August 1, 2025

FoodHub Data Analysis

Python Foundations

Rex King

Contents

1. Executive Summary
2. Problem Overview and Solution Approach
3. Data Overview
4. Univariate Analysis
5. Multivariate Analysis

Executive Summary

Conclusions:

- Japanese, American, Italian, and Chinese are the most popular types of cuisines
- Certain restaurants are much more popular than others
  - Shack Shack, The Meatball Shop, and Blue Ribbon sushi are the top 3 most popular restaurants
  - These restaurants also receive a significant amount of ratings compared to other restaurants in the data set
  - The Meatball Shop has the second greatest number of orders, but has the highest average rating
- The costs of certain types of cuisines range very diversey
  - Vietnamese is the cheapest
  - French is the most expensive
- There is a much higher number of orders on the weekends compared to weekdays
- About 38.7% of restaurants have not been rated

Executive Summary (cont.)

Recommendations:

- Preparation and delivery practices should be optimized
  - The lower the time from when a order is placed to when an order is delivered can result in better ratings from customers
- Cuisines that are more expensive should emphasis quality so that the customer feels that the price of an order is justified
- Cuisines that are less expensive could sell at a higher price if they were to serve larger quantities of food in each order
- On weekends, there should be a greater number of delivery drivers to meet the higher demand of orders
  - On the other hand, there should be less delivery drivers on weekdays since the demand is lower
- There should be some form of incentive for customers to rate each restaurant
  - This can help restaurants improve their quality and service
  - Forms of incentives could be discounts or coupons
- Restaurants that sell more expensive cuisines can introduce a loyalty program to help retain customers who may not be able to pay an expensive price everytime

Problem Overview and Solution Approach

- Problem:
  - Restaurants seek to optimize their quality and service but understanding their performance and the behavior of their customers. These restaurants aim to ensure efficient delivery, reasonable and justified pricing, high quality, and high satisfaction in order to compete with other restaurants.

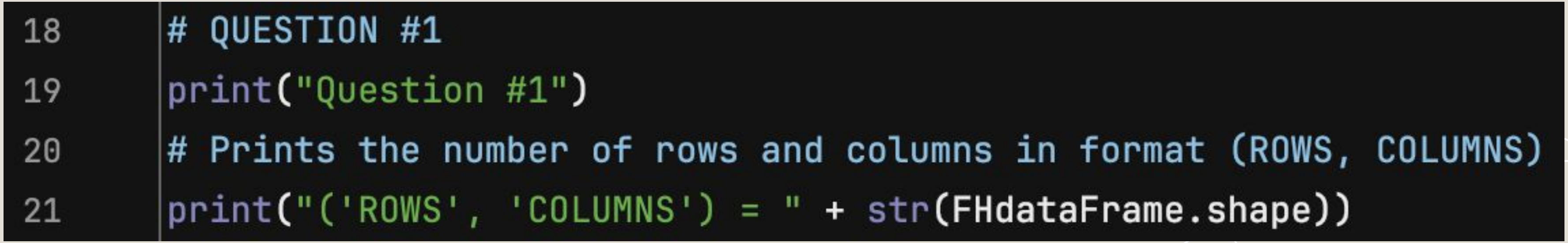
- Solution Approach
  - These restaurants need to understand the data obtained from them. In order to do so, categories crucial to their success must be studied and compared. Data like types of cuisine and their cost, analyzing the total time it takes for a customer to receive their order, ratings they receive, and the outliers of these categories will play a crucial role in allowing the restaurants to understand the type of service they are providing. Once this data has been analyzed, conclusions and recommendations will be able to be drawn.

Data Overview

- There are 1,989 total columns in the data sheet
- There are 9 total columns in the data sheet
- The 9 columns consist of:
  - order_id
  - customer_id
  - restaurant_name
  - cuisine_type
  - cost_of_the_order
  - day_of_the_week
  - rating
  - food_preparation_time
  - delivery_time

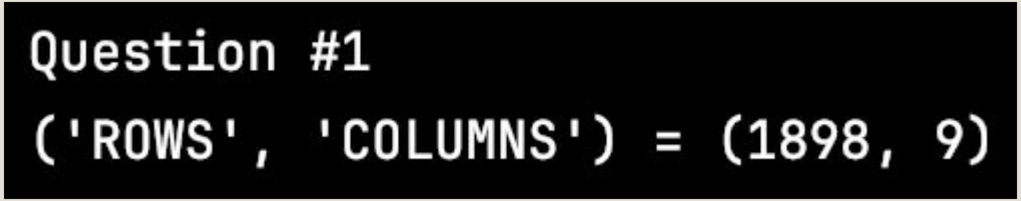
Question #1

Input:



```python
# QUESTION #1
print("Question #1")
# Prints the number of rows and columns in format (ROWS, COLUMNS)
print("('ROWS', 'COLUMNS') = " + str(FHdataFrame.shape))
```

Output:

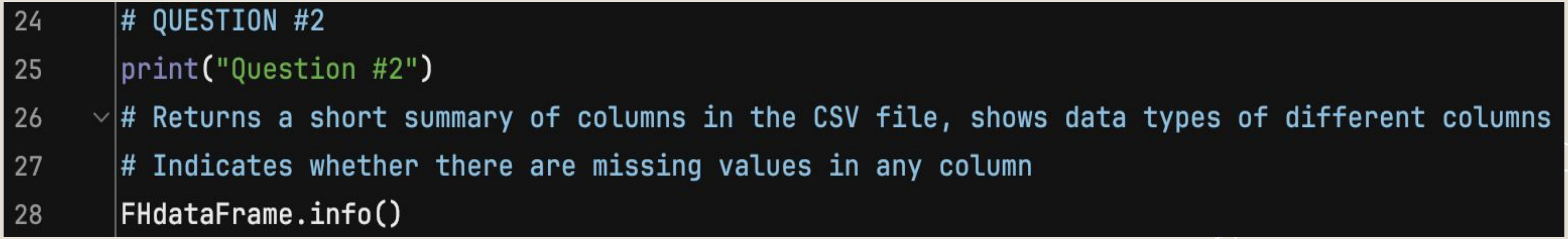


```
Question #1
('ROWS', 'COLUMNS') = (1898, 9)
```

- Observations:
  - This data set is made up of 1,989 rows and 9 columns

Question #2

Input:

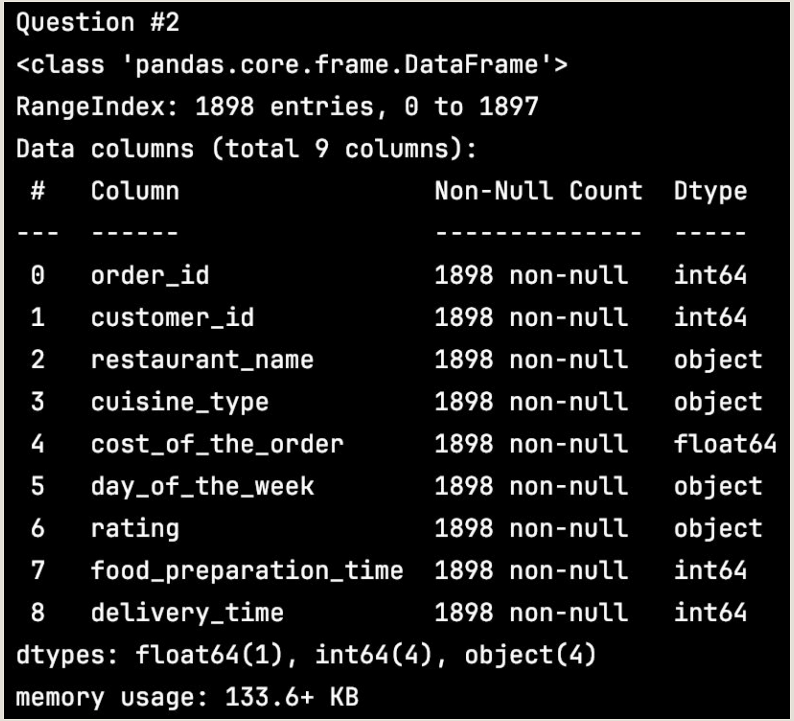


```
# QUESTION #2
print("Question #2")
# Returns a short summary of columns in the CSV file, shows data types of different columns
# Indicates whether there are missing values in any column
FHdataFrame.info()
```

Output on next slide

Question #2 (cont.)

Output:

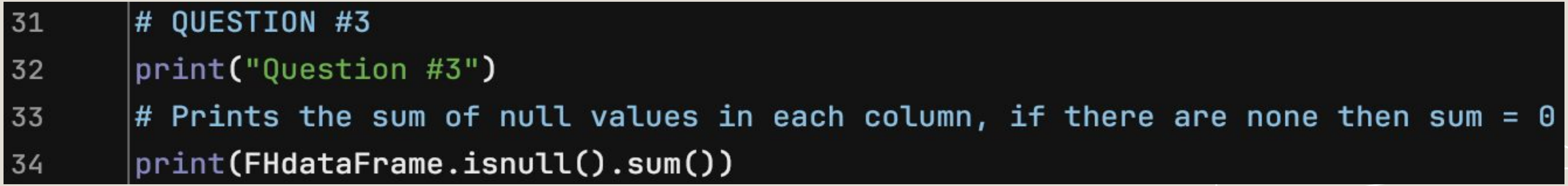


```
Question #2
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1898 entries, 0 to 1897
Data columns (total 9 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   order_id               1898 non-null   int64
 1   customer_id            1898 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:
  - All columns have 1,989 non-null values. This means that there are no empty values in each of the 9 columns. There are 3 different types of data in the data sheet. The different types are float64, int64, and object. The data uses 133.6kb of memory.

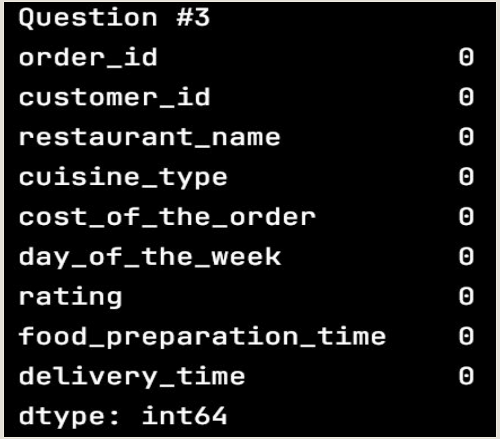
Question #3

Input:



```python
# QUESTION #3
print("Question #3")
# Prints the sum of null values in each column, if there are none then sum = 0
print(FHdataFrame.isnull().sum())
```

Output:

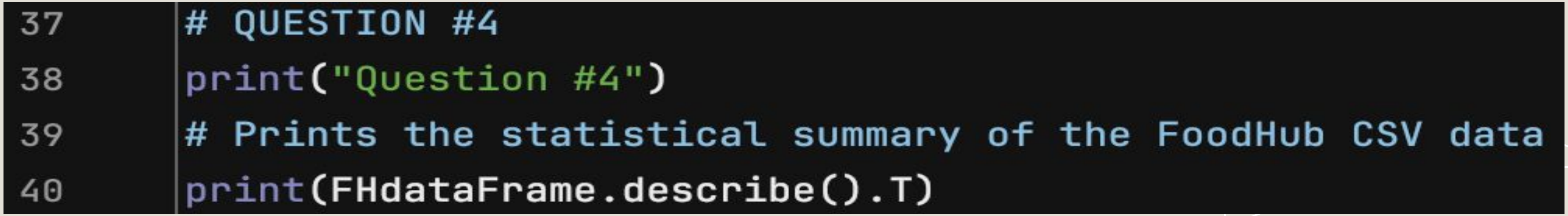


```
Question #3
order_id                  0
customer_id               0
restaurant_name           0
cuisine_type              0
cost_of_the_order         0
day_of_the_week           0
rating                    0
food_preparation_time     0
delivery_time             0
dtype: int64
```

- Observations:
  - There are no missing values. This is shown by the zeroes in the column on the right side.

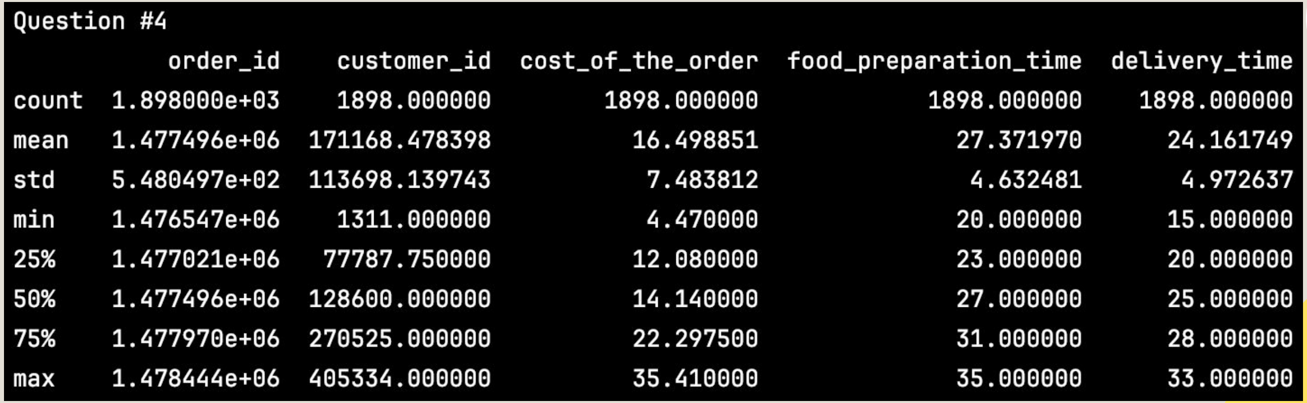
Question #4

Input:



```
# QUESTION #4
print("Question #4")
# Prints the statistical summary of the FoodHub CSV data
print(FHdataFrame.describe().T)
```

Output:



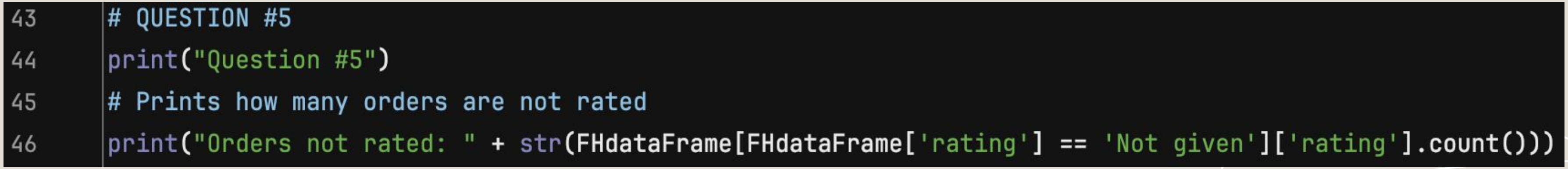
```
Question #4
          order_id    customer_id  cost_of_the_order  food_preparation_time  delivery_time
count  1.898000e+03    1898.000000        1898.000000            1898.000000    1898.000000
mean   1.477496e+06  171168.478398          16.498851              27.371970      24.161749
std    5.480497e+02  113698.139743           7.483812               4.632481       4.972637
min    1.476547e+06    1311.000000           4.470000              20.000000      15.000000
25%    1.477021e+06   77787.750000          12.080000              23.000000      20.000000
50%    1.477496e+06  128600.000000          14.140000              27.000000      25.000000
75%    1.477970e+06  270525.000000          22.297500              31.000000      28.000000
max    1.478444e+06  405334.000000          35.410000              35.000000      33.000000
```

Question #4 (cont.)

- Observations:
  - Customer_id and order_id are only used as identifiers. The average cost of an order is $16.50, but the cost of an order ranges from $4.47 to $35.41 with a standard deviation of 7.48. The average preparation time is around 27 minutes, but the time ranges from 20-35 minutes with a standard deviation of 4.63. The average delivery time is around 24 minutes, but it ranges from 15-33 minutes with a standard deviation of 4.97.

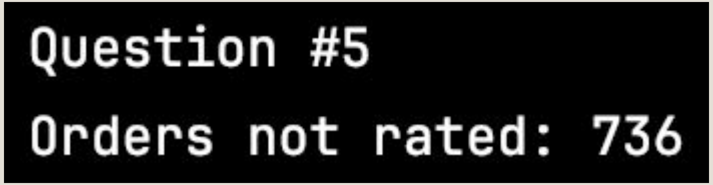
Question #5

Input:



```python
# QUESTION #5
print("Question #5")
# Prints how many orders are not rated
print("Orders not rated: " + str(FHdataFrame[FHdataFrame['rating'] == 'Not given']['rating'].count()))
```

Output:



```
Question #5
Orders not rated: 736
```

- Observations:
  - There are 736 orders that are not rated in the data set

Univariate Analysis

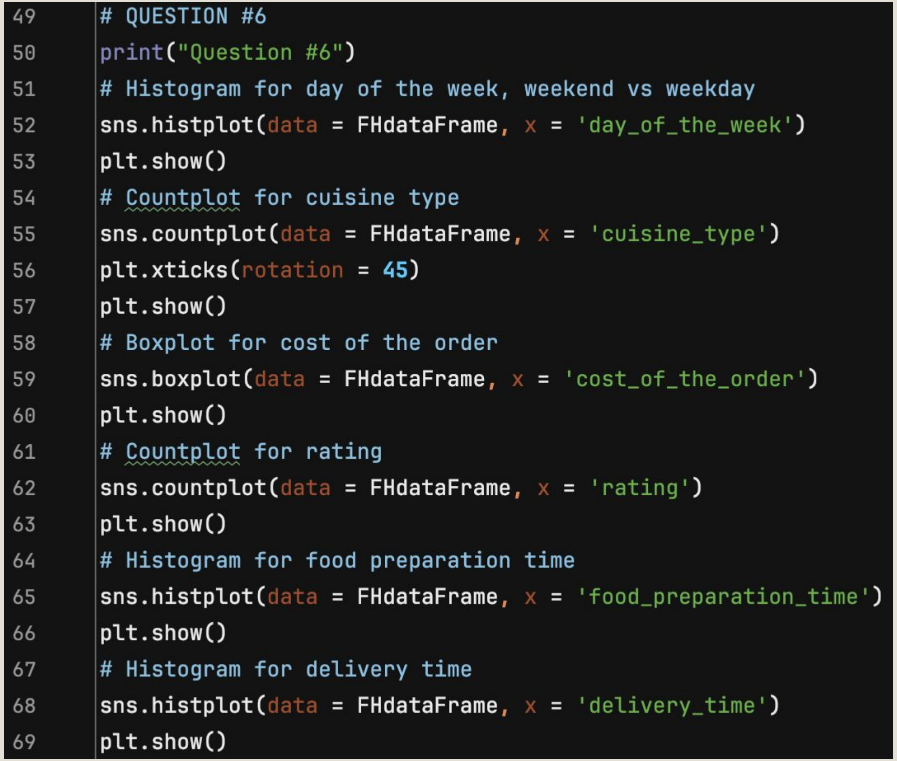
- Univariate analysis is essential because it allows all columns in the data set to be understood statistically. It can identify popular, as well as unpopular practices of the restaurants in the data set. This can help restaurants determine with they are doing correct, and what they need to improve upon.

Multivariate Analysis

- Multivariate analysis is important because it allows multiple categories to be studied at the same time, as well as the relationships between them. It can discover correlations of multiple categories, as well as identify patterns and trends between categories. This allows the restaurants to understand combined effects of different practices and results.

Question #6

Input:

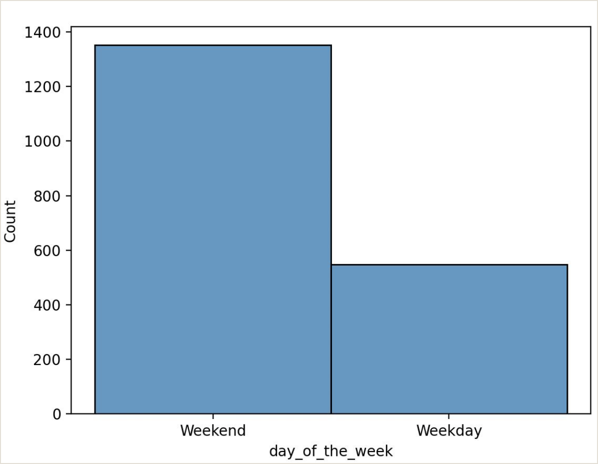


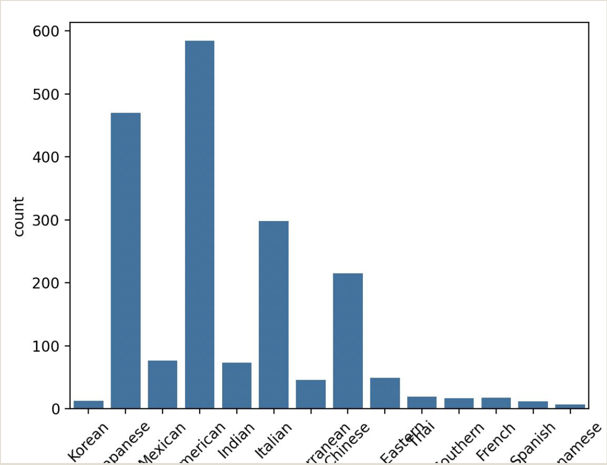
```
# QUESTION #6
print("Question #6")
# Histogram for day of the week, weekend vs weekday
sns.histplot(data = FHdataFrame, x = 'day_of_the_week')
plt.show()
# Countplot for cuisine type
sns.countplot(data = FHdataFrame, x = 'cuisine_type')
plt.xticks(rotation = 45)
plt.show()
# Boxplot for cost of the order
sns.boxplot(data = FHdataFrame, x = 'cost_of_the_order')
plt.show()
# Countplot for rating
sns.countplot(data = FHdataFrame, x = 'rating')
plt.show()
# Histogram for food preparation time
sns.histplot(data = FHdataFrame, x = 'food_preparation_time')
plt.show()
# Histogram for delivery time
sns.histplot(data = FHdataFrame, x = 'delivery_time')
plt.show()
```

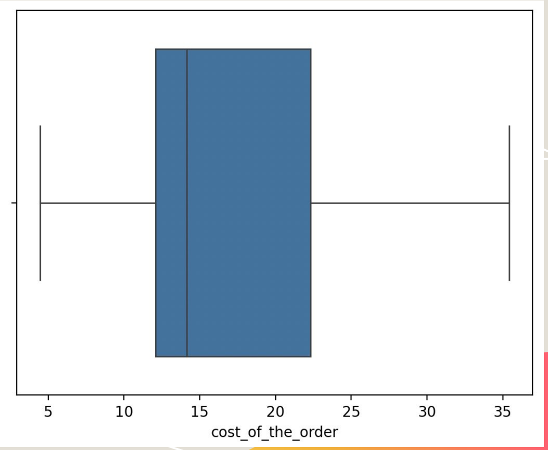
Output on following slides

Question #6 (cont.)

Output:



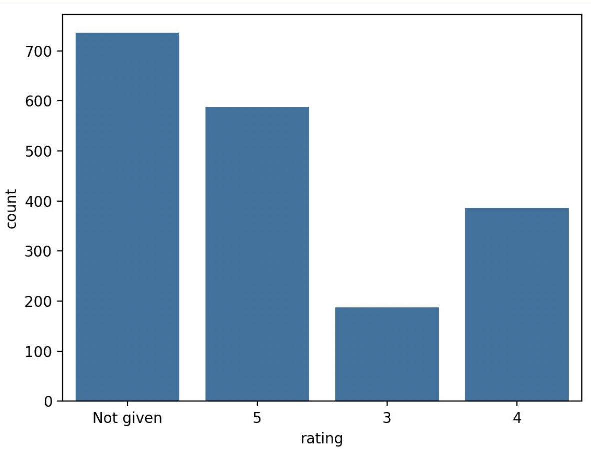


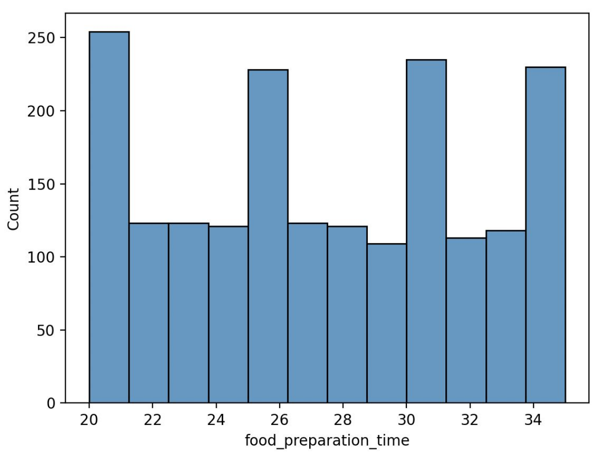


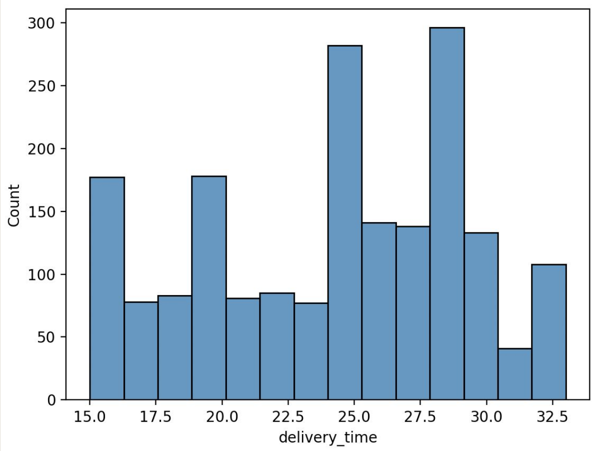
- The number of orders placed on the weekends is much higher than on weekdays.
- Japanese, American, Italian, and Chinese are the most popular types of cuisine
- The graph is right skewed which means that average cost is on the lower end of the range of costs.

Question #6 (cont.)

Output:



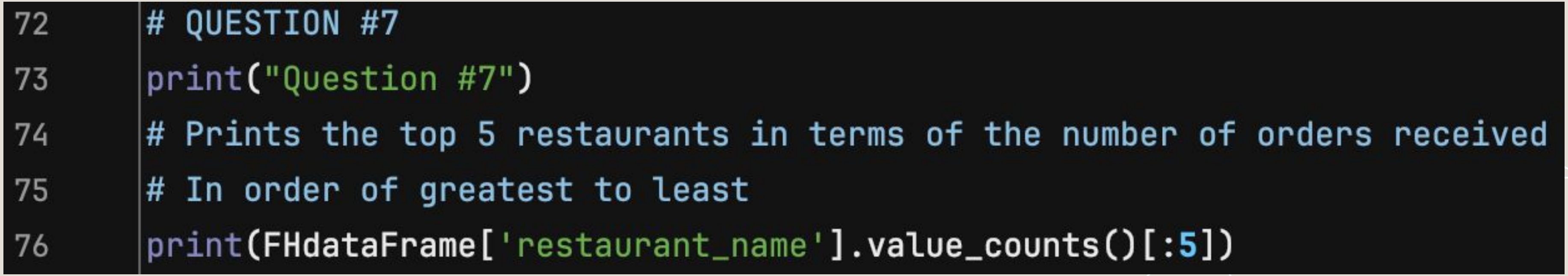




- Around 700 orders were not rated. Around 550 orders were given a rating of 5. Around 200 gave a rating of 3. Around 400 gave a rating of 4.
- The preparation time count is much higher around 20, 26, 30, and 34 minutes. Counts around 100 are nearly equal.
- The count for delivery time is much higher around 25 and 29 minutes.

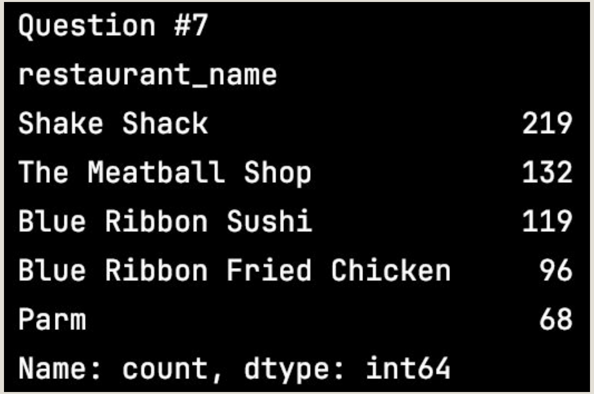
Question #7

Input:



```
# QUESTION #7
print("Question #7")
# Prints the top 5 restaurants in terms of the number of orders received
# In order of greatest to least
print(FHdataFrame['restaurant_name'].value_counts()[:5])
```

Output:

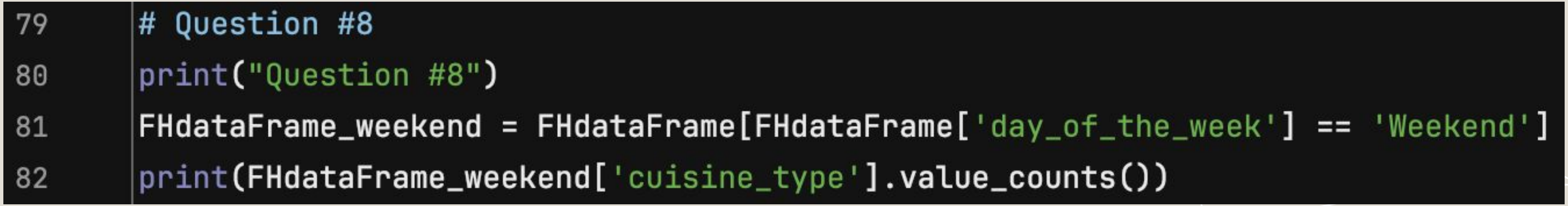


```
Question #7
restaurant_name
Shake Shack                  219
The Meatball Shop            132
Blue Ribbon Sushi            119
Blue Ribbon Fried Chicken     96
Parm                          68
Name: count, dtype: int64
```

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

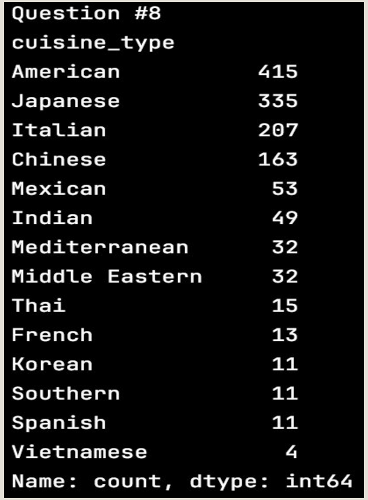
Question #8

Input:



```python
# Question #8
print("Question #8")
FHdataFrame_weekend = FHdataFrame[FHdataFrame['day_of_the_week'] == 'Weekend']
print(FHdataFrame_weekend['cuisine_type'].value_counts())
```

Output:

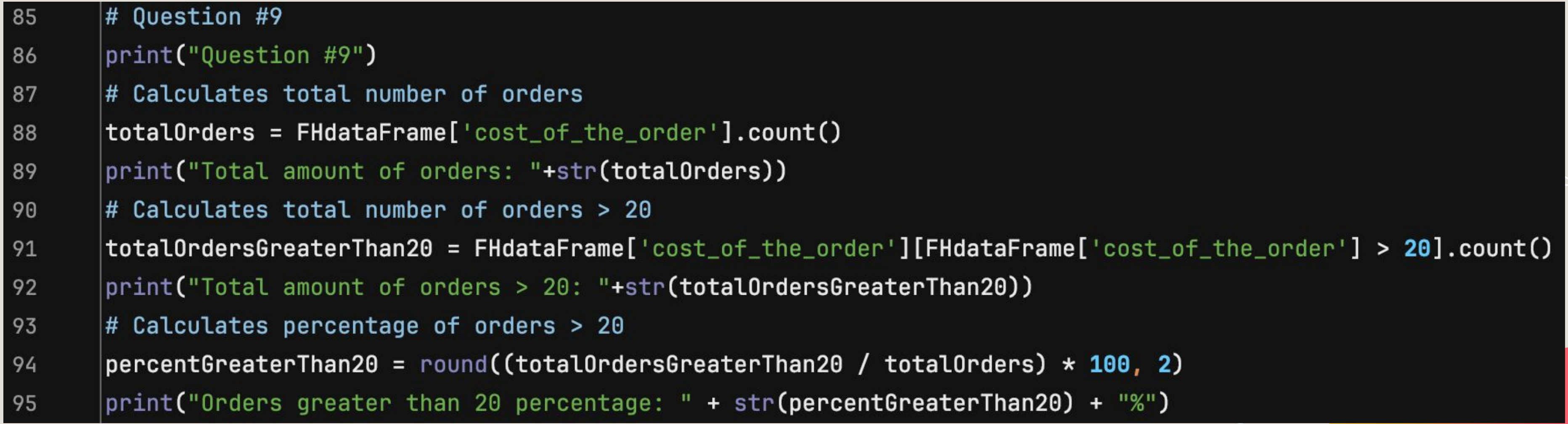


```
Question #8
cuisine_type
American          415
Japanese          335
Italian           207
Chinese           163
Mexican            53
Indian             49
Mediterranean      32
Middle Eastern     32
Thai               15
French             13
Korean             11
Southern           11
Spanish            11
Vietnamese          4
Name: count, dtype: int64
```

- Observations:
  - The most popular type of cuisine on the weekend is American. The least popular top of cuisine on the weekend is Vietnamese. There are only 4 types of cuisine with a count greater than 100.

Question #9

Input:

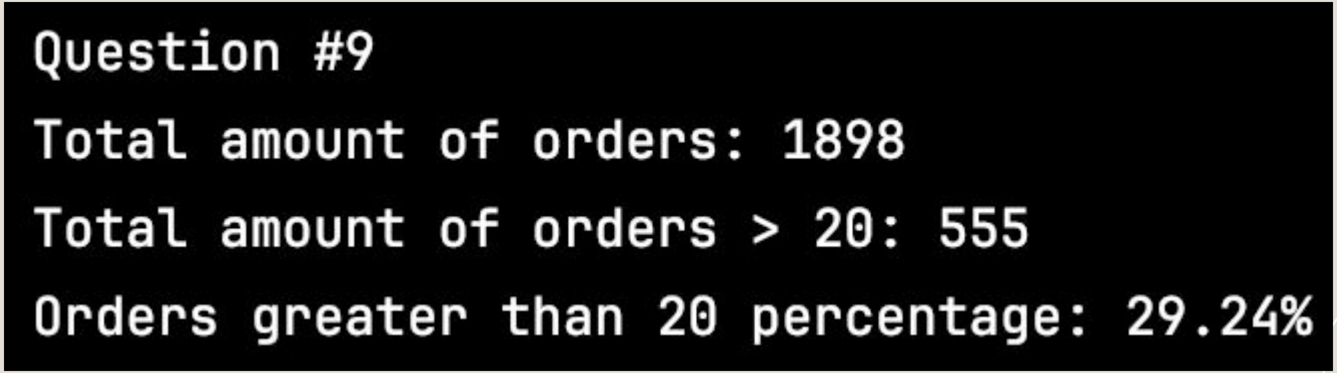


```python
# Question #9
print("Question #9")
# Calculates total number of orders
totalOrders = FHdataFrame['cost_of_the_order'].count()
print("Total amount of orders: "+str(totalOrders))
# Calculates total number of orders > 20
totalOrdersGreaterThan20 = FHdataFrame['cost_of_the_order'][FHdataFrame['cost_of_the_order'] > 20].count()
print("Total amount of orders > 20: "+str(totalOrdersGreaterThan20))
# Calculates percentage of orders > 20
percentGreaterThan20 = round((totalOrdersGreaterThan20 / totalOrders) * 100, 2)
print("Orders greater than 20 percentage: " + str(percentGreaterThan20) + "%")
```

Output on next slide

Question #9 (cont.)

Output:

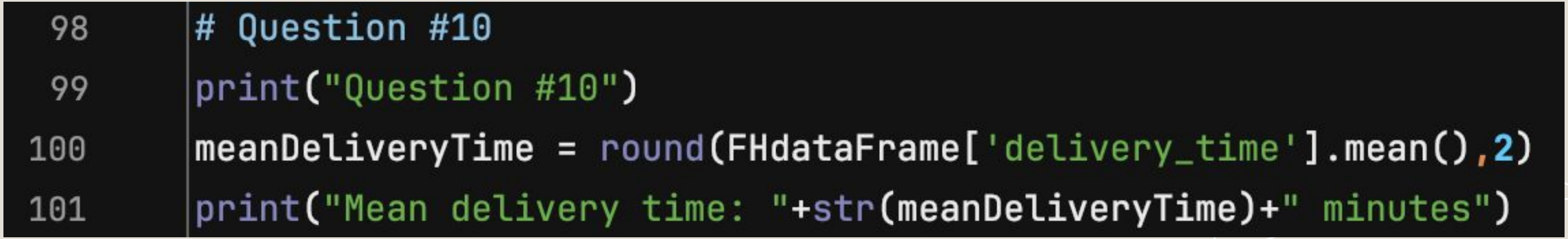


```
Question #9
Total amount of orders: 1898
Total amount of orders > 20: 555
Orders greater than 20 percentage: 29.24%
```

- Observations:
  - There are 555 total orders that cost above $20. This amounts to 29.24% of orders in the data.

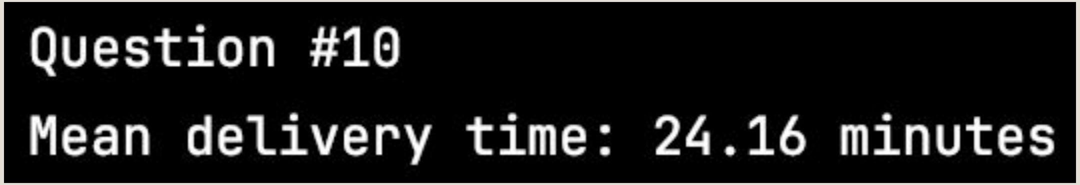
Question #10

Input:



```
# Question #10
print("Question #10")
meanDeliveryTime = round(FHdataFrame['delivery_time'].mean(),2)
print("Mean delivery time: "+str(meanDeliveryTime)+" minutes")
```

Output:

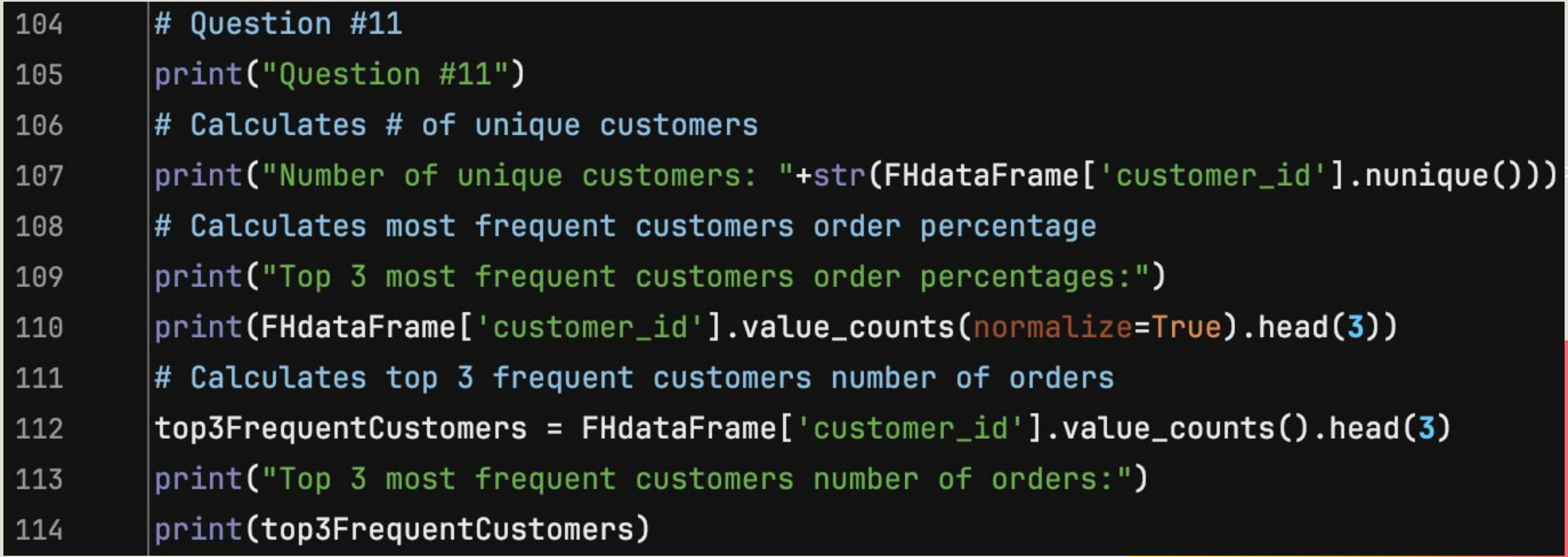


```
Question #10
Mean delivery time: 24.16 minutes
```

- Observations:
  - The mean delivery time is around 24.16 minutes.

Question #11

Input:

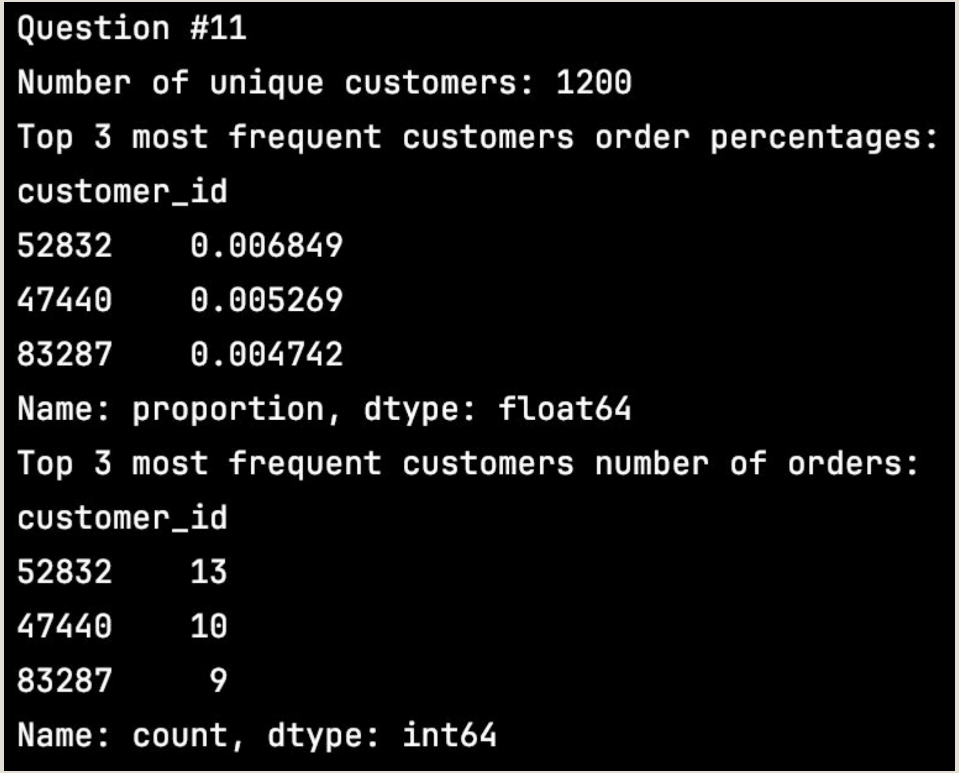


```python
# Question #11
print("Question #11")
# Calculates # of unique customers
print("Number of unique customers: "+str(FHdataFrame['customer_id'].nunique()))
# Calculates most frequent customers order percentage
print("Top 3 most frequent customers order percentages:")
print(FHdataFrame['customer_id'].value_counts(normalize=True).head(3))
# Calculates top 3 frequent customers number of orders
top3FrequentCustomers = FHdataFrame['customer_id'].value_counts().head(3)
print("Top 3 most frequent customers number of orders:")
print(top3FrequentCustomers)
```

Output on next slide

Question #11 (cont.)

Output:

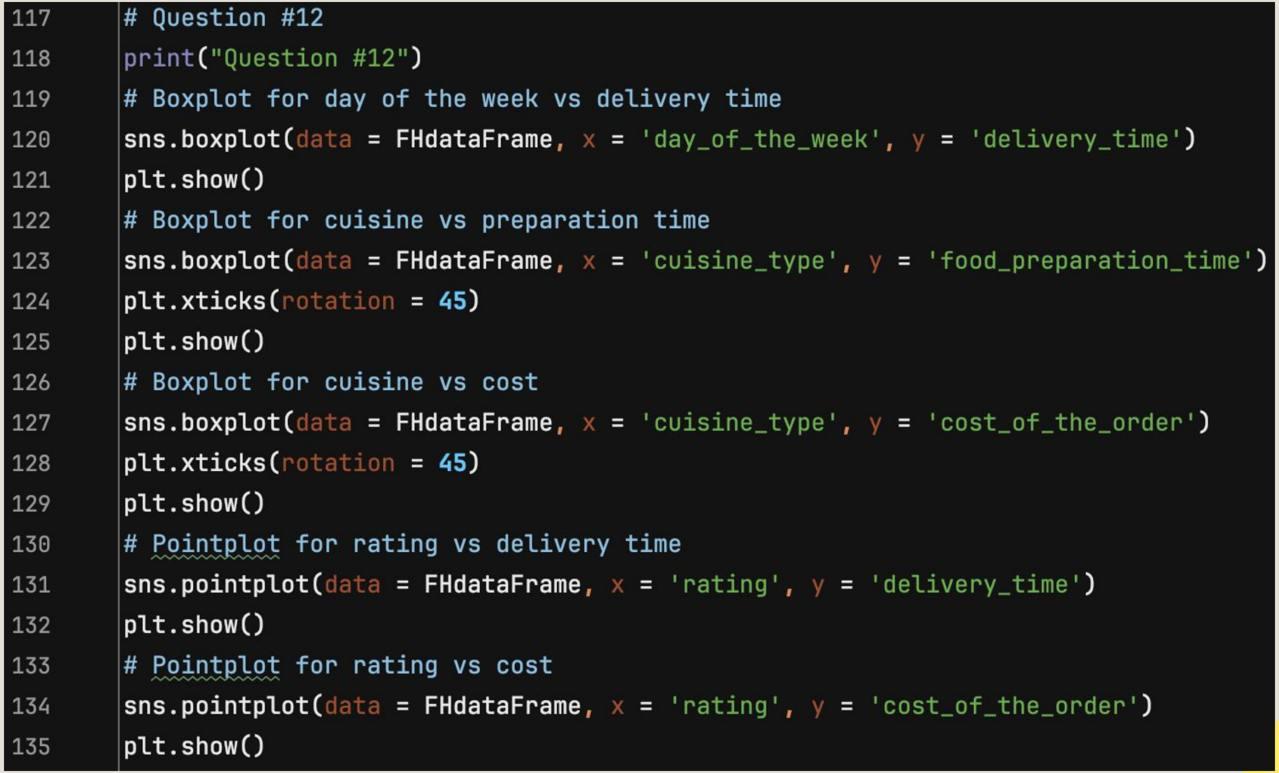


```
Question #11
Number of unique customers: 1200
Top 3 most frequent customers order percentages:
customer_id
52832    0.006849
47440    0.005269
83287    0.004742
Name: proportion, dtype: float64
Top 3 most frequent customers number of orders:
customer_id
52832    13
47440    10
83287     9
Name: count, dtype: int64
```

- Observations:
  - The top 3 most frequent customer IDs are 52832, 47440, 83287. Customer 52832 has placed 13 orders. Customer 47440 has placed 10 orders. Customer 83287 has placed 9 orders.

Question #12

Input:

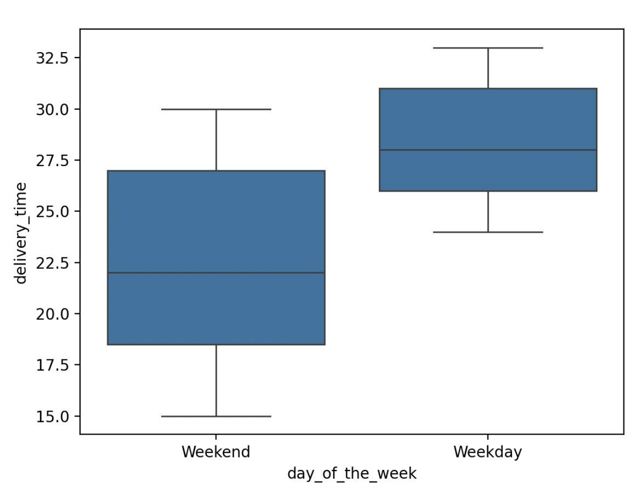


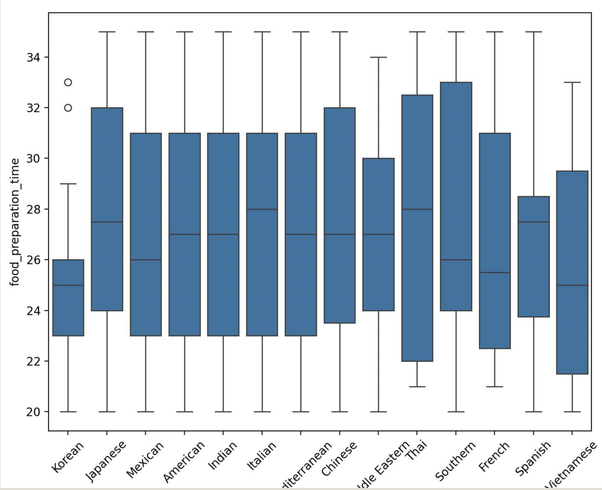
```
# Question #12
print("Question #12")
# Boxplot for day of the week vs delivery time
sns.boxplot(data = FHdataFrame, x = 'day_of_the_week', y = 'delivery_time')
plt.show()
# Boxplot for cuisine vs preparation time
sns.boxplot(data = FHdataFrame, x = 'cuisine_type', y = 'food_preparation_time')
plt.xticks(rotation = 45)
plt.show()
# Boxplot for cuisine vs cost
sns.boxplot(data = FHdataFrame, x = 'cuisine_type', y = 'cost_of_the_order')
plt.xticks(rotation = 45)
plt.show()
# Pointplot for rating vs delivery time
sns.pointplot(data = FHdataFrame, x = 'rating', y = 'delivery_time')
plt.show()
# Pointplot for rating vs cost
sns.pointplot(data = FHdataFrame, x = 'rating', y = 'cost_of_the_order')
plt.show()
```

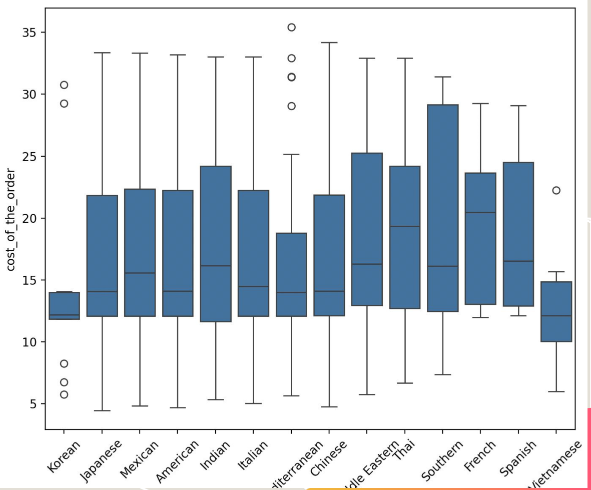
Output on following slides

Question #12 (cont.)

Output:





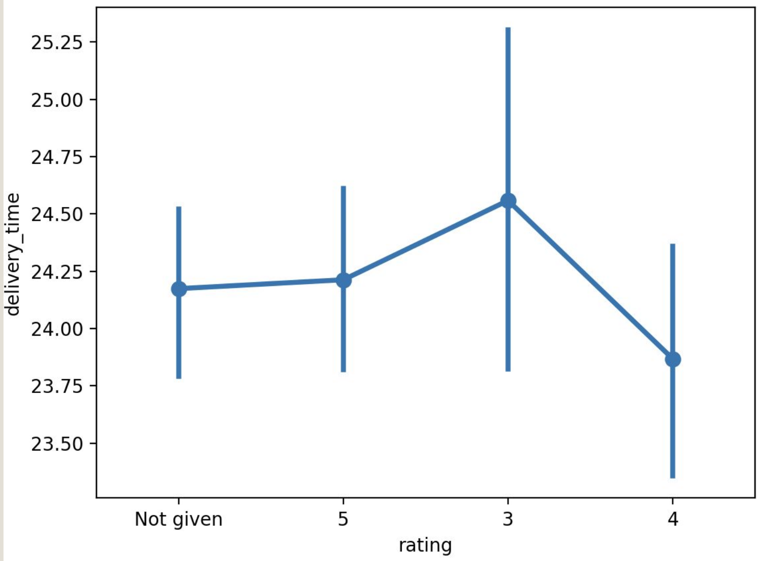


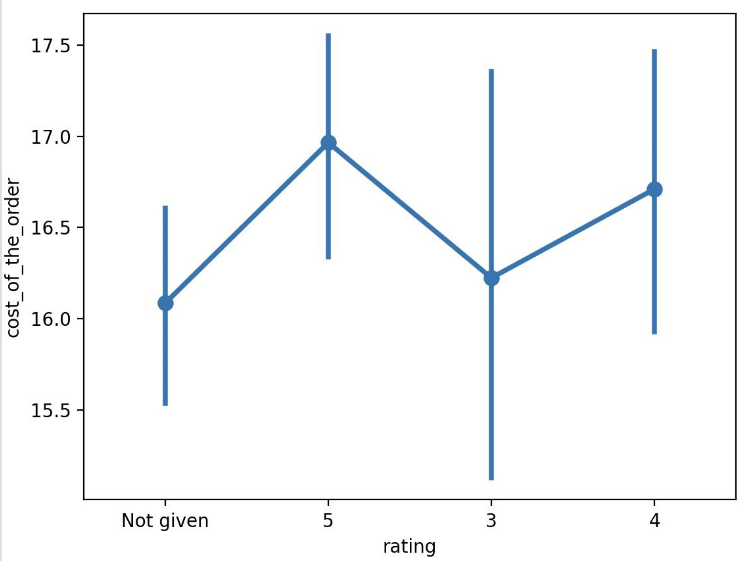
- Delivery time on the weekends is much less than that on weekdays.

- Food preparation time is mostly consistent with outliers in Korean and Spanish.

- Korean, Mediterranean, and Vietnamese cuisines cost much less than the others. The cost of the rest of the cuisines are similar.

Question #12 (cont.)



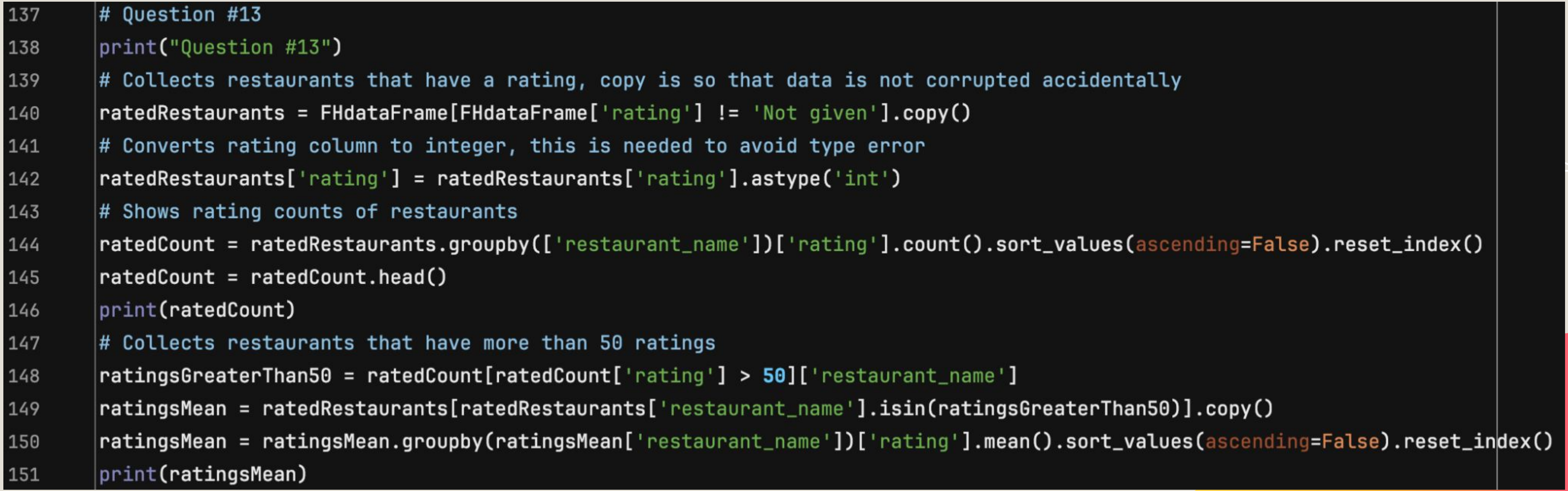


- The lowest delivery times have a high rating. There is not much difference between no rating and a rating of 5.

- The most expensive orders have the highest rating, while the cheapest orders have not been rated.

Question #13

Input:

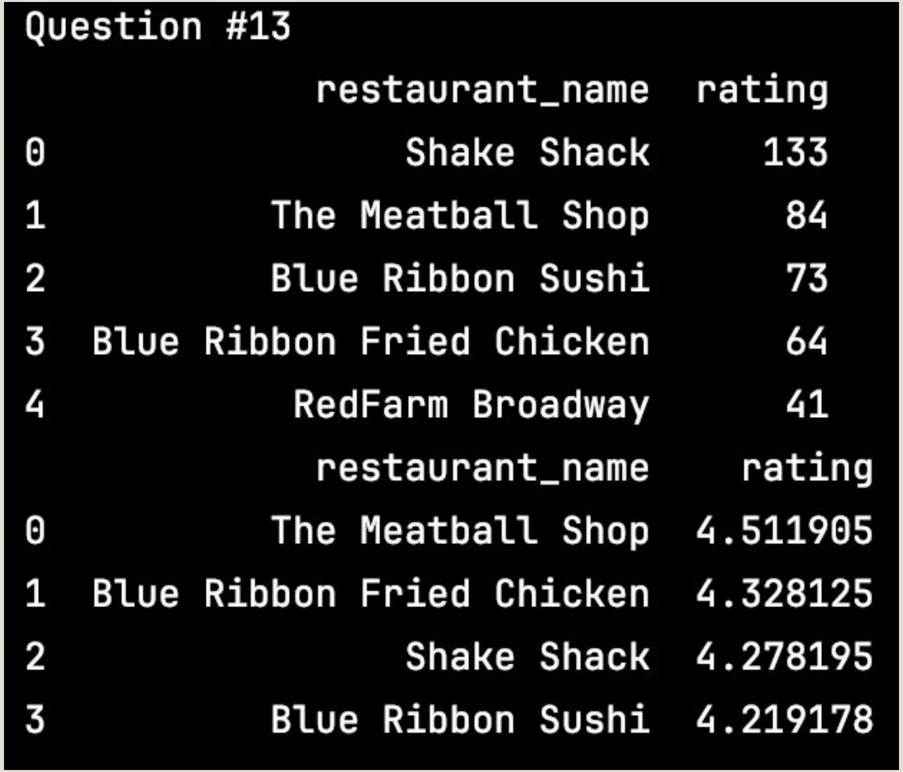


```
# Question #13
print("Question #13")
# Collects restaurants that have a rating, copy is so that data is not corrupted accidentally
ratedRestaurants = FHdataFrame[FHdataFrame['rating'] != 'Not given'].copy()
# Converts rating column to integer, this is needed to avoid type error
ratedRestaurants['rating'] = ratedRestaurants['rating'].astype('int')
# Shows rating counts of restaurants
ratedCount = ratedRestaurants.groupby(['restaurant_name'])['rating'].count().sort_values(ascending=False).reset_index()
ratedCount = ratedCount.head()
print(ratedCount)
# Collects restaurants that have more than 50 ratings
ratingsGreaterThan50 = ratedCount[ratedCount['rating'] > 50]['restaurant_name']
ratingsMean = ratedRestaurants[ratedRestaurants['restaurant_name'].isin(ratingsGreaterThan50)].copy()
ratingsMean = ratingsMean.groupby(ratingsMean['restaurant_name'])['rating'].mean().sort_values(ascending=False).reset_index()
print(ratingsMean)
```

Output on next slide

Question #13

Output:



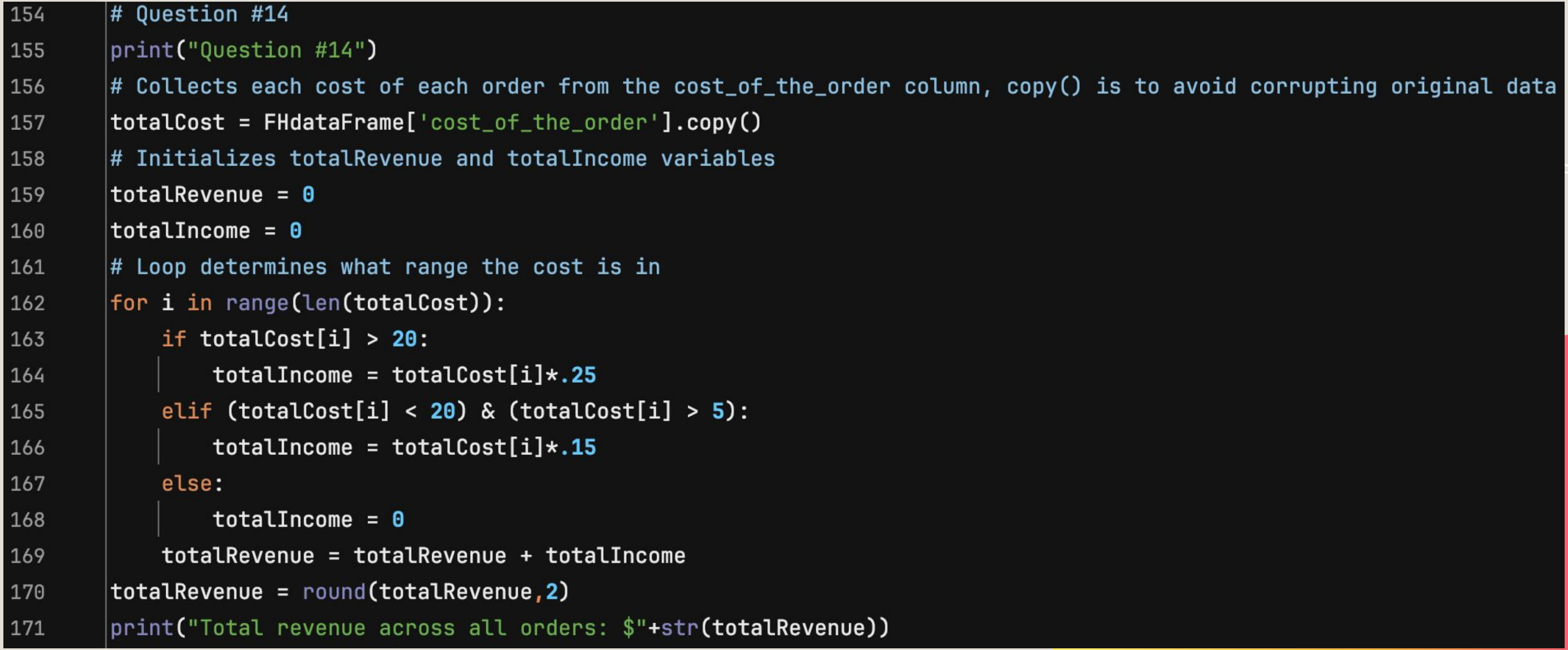
```
Question #13
             restaurant_name  rating
0                Shake Shack     133
1          The Meatball Shop      84
2          Blue Ribbon Sushi      73
3  Blue Ribbon Fried Chicken      64
4           RedFarm Broadway      41
             restaurant_name    rating
0          The Meatball Shop  4.511905
1  Blue Ribbon Fried Chicken  4.328125
2                Shake Shack  4.278195
3          Blue Ribbon Sushi  4.219178
```

- Observations:
  - Shake Shack, The Meatball Shop, Blue Ribbon, and Blue Ribbon Fried Chicken meet the requirements for the promotion. Shack Shack has the most ratings while The Meatball Shop has the highest average rating.

Question #14

Input:

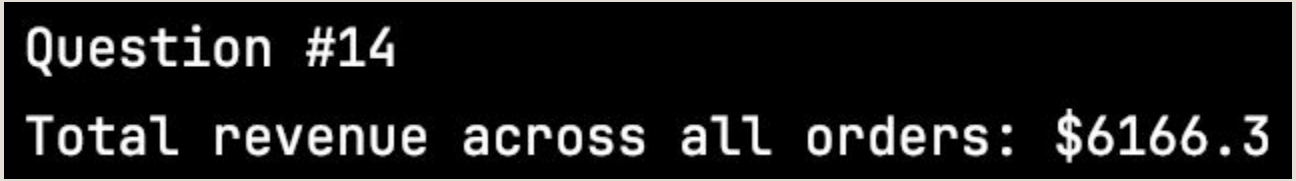
Output on next slide



```
# Question #14
print("Question #14")
# Collects each cost of each order from the cost_of_the_order column, copy() is to avoid corrupting original data
totalCost = FHdataFrame['cost_of_the_order'].copy()
# Initializes totalRevenue and totalIncome variables
totalRevenue = 0
totalIncome = 0
# Loop determines what range the cost is in
for i in range(len(totalCost)):
    if totalCost[i] > 20:
        totalIncome = totalCost[i]*.25
    elif (totalCost[i] < 20) & (totalCost[i] > 5):
        totalIncome = totalCost[i]*.15
    else:
        totalIncome = 0
    totalRevenue = totalRevenue + totalIncome
totalRevenue = round(totalRevenue,2)
print("Total revenue across all orders: $"+str(totalRevenue))
```

Question #14 (cont.)

Output:



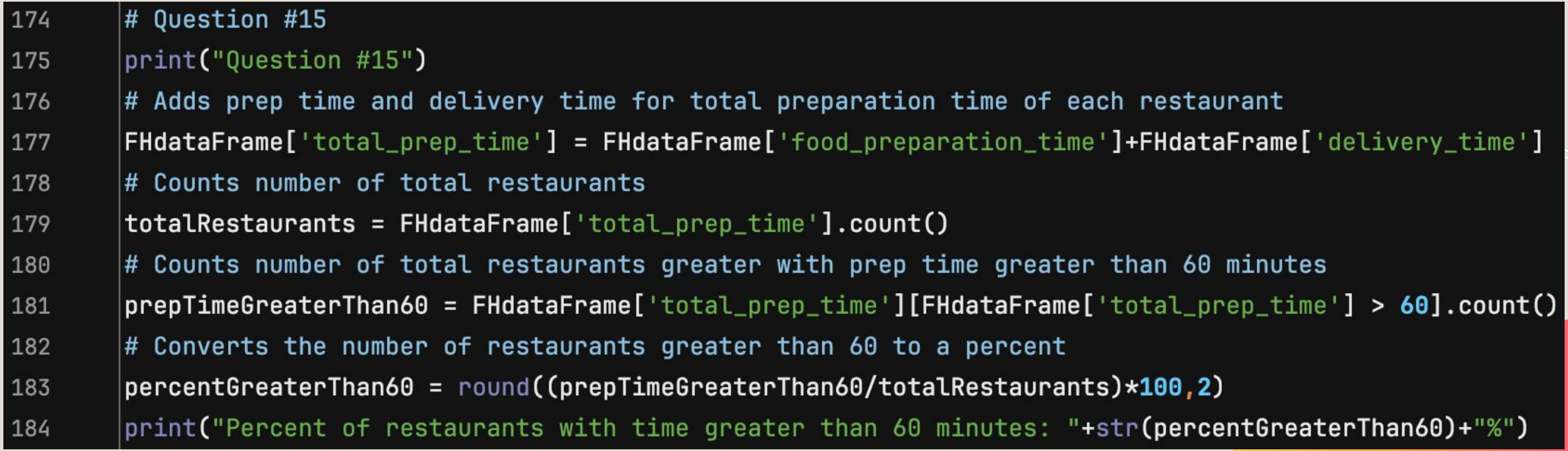
```
Question #14

Total revenue across all orders: $6166.3
```

- Observations:
  - The total revenue across all orders is the data set is $6166.30

Question #15

Input:

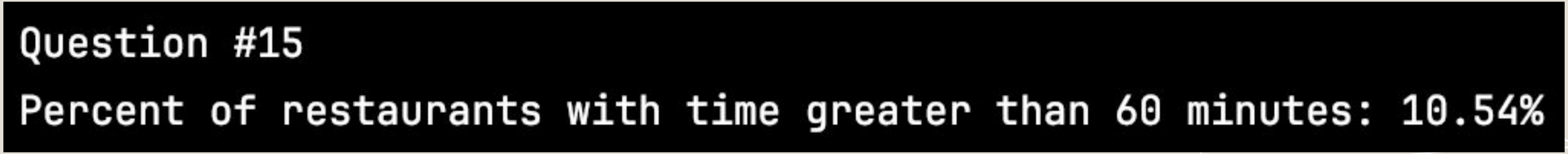


```python
# Question #15
print("Question #15")
# Adds prep time and delivery time for total preparation time of each restaurant
FHdataFrame['total_prep_time'] = FHdataFrame['food_preparation_time']+FHdataFrame['delivery_time']
# Counts number of total restaurants
totalRestaurants = FHdataFrame['total_prep_time'].count()
# Counts number of total restaurants greater with prep time greater than 60 minutes
prepTimeGreaterThan60 = FHdataFrame['total_prep_time'][FHdataFrame['total_prep_time'] > 60].count()
# Converts the number of restaurants greater than 60 to a percent
percentGreaterThan60 = round((prepTimeGreaterThan60/totalRestaurants)*100,2)
print("Percent of restaurants with time greater than 60 minutes: "+str(percentGreaterThan60)+"%")
```

Output on next slide

Question #15 (cont.)

Output:



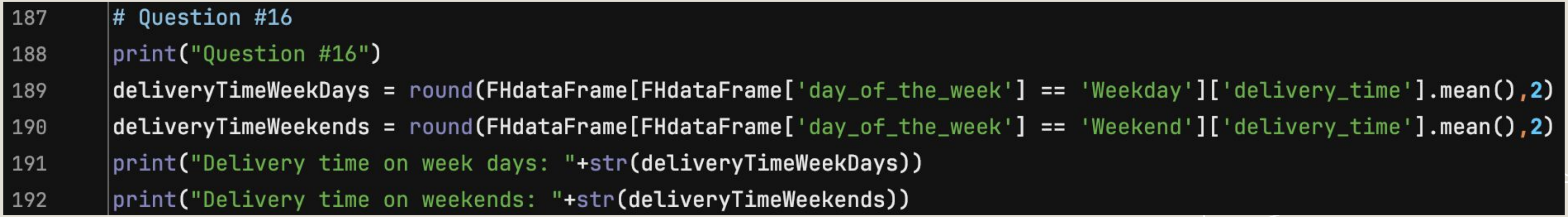
```
Question #15

Percent of restaurants with time greater than 60 minutes: 10.54%
```

- Observations:
  - 10.54% of restaurants have a total time to deliver food greater than 60 minutes

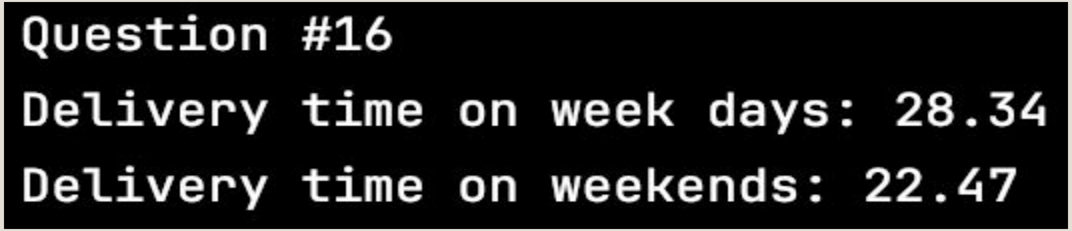
Question #16

Input:



```
# Question #16
print("Question #16")
deliveryTimeWeekDays = round(FHdataFrame[FHdataFrame['day_of_the_week'] == 'Weekday']['delivery_time'].mean(),2)
deliveryTimeWeekends = round(FHdataFrame[FHdataFrame['day_of_the_week'] == 'Weekend']['delivery_time'].mean(),2)
print("Delivery time on week days: "+str(deliveryTimeWeekDays))
print("Delivery time on weekends: "+str(deliveryTimeWeekends))
```

Output:



```
Question #16
Delivery time on week days: 28.34
Delivery time on weekends: 22.47
```

- Observations:
  - The average delivery time on weekdays is around 28.34 minutes, while only 22.47 minutes on weekends.

Thank you