Food-Hub Ordering Project

Python Foundations

April 10, 2023

Contents / Agenda

- Executive Summary
- Business Problem Overview and Solution Approach
- Data Overview Summary and Observations Analysis, including:
  - EDA - Univariate Analysis
  - EDA - Multivariate Analysis
- Appendix

Executive Summary

- Summary of observations and conclusions
  - No rating is the highest contribution and second rating is 5. It is suspected most satisfied customer did not bother to vote and submit the rating.
  - Cuisine type ordered was consistent regardless weekdays or weekends. American, Japanese, Italian and Chinese food is the top 4 contributor of cuisine type.
  - Weekends order is the highest compared with weekdays. However, delivery time for weekdays is longer than weekends.
  - Highest rating may be associated with faster delivery time regardless cuisine type

Business Problem Overview and Solution Approach

- Business problem overview:
  - Cuisine type needs to be more diversify to tackle low cuisine type order
  - Business owner need to focus more improvement on weekdays sales
  - Expensive food cost may not attractive more customers
- Solution approach/business improvement/recommendation
  - Improve customer rating submission, by giving them voucher discount or reward for the next purchase
  - Business owner with the lowest cuisine type may need to improve by giving a food discount for certain amount of time window
  - Better food promotion or discount need to focus to boost sales on weekday
  - Delivery time at weekdays need to improve to boost restaurant rating
  - Business owner need to improve which total order is more than 60 minutes, by giving customer some food discount for the next purchase as a apologize gesture

Data Overview Summary and Observations Analysis

- It is found the total analysed restaurant took part in the single smartphone app for restaurant in NY is 178 with total 14 cuisine type.
- Below is the overall summary for analysed data. The order cost is ranging between 4 dollars to 35 dollars, and average of foods to be prepared is around 27 minutes.
- Based on histogram, delivery time is ranging between 15 to 33 minutes, with the most highest delivery time is between 25-28 minutes.

	Min	Average	Max
cost_of_the_order (dollars)	4.47	16.50	35.41
food_preparation_time (minutes)	20	27	35
delivery_time (minutes)	15	24	33

Data Overview Summary and Observations Analysis

- The histogram also found the highest count of cost_of_the_order is between 11-12 dollars, while 34-35 dollars is the lowest count.
- The countplot divided data into 2, which is weekend and weekdays with weekend is the higher contributor of the food ordered compared with weekdays. It is assumed that weekend is most people stay at home and recuperate with their family, hence food delivery is expected.
- This assumption may be true due to we found estimation 1350 orders in weekend while weekday was found only 550 orders.
- Countplot also disclosed 14 cuisine types were divided during this analysis, with the top 4 cuisine is determined to be American, Japanese, Italian and Chinese food (the top 4 cuisine also similar on weekend) while countplot also disclosed the lowest cuisine type is Vietnamese and Spanish and Korean estimated share the same lowest spot.

Data Overview Summary and Observations Analysis

- Below is the top 5 restaurant:
  - Shake Shack
  - The Meatball Shop
  - Blue Ribbon Sushi
  - Blue Ribbon Fried Chicken
  - Parm
- We found there is 555 orders/29.24% for total order more than 20 dollars.
- Heatmap plot shows very weak correlation between the variables.

Data Overview Summary and Observations Analysis

- Boxplot analysis (multivariate) analysis shows:
  - cost_of_the_order is varied with cuisine_type, and observed Korean, Mediterranean and Vietnamese having some outliers.
  - Food_preparation_time is more or less having a similar average regardless cuisine_type
  - Delivery time generally takes longer on weekdays
- Pointplot analysis (multivariate) analysis shows:
  - Highest rating 4 may be associated with faster delivery_time regardless cuisine_type, while rating 3 is having larger spread compared with rating 5 and no rating.
  - Average rating 5 and 4 for cost_of_the_order is higher than rating 3 and no rating, with estimated around 16.8 dollars.

Data Overview Summary and Observations Analysis

- Restaurant who is able to fulfil the rating count more than 50 and rating 4 and able to get promotional offer: Shake Shack, The Meatball Shop, Blue Ribbon Sushi, Blue Ribbon Fried Chicken, RedFarm Broadway
- Net revenue for total all restaurant: 6166.3 dollars
- We found total order for more than 60 minutes is 200 dollars/10.54%
- Mean delivery weekdays=28 minutes, mean delivery weekends=22 minutes

APPENDIX

Data Overview

- Q1: How many rows and columns present in the data
  - By using df.shape, it has 1898 columns and 9 rows

```
In [4]: #rows and columns
        df.shape

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

- Q2: What are the datatypes of the different columns in the dataset
  - By using df.info(), it will print to concise the dataframe summary

```
In [39]: 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
```

Data Overview

- Q3: Are there any missing values in the data?
  - No missing values found in the data

```
In [40]: # Checking for missing values in the data
         print(df.isnull().sum())
```

```
order_id                0
customer_id             0
restaurant_name         0
cuisine_type            0
cost_of_the_order       0
day_of_the_week         0
rating                  0
food_preparation_time   0
delivery_time           0
dtype: int64
```

Data Overview

- Q4: Check the statistical summary of the data. What is the minimum, average, and maximum time it takes for food to be prepared once an order is placed?

```
In [23]: #statistical summary
         df.describe().T
```

Out[23]:

	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

- Minimum time takes for food to be prepared: 20 minutes
- Maximum time takes for food to be prepared: 35 minutes
- Average time takes for food to be prepared: 27 minutes

Data Overview

- Q5: How many orders are not rated?

```
In [41]: df['rating'].value_counts()

Out[41]: Not given    736
         5            588
         4            386
         3            188
         Name: rating, dtype: int64
```

  - It is found that 736 of food order were not given a rating by customers

Univariate Analysis

- Q6: Explore all the variables and provide observations on their distributions. (Generally, histograms, boxplots, countplots, etc. are used for univariate exploration.)

Order ID

```
In [42]: # check unique order ID
         df['order_id'].nunique()
Out[42]: 1898
```

Customer ID

```
In [43]: # check unique customer ID
         df['customer_id'].nunique()
Out[43]: 1200
```

Restaurant name

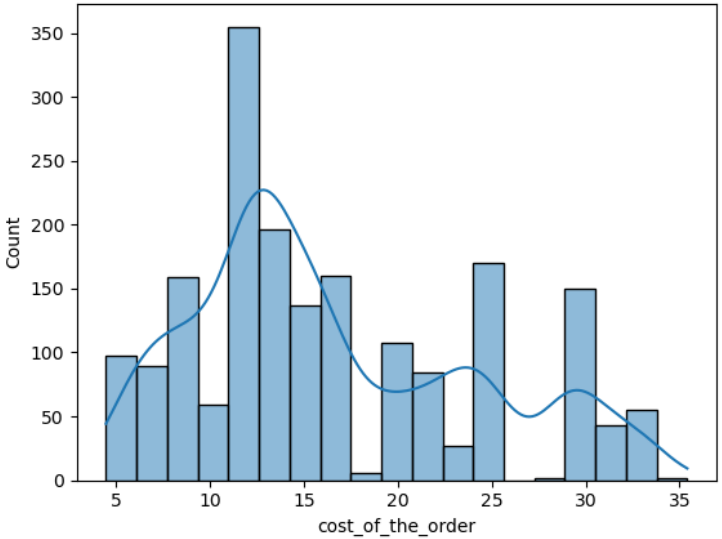
```
In [44]: # check unique Restaurant Name
         df['restaurant_name'].nunique()
Out[44]: 178
```

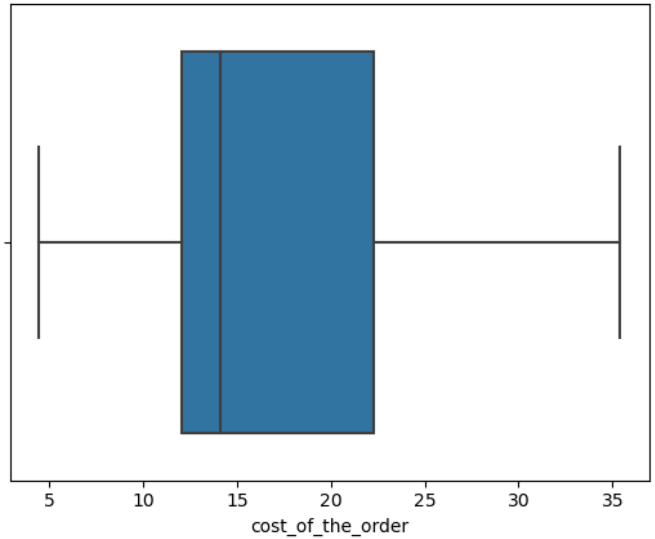
Cuisine type

```
In [45]: # Check unique cuisine type
         df['cuisine_type'].nunique()
Out[45]: 14
```

Univariate Analysis

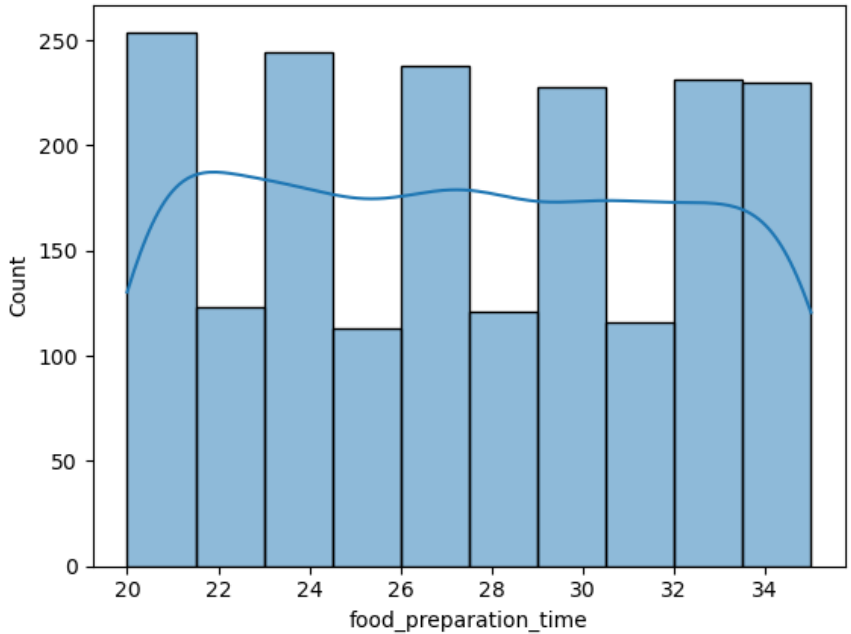
- Q6: Explore all the variables and provide observations on their distributions. (Generally, histograms, boxplots, countplots, etc. are used for univariate exploration.)

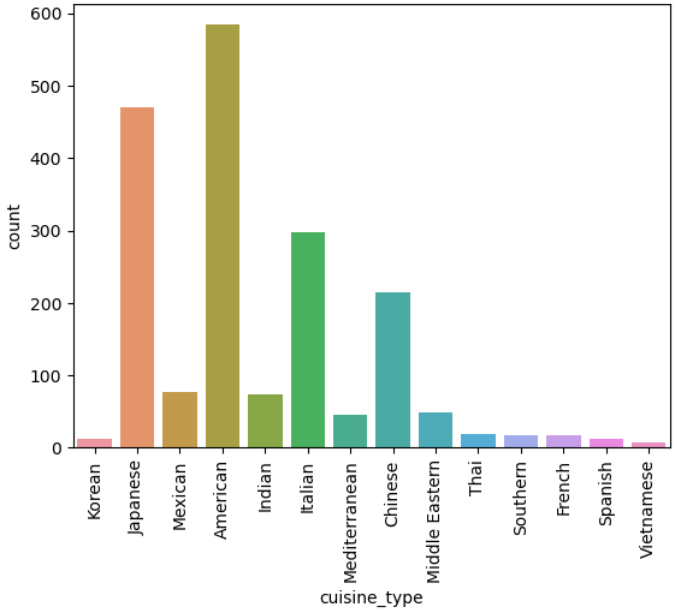




Univariate Analysis

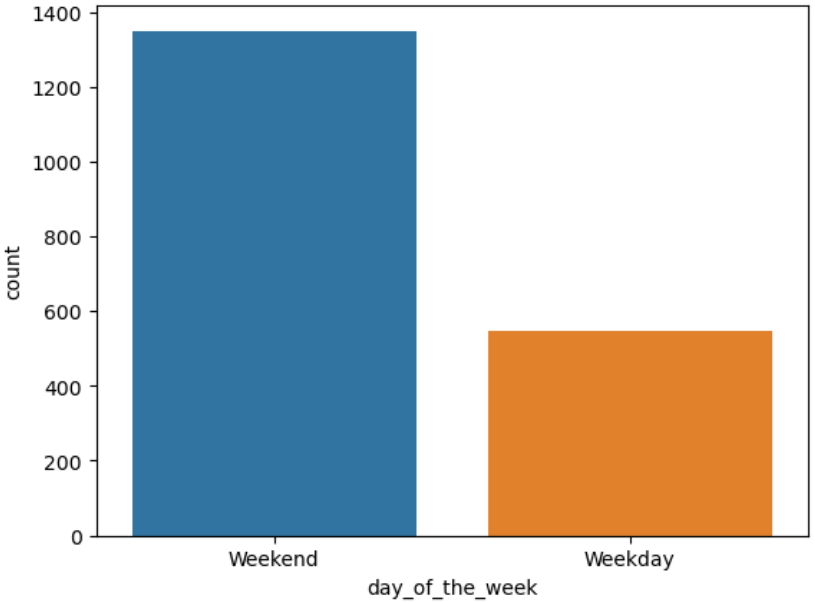
- Q6: Explore all the variables and provide observations on their distributions. (Generally, histograms, boxplots, countplots, etc. are used for univariate exploration.)





Univariate Analysis

- Q6: Explore all the variables and provide observations on their distributions. (Generally, histograms, boxplots, countplots, etc. are used for univariate exploration.)



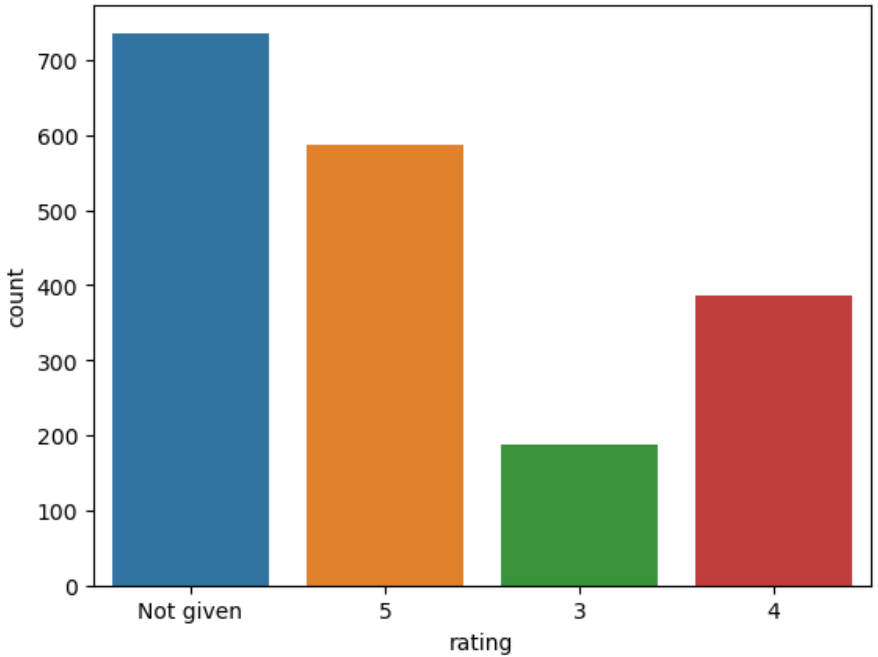
Day of the week

```
In [17]: # # Check the unique values
         df['day_of_the_week'].nunique()

Out[17]: 2
```

Univariate Analysis

- Q6: Explore all the variables and provide observations on their distributions. (Generally, histograms, boxplots, countplots, etc. are used for univariate exploration.)



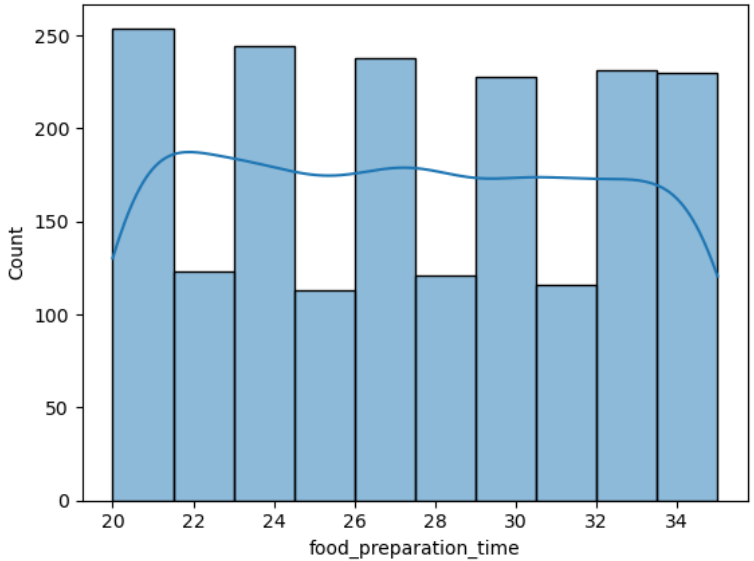
Rating

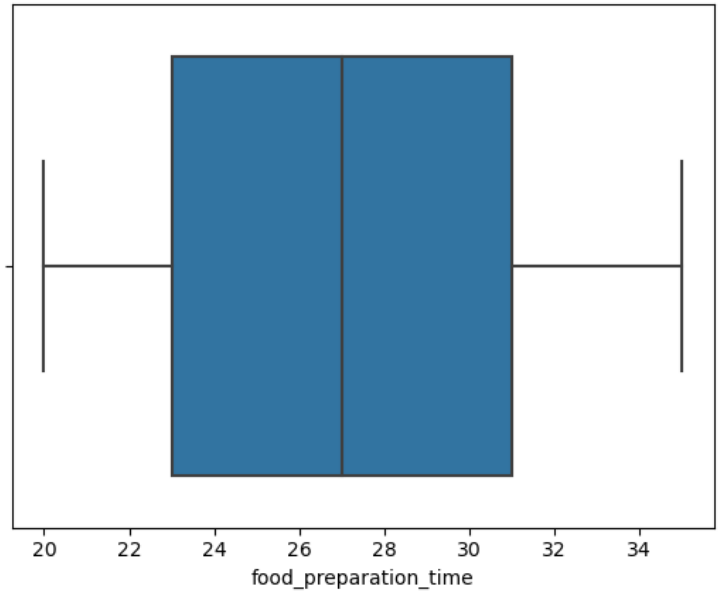
```
In [18]: # Check the unique values
         df['rating'].nunique()

Out[18]: 4
```

Univariate Analysis

- Q6: Explore all the variables and provide observations on their distributions. (Generally, histograms, boxplots, countplots, etc. are used for univariate exploration.)





Univariate Analysis

- Q7: Which are the top 5 restaurants in terms of the number of orders received?

```
df[['restaurant_name','order_id']].groupby('restaurant_name').count().sort_values(by = 'order_id',axis = 0, ascending=False).head()
```

Out[19]:

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

Univariate Analysis

- Q8: Which is the most popular cuisine on weekends?

```
df[df['day_of_the_week'] =='Weekend'].groupby('cuisine_type').count().sort_values(by = 'order_id',axis = 0, ascending=False).head()
```

Out[25]:

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

Univariate Analysis

- Q9: What percentage of the orders cost more than 20 dollars?

```
df_greater_than_20 = df[df['cost_of_the_order']>20]
print('The number of total orders that cost above 20 dollars is:', df_greater_than_20.shape[0])
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 orders
Percentage of orders above 20 dollars: 29.24%
```

Univariate Analysis

- Q10: What is the mean order delivery time?

```
mean_del_time = df['delivery_time'].mean()
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: 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.

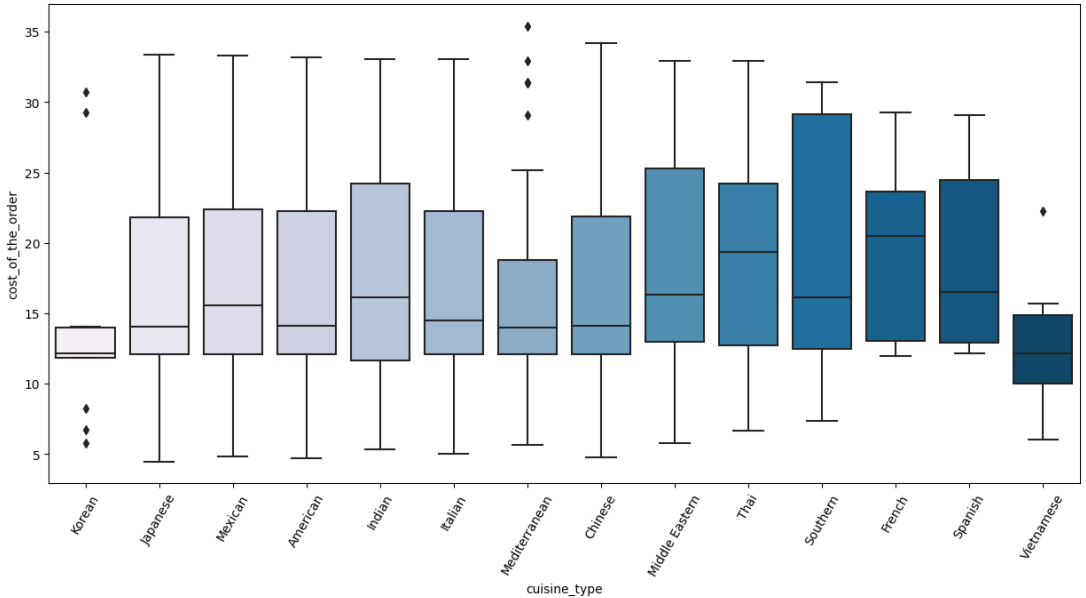
```
df[['customer_id','order_id']].groupby('customer_id').count().sort_values(by = 'order_id',axis = 0, ascending=False).head(3)
```

Out[32]:

customer_id	order_id
52832	13
47440	10
83287	9

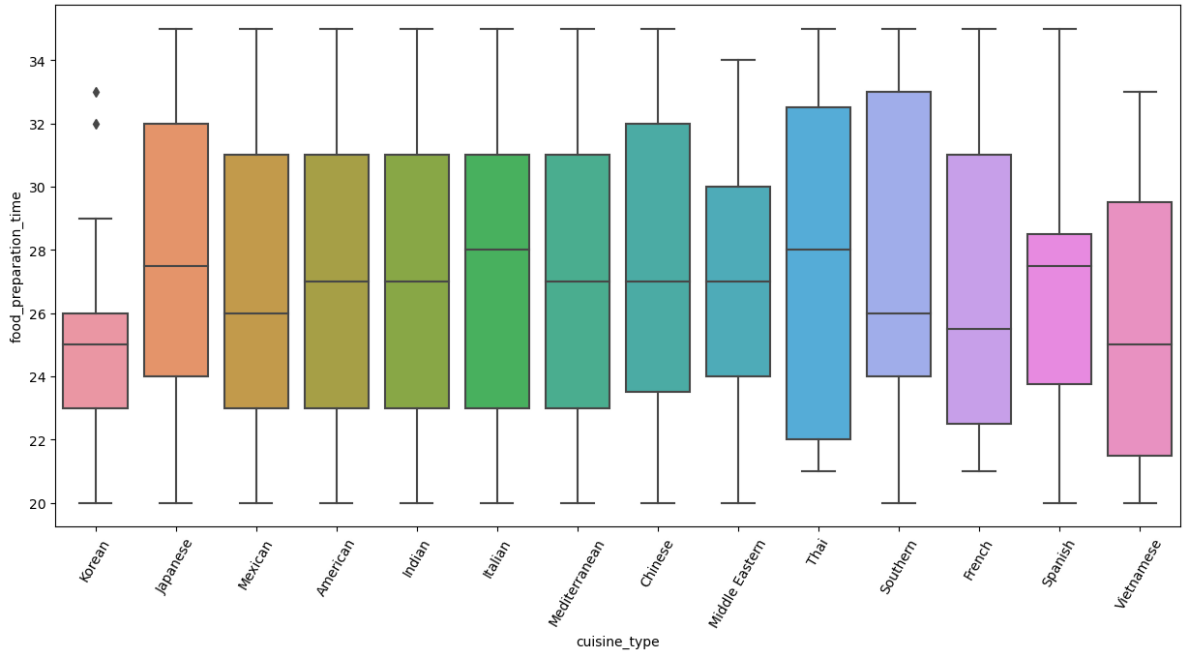
Multivariate Analysis

- Q12: Perform a multivariate analysis to explore relationships between the important variables in the dataset.



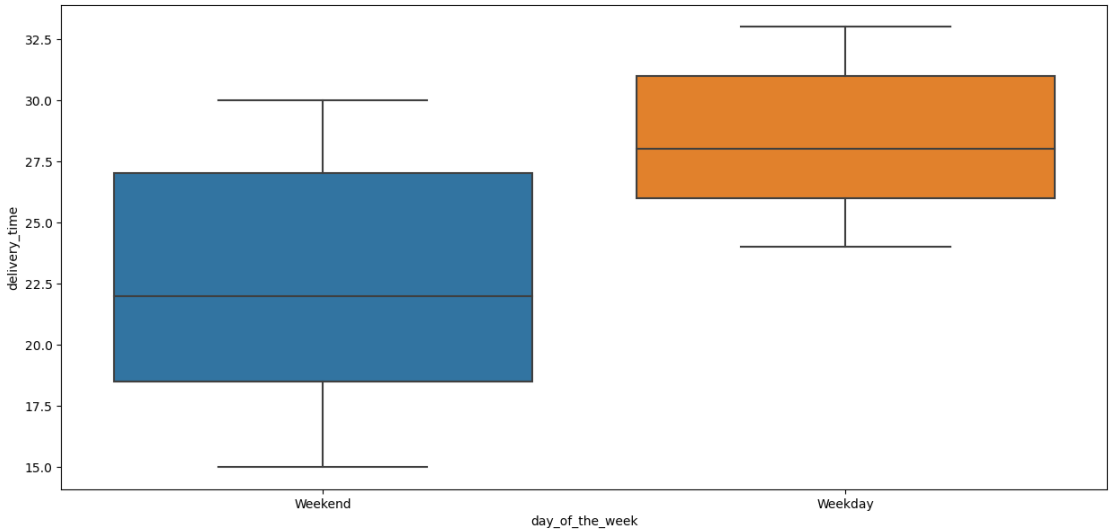
Multivariate Analysis

- Q12: Perform a multivariate analysis to explore relationships between the important variables in the dataset.



Multivariate Analysis

- Q12: Perform a multivariate analysis to explore relationships between the important variables in the dataset.



Multivariate Analysis

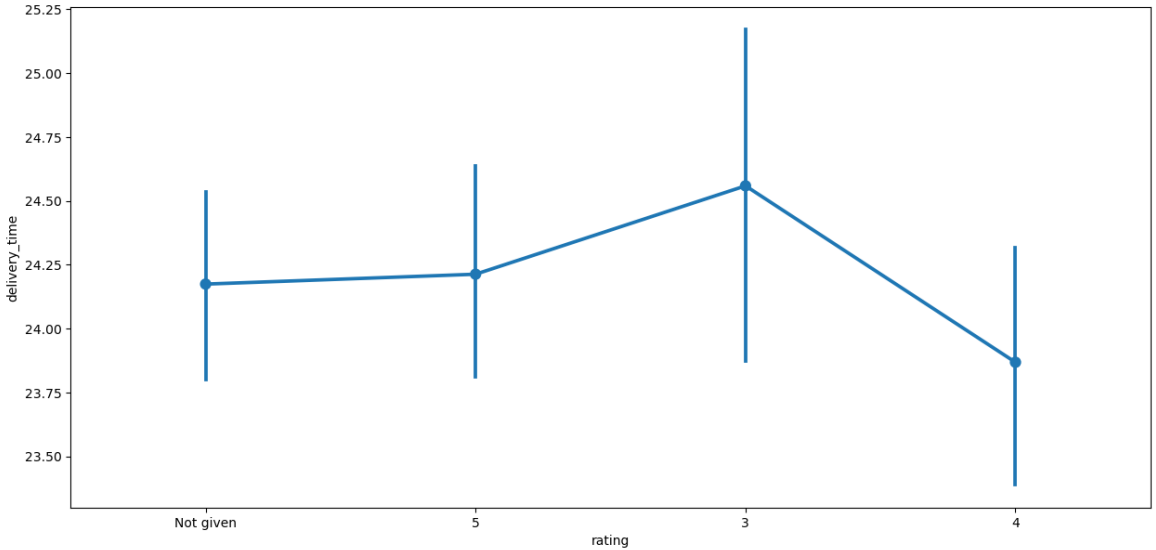
- Q12: Perform a multivariate analysis to explore relationships between the important variables in the dataset.
- Revenue generated

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

```
Out[47]: 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
```

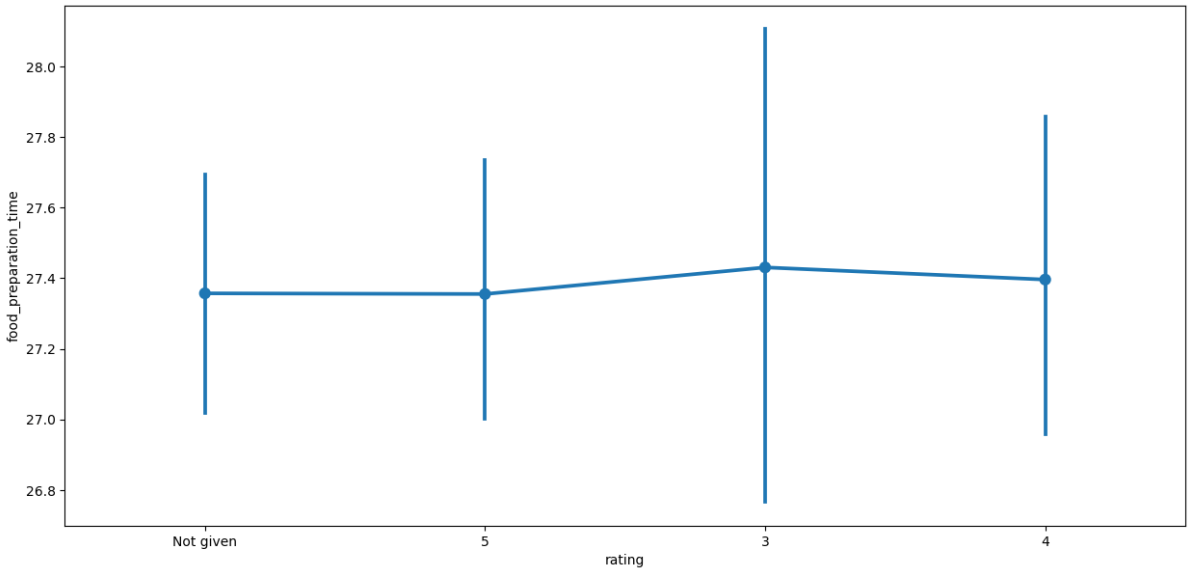
Multivariate Analysis

- Q12: Perform a multivariate analysis to explore relationships between the important variables in the dataset.



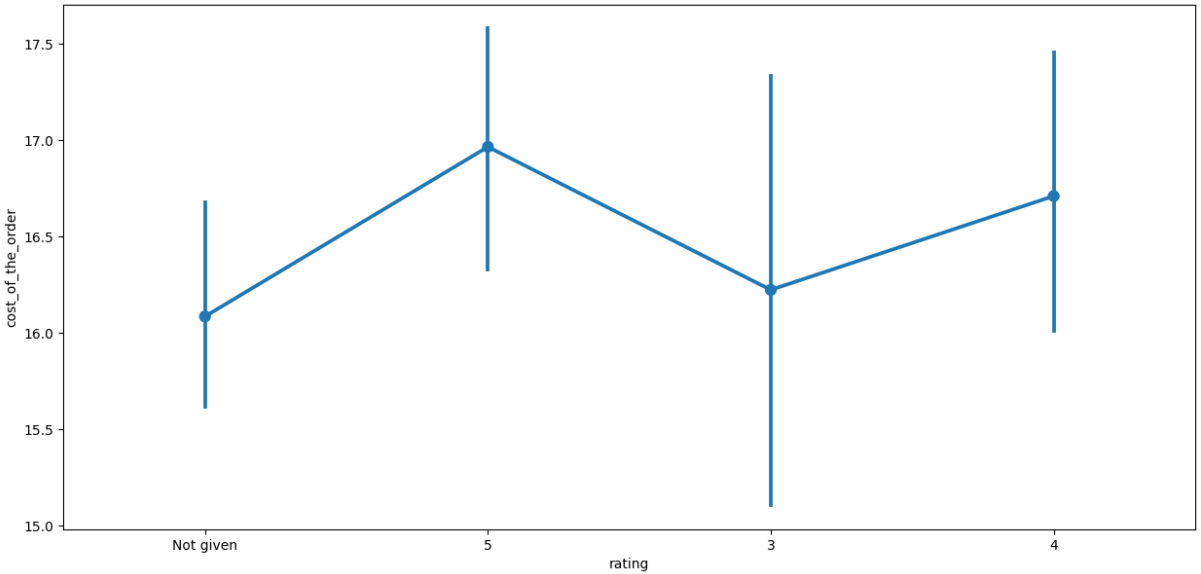
Multivariate Analysis

- Q12: Perform a multivariate analysis to explore relationships between the important variables in the dataset.



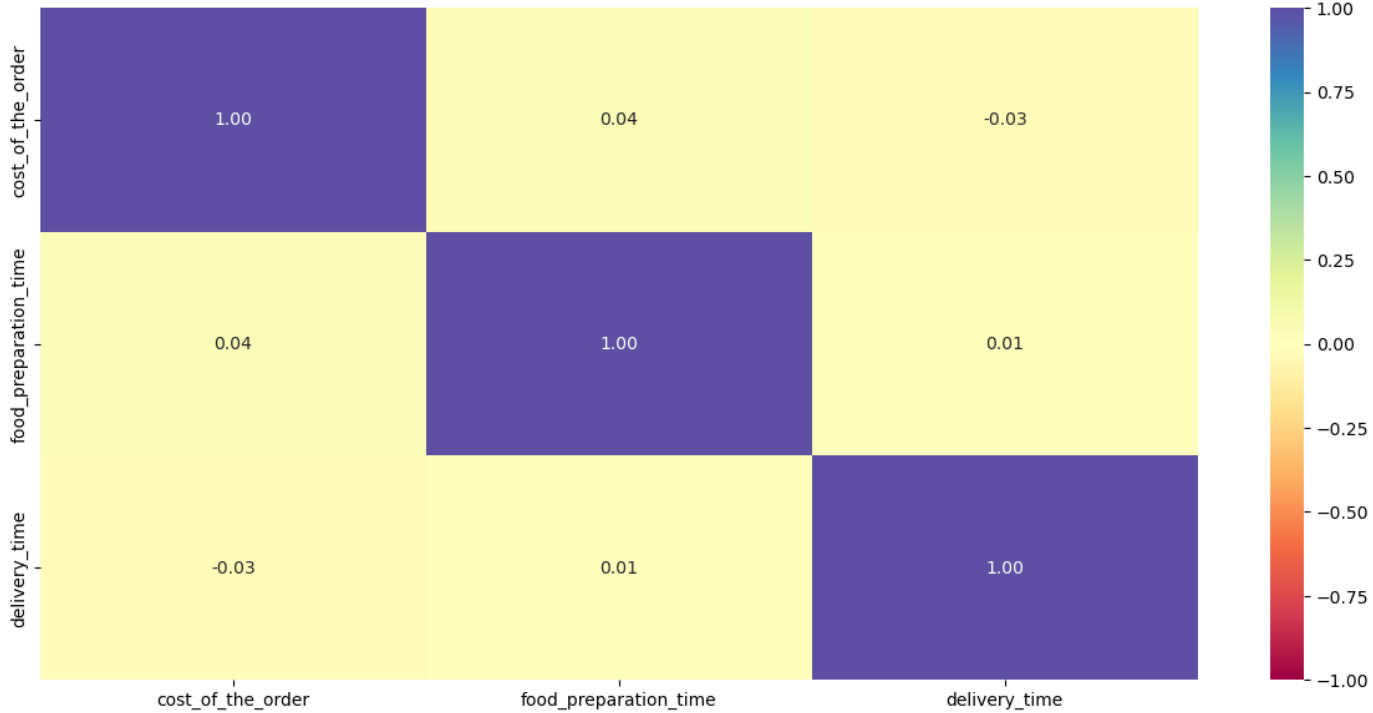
Multivariate Analysis

- Q12: Perform a multivariate analysis to explore relationships between the important variables in the dataset.



Multivariate Analysis

- Q12: Perform a multivariate analysis to explore relationships between the important variables in the dataset.



Multivariate Analysis

- Q13: 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.

```
# 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[48]:

	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

Multivariate Analysis

- Q13: 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.

```
# Get the restaurant names that have rating count more than 50
rest_names = df_rating_count[df_rating_count['rating']>50]['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(df_mean_4['restaurant_name'])['rating'].mean().sort_values(ascending =
False).reset_index().dropna()
```

Out[49]:

	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

Multivariate Analysis

- Q14: 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.

```
#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)
df.head()
```

Multivariate Analysis

- Q14: 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.

Out[50]:

	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

```
# total revenue and print it
total_rev = df['Revenue'].sum()
print('The net revenue is around', round(total_rev, 2), 'dollars')
```

The net revenue is around 6166.3 dollars

Multivariate Analysis

- Q15: 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.)

```
#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']
print(df.head())

df_greater_than_60 = df[df['total_time']>60]

print('The total orders that take more than 60 minutes is:', df_greater_than_60.shape[0])

# Calculate percentage of such orders in the dataset
df_percentage_60 = (df_greater_than_60.shape[0] / df.shape[0]) * 100

print("Percentage of total orders that take more than 60 minutes is:", round(percentage_60, 2), '%')
```

Multivariate Analysis

- Q15: 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.)

```
   order_id  customer_id           restaurant_name cuisine_type  \
0   1477147       337525                   Hangawi       Korean
1   1477685       358141  Blue Ribbon Sushi Izakaya     Japanese
2   1477070        66393               Cafe Habana      Mexican
3   1477334       106968  Blue Ribbon Fried Chicken     American
4   1478249        76942           Dirty Bird to Go     American

   cost_of_the_order day_of_the_week     rating  food_preparation_time  \
0              30.75         Weekend  Not given                     25
1              12.08         Weekend  Not given                     25
2              12.23         Weekday          5                     23
3              29.20         Weekend          3                     25
4              11.59         Weekday          4                     25

   delivery_time  Revenue  Total  total_time
0             20   7.6875     45          45
1             23   1.8120     48          48
2             28   1.8345     51          51
3             15   7.3000     40          40
4             24   1.7385     49          49
The total orders that take more than 60 minutes is: 200
Percentage of total orders that take more than 60 minutes is: 10.54 %
```

The total orders that take more than 60 minutes is: 200
Percentage of total orders that take more than 60 minutes is: 10.54%

Multivariate Analysis

- Q16: 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?

```
# 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')
```

```
# 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')
```

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

Conclusion and Recommendation

- Q17: What are your conclusions from the analysis? What recommendations would you like to share to help improve the business? (You can use cuisine type and feedback ratings to drive your business recommendations.)
- Conclusions
  - No rating is the highest contribution and second rating is 5. It is suspected most satisfied customer did not bother to vote and submit the rating.
  - Cuisine type ordered was consistent regardless weekdays or weekends. American, Japanese, Italian and Chinese food is the top 4 contributor of cuisine type.
  - Weekends order is the highest compared with weekdays. However, delivery time for weekdays is longer than weekends.
  - Highest rating may be associated with faster delivery time regardless cuisine type

Conclusion and Recommendation

- Q17: What are your conclusions from the analysis? What recommendations would you like to share to help improve the business? (You can use cuisine type and feedback ratings to drive your business recommendations.)
- Recommendations
  - Improve customer rating submission, by giving them voucher discount or reward for the next purchase
  - Business owner with the lowest cuisine type may need to improve by giving a food discount for certain amount of time window
  - Cuisine type needs to be more diversify to tackle low cuisine type order
  - Better food promotion or discount need to focus to boost sales on weekday
  - Delivery time at weekdays need to improve to boost restaurant rating
  - Business owner need to improve which total order is more than 60 minutes, by giving customer some food discount for the next purchase as a apologize gesture

Great Learning

Happy Learning !