 24.16 minutes

Question 11: The company has decided to give 20% discount vouchers to the top 3 most frequent customers. Find the IDs of these customers and the number of orders they placed.

[95]:
```
# Get the counts of  each customer_id
df['customer_id'].value_counts().head(5)
```

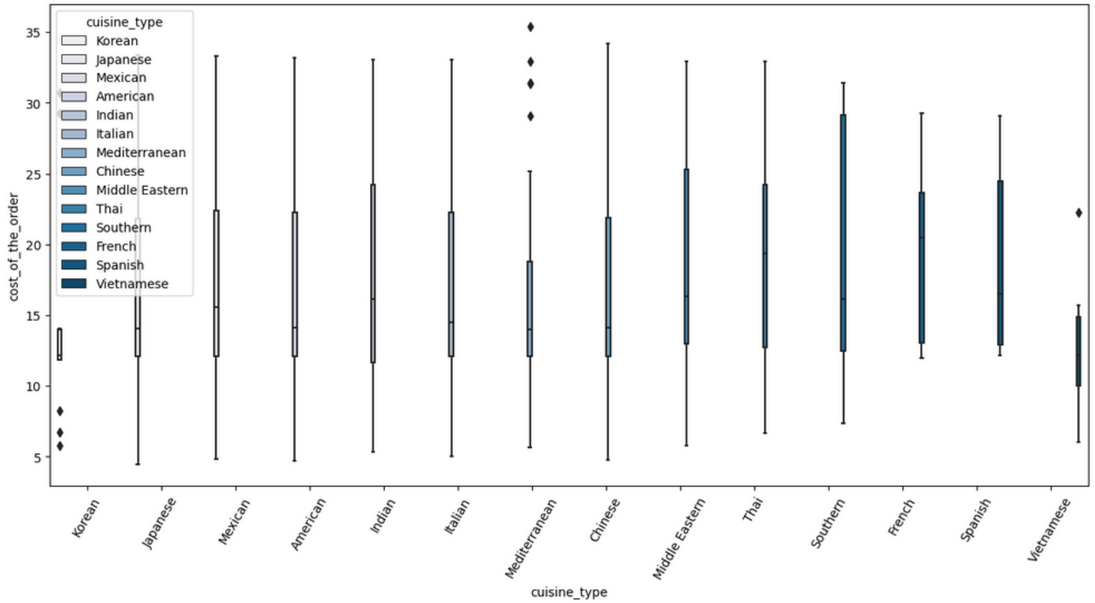
[95]:
```
customer_id
52832     13
47440     10
83287      9
250494     8
259341     7
Name: count, dtype: int64
```

Multivariate Analysis

Question 12: Perform a multivariate analysis to explore relationships between the important variables in the dataset.

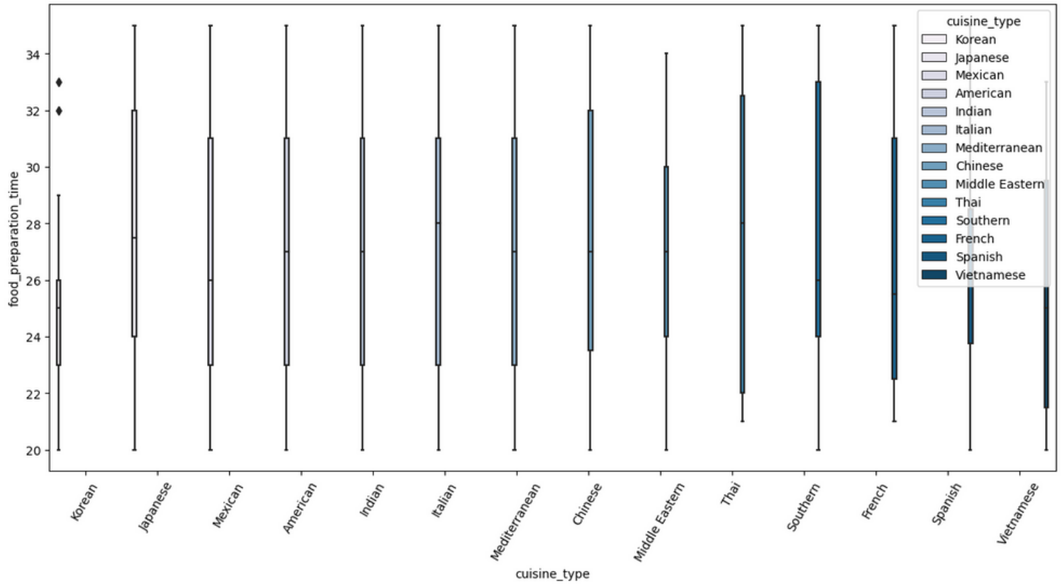
Cuisine vs Cost of the order

[99]:
```
# Relationship between cost of the order and cuisine type
plt.figure(figsize=(15,7)) sns.boxplot(x = "cuisine_type", y = "cost_of_the_order",
data = df, palette =␣
↪'PuBu',hue="cuisine_type")plt.xticks(rotation=60)
plt.show()
```



Cuisine vs Food Preparation time

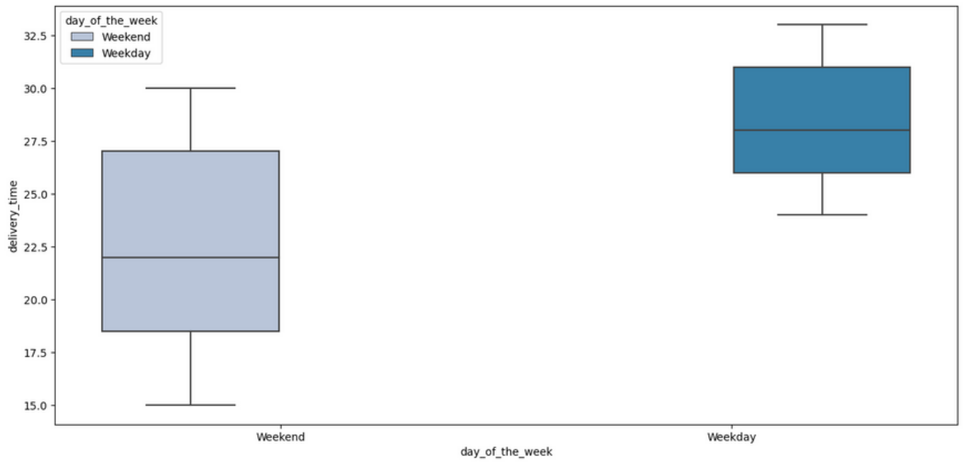
[103]:
```
# Relationship between food preparation time and cuisine type
plt.figure(figsize=(15,7)) sns.boxplot(x = "cuisine_type", y =
"food_preparation_time", data = df, palette␣
↪='PuBu',hue="cuisine_type")plt.xticks(rotation=60)
plt.show()
```



Multivariate Analysis - Cont.

Day of the Week vs Delivery time

[107]:
```
# Relationship between day of the week and delivery time
plt.figure(figsize=(15,7)) sns.boxplot(x = "day_of_the_week", y = "delivery_time", data = df, palette = ␣
↪'PuBu',hue="day_of_the_week")plt.show()
```



Run the below code and write your observations on the revenue generated by the restaurants.

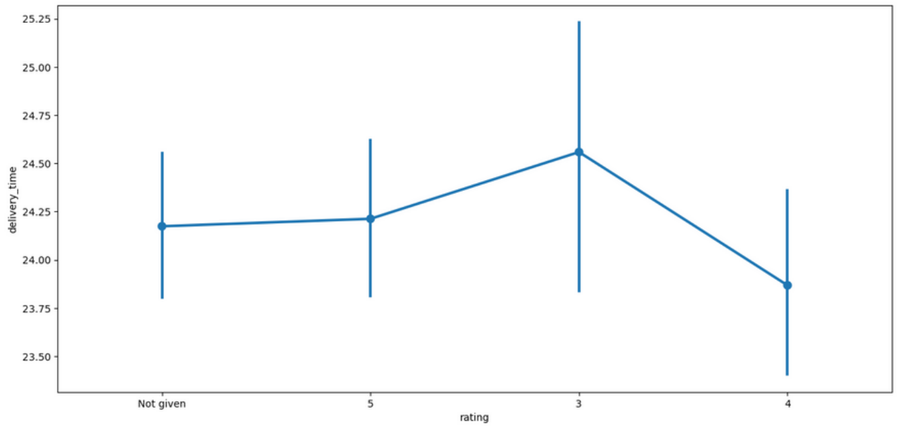
[109]:
```
df.groupby(['restaurant_name'])['cost_of_the_order'].sum().
↪sort_values(ascending = False).head(14)
```

[109]:
```
restaurant_name
Shake Shack                      3579.53
The Meatball Shop                2145.21
Blue Ribbon Sushi                1903.95
Blue Ribbon Fried Chicken        1662.29
Parm                             1112.76
RedFarm Broadway                  965.13
RedFarm Hudson                    921.21
TAO                               834.50
Han Dynasty                       755.29
Blue Ribbon Sushi Bar & Grill     666.62
Rubirosa                          660.45
Sushi of Gari 46                  640.87
Nobu Next Door                    623.67
Five Guys Burgers and Fries       506.47
Name: cost_of_the_order, dtype: float64
```

Multivariate Analysis - Cont.

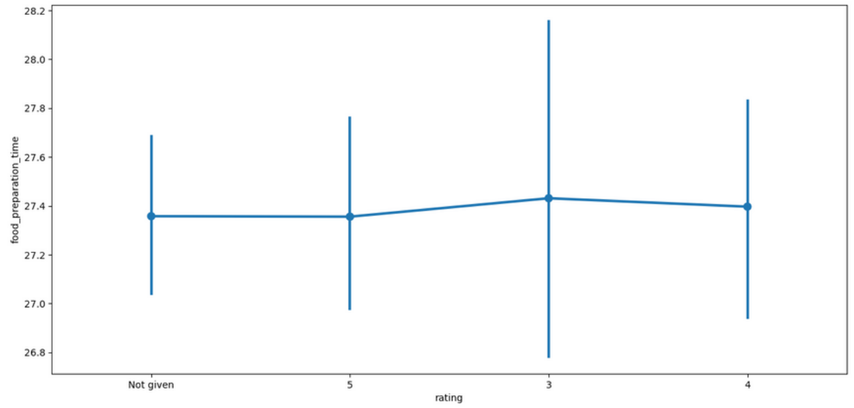
Rating vs Delivery time

[111]:
```
# Relationship between rating and delivery time
plt.figure(figsize=(15, 7)) sns.pointplot(x = 'rating', y =
'delivery_time', data = df) plt.show()
```



Rating vs Food preparation time

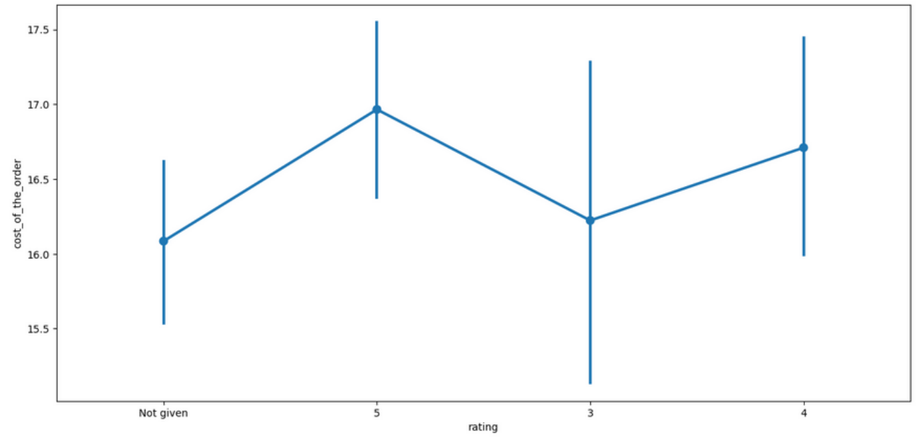
[115]:
```
# Relationship between rating and food preparation
time
plt.figure(figsize=(15, 7)) sns.pointplot(x = 'rating', y =
'food_preparation_time', data = df) plt.show()
```



Rating vs Cost of the order

[121]:
```
# Relationship between rating and cost of the order
plt.figure(figsize=(15, 7))
```

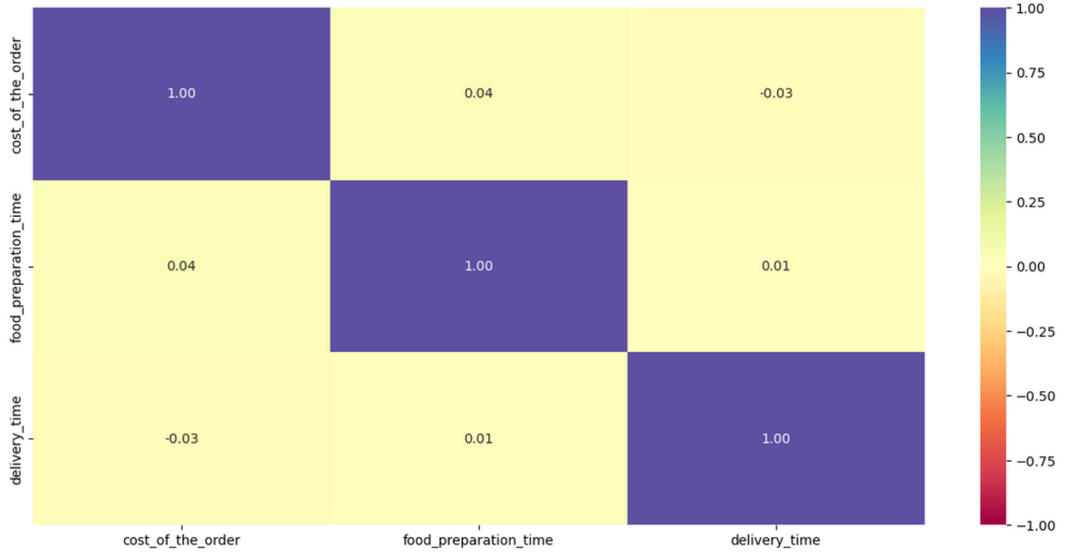
```
sns.pointplot(x = 'rating', y = 'cost_of_the_order', data = df)
plt.show()
```

Multivariate Analysis - Cont.

Correlation among variables

[123]:
```
# Plot the heatmap
col_list = ['cost_of_the_order', 'food_preparation_time', 'delivery_time']
plt.figure(figsize=(15, 7)) sns.heatmap(df[col_list].corr(), annot=True,
vmin=-1, vmax=1, fmt=".2f",␣
↪cmap="Spectral")plt.show()
```



Question 13: The company wants to provide a promotional offer in the advertisement of the restaurants. The condition to get the offer is that the restaurants must have a rating count of more than 50 and the average rating should be greater than 4. Find the restaurants fulfilling the criteria to get the promotional offer.

[125]:
```
# Filter the rated restaurants
df_rated = df[df['rating'] != 'Not given'].copy()

# Convert rating column from object to integer
df_rated['rating'] = df_rated['rating'].astype('int')

# Create a dataframe that contains the restaurant names with their rating

counts
df_rating_count = df_rated.groupby(['restaurant_name'])['rating'].count().

↪sort_values(ascending=False).reset_index()df_rating_count.head()
```

[125]:

	restaurant_name	rating
0	Shake Shack	133
1	The Meatball Shop	84
2	Blue Ribbon Sushi	73
3	Blue Ribbon Fried Chicken	64
4	RedFarm Broadway	41

[153]:
```
# Get the restaurant names that have rating count more than 50
rest_names = df_rating_count[df_rating_count['rating'] > 50].copy()
rest_list = rest_names['restaurant_name'].tolist()

# Filter to get the data of restaurants that have rating count more than 50
df_mean_4 = df_rated[df_rated['restaurant_name'].isin(rest_list)].copy()

# Group the restaurant names with their ratings and find the mean rating of␣
↪each restaurant
df_mean_4.groupby(['restaurant_name'])['rating'].mean().sort_values(ascending =␣
↪False).reset_index().dropna()
```

[153]:

	restaurant_name	rating
0	The Meatball Shop	4.511905
1	Blue Ribbon Fried Chicken	4.328125
2	Shake Shack	4.278195
3	Blue Ribbon Sushi	4.219178

Multivariate Analysis - Cont.

Question 14: The company charges the restaurant 25% on the orders having cost greater than 20 dollars and 15% on the orders having cost greater than 5 dollars. Find the net revenue generated by the company across all orders.

[155]:
```
#function to determine the revenue
def compute_rev(x):
    if x > 20:
        return x*0.25
    elif x > 5:
        return x*0.15
    else:
        return x*0

df['Revenue'] = df['cost_of_the_order'].apply(compute_rev)
df.head()
```

[155]:

	order_id	customer_id	restaurant_name	cuisine_type	cost_of_the_order	day_of_the_week	rating	food_preparation_time	delivery_time	Revenue
0	1477147	337525	Hangawi	Korean	30.75	Weekend	Not given	25	20	7.6875
1	1477685	358141	Blue Ribbon Sushi Izakaya	Japanese	12.08	Weekend	Not given	25	23	1.8120
2	1477070	66393	Cafe Habana	Mexican	12.23	Weekday	5	23	28	1.8345
3	1477334	106968	Blue Ribbon Fried Chicken	American	29.20	Weekend	3	25	15	7.3000
4	1478249	76942	Dirty Bird to Go	American	11.59	Weekday	4	25	24	1.7385

[157]:
```
# get the total revenue and print it
total_rev = df['Revenue'].sum()
print('The net revenue is around', round(total_rev, 2), 'dollars')
```

The net revenue is around 6166.3 dollars

Question 15: The company wants to analyze the total time required to deliver the food. What percentage of orders take more than 60 minutes to get de- livered from the time the order is placed? (The food has to be prepared and then delivered.)

[161]:
```
# Calculate total delivery time and add a new column to the dataframe df to store the total delivery time
df['total_time'] = df['food_preparation_time'] + df['delivery_time']


## Write the code below to find the percentage of orders that have more than 60 minutes of total delivery time (see Question 9 for reference)
# Get orders that cost above 20 dollars
df_greater_than_60 = df[df['total_time']>60]

# Calculate the number of total orders where the total time is more than 60 minutes
print('The number of total orders that take more than 60 minutes:', df_greater_than_60.shape[0])

# Calculate percentage of such orders in the dataset
percentage = (df_greater_than_60.shape[0] / df.shape[0]) * 100

print("Percentage of orders above 60 minutes:", round(percentage, 2), '%')
```

The number of total orders that take more than 60 minutes: 200
Percentage of orders above 60 minutes: 10.54 %

Question 16: The company wants to analyze the delivery time of the orders on weekdays and weekends. How does the mean delivery time vary during weekdays and weekends?

[163]:
```
# Get the mean delivery time on weekdays and print it
print('The mean delivery time on weekdays is around',
      round(df[df['day_of_the_week'] == 'Weekday']['delivery_time'].mean()),
      'minutes')


## Write the code below to get the mean delivery time on weekends and print it
print('The mean delivery time on weekends is around',
      round(df[df['day_of_the_week'] == 'Weekend']['delivery_time'].mean()),
      'minutes')
```

The mean delivery time on weekdays is around 28 minutes
The mean delivery time on weekdays is around 22 minutes

Great Learning

Happy Learning !