In [1]:
```
#I am importing all libraries i will need.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
#I am loading the data.
df = pd.read_csv('/Users/conne/Downloads/foodhub_order.csv')
```

In [3]:
```
#Here i am checking to make sure it uploaded.
df
```

Out[3]:

	order_id	customer_id	restaurant_name	cuisine_type	cost_of_the_order	day_of_the_week	rati
0	1477147	337525	Hangawi	Korean	30.75	Weekend	N giv
1	1477685	358141	Blue Ribbon Sushi Izakaya	Japanese	12.08	Weekend	N giv
2	1477070	66393	Cafe Habana	Mexican	12.23	Weekday	
3	1477334	106968	Blue Ribbon Fried Chicken	American	29.20	Weekend	
4	1478249	76942	Dirty Bird to Go	American	11.59	Weekday	
...	...	...	...	...	...	...	
1893	1476701	292602	Chipotle Mexican Grill $1.99 Delivery	Mexican	22.31	Weekend	
1894	1477421	397537	The Smile	American	12.18	Weekend	
1895	1477819	35309	Blue Ribbon Sushi	Japanese	25.22	Weekday	N giv
1896	1477513	64151	Jack's Wife Freda	Mediterranean	12.18	Weekday	
1897	1478056	120353	Blue Ribbon Sushi	Japanese	19.45	Weekend	N giv

1898 rows × 9 columns

In [4]:
```
#Question 1
#this lists the number of rows and columns.
df.shape
```

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

In [5]:
```
#Question 1
#This has 1898 resturants listed with 9 variables for each resturant.
```

In [6]:
```
#This shows us our columns null count and its dtype
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
```

In [7]:
```
#Question 2
#dtypes: float64(2), int64(4), object(3)
#Question 3
#We have no missing values and we 3 different data types. Rating should be a number sc
```

In [8]:
```
#question 3
#we changed rating to a float and defined not given as NaN.
df["rating"] = df["rating"].replace(["Not given"], np.nan)
df["rating"] = df["rating"].astype(float)
```

In [9]:
```
#This gives me a statistical overview of the dataframe
df.describe()
```

Out[9]:

	order_id	customer_id	cost_of_the_order	rating	food_preparation_time	delivery_ti
count	1.898000e+03	1898.000000	1898.000000	1162.000000	1898.000000	1898.000(
mean	1.477496e+06	171168.478398	16.498851	4.344234	27.371970	24.161
std	5.480497e+02	113698.139743	7.483812	0.741478	4.632481	4.972(
min	1.476547e+06	1311.000000	4.470000	3.000000	20.000000	15.000(
25%	1.477021e+06	77787.750000	12.080000	4.000000	23.000000	20.000(
50%	1.477496e+06	128600.000000	14.140000	5.000000	27.000000	25.000(
75%	1.477970e+06	270525.000000	22.297500	5.000000	31.000000	28.000(
max	1.478444e+06	405334.000000	35.410000	5.000000	35.000000	33.000(

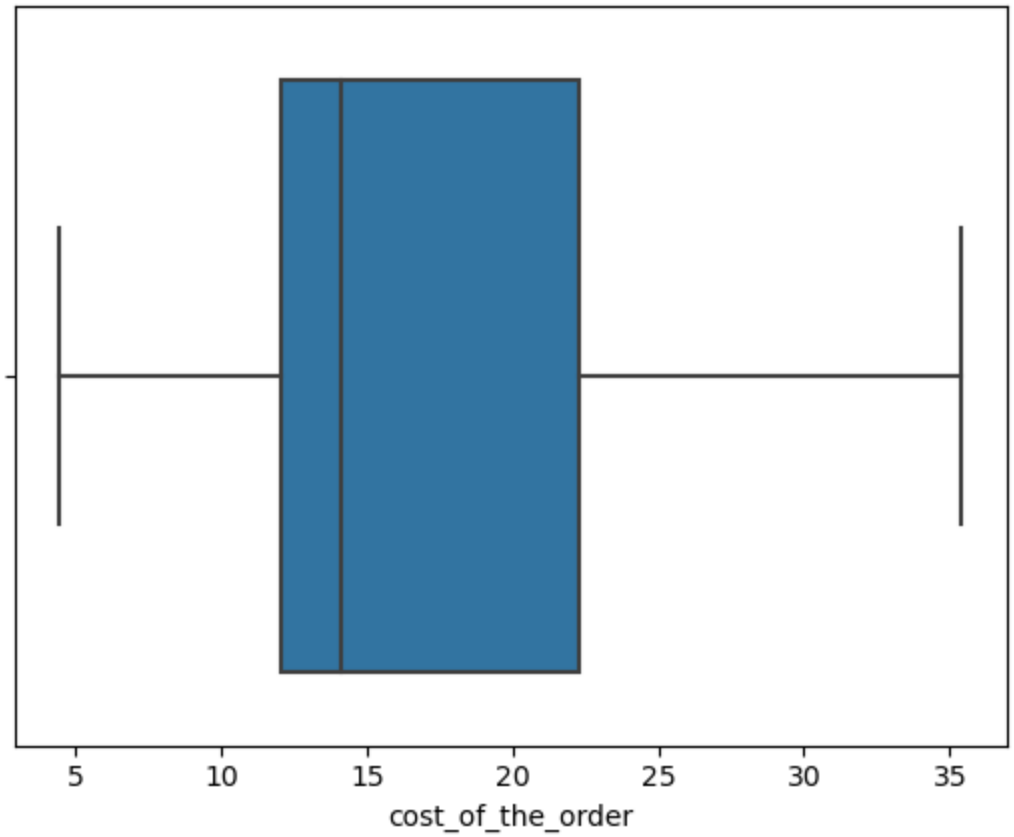
In [10]:
```
#Question 4
#The .describe method shows the min, mean, and max of food_perpartion_time which is 20
```

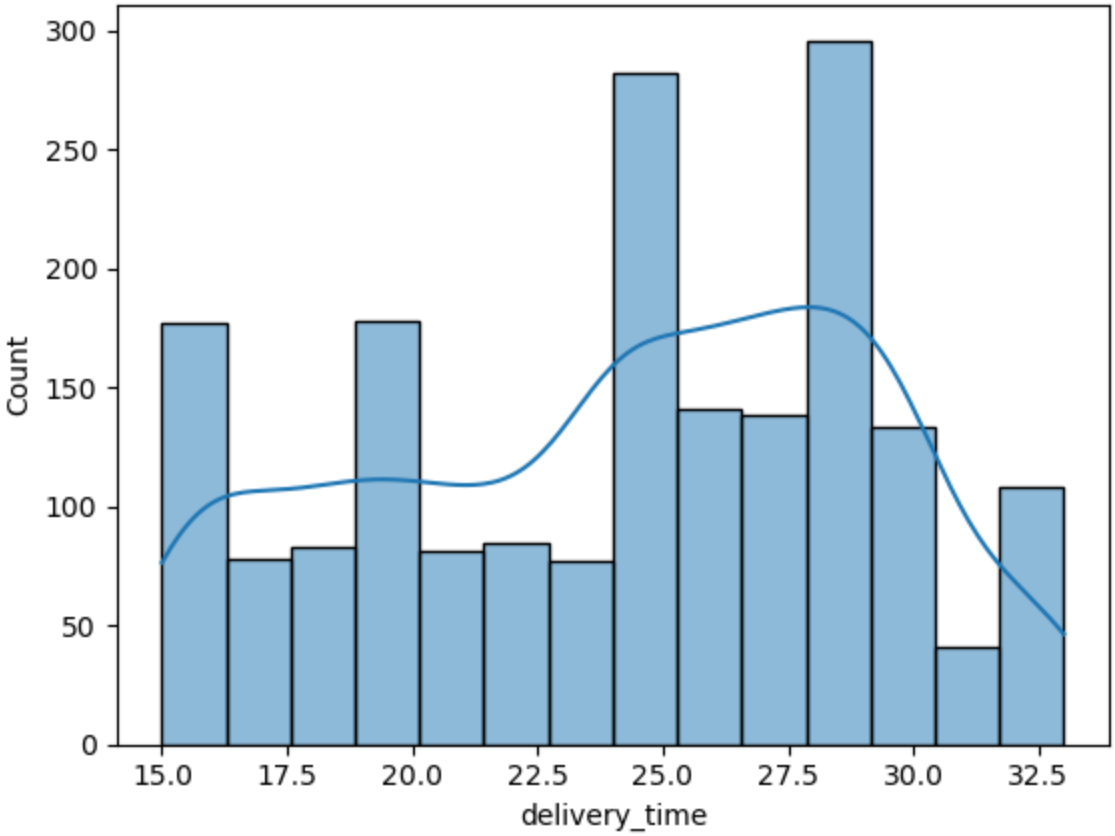
In [11]:
```
#so i can see how many null values were created.
df.info()
```

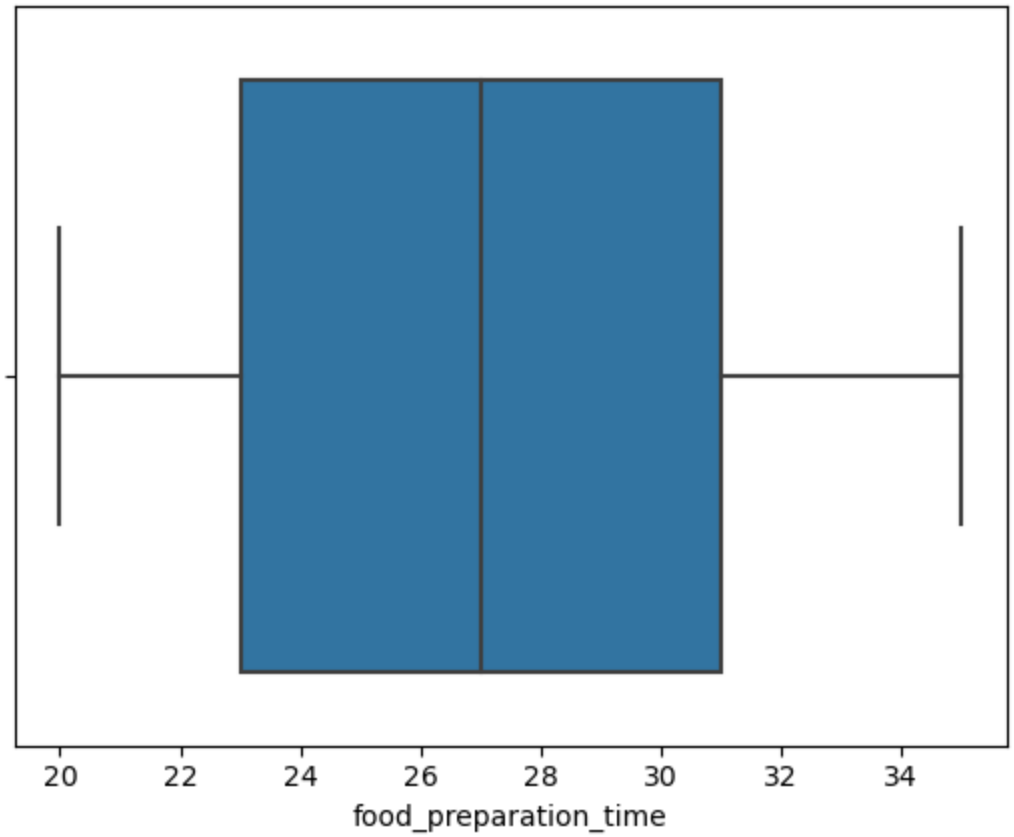
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1898 entries, 0 to 1897
Data columns (total 9 columns):
 #   Column                 Non-Null Count  Dtype  
---  ------                 --------------  -----  
 0   order_id               1898 non-null   int64  
 1   customer_id            1898 non-null   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                 1162 non-null   float64
 7   food_preparation_time  1898 non-null   int64  
 8   delivery_time          1898 non-null   int64  
dtypes: float64(2), int64(4), object(3)
memory usage: 133.6+ KB
```

In [12]:
```
#Question 5
# By subtracting 1898 from 1162 which gives us 736 meaning we had 736 ratings that wer
#We have 736 missing ratings.
```

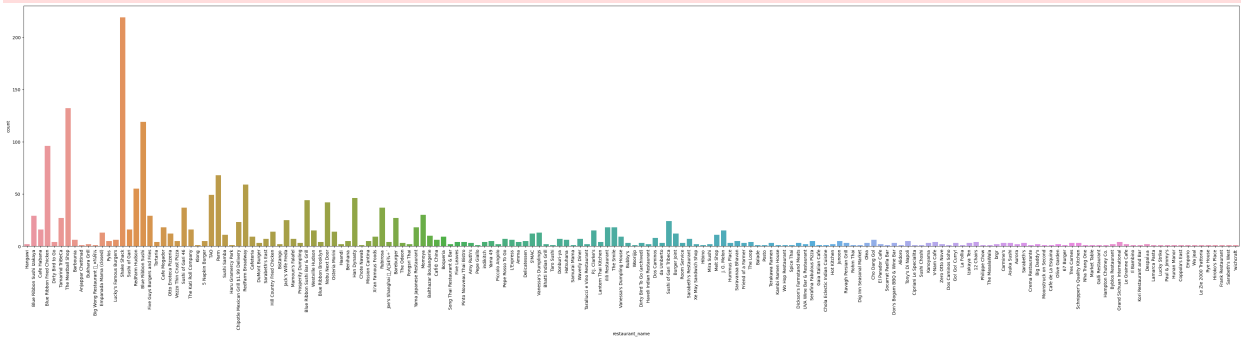
In [13]:
```
#Creating plots to examine cost of the order, delivery time, and food prep time.
sns.boxplot(data=df,x='cost_of_the_order')
plt.show()
sns.histplot(data=df, x='delivery_time', kde=True,)
plt.show()
sns.boxplot(data=df,x='food_preparation_time')
plt.show()
plt.figure(figsize=(50,10))
sns.countplot(data=df, x='restaurant_name')
plt.xticks(rotation=90)
plt.show()
sns.histplot(data=df, x='rating')
plt.show()
sns.histplot(data=df, x='day_of_the_week')
plt.show()
```

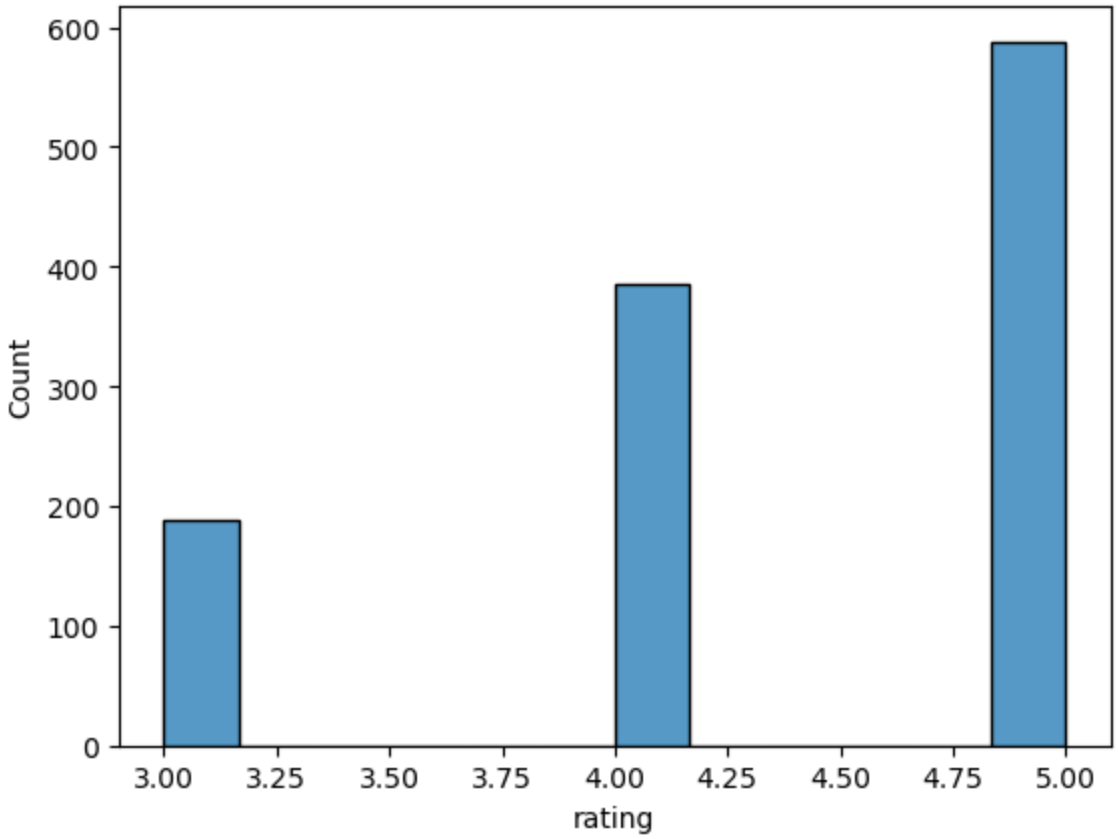

5
10
15
20
25
30
35
cost_of_the_order

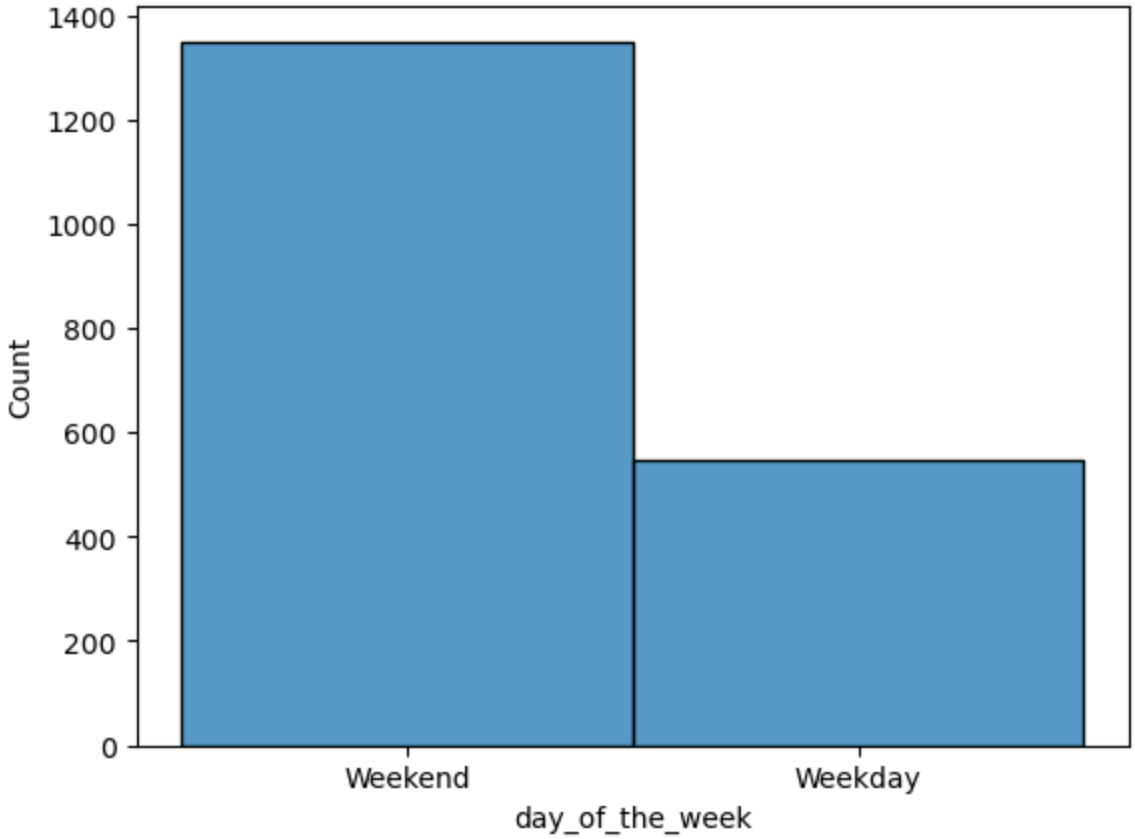

300
250
200
150
100
50
0
Count
15.0
17.5
20.0
22.5
25.0
27.5
30.0
32.5
delivery_time




```
C:\Users\conne\anaconda3\lib\site-packages\IPython\core\pylabtools.py:152: UserWarnin
g: Glyph 140 (\x8c) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\conne\anaconda3\lib\site-packages\IPython\core\pylabtools.py:152: UserWarnin
g: Glyph 142 (\x8e) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```







In [14]:
```
#Question 6
#This boxplot tells us  that over half the people ordering food order at a cost less t
#This histogram show that delivery time distubition is fairly even with some mild spik
#Based on the box plot we can see the food prep time is fairly even in disturbution wi
#It seems as though the higher preforming restaurants may have multiple locatons that
```

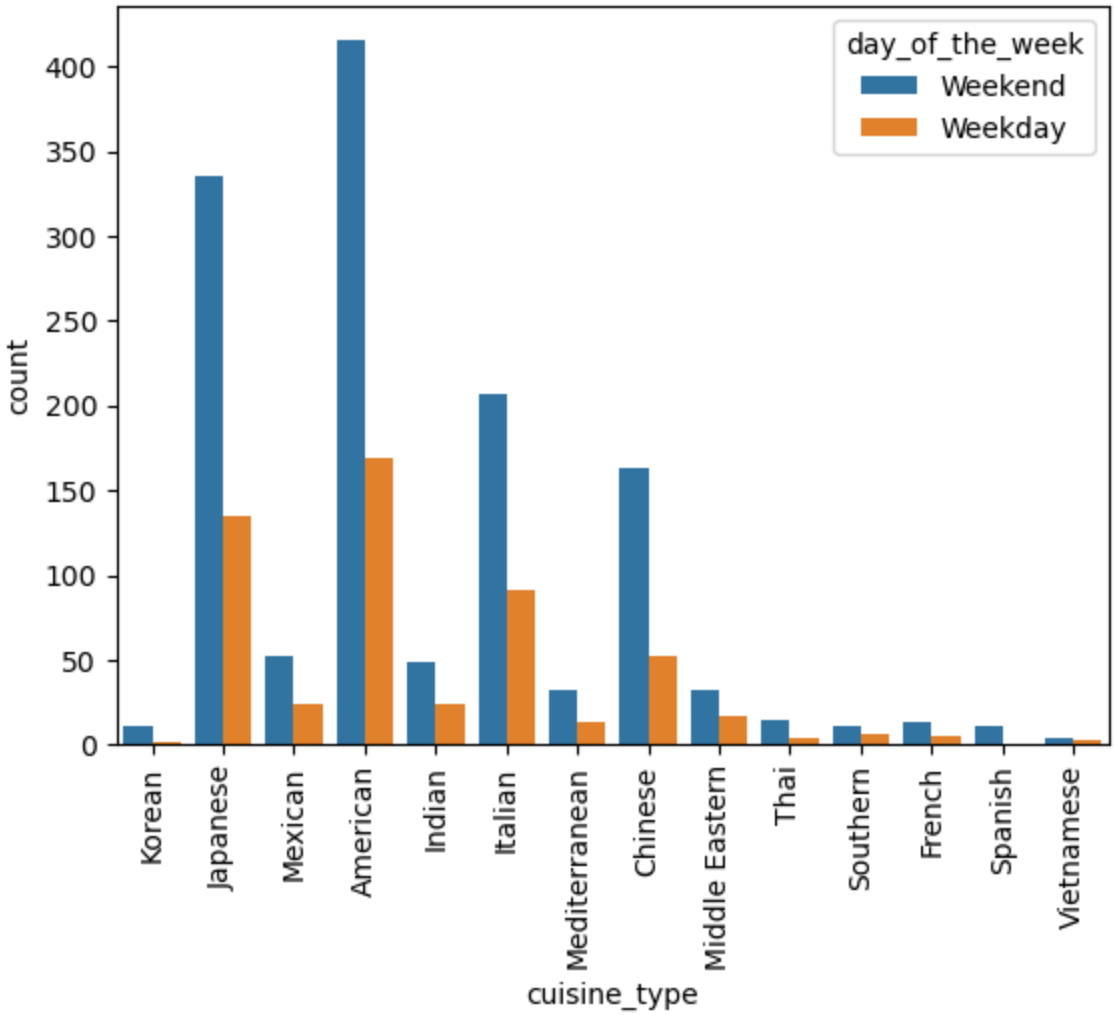
```
#Here we see rating is skedwed to the left. Note the graph doesnt include those with n
#We can see the lowest rating is 3 and is the smallest data point.
#This tells us over 60% of orders are placed on the weekend.
```

In [15]:
```
#Calling restaurant names in order of most frequent to least frequent.
df.value_counts('restaurant_name')
```

Out[15]:
```
restaurant_name
Shake Shack                  219
The Meatball Shop            132
Blue Ribbon Sushi            119
Blue Ribbon Fried Chicken     96
Parm                          68
                            ...
Klong                          1
Kambi Ramen House              1
Il Bambino                     1
Hunan Manor                    1
Lamarca Pasta                  1
Length: 178, dtype: int64
```

In [16]:
```
#Question 7
#Here we see the top 5 restaurants in terms of sales is from least of greatest are par
#top five are parm, blue ribbon fried chicken, blue ribbon sushi, the meatball shop, a
```

In [17]:
```
#Creating a countplot to compare weekend and weekday sales.
sns.countplot(data=df, x='cuisine_type', hue='day_of_the_week' )
plt.xticks(rotation=90)
plt.show()
```



In [18]:
```
#Question 8
#Here we can see that the most ordered cuisine type on weekends is american.
```

In [19]:
```
#Here i am asking for just the values greater than 20 then counting how many instances
df.query('cost_of_the_order > 20')['cost_of_the_order'].count()
```

Out[19]:
```
555
```

In [20]:
```
#Question 9
#We can divide 555 by 1,898 and getting .2924 meaning that 29.24% of orders costing mo
```

In [21]:
```
#This preforms statistical analysis of a specified column.
df.describe()['delivery_time']
```

Out[21]:
```
count    1898.000000
mean       24.161749
std         4.972637
min        15.000000
25%        20.000000
50%        25.000000
75%        28.000000
max        33.000000
Name: delivery_time, dtype: float64
```

In [22]:
```
#Question 10
#The mean of delivery time is 24.16
```

In [23]:
```
#this will give me a list of the number of times a customer orders from most frequent
df.value_counts('customer_id')
```

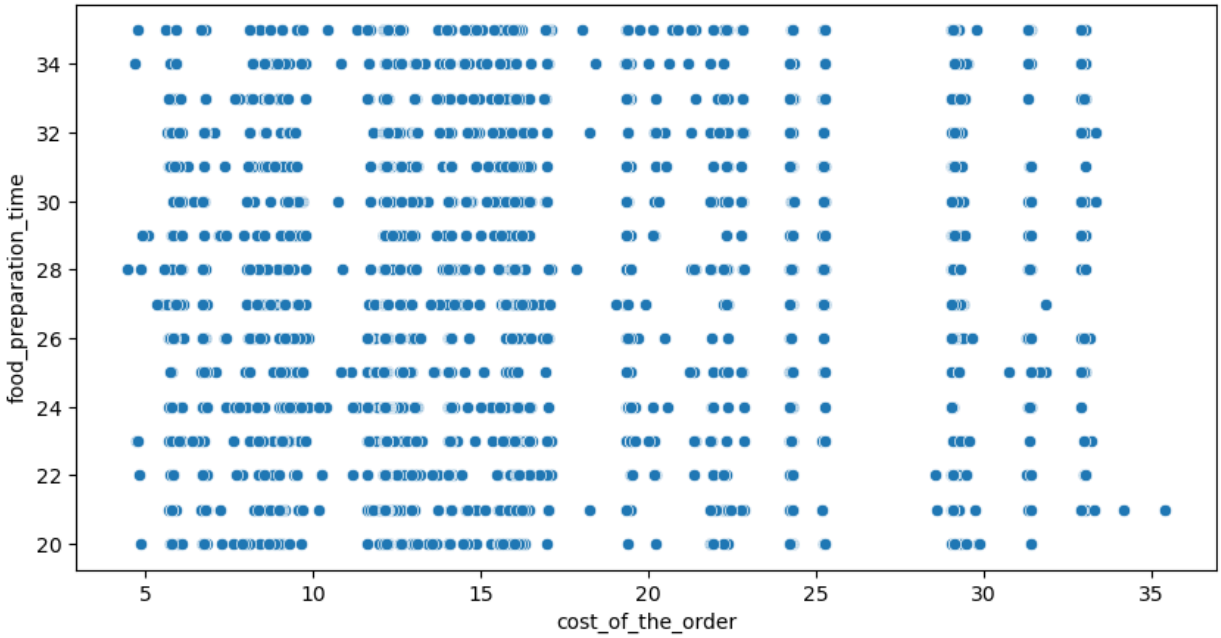
Out[23]:
```
customer_id
52832     13
47440     10
83287      9
250494     8
65009      7
          ..
105903     1
105992     1
106006     1
106324     1
405334     1
Length: 1200, dtype: int64
```

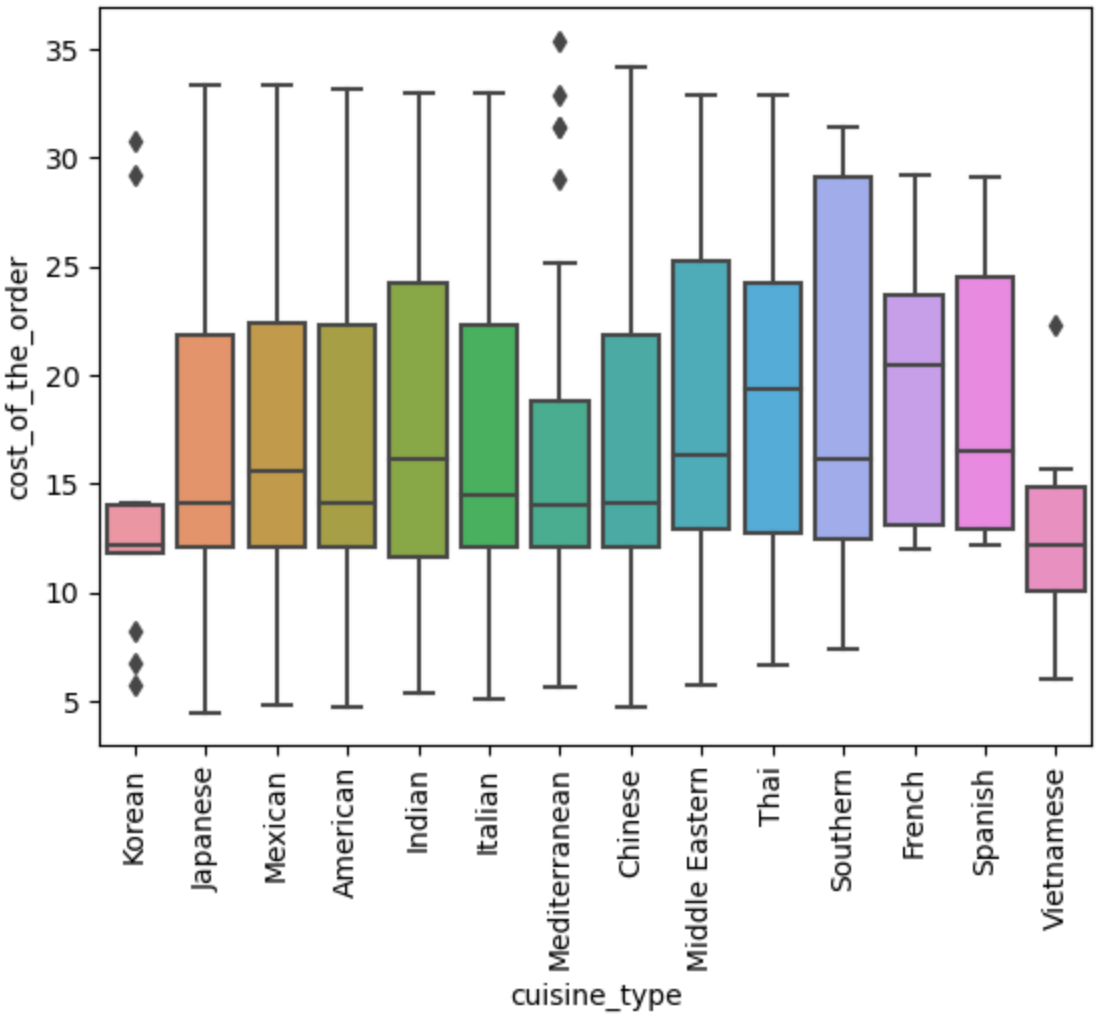
In [24]:
```
#Question 11
#the top 3 customers are,c_id 83287 with 9 orders, c_id 47440 with 10 orders, and c_id
```

In [26]:
```
#Checking to make sure it dropped.
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                 1162 non-null   float64
 7   food_preparation_time  1898 non-null   int64  
 8   delivery_time          1898 non-null   int64  
dtypes: float64(2), int64(4), object(3)
memory usage: 133.6+ KB
```

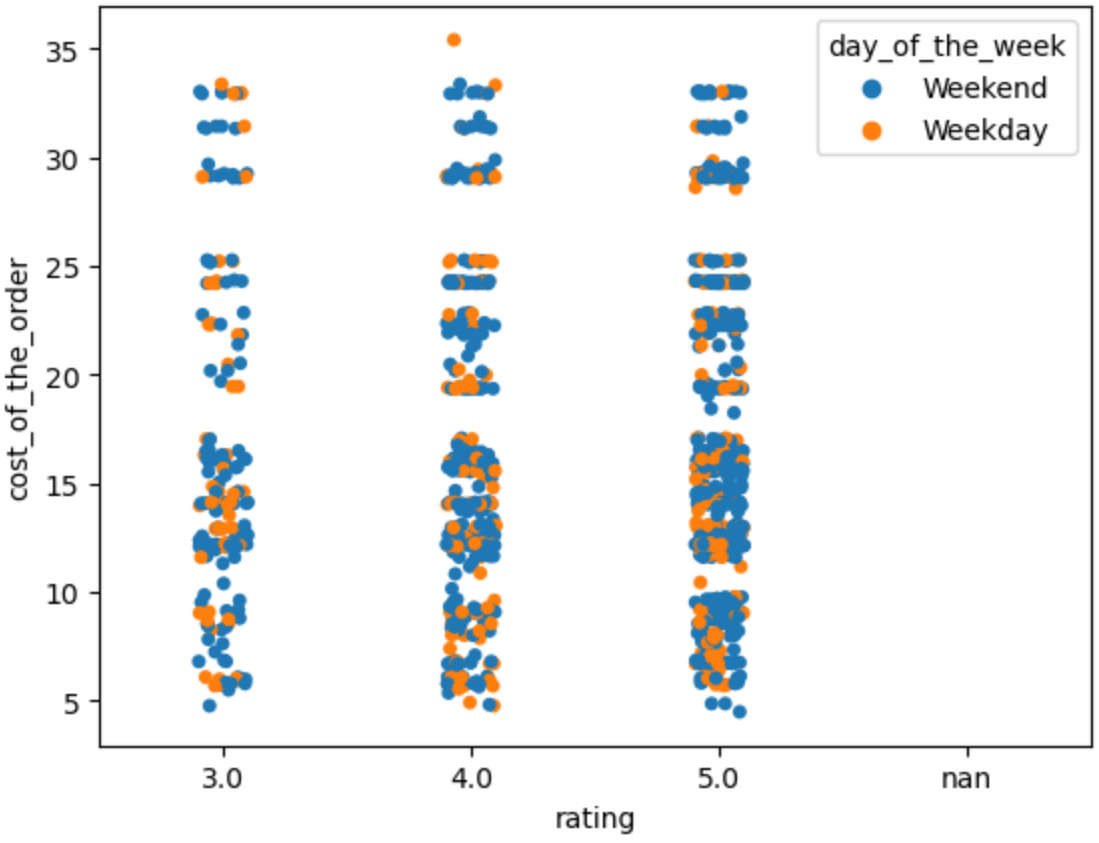
In [27]:
```
plt.figure(figsize=(10,5))
sns.scatterplot(data=df,x='cost_of_the_order',y='food_preparation_time')
plt.show()
sns.boxplot(data=df, x='cuisine_type', y='cost_of_the_order')
plt.xticks(rotation=90)
plt.show()
plt.figure(figsize=(10,5))
sns.heatmap(df.corr(),annot=True,cmap='Spectral',vmin=-1,vmax=1)
plt.show()
sns.stripplot(data=df, x='rating', y='cost_of_the_order', hue='day_of_the_week')
plt.show()
sns.catplot(data=df, x="cuisine_type", y="delivery_time", hue='day_of_the_week', kind=
plt.xticks(rotation=90)
plt.figure(figsize=(50,10))
plt.show()
```

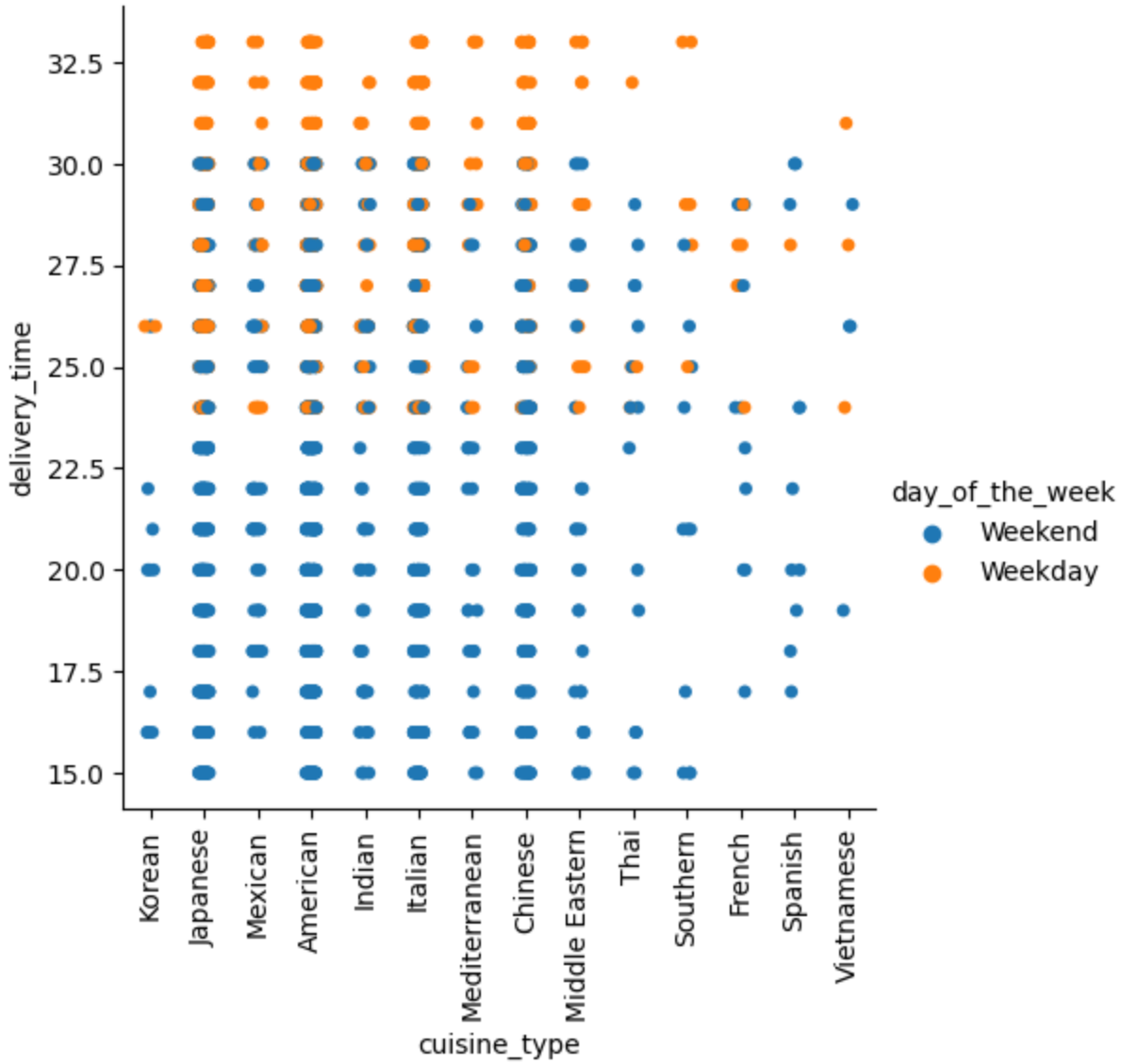





```
C:\Users\conne\AppData\Local\Temp\ipykernel_10972\1975719195.py:8: FutureWarning: The
default value of numeric_only in DataFrame.corr is deprecated. In a future version, i
t will default to False. Select only valid columns or specify the value of numeric_on
ly to silence this warning.
  sns.heatmap(df.corr(),annot=True,cmap='Spectral',vmin=-1,vmax=1)
```


order_id
customer_id
cost_of_the_order
rating
food_preparation_time
delivery_time
1 -0.014 0.021 0.0063 -0.007 0.0097
-0.014 1 -0.0013 0.0078 -0.03 -0.0099
0.021 -0.0013 1 0.034 0.042 -0.03
0.0063 0.0078 0.034 1 -0.0061 -0.0098
-0.007 -0.03 0.042 -0.0061 1 0.011
0.0097 -0.0099 -0.03 -0.0098 0.011 1
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00


day_of_the_week
Weekend
Weekday
cost_of_the_order
35
30
25
20
15
10
5
3.0
4.0
5.0
nan
rating




```
<Figure size 5000x1000 with 0 Axes>
```

In [28]:
```
#Question 12
#There appears to be no relation between cost of the order and food prep time.
#We can see that when comparing cost to cuisine type korean has the smallest IQR while
#almost all korean orders cost less than $15. Spanish and French share almost the same
#Korean, Mediterranean,and vietnamese all have outliers with only korean having an out
#We can see when using a heat map there is little correlation between the numeric colu
#The first strip plot shows that the density of 3 and 4 star ratings is higher on the
#The second strip plot shows that delivery time on weekend is much higher and cuisine
```

In [29]:
```
#I am asking for restaurant names with a raing greater than 2 then 3 then 4 to determi
q=df.query('rating > 2')['restaurant_name'].value_counts()
print(q)
r=df.query('rating > 3')['restaurant_name'].value_counts()
print(r)
a=df.query('rating > 4')['restaurant_name'].value_counts()
print(a)
```

```
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
Shake Shack                  110
The Meatball Shop             74
Blue Ribbon Sushi             57
Blue Ribbon Fried Chicken     53
RedFarm Broadway              33
                             ...
Izakaya Ten                    1
Taro Sushi                     1
Bhatti Indian Grill            1
Philippe Chow                  1
'wichcraft                     1
Name: restaurant_name, Length: 148, dtype: int64
Shake Shack                  60
The Meatball Shop            53
Blue Ribbon Fried Chicken    32
Blue Ribbon Sushi            32
RedFarm Broadway             18
                             ..
UVA Wine Bar & Restaurant     1
V-Nam Cafe                    1
Amma                          1
Hatsuhana                     1
'wichcraft                    1
Name: restaurant_name, Length: 121, dtype: int64
```

In [30]:
```
#Question 13
#Shake shack has 23 three star ratings, 26550 four star ratings, and 60 five star rati
#The meatball shop has 10 three star ratings, 21 four star ratings, and 53 five star r
#Blue ribbon sushi has 16 three star ratings, 25 four star ratings, and 32 five star r
#Blue ribbon fried chicken has 11 three star ratings, 21 four star ratings, and 32 fiv
#There are 4 restaurants that are eligible for the promo The meatball shop with an ave
```

In [31]:
```
#I am Looking for the cost of order under $5, between $5-$20, over $20.
L=df.query('cost_of_the_order <5.01').sum()
print(L)
K=df.query('cost_of_the_order <20.01').sum()
print(K)
P=df.query('cost_of_the_order >20').sum()
print(P)
```

```
order_id                                                    13297784
customer_id                                                  1405698
restaurant_name          Shake ShackCafe HabanaThe LoopP.J. Clarke'sNob...
cuisine_type             AmericanMexicanJapaneseAmericanJapaneseJapanes...
cost_of_the_order                                             42.74
day_of_the_week          WeekdayWeekendWeekendWeekendWeekendWeekendWeek...
rating                                                         30.0
food_preparation_time                                           242
delivery_time                                                   218
dtype: object

order_id                                                   1984260070
customer_id                                                 230054328
restaurant_name          Blue Ribbon Sushi IzakayaCafe HabanaDirty Bird...
cuisine_type             JapaneseMexicanAmericanItalianMediterraneanInd...
cost_of_the_order                                          16559.91
day_of_the_week          WeekendWeekdayWeekdayWeekendWeekdayWeekdayWeek...
rating                                                      3493.0
food_preparation_time                                         36654
delivery_time                                                 32533
dtype: object

order_id                                                    820026389
customer_id                                                  94823444
restaurant_name          HangawiBlue Ribbon Fried ChickenTamarind TriBe...
cuisine_type             KoreanAmericanIndianAmericanJapaneseAmericanAm...
cost_of_the_order                                          14754.91
day_of_the_week          WeekendWeekendWeekdayWeekendWeekendWeekendWeek...
rating                                                      1555.0
food_preparation_time                                         15298
delivery_time                                                 13326
dtype: object
```

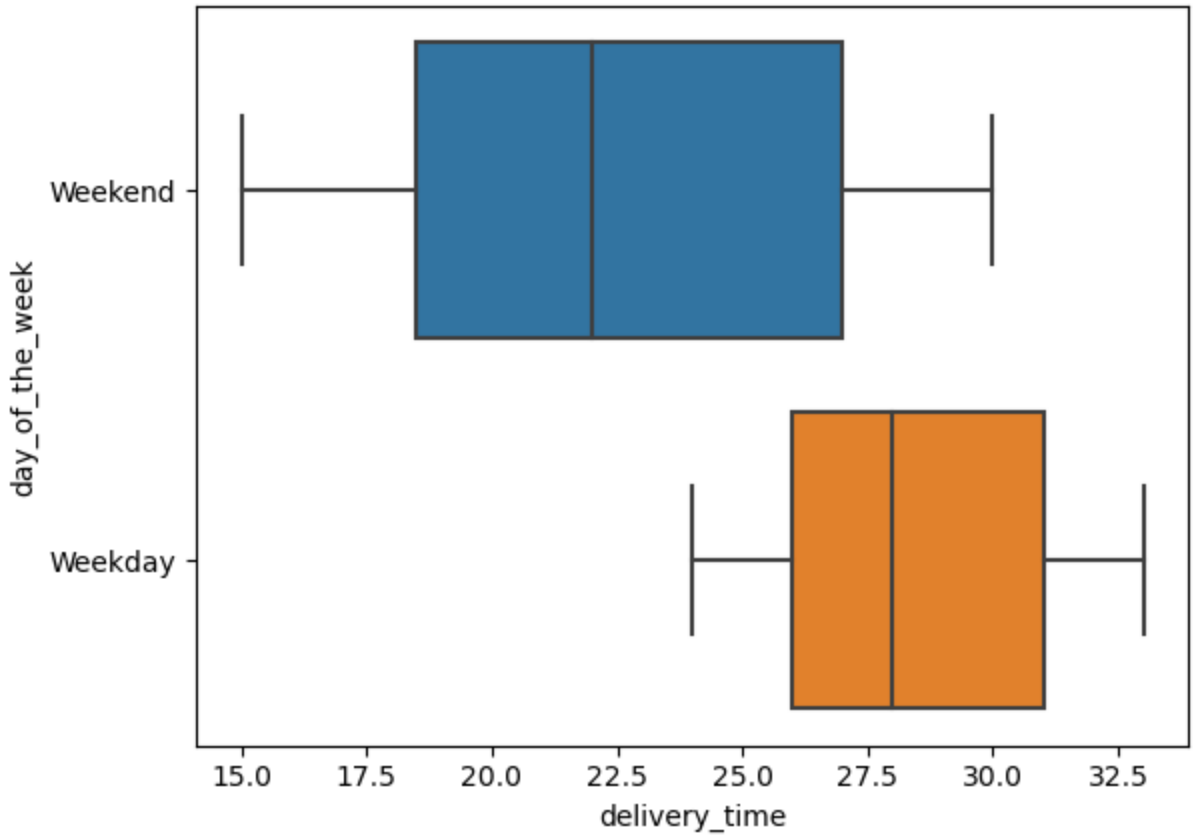
In [32]:
```
#Question 14
# $42.74 was made from orders under $5
# By subtracting 16,559.19 from 42.74 we can see $16,517.17 was made from orders costi
# $14,754.91 was made from orders over $20
#Net revenue from orders is $6,166.30
```

In [33]:
```
#I created a var named total time that had both food prep time and delivery time added
total_time=df['food_preparation_time']+df['delivery_time']
new_total_time=0
for var in total_time:
    if var>60:
        new_total_time=new_total_time+1
print(new_total_time)
```

```
200
```

In [34]:
```
#Question 15
#There are 200 out of 1898 instances where total time was above 60 minutes and when di
# 10.54% of orders take longer than 60 minutes.
```

In [35]:
```
#This shows a boxplot comparing weekday and weekend delivery times.
sns.boxplot(data=df, x="delivery_time", y='day_of_the_week')
plt.show()
```



In [36]:
```
#Question 16
#we can see that weekend has a mean of roughly 22.5 minutes and week day has a mean of
#The weekend data is less dense with faster times where as the week day data is more a
```

In [37]:
```
#Question 17
# In conclusion it seems as though lower cost items preform better,cost has little inp
#I recommend that the company look at promoting mpre expensive items because depite it
#I also recommend running a program to incentivize people to leave ratings as currentl
```