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 a Data Scientist at Foodhub 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_of_the_order: 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

Understanding the structure of the data

In [29]:

Out[29]:

	order_id	customer_id	restaurant_name	cuisine_type	cost_of_the_order	day_of_the_week	rating	food_preparati
0	1477147	337525	Hangawi	Korean	30.75	Weekend	Not given	
1	1477685	358141	Blue Ribbon Sushi Izakaya	Japanese	12.08	Weekend	Not given	
2	1477070	66393	Cafe Habana	Mexican	12.23	Weekday	5	
3	1477334	106968	Blue Ribbon Fried Chicken	American	29.20	Weekend	3	
4	1478249	76942	Dirty Bird to Go	American	11.59	Weekday	4	
5	1477224	147468	Tamarind TriBeCa	Indian	25.22	Weekday	3	
6	1477894	157711	The Meatball Shop	Italian	6.07	Weekend	Not given	
7	1477859	89574	Barbounia	Mediterranean	5.97	Weekday	3	
8	1477174	121706	Anjappar Chettinad	Indian	16.44	Weekday	5	
9	1477311	39705	Bukhara Grill	Indian	7.18	Weekday	5	

Observation -View of the first few rows of the dataset -The dataset looks clean and consistent with the description provided in the Data Dictionary -The restaurant name, cusine type and day of the week are categorical variables. -The other variables are numerical -The rating is based on a maximum of 5 and minimum of 1 -Not all are rated

In [ ]:

```
df.shape
```

-We have 1,898 rows and 11 columns

In [3]:

```
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 -All columns have 1,898 observations indicating that there are no missing values in it -restaurant name, cusine type and day of the week should be categorical variables -rating is an object data type -The 'info()' function is used to print a concise summary of the DataFrame

In [4]:

```
df.restaurant_name = df.restaurant_name.astype('category')
df.cuisine_type = df.cuisine_type.astype('category')
df.day_of_the_week = df.day_of_the_week.astype('category')
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   category
 3   cuisine_type           1898 non-null   category
 4   cost_of_the_order      1898 non-null   float64
 5   day_of_the_week        1898 non-null   category
 6   rating                 1898 non-null   object
 7   food_preparation_time  1898 non-null   int64
 8   delivery_time          1898 non-null   int64
dtypes: category(3), float64(1), int64(4), object(1)
memory usage: 102.7+ KB
```

Observation -Converting "objects" to "category" reduces the data space required to store the dataframe -Every class in the categorical column will be represented by a number -It makes building models eaisier -There are 5 numeric columns, 3 categorical columns and 'rating' column

Statistical summary of the data

In [21]:

```
df.describe().T
```

Out[21]:

	count	mean	std	min	25%	50%	75%	
order_id	1898.0	1.477496e+06	548.049724	1476547.00	1477021.25	1477495.50	1.477970e+06	1478
customer_id	1898.0	1.711685e+05	113698.139743	1311.00	77787.75	128600.00	2.705250e+05	405
cost_of_the_order	1898.0	1.649885e+01	7.483812	4.47	12.08	14.14	2.229750e+01	
food_preparation_time	1898.0	2.737197e+01	4.632481	20.00	23.00	27.00	3.100000e+01	
delivery_time	1898.0	2.416175e+01	4.972637	15.00	20.00	25.00	2.800000e+01	

Observation

-The mean food preparation time and delivery time is close to the 50% percentile of the data, indicating a consistency -On the contrary there is a large gap between the minimum and maximum cost of order, indicating a right (positive) skew

In [40]:

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

Out[40]:

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

-A little bit less than half of the the sum of rating was not given -Therefore we will proceed with caution in sampling this to avoid very high or low values

Exploratory Data Analysis (EDA)

Univariate Analysis

-These involve just one variable from our data frame, so we will be using histograms, boxplots and countplots for this

Cuisine type

In [11]:

```
df['cuisine_type'].unique()
```

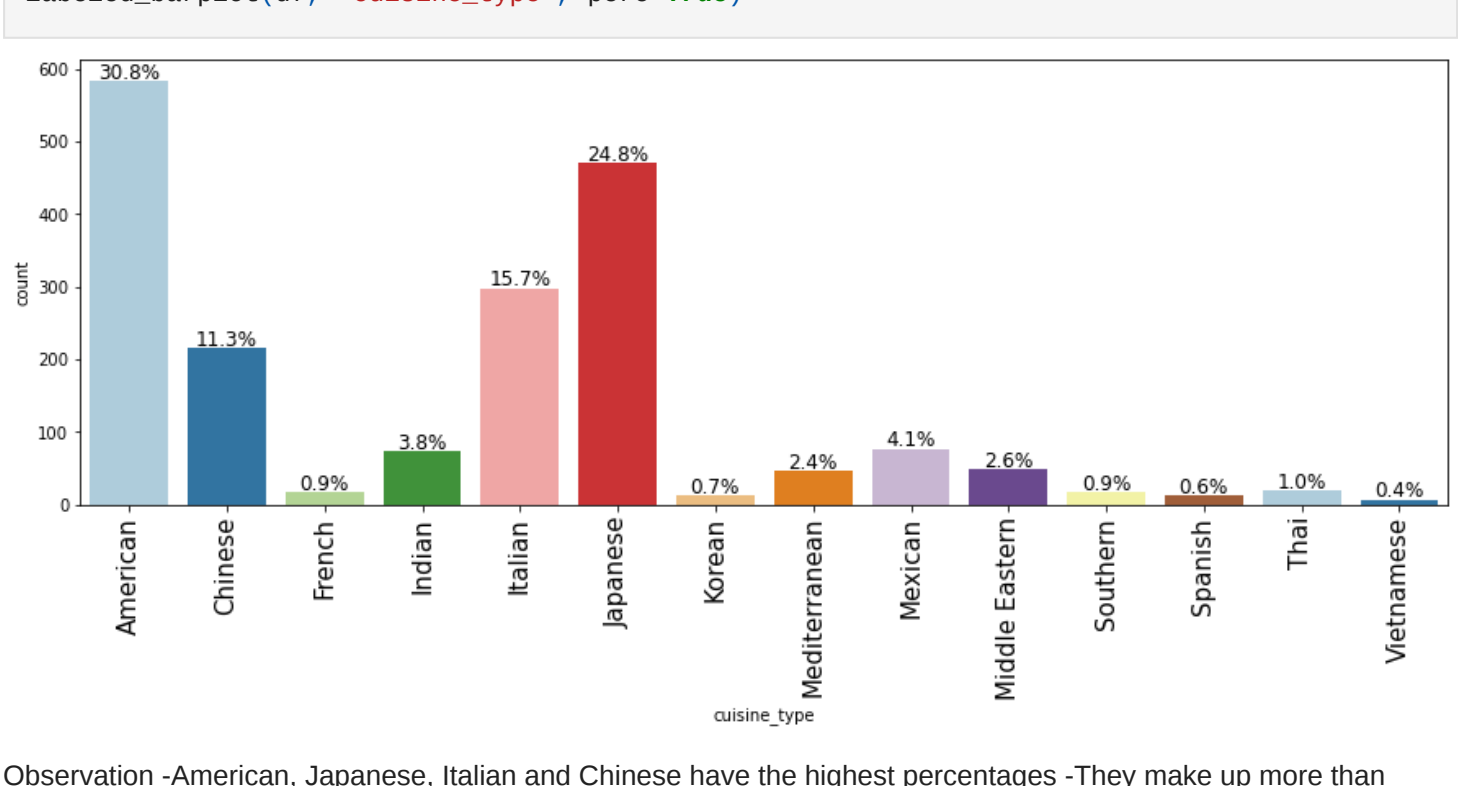
Out[11]:

```
array(['Korean', 'Japanese', 'Mexican', 'American', 'Indian', 'Italian',
       'Mediterranean', 'Chinese', 'Middle Eastern', 'Thai', 'Southern',
       'French', 'Spanish', 'Vietnamese'], dtype=object)
```

-We have 14 different kinds of cusines

In [42]:

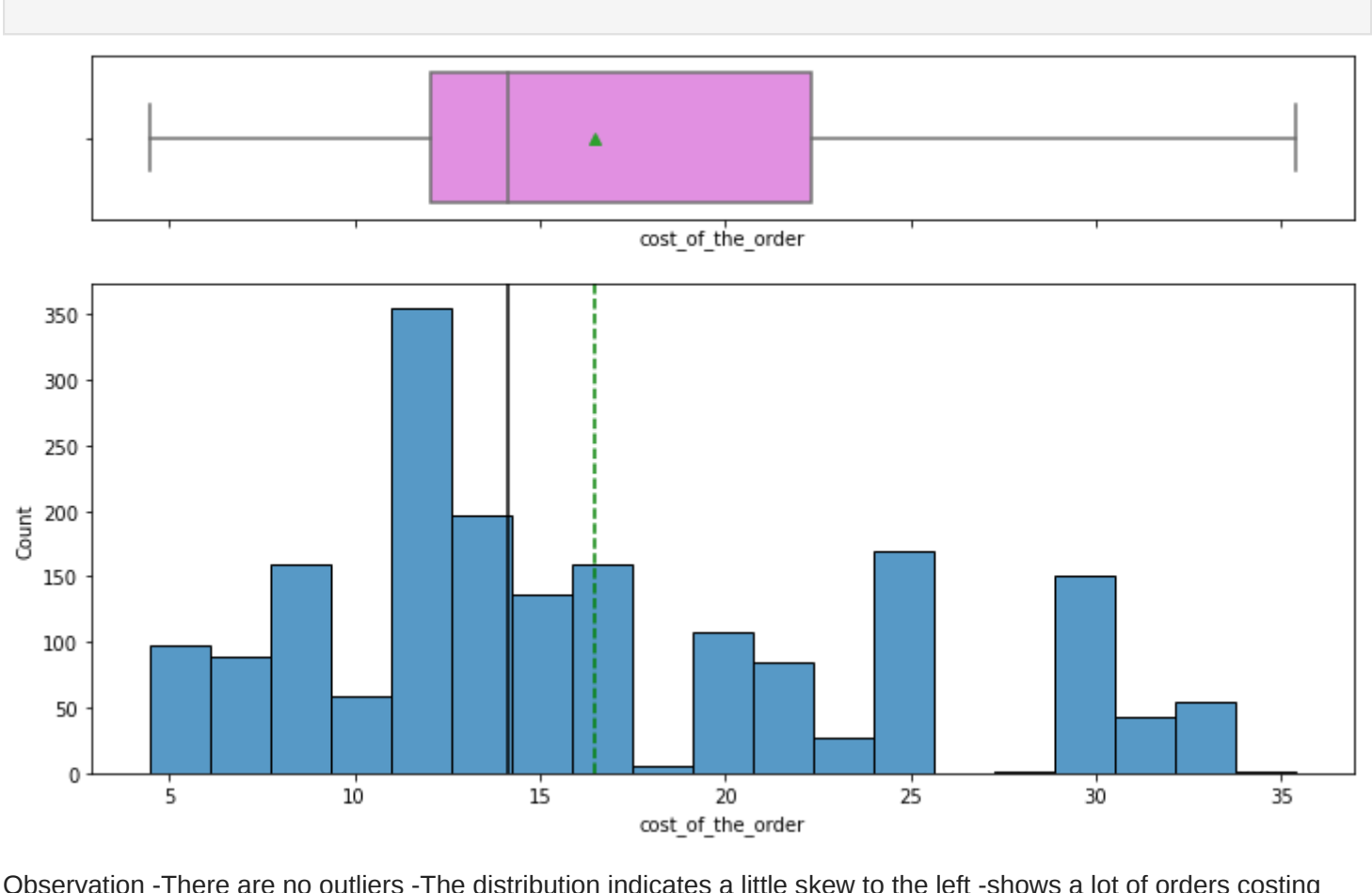
```
labeled_barplot(df, 'cuisine_type', perc=True)
```



Observation -American, Japanese, Italian and Chinese have the highest percentages -They make up more than 80% of the given statistics

Cost of the order

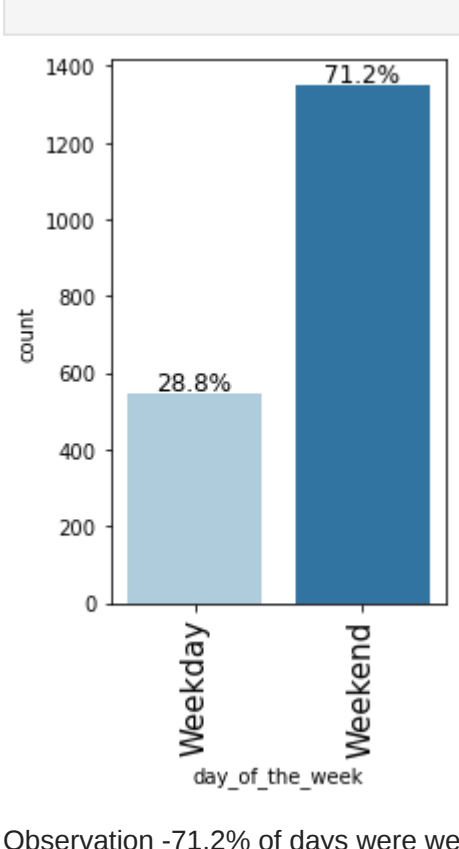
In [43]:



Observation -There are no outliers -The distribution indicates a little skew to the left -shows a lot of orders costing $11

Day of the week

In [46]:

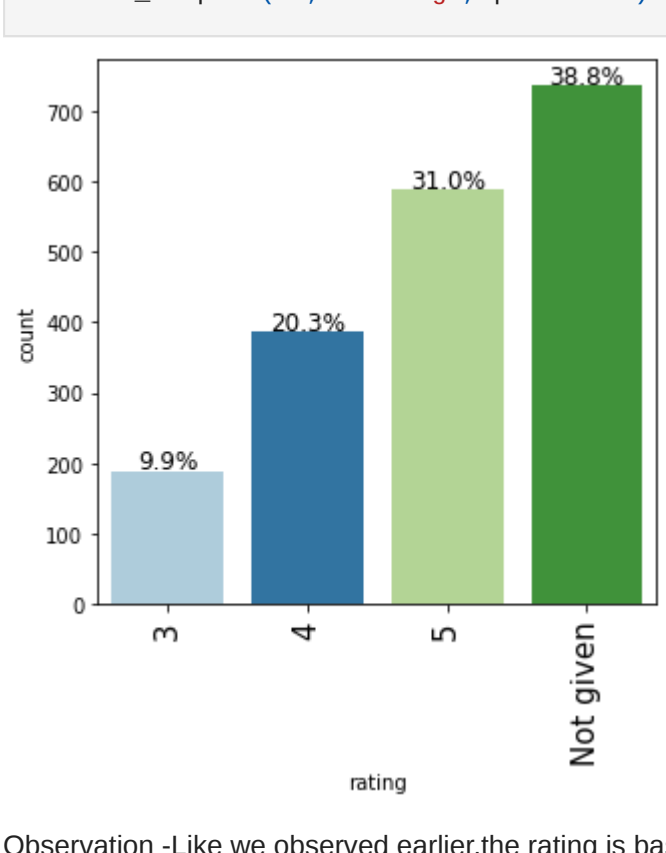


Observation -71.2% of days were weekends in the data that we are analyzing

Rating

In [50]:

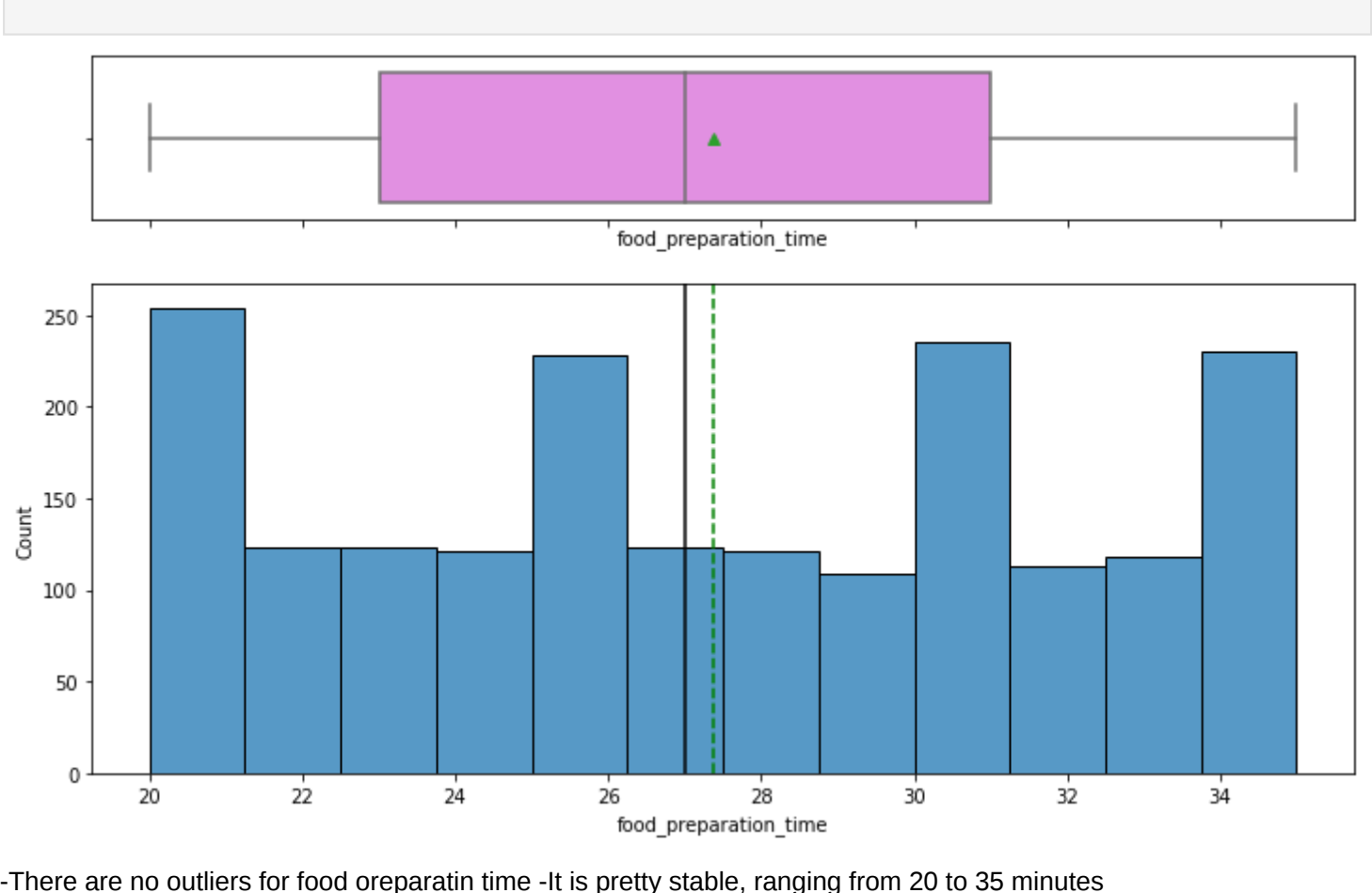
```
labeled_barplot(df, 'rating', perc=True)
```



Observation -Like we observed earlier,the rating is based on a maximum of 5 and minimum of 1 -Not all are rate -Around have of the ratings given are the maximum number 5

Food Preparation time

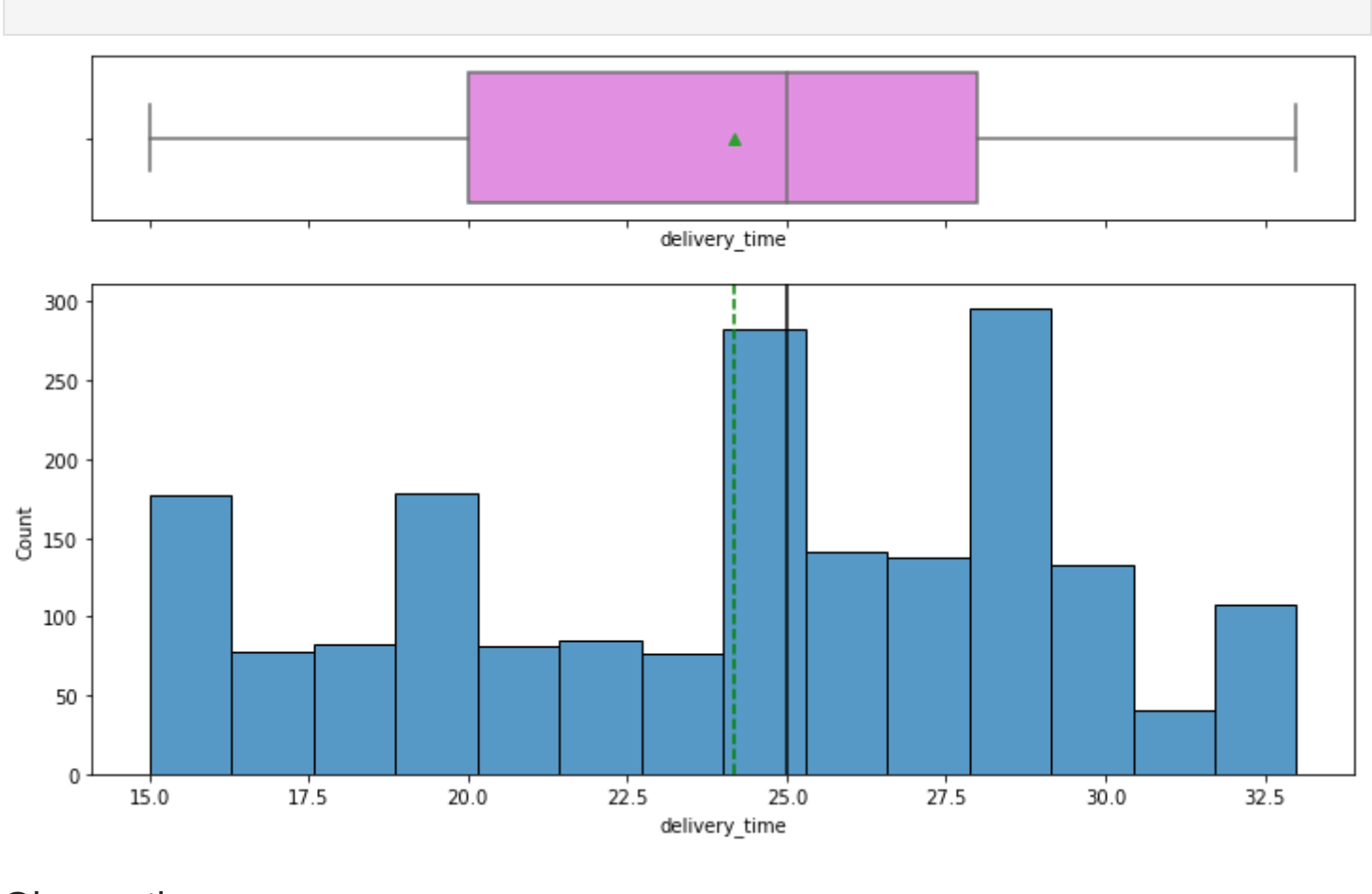
In [51]:



Observation -There are no outliers for food oreparatin time -It is pretty stable, ranging from 20 to 35 minutes

Delivery time

In [52]:



Observation

-There are no outliers for delivery time either -The distribution is similar to that of food preparation time -With the highest points around 24 and 28 minutes

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

In [53]:

Out[53]:

```
Shake Shack                  219
The Meatball Shop            132
Blue Ribbon Sushi            119
Blue Ribbon Fried Chicken     96
Parm                          68
                            ...
Klong                          1
Kambi Ramen House              1
Il Bambino                     1
Hunan Manor                    1
Lamarca Pasta                  1
Name: restaurant_name, Length: 178, dtype: int64
```

The most popular cuisine on weekends

In [19]:

Out[19]:

```
array(['Korean', 'Japanese', 'American', 'Italian', 'Mexican',
       'Mediterranean', 'Chinese', 'Indian', 'Thai', 'Southern', 'French',
       'Spanish', 'Middle Eastern', 'Vietnamese'], dtype=object)
```

Percentage of the orders cost more than 20 dollars

In [55]:

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

The mean order delivery time

In [56]:

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

The company has decided to give 20% discount vouchers to the top 5 most frequent customers. The IDs of these customers and the number of orders they placed are:

In [57]:

Out[57]:

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

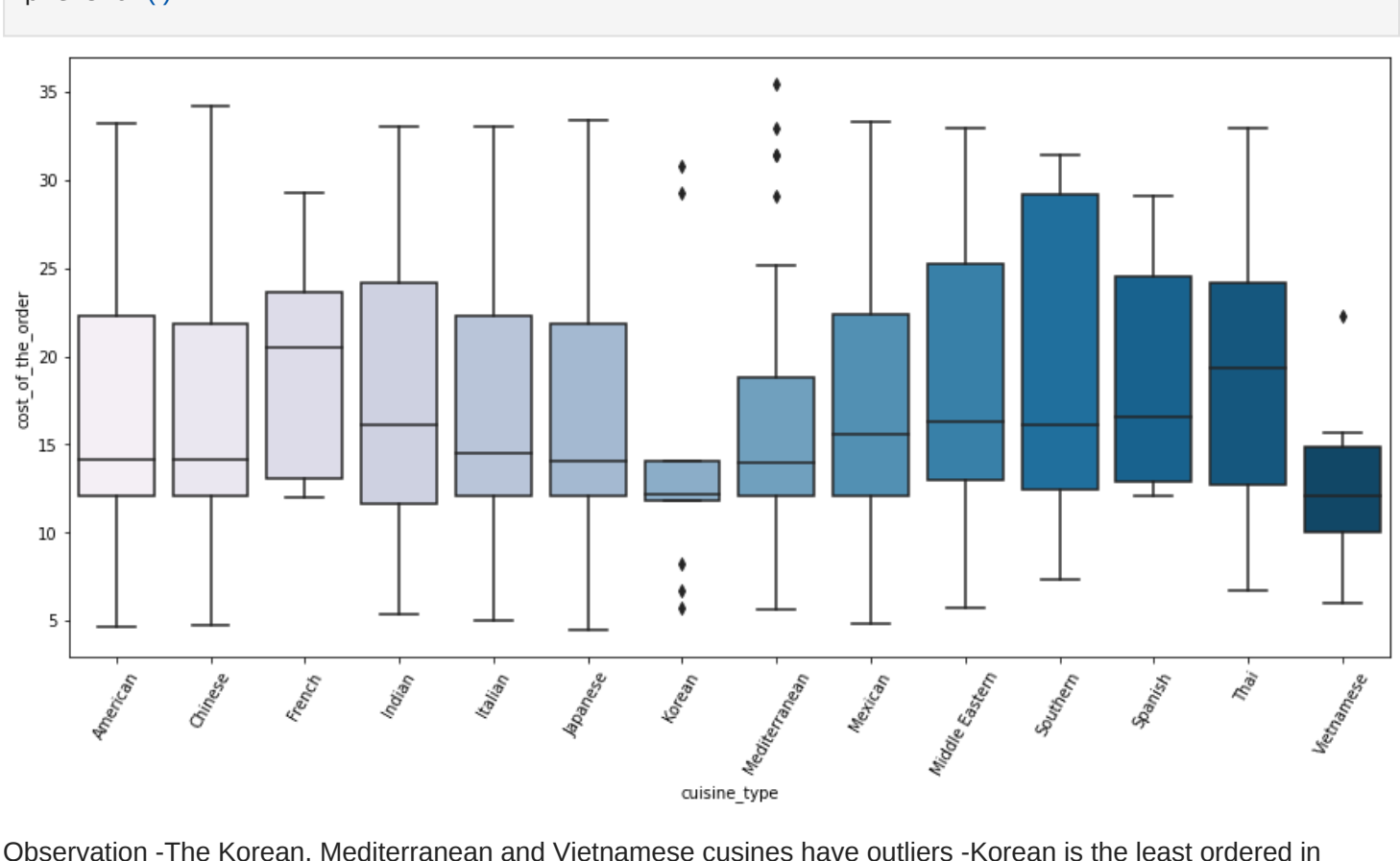
Multivariate Analysis

Performing a multivariate analysis to explore relationships between the important variables in the dataset.

Cuisine vs Cost of the order

In [58]:

```
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()
```

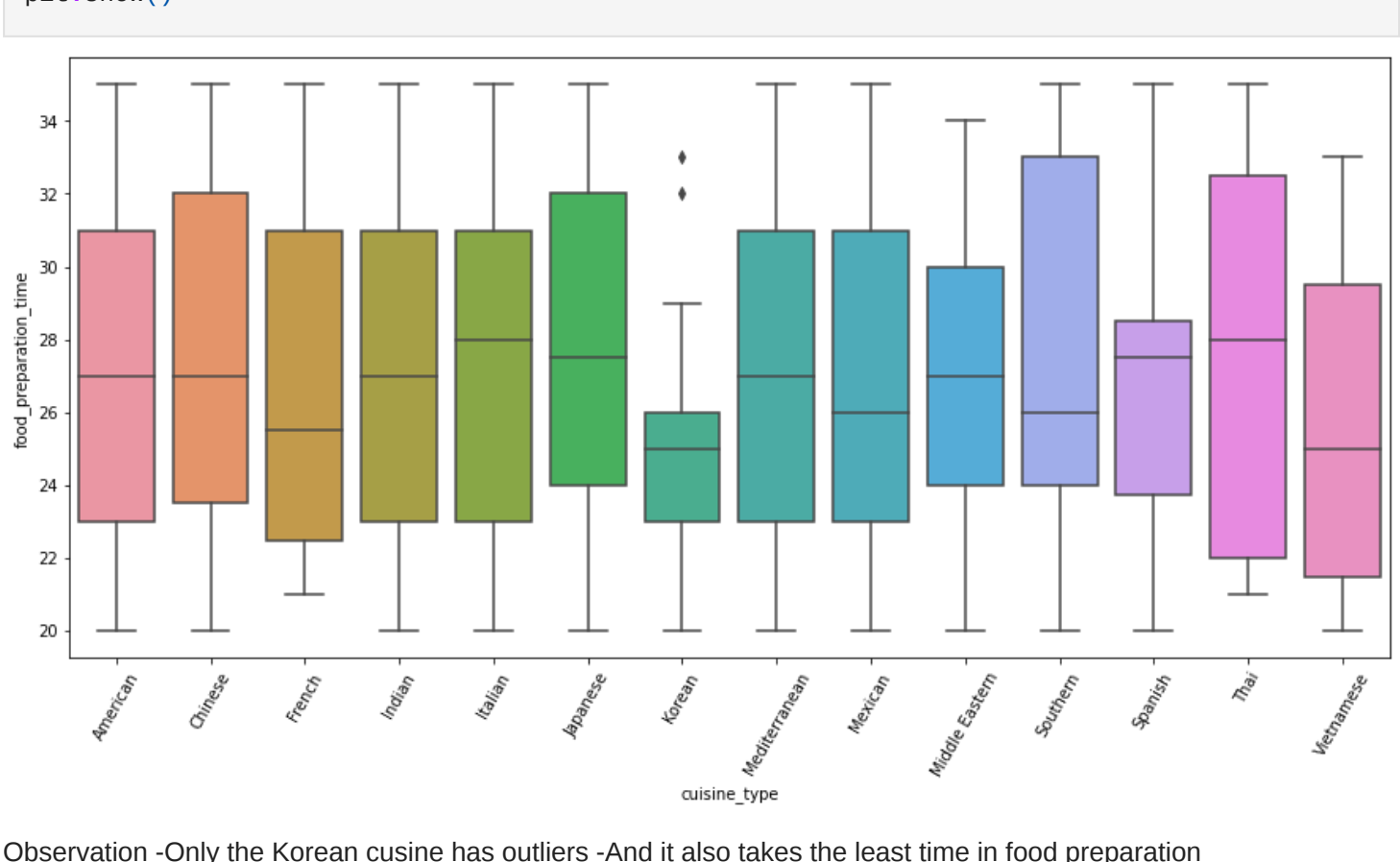


Observation -The Korean, Mediterranean and Vietnamese cusines have outliers -Korean is the least ordered in relation to the cost of order

Cuisine vs Food Preparation time

In [59]:

```
plt.figure(figsize=(15,7))
sns.boxplot(x = "cuisine_type", y = "food_preparation_time", data = df)
plt.show()
```

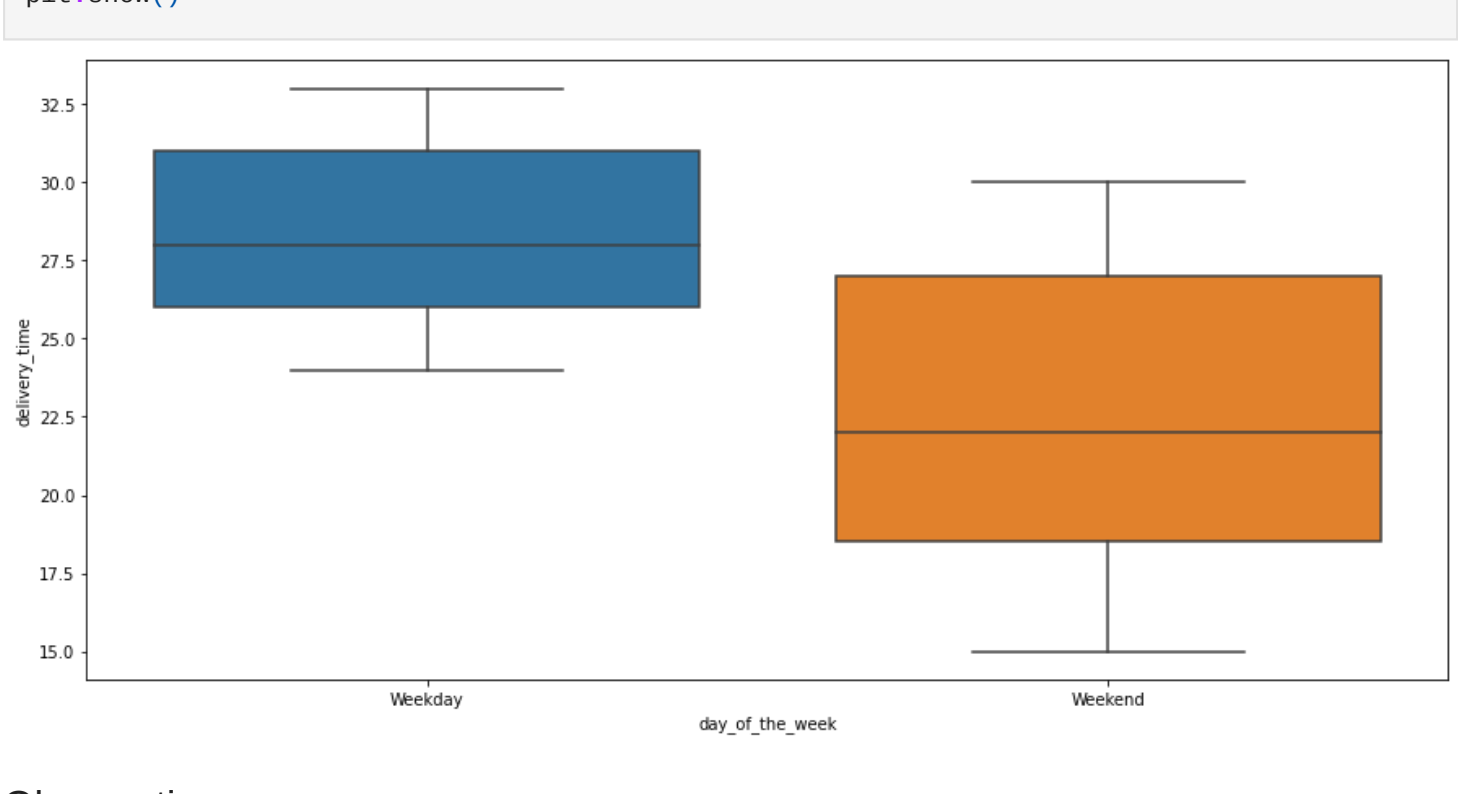


Observation -Only the Korean cusine has outliers -And it also takes the least time in food preparation

Day of the Week vs Delivery time

In [60]:

```
plt.figure(figsize=(15,7))
sns.boxplot(x = "day_of_the_week", y = "delivery_time", data = df)
plt.show()
```



Observation

Observation -There are no outlers -The weekdays have a higher delivery time than the weekends -It might be attributed to the fact that there will be more commuters and road users causing delay on a working day -And more orders coming in

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

In [25]:

```
df.groupby(['restaurant_name'])['cost_of_the_order'].sum().sort_values(ascending = False).he
```

Out[25]:

```
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
Name: cost_of_the_order, dtype: float64
```

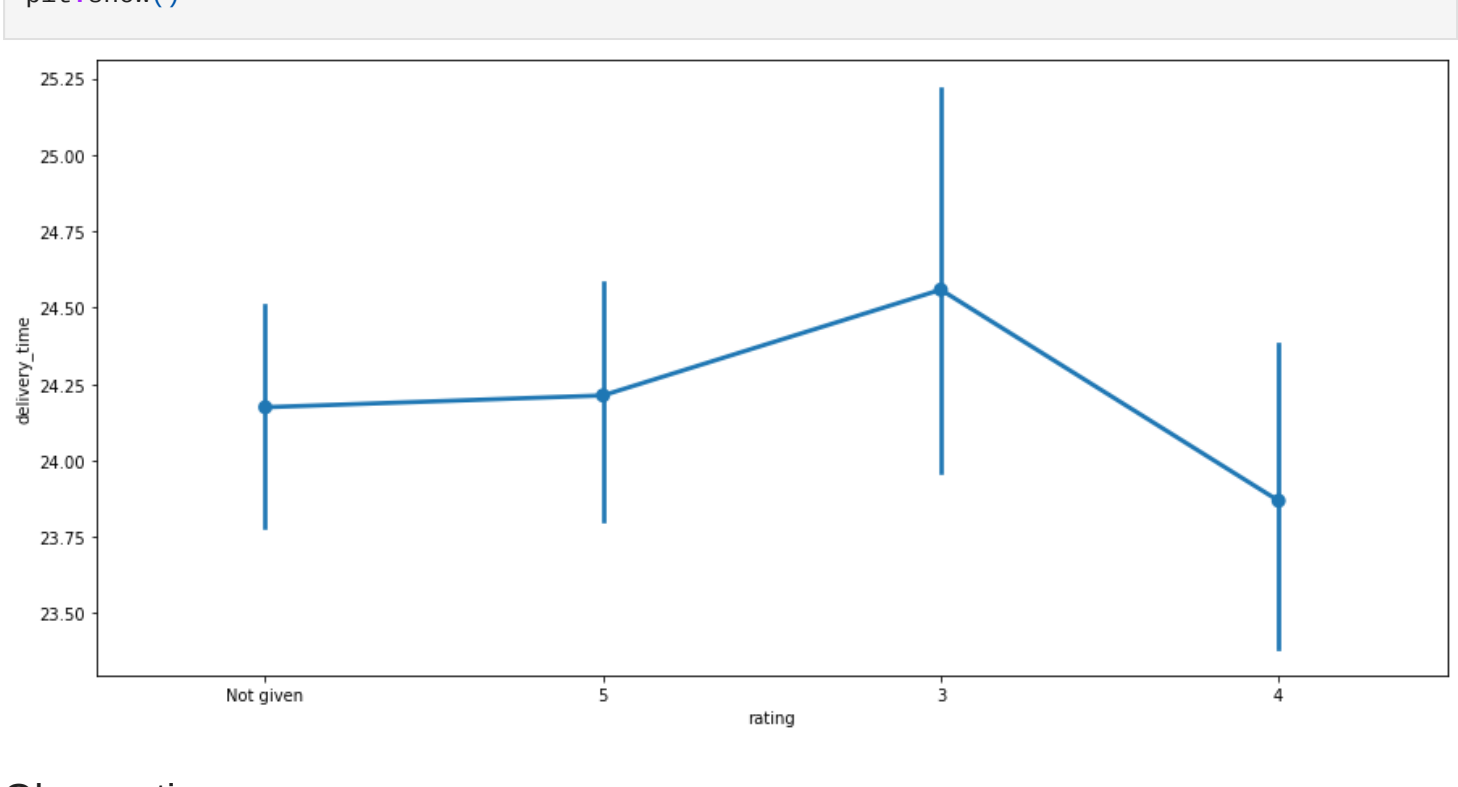
Observation

-There is a gradual increase in revenue from the last restuarant -But in comparision seperately, it is about a 200% to 700% increase -That is a very huge margin

Rating vs Delivery time

In [62]:

```
plt.figure(figsize=(15, 7))
sns.pointplot(x = 'rating', y = 'delivery_time', data = df)
plt.show()
```



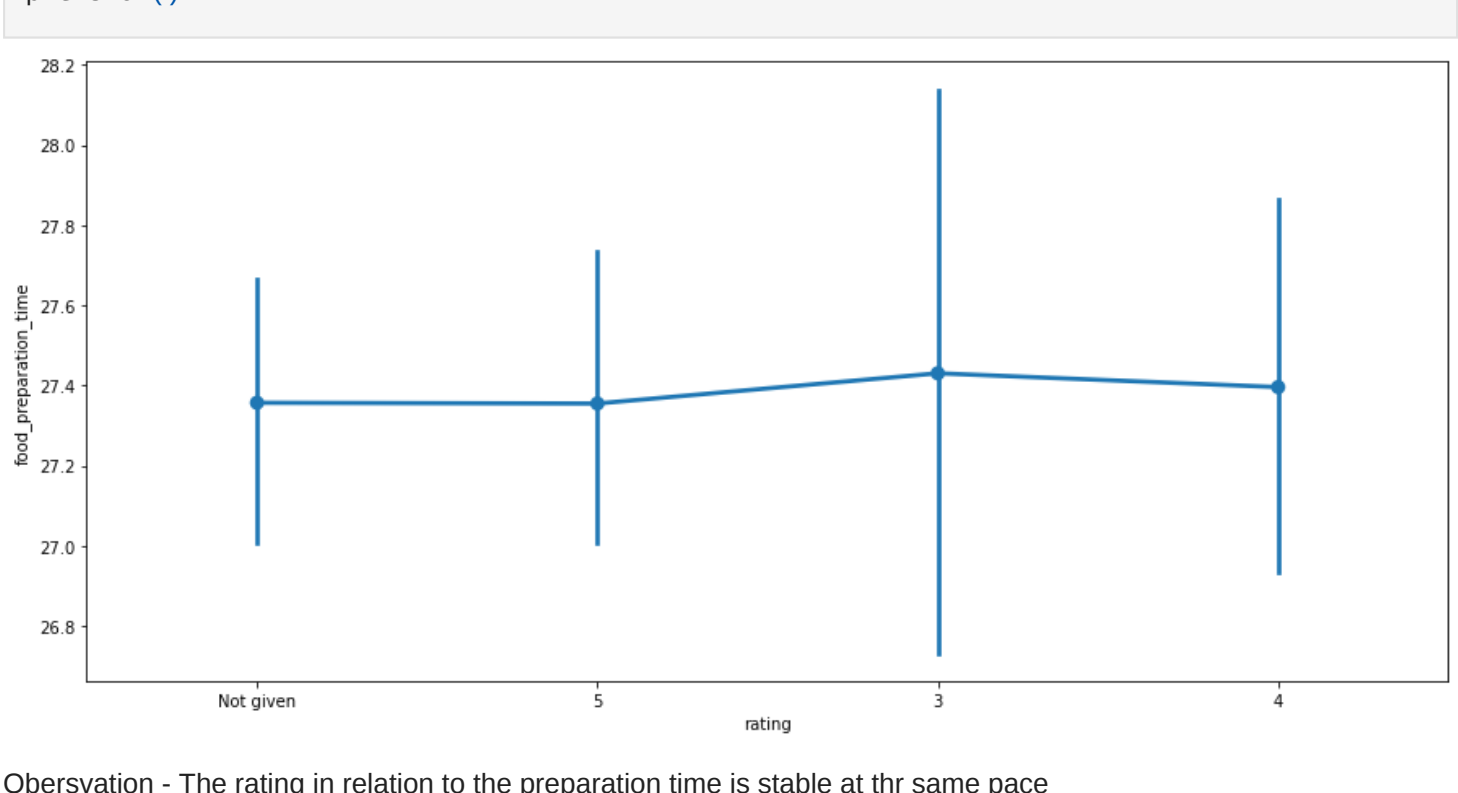
Observation

-The lowest rating is seen to have the highest derlivery time

Rating vs Food preparation time

In [63]:

```
plt.figure(figsize=(15, 7))
sns.pointplot(x = "rating", y = "food_preparation_time", data = df)
plt.show()
```

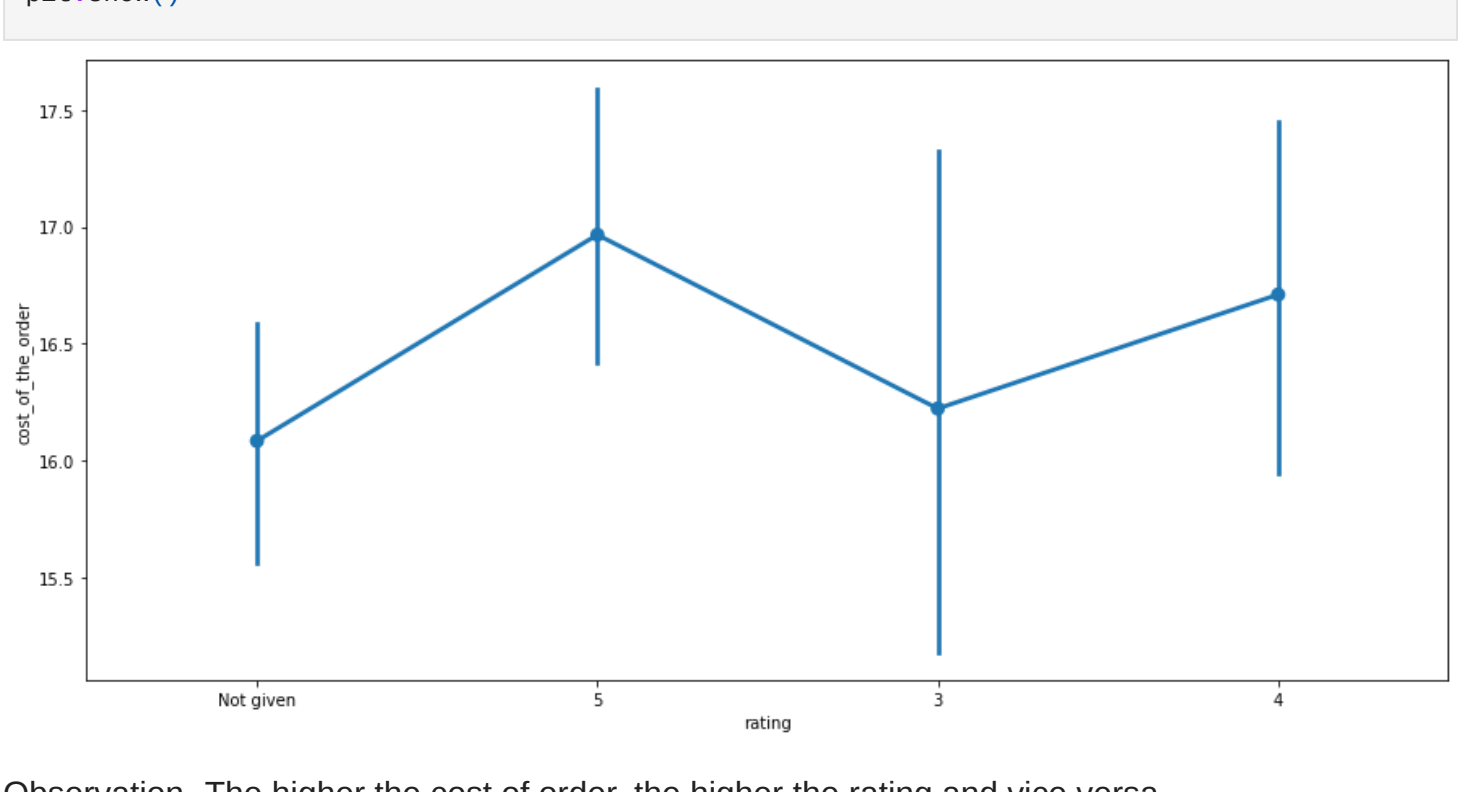


Obersvation - The rating in relation to the preparation time is stable at thr same pace

Rating vs Cost of the order

In [64]:

```
plt.figure(figsize=(15, 7))
sns.pointplot(x = "rating", y = "cost_of_the_order", data = df)
plt.show()
```

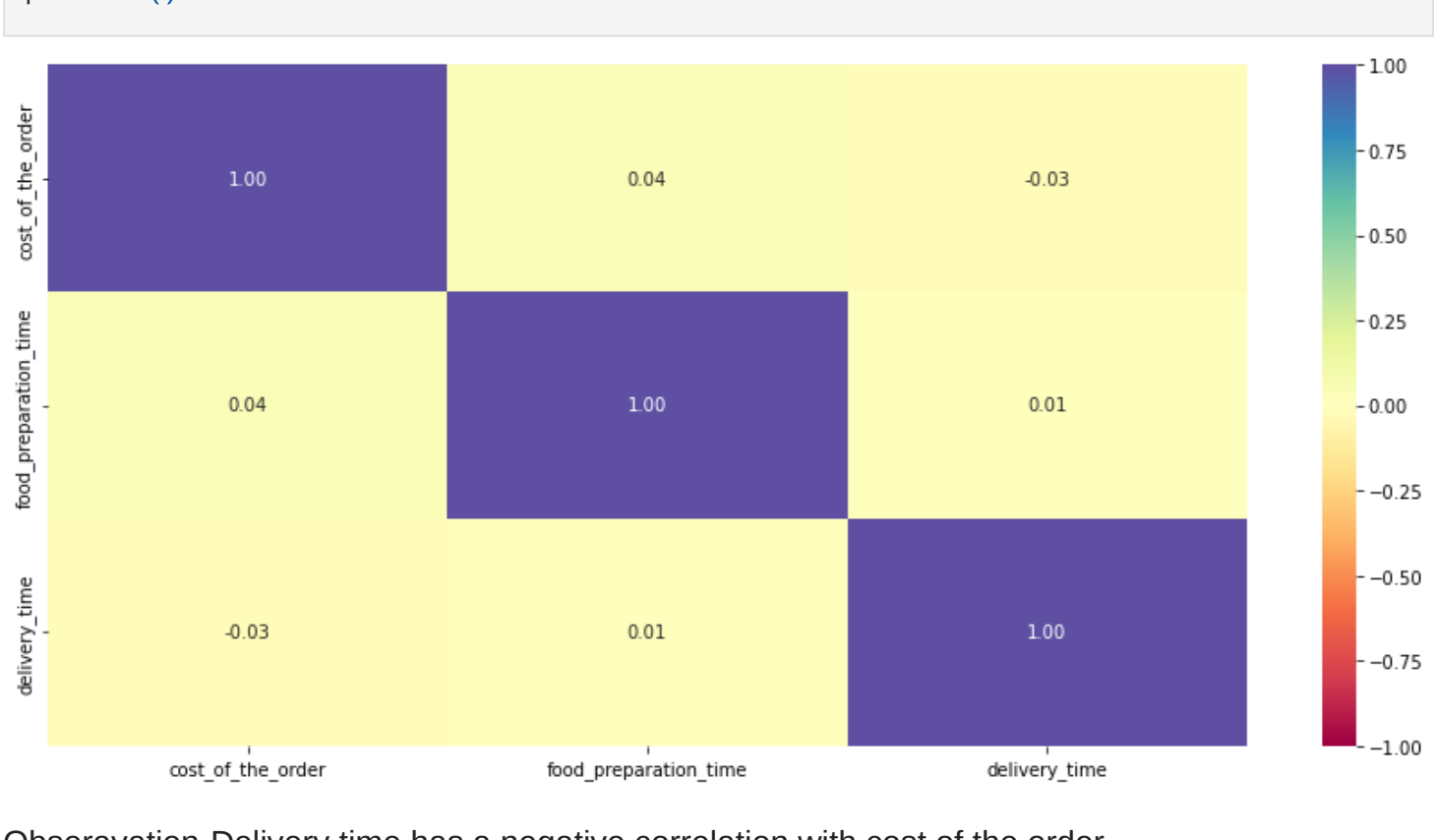


Observation- The higher the cost of order, the higher the rating and vice versa

Correlation among variables

In [65]:

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



Obseravation-Delivery time has a negative correlation with cost of the order

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. The restaurants fulfilling the criteria to get the promotional offer are:

In [66]:

```
df_rated = df[df['rating'] != 'Not given'].copy()
df_rated['rating'] = df_rated['rating'].astype('int')
lues(ascending = False).reset_index()
df_rating_count.head()
```

Out[66]:

	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 [19]:

```
rest_names = df_rating_count[df_rating_count['rating']>50]['restaurant_name']
df_mean_4 = df_rated[df_rated['restaurant_name'].isin(rest_names)].copy()
df_mean_4.groupby(['restaurant_name'])['rating'].mean().sort_values(ascending = False).reset
```

Out[19]:

	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

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. Finding the net revenue generated by the company across all orders.

In [20]:

Out[20]:

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

In [28]:

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

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?

In [35]:

```
The number of orders that have more than 60 minutes is: 200
Percentage of orders that have more than 60 minutes is 10.54 %
```

Out[35]:

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

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 [30]:

```
print('The mean delivery time on weekdays is around',
      round(df[df['day_of_the_week'] == 'Weekday']['delivery_time'].mean()),
     'minutes')
print('The mean delivery time on weekdays is around',
      round(df[df['day_of_the_week'] =='Weekend']['delivery_time'].mean()),
      'minutes')
```

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

Conclusion and Recommendations

What are your conclusions from the analysis? What recommendations would you like to share to help improve the business?

Conclusion -We can conclude that the total cost of order is on a positive skew.In relation to rating we found out that the higher the cost of order, the higher the rating. The customers seem to be getting their money's worth. -This also depended on the cuisine type. the more prefered cuisine types had a higher prepation and delivery time. -Out of the 14 different cuisines, we observed that American, Japanese, Italian and Chinese have the highest percentages and they make up more than 80% of the given statistics. -As observed, the least ordered cuisines had lesser preparation time and quicker delivery time. -Also,we observed that the weekdays have a higher delivery time than the weekends. This is somewhat expected as there maybe more road users commutting and more orders coming in.

In [ ]:

Based on the findings and conclusions in this study, the following recommendations are made: -Offering loyalty points to customers for walk-ins especially during weekdays to manage the delivery time -Setting up group ordering for customers at a reduced fee -Encouraging a duo meal plan, to make customers order more at once -delivery services can also be out-soured to meet up with the demands -To manage the disparity in the preference for particular cusines, reward reviews can be created to encourage feedback on reasons for thee prferences