Project Foundations for Data Science: FoodHub Data Analysis

Marks: 60

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

```
# 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
```

Understanding the structure of the data

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

	order_id	customer_id	restaurant_name	cuisine_type	cost_of_the_order	day_of_the_week	rating	food_preparation_time	del
0	1477147	337525	Hangawi	Korean	30.75	Weekend	Not given	25	
1	1477685	358141	Blue Ribbon Sushi Izakaya	Japanese	12.08	Weekend	Not given	25	

Observations: The DataFrame has 9 columns as mentioned in the Data Dictionary. Data in each row corresponds to the order placed by a customer.

Question 1: How many rows and columns are present in the data? [0.5 mark]

```
# Write your code here
print('Rows =',df.shape[0])
print('Columns =',df.shape[1])
```

```
Rows = 1898
Columns = 9
```

Observations:

1. The dataset contains 1898 rows and 9 columns.
2. Overall, the rows and columns hold the information regarding the orders placed on the foodhub website.

Question 2: What are the datatypes of the different columns in the dataset? (The info() function can be used) [0.5 mark]

```
# Use info() to print a concise 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:

1. There is 2 float type, 5 object type (rating colums is not object), and 2 integer types.
2. No null values.
3. The rating column should be integer type instead of object type.
4. Categorical Variables: order id, customer id, resturant name, cuisine type, rating, day of the week.
5. Numerical Variables: order cost, food prep time, delivery time.

```
#Fixing 'rating' column
df['rating'].unique()
```

```
array(['Not given', '5', '3', '4'], dtype=object)
```

```
df['rating'] = df['rating'].replace(['Not given'],0)

df = df.astype({"rating": float, "order_id": object, "customer_id": object})

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   object
 1   customer_id            1898 non-null   object
 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   float64
 7   food_preparation_time  1898 non-null   int64
 8   delivery_time          1898 non-null   int64
dtypes: float64(2), int64(2), object(5)
memory usage: 133.6+ KB
```

Question 3: Are there any missing values in the data? If yes, treat them using an appropriate method. [1 mark]

```
# Write your code here
df.isnull().sum()
```

```
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
```

Observations:

1. The dataset has no null values.
2. Replaced the not given values in the rating column with zero

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? [2 marks]

```
# Write your code here
df.describe()
```

	cost_of_the_order	rating	food_preparation_time	delivery_time
count	1898.000000	1898.000000	1898.000000	1898.000000
mean	16.498851	2.659642	27.371970	24.161749
std	7.483812	2.195280	4.632481	4.972637
min	4.470000	0.000000	20.000000	15.000000
25%	12.080000	0.000000	23.000000	20.000000
50%	14.140000	4.000000	27.000000	25.000000
75%	22.297500	5.000000	31.000000	28.000000
max	35.410000	5.000000	35.000000	33.000000

Observations:

1. First quartile or 25% (Q1) of the orders have an order cost of 12.08 with ratings not included. The order preperation time is 23 mins or below and the delivery time is 20 mins or below.
2. The Median (Q2) or 50% of the orders have an order cost of 14.14 with ratings of 4. The order prep time is 31 mins or below and the delivery time is 28 mins or below.
3. The third quartile (Q3) or 75% of the orders have an order cost of 35.41 with an order rating of 5. The order prep time is 35 mins or below and the delivery time 33 mins or below.

Question 5: How many orders are not rated? [1 mark]

```
# Write the code here
df['rating'].value_counts(dropna=False)
```

```
0.0    736
5.0    588
4.0    386
3.0    188
Name: rating, dtype: int64
```

Observations:

1. There are 736 orders with no rating.

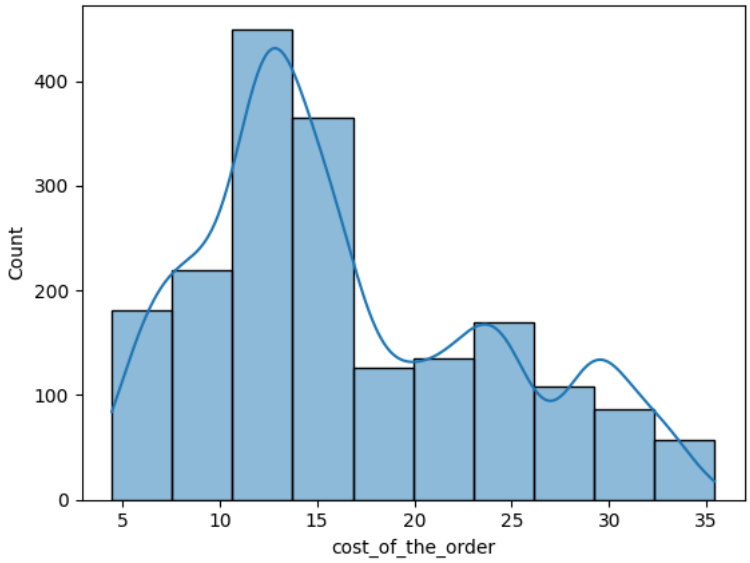
Exploratory Data Analysis (EDA)

Univariate Analysis

Question 6: Explore all the variables and provide observations on their distributions. (Generally, histograms, boxplots, countplots, etc. are used for univariate exploration.) [9 marks]

```
# Cost of the order
sns.histplot(data = df, x='cost_of_the_order',bins = 10, stat = 'count',kde=True)
```

```
<Axes: xlabel='cost_of_the_order', ylabel='Count'>
```

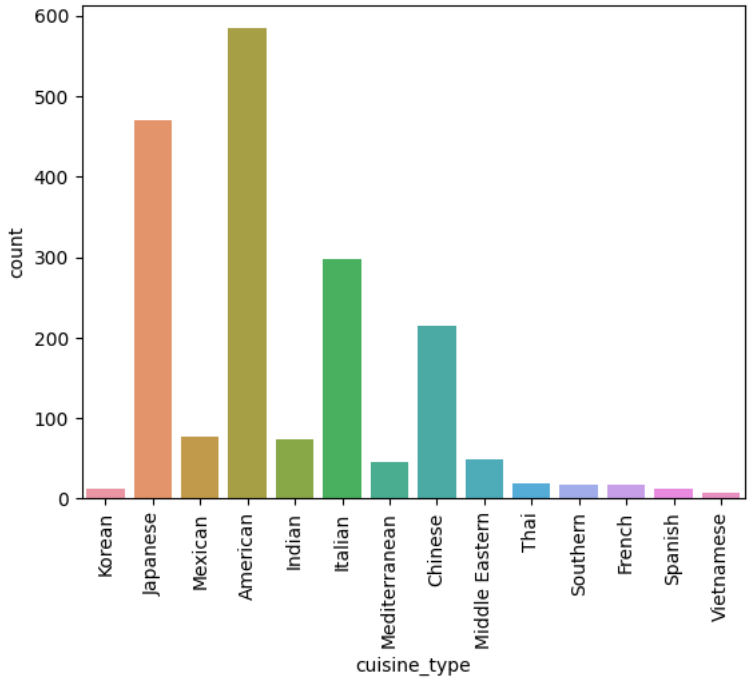



```
sns.boxplot(data = df, x='cost_of_the_order')
plt.show()
```

Observations: Order Cost

1. The data of cost of orders is skewed to the left, indicating low order prices are more frequent.
2. According to the boxplot, the median cost of the orders is $14 with skewness towards the right.
3. As students and working professionals are out target audiences, it is good to assume that students are more towards the left, where the order prices are lower and the working professionals more towards the right, with higher cost of orders.

```
# Cuisine Type
sns.countplot(data = df, x='cuisine_type')
plt.xticks(rotation=90)
plt.show()
```



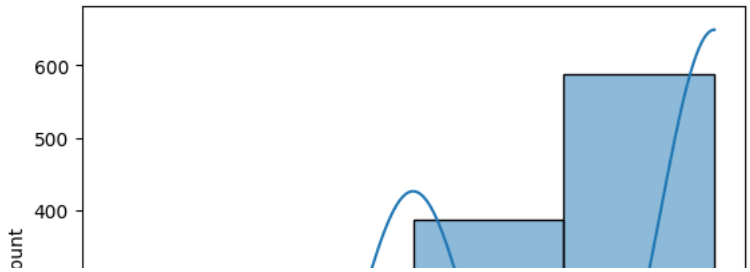
Obersvations: Cuisine Type

1. The top three frequently ordered cusines are American, Japaenese, and Italian.

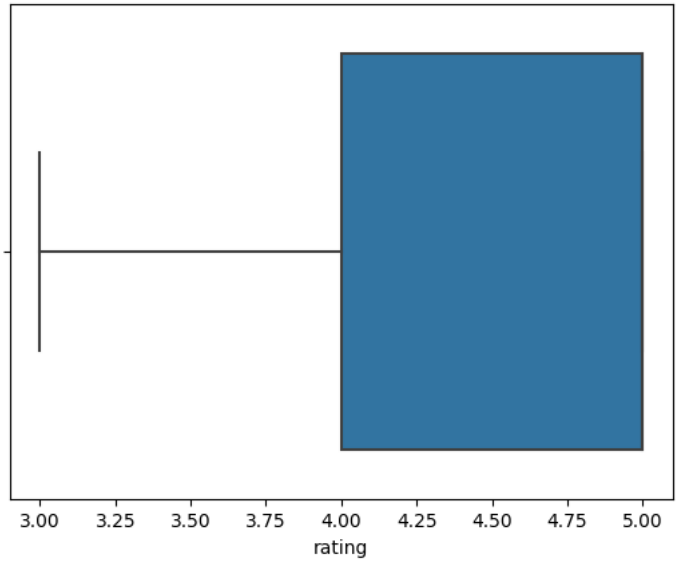
```
# Rating

#Removing Zero ratings
df_rating = df[df['rating']!= 0]

sns.histplot(data = df_rating, x='rating',bins = 4,stat = 'count',kde = True)
plt.show()
```



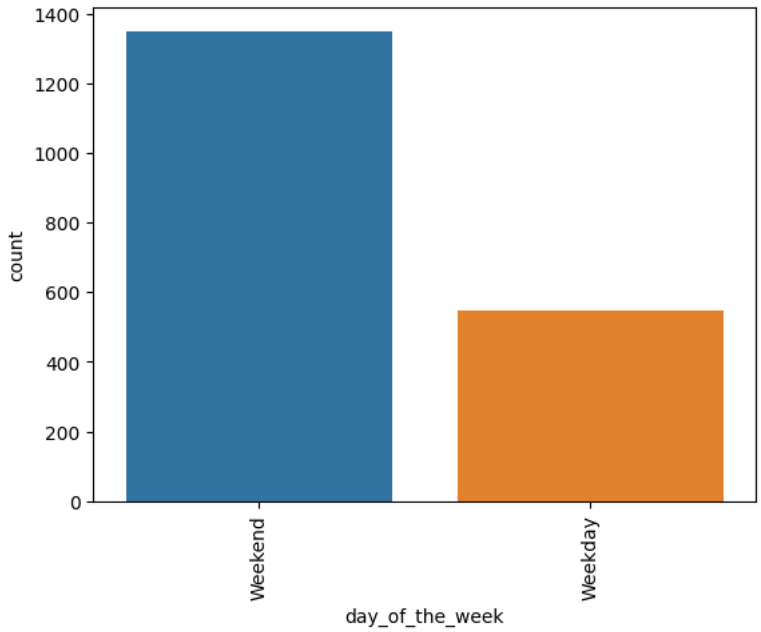

```
sns.boxplot(data = df_rating, x='rating')
plt.show()
```



Observations: Rating

1. The ratings are highly skewed to the right, show that ratings are mostly in the top rankings. However, we keep in-mind that 736 or 38.8% of the ratings are missing or zero.

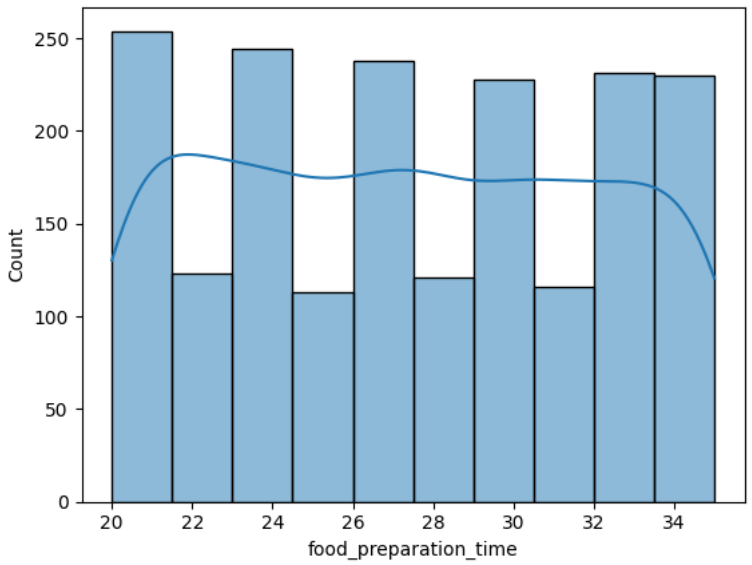
```
# Day of the week
sns.countplot(data=df,x='day_of_the_week')
plt.xticks(rotation=90)
plt.show()
```



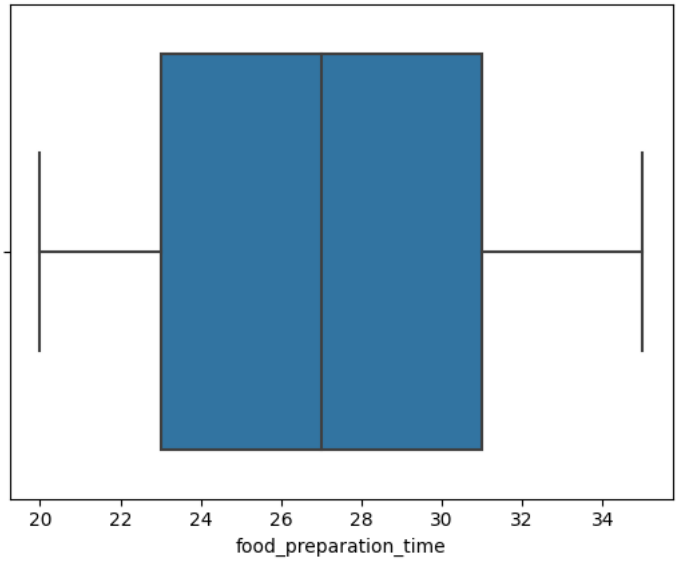
Observations: Day of the week

1. Order volume is higher on the weekends than weekdays.

```
# Food Prep
sns.histplot(data = df, x='food_preparation_time',bins = 10, stat = 'count',kde = True)
plt.show()
```



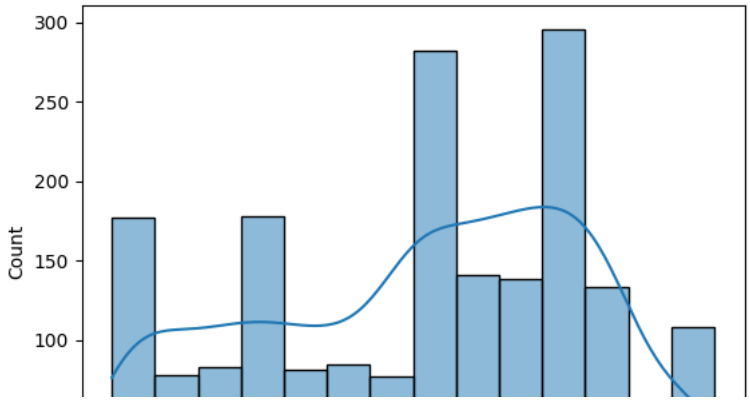

```
sns.boxplot(data = df, x='food_preparation_time')
plt.show();
```



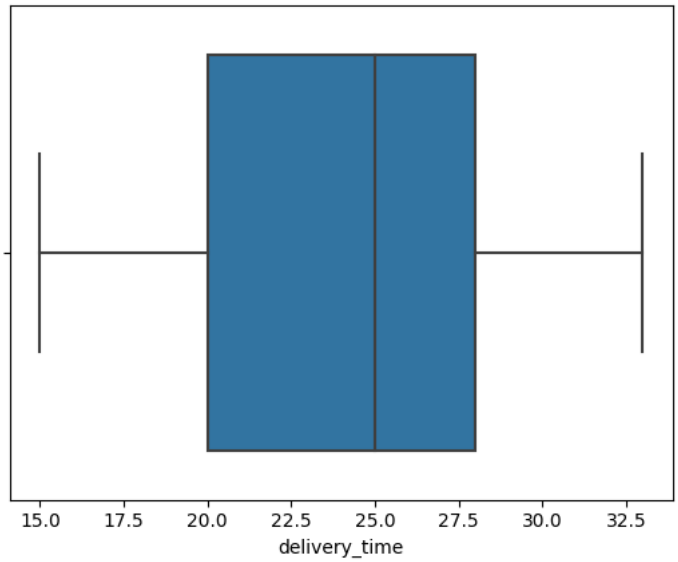
Observation: Food Prep Time

1. The common preparation times are between 20 mins and 36 mins with 27 mins as the median value.

```
# Delivery time
sns.histplot(data = df, x='delivery_time', stat = 'count',kde = True,)
plt.show()
```




```
sns.boxplot(data = df, x='delivery_time')
plt.show()
```



Observations: Delivery Time

1. The data is skewed to the left and the median delivery time is approx. 25 mins.
2. A majority of the order take delivery time between 25 and 28 minutes.

Question 7: Which are the top 5 restaurants in terms of the number of orders received? [1 mark]

```
# Write the code here
df[['restaurant_name', 'order_id']].groupby('restaurant_name').count().sort_values(by = 'order_id', axis = 0, ascending=False).head(5)
```

restaurant_name	order_id
Shake Shack	219
The Meatball Shop	132
Blue Ribbon Sushi	119
Blue Ribbon Fried Chicken	96
Parm	68

Observations:

1. The top five ordered resturants are (1) shake shack, (2) the meatball shop, (3) blue ribbon sushi, (4) blue ribbon fried chicken, and (5) parm.

Question 8: Which is the most popular cuisine on weekends? [1 mark]

```
# Write the code here
df[df['day_of_the_week'] =='Weekend'].groupby('cuisine_type').count().sort_values(by = 'order_id',axis = 0, ascending=False).head(5)
```

cuisine_type	order_id	customer_id	restaurant_name	cost_of_the_order	day_of_the_week	rating	food_preparation_time	deliv
American	415	415	415	415	415	415	415	
Japanese	335	335	335	335	335	335	335	
Italian	207	207	207	207	207	207	207	
Chinese	163	163	163	163	163	163	163	
Mexican	53	53	53	53	53	53	53	

Observations:

1. American is the most popular cuisine on the weekends.

Question 9: What percentage of the orders cost more than 20 dollars? [2 marks]

```
# Write the code here
total_observations = df['cost_of_the_order'].count()

orders_greater20 = df['cost_of_the_order'][df['cost_of_the_order']> 20].count()

percentage_greater20 = round((orders_greater20/total_observations)*100,2)

percentage_greater20
```

```
29.24
```

Observations:

29.2% of the orders cost mroe than $20

Question 10: What is the mean order delivery time? [1 mark]

```
# Write the code here
average_del_time = round(df[['delivery_time']].agg('mean'),2)
print(average_del_time)
```

```
delivery_time    24.16
dtype: float64
```

Observations:

1. The average time for deliveries is 24.2 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. [1 mark]

```
# Write the code here
df[['order_id','customer_id']].groupby('customer_id').count().sort_values(by = 'order_id',axis = 0, ascending=False).head(10)
```

customer_id	order_id
52832	13
47440	10
83287	9
250494	8
65009	7

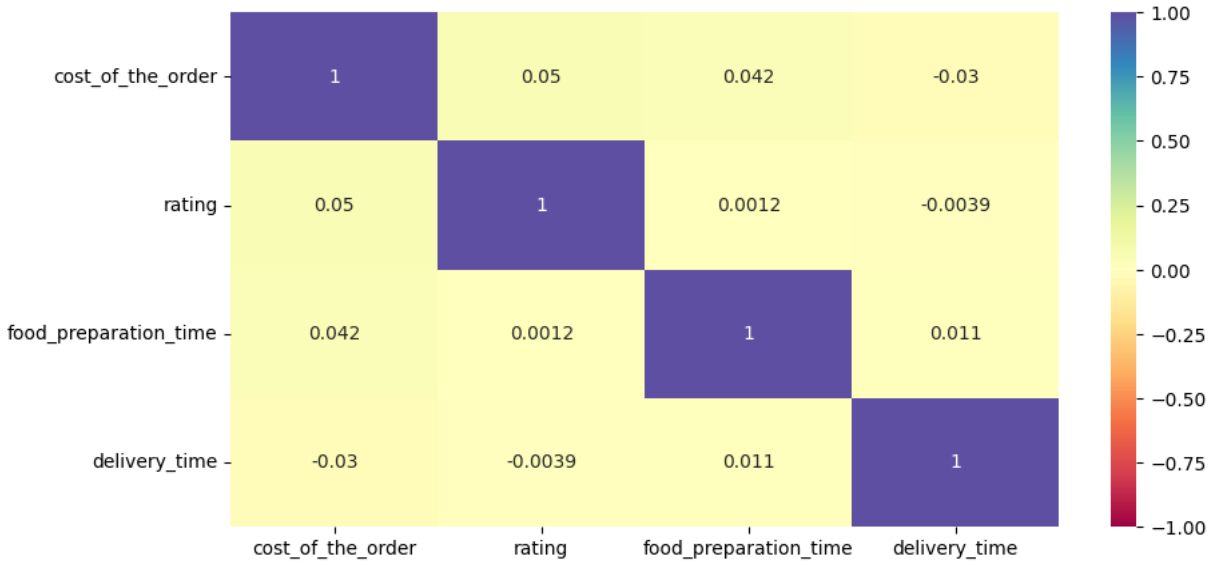
Observations:

1. There are 4 customers with 7 order counts and 2 with 6 order counts. The company would have to come up with a strategy to address this issue such as increasing the number of awards given to customers.

Multivariate Analysis

Question 12: 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) [10 marks]

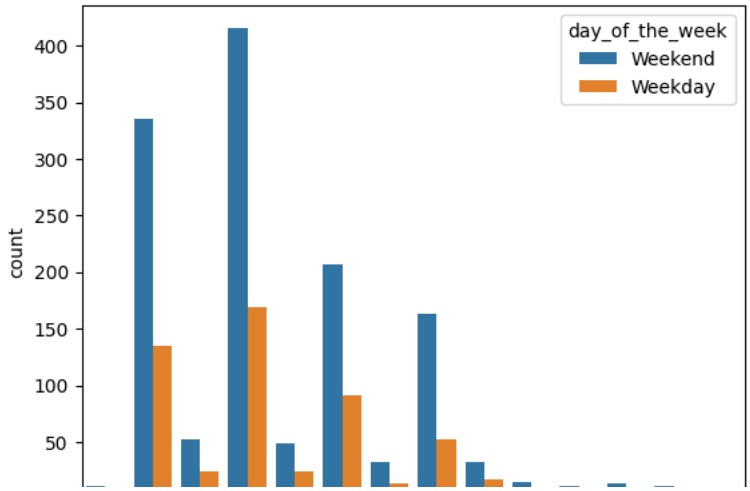
```
# Write the code here
plt.figure(figsize=(10,5))
sns.heatmap(df.corr(),annot=True,cmap='Spectral',vmin=-1,vmax=1)
plt.show()
```



Observations:

1. All the variables show weak correlation with each other.

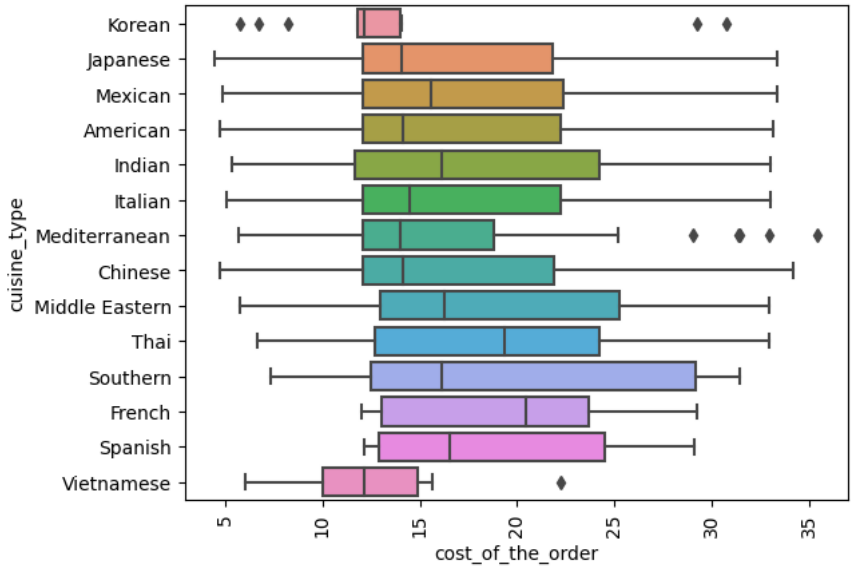
```
sns.countplot(data=df,x='cuisine_type',hue= 'day_of_the_week')
plt.xticks(rotation=90)
plt.show()
```



- Observations: Orders per cuisine per day of the week

1. Similar to the weekend order per cuisine. However, the order volume is low on weekdays compared to weekends.

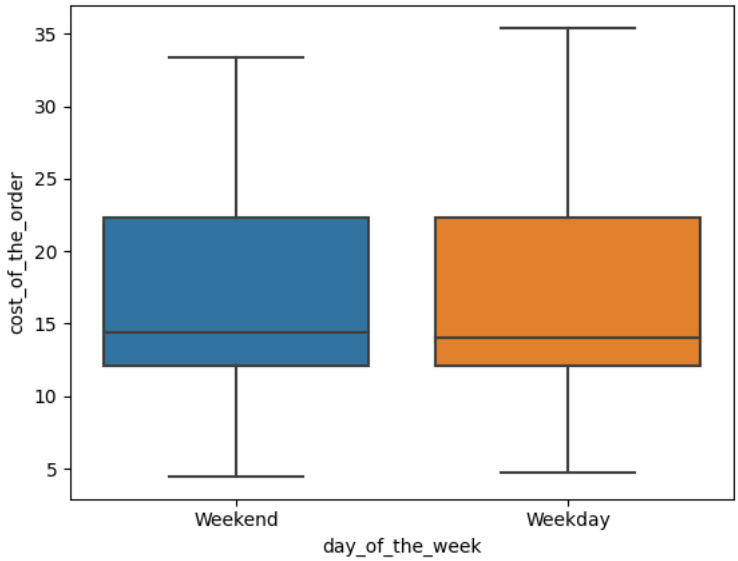
```
sns.boxplot(data=df,x='cost_of_the_order',y='cuisine_type')
plt.xticks(rotation=90)
plt.show();
```

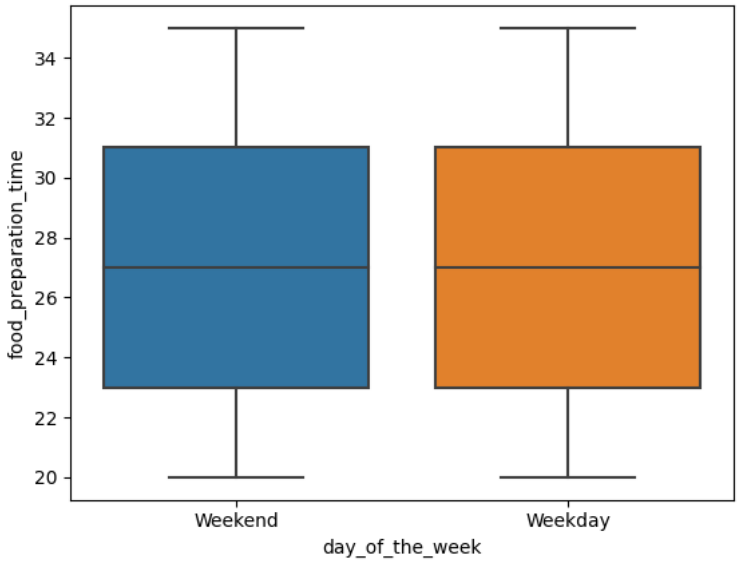


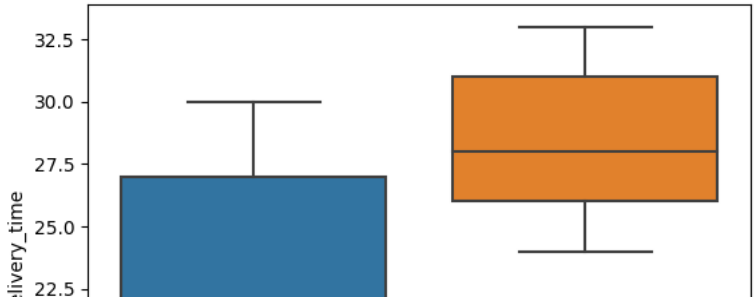
- Observation: Cost of order based on cuisine types.

1. Korean, Mediterranean, and vietnamese cuisines shows to have outliers.

```
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()
```



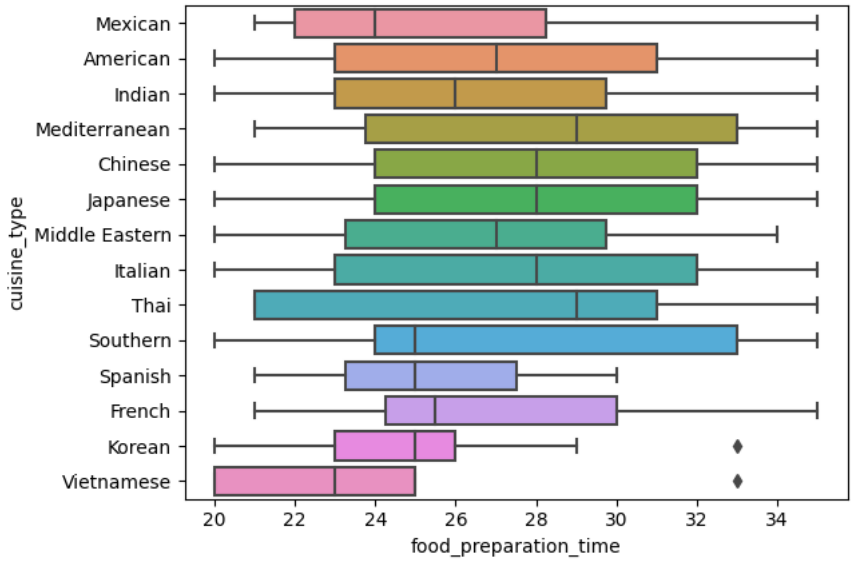


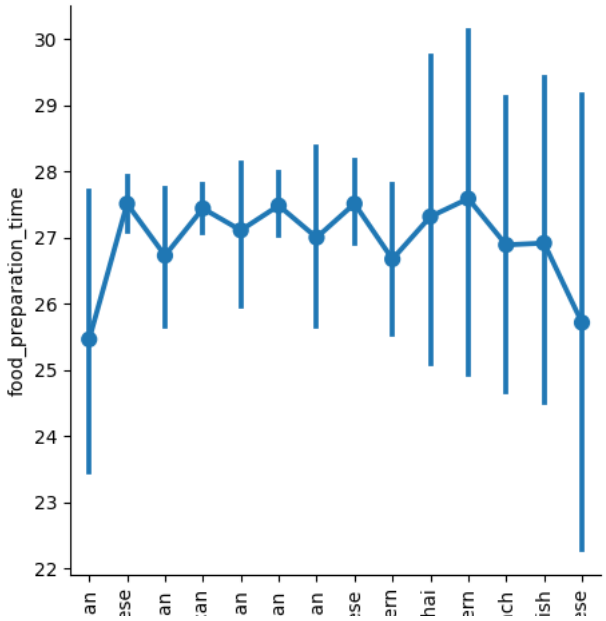


- Observations: Day of the week with other variables

1. Delivery times are higher on weekdays than weekends.
2. The median delivery time for weekends is 22.5 and weekdays is 28.5

```
sns.boxplot(data=df_rating,x='food_preparation_time',y='cuisine_type')
plt.show()
sns.catplot(data = df, x = 'cuisine_type',y = 'food_preparation_time',kind = 'point')
plt.xticks(rotation=90)
plt.show();
```





Observation:

1. There is no significant difference between delivery times among different cuisines.

cuisine_type

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. [3 marks]

```
# Write the code here
df2 = df[['restaurant_name','rating']][df['rating']!= 0].groupby('restaurant_name').mean()> 4

df2.reset_index(inplace=True)

df3 = df[['restaurant_name','rating']][df['rating']!= 0].groupby('restaurant_name').count()> 50

df3.reset_index(inplace=True)

qualified_restaurants = pd.merge(df3[df3['rating']== True], df2[df2['rating']== True],on ='restaurant_name')

qualified_restaurants
```

	restaurant_name	rating_x	rating_y
0	Blue Ribbon Fried Chicken	True	True
1	Blue Ribbon Sushi	True	True
2	Shake Shack	True	True
3	The Meatball Shop	True	True

Observations:

1. The table above shows the four resturants that qualify gor the promotional offer.

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. [3 marks]

```
# Write the code here
revenue = 0
income = 0

cost = pd.Series(df['cost_of_the_order'])

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),'assuming no commission charged for orders with cost below $5')
```

```
Net revenue generated is:  6166.3 assuming no commission charged for orders with cost below $5
```

Observations:

1. Foodhub generated $6166.3 in revenue, excluding the commision charged for order below 5.

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 delivered from the time the order is placed? (The food has to be prepared and then delivered.) [2 marks]

```
# Write the code here
df['total_prep_time'] = df['food_preparation_time']+df['delivery_time']
df.head()
```

	order_id	customer_id	restaurant_name	cuisine_type	cost_of_the_order	day_of_the_week	rating	food_preparation_time	del
0	1477147	337525	Hangawi	Korean	30.75	Weekend	0.0	25	
1	1477685	358141	Blue Ribbon Sushi Izakaya	Japanese	12.08	Weekend	0.0	25	
2	1477070	66393	Cafe Habana	Mexican	12.23	Weekday	5.0	23	
3	1477334	106968	Blue Ribbon Fried Chicken	American	29.20	Weekend	3.0	25	
4	1478249	76942	Dirty Bird to Go	American	11.59	Weekday	4.0	25	

```
# Calculating the percentage of order that take more than 60 mins (prep + deliver)
total_observations = df['total_prep_time'].count()
ordertime_greater60 = df['total_prep_time'][df['total_prep_time']> 60].count()

percent_greater60 = round((ordertime_greater60/total_observations)*100,2)
percent_greater60
```

```
10.54
```

Observations:

1. 10.5% of orders take more than 60 mins to prep and deliver.

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? [2 marks]

```
# Write the code here
average_del_time = df.groupby('day_of_the_week')[['delivery_time']].agg(['mean','sum','count','std'])
average_del_time
```

	delivery_time			
	mean	sum	count	std
day_of_the_week				
Weekday	28.340037	15502	547	2.891428
Weekend	22.470022	30357	1351	4.628938

Observations:

1. Mean delivery time on the weekend is 22.5 and weekday 28.3. This is due to people ordering out on the weekends than weekdays.

Conclusion and Recommendations

Question 17: 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.) [6 marks]

Conclusions:

- The rating seemed to be in top range, closer to 5. However, adding "Not Given" values can skew that data as it accounts for more than 30% of the dataset.
- The dataset has two peaks of order cost (one around $13 and the other 25$)
- The order demand on the weekneds is way higher than weekdays.

Business Recommendations:

- Entice customers to submit ratings
- The customer pool is sperated into students and working professionals
- More marketing geared towards weekend orders as the order flow is higher than weekdays.
- Consideration for the tie-breaker customers with same score. In additon, we have to consider financial impact of the strategy.

Work Cited:

1. Kaggle: Foodhub Review
2. Stackoverflow

✓ 0s completed at 10:53 PM