Project Python Foundations: FoodHub Data Analysis

Marks: 50

Context

The number of restaurants in New York is increasing day by day. Lots of students and busy professionals rely on those restaurants due to their hectic lifestyles. Online food delivery service is a great option for them. It provides them with good food from their favorite restaurants. A food aggregator company FoodHub offers access to multiple restaurants through a single smartphone app.

The app allows the restaurants to receive a direct online order from a customer. The app assigns a delivery person from the company to pick up the order after it is confirmed by the restaurant. The delivery person then uses the map to reach the restaurant and waits for the food package. Once the food package is handed over to the delivery person, he/she confirms the pick-up in the app and travels to the customer's location to deliver the food. The delivery person confirms the drop-off in the app after delivering the food package to the customer. The customer can rate the order in the app. The food aggregator earns money by collecting a fixed margin of the delivery order from the restaurants.

Objective

The food aggregator company has stored the data of the different orders made by the registered customers in their online portal. They want to analyze the data to get a fair idea about the demand of different restaurants which will help them in enhancing their customer experience. Suppose you are hired as a Data Scientist in this company and the Data Science team has shared some of the key questions that need to be answered. Perform the data analysis to find answers to these questions that will help the company to improve the business.

Data Description

The data contains the different data related to a food order. The detailed data dictionary is given below.

Data Dictionary

- order_id: Unique ID of the order
- customer_id: ID of the customer who ordered the food
- restaurant_name: Name of the restaurant
- cuisine_type: Cuisine ordered by the customer
- cost: Cost of the order
- day_of_the_week: Indicates whether the order is placed on a weekday or weekend (The weekday is from Monday to Friday and the weekend is Saturday and Sunday)
- rating: Rating given by the customer out of 5
- food_preparation_time: Time (in minutes) taken by the restaurant to prepare the food. This is calculated by taking the difference between the timestamps of the restaurant's order confirmation and the delivery person's pick-up confirmation.
- delivery_time: Time (in minutes) taken by the delivery person to deliver the food package. This is calculated by taking the difference between the timestamps of the delivery person's pick-up confirmation and drop-off information

Let us start by importing the required libraries

In [6]:
```
# import libraries for data manipulation
import numpy as np
import pandas as pd

# import libraries for data visualization
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
```

Understanding the structure of the data

In [13]:
```
# read the data
df = pd.read_csv('foodhub_order.csv')
# returns the first 5 rows
df.head()
```

Out[13]:

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

Observations:

The DataFrame has 9 columns as mentioned in the Data Dictionary. Data in each row corresponds to the order placed by a customer. There are no null values in the dataset.

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

In [11]:
```
# Checking structure and dimension of the data

df.shape
```

Out[11]: (1898, 9)

Observations:

There are 1898 rows and 9 columns in the data.

Question 2: What are the datatypes of the different columns in the dataset? (The info() function can be used) [1 mark]

In [14]:
```
# Checking summary of the DataFrame
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
```

Observations:

The different datatypes in the dataset are float, integer and objects. Datatype of the columns order_id, customer_id, food_preparation_time, delivery_time - integer Datatype of the column cost_of_the_order - float Datatype of the columns restaurant_name, cuisine_type, day_of_the_week, rating - object

Question 3: 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? [2 marks]

In [18]:
```
# Descriptive statistics summary

df.describe()
```

Out[18]:

	order_id	customer_id	cost_of_the_order	food_preparation_time	delivery_time
count	1.898000e+03	1898.000000	1898.000000	1898.000000	1898.000000
mean	1.477496e+06	171168.478398	16.498851	27.371970	24.161749
std	5.480497e+02	113698.139743	7.483812	4.632481	4.972637
min	1.476547e+06	1311.000000	4.470000	20.000000	15.000000
25%	1.477021e+06	77787.750000	12.080000	23.000000	20.000000
50%	1.477496e+06	128600.000000	14.140000	27.000000	25.000000
75%	1.477970e+06	270525.000000	22.297500	31.000000	28.000000
max	1.478444e+06	405334.000000	35.410000	35.000000	33.000000

Observations:

It takes minimum time of 20 minutes and maximum of 35 minutes for the food to be prepared once the order is placed. On an average, it takes 27.37 minutes for food preparation.

Question 4: How many orders are not rated? [1 mark]

In [34]:
```
# counts not rated orders
df['rating'].value_counts()['Not given']
```

Out[34]: 736

Observations:

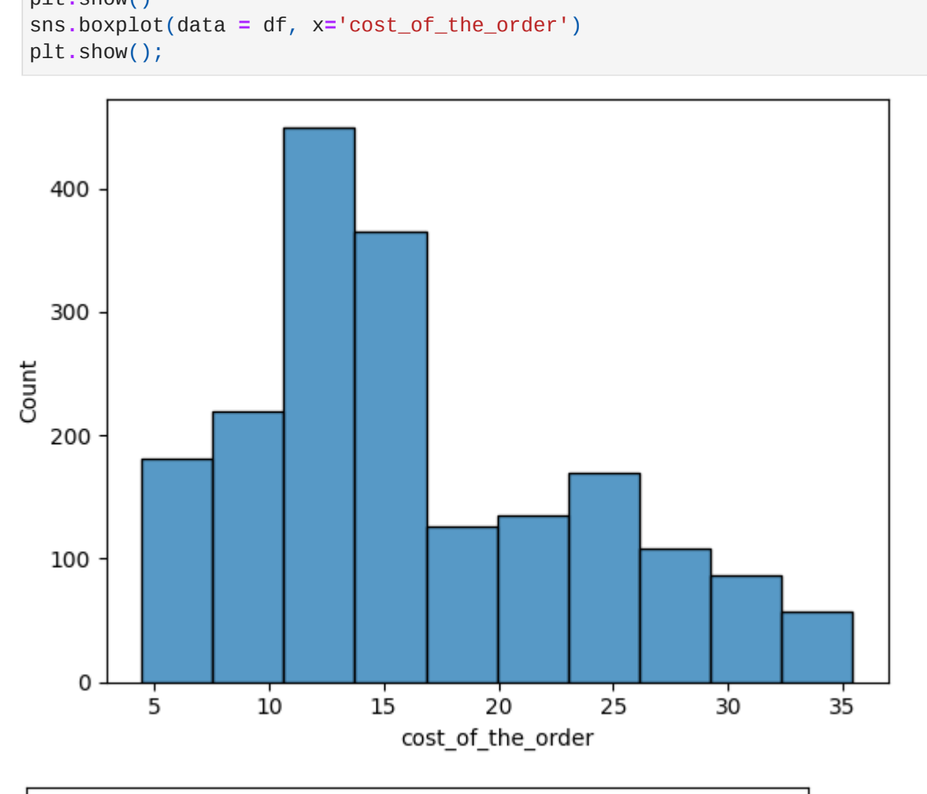
There are 736 orders where customer has provided any rating.

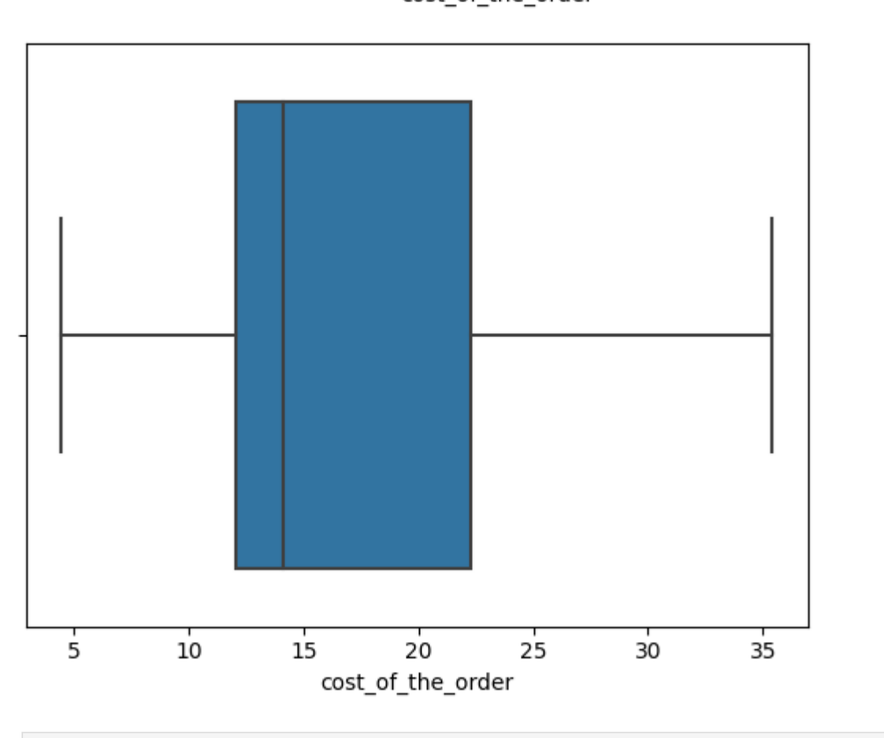
Exploratory Data Analysis (EDA)

Univariate Analysis

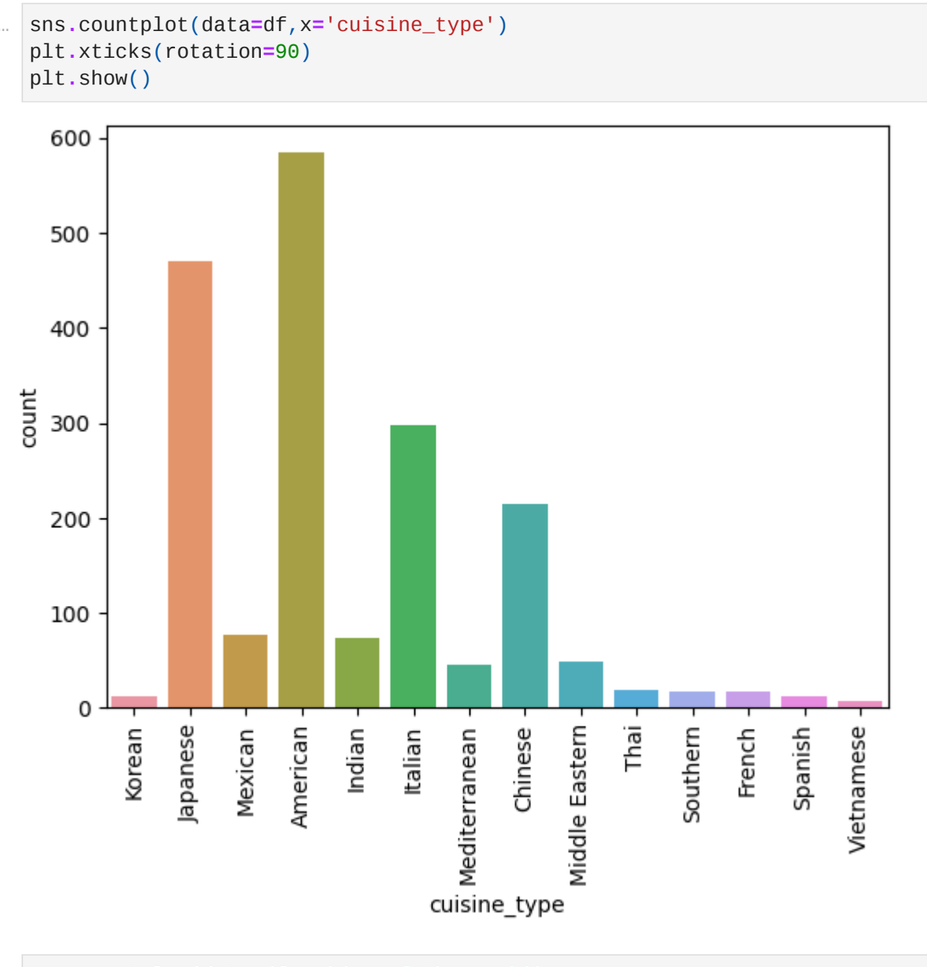
Question 5: Explore all the variables and provide observations on their distributions. (Generally, distinct number of values, histograms, boxplots, countplots, etc. are used for univariate exploration.) [9 marks]

In [115…
```
sns.histplot(data = df, x='cost_of_the_order',bins = 10, stat = 'count')
plt.show()
sns.boxplot(data = df, x='cost_of_the_order')
plt.show();
```

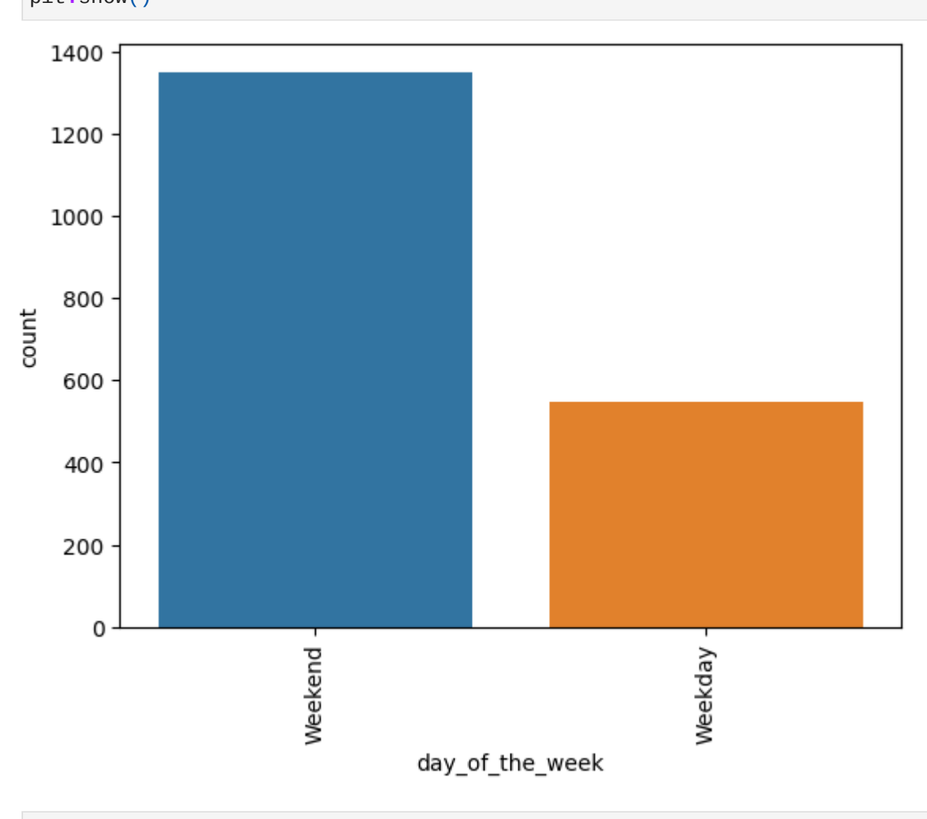




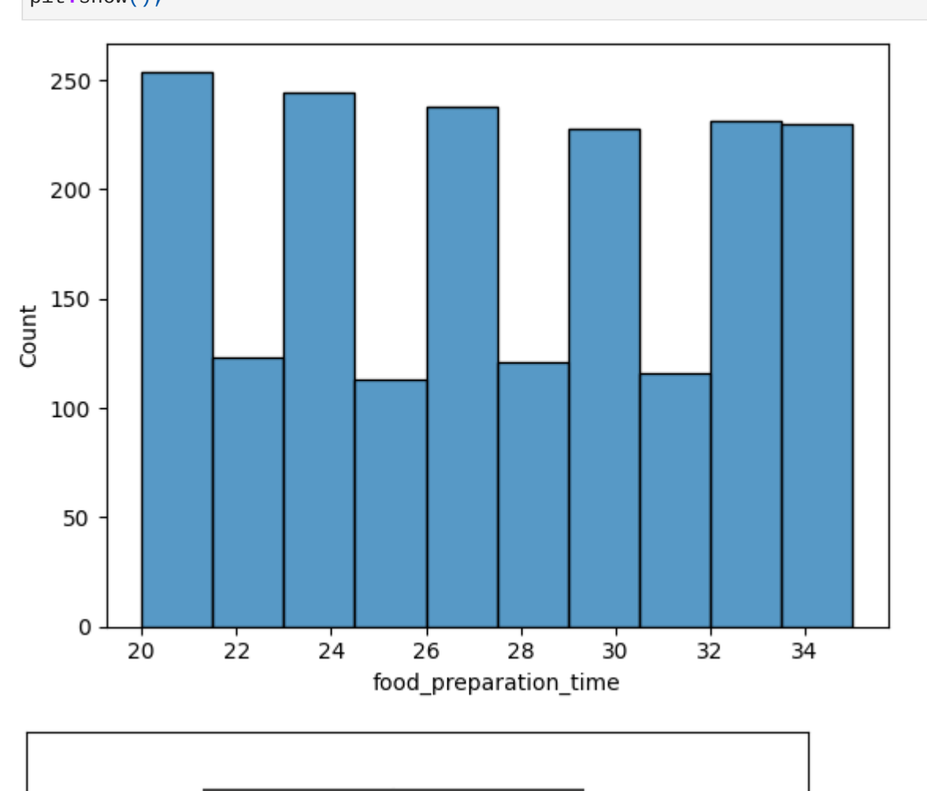
In [116…
```
sns.countplot(data=df,x='cuisine_type')
plt.xticks(rotation=90)
plt.show()
```

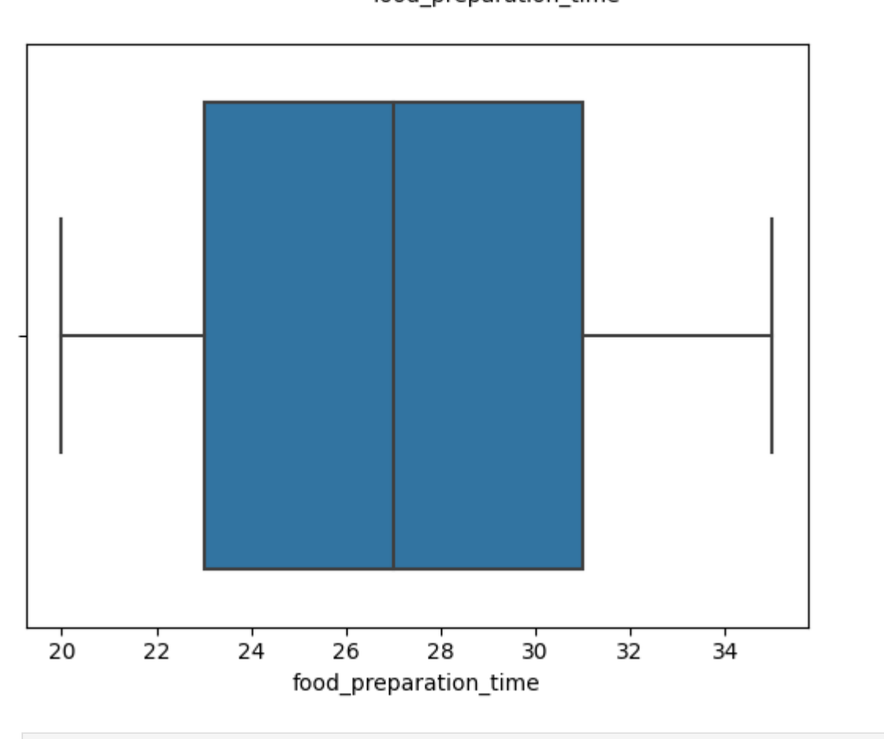


In [122…
```
sns.countplot(data=df,x='day_of_the_week')
plt.xticks(rotation=90)
plt.show()
```

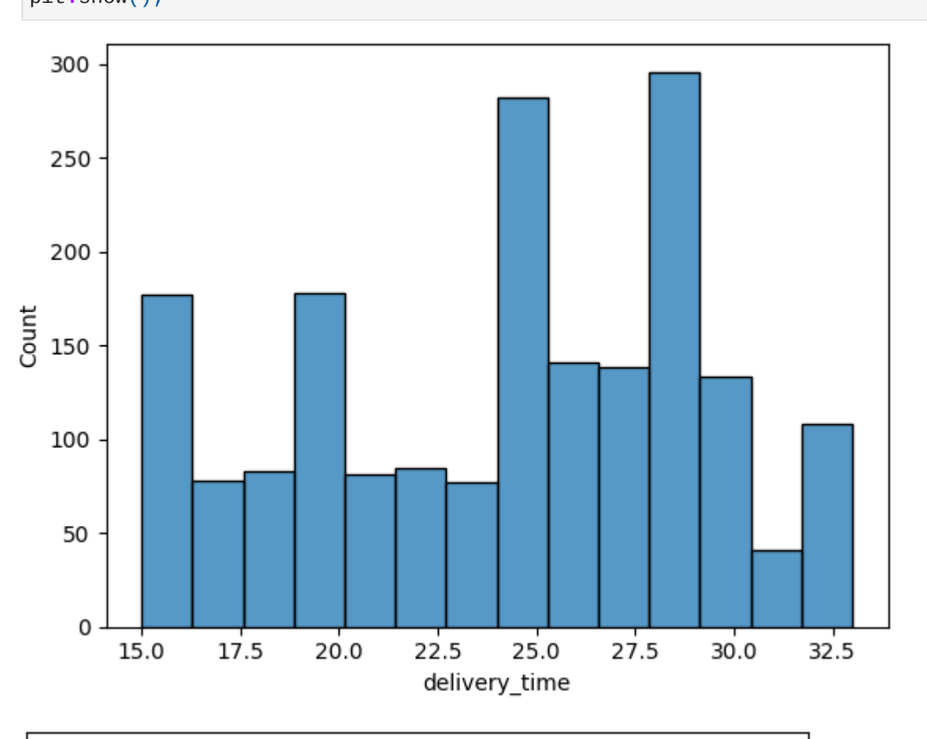


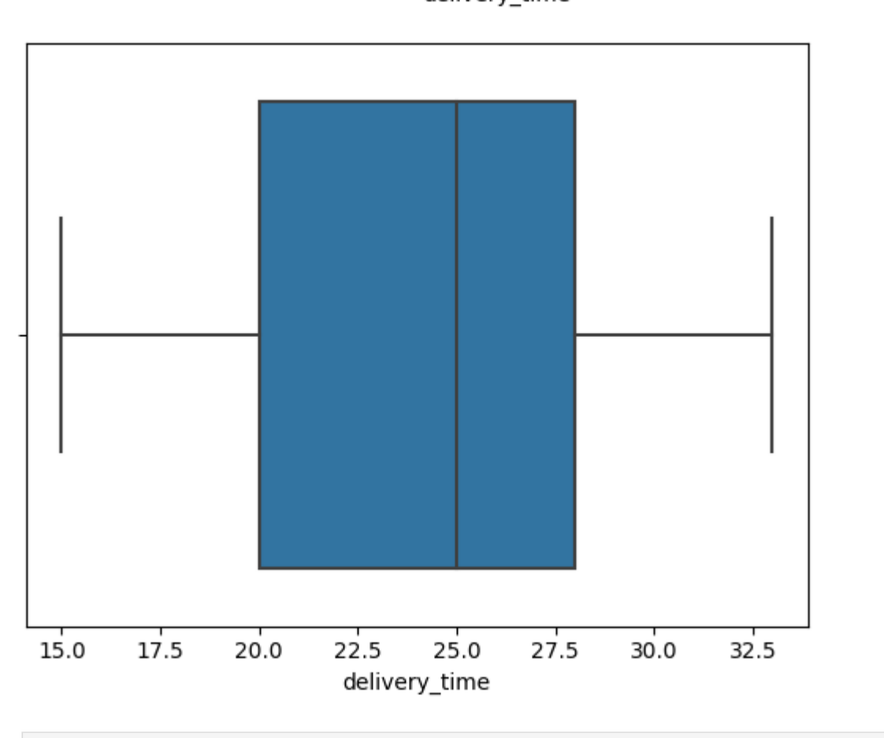
In [124…
```
sns.histplot(data = df, x='food_preparation_time',bins = 10, stat = 'count')
plt.show()
sns.boxplot(data = df, x='food_preparation_time')
plt.show();
```





In [126…
```
sns.histplot(data = df, x='delivery_time', stat = 'count')
plt.show()
sns.boxplot(data = df, x='delivery_time')
plt.show();
```





In [ ]:

Question 6: Which are the top 5 restaurants in terms of the number of orders received? [1 mark]

In [37]:
```
# Gives the list of top 5 restaurants
df['restaurant_name'].value_counts().head(5)
```

Out[37]:
```
restaurant_name
Shake Shack                  219
The Meatball Shop            132
Blue Ribbon Sushi            119
Blue Ribbon Fried Chicken     96
Parm                          68
Name: count, dtype: int64
```

Observations:

The top 5 restaurants in terms of the number of orders received are
Shake Shack The Meatball Shop Blue Ribbon Sushi Blue Ribbon Fried Chicken Parm

Question 7: Which is the most popular cuisine on weekends? [1 mark]

In [49]:
```
# Filtering the data by Weekend and grouping the data by cuisine type and couting the numbers.
#Then sorting the data in descending order.
#head() is used to print only top 1 value.

df[df['day_of_the_week'] =='Weekend'].groupby('cuisine_type').count().sort_values(by = 'order_id', ascending=False).head(1)
```

Out[49]:

	order_id	customer_id	restaurant_name	cost_of_the_order	day_of_the_week	rating	food_preparation_time	delivery_time
cuisine_type								
American	415	415	415	415	415	415	415	415

Observations:

The most popular(ordered) cuisine type on weekends is "American"

Question 8: What percentage of the orders cost more than 20 dollars?(use .round function to round the final percentage) [2 marks]

In [60]:
```
# Counts total orders
total = df['order_id'].count()
# counts the orders that cost above $20
above20orders = df['order_id'][df['cost_of_the_order']> 20].count()
# Calc the percent of the orders more than $20
percentage_calc = (above20orders/total)*100
print(round(percentage_calc, 2), '%') # Rounded to two decimal places
```

```
29.24 %
```

Observations:

29.24% of the total orders cost more than $20

Question 9: What is the mean order delivery time? [1 mark]

In [66]:
```
# Aggregated by mean of the column delivery time

mean_delivery_time = df[['delivery_time']].agg('mean') # Rounded to two decimal places
print(round(mean_delivery_time, 2)) # Rounded to two decimal places
```

```
delivery_time    24.16
dtype: float64
```

Observations:

The average(mean) time taken to deliver an order is 24.16 minutes.

Question 10: 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. [1 mark]

In [69]:
```
# Grouping the data by customer ID and count of order IDs. Then sorting the data by count of order IDs in descending order

df[['order_id','customer_id']].groupby('customer_id').count().sort_values(by='order_id',ascending = False).head(3)
```

Out[69]:

	order_id
customer_id	
52832	13
47440	10
83287	9

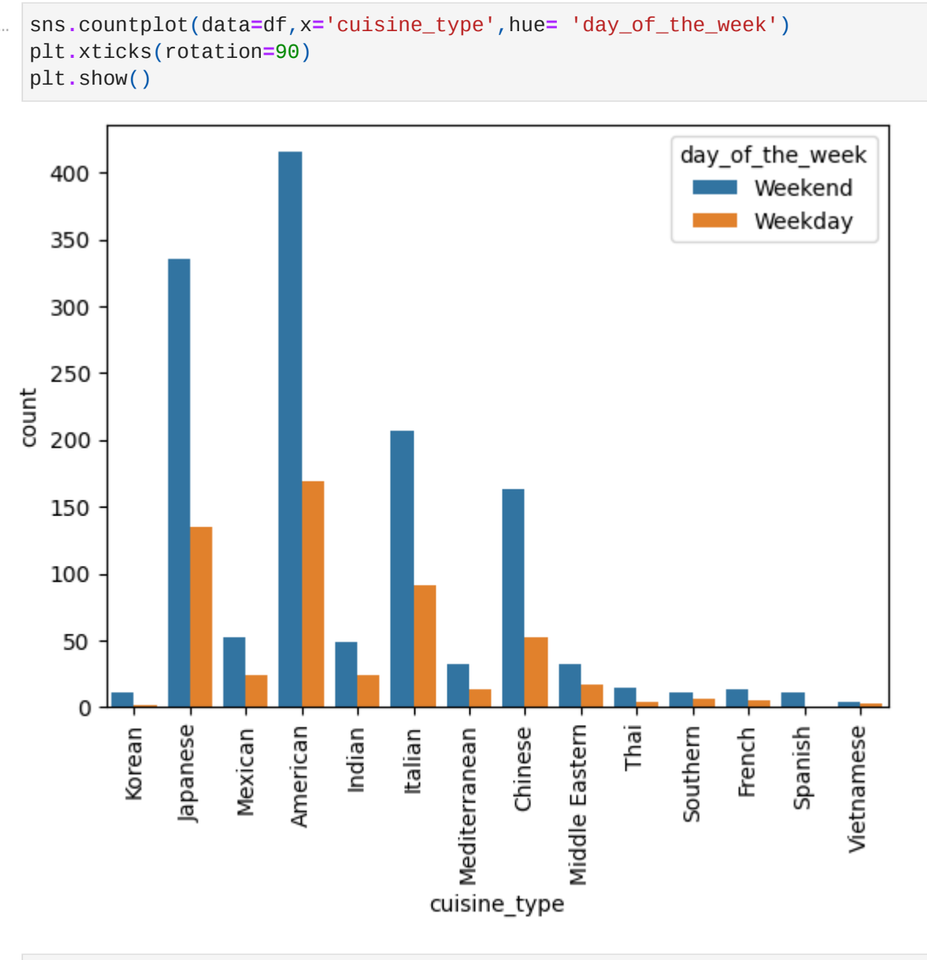
Observations:

The customer IDs of top 3 customers with most frequent orders are 52832 47440 83287

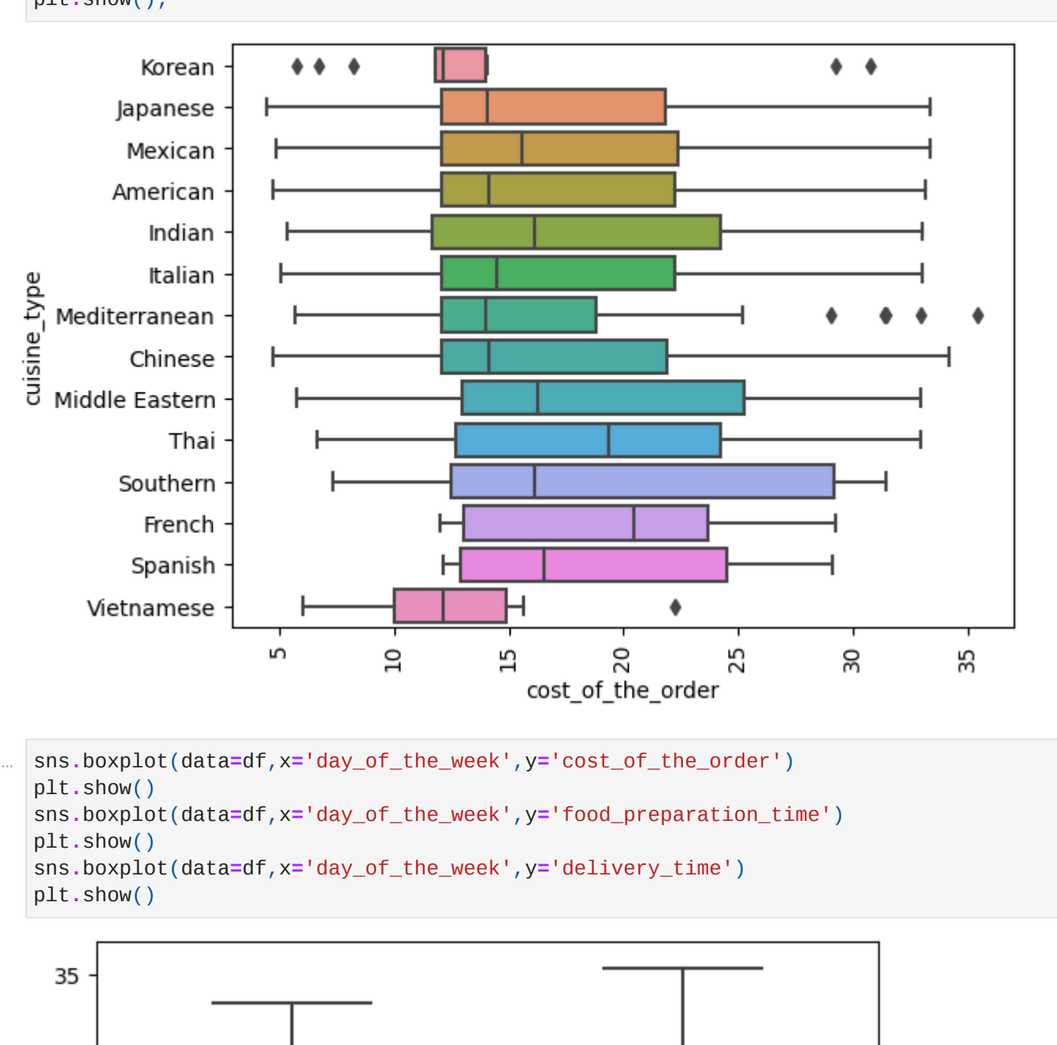
Multivariate Analysis

Question 11: Perform a multivariate analysis to explore relationships between the important variables in the dataset. (It is a good idea to explore relations between numerical variables as well as relations between numerical and categorical variables) [8 marks]

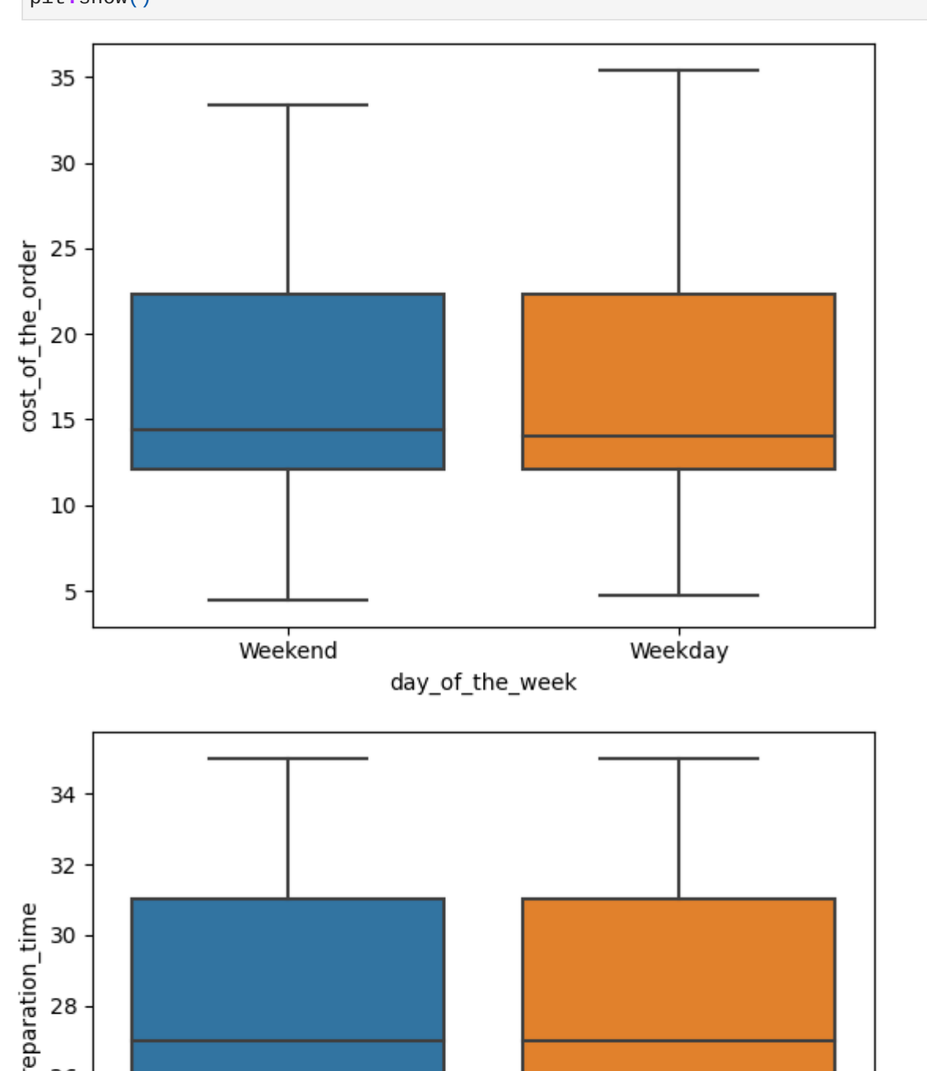
In [132…
```
sns.countplot(data=df,x='cuisine_type',hue= 'day_of_the_week')
plt.xticks(rotation=90)
plt.show()
```

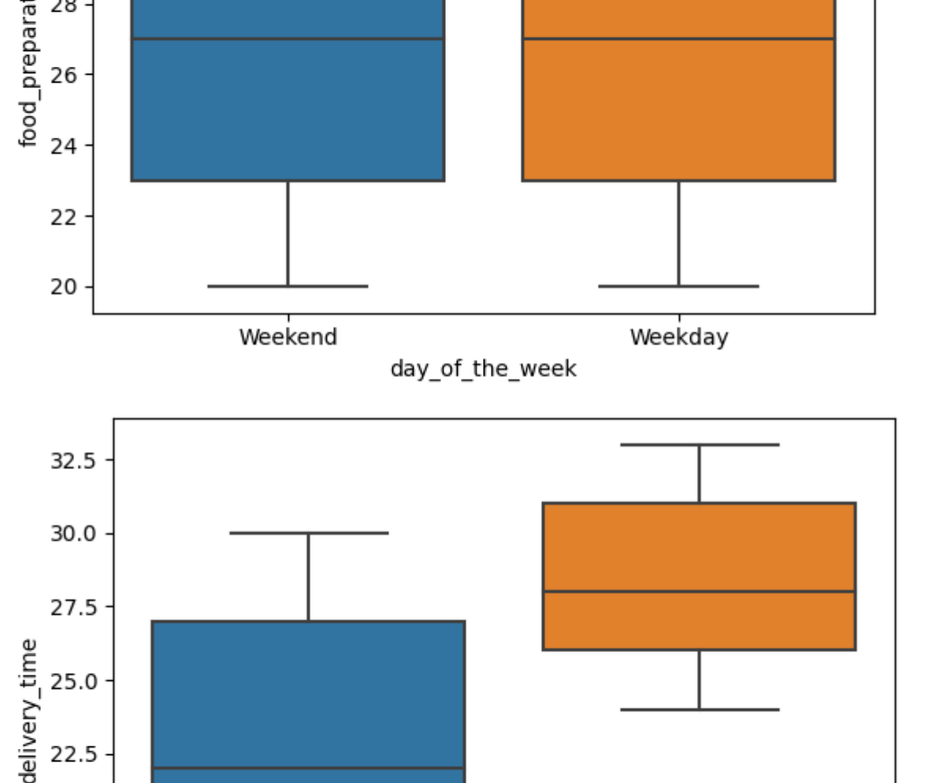


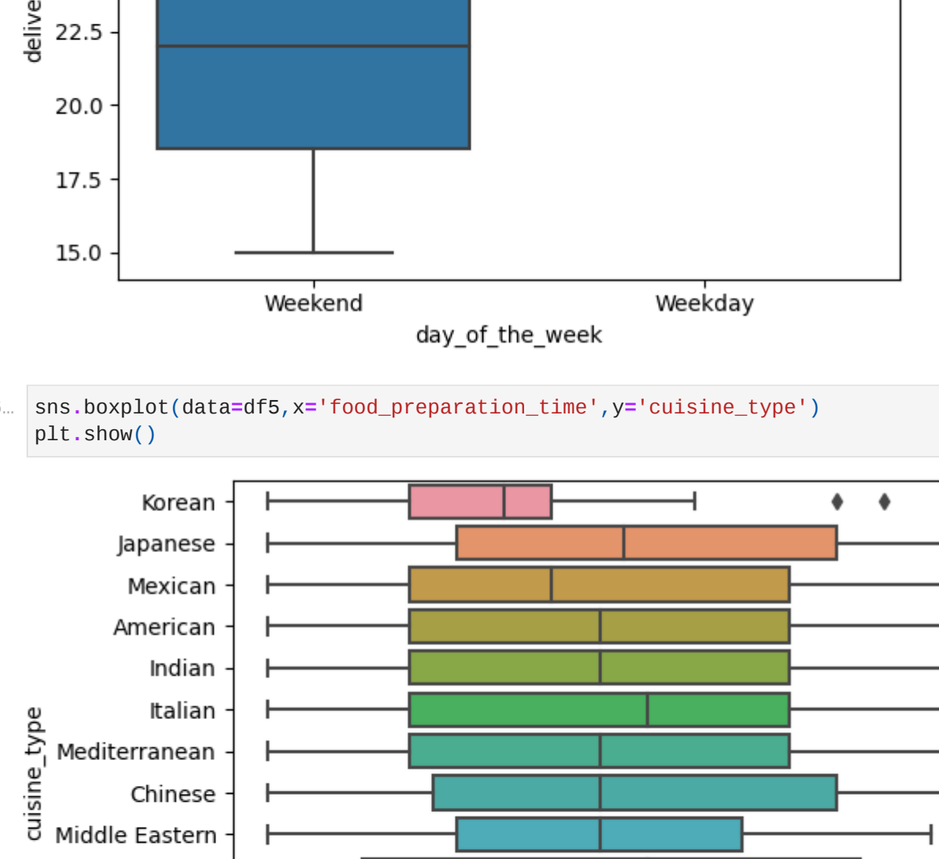
In [133…
```
sns.boxplot(data=df,x='cost_of_the_order',y='cuisine_type')
plt.xticks(rotation=90)
plt.show();
```



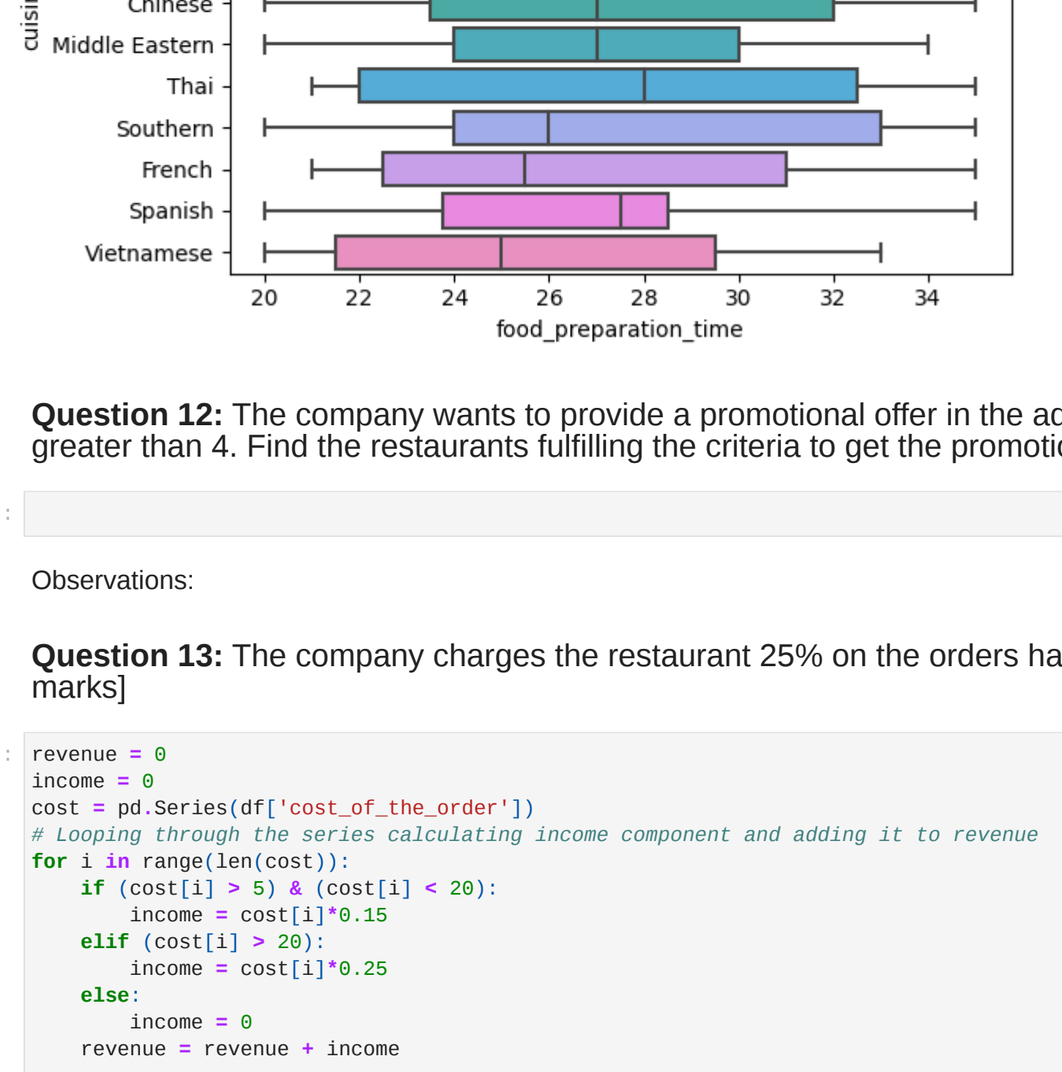
In [134…
```
sns.boxplot(data=df,x='day_of_the_week',y='cost_of_the_order')
plt.show()
sns.boxplot(data=df,x='day_of_the_week',y='food_preparation_time')
plt.show()
sns.boxplot(data=df,x='day_of_the_week',y='delivery_time')
plt.show()
```







In [136…
```
sns.boxplot(data=df5,x='food_preparation_time',y='cuisine_type')
plt.show()
```



Question 12: 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. [3 marks]

In [ ]:

Observations:

Question 13: 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. [2 marks]

In [78]:
```
revenue = 0
income = 0
cost = pd.Series(df['cost_of_the_order'])
# Looping through the series calculating income component and adding it to revenue
for i in range(len(cost)):
    if (cost[i] > 5) & (cost[i] < 20):
        income = cost[i]*0.15
    elif (cost[i] > 20):
        income = cost[i]*0.25
    else:
        income = 0
    revenue = revenue + income

print('Net revenue generated is: ',round(revenue,2))
```

```
Net revenue generated is:  6166.3
```

Observations:

The net revenue generated by the FoodHub company is $6166.3

Question 14: The company wants to analyze the total time required to deliver the food. What percentage of orders take more than 60 minutes to get delivered from the time the order is placed? (The food has to be prepared and then delivered.) (Use .round function to round value to nearest zero) [2 marks]

In [105…
```
df['total_prep_time'] = df['food_preparation_time']+df['delivery_time']
total = df['total_prep_time'].count() #calc total time
ordertime = df['total_prep_time'][df['total_prep_time']> 60].count() #filters the data with total prep time more than 60 mins

percent_calc = (ordertime/total)*100 #calc percentage of total
print(round(percent_calc, 0), '%')
```

```
11.0 %
```

Observations:

Almost 11% of the orders take more than 60 minutes to get delivered.

Question 15: 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? [2 marks]

In [108…
```
mean_del_time = df.groupby('day_of_the_week')[['delivery_time']].agg(['mean','sum','count','std'])
print(round(mean_del_time, 2))
```

```
                delivery_time
                         mean    sum count   std
day_of_the_week
Weekday                 28.34  15502   547  2.89
Weekend                 22.47  30357  1351  4.63
```

Observations:

The mean delivery time on weekdays is 28.34 minute and on weekends is 22.47 minutes. It is noticed that it takes less time to deliver orders on weekends.

Conclusion and Recommendations

Question 16: What are your conclusions from the analysis? What recommendations would you like to share to help improve the business? (You can use cuisine type and feedback ratings to drive your business recommendations.) [4 marks]

Conclusions:

Below is a summary of Key observations and conclusions

1. Ratings provide a great source of analysing but alot of data about (38%) is lost in the unrated orders.
2. On average it takes 24.16 mins to deliver an order.
3. The demand is higher over the weekends.
4. Preparation time is relatively consistent as compared to delivery time, meaning delivery time is the significant variable in total preparation time (total time between order and delivery to customer).
5. There is a tie of 4 customers on the the 5th position given the reward criteria proposed. It requires further analysis or decision on how to handle such.

Recommendations:

Based on the observations in this analysis below is recommended.

1. Improve the customers' response rating their orders, that is, reduce the rating 'Not given' on the orders.
2. Reduce the deliver time further on weekdays.
3. Better marketing can be focused on the weekday to improve sales.
4. Improve ordering rate during weekends.
5. Provide more promotional offers to improve orders.