# FoodHub Presentation

## Python Foundations and DSBA

Date:- 04/23/2023

# Contents / Agenda

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

# Executive Summary

There are some conclusions and recommendations regarding the Foodhub project.

**Conclusions:-**

- There is a persistent popularity content of cuisine over the day of the week although demand is significantly higher over the weekends.
- Preparation time is relatively consistent towards the delivery time and delivery time is significantly variable in total preparation time.
- In case of rating there are many customers who have rated as 5 but it can be possible that they are very much satisfied or it can even be possible that they have not rated the order .So then it comes a question that do unrated people will go to different direction of cuisine or any different app to order the food.

# Recommendations

- As there is a proper consistency in the cuisines across the days of the well I would recommended to focus more on marketing strategies for the weekday to boost sales.

- Improve the customer response on rating their orders try to reduce the rating for not given. A survey can also be conducted as in why the consumers switch to different restaurants or if they are the consistent customers than why do they not rate the order.

- Try to increase customer satisfaction and enable service improvements.

# Business Problem Overview and Solution Approach

❑ Define the Problem & Solution

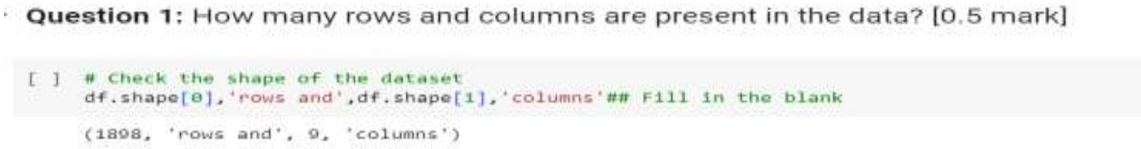
- Foodhub is an aggregator company that offers a access to multiple restaurants as well as different kind of cuisines through a single smartphone app.
- As it takes an order from different restaurants it has access to store the data which are made from different orders by the registered customers.
- The company revenue is based on charging the orders of each restaurant depending on the order price which is nothing but a commission kind of from every restaurant foodhub charges.
- To define the problem the company want to analyze the data and want to understand mainly two things as in whether their business is viable or not.
- Secondly they want to analyze based on the monetary terms as if they are able to monetize or they can improve on what they are focusing at customer point of view as in more customer than more money they can make.

# Data Overview

- The data contains different type of observations related to food order. It includes order id, customer id, ratings given by the customer, restaurant name, cuisine type, cost of order, delivery time, food preparation time.

- It also checks in the data as in the order is placed on weekdays or weekends and what delivery time it was taken from the restaurant to reach to their customer.

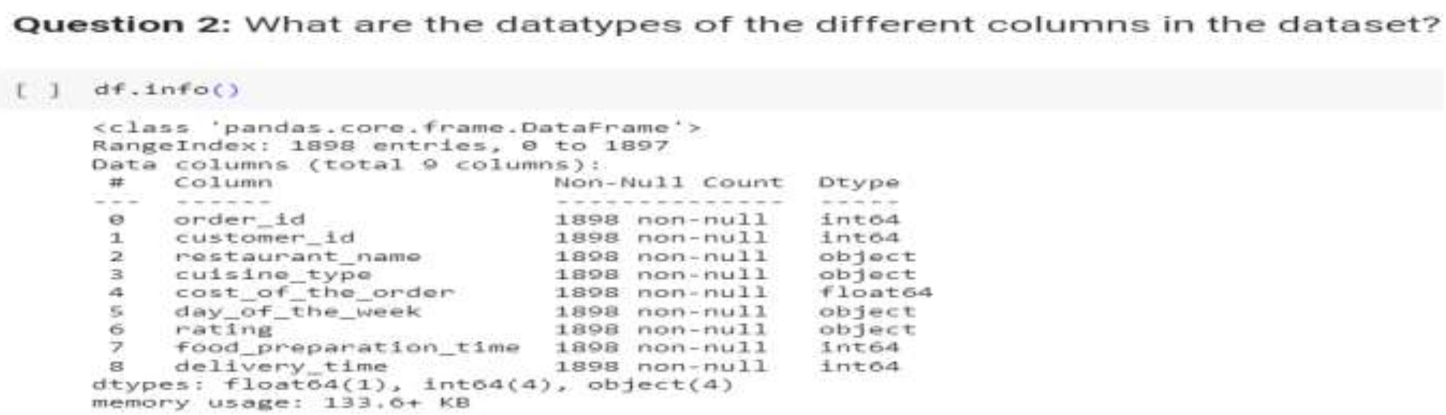
- Lets overview the data and give its observation according the questions.



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

```
[ ]  # Check the shape of the dataset
     df.shape[0],'rows and',df.shape[1],'columns'## Fill in the blank

     (1898, 'rows and', 9, 'columns')
```

➢ From question 1 I have observed that we have 1898 rows and 9 columns in the data for all order processed.

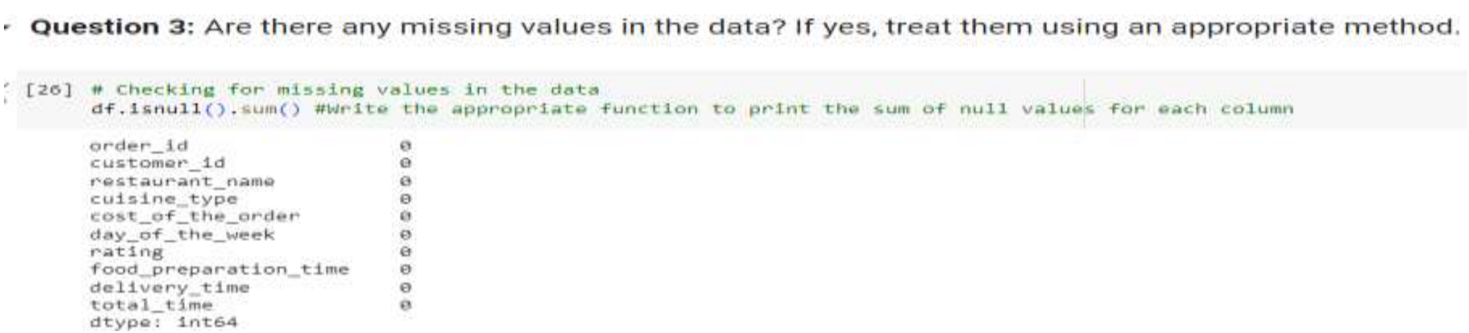


Question 2: What are the datatypes of the different columns in the dataset?

```
[ ]  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
```

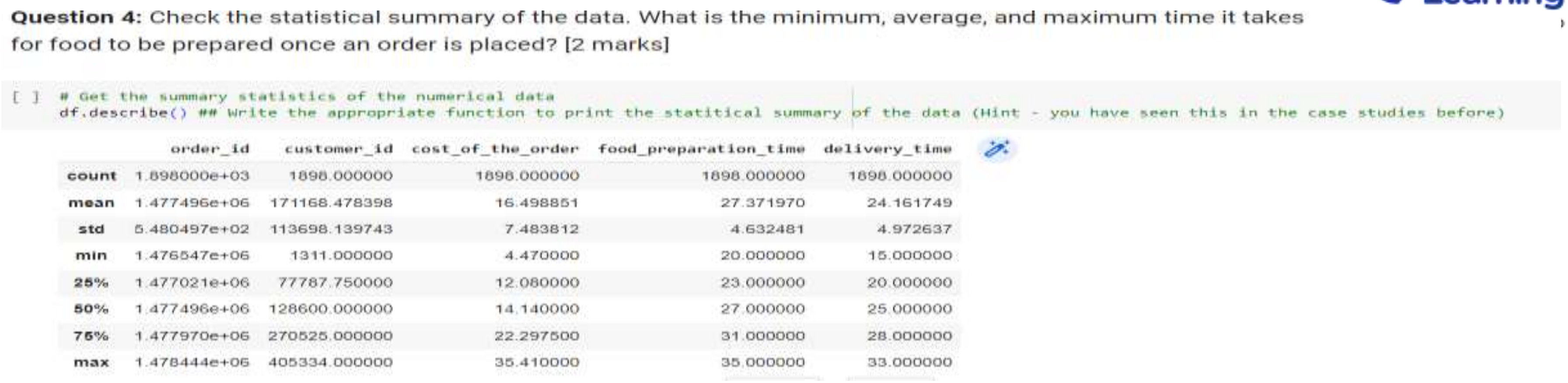
- Here we can observe that we have 1 float data type, 4 integer data type, and 4 object data type. From this we can see that rating column has wrong data type. We need to modify them accordingly. To enable the change we can reassign the Not given rating to the number and this will help us to replace rating column to a numerical value.



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

```
[26] # Checking for missing values in the data
     df.isnull().sum() #Write the appropriate function to print the sum of null values for 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
total_time                0
dtype: int64
```

- ➢ We do not have any missing values in the data, hence it requires not further treatment with any kind of appropriate method.

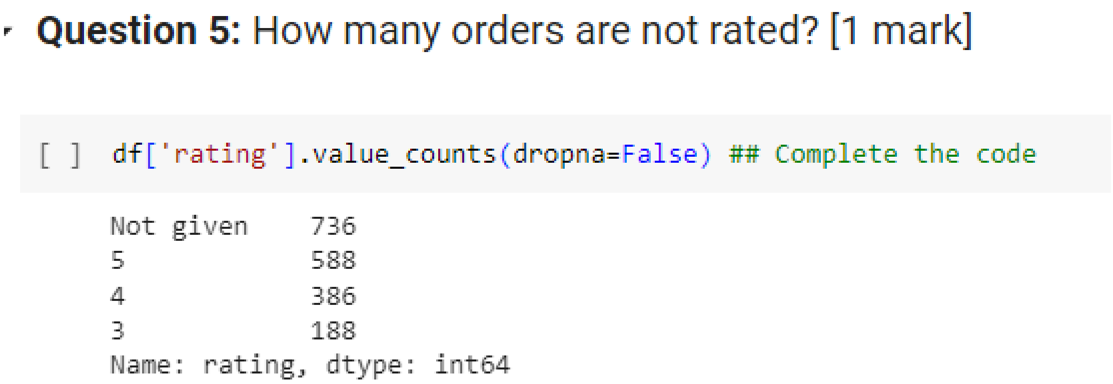


**Question 4:** Check the statistical summary of the data. What is the minimum, average, and maximum time it takes for food to be prepared once an order is placed? [2 marks]

```
[ ]  # Get the summary statistics of the numerical data
     df.describe() ## Write the appropriate function to print the statitical summary of the data (Hint - you have seen this in the case studies before)
```

|  | order_id | customer_id | cost_of_the_order | food_preparation_time | delivery_time |
|---|---|---|---|---|---|
| **count** | 1.898000e+03 | 1898.000000 | 1898.000000 | 1898.000000 | 1898.000000 |
| **mean** | 1.477496e+06 | 171168.478398 | 16.498851 | 27.371970 | 24.161749 |
| **std** | 5.480497e+02 | 113698.139743 | 7.483812 | 4.632481 | 4.972637 |
| **min** | 1.476547e+06 | 1311.000000 | 4.470000 | 20.000000 | 15.000000 |
| **25%** | 1.477021e+06 | 77787.750000 | 12.080000 | 23.000000 | 20.000000 |
| **50%** | 1.477496e+06 | 128600.000000 | 14.140000 | 27.000000 | 25.000000 |
| **75%** | 1.477970e+06 | 270525.000000 | 22.297500 | 31.000000 | 28.000000 |
| **max** | 1.478444e+06 | 405334.000000 | 35.410000 | 35.000000 | 33.000000 |

- We have total 1898 observations and from that we can see that 25% of the cost has 12.08 and has no rating and it take 23 minutes to prepare the food and 20 minutes to deliver the food, 50% of the observations has values as 14.14, 27, And 25 for the cost, food preparation, and delivery time. For the 75% we have 22,31,and 28 for same parameters. **From all this it takes min 20 minutes on a mean of 27.37 minutes and max 35 minutes to prepare the food once an order is placed.**



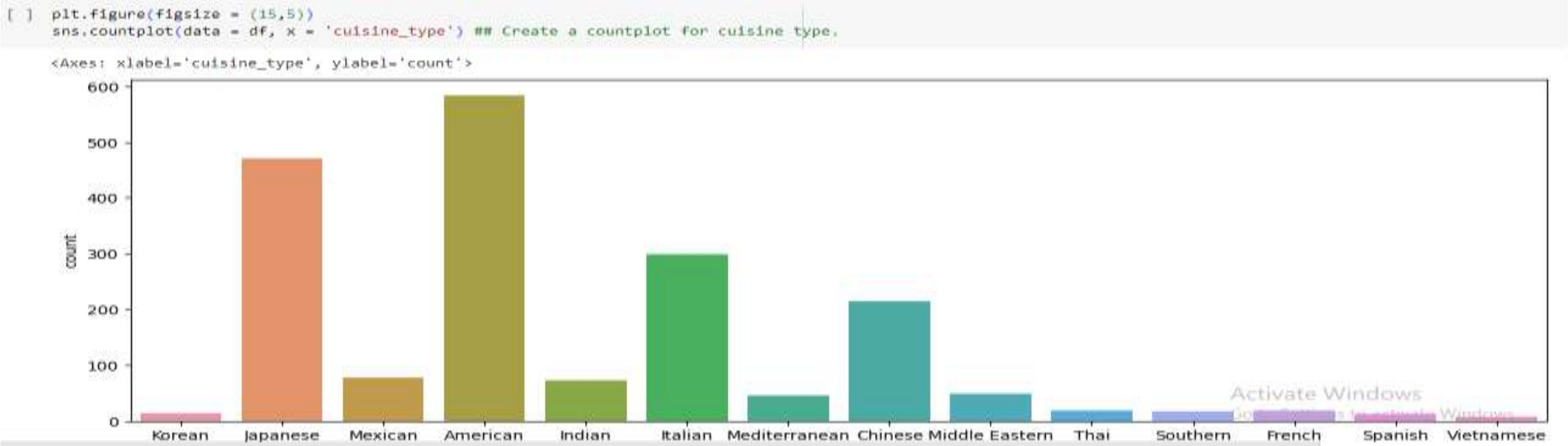
**Question 5:** How many orders are not rated? [1 mark]

```
[ ]  df['rating'].value_counts(dropna=False) ## Complete the code
```

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

- As we can see here that the orders which are not rated are specified as ’Not Given’ one which are about 736 orders are not rated.

# Univariate Analysis

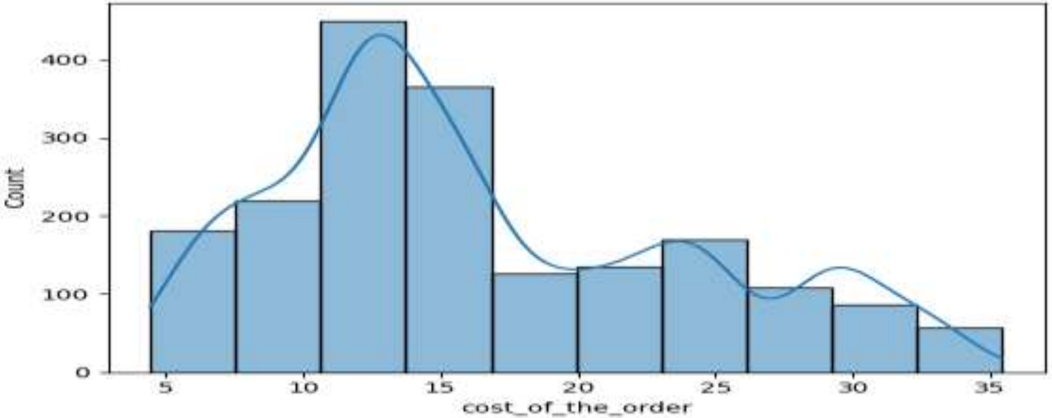


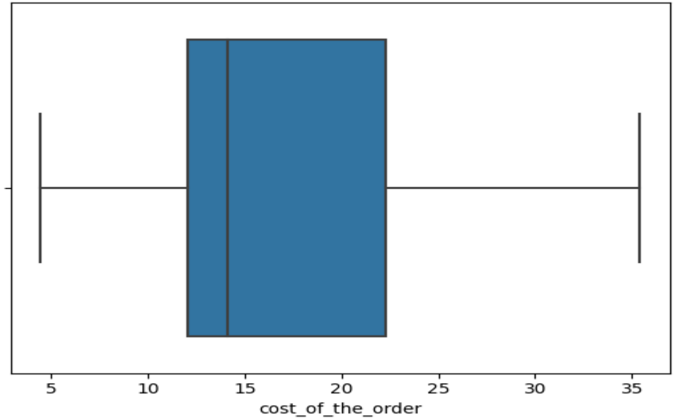
```
[ ]  plt.figure(figsize = (15,5))
     sns.countplot(data = df, x = 'cuisine_type') ## Create a countplot for cuisine type.
```

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

- ➢ From the following results we can conclude that the most popular cuisines are American, Japanese, Italian, and Chinese with a close connection or tie with Mexican and Indian.

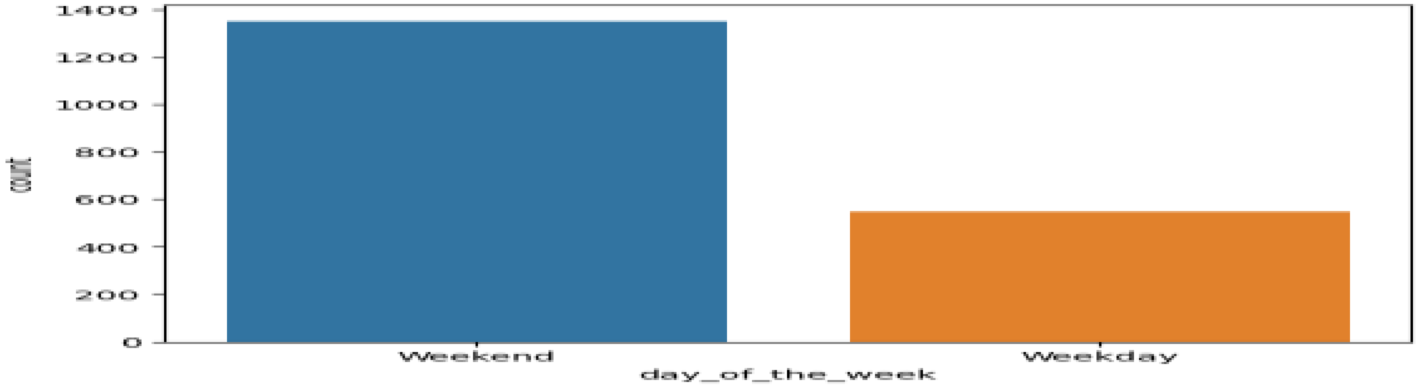
## Observations on Cost Of Order





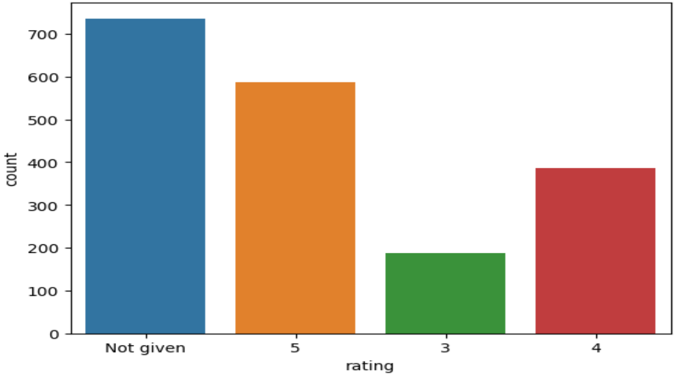
- ➢ The histplot is skewed to the left so we can conclude from that there is a slight peak at 25 dollars and more is towards lower cost. However the box plot has the median of 14 which can be considered as the orders are being right skewed. These are the observations for the cost of orders.

# Observations on Day of the week



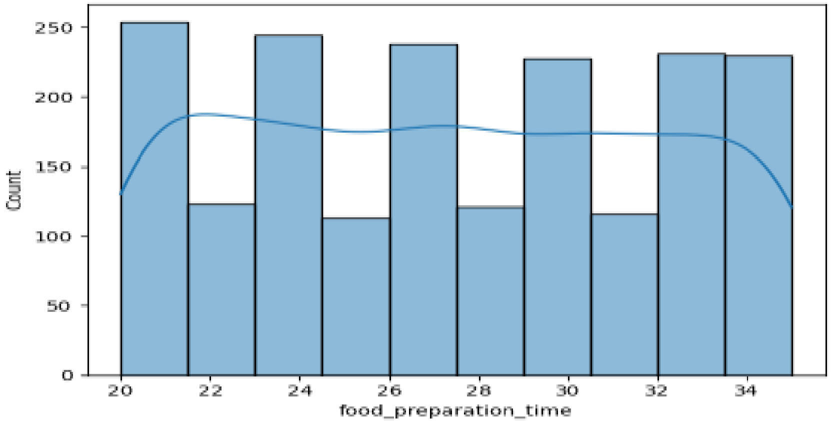
- From this we can observe that highest number of orders are being placed and prepared on weekend over the weekdays may  be considered a reason could be the highest promotional offers or any discount or people would consider holiday on weekends.

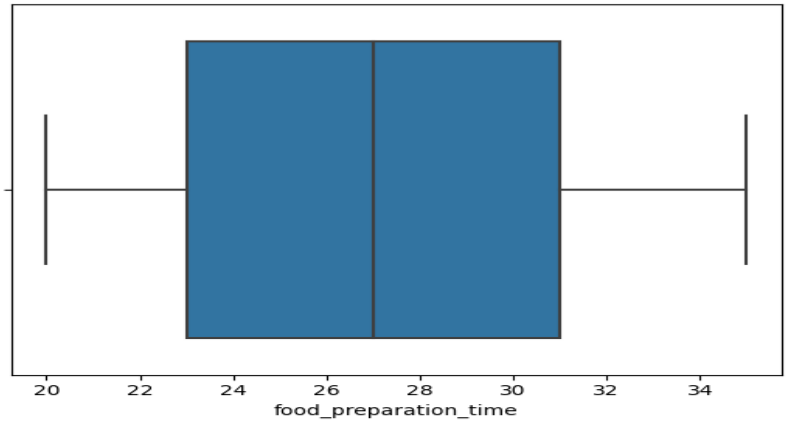
# Observation on Rating



- From the bar graph we can conclude that many of the customers have not given the rating after ordering the food and maximum rating given is 5 by the consumers and minimum is 3. So this can be considered an average app for the consumers for placing order and work on it.

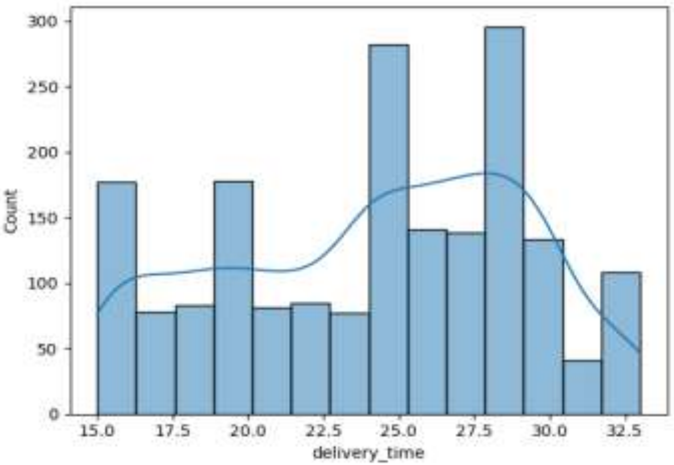
# Observation on Food preparation time

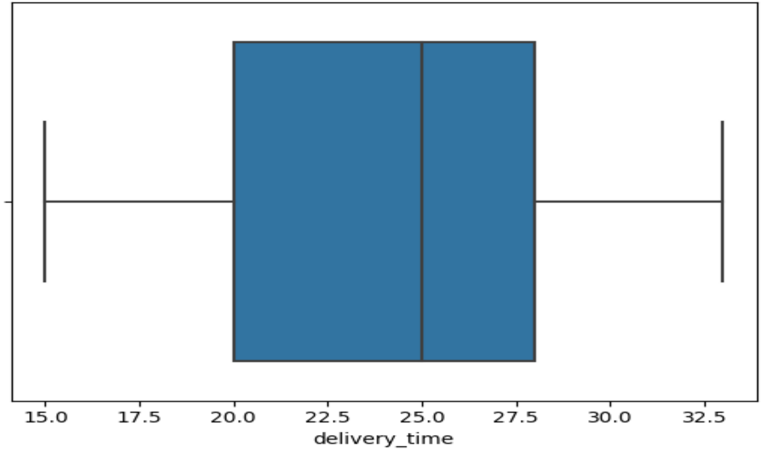




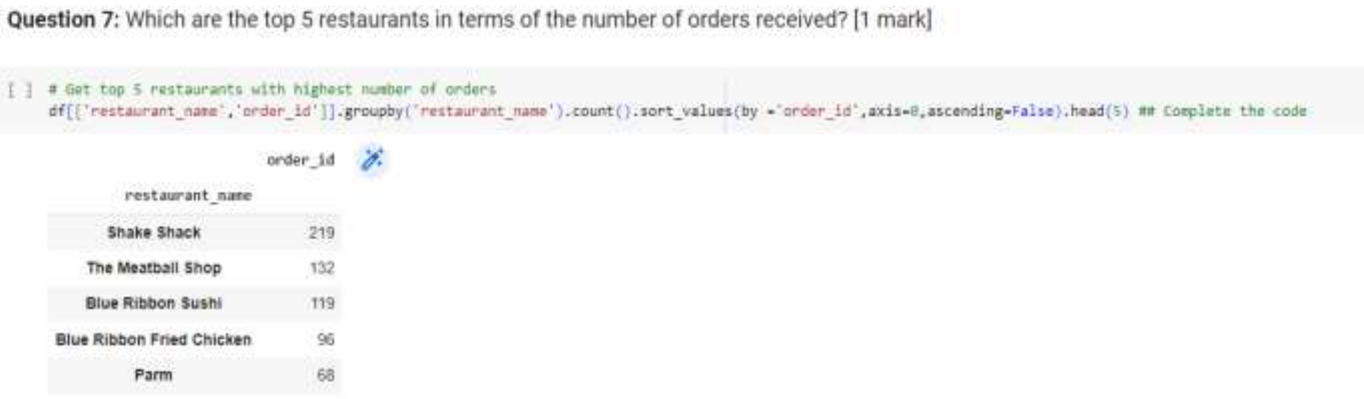
- From the above graphs we can observe that it has a normal distribution of time for preparing the food where normal it takes 20 minutes minimum time to max 34 minutes with a median of 27 minutes.

# Observations on Delivery Time





- From the above graphs we can conclude that the median delivery time is about 25 minutes with skewed to left while most orders takes 25 to 28 minutes for delivery of the food.



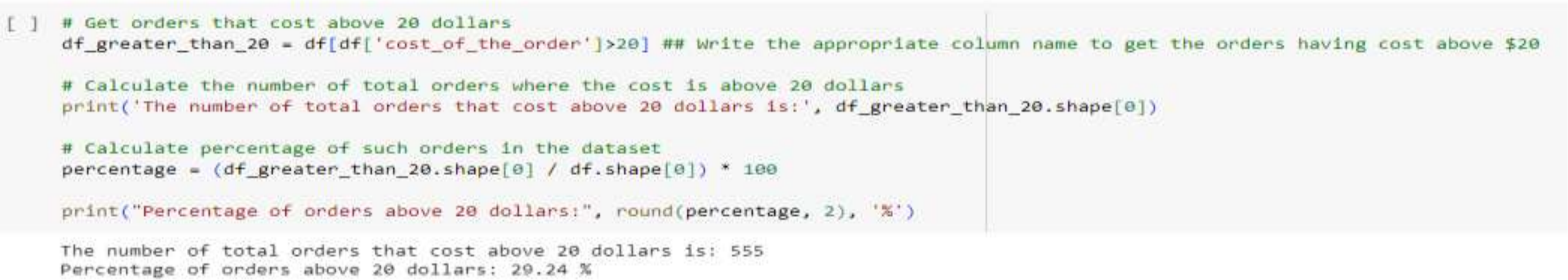
**Question 7:** Which are the top 5 restaurants in terms of the number of orders received? [1 mark]

```
[ ]  # Get top 5 restaurants with highest number of orders
     df[['restaurant_name','order_id']].groupby('restaurant_name').count().sort_values(by ='order_id',axis=0,ascending=False).head(5) ## Complete the code
```

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

- ➢ From the data we can observe that Shake Shack is the leading restaurant with 219 orders and other restaurants such as The Meatball, Blue Ribbon Sushi, Blue Ribbon Fired Chicken, and Parm have orders such as 132, 119, 96,and 68 which collectively makes top 5 restaurants for the orders placed.
- ➢ The most popular cuisine is American with a count of 415 orders.

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

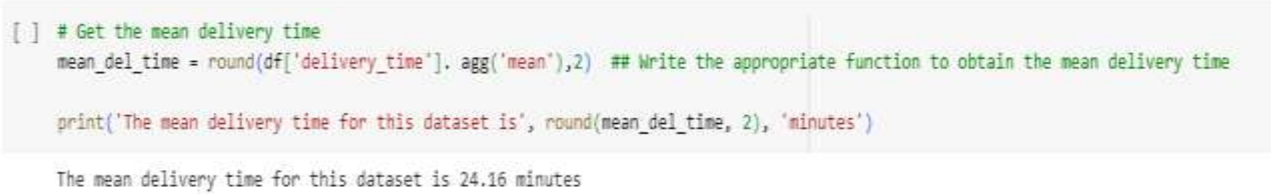


```
[ ]  # Get orders that cost above 20 dollars
     df_greater_than_20 = df[df['cost_of_the_order']>20] ## Write the appropriate column name to get the orders having cost above $20

     # 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), '%')
```

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

- ➢ We can observe here that 29.24% of the orders cost more than 20 dollars.

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



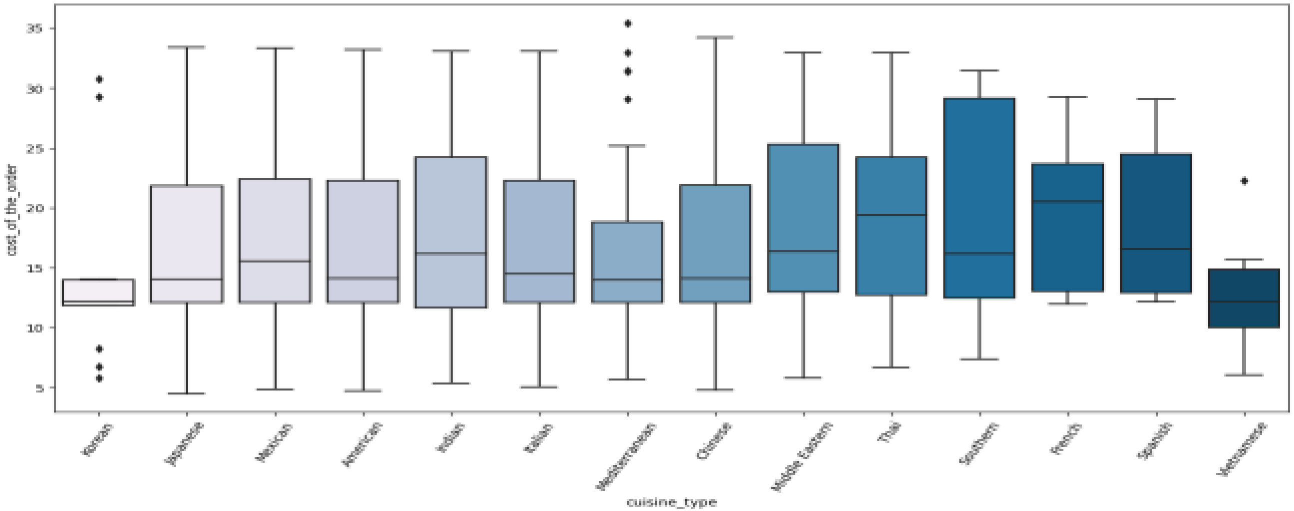
```
[ ]  # Get the mean delivery time
     mean_del_time = round(df['delivery_time']. agg('mean'),2)  ## Write the appropriate function to obtain the mean delivery time

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

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

- ➢ We can observe here that the mean of order delivery time is about 24.16 minutes.

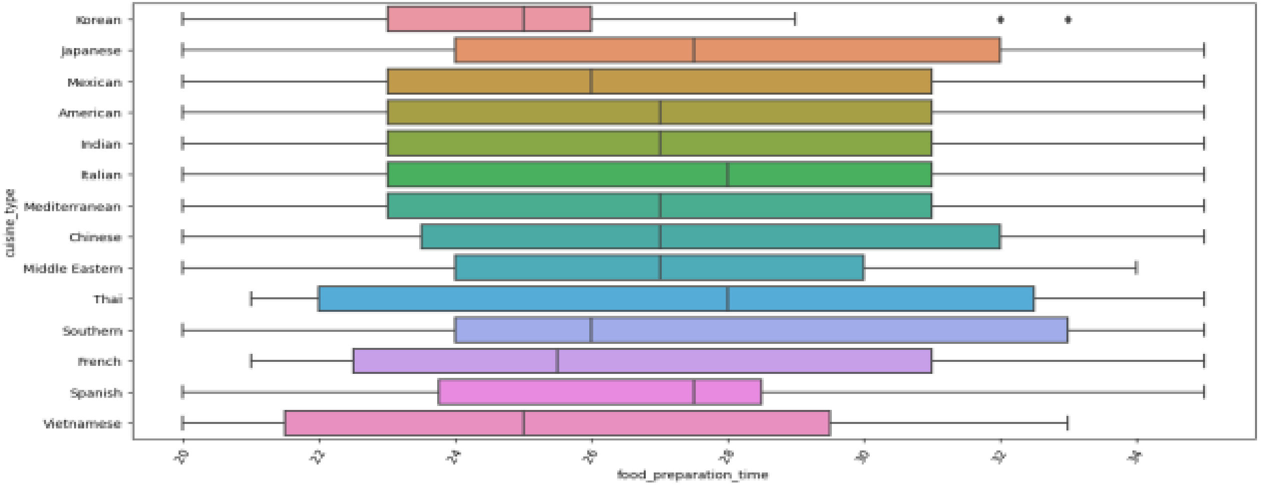
# Multivariate Analysis

Observations on Cuisine v/s Cost of the order:-



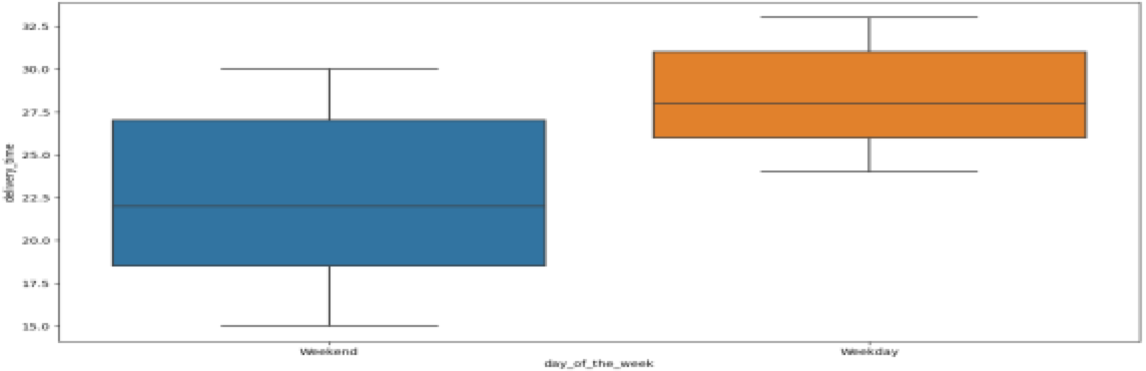
- ➢ From the above graph we can observe here that most of the cuisine have assorted degrees of skewed cost and some of the cuisine have outliners such as Korean, Mediterranean, and Vietnamese.

# Cuisine V/S Food Preparation Time



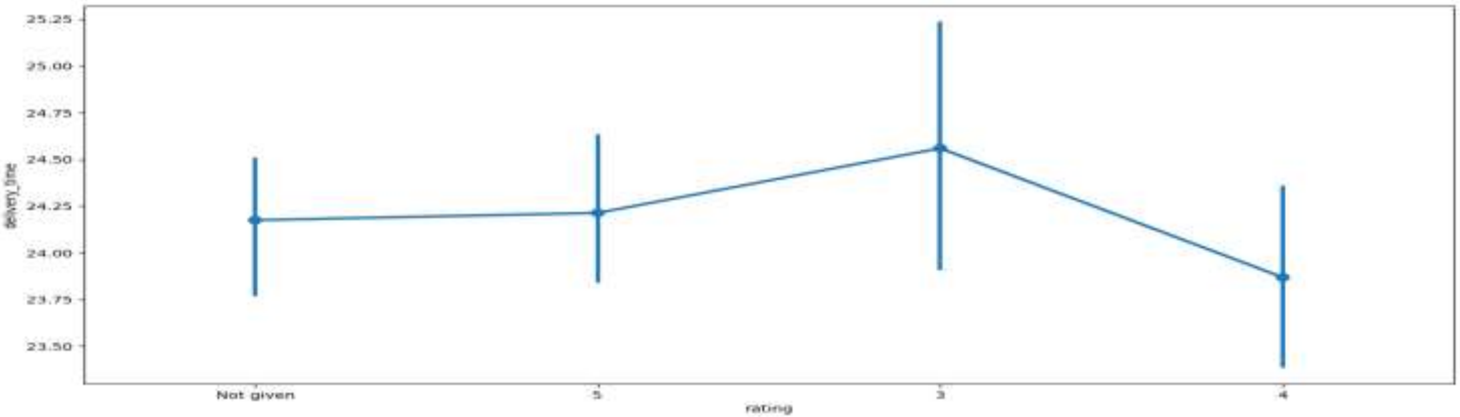
- ➢ We can observe here that some of the cuisine has more or less same average preparation time while some have larger spread with a 95% of confidence interval.

# Day of the Week v/s Delivery Time



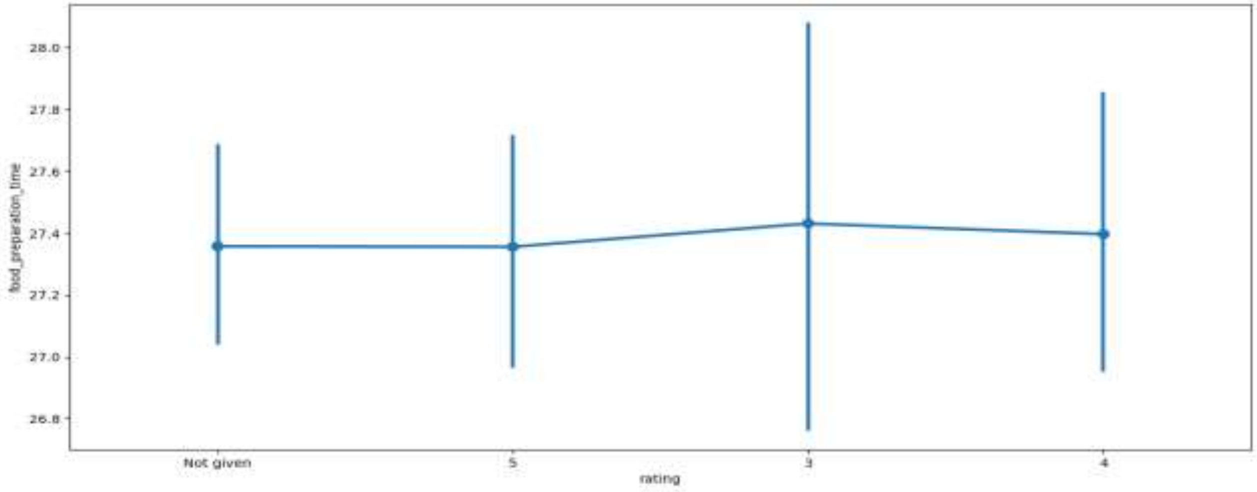
- The median delivery time for weekends is about 22.5 minutes while on weekday is about 28.5 minutes. So here we can observe that in weekdays it takes longer time as compared to weekends in delivery of food to customers for the respective restaurants.

# Rating v/s Delivery time



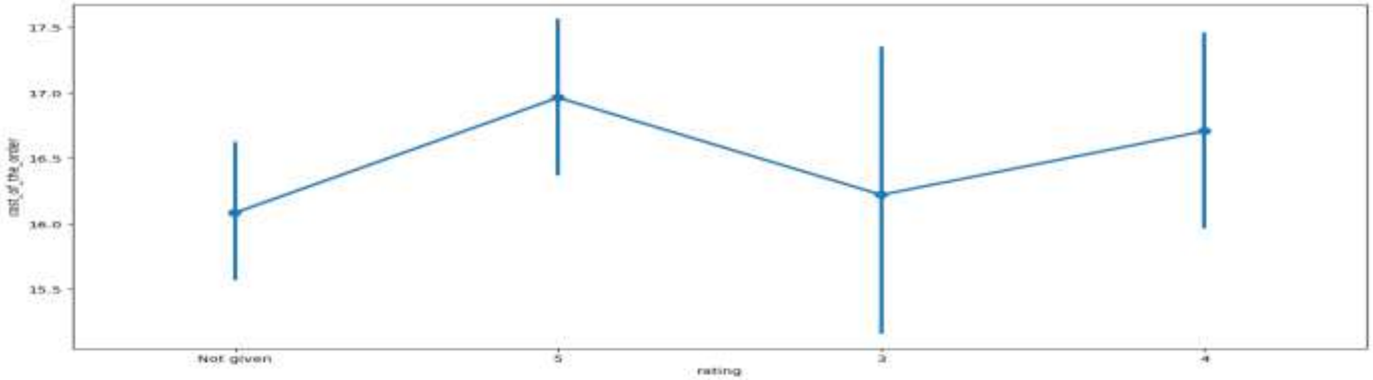
- Here we can observe that when delivery time take about 24 minutes approx. the rating is either given is 5 or no rating is provided. When the rating given is 3 which takes approx. on an average of 25 minutes and when it takes less time it gives rating upto 4. So we can conclude that it's a direct relation between rating and delivery time.

# Rating v/s Food preparation time



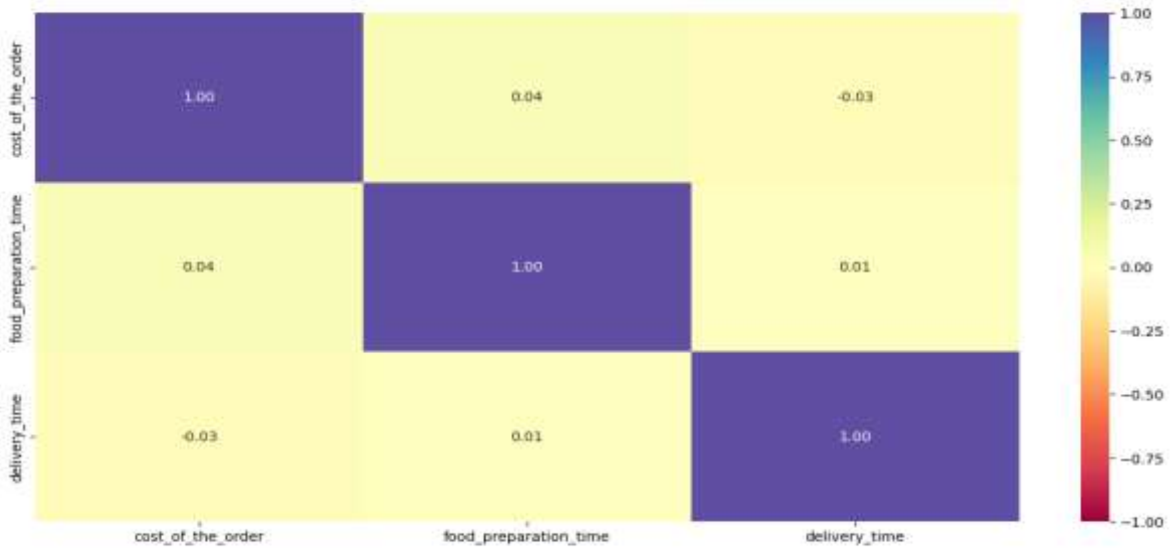
- We can observe here that there is not much difference between the rating and food preparation time as it takes the same observation where it was in delivery time or food preparation time.

# Rating v/s Cost of Order

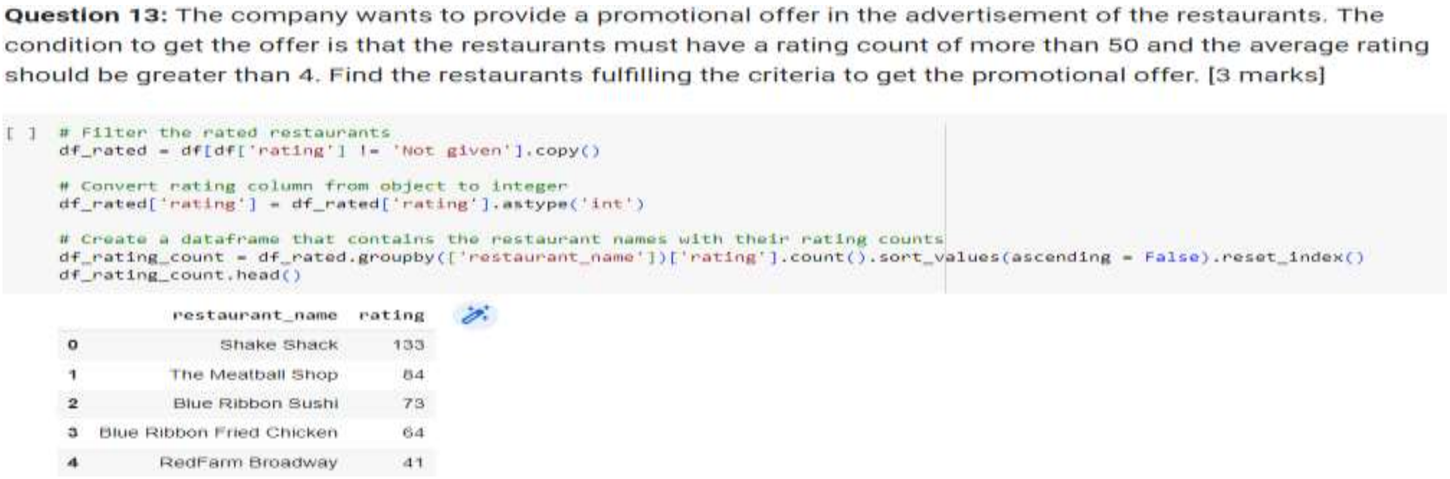


- ➢ From the above graph we can observe that when the cost of order is high the rating provided is 5 while when the cost of order is average than rating of 3 is provided and at max 3 rating is provided when the cost of order is on an average of 17.

# Heatmap of cost of order, food preparation time, &delivery time



- We can observe here as heatmap shows a weak correlation between all the data points.

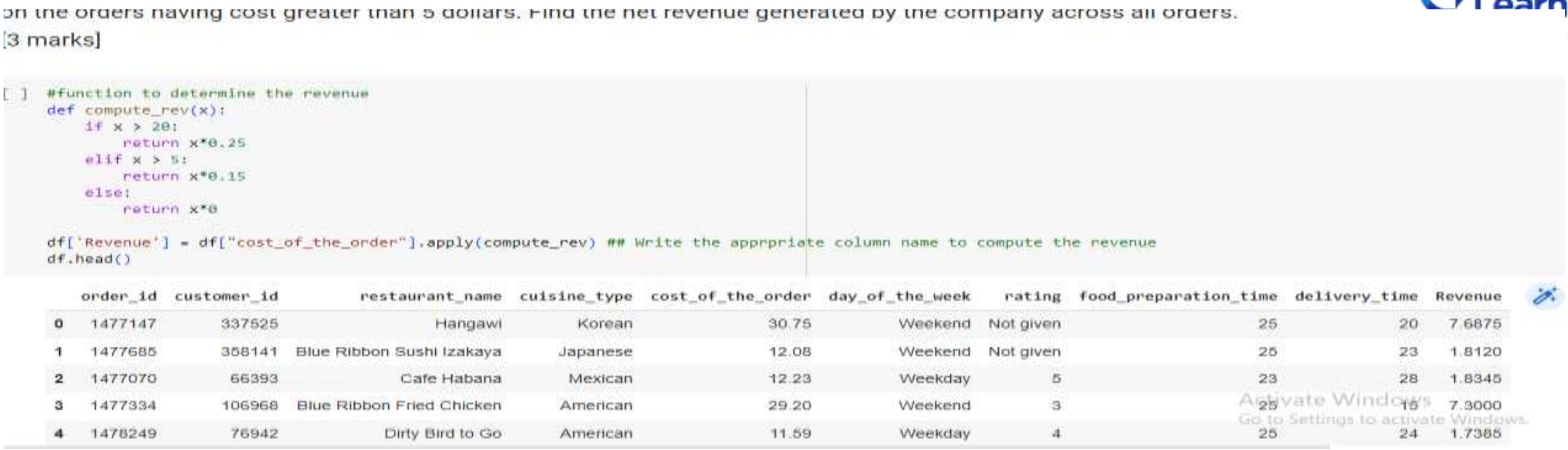


**Question 13:** The company wants to provide a promotional offer in the advertisement of the restaurants. The condition to get the offer is that the restaurants must have a rating count of more than 50 and the average rating should be greater than 4. Find the restaurants fulfilling the criteria to get 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()
```

|   | 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 |

- We can observe here that the restaurants named Shake Shack has high rating of 133 rating and even if we compare related to all other restaurants they are competent for the promotional offers.

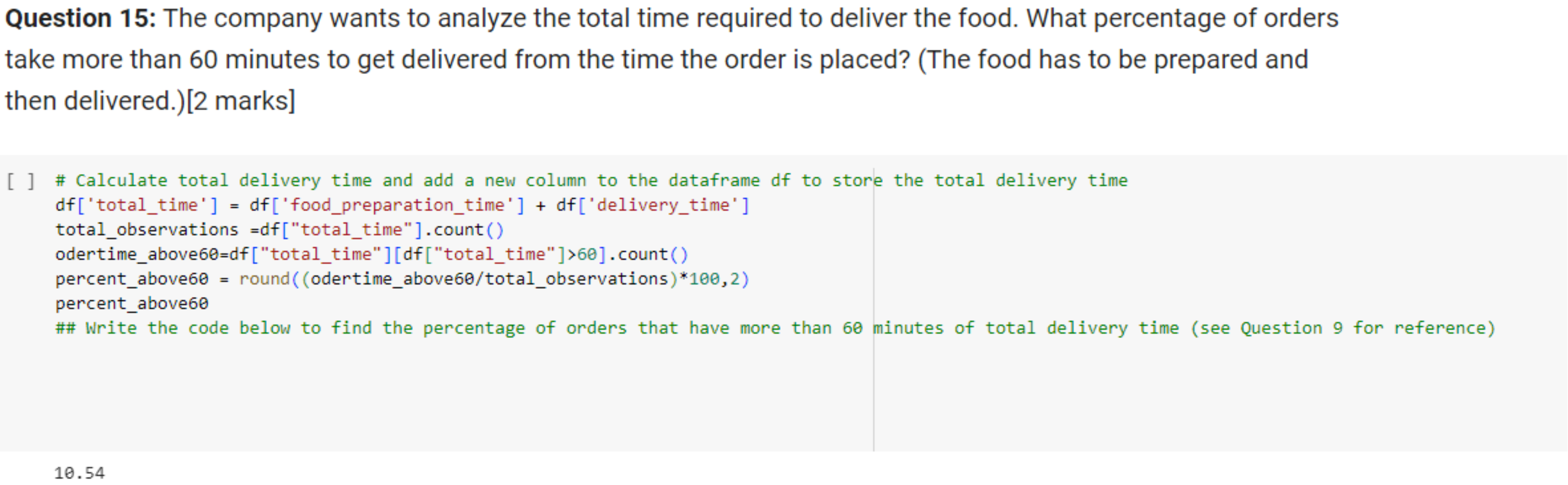


on the orders having cost greater than 5 dollars. Find the net revenue generated by the company across all orders.
[3 marks]

```
[ ] #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()
```

| | 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 |

- ➢ We observed here as in the net revenue for the restaurant Hangawi has the highest revenue of 7.68 on weekends and even though rating is not provided to that restaurant so it means that people prefer more Korean cuisine type.



**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]

```
[ ] # 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']
    total_observations =df["total_time"].count()
    odertime_above60=df["total_time"][df["total_time"]>60].count()
    percent_above60 = round((odertime_above60/total_observations)*100,2)
    percent_above60
    ## 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)
```

```
10.54
```

- We observed that about 10.54% of orders take more than 60 minutes to get delivered from the time the order is placed.
- The mean delivery time on weekdays is around 28 minutes and the mean delivery time on weekends is around 23 minutes.

Happy Learning !