In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import plotly.figure_factory as ff
```

In [2]:
```
df=pd.read_csv(r'C:\Users\abhis\Downloads\Compressed\Data\3-Zomato Data Analysis/zomato.
df.head()
```

Out[2]:

	url	address	name	online_order	book_table	rate	votes	
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1/5	775	4
1	https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1/5	787	
2	https://www.zomato.com/SanchurroBangalore?cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8/5	918	-
3	https://www.zomato.com/bangalore/addhuri-udupi...	1st Floor, Annakuteera, 3rd Stage, Banashankar...	Addhuri Udupi Bhojana	No	No	3.7/5	88	-
4	https://www.zomato.com/bangalore/grand-village...	10, 3rd Floor, Lakshmi Associates, Gandhi Baza...	Grand Village	No	No	3.8/5	166	802

In [3]:
```
df.shape
```

Out[3]:
```
(51717, 17)
```

In [4]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51717 entries, 0 to 51716
Data columns (total 17 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   url                          51717 non-null  object
 1   address                      51717 non-null  object
 2   name                         51717 non-null  object
 3   online_order                 51717 non-null  object
 4   book_table                   51717 non-null  object
 5   rate                         43942 non-null  object
 6   votes                        51717 non-null  int64 
```

```
 7   phone                        50509 non-null  object
 8   location                     51696 non-null  object
 9   rest_type                    51490 non-null  object
 10  dish_liked                   23639 non-null  object
 11  cuisines                     51672 non-null  object
 12  approx_cost(for two people)  51371 non-null  object
 13  reviews_list                 51717 non-null  object
 14  menu_item                    51717 non-null  object
 15  listed_in(type)              51717 non-null  object
 16  listed_in(city)              51717 non-null  object
dtypes: int64(1), object(16)
memory usage: 6.7+ MB
```

In [5]:
```
df.isna().sum()
```

Out[5]:
```
url                                0
address                            0
name                               0
online_order                       0
book_table                         0
rate                            7775
votes                              0
phone                           1208
location                          21
rest_type                        227
dish_liked                     28078
cuisines                          45
approx_cost(for two people)      346
reviews_list                       0
menu_item                          0
listed_in(type)                    0
listed_in(city)                    0
dtype: int64
```

In [6]:
```
missing_value_list=[]
for i in df.columns:
    if df[i].isna().sum()>0:
        missing_value_list.append(i)
missing_value_list
```

Out[6]:
```
['rate',
 'phone',
 'location',
 'rest_type',
 'dish_liked',
 'cuisines',
 'approx_cost(for two people)']
```

In [7]:
```
# % of missing values
for i in missing_value_list:
    print('{} has {} % missing values'.format(i,np.round(df[i].isna().sum()/len(df[i])*1
```

```
rate has 15.0337 % missing values
phone has 2.3358 % missing values
location has 0.0406 % missing values
rest_type has 0.4389 % missing values
dish_liked has 54.2916 % missing values
cuisines has 0.087 % missing values
approx_cost(for two people) has 0.669 % missing values
```

In [8]:
```
#Analysis on Rate
```

In [9]:
```
df['rate'].unique()
```

Out[9]:
```
array(['4.1/5', '3.8/5', '3.7/5', '3.6/5', '4.6/5', '4.0/5', '4.2/5',
       '3.9/5', '3.1/5', '3.0/5', '3.2/5', '3.3/5', '2.8/5', '4.4/5',
```

```
       '4.3/5', 'NEW', '2.9/5', '3.5/5', nan, '2.6/5', '3.8 /5', '3.4/5',
       '4.5/5', '2.5/5', '2.7/5', '4.7/5', '2.4/5', '2.2/5', '2.3/5',
       '3.4 /5', '-', '3.6 /5', '4.8/5', '3.9 /5', '4.2 /5', '4.0 /5',
       '4.1 /5', '3.7 /5', '3.1 /5', '2.9 /5', '3.3 /5', '2.8 /5',
       '3.5 /5', '2.7 /5', '2.5 /5', '3.2 /5', '2.6 /5', '4.5 /5',
       '4.3 /5', '4.4 /5', '4.9/5', '2.1/5', '2.0/5', '1.8/5', '4.6 /5',
       '4.9 /5', '3.0 /5', '4.8 /5', '2.3 /5', '4.7 /5', '2.4 /5',
       '2.1 /5', '2.2 /5', '2.0 /5', '1.8 /5'], dtype=object)
```

In [10]:
```
#NEW,nan and _ are invalid datapoints and we only want the rating and not out of 5
df.dropna(axis='index',subset= ['rate'], inplace=True)
```

In [11]:
```
df.shape
```

Out[11]:
```
(43942, 17)
```

In [12]:
```
df['rate'].unique()
```

Out[12]:
```
array(['4.1/5', '3.8/5', '3.7/5', '3.6/5', '4.6/5', '4.0/5', '4.2/5',
       '3.9/5', '3.1/5', '3.0/5', '3.2/5', '3.3/5', '2.8/5', '4.4/5',
       '4.3/5', 'NEW', '2.9/5', '3.5/5', '2.6/5', '3.8 /5', '3.4/5',
       '4.5/5', '2.5/5', '2.7/5', '4.7/5', '2.4/5', '2.2/5', '2.3/5',
       '3.4 /5', '-', '3.6 /5', '4.8/5', '3.9 /5', '4.2 /5', '4.0 /5',
       '4.1 /5', '3.7 /5', '3.1 /5', '2.9 /5', '3.3 /5', '2.8 /5',
       '3.5 /5', '2.7 /5', '2.5 /5', '3.2 /5', '2.6 /5', '4.5 /5',
       '4.3 /5', '4.4 /5', '4.9/5', '2.1/5', '2.0/5', '1.8/5', '4.6 /5',
       '4.9 /5', '3.0 /5', '4.8 /5', '2.3 /5', '4.7 /5', '2.4 /5',
       '2.1 /5', '2.2 /5', '2.0 /5', '1.8 /5'], dtype=object)
```

In [13]:
```
def split(x):
    return x.split('/')[0]
```

In [14]:
```
df['rate'] = df['rate'].apply(split)
```

In [15]:
```
df.head(1)
```

Out[15]:

	url	address	name	online_order	book_table	rate	votes	ph
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1	775	42297555\r\n974377

In [16]:
```
df['rate'] = df['rate'].replace('NEW',0)
df['rate'] = df['rate'].replace('-', 0)
```

In [17]:
```
df['rate']=df['rate'].astype(float)
```

In [18]:
```
df['rate'].unique()
```

Out[18]:
```
array([4.1, 3.8, 3.7, 3.6, 4.6, 4. , 4.2, 3.9, 3.1, 3. , 3.2, 3.3, 2.8,
       4.4, 4.3, 0. , 2.9, 3.5, 2.6, 3.4, 4.5, 2.5, 2.7, 4.7, 2.4, 2.2,
       2.3, 4.8, 4.9, 2.1, 2. , 1.8])
```

In [19]:
```
df3 = df.copy()
```

Avg rating of each restuarant

In [20]:
```
df['name'].value_counts()
```

Out[20]:
```
Cafe Coffee Day                89
Onesta                         85
Empire Restaurant              71
Just Bake                      68
Five Star Chicken              68
                               ..
SV Food Garden                  1
Ginger restaurant               1
Darjeeling Hot Momos            1
Sri Annapoorna Andhra Mess      1
SeeYa Restaurant                1
Name: name, Length: 7162, dtype: int64
```

In [21]:
```
avg_rating = df.groupby('name')['rate'].mean().to_frame()
avg_rating
```

Out[21]:

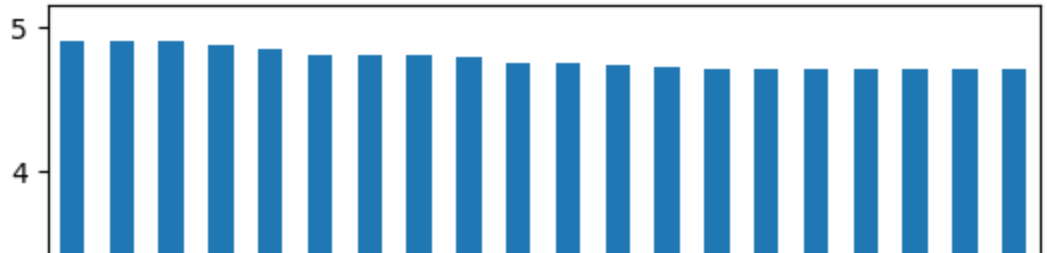
	rate
name	
#FeelTheROLL	3.400000
#L-81 Cafe	3.900000
#refuel	3.700000
1000 B.C	3.200000
100Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â°C	3.700000
...	...
i-Bar - The Park Bangalore	3.800000
iFruit Live Ice Creams	3.400000
iSpice Resto Cafe	3.700000
nu.tree	4.314286
re:cess - Hilton Bangalore Embassy GolfLinks	4.100000

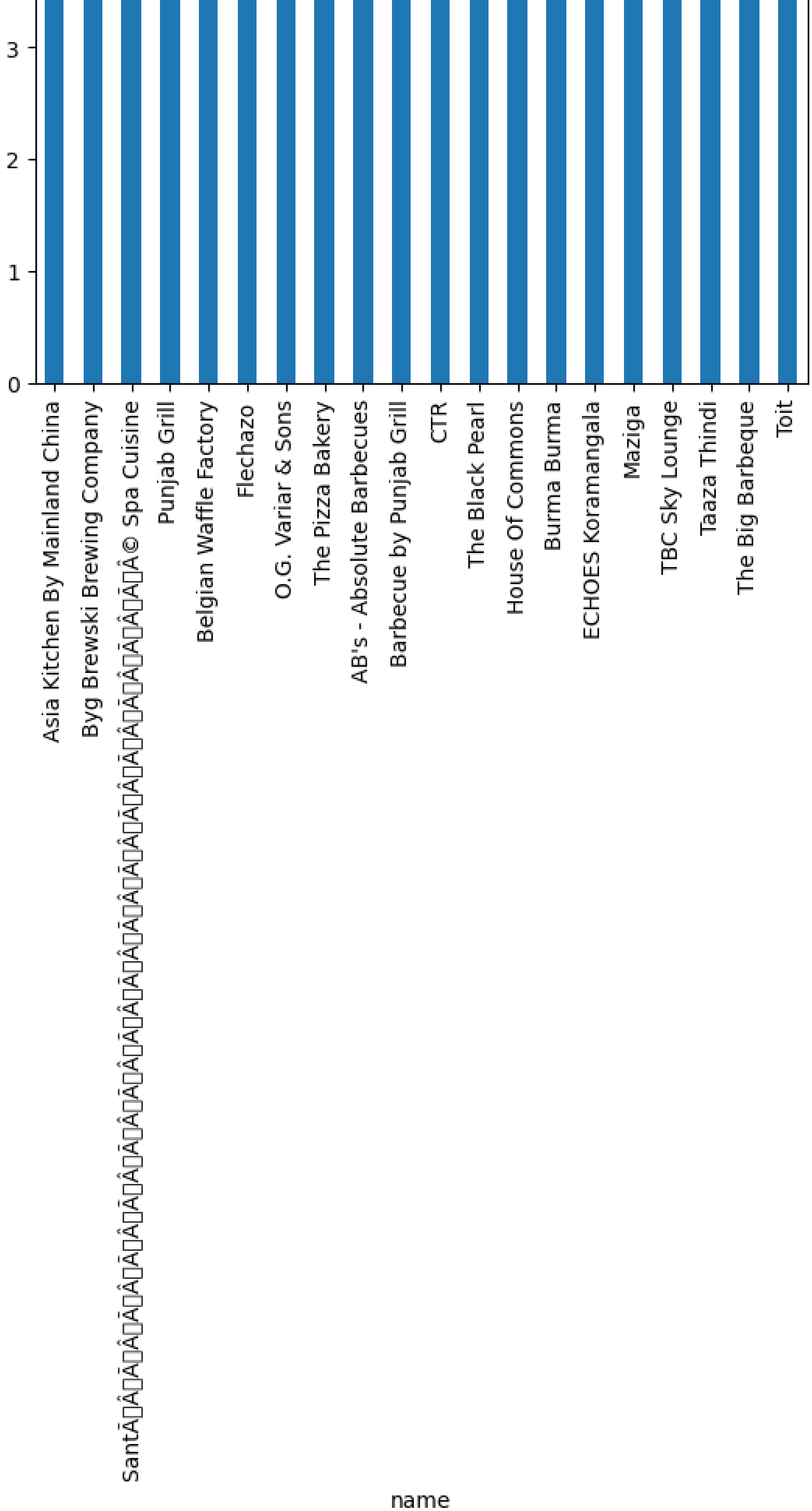
7162 rows × 1 columns

In [22]:
```
df.groupby('name')['rate'].mean().nlargest(20).plot.bar()
```

Out[22]:
```
<AxesSubplot:xlabel='name'>
```

```
C:\Users\abhis\anaconda3\lib\site-packages\IPython\core\pylabtools.py:151: UserWarning:
Glyph 131 (\x83) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\abhis\anaconda3\lib\site-packages\IPython\core\pylabtools.py:151: UserWarning:
Glyph 130 (\x82) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```





In [23]:
```
#plotly
x = avg_rating.nlargest(20,'rate')
```

```python
fig = px.bar(x,x=x.index,y=x['rate'], height = 800)
fig.show()
```

In [24]:
```python
rate_df = df.groupby('name')['rate'].mean().to_frame().reset_index().rename(columns={'na
```

In [25]:
```python
rate_df
```

Out[25]:

	restuarant	rating
0	#FeelTheROLL	3.400000
1	#L-81 Cafe	3.900000
2	#refuel	3.700000

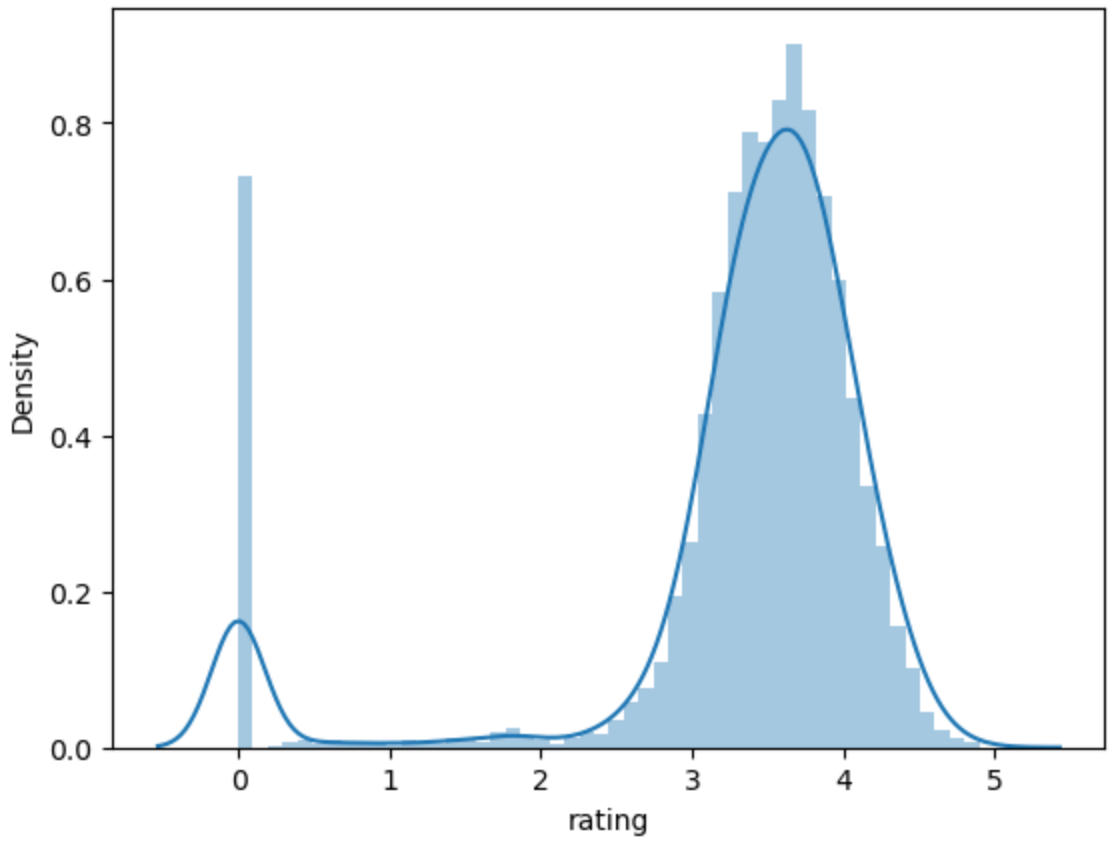
3	1000 B.C	3.200000
4	100Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â°C	3.700000
...	...	...
7157	i-Bar - The Park Bangalore	3.800000
7158	iFruit Live Ice Creams	3.400000
7159	iSpice Resto Cafe	3.700000
7160	nu.tree	4.314286
7161	re:cess - Hilton Bangalore Embassy GolfLinks	4.100000

7162 rows × 2 columns

In [26]:
```
sns.distplot(rate_df['rating'])
```

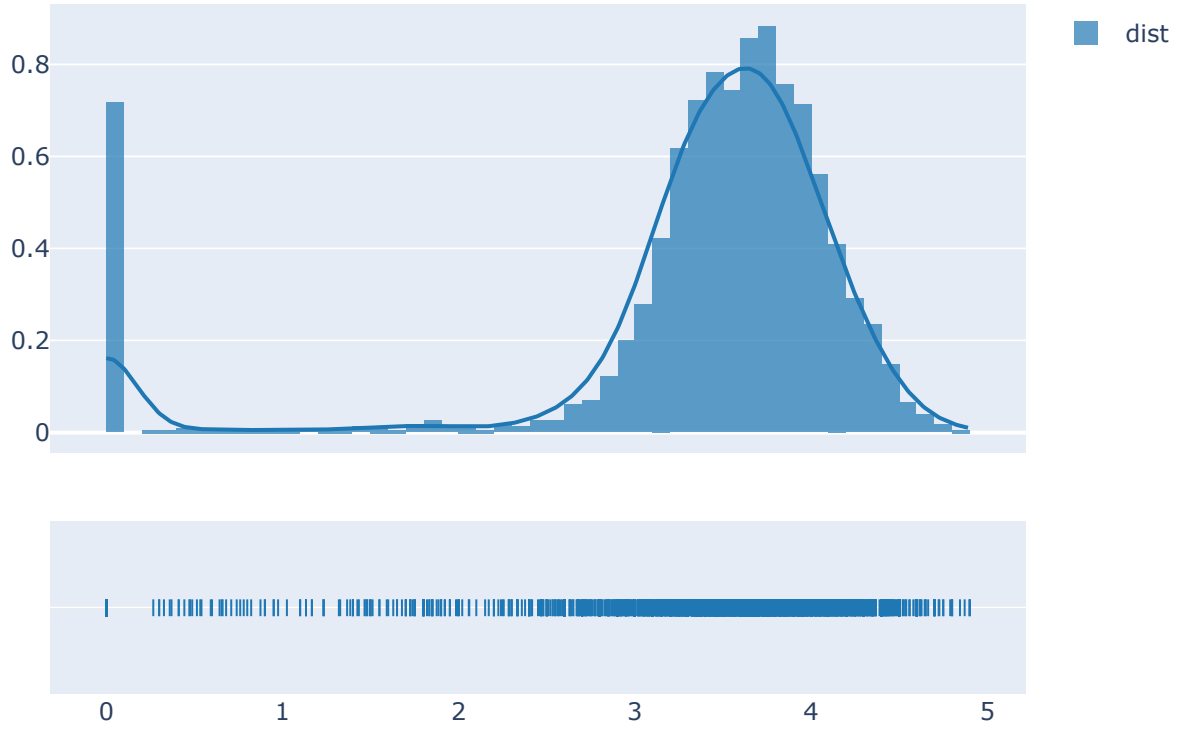
```
C:\Users\abhis\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
```

Out[26]:
```
<AxesSubplot:xlabel='rating', ylabel='Density'>
```



In [27]:
```
x =[rate_df['rating'].values.tolist()]
```

In [28]:
```
fig = ff.create_distplot(hist_data = [rate_df['rating'].values.tolist()],group_labels =
fig.show()
```



Almost more than 50 percent of restaurants has rating between 3 and 4. Restaurants having rating more than 4.5 are very rare

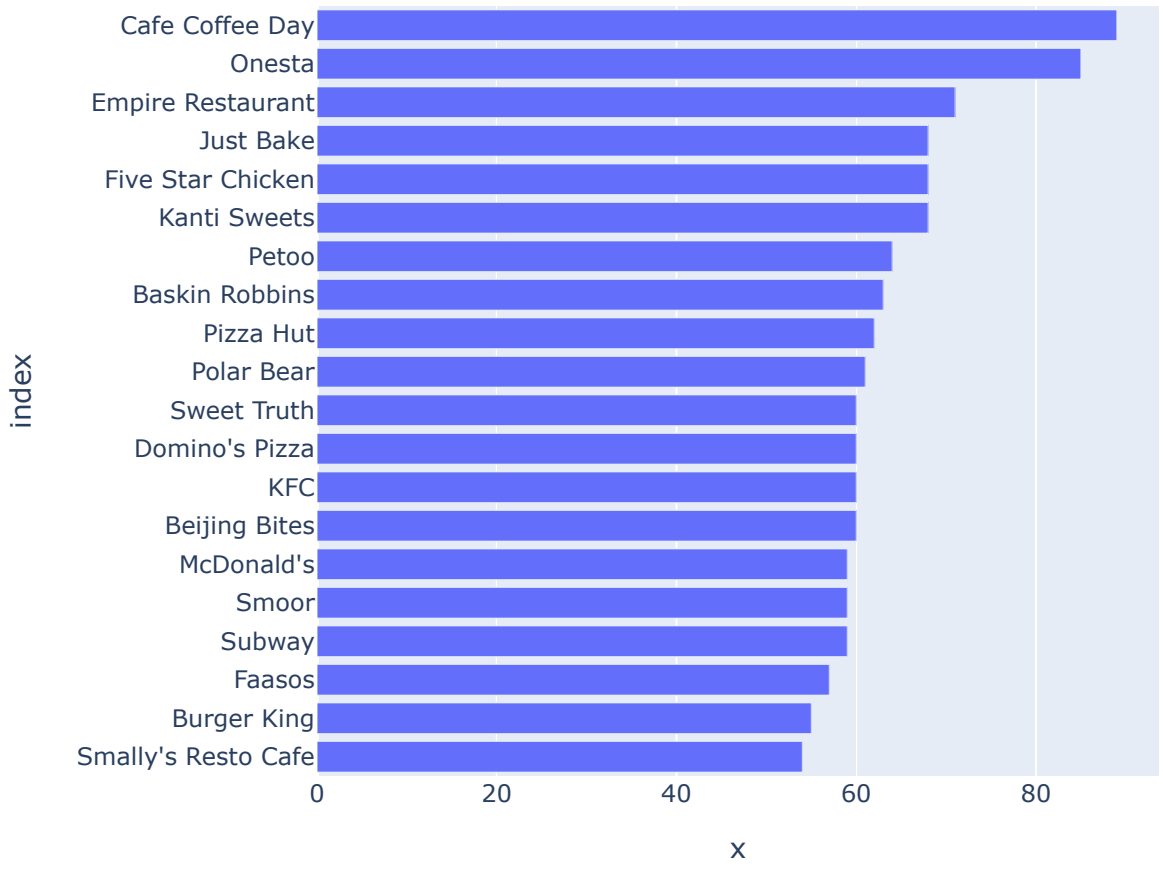
Top restuarants in Bengaluru

In [29]:
```
chains = df['name'].value_counts().nlargest(20).sort_values(ascending = True)
```

In [30]:
```
chains
```

Out[30]:
```
Smally's Resto Cafe     54
Burger King             55
Faasos                  57
Subway                  59
Smoor                   59
McDonald's              59
Beijing Bites           60
KFC                     60
Domino's Pizza          60
Sweet Truth             60
Polar Bear              61
Pizza Hut               62
Baskin Robbins          63
Petoo                   64
Kanti Sweets            68
Five Star Chicken       68
Just Bake               68
Empire Restaurant       71
Onesta                  85
Cafe Coffee Day         89
Name: name, dtype: int64
```

In [31]:
```
fig = px.bar(chains,x=chains, y=chains.index)
fig.show()
```



How many restuarants do not accept online order

In [32]:
```
df.head(3)
```

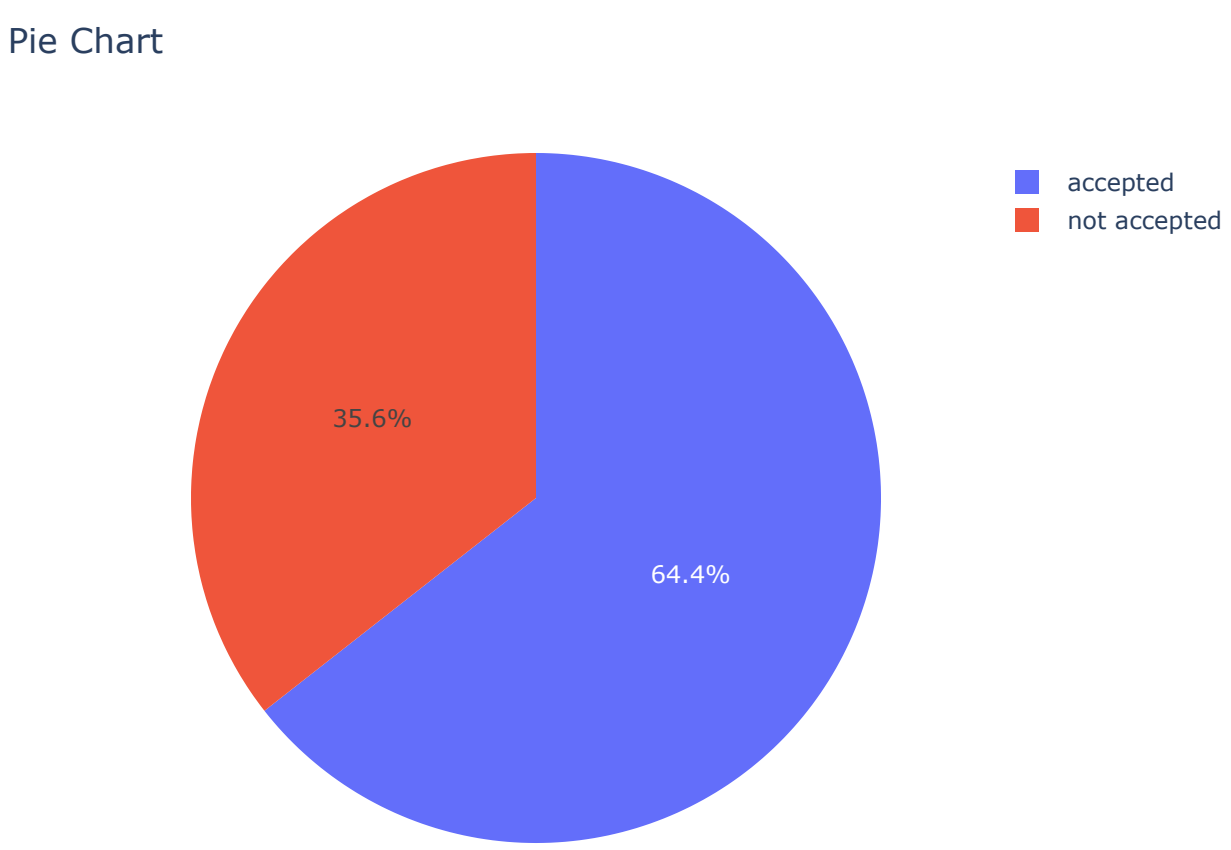
Out[32]:

	url	address	name	online_order	book_table	rate	votes	
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1	775	4229
1	https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1	787	0
2	https://www.zomato.com/SanchurroBangalore?cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8	918	+91

In [33]:
```
df['online_order'].value_counts()
```

Out[33]:
```
Yes     28308
No      15634
Name: online_order, dtype: int64
```

In [34]:
```
fig = px.pie(df, values = df['online_order'].value_counts(), names = ['accepted','not ac
fig.show()
```



In [ ]:
```

```

Ratio b/w restaurants that provide and do not provide table booking

In [35]:
```
df.head(3)
```

Out[35]:

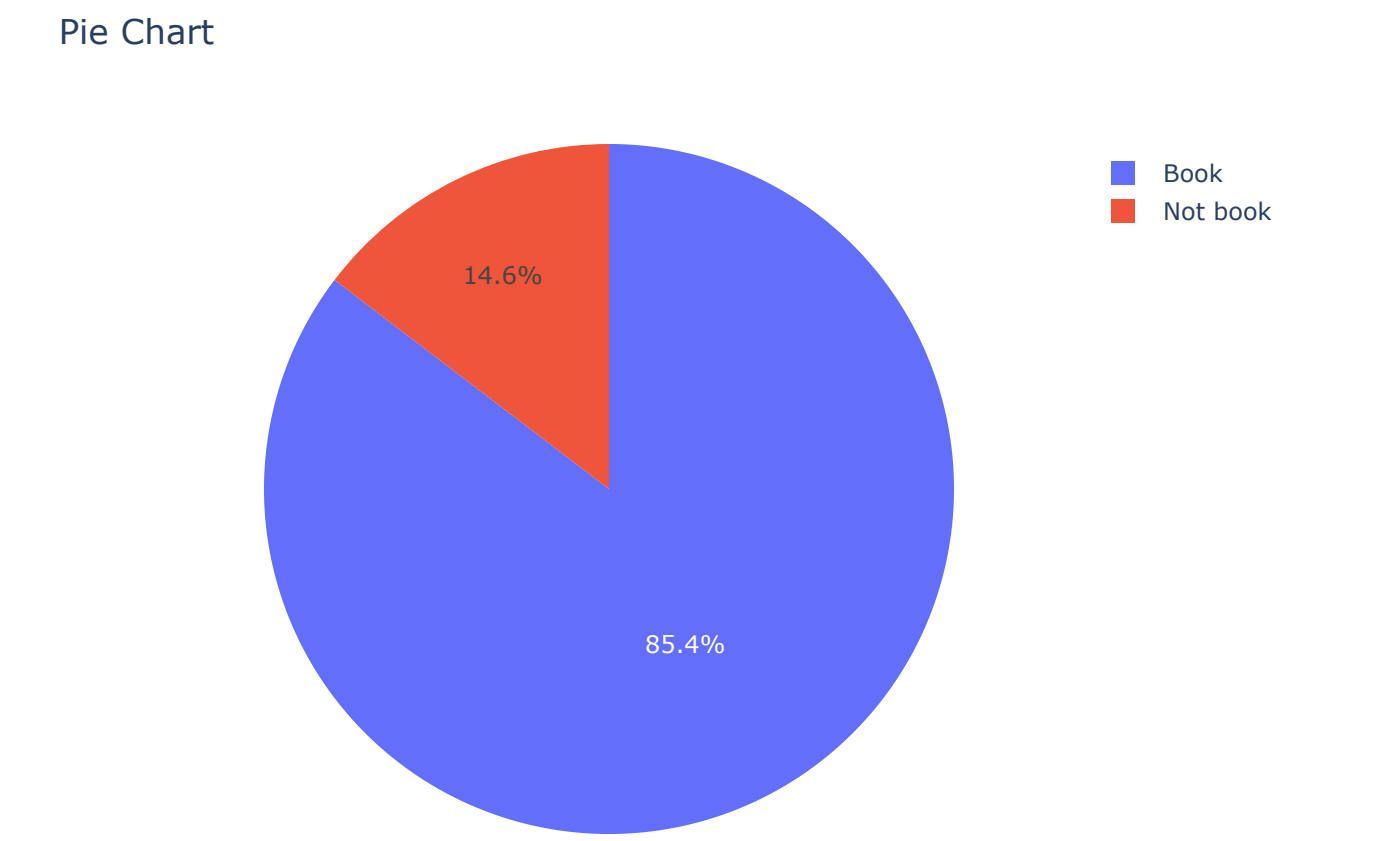
	url	address	name	online_order	book_table	rate	votes	
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1	775	4229

1	https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1	787	0
2	https://www.zomato.com/SanchurroBangalore?cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8	918	+91

In [36]:
```
df['book_table'].value_counts()
```

Out[36]:
```
No      37509
Yes      6433
Name: book_table, dtype: int64
```

In [37]:
```
fig = px.pie(df,values=df['book_table'].value_counts(),names=['Book','Not book'],title =
fig.show()
```



How many type of restuarants we have?

In [38]:
```
df.head(3)
```

Out[38]:

	url	address	name	online_order	book_table	rate	votes

0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1	775	4229
1	https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1	787	0
2	https://www.zomato.com/SanchurroBangalore?cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8	918	+91

In [39]:
```
df['rest_type'].isna().sum()
```

Out[39]:
```
151
```

In [40]:
```
rest_df = df['rest_type'].dropna()
```

In [41]:
```
rest_df.isna().sum()
```

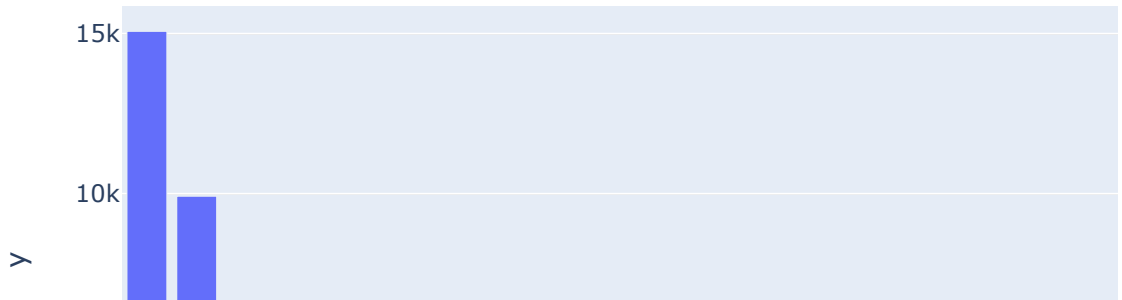
Out[41]:
```
0
```

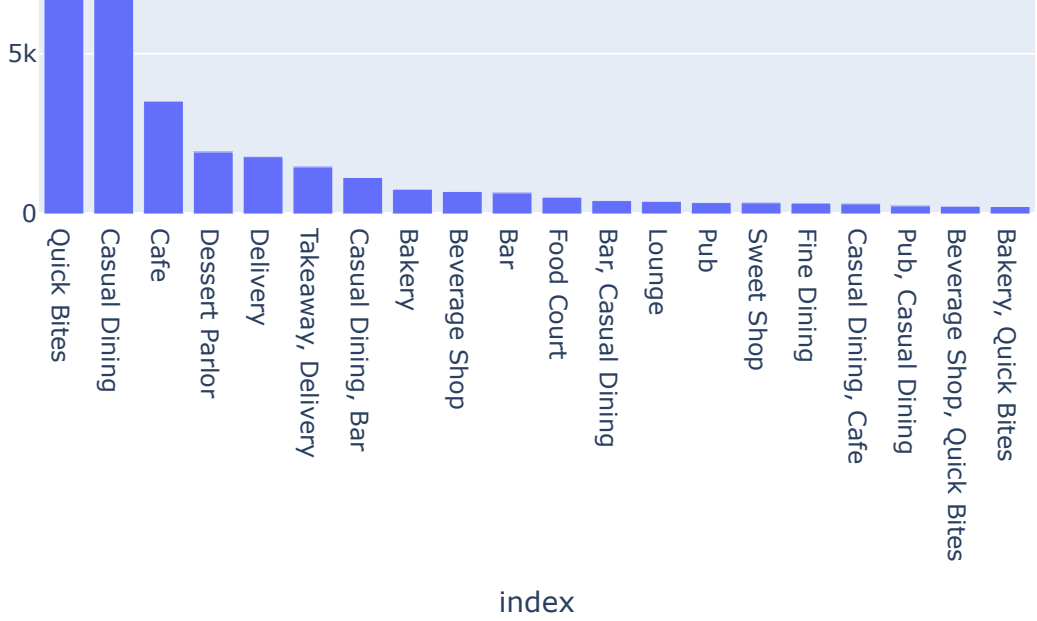
In [42]:
```
rest_df.value_counts()
```

Out[42]:
```
Quick Bites                   15073
Casual Dining                  9923
Cafe                           3527
Dessert Parlor                 1939
Delivery                       1791
                              ...
Bakery, Food Court                2
Food Court, Beverage Shop         2
Dessert Parlor, Food Court        2
Dessert Parlor, Kiosk             2
Quick Bites, Kiosk                1
Name: rest_type, Length: 87, dtype: int64
```

In [43]:
```
rest_df = rest_df.value_counts().nlargest(20)
```

In [44]:
```
fig = px.bar(rest_df, x=rest_df.index, y =rest_df)
fig.show()
```





At all, Banglore is known as the tech capital of India,people having busy and modern life will prefer Quick Bites.We can observe tha Quick Bites type restaurants dominates.

In [ ]:

Highest voted restuarant

In [45]:
```
df.head(3)
```

Out[45]:

	url	address	name	online_order	book_table	rate	votes	
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1	775	4229
1	https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1	787	0
2	https://www.zomato.com/SanchurroBangalore?cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8	918	+91

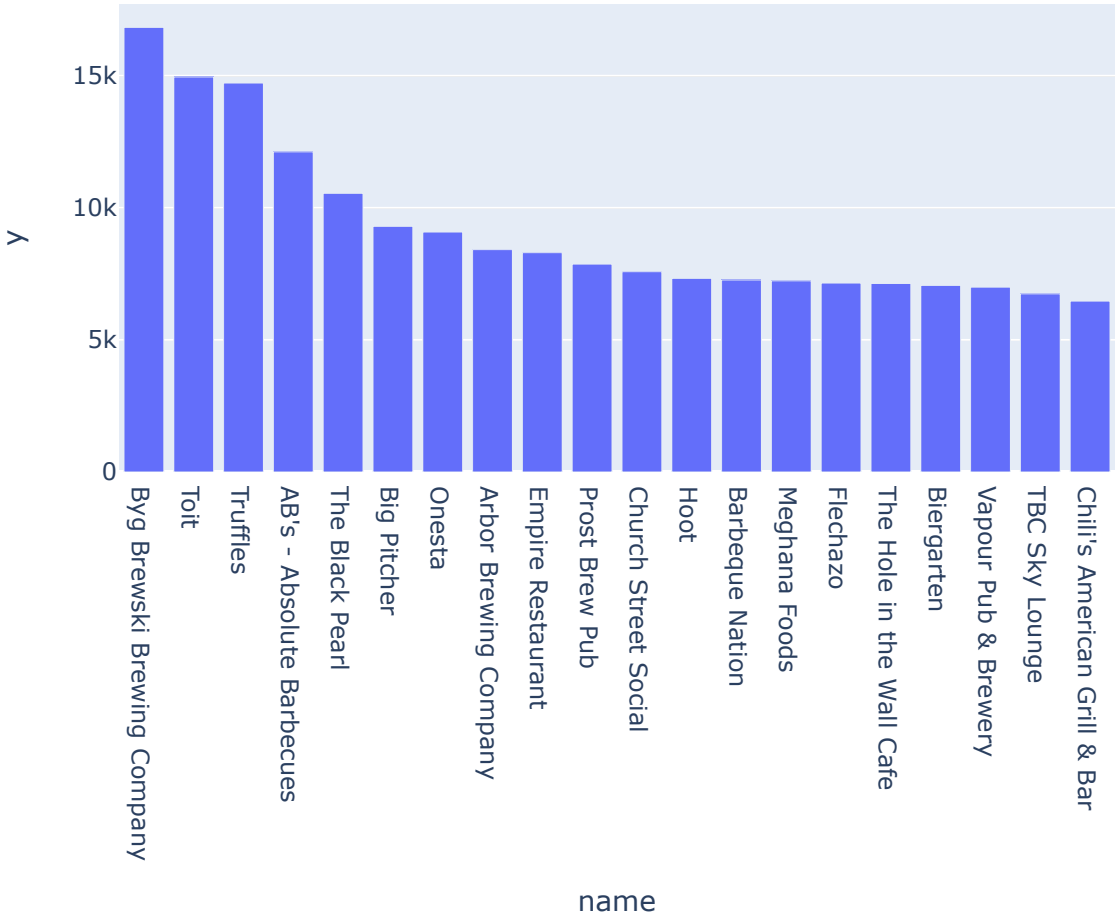
In [46]:
```
df['votes'].isna().sum()
```

Out[46]:
```
0
```

In [47]:
```
votes_df = df.groupby('name')['votes'].max().nlargest(20)
votes_df.head()
```

Out[47]:
```
name
Byg Brewski Brewing Company    16832
Toit                           14956
Truffles                       14726
AB's - Absolute Barbecues      12121
The Black Pearl                10550
Name: votes, dtype: int64
```

In [48]:
```
fig = px.bar(votes_df, x = votes_df.index, y = votes_df)
fig.show()
```



In [ ]:

Total restuarants at different locations of Bengaluru

In [49]:
```
df.head(3)
```

Out[49]:

	url	address	name	online_order	book_table	rate	votes	
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1	775	4229

1	https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1	787	0
2	https://www.zomato.com/SanchurroBangalore?cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8	918	+91

In [50]:
```
df['location'].isna().sum()
```

Out[50]:
```
0
```

In [51]:
```
df['name'].isna().sum()
```

Out[51]:
```
0
```

In [52]:
```
loc_res_df = df.groupby('location')['name'].unique()
loc_res_df
```

Out[52]:
```
location
BTM                    [Sankranthi Veg Restaurant, Hearts Unlock Cafe...
Banashankari           [Jalsa, Spice Elephant, San Churro Cafe, Addhu...
Banaswadi              [Cafe Nibras, The Sanctuary, Crunch Pizzas, Pi...
Bannerghatta Road      [Deja Vu Resto Bar, Fattoush, Empire Restauran...
Basavanagudi           [Grand Village, Timepass Dinner, Srinathji's C...
                                                 ...
West Bangalore         [FreshMenu, Fit Dish Fetish, Garden City Mobil...
Whitefield             [Imperio Cafe, Night Diaries, LocalHost, AB's ...
Wilson Garden          [Tree Top, Sahana's (Nati Style), Karavali Kol...
Yelahanka              [Prashanth Naati Corner, Cheta's Kitchen, Twis...
Yeshwantpur            [Chef's Bank, New Agarwal Bhavan, Fishing Boat...
Name: name, Length: 92, dtype: object
```

In [53]:
```
tr=[]
for i in loc_res_df:
    tr.append(len(i))
```

In [54]:
```
location = []
for i in loc_res_df.index:
    location.append(i)
```

In [55]:
```
location_res_df = pd.DataFrame(zip(location,tr))
location_res_df.columns = ['location','total_restuarants']
location_res_df.set_index('location', inplace = True)
location_res_df.head()
```

Out[55]:

location	total_restuarants
BTM	581
Banashankari	238
Banaswadi	151
Bannerghatta Road	362
Basavanagudi	195

In [56]:
```
location_res_df.nlargest(20,'total_restuarants').sort_values(by ='total_restuarants',asc
```

Out[56]:
```
<pandas.plotting._core.PlotAccessor object at 0x000001F9E6C37DC0>
```

In [ ]:
```

```

In [57]:
```
df.isna().sum()
```

Out[57]:
```
url                                0
address                            0
name                               0
online_order                       0
book_table                         0
rate                               0
votes                              0
phone                            832
location                           0
rest_type                        151
dish_liked                     20333
cuisines                          11
approx_cost(for two people)      252
reviews_list                       0
menu_item                          0
listed_in(type)                    0
listed_in(city)                    0
dtype: int64
```

Different type of restuarants

In [58]:
```
df.head(3)
```

Out[58]:

	url	address	name	online_order	book_table	rate	votes	
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1	775	4229
1	https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1	787	0
2	https://www.zomato.com/SanchurroBangalore?cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8	918	+91

In [59]:
```
rest_type_percent = (df['rest_type'].value_counts()/len(df))*100
rest_type_percent.nlargest(20)
```

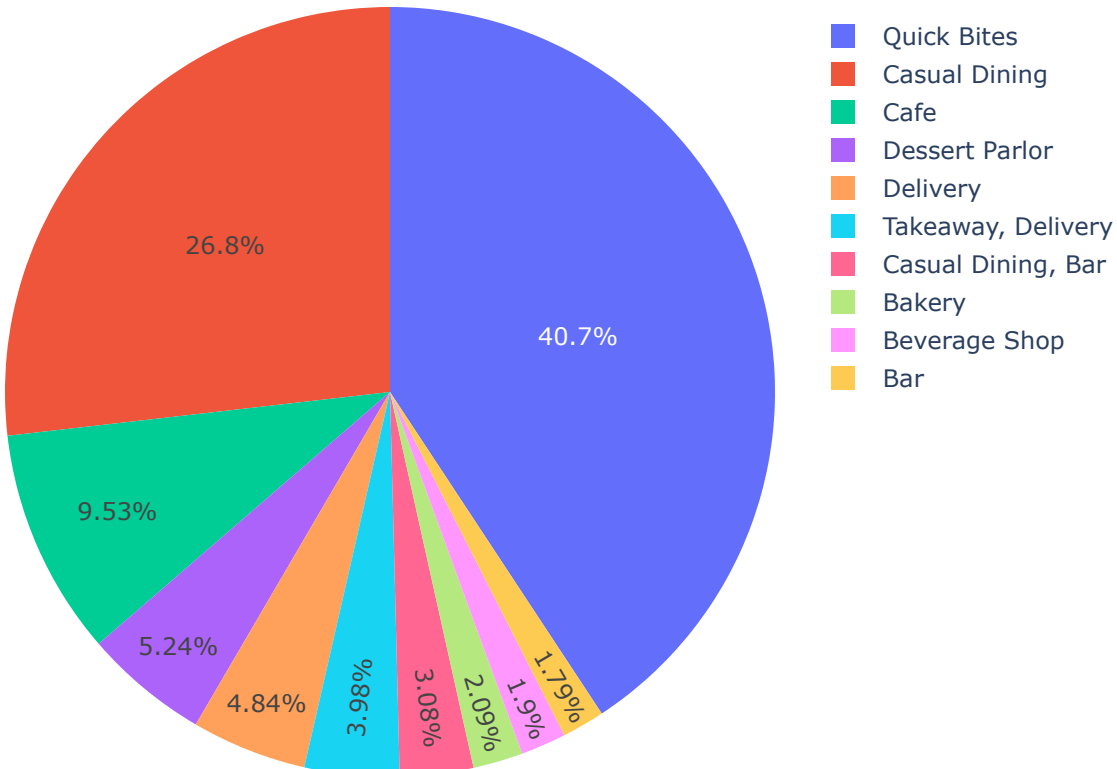
Out[59]:
```
Quick Bites                   34.302035
Casual Dining                 22.582040
Cafe                           8.026489
```

```
Dessert Parlor                 4.412635
Delivery                       4.075827
Takeaway, Delivery             3.349870
Casual Dining, Bar             2.596605
Bakery                         1.763688
Beverage Shop                  1.602112
Bar                            1.504256
Food Court                     1.197032
Bar, Casual Dining             0.955805
Lounge                         0.901188
Pub                            0.810159
Sweet Shop                     0.798780
Fine Dining                    0.780574
Casual Dining, Cafe            0.725957
Pub, Casual Dining             0.580310
Beverage Shop, Quick Bites     0.555277
Bakery, Quick Bites            0.532520
Name: rest_type, dtype: float64
```

In [60]:
```
rest_type_names = []
for i in rest_type_percent.nlargest(10).index:
    rest_type_names.append(i)

rest_type_names
```

Out[60]:
```
['Quick Bites',
 'Casual Dining',
 'Cafe',
 'Dessert Parlor',
 'Delivery',
 'Takeaway, Delivery',
 'Casual Dining, Bar',
 'Bakery',
 'Beverage Shop',
 'Bar']
```

In [61]:
```
fig = px.pie(df,values = rest_type_percent.nlargest(10), names = rest_type_names)
fig.show()
```



Total number of variety of restuarants

In [62]:
```
df.head(3)
```

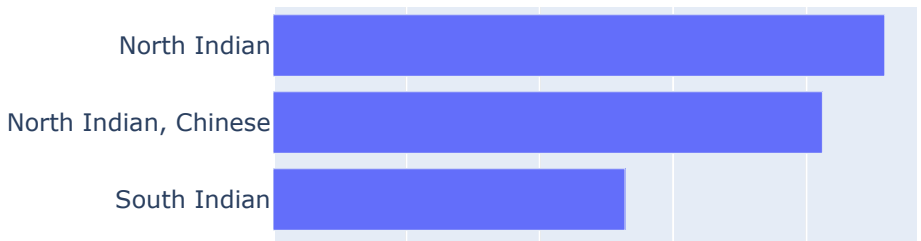
Out[62]:

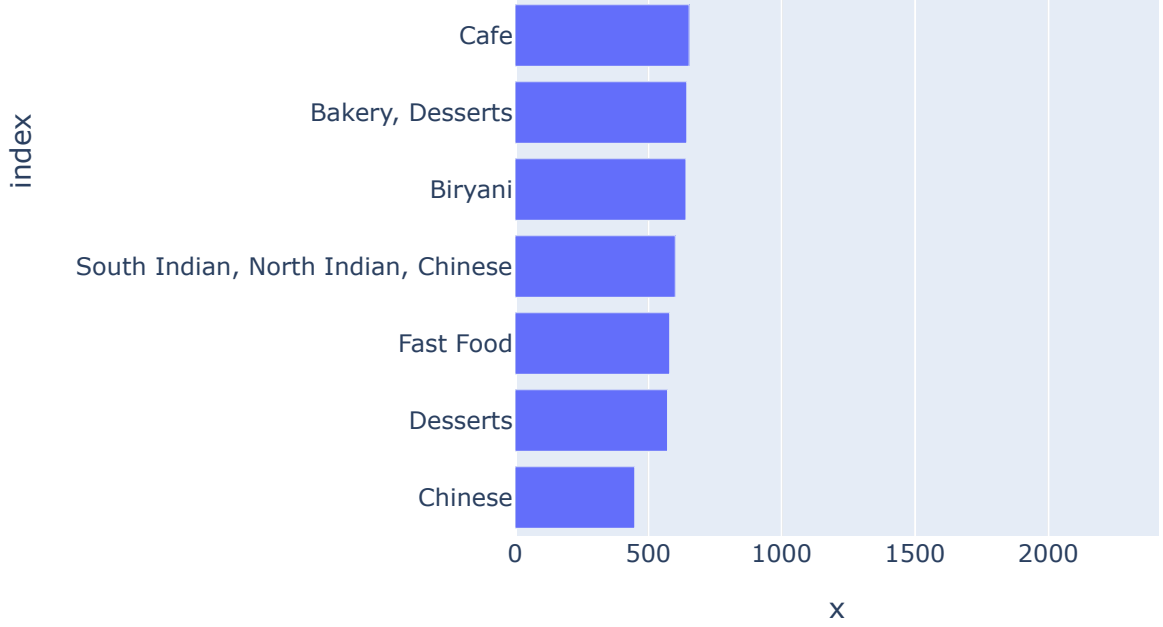
	url	address	name	online_order	book_table	rate	votes	
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1	775	4229
1	https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1	787	0
2	https://www.zomato.com/SanchurroBangalore?cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8	918	+91

In [63]:
```
cuisines_df = df['cuisines'].value_counts().nlargest(10).sort_values(ascending=True)
cuisines_df
```

Out[63]:
```
Chinese                                  449
Desserts                                 572
Fast Food                                580
South Indian, North Indian, Chinese      601
Biryani                                  641
Bakery, Desserts                         644
Cafe                                     653
South Indian                            1320
North Indian, Chinese                   2060
North Indian                            2294
Name: cuisines, dtype: int64
```

In [64]:
```
fig = px.bar(cuisines_df, y = cuisines_df.index , x= cuisines_df)
fig.show()
```





North Indian,chinese,South Indian and Biriyani are most common and Bengaluru is more influenced by North Indian culture than the south

Approx cost for two people

In [65]:
```
df.head(2)
```

Out[65]:

	url	address	name	online_order	book_table	rate	votes	
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1	775	42297555 974
1	https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1	787	080 4

In [66]:
```
df['approx_cost(for two people)'].unique()
```

Out[66]:
```
array(['800', '300', '600', '700', '550', '500', '450', '650', '400',
       '900', '200', '750', '150', '850', '100', '1,200', '350', '250',
       '950', '1,000', '1,500', '1,300', '199', '80', '1,100', '160',
       '1,600', '230', '130', '1,700', '1,400', '1,350', nan, '2,200',
       '2,000', '1,800', '1,900', '180', '330', '2,500', '2,100', '3,000',
       '2,800', '3,400', '50', '40', '1,250', '3,500', '4,000', '2,400',
       '2,600', '1,450', '70', '3,200', '560', '240', '360', '6,000',
       '1,050', '2,300', '4,100', '120', '5,000', '3,700', '1,650',
       '2,700', '4,500'], dtype=object)
```

In [67]:
```
df['approx_cost(for two people)'].isna().sum()
```

Out[67]: 252

In [68]:
```
ac_df = df.dropna(axis = 'index', subset='approx_cost(for two people)')
```

In [69]:
```
ac_df['approx_cost(for two people)'].isna().sum()
```

Out[69]: 0

In [70]:
```
ac_df['approx_cost(for two people)'] = ac_df['approx_cost(for two people)'].apply(lambda
```

In [71]:
```
ac_df['approx_cost(for two people)'].unique()
```

Out[71]:
```
array(['800', '300', '600', '700', '550', '500', '450', '650', '400',
       '900', '200', '750', '150', '850', '100', '1200', '350', '250',
       '950', '1000', '1500', '1300', '199', '80', '1100', '160', '1600',
       '230', '130', '1700', '1400', '1350', '2200', '2000', '1800',
       '1900', '180', '330', '2500', '2100', '3000', '2800', '3400', '50',
       '40', '1250', '3500', '4000', '2400', '2600', '1450', '70', '3200',
       '560', '240', '360', '6000', '1050', '2300', '4100', '120', '5000',
       '3700', '1650', '2700', '4500'], dtype=object)
```

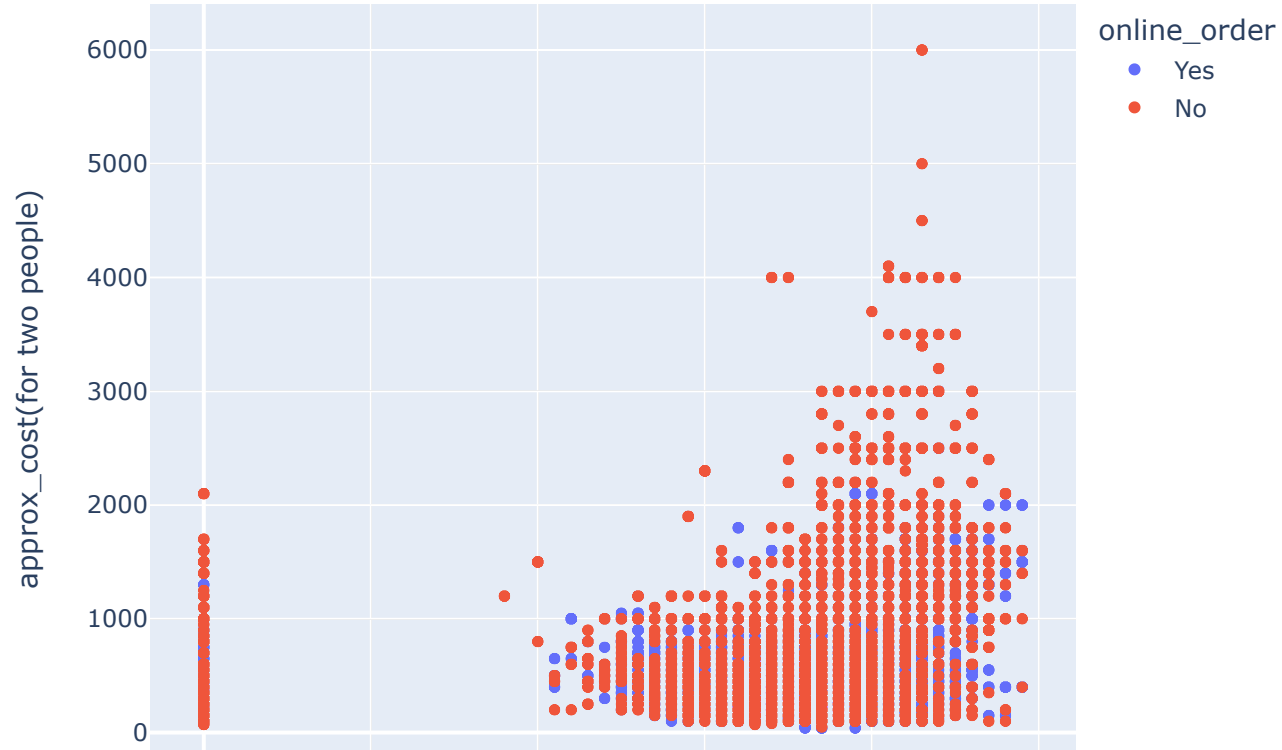
In [72]:
```
ac_df['approx_cost(for two people)'] = ac_df['approx_cost(for two people)'].astype(int)
```

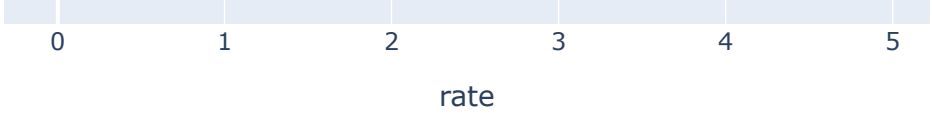
In [73]:
```
ac_df['approx_cost(for two people)'].unique()
```

Out[73]:
```
array([ 800,  300,  600,  700,  550,  500,  450,  650,  400,  900,  200,
        750,  150,  850,  100, 1200,  350,  250,  950, 1000, 1500, 1300,
        199,   80, 1100,  160, 1600,  230,  130, 1700, 1400, 1350, 2200,
       2000, 1800, 1900,  180,  330, 2500, 2100, 3000, 2800, 3400,   50,
         40, 1250, 3500, 4000, 2400, 2600, 1450,   70, 3200,  560,  240,
        360, 6000, 1050, 2300, 4100,  120, 5000, 3700, 1650, 2700, 4500])
```

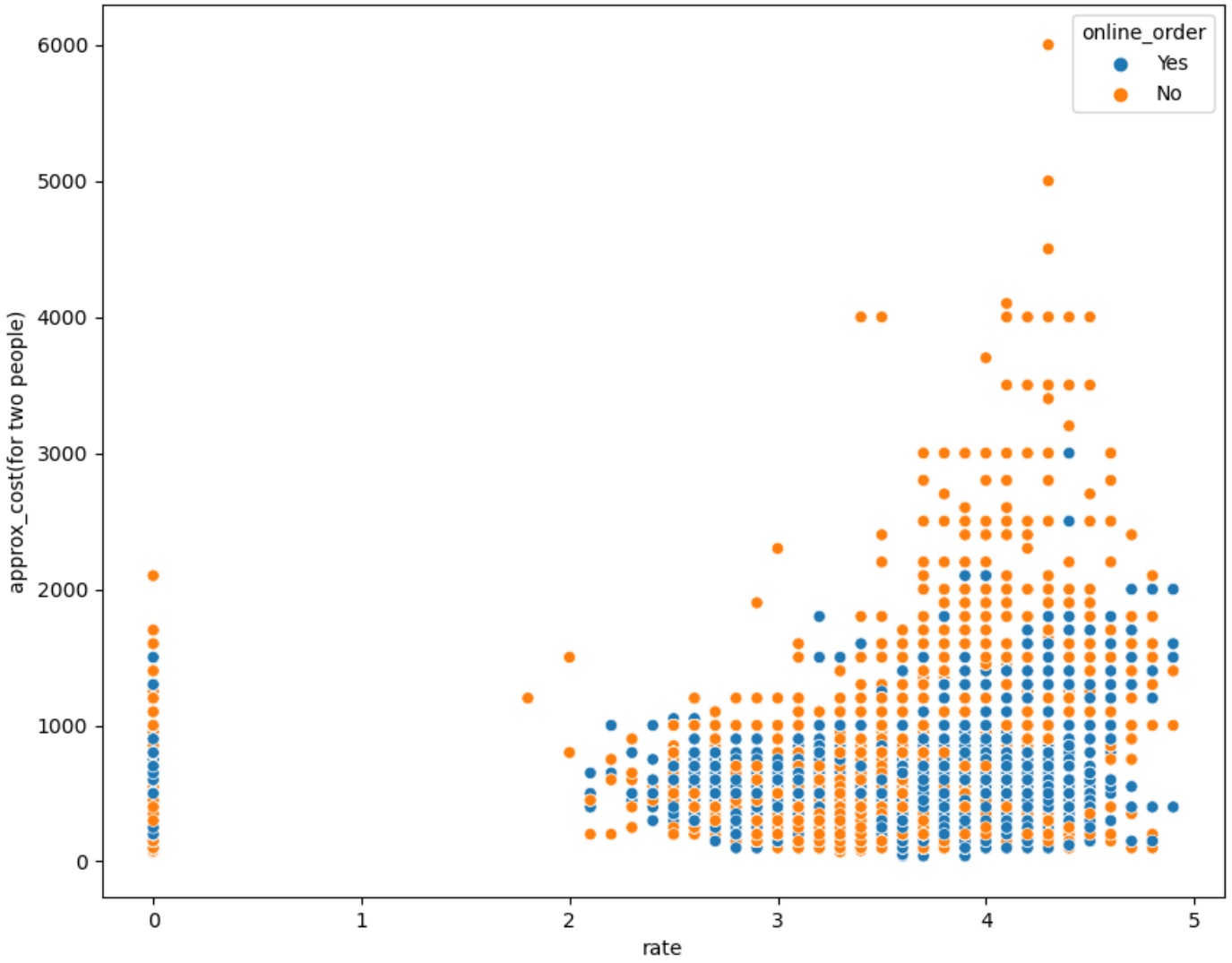
Cost vs Rating

In [74]:
```
fig = px.scatter(ac_df, x = ac_df['rate'], y = ac_df['approx_cost(for two people)'], col
fig.show()
```





In [75]:
```
plt.figure(figsize=(10,8))
sns.scatterplot(x = ac_df['rate'], y = ac_df['approx_cost(for two people)'], hue = ac_df
plt.show()
```



Most of the High rated restuarants accept online order and they are budgeted too

In [76]:
```
df = ac_df.copy()
```

In [77]:
```
df.head(3)
```

Out[77]:

	url	address	name	online_order	book_table	rate	votes	
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1	775	4229
1	https://www.zomato.com/bangalore/spice-	2nd Floor, 80	Spice	Yes	No	4.1	787	0

	elephan...	Feet Road, Near Big Bazaar, 6th ...	Elephant					
2	https://www.zomato.com/SanchurroBangalore?cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8	918	+91

In [ ]:

Difference between votes of restaurants accepting and not accepting online orders

In [78]:
```
df.head(2)
```

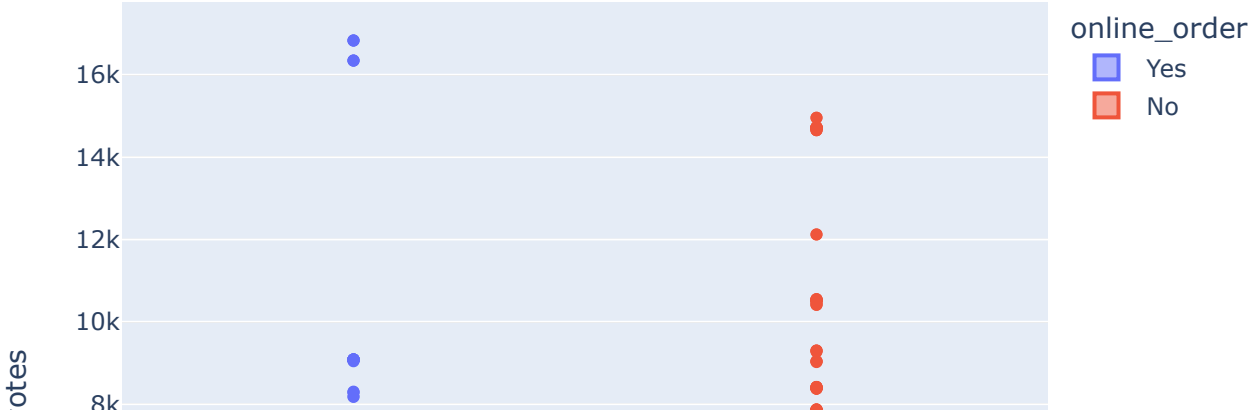
Out[78]:

	url	address	name	online_order	book_table	rate	votes	
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1	775	42297555 974
1	https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1	787	080 4

In [79]:
```
df['votes'].isna().sum()
```

Out[79]:
```
0
```

In [80]:
```
fig = px.box(df, x=df['online_order'], y=df['votes'], color =df['online_order'])
fig.show()
```



Median number of votes for both categories vary. Restaurants accepting online orders get more votes from customers as there is a rating option poping up after each order through zomato application

Difference between price of restaurants accepting and not accepting online orders

In [81]:
```
df.head(3)
```

Out[81]:

	url	address	name	online_order	book_table	rate	votes	
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1	775	4229
1	https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1	787	0
2	https://www.zomato.com/SanchurroBangalore?cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8	918	+91

In [82]:
```
fig = px.box(df, x= df['online_order'], y=df['approx_cost(for two people)'], color=df['o
fig.show()
```

Restaurants accepting online orders are more affordable than Restaurants who are accepting online orders

Cheapest and Most expensive for two people

In [83]:
```
df['approx_cost(for two people)'].min()
```

Out[83]:
```
40
```

In [84]:
```
df['approx_cost(for two people)'].max()
```

Out[84]:
```
6000
```

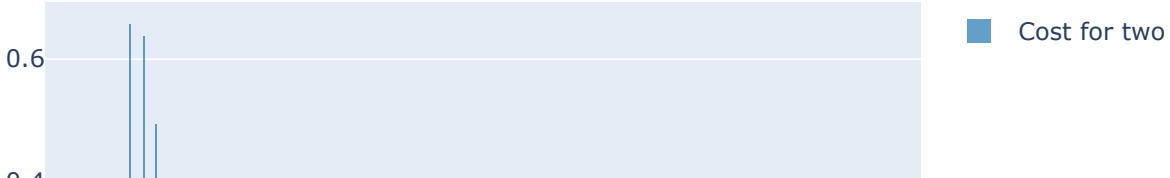
In [85]:
```
df[df['approx_cost(for two people)']==6000]['name']
```

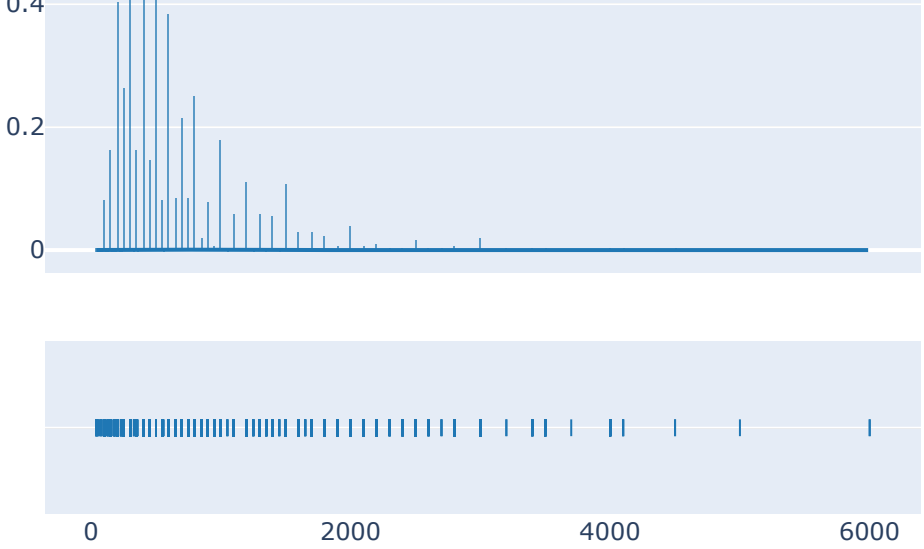
Out[85]:
```
19139    Le Cirque Signature - The Leela Palace
45618    Le Cirque Signature - The Leela Palace
Name: name, dtype: object
```

In [ ]:
```

```

Distribution of cost for two people

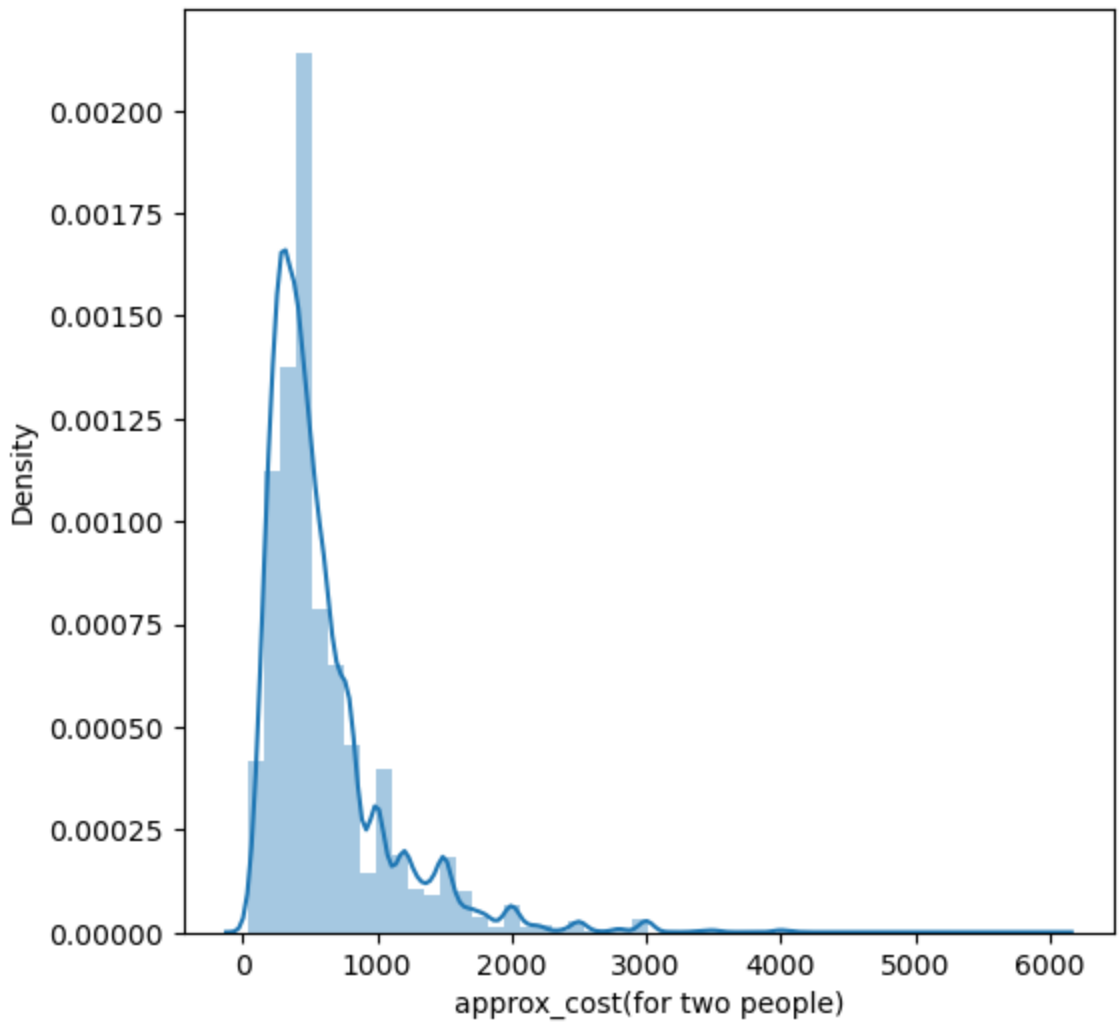
In [86]:
```
fig = ff.create_distplot(hist_data= [df['approx_cost(for two people)'].values.tolist()],
fig.show()
```





In [87]:
```
plt.figure(figsize=(6,6))
sns.distplot(df['approx_cost(for two people)'])
plt.show()
```

```
C:\Users\abhis\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
```



Most of the price lies between in a range of under 1000,it means most are affordable & very few are luxurious

In [ ]:

Most costly Rate for 2 people is served at which Restaurant what exactly is the dish involved in this and liked dish of that restaurant?

In [88]:
```
df.head(1)
```

Out[88]:

	url	address	name	online_order	book_table	rate	votes	ph
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1	775	42297555\r\n 974377

In [89]:
```
df['approx_cost(for two people)'].max()
```

Out[89]:
```
6000
```

In [90]:
```
df[df['approx_cost(for two people)']==6000][['name','cuisines','dish_liked']]
```

Out[90]:

	name	cuisines	dish_liked
19139	Le Cirque Signature - The Leela Palace	French, Italian	Wine, Asparagus Soup, Creme Brulee, Pasta, Rav...
45618	Le Cirque Signature - The Leela Palace	French, Italian	Wine, Asparagus Soup, Creme Brulee, Pasta, Rav...

In [ ]:

Top 10 Most Expensive restaurant with approx cost for 2 people

In [91]:
```
data1 = df.copy()
```

In [92]:
```
data1.dtypes
```

Out[92]:
```
url                            object
address                        object
name                           object
online_order                   object
book_table                     object
rate                          float64
votes                           int64
phone                          object
location                       object
rest_type                      object
dish_liked                     object
```

```
cuisines                        object
approx_cost(for two people)      int32
reviews_list                    object
menu_item                       object
listed_in(type)                 object
listed_in(city)                 object
dtype: object
```

In [93]:
```
eac = data1.groupby('approx_cost(for two people)')['name'].unique()[::-1][0:10]
eac
```

Out[93]:
```
approx_cost(for two people)
6000              [Le Cirque Signature - The Leela Palace]
5000                          [Royal Afghan - ITC Windsor]
4500                              [Malties - Radisson Blu]
4100                          [La Brasserie - Le Meridien]
4000    [Alba - JW Marriott Bengaluru, Edo Restaurant ...
3700                 [Baluchi - The Lalit Ashok Bangalore]
3500    [Karavalli - The Gateway Hotel, Kebabs & Kurri...
3400    [The Market - The Ritz-Carlton, b CafÃ□Â□Ã□Â□Ã...
3200                       [Blue Ginger - The Taj West End]
3000    [Rim Naam - The Oberoi, Le Jardin - The Oberoi...
Name: name, dtype: object
```

In [104…
```
main_df = df.copy()
```

In [105…
```
df.set_index('location',inplace=True)
```

In [106…
```
df['approx_cost(for two people)'].sort_values(ascending=False)[0:10]
```

Out[106]:
```
location
Old Airport Road    6000
Old Airport Road    6000
Sankey Road         5000
Marathahalli        4500
Marathahalli        4500
Sankey Road         4100
Sankey Road         4100
Sankey Road         4100
Sankey Road         4100
Residency Road      4000
Name: approx_cost(for two people), dtype: int32
```

All the restautant that are below than 500(budget hotel)

In [107…
```
df_budget = df[df['approx_cost(for two people)']<=500]['approx_cost(for two people)'].to
```

In [108…
```
df_budget.reset_index(inplace= True)
```

In [109…
```
df_budget
```

Out[109]:

	location	approx_cost(for two people)
0	Banashankari	300
1	Banashankari	500
2	Banashankari	500
3	Banashankari	450

4	Banashankari	300
...	...	...
26325	Brookefield	200
26326	KR Puram	400
26327	Brookefield	300
26328	KR Puram	400
26329	Brookefield	300

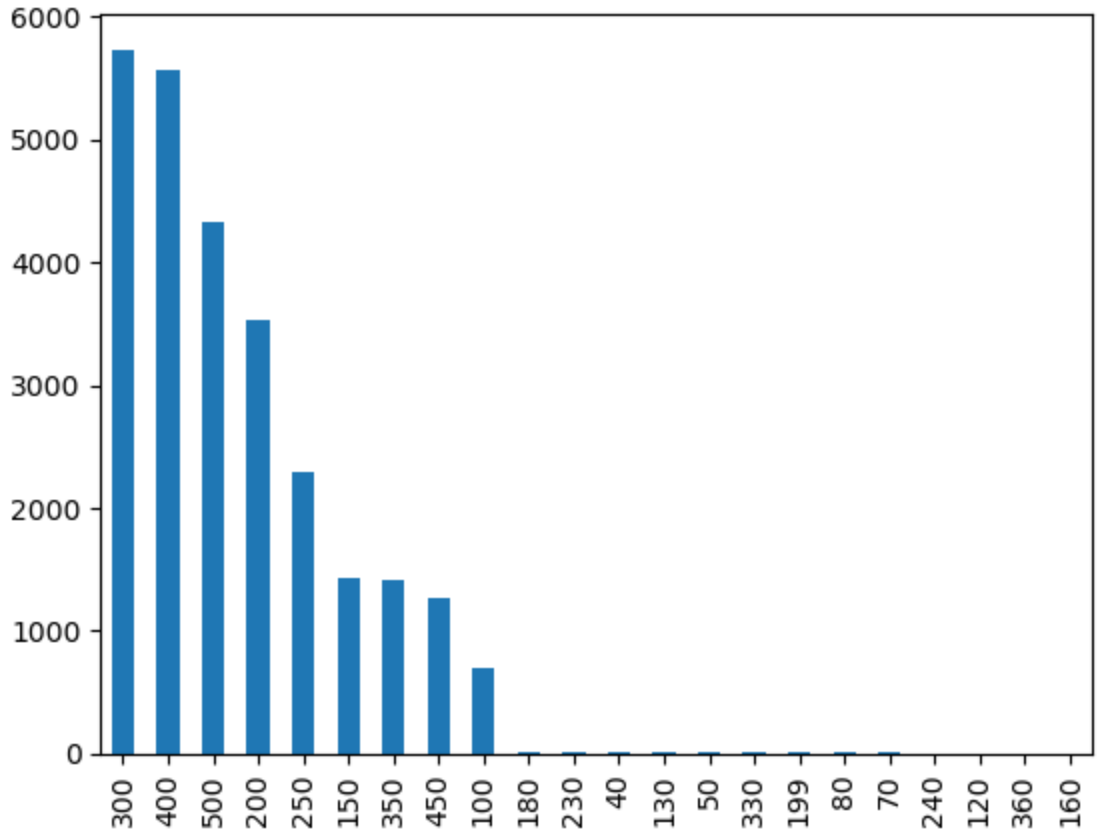
26330 rows × 2 columns

In [110…
```
df_budget2 = df_budget['approx_cost(for two people)'].value_counts()
df_budget2
```

Out[110]:
```
300    5735
400    5562
500    4326
200    3527
250    2293
150    1432
350    1416
450    1267
100     702
180      17
230      10
40        8
130       8
50        6
330       4
199       4
80        4
70        3
240       2
120       2
360       1
160       1
Name: approx_cost(for two people), dtype: int64
```

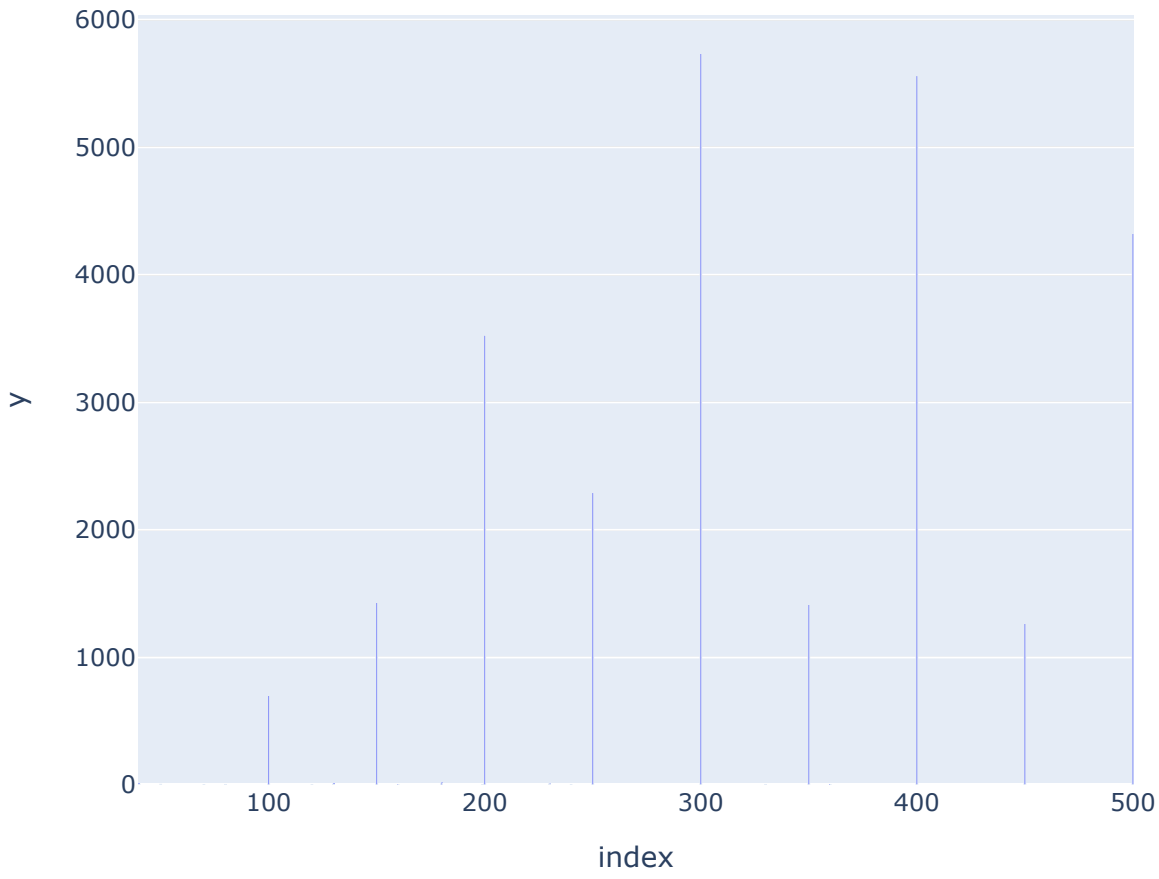
In [111…
```
df_budget2.plot.bar()
```

Out[111]:
```
<AxesSubplot:>
```



In [112…

```
fig = px.bar(df_budget2, x=df_budget2.index, y =df_budget2)
fig.show()
```



In [ ]:

Restaurants that have better rating >4 and that are under budget too

In [113…
```
df_new = df[(df['rate']>=4) & (df['approx_cost(for two people)']<=500)]
```

In [114…
```
len(df_new['name'].unique())
```

Out[114]:
```
628
```

Total such various affordable hotels at different location

In [115…
```
df_new.head()
```

Out[115]:

location	url	address	
Banashankari	https://www.zomato.com/bangalore/caf%C3%A9-dow...	12,29 Near PES University Back Gate, D'Souza N...	CafÃ□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Ã□ Dow
Banashankari	https://www.zomato.com/bangalore/the-coffee-sh...	6th Block, 3rd Stage, Banashankari, Bangalore	The Cof
Banashankari	https://www.zomato.com/bangalore/caf-eleven-ba...	111, Sapphire Toys Building, 100 Feet Ring Roa...	C
Banashankari	https://www.zomato.com/bangalore/faasos-banash...	80, BDA Complex, 2nd Stage, Banashankari, Bang...	
Banashankari	https://www.zomato.com/bangalore/maruthi-davan...	556/A, Water Tank Road, 3rd Stage, Banashankar...	Maruthi Davangere Be

In [116…
```
loc_total = df_new.groupby('location')['name'].unique()
loc_total
```

Out[116]:
```
location
BTM                                    [eat.fit, Hiyar Majhe Kolkata, XO Belgian Waff...
Banashankari                           [CafÃ□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â© Down The ...
Banaswadi                              [Pooja Dosa Camp, Corner House Ice Cream, The ...
Bannerghatta Road                      [BOX8- Desi Meals, Krishna Kuteera, Krishna Ku...
Basavanagudi                           [Kabab Magic, Sri Guru Kottureshwara Davangere...
                                                                   ...
Varthur Main Road, Whitefield          [Easy Bites, Polar Bear, Chaat Street, Vegetarea]
Vasanth Nagar                          [Bangalore Masala House, Step Right Up, Belgia...
Vijay Nagar                            [Shree Cool Point, Sri Ayodhya Veg, Al-Sidique...
Whitefield                             [Roti Ghar Ki, BOX8- Desi Meals, Warm Oven, Th...
Yeshwantpur                            [New Agarwal Bhavan, Fishing Boat, Bhavani Vad...
Name: name, Length: 68, dtype: object
```

In [117…
```
location = []
total = []
for loc in loc_total.index:
    location.append(loc)
for tot in loc_total:
    total.append(len(tot))
```

In [118…
```
loc_total = pd.DataFrame(zip(location,total))
```

In [119…
```
loc_total.columns=['location','total']
```

In [120…
```
loc_total = loc_total.set_index('location')
```

In [121…
```
loc_total
```

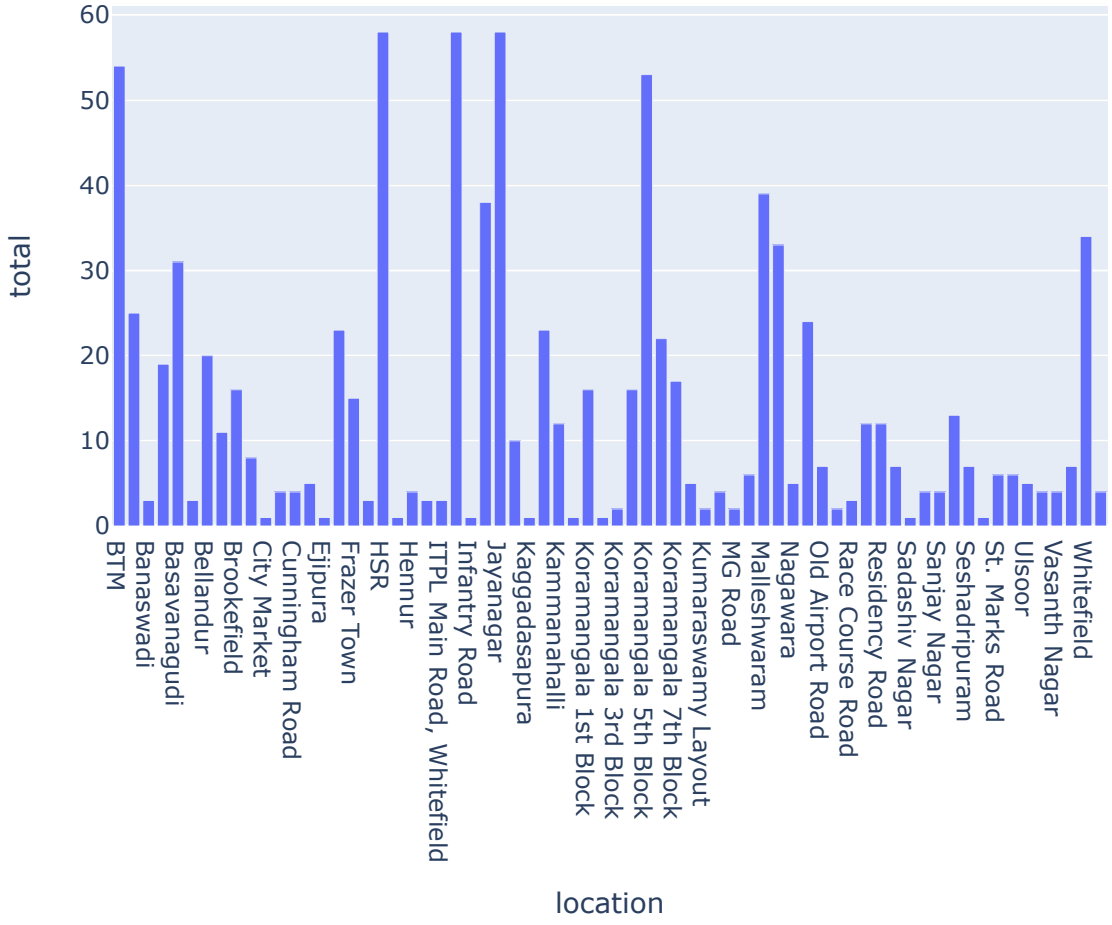
Out[121]:

location	total
BTM	54
Banashankari	25
Banaswadi	3
Bannerghatta Road	19
Basavanagudi	31
...	...
Varthur Main Road, Whitefield	4
Vasanth Nagar	4
Vijay Nagar	7
Whitefield	34
Yeshwantpur	4

68 rows × 1 columns

In [122…
```
loc_total1=loc_total.nlargest(10,'total')
```

In [123…
```
fig = px.bar(loc_total, y=loc_total['total'], x=loc_total.index)
fig.show()
```



In [ ]:

Geographic Analysis

In [127…
```
df = main_df.copy()
```

In [128…
```
df.shape
```

Out[128]:
```
(43690, 17)
```

In [129…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 43690 entries, 0 to 51716
Data columns (total 17 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   url                          43690 non-null  object 
 1   address                      43690 non-null  object 
 2   name                         43690 non-null  object 
 3   online_order                 43690 non-null  object 
 4   book_table                   43690 non-null  object 
 5   rate                         43690 non-null  float64
 6   votes                        43690 non-null  int64  
 7   phone                        43099 non-null  object 
 8   location                     43690 non-null  object 
 9   rest_type                    43541 non-null  object 
 10  dish_liked                   23474 non-null  object 
 11  cuisines                     43682 non-null  object 
 12  approx_cost(for two people)  43690 non-null  int32  
 13  reviews_list                 43690 non-null  object 
```

```
 14  menu_item                    43690 non-null  object
 15  listed_in(type)              43690 non-null  object
 16  listed_in(city)              43690 non-null  object
dtypes: float64(1), int32(1), int64(1), object(14)
memory usage: 5.8+ MB
```

In [ ]:

In [130…
```
locations = pd.DataFrame({'name':df['location'].unique()})
```

In [131…
```
locations.head()
```

Out[131]:

	name
0	Banashankari
1	Basavanagudi
2	Mysore Road
3	Jayanagar
4	Kumaraswamy Layout

In [132…
```
!pip install geopy
```

```
Requirement already satisfied: geopy in c:\users\abhis\anaconda3\lib\site-packages (2.3.
0)
Requirement already satisfied: geographiclib<3,>=1.52 in c:\users\abhis\anaconda3\lib\si
te-packages (from geopy) (2.0)
```

In [133…
```
from geopy.geocoders import Nominatim
```

In [134…
```
lat=[]
lon=[]
geolocator = Nominatim(user_agent='app')
for location in locations['name']:
    loc = geolocator.geocode(location)
    if loc is None:
        lat.append(np.nan)
        lon.append(np.nan)
    else:
        lat.append(loc.latitude)
        lon.append(loc.longitude)
```

In [135…
```
locations['lattitude'] = lat
locations['longitude'] = lon
```

In [136…
```
locations.head()
```

Out[136]:

	name	lattitude	longitude
0	Banashankari	15.887678	75.704678
1	Basavanagudi	12.941726	77.575502
2	Mysore Road	12.946662	77.530090
3	Jayanagar	27.643927	83.052805
4	Kumaraswamy Layout	12.908149	77.555318

In [137…
```
total = pd.DataFrame(df['location'].value_counts())
```

In [138…
```
total.reset_index(inplace=True)
```

In [139…
```
total.columns = ['name','total']
```

In [140…
```
total.head()
```

Out[140]:

	name	total
0	BTM	4237
1	Koramangala 5th Block	2358
2	HSR	2113
3	Indiranagar	1892
4	JP Nagar	1849

In [141…
```
total.shape
```

Out[141]:
```
(92, 2)
```

In [142…
```
locations.shape
```

Out[142]:
```
(92, 3)
```

In [143…
```
total_loc=total.merge(locations,on='name',how='left')
```

In [144…
```
total_loc.head()
```

Out[144]:

	name	total	lattitude	longitude
0	BTM	4237	45.954851	-112.496595
1	Koramangala 5th Block	2358	12.934843	77.618977
2	HSR	2113	18.147500	41.538889
3	Indiranagar	1892	12.973291	77.640467
4	JP Nagar	1849	12.265594	76.646540

In [145…
```
total_loc.head()
```

Out[145]:

	name	total	lattitude	longitude
0	BTM	4237	45.954851	-112.496595
1	Koramangala 5th Block	2358	12.934843	77.618977
2	HSR	2113	18.147500	41.538889
3	Indiranagar	1892	12.973291	77.640467
4	JP Nagar	1849	12.265594	76.646540

In [146…
```
total_loc.dropna(inplace=True)
```

In [147…
```
!pip install folium
```

```
Requirement already satisfied: folium in c:\users\abhis\anaconda3\lib\site-packages (0.1
4.0)
```

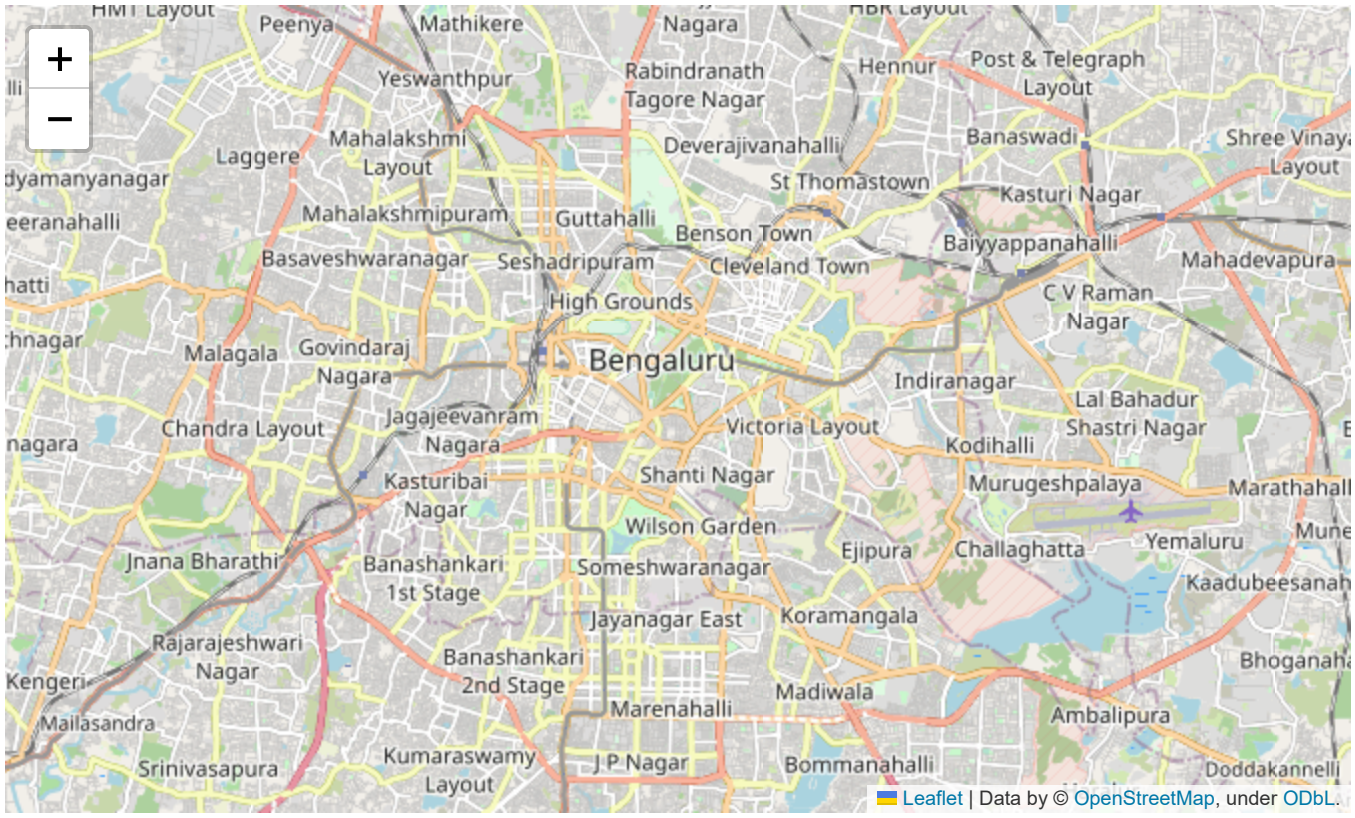
```
Requirement already satisfied: jinja2>=2.9 in c:\users\abhis\anaconda3\lib\site-packages
(from folium) (2.11.3)
Requirement already satisfied: numpy in c:\users\abhis\anaconda3\lib\site-packages (from
folium) (1.21.5)
Requirement already satisfied: requests in c:\users\abhis\anaconda3\lib\site-packages (f
rom folium) (2.28.1)
Requirement already satisfied: branca>=0.6.0 in c:\users\abhis\anaconda3\lib\site-packag
es (from folium) (0.6.0)
Requirement already satisfied: MarkupSafe>=0.23 in c:\users\abhis\anaconda3\lib\site-pac
kages (from jinja2>=2.9->folium) (2.0.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\abhis\anaconda3\lib\site-p
ackages (from requests->folium) (2022.9.14)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\abhis\anaconda3\lib\sit
e-packages (from requests->folium) (1.26.11)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\abhis\anaconda3\lib
\site-packages (from requests->folium) (2.0.4)
Requirement already satisfied: idna<4,>=2.5 in c:\users\abhis\anaconda3\lib\site-package
s (from requests->folium) (3.3)
```

In [148…]
```
def generateheatmap(default_location=[12.97,77.59], default_zoom_start=12):
    base_map = folium.Map(location=default_location, zoom_start=default_zoom_start)
    return base_map
```

In [149…]
```
import folium
from folium.plugins import HeatMap
```

In [150…]
```
basemap = generateheatmap()
basemap
```

Out[150]:



In [151…]
```
total_loc[['lattitude','longitude','total']].values.tolist()
```

Out[151]:
```
[[45.95485055, -112.49659530324134, 4237.0],
 [12.9348429, 77.6189768, 2358.0],
 [18.1475, 41.538889, 2113.0],
 [12.9732913, 77.6404672, 1892.0],
 [12.2655944, 76.6465404, 1849.0],
 [27.64392675, 83.05280519687284, 1711.0],
 [53.5533682, -2.2969019, 1688.0],
 [12.9552572, 77.6984163, 1488.0],
```

```
[12.9521804, 77.6041899, 1318.0],
[12.9390255, 77.6238477, 1103.0],
[12.9736132, 77.6074716, 1084.0],
[12.9364846, 77.6134783, 1084.0],
[12.93103185, 77.6782471, 1078.0],
[12.9277245, 77.6327822, 964.0],
[15.67509025, 73.81083634836561, 963.0],
[12.920441, 77.6653284, 913.0],
[12.9778793, 77.6246697, 901.0],
[12.9327778, 77.6294052, 864.0],
[12.9755264, 77.6067902, 818.0],
[15.8876779, 75.7046777, 805.0],
[13.0221416, 77.6403368, 745.0],
[13.0027353, 77.5703253, 657.0],
[12.9417261, 77.5755021, 628.0],
[40.5758225, -74.123824, 626.0],
[32.729455, 74.8708533, 607.0],
[33.5935063, -79.0345627, 579.0],
[12.9986827, 77.615525, 574.0],
[40.7165743, -74.0066546, 550.0],
[13.0093455, 77.6377094, 522.0],
[13.0170461, 77.5726933, 521.0],
[40.7652844, -76.373824, 493.0],
[13.0141618, 77.6518539, 491.0],
[12.9882338, 77.554883, 487.0],
[36.5348643, -79.0905056, 475.0],
[12.9624669, 77.6381958, 426.0],
[22.478459, 88.3541291, 390.0],
[40.1460436, -90.5382275, 383.0],
[54.1339384, -4.6097098, 343.0],
[51.5153811, -0.0719678, 309.0],
[12.945245, 77.6269144, 292.0],
[18.532248199999998, 73.84990967997429, 277.0],
[12.9678074, 77.6568367, 262.0],
[12.988721250000001, 77.58516877601824, 252.0],
[12.9408685, 77.617338, 227.0],
[12.9489339, 77.5968273, 203.0],
[12.9282918, 77.6254034, 193.0],
[12.9081487, 77.5553179, 168.0],
[13.2227, 78.5541977, 165.0],
[12.9089453, 77.6239038, 157.0],
[12.973936, 77.6509982, 152.0],
[12.9931876, 77.5753419, 142.0],
[12.2949193, 76.6155101, 141.0],
[30.314117, -89.80857, 140.0],
[52.7678265, 0.2986572, 137.0],
[13.0258087, 77.6305067, 129.0],
[1.2847055, 103.84320655721689, 118.0],
[13.02383, 77.5529215, 112.0],
[13.0358698, 77.6323597, 111.0],
[12.9243692, 77.6242433, 92.0],
[12.9846713, 77.6790908, 91.0],
[12.92613325, 77.57848796011083, 91.0],
[12.9413238, 77.7471103, 91.0],
[12.9862452, 77.731685, 84.0],
[39.76880625, -86.15345077251979, 77.0],
[12.7687114, 77.78836, 74.0],
[18.5384853, 75.5569267, 69.0],
[17.2510682, 80.1651978, 66.0],
[13.0227204, 77.595715, 63.0],
[15.8782951, 74.5084834, 58.0],
[23.1485712, 81.6048241, 50.0],
[18.4900796, 73.8475301, 47.0],
[12.9340114, 77.6222304, 31.0],
[50.84612745, 0.50281715, 30.0],
[12.957998, 77.6037312, 27.0],
```

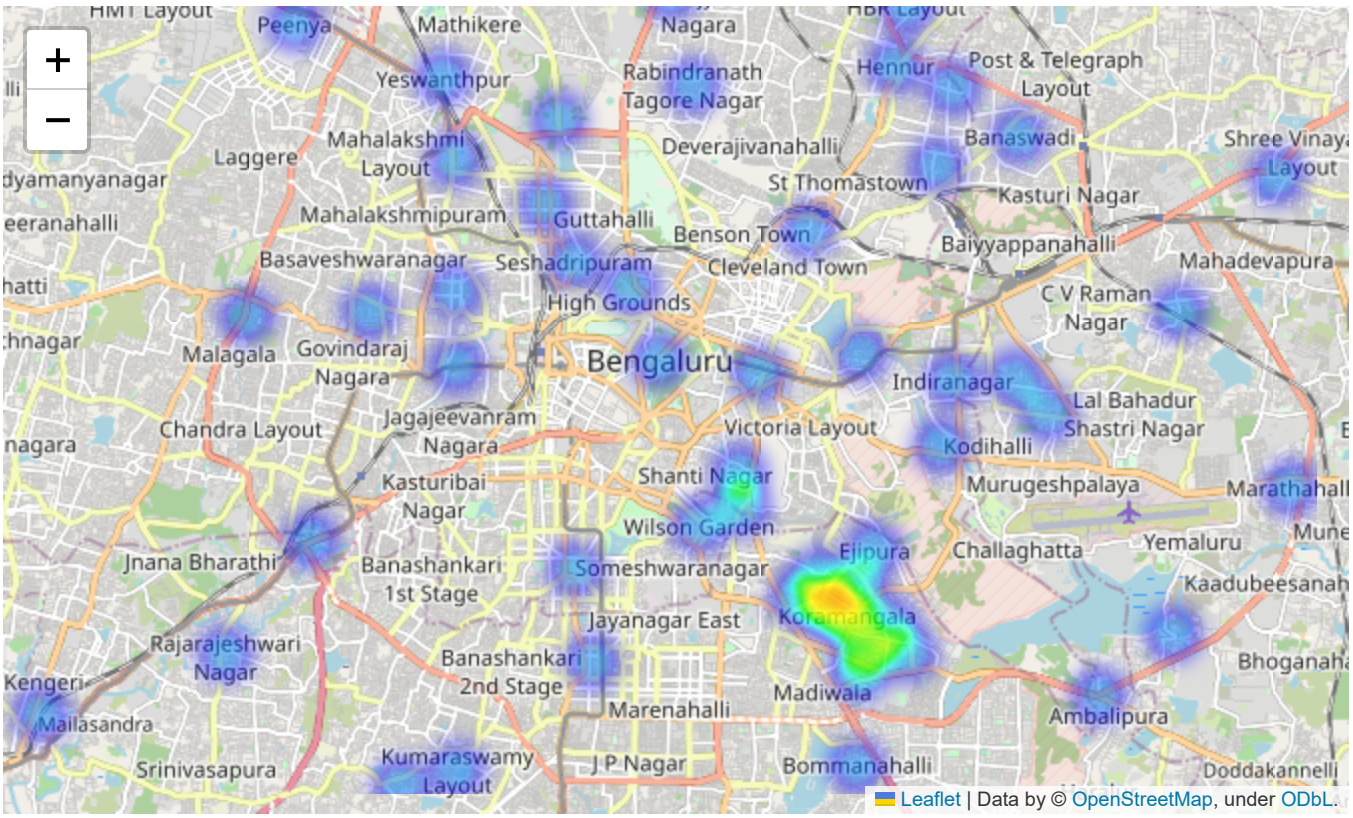
```
[38.7801076, -121.5056438, 26.0],
[12.9756527, 77.5553548, 24.0],
[13.0464531, 77.5483803, 23.0],
[13.0227657, 77.7140616, 22.0],
[12.5442176, 77.4213833, 19.0],
[12.9466619, 77.5300896, 18.0],
[12.9767936, 77.590082, 14.0],
[13.007516, 77.695935, 11.0],
[13.0382184, 77.5919, 10.0],
[12.9176571, 77.4837568, 9.0],
[12.9055682, 77.5455438, 9.0],
[13.0101286, 77.5548006, 8.0],
[12.9848519, 77.5400626, 5.0],
[13.1006982, 77.5963454, 4.0],
[12.9274413, 77.5155224, 2.0],
[12.9841958, 77.51919296908994, 1.0],
[13.0329419, 77.5273253, 1.0]]
```

In [152…
```
HeatMap(total_loc[['lattitude','longitude','total']].values.tolist(),zoom=22,radius=15).
```

Out[152]:
```
<folium.plugins.heat_map.HeatMap at 0x1f9eb2c8c40>
```

In [153…
```
basemap
```

Out[153]:



In [ ]:
```
```

Heatmap of North Indian Restuarants

In [154…
```
ni_df = main_df[main_df['cuisines']=='North Indian'].copy()
```

In [155…
```
ni_df.shape
```

Out[155]:
```
(2254, 17)
```

In [156…
```
ni_df.isna().sum()
```

Out[156]:

```
url                            0
address                        0
name                           0
online_order                   0
book_table                     0
rate                           0
votes                          0
phone                         51
location                       0
rest_type                     10
dish_liked                  1089
cuisines                       0
approx_cost(for two people)    0
reviews_list                   0
menu_item                      0
listed_in(type)                0
listed_in(city)                0
dtype: int64
```

In [157…

```
ni_df.head(2)
```

Out[157]:

	url	address	name	online_order	book_table	rate	votes	
5	https://www.zomato.com/bangalore/timepass-dinn...	37, 5-1, 4th Floor, Bosco Court, Gandhi Bazaar...	Timepass Dinner	Yes	No	3.8	286	9980040 9
50	https://www.zomato.com/bangalore/petoo-banasha...	276, Ground Floor, 100 Feet Outer Ring Road, B...	Petoo	No	No	3.7	21	+91 8

In [158…

```
locations = pd.DataFrame({'name': ni_df['location'].unique()})
```

In [159…

```
locations.head()
```

Out[159]:

	name
0	Basavanagudi
1	Banashankari
2	Kumaraswamy Layout
3	Jayanagar
4	JP Nagar

In [160…

```
lat = []
lon = []
geolocator = Nominatim(user_agent='app')
for location in locations['name']:
    loc = geolocator.geocode(location)
    if loc is None:
```

```
        lat.append(np.nan)
        lon.append(np.nan)
    else:
        lat.append(loc.latitude)
        lon.append(loc.longitude)
```

In [161…
```
locations['lat'] = lat
locations['lon'] = lon
```

In [162…
```
locations.head()
```

Out[162]:

	name	lat	lon
0	Basavanagudi	12.941726	77.575502
1	Banashankari	15.887678	75.704678
2	Kumaraswamy Layout	12.908149	77.555318
3	Jayanagar	27.643927	83.052805
4	JP Nagar	12.265594	76.646540

In [163…
```
total = ni_df['location'].value_counts().to_frame().reset_index()
```

In [164…
```
total.columns = ['name','total']
```

In [165…
```
total.head()
```

Out[165]:

	name	total
0	BTM	262
1	Whitefield	146
2	JP Nagar	135
3	Bellandur	131
4	HSR	117

In [166…
```
north_india = total.merge(locations,on='name',how='left')
```

In [167…
```
north_india.isna().sum()
```

Out[167]:
```
name     0
total    0
lat      0
lon      0
dtype: int64
```

In [168…
```
north_india
```

Out[168]:

	name	total	lat	lon
0	BTM	262	45.954851	-112.496595
1	Whitefield	146	53.553368	-2.296902
2	JP Nagar	135	12.265594	76.646540
3	Bellandur	131	12.931032	77.678247
4	HSR	117	18.147500	41.538889

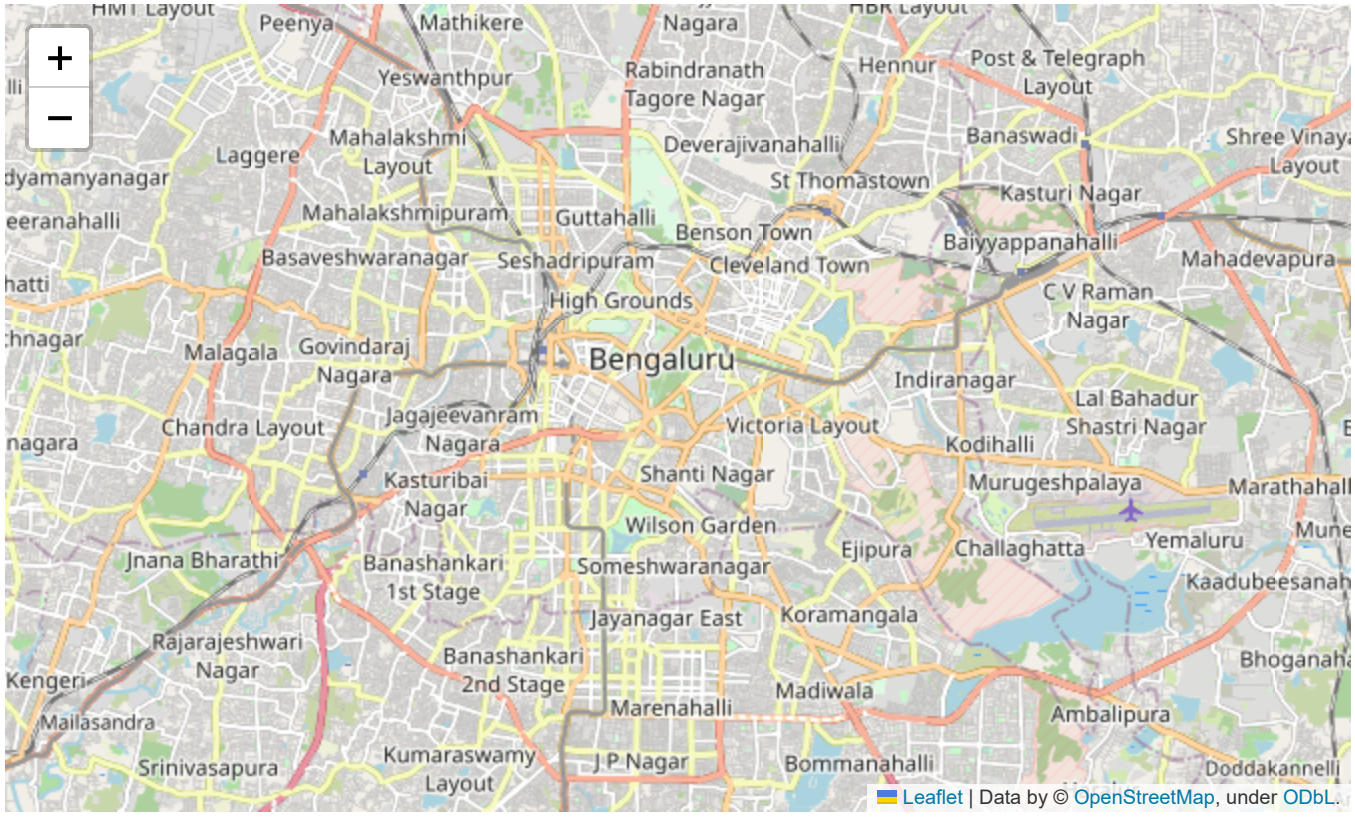
...	...	...	...	...
58	Hennur	2	13.025809	77.630507
59	East Bangalore	2	50.846127	0.502817
60	RT Nagar	2	13.022720	77.595715
61	North Bangalore	1	12.976794	77.590082
62	Basaveshwara Nagar	1	12.294919	76.615510

63 rows × 4 columns

In [169…
```
import folium
from folium.plugins import HeatMap
```

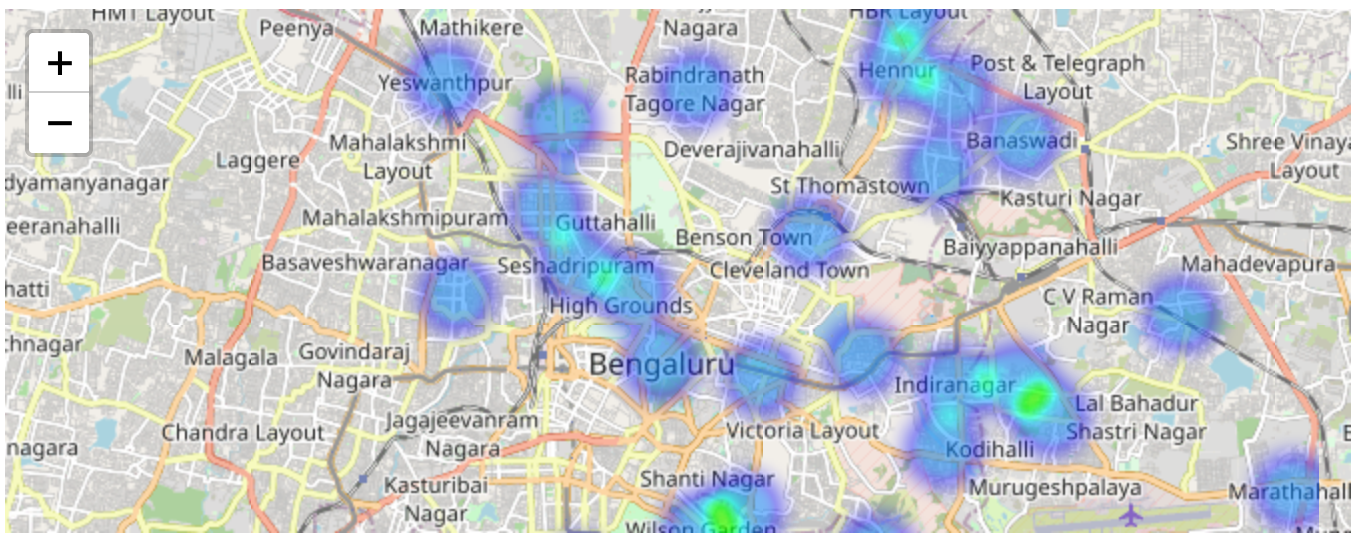
In [170…
```
basemap = generateheatmap()
basemap
```

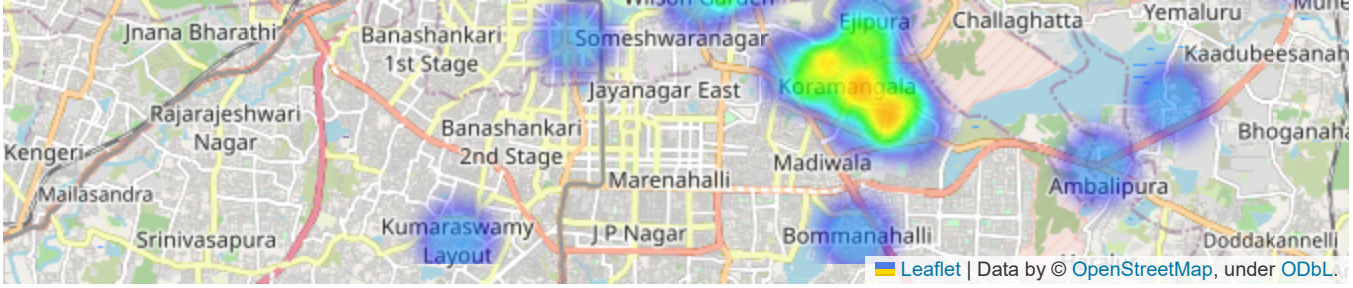
Out[170]:



In [171…
```
HeatMap(north_india[['lat','lon','total']].values.tolist(),zoom=25, radius=20).add_to(ba
basemap
```

Out[171]:





In [ ]:

South Indian Heatmap

In [172…
```
main_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 43690 entries, 0 to 51716
Data columns (total 17 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   url                          43690 non-null  object 
 1   address                      43690 non-null  object 
 2   name                         43690 non-null  object 
 3   online_order                 43690 non-null  object 
 4   book_table                   43690 non-null  object 
 5   rate                         43690 non-null  float64
 6   votes                        43690 non-null  int64  
 7   phone                        43099 non-null  object 
 8   location                     43690 non-null  object 
 9   rest_type                    43541 non-null  object 
 10  dish_liked                   23474 non-null  object 
 11  cuisines                     43682 non-null  object 
 12  approx_cost(for two people)  43690 non-null  int32  
 13  reviews_list                 43690 non-null  object 
 14  menu_item                    43690 non-null  object 
 15  listed_in(type)              43690 non-null  object 
 16  listed_in(city)              43690 non-null  object 
dtypes: float64(1), int32(1), int64(1), object(14)
memory usage: 5.8+ MB
```

In [173…
```
main_df['cuisines'].value_counts()
```

Out[173]:
```
North Indian                                                          2254
North Indian, Chinese                                                 2055
South Indian                                                          1318
Bakery, Desserts                                                       642
Biryani                                                                641
                                                                      ...
Chinese, Bakery                                                          1
South Indian, Fast Food, Chinese                                         1
Asian, Continental, Healthy Food                                         1
South Indian, North Indian, Chinese, Desserts, Fast Food, Ice Cream      1
North Indian, Chinese, Arabian, Momos                                    1
Name: cuisines, Length: 2495, dtype: int64
```

In [174…
```
si_df = main_df[main_df['cuisines']=='South Indian']
```

In [175…
```
si_df.head(3)
```

Out[175]:

	url	address	name	online_order	book_table	rate	votes

39	https://www.zomato.com/bangalore/maruthi-davan...	556/A, Water Tank Road, 3rd Stage, Banashankar...	Maruthi Davangere Benne Dosa	Yes	No	4.0	17	
41	https://www.zomato.com/bangalore/havyaka-mess-...	Opp Vinayaka hospital, Near Seetha Circle , 80...	Havyaka Mess	No	No	3.9	28	92
66	https://www.zomato.com/bangalore/namma-brahmin...	30th Cross, 8th Main, Near Jain Temple, 4th Bl...	Namma Brahmin's Idli	Yes	No	3.6	34	

In [176…
```
locations = pd.DataFrame({'name':si_df['location'].unique()})
```

In [177…
```
locations.head()
```

Out[177]:

	name
0	Banashankari
1	Jayanagar
2	Basavanagudi
3	Kumaraswamy Layout
4	JP Nagar

In [178…
```
locations.isna().sum()
```

Out[178]:
```
name    0
dtype: int64
```

In [179…
```
lat = []
lon = []
geolocator = Nominatim(user_agent='app')
for loc in locations['name']:
    location = geolocator.geocode(loc)
    if location is None:
        lat.append(np.nan)
        lon.append(np.nan)
    else:
        lat.append(location.latitude)
        lon.append(location.longitude)
```

In [180…
```
locations['lat'] = lat
locations['lon'] = lon
```

In [181…
```
locations.head()
```

Out[181]:

	name	lat	lon
0	Banashankari	15.887678	75.704678
1	Jayanagar	27.643927	83.052805
2	Basavanagudi	12.941726	77.575502

3	Kumaraswamy Layout	12.908149	77.555318
4	JP Nagar	12.265594	76.646540

In [182…
```
locations.isna().sum()
```

Out[182]:
```
name    0
lat     1
lon     1
dtype: int64
```

In [183…
```
locations.dropna(inplace = True)
```

In [184…
```
locations.isna().sum()
```

Out[184]:
```
name    0
lat     0
lon     0
dtype: int64
```

In [185…
```
locations.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 68 entries, 0 to 68
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   name    68 non-null     object 
 1   lat     68 non-null     float64
 2   lon     68 non-null     float64
dtypes: float64(2), object(1)
memory usage: 2.1+ KB
```

In [186…
```
total = si_df['location'].value_counts().to_frame().reset_index()
```

In [187…
```
total.columns = ['name', 'total']
```

In [188…
```
total.head()
```

Out[188]:

	name	total
0	Jayanagar	94
1	Basavanagudi	86
2	BTM	66
3	Banashankari	60
4	Malleshwaram	59

In [189…
```
total.isna().sum()
```

Out[189]:
```
name     0
total    0
dtype: int64
```

In [190…
```
total.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 69 entries, 0 to 68
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
```

```
 0   name    69 non-null     object
 1   total   69 non-null     int64
dtypes: int64(1), object(1)
memory usage: 1.2+ KB
```

In [191…
```
south_indian = total.merge(locations,on='name',how='left')
```

In [192…
```
south_indian.head()
```

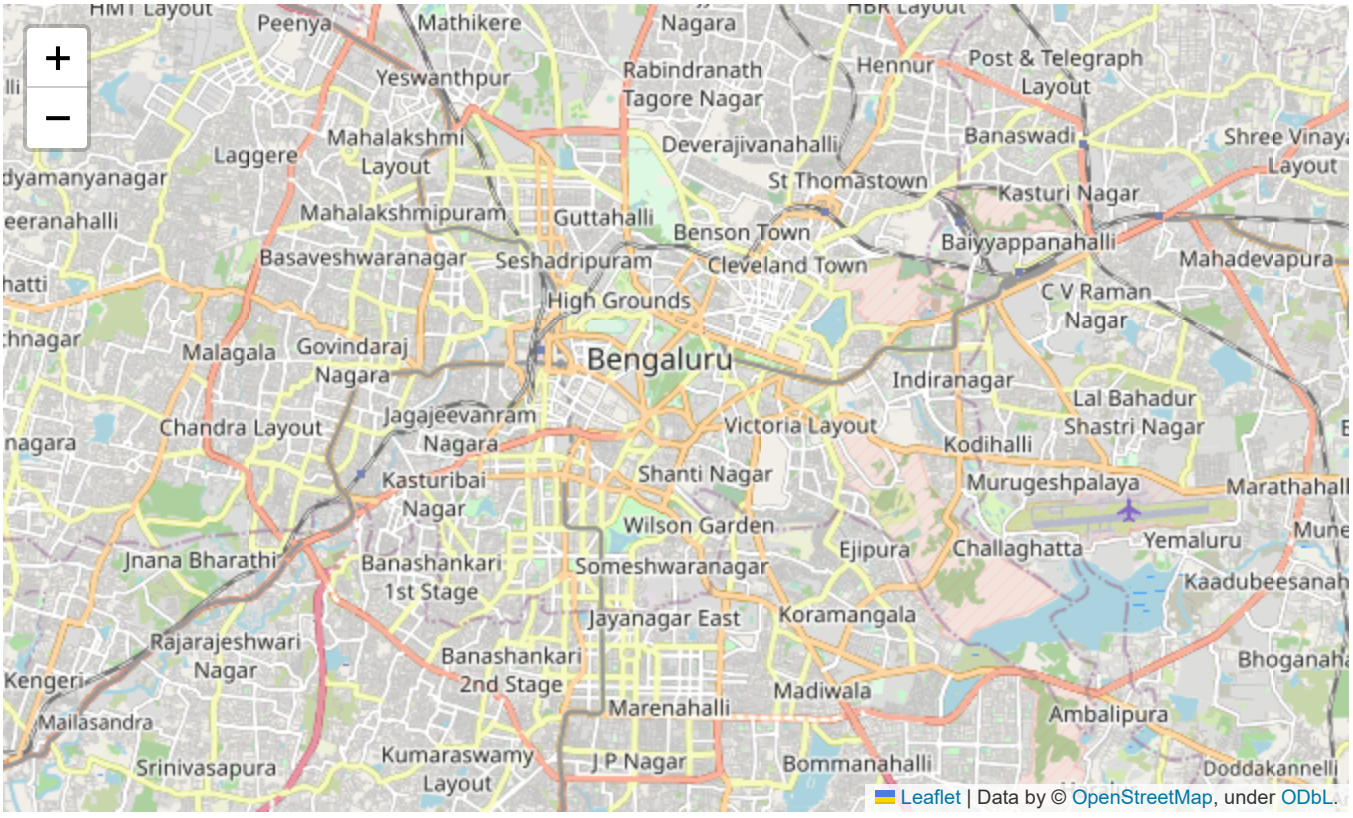
Out[192]:

	name	total	lat	lon
0	Jayanagar	94	27.643927	83.052805
1	Basavanagudi	86	12.941726	77.575502
2	BTM	66	45.954851	-112.496595
3	Banashankari	60	15.887678	75.704678
4	Malleshwaram	59	13.002735	77.570325

In [193…
```
south_indian.dropna(inplace=True)
```

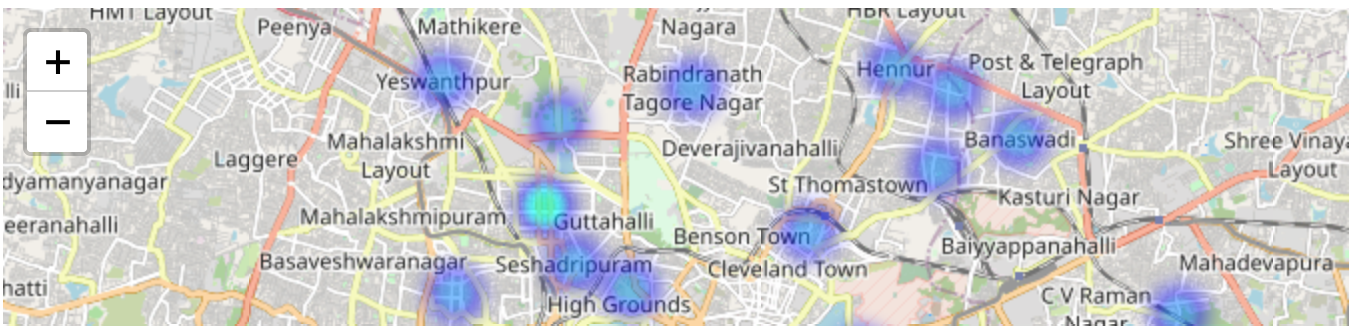
In [194…
```
basemap = generateheatmap()
basemap
```

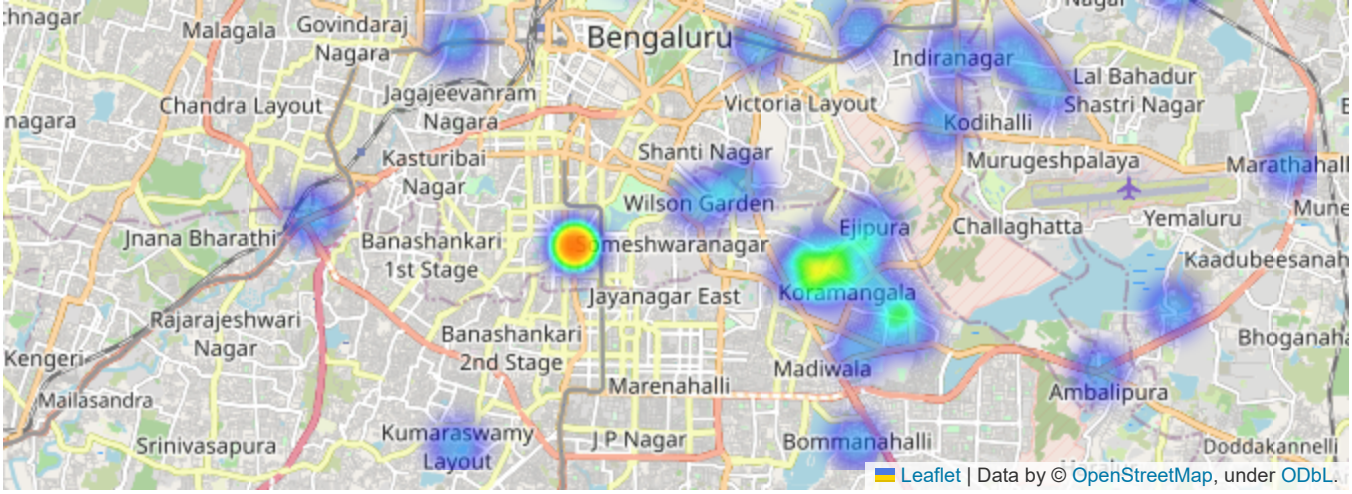
Out[194]:



In [195…
```
HeatMap(south_indian[['lat','lon','total']].values.tolist(),zoom=25,radius=15).add_to(ba
basemap
```

Out[195]:




Malagala
Govindaraj Nagara
Bengaluru
Indiranagar
Lal Bahadur Shastri Nagar
Chandra Layout
Jagajeevanram Nagara
Victoria Layout
Kodihalli
Kasturibai Nagar
Shanti Nagar
Murugeshpalaya
Wilson Garden
Jnana Bharathi
Banashankari 1st Stage
Ejipura
Challaghatta
Yemaluru
Jayanagar East
Koramangala
Kaadubeesanah
Rajarajeshwari Nagar
Banashankari 2nd Stage
Madiwala
Bhoganah
Kengeri
Marenahalli
Ambalipura
Mailasandra
Kumaraswamy Layout
J P Nagar
Bommanahalli
Srinivasapura
Doddakannelli
Leaflet | Data by © OpenStreetMap, under ODbL.