In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
data = pd.read_csv(r'D:\Career\Udemy\DA\Zomato Data Analysis/zomato.csv')
data.head()
```

Out[2]:

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

Missing Values in Percentage

In [3]:

```
data.isnull().sum()
```

Out[3]:

```
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 [4]:

```
feature_na = [feature for feature in data if data[feature].isnull().sum()>0]
feature_na
```

Out[4]:

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

In [5]:

```
for feature in feature_na:
    print("{} has {} % missing values".format(feature,np.round(data[feature].isnull().sum()
```

```
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 [6]:

```
data['rate'].unique()
```

Out[6]:

```
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 [7]:

```
data.dropna(axis = 'index', subset = ['rate'],inplace = True)
```

In [8]:

```
data['rate'].unique()
```

Out[8]:

```
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 [9]:

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

In [10]:

```
data['rate'] = data['rate'].apply(split)
```

In [11]:

```
data.replace('NEW',0,inplace = True)
data.replace('-',0,inplace = True)
```

In [12]:

```
data['rate'] = data['rate'].astype(float)
```

In [13]:

```
data['rate'].unique()
```

Out[13]:

```
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 [14]:

```
data.dropna(axis = 'index',subset = ['approx_cost(for two people)'],inplace = True)
```

Average Rating of Restaurants

In [15]:

```
data_rate = data.groupby('name')['rate'].mean().to_frame().reset_index()
```

In [16]:

```
data_rate
```

Out[16]:

	name	rate
0	#FeelTheROLL	3.400000
1	#L-81 Cafe	3.900000
2	#refuel	3.700000
3	1000 B.C	3.200000
4	100ÃƒÂƒÃ, ÂƒÃƒÂ, Ã, Â, ÃƒÂƒÃ, Â, ÃƒÂ, Ã, Â°C	3.700000
...	...	...
7131	i-Bar - The Park Bangalore	3.800000
7132	iFruit Live Ice Creams	3.400000
7133	iSpice Resto Cafe	3.700000
7134	nu.tree	4.314286
7135	re:cess - Hilton Bangalore Embassy GolfLinks	4.100000

7136 rows × 2 columns

Distribution of Ratings

In [17]:

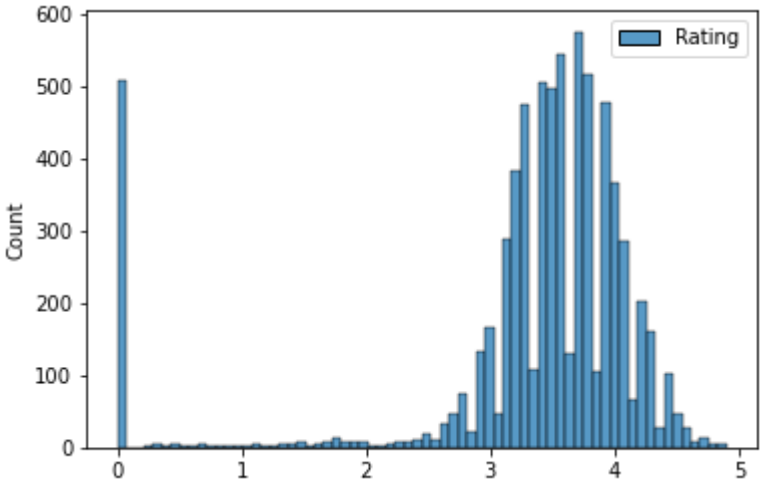
```
data_rate.columns = [['Restaurant_name','Rating']]
```

In [18]:

```
sns.histplot(data_rate['Rating'])
```

Out[18]:

```
<AxesSubplot:ylabel='Count'>
```



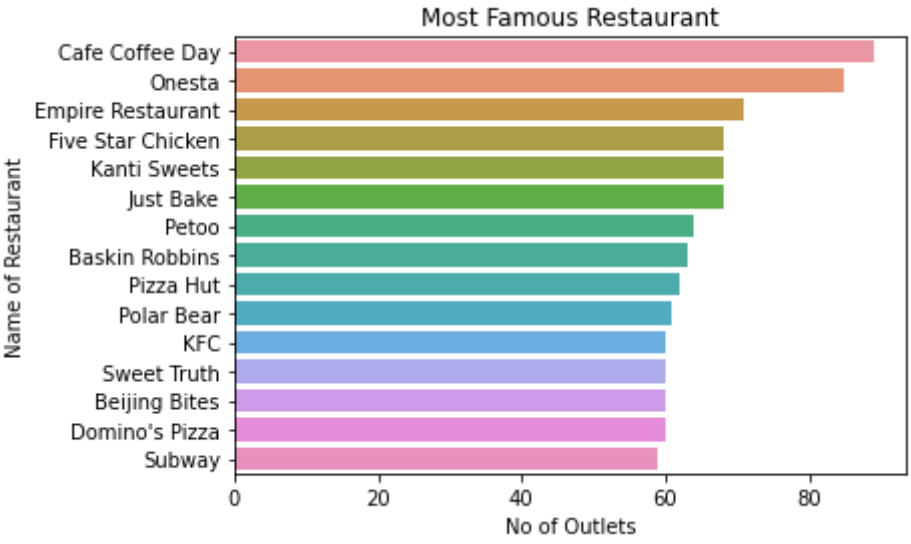
Most Famous Restaurants

In [19]:

```
a = data['name'].value_counts()[0:15]
sns.barplot(x = a, y = a.index)
plt.title('Most Famous Restaurant')
plt.xlabel('No of Outlets')
plt.ylabel('Name of Restaurant')
```

Out[19]:

```
Text(0, 0.5, 'Name of Restaurant')
```

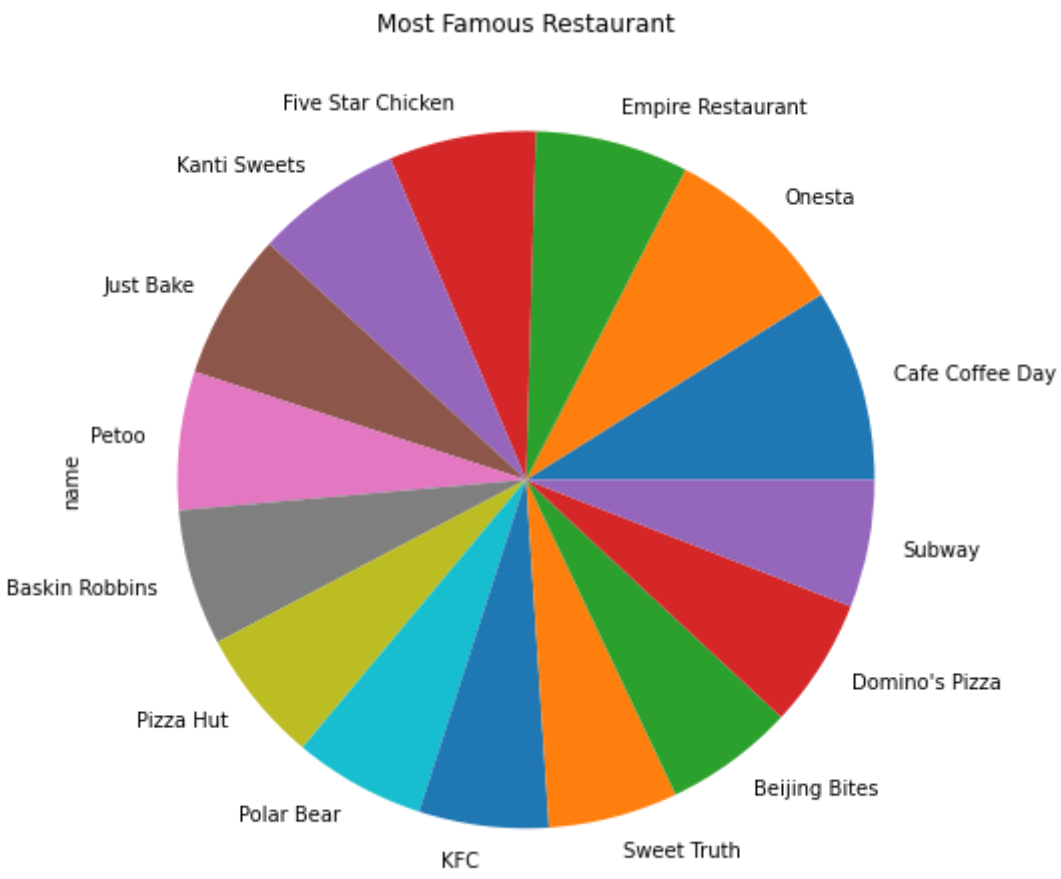


In [20]:

```
a = data['name'].value_counts()[0:15].plot(kind = 'pie',figsize = (8,8))
plt.title('Most Famous Restaurant')
```

Out[20]:

```
Text(0.5, 1.0, 'Most Famous Restaurant')
```



Restaurants that does not accept online order

In [21]:

```
x = data['online_order'].value_counts()
x
```

Out[21]:

```
Yes    28308
No     15382
Name: online_order, dtype: int64
```

In [22]:

```
import plotly.express as px
```

In [23]:

```
Online_order = ['Accpeted','Not Accepted']
```

In [24]:

```
px.pie(data,values = x,names = Online_order,title = 'Online Order')
```

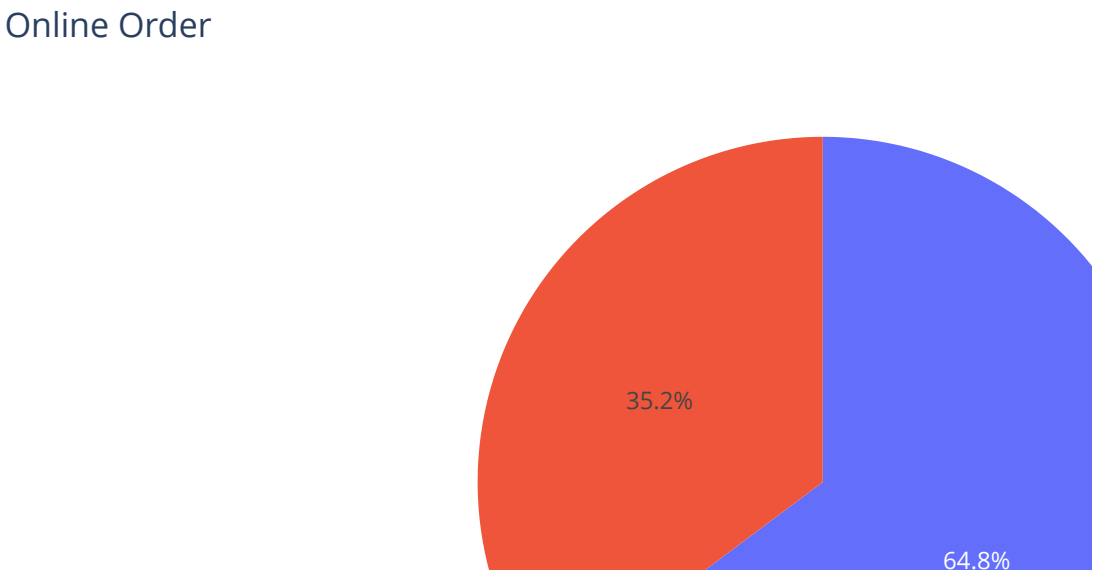


Table Providing or not

In [25]:

```
y = data['book_table'].value_counts()
```

In [26]:

```
import plotly.graph_objs as go
from plotly.offline import iplot
```

In [27]:

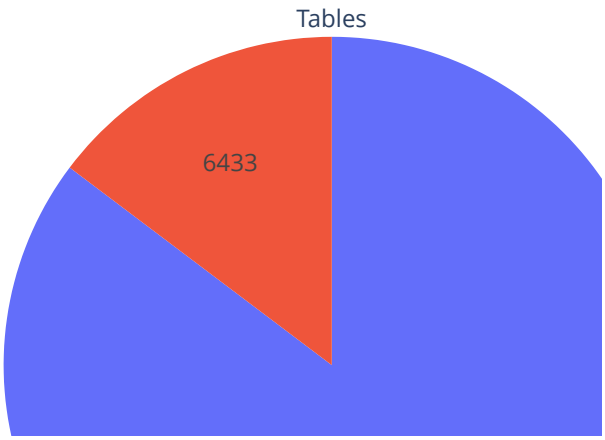
```
names = ['Not Providing','Providing']
```

In [28]:

```
trace = go.Pie(labels = names, values = y, hoverinfo = 'label+percent',textinfo = 'value',t
```

In [29]:

```
iplot([trace])
```



Indepth analysis of types of restaurants

In [30]:

```
data['rest_type'].isnull().sum()
```

Out[30]:

```
149
```

In [31]:

```
data.dropna(axis = 'index',subset = ['rest_type'],inplace = True)
```

In [32]:

```
z = data['rest_type'].value_counts().nlargest(10)
```

In [33]:

```
z
```

Out[33]:

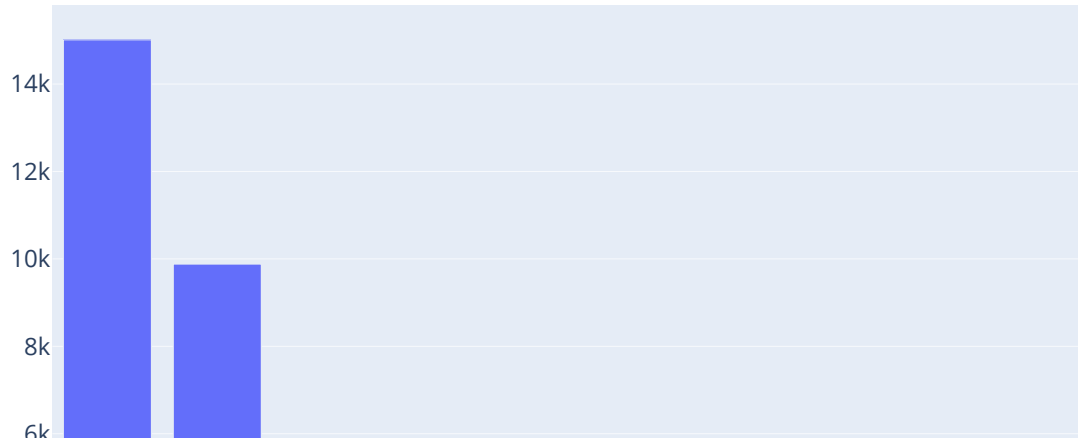
```
Quick Bites           15011
Casual Dining          9880
Cafe                   3491
Dessert Parlor         1925
Delivery               1781
Takeaway, Delivery     1458
Casual Dining, Bar     1123
Bakery                  775
Beverage Shop           704
Bar                     650
Name: rest_type, dtype: int64
```

In [34]:

```
trace2 = go.Bar(x = data['rest_type'].value_counts().nlargest(15).index,
                y = data['rest_type'].value_counts().nlargest(15))
```

In [35]:

```
iplot([trace2])
```



Highest Voted Restaurant

In [36]:

```
high_votes = data.groupby('name')['votes'].sum().nlargest(20)
```

In [37]:

```
high_votes
```

Out[37]:

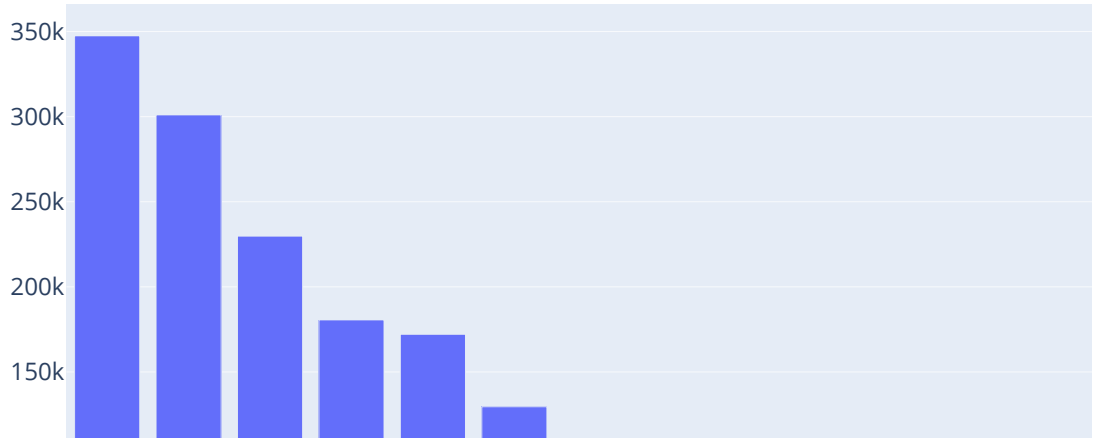
```
name
Onesta                          347520
Truffles                        301059
Empire Restaurant               229808
Hammered                        180602
The Black Pearl                 172122
Meghana Foods                   129557
Barbeque Nation                 108425
Smally's Resto Cafe             102877
Byg Brewski Brewing Company      99531
Gilly's Restobar                 98808
Arbor Brewing Company            92362
House Of Commons                 90573
Chutney Chang                    89910
Fenny's Lounge And Kitchen       89183
AB's - Absolute Barbecues        86418
Church Street Social             83179
Prost Brew Pub                   78609
The Biere Club                   76649
Stoner                           75194
Koramangala Social               75021
Name: votes, dtype: int64
```

In [38]:

```
trace3 = go.Bar(x = data.groupby('name')['votes'].sum().nlargest(20).index,
                y = data.groupby('name')['votes'].sum().nlargest(20))
```

In [39]:

```
iplot([trace3])
```



Total Restaurants in Different Cities in Bangalore

In [40]:

```
data.groupby('name')['location'].sum()
```

Out[40]:

```
name
#FeelTheROLL
BellandurBellandur
#L-81 Cafe                                                        HSRHSR
HSRHSRHSRHSRHSRHSRHSR
#refuel                                          Bannerghatta RoadBannerghatt
a RoadBannerghatta...
1000 B.C                                         Koramangala 5th BlockKoraman
gala 5th BlockKora...
100ÃƒÂƒÃ‚Â, Â ÃƒÂƒÃ‚Â, Ã‚Â, Â, ÃƒÂƒÃ‚Â, Â, ÃƒÂ‚Â, Ã‚Â, Â°C
BTMBTMBTM
                                                                    ...
i-Bar - The Park Bangalore                       MG RoadMG RoadMG RoadMG Road
MG RoadMG RoadMG R...
iFruit Live Ice Creams                           Koramangala 1st BlockKoraman
gala 1st BlockKora...
iSpice Resto Cafe                                IndiranagarIndiranagarIndira
nagarIndiranagarIn...
nu.tree                                          Sarjapur RoadWhitefieldSarja
pur RoadWhitefield...
re:cess - Hilton Bangalore Embassy GolfLinks
DomlurDomlurDomlur
Name: location, Length: 7116, dtype: object
```

In [41]:

```
restaurant = []
location = []
for key,location_df in data.groupby('location'):
    location.append(key)
    restaurant.append(len(location_df['name'].unique()))
```

In [42]:

```
df_total = pd.DataFrame(zip(location,restaurant))
df_total.head()
```

Out[42]:

	0	1
0	BTM	578
1	Banashankari	238
2	Banaswadi	147
3	Bannerghatta Road	360
4	Basavanagudi	195

In [43]:

```
df_total.columns = ['Location','No_of_Restaurants']
df_total.set_index('Location',inplace = True)
```

In [44]:

```
df_total.head()
```

Out[44]:

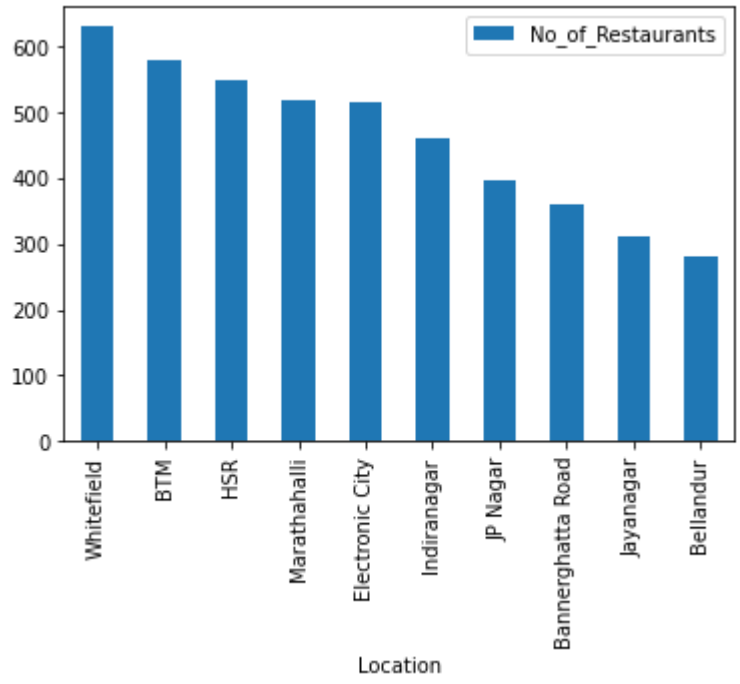
Location	No_of_Restaurants
BTM	578
Banashankari	238
Banaswadi	147
Bannerghatta Road	360
Basavanagudi	195

In [45]:

```
df_total.sort_values(by = 'No_of_Restaurants',ascending = False)[0:10].plot(kind = 'bar')
```

Out[45]:

```
<AxesSubplot:xlabel='Location'>
```



No of variety of Restaurants in Bangalore

In [46]:

```
data['cuisines'].value_counts()[0:10]
```

Out[46]:

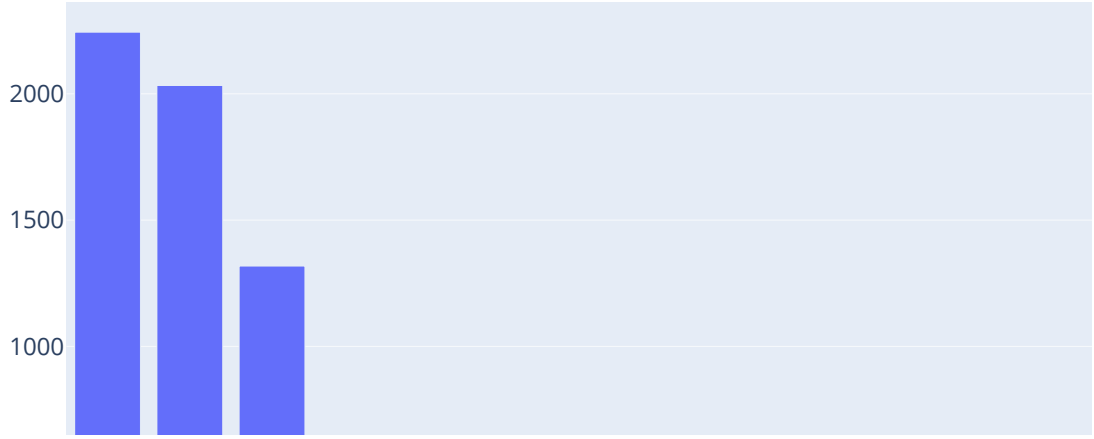
```
North Indian                            2244
North Indian, Chinese                   2033
South Indian                            1318
Bakery, Desserts                         642
Biryani                                  632
Cafe                                     631
South Indian, North Indian, Chinese      601
Fast Food                                576
Desserts                                 566
Chinese                                  449
Name: cuisines, dtype: int64
```

In [47]:

```
trace4 = go.Bar(x = data['cuisines'].value_counts()[0:20].index,
                y = data['cuisines'].value_counts()[0:20])
```

In [48]:

```
iplot([trace4])
```



Approx Cost of 2 feature

In [49]:

```
data['approx_cost(for two people)'].isna().sum()
```

Out[49]:

```
0
```

In [50]:

```
data['approx_cost(for two people)'].unique()
```

Out[50]:

```
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', '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 [51]:

```
data['approx_cost(for two people)'] = data['approx_cost(for two people)'].apply([lambda x :
```

In [52]:

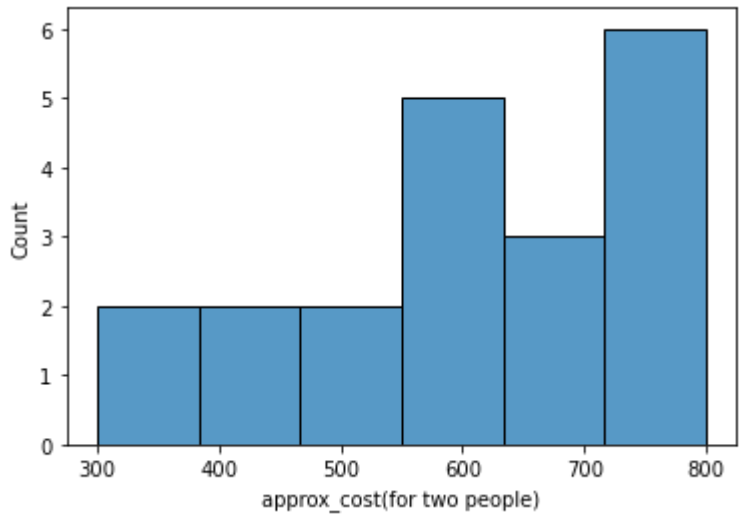
```
data['approx_cost(for two people)'] = data['approx_cost(for two people)'].astype(int)
```

In [53]:

```
sns.histplot(data['approx_cost(for two people)'][0:20])
```

Out[53]:

```
<AxesSubplot:xlabel='approx_cost(for two people)', ylabel='Count'>
```



Approx cost of 2 people vs Rating

In [54]:

```
data['approx_cost(for two people)'].dtype
```

Out[54]:

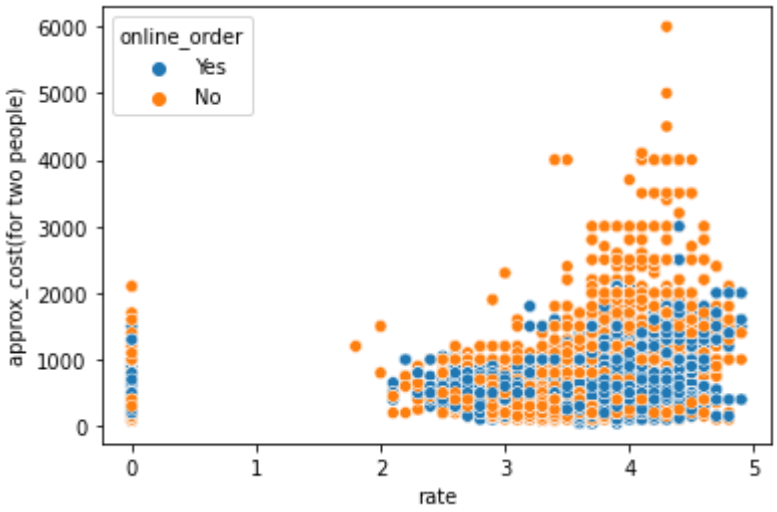
```
dtype('int32')
```

In [55]:

```
sns.scatterplot(x = 'rate',y = 'approx_cost(for two people)',hue = 'online_order',data = da
```

Out[55]:

```
<AxesSubplot:xlabel='rate', ylabel='approx_cost(for two people)'>
```



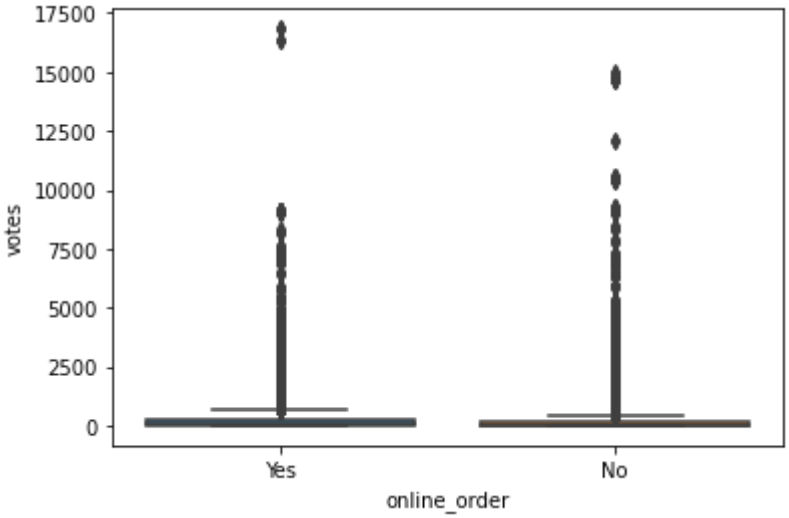
Votes of Hotels accepting Online order and Not

In [56]:

```
sns.boxplot(x = 'online_order',y = 'votes',data = data)
```

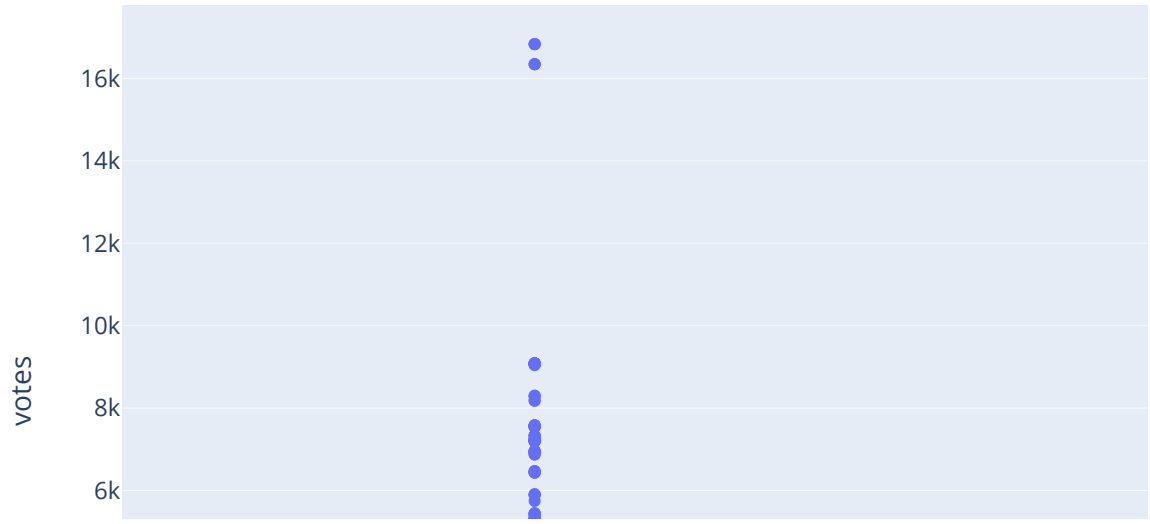
Out[56]:

```
<AxesSubplot:xlabel='online_order', ylabel='votes'>
```



In [57]:

```
px.box(data,x = 'online_order',y = 'votes')
```



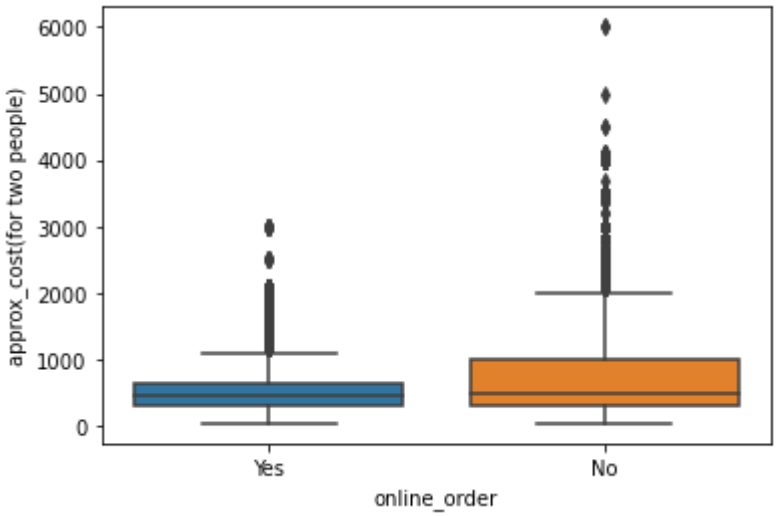
Prices of hotels accepting online order and not

In [58]:

```
sns.boxplot(x = 'online_order',y = 'approx_cost(for two people)',data = data)
```

Out[58]:

```
<AxesSubplot:xlabel='online_order', ylabel='approx_cost(for two people)'>
```



In [59]:

```
px.box(data,x = 'online_order', y = 'approx_cost(for two people)')
```

Most Luxurious Restaurant in Bangalore

In [60]:

```
data['approx_cost(for two people)'].max()
```

Out[60]:

```
6000
```

In [61]:

```
data[data['approx_cost(for two people)'] == 6000]['name']
```

Out[61]:

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

Top 10 Expensive Restaurants in Bangalore

In [62]:

```
data2 = data.copy()
```

In [63]:

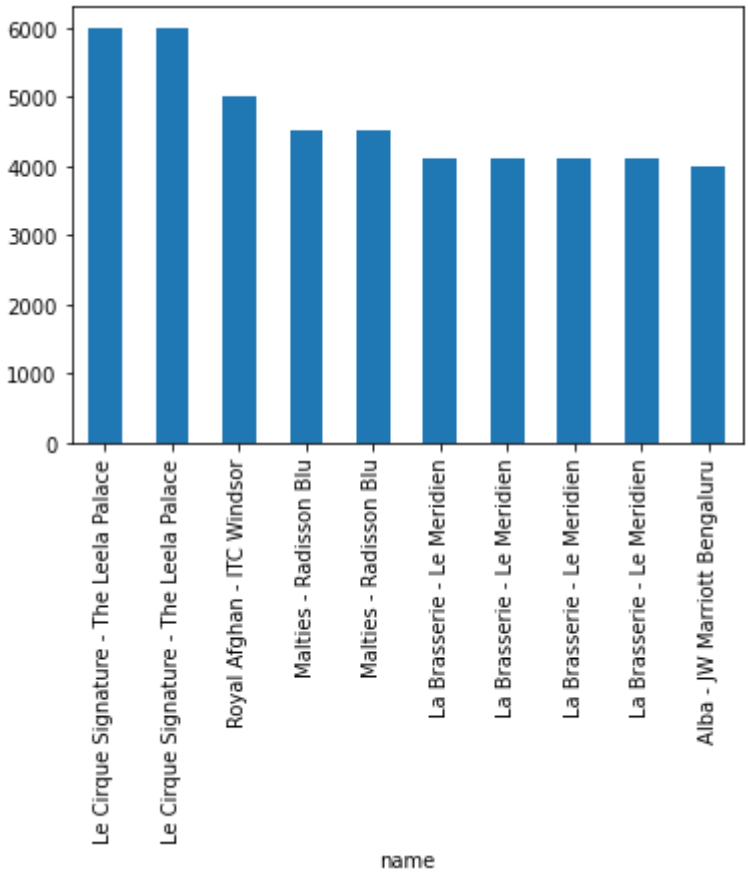
```
data2.set_index('name',inplace = True)
```

In [64]:

```
data2['approx_cost(for two people)'].nlargest(10).plot(kind = 'bar')
```

Out[64]:

```
<AxesSubplot:xlabel='name'>
```

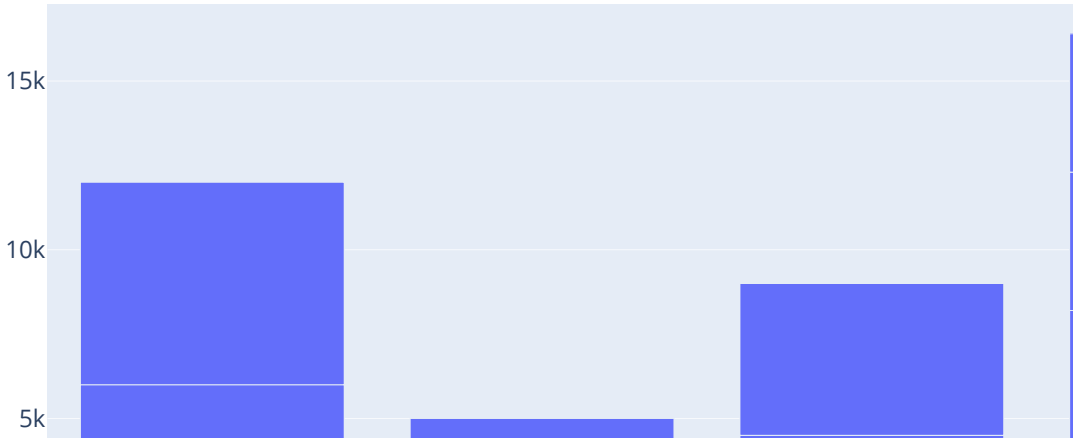


In [65]:

```
trace5 = go.Bar(x = data2['approx_cost(for two people)'].nlargest(10).index,
      y = data2['approx_cost(for two people)'].nlargest(10))
```

In [66]:

```
iplot([trace5])
```



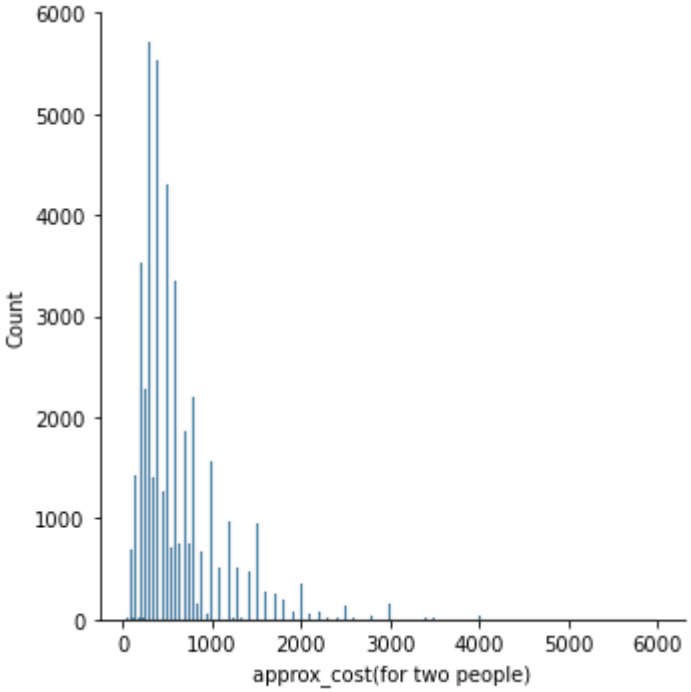
Distribution of Approx cost of 2 people

In [67]:

```
sns.displot(data2['approx_cost(for two people)'])
```

Out[67]:

```
<seaborn.axisgrid.FacetGrid at 0x23caf4ca160>
```



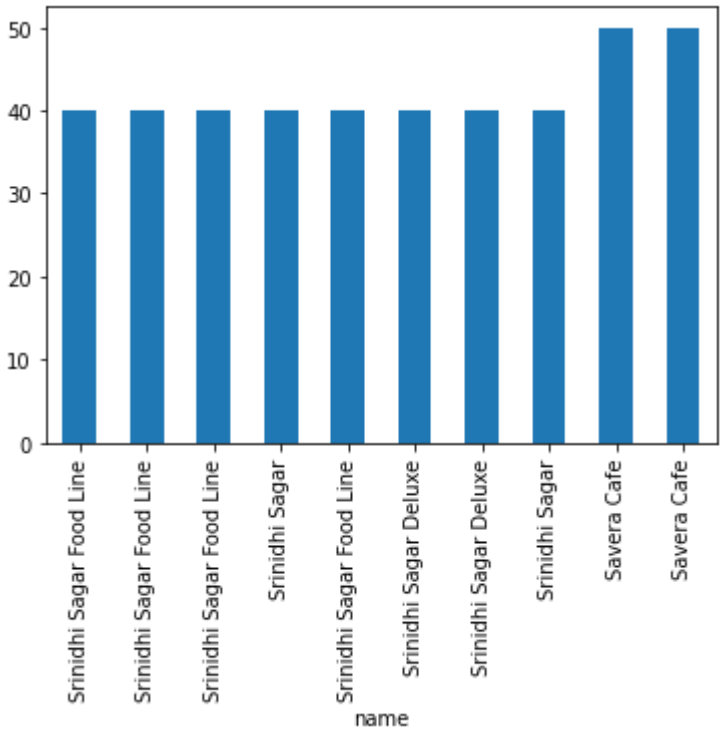
Top 10 Cheapest Restaurants in Bangalore

In [68]:

```
data2['approx_cost(for two people)'].nsmallest(10).plot.bar()
```

Out[68]:

```
<AxesSubplot:xlabel='name'>
```

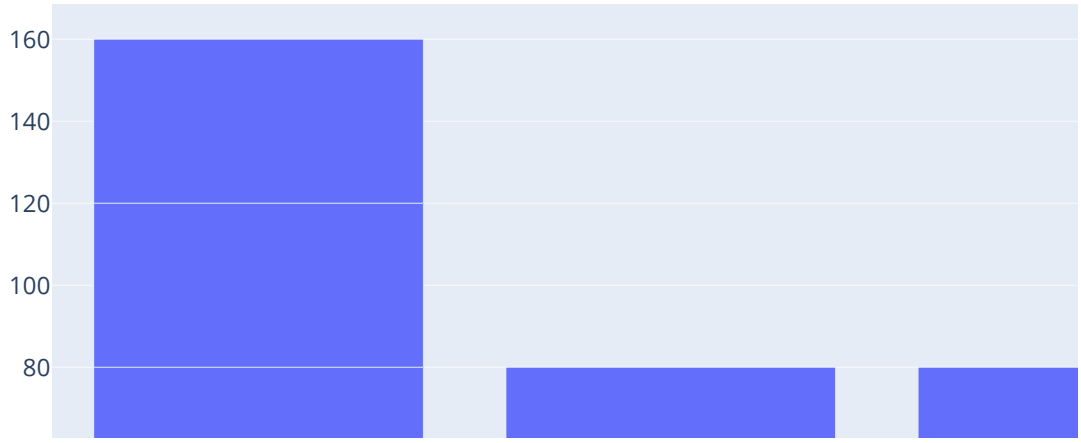


In [69]:

```
trace6 = go.Bar(x = data2['approx_cost(for two people)'].nsmallest(10).index,
                y  =data2['approx_cost(for two people)'].nsmallest(10))
```

In [70]:

```
iplot([trace6])
```



Restarant with cost below 500 as well as affordable

In [71]:

```
data2_budget = data2[data2['approx_cost(for two people)'] < 500].loc[:,('approx_cost(for tw
```

In [72]:

```
data2_budget = data2_budget.reset_index()
```

In [73]:

```
data2_budget.head()
```

Out[73]:

	name	approx_cost(for two people)
0	Addhuri Udupi Bhojana	300
1	Caf-Eleven	450
2	T3H Cafe	300
3	360 Atoms Restaurant And Cafe	400
4	The Vintage Cafe	400

Restaurants having rate > 4 and are of low budget

In [74]:

```
data[(data['rate'] > 4) & (data['approx_cost(for two people)'] <= 500)]
```

Out[74]:

	url	address	
10	https://www.zomato.com/bangalore/caf%C3%A9-dow...	12,29 Near PES University Back Gate, D'Souza N...	CafÃƒÂƒÃ‚Â, Ã‚Â, Ã, ÂƒÃ‚ÂƒÃ‚Â, Ã D
12	https://www.zomato.com/bangalore/the-coffee-sh...	6th Block, 3rd Stage, Banashankari, Bangalore	The C
34	https://www.zomato.com/bangalore/faasos-banash...	80, BDA Complex, 2nd Stage, Banashankari, Bang...	
51	https://www.zomato.com/bangalore/shree-cool-po...	1514, 4th Cross, 7th Main, RPC layout, 2nd Sta...	Shre
52	https://www.zomato.com/bangalore/corner-house-...	808/6-1, 24th A Cross, K.R Road, 2nd Stage, Ba...	Corner Hous
...	...	...	
51312	https://www.zomato.com/bangalore/biryani-kitch...	FB 11, 1st Floor, Inorbit Mall, EPIP Area, Whi...	Bi
51313	https://www.zomato.com/bangalore/stoner-whitef...	120/57, Ground Floor, Azeem's Gold Building, W...	
51345	https://www.zomato.com/bangalore/the-wok-shop-...	S 26, 2nd Floor, Phoenix Market City, Whitefie...	Th
51376	https://www.zomato.com/bangalore/nu-tree-1-whi...	12th Floor, Gamma Building, Sigma Soft Tech Pa...	

	url	address
51437	https://www.zomato.com/bangalore/captain-egg-w...	V R Bengaluru Mall, Mahadevapura Main Rd, Whit...

2436 rows × 17 columns

In [75]:

```
len(data[(data['rate'] > 4) & (data['approx_cost(for two people)'] <= 500)]['name'].unique(
```

Out[75]:

372

Total various afordable hotels at all locations in Bangalore

In [76]:

```
new_df = data[(data['rate'] > 4) & (data['approx_cost(for two people)'] <= 500)]
```

In [77]:

```
location = []
total = []
for loc,location_df in new_df.groupby('location'):
    location.append(loc)
    total.append(len(location_df['name'].unique()))
```

In [78]:

```
df_new = pd.DataFrame(zip(location,total))
```

In [79]:

```
df_new.columns = ['Location','Restaurants']
```

In [80]:

```
df_new.head()
```

Out[80]:

	Location	Restaurants
0	BTM	28
1	Banashankari	16
2	Banaswadi	1
3	Bannerghatta Road	9
4	Basavanagudi	24

Best Budget Restaurants in any Location

In [81]:

```
def return_budget(location,restaurant):
    budget = data[(data['approx_cost(for two people)'] <= 400) & (data['rate'] > 4) & (data
    return budget['name'].unique()
```

In [82]:

```
return_budget('BTM','Quick Bites')
```

Out[82]:

```
array(['Swadista Aahar', 'Litti Twist', 'The Shawarma Shop', 'Gorbandh',
       'Yum In My Tum', 'Chaatimes', "Muthashy's", 'Swad Punjab Da',
       "Domino's Pizza", 'Roti Wala', 'Andhra Kitchen'], dtype=object)
```

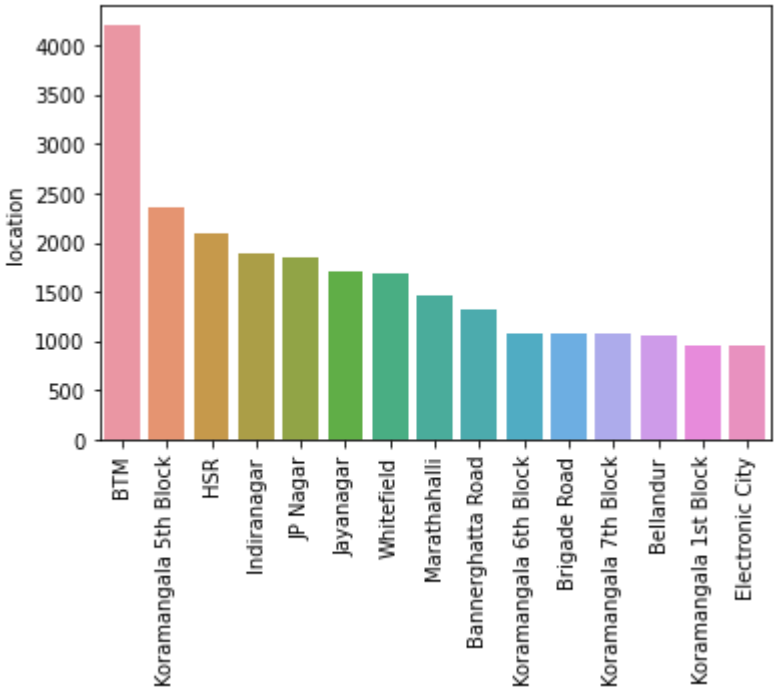
Foodie Areas

In [83]:

```
foodie = data['location'].value_counts()[0:15]
sns.barplot(y = foodie, x = foodie.index)
plt.xticks(rotation = 'vertical')
```

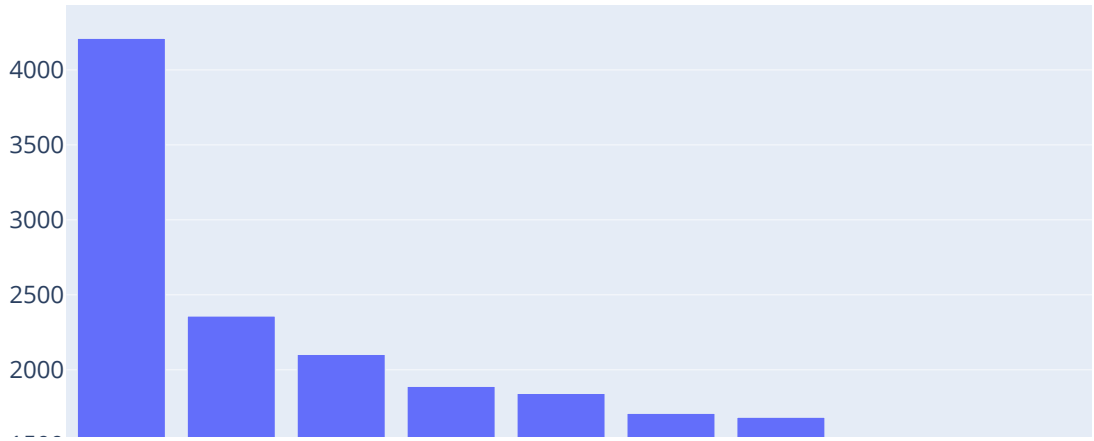
Out[83]:

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14]),
 [Text(0, 0, 'BTM'),
  Text(1, 0, 'Koramangala 5th Block'),
  Text(2, 0, 'HSR'),
  Text(3, 0, 'Indiranagar'),
  Text(4, 0, 'JP Nagar'),
  Text(5, 0, 'Jayanagar'),
  Text(6, 0, 'Whitefield'),
  Text(7, 0, 'Marathahalli'),
  Text(8, 0, 'Bannerghatta Road'),
  Text(9, 0, 'Koramangala 6th Block'),
  Text(10, 0, 'Brigade Road'),
  Text(11, 0, 'Koramangala 7th Block'),
  Text(12, 0, 'Bellandur'),
  Text(13, 0, 'Koramangala 1st Block'),
  Text(14, 0, 'Electronic City')])
```



In [84]:

```
trace9 = go.Bar(x = foodie.index,y = foodie)
iplot([trace9])
```



In [85]:

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

Out[85]:

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

In [86]:

```
from geopy.geocoders import Nominatim
```

In [87]:

```
geolocator = Nominatim(user_agent = 'app')
```

In [88]:

```
lat_lon = []
for location in locations['name']:
    location = geolocator.geocode(location)
    if location is None:
        lat_lon.append(np.nan)
    else:
        geo = (location.latitude,location.longitude)
        lat_lon.append(geo)
```

Latitudes and Longitudes of Each Location in Bangalore

In [89]:

```
locations['Geo_loc'] = lat_lon
```

In [90]:

```
locations.head()
```

Out[90]:

	name	Geo_loc
0	Banashankari	(15.8876779, 75.7046777)
1	Basavanagudi	(12.9417261, 77.5755021)
2	Mysore Road	(12.9466619, 77.5300896)
3	Jayanagar	(27.64392675, 83.05280519687284)
4	Kumaraswamy Layout	(12.9081487, 77.5553179)

In [91]:

```
Rest_locations = pd.DataFrame(data['location'].value_counts().reset_index())
Rest_locations.columns = ['name','Count']
Rest_locations.head()
```

Out[91]:

	name	Count
0	BTM	4210
1	Koramangala 5th Block	2358
2	HSR	2102
3	Indiranagar	1889
4	JP Nagar	1842

In [92]:

```
Restaurant_locations = pd.merge(locations,Rest_locations, on = 'name')
```

In [93]:

```
Restaurant_locations.head()
```

Out[93]:

	name	Geo_loc	Count
0	Banashankari	(15.8876779, 75.7046777)	805
1	Basavanagudi	(12.9417261, 77.5755021)	628
2	Mysore Road	(12.9466619, 77.5300896)	18
3	Jayanagar	(27.64392675, 83.05280519687284)	1709
4	Kumaraswamy Layout	(12.9081487, 77.5553179)	167

In [94]:

```
Restaurant_locations['Geo_loc'] = np.asarray(Restaurant_locations['Geo_loc'])
```

In [95]:

```
a = []
b = []
for i in Restaurant_locations['Geo_loc']:
    if type(i) == tuple:
        a.append(i[0])
        b.append(i[1])
```

In [96]:

```
a = np.array(a)
```

In [97]:

```
b = np.array(b)
```

In [98]:

```
df = pd.DataFrame({'Lat':a,'Lon':b})
```

In [99]:

```
df
```

Out[99]:

	Lat	Lon
0	15.887678	75.704678
1	12.941726	77.575502
2	12.946662	77.530090
3	27.643927	83.052805
4	12.908149	77.555318
...	...	...
86	18.490080	73.847530
87	13.651058	77.430522
88	12.959618	77.511267
89	13.032942	77.527325
90	13.007516	77.695935

91 rows × 2 columns

In [100]:

```
Restaurant_locations = Restaurant_locations.join(df)
```

In [101]:

```
Restaurant_locations.dropna(axis = 'index',subset = ['Lat','Lon'],inplace = True)
```

In [102]:

```
import folium
```

In [103]:

```
from folium.plugins import HeatMap
```

In [104]:

```
def generatebasemap(default_location = [12.97,77.59],default_zoom_start = 12):
    basemap = folium.Map(location = default_location,zoom_start = default_zoom_start)
    return basemap
```

In [105]:

```
basemap = generatebasemap()
```

In [106]:

```
basemap
```

Out[106]:

Heatmap of Restaurants

In [107]:

```
HeatMap(Restaurant_locations[['Lat','Lon','Count']].values.tolist(),zoom = 20,radius = 15).
```

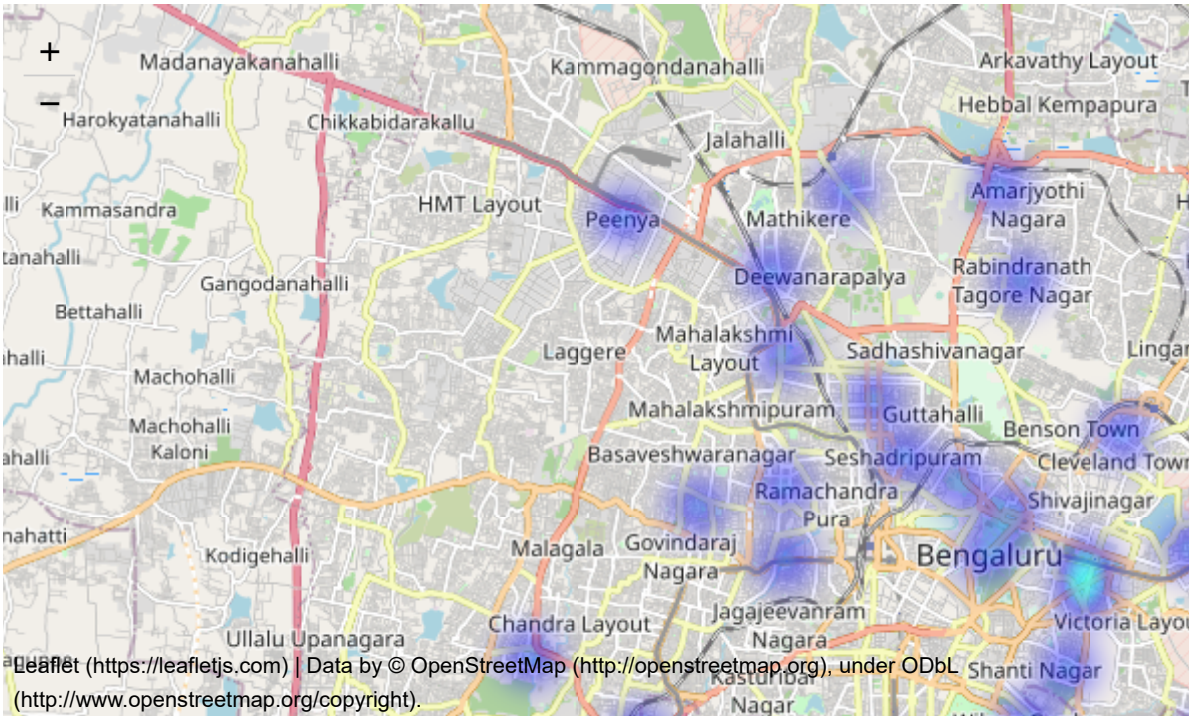
Out[107]:

```
<folium.plugins.heat_map.HeatMap at 0x23cb146c430>
```

In [108]:

```
basemap
```

Out[108]:



Heatmap of North Indian Restaurants

In [109]:

```
df2  = data[data['cuisines'] == 'North Indian']
```

In [110]:

```
north_india = df2.groupby(['location'],as_index = False)['url'].agg('count')
```

In [111]:

```
north_india.columns = ['name','Count']
```

In [112]:

```
north_india = north_india.merge(locations,on = 'name',how = 'left').dropna()
```

In [113]:

```
north_india['lat'],north_india['lon'] = zip(*north_india['Geo_loc'])
```

In [114]:

```
north_india.drop('Geo_loc',axis = 1,inplace = True)
```

In [115]:

```
north_india
```

Out[115]:

	name	Count	lat	lon
0	BTM	262	45.954851	-112.496595
1	Banashankari	35	15.887678	75.704678
2	Banaswadi	5	13.014162	77.651854
3	Bannerghatta Road	60	12.888586	77.597307
4	Basavanagudi	17	12.941726	77.575502
...	...	...	...	...
58	Varthur Main Road, Whitefield	3	12.941324	77.747110
59	Vasanth Nagar	12	12.988721	77.585169
60	Whitefield	145	53.553368	-2.296902
61	Wilson Garden	37	12.948934	77.596827
62	Yeshwantpur	3	13.023830	77.552921

63 rows × 4 columns

In [116]:

```
basemap = generatebasemap()
HeatMap(north_india[['lat','lon','Count']].values.tolist(),zoom = 20,radius = 15).add_to(ba
```

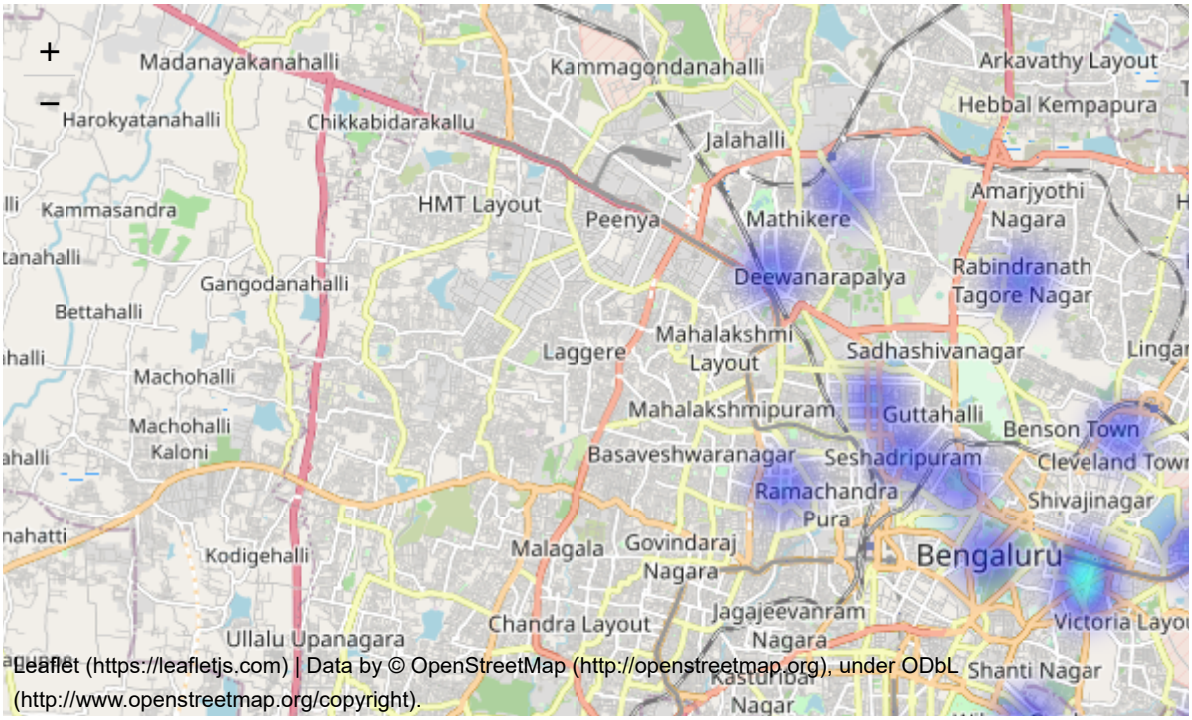
Out[116]:

```
<folium.plugins.heat_map.HeatMap at 0x23caf866400>
```

In [117]:

```
basemap
```

Out[117]:



Most Popular Casual Dining Restaurant Chains

In [118]:

```
df3 = data.groupby(['rest_type','name']).agg('count')
```

In [119]:

```
df4 = df3.sort_values(['url'],ascending = False).groupby(['rest_type'],as_index = False).ap
```

In [120]:

```
df4.head()
```

Out[120]:

	level_0	rest_type	name	count
0	0	Bakery	Just Bake	44
1	0	Bakery	Warm Oven	28
2	0	Bakery	INDULGE by InnerChef	28
3	0	Bakery	Karachi Bakery	26
4	0	Bakery	CakeZone	21

In [121]:

```
casual = df4[df4['rest_type'] == 'Casual Dining']
casual[0:10]
```

Out[121]:

	level_0	rest_type	name	count
1001	24	Casual Dining	Empire Restaurant	58
1002	24	Casual Dining	Beijing Bites	48
1003	24	Casual Dining	Mani's Dum Biryani	47
1004	24	Casual Dining	Chung Wah	46
1005	24	Casual Dining	Oye Amritsar	41
1006	24	Casual Dining	Barbeque Nation	41
1007	24	Casual Dining	Toscano	40
1008	24	Casual Dining	A2B - Adyar Ananda Bhavan	39
1009	24	Casual Dining	New Prashanth Hotel	38
1010	24	Casual Dining	Pizza Hut	38