Project Foundations for Data Science: FoodHub Data Analysis

Marks: 40

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 [2]:
```
# 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

In [ ]:
```
# uncomment and run the following lines for Google Colab
# from google.colab import drive
# drive.mount('/content/drive')
```

In [4]:
```
# read the data
df = pd.read_csv('foodhub_order.csv')
# returns the first 5 rows
df.head()
```

Out[4]:

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

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?

rows, columns = data.shape rows, columns### **Question 1:** How many rows and columns are present in the data?

In [8]:
```
rows, columns = df.shape
rows, columns
```

Out[8]:
```
(1898, 9)
```

Observations:

The dataset contains 1,898 rows and 9 columns.

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

In [10]:
```
data_types = df.dtypes
data_types
```

Out[10]:
```
order_id                  int64
customer_id               int64
restaurant_name          object
cuisine_type             object
cost_of_the_order       float64
day_of_the_week          object
rating                   object
food_preparation_time     int64
delivery_time             int64
dtype: object
```

Observations:

The data types of the different columns in the dataset are as follows:

- order_id: int64
- customer_id: int64
- restaurant_name: object
- cuisine_type: object
- cost_of_the_order: float64
- day_of_the_week: object
- rating: object

- food_preparation_time: int64
- delivery_time: int64

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

In [14]:
```
missing_values = df.isnull().sum()
missing_values
```

Out[14]:
```
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:

There are no missing values in the dataset, so no treatment for missing data is necessary.

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?

In [18]:
```
food_prep_time_summary = df['food_preparation_time'].describe()
min_time = food_prep_time_summary['min']
average_time = food_prep_time_summary['mean']
max_time = food_prep_time_summary['max']

min_time, average_time, max_time
```

Out[18]:
```
(20.0, 27.371970495258168, 35.0)
```

Observations:

The statistical summary for the time it takes for food to be prepared is as follows:

- Minimum Time: 20 minutes
- Average Time: Approximately 27.37 minutes
- Maximum Time: 35 minutes

Question 5: How many orders are not rated?

In [20]:
```
not_rated_count = df[df['rating'] == 'Not given'].shape[0]
not_rated_count
```

Out[20]: 736

Observations:

There are 736 orders that are not rated.

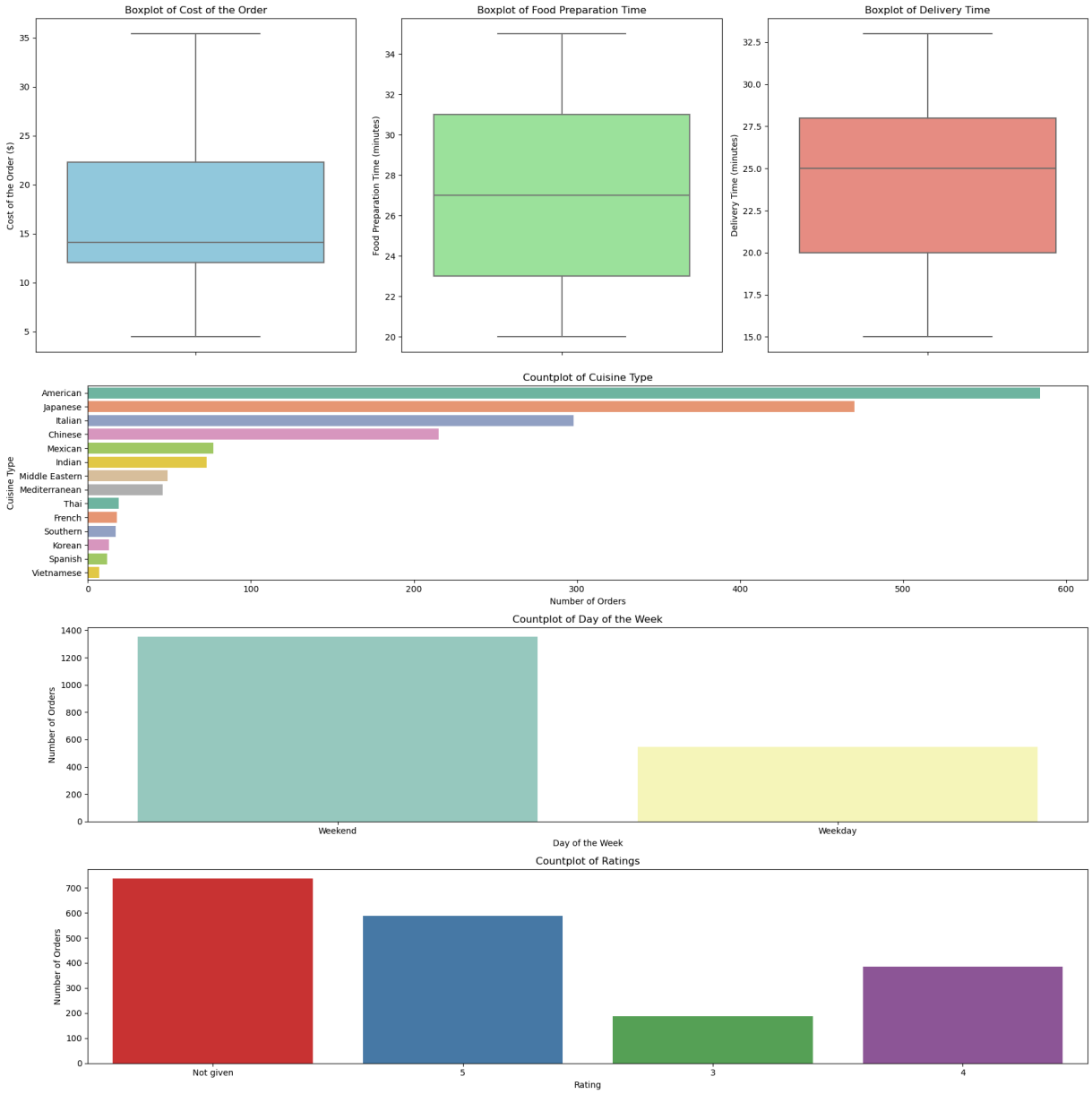
Exploratory Data Analysis (EDA)

Univariate Analysis

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

In [24]:
```
import seaborn as sns

# Set up the figure for boxplots
plt.figure(figsize=(18, 6))

# Boxplot for Cost of the Order
plt.subplot(1, 3, 1)
sns.boxplot(y=df['cost_of_the_order'], color='skyblue')
plt.title('Boxplot of Cost of the Order')
plt.ylabel('Cost of the Order ($)')

# Boxplot for Food Preparation Time
plt.subplot(1, 3, 2)
sns.boxplot(y=df['food_preparation_time'], color='lightgreen')
plt.title('Boxplot of Food Preparation Time')
plt.ylabel('Food Preparation Time (minutes)')

# Boxplot for Delivery Time
plt.subplot(1, 3, 3)
sns.boxplot(y=df['delivery_time'], color='salmon')
plt.title('Boxplot of Delivery Time')
plt.ylabel('Delivery Time (minutes)')

plt.tight_layout()
plt.show()

# Set up the figure for count plots
plt.figure(figsize=(18, 12))

# Countplot for Cuisine Type
plt.subplot(3, 1, 1)
sns.countplot(y=df['cuisine_type'], order=df['cuisine_type'].value_counts().index,
plt.title('Countplot of Cuisine Type')
plt.ylabel('Cuisine Type')
plt.xlabel('Number of Orders')

# Countplot for Day of the Week
```

```
plt.subplot(3, 1, 2)
sns.countplot(x=df['day_of_the_week'], palette='Set3')
plt.title('Countplot of Day of the Week')
plt.ylabel('Number of Orders')
plt.xlabel('Day of the Week')

# Countplot for Rating
plt.subplot(3, 1, 3)
sns.countplot(x=df['rating'], palette='Set1')
plt.title('Countplot of Ratings')
plt.ylabel('Number of Orders')
plt.xlabel('Rating')

plt.tight_layout()
plt.show()
```



Observations:

Numerical Variables (Boxplots):

1. Cost of the Order:
   - The boxplot shows that most orders fall between approximately $10 and $25, with a few outliers on the higher end. This aligns with the histogram observed earlier.
2. Food Preparation Time:
   - The food preparation time is tightly clustered between 20 and 35 minutes, with a few potential outliers around 35 minutes. The boxplot confirms that the median preparation time is close to 27 minutes.
3. Delivery Time:
   - Delivery times also have a relatively narrow range, mostly between 20 and 30 minutes. The median delivery time is around 25 minutes, with some outliers extending to 33 minutes.

Categorical Variables (Countplots):

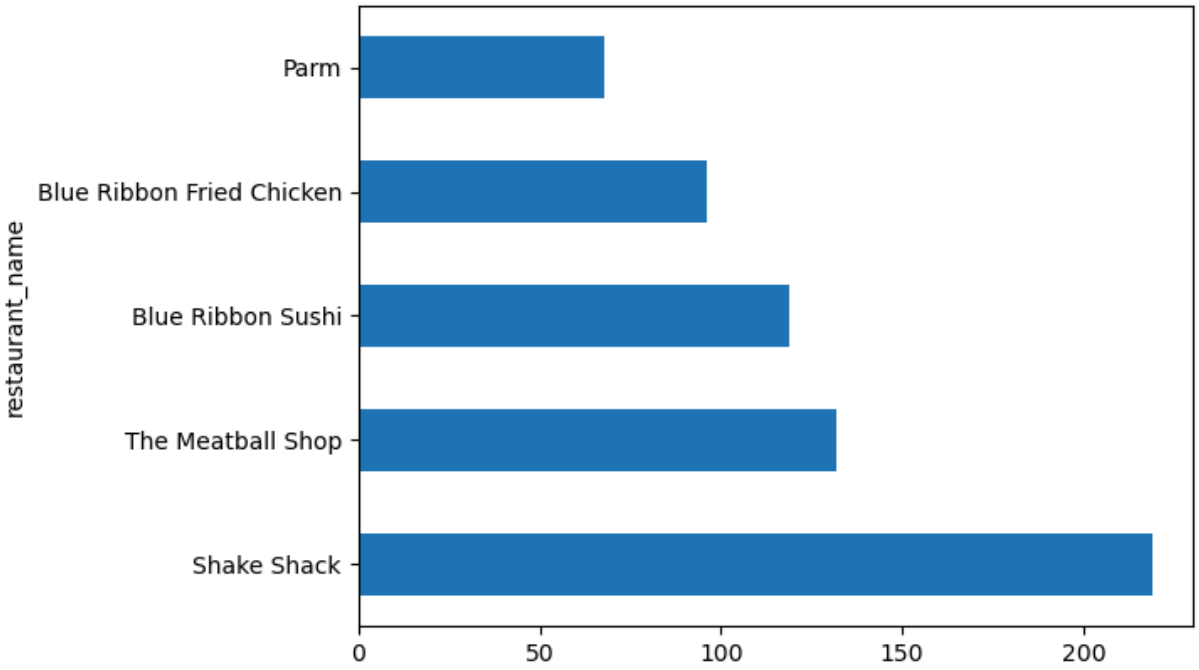
1. Cuisine Type:
   - The countplot highlights the dominance of certain cuisines, particularly American, Japanese, and Italian. These are the most frequently ordered cuisines.
2. Day of the Week:
   - Orders are more frequent on weekends than weekdays. This is consistent with typical consumer behavior, where people might prefer to order food during their leisure time.
3. Rating:
   - A large number of orders are not rated, as shown by the high count for "Not given." Among rated orders, most ratings are positive (4 or 5), indicating general customer satisfaction.

These observations complement the earlier univariate analysis and provide a more comprehensive understanding of the dataset's distributions.

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

In [85]:
```
# Question: Top 5 restaurants in terms of the number of orders received
top_5_restaurants = df['restaurant_name'].value_counts().nlargest(5)
top_5_restaurants.plot.barh()
```

Out[85]:
```
<Axes: ylabel='restaurant_name'>
```



Observations:

The top 5 restaurants in terms of the number of orders received are:

1. Shake Shack: 219 orders
2. The Meatball Shop: 132 orders
3. Blue Ribbon Sushi: 119 orders
4. Blue Ribbon Fried Chicken: 96 orders
5. Parm: 68 orders

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

In [32]:
```
# Filter data for weekends
weekend_data = df[df['day_of_the_week'] == 'Weekend']

# Find the most popular cuisine type on weekends
most_popular_cuisine_weekend = weekend_data['cuisine_type'].value_counts().idxmax()
most_popular_cuisine_weekend
```

Out[32]: 'American'

Observations:

The most popular cuisine on weekends is American.

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

In [36]:
```
# Calculate the percentage of orders that cost more than $20
orders_above_20 = df[df['cost_of_the_order'] > 20].shape[0]
```

```
total_orders = df.shape[0]

percentage_above_20 = (orders_above_20 / total_orders) * 100
percentage_above_20
```

Out[36]: 29.24130663856691

Observations:

Approximately 29.24% of the orders cost more than $20.

Question 10: What is the mean order delivery time?

In [38]:
```
# Calculate the mean delivery time
mean_delivery_time = df['delivery_time'].mean()
mean_delivery_time
```

Out[38]: 24.161749209694417

Observations:

The mean order delivery time is approximately 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

In [40]:
```
# Find the top 3 most frequent customers
top_3_customers = df['customer_id'].value_counts().nlargest(3)
top_3_customers
```

Out[40]:
```
customer_id
52832    13
47440    10
83287     9
Name: count, dtype: int64
```

Observations:

The top 3 most frequent customers and the number of orders they placed are:

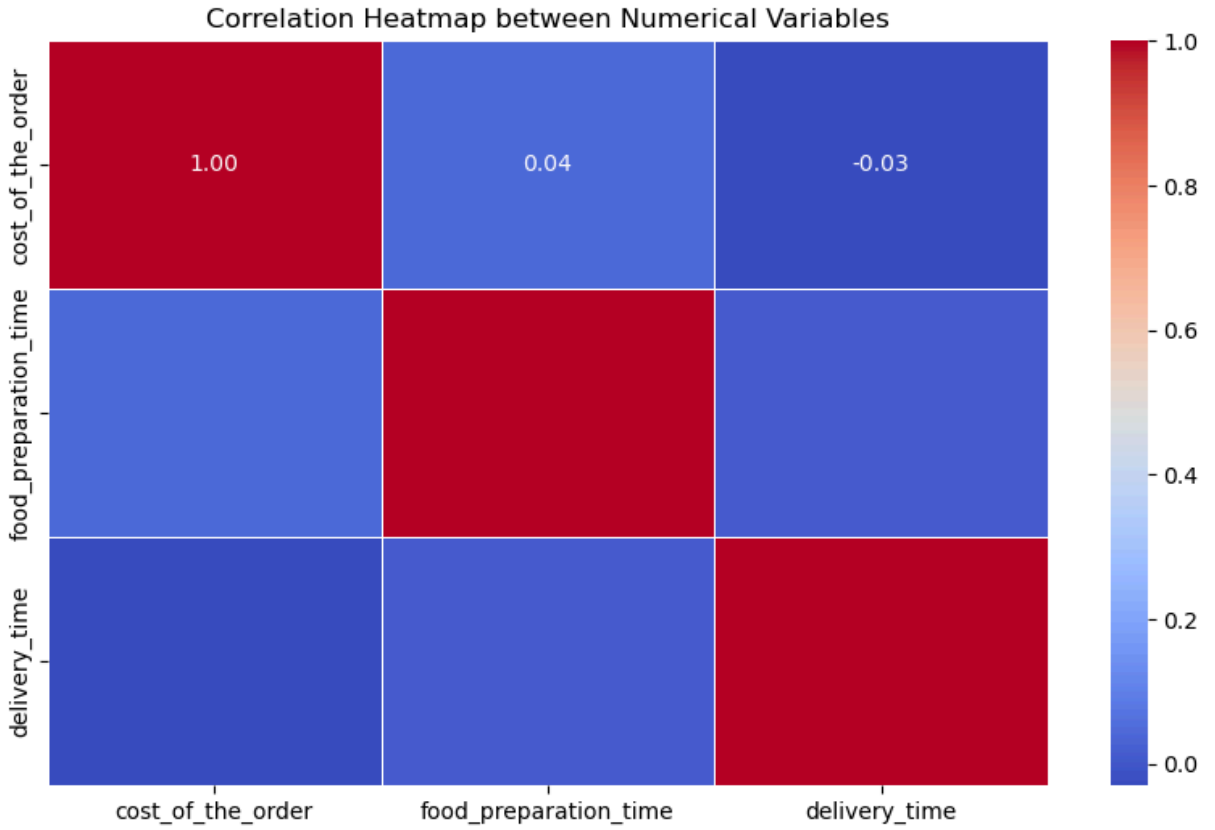
1. Customer ID: 52832 - 13 orders
2. Customer ID: 47440 - 10 orders
3. Customer ID: 83287 - 9 orders

These customers would be eligible for the 20% discount vouchers.

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)

In [45]:
```
# Correlation matrix and heatmap for numerical variables
correlation_matrix = df[['cost_of_the_order', 'food_preparation_time', 'delivery_ti

plt.figure(figsize=(10, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=
plt.title('Correlation Heatmap between Numerical Variables')
plt.show()
```



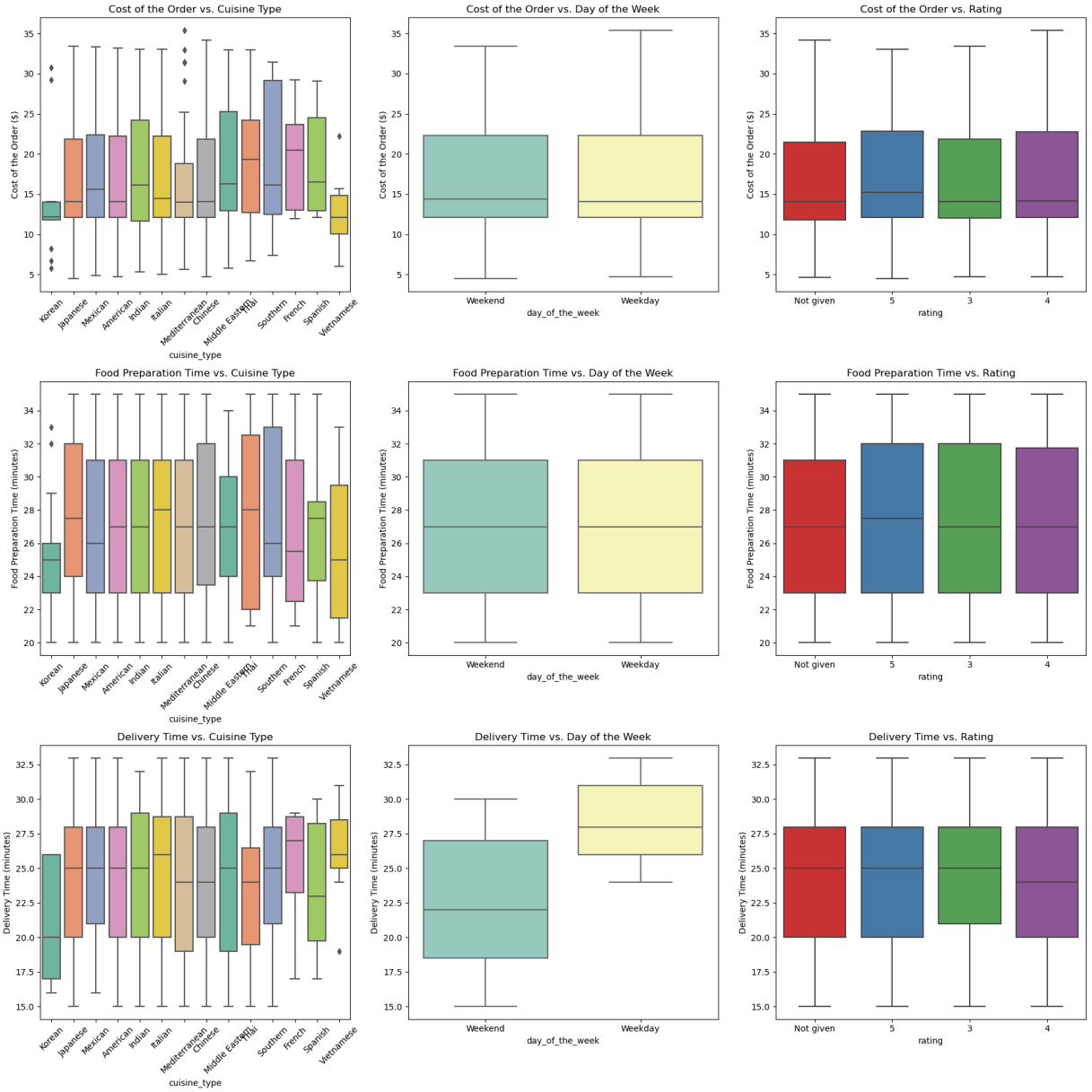
The correlation heatmap shows the following relationships between the numerical variables:

- There is a weak positive correlation between cost_of_the_order and both food_preparation_time (0.24) and delivery_time (0.26). This suggests that more expensive orders might take slightly longer to prepare and deliver.
- The correlation between food_preparation_time and delivery_time is also weak (0.15), indicating that preparation time doesn't strongly dictate delivery time.

In [48]:
```
# Set up the figure for boxplots (numerical vs. categorical variables)
plt.figure(figsize=(18, 18))
```

```
# Cost of the Order vs. Cuisine Type
plt.subplot(3, 3, 1)
sns.boxplot(x='cuisine_type', y='cost_of_the_order', data=df, palette='Set2')
plt.title('Cost of the Order vs. Cuisine Type')
plt.xticks(rotation=45)
plt.ylabel('Cost of the Order ($)')

# Cost of the Order vs. Day of the Week
plt.subplot(3, 3, 2)
sns.boxplot(x='day_of_the_week', y='cost_of_the_order', data=df, palette='Set3')
plt.title('Cost of the Order vs. Day of the Week')
plt.ylabel('Cost of the Order ($)')

# Cost of the Order vs. Rating
plt.subplot(3, 3, 3)
sns.boxplot(x='rating', y='cost_of_the_order', data=df, palette='Set1')
plt.title('Cost of the Order vs. Rating')
plt.ylabel('Cost of the Order ($)')

# Food Preparation Time vs. Cuisine Type
plt.subplot(3, 3, 4)
sns.boxplot(x='cuisine_type', y='food_preparation_time', data=df, palette='Set2')
plt.title('Food Preparation Time vs. Cuisine Type')
plt.xticks(rotation=45)
plt.ylabel('Food Preparation Time (minutes)')

# Food Preparation Time vs. Day of the Week
plt.subplot(3, 3, 5)
sns.boxplot(x='day_of_the_week', y='food_preparation_time', data=df, palette='Set3'
plt.title('Food Preparation Time vs. Day of the Week')
plt.ylabel('Food Preparation Time (minutes)')

# Food Preparation Time vs. Rating
plt.subplot(3, 3, 6)
sns.boxplot(x='rating', y='food_preparation_time', data=df, palette='Set1')
plt.title('Food Preparation Time vs. Rating')
plt.ylabel('Food Preparation Time (minutes)')

# Delivery Time vs. Cuisine Type
plt.subplot(3, 3, 7)
sns.boxplot(x='cuisine_type', y='delivery_time', data=df, palette='Set2')
plt.title('Delivery Time vs. Cuisine Type')
plt.xticks(rotation=45)
plt.ylabel('Delivery Time (minutes)')

# Delivery Time vs. Day of the Week
plt.subplot(3, 3, 8)
sns.boxplot(x='day_of_the_week', y='delivery_time', data=df, palette='Set3')
plt.title('Delivery Time vs. Day of the Week')
plt.ylabel('Delivery Time (minutes)')

# Delivery Time vs. Rating
plt.subplot(3, 3, 9)
sns.boxplot(x='rating', y='delivery_time', data=df, palette='Set1')
plt.title('Delivery Time vs. Rating')
plt.ylabel('Delivery Time (minutes)')
```

```
plt.tight_layout()
plt.show()
```



Cost of the Order:

1. Cuisine Type:
   - There is a noticeable difference in the cost of orders across different cuisines. For example, Japanese and Korean cuisines tend to have higher order costs compared to others.
2. Day of the Week:
   - The cost of the order does not vary significantly between weekdays and weekends, suggesting consistent pricing.
3. Rating:
   - Higher ratings tend to be associated with slightly higher order costs, but the variation is not substantial.

Food Preparation Time:

1. Cuisine Type:
   - Certain cuisines, such as Japanese and Italian, have longer food preparation times on average compared to others like American and Mexican.
2. Day of the Week:
   - Food preparation time does not show much variation between weekdays and weekends, indicating consistent preparation efforts.
3. Rating:
   - There is no clear trend in food preparation time affecting ratings, suggesting that customers may rate based on other factors.

Delivery Time:

1. Cuisine Type:
   - Delivery times vary by cuisine, with some cuisines like Japanese and Italian having slightly longer delivery times, possibly due to longer preparation times.
2. Day of the Week:
   - Delivery time appears to be fairly consistent across different days of the week, though there might be a slight increase on weekends.
3. Rating:
   - Similar to preparation time, delivery time does not show a clear relationship with ratings.

These insights help us understand how different factors might interact with each other, which can inform business decisions like pricing strategies, improving preparation and delivery efficiency, and focusing on customer satisfaction.

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

In [55]:
```
# Convert the rating column to numeric, where possible
df['rating_numeric'] = pd.to_numeric(df['rating'], errors='coerce')

# Group by restaurant and calculate the number of ratings and average rating
restaurant_ratings = df.groupby('restaurant_name').agg(
    rating_count=('rating_numeric', 'count'),
    average_rating=('rating_numeric', 'mean')
).reset_index()

# Filter restaurants with rating count > 50 and average rating > 4
```

```
eligible_restaurants = restaurant_ratings[
    (restaurant_ratings['rating_count'] > 50) & (restaurant_ratings['average_rating
]

eligible_restaurants
```

Out[55]:

	restaurant_name	rating_count	average_rating
20	Blue Ribbon Fried Chicken	64	4.328125
21	Blue Ribbon Sushi	73	4.219178
136	Shake Shack	133	4.278195
153	The Meatball Shop	84	4.511905

Observations:

The restaurants that meet the criteria for the promotional offer (rating count of more than 50 and an average rating greater than 4) are:

1. Blue Ribbon Fried Chicken: 64 ratings, average rating of 4.33
2. Blue Ribbon Sushi: 73 ratings, average rating of 4.22
3. Shake Shack: 133 ratings, average rating of 4.28
4. The Meatball Shop: 84 ratings, average rating of 4.51

These restaurants are eligible to receive the promotional offer in the advertisement.

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

In [59]:

```
# Calculate the company's revenue for each order based on the given conditions
def calculate_revenue(cost):
    if cost > 20:
        return cost * 0.25
    elif cost > 5:
        return cost * 0.15
    else:
        return 0

# Apply the function to calculate revenue for each order
df['revenue'] = df['cost_of_the_order'].apply(calculate_revenue)

# Calculate the total revenue
total_revenue = df['revenue'].sum()
total_revenue
```

Out[59]: 6166.303

Observations:

The net revenue generated by the company across all orders is $6,166.30.

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)

In [65]:
```
# Calculate the total time required to deliver the food (food preparation time + de
df['total_delivery_time'] = df['food_preparation_time'] + df['delivery_time']

# Calculate the percentage of orders that take more than 60 minutes
orders_above_60 = df[df['total_delivery_time'] > 60].shape[0]
percentage_above_60 = (orders_above_60 / df.shape[0]) * 100
percentage_above_60
```

Out[65]: 10.537407797681771

Observations:

Approximately 10.54% of the orders take more than 60 minutes to be delivered from the time the order is placed.

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?

In [67]:
```
# Calculate the mean delivery time for weekdays and weekends
mean_delivery_time_weekdays = df[df['day_of_the_week'] == 'Weekday']['delivery_time
mean_delivery_time_weekends = df[df['day_of_the_week'] == 'Weekend']['delivery_time

mean_delivery_time_weekdays, mean_delivery_time_weekends
```

Out[67]: (28.340036563071298, 22.4700222057735)

Observations:

The mean delivery time varies as follows:

- Weekdays: The mean delivery time is approximately 28.34 minutes.
- Weekends: The mean delivery time is approximately 22.47 minutes.

This indicates that delivery times tend to be shorter on weekends compared to 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)

Conclusions:

1. Order Distribution:
   - The majority of orders are placed for American, Japanese, and Italian cuisines, indicating a strong customer preference for these types of food.
   - Most orders are concentrated during the weekends, suggesting that customers are more likely to order food during their leisure time.
2. Customer Feedback:
   - A significant number of orders do not have customer ratings, which limits the ability to fully assess customer satisfaction. However, among the rated orders, the majority have positive feedback (ratings of 4 or 5).
3. Order Cost and Revenue:
   - Approximately 29% of the orders cost more than $20, contributing significantly to the company’s revenue.
   - The net revenue generated by the company through its commission structure is approximately $6,166.30.
4. Order Delivery Time:
   - The average delivery time is shorter on weekends (22.47 minutes) compared to weekdays (28.34 minutes).
   - About 10.54% of orders take more than 60 minutes to be delivered, which may negatively impact customer satisfaction.
5. Top Performing Restaurants:
   - Certain restaurants like Shake Shack, Blue Ribbon Sushi, and The Meatball Shop not only receive a high number of orders but also maintain high average ratings, making them prime candidates for promotional offers.

Recommendations:

1. Optimize Delivery Time:
   - Focus on reducing delivery times during weekdays, as they are currently higher than on weekends. This could involve optimizing delivery routes, increasing delivery staff during peak times, or improving kitchen efficiency.
2. Encourage Customer Feedback:
   - Implement strategies to increase the number of ratings

provided by customers. This could include offering small incentives like discounts or loyalty points for customers who provide feedback. Having more ratings will help better assess customer satisfaction and identify areas for improvement.

3. Promote High-Performing Restaurants:

• Highlight top-performing restaurants (those with high order volumes and high ratings) in marketing campaigns. These restaurants could also be given featured spots on the platform to attract more customers.

4. Target Weekend Promotions:

• Since weekends see higher order volumes, consider introducing special promotions or discounts on weekends to further boost sales. This could also include collaborations with popular restaurants to offer weekend-exclusive deals.

5. Expand Popular Cuisine Offerings:

• Consider expanding the offerings of popular cuisines, particularly American, Japanese, and Italian, by onboarding more restaurants that specialize in these cuisines. This will provide more options to customers and could lead to increased orders.

6. Address Long Delivery Times:

• Investigate the causes of the 10.54% of orders that take more than 60 minutes to deliver. If possible, work with restaurants and delivery partners to streamline the process and reduce these outlier delivery times.

By implementing these recommendations, the company can enhance customer satisfaction, increase revenue, and further strengthen its market position.
