



zomato

Better food for more people

Capstone Project

BY FAIZ KHAN

Lets perform some Data Analytics on Zomato's DataSet

First we have to import major libraries that we are going to use in our project .

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
import warnings
warnings.filterwarnings("ignore")
```

In [2]: *# Load DataSet*

```
df=pd.read_csv("Indian-Resturants.csv")
```

Now we have to load DataSet

Lets check null
values, drop
duplicates and
fill null values

```
In [6]: df.isna().sum()
```

```
Out[6]: 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
```

```
In [7]: df.duplicated().sum()
```

```
Out[7]: 151527
```

```
In [8]: df.drop_duplicates(inplace=True)
```

```
In [9]: df.shape
```

```
Out[9]: (60417, 26)
```

```
In [13]: df.address=df.address.fillna("Unknown")
```

```
In [14]: df.cuisines=df.cuisines.fillna("Unknown")
```

```
In [15]: df.opentable_support=df.opentable_support.fillna(df.opentable_support.mean())
```

```
In [16]: df.zipcode=df.zipcode.fillna("unknown")
```

```
In [17]: a=df.timings.mode()
```

```
In [18]: a
```

```
Out[18]: 0    11 AM to 11 PM
         Name: timings, dtype: object
```

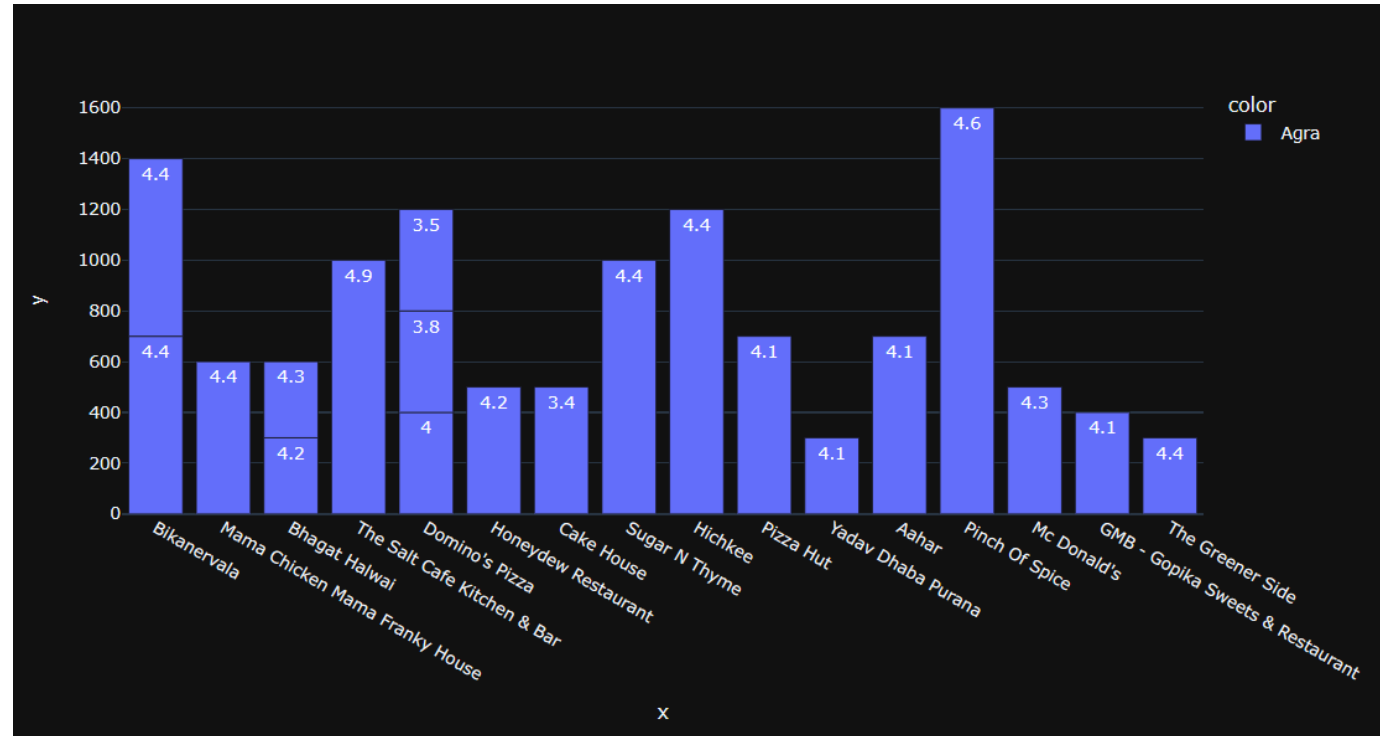
```
In [19]: df.timings=df.timings.fillna("a")
```

Now we can
see that
there is no
null value in
DataSet

```
df.isna().sum()
```

res_id	0
name	0
establishment	0
url	0
address	0
city	0
city_id	0
locality	0
latitude	0
longitude	0
zipcode	0
country_id	0
locality_verbose	0
cuisines	0
timings	0
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	0
delivery	0
takeaway	0
dtype: int64	

PLOTTING HORIZONTAL BAR CHATS (MULTICOLOR)



Average Cost for two people in the best restaurant of the city Agra that is Pinch Of Spices with the Aggregate ratings of 4.6 is Rs1600 and cost of two people in least rated restaurant of Agra that is Yadav Dhaba Purana with the ratings of 4.1 is Rs 300 only

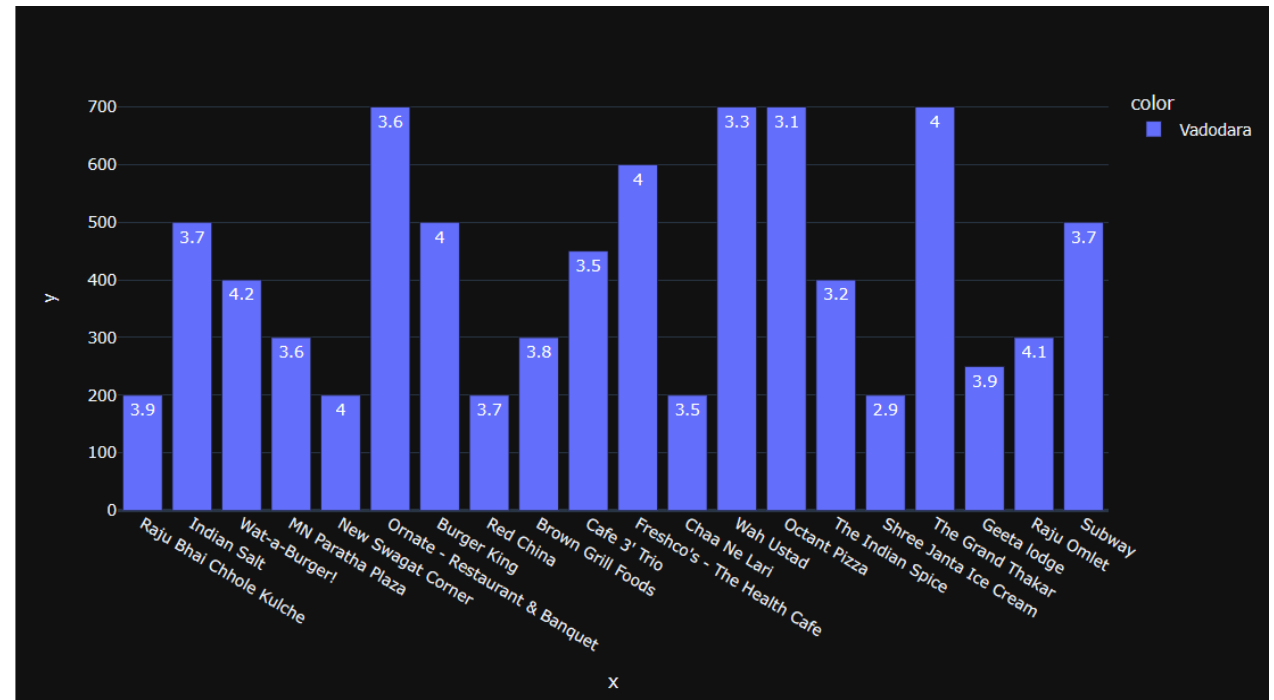
- CODE:-

```
px.bar(x=df.name.head(20),y=df.average_cost_for_two.  
head(20),color=df.city.head(20),text=df.aggregate_ratin  
g.head(20),template="plotly_dark")
```

In Vadodara city

- CODE:-

```
px.bar(x=df.name.tail(20),y=df.average_cost_for_two.tail(20),color=df.city.tail(20),text=df.aggregate_rating.tail(20),template="plotly_dark")
```



In Vadodara city average cost of meal for 2 people is Rs 700 in the best restaurant that is The Grand Thakar with the ratings of 4 and for average cost for 2 people in least rated restaurant that is Shree Janta Ice Cream with the ratings of 2.9 is Rs 200 only

Using Groupby, Apply, and Unique Functions

First we use Groupby

```
In [24]: df.groupby(["name", "city", "cuisines", "rating_text"])["average_cost_for_two"].unique().reset_index()
```

Out[24]:

	name	city	cuisines	rating_text	average_cost_for_two
0	# Wednesday	Kanpur	Fast Food, Chinese, North Indian	Good	[250]
1	#1, Culinary Avenue - The Red Maple	Indore	North Indian, Chinese, Continental	Good	[1500]
2	#788 Avenue	Hyderabad	Continental, Italian, North Indian	Good	[1900]
3	#BC	Jaipur	North Indian, Continental, Finger Food	Very Good	[2000]
4	#BEiR	Bangalore	Finger Food, Continental, Mexican, Burger	Very Good	[1100]
...
51528	Food Street - Veg	Kota	Fast Food	Average	[250]
51529	ट 4 Tasty	Goa	Street Food	Good	[300]
51530	द वेगे टेबल	Pune	North Indian, Continental, Italian	Very Good	[1000]
51531	स्पेस Bar	Pune	North Indian, Chinese, Continental	Very Good	[1700]
51532	ह-tea The Tea Hut	Ludhiana	Cafe, North Indian, Italian, Chinese, Fast Foo...	Very Good	[550]

51533 rows × 5 columns

By using Groupby method we can see Name of restaurant , City in which restaurant is located , Cuisine restaurant is offering their ratings and average cost of 2 people

Apply Method

- we are making new column in which we are giving 5% discount on average cost of two people

```
In [25]: df.insert(21, "Discount", df.average_cost_for_two*5/100)
```

```
In [26]: df
```

```
Out[26]:
```

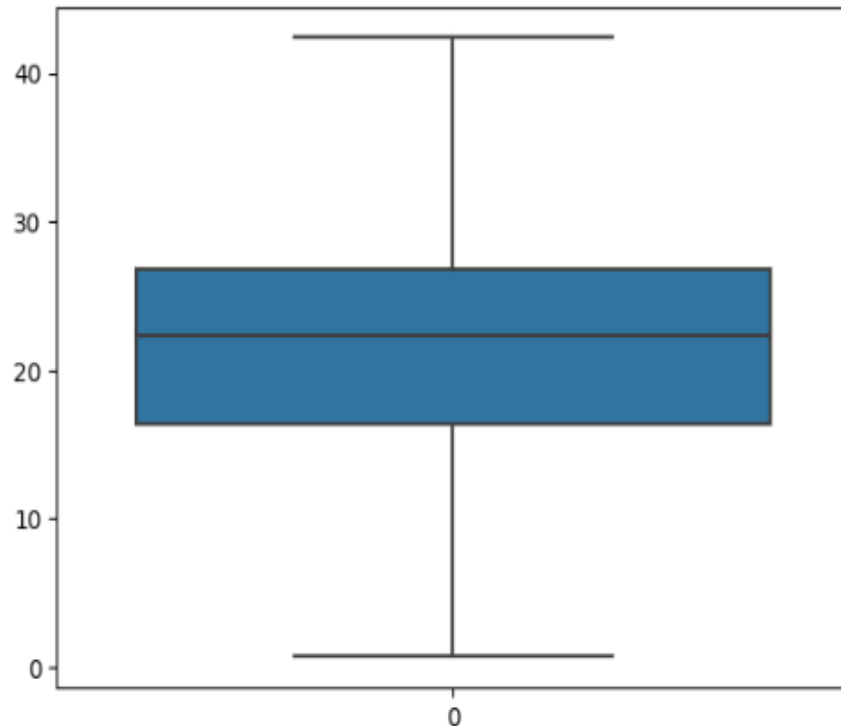
city	city_id	locality	latitude	longitude	...	currency	highlights	aggregate_rating	rating_text	Discount	votes	photo_count	opentable_support	delivery
Agra	34	Khandari	27.211450	78.002381	...	Rs.	['Lunch', 'Takeaway Available', 'Credit Card', ...]	4.4	Very Good	35.0	814	154	0.0	-1
Agra	34	Agra Cantt	27.160569	78.011583	...	Rs.	['Delivery', 'No Alcohol Available', 'Dinner', ...]	4.4	Very Good	30.0	1203	161	0.0	-1
Agra	34	Shahganj	27.182938	77.979684	...	Rs.	['No Alcohol Available', 'Dinner', 'Takeaway A...]	4.2	Very Good	15.0	801	107	0.0	1
Agra	34	Civil Lines	27.205668	78.004799	...	Rs.	['Takeaway Available', 'Credit Card', 'Lunch', ...]	4.3	Very Good	15.0	693	157	0.0	1
Agra	34	Tajganj	27.157709	78.052421	...	Rs.	['Lunch', 'Serves Alcohol', 'Cash', 'Credit Ca...]	4.9	Excellent	50.0	470	291	0.0	1

NOW WE WILL REMOVE OUTLIERS

Outliers in Latitude

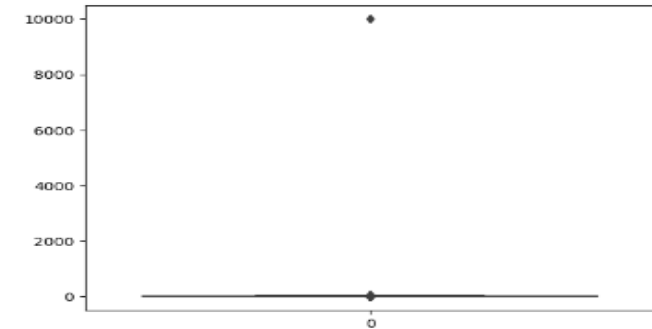
```
In [37]: sns.boxplot(df.latitude)
```

```
Out[37]: <Axes: >
```



```
In [27]: sns.boxplot(df.latitude)
```

```
Out[27]: <Axes: >
```



```
In [28]: Q1,Q3=df.latitude.quantile([0.25,0.75])
```

```
In [29]: Q1,Q3
```

```
Out[29]: (16.324755463800003, 26.744388798800003)
```

```
In [30]: IQR=Q3-Q1
```

```
In [31]: IQR
```

```
Out[31]: 10.419633335
```

```
In [32]: UL=Q3+1.5*IQR  
LL=Q1-1.5*IQR
```

```
In [33]: UL,LL
```

```
Out[33]: (42.3738388013, 0.6953054613000002)
```

```
In [34]: len(df[df.latitude<LL])
```

```
Out[34]: 951
```

```
In [35]: len(df[df.latitude>UL])
```

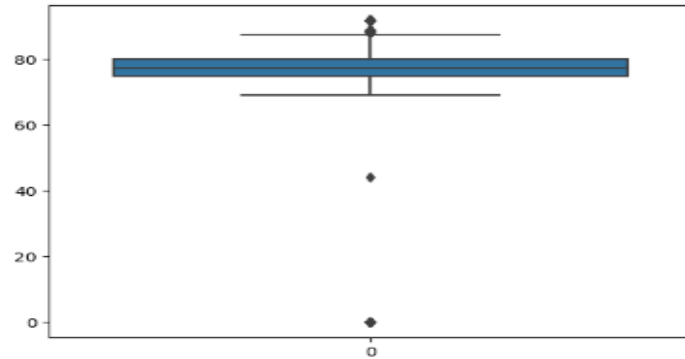
```
Out[35]: 2
```

```
In [36]: df.latitude=np.where(df.latitude<LL,LL,df.latitude)  
df.latitude=np.where(df.latitude>UL,UL,df.latitude)
```

Outliers in Longitude

```
In [38]: sns.boxplot(df.longitude)
```

```
Out[38]: <Axes: >
```



```
In [39]: Q1,Q3=df.longitude.quantile([0.25,0.75])
```

```
In [40]: Q1,Q3
```

```
Out[40]: (74.65402912350001, 79.9281898141)
```

```
In [41]: IQR=Q3-Q1
```

```
In [42]: IQR
```

```
Out[42]: 5.2741606905999845
```

```
In [43]: UL=Q3+1.5*IQR  
LL=Q1-1.5*IQR
```

```
In [44]: UL,LL
```

```
Out[44]: (87.83943084999997, 66.74278808760003)
```

```
In [45]: len(df[df.longitude>UL])
```

```
Out[45]: 2938
```

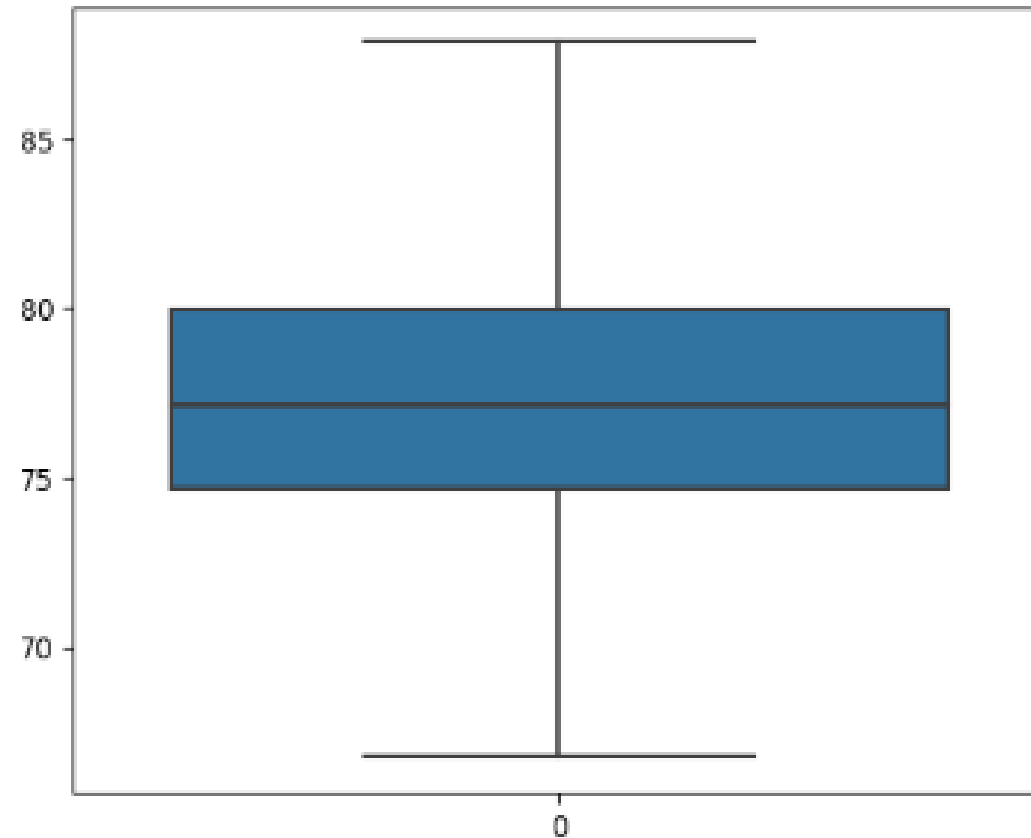
```
In [46]: len(df[df.longitude<LL])
```

```
Out[46]: 957
```

```
In [47]: df.longitude=np.where(df.longitude>UL,UL,df.longitude)  
df.longitude=np.where(df.longitude<LL,LL,df.longitude)
```

```
In [48]: sns.boxplot(df.longitude)
```

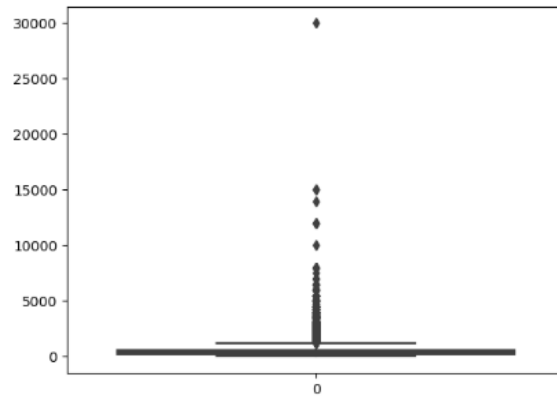
```
Out[48]: <Axes: >
```



Outliers in Average Cost For Two

```
In [49]: sns.boxplot(df.average_cost_for_two)
```

```
Out[49]: <Axes: >
```



```
In [50]: Q1,Q3=df.average_cost_for_two.quantile([0.25,0.75])
```

```
In [51]: Q1,Q3
```

```
Out[51]: (200.0, 600.0)
```

```
In [52]: IQR=Q3-Q1
```

```
In [53]: UL=Q3+1.5*IQR  
LL=Q1-1.5*IQR
```

```
In [54]: UL,LL
```

```
Out[54]: (1200.0, -400.0)
```

```
In [55]: len(df[df.average_cost_for_two>UL])
```

```
Out[55]: 5291
```

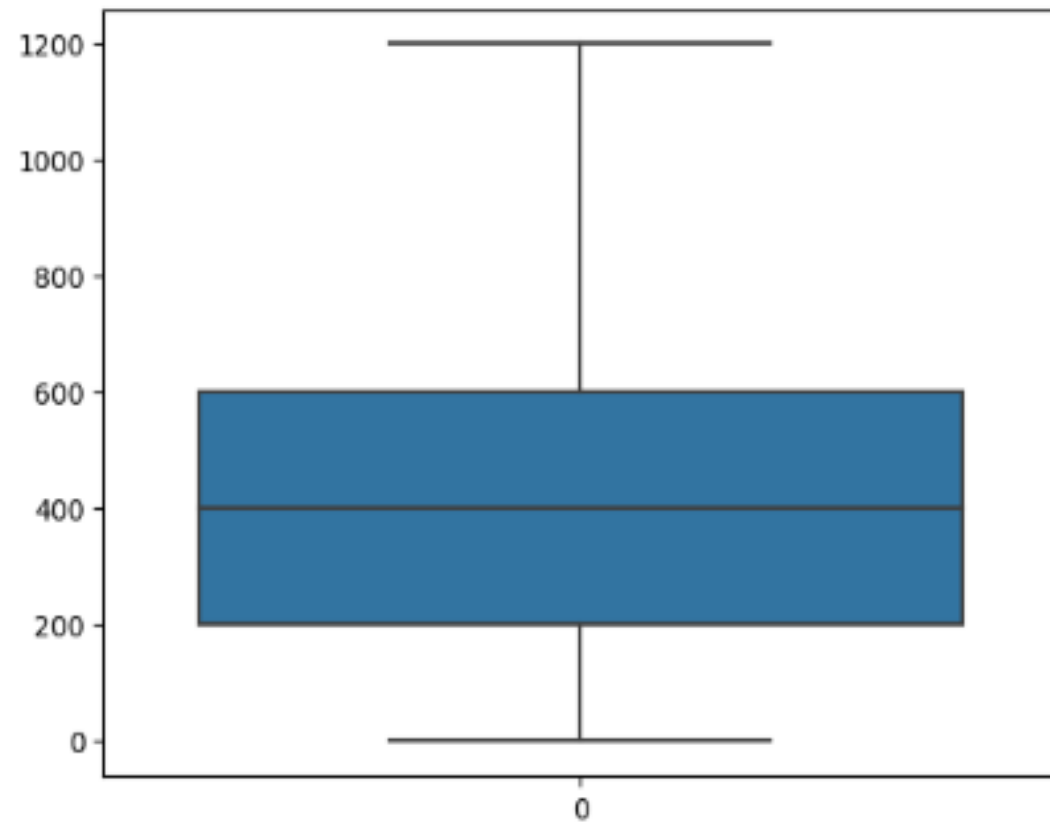
```
In [56]: len(df[df.average_cost_for_two<LL])
```

```
Out[56]: 0
```

```
In [57]: df.average_cost_for_two=np.where(df.average_cost_for_two>UL,UL,df.average_cost_for_two)
```

```
In [58]: sns.boxplot(df.average_cost_for_two)
```

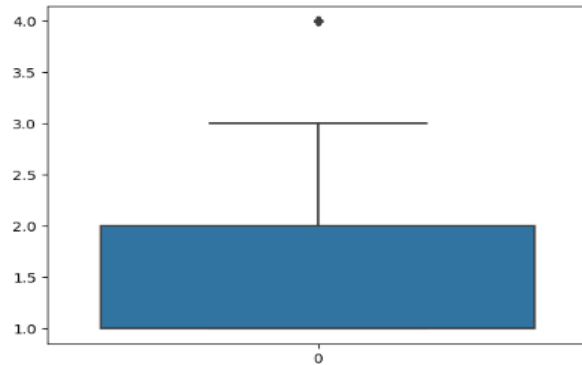
```
Out[58]: <Axes: >
```



Outliers in Price Range

```
In [60]: sns.boxplot(df.price_range)
```

```
Out[60]: <Axes: >
```



```
In [61]: Q1,Q3=df.price_range.quantile([0.25,0.75])
```

```
In [62]: Q1,Q3
```

```
Out[62]: (1.0, 2.0)
```

```
In [63]: IQR=Q3-Q1
```

```
In [64]: IQR
```

```
Out[64]: 1.0
```

```
In [65]: UL=Q3+1.5*IQR  
LL=Q1-1.5*IQR
```

```
In [66]: UL,LL
```

```
Out[66]: (3.5, -0.5)
```

```
In [67]: len(df[df.price_range>UL])
```

```
Out[67]: 3058
```

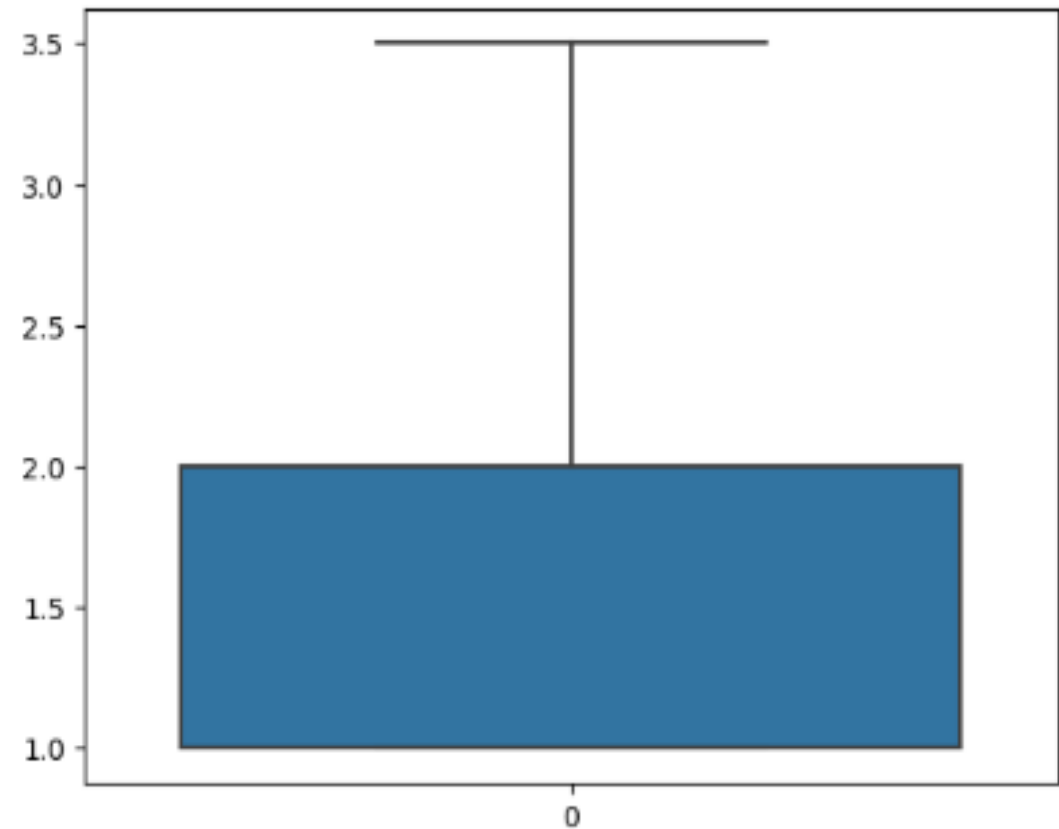
```
In [68]: len(df[df.price_range<LL])
```

```
Out[68]: 0
```

```
In [69]: df.price_range=np.where(df.price_range>UL,UL,df.price_range)
```

```
In [70]: sns.boxplot(df.price_range)
```

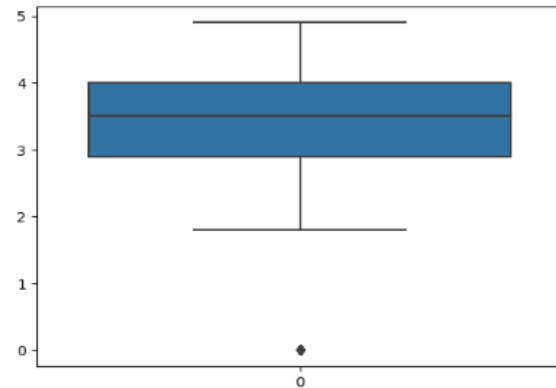
```
Out[70]: <Axes: >
```



Outliers in Aggregate Ratings

```
In [71]: sns.boxplot(df.aggregate_rating)
```

```
Out[71]: <Axes: >
```



```
In [72]: Q1,Q3=df.aggregate_rating.quantile([0.25,0.75])
```

```
In [73]: Q1,Q3
```

```
Out[73]: (2.9, 4.0)
```

```
In [74]: IQR=Q3-Q1
```

```
In [75]: IQR
```

```
Out[75]: 1.1
```

```
In [76]: UL=Q3+1.5*IQR  
LL=Q1-1.5*IQR
```

```
In [77]: len(df[df.aggregate_rating>UL])
```

```
Out[77]: 0
```

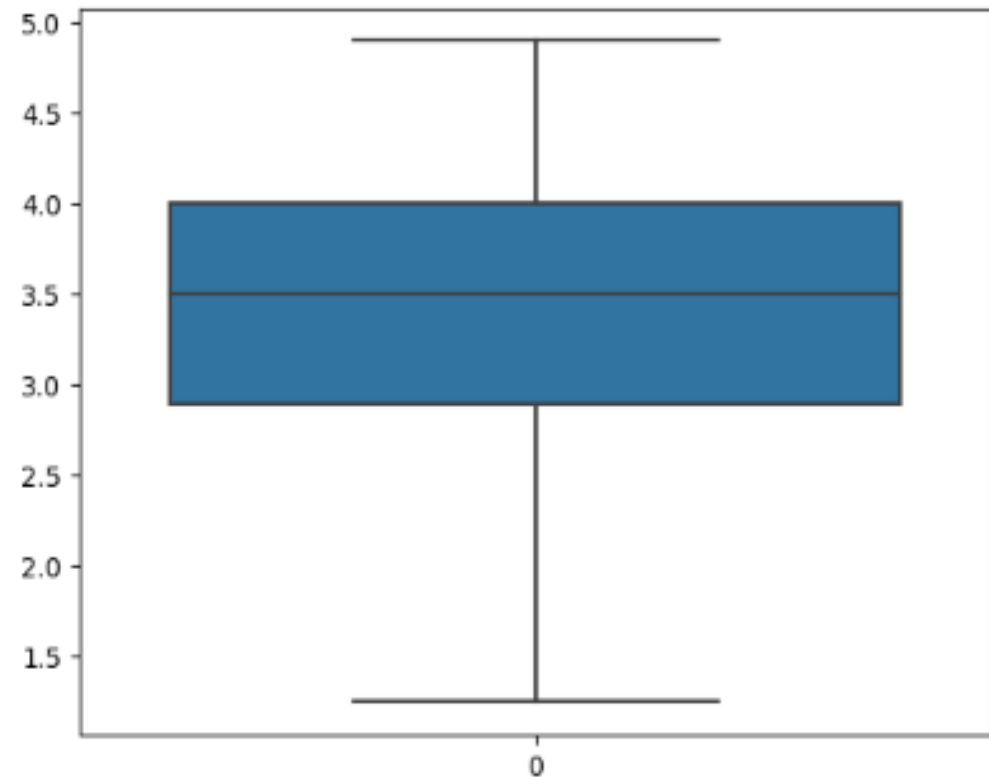
```
In [78]: len(df[df.aggregate_rating<LL])
```

```
Out[78]: 10159
```

```
In [79]: df.aggregate_rating=np.where(df.aggregate_rating<LL,LL,df.aggregate_rating)
```

```
In [80]: sns.boxplot(df.aggregate_rating)
```

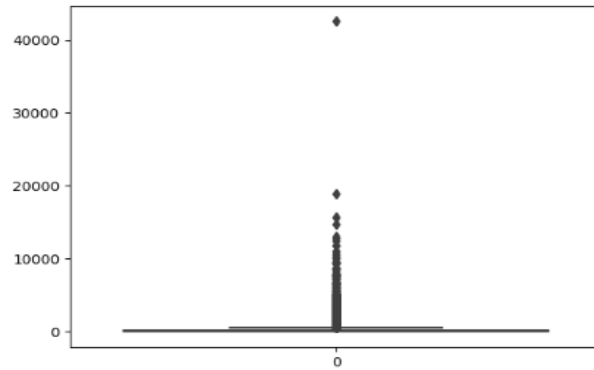
```
Out[80]: <Axes: >
```



Outliers in Votes

```
In [81]: sns.boxplot(df.votes)
```

```
Out[81]: <Axes: >
```



```
In [82]: Q1,Q3=df.votes.quantile([0.25,0.75])
```

```
In [83]: Q1,Q3
```

```
Out[83]: (7.0, 207.0)
```

```
In [84]: IQR=Q3-Q1
```

```
In [85]: IQR
```

```
Out[85]: 200.0
```

```
In [86]: UL=Q3+1.5*IQR  
LL=Q1-1.5*IQR
```

```
In [87]: UL,LL
```

```
Out[87]: (507.0, -293.0)
```

```
In [88]: len(df[df.votes>UL])
```

```
Out[88]: 7962
```

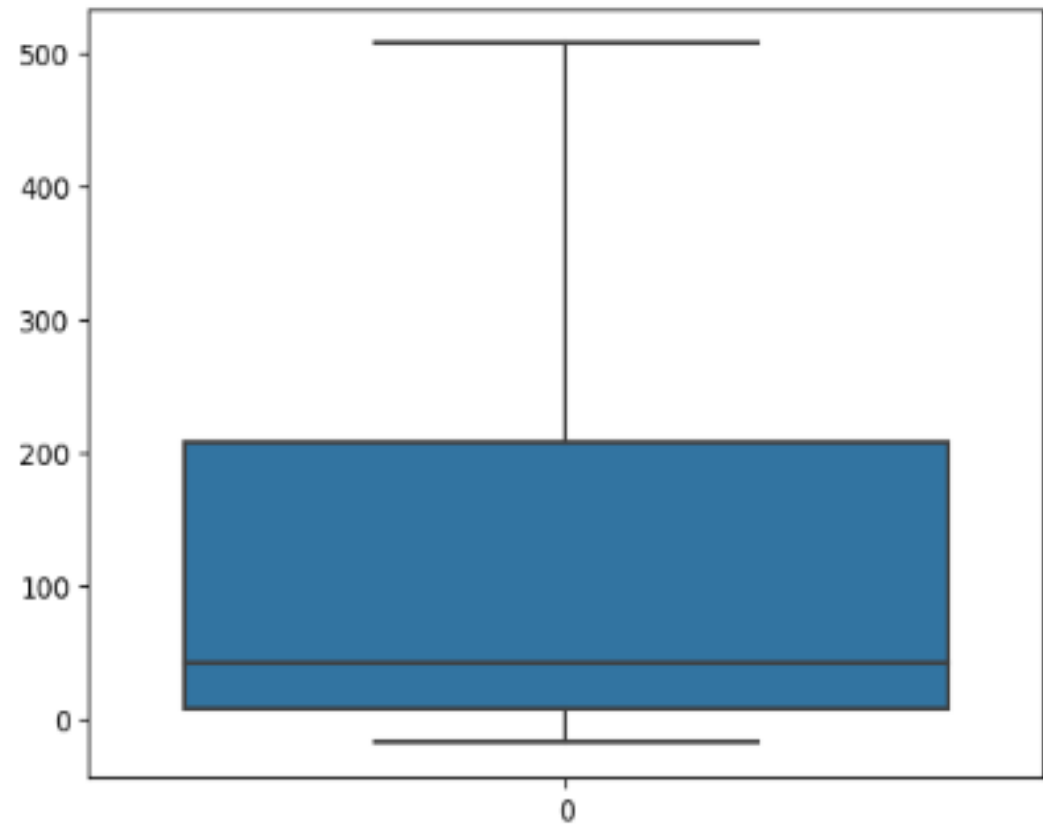
```
In [89]: len(df[df.votes<LL])
```

```
Out[89]: 0
```

```
In [90]: df.votes=np.where(df.votes>UL,UL,df.votes)
```

```
In [91]: sns.boxplot(df.votes)
```

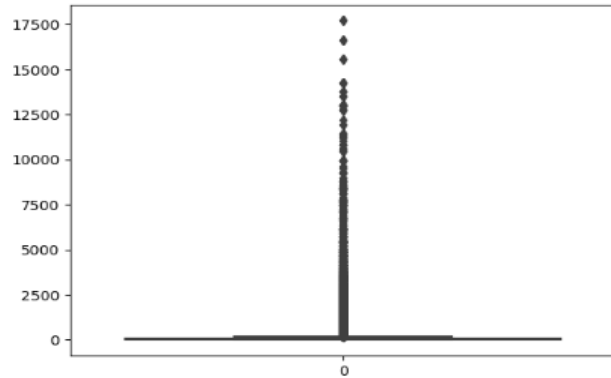
```
Out[91]: <Axes: >
```



Outliers in Photo Count

```
In [92]: sns.boxplot(df.photo_count)
```

```
Out[92]: <Axes: >
```



```
In [93]: Q1,Q3=df.photo_count.quantile([0.25,0.75])
```

```
In [94]: Q1,Q3
```

```
Out[94]: (1.0, 82.0)
```

```
In [95]: IQR=Q3-Q1
```

```
In [96]: IQR
```

```
Out[96]: 81.0
```

```
In [97]: UL=Q3+1.5*IQR  
LL=Q1-1.5*IQR
```

```
In [98]: UL,LL
```

```
Out[98]: (203.5, -120.5)
```

```
In [99]: len(df[df.photo_count>UL])
```

```
Out[99]: 9523
```

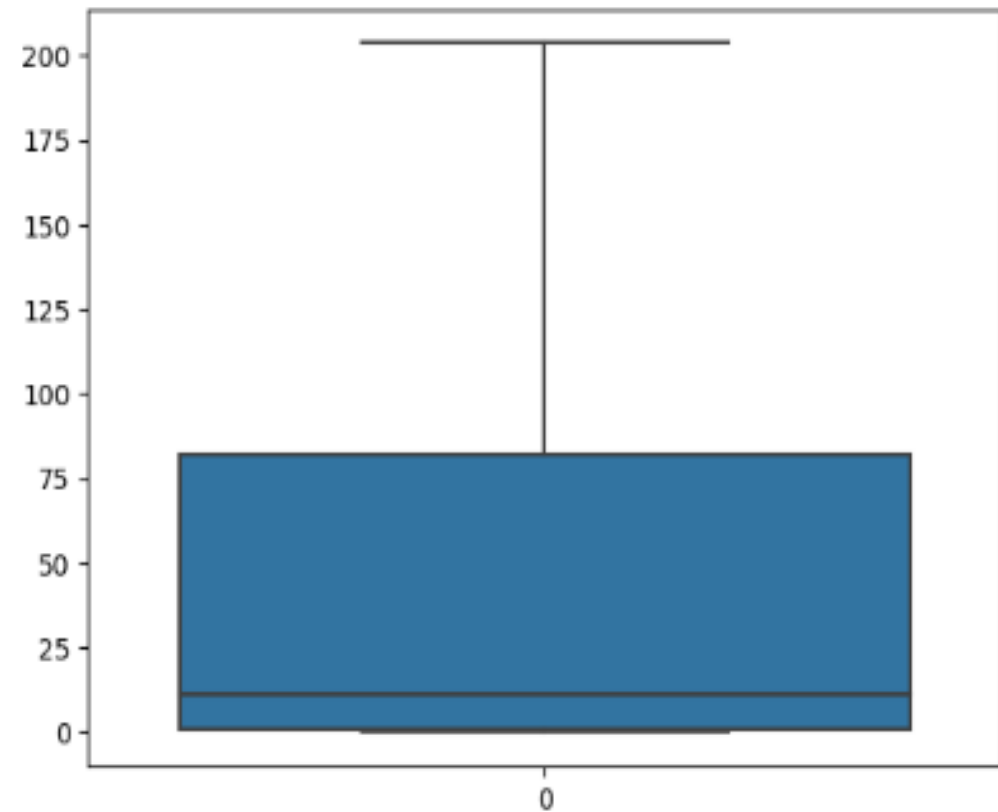
```
In [100]: len(df[df.photo_count<LL])
```

```
Out[100]: 0
```

```
In [101]: df.photo_count=np.where(df.photo_count>UL,UL,df.photo_count)
```

```
In [102]: sns.boxplot(df.photo_count)
```

```
Out[102]: <Axes: >
```

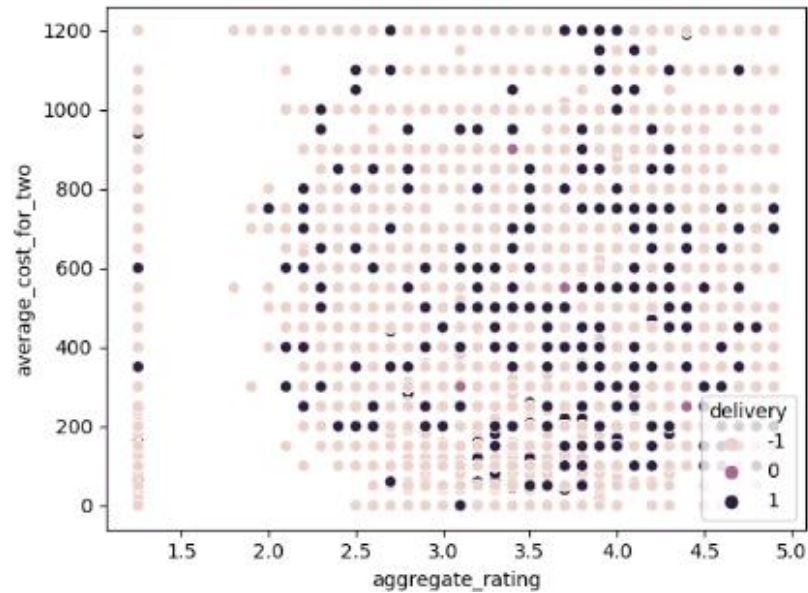


Scatter Plot

CODE:-

```
sns.scatterplot(x=df.aggregate_rating,y=df.average_cost_for_two,hue=df.delivery)
```

```
Out[103]: <Axes: xlabel='aggregate_rating', ylabel='average_cost_for_two'>
```

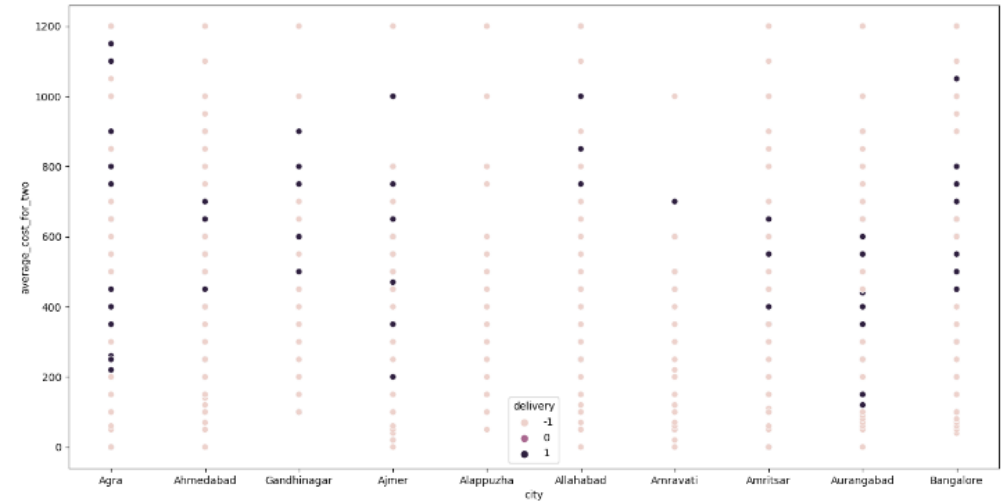


CODE:-

```
plt.figure(figsize=(16,8))
```

```
sns.scatterplot(x=df.city.head(6000),y=df.average_cost_for_two.head(6000),hue=df.delivery.y.head(6000))
```

```
Out[104]: <Axes: xlabel='city', ylabel='average_cost_for_two'>
```



Word Cloud

First install word cloud

```
In [105]: pip install WordCloud
```

```
Requirement already satisfied: WordCloud in c:\users\kfaiz\anaconda3\lib\site-packages (1.9.3)
Requirement already satisfied: numpy>=1.6.1 in c:\users\kfaiz\anaconda3\lib\site-packages (from WordCloud) (1.24.3)
Requirement already satisfied: pillow in c:\users\kfaiz\anaconda3\lib\site-packages (from WordCloud) (9.4.0)
Requirement already satisfied: matplotlib in c:\users\kfaiz\anaconda3\lib\site-packages (from WordCloud) (3.7.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\kfaiz\anaconda3\lib\site-packages (from matplotlib->WordCloud) (1.0.5)
Requirement already satisfied: cycler>=0.10 in c:\users\kfaiz\anaconda3\lib\site-packages (from matplotlib->WordCloud) (0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\kfaiz\anaconda3\lib\site-packages (from matplotlib->WordCloud) (4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\kfaiz\anaconda3\lib\site-packages (from matplotlib->WordCloud) (1.4.4)
Requirement already satisfied: packaging>=20.0 in c:\users\kfaiz\anaconda3\lib\site-packages (from matplotlib->WordCloud) (23.1)
Requirement already satisfied: pyparsing<3.1,>=2.3.1 in c:\users\kfaiz\anaconda3\lib\site-packages (from matplotlib->WordCloud) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\kfaiz\anaconda3\lib\site-packages (from matplotlib->WordCloud) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\kfaiz\anaconda3\lib\site-packages (from python-dateutil->matplotlib->WordCloud) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

CODE:-

```
In [106]: from wordcloud import WordCloud
```

```
In [107]: text = "ZOMATO"
wordcloud = WordCloud().generate(text)
```

```
In [108]: plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

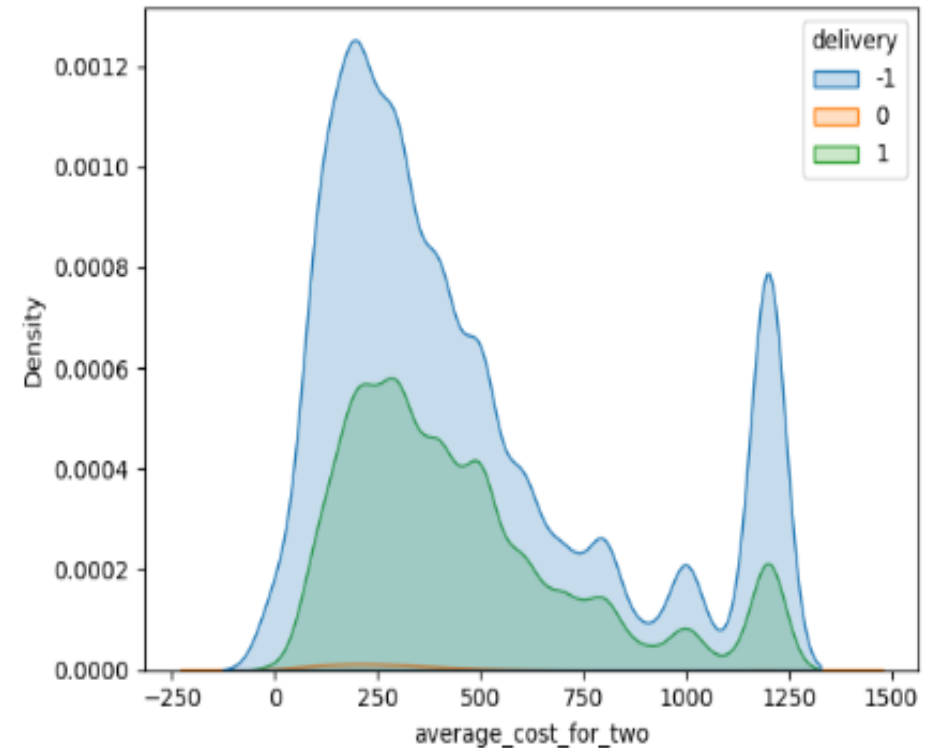


Density Plot

- The provided uses the Seaborn library's `kdeplot` function to create a kernel density estimate plot of the "average_cost_for_two" column of a DataFrame "df", with the "delivery" column used for hue and fill. The palette is set to "tab10".
- Here's a brief explanation of the arguments used:
- `x=df.average_cost_for_two`: This is the column to be plotted.
- `hue=df.delivery`: This argument is used to differentiate the plot between different categories of the "delivery" column.
- `fill=True`: This argument is used to fill the area under the curve of the kernel density estimate plot.
- `palette="tab10"`: This argument is used to set the color palette for the different categories of the "delivery" column.

```
In [109]: sns.kdeplot(x=df.average_cost_for_two, hue=df.delivery, fill=True, palette="tab10")
```

```
Out[109]: <Axes: xlabel='average_cost_for_two', ylabel='Density'>
```

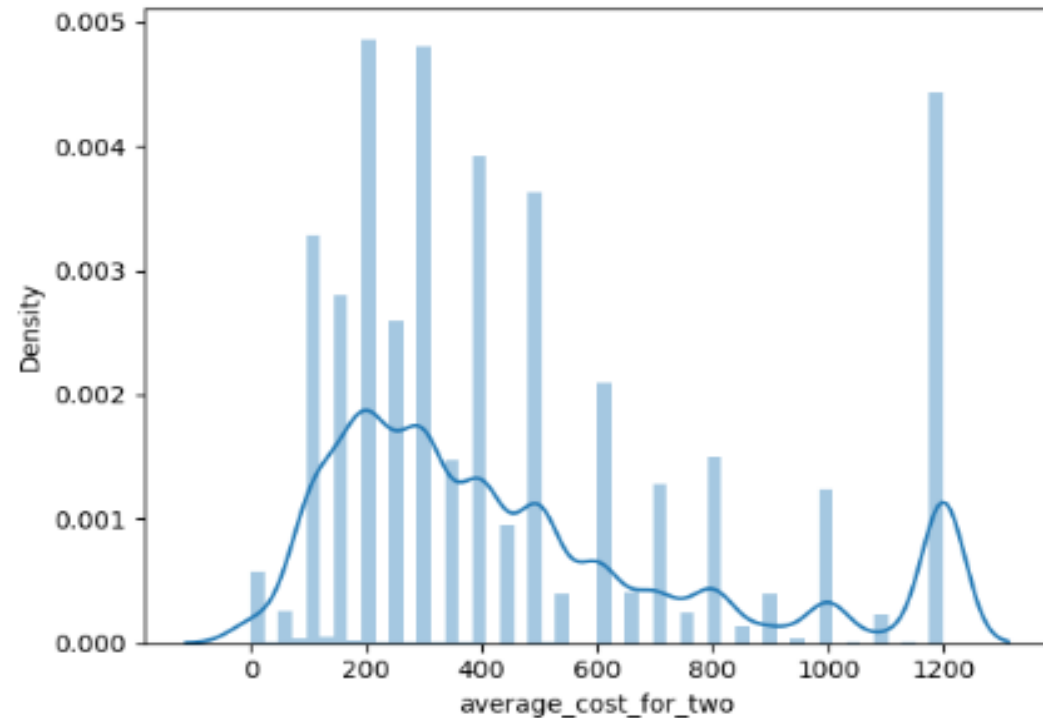


Dist Plot

- The `sns.distplot(df.average_cost_for_two)` function call is used to create a distribution plot (histogram) of the values in the "average_cost_for_two" column of the DataFrame `df` using the Seaborn library. This plot provides insights into the distribution of the data in that specific column, showing the frequency of different values.
- By executing this code, you will generate a histogram that visualizes the distribution of the "average_cost_for_two" values in your DataFrame `df`. This can help you understand the spread and frequency of these values, providing a quick overview of the data distribution.

```
In [110]: sns.distplot(df.average_cost_for_two)
```

```
Out[110]: <Axes: xlabel='average_cost_for_two', ylabel='Density'>
```

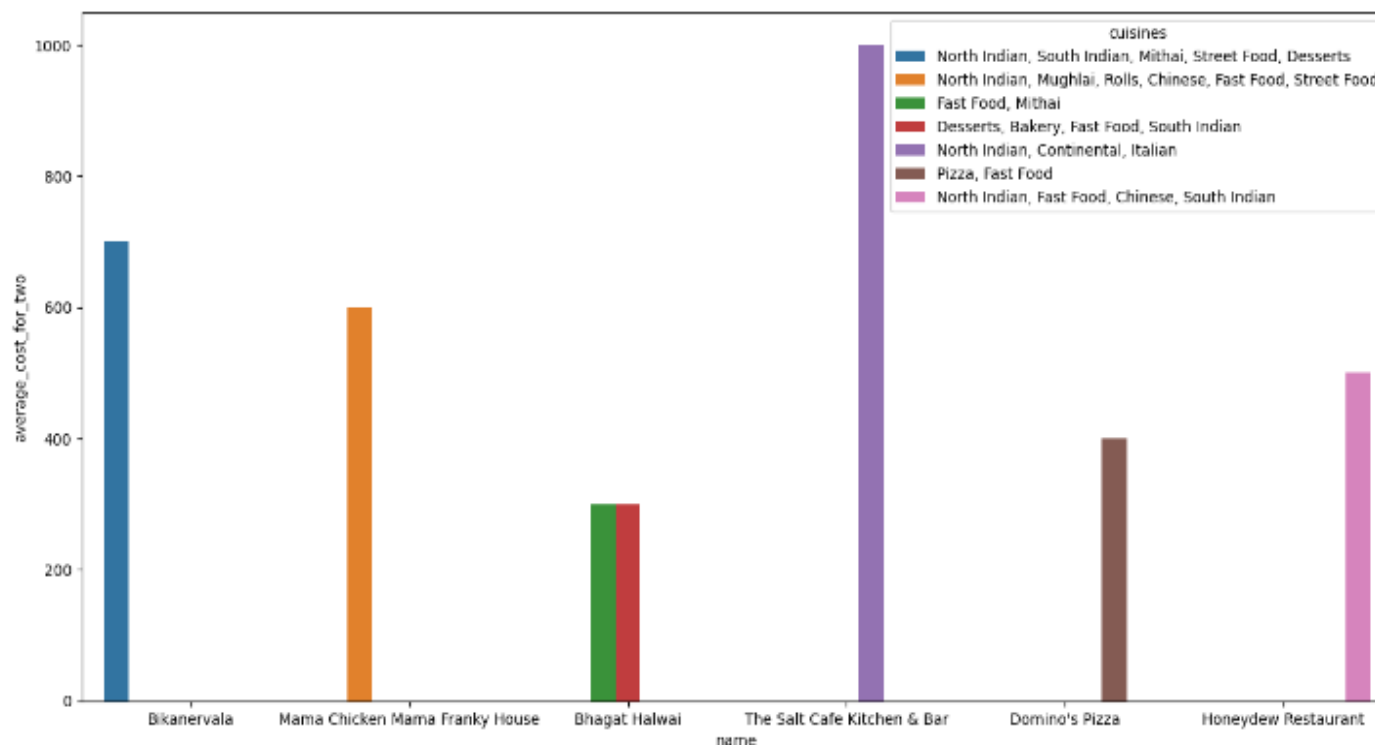


Bar Plot

- The provided code creates a bar plot using the Seaborn library's barplot function to visualize the average cost for two people in the first eight restaurants (based on the index) in the DataFrame df. The x-axis represents the names of the restaurants, and the y-axis shows the average cost for two people. The hue argument is used to differentiate the bars based on the cuisine of the restaurants.
- Here's a brief explanation of the arguments used:
- `x=df.name.head(8)`: This is the x-axis variable, which represents the names of the first eight restaurants in the DataFrame df.
- `y=df.average_cost_for_two.head(8)`: This is the y-axis variable, which represents the average cost for two people in the first eight restaurants in the DataFrame df.
- `hue=df.cuisines.head(8)`: This argument is used to differentiate the bars based on the cuisine of the first eight restaurants in the DataFrame df.
- Additionally, the `plt.figure(figsize=(16,8))` line sets the size of the plot to 16 units wide and 8 units tall, which provides a larger and clearer visualization of the data.

BAR CHART

```
In [111]: plt.figure(figsize=(16,8))  
  
sns.barplot(x=df.name.head(8),y=df.average_cost_for_two.head(8),hue=df.cuisines.head(8))  
  
Out[111]: <Axes: xlabel='name', ylabel='average_cost_for_two'>
```



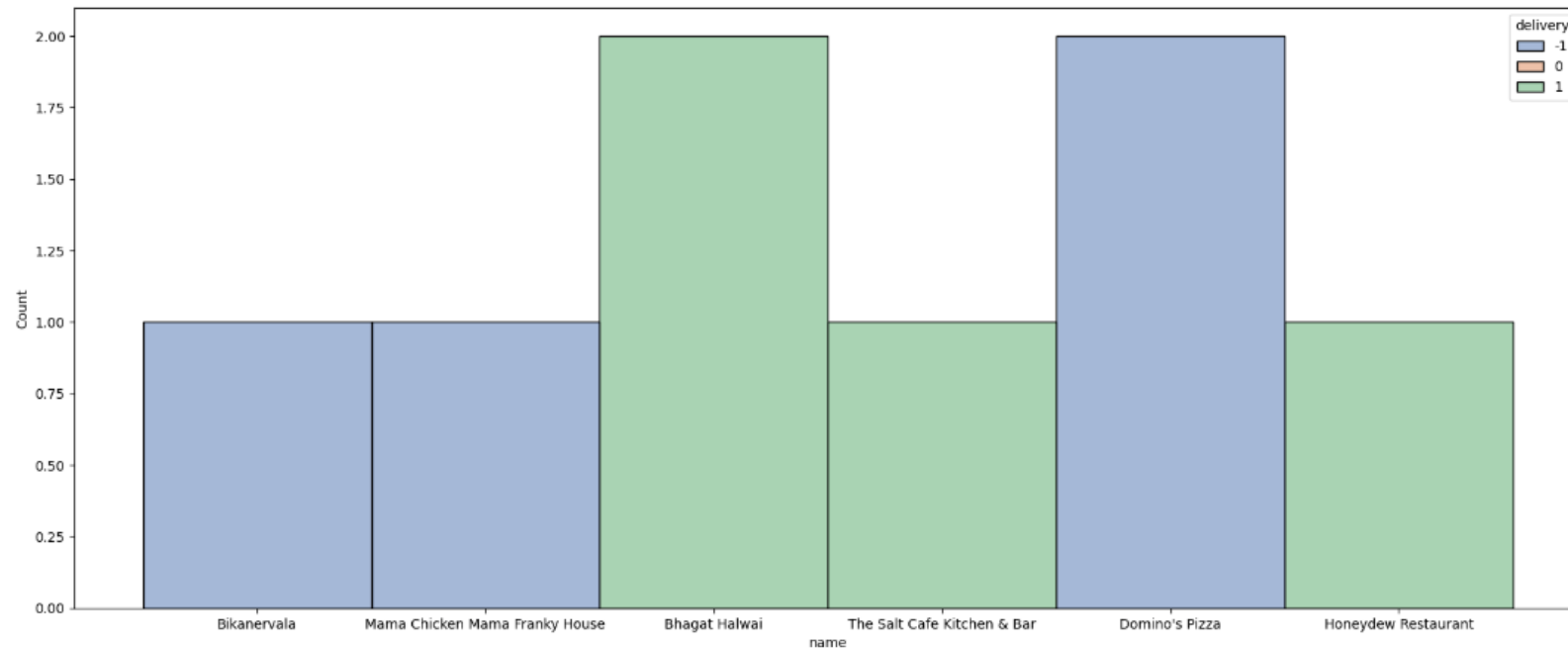
Bar Plot

- The provided code creates a histogram using the Seaborn library's histplot function to visualize the number of occurrences of the names of the first eight restaurants (based on the index) in the DataFrame df. The x-axis represents the names of the restaurants, and the y-axis shows the count of each restaurant name. The hue argument is used to differentiate the histogram based on the delivery method of the restaurants, and the palette argument is used to set the color palette for the different delivery methods.
- Here's a brief explanation of the arguments used:
- `x=df.name.head(8)`: This is the variable to be plotted, which represents the names of the first eight restaurants in the DataFrame df.
- `hue=df.delivery`: This argument is used to differentiate the histogram based on the delivery method of the restaurants.
- `palette="deep"`: This argument is used to set the color palette for the different delivery methods.
- Additionally, the `plt.figure(figsize=(20,8))` line sets the size of the plot to 20 units wide and 8 units tall, which provides a larger and clearer visualization of the data.

In [112]:

```
plt.figure(figsize=(20,8))
sns.histplot(x=df.name.head(8),hue=df.delivery,palette="deep")
```

Out[112]: <Axes: xlabel='name', ylabel='Count'>

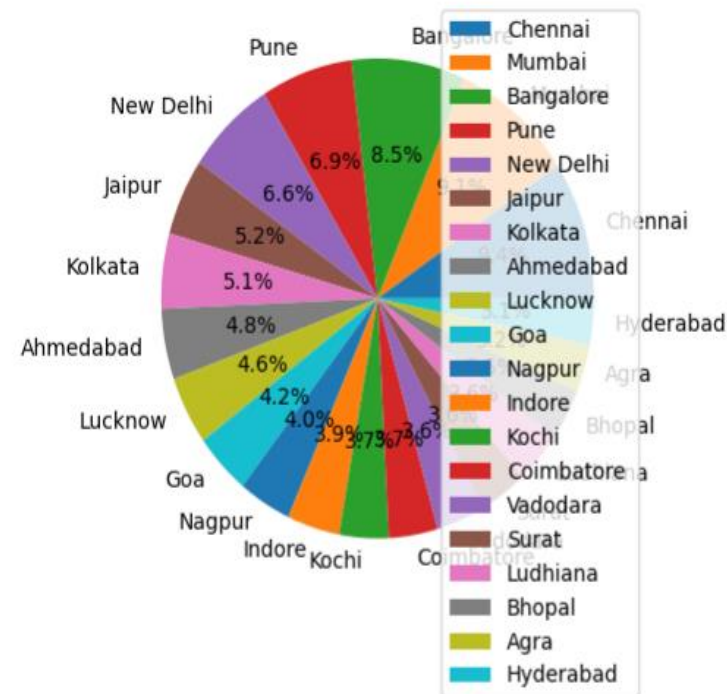


Pie Chart

- The provided code uses Matplotlib to create a pie chart based on the value counts of the "city" column in the DataFrame df. It displays the distribution of the top 20 cities by count in the dataset. The autopct="%1.1f%%" argument formats the percentage shown on each wedge of the pie chart with one decimal place.
- By adding the plt.legend() function with the appropriate parameters, you can include a legend in the pie chart to indicate the cities represented by each slice. This will enhance the readability and understanding of the chart.

```
In [113]: plt.pie(df.city.value_counts().head(20), labels=df.city.value_counts().head(20).index, autopct="%1.1f%%")  
plt.legend()
```

```
Out[113]: <matplotlib.legend.Legend at 0x25c2441dfd0>
```



Chennai has most number of resturants

```
In [129]: df.aggregate_rating.median()
```

```
Out[129]: 3.5
```

```
In [130]: average_rating=df[df.aggregate_rating==3.5]
```

```
In [131]: average_rating
```

```
Out[131]:
```

	res_id	name	establishment	url	address	city	city_id	locality	latitude	longitude	...	p
15	3400811	Domino's Pizza	[Quick Bites]	https://www.zomato.com/agra/dominos-pizza-kamla...	Ground & First Floor, City Mall, Balkeshwar Co...	Agra	34	Kamla Nagar	27.212943	78.026549	...	
77	3400385	Sattolala Food Court	[Quick Bites]	https://www.zomato.com/agra/sattolala-food-cou...	6, M.I.G., New Shahganj Rd, Kothi Meena Bazaar...	Agra	34	Shahganj	27.178337	77.989000	...	
79	3400029	Capri Restaurant	[Casual Dining]	https://www.zomato.com/agra/capri-restaurant-c...	Shop no.6-7, upadhyay market, hari parvat, agra, u...	Agra	34	Civil Lines	27.196596	78.000893	...	
127	3400171	Rendezvous - Hotel Howard Plaza	[Fine Dining]	https://www.zomato.com/agra/rendezvous-hotel-h...	Hotel Howard Plaza, Fatehabad Road, Tajganj, Agra	Agra	34	Hotel Howard Plaza, Tajganj	27.162926	78.036835	...	
225	3401672	Laxman Momo's Corner	[Quick Bites]	https://www.zomato.com/agra/laxman-momos-corne...	Rajpur Chungi, Tajganj, Agra	Agra	34	Tajganj	27.150113	78.035584	...	
...
211075	18754495	Shaan-E-Punjab	[Quick Bites]	https://www.zomato.com/vadodara/shaan-e-punjab...	C 20, Dayal Nagar Society, Beside Modi Medical...	Vadodara	32	Karelibaug	22.310883	73.224911	...	
211097	3202271	The Square Meal-Ginger Hotel RCR	[Casual Dining]	https://www.zomato.com/vadodara/the-square-mea...	Ginger Hotel, Near Natubhai Circle, Race Cours...	Vadodara	32	Vadiwadi	22.308823	73.159515	...	
211705	18878732	Vijay Panjabi Khana	[Quick Bites]	https://www.zomato.com/vadodara/vijay-panjabi-...	GF/5, Almiya Heights, Maneja Crossing, Makarpur...	Vadodara	32	Makarpura	22.236352	73.188811	...	
211724	3201287	Cafe 3' Trio	[Café]	https://www.zomato.com/vadodara/cafe-3-trio-ke...	103, Vraj Residency, Above A-One Sport Saloon,...	Vadodara	32	Kendranagar	22.286610	73.233724	...	
211730	3201920	Chas Ne Lari	[Quick Bites]	https://www.zomato.com/vadodara/chas-ne-lari-a...	Opposite Rajkot Nagrik Sahkari Bank, Back side...	Vadodara	32	Alkapuri	22.315174	73.170885	...	

2970 rows x 27 columns

Basic Statistics with Data

Calculate and visualize the average rating of restaurants


```
In [132]: average_rating.groupby(["name","city","cuisines"])["average_cost_for_two"].sum().reset_index()
```

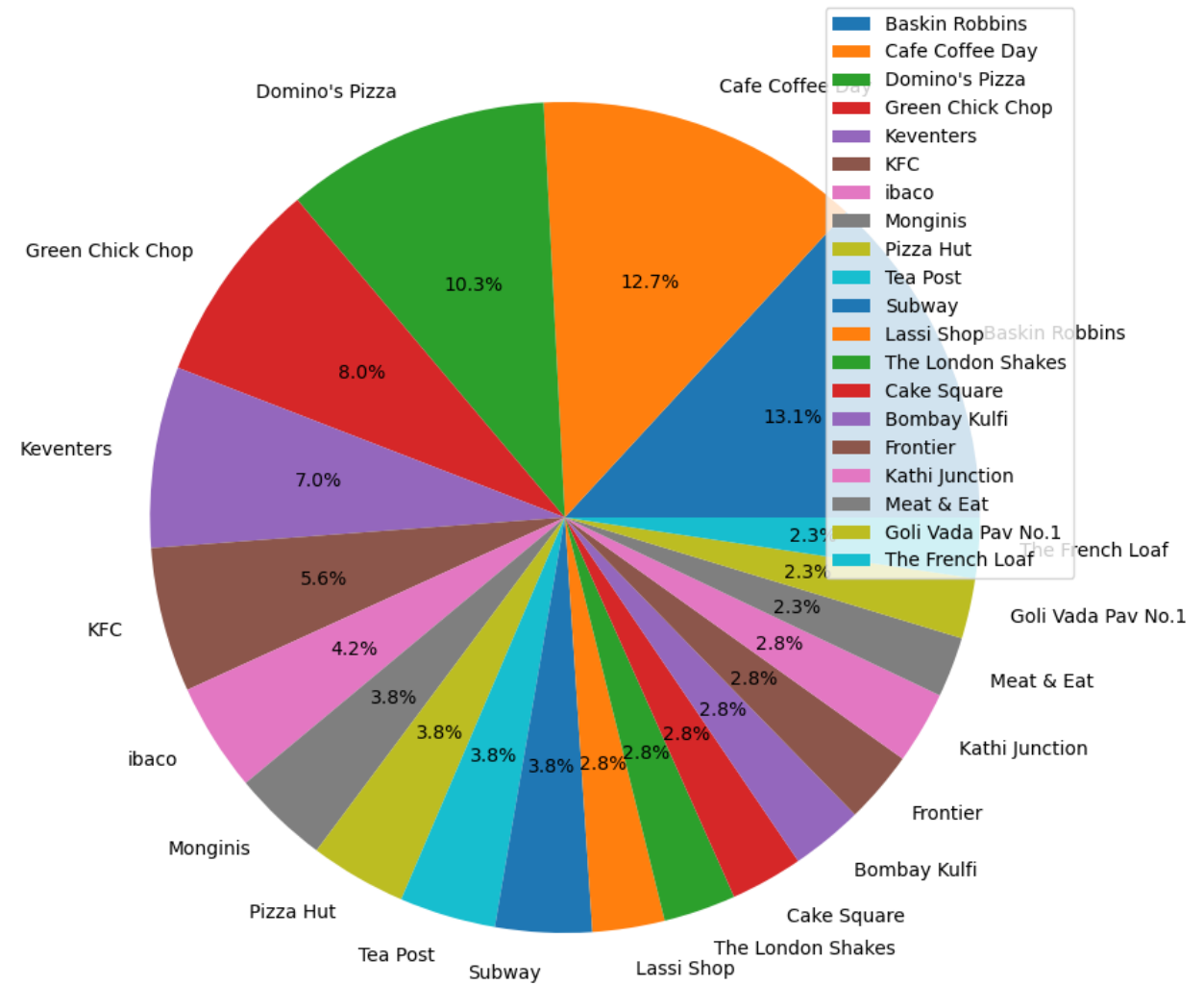
Out[132]:

	name	city	cuisines	average_cost_for_two
0	# Wednesday	Kanpur	Fast Food, Chinese, North Indian	250.0
1	009 Restaurant	Varanasi	Fast Food, North Indian, Chinese	500.0
2	14 Empire	Patna	North Indian, Chinese, Fast Food	600.0
3	17th Avenue Cafe	Goa	Cafe	250.0
4	18 Degree Icecream Zone	Trichy	Ice Cream, Desserts, Juices	150.0
...
2653	Zuma - Hampton by Hilton	Vadodara	Thai, North Indian, Chinese, Afghan, American	1200.0
2654	ibaco	Chennai	Ice Cream, Desserts	1400.0
2655	ibaco	Palakkad	Desserts, Ice Cream	600.0
2656	lazeezo shawarna	Kolhapur	Kebab, Turkish	200.0
2657	pizza zone	Jamnagar	American	250.0

2658 rows × 4 columns

- **CODE:-**

- `plt.figure(figsize=(10,10))`
- `plt.pie(average_rating.name.value_counts().head(20),labels=average_rating.name.value_counts().head(20).index,autopct="%1.1f%%")`
- `plt.legend()`

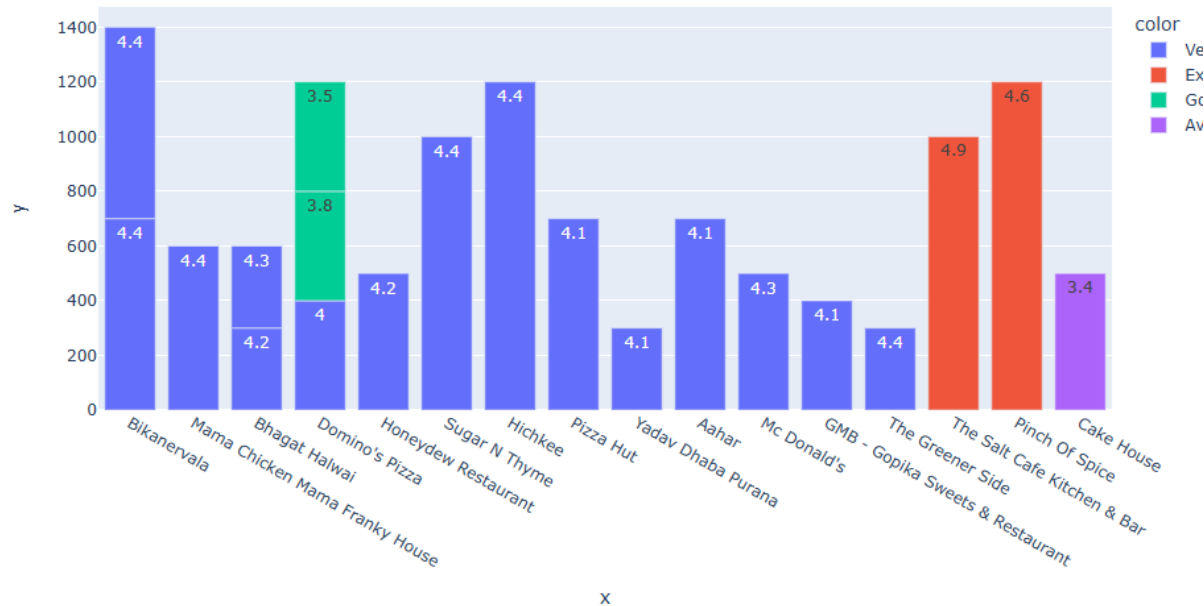


Baskin Robbins has highest ratings

Distribution of restaurant ratings to understand the overall rating

57]:

```
px.bar(x=df.name.head(20),y=df.average_cost_for_two.head(20),color=df.rating_text.head(20),text=df.aggregate_rating.head(20))
```



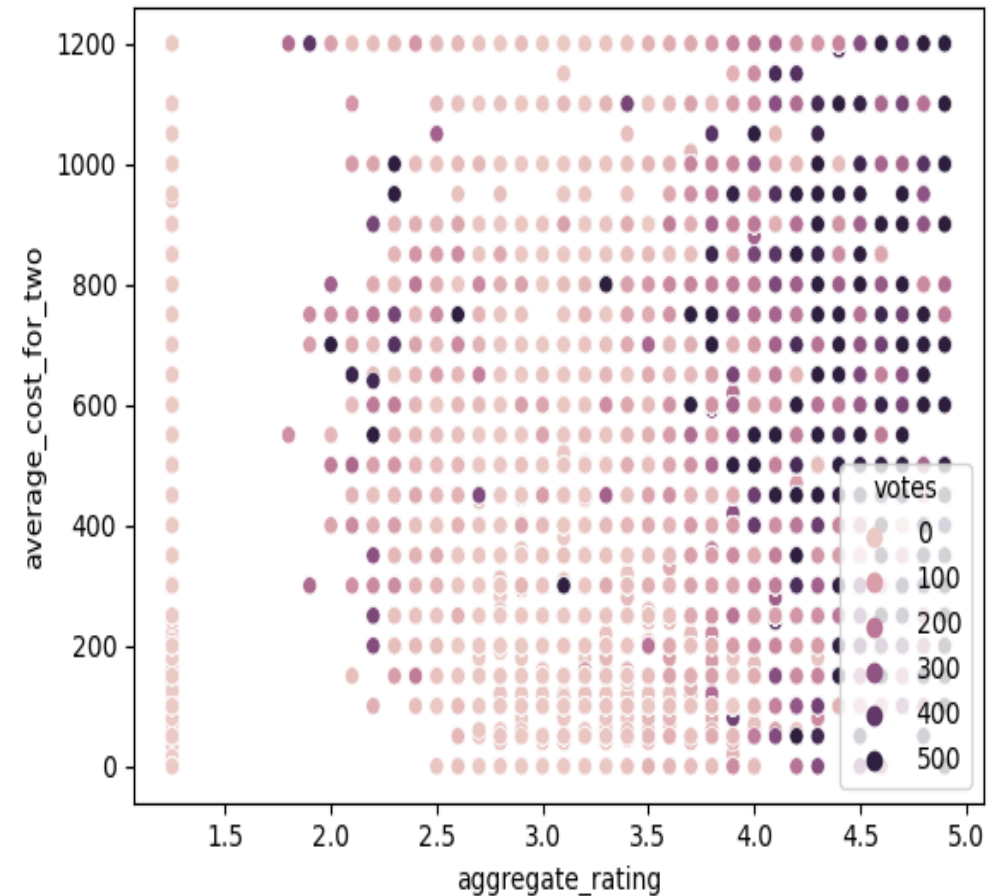
- The provided code seems to be using Plotly Express to create a bar chart. It visualizes the average cost for two people, with the names of the first 20 restaurants on the x-axis, the average cost on the y-axis, and the rating text as the color of the bars. Additionally, the aggregate rating is displayed as text on the bars.

CODE:-

```
sns.scatterplot(x=df.aggregate_rating,y=df.average_cost_for_two,hue=df.votes)
```

- The Seaborn library is a popular data visualization library in Python, and the scatterplot function is a widely used tool for visualizing statistical relationships between two variables. In your code, `sns.scatterplot(x=df.aggregate_rating, y=df.average_cost_for_two, hue=df.votes)`, you are using the scatterplot function to visualize the relationship between `aggregate_rating` and `average_cost_for_two`, with the `hue` parameter set to `df.votes` to differentiate the points based on the votes variable.
- The scatterplot function is particularly useful for understanding the joint distribution of two variables, as it depicts the relationship between them using a cloud of points, where each point represents an observation in the dataset. This allows the eye to infer a substantial amount of information about whether there is any meaningful relationship between the variables.
- The `hue` parameter is a powerful feature of the scatterplot function, as it allows you to map a third variable to the color of the points, adding another dimension to the plot. This can be particularly useful for understanding how the relationship between two variables is influenced by a third variable.
- In your code, the `hue` parameter is set to `df.votes`, which means that the points will be colored based on the votes variable. This will allow you to see if there are any patterns or trends in the relationship between `aggregate_rating` and `average_cost_for_two` that are influenced by the votes variable.

```
Out[136]: <Axes: xlabel='aggregate_rating', ylabel='average_cost_for_two'>
```



Location Analysis

```
In [160]: x=df.groupby(["city"])["name"].nunique().sort_values(ascending=False)
```

```
In [161]: x
```

```
Out[161]: city
Bangalore      1504
Mumbai         1474
Pune           1462
New Delhi      1271
Chennai        1271
...
Udupi          59
Howrah         49
Neemrana       26
Greater Noida  20
Nayagaon       15
Name: name, Length: 99, dtype: int64
```

```
In [162]: a=pd.DataFrame(x).reset_index()
```

```
In [163]: a
```

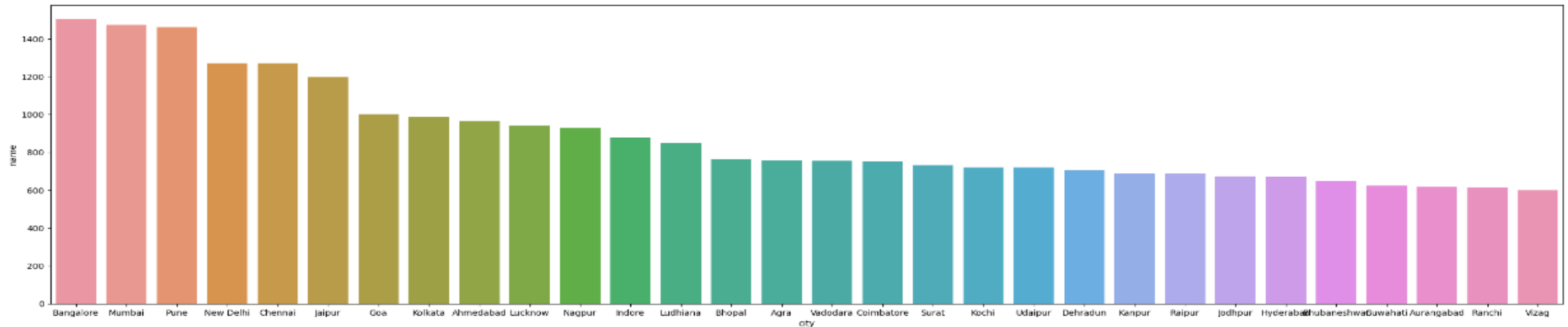
```
Out[163]:
```

	city	name
0	Bangalore	1504
1	Mumbai	1474
2	Pune	1462
3	New Delhi	1271
4	Chennai	1271
...
94	Udupi	59
95	Howrah	49
96	Neemrana	26
97	Greater Noida	20
98	Nayagaon	15

99 rows x 2 columns

```
In [164]: plt.figure(figsize=(30,7))  
sns.barplot(x=a.city.head(30),y=a.name.head(30))
```

```
Out[164]: <Axes: xlabel='city', ylabel='name'>
```



Bangalore has highest number of resturants

Ratings of restruratnts according to diffrent cities

```
In [145]: ratings.head(15)
```

Out[145]:

	name	city	aggregate_rating
0	# Wednesday	Kanpur	3.50
1	#1, Culinary Avenue - The Red Maple	Indore	3.90
2	#788 Avenue	Hyderabad	3.90
3	#BC	Jaipur	4.20
4	#BEiR	Bangalore	4.10
5	#Dilliwaala6	Puducherry	4.20
6	#Jaipur Darbar	Jaipur	3.40
7	#OFF Campus	New Delhi	4.00
8	#Selfieccino	Rajkot	3.30
9	#Underground Restro & Bar	Jaipur	3.20
10	#Waffles	Ghaziabad	7.90
11	#Waffles	New Delhi	7.90
12	#World of Waffles	Puducherry	1.25
13	#gachKhana	Hyderabad	2.60
14	#nofilter	New Delhi	3.90

```
In [143]: ratings=df.groupby(["name","city"])["aggregate_rating"].sum().reset_index()
```

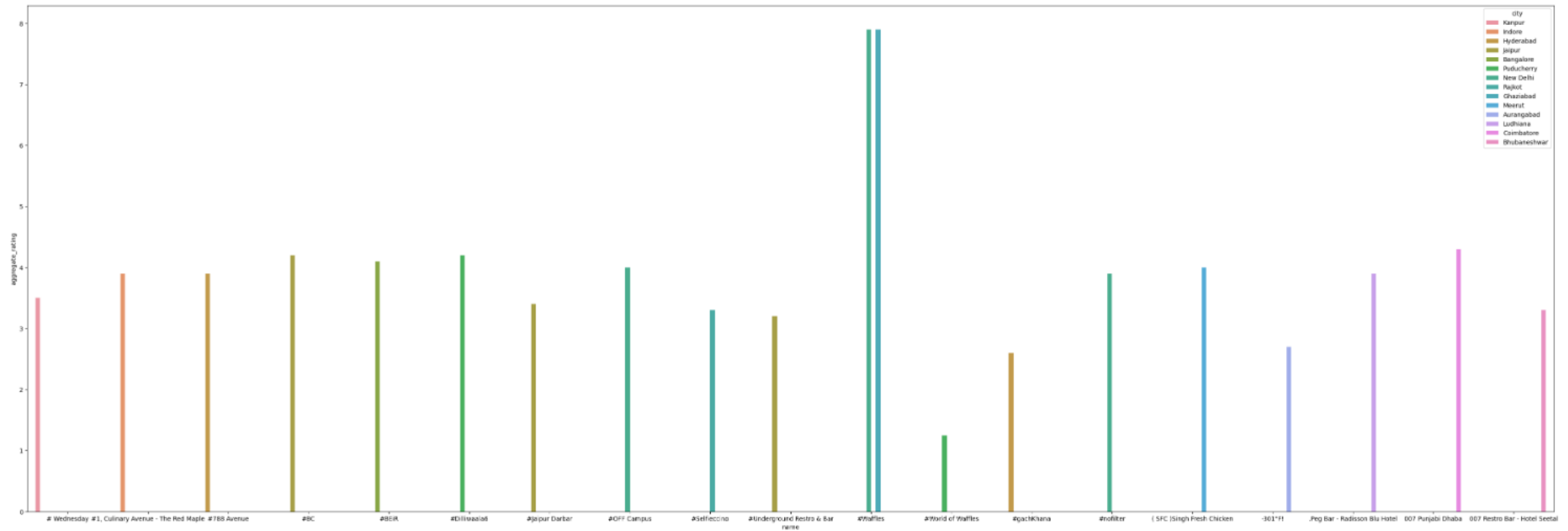
```
In [144]: ratings.max()
```

```
Out[144]: name          ह-tea The Tea Hut
city              Zirakpur
aggregate_rating      270.85
dtype: object
```

CODE:-

```
plt.figure(figsize=(45,15))
sns.barplot(x=ratings.name.head(20),y=ratings.aggregate_rating.head(20),hue=ratings.city.head(20))
```

<Axes: xlabel='name', ylabel='aggregate_rating'>



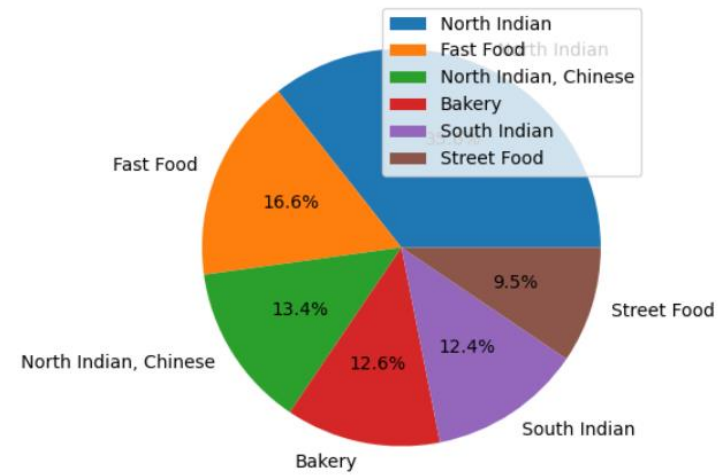
CUISINE ANALYSIS- The most popular cuisine under listed resturants

```
In [147]: df.cuisines.value_counts()
```

```
Out[147]: cuisines
North Indian      4587
Fast Food         2137
North Indian, Chinese 1720
Bakery            1618
South Indian      1600
...
North Indian, Fast Food, Healthy Food      1
Beverages, Healthy Food, Fast Food         1
Fast Food, Street Food, South Indian, Bakery 1
Tibetan, Street Food                       1
North Indian, Lucknowi, Chinese             1
Name: count, Length: 9383, dtype: int64
```

```
In [148]: plt.figure(figsize=(10,5))
plt.pie(df.cuisines.value_counts().head(6),labels=df.cuisines.value_counts().head(6).index,autopct="%1.1f%%",
plt.legend()
```

```
Out[148]: <matplotlib.legend.Legend at 0x25c33e3c110>
```

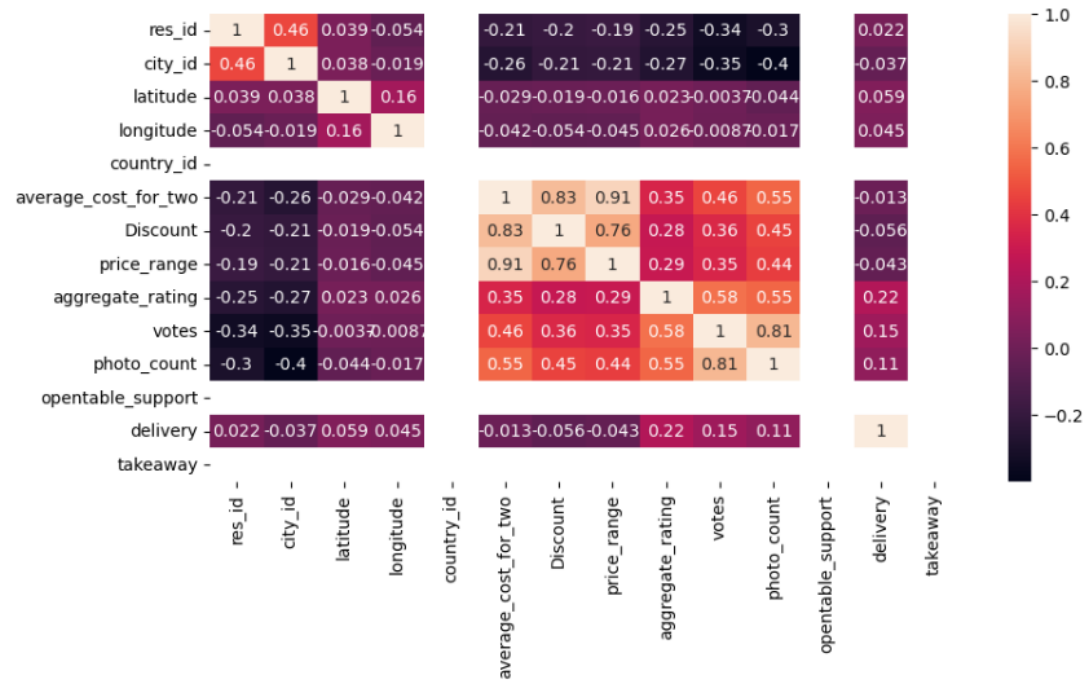


From the given pie chart we can clearly see that the North Indian cuisine is most popular

Relationship between Price , Range and Ratings

```
In [150]: plt.figure(figsize=(10,5))  
sns.heatmap(df.corr(numeric_only=True),annot=True)
```

Out[150]: <Axes: >

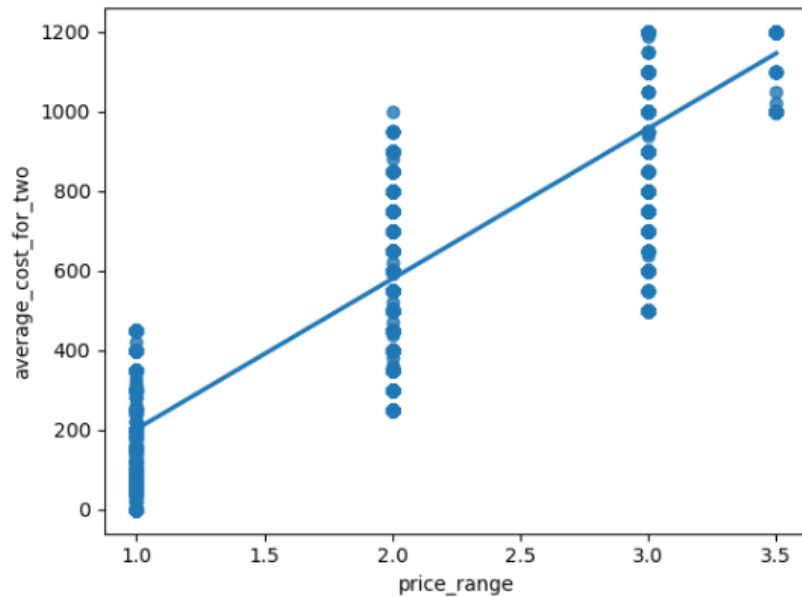


- The provided code creates a heatmap of the correlation matrix for the DataFrame df, using the Seaborn library's heatmap function. The figsize parameter is used to set the size of the plot, while the corr function is used to calculate the correlation matrix for the numeric columns in the DataFrame. The annot=True parameter is used to display the correlation coefficients in each cell of the heatmap.

Visualization of the average cost for two people in different price categories.

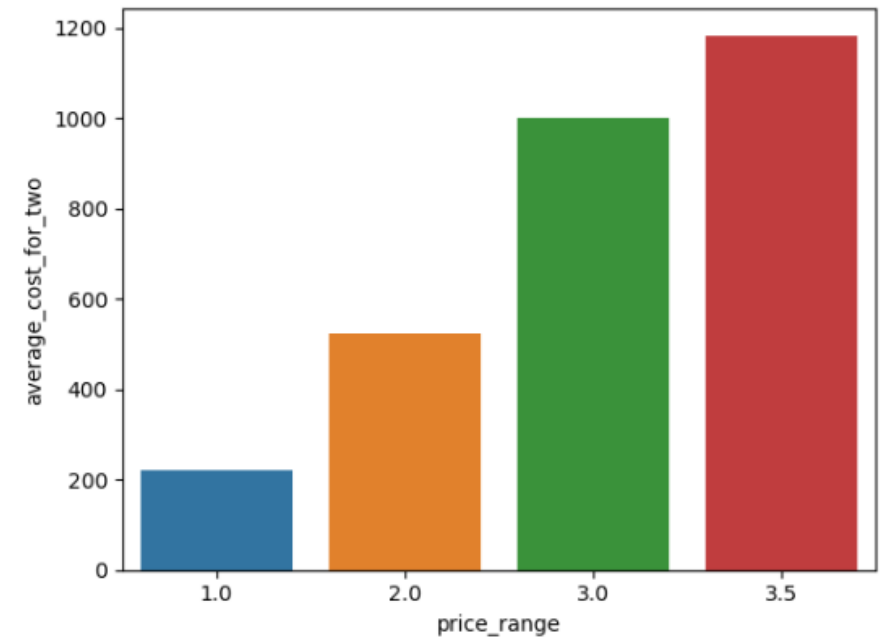
```
In [155]: sns.regplot(x=df.price_range,y=df.average_cost_for_two)
```

```
Out[155]: <Axes: xlabel='price_range', ylabel='average_cost_for_two'>
```



```
In [167]: sns.barplot(x=df.price_range,y=df.average_cost_for_two,ci=True)
```

```
Out[167]: <Axes: xlabel='price_range', ylabel='average_cost_for_two'>
```



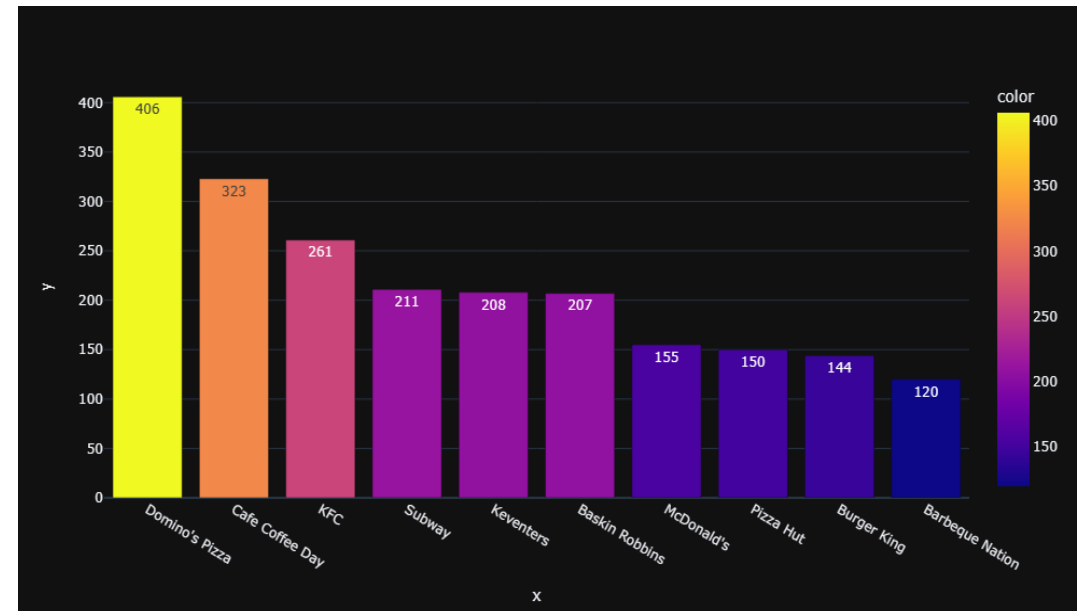
Top Restaurant Chains:

visualization the top restaurant chains based on the number of outlets and the ratings of top chains

```
In [139]: df.name.value_counts()
```

```
Out[139]: name
Domino's Pizza      406
Cafe Coffee Day     323
KFC                 261
Subway              211
Keventers           208
...
Jai Bhole ki Vaishnav Bhojnalaya      1
Shri Hari Snacks                      1
Greeno Restaurant                    1
The Dark Mustache ( Kathi Roll Barbeque Chicken ) 1
Geeta lodge                          1
Name: count, Length: 41100, dtype: int64
```

- **CODE:-**
`px.bar(x=df.name.value_counts().head(10).index,y=df.name.value_counts().head(10),text=df.name.value_counts().head(10),color=df.name.value_counts().head(10),template="plotly_dark")`



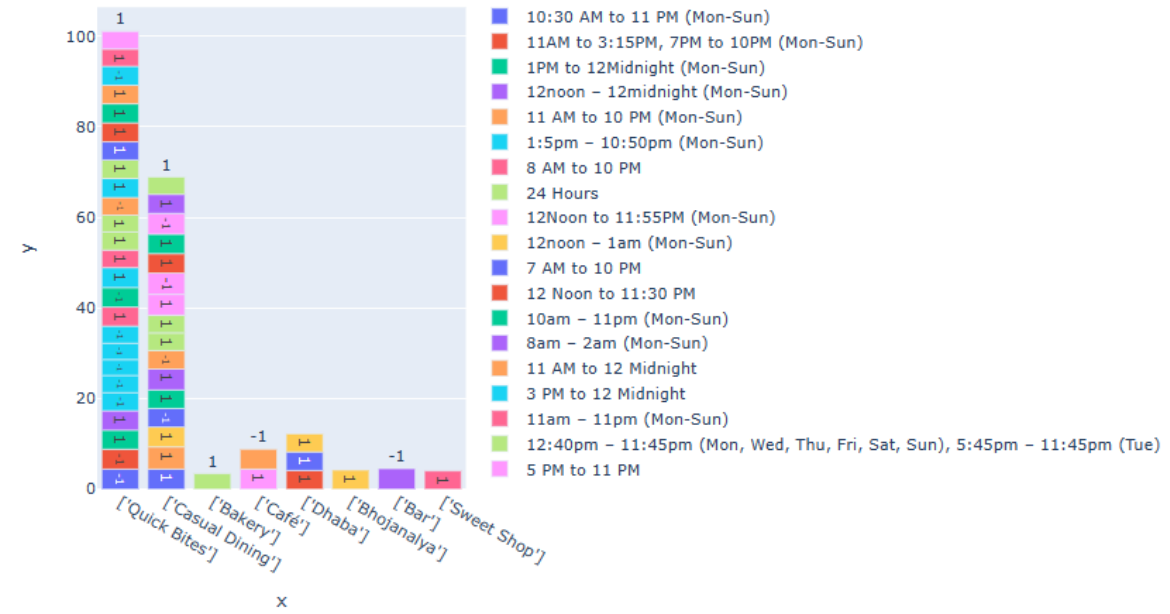
Finding Average Ratings

```
Out[139]: name
Villa Maya Heritage Restaurant      4.90
Zolocrust - Hotel Clarks Amer      4.90
Trio-La Patisserie                  4.90
Jumjoji - The Parsi Diner           4.90
White Charcoal - The Empresa Hotel  4.90
...
TAPRI - The Food Stall              1.25
Buon Appetito                      1.25
Burfi Kadaï                        1.25
Friend's Cafe                      1.25
Ruchiram Bhojnalaya                 1.25
Name: aggregate_rating, Length: 41100, dtype: float64
```

- **CODE:-**
`df.groupby("name")["aggregate_rating"].mean().sort_values(ascending=False)`

Investigate if the presence of certain features correlates with higher ratings. 1

```
In [142]: plt.figure(figsize=(25,5))
px.bar(x=df.establishment.head(50),y=df.aggregate_rating.head(50),text=df.delivery.head(50),color=df.timings.head(50))
```



<Figure size 2500x500 with 0 Axes>

- This code creates a bar chart using the Plotly Express library in Python. The chart will have 50 bars, with the x-axis representing the names of the top 50 establishments from the DataFrame df, and the y-axis representing the aggregate rating of each establishment. The text displayed above each bar will be the delivery status of the corresponding establishment, and the color of each bar will be determined by the timings of the establishment.
- To create the chart, the code first calls the plt.figure function from the matplotlib library to create a new figure with a specified size. Then, the px.bar function from the Plotly Express library is called to create the bar chart. The x parameter is set to the establishment column of the first 50 rows of the DataFrame df, the y parameter is set to the aggregate_rating column of the first 50 rows of df, the text parameter is set to the delivery column of the first 50 rows of df, and the color parameter is set to the timings column of the first 50 rows of df.

Seasonal Trends

Visualization of the distribution of ratings during different times of the year

```
In [143]: df.groupby("timings")["aggregate_rating"].sum()
```

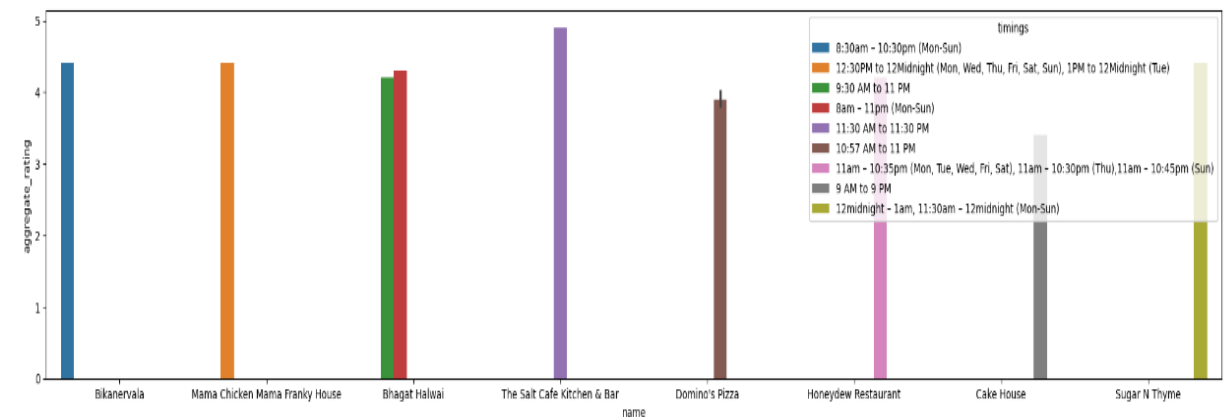
```
Out[143]: timings
(Mon),12noon - 12midnight (Tue-Sun)      3.40
(Mon, Tue, Fri), 8am - 12midnight (Wed, Sat, Sun), 8:40am - 11:55pm (Thu)  7.40
(Mon, Tue, Thu, Fri, Sat, Sun), 12:5AM - 12midnight (Wed)  7.80
(Mon-Sat),11am - 12:5pm (Sun)      1.25
(Mon-Sun)      35.45
...
Mon-Tue, Thu-Sun (11 AM to 11:59 PM), Wed (11 AM to 11:55 PM)  7.40
Mon-Tue, Thu-Sun (Closed), Wed (8 AM to 8:15 AM)  3.80
Mon-Wed (11 AM to 10 PM), Thu-Sun (Closed)  3.10
Mon-Wed, Fri-Sun (11 AM to 11 PM), Thu (Closed)  2.60
a      2537.65
Name: aggregate_rating, Length: 7741, dtype: float64
```

CODE:-

```
plt.figure(figsize=(25,5))
```

```
sns.barplot(x=df.name.head(10),y=df.aggregate_rating.head(10),hue=df.timings.head(10))
```

<Axes: xlabel='name', ylabel='aggregate_rating'>

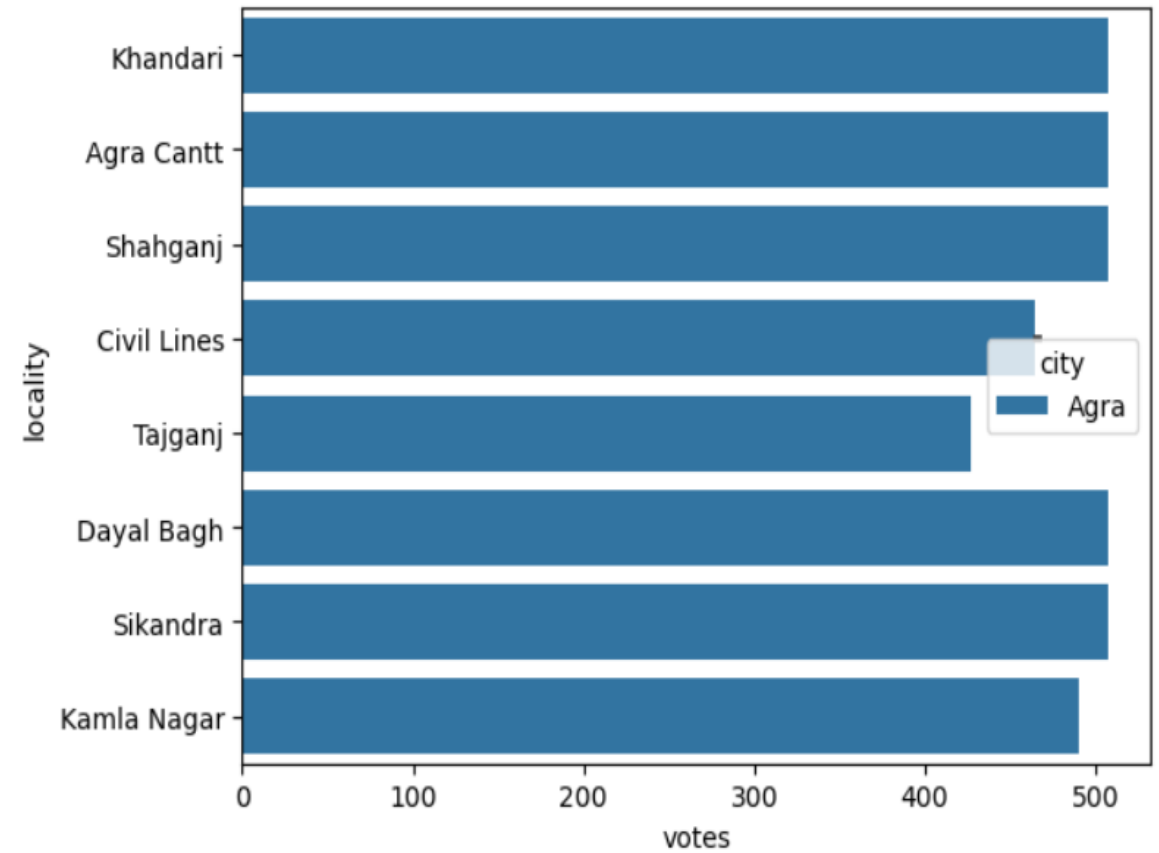


Votes of the localities

- This code creates a bar chart using the Seaborn library in Python. The chart will have 20 bars, with the x-axis representing the number of votes of the top 20 localities from the DataFrame df, and the y-axis representing the name of the localities. The color of the bars will be determined by the city of the corresponding locality.
- To create the chart, the code first calls the plt.figure function from the matplotlib library to create a new figure with a specified size. Then, the sns.barplot function from the Seaborn library is called to create the bar chart. The x parameter is set to the votes column of the first 20 rows of the DataFrame df, the y parameter is set to the locality column of the first 20 rows of df, and the hue parameter is set to the city column of the first 20 rows of df. The ci parameter is set to True, which means that the confidence intervals will be displayed on the chart.

```
plt.figure(figsize=(25,7))  
sns.barplot(x=df.votes.head(20),y=df.locality.head(20),hue=df.city.head(20),ci=True)
```

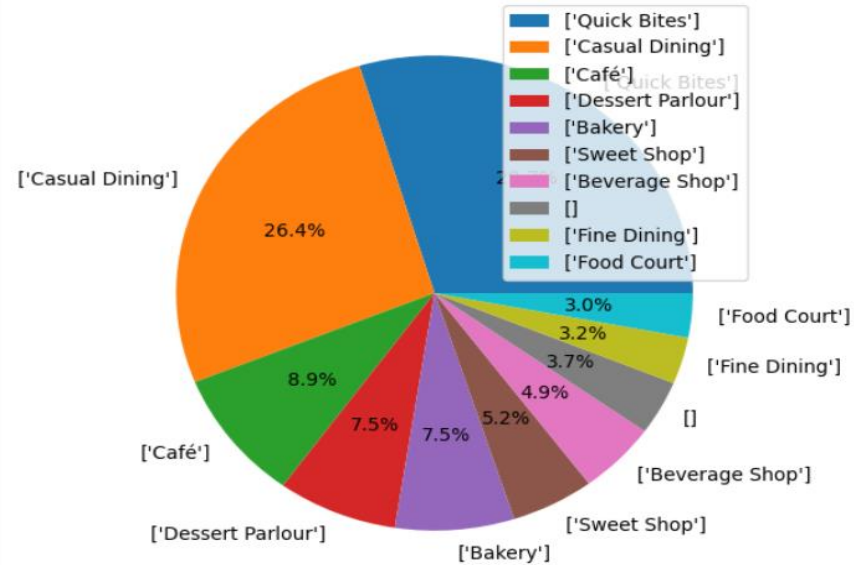
<Axes: xlabel='votes', ylabel='locality'>



Top Establishments

```
In [146]: plt.figure(figsize=(12,6))
plt.pie(df.establishment.value_counts().head(10),labels=df.establishment.value_counts().head(10).index,autopct="%1.1f%%")
plt.legend()
```

```
Out[146]: <matplotlib.legend.Legend at 0x1c3be5a6550>
```

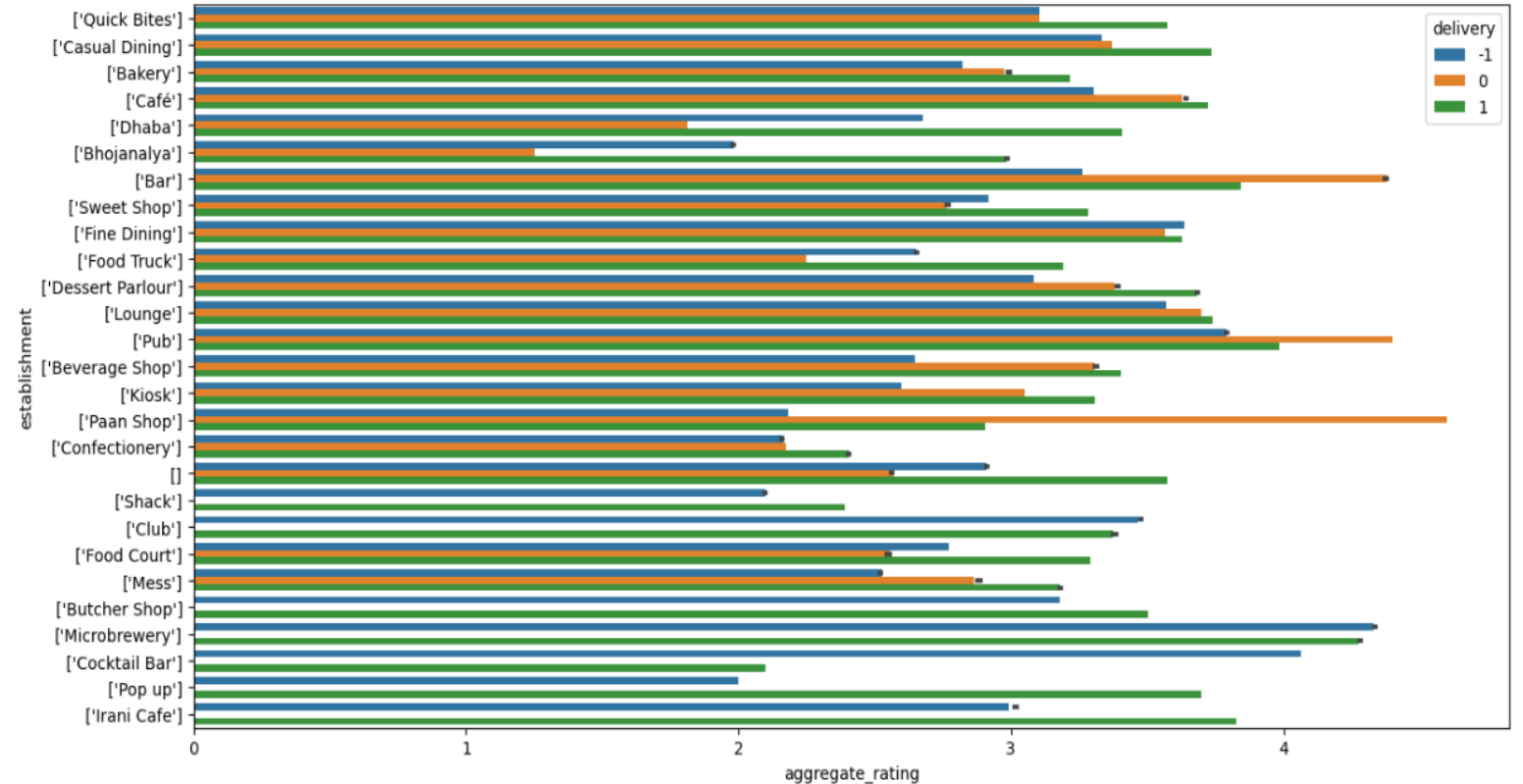


Top Establishment= Quick Bites

- This code creates a pie chart using the Matplotlib library in Python. The chart will show the top 10 establishments from the DataFrame df and their corresponding counts as a percentage of the total.
- To create the chart, the code first calls the plt.figure function from the Matplotlib library to create a new figure with a specified size. Then, the plt.pie function is called to create the pie chart. The x parameter is set to the top 10 establishments and their corresponding counts from the DataFrame df using the value_counts() method. The labels parameter is set to the same values as the x parameter to display the names of the establishments on the chart. The autopct parameter is set to "%1.1f%" to display the percentage of each slice on the chart.
- Finally, the plt.legend() function is called to display a legend for the chart.
- Note that the value_counts() method is used to count the number of occurrences of each establishment in the DataFrame df, and the head(10) function is used to select the top 10 establishments

Aggregate Ratings of these Establishments

<Axes: xlabel='aggregate_rating', ylabel='establishment'>



- **CODE:-**
- `plt.figure(figsize=(15,7.5))`
- `sns.barplot(x=df.aggregate_rating,y=df.establishment,hue=df.delivery,ci=True)`

Delhi Data Analysis

First we have to fetch Data of New Delhi from whole DataSet

```
In [148]: Delhi=df[df.city=="New Delhi"]
```

```
In [149]: Delhi
```

```
Out[149]:
```

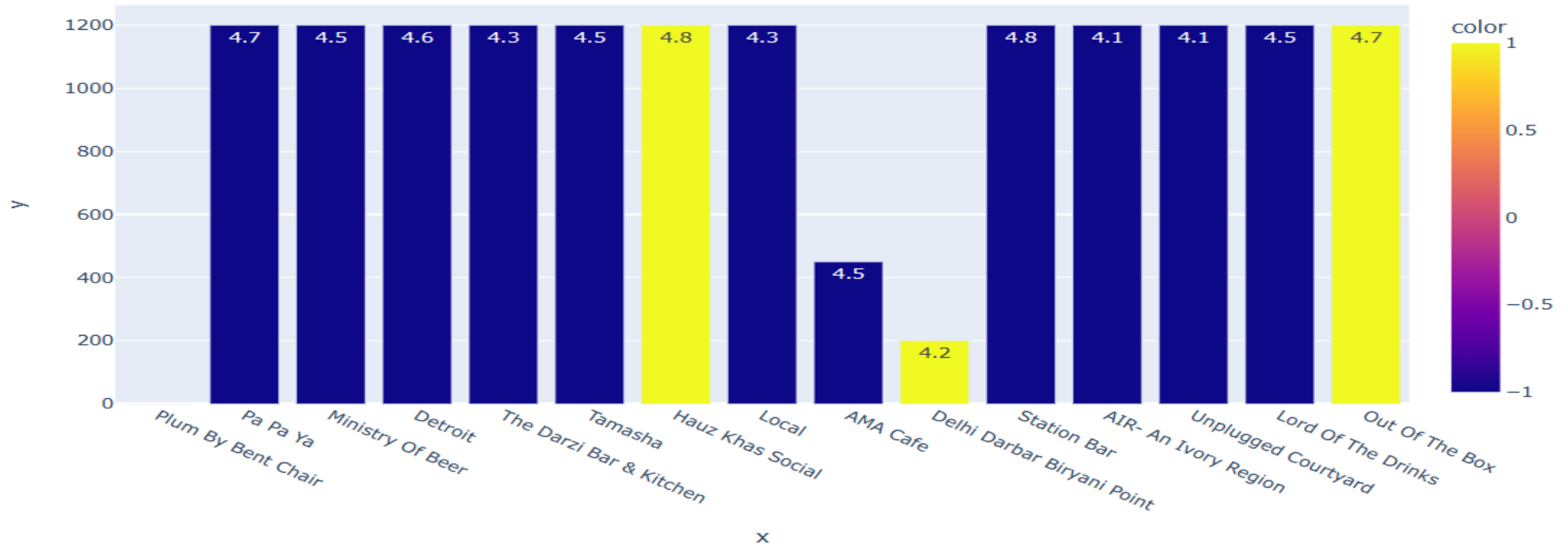
	res_id	name	establishment	uri	address	city	city_id	locality	latitude	longitude	...	currency
53884	18895645	Plum By Bent Chair	[Casual Dining]	https://www.zomato.com/ncr/plum-by-bent-chair-...	The Walk, Worldmark 2, Aerocity, New Delhi	New Delhi	1	Aerocity	28.551540	77.122704	...	Rs.
53885	18429148	Pa Pa Ya	[Fine Dining]	https://www.zomato.com/ncr/pa-pa-ya-saket-new-...	Dome, Level 4, Select Citywalk, A-3, District ...	New Delhi	1	Select Citywalk Mall, Saket	28.528391	77.218611	...	Rs.
53886	18418277	Ministry Of Beer	[Lounge]	https://www.zomato.com/ncr/ministry-of-beer-co-...	M 44, Outer Circle, Connaught Place, New Delhi	New Delhi	1	Connaught Place	28.633286	77.222787	...	Rs.
53887	18625812	Detroit	[Bar]	https://www.zomato.com/ncr/detroit-connaught-p-...	F40, 2nd Floor, Inner Circle, Connaught Place...	New Delhi	1	Connaught Place	28.631357	77.220251	...	Rs.
53888	18423151	The Darzi Bar & Kitchen	[Casual Dining]	https://www.zomato.com/TheDarziBar?utm_source=...	H 55, 1st Floor, Outer Circle, Connaught Place...	New Delhi	1	Connaught Place	28.635508	77.219096	...	Rs.
...
58538	310776	Gastronomica Kitchen & Bar	[Bar]	https://www.zomato.com/ncr/gastronomica-kitchen-...	M-55, 2nd Floor, M Block Market, Greater Kailash...	New Delhi	1	M Block Market, Greater Kailash 1 (GK1)	28.550421	77.235138	...	Rs.
58546	18168125	Informal by Imperfecto	[Lounge]	https://www.zomato.com/ncr/informal-by-imperfe-...	52, Tolstoy Lane, Near Connaught Place, Janpat...	New Delhi	1	Janpath	28.627185	77.220571	...	Rs.
58550	18281985	Teddy Boy	[Lounge]	https://www.zomato.com/ncr/teddy-boy-connaught-...	N-86, 1st Floor, Outer Circle, Connaught Place...	New Delhi	1	Connaught Place	28.630120	77.219803	...	Rs.
58565	18500811	Sixth Empirica	[Casual Dining]	https://www.zomato.com/ncr/sixth-empirica-punj-...	Building 6, 2nd & 3rd Floor, NWA, Punjabi Bagh...	New Delhi	1	Punjabi Bagh	28.666524	77.123032	...	Rs.
58566	18128893	Lavaash By Saby	[Casual Dining]	https://www.zomato.com/ncr/lavaash-by-saby-meh-...	H-5/1, Ambawatta One, Kalkadass Marg, Mehrauli...	New Delhi	1	Mehrauli	28.526504	77.185070	...	Rs.

1847 rows × 27 columns

CODE:-

```
px.bar(x=Delhi.name.head(15),y=Delhi.
average_cost_for_two.head(15),text=D
elhi.aggregate_rating.head(15),color=D
elhi.delivery.head(15))
```

- This code creates a bar chart using the Plotly Express library in Python. The chart will have 15 bars, with the x-axis representing the names of the top 15 restaurants in Delhi from the DataFrame Delhi, and the y-axis representing the average cost for two at each restaurant. The text displayed above each bar will be the aggregate rating of the corresponding restaurant, and the color of each bar will be determined by the delivery status of the restaurant.
- To create the chart, the px.bar function from the Plotly Express library is called to create the bar chart. The x parameter is set to the name column of the first 15 rows of the DataFrame Delhi, the y parameter is set to the average_cost_for_two column of the first 15 rows of Delhi, the text parameter is set to the aggregate_rating column of the first 15 rows of Delhi, and the color parameter is set to the delivery column of the first 15 rows of Delhi.



The average cost for two in Hauz Khas Social with the most ratings of 4.8 is Rs1200 and home delivery is also available

Aggregate Ratings

```
In [151]: Delhi.aggregate_rating.max()
```

```
Out[151]: 4.9
```

```
In [152]: Delhi[Delhi.aggregate_rating==4.9]
```

```
Out[152]:
```

	res_id	name	establishment	uri	address	city	city_id	locality	latitude	longitude	...	currency
53884	18895645	Plum By Bent Chair	[Casual Dining]	https://www.zomato.com/ncr/plum-by-bent-chair-...	The Walk, Worldmark 2, Aerocity, New Delhi	New Delhi	1	Aerocity	28.551540	77.122704	...	Rs.
53926	18807069	Chili's Grill & Bar	[Casual Dining]	https://www.zomato.com/ncr/chilis-grill-bar-ta...	S 8 & 9, 2nd Floor, Pacific Mall, Tagore Garden...	New Delhi	1	Pacific Mall, Tagore Garden	28.642270	77.106436	...	Rs.
53927	4751	Chili's Grill & Bar	[Casual Dining]	https://www.zomato.com/ncr/chilis-grill-bar-va...	3rd Floor, Ambience Mall, Nelson Mandela Road, ...	New Delhi	1	Ambience Mall, Vasant Kunj	28.540954	77.154802	...	Rs.
54079	18896689	Dragonfly Experience	[Cocktail Bar]	https://www.zomato.com/ncr/dragonfly-experienc...	Worldmark 1, Aerocity, New Delhi	New Delhi	1	Worldmark 1, Aerocity	28.550590	77.122990	...	Rs.
55792	19062617	Xero Degrees	[Café]	https://www.zomato.com/ncr/xero-degrees-vijay-...	G-18B, Hudson Lane, Vijay Nagar, New Delhi	New Delhi	1	Vijay Nagar	28.694923	77.203577	...	Rs.
55822	19062617	Xero Degrees	[Café]	https://www.zomato.com/ncr/xero-degrees-vijay-...	G-18B, Hudson Lane, Vijay Nagar, New Delhi	New Delhi	1	Vijay Nagar	28.694923	77.203577	...	Rs.
57828	18895645	Plum By Bent Chair	[Casual Dining]	https://www.zomato.com/ncr/plum-by-bent-chair-...	The Walk, Worldmark 2, Aerocity, New Delhi	New Delhi	1	Aerocity	28.551540	77.122704	...	Rs.
57872	18807069	Chili's Grill & Bar	[Casual Dining]	https://www.zomato.com/ncr/chilis-grill-bar-ta...	S 8 & 9, 2nd Floor, Pacific Mall, Tagore Garden...	New Delhi	1	Pacific Mall, Tagore Garden	28.642270	77.106436	...	Rs.
57875	4751	Chili's Grill & Bar	[Casual Dining]	https://www.zomato.com/ncr/chilis-grill-bar-va...	3rd Floor, Ambience Mall, Nelson Mandela Road, ...	New Delhi	1	Ambience Mall, Vasant Kunj	28.540954	77.154802	...	Rs.
58134	18983449	ABB - Andrea's Bar And Brasserie	[Casual Dining]	https://www.zomato.com/ncr/abb-andreas-bar-and...	Shop 12, Middle Lane, Khan Market, New Delhi	New Delhi	1	Khan Market	28.600624	77.227395	...	Rs.
58137	19081152	Doodle's Garden	[Casual Dining]	https://www.zomato.com/ncr/doodles-garden-punj...	Plot 1, 2nd Floor, NWA Club Road, Punjabi Bagh...	New Delhi	1	Punjabi Bagh	28.686740	77.123331	...	Rs.

11 rows × 27 columns

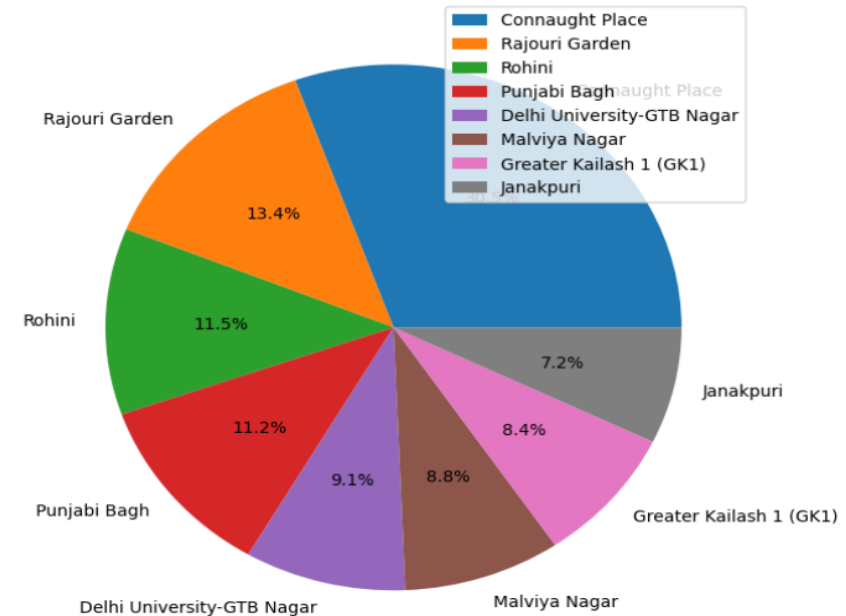
Number of Restaurants in different localities of Delhi

```
In [153]: Delhi.locality.value_counts()
```

```
Out[153]: locality
Connaught Place      128
Rajouri Garden       56
Rohini                48
Punjabi Bagh         47
Delhi University-GTB Nagar  38
...
Vardhaman Crown Mall      1
JMD Kohinoor Mall, Greater Kailash  1
Basant Lok Market, Vasant Vihar    1
Spark Mall, Kamla Nagar    1
West Gate Mall, Rajouri Garden    1
Name: count, Length: 257, dtype: int64
```

```
In [154]: plt.figure(figsize=(15,7.5))
plt.pie(Delhi.locality.value_counts().head(8),labels=Delhi.locality.value_counts().head(8).index,autopct="%1.1f%%")
plt.legend()
```

```
Out[154]: <matplotlib.legend.Legend at 0x1c3bf5f52d0>
```



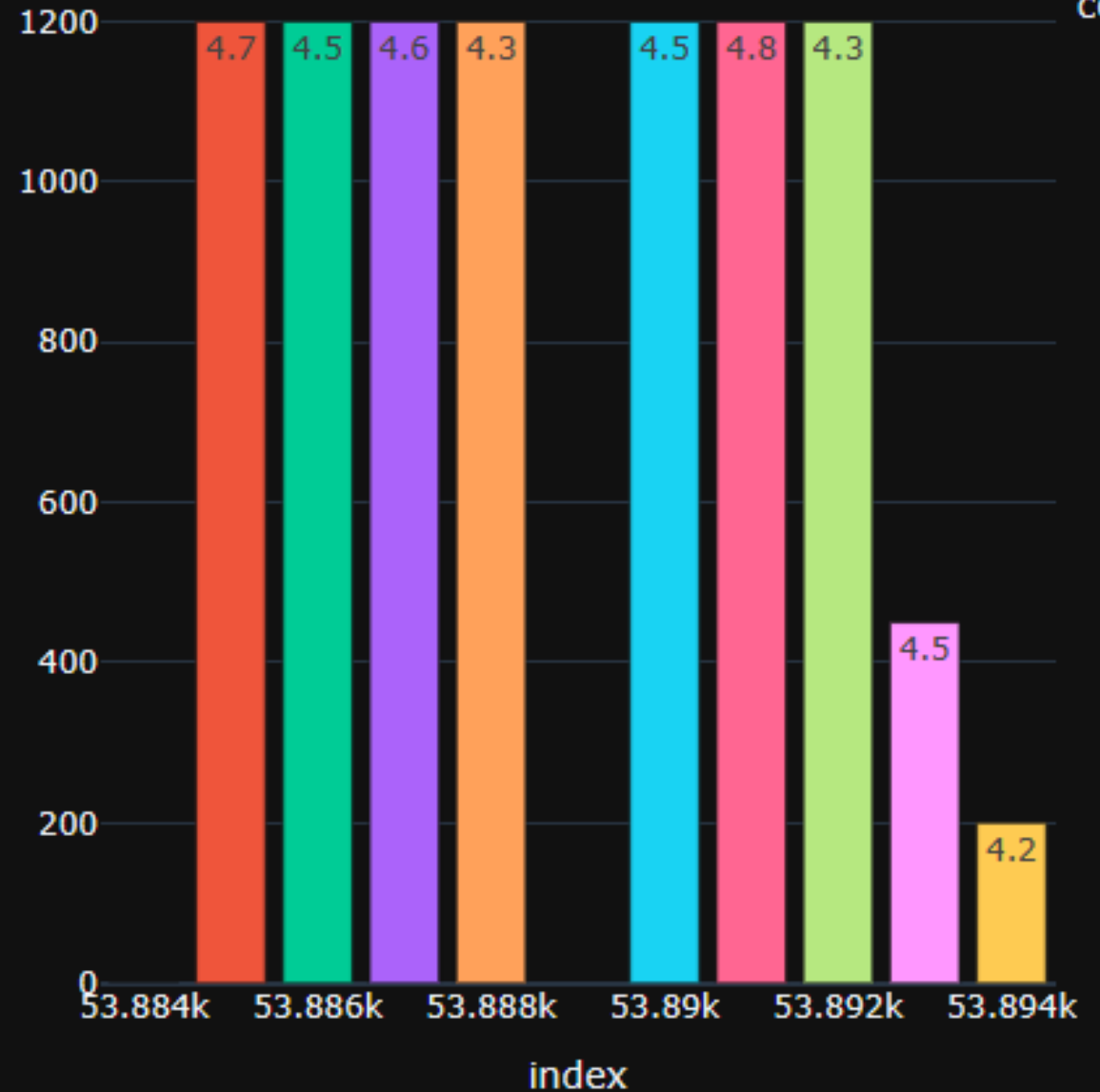
Connaught Place has most number of restaurants in Delhi

Restaurants offering different cuisines in Delhi

- **CODE:-**
plt.figure(figsize=(45,5))
- px.bar(Delhi.name.head(10),y=Delhi.average_cost_for_two.head(10),text=Delhi.aggregate_rating.head(10),color=Delhi.cuisines.head(10),template="plotly_dark")

```
5]: Delhi.cuisines.value_counts()
```

```
5]: cuisines
North Indian      133
Street Food       46
Desserts, Ice Cream, Beverages  41
Paan              40
North Indian, Chinese  34
...
Cafe, Italian, Continental, Healthy Food, Pizza, Coffee  1
Cafe, Continental, European  1
Cafe, American, Italian  1
Cafe, Mediterranean, Sushi  1
North Indian, South Indian, Chinese, Fast Food  1
Name: count, Length: 695, dtype: int64
```



Specific Cuisine offered by restaurants in Delhi

In [157]: `a=Delhi[Delhi.cuisines=="North Indian"]`

In [158]: `a`

Out[158]:

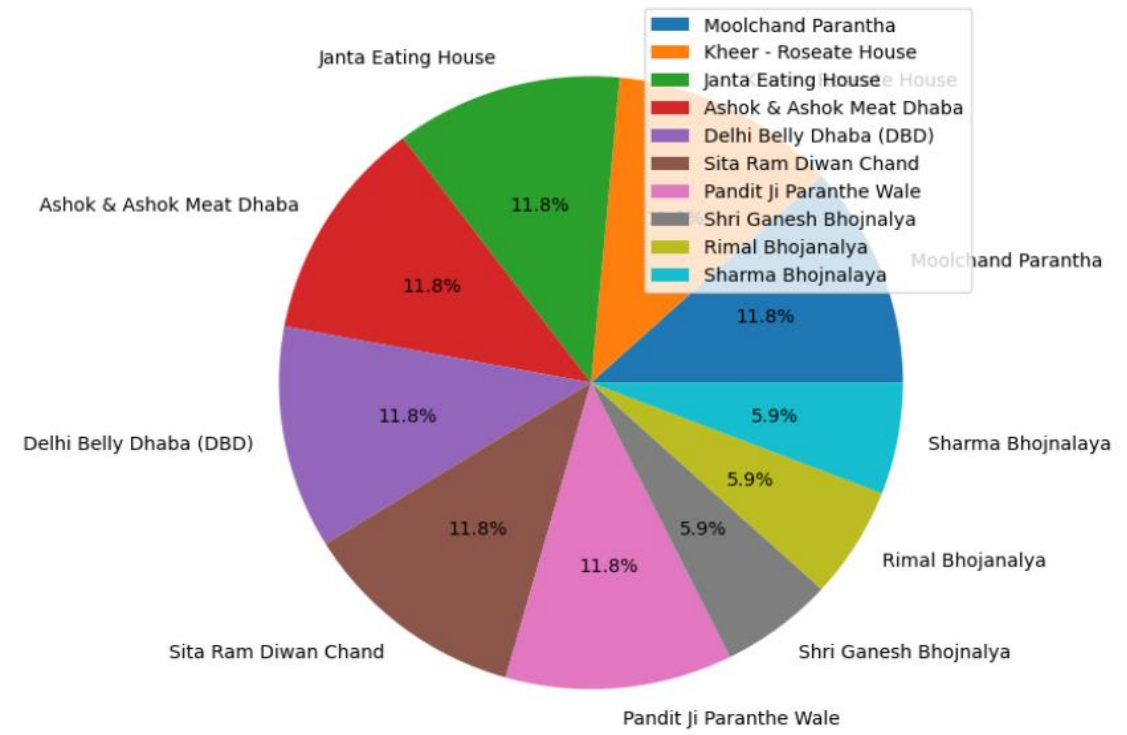
	res_id	name	establishment	url	address	city	city_id	locality	latitude	longitude	...	currenc
53901	310470	Moolchand Parantha	☐	https://www.zomato.com/ncr/moolchand-parantha-...	Moolchand Metro Station, Lajpat Nagar 4, New D...	New Delhi	1	Lajpat Nagar 4	28.564135	77.234348	...	R
53906	302424	Janta Eating House	['Quick Bites']	https://www.zomato.com/ncr/janta-eating-house-...	Loni Road, Shahdara, New Delhi	New Delhi	1	Shahdara	28.682230	77.292380	...	R
53914	308969	Shree Gopal Ji Chole Bhature	☐	https://www.zomato.com/ncr/shree-gopal-ji-chole...	Flat 148, Pocket 7, Rohini, New Delhi	New Delhi	1	Rohini	28.700392	77.102034	...	R
53924	6574	Pandit Ji Paranthhe Wale	['Quick Bites']	https://www.zomato.com/ncr/pandit-ji-paranthhe-...	Shop 14, Ashok Vihar Phase 2, New Delhi	New Delhi	1	Ashok Vihar Phase 2	28.696565	77.179951	...	R
53970	6075	Sita Ram Diwan Chand	['Quick Bites']	https://www.zomato.com/ncr/sita-ram-diwan-cha...	2248, Near Imperial Cinema, Paharganj, New Delhi	New Delhi	1	Paharganj	28.642408	77.210479	...	R
...
58261	18610363	UK Chicken	['Quick Bites']	https://www.zomato.com/ncr/uk-chicken-rajouri-...	C 1, Shop 5, Near Ramesh Nager Metro Station, ...	New Delhi	1	Rajouri Garden	28.614040	77.076348	...	R
58274	18273527	Bajaj Vaishno Dhaba	['Dhaba']	https://www.zomato.com/ncr/bajaj-vaishno-dhaba...	117, Amrit Kaur Market, Opposite New Delhi Rai...	New Delhi	1	Paharganj	28.642217	77.218356	...	R
58289	7788	Punjabi Rasoi	['Quick Bites']	https://www.zomato.com/ncr/punjabi-rasoi-ina? u...	Dilli Haat, INA, New Delhi	New Delhi	1	Dilli Haat, INA	28.573095	77.206154	...	R
58292	305606	Zustt Yummy	['Quick Bites']	https://www.zomato.com/zusttyummyrestaurant? ut...	70 B, LGF, Safdarjung Enclave, Safdarjung, New...	New Delhi	1	Safdarjung	28.564978	77.198894	...	R


```
: a.name.value_counts().head(10)
```

name	
Moolchand Parantha	2
Kheer - Roseate House	2
Janta Eating House	2
Ashok & Ashok Meat Dhaba	2
Delhi Belly Dhaba (DBD)	2
Sita Ram Diwan Chand	2
Pandit Ji Paranthe Wale	2
Shri Ganesh Bhojnalya	1
Rimal Bhojanalya	1
Sharma Bhojnalaya	1
Name: count, dtype: int64	

```
In [160]: plt.figure(figsize=(15,7.5))
plt.pie(a.name.value_counts().head(10),labels=a.name.value_counts().head(10).index,autopct="%1.1f%%")
plt.legend()

Out[160]: <matplotlib.legend.Legend at 0x1c3bf636550>
```



This code creates a pie chart using the Matplotlib library in Python. The chart will display the distribution of the top 10 values in the 'name' column of the DataFrame 'a' as a percentage of the total.

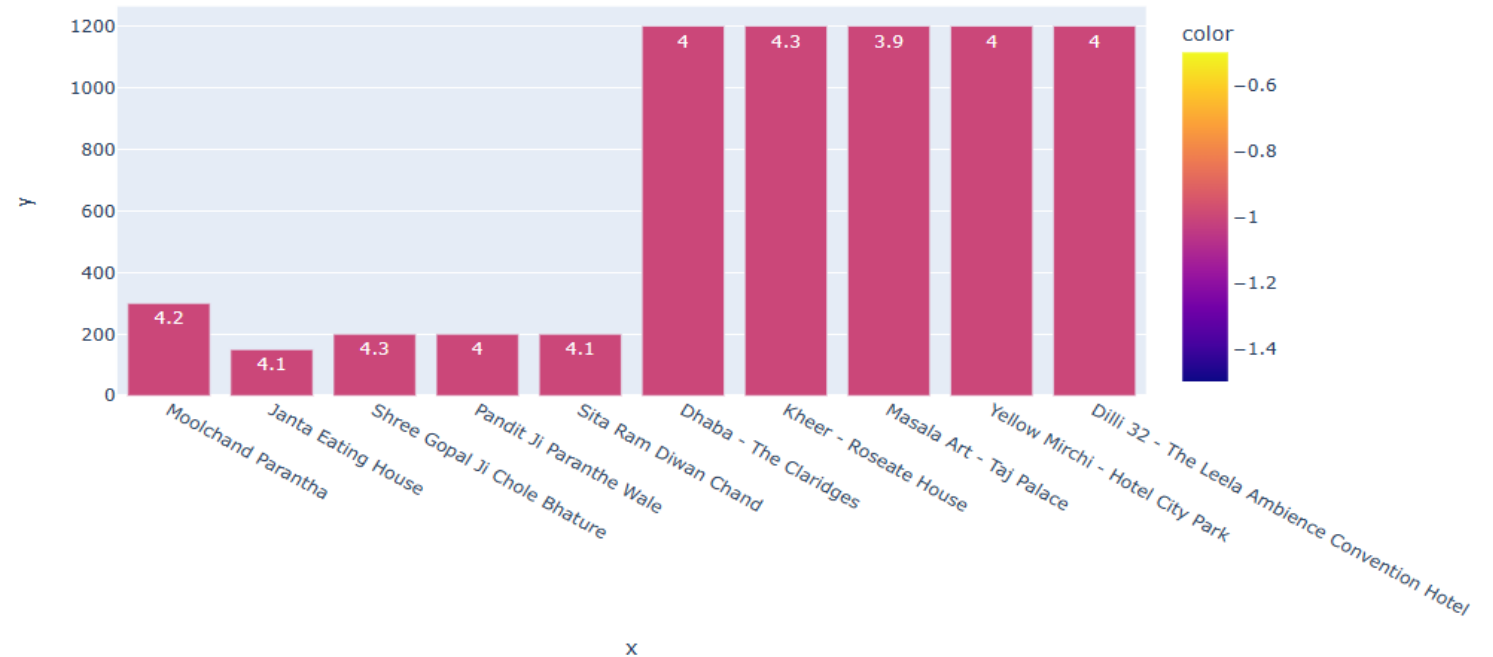
Here's a breakdown of the code:

- plt.figure(figsize=(15,7.5)): This line sets the figure size for the plot.
- plt.pie(a.name.value_counts().head(10), labels=a.name.value_counts().head(10).index, autopct="%1.1f%%"): This line creates the pie chart. It uses the 'value_counts()' method to count the occurrences of each unique value in the 'name' column of DataFrame 'a', selects the top 10 values, and then plots them as a pie chart. The 'labels' parameter sets the labels for each slice of the pie chart, and 'autopct="%1.1f%%"' formats the percentage display on each slice.
- plt.legend(): This line adds a legend to the pie chart.

Average Cost for two in Delhi for North Indian cuisine

- The provided code snippet aims to create a bar chart using Plotly Express in Python. The chart will display information from the DataFrame 'a' for the top 10 entries. Here is a breakdown of the code:
- `plt.figure(figsize=(12,6))`: This line sets the figure size for the plot using Matplotlib.
- `px.bar(x=a.name.head(10), y=a.average_cost_for_two.head(10), text=a.aggregate_rating.head(10), color=a.delivery.head(10), template="plotly_dark")`: This line attempts to create a bar chart using Plotly Express. The 'x' parameter represents the names of the top 10 entries from the 'name' column of DataFrame 'a', the 'y' parameter represents the average cost for two from the 'average_cost_for_two' column, the 'text' parameter displays the aggregate rating from the 'aggregate_rating' column, and the 'color' parameter is based on the delivery status from the 'delivery' column. The 'template="plotly_dark"' sets the dark theme for the plot.

```
In [327]: plt.figure(figsize=(12,6))
px.bar(x=a.name.head(10),y=a.average_cost_for_two.head(10),text=a.aggregate_rating.head(10),color=a.delivery.head(10),)
```



<Figure size 1200x600 with 0 Axes>

Average cost for two in Delhi for North Indian Cuisine is Rs1200 and the best restaurant offering North Indian in Delhi with the ratings of 4.3 is kheer-Roseate House

CONCLUSION:- From the given data we can conclude that

- **1. Average Cost in Agra:**

- The average cost of a meal for two people in Agra is approximately Rs. 1600.
- Most restaurants fall within the price range of Rs. 1000 to Rs. 2000 for two people.

- **2. Specific Restaurant Insights:**

- "Pinch of Spice" stands out with a high rating of 4.6 and an average cost of Rs. 1600 for two people.
- "Bikaner" has a good rating of 4.4 with a slightly lower average cost of Rs. 1400 for two people.
- "Dominos Pizza" has a lower rating of 3.5 but a more affordable average cost of Rs. 1200 for two people.

- **3. Delivery Status Differentiation:**

- The scatter plot analysis revealed that the delivery status significantly impacts the restaurant landscape.
- Restaurants with delivery services generally have diverse ratings and costs, while those without delivery or with
- unknown status cluster differently.

City-Specific Exploration - Delhi

- **The average cost for two** in Hauz Khas Social with the most ratings of 4.8 is Rs1200 and home delivery is also available.
- **Connaught Place** has the most number of restaurants in Delhi (128 Restaurants).
- **North Indian** cuisine is the most popular cuisine in Delhi .
- **Average cost for two in Delhi for North Indian Cuisine** is Rs1200 and the best restaurant offering North Indian in Delhi with the ratings of 4.3 is kheer-Roseate House

RECOMMENDATIONS

- **1. Focus on Quality and Affordability:**

- Restaurants should prioritize maintaining high-quality services to garner positive ratings.
- Offering competitive pricing, especially within the Rs. 1000 to Rs. 2000 range, can attract a
- broader customer base.

- **2. Enhance Online Presence:**

- Given the influence of delivery status on the restaurant landscape, owners should consider
- providing clear and reliable information on online delivery services.

- **3. Improve Ratings:**

- Restaurants aiming for success should prioritize aspects that contribute to positive ratings, such
- as excellent customer service, quality food, and a pleasant dining experience.

- **4. Diversify Menu and Pricing:**

- Offering a diverse menu with different price points can cater to a wider audience, accommodating
- various preferences and budgets.

- **5. Regularly Monitor and Update Information:**

- Regularly updating information on Zomato regarding delivery status, menu, and pricing ensures
- that potential customers have accurate and up-to-date details

6. Analyze Competitor Landscape:-

Understanding the strengths and weaknesses of competitors, especially those with similar ratings, can help restaurants identify areas for improvement and differentiation

*For those seeking an authentic North Indian culinary experience in Delhi, we recommend exploring the offerings of [highlighted restaurants].

*Considerations for budget-friendly options and premium dining experiences can be tailored based on individual preferences and expectations

This project structure provides a comprehensive framework for understanding and improving restaurant success factors. Restaurant owners and Zomato users can benefit from these recommendations to enhance their overall experience and engagement within the platform.