Name : Ann Mariya Jomon

Project 1

FoodHub

Python Foundations

23rd September 2022

Contents / Agenda

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

Executive Summary

| Actionable Insights |
| Recommendations |

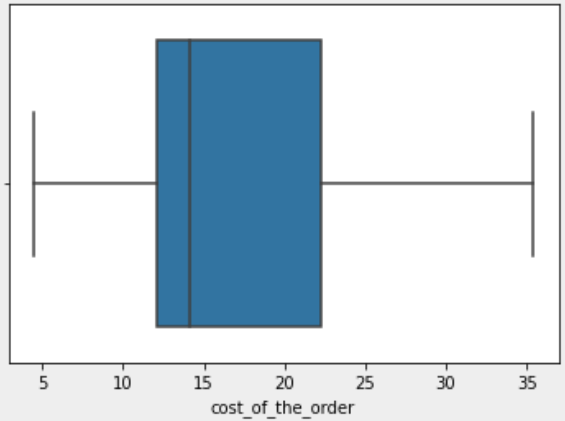
Actionable Insights

Demand

- American cuisine is the most popular and highly demanded cuisine. It is also the most popular cuisine on the weekend.
- The top 5 restaurants in terms of number of orders are:
    1. Shake Shack
    2. The Meatball Shop
    3. Blue Ribbon Sushi
    4. Blue Ribbon Fried Chicken
    5. Parm
- Majority (71%) of the orders are made during the weekends.
- The top 3 frequent customers are:
    1. 52832
    2. 47440
    3. 83278

Cost

- 29.24% of total orders cost more than $20.
- 50% of orders of all the cuisines is above $10.
- 75% of orders of American cuisine is below $25.
- Orders with higher costs have higher ratings.
- The cost is not impacted much by the food preparation time or delivery time.



Actionable Insights

Time

- 50% of the orders take more than 27 mins for food preparation.
- The average food preparation time is almost equal for all the ratings. But the variance decreases for higher ratings.
- 50% of the orders take more than 25 mins to deliver.
- Delivery time is higher during the weekdays than the weekends.
- Delivery time has an impact on the ratings. Orders rated 5 have a delivery time ranging between 23.75 mins and 24.75 mins.
- 10.5% of the orders take more than an hour to get delivered.

Revenue

- FoodHub's revenue varies in accordance with the cost of the orders.
- Shake Shack generated the highest revenue.

Restaurant name (top 10)	Revenue generated
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

Recommendations

- Customers can be provided with offers, discounts, or any other benefit when they cross a particular limit of orders. For example, a 10% discount can be availed by the customer if they have used the app to place more than 10 orders.
- Such offers and discounts can help in increasing the number of orders which in turn helps FoodHub to generate more revenue.
- FoodHub can focus on the American cuisine to improve their revenue. They can work with restaurants that have American cuisine.
- Special promotions can be made for the top 5 restaurants. This will act as an incentive for other restaurants to perform better and increase their demand. The promotional offer mentioned in Q13 will also be beneficial.
- Most orders are placed during the weekends and hence there should be sufficient availability of delivery vehicles and drivers to meet the demand during the weekends.

Recommendations

- Orders that take more than an hour to deliver will need to be investigated further to identify the reason for the delay.
- Delivery time is higher during the weekdays. Appropriate timings and faster routes should be selected using the maps during the weekdays to avoid traffic congestion and ensure faster delivery.
- The delivery time taken for orders rated 5 can be analyzed to identify the ideal time that could be taken to deliver the orders.
- Further analysis needs to be done on the ratings. In order to provide a better customer service and improved ratings, FoodHub must understand the actual reasons for the given ratings and use that to improve the ratings from their end.
- Food preparation time needs to be analyzed separately for each cuisine within each restaurant and find ways to reduce unnecessary time taken. This will help in ensuring faster delivery.

Business Problem Overview and Solution Approach

Problem Overview

- **FoodHub** is a **food aggregator** company providing access to multiple restaurants through its mobile application. The app helps customers to place orders from various restaurants. The company has stored all the **data** of the orders placed through their online portal.
- They wish to **analyze** this data to understand the **demand** for different restaurants and improve their **customer experience**.

Solution Approach

- The data was analyzed using **Python** on **Jupyter** Notebooks.
- Data visualization tools such as **boxplot, countplot, histplot, pointplot, and heatmap** were used. **Univariate and multivariate** analysis was performed using the above-mentioned data visualization tools.
- Insights and Recommendations were derived based on the observations from the output of all the codes as per the questions mentioned.

Data Overview

Introduction to the dataset and its properties

| Description of the data |
| Introducing variables |
| First 5 rows of dataset |
| Question-wise analysis (Q1 – Q5) |

Data Overview

- The dataset contains details of the orders placed on the FoodHub app by different customers including the respective restaurant names and cuisines selected along with the cost.
- It also contains details regarding the time taken for food preparation and delivery on both weekends and weekdays for each order.
- The ratings for each order is also provided.

Variable	Description
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_of_the_order	Cost of the order
day_of_the_week	Weekend / Weekday
rating	Rating by customer (out of 5)
food_preparation_time	Time taken to prepare food (in mins.)
delivery_time	Time taken to deliver the food (in mins.)

Data Overview

First 5 rows of the data set

In [3]:
```
# Read the data
df = pd.read_csv('C:/Users/jmnjo/Documents/DSBA Course/Python Foundations/Project/foodhub_order.csv') ## Fill the blank to re
# Returns the first 5 rows
df.head()
```

Out[3]:

	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

Data Overview (Answers to Q1 – Q5)

Q1

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

```
In [4]: # Check the shape of the dataset
        df.shape ## Fill in the blank

Out[4]: (1898, 9)
```

Q2

Question 2: What are the datatypes of the different columns in the dataset? [0.5 mark]

```
In [5]: 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
```

Notes:

- **Rows** (No. of orders) : 1898
- **Columns** : 9
- **Datatypes** :
  - integer – 4
  - object – 4
  - float – 1
- **No null values** - 1898 non-null values for all columns

Data Overview (Answers to Q1 – Q5)

Q3

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

```
In [6]: # Checking for missing values in the data
        df.isnull().sum()  #Write the appropriate function to print the sum of null values for each column

Out[6]: 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
```

Q4

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]

```
In [8]: # Get the summary statistics of the numerical data
        df.describe().T ## Write the appropriate function to print the statitical summary of the data (Hint - you have seen this in t
```

Out[8]:

	count	mean	std	min	25%	50%	75%	max
order_id	1898.0	1.477496e+06	548.049724	1476547.00	1477021.25	1477495.50	1.477970e+06	1478444.00
customer_id	1898.0	1.711685e+05	113698.139743	1311.00	77787.75	128600.00	2.705250e+05	405334.00
cost_of_the_order	1898.0	1.649885e+01	7.483812	4.47	12.08	14.14	2.229750e+01	35.41
food_preparation_time	1898.0	2.737197e+01	4.632481	20.00	23.00	27.00	3.100000e+01	35.00
delivery_time	1898.0	2.416175e+01	4.972637	15.00	20.00	25.00	2.800000e+01	33.00

Notes:

- There are **no missing values** in the data.
- The **minimum** time taken to prepare food is **20 mins**.
- The **average** time taken to prepare food is **27 mins**.
- The **maximum** time taken to prepare food is **35 mins**.

Data Overview (Answers to Q1 – Q5)

Q5

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

```
In [9]: df['rating'].value_counts() ## Complete the code

Out[9]: Not given    736
        5            588
        4            386
        3            188
        Name: rating, dtype: int64
```

Notes:

- **736** orders were not rated.
- It represents approximately **39%** of total orders.

Univariate Analysis

Univariate analysis is performed on a single variable separately

| Used Histplots, Countplots, and Boxplots |

| Column-wise analysis |

| Question-wise analysis (Q6 – Q11) |

Univariate Analysis (Answers to Q6 – Q11)

Q6 *Order ID / Customer ID / Restaurant name*

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

Order ID

In [10]:
```
# check unique order ID
df['order_id'].nunique()
```

Out[10]: 1898

Customer ID

In [11]:
```
# check unique customer ID
df['customer_id'].nunique()  ## Complete the code to find out number of unique Customer ID
```

Out[11]: 1200

Restaurant name

In [12]:
```
# check unique Restaurant Name
df['restaurant_name'].nunique()  ## Complete the code to find out number of unique Restaurant Name
```

Out[12]: 178

Observations:

- There are a total of **1898 orders** made.
- The orders were placed by **1200 customers**. This means customers have made multiple orders.
- There are a total of **178 different restaurants** towards which the orders were placed.

Univariate Analysis (Answers to Q6 – Q11)

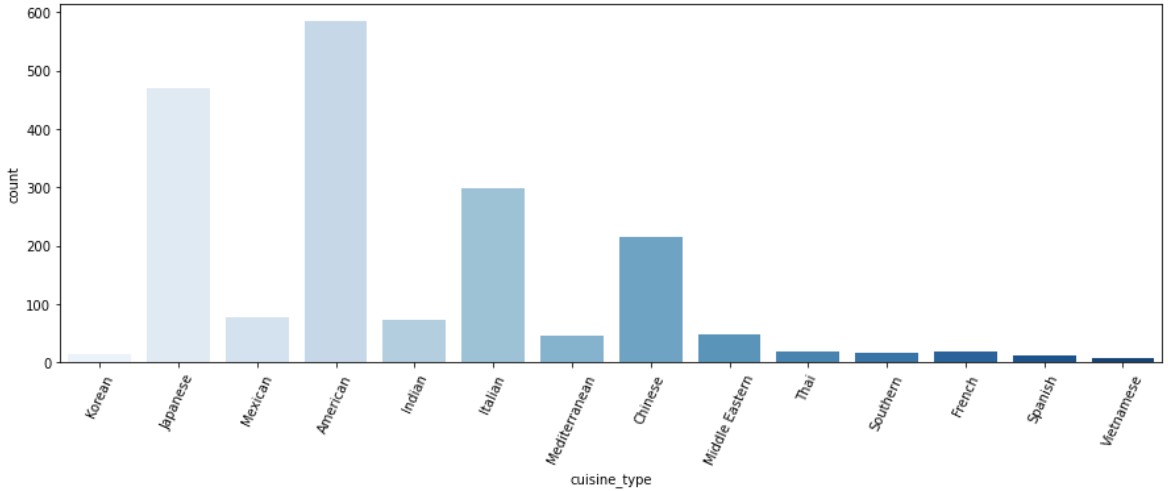
Q6 *Cuisine type*

Cuisine type

In [13]:
```
# Check unique cuisine type
df['cuisine_type'].nunique() ## Complete the code to find out  number of  unique cuisine type
```

Out[13]: 14

In [19]:
```
plt.figure(figsize = (15,5))
sns.countplot(data = df, x = 'cuisine_type',palette='Blues'); ## Create a countplot for cuisine type.
plt.xticks(rotation = 65)
plt.show()
```



Observations:

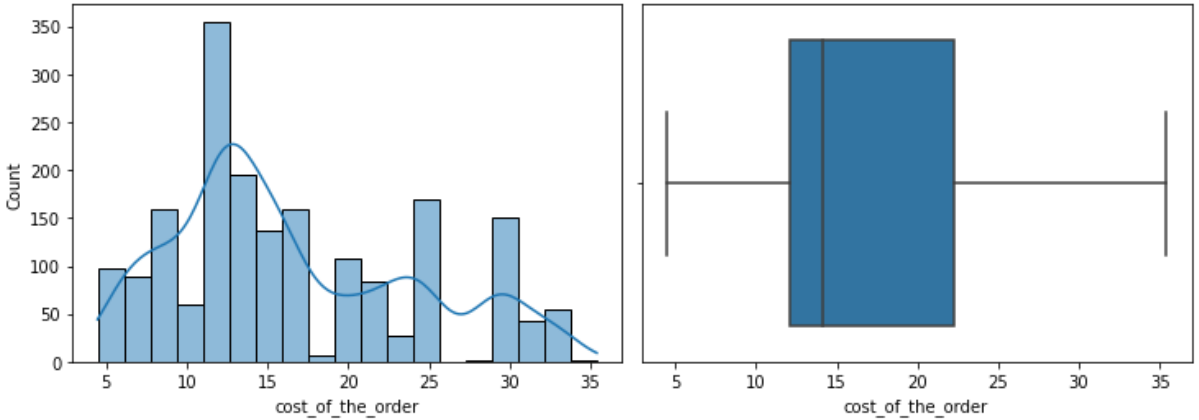
- There are **14 cuisine** types.
- **American cuisine** is the most **frequently ordered** with having 584 orders. It represents 30.76% of total orders.
- **Vietnamese** is the **least frequently** ordered cuisine with only 7 orders.

Univariate Analysis (Answers to Q6 – Q11)

Q6 *Cost of the order*

Cost of the order

In [4]:
```
sns.histplot(data=df,x='cost_of_the_order', kde = True) ## 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()
```



Observations:

- The distribution of cost of order is **right skewed.**
- The cost most charged among the orders is $**12.**
- There are **no outliers** present in the boxplot.
- The cost ranges from **$5 to $35.**
- The **median** cost of order is around **$14**.
- 75% of the cost is below $23.

Univariate Analysis (Answers to Q6 – Q11)

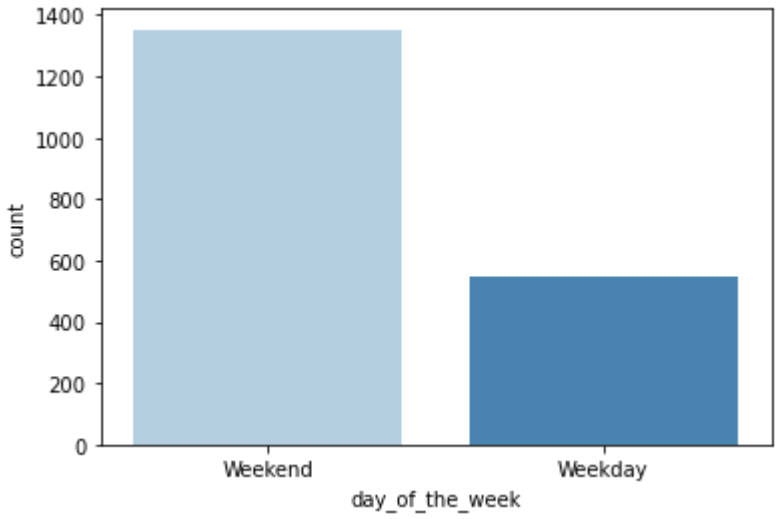
Q6 *Day of the week*

Day of the week

In [19]:
```
# # Check the unique values
df['day_of_the_week'].unique() ## Complete the code to check unique values for the 'day_of_the_week' column
```

Out[19]:
```
array(['Weekend', 'Weekday'], dtype=object)
```

In [20]:
```
sns.countplot(data = df, x = 'day_of_the_week', palette = 'Blues'); ## Complete the code to plot a bar graph for 'day_of_the_
```



Observations:

- The “day of the week” column consists of ‘Weekend’ and ‘Weekday’.
- Approx. **71%** (1351) of the orders are made during the **weekend.**
- Therefore, majority of the orders are made during the weekends.

Univariate Analysis (Answers to Q6 – Q11)

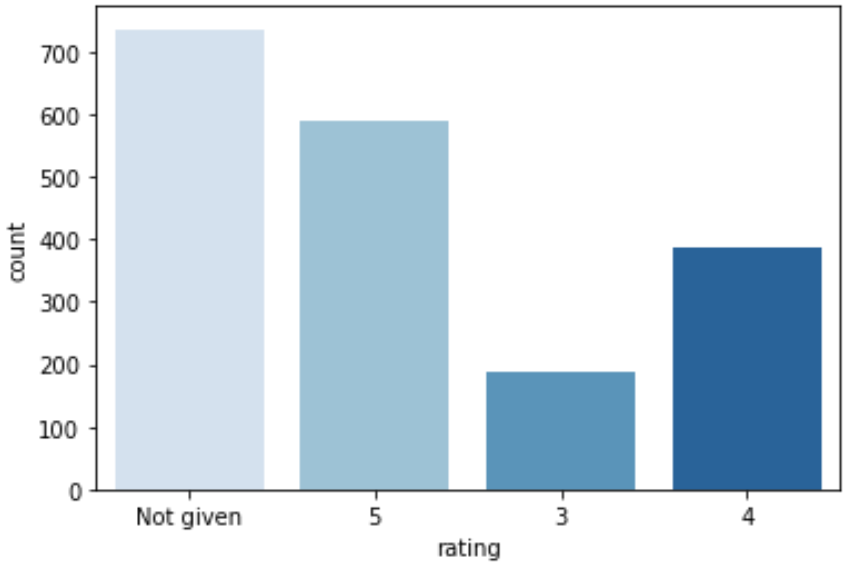
Q6 *Rating*

Rating

In [5]:
```
# Check the unique values
df['rating'].unique() ## Complete the code to check unique values for the 'rating' column
```

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

In [21]:
```
sns.countplot(data = df, x = 'rating', palette = 'Blues'); ## Complete the code to plot bar graph for 'rating' column
```



Observations:

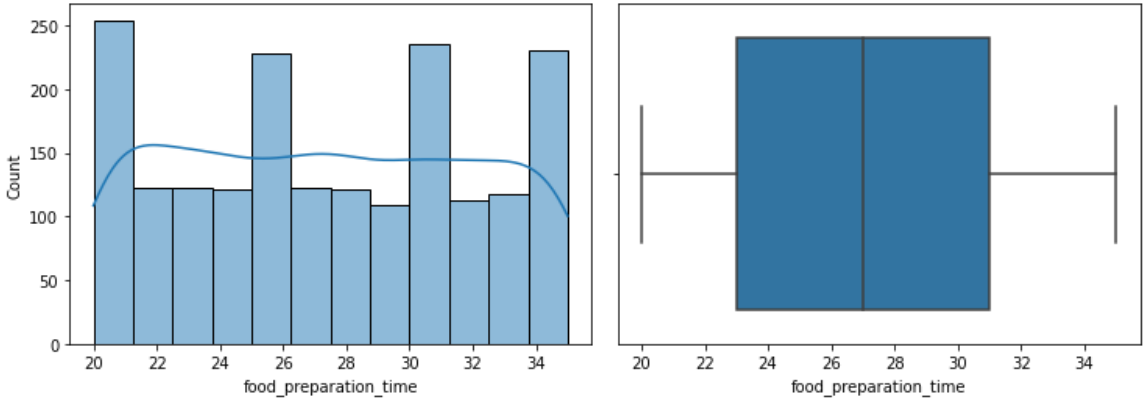
- The ratings for the orders are **out of 5**.
- Customers have provided ratings of **3, 4, and 5**.
- Highest number of orders have a rating of 5.
- Rating of 3 is the least provided rating.
- **736** orders do not have any rating.

Univariate Analysis (Answers to Q6 – Q11)

Q6 *Food Preparation Time*

Food Preparation time

```
In [12]: sns.histplot(data=df,x='food_preparation_time', kde= True) ## Complete the code to plot the histogram for the cost of order
         plt.show()
         sns.boxplot(data=df,x='food_preparation_time') ## Complete the code to plot the boxplot for the cost of order
         plt.show()
```



Observations:

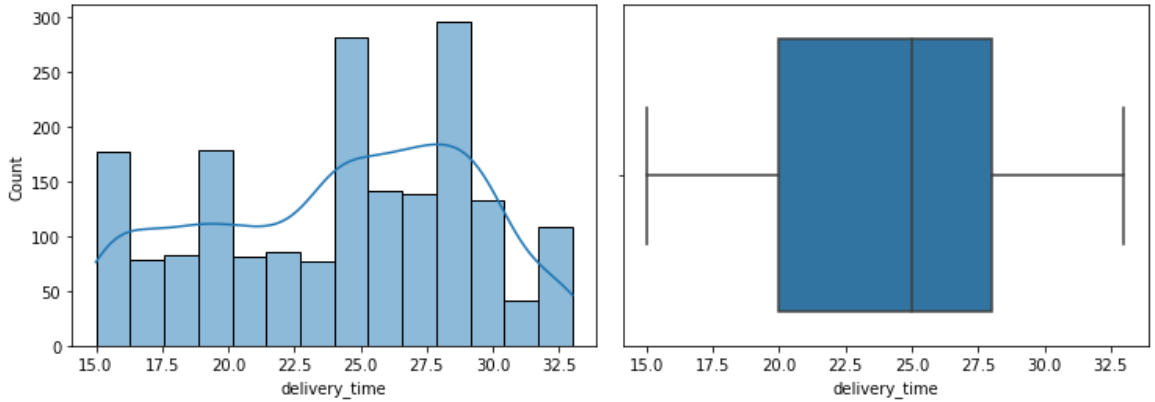
- The food preparation time ranges between **20 - 35 mins**.
- Distribution shows a **multimodal** data set.
- There are **no outliers** present.
- **25%** of the orders take 20 – 23 mins.
- Median time is **27 mins.**

Univariate Analysis (Answers to Q6 – Q11)

Q6 *Delivery time*

Delivery time

```
In [5]: sns.histplot(data=df,x='delivery_time', kde = True) ## Complete the code to plot the histogram for the delivery time
        plt.show()
        sns.boxplot(data=df,x='delivery_time') ## Complete the code to plot the boxplot for the delivery time
        plt.show()
```



Observations:

- The delivery time ranges between **15 mins to 33 mins.**
- The distribution is **slightly left skewed** and represents a **bimodal** data set.
- There are **no outliers** present in the boxplot.
- The median delivery time is **25 mins**.
- 75% of the orders have delivery time below 28 mins.

Univariate Analysis (Answers to Q6 – Q11)

Q7

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

```
In [36]: # Get top 5 restaurants with highest number of orders
         df['restaurant_name'].value_counts().head() ## Complete the code

Out[36]: Shake Shack                  219
         The Meatball Shop            132
         Blue Ribbon Sushi            119
         Blue Ribbon Fried Chicken     96
         Parm                          68
         Name: restaurant_name, dtype: int64
```

Q8

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

```
In [31]: # Get most popular cuisine on weekends
         df_weekend = df[df['day_of_the_week'] == 'Weekend']
         df_weekend['cuisine_type'].value_counts() ## Complete the code to check unique values for th

Out[31]: American          415
         Japanese          335
         Italian           207
         Chinese           163
         Mexican            53
         Indian             49
         Mediterranean      32
         Middle Eastern     32
         Thai               15
         French             13
         Korean             11
         Southern           11
         Spanish            11
         Vietnamese          4
         Name: cuisine_type, dtype: int64
```

Notes:

- **Top 5** restaurants in terms of number of orders:

Restaurant name	Number of orders
Shake Shack	219
The Meatball Shop	132
Blue Ribbon Sushi	119
Blue Ribbon Fried Chicken	96
Parm	68

- Most popular cuisine on weekends : **American (415 orders)**

Univariate Analysis (Answers to Q6 – Q11)

Q9

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

```
In [32]:
# Get orders that cost above 20 dollars
df_greater_than_20 = df[df['cost_of_the_order']>20] ## Write the appropriate column name to get the

# 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 %
```

Q10

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

```
In [33]:
# Get the mean delivery time
mean_del_time = df['delivery_time'].mean()  ## Write the appropriate function to obtain the m

print('The mean delivery time for this dataset is', round(mean_del_time, 2), 'minutes')
```

```
The mean delivery time for this dataset is 24.16 minutes
```

Q11

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]

```
In [37]:
# Get the counts of  each customer_id
df['customer_id'].value_counts().head(3)  ## Write the appropriate column name to get the top 5 cmost frequent customers
```

```
Out[37]: 52832    13
         47440    10
         83287     9
         Name: customer_id, dtype: int64
```

Notes:

- **29.24%** (555 orders) of the total orders cost more than $20.
- The mean delivery time is **24.16 mins**.
- The **top 3** most frequent customers:

Customer ID	Number of orders
52832	13
47440	10
83287	9

Multivariate Analysis

Multivariate analysis is done to analyze between 2 or more variables

| Used Boxplots, Pointplots, Heatmap |
| Analyzed between multiple columns |
| Correlation between variables |
| Question-wise analysis (Q12 – Q16) |

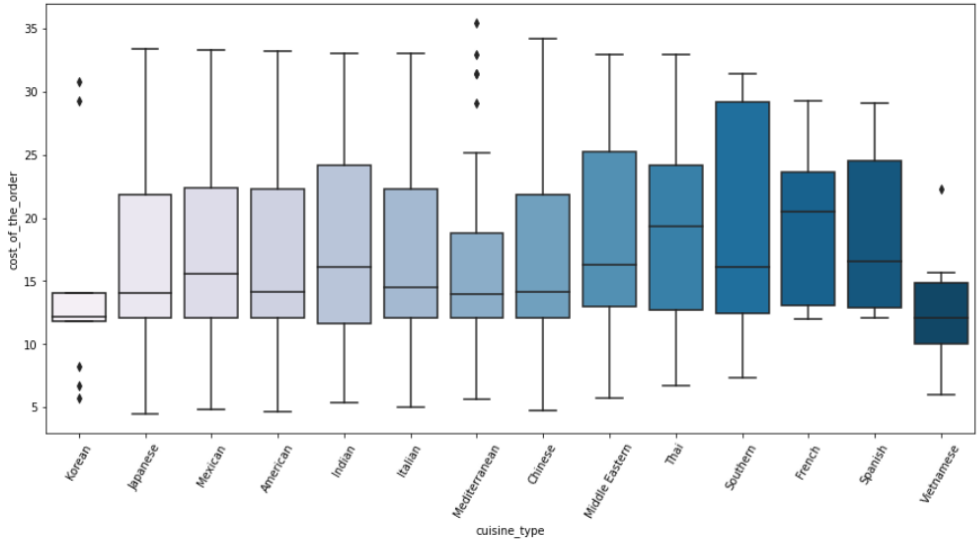
Multivariate Analysis (Answers to Q12 – Q16)

Q12 *Cuisine vs Cost of the order*

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]

Cuisine vs Cost of the order

In [38]:
```
# 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')
plt.xticks(rotation = 60)
plt.show()
```



Observations:

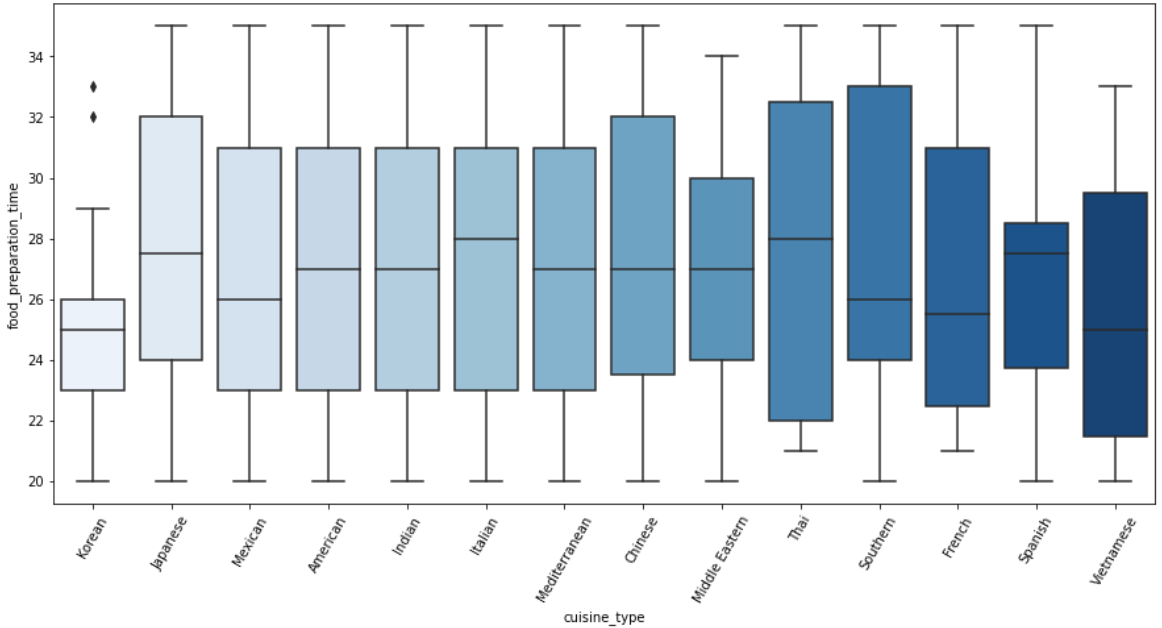
- The **range of cost** is equal in **Japanese, Mexican, and American cuisine.**
- The **median cost** is same for **Japanese, American, and Chinese cuisine.**
- **50%** of orders of all the cuisines is above $10.
- **75%** of orders of American cuisine is below $25.
- There are **no whiskers for Korean** cuisine as the lower quartile is equal to the minimum cost and the upper quartile is equal to the maximum cost.
- There are **outliers for Korean, Mediterranean, and Vietnamese** cuisines.
- The range of cost is lowest for **Vietnamese cuisine.**

Multivariate Analysis (Answers to Q12 – Q16)

Q12 *Cuisine vs Food Preparation Time*

Cuisine vs Food Preparation time

In [14]:
```
# Relationship between food preparation time and cuisine type
plt.figure(figsize=(15,7))
sns.boxplot(data = df, x = 'cuisine_type', y = 'food_preparation_time', palette = 'Blues')  ## Complete the code to visualize
plt.xticks(rotation = 60)
plt.show()
```



Observations:

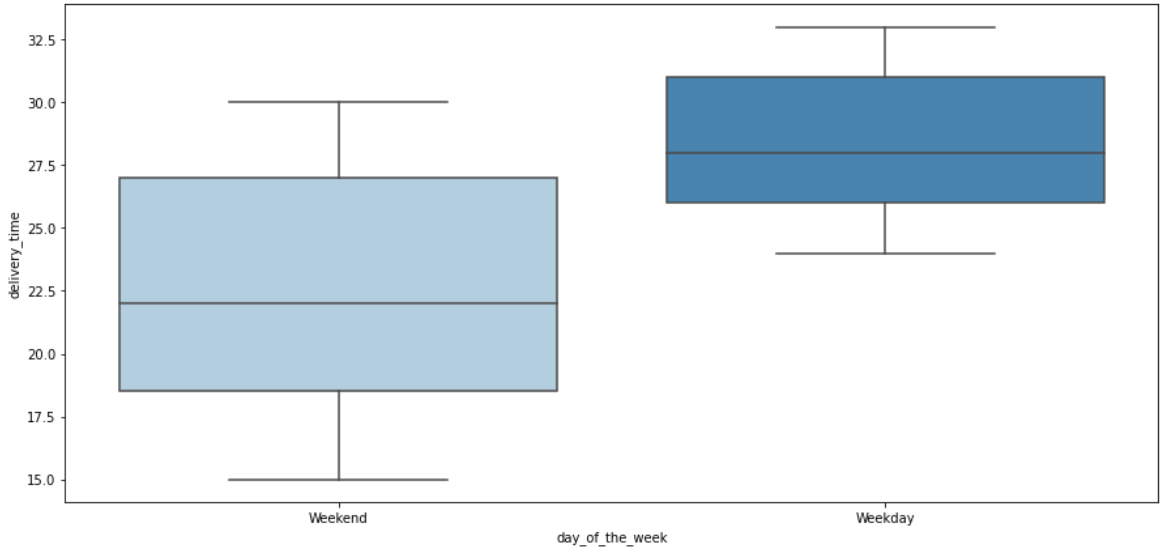
- There are **no significant variations** between food preparation time for most of the cuisines.
- **Korean** cuisine has the **lowest** food preparation time.
- The distribution of food preparation time is equal between Mexican, American, Indian, Italian, and Mediterranean cuisines.
- The **median time is equal** for American, Indian, Mediterranean, Chinese, and Middle Eastern.
- There are **outliers for Korean** cuisine.
- **Highest median**: Italian & Thai cuisine
- **Lowest median:** Vietnamese cuisine

Multivariate Analysis (Answers to Q12 – Q16)

Q12 *Day of the Week vs Delivery time*

Day of the Week vs Delivery time

In [11]:
```
# Relationship between day of the week and delivery time
plt.figure(figsize=(15,7))
sns.boxplot(data = df,x = 'day_of_the_week', y = 'delivery_time', palette = 'Blues')  ## Complete the code to visualize the r
plt.show()
```



Observations:

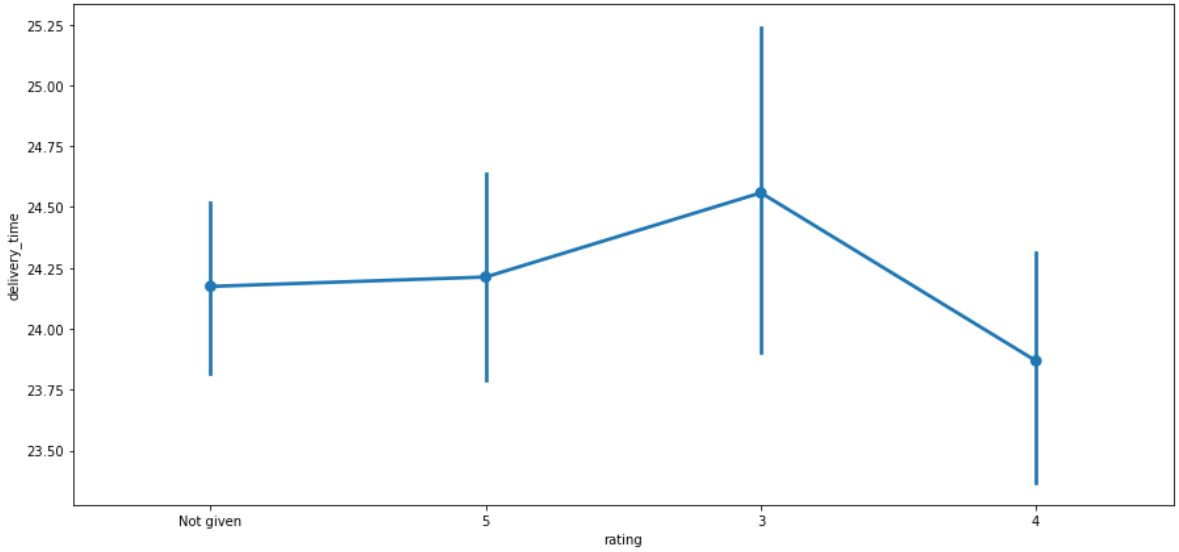
- The **delivery time** is **higher** during the **weekdays** than the weekends.
- This could be due to the **increased traffic** from office-goers and school buses during the weekdays.
- Delivery time could go **beyond 30 mins during the weekday**.

Multivariate Analysis (Answers to Q12 – Q16)

Q12 *Rating vs Delivery time*

Rating vs Delivery time

In [43]:
```
# Relationship between rating and delivery time
plt.figure(figsize=(15, 7))
sns.pointplot(x = 'rating', y = 'delivery_time', data = df)
plt.show()
```



Observations:

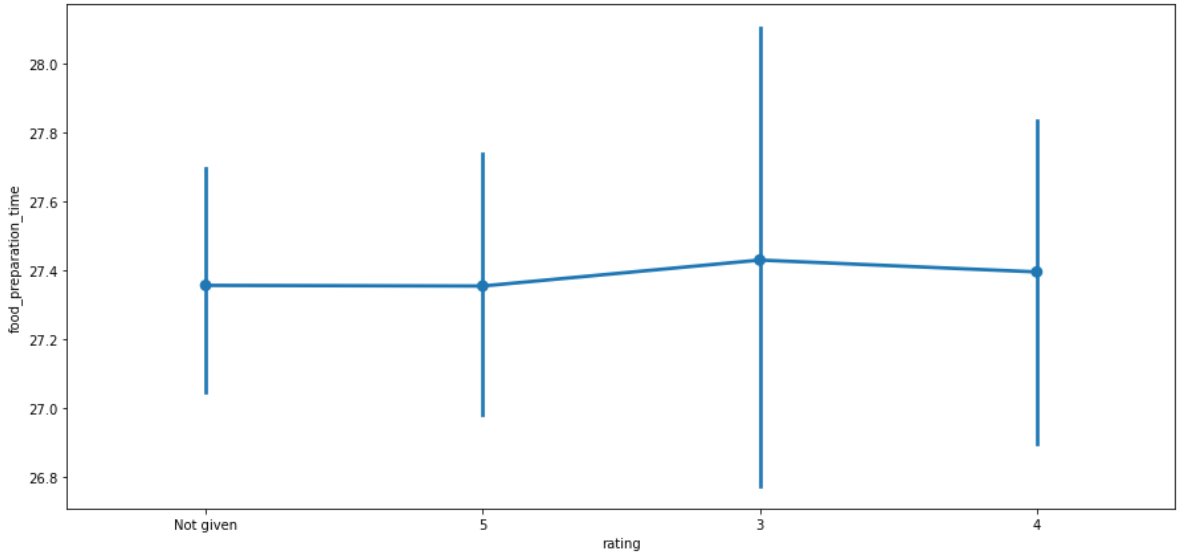
- There is **great variance** in the delivery time for orders with the **rating of 3**.
- Orders rated **4 have lowest** delivery time.
- Orders rated **5** have a delivery time ranging between **23.75 mins and 24.75 mins**.
- The **delivery time has an impact on the rating** of the orders.

Multivariate Analysis (Answers to Q12 – Q16)

Q12 *Rating vs Food preparation time*

Rating vs Food preparation time

```
In [44]: # Relationship between rating and food preparation time
         plt.figure(figsize=(15, 7))
         sns.pointplot(data = df, x = 'rating', y = 'food_preparation_time')  ## Complete the code to visualize the relationship betwe
         plt.show()
```



Observations:

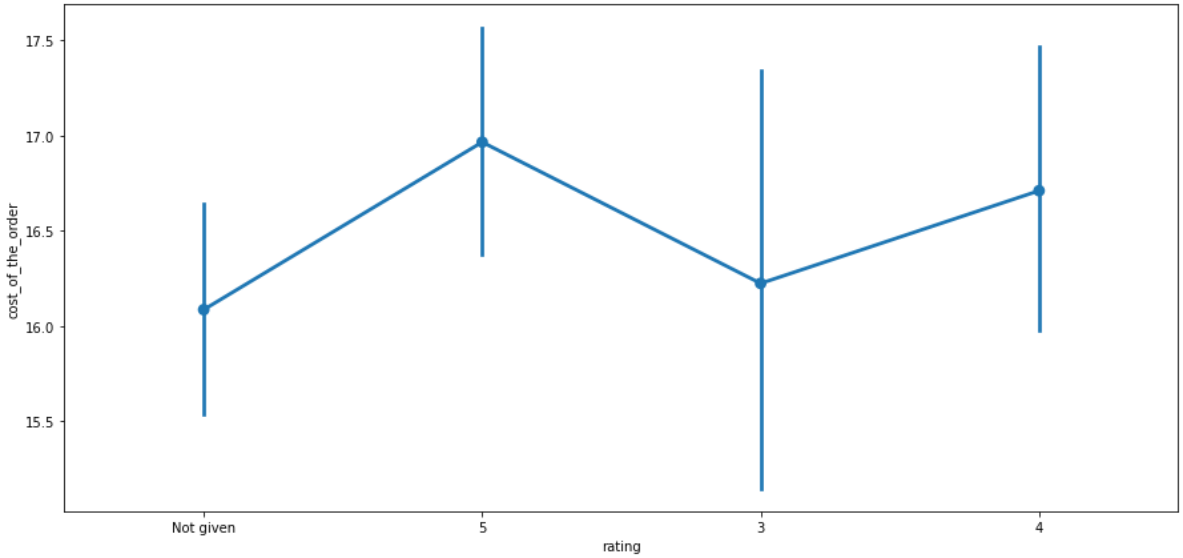
- The **mean** food preparation time is **almost same** for all ratings.
- Orders **rated 3 have significant variance** in the food preparation time.
- The **variance decreases as the rating increases**.
- It could be due to the food preparation time having some **impact on the delivery time**.
- This in turn **impacts the rating** of the order.
- However, the **average** food preparation time is almost equal for all the ratings

Multivariate Analysis (Answers to Q12 – Q16)

Q12 *Rating vs Cost of the order*

Rating vs Cost of the order

In [45]:
```
# Relationship between rating and cost of the order
plt.figure(figsize=(15, 7))
sns.pointplot(data = df, x = 'rating', y = 'cost_of_the_order')   ## Complete the code to visualize the relationship between
plt.show()
```



Observations:

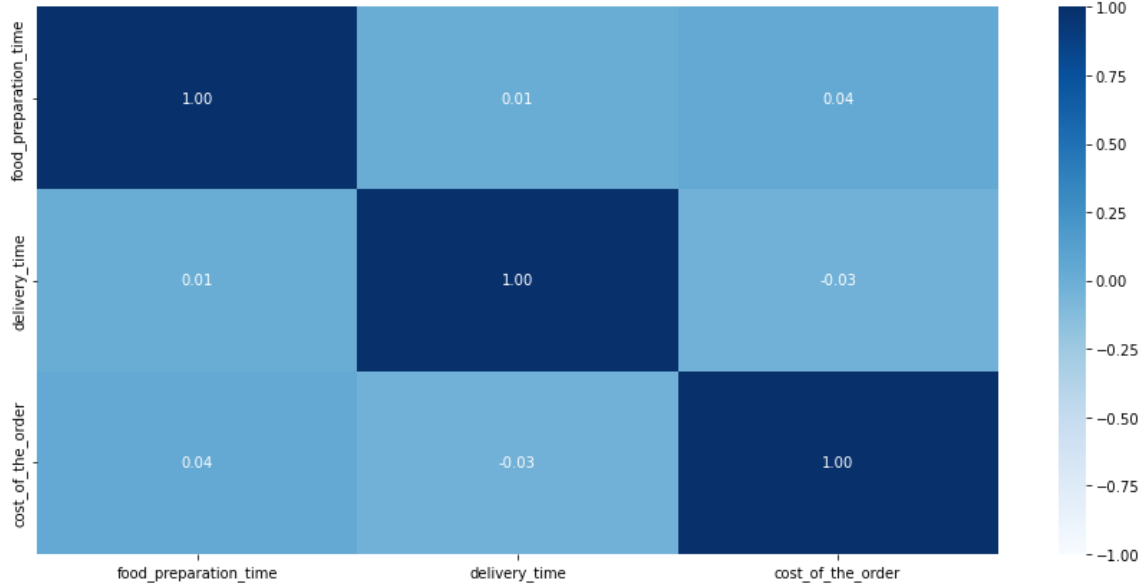
- Orders with **higher costs have higher rating.**
- Orders rated **3 have greater variance** in their cost.
- Most **popular cuisines have higher cost** and hence the ratings would also be higher for those orders.

Multivariate Analysis (Answers to Q12 – Q16)

Q12 *Correlation among variables*

Correlation among variables

In [9]:
```
# Plot the heatmap
col_list = ['food_preparation_time', 'delivery_time','cost_of_the_order']
plt.figure(figsize=(15, 7))
sns.heatmap(df[col_list].corr(), annot=True, vmin=-1, vmax=1, fmt=".2f", cmap="Blues")
plt.show()
```



Observations:

- Cost of order has a **weak negative correlation** with delivery time.
- Cost of order has a **weak positive correlation** with food preparation time.
- Food preparation time has a **weak positive correlation** with delivery time.
- The cost is not impacted much by the food preparation time or delivery time.

Multivariate Analysis (Answers to Q12 – Q16)

Q12

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

```
In [42]: df.groupby(['restaurant_name'])['cost_of_the_order'].sum().sort_values(ascending = False).head(14)

Out[42]: 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
```

Notes:

- **Shake Shack generates the highest revenue**. It is the top restaurant in number of orders (refer question 7).
- **Five Guys Burgers and Fries generates the lowest revenue.**
- The top 5 restaurants in number of orders are the top 5 with highest revenue generated.

Multivariate Analysis (Answers to Q12 – Q16)

Q13

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]

In [47]:
```
# 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()
```

Out[47]:

	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

In [48]:
```
# Get the restaurant names that have rating count more than 50
rest_names = df_rating_count[df_rating_count['rating']>50]['restaurant_name'] ## Complete the code to get the restaurant name

# 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_names)].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() ## Complete the
```

Out[48]:

	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

Notes:

- The restaurants with **rating count of more than 50** are:

Restaurant name	Rating count
Shake Shack	133
The Meatball Shop	84
Blue Ribbon Sushi	73
Blue Ribbon Fried Chicken	64
Parm	41

- **All the above restaurants have an average rating of more than 4.**
- The promotional offer can be provided to these restaurants.

Multivariate Analysis (Answers to Q12 – Q16)

Q14

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]

In [49]:
```
#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) ## Write the apprpriate column name to compute the revenue
df.head()
```

Out[49]:

	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

In [52]:
```
# get the total revenue and print it
total_rev = df['Revenue'].sum() ## Write the appropriate function to get the total revenue
print('The net revenue is around', round(total_rev, 2), 'dollars')
```

```
The net revenue is around 6166.3 dollars
```

Notes:

- FoodHub has generated a total revenue of **$6166.3**.
- This is taken from orders costing more than $20 and $5 by charging **25% and 15%** respectively.
- Therefore, their **revenue is dependent on the cost** of the orders.

Multivariate Analysis (Answers to Q12 – Q16)

Q15

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]

In [6]:

```
# 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
orders = df[df['total_time']>60]
perc_of_orders = orders.shape[0]/df.shape[0] * 100
print('Percentage of orders taking more 60 mins to get delivered : ',perc_of_orders,"%")
```

```
Percentage of orders taking more 60 mins to get delivered :  10.537407797681771 %
```

Q16

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]

In [7]:

```
# 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 weekdends 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 weekdends is around 22 minutes
```

Notes:

- **10.5% of the orders take more than 60 mins** to get the food delivered from the time the order is placed.
- The **mean delivery time on weekdays is higher** than that on the weekends.
- There is a **difference of 6 mins** between the two.

Great Learning

Happy Learning !