# Zomato Exploratory Data Analysis

In [1]:

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

# importing libraries

In [2]:

```
%matplotlib inline
```

# Loading dataset

In [5]:

```
data=pd.read_csv(r"C:\Users\Tesla\Downloads\zomato dataset\zomato.csv")
```

In [6]:

```
data.head()
```

Out[6]:

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

In [13]:

```
data.columns
```

Out[13]:

```
Index(['url', 'address', 'name', 'online_order', 'book_table', 'rate', 'votes',
       'phone', 'location', 'rest_type', 'dish_liked', 'cuisines',
       'approx_cost(for two people)', 'reviews_list', 'menu_item',
       'listed_in(type)', 'listed_in(city)'],
      dtype='object')
```

In [8]:

```
data.tail()
```

Out[8]:

| | url | address | name | online_order | book_table |
|---|---|---|---|---|---|
| **51712** | https://www.zomato.com/bangalore/best-brews-fo... | Four Points by Sheraton Bengaluru, 43/3, White... | Best Brews - Four Points by Sheraton Bengaluru... | No | No |
| **51713** | https://www.zomato.com/bangalore/vinod-bar-and... | Number 10, Garudachar Palya, Mahadevapura, Whi... | Vinod Bar And Restaurant | No | No |
| **51714** | https://www.zomato.com/bangalore/plunge-sherat... | Sheraton Grand Bengaluru Whitefield Hotel & Co... | Plunge - Sheraton Grand Bengaluru Whitefield H... | No | No |
| **51715** | https://www.zomato.com/bangalore/chime-sherato... | Sheraton Grand Bengaluru Whitefield Hotel & Co... | Chime - Sheraton Grand Bengaluru Whitefield Ho... | No | Yes |
| **51716** | https://www.zomato.com/bangalore/the-nest-the-... | ITPL Main Road, KIADB Export Promotion Industr... | The Nest - The Den Bengaluru | No | No |

In [16]:

```
data=data.drop(['url','address','phone','menu_item','listed_in(type)','listed_in(city
```

In [ ]:

```
#removing columns
```

In [17]:

```
data.head()
```

Out[17]:

| | name | online_order | book_table | rate | votes | location | rest_type | dish_liked | cuisines | ap |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Jalsa | Yes | Yes | 4.1/5 | 775 | Banashankari | Casual Dining | Pasta, Lunch Buffet, Masala Papad, Paneer Laja... | North Indian, Mughlai, Chinese | |
| **1** | Spice Elephant | Yes | No | 4.1/5 | 787 | Banashankari | Casual Dining | Momos, Lunch Buffet, Chocolate Nirvana, Thai G... | Chinese, North Indian, Thai | |
| **2** | San Churro Cafe | Yes | No | 3.8/5 | 918 | Banashankari | Cafe, Casual Dining | Churros, Cannelloni, Minestrone Soup, Hot Choc... | Cafe, Mexican, Italian | |
| **3** | Addhuri Udupi Bhojana | No | No | 3.7/5 | 88 | Banashankari | Quick Bites | Masala Dosa | South Indian, North Indian | |
| **4** | Grand Village | No | No | 3.8/5 | 166 | Basavanagudi | Casual Dining | Panipuri, Gol Gappe | North Indian, Rajasthani | |

In [19]:

```
data.shape
```

Out[19]:

```
(51717, 11)
```

In [20]:

```
data.columns
```

Out[20]:

```
Index(['name', 'online_order', 'book_table', 'rate', 'votes', 'location',
       'rest_type', 'dish_liked', 'cuisines', 'approx_cost(for two people)',
       'reviews_list'],
      dtype='object')
```

In [22]:

```
data.isna().sum()
```

Out[22]:
```
name                              0
online_order                      0
book_table                        0
rate                           7775
votes                             0
location                         21
rest_type                       227
dish_liked                    28078
cuisines                         45
approx_cost(for two people)     346
reviews_list                      0
dtype: int64
```

In [23]:
```
data.isnull().sum()
```

Out[23]:
```
name                              0
online_order                      0
book_table                        0
rate                           7775
votes                             0
location                         21
rest_type                       227
dish_liked                    28078
cuisines                         45
approx_cost(for two people)     346
reviews_list                      0
dtype: int64
```

# Data cleaning

In [26]:
```
feature_na=[i for i in data.columns if data[i].isnull().sum()>0 ]
```

In [27]:
```
feature_na
```

Out[27]:
```
['rate',
 'location',
 'rest_type',
 'dish_liked',
 'cuisines',
 'approx_cost(for two people)']
```

In [31]:
```
for i in feature_na:
 print(f"{i} has {np.round(data[i].isnull().sum()/len(data[i])*100,2)}% null values")
```

```
rate has 15.03% null values
location has 0.04% null values
rest_type has 0.44% null values
dish_liked has 54.29% null values
cuisines has 0.09% null values
approx_cost(for two people) has 0.67% null values
```

In [33]:
```
data.rate.unique()
```

Out[33]:

```
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 [34]:

```
data.rate
```

Out[34]:

```
0          4.1/5
1          4.1/5
2          3.8/5
3          3.7/5
4          3.8/5
           ...
51712    3.6 /5
51713       NaN
51714       NaN
51715    4.3 /5
51716    3.4 /5
Name: rate, Length: 51717, dtype: object
```

In [38]:

```
data.rate
```

Out[38]:

```
0          4.1/5
1          4.1/5
2          3.8/5
3          3.7/5
4          3.8/5
           ...
51712    3.6 /5
51713       NaN
51714       NaN
51715    4.3 /5
51716    3.4 /5
Name: rate, Length: 51717, dtype: object
```

In [41]:

```
data.dropna(subset=['rate'],inplace=True,axis=0)
```

In [42]:

```
data.rate
```

Out[42]:

```
0          4.1/5
1          4.1/5
2          3.8/5
3          3.7/5
4          3.8/5
           ...
51709    3.7 /5
51711    2.5 /5
51712    3.6 /5
51715    4.3 /5
51716    3.4 /5
Name: rate, Length: 43942, dtype: object
```

In [43]:

```
data['rate'] = data['rate'].apply(lambda x : x.split('/')[0].strip())
```

In [44]:

```
data.rate
```

Out[44]:

```
0        4.1
1        4.1
2        3.8
3        3.7
4        3.8
        ... 
51709    3.7
51711    2.5
51712    3.6
51715    4.3
51716    3.4
Name: rate, Length: 43942, dtype: object
```

In [45]:

```
data.rate.unique()
```

Out[45]:

```
array(['4.1', '3.8', '3.7', '3.6', '4.6', '4.0', '4.2', '3.9', '3.1',
       '3.0', '3.2', '3.3', '2.8', '4.4', '4.3', 'NEW', '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', '2.0', '1.8'], dtype=object)
```

In [46]:

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

In [56]:

```
data.rate
```

Out[56]:

```
0        4.1
1        4.1
2        3.8
3        3.7
4        3.8
        ... 
51709    3.7
51711    2.5
51712    3.6
51715    4.3
51716    3.4
Name: rate, Length: 43942, dtype: float64
```

In [48]:

```
data.rate
```

Out[48]:

```
0        4.1
1        4.1
2        3.8
3        3.7
4        3.8
        ... 
51709    3.7
51711    2.5
51712    3.6
51715    4.3
51716    3.4
Name: rate, Length: 43942, dtype: object
```

In [49]:

```
data.rate.unique()
```

Out[49]:

```
array(['4.1', '3.8', '3.7', '3.6', '4.6', '4.0', '4.2', '3.9', '3.1',
       '3.0', '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', '2.0', '1.8'], dtype=object)
```

In [50]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 43942 entries, 0 to 51716
Data columns (total 11 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   name                         43942 non-null  object
 1   online_order                 43942 non-null  object
 2   book_table                   43942 non-null  object
 3   rate                         43942 non-null  object
 4   votes                        43942 non-null  int64 
 5   location                     43942 non-null  object
 6   rest_type                    43791 non-null  object
 7   dish_liked                   23609 non-null  object
 8   cuisines                     43931 non-null  object
 9   approx_cost(for two people)  43690 non-null  object
 10  reviews_list                 43942 non-null  object
dtypes: int64(1), object(10)
memory usage: 4.0+ MB
```

In [51]:

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

Out[51]:

```
name                               0
online_order                       0
book_table                         0
rate                               0
votes                              0
location                           0
rest_type                        151
dish_liked                     20333
cuisines                          11
approx_cost(for two people)      252
reviews_list                       0
dtype: int64
```

In [52]:

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

In [55]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 43942 entries, 0 to 51716
Data columns (total 11 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   name                         43942 non-null  object 
 1   online_order                 43942 non-null  object 
 2   book_table                   43942 non-null  object 
 3   rate                         43942 non-null  float64
 4   votes                        43942 non-null  int64  
 5   location                     43942 non-null  object 
 6   rest_type                    43791 non-null  object 
 7   dish_liked                   23609 non-null  object 
 8   cuisines                     43931 non-null  object 
 9   approx_cost(for two people)  43690 non-null  object 
 10  reviews_list                 43942 non-null  object 
dtypes: float64(1), int64(1), object(9)
memory usage: 4.0+ MB
```

# calculate average rating for each restaurant

In [89]:
```
rating = pd.pivot_table(data,index='name',values='rate')
rating
```

Out[89]:

| name | rate |
|---:|---:|
| **#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 [90]:
```
rating=rating.sort_values(['rate'],ascending=False)
rating[0:15]
```

Out[90]:

| name | rate |
|---:|---:|
| Asia Kitchen By Mainland China | 4.900000 |
| Byg Brewski Brewing Company | 4.900000 |
| SantÃ□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â© Spa Cuisine | 4.900000 |
| Punjab Grill | 4.871429 |
| Belgian Waffle Factory | 4.844828 |
| O.G. Variar & Sons | 4.800000 |
| Flechazo | 4.800000 |
| The Pizza Bakery | 4.800000 |
| AB's - Absolute Barbecues | 4.789474 |
| Barbecue by Punjab Grill | 4.750000 |
| CTR | 4.750000 |
| The Black Pearl | 4.727778 |
| House Of Commons | 4.723810 |
| The Big Barbeque | 4.700000 |
| ECHOES Koramangala | 4.700000 |

In [91]:

```
plt.figure(figsize=(15,8))
```

Out[91]:

```
<Figure size 1500x800 with 0 Axes>

<Figure size 1500x800 with 0 Axes>
```

In [73]:

```
rating.info()
```

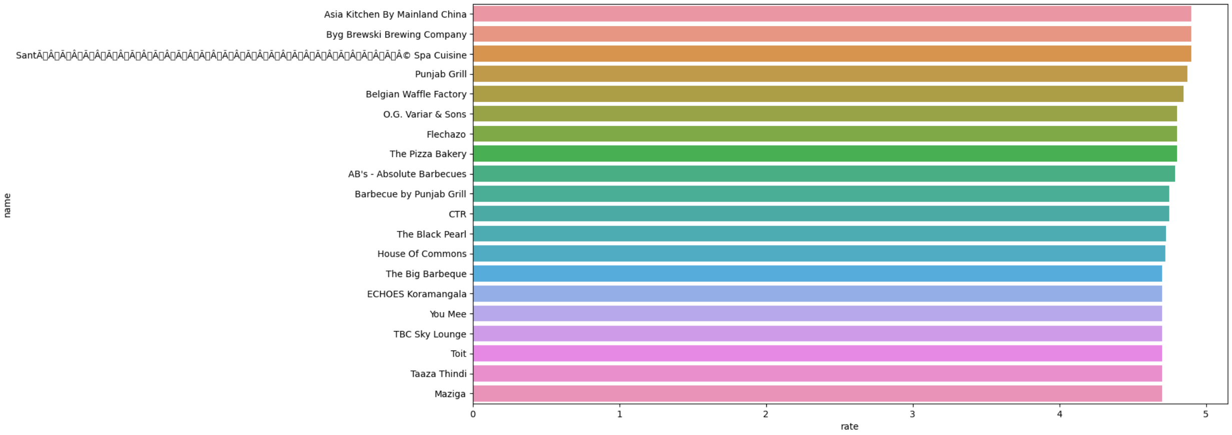
```
<class 'pandas.core.series.Series'>
Index: 7162 entries, Asia Kitchen By Mainland China to Lazzet Lee
Series name: rate
Non-Null Count  Dtype
--------------  -----
7162 non-null   float64
dtypes: float64(1)
memory usage: 369.9+ KB
```

In [ ]:

In [105…

```
plt.figure(figsize=(15,8))
sns.barplot(x=rating[0:20].rate,y=rating[0:20].index,orient='h')
plt.show()
```

```
C:\Users\Tesla\anaconda3\lib\site-packages\IPython\core\pylabtools.py:152: UserWarnin
g: Glyph 131 (\x83) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\Tesla\anaconda3\lib\site-packages\IPython\core\pylabtools.py:152: UserWarnin
g: Glyph 130 (\x82) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```



In [95]:
```
rating[0:20].index
```

Out[95]:
```
Index(['Asia Kitchen By Mainland China', 'Byg Brewski Brewing Company',
       'SantÃ⍰Â⍰Ã⍰Â⍰Ã⍰Â⍰Ã⍰Â⍰Ã⍰Â⍰Ã⍰Â⍰Ã⍰Â⍰Ã⍰Â⍰Ã⍰Â⍰Ã⍰Â⍰Ã⍰Â⍰Ã⍰Â⍰Ã⍰Â⍰Ã⍰Â⍰Ã⍰Â⍰Ã⍰Â© Spa Cuis
ine',
       'Punjab Grill', 'Belgian Waffle Factory', 'O.G. Variar & Sons',
       'Flechazo', 'The Pizza Bakery', 'AB's - Absolute Barbecues',
       'Barbecue by Punjab Grill', 'CTR', 'The Black Pearl',
       'House Of Commons', 'The Big Barbeque', 'ECHOES Koramangala', 'You Mee',
       'TBC Sky Lounge', 'Toit', 'Taaza Thindi', 'Maziga'],
      dtype='object', name='name')
```

In [96]:
```
rating[0:20].rate
```

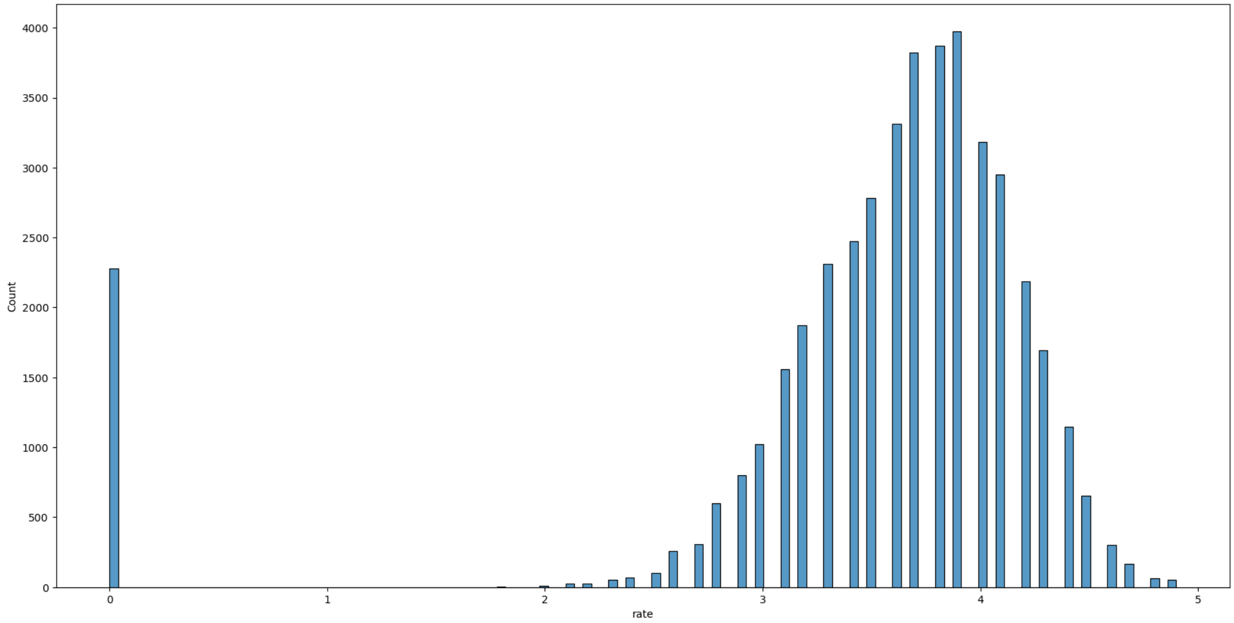
Out[96]:

```
name
Asia Kitchen By Mainland China
4.900000
Byg Brewski Brewing Company
4.900000
SantÃ�Â�Ã�Â�Ã�Â�Ã�Â�Ã�Â�Ã�Â�Ã�Â�Ã�Â�Ã�Â�Ã�Â�Ã�Â�Ã�Â�Ã�Â�Ã�Â�Ã�Â�Ã�Â© Spa Cuisine
4.900000
Punjab Grill
4.871429
Belgian Waffle Factory
4.844828
O.G. Variar & Sons
4.800000
Flechazo
4.800000
The Pizza Bakery
4.800000
AB's - Absolute Barbecues
4.789474
Barbecue by Punjab Grill
4.750000
CTR
4.750000
The Black Pearl
4.727778
House Of Commons
4.723810
The Big Barbeque
4.700000
ECHOES Koramangala
4.700000
You Mee
4.700000
TBC Sky Lounge
4.700000
Toit
4.700000
Taaza Thindi
4.700000
Maziga
4.700000
Name: rate, dtype: float64
```

# distribution of Ratings

In [111…

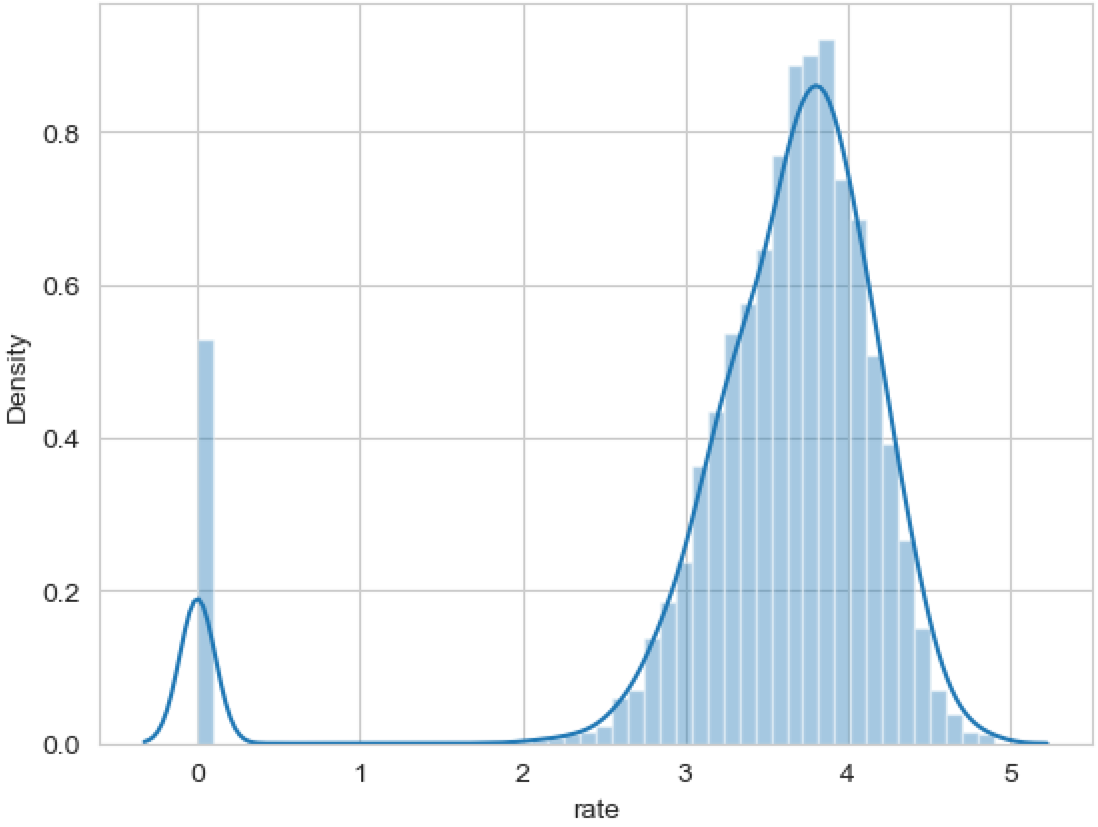
```python
plt.figure(figsize=(20,10))
sns.histplot(data['rate'])
plt.show()
```



In [112…

```
sns.set_style('whitegrid')
sns.distplot(data['rate'])

plt.show()
```

```
C:\Users\Tesla\AppData\Local\Temp\ipykernel_3280\2668192239.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['rate'])
```



In [113…

```
from scipy.stats import normaltest

DataToTest = data['rate']

stat,p = normaltest(DataToTest)

print("stat=%0.2f,p=%0.30f" % (stat,p))

if p>0.05:
    print("Normal distribution")
else:
    print("Not a normal distribution")
```

```
stat=24555.12,p=0.000000000000000000000000000000
Not a normal distribution
```

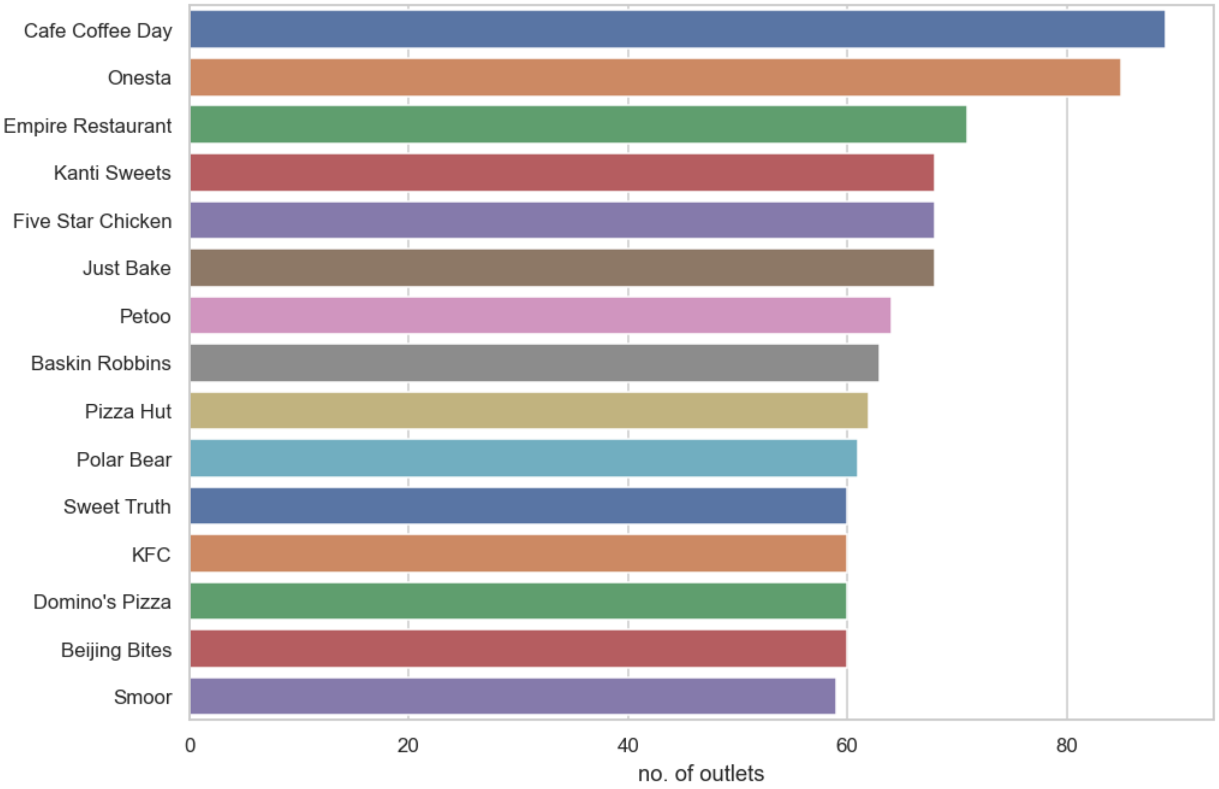
# Top restaurant chains

In [117…

```
data.name.value_counts()
```

```
Out[117]: 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 [295…
plt.figure(figsize=(10,7),dpi=110)

chains = data['name'].value_counts()[0:15]

sns.barplot(x=chains,y=chains.index,palette='deep')
plt.xlabel("no. of outlets")

plt.show()
```



```
In [128…
plt.show()
```

# How many restaurants does not accept orders online

```
In [129…
data.columns
```

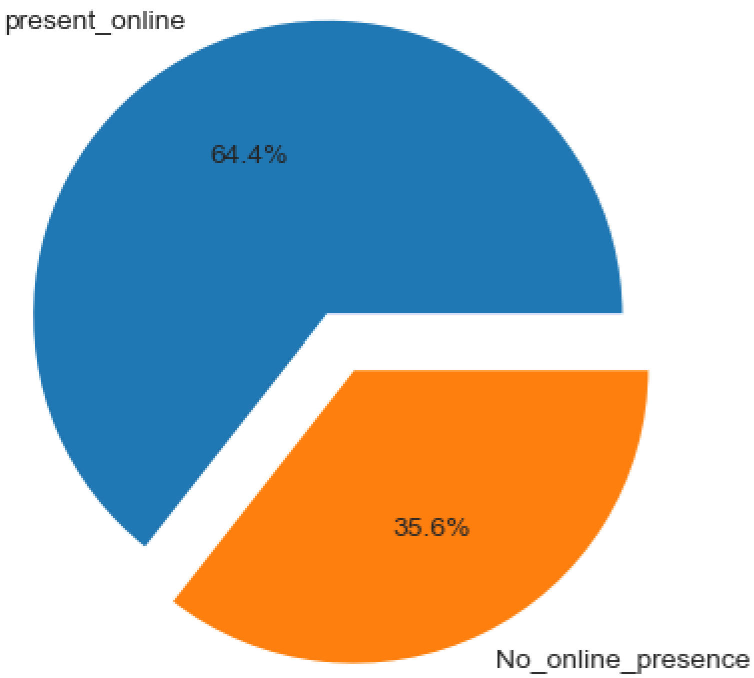
Out[129]:
```
Index(['name', 'online_order', 'book_table', 'rate', 'votes', 'location',
       'rest_type', 'dish_liked', 'cuisines', 'approx_cost(for two people)',
       'reviews_list'],
      dtype='object')
```

In [130…
```python
x_online=data.online_order.value_counts()
```

In [131…
```python
x_online
```

Out[131]:
```
Yes    28308
No     15634
Name: online_order, dtype: int64
```

In [146…
```python
plt.pie(x_online,labels=["present_online","No_online_presence"],explode=[0.1,0.1],auto
plt.show()
```



# how many restaurants have Table Bookings

In [148…
```python
data.columns
```

Out[148]:
```
Index(['name', 'online_order', 'book_table', 'rate', 'votes', 'location',
       'rest_type', 'dish_liked', 'cuisines', 'approx_cost(for two people)',
       'reviews_list'],
      dtype='object')
```

In [149…
```python
data.book_table.value_counts()
```

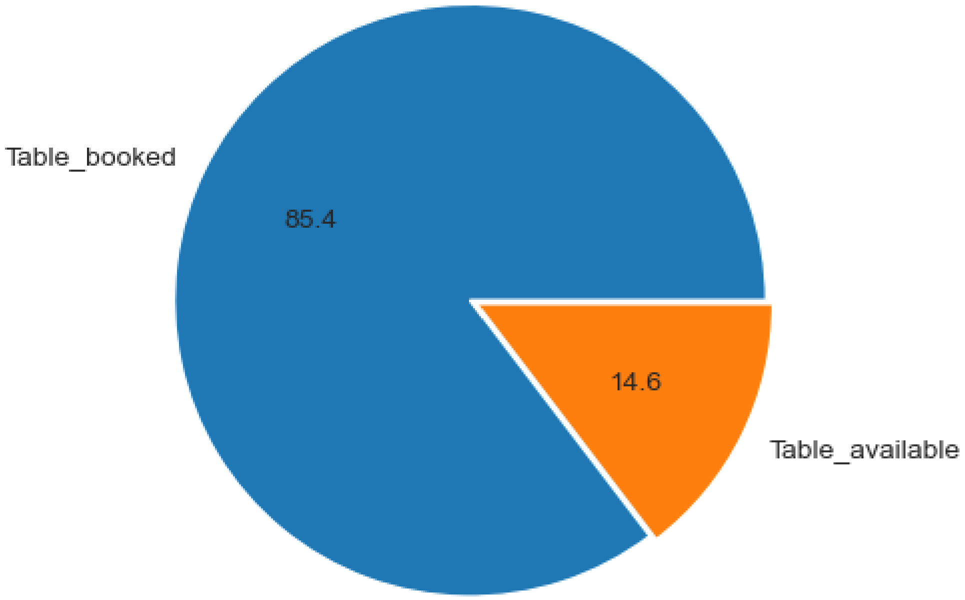
Out[149]:
```
No     37509
Yes     6433
Name: book_table, dtype: int64
```

In [150…
```python
plt.figure(figsize=(10,6))
```

Out[150]:
```
<Figure size 1000x600 with 0 Axes>

<Figure size 1000x600 with 0 Axes>
```

In [154…
```
plt.pie(data.book_table.value_counts(),labels=['Table_booked','Table_available'],expl
```

Out[154]:
```
([<matplotlib.patches.Wedge at 0x2066983c100>,
  <matplotlib.patches.Wedge at 0x206695a3430>],
 [Text(-0.9856961240827233, 0.4882654513359477, 'Table_booked'),
  Text(1.0080983087209672, -0.4993623934117643, 'Table_available')],
 [Text(-0.537652431317849, 0.26632660981960776, '85.4'),
  Text(0.5600546159560928, -0.27742355189542456, '14.6')])
```



# Analysis on type of restarants

In [155…
```
data.columns
```

Out[155]:
```
Index(['name', 'online_order', 'book_table', 'rate', 'votes', 'location',
       'rest_type', 'dish_liked', 'cuisines', 'approx_cost(for two people)',
       'reviews_list'],
      dtype='object')
```

In [156…
```
data.rest_type
```

Out[156]:

```
0                Casual Dining
1                Casual Dining
2          Cafe, Casual Dining
3                  Quick Bites
4                Casual Dining
                      ...
51709       Casual Dining, Bar
51711       Casual Dining, Bar
51712                      Bar
51715                      Bar
51716       Bar, Casual Dining
Name: rest_type, Length: 43942, dtype: object
```

In [160…

```
data.rest_type.unique()
```

Out[160]:

```
array(['Casual Dining', 'Cafe, Casual Dining', 'Quick Bites',
       'Casual Dining, Cafe', 'Cafe', 'Quick Bites, Cafe',
       'Cafe, Quick Bites', 'Delivery', 'Mess', 'Dessert Parlor',
       'Bakery, Dessert Parlor', 'Pub', 'Bakery', 'Takeaway, Delivery',
       'Fine Dining', 'Beverage Shop', 'Sweet Shop', 'Bar',
       'Dessert Parlor, Sweet Shop', 'Bakery, Quick Bites',
       'Sweet Shop, Quick Bites', 'Kiosk', 'Food Truck',
       'Quick Bites, Dessert Parlor', 'Beverage Shop, Quick Bites',
       'Beverage Shop, Dessert Parlor', 'Takeaway', 'Pub, Casual Dining',
       'Casual Dining, Bar', 'Dessert Parlor, Beverage Shop',
       'Quick Bites, Bakery', 'Microbrewery, Casual Dining', 'Lounge',
       'Bar, Casual Dining', 'Food Court', 'Cafe, Bakery', nan, 'Dhaba',
       'Quick Bites, Sweet Shop', 'Microbrewery',
       'Food Court, Quick Bites', 'Quick Bites, Beverage Shop',
       'Pub, Bar', 'Casual Dining, Pub', 'Lounge, Bar',
       'Dessert Parlor, Quick Bites', 'Food Court, Dessert Parlor',
       'Casual Dining, Sweet Shop', 'Food Court, Casual Dining',
       'Casual Dining, Microbrewery', 'Lounge, Casual Dining',
       'Cafe, Food Court', 'Beverage Shop, Cafe', 'Cafe, Dessert Parlor',
       'Dessert Parlor, Cafe', 'Dessert Parlor, Bakery',
       'Microbrewery, Pub', 'Bakery, Food Court', 'Club',
       'Quick Bites, Food Court', 'Bakery, Cafe', 'Pub, Cafe',
       'Casual Dining, Irani Cafee', 'Fine Dining, Lounge',
       'Bar, Quick Bites', 'Confectionery', 'Pub, Microbrewery',
       'Microbrewery, Lounge', 'Fine Dining, Microbrewery',
       'Fine Dining, Bar', 'Dessert Parlor, Kiosk', 'Bhojanalya',
       'Casual Dining, Quick Bites', 'Cafe, Bar', 'Casual Dining, Lounge',
       'Bakery, Beverage Shop', 'Microbrewery, Bar', 'Cafe, Lounge',
       'Bar, Pub', 'Lounge, Cafe', 'Club, Casual Dining',
       'Quick Bites, Mess', 'Quick Bites, Meat Shop',
       'Quick Bites, Kiosk', 'Lounge, Microbrewery',
       'Food Court, Beverage Shop', 'Dessert Parlor, Food Court',
       'Bar, Lounge'], dtype=object)
```

In [161…

```
len(data.rest_type.unique())
```

Out[161]:

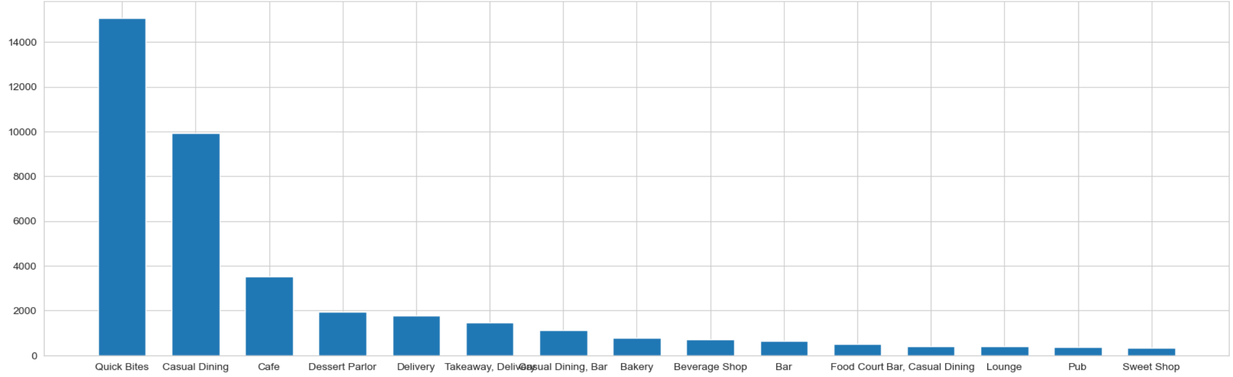
```
88
```

In [162…

```
data.rest_type.value_counts()
```

Out[162]:
```
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 [173…
```
plt.figure(figsize=(20,6))
rest_typ = data.rest_type.value_counts()[0:15]
plt.bar(rest_typ.index,rest_typ,width=0.65)
plt.show()
```



# which restaurant has highest rating count given

In [174…
```
data.columns
```

Out[174]:
```
Index(['name', 'online_order', 'book_table', 'rate', 'votes', 'location',
       'rest_type', 'dish_liked', 'cuisines', 'approx_cost(for two people)',
       'reviews_list'],
      dtype='object')
```

In [178…
```
data.votes
```

Out[178]:
```
0        775
1        787
2        918
3         88
4        166
        ... 
51709     34
51711     81
51712     27
51715    236
51716     13
Name: votes, Length: 43942, dtype: int64
```

In [177…
```
voting = data.groupby('name')[['votes']].mean()
voting
```

Out[177]:

| | votes |
|---|---|
| **name** | |
| **#FeelTheROLL** | 7.000000 |
| **#L-81 Cafe** | 48.000000 |
| **#refuel** | 37.000000 |
| **1000 B.C** | 49.000000 |
| **100Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â□Ã□Â°C** | 41.000000 |
| **...** | ... |
| **i-Bar - The Park Bangalore** | 619.727273 |
| **iFruit Live Ice Creams** | 10.000000 |
| **iSpice Resto Cafe** | 30.142857 |
| **nu.tree** | 206.142857 |
| **re:cess - Hilton Bangalore Embassy GolfLinks** | 146.000000 |

7162 rows × 1 columns

In [179…

```
voting['votes'].describe()
```

Out[179]:

```
count     7162.000000
mean       201.484245
std        607.015608
min          0.000000
25%         10.400000
50%         38.750000
75%        146.797414
max      16588.500000
Name: votes, dtype: float64
```

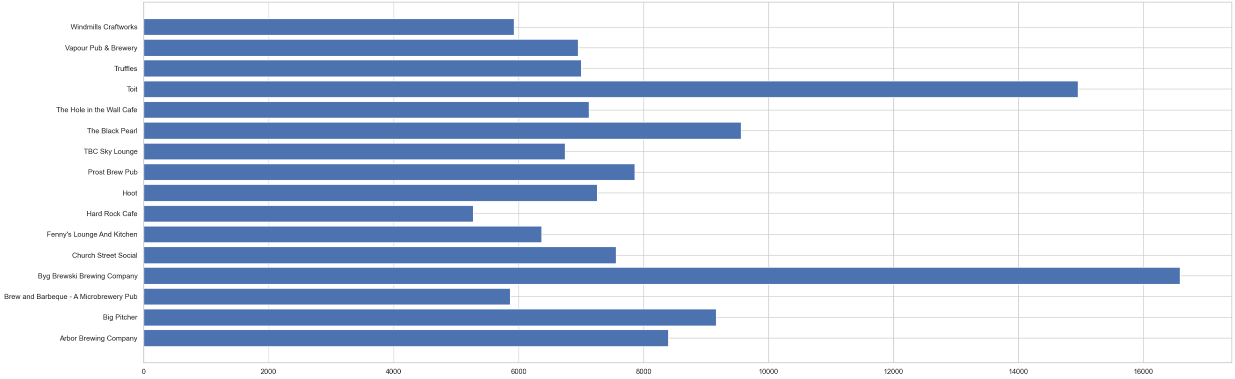
In [183…

```
high_vot = voting[voting['votes'] >5000]
```

# high_vot

In [296…

```
plt.figure(figsize=(30,10),dpi=100)
plt.barh(high_vot.index,high_vot['votes'])
plt.show()
```



# Let's analyse the highly rated restaurants and worst restaurants based on number of ratings given

In [195…

```
data.columns
```

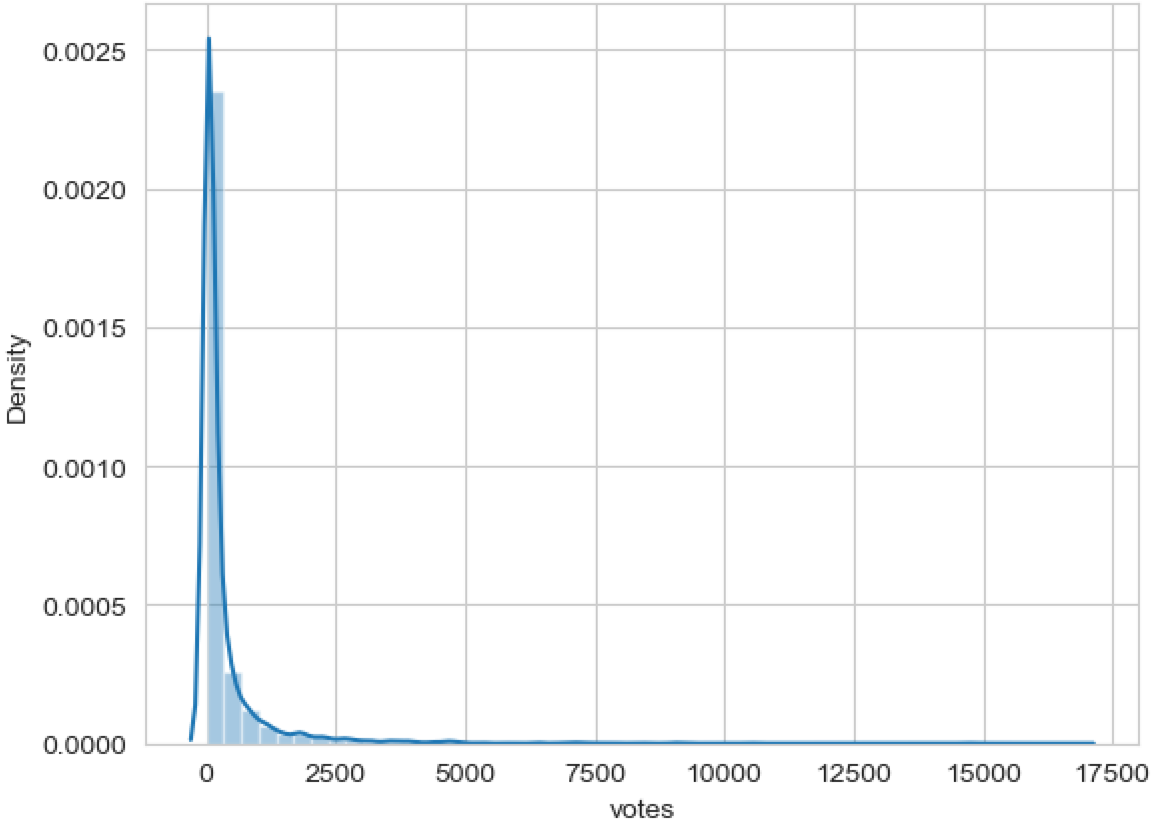
Out[195]:

```
Index(['name', 'online_order', 'book_table', 'rate', 'votes', 'location',
       'rest_type', 'dish_liked', 'cuisines', 'approx_cost(for two people)',
       'reviews_list'],
      dtype='object')
```

In [200…

```
sns.distplot(data['votes'])
plt.show()
```

```
C:\Users\Tesla\AppData\Local\Temp\ipykernel_3280\2731871164.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(data['votes'])
```



In [201…

```
voting_rating=data.groupby('name')[['votes']].mean().sort_values('votes',ascending=Fal
```

In [202…

```
voting_rating
```

Out[202]:

| | votes |
|---|---|
| **name** | |
| **Byg Brewski Brewing Company** | 16588.500000 |
| **Toit** | 14956.000000 |
| **The Black Pearl** | 9562.333333 |
| **Big Pitcher** | 9164.500000 |
| **Arbor Brewing Company** | 8396.545455 |
| **...** | ... |
| **NutriFit** | 0.000000 |
| **Cross Roads Inn** | 0.000000 |
| **Crown Resto** | 0.000000 |
| **Nourich** | 0.000000 |
| **Lazzet Lee** | 0.000000 |

7162 rows × 1 columns

In [203…

```
voting_rating['name'] = voting_rating.index
```

In [204…

```
voting_rating.index
```

Out[204]:

```
Index(['Byg Brewski Brewing Company', 'Toit', 'The Black Pearl', 'Big Pitcher',
       'Arbor Brewing Company', 'Prost Brew Pub', 'Church Street Social',
       'Hoot', 'The Hole in the Wall Cafe', 'Truffles',
       ...
       'ONS Food', 'Costal Kadai', 'Crave Bit Cafe',
       'CraveBelly Veg & Non veg', 'Cravisthan', 'NutriFit', 'Cross Roads Inn',
       'Crown Resto', 'Nourich', 'Lazzet Lee'],
      dtype='object', name='name', length=7162)
```

In [205…

```
voting_rating = voting_rating.reset_index(drop=True)

voting_rating.head()
```

Out[205]:

| | votes | name |
|---|---|---|
| **0** | 16588.500000 | Byg Brewski Brewing Company |
| **1** | 14956.000000 | Toit |
| **2** | 9562.333333 | The Black Pearl |
| **3** | 9164.500000 | Big Pitcher |
| **4** | 8396.545455 | Arbor Brewing Company |

In [206…

```
voting_rating=pd.merge(voting_rating,data[['rate','name']])
voting_rating
```

Out[206]:

| | votes | name | rate |
|---|---|---|---|
| **0** | 16588.5 | Byg Brewski Brewing Company | 4.9 |
| **1** | 16588.5 | Byg Brewski Brewing Company | 4.9 |
| **2** | 16588.5 | Byg Brewski Brewing Company | 4.9 |
| **3** | 16588.5 | Byg Brewski Brewing Company | 4.9 |
| **4** | 16588.5 | Byg Brewski Brewing Company | 4.9 |
| **...** | ... | ... | ... |
| **43937** | 0.0 | Lazzet Lee | 0.0 |
| **43938** | 0.0 | Lazzet Lee | 0.0 |
| **43939** | 0.0 | Lazzet Lee | 0.0 |
| **43940** | 0.0 | Lazzet Lee | 0.0 |
| **43941** | 0.0 | Lazzet Lee | 0.0 |

43942 rows × 3 columns

In [207…

```
vote_top = voting_rating[voting_rating['votes'] > 5000]
vote_top.head()
```

Out[207]:

| | votes | name | rate |
|---|---|---|---|
| **0** | 16588.5 | Byg Brewski Brewing Company | 4.9 |
| **1** | 16588.5 | Byg Brewski Brewing Company | 4.9 |
| **2** | 16588.5 | Byg Brewski Brewing Company | 4.9 |
| **3** | 16588.5 | Byg Brewski Brewing Company | 4.9 |
| **4** | 16588.5 | Byg Brewski Brewing Company | 4.9 |

In [208…

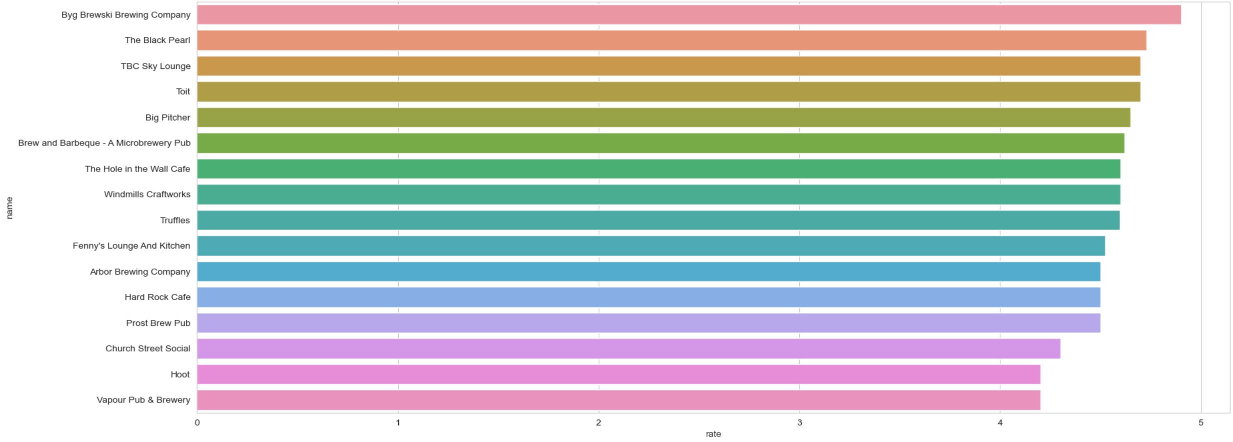
```
vote_top = vote_top.groupby('name')[['rate']].mean().sort_values('rate',ascending=Fals
vote_top
```

Out[208]:

| | rate |
|---|---|
| **name** | |
| **Byg Brewski Brewing Company** | 4.900000 |
| **The Black Pearl** | 4.727778 |
| **TBC Sky Lounge** | 4.700000 |
| **Toit** | 4.700000 |
| **Big Pitcher** | 4.650000 |
| **Brew and Barbeque - A Microbrewery Pub** | 4.620000 |
| **The Hole in the Wall Cafe** | 4.600000 |
| **Windmills Craftworks** | 4.600000 |
| **Truffles** | 4.595349 |
| **Fenny's Lounge And Kitchen** | 4.521429 |
| **Arbor Brewing Company** | 4.500000 |
| **Hard Rock Cafe** | 4.500000 |
| **Prost Brew Pub** | 4.500000 |
| **Church Street Social** | 4.300000 |
| **Hoot** | 4.200000 |
| **Vapour Pub & Brewery** | 4.200000 |

In [214…

```
plt.figure(figsize=(20,8),dpi=100)

sns.barplot(x=vote_top['rate'],y=vote_top.index,orient='h')

plt.show()
```



In [215…

```
bad_count = voting_rating[(voting_rating['rate'] < 3) & (voting_rating['rate'] > 0)]
bad_count
```

Out[215]:

| | votes | name | rate |
|---|---|---|---|
| **3725** | 967.681818 | Kritunga Restaurant | 2.6 |
| **3726** | 967.681818 | Kritunga Restaurant | 2.9 |
| **3727** | 967.681818 | Kritunga Restaurant | 2.9 |
| **3742** | 967.681818 | Kritunga Restaurant | 2.7 |
| **3743** | 967.681818 | Kritunga Restaurant | 2.7 |
| **...** | ... | ... | ... |
| **42110** | 4.000000 | Special Biriyani | 2.9 |
| **42111** | 4.000000 | Special Biriyani | 2.9 |
| **42198** | 4.000000 | Sandesh Restaurant | 2.9 |
| **42311** | 2.500000 | Kushi Meals | 2.8 |
| **42312** | 2.500000 | Kushi Meals | 2.8 |

2257 rows × 3 columns

In [ ]:

In [218…

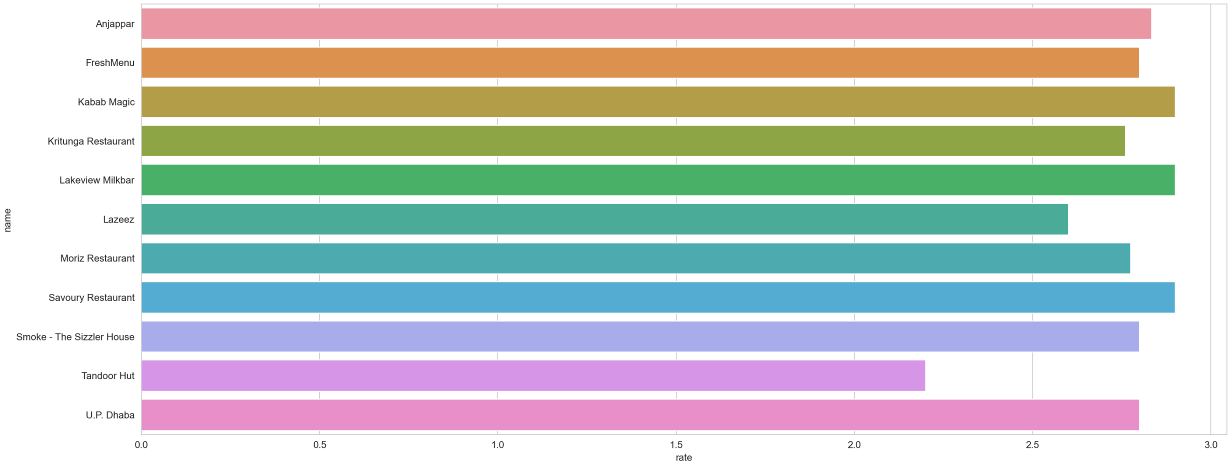
```
bad_resturants = bad_count[bad_count['votes'] > 500].groupby('name')[['rate']].mean()
bad_resturants
```

Out[218]:

| | rate |
|---|---|
| **name** | |
| **Anjappar** | 2.833333 |
| **FreshMenu** | 2.800000 |
| **Kabab Magic** | 2.900000 |
| **Kritunga Restaurant** | 2.760000 |
| **Lakeview Milkbar** | 2.900000 |
| **Lazeez** | 2.600000 |
| **Moriz Restaurant** | 2.775000 |
| **Savoury Restaurant** | 2.900000 |
| **Smoke - The Sizzler House** | 2.800000 |
| **Tandoor Hut** | 2.200000 |
| **U.P. Dhaba** | 2.800000 |

In [221…

```
plt.figure(figsize=(20,8),dpi=160)

sns.barplot(x=bad_resturants['rate'],y=bad_resturants.index)

plt.show()
```



# Location have highest number of restaurants

In [222…

```
data.head()
```

Out[222]:

| | name | online_order | book_table | rate | votes | location | rest_type | dish_liked | cuisines | ap |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Jalsa | Yes | Yes | 4.1 | 775 | Banashankari | Casual Dining | Pasta, Lunch Buffet, Masala Papad, Paneer Laja... | North Indian, Mughlai, Chinese | |
| **1** | Spice Elephant | Yes | No | 4.1 | 787 | Banashankari | Casual Dining | Momos, Lunch Buffet, Chocolate Nirvana, Thai G... | Chinese, North Indian, Thai | |
| **2** | San Churro Cafe | Yes | No | 3.8 | 918 | Banashankari | Cafe, Casual Dining | Churros, Cannelloni, Minestrone Soup, Hot Choc... | Cafe, Mexican, Italian | |
| **3** | Addhuri Udupi Bhojana | No | No | 3.7 | 88 | Banashankari | Quick Bites | Masala Dosa | South Indian, North Indian | |
| **4** | Grand Village | No | No | 3.8 | 166 | Basavanagudi | Casual Dining | Panipuri, Gol Gappe | North Indian, Rajasthani | |

In [223…

```
data.columns
```

Out[223]:

```
Index(['name', 'online_order', 'book_table', 'rate', 'votes', 'location',
       'rest_type', 'dish_liked', 'cuisines', 'approx_cost(for two people)',
       'reviews_list'],
      dtype='object')
```

In [225…

```
data.location.value_counts()[0:20]
```

Out[225]:

```
BTM                      4261
Koramangala 5th Block    2381
HSR                      2128
Indiranagar              1936
JP Nagar                 1849
Jayanagar                1718
Whitefield               1693
Marathahalli             1503
Bannerghatta Road        1324
Koramangala 6th Block    1111
Koramangala 7th Block    1089
Brigade Road             1084
Bellandur                1078
Koramangala 1st Block     965
Electronic City           964
Sarjapur Road             919
Ulsoor                    901
Koramangala 4th Block     864
MG Road                   836
Banashankari              805
Name: location, dtype: int64
```

In [228…

```
location_restro = data[['location']].value_counts()[0:20]
location_restro
```

Out[228]:

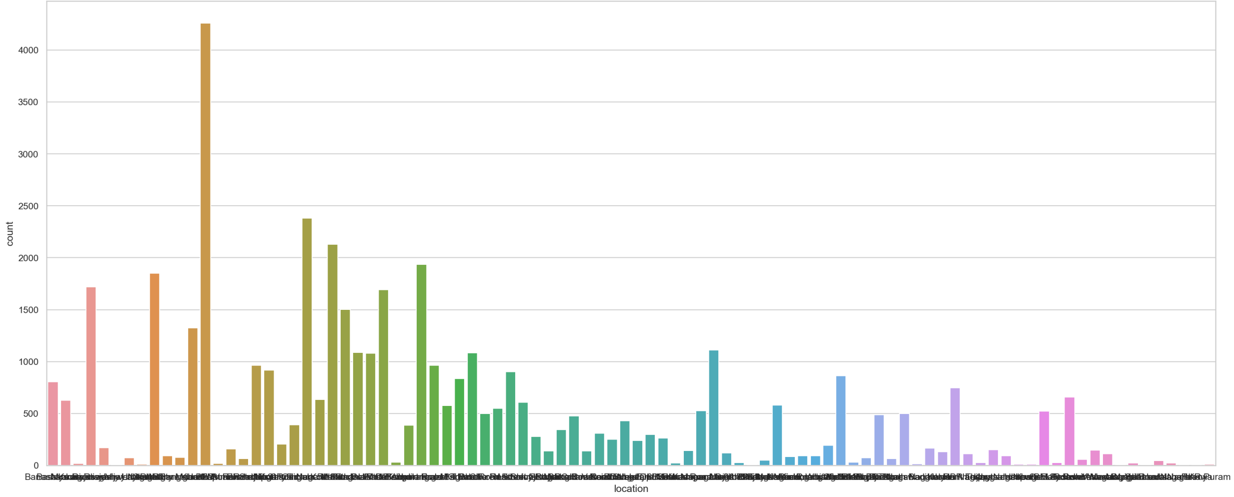
```
location
BTM                      4261
Koramangala 5th Block    2381
HSR                      2128
Indiranagar              1936
JP Nagar                 1849
Jayanagar                1718
Whitefield               1693
Marathahalli             1503
Bannerghatta Road        1324
Koramangala 6th Block    1111
Koramangala 7th Block    1089
Brigade Road             1084
Bellandur                1078
Koramangala 1st Block     965
Electronic City           964
Sarjapur Road             919
Ulsoor                    901
Koramangala 4th Block     864
MG Road                   836
Banashankari              805
dtype: int64
```

In [244…

```
plt.figure(figsize=(25,10),dpi=150)
sns.set_theme(style="whitegrid")
sns.countplot(x='location',data=data)
```

Out[244]:

```
<Axes: xlabel='location', ylabel='count'>
```



# Identify the top 10 cuisines

In [245…

```
data.columns
```

Out[245]:

```
Index(['name', 'online_order', 'book_table', 'rate', 'votes', 'location',
       'rest_type', 'dish_liked', 'cuisines', 'approx_cost(for two people)',
       'reviews_list'],
      dtype='object')
```

In [250…

```
cusine=data.cuisines.value_counts()[0:20]
cusine
```

Out[250]:

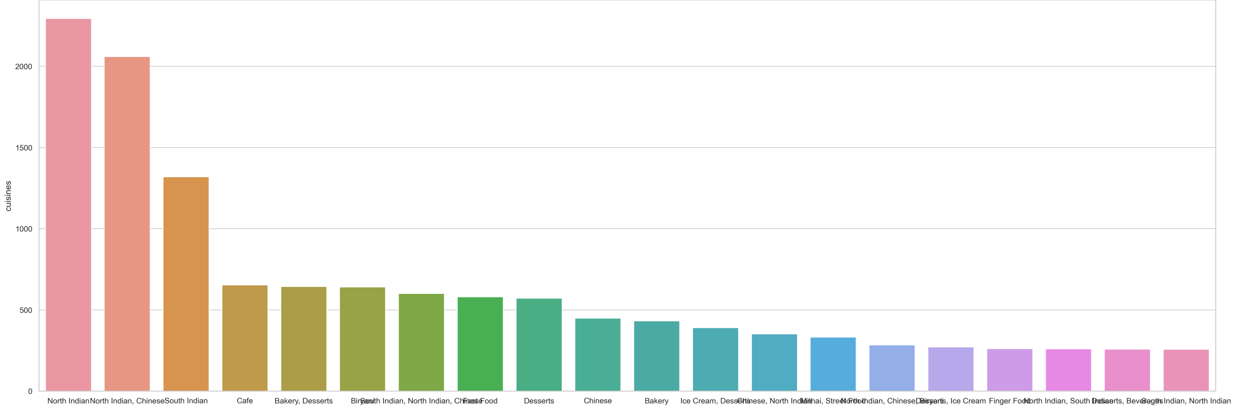
```
North Indian                              2294
North Indian, Chinese                     2060
South Indian                              1320
Cafe                                       653
Bakery, Desserts                           644
Biryani                                    641
South Indian, North Indian, Chinese        601
Fast Food                                  580
Desserts                                   572
Chinese                                    449
Bakery                                     432
Ice Cream, Desserts                        390
Chinese, North Indian                      352
Mithai, Street Food                        332
North Indian, Chinese, Biryani             284
Desserts, Ice Cream                        272
Finger Food                                261
North Indian, South Indian                 260
Desserts, Beverages                        258
South Indian, North Indian                 257
Name: cuisines, dtype: int64
```

In [259…

```
plt.figure(figsize=(30,10),dpi=300)
sns.set_theme(style="whitegrid")
sns.barplot(x=cusine.index,y=cusine,width=0.78)
plt.show
```

Out[259]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



# distribution for cost of 2 people

In [260…

```
data.columns
```

Out[260]:

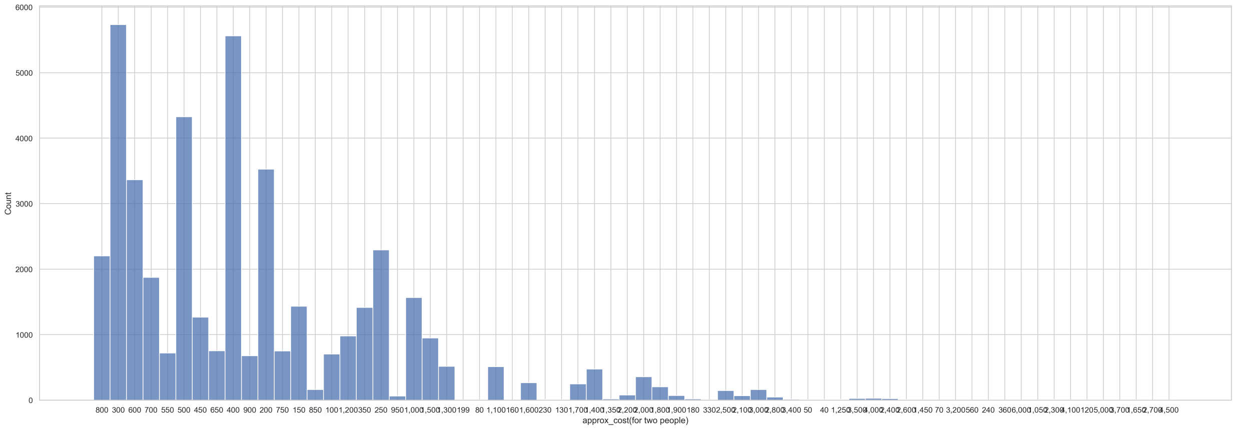
```
Index(['name', 'online_order', 'book_table', 'rate', 'votes', 'location',
       'rest_type', 'dish_liked', 'cuisines', 'approx_cost(for two people)',
       'reviews_list'],
      dtype='object')
```

In [264…

```
plt.figure(figsize=(30,10),dpi=300)
sns.histplot(x=data['approx_cost(for two people)'])
plt.show
```

Out[264]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [265…

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

Out[265]:

```
252
```

In [267…

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

In [268…

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

Out[268]:

```
252
```

In [269…

```
data.dropna(axis=0,subset=['approx_cost(for two people)'],inplace=True)
```

In [270…

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

Out[270]:

```
0
```

In [281…

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

Out[281]:

```
300      5735
400      5562
500      4326
200      3527
600      3365
         ...
560         1
360         1
5,000       1
3,700       1
160         1
Name: approx_cost(for two people), Length: 66, dtype: int64
```

In [282…

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

In [283…

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

Out[283]:

```
0         800
1         800
2         800
3         300
4         600
         ...
51709     800
51711     800
51712    1500
51715    2500
51716    1500
Name: approx_cost(for two people), Length: 43690, dtype: object
```

In [286…

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

In [287…

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

Out[287]:

```
0         800
1         800
2         800
3         300
4         600
         ...
51709     800
51711     800
51712    1500
51715    2500
51716    1500
Name: approx_cost(for two people), Length: 43690, dtype: int32
```

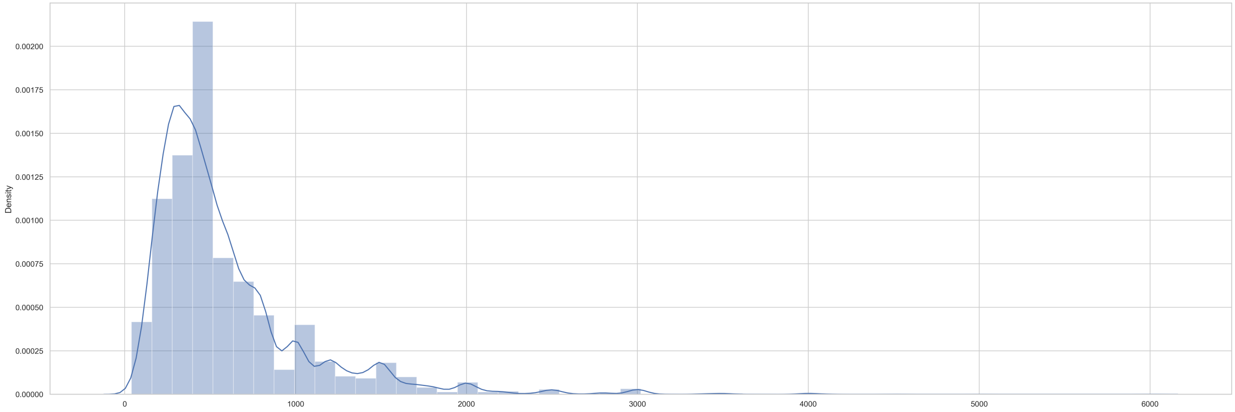
In [289…

```
plt.figure(figsize=(30,10),dpi=300)
sns.distplot(x=data['approx_cost(for two people)'])
plt.show
```

```
C:\Users\Tesla\AppData\Local\Temp\ipykernel_3280\3572057511.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(x=data['approx_cost(for two people)'])
```

Out[289]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [290…

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

Out[290]:

```
count    43690.000000
mean       594.270222
std        458.761551
min         40.000000
25%        300.000000
50%        450.000000
75%        700.000000
max       6000.000000
Name: approx_cost(for two people), dtype: float64
```

In [291…

```
data.describe()
```

Out[291]:

| | rate | votes | approx_cost(for two people) |
|---|---|---|---|
| **count** | 43690.000000 | 43690.000000 | 43690.000000 |
| **mean** | 3.507924 | 333.435294 | 594.270222 |
| **std** | 0.926878 | 862.980555 | 458.761551 |
| **min** | 0.000000 | 0.000000 | 40.000000 |
| **25%** | 3.300000 | 17.000000 | 300.000000 |
| **50%** | 3.700000 | 64.000000 | 450.000000 |
| **75%** | 4.000000 | 254.000000 | 700.000000 |
| **max** | 4.900000 | 16832.000000 | 6000.000000 |

In [292…

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

In [293…

```
return_budget('Whitefield','Casual Dining',1000)
```

Out[293]:

```
array(['The Square Table', "Galito's", 'Suryawanshi', 'NH8',
       "Kapoor's Cafe", 'Salt Mango Tree', 'Chez Mariannick',
       'Imperio Restaurant', 'Nasi And Mee', 'Sorse'], dtype=object)
```

# Insights:

1. Asia Kitchen By Mainland China,Spa Cuisine,Byg Brewski Brewing Company,Punjab Grill were the best rated Restaurants.
2. Cafe Coffee Day, Onesta, Empire Restaurant, Just Bake ,Kanti sweets were the top restaurant chains.
3. 64.4% of resturants accept online orders where as 35.6% donot accept online orders.
4. Based on most voted resaturants Byg Brewski Brewing Company and The Black Pearl were the best rated restaurants.
5. Based on most voted resaturants Lazeez and Tandoor Hut were the worst rated restaurants.
6. BTM, Koramangala 5th Block has the most numbers of restaurants.
7. North Indian and Chinese were the top favourite of the people in Bangalore.
8. Cost for 2 people for most restaurants was varing from 500 to 1000 bucks.

In [ ]: