Name: Ritik

Registration no.: 12020439

Roll no.: RK20MPB53

Section: K20MP

Int 353

Ca-1

Exploratory Data Analysis Project

Topic: Zomato Restaurants in India

About Dataset:

This Zomato exploratory data analysis is for the foodies to find best restaurants, value for money restaurants in their locality. It also helps to find their required cuisines in their locality

Additionally, this analysis caters the needs of people who are striving to get the best cuisine of the country and which locality of that country serves those cuisines with maximum number of restaurants.

The collected data has been stored in the Comma Separated Value file Zomato.csv. Each restaurant in the dataset is uniquely identified by its Restaurant Id. Every Restaurant contains the following variables:

- Restaurant Id: Unique id of every restaurant across various cities of the world
- Restaurant Name: Name of the restaurant
- City id: City id in which restaurant is located
- City: City in which restaurant is located
- Address: Address of the restaurant
- Locality: Location in the city
- Longitude: Longitude coordinate of the restaurant's location
- Latitude: Latitude coordinate of the restaurant's location
- Cuisines: Cuisines offered by the restaurant

- Average Cost for two: Cost for two people in different currencies
- Currency: Currency of the country
- Highlights: Highlights of restaurants
- Price range: range of price of food
- Aggregate Rating: Average rating out of 5
- Rating text: text on the basis of rating of rating
- Votes: Number of ratings casted by people

About Domain of Zomato:

Zomato is an Indian multinational restaurant aggregator and food delivery company founded by Deepinder Goyal and Pankaj Chaddah in 2008. Zomato provides information, menus and user-reviews of restaurants as well as food delivery options from partner restaurants in select cities. As of 2019, the service is available in 24 countries and in more than 10,000 cities.

Zomato focuses on the **QAAA model**, which promises and offers customers and partners quality, accessibility, and portfolio. The following are the values of Zomato, making Zomato unique and admirable: Zomato works to maintain an impartial platform. Hyper pure constantly helps to keep the food supplies high.

Why I Choose This Dataset?

Everyone wants to eat good and famous food but it’s very difficult to choose good restaurants and location of the restaurants. By this dataset I will able to visualize which restaurant is good, which restaurants price is pocket friendly, rating the restaurants and where it’s located.

Dataset Information:

This dataset consists **211944 rows** and **26 columns**

Columns Names are: ['res_id', 'name', 'establishment', 'url', 'address', 'city', 'city_id', 'locality', 'latitude', 'longitude', 'zipcode', 'country_id', 'locality_verbose', 'cuisines', 'timings', 'average_cost_for_two', 'price_range', 'currency', 'highlights', 'aggregate_rating', 'rating_text', 'votes', 'photo_count', 'opentable_support', 'delivery', 'takeaway']

Questions:

1) How many Rows and Columns are there in the dataset?
2) How many columns consists null values?
3) What are the ways to remove null values?
4) How many restaurants have more than one outlet?
5) Which restaurants have highest number of outlets?
6) Which city have highest number of restaurants?
7) How many restaurants supports delivery?

INT-353

CA-2

EDA PROJECT

NAME: RITIK

SECTION: K20MP

ROLL NO.: RK20MPB53

REGESTRATION NO.: 12020439

DATA CLEANING

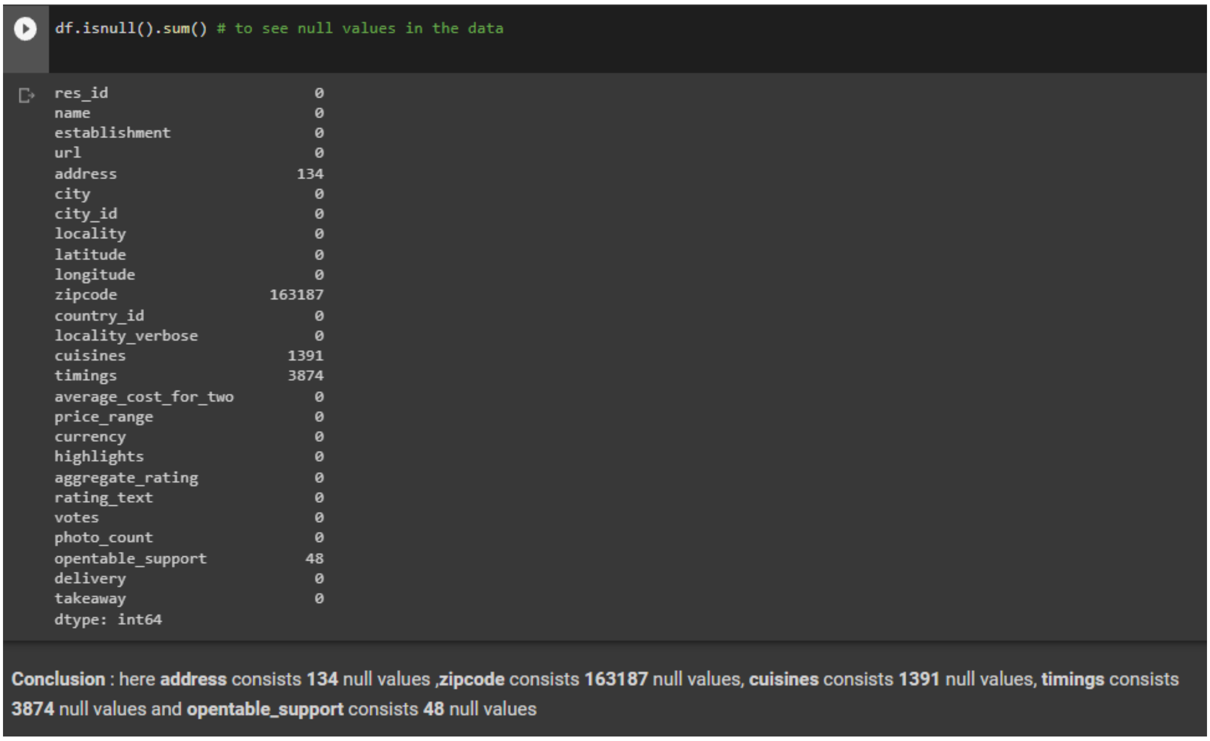
Data cleaning means fixing bad data in your data set.

Bad data could be:

- Empty cells
- Data in wrong format
- Wrong data
- Duplicates

Every Dataset consists missing values which interrupt’s the visualization of the data, to avoid this interruption we have to fill this missing values.

In this Zomato dataset missing values are:



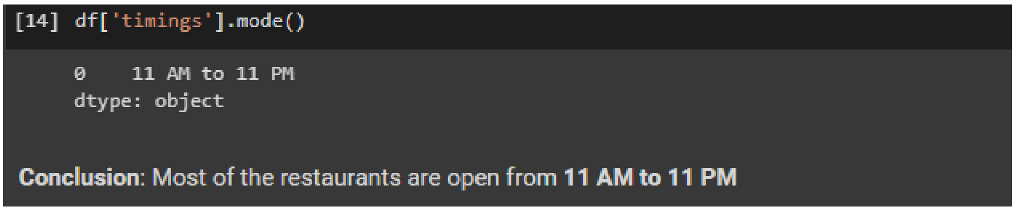
```
df.isnull().sum() # to see null values in the data

res_id                      0
name                        0
establishment               0
url                         0
address                   134
city                        0
city_id                     0
locality                    0
latitude                    0
longitude                   0
zipcode                163187
country_id                  0
locality_verbose            0
cuisines                 1391
timings                  3874
average_cost_for_two        0
price_range                 0
currency                    0
highlights                  0
aggregate_rating            0
rating_text                 0
votes                       0
photo_count                 0
opentable_support          48
delivery                    0
takeaway                    0
dtype: int64
```

Conclusion : here **address** consists **134** null values ,**zipcode** consists **163187** null values, **cuisines** consists **1391** null values, **timings** consists **3874** null values and **opentable_support** consists **48** null values

Conclusion: Here **address** consists **134** null values, **zipcode** consists **163187** null values, **cuisines** consist **1391** null values, **timings** consist **3874** null values and **opentable_support** consists **48** null values.

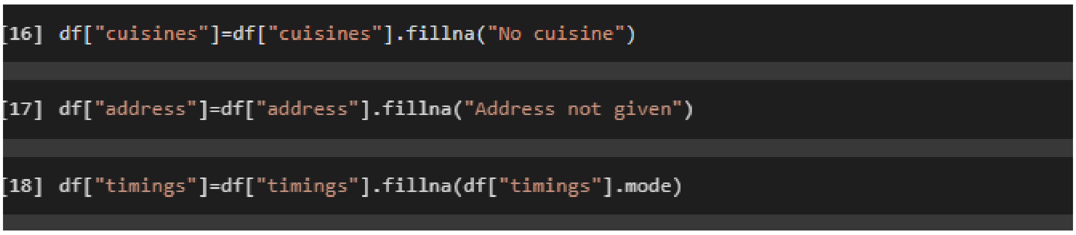
To Fix this missing data we fill the suitable values or we drop some data if that does not affect the dataset.



```
[14] df['timings'].mode()

     0    11 AM to 11 PM
     dtype: object
```

Conclusion: Most of the restaurants are open from **11 AM to 11 PM**

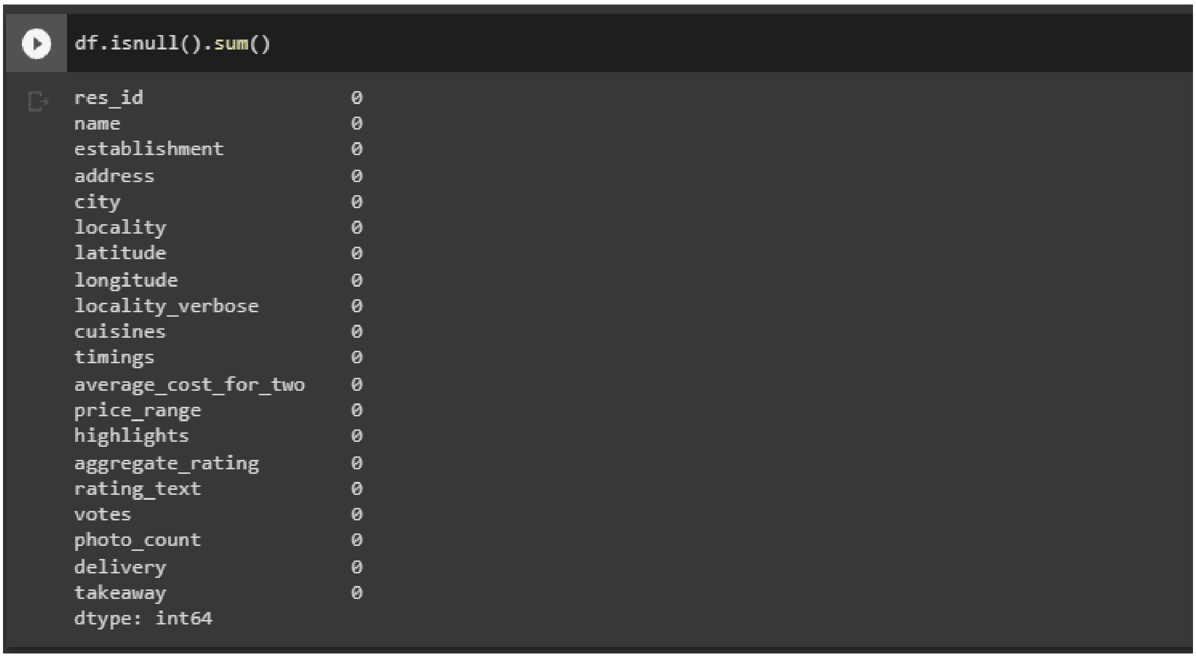


```
[16] df["cuisines"]=df["cuisines"].fillna("No cuisine")

[17] df["address"]=df["address"].fillna("Address not given")

[18] df["timings"]=df["timings"].fillna(df["timings"].mode)
```

Conclusion: The dateTypes of **address**, **cusines**, **timings** is object and these columns contain nullValues, therefore we can fill these columns by **"No cuisine", "Address not given", "timings"** respectively.



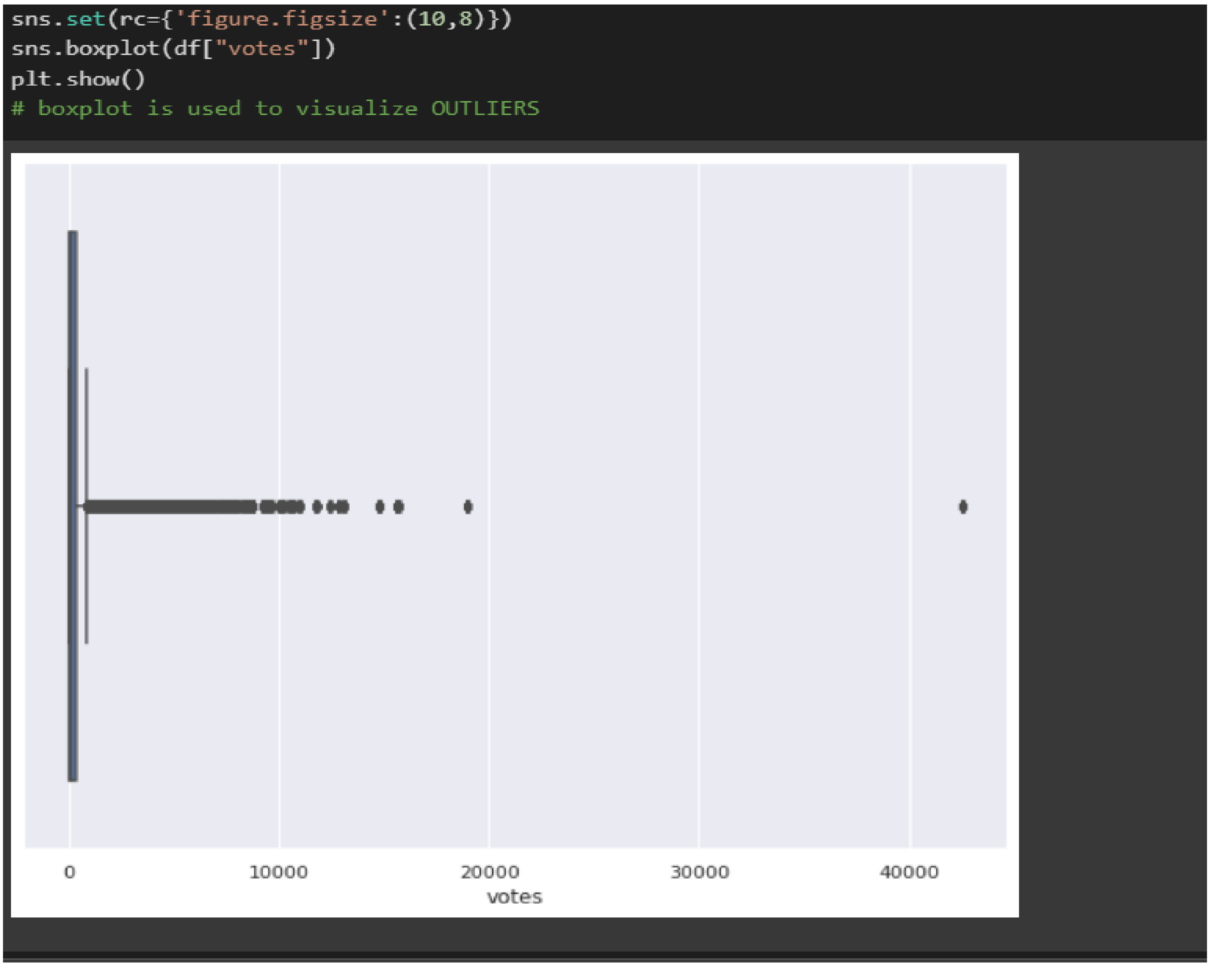
```
df.isnull().sum()

res_id                   0
name                     0
establishment            0
address                  0
city                     0
locality                 0
latitude                 0
longitude                0
locality_verbose         0
cuisines                 0
timings                  0
average_cost_for_two     0
price_range              0
highlights               0
aggregate_rating         0
rating_text              0
votes                    0
photo_count              0
delivery                 0
takeaway                 0
dtype: int64
```

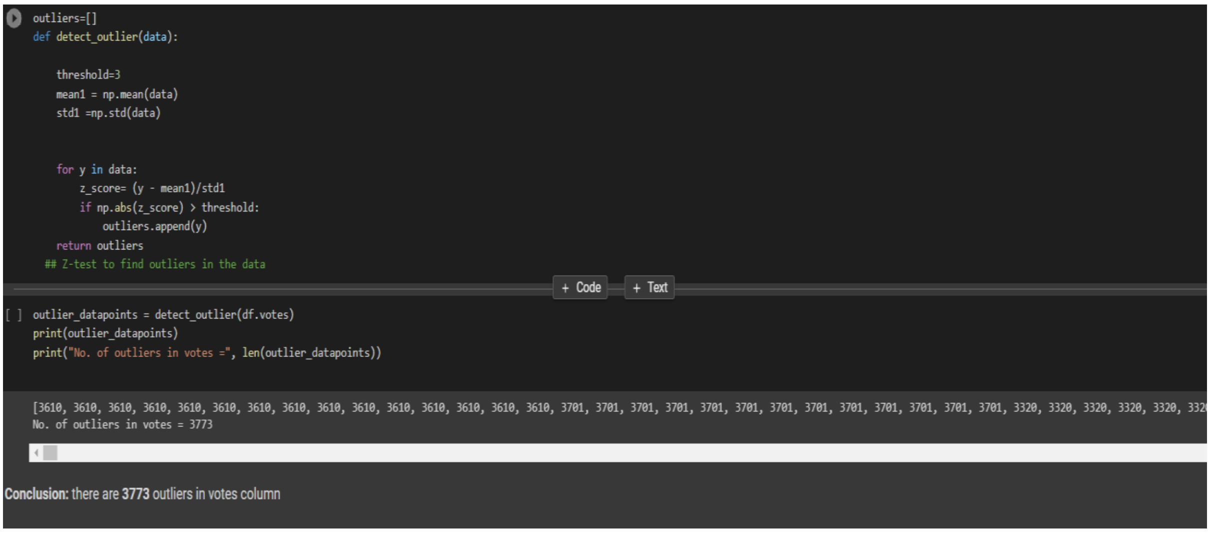
Now all columns have zero null values.

Outliers

An Outlier is a data-item/object that deviates significantly from the rest of the (so-called normal) objects. They can be caused by measurement or execution errors. The analysis for outlier detection is referred to as outlier mining. There are many ways to detect the outliers, and the removal process is the data frame same as removing a data item from the panda's data frame.



```
sns.set(rc={'figure.figsize':(10,8)})
sns.boxplot(df["votes"])
plt.show()
# boxplot is used to visualize OUTLIERS
```



```
outliers=[]
def detect_outlier(data):

    threshold=3
    mean1 = np.mean(data)
    std1 =np.std(data)


    for y in data:
        z_score= (y - mean1)/std1
        if np.abs(z_score) > threshold:
            outliers.append(y)
    return outliers
  ## Z-test to find outliers in the data
```

```
outlier_datapoints = detect_outlier(df.votes)
print(outlier_datapoints)
print("No. of outliers in votes =", len(outlier_datapoints))
```

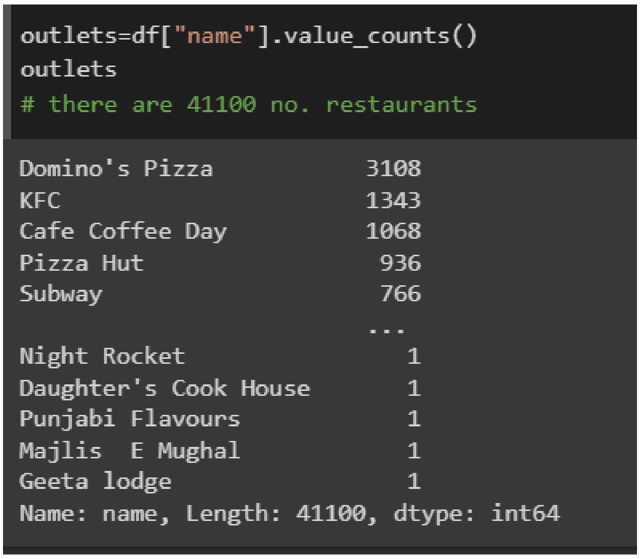
```
[3610, 3610, 3610, 3610, 3610, 3610, 3610, 3610, 3610, 3610, 3610, 3610, 3610, 3610, 3610, 3701, 3701, 3701, 3701, 3701, 3701, 3701, 3701, 3701, 3701, 3701, 3701, 3701, 3320, 3320, 3320, 3320, 3320, 332
No. of outliers in votes = 3773
```

Conclusion: there are **3773** outliers in votes column

Conclusion: There are **3773** outliers in votes column.

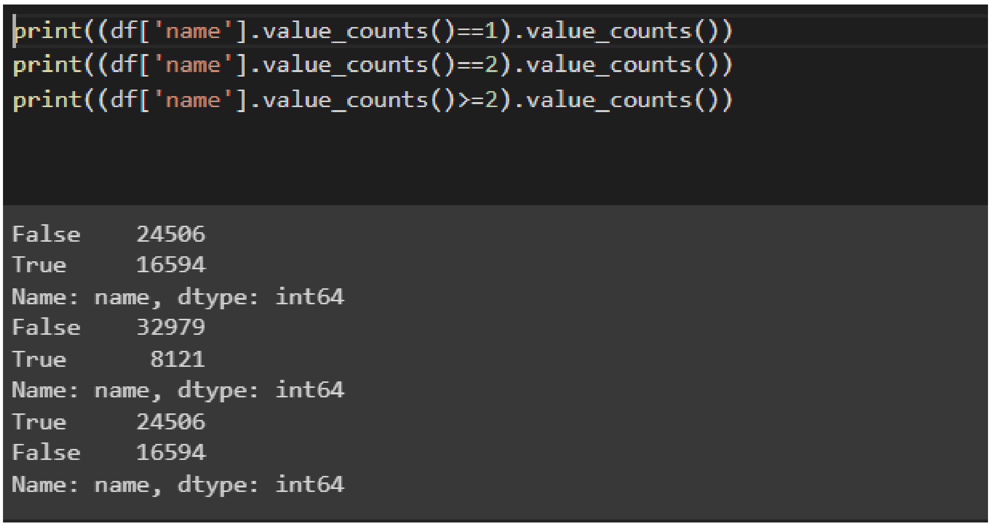
Univariant Analysis

Univariate analysis is the simplest form of analyzing data. “Uni” means “one”, so in other words your data has only one variable. It doesn’t deal with causes or relationships (unlike regression) and its major purpose is to describe; It takes data, summarizes that data and finds patterns in the data.



```
outlets=df["name"].value_counts()
outlets
# there are 41100 no. restaurants
```

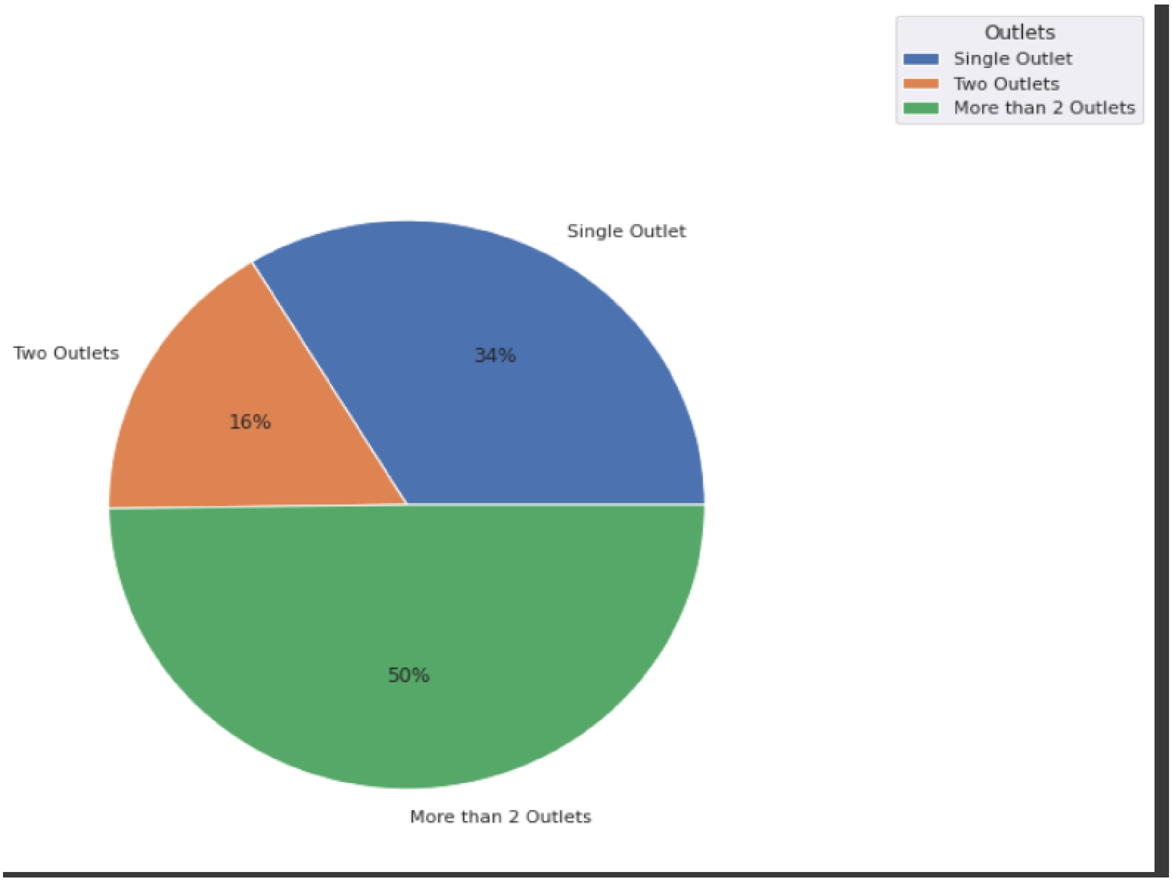
```
Domino's Pizza          3108
KFC                     1343
Cafe Coffee Day         1068
Pizza Hut                936
Subway                   766
                        ... 
Night Rocket               1
Daughter's Cook House      1
Punjabi Flavours           1
Majlis  E Mughal           1
Geeta lodge                1
Name: name, Length: 41100, dtype: int64
```



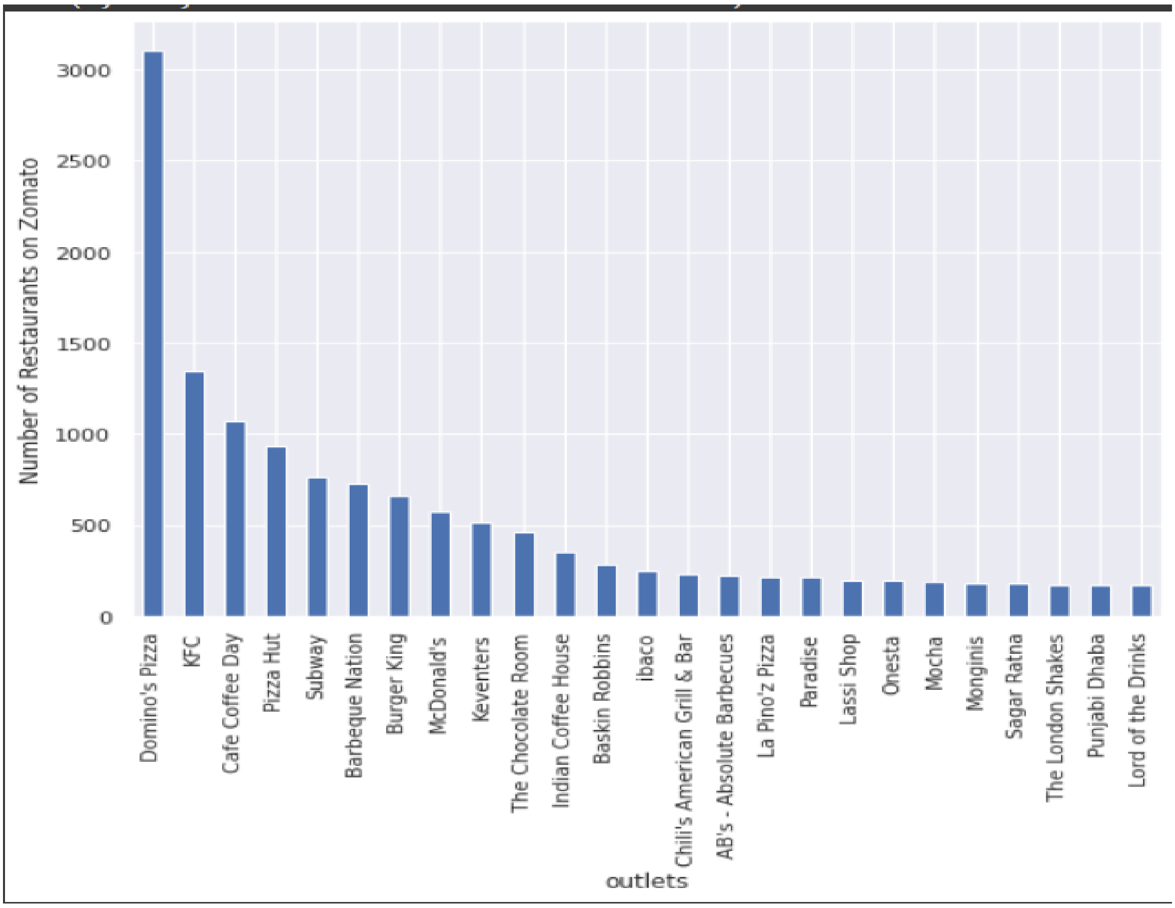
```
print((df['name'].value_counts()==1).value_counts())
print((df['name'].value_counts()==2).value_counts())
print((df['name'].value_counts()>=2).value_counts())
```

```
False    24506
True     16594
Name: name, dtype: int64
False    32979
True      8121
Name: name, dtype: int64
True     24506
False    16594
Name: name, dtype: int64
```

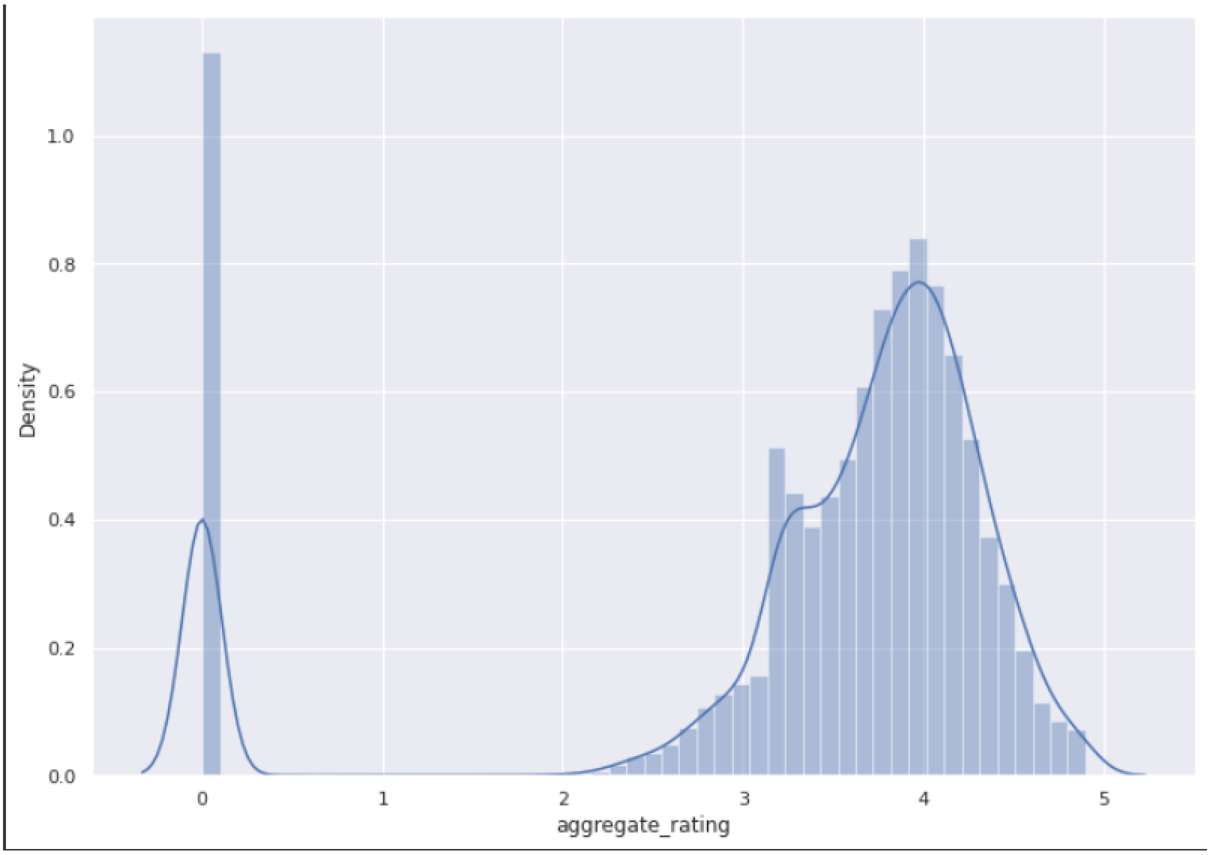
Conclusion: There are **41100** restaurants in which **16594** have **single outlets**, **8121** have **double outlets**, and **24506** have **more than 2 outlets**



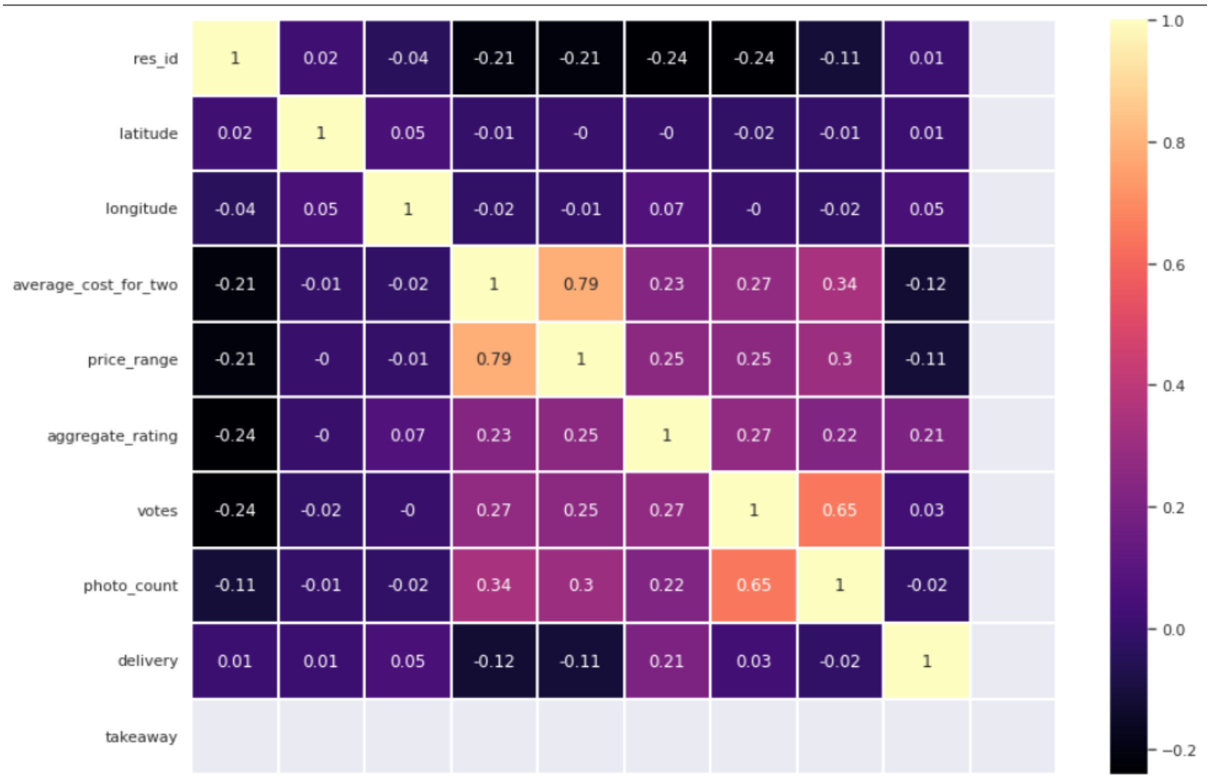
Conclusion: 50% of the restaurants have **more than two outlets**, **34%** restaurants have **single outlets** and rest **16%** restaurants have **double outlets**.



Conclusion: Domino's Pizza have highest numbers of outlets.

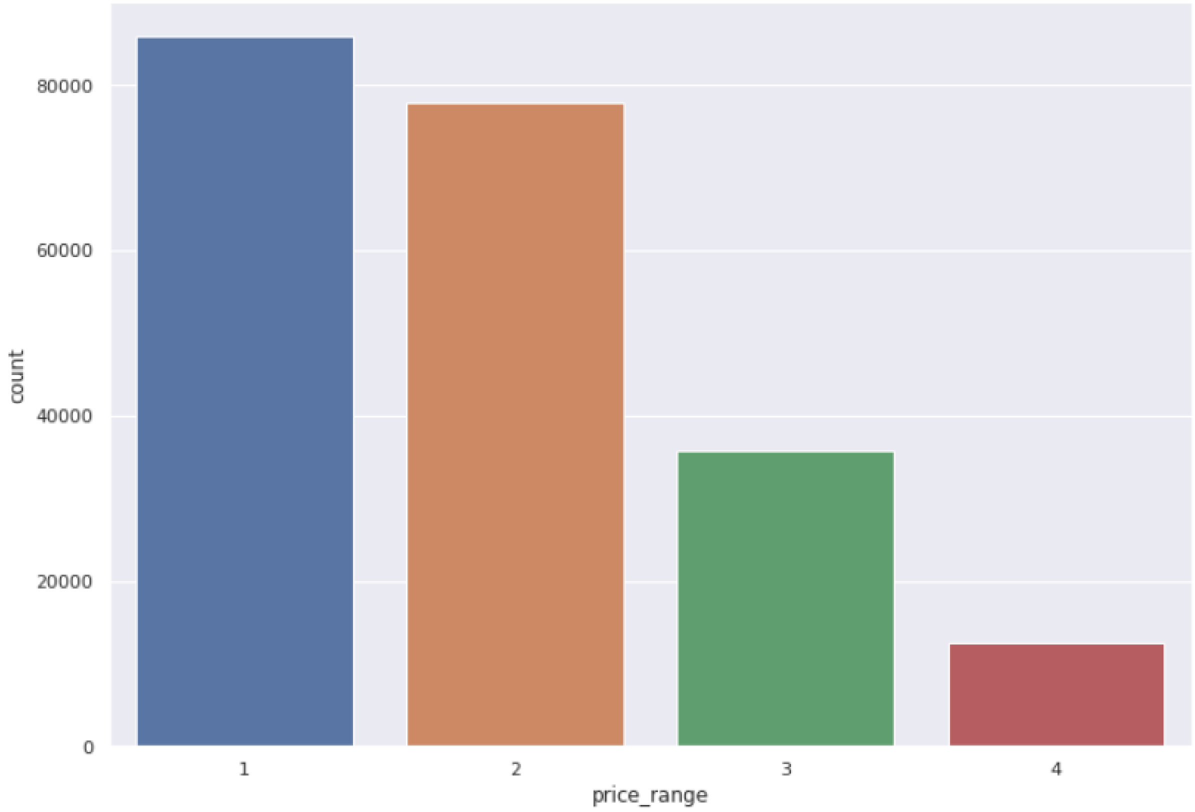


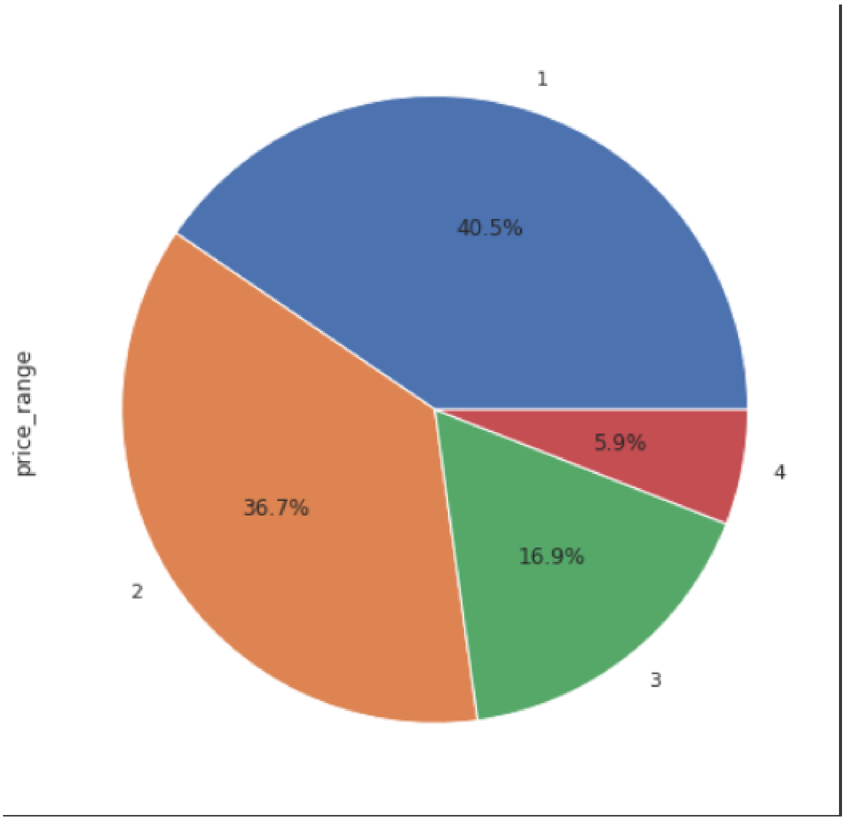
Conclusion: Most of the people don't give rating to the restaurants. But Average rating of Zomato restaurants is 4.



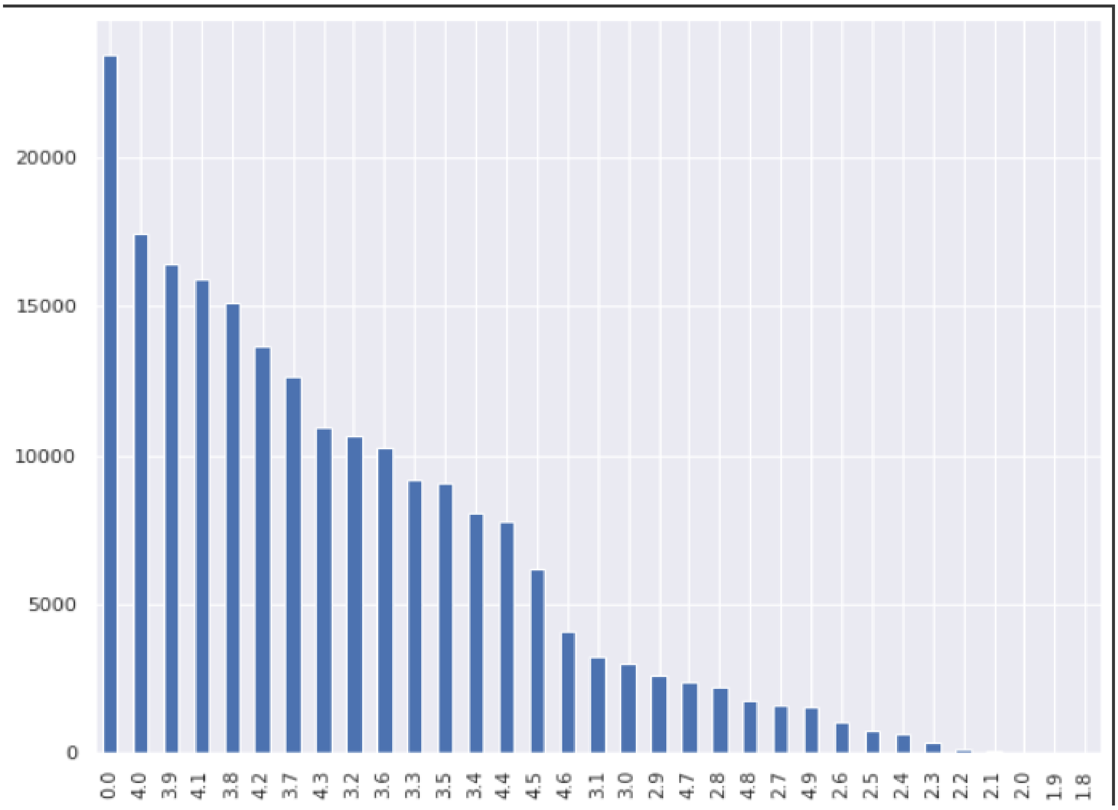
res_id	1	0.02	-0.04	-0.21	-0.21	-0.24	-0.24	-0.11	0.01	
latitude	0.02	1	0.05	-0.01	-0	-0	-0.02	-0.01	0.01	
longitude	-0.04	0.05	1	-0.02	-0.01	0.07	-0	-0.02	0.05	
average_cost_for_two	-0.21	-0.01	-0.02	1	0.79	0.23	0.27	0.34	-0.12	
price_range	-0.21	-0	-0.01	0.79	1	0.25	0.25	0.3	-0.11	
aggregate_rating	-0.24	-0	0.07	0.23	0.25	1	0.27	0.22	0.21	
votes	-0.24	-0.02	-0	0.27	0.25	0.27	1	0.65	0.03	
photo_count	-0.11	-0.01	-0.02	0.34	0.3	0.22	0.65	1	-0.02	
delivery	0.01	0.01	0.05	-0.12	-0.11	0.21	0.03	-0.02	1	
takeaway										

Conclusion: Price_range and Average_count_for_two have highest correlation

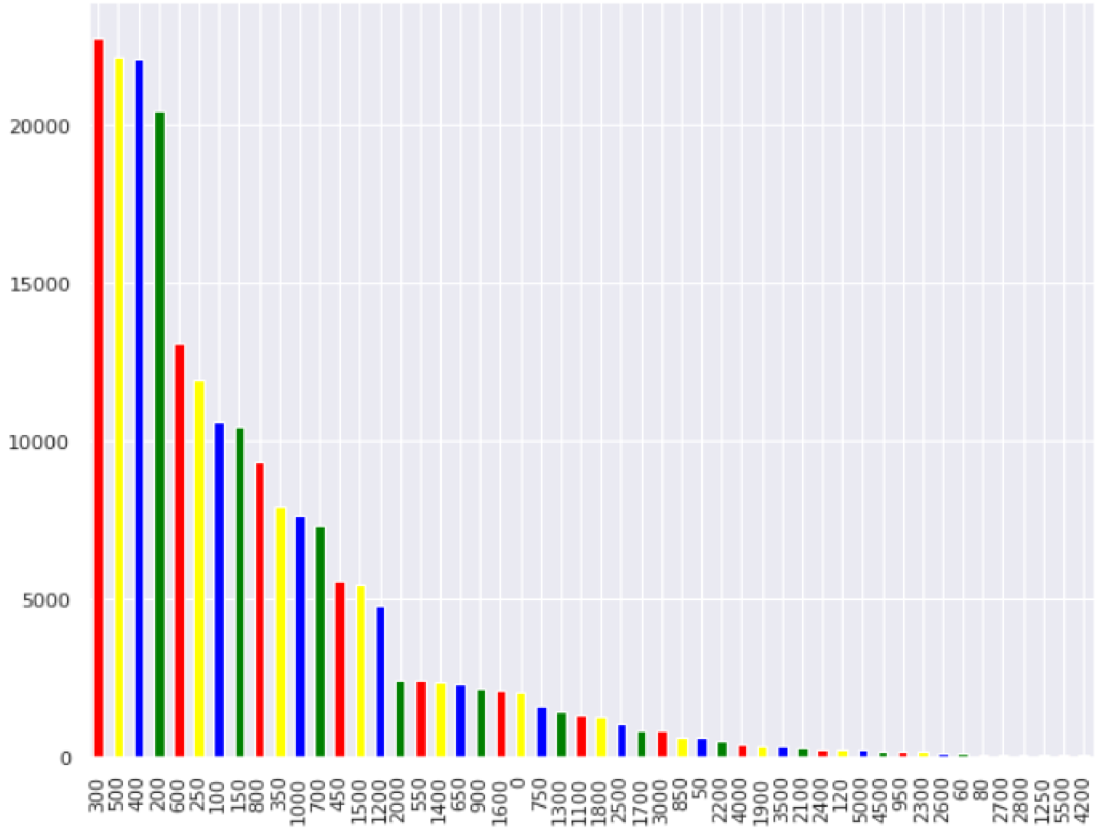




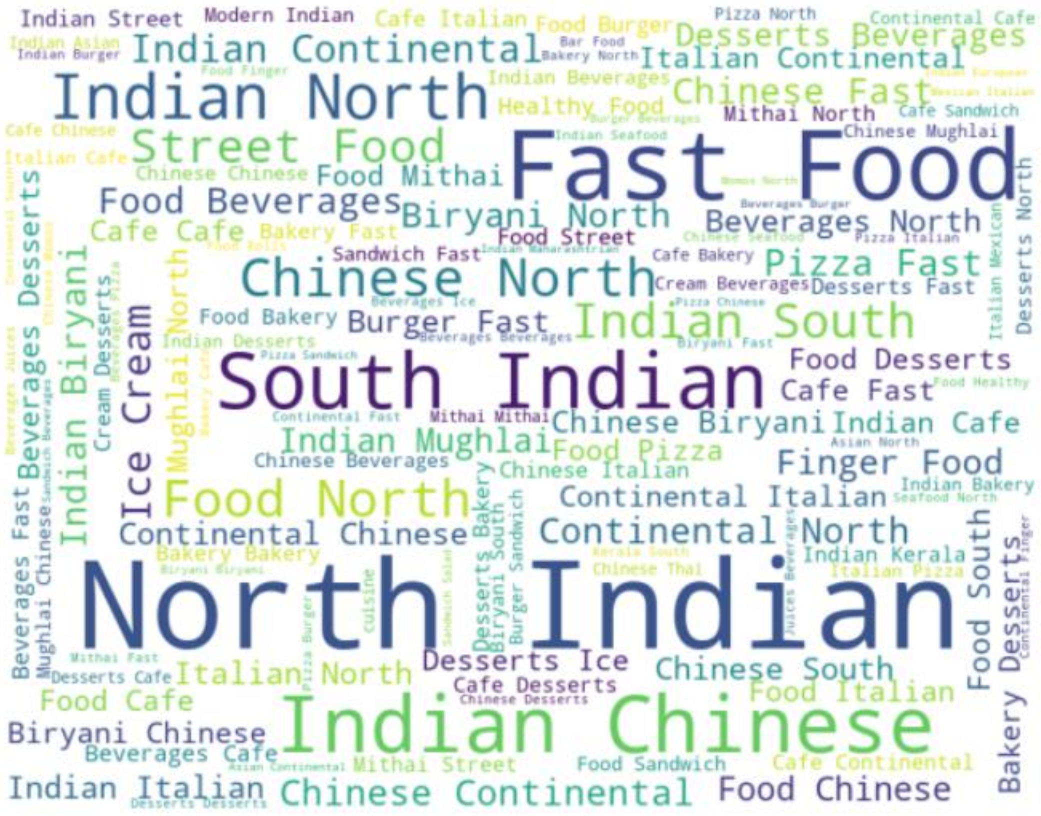
Conclusion: Price range 1 have highest percentage of 40% which means most of the people like to order price range of 1 food.



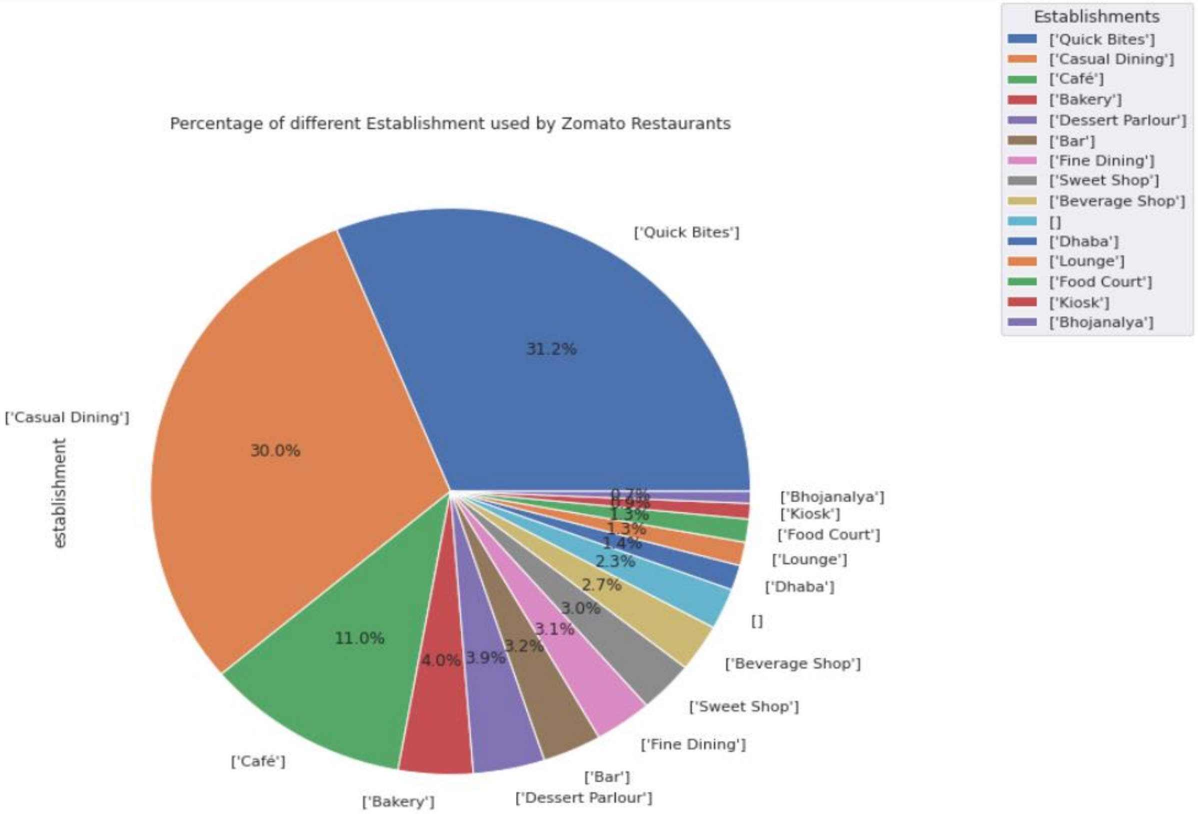
Conclusion: Most of the people close the app without giving rating but most of the restaurants average rating of 4.0



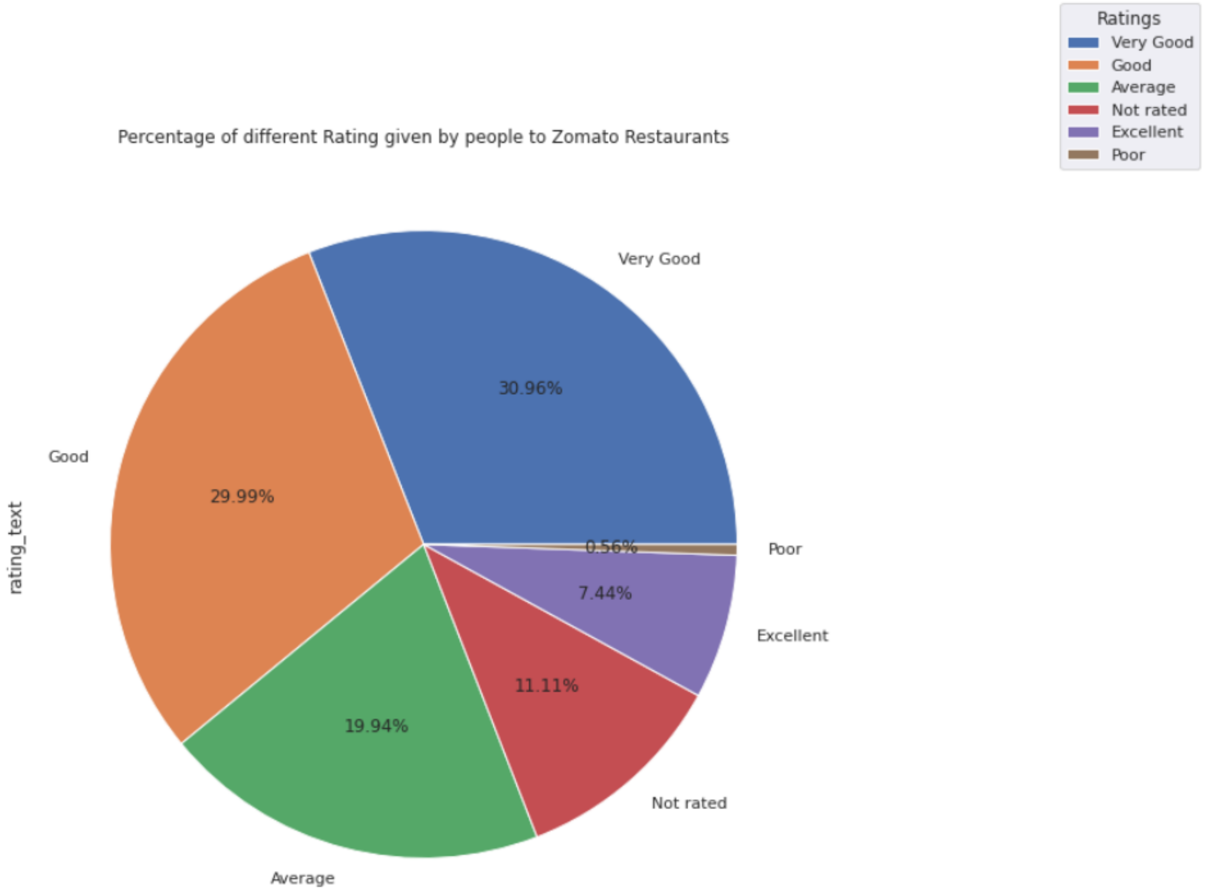
Conclusion: The Highest Average_cost_for_two is 300.



Conclusion: The **North Indian** cuisines is the famouse cuisines among all the cuisions.



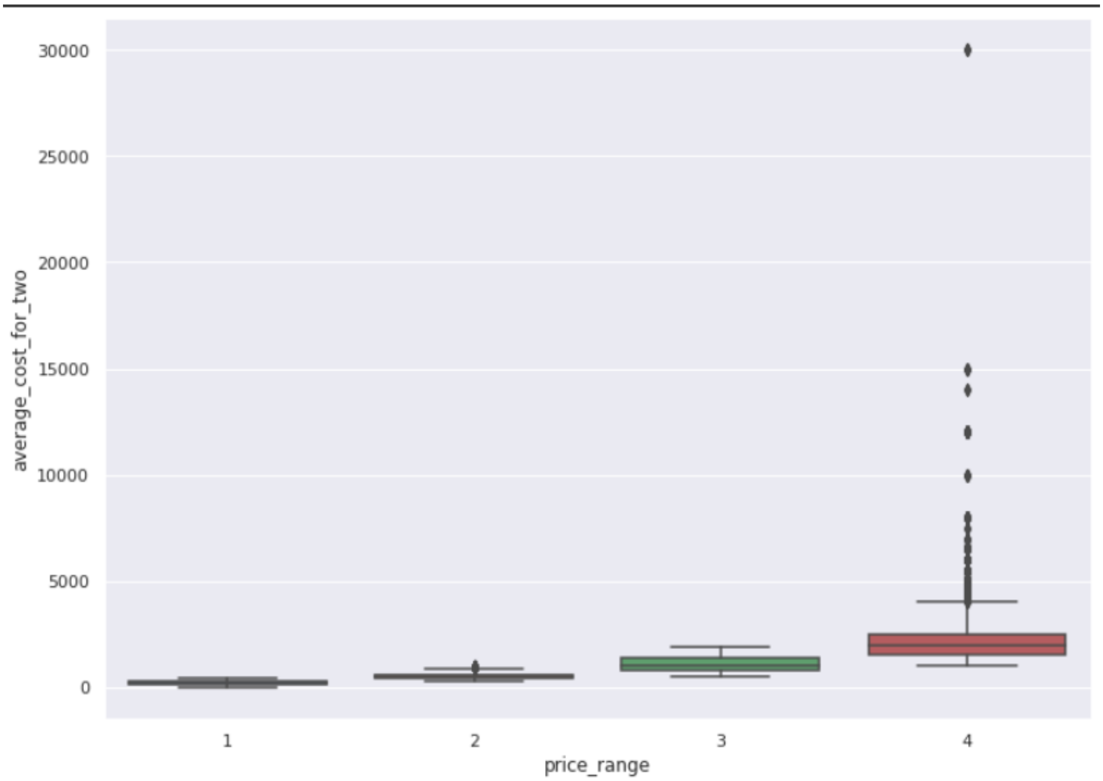
Conclusion: Quick Bites has the **highest** number of establishments of **31.2%**.

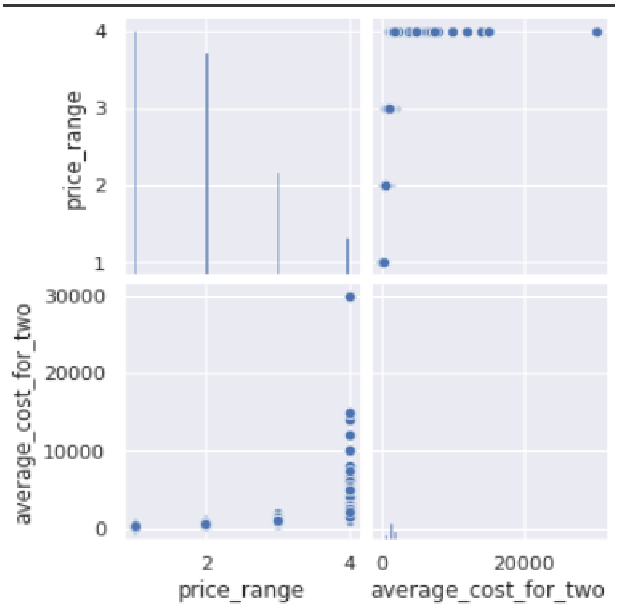


Conclusion: Most of the restaurants have **'very good'** rating of **30.96%.**

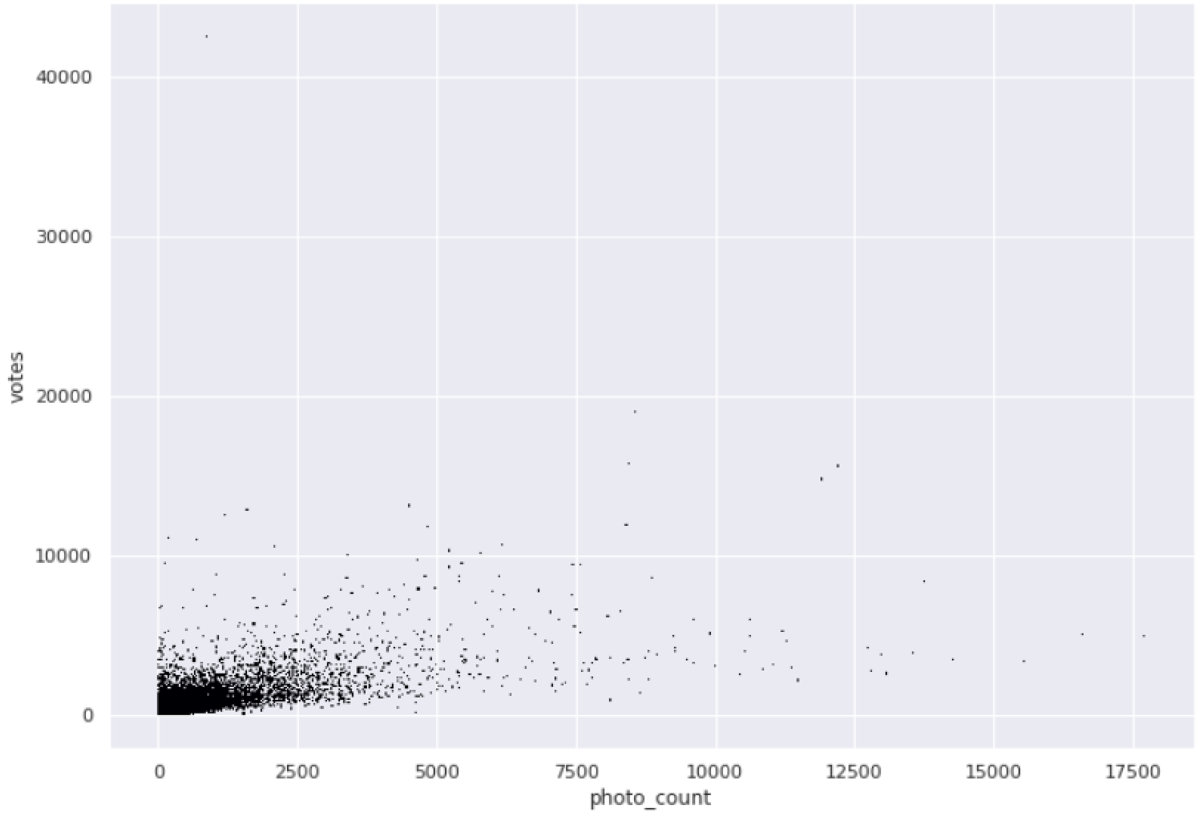
Bivariant Analysis

Bivariate analysis is one of the simplest forms of quantitative analysis. It involves the analysis of two variables, for the purpose of determining the empirical relationship between them. Bivariate analysis can be helpful in testing simple hypotheses of association.



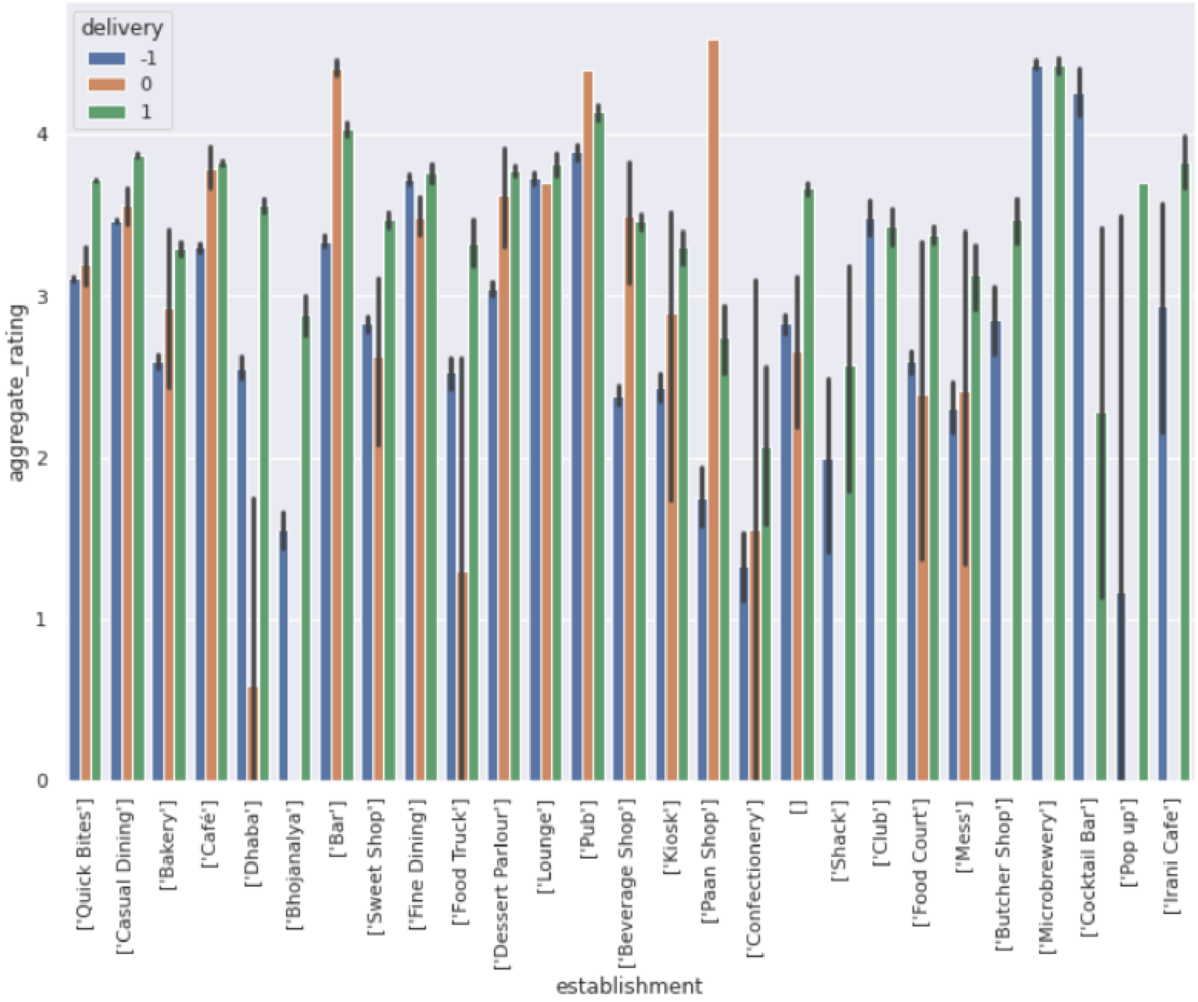


Conclusion: Most of the people order expensive food in a group of two and less expensive foods are order by single person.


40000
30000
20000
10000
0
votes
0
2500
5000
7500
10000
12500
15000
17500
photo_count

Multivariant Analysis

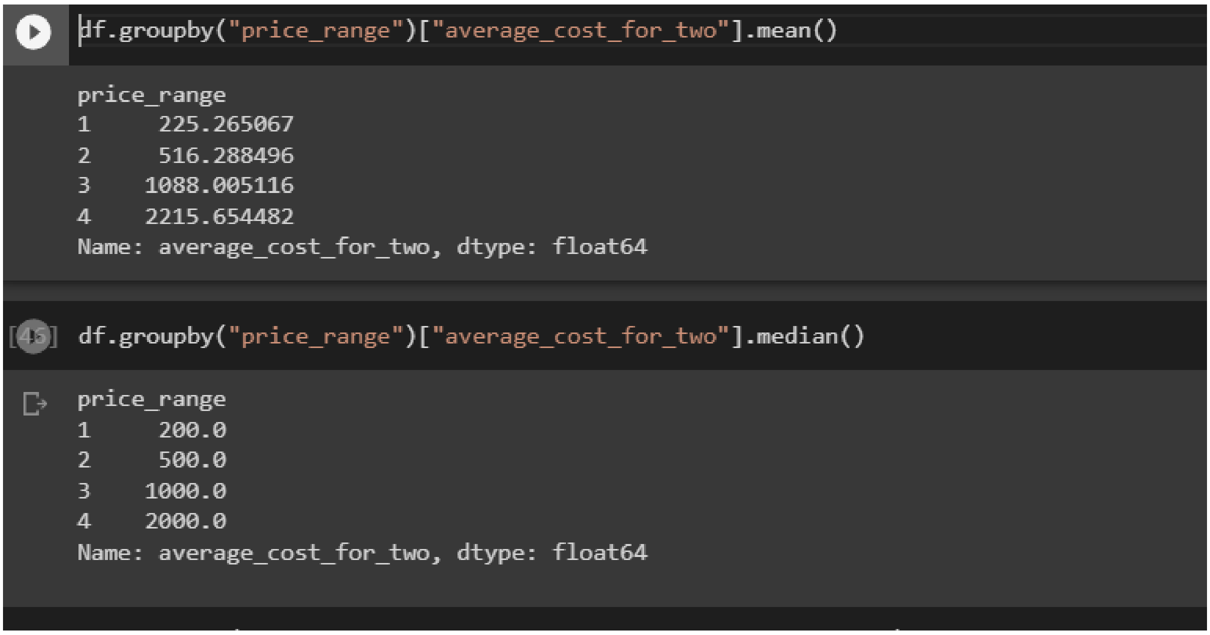
Multivariate analysis is defined as: The statistical study of data where multiple measurements are made on each experimental unit and where the relationships among multivariate measurements and their structure are important.



Conclusion: Delivery rating of **Microbrewery** has the highest delivery rating.

Statistical Analysis

Statistical analysis is the collection and interpretation of data in order to uncover patterns and trends. It is a component of data analytics. Statistical analysis can be used in situations like gathering research interpretations, statistical modelling or designing surveys and studies. It can also be useful for business intelligence organizations that have to work with large data volumes.



```
df.groupby("price_range")["average_cost_for_two"].mean()
```

```
price_range
1     225.265067
2     516.288496
3    1088.005116
4    2215.654482
Name: average_cost_for_two, dtype: float64
```

```
df.groupby("price_range")["average_cost_for_two"].median()
```

```
price_range
1     200.0
2     500.0
3    1000.0
4    2000.0
Name: average_cost_for_two, dtype: float64
```

Summary:

- Here address consists 134 null values, zipcode consists 163187 null values, cuisines consist 1391 null values, timings consist 3874 null values and opentable_support consists 48 null values.
- The dateTypes of **address**, **cusines**, **timings** is object and these columns contain nullValues, therefore we can fill these columns by **"No cuisine", "Address not given", "timings"** respectively.
- There are **3773** outliers in votes column.
- **50%** of the restaurants have **more than two outlets**, **34%** restaurants have **single outlets** and rest **16%** restaurants have **double outlets**.

- Domino's Pizza have highest numbers of outlets.
- Most of the people don't give rating to the restaurants. But Average rating of Zomato restaurants is 4.
- Price_range and Average_count_for_two have highest correlation
- Price range 1 have highest percentage of 40% which means most of the people like to order price range of 1 food.
- Most of the people close the app without giving rating but most of the restaurants average rating of 4.0
- The Highest Average_cost_for_two is 300
- The **North Indian** cuisines is the famouse cuisines among all the cuisions.
- **Quick Bites** has the **highest** number of establishments of **31.2%**.
- Most of the restaurants have **'very good'** rating of **30.96%.**
- Most of the people order expensive food in a group of two and less expensive foods are order by single person.