A strategy to boost rating and drive new customers

FoodHub aggregator online food delivery

Austin University – Post Graduate Program in Artificial Intelligence

12/07/2025

Contents / Agenda

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

Executive Summary - Opening

Usually, when we talk about customer satisfaction in food delivery, we obsess over **speed**: *'How can we shave two minutes off delivery time?'* or *'How can we get the food out of the kitchen faster?*

However, our analysis of the latest dataset reveals a critical paradox: Our operations are actually running with incredible consistency. Our delivery times are stable, and our kitchen prep is standardized. But, statistically speaking, **speed is no longer moving the needle on our customer ratings.** A delivery that arrives in 24 minutes gets the exact same rating as one that arrives in 28.

Instead, the data points to a new driver: **Perceived Value.** We found that our highest satisfaction scores come exclusively from our highest-priced orders—specifically within our Premium segments like French and Southern cuisine. Conversely, our 'budget' options are where we see satisfaction dropping.

Executive Summary - Conclusions

Conclusions

The FoodHub business is solid; it has a consistent process to prepare and deliver the food to customers on time and with quality. The FoodHub has 5-star ratings on more than 50% of its reviews and an overall rating of 4.34. It shows that customers are overall satisfied with the food quality and delivery.

There is a clear positive correlation between Cost and rating. Customers ordering more expensive means, such as French and Southern, tend to give 5-star ratings. On the other hand, orders with low rate and unrated are basically low price orders. High price is usually associated with higher satisfaction.

Korean, Vietnamese, and Japanese food are the most affordable ranging average $12-$14 and driving the majority of the orders. The French food is the most expensive(median > $20) overall. It propels high ratings, probably as the result, perception, or marketing of the food quality.

Though the Korean food is generally fast to prepare, according to the multi-variable cuisine vs preparation time. There is a particular subset that suffers from significant delays(32,33) minutes, which are statistical outliers.

Executive Summary - Recommendations

Based on rates and price correlation

The low-cost orders are associated with lower rates or no ratings. The entry-level menus may be lacking quality or size.

Proposal:

- Review and possibly increase portion sizes
- Change or remove the menu with options that would better suit your audience(Vegan, Youth, Older)
- Review the food presentation

The delivery logistics

Given that the delivery logistics are not affecting the overall rate of the experience, as inferred from the data, since delivery time was not rated separately, restaurants should focus on improving the quality of the food, presentation, and portions. Considering optimizing for faster delivery will not drive customer satisfaction.

Low-engagement customers

Given that more than 700 customers did not rate the experience, it would be interesting to implement a post-order incentive (e.g., rate your experience and get 10% back on your next order). Urated orders are also associated with cheaper orders; getting feedback may be particularly important for determining whether the lower rate for more affordable food persists.

Marketing

Outside of American, Japanese, Italian, Chinese, Mexican, and Indian, the other restaurants received fewer than 30 orders in the period. FoodHub should focus its marketing to expand the offer to communities represented by the other cuisine type. The approach can increase the overall number of orders and customers.

Restaurants such as Five Guys are not doing well; they probably cater to the youth community. Campaigns to address the youth community may also lead to new customers(e.g., University promotions)

Business Problem Overview and Solution Approach

Overview

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 answer these questions and help the company improve its business. Please mention the solution approach / methodology

Approach

Approach & Methodology

Objective: To diagnose root causes of customer satisfaction variance and identify revenue optimization opportunities.

Methodology: Applied **Exploratory Data Analysis (EDA)** to decode the operational dataset.

Techniques:

- **Univariate Analysis:** Audited individual performance metrics (Cost, Time, Ratings) for consistency and outliers
- **Multivariate Analysis:** Examined the correlations between operational variables to isolate the true drivers of 5-star ratings.

Data Overview

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

```
# Check the shape of the dataset
df.shape ## Fill in the blank
```

```
(1898, 9)
```

The .csv file contains 1898 rows and 9 columns. It is a moderately sized dataset suitable for analysis and allow for fast processing.

Question 2 - What are the data types of the different columns in the dataset? [0.5 mark]

```
df.info()
```

```
(1898, 9)
<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
```

The 9 columns follow a basic pattern for column types. It uses int64 for integer numbers and Object to represent String or mixed data types.

The cost is represented as a float, as it should, because it allows fractions of dolar representations.

There are 4 Strings/Objects, 4 integers, and 1 Float number

Data Overview

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

```
# Checking for missing values in the data
def check_missing_values(dataframe):
    ''' returns te missing values in each column of the dataframe'''
    return dataframe.isnull().sum()


missing_values = check_missing_values(df)


print (f"Missing Values of each column:\n{missing_values}")
  #Write the appropriate function to print the sum of null values for each column
```

```
Missing Values of each column:
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: int64int64
```

There are no columns with null values
The dataset is whole in complete

Data Overview

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]

```
df.describe()
```

	order_id	customer_id	cost_of_the_order	food_preparation_time	delivery_time
count	1.90E+03	1898	1898	1898	1898
mean	1.48E+06	171168.4784	16.498851	27.37197	24.161749
std	5.48E+02	113698.1397	7.483812	4.632481	4.972637
min	1.48E+06	1311	4.47	20	15
25%	1.48E+06	77787.75	12.08	23	20
50%	1.48E+06	128600	14.14	27	25
75%	1.48E+06	270525	22.2975	31	28
max	1.48E+06	405334	35.41	35	33

```
####
#### The minimum time to prepare the food was 20.00 Min
#### The average time to prepare the food was 27.37 Min
#### The max time to prepare the food was    35.00 Min
####
```

Data Overview

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

```
#df['rating'].value_counts()['Not given'] ## Complete the code

# disregard of capitalization
count = (df['rating'].str.lower() == 'not given').sum()
print(f"Number of 'Not given' ratings: {count}")
```

Number of 'Not given' ratings: 736

736 orders were not rated out of 1898, meaning 39% of the orders were not rate

Univariate Analysis

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

Order ID

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

[19] ✓ 0.0s Python

```
1898
```

Customer ID

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

[20] ✓ 0.0s Python

```
1200
```

Restaurant name

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

[22] ✓ 0.0s Python

```
178
```

Cuisine type

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

[23] ✓ 0.0s Python

```
14
```

Out of 1898 customer orders there are 1200 unique customers, from 178 Restaurantes providing 14 Cuisine Types.

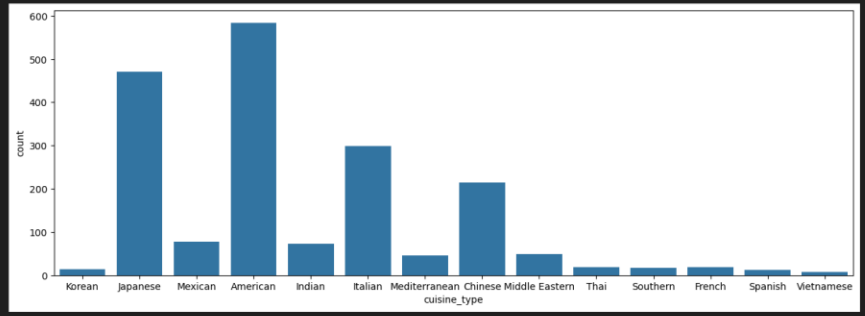
Univariate Analysis

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

```
plt.figure(figsize = (15,5))
sns.countplot(data = df, x = 'cuisine_type') ## Create a countplot for cuisine type.
```

0.3s Python

```
<Axes: xlabel='cuisine_type', ylabel='count'>
```




```
df.groupby('cuisine_type')['restaurant_name'].nunique().sort_values(ascending=False)
```

0.0s Python

cuisine_type	# restaurant_name
American	41
Italian	31
Japanese	29
Chinese	16
Indian	14
Mexican	11
Thai	9
Middle Eastern	7
Korean	5
Mediterranean	5
French	3
Spanish	3
Vietnamese	3
Southern	2

14 rows x 1 cols 25 per page Page 1 of 1

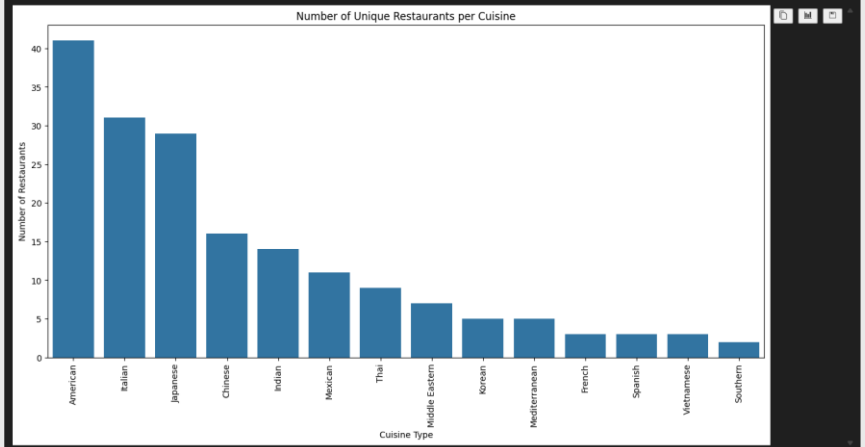
The restaurants that most sell are americans and japaneses, followed by Italians and chinese.
It appears that customer do not prefer Vietnameses, spanishes, and koreans, considering the orders. Despite, there are more Korean restaurants than vietnamese and sourthern.

Univariate Analysis

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

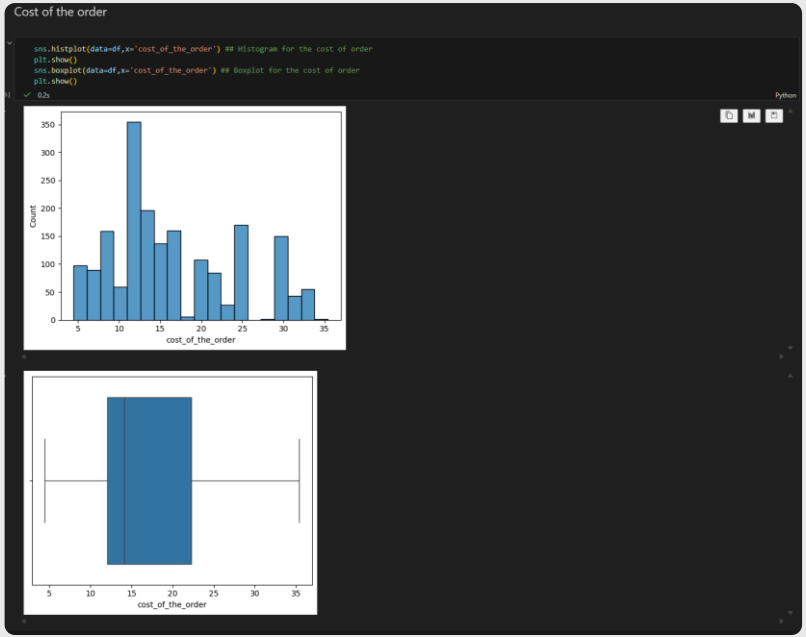
```
# Create a figure
plt.figure(figsize=(15, 7))

restaurants_per_cuisine = df.groupby('cuisine_type')['restaurant_name'].nunique().sort_values(ascending=False)
# Create the countplot
# Note: We use the original dataframe but we need to ensure we only count unique restaurants
# It is often easier to plot the processed series we created above
sns.barplot(x=restaurants_per_cuisine.index, y=restaurants_per_cuisine.values)

plt.xticks(rotation=90) # Rotate labels so they don't overlap
plt.title('Number of Unique Restaurants per Cuisine')
plt.ylabel('Number of Restaurants')
plt.xlabel('Cuisine Type')
plt.show()
```

0.2s Python



Univariate Analysis

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



The typical order cost is around 14. While the costs range from roughly 5 to 35
Most people tend to buy cheap below 14
The bulk revenue comes from orders from 12 to 22/23 dolars
There are no visible outliers represented may be for the fact there are no party orders
The higher orders are around 35/36 dolares, probably from families.

Univariate Analysis

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

Day of the week

Generate + Code + Markdown

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

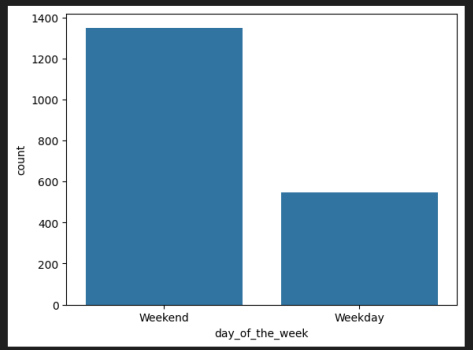
[26] ✓ 0.0s Python

```
2
```

```
sns.countplot(data = df, x = 'day_of_the_week') ## Complete the code to plot a bar graph for 'day_of_the_week' column
```

[27] ✓ 0.1s Python

```
<Axes: xlabel='day_of_the_week', ylabel='count'>
```



Most customers buy on weekends as opposed to weekdays. Probably, because people want to enjoy the weekend out.

Univariate Analysis

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

Rating

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

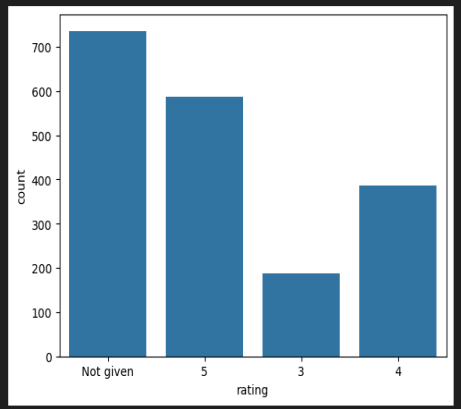
[28] ✓ 0.0s Python

```
4
```

```
sns.countplot(data = df, x = 'rating') ## Complete the code to plot bar graph for 'rating' column
```

[29] ✓ 0.2s Python

```
<Axes: xlabel='rating', ylabel='count'>
```



People do not rate all the time. It may happen because of the familiarity with technology or the rush of the day. It seems we do not have bad restaurants or food rated 1 and 2. Most restaurants are well rated 4 and 5. While some restaurants are average they are not the majority.

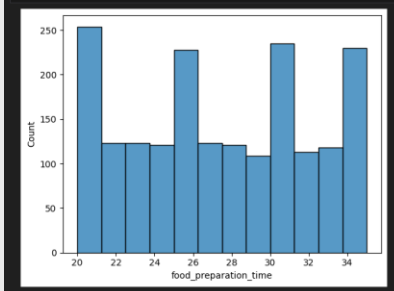
Univariate Analysis

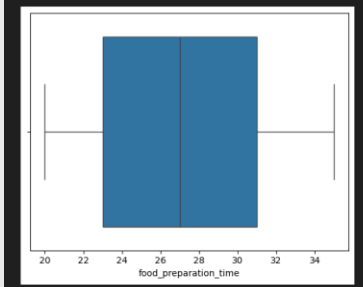
Question 6 - Explore all the variables and provide observations on their distributions. (Generallyhistograms, boxplots, countplots, etc. are used for univariate exploration.) [9 marks]

Food Preparation Time

```
sns.histplot(data=df,x='food_preparation_time') ## Complete the code to plot the histogram for the food preparation time
plt.show()
sns.boxplot(data=df,x='food_preparation_time') ## Complete the code to plot the boxplot for the cost of order
plt.show()
```

0.3s Python






```
df.groupby('cuisine_type')['food_preparation_time'].describe().sort_values(by='mean', ascending=False)
```

0.0s Python

cuisine_type	count	mean	std	min	25%	50%
Southern	17.0	27.588235294117655	5.523346172491497	20.0	24.0	28
Chinese	215.0	27.511627906976745	4.758524534847989	20.0	23.5	27
Japanese	470.0	27.51063829787234	4.654491531999785	20.0	24.0	27
Italian	298.0	27.483221476510067	4.519473803533745	20.0	23.0	26
American	584.0	27.440068493150687	4.658473604586773	20.0	23.0	27
Thai	19.0	27.315789473684214	5.496006018134084	21.0	22.0	28
Indian	73.0	27.109589041095893	4.620490726299648	20.0	23.0	27
Mediterranean	46.0	27.0	4.908043285148256	20.0	23.0	27
Spanish	12.0	26.916666666666668	4.501683186892826	20.0	23.75	27
French	18.0	26.888888888888889	5.039771158008645	21.0	22.5	25
Mexican	77.0	26.727272727272727	4.604449688890385	20.0	23.0	26
Middle Eastern	49.0	26.673469387755	4.012416995902876	20.0	24.0	27
Vietnamese	7.0	25.714285714285715	5.154748157905347	20.0	21.5	25
Korean	13.0	25.461538461538464	3.971049076658556	20.0	23.0	25

14 rows x 8 cols 25 per page Page 1 of 1

Around 25% of the food took less than 23 min to prepare, being the minimun 20 min
50% of the food took from 23 to 33 min to prepare.

Overall the preparation time for all restaurants/cuisines are the same. Deeper looking at the average, min, and max of all the restaurants there are no outliers. Restaurants probably try to be closer to preparation time of other restaurants to stay competitive.

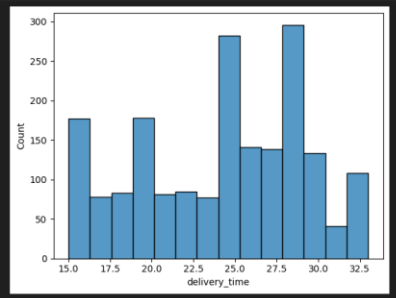
Univariate Analysis

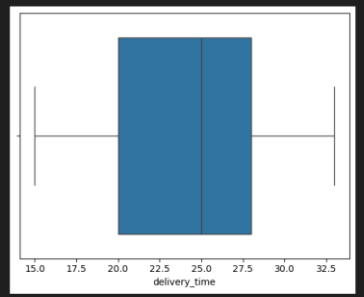
Question 6 - Explore all the variables and provide observations on their distributions. (Generallyhistograms, boxplots, countplots, etc. are used for univariate exploration.) [9 marks]

Delivery time

```
sns.histplot(data=df,x='delivery_time') ## 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()
```

[54] ✓ 0.3s Python





The fastest delivery time is 15 min
The common delivery is 25 min
The longest delivery is 33 min.
Because the median line is slightly closer to the right side of the box. It implies that
most deliveries are around 25-28 min.
Looking from the left skew to the mid skew we can infer that there are more fast deliveries than slow.

Considering cost of order and delivery time it is fair to say the low cost (14) is delivered in average time.
The logistic is predictable for all restaurants types. It may be a good place to improve in the future to attract more customers.

Univariate Analysis

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

```
df[['restaurant_name', 'cuisine_type']].value_counts().head(100) ## Complete the code
```

0.0s Python

('restaurant_name', 'cuisi...	# 0
('Shake Shack', 'American')	219
('Blue Ribbon Sushi', 'Japanese')	119
('The Meatball Shop', 'Italian')	112
('Blue Ribbon Fried Chicken', 'Americ	96
('Parm', 'Italian')	68
('RedFarm Broadway', 'Chinese')	59
('RedFarm Hudson', 'Chinese')	55
('TAO', 'Japanese')	49
('Han Dynasty', 'Chinese')	46
('Blue Ribbon Sushi Bar & Grill', 'Japa	44

100 rows x 1 cols 10 per page Page 1 of 10

```
df[['cuisine_type']].value_counts().head(100) ## Complete the code
```

0.0s Python

index	# 0
('Southern',)	17
('Korean',)	13
('Spanish',)	12
('Vietnamese',)	7

14 rows x 1 cols 10 per page Page 2 of 2

The five restaurants in terms of orders are

restaurant_name	cuisine_type	
Shake Shack	American	219
Blue Ribbon Sushi	Japanese	119
The Meatball Shop	Italian	112
Blue Ribbon Fried Chicken	American	96
Parm	Italian	68

Univariate Analysis

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

```
# 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 the cuisine type on weekend
```

0.0s Python

	# cuisine_type
American	415
Japanese	335
Italian	207
Chinese	163
Mexican	53
Indian	49
Mediterranea	32
Middle Easter	32
Thai	15
French	13

14 rows x 1 cols 10 per page Page 1 of 2

The most popular is American with 415 orders

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

Univariate Analysis

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

```
# 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 orders having cost above $20

# 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), '%')
```

✓ 0.0s Python

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

The number of total orders that cost above 20 dollars is: 555

Percentage of orders above 20 dollars: 29.24 %

Univariate Analysis

Question 10 - What percentage of the orders cost more than 20 dollars? [2 marks]

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

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

0.0s Python

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

The mean delivery time for the order is 24.16 minutes

Univariate Analysis

Question 11 - The company has decided to give 20% discount vouchers to the top 5 most frequent

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

0.0s Python

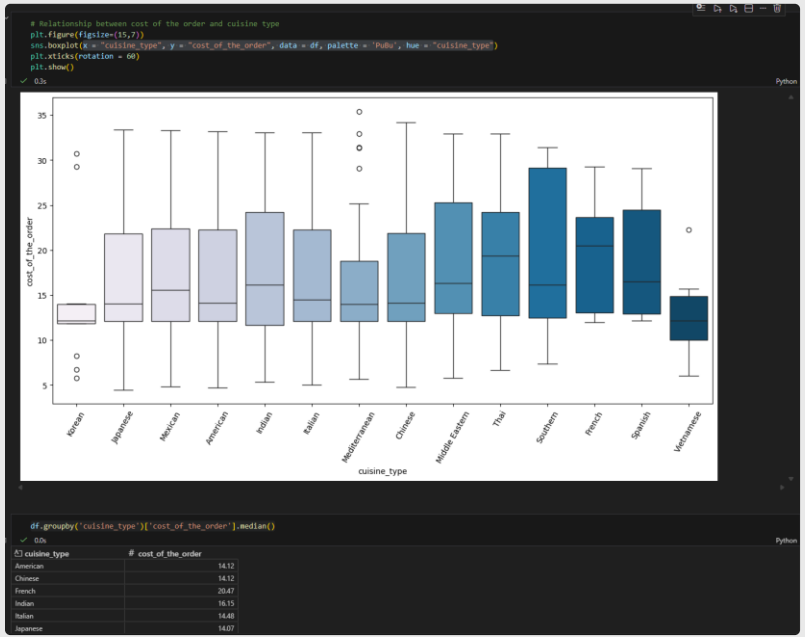
	# customer_id
52832	13
47440	10
83287	9
250494	8
259341	7

5 rows x 1 cols 10 per page Page 1 of 1

customer_id	count
52832	13
47440	10
83287	9
250494	8
259341	7

Multivariate Analysis

Question 11 - Perform a multivariate analysis to explore relationships between the important variables in the dataset. (It is a good idea to explore relations between numerical variables as well as relations between numerical and categorical variables) [10 marks]




```
df.groupby('cuisine_type')['cost_of_the_order'].median()
```

cuisine_type	cost_of_the_order
American	14.12
Chinese	14.12
French	20.47
Indian	16.15
Italian	14.48
Japanese	14.07

The graph shows that Frent has the highest median cost.
it also has the majority of the order above mediam.
It is typical order is $6 to 7
dollars(df.groupby('cuisine_type')['cost_of_the_order'].median()) more expensive than almost any other cuisine

cuisine_type	mediam
American	14.120
Chinese	14.120
French	20.470
Indian	16.150
Italian	14.480
Japanese	14.070
Korean	12.180
Mediterranean	13.995
Mexican	15.570
Middle Eastern	16.300
Southern	16.110
Spanish	16.520
Thai	19.350
Vietnamese	12.130

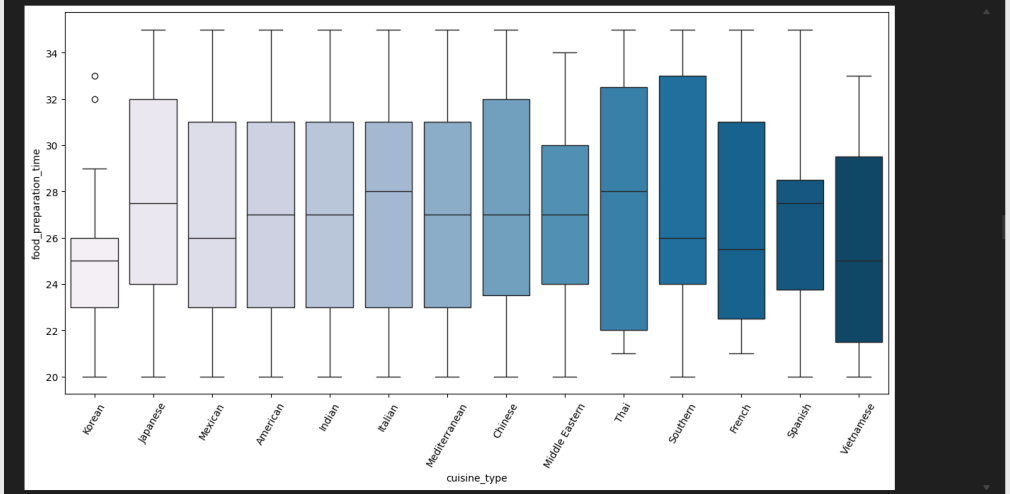
Multivariate Analysis

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]

Cuisine vs Food Preparation time

```
# Relationship between food preparation time and cuisine type
plt.figure(figsize=(15,7))
sns.boxplot(x = "cuisine_type", y = "food_preparation_time", data = df, palette = 'PuBu', hue = "cuisine_type")  ## Complete the code to visualize the relationship between food
plt.xticks(rotation = 60)
plt.show()
```

0.3s Python



The graphic shows that the median food preparation time is close to all cuisine.
Korean cuisine is the fastest preparation given 50% is from 23 to 26 minutes, and the max 29 min. It is the lowest is 20 min. Korean is also close to Vietnamese, though it has some outliers.

Italian, Thai, and Southern have median closer to 28 minutes, they are slightly slower.

The lowest time to prepare is not much different for all restaurants/cuisine. It ranges from 20 to 21 min.

The graphic talks a lot about the processes in the kitchen, they seem to be the same for all order, for the
outcome time is basically the same.

One can also predict given the graphic that any order is going to be ready in the range of 20-35 min.

The Korean outliers 32, 33

```
df[df['cuisine_type'] == 'Korean'][['cuisine_type', 'food_preparation_time']].value_counts()
```

cuisine_type	food_preparation_time	
Korean	25	4
	21	2
	26	2
	20	1
	23	1
	29	1
	32	1
	33	1

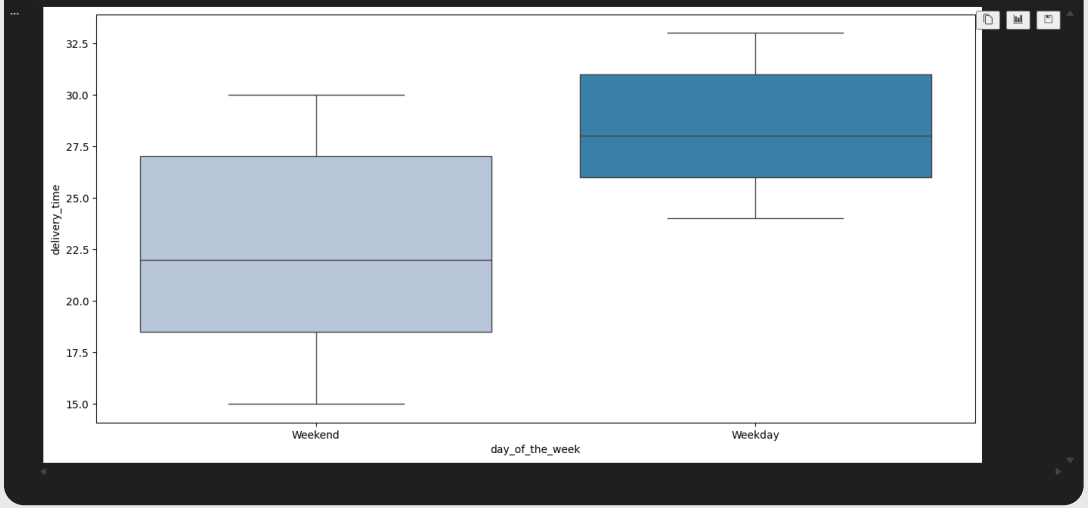
Multivariate Analysis

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]

Day of the Week vs Delivery time

```
# Relationship between day of the week and delivery time
plt.figure(figsize=(15,7))
sns.boxplot(x = "day_of_the_week", y = "delivery_time", data = df, palette = 'PuBu', hue = "day_of_the_week")  ## Complete the code to visualize the relationship between day of
plt.show()
```

0.1s Python



The graphic shows that the delivery time during week days is more than the median delivery time during weekend. Probably because of the amount of deliverers available, giving there are more delivers during weekend. It may also be the result of low traffic on the weekend.

The lowest delivery time is 15 min on weekend, and the highest 35 min on weekday.

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]

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

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

0.0s Python

restaurant_name	# cost_of_the_order
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	834.5
Han Dynasty	755.29
Blue Ribbon Sushi Bar & Grill	666.62

14 rows x 1 cols 10 per page Page 1 of 2

The code generates a slice of restaurantes and the total value of the orders for the analysed period. Shake Shack, the American Cuisine Restaurant, sells the most. It is followed by the Meatball Shop restaurant with $2145.21.

The code aims to display only 14 snapshots of the result.

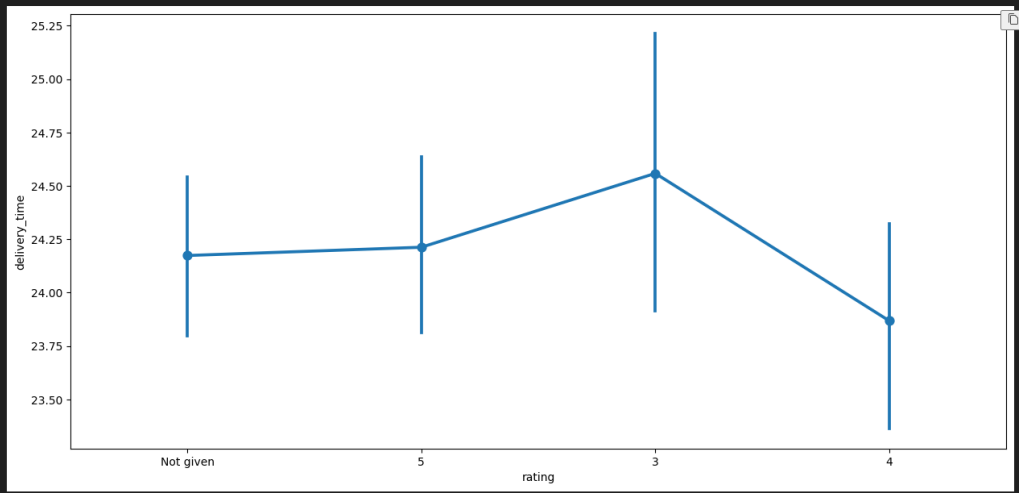
Multivariate Analysis

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]

Rating vs Delivery time

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

0.2s Python



The data shows that there is no strong correlation between the rate and the average delivery time,
which ranges from slightly higher than 23.75 to 24.5, a thin difference.

There is a slightly higher delivery time on same samples for orders with rate 3.

There are lots of overlaps, line at the same swimlane, height and low, which indicates it the difference is not meaningful

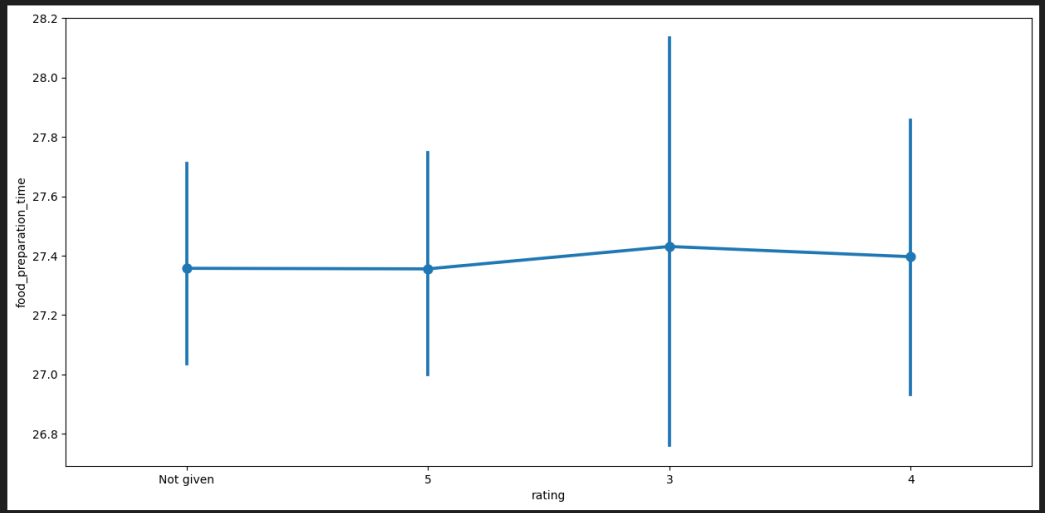
Multivariate Analysis

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]

Rating vs Food preparation time

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

0.3s Python



The graphic shows that the average of the rating to food preparation time is the same for all the rates.
It shows that overall the preparation time is not related to rate.

it seems that time is not related to rating.

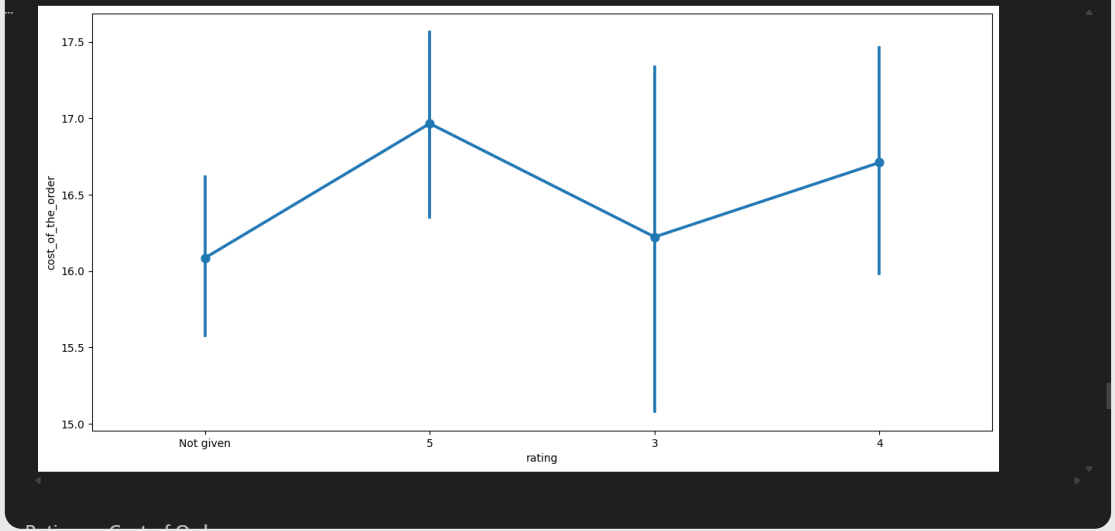
Multivariate Analysis

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]

Rating vs Cost of the order

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

0.2s Python



It shows that higher spending are related to higher ratings. One can see from the spike in the rating 5 median, going up to $17

The lower rates are associated to cheaper order. May be the lower level food is not impressing customer, it is just as expected.

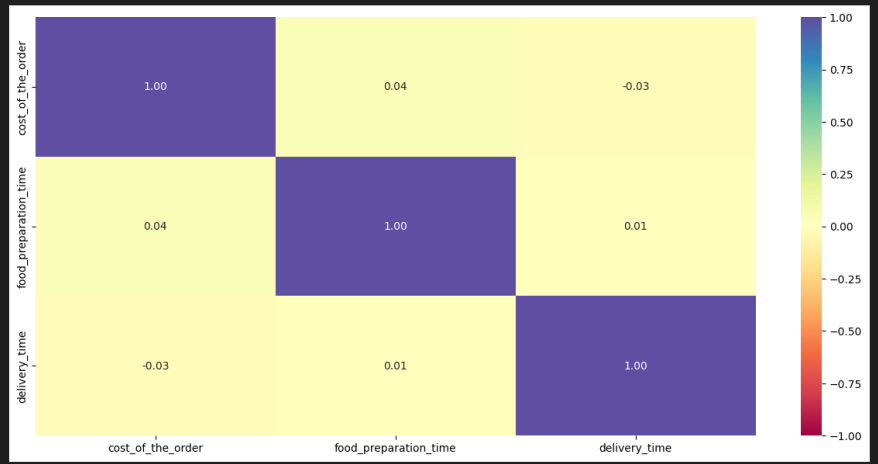
Multivariate Analysis

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]

Correlation among variables

```
# Plot the heatmap
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()
```

[96] ✓ 0.2s Python



Correlation among variable (cost_of_the_order, food_preparation_time, delivery_time,)

Cost does not affect preparation_time, it is the opposite of what normally happens, more expensive food tend
to be take more time to prepare. It seems the restaurants analysed have a solid approach to prepare the food.

Cost also does not affect delivery time, meaning the logistic employed to deliver is well round.

Delivery time does not affect food preparation time or vice versa.

Multivariate Analysis

- **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]

```
# 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()
```

0.0s Open 'df_rating_count' in Data Wrangler Python

	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

5 rows x 2 cols 10 per page Page 1 of 1

Restaurants and their rating counts

restaurant_name	rating
Shake Shack	133
The Meatball Shop	84
Blue Ribbon Sushi	73
Blue Ribbon Fried Chicken	64
RedFarm Broadway	41

Multivariate Analysis

- **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]

```
# 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 names having rating count more than 50

# 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_rating = df_mean_4.groupby(['restaurant_name'])['rating'].mean().sort_values(ascending = False).reset_index().dropna() ## Complete the code to find the mean rating

# filter for average rating greater than 4
df_avg_rating_greater_than_4 = df_mean_4_rating[df_mean_4_rating['rating'] > 4].sort_values(by='rating', ascending=False).reset_index(drop=True)  ## Complete the code to find r

df_avg_rating_greater_than_4
```

0.0s Open 'df_avg_rating_greater_than_4' in Data Wrangler Python

	restaurant_name	rating
0	The Meatball Shop	4.511904761904762
1	Blue Ribbon Fried Chicken	4.328125
2	Shake Shack	4.278195488721804
3	Blue Ribbon Sushi	4.219178082191781

4 rows x 2 cols 10 per page Page 1 of 1

Restaurants and their rating greater than 4
The Meatball Shop exceed in rating getting a 4.5+ rate

Multivariate Analysis

- **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]

```
#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()
df.groupby('restaurant_name')['Revenue'].sum().sort_values(ascending=False).head()
```

0.0s Python

restaurant_name	# Revenue
Shake Shack	703.607
The Meatball Shop	419.8285
Blue Ribbon Sushi	360.46049999999997
Blue Ribbon Fried Chicken	340.2035
Parm	218.557

5 rows x 1 cols 10 per page Page 1 of 1

The column is cost of the order

restaurant_name	Revenue
Shake Shack	703.61
The Meatball Shop	419.83
Blue Ribbon Sushi	360.46
Blue Ribbon Fried Chicken	340.20
Parm	218.56

Total Revenue
$6166.3 Dollars

```
# 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')
```

0.0s Python

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

Total Revenue

The net revenue is around 6166.3 dollars

Multivariate Analysis

- **Question 15 -** he 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]

```
# 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 9 for reference)

# Calculate the percentage of orders with total time > 60 minutes
df_60 = df[df['total_time'] > 60]
percentage_over_60 = (df_60.shape[0] / df.shape[0]) * 100

print(f"Percentage of orders taking more than 60 minutes: {percentage_over_60:.2f}%")
```

0.0s Python

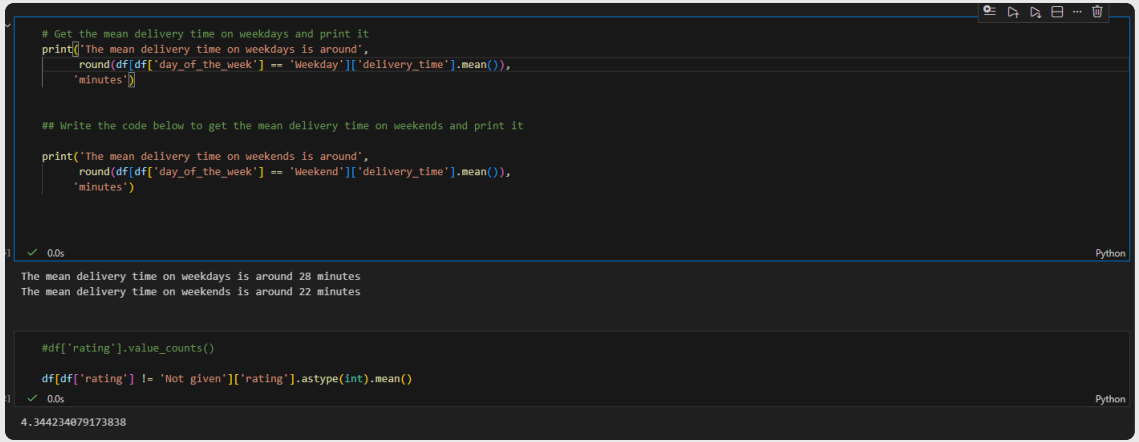
```
Percentage of orders taking more than 60 minutes: 10.54%
```

The column is cost of the order

restaurant_name	Revenue
Shake Shack	703.61
The Meatball Shop	419.83
Blue Ribbon Sushi	360.46
Blue Ribbon Fried Chicken	340.20
Parm	218.56

Multivariate Analysis

- **Question 16 - 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]**



```
# 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 weekends is around',
      round(df[df['day_of_the_week'] == 'Weekend']['delivery_time'].mean()),
      'minutes')
```

0.0s Python

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

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

df[df['rating'] != 'Not given']['rating'].astype(int).mean()
```

0.0s Python

```
4.344234079173838
```

The mean delivery time on weekdays is around 28 min
The mean delivery time on weekends is around 22 min

APPENDIX

Great Learning

Happy Learning !