Project Foundations for Data Science: FoodHub Data Analysis

Marks: 40

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 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 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: 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

Let us start by importing the required libraries

In [ ]:
```
# import libraries for data manipulation
import numpy as np
import pandas as pd

# import libraries for data visualization
import matplotlib.pyplot as plt
import seaborn as sns
```

Understanding the structure of the data

In [ ]:
```
from google.colab import drive
drive.mount("/content/drive", force_remount=True)
```

```
Mounted at /content/drive
```

In [ ]:
```
# read the data
df=pd.read_csv('foodhub_order.csv')
# returns the first 5 rows
df.head()
```

Out[ ]:

	order_id	customer_id	restaurant_name	cuisine_type	cost_of_the_order	day_of_the_week	rating	f
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	

Observations:

The DataFrame has 9 columns as mentioned in the Data Dictionary. Data in each row corresponds to the order placed by a customer.

Question 1: How many rows and columns are present in the data?

In [ ]:
```
count=df.shape # provides column and row structure of data#
print(count)
### There are 1898 rows and 9 columns####
```

```
(1898, 9)
```

Observations: There are 1898 rows and 9 columns present.

Question 2: What are the datatypes of the different columns in the dataset? (The info() function can be used)

In [ ]:
```
# Use info() to print a concise summary of the DataFrame
datatypes=df.info()
print(datatypes)
```

```
<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
None
```

Observations: The customer_id and order_id columns are presently set as integers. This may want to be changed to an object long term to create letter flexibility in id assignment.

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

In [ ]:
```
# The previous info() code lets us know that there are no missing values as we know th
print(df.isna().sum())
print()


isthere_null=pd.isnull(df)
count_null=isthere_null.sum()
print(count_null)
```

```
print()

print(df[df['rating']=='0']) ###verify that there are no 0's in the ratings column
print(df[df['food_preparation_time']==0]) ###verify that there are no 0's in the food
print(df[df['delivery_time']==0]) ###verify that there are no 0's in the delivery time
```

```
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

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

Empty DataFrame
Columns: [order_id, customer_id, restaurant_name, cuisine_type, cost_of_the_order, da
y_of_the_week, rating, food_preparation_time, delivery_time]
Index: []
Empty DataFrame
Columns: [order_id, customer_id, restaurant_name, cuisine_type, cost_of_the_order, da
y_of_the_week, rating, food_preparation_time, delivery_time]
Index: []
Empty DataFrame
Columns: [order_id, customer_id, restaurant_name, cuisine_type, cost_of_the_order, da
y_of_the_week, rating, food_preparation_time, delivery_time]
Index: []
```

Observations: There are no missing values within the data

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?

In [ ]:
```
print(df["food_preparation_time"].describe())  #statistical data for only 'food_prepar
```

```
count    1898.000000
mean       27.371970
std         4.632481
min        20.000000
25%        23.000000
50%        27.000000
75%        31.000000
max        35.000000
Name: food_preparation_time, dtype: float64
```

Observations: The minimum time for food to be prepared is 20 minutes. The average tme for food to be prepared is 27.37 minutes. The maximum time for food to be prepared is 35 minutes.

Question 5: How many orders are not rated?

In [ ]:
```
df['rating'].value_counts()['Not given']
```

Out[ ]:
```
736
```

Observations:

There are 736 orders with the data 'Not Rated' in the ratings column

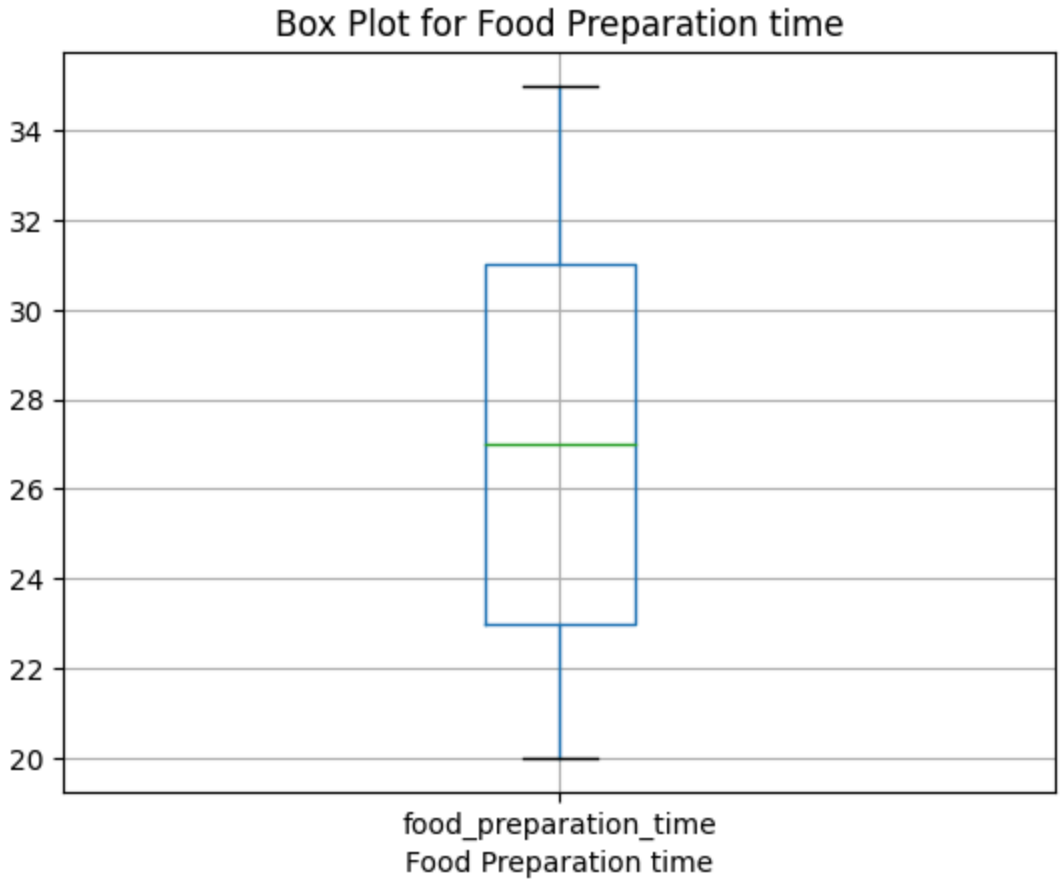
Exploratory Data Analysis (EDA)

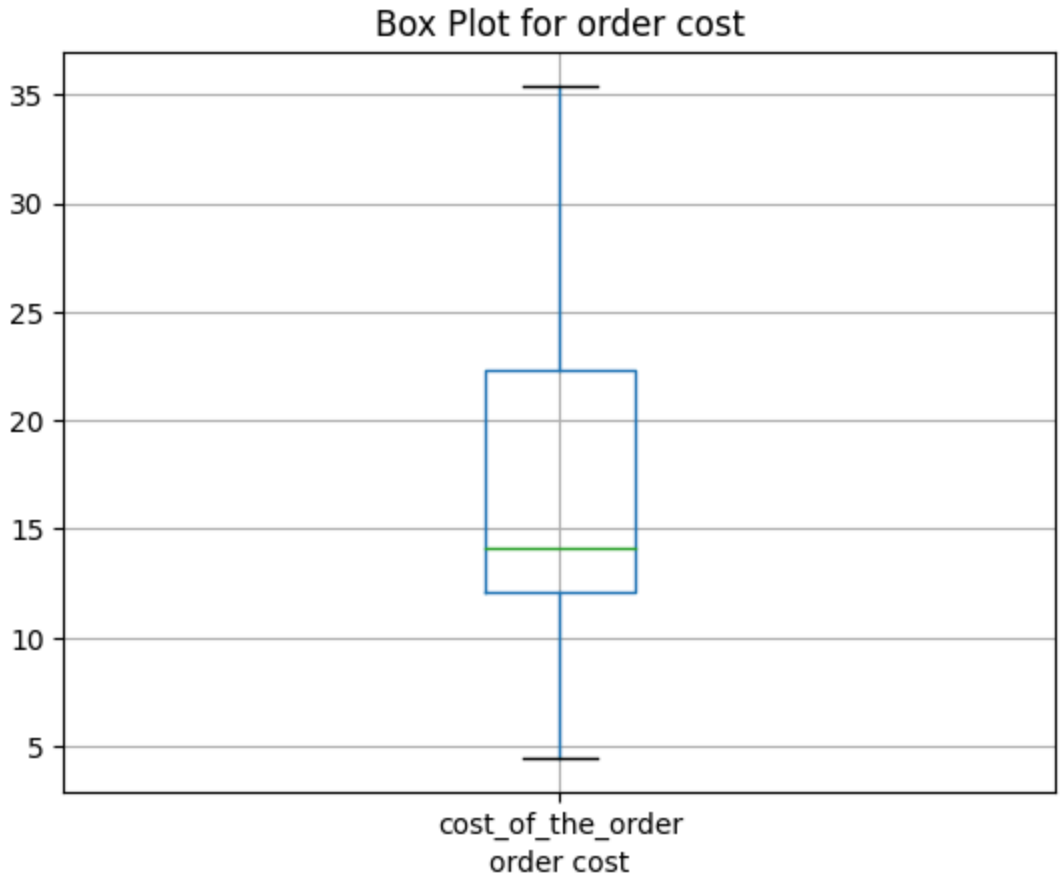
Univariate Analysis

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

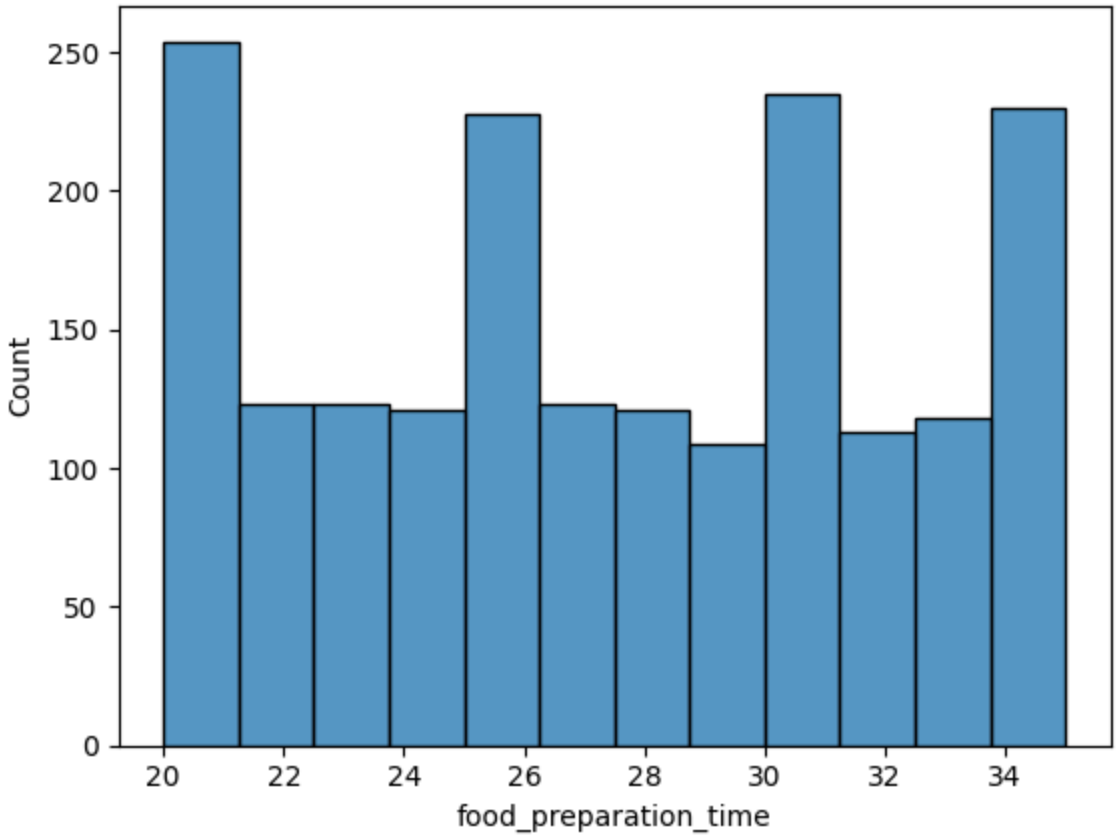
In [ ]:
```
plt.figure()
plt.title('Box Plot for Food Preparation time')
plt.xlabel('Food Preparation time')
df.boxplot(column='food_preparation_time')




plt.figure()
plt.title('Box Plot for order cost')
plt.xlabel('order cost')
df.boxplot(column='cost_of_the_order')

plt.figure()
plt.title('Box Plot for delivery time')
plt.xlabel('delivery time in minutes')
df.boxplot(column='delivery_time')
plt.show()

sns.histplot(data=df, x='food_preparation_time')
plt.show()
sns.histplot(data=df, x='cost_of_the_order')
plt.show()
sns.histplot(data=df, x='delivery_time')
plt.show()

sns.histplot(data=df, x='day_of_the_week')
plt.show()

plt.figure(figsize=(20,30))
```

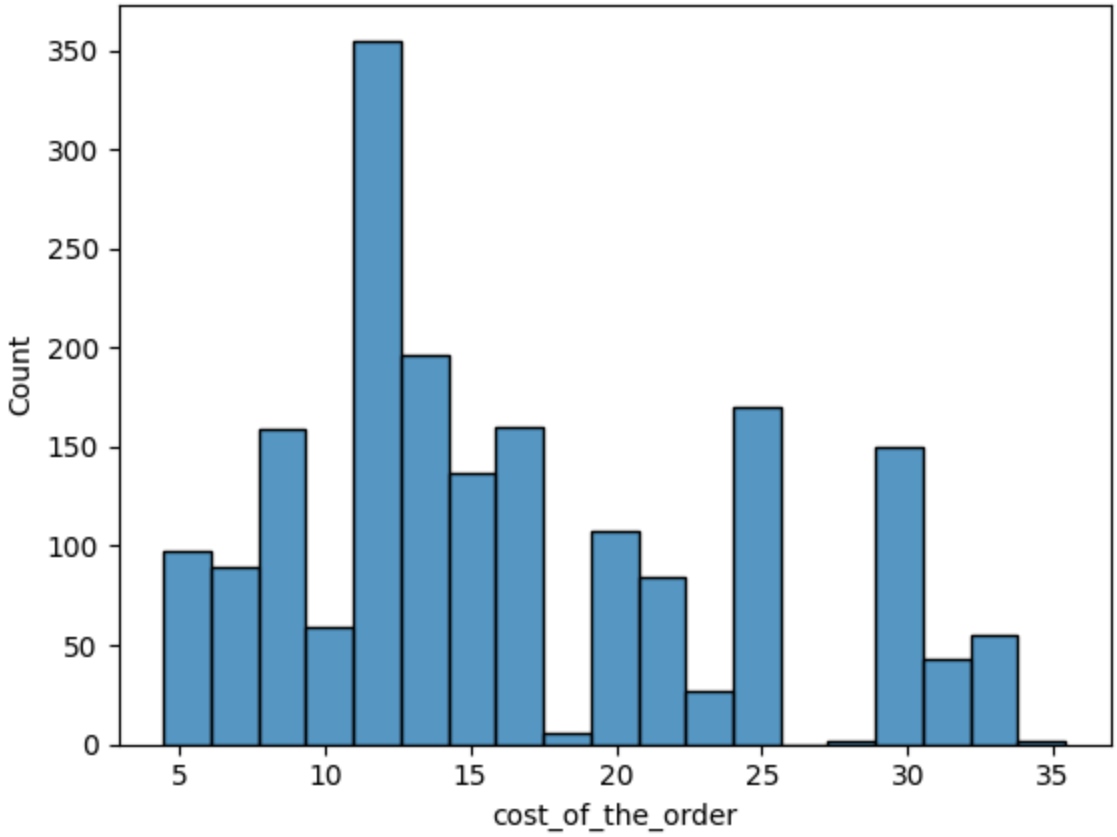
```
sns.countplot(data=df, y='restaurant_name')
plt.show()

sns.countplot(data=df, y='cuisine_type')
plt.show()
```

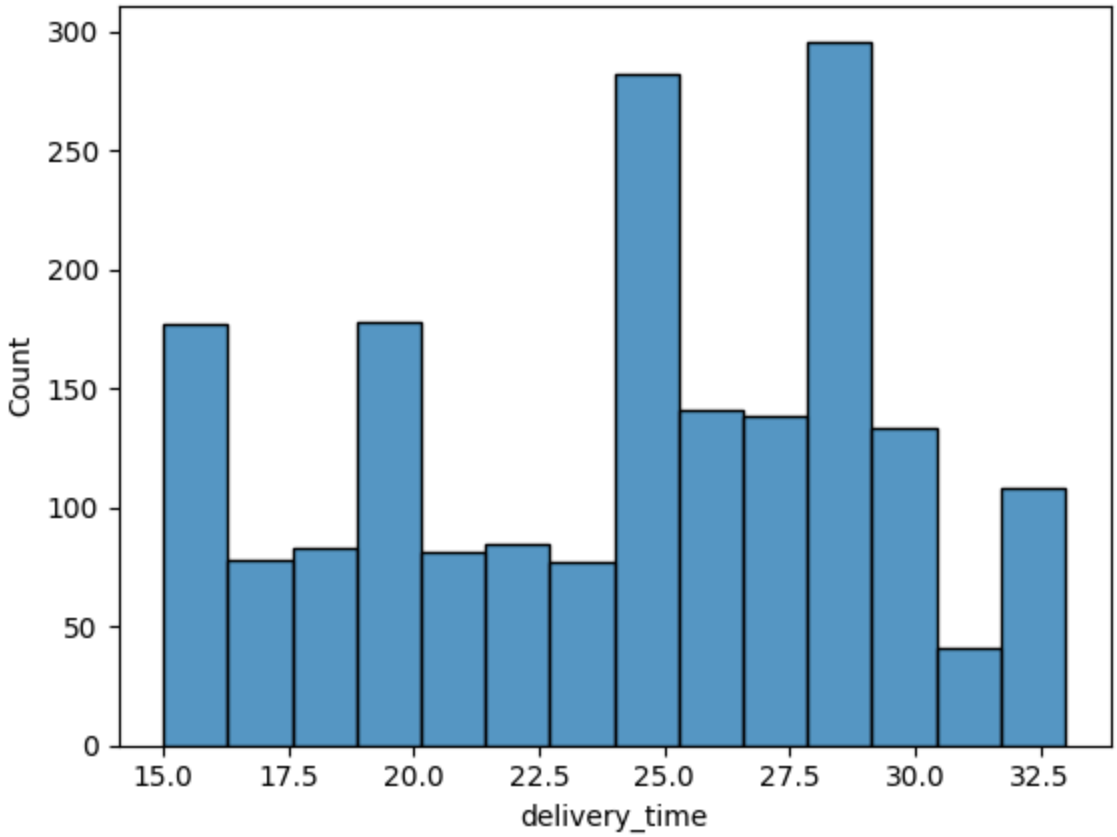


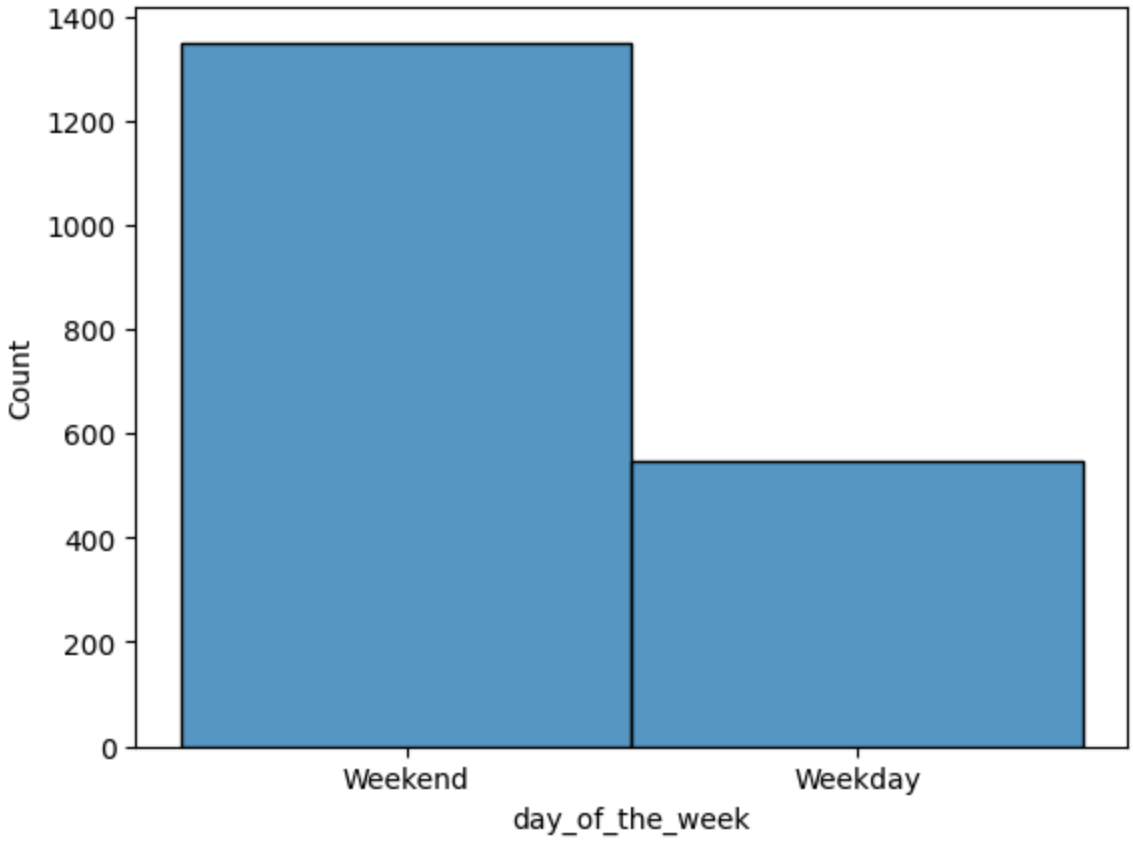

Box Plot for order cost
35
30
25
20
15
10
5
cost_of_the_order
order cost


Box Plot for delivery time
32.5
30.0
27.5
25.0
22.5
20.0
17.5
15.0
delivery_time
delivery time in minutes

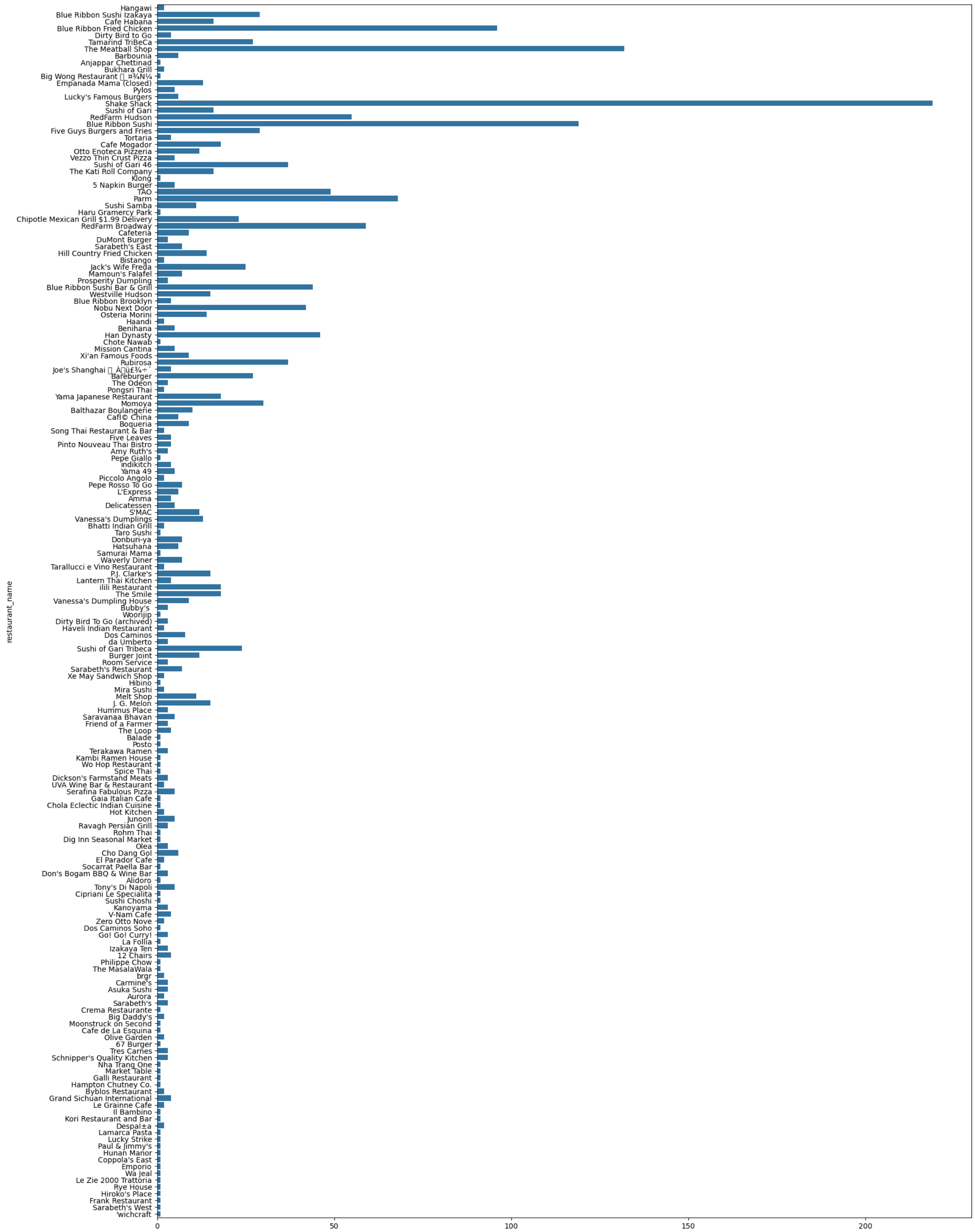

Count
250
200
150
100
50
0
20
22
24
26
28
30
32
34
food_preparation_time

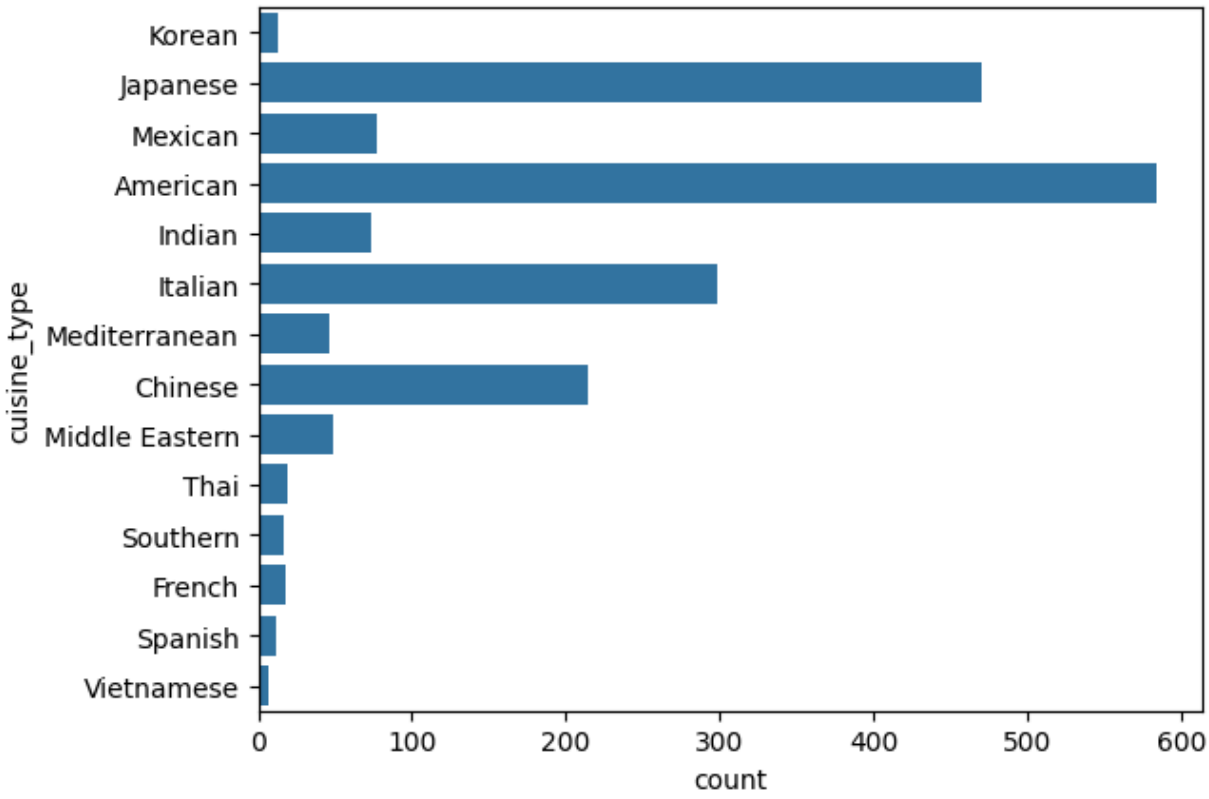

Count
350
300
250
200
150
100
50
0
5
10
15
20
25
30
35
cost_of_the_order






```
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
Glyph 140 (\x8c) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning:
Glyph 142 (\x8e) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```


Hangawi
Blue Ribbon Sushi Izakaya
Cafe Habana
Blue Ribbon Fried Chicken
Dirty Bird to Go
Tamarind TriBeCa
The Meatball Shop
Barbounia
Anjappar Chettinad
Bukhara Grill
Big Wong Restaurant
Empanada Mama (closed)
Pylos
Lucky's Famous Burgers
Shake Shack
Sushi of Gari
RedFarm Hudson
Blue Ribbon Sushi
Five Guys Burgers and Fries
Tortaria
Cafe Mogador
Otto Enoteca Pizzeria
Vezzo Thin Crust Pizza
Sushi of Gari 46
The Kati Roll Company
Klong
5 Napkin Burger
TAO
Parm
Sushi Samba
Haru Gramercy Park
Chipotle Mexican Grill $1.99 Delivery
RedFarm Broadway
Cafeteria
DuMont Burger
Sarabeth's East
Hill Country Fried Chicken
Bistango
Jack's Wife Freda
Mamoun's Falafel
Prosperity Dumpling
Blue Ribbon Sushi Bar & Grill
Westville Hudson
Blue Ribbon Brooklyn
Nobu Next Door
Osteria Morini
Haandi
Benihana
Han Dynasty
Chote Nawab
Mission Cantina
Xi'an Famous Foods
Rubirosa
Joe's Shanghai
Bareburger
The Odeon
Pongsri Thai
Yama Japanese Restaurant
Momoya
Balthazar Boulangerie
Cafl© China
Boqueria
Song Thai Restaurant & Bar
Five Leaves
Pinto Nouveau Thai Bistro
Amy Ruth's
Pepe Giallo
indikitch
Yama 49
Piccolo Angolo
Pepe Rosso To Go
L'Express
Amma
Delicatessen
S'MAC
Vanessa's Dumplings
Bhatti Indian Grill
Taro Sushi
Donburi-ya
Hatsuhana
Samurai Mama
Waverly Diner
Tarallucci e Vino Restaurant
P.J. Clarke's
Lantern Thai Kitchen
ilili Restaurant
The Smile
Vanessa's Dumpling House
Bubby's
Woorijip
Dirty Bird To Go (archived)
Haveli Indian Restaurant
Dos Caminos
da Umberto
Sushi of Gari Tribeca
Burger Joint
Room Service
Sarabeth's Restaurant
Xe May Sandwich Shop
Hibino
Mira Sushi
Melt Shop
J. G. Melon
Hummus Place
Saravanaa Bhavan
Friend of a Farmer
The Loop
Balade
Posto
Terakawa Ramen
Kambi Ramen House
Wo Hop Restaurant
Spice Thai
Dickson's Farmstand Meats
UVA Wine Bar & Restaurant
Serafina Fabulous Pizza
Gaia Italian Cafe
Chola Eclectic Indian Cuisine
Hot Kitchen
Junoon
Ravagh Persian Grill
Rohm Thai
Dig Inn Seasonal Market
Olea
Cho Dang Gol
El Parador Cafe
Socarrat Paella Bar
Don's Bogam BBQ & Wine Bar
Alidoro
Tony's Di Napoli
Cipriani Le Specialita
Sushi Choshi
Kanoyama
V-Nam Cafe
Zero Otto Nove
Dos Caminos Soho
Go! Go! Curry!
La Follia
Izakaya Ten
12 Chairs
Philippe Chow
The MasalaWala
brgr
Carmine's
Asuka Sushi
Aurora
Sarabeth's
Crema Restaurante
Big Daddy's
Moonstruck on Second
Cafe de La Esquina
Olive Garden
67 Burger
Tres Carnes
Schnipper's Quality Kitchen
Nha Trang One
Market Table
Galli Restaurant
Hampton Chutney Co.
Byblos Restaurant
Grand Sichuan International
Le Grainne Cafe
Il Bambino
Kori Restaurant and Bar
Despal±a
Lamarca Pasta
Lucky Strike
Paul & Jimmy's
Hunan Manor
Coppola's East
Emporio
Wa Jeal
Le Zie 2000 Trattoria
Rye House
Hiroko's Place
Frank Restaurant
Sarabeth's West
'wichcraft
restaurant_name
0
50
100
150
200
count



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

In [ ]:
```
n=5
df['restaurant_name'].value_counts()[:n]
```

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

Observations: Shake Shack, The Meatball Shop, Blue Ribbon Sushi, Blue Ribbon Fried Chicken, and Parm are the top 5 restaurants in terms of orders received.

Question 8: Which is the most popular cuisine on weekends?

In [ ]:
```
list_weekends=list(df[df['day_of_the_week']=='Weekend']['cuisine_type']) # Creates lis
weekend_cuisine_count = {i:list_weekends.count(i) for i in list_weekends} # Counts dup
print(weekend_cuisine_count) #prints cuisines with number of times that they duplicate
```

```
{'Korean': 11, 'Japanese': 335, 'American': 415, 'Italian': 207, 'Mexican': 53, 'Medi
terranean': 32, 'Chinese': 163, 'Indian': 49, 'Thai': 15, 'Southern': 11, 'French': 1
3, 'Spanish': 11, 'Middle Eastern': 32, 'Vietnamese': 4}
```

Observations: 'American' is the most popular cuisine on the weekends'

Question 9: What percentage of the orders cost more than 20 dollars?

In [ ]:
```
# Using dataframe df: create list of orders that cost more than 20 dollars

over_twenty = df[df['cost_of_the_order'] > 20]   #creates new dataframe called over_tw
print(over_twenty.shape) #providing shape tells us how many rows or orders  are over $
print(555/1898) #dividing number of orders over $20 by the total number of orders in c
```

```
(555, 9)
0.2924130663856691
```

Observations: Approximately 29% of orders were over $20

Question 10: What is the mean order delivery time?

In [ ]:
```
df["delivery_time"].mean() #provide the mean of all the delivery times provided#
```

Out[ ]:
```
24.161749209694417
```

Observations: The approximate mean delivery time is 24 minutes

Question 11: The company has decided to give 20% discount vouchers to the top 3 most frequent customers. Find the IDs of these customers and the number of orders they placed

In [ ]:
```
n=3 #variable assigned to be limiter. set to 3 for top 3 returned values
df['customer_id'].value_counts()[:n] #counts number of times a customer id repeats and
```

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

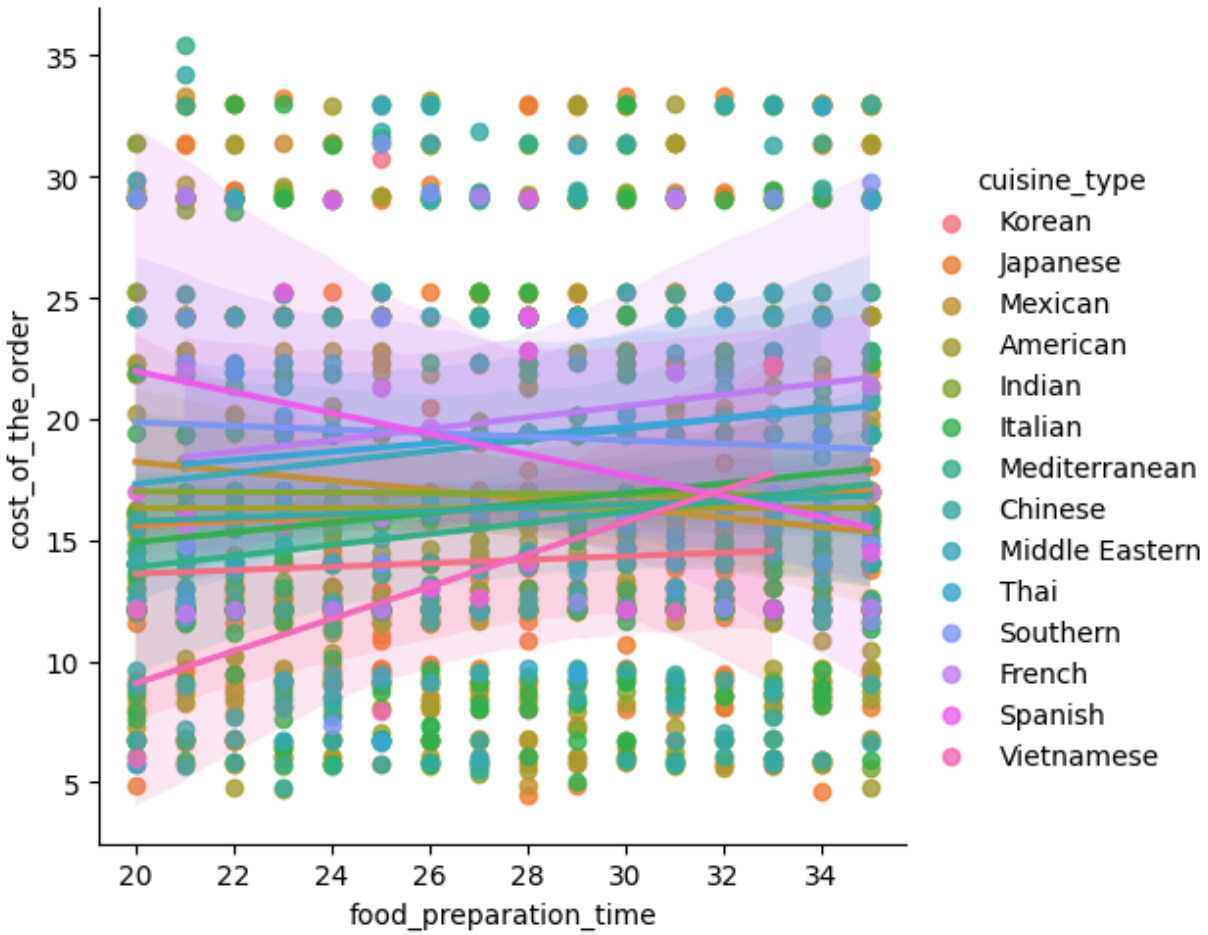
Observations: Customer ID 52832 placed 13 orders. Customer Id 47440 placed 10 orders, and customer ID 83287 placed 9 orders. These are the top 3 most frequent customers

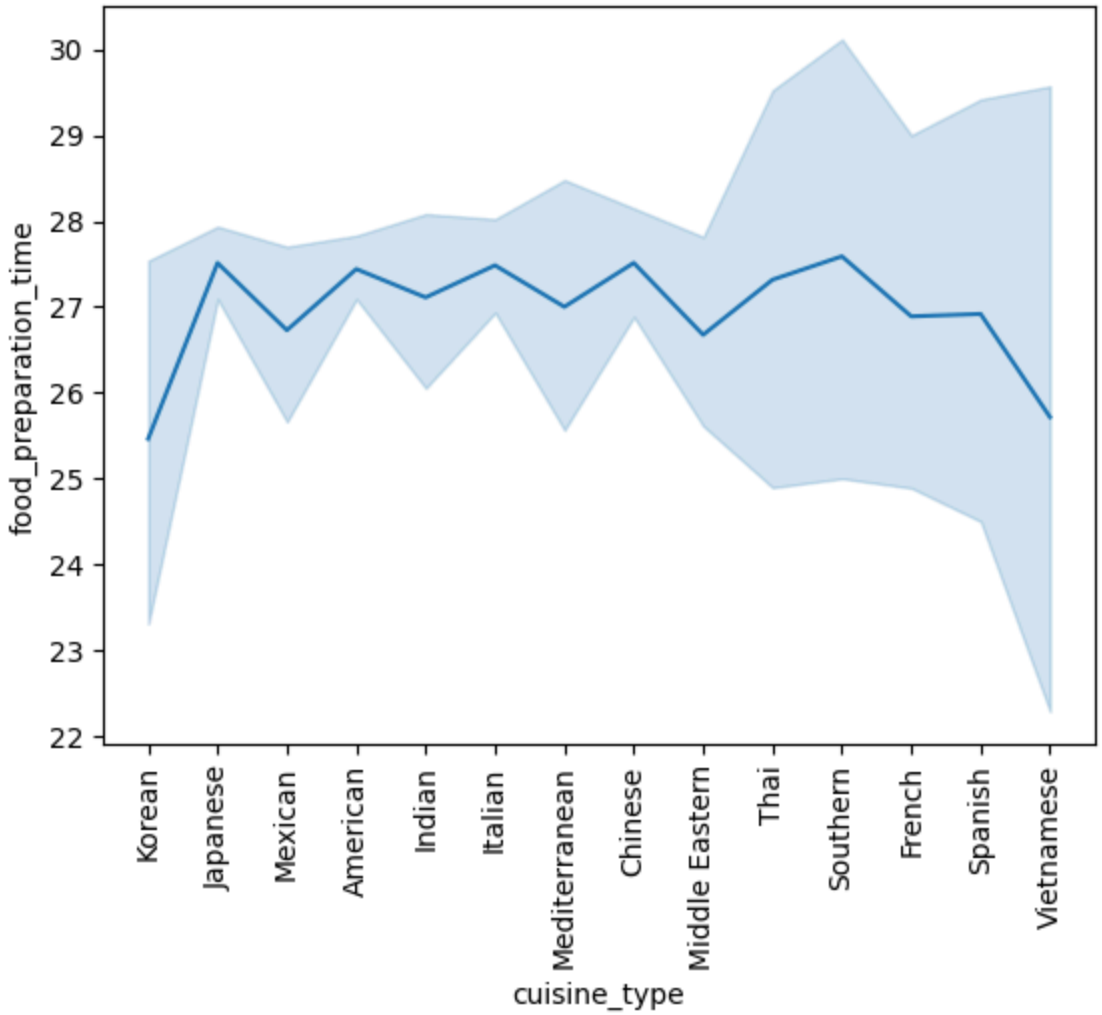
Multivariate Analysis

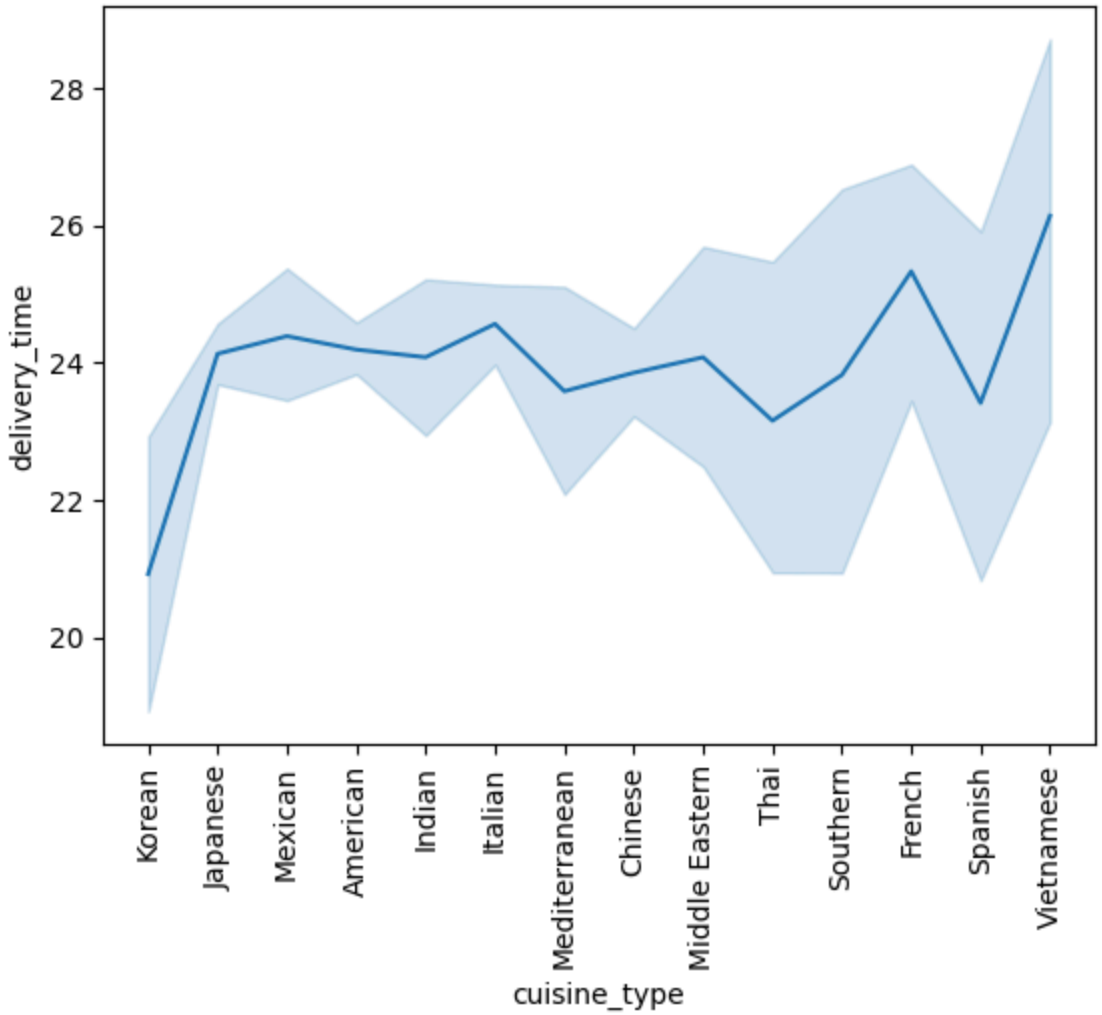
Question 12: Perform a multivariate analysis to explore relationships between the important variables in the dataset. (It is a good idea to explore relations between numerical variables as well as relations between numerical and categorical variables)

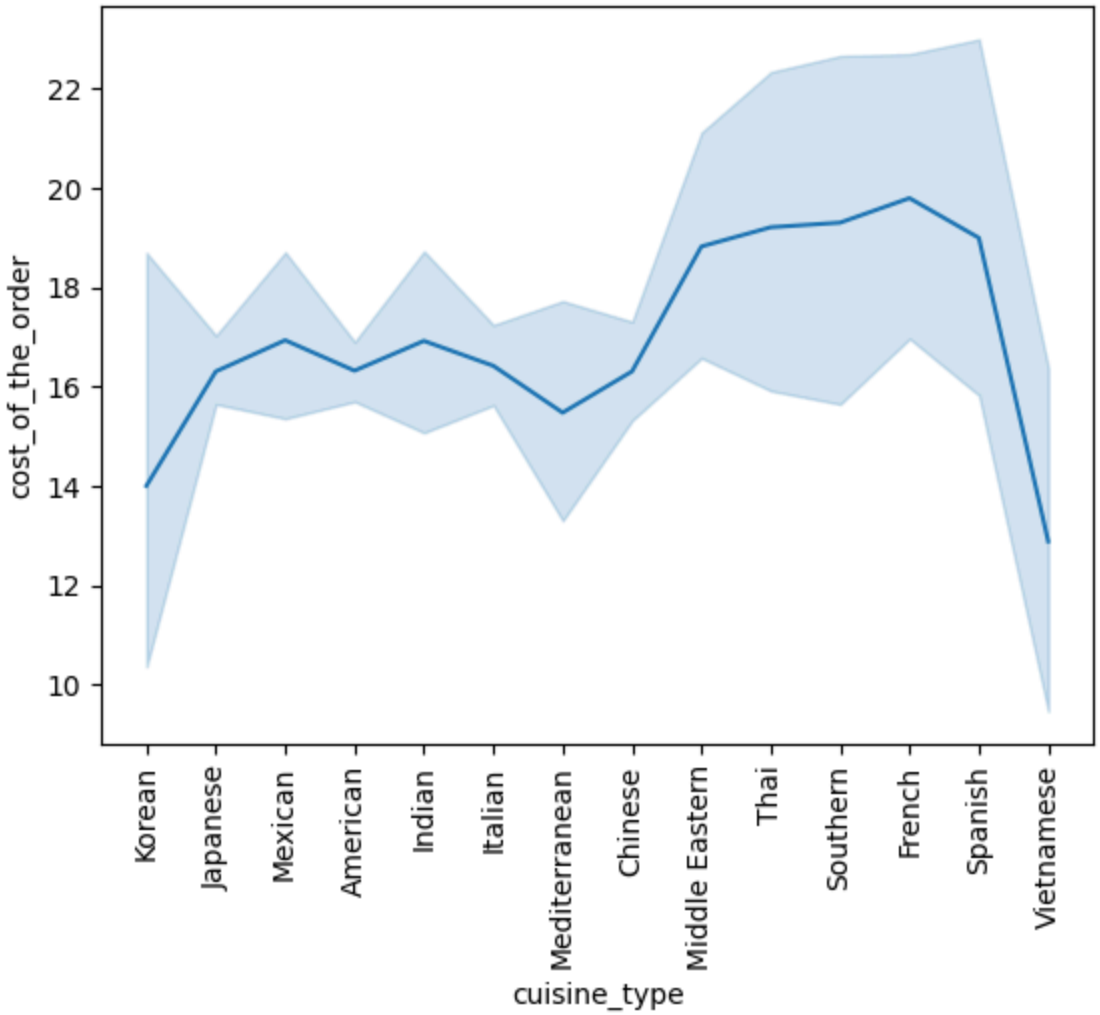
In [ ]:
```
### lmplot of delivey cost vs delivery time ###

sns.lmplot(data=df, x='food_preparation_time', y='cost_of_the_order', hue='cuisine_typ
plt.show()
```

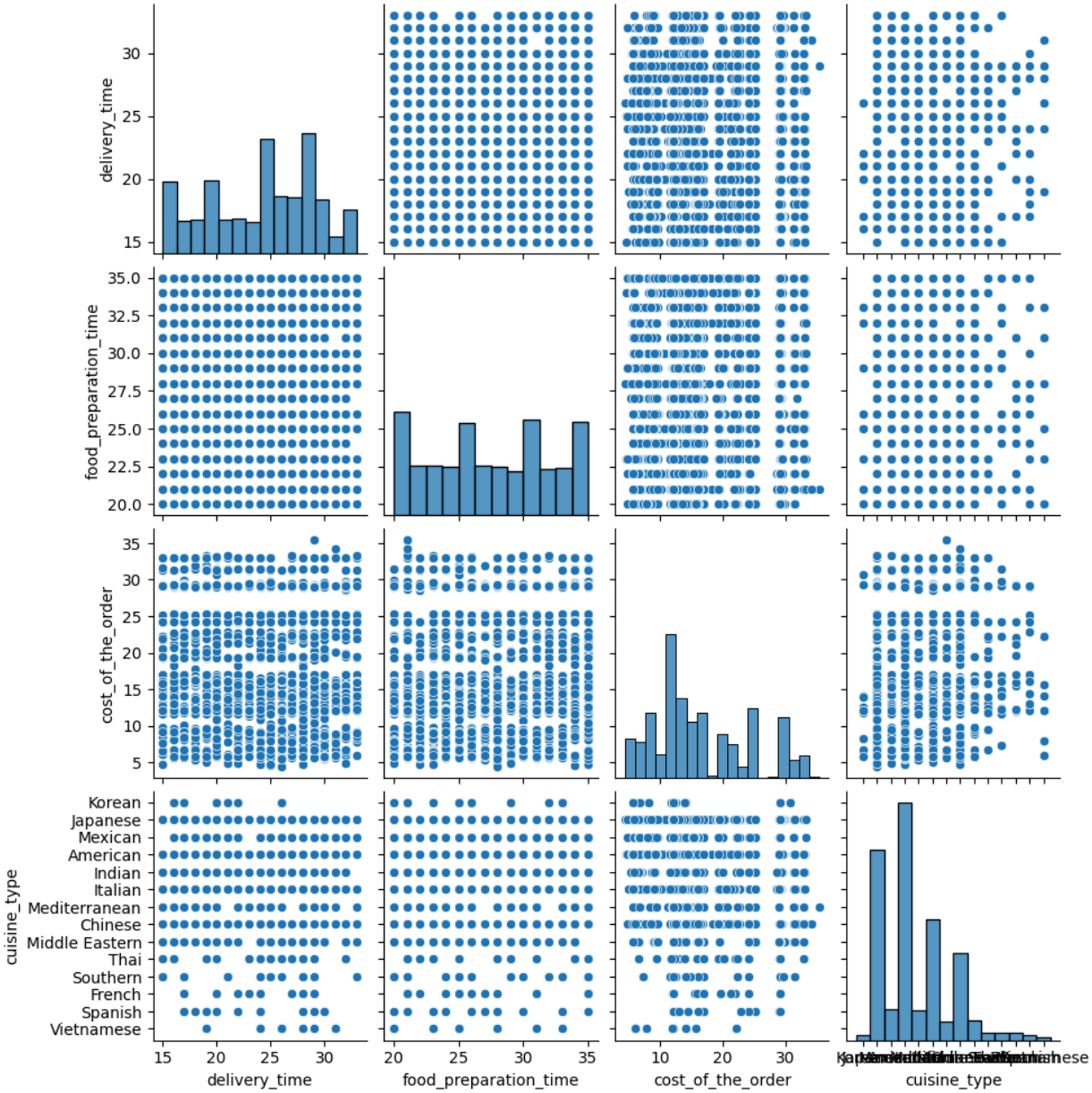
```
# Lineplot of preparation time across cuisine type#
sns.lineplot(data=df, y='food_preparation_time', x='cuisine_type')
plt.xticks(rotation=90)
plt.show()

#Lineplot of delivery time across cuisine type#
sns.lineplot(data=df, y='delivery_time', x='cuisine_type')
plt.xticks(rotation=90)
plt.show()

sns.lineplot(data=df, y='cost_of_the_order', x='cuisine_type')
plt.xticks(rotation=90)
plt.show()

sns.pairplot(data=df, vars=["delivery_time","food_preparation_time", "cost_of_the_orde
plt.show()

sns.histplot(data=df, x="cuisine_type", y='customer_id')
plt.xticks(rotation=90)
plt.show()

plt.figure(figsize=(20,20))
sns.histplot(data=df, y='cuisine_type', x='customer_id')
plt.xticks(rotation=90)
plt.show()
```

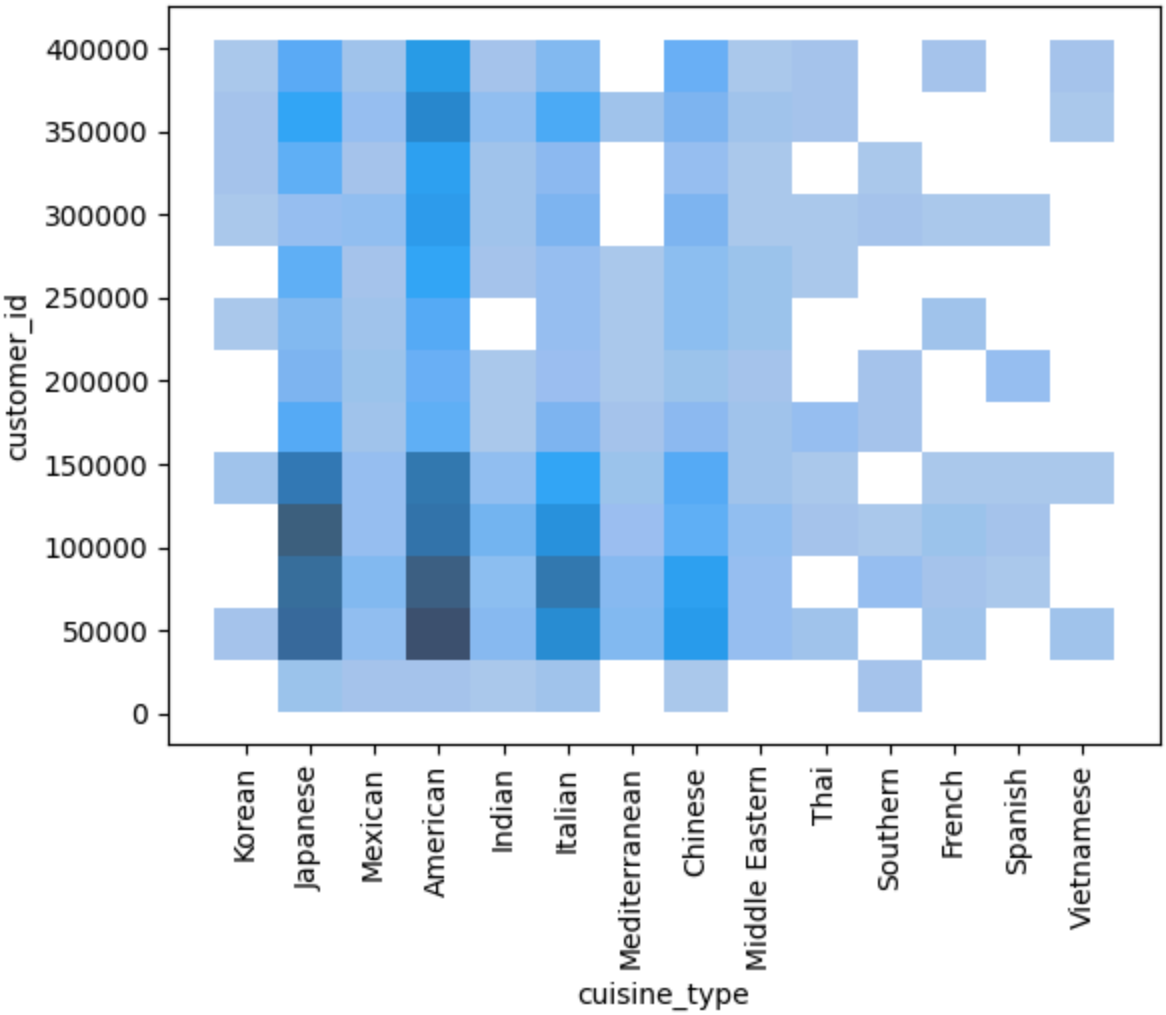


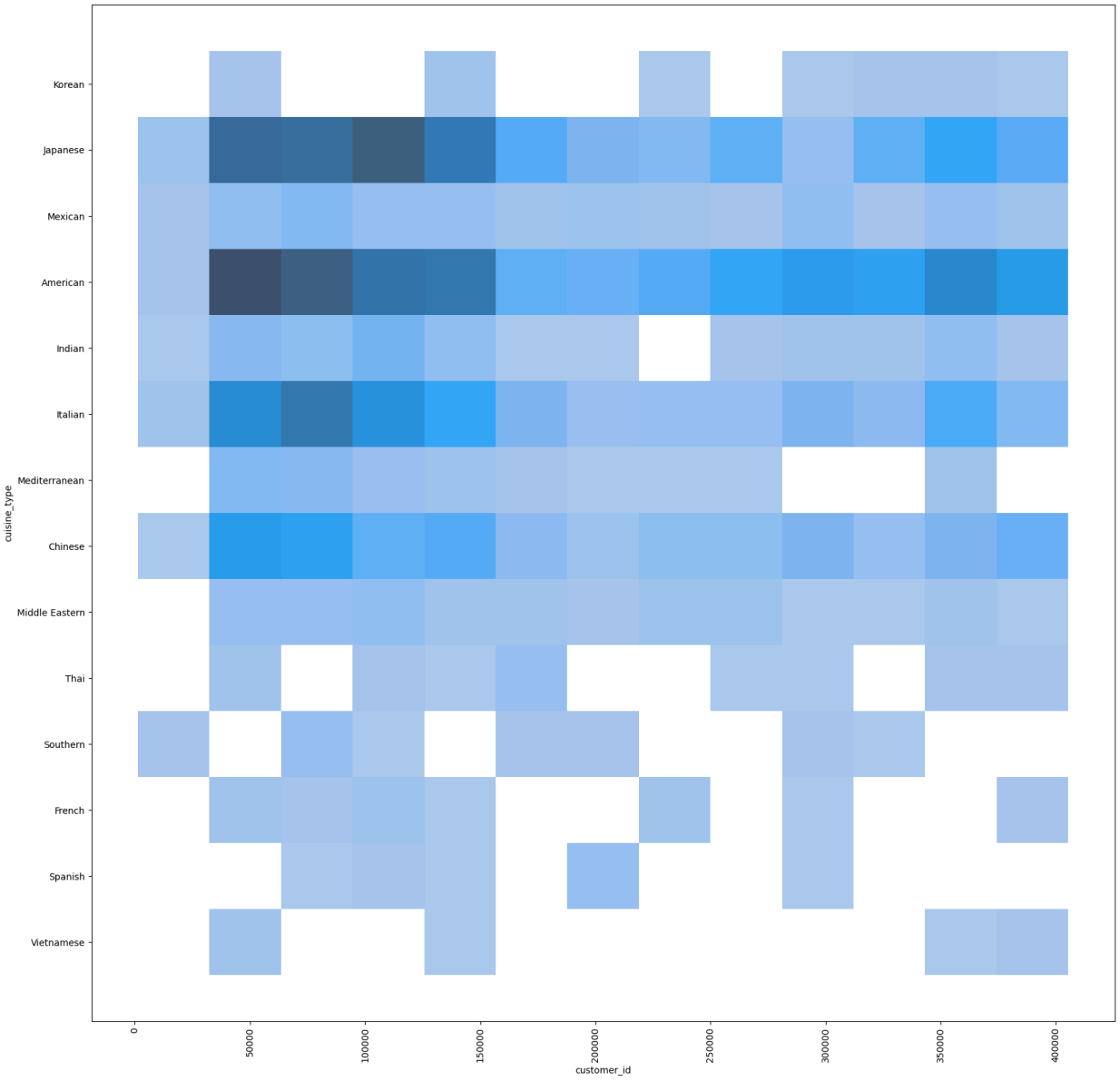

food_preparation_time
30
29
28
27
26
25
24
23
22
Korean
Japanese
Mexican
American
Indian
Italian
Mediterranean
Chinese
Middle Eastern
Thai
Southern
French
Spanish
Vietnamese
cuisine_type


28
26
24
22
20
delivery_time
Korean
Japanese
Mexican
American
Indian
Italian
Mediterranean
Chinese
Middle Eastern
Thai
Southern
French
Spanish
Vietnamese
cuisine_type


cost_of_the_order
22
20
18
16
14
12
10
Korean
Japanese
Mexican
American
Indian
Italian
Mediterranean
Chinese
Middle Eastern
Thai
Southern
French
Spanish
Vietnamese
cuisine_type


delivery_time
food_preparation_time
cost_of_the_order
cuisine_type
Korean
Japanese
Mexican
American
Indian
Italian
Mediterranean
Chinese
Middle Eastern
Thai
Southern
French
Spanish
Vietnamese


customer_id
400000
350000
300000
250000
200000
150000
100000
50000
0
Korean
Japanese
Mexican
American
Indian
Italian
Mediterranean
Chinese
Middle Eastern
Thai
Southern
French
Spanish
Vietnamese
cuisine_type



In [ ]:

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

In [ ]:
```
only_rated=df[df['rating'].isin(['1', '2', '3', '4', '5'])] #creates new df limited to
only_rated['rating'] = only_rated['rating'].astype('int') #changes rating row from obj


print(only_rated['restaurant_name'].value_counts()) #counts the number of times each r
Shake_Shack=only_rated[only_rated['restaurant_name']=='Shake Shack'] #creates df of or
print(Shake_Shack.head())
Shake_Shack_avg=Shake_Shack['rating'].mean() #assigns the mean of ss ratings to a valu
print("Shake Shop's average rating is:", Shake_Shack_avg) #provides string with avg ra

The_Meatball_Shop=only_rated[only_rated['restaurant_name']=='The Meatball Shop'] #crea
```

```
print(The_Meatball_Shop.head())
Meatball_Shop_avg=The_Meatball_Shop['rating'].mean()  #assigns the mean of mbs ratings
print("The Meatball Shop's average rating is:", Meatball_Shop_avg) #prints string with

Blue_Ribbon_Sushi=only_rated[only_rated['restaurant_name']=='Blue Ribbon Sushi'] #creates
print(Blue_Ribbon_Sushi.head())
Blue_Ribbon_Sushi_avg=Blue_Ribbon_Sushi['rating'].mean() #assigns mean of brs to value
print("Blue Ribbon Sushi's average rating is:", Blue_Ribbon_Sushi_avg)  #prints string

Blue_Ribbon_Chicken=only_rated[only_rated['restaurant_name']=='Blue Ribbon Fried Chick
print(Blue_Ribbon_Chicken.head())
Blue_Ribbon_Chicken_avg=Blue_Ribbon_Chicken['rating'].mean() #assigns mean of ratings
print("Blue Ribbon Chicken's average rating is:", Blue_Ribbon_Chicken_avg) #prints str

print("The meatball Shop's average rating is:", Meatball_Avg,"/n","Shake Shack's
      "Blue Ribbon Fried Chicken's average rating is:", Blue_Ribbon_Chicken_avg)   # p

print()
print()
```

```
Shake Shack                   133
The Meatball Shop              84
Blue Ribbon Sushi              73
Blue Ribbon Fried Chicken      64
RedFarm Broadway               41
                              ...
Philippe Chow                   1
Dirty Bird To Go (archived)     1
The MasalaWala                  1
Kambi Ramen House               1
'wichcraft                      1
Name: restaurant_name, Length: 156, dtype: int64
    order_id  customer_id restaurant_name cuisine_type  cost_of_the_order  \
15   1477414        66222     Shake Shack     American              16.20
22   1478287       150599     Shake Shack     American              29.10
71   1476651        58092     Shake Shack     American               8.00
80   1477975        56722     Shake Shack     American               9.75
82   1477790       133617     Shake Shack     American               4.75

   day_of_the_week  rating  food_preparation_time  delivery_time
15         Weekend       5                     33             25
22         Weekday       5                     21             30
71         Weekend       5                     27             23
80         Weekend       5                     33             25
82         Weekday       4                     35             28
Shake Shop's average rating is: 4.2781954887218046
     order_id  customer_id    restaurant_name cuisine_type  cost_of_the_order  \
26    1476995       371590  The Meatball Shop      Italian              21.88
37    1476871       118709  The Meatball Shop      Italian              24.30
45    1476581       322162  The Meatball Shop      Italian               6.74
127   1477405       128243  The Meatball Shop      Italian               6.74
144   1478269       250494  The Meatball Shop      Italian              11.16

    day_of_the_week  rating  food_preparation_time  delivery_time
26          Weekday       5                     24             27
37          Weekday       4                     31             29
45          Weekend       5                     29             23
127         Weekend       5                     26             30
144         Weekday       5                     22             28
The Meatball Shop's average rating is: 4.511904761904762
     order_id  customer_id    restaurant_name cuisine_type  cost_of_the_order  \
19    1477354        67487  Blue Ribbon Sushi     Japanese              16.20
36    1478017       148649  Blue Ribbon Sushi     Japanese              16.01
102   1477240       142356  Blue Ribbon Sushi     Japanese              11.83
106   1477617        38050  Blue Ribbon Sushi     Japanese              12.18
109   1478223       234089  Blue Ribbon Sushi     Japanese               8.68

    day_of_the_week  rating  food_preparation_time  delivery_time
19          Weekend       4                     35             26
36          Weekday       4                     23             31
102         Weekend       4                     26             21
106         Weekend       5                     28             25
109         Weekend       5                     34             27
Blue Ribbon Sushi's average rating is: 4.219178082191781
     order_id  customer_id            restaurant_name cuisine_type  \
3     1477334       106968  Blue Ribbon Fried Chicken     American
12    1476966       129969  Blue Ribbon Fried Chicken     American
69    1477475        65009  Blue Ribbon Fried Chicken     American
96    1476921       121476  Blue Ribbon Fried Chicken     American
117   1476770        65009  Blue Ribbon Fried Chicken     American
```

```
     cost_of_the_order day_of_the_week  rating  food_preparation_time  \
3                29.20         Weekend       3                     25
12               24.30         Weekend       5                     23
69               32.93         Weekend       5                     24
96               12.18         Weekday       3                     29
117               7.86         Weekday       4                     22

     delivery_time
3               15
12              17
69              23
96              27
117             33
Blue Ribbon Chicken's average rating is: 4.328125
The meatball Shop's average rating is: 4.511904761904762 /n Shake Shack's average rat
ing is: 4.2781954887218046 /n Blue Ribbon Sushi's average rating is: 4.21917808219178
1 Blue Ribbon Fried Chicken's average rating is: 4.328125
```

```
<ipython-input-74-38338a3e5b04>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/us
er_guide/indexing.html#returning-a-view-versus-a-copy
  only_rated['rating'] = only_rated['rating'].astype('int') #changes rating row from
object to integer
```

Observations: Only 4 restaurants had more than 50 ratings and all 4 of them had an average rating of over 4. Those restaurants are, "The Shake Shack", "The Meatball Shop", "Blue Ribbon Sushi", and "Blue Ribbon Fried Chicken".

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

In [ ]:
```
over_twenty = df[df['cost_of_the_order'] > 20] #creates data frame of orders over 20 d
over_five = df[df['cost_of_the_order'] > 5] #creates data frame of orders over 5 dolla
print(over_twenty)
print(over_twenty_five)

net_revenue=((over_twenty['cost_of_the_order'].sum())*.25)+((over_five['cost_of_the_or
print("The total revenue generated across all orders is:", net_revenue)
```

```
      order_id  customer_id                         restaurant_name  \
0      1477147       337525                                 Hangawi
3      1477334       106968              Blue Ribbon Fried Chicken
5      1477224       147468                       Tamarind TriBeCa
12     1476966       129969              Blue Ribbon Fried Chicken
17     1477373       139885              Blue Ribbon Sushi Izakaya
...        ...          ...                                    ...
1884   1477437       304993                             Shake Shack
1885   1477550        97324                             Shake Shack
1892   1477473        97838                             Han Dynasty
1893   1476701       292602  Chipotle Mexican Grill $1.99 Delivery
1895   1477819        35309                        Blue Ribbon Sushi

     cuisine_type  cost_of_the_order day_of_the_week     rating  \
0          Korean              30.75         Weekend  Not given
3        American              29.20         Weekend          3
5          Indian              25.22         Weekday          3
12       American              24.30         Weekend          5
17       Japanese              33.03         Weekend  Not given
...           ...                ...             ...        ...
1884     American              31.43         Weekend          3
1885     American              29.05         Weekday          4
1892      Chinese              29.15         Weekend  Not given
1893      Mexican              22.31         Weekend          5
1895     Japanese              25.22         Weekday  Not given

      food_preparation_time  delivery_time
0                        25             20
3                        25             15
5                        20             24
12                       23             17
17                       21             22
...                     ...            ...
1884                     31             24
1885                     27             29
1892                     29             21
1893                     31             17
1895                     31             24

[555 rows x 9 columns]
      order_id  customer_id            restaurant_name cuisine_type  \
0      1477147       337525                    Hangawi       Korean
3      1477334       106968  Blue Ribbon Fried Chicken     American
5      1477224       147468           Tamarind TriBeCa       Indian
17     1477373       139885  Blue Ribbon Sushi Izakaya     Japanese
22     1478287       150599                Shake Shack     American
...        ...          ...                        ...          ...
1872   1477000       328731  Blue Ribbon Fried Chicken     American
1884   1477437       304993                Shake Shack     American
1885   1477550        97324                Shake Shack     American
1892   1477473        97838                Han Dynasty      Chinese
1895   1477819        35309          Blue Ribbon Sushi     Japanese

      cost_of_the_order day_of_the_week     rating  food_preparation_time  \
0                 30.75         Weekend  Not given                     25
3                 29.20         Weekend          3                     25
5                 25.22         Weekday          3                     20
17                33.03         Weekend  Not given                     21
22                29.10         Weekday          5                     21
...                 ...             ...        ...                    ...
```

```
1872              29.59         Weekend          5                    23
1884              31.43         Weekend          3                    31
1885              29.05         Weekday          4                    27
1892              29.15         Weekend  Not given                    29
1895              25.22         Weekday  Not given                    31

      delivery_time
0                20
3                15
5                24
17               22
22               30
...             ...
1872             25
1884             24
1885             29
1892             21
1895             24

[305 rows x 9 columns]
The total revenue generated across all orders is: 8379.539499999999
3688.7275
4690.812
```

Observations: The total revenue genearated across all orders is approximately $8379.54

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)

In [ ]:
```
Total_processing_time=(df['food_preparation_time']+df['delivery_time']) #creates new c
new_df = [df['food_preparation_time'], df['delivery_time'], Total_processing_time] #cr
display(new_df)
```

```
[0       25
 1       25
 2       23
 3       25
 4       25
         ..
 1893    31
 1894    31
 1895    31
 1896    23
 1897    28
 Name: food_preparation_time, Length: 1898, dtype: int64,
 0       20
 1       23
 2       28
 3       15
 4       24
         ..
 1893    17
 1894    19
 1895    24
 1896    31
 1897    24
 Name: delivery_time, Length: 1898, dtype: int64,
 0       45
 1       48
 2       51
 3       40
 4       49
         ..
 1893    48
 1894    50
 1895    55
 1896    54
 1897    52
 Length: 1898, dtype: int64]
```

Observations: It appears that the largest combined time for food prep and delivery across all of the orders is 52 minutes. This leads to the observation that 0% of the orders take more than 60 minutes to get delivered from the time the order is placed.

Question 16: The company wants to analyze the delivery time of the orders on weekdays and weekends. How does the mean delivery time vary during weekdays and weekends?

In [ ]:
```
list_weekends_delivery=list(df[df['day_of_the_week']=='Weekend']['delivery_time']) #cr
list_weekdays_delivery=list(df[df['day_of_the_week']=='Weekday']['delivery_time']) #cr
weekend_delivery_mean = np.average(list_weekends_delivery) #assigns average of weekend
weekday_delivery_mean = np.average(list_weekdays_delivery) #assigns average of weekday
print("The mean for weekend deliveries is:", weekend_delivery_mean)
print("The mean for weekday deliveries is:", weekday_delivery_mean)
print("Delivery difference is", (weekend_delivery_mean-weekday_delivery_mean))
print("Weekday deliveries, on average, are taking approximately", ((weekday_delivery_m
```

```
The mean for weekend deliveries is: 22.4700222057735
The mean for weekday deliveries is: 28.340036563071298
Delivery difference is -5.870014357297798
Weekday deliveries, on average, are taking approximately 6.0 more minutes than weeken
d deliveries.
```

Observations: Rounding up, the average difference in weekend vs weekday delivery times results in weekday deliveries taking 6 more minutes than weekend deliveries. The more exact difference to two decimal points is 5.8 minutes.

Conclusion and Recommendations

Question 17: 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: Our histogram shows us that most of our orders occur on the weekends which is also when we are faster with our delivery time. If we are ok with our weekday delivery times, we may be just slightly over staffed on the weekends.

There is a lot of inconsistency in the food prep time for, Southern, French, Spanish, and, Vietnamese cuisines. These orders also tend to have a large upward and downward swing in cost also. It is possible that these orders are of larger quantities resulting in longer food prep times. Either way, it may be good customer service to give an advisement of possible longer total time from order to delivery. Also, while they do some of our most expensive and hence, most profitable orders, they also do some of our least profitable, as well. Could be beneficial to add a minimum order size for delivery from these locations. It doesn't look like we do a majority of our orders in these cuisine types, so it shouldn't be too affecting to overall numbers. Also, there does not appear to be a large count of customer id's ordering from these locations, so there is the possibility that the majority of those ordering are very specific niche clients and would be willing to pay a premium for what they like.

-

Recommendations: Advise customers of potential longer delivery times for restaurants providing Southern, French, Spanish, and Vietnamese cuisines. Add a minimum order size for delivery from restaurants providding Southern, French, Spanish, and Vietnamese cuisines.

-
