FoodHub Data Analysis

Project 1 - Python Programming

Date: May 24, 2026

Kimberly Wiethoff

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Contents / Agenda

- Executive Summary
- Business Problem Overview and Solution Approach
- Recommendations – Executive Summary
- Data Overview
- EDA - Univariate Analysis
- EDA - Multivariate Analysis
- Appendix – Python Code and Visualizations

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Executive Summary

Objective – Analyze customer data for the Food Hub platform to:

- Understand ordering behavior for customers
- Identify the restaurants and cuisines that are performing best
- Improve customer satisfaction
- Optimize delivery operations
- Increase revenue opportunities

Conclusion – Recommendations are based on the categories listed below

Recommendations

- Delivery and Preparation Time
- Revenue Review Model
- Top Rated Cuisines
- Day of the Week Comparison
- Customer Order Frequency and Repeat Business

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Business Problem Overview

FoodHub operates in a highly competitive food delivery market where customer experience, delivery efficiency, and restaurant performance directly impact revenue growth and customer retention. As order volumes continue to increase, the company requires greater visibility into customer ordering behavior, restaurant demand, delivery performance, and the factors influencing customer satisfaction.

To support strategic decision-making, operational and customer order data will be analyzed to identify high-performing restaurants and cuisines, uncover the key drivers of customer ratings, detect delivery and preparation bottlenecks, and evaluate revenue and demand trends across weekdays and weekends.

The goal of this analysis is to provide actionable, data-driven insights that enhance the customer experience, optimize delivery operations, strengthen restaurant partnerships, and support sustainable business growth.

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Solution Approach / Methodology

The data analysis approach uses exploratory data analysis (EDA) techniques to evaluate customer orders, restaurant performance, operational efficiency, and customer satisfaction trends within the FoodHub platform.

The solution methodology includes:

- Data validation and cleaning to address missing values and inconsistencies
- Descriptive statistical analysis to identify order, revenue, and customer behavior trends
- Comparative analysis of restaurants, cuisines, weekday versus weekend, revenue, and customer ratings
- Operational performance analysis focused on food preparation and delivery times
- Correlation analysis to identify factors influencing customer satisfaction and business performance
- Data visualization techniques to highlight trends, bottlenecks, and strategic opportunities

The final output will translate analytical findings into actionable business recommendations to improve operational efficiency, revenue growth, and customer experience.

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Recommendations

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Executive Recommendations Overview

Key Strategic Priorities

- Improve delivery efficiency and customer satisfaction by reducing total order fulfillment time
- Optimize the revenue model to better align higher-value orders with revenue growth
- Leverage top-rated cuisines to drive customer acquisition and retention
- Increase customer repeat business through loyalty and personalization strategies
- Optimize operations and marketing strategies based on weekday and weekend demand trends

Anticipated Outcomes

- Faster delivery and preparation times
- Higher customer satisfaction and ratings
- Increased repeat customer activity
- Improved restaurant partner performance
- Stronger revenue growth and operational scalability

Strategic Value These recommendations position FoodHub to improve operational efficiency, strengthen customer loyalty, and create a sustainable competitive advantage in the food delivery market

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Delivery & Operational Efficiency Recommendations

Key Insights

- total_time is strongly correlated with:
  - delivery_time (0.74)
  - food_preparation_time (0.69)
- Total fulfillment time is a key driver of customer experience

Recommendations

- Optimize delivery routing, driver allocation, and logistics processes
- Partner with restaurants to improve food preparation efficiency
- Monitor delivery and preparation KPIs to identify bottlenecks
- Prioritize operational improvements during peak demand periods

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Revenue Optimization Recommendations

Key Insights

- cost_of_the_order and revenue show minimal correlation (-0.03).
- Current commission thresholds may not maximize revenue from higher-value orders

Recommendations

- Reassess commission percentages and pricing thresholds
- Evaluate alternative revenue models for premium or high-value orders
- Introduce dynamic pricing or tiered commission strategies
- Analyze profitability by cuisine, restaurant, and order size

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Cuisine Opportunities Recommendations

Key Insights

- **Top-Rated Cuisines -** Spanish. Thai, Indian, Mexican
- **Lowest-Rated Cuisines** – Vietnamese, Korean, Mediterranean, Middle Eastern

Recommendations

- Feature top-rated cuisines prominently in marketing campaigns and app placement
- Partner closely with high-performing restaurants for promotions and visibility
- Prioritize and incentivize onboarding similar cuisine types in expansion markets
- Investigate lower-rated cuisines to identify quality or service gaps
- Determine ways to encourage customers to give reviews and avoid the ‘Not Given’ option

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Customer Retention & Repeat Business Recommendations

Key Insights

- Average customer order frequency is only 1.58 orders
- Significant opportunity exists to improve customer retention

Recommendations

- Launch tiered loyalty and rewards programs
- Expand targeted promotions beyond top customers
- Develop win-back campaigns for infrequent users
- Improve first-order experience to encourage repeat purchases
- Use personalized recommendations based on order history
- Explore subscription or membership models with delivery benefits

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Day of Week Operational Recommendations

Key Insights

- Weekend order volume (1,351 orders) is more than double weekday volume (547 orders)
- American cuisine is the most popular cuisine on weekends
- Spanish and Thai cuisines generate higher average weekend order values
- Weekday delivery times average 28.34 minutes compared to 22.47 minutes on weekends
- Vietnamese cuisine receives lower ratings during weekdays

Recommendations

- Increase delivery staffing and customer support coverage during weekends
- Launch targeted weekend promotions for popular and high-value cuisines
- Investigate weekday delivery inefficiencies, including routing, staffing, and traffic impacts
- Work with Vietnamese restaurant partners to improve weekday customer experience and service quality

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Data Analysis

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Data Overview

Historical customer order information is provided by FoodHub, which captures key operational, customer, and restaurant-related metrics associated with each order transaction.

The dataset includes:

- Order and customer identifiers
- Restaurant and cuisine information
- Order cost details
- Weekday versus weekend ordering activity
- Customer ratings and satisfaction data
- Food preparation time
- Delivery performance metrics

Analysis of customer ordering behavior, restaurant demand, operational efficiency, delivery performance, and factors that influence customer satisfaction will help determine operational insights and strategic business recommedations to improve service quality and business performance.

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Understanding the Structure of the Data

- There are 1989 rows and 9 columns in the Foodhub_order.csv file
  - There are 1200 unique customers
  - Customers placed an average of 1.58 orders each, with order frequency ranging from a minimum of 1 order to a maximum of 13 orders per customer
  - There are 178 unique restaurants with 14 unique cuisine types
  - Food preparation time averages 27.37 minutes, with times ranging from a minimum of 20 minutes to a maximum of 35 minutes
  - There are 547 Weekday orders and 1351 Weekend orders
  - The food preparation and delivery process takes an average of 51.53 minutes, with completion times ranging from a minimum of 35 minutes to a maximum of 68 minutes
  - Order costs averaged $16.50, with values ranging from a minimum of $4.47 to a maximum of $35.41.
- There are 3 datatypes in the data: int64, object, and float64
- There are no missing values in the data
- However, the rating column does include ‘Not Given’ for 736 orders

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Univariate Analysis

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Univariate Analysis

Univariate analysis examines the distribution and characteristics of individual variables to identify patterns.

- **Customer ID and Order ID** analysis did not provide meaningful analytical insights because these fields are identifier columns rather than measurable business variables, despite being stored as numeric data
- **Order Cost** distribution is right-skewed, with most orders ranging between $11 and $17
- **Order volume** showed concentrations around $11, $17, $25, and $30, while most orders fall within the $11 to $23 price range
- **Food preparation time** distribution indicates that most orders are prepared within 20 to 35 minutes
- **Delivery time** analysis shows that most orders were delivered in either 25 or 28 minutes
- **Total time** distribution, calculated as the sum of food preparation and delivery time, is approximately normal, with most orders completed within 48 to 56 minutes
- There were no significant outliers in the visualizations of the numeric values

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Univariate Analysis (Continued)

- **Weekend order volume** is more than double that of weekday orders
- For orders with **customer ratings**, the distribution is left-skewed, with most ratings between 3 and 5, with the highest concentration at 5
- **Top 5 restaurants** are Shake Shack, The Meatball Shop, Blue Ribbon Sushi, Blue Ribbon Fried Chicken, and Parm
- **American cuisine** is the most popular cuisine on weekends, with 415 orders
- **29.24%** of the **orders cost more than 20 dollars**
- The **average order delivery time** is 24.16 minutes
- Customers 52832, 47440, and 83287 were the **most frequent users**, placing 13, 10, and 9 orders, respectively

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Multivariate Analysis

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Multivariate Analysis

Multivariate analysis examines relationships and interactions among multiple variables to identify correlations, patterns, and factors that can influence business outcomes.

- The **cost of the order on the weekend** as compared to the weekday did not show a recognizable difference
- The **average order cost across cuisines** ranged from $13 to $20. Mediterranean cuisine had the highest-priced maximum order at $35.41, while Vietnamese cuisine had the lowest maximum order at $22.26
- **Spanish and Thai cuisines** showed the average order cost was noticeably higher on weekends compared to weekdays
- For all other cuisines, the difference in order cost between **weekends and weekdays** was insignificant
- The **restaurants where the rating count is greater than 50, and the average rating is greater than 4** are Blue Ribbon Fried Chicken, Shake Shack, Blue Ribbon Sushi, and The Meatball Shop
- There were no weekday ratings for Spanish cuisine
- Vietnamese cuisine received significantly lower ratings on weekdays, while ratings for all other cuisines showed no meaningful difference between weekdays and weekends

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Multivariate Analysis (Continued)

- **Total revenue** generated was $6,166.30, with revenue per order ranging from $0.00 to a maximum of $8.85 and an average of $3.25 per order
- 9 orders generated no revenue
- **American, Japanese, Italian, and Chinese cuisines** earned the most revenue, while Vietnamese, Korean, Spanish, and Southern cuisine earned the least revenue
- 10.5% of the orders take more than 60 minutes to get delivered from the time the order is placed
- The average (mean) delivery time is 28.34 minutes on weekdays and 22.47 minutes on weekends
- **Spanish cuisine, Thai, Indian, and Mexican cuisines** were rated the highest in customer ratings in that order, while Vietnamese, Korean, Mediterranean, and Middle Eastern ranked the lowest

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Correlation Analysis

Focus on Delivery and Preparation Time: total_time has a strong positive correlation with both delivery_time (0.74) and food_preparation_time (0.69). To improve overall delivery efficiency and customer satisfaction (which is often linked to total delivery time), the company should prioritize optimizing both food preparation and delivery processes.

- **Delivery Optimization:** Investigate routes, delivery person allocation, and logistics to reduce delivery times.
- **Food Preparation Efficiency:** Work with restaurants to streamline food preparation without compromising quality.

Review Revenue Model: The order cost and revenue show a very low correlation (-0.03). Given that revenue is calculated based on the cost of the order, this suggests that the current revenue structure (25% for orders > $20, 15% for orders > $5) might not be effectively translating higher order values into proportionally higher revenue. The company should review its revenue model to ensure it aligns with business goals and potentially adjust the percentage commissions or thresholds to optimize earnings from orders across different price ranges.

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Top-Rated Cuisines Analysis

- **Highlight Top-Rated Cuisines:** Spanish, Thai, Indian, and Mexican cuisines consistently receive the highest average ratings. The company should leverage these insights in their marketing campaigns, highlighting the exceptional quality and customer satisfaction associated with these cuisine types. This can attract new customers and reinforce loyalty among existing ones.
- **Promote Restaurants Specializing in Top Cuisines:** Work closely with restaurants offering these top-rated cuisines. Consider offering special promotions or featured placements within the app for these establishments to further boost their visibility and order volume.
- **Investigate Lower-Rated Cuisines:** Explore the reasons behind lower ratings for cuisines like Vietnamese, Korean, Mediterranean, and Middle Eastern. This could involve gathering more specific customer feedback, assessing food quality, preparation times, or delivery experience for these categories, and working with partner restaurants to implement improvements.
- **Expansion Opportunities:** If the company plans to expand to new areas or onboard new restaurants, prioritize partners offering these highly-rated cuisines to maintain a high standard of customer satisfaction and capitalize on proven popularity.

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Day of the Week Analysis

- **Optimize Weekend Operations for Higher Demand:** Weekends have significantly more orders (1351) compared to weekdays (547). This indicates a higher demand during these periods. The company should ensure adequate staffing for delivery personnel and customer support on weekends to handle the increased volume and maintain service quality.
- **Capitalize on Weekend Cuisine Popularity:** American cuisine is the most popular on weekends. This insight can be used for targeted marketing campaigns during weekends, perhaps offering special deals on popular American dishes or promoting American restaurants. For Spanish and Thai cuisines, where weekend average order costs are noticeably higher, consider special weekend promotions to further boost these high-value orders.
- **Improve Weekday Delivery Efficiency:** The average delivery time is longer on weekdays (28.34 minutes) compared to weekends (22.47 minutes). Investigate the reasons for this difference, which could include traffic patterns, delivery route optimization, or staffing levels during peak weekday hours. Strategies to reduce weekday delivery times could significantly improve customer satisfaction.
- **Address Weekday Rating Discrepancies:** While most cuisines show no significant difference in ratings between weekdays and weekends, Vietnamese cuisine receives significantly lower ratings on weekdays. This is an area that needs immediate attention. The company should investigate the specific issues affecting Vietnamese cuisine on weekdays and work with partner restaurants to improve quality or service during these times.

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Customer Order Frequency and Repeat Business

- **Targeted Loyalty Programs:** With customers averaging only 1.58 orders, there is strong potential to increase repeat business. Expand the current 20% discount voucher initiative for top customers (52832, 47440, 83287) into a tiered rewards program based on order frequency or total spend to engage a broader customer base.
- **Win-Back Campaigns for Infrequent Customers:** Many customers likely order only once or twice. Re-engage them with personalized promotions, discounts on future orders, and recommendations for new restaurants or cuisines aligned with their past preferences.
- **Enhance the First-Order Experience:** Since repeat ordering is low, creating an excellent first impression is critical. Focus on fast delivery, order accuracy, and responsive customer support to improve retention and encourage second orders.
- **Personalized Recommendations:** Use historical order data to deliver tailored restaurant and cuisine suggestions, making it easier for customers to discover relevant options and increasing the likelihood of repeat purchases.
- **Customer Feedback & Service Improvements:** Continuously gather and analyze feedback, especially from low-frequency customers, to identify pain points and improve delivery, service quality, and customer satisfaction.
- **Subscription or Membership Benefits:** Introduce a membership program offering perks such as free delivery, exclusive discounts, and early access to restaurant partners to encourage recurring usage and generate more predictable revenue.

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Appendix Python Code and Visualizations

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Libraries and Google Drive

```
# Installing the libraries with the specified version.
!pip install numpy==2.0.2 pandas==2.2.2 matplotlib==3.10.0 seaborn==0.13.2 -q --user
```

```
# 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

```
# uncomment and run the below code snippets if the dataset is present in the Google Drive
from google.colab import drive
drive.mount('/content/drive')
```

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

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Data Analysis

```
# Write your code here to view the first 5 rows
df.head()
```

	order_id	customer_id	restaurant_name	cuisine_type	cost_of_the_order	day_of_the_week	rating	food_preparation_time	delivery_time
0	1477147	337525	Hangawi	Korean	30.75	Weekend	Not given	25	20
1	1477685	358141	Blue Ribbon Sushi Izakaya	Japanese	12.08	Weekend	Not given	25	23
2	1477070	66393	Cafe Habana	Mexican	12.23	Weekday	5	23	28
3	1477334	106968	Blue Ribbon Fried Chicken	American	29.20	Weekend	3	25	15
4	1478249	76942	Dirty Bird to Go	American	11.59	Weekday	4	25	24

```
# number of unique customers
df['customer_id'].nunique()
```

```
1200
```

```
# count the number of orders per customer
df['customer_id'].value_counts()
```

customer_id	count
52832	13
47440	10
83287	9
250494	8
259341	7
...	...
143926	1
89574	1
157711	1
66393	1
358141	1

1200 rows × 1 columns

dtype: int64

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Data Analysis

```
# count the max and min and average number of orders per customer
print(df['customer_id'].value_counts().max())
print(df['customer_id'].value_counts().min())
print(df['customer_id'].value_counts().mean())
```

```
13
1
1.5816666666666668
```

```
# count the number of unique restaurants
df['restaurant_name'].nunique()
```

```
178
```

```
# count the number of quisine types
df['cuisine_type'].nunique()
```

```
14
```

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Data Analysis

```
# count the number of orders where day of the week = weekday and day of the week = weekend
print(df[df['day_of_the_week'] == 'Weekday'].shape[0])
print(df[df['day_of_the_week'] == 'Weekend'].shape[0])
```

```
547
1351
```

```
# use a function to calculate total time
def total_time(prep, delivery):
  return prep + delivery
#use the function total time to get the maximum, minimum, and average total_time
df['total_time'] = total_time(df['food_preparation_time'], df['delivery_time'])
print(df['total_time'].max())
print(df['total_time'].min())
print(df['total_time'].mean())
```

```
68
35
51.53371970495258
```

```
# calculate the maximum, minimum, and average cost of the order
print(df['cost_of_the_order'].max())
print(df['cost_of_the_order'].min())
print(df['cost_of_the_order'].mean())
```

```
35.41
4.47
16.498851422550054
```

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Data Analysis

```
# Write your code here
df.shape
```

```
(1898, 9)
```

Observations: There are 1898 rows and 9 columns in the data

```
# Write your code here
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
```

Observations: There are 3 datatypes in the data: int64, object, and float64

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Data Analysis

```
# Write your code here
df.isnull().sum()
```

	0
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

Observations: There are no missing values in the data.

```
# Write your code here
df.describe().T
print("Average = ", df['food_preparation_time'].mean())
print("Minimum = ", df['food_preparation_time'].min())
print("Maximum = ", df['food_preparation_time'].max())
```

```
Average =  27.371970495258168
Minimum =  20
Maximum =  35
```

Observations: Minimum = 20.00, Maximum = 35.00, Average = 27.37

Average = 27.371970495258168, Minimum = 20, Maximum = 35

```
# Write the code here
df.loc[df['rating'] == 'Not given'].shape[0]
```

```
736
```

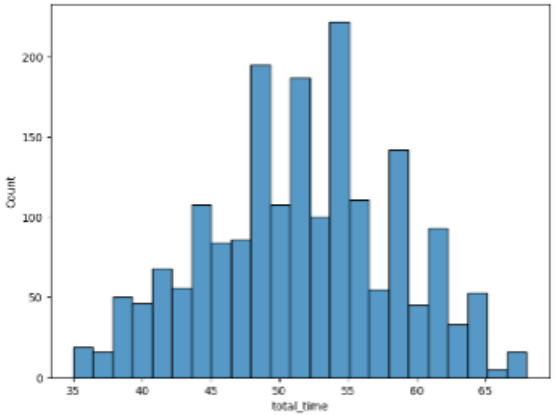
Observations: 736

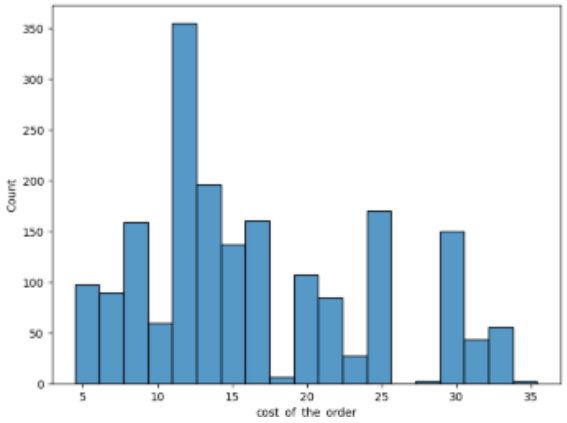
Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

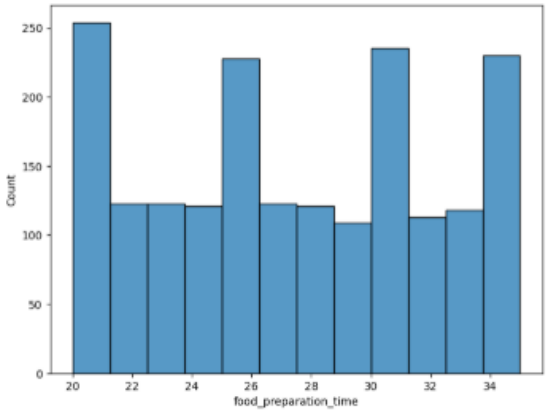
Histogram Plots

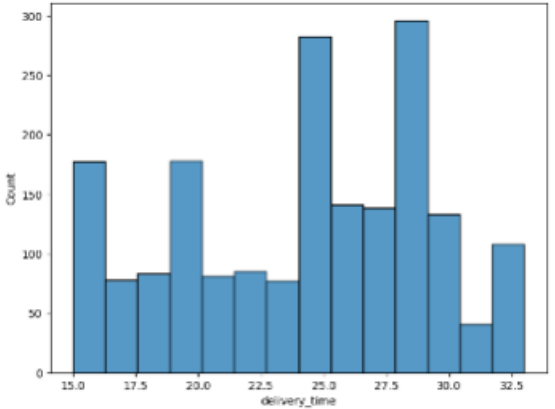
```
# Write the code here
# get the columns that are numeric
numeric_cols = df.select_dtypes(include=['int64', 'float64']).columns

# explore the numeric columns to provide observations on their distributions
for col in numeric_cols:
    plt.figure(figsize=(8, 6))
    sns.histplot(data=df, x=col)
    plt.show()
```









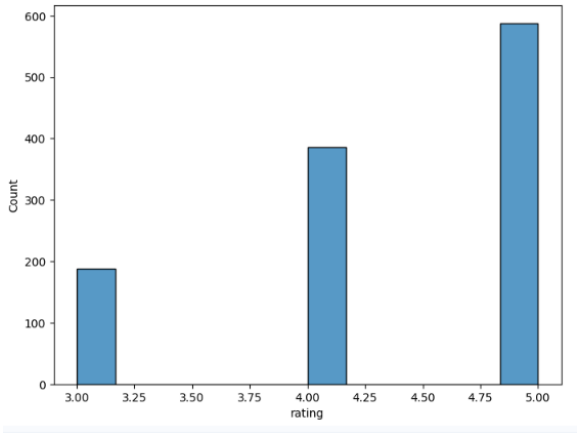
Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

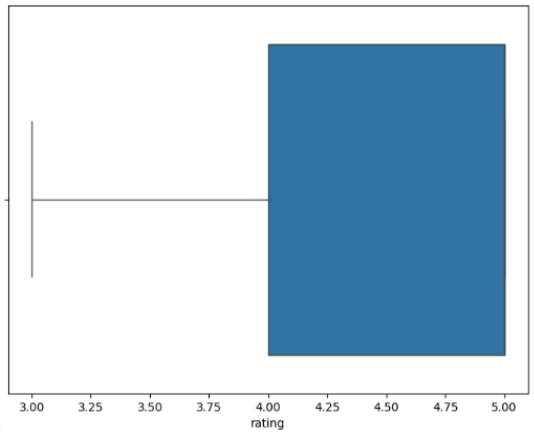
Ratings

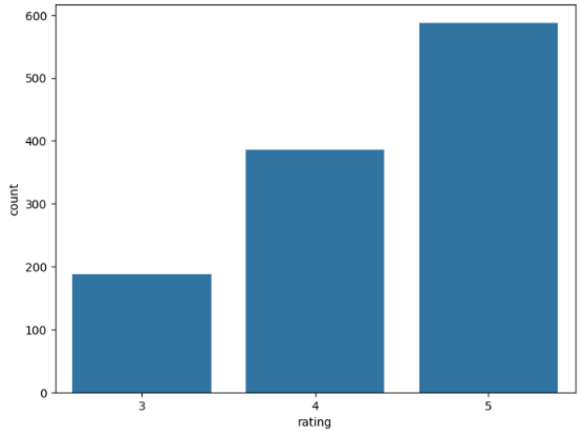
```
# Explore the ratings column.  First we need to create a new dataframe
df_ratings = df.drop(df[df['rating'] == 'Not given'].index)
df_ratings['rating'] = df_ratings['rating'].astype(int)
df_ratings.shape
```

```
(1162, 11)
```

```
# do the histogram for the column ratin g in the df_ratings dataframe
plt.figure(figsize=(8, 6))
sns.histplot(data=df_ratings, x='rating')
plt.show()
# boxplot
plt.figure(figsize=(8, 6))
sns.boxplot(data=df_ratings, x='rating')
plt.show()
# countplot()
plt.figure(figsize=(8, 6))
sns.countplot(data=df_ratings, x='rating')
plt.show()
```







Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Top 5 Restaurants (Orders Received)

```
# Write the code here
df['restaurant_name'].value_counts().head(5)
```

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

dtype: int64

Observations:

The top 5 restaurants include: Shake Shack, The Meatball Shop, Blue Ribbon Sushi, Blue Ribbon Fried Chicken, and Parm

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Percentage of Orders > $20, Most popular Cuisine (Weekends)

```
# Write the code here
(df['cost_of_the_order'] > 20).mean() * 100
```

```
np.float64(29.24130663856691)
```

Observations:

29.24% of the orders cost more than 20 dollars.

```
# Write the code here
df[df['day_of_the_week'] == 'Weekend']['cuisine_type'].value_counts().head(1)
```

cuisine_type	count
American	415

dtype: int64

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Average Delivery Time and Top 3 Customers

```
# Write the code here
df['delivery_time'].mean()
```

```
np.float64(24.161749209694417)
```

Observations:

The mean order delivery time is 24.16 minutes.

```
# Write the code here
df['customer_id'].value_counts().head(3)
```

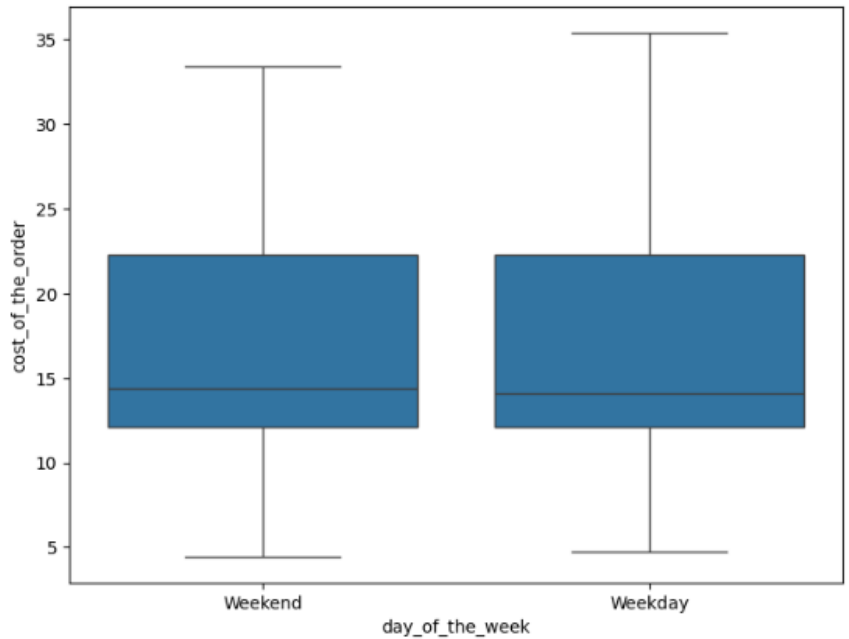
customer_id	count
52832	13
47440	10
83287	9

dtype: int64

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Order Cost – Day of the Week Boxplot

```
# explore the relationship between the day_of_the_week and the cost_of_the_order
plt.figure(figsize=(8, 6))
sns.boxplot(data=df, x='day_of_the_week', y='cost_of_the_order')
plt.show()
```



Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Order Cost – Cuisine Type – Max, Min, Average

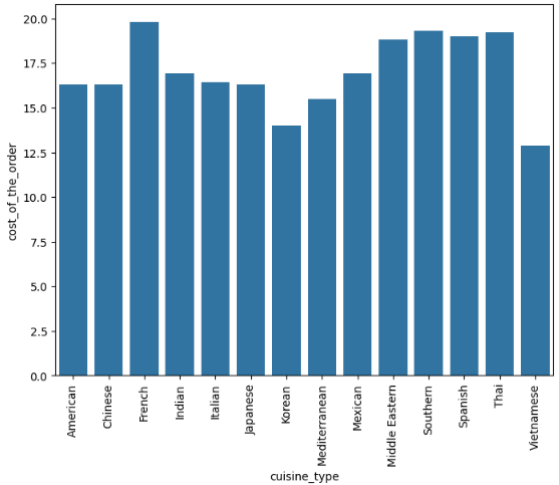
```
# find the max and min cost of the order for cuisine_type and sort by max
df.groupby('cuisine_type')['cost_of_the_order'].agg(['max', 'min', 'mean']).sort_values(by='max', ascending=False)
```

cuisine_type	max	min	mean
Mediterranean	35.41	5.67	15.474783
Chinese	34.19	4.75	16.305209
Japanese	33.37	4.47	16.304532
Mexican	33.32	4.85	16.933117
American	33.18	4.71	16.319829
Indian	33.03	5.34	16.919726
Italian	33.03	5.05	16.418691
Middle Eastern	32.93	5.77	18.820612
Thai	32.93	6.69	19.207895
Southern	31.43	7.38	19.300588
Korean	30.75	5.77	14.001538
French	29.25	11.98	19.793889
Spanish	29.10	12.13	18.994167
Vietnamese	22.26	6.01	12.882857

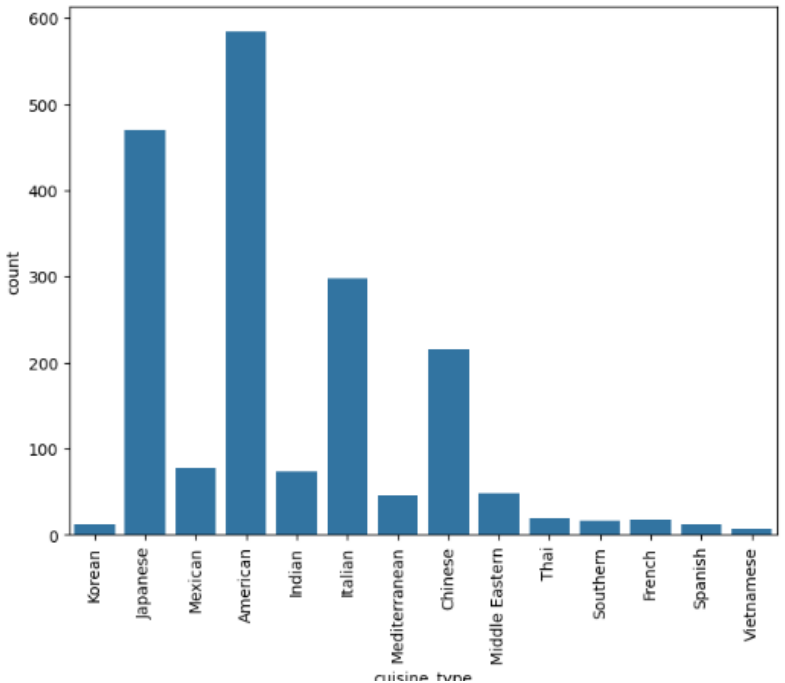
Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Cuisine Type – Average Cost, Count of Orders

```
# create a histogram for the average cost of the order for cuisine_type
plt.figure(figsize=(8, 6))
sns.barplot(
    data=df.groupby('cuisine_type')['cost_of_the_order'].mean().reset_index(),
    x='cuisine_type',
    y='cost_of_the_order')
plt.xticks(rotation=90)
plt.show()
```



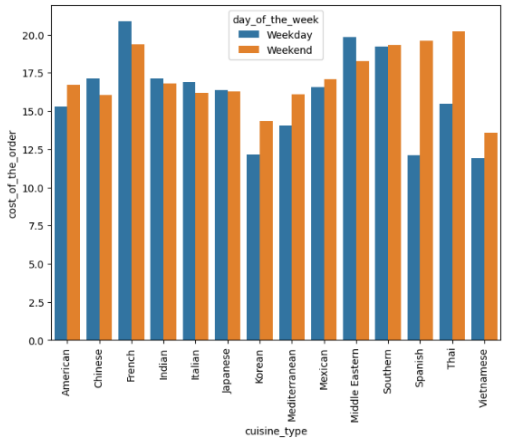

```
# create a countplot for cuisine_type rotate the titles in the x axis by 90 degrees
plt.figure(figsize=(8, 6))
sns.countplot(data=df, x='cuisine_type')
plt.xticks(rotation=90)
plt.show()
```



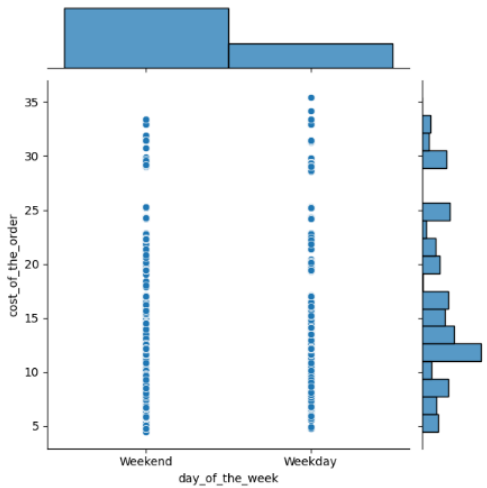
Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Order Cost – Day of Week by Cuisine Type

```
# plot the average cost of the order for weekend and weekdays by cuisine and set the color to be different for each
plt.figure(figsize=(8, 6))
sns.barplot(
    data=df.groupby(['day_of_the_week', 'cuisine_type'])['cost_of_the_order'].mean().reset_index(),
    x='cuisine_type',
    y='cost_of_the_order',
    hue='day_of_the_week')
plt.xticks(rotation=90)
plt.show()
```



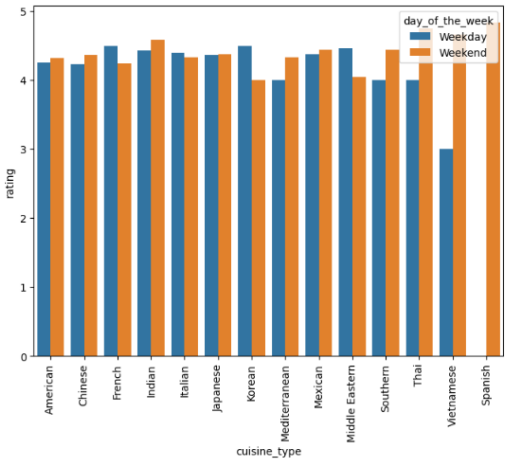

```
# create a jointplot for day of the week and cost of the order
sns.jointplot(data=df, x='day_of_the_week', y='cost_of_the_order')
plt.show()
```



Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Ratings – Day of Week by Cuisine Type - Average

```
# plot the rating in df_ratings for weekend and weekdays by cuisine and set the color to be different for each
plt.figure(figsize=(8, 6))
sns.barplot(
    data=df_ratings.groupby(['day_of_the_week', 'cuisine_type'])['rating'].mean().reset_index(),
    x='cuisine_type',
    y='rating',
    hue='day_of_the_week')
plt.xticks(rotation=90)
plt.show()
```



Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Promotion

```
# find the restaurants with a rating count more than 50 and the average rating greater than 4
# use df_ratings dataframe to add a new column for count of ratings per restaurant and the average rating

df_ratings['rating_count'] = (
    df_ratings.groupby('restaurant_name')['rating']
    .transform('count')
)

df_ratings['avg_rating'] = (
    df_ratings.groupby('restaurant_name')['rating']
    .transform('mean')
)
# now add a new column in df_ratings to show true when the rating_count > 50 and the avg_rating > 4
df_ratings['promo'] = (df_ratings['rating_count'] > 50) & (df_ratings['avg_rating'] > 4)
```

```
# create a new dataFrame for unique restaurant_name where the promo == True
df_promo = df_ratings[df_ratings['promo'] == True]['restaurant_name'].unique()
print(df_promo)
```

```
['Blue Ribbon Fried Chicken' 'Shake Shack' 'Blue Ribbon Sushi'
 'The Meatball Shop']
```

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Revenue

```
# define a function to calculate the revenue of each order and add to a column named revenue,
def revenue(cost):
    if cost > 20:
        return cost * 0.25
    elif cost > 5:
        return cost * 0.15
    else:
        return 0
df['revenue'] = df['cost_of_the_order'].apply(revenue)
print('Total_Revenue = ', df['revenue'].sum())
# print the maximum, minimum, and mean revenue
print('Maximum = ', df['revenue'].max())
print('Minimum = ', df['revenue'].min())
print('Mean = ', df['revenue'].mean())

# print the total revenue for each cuisine and order by revenue from the highest to lowest
df_revenue_cuisine = df.groupby('cuisine_type')['revenue'].sum()
df_revenue_cuisine.sort_values(ascending=False)
```

```
Total_Revenue =  6166.303
Maximum =  8.8525
Minimum =  0.0
Mean =  3.2488424657534245
```

cuisine_type	revenue
American	1878.2740
Japanese	1478.2445
Italian	966.8845
Chinese	690.0405
Mexican	256.6930
Indian	246.9940
Middle Eastern	193.5535
Mediterranean	136.3280
Thai	78.0525
French	76.1925
Southern	68.7535
Spanish	47.2365
Korean	33.3030
Vietnamese	15.7530

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Total Time, Delivery Time Day of Week

```
# total_time was calculated above and added to df as a column named "total_time"
(df['total_time'] > 60).mean() * 100
```

```
np.float64(10.537407797681771)
```

Observations: * 10.5% of the orders takes more than 60 minutes to get delivered from the time the order is placed.

```
# calculate the mean delivery time on the weekdays and the weekends
df.groupby('day_of_the_week')['delivery_time'].mean()
```

day_of_the_week	delivery_time
Weekday	28.340037
Weekend	22.470022

dtype: float64

Observations: * The average (mean) delivery time is 28.34 minutes on weekdays and 22.47 minutes on weekends

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Cuisines in order of Ratings

```
# from df_ratings determine which cuisines had the highest ratings
df_ratings.groupby('cuisine_type')['rating'].mean().sort_values(ascending=False)
```

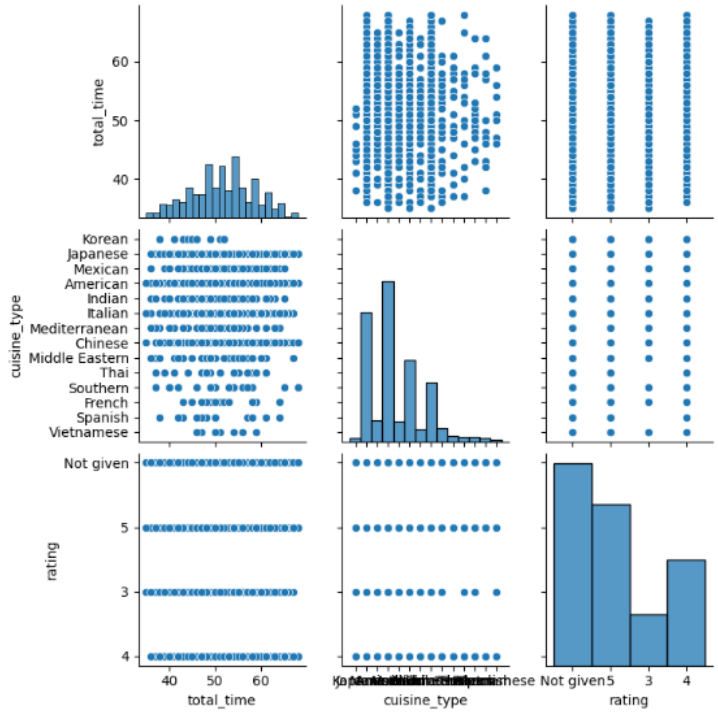
cuisine_type	rating
Spanish	4.833333
Thai	4.666667
Indian	4.540000
Mexican	4.416667
Japanese	4.373626
Italian	4.360465
Chinese	4.338346
Southern	4.307692
French	4.300000
American	4.298913
Middle Eastern	4.235294
Mediterranean	4.218750
Korean	4.111111
Vietnamese	4.000000

Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

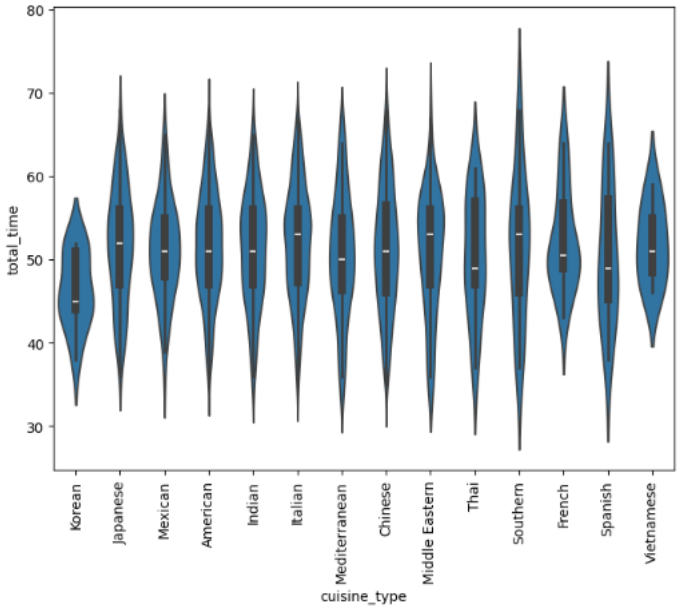
Total Time, Cuisine Type, Rating

```
# create a pair plot for total time, cuisine_type, and rating
plt.figure(figsize=(8, 6))
sns.pairplot(data=df, vars=['total_time', 'cuisine_type', 'rating'])
plt.show()
```

```
<Figure size 800x600 with 0 Axes>
```




```
# create a violin plot for total time, cuisine_type, and rating
plt.figure(figsize=(8, 6))
sns.violinplot(data=df, x='cuisine_type', y='total_time')
plt.xticks(rotation=90)
plt.show()
```



Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Heat Map Preparation

```
df.select_dtypes(exclude=['object', 'category']).columns
```

```
Index(['order_id', 'customer_id', 'cost_of_the_order', 'food_preparation_time',
       'delivery_time', 'total_time', 'revenue'],
      dtype='object')
```

```
df.head()
```

	order_id	customer_id	restaurant_name	cuisine_type	cost_of_the_order	day_of_the_week	rating	food_preparation_time	delivery_time	total_time	revenue
0	1477147	337525	Hangawi	Korean	30.75	Weekend	Not given	25	20	45	7.6875
1	1477685	358141	Blue Ribbon Sushi Izakaya	Japanese	12.08	Weekend	Not given	25	23	48	1.8120
2	1477070	66393	Cafe Habana	Mexican	12.23	Weekday	5	23	28	51	1.8345
3	1477334	106968	Blue Ribbon Fried Chicken	American	29.20	Weekend	3	25	15	40	7.3000
4	1478249	76942	Dirty Bird to Go	American	11.59	Weekday	4	25	24	49	1.7385

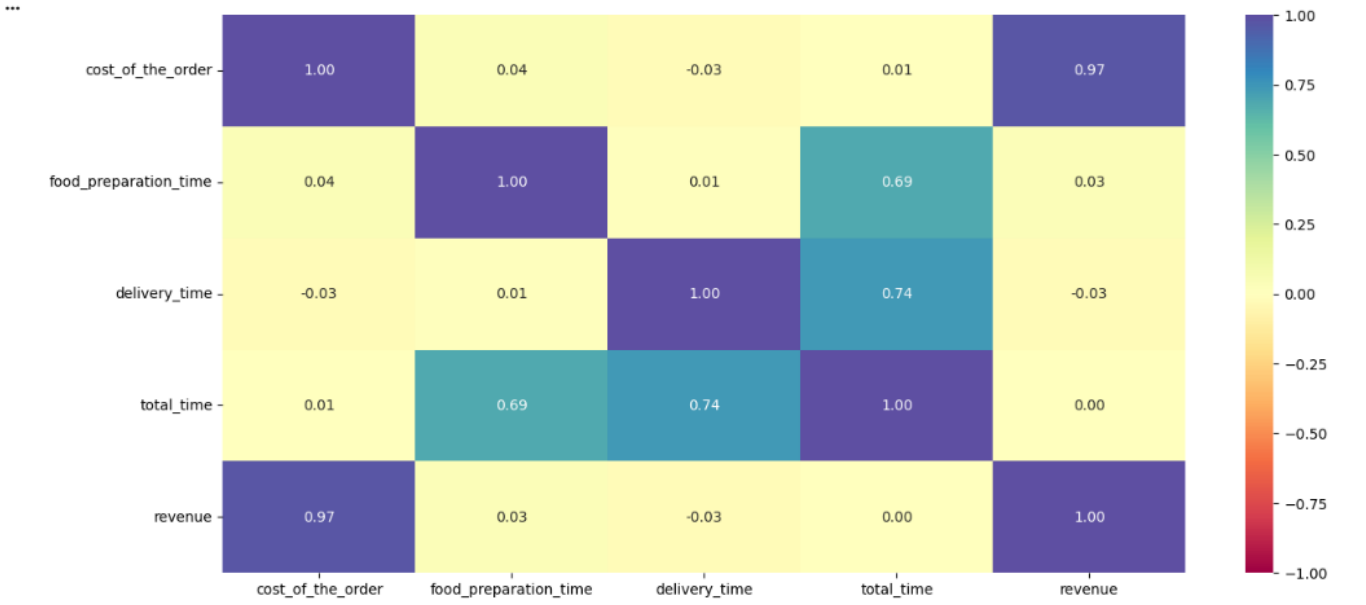
Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Heat Map

```
# create a heatmap of df_ratings to see where the highest correlation is for the numeric values
num_var = df[['cost_of_the_order', 'food_preparation_time',
       'delivery_time', 'total_time', 'revenue']].columns
corr = df[num_var].corr(numeric_only = True)

# plot the heatmap

plt.figure(figsize=(15, 7))
sns.heatmap(corr, annot=True, vmin=-1, vmax=1, fmt=".2f", cmap="Spectral")
plt.show()
```



Proprietary content. © Great Learning. All Rights Reserved. Unauthorized use or distribution prohibited.

Great Learning

Happy Learning !

