Zomato Data Set Analysis and Visualization

Importing Libraries

In [4]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Reading CSV

In [6]:
```
df=pd.read_csv(r"D:\CAPSTONE PROJECT_DEPLOYMENT\11. CAPSTONE PROJECT_DEPLOYMENT\
df.head()
```

Out[6]:

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

In [7]:

```
df.shape
```

Out[7]:

```
(51717, 17)
```

In [8]:

```
df.columns
```

Out[8]:

```
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 [9]:

```
df=df.drop(['url','address','phone','menu_item','dish_liked','reviews_list'], ax
df.head(5)
```

Out[9]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines
0	Jalsa	Yes	Yes	4.1/5	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese
1	Spice Elephant	Yes	No	4.1/5	787	Banashankari	Casual Dining	Chinese, North Indian, Thai
2	San Churro Cafe	Yes	No	3.8/5	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian
3	Addhuri Udupi Bhojana	No	No	3.7/5	88	Banashankari	Quick Bites	South Indian, North Indian
4	Grand Village	No	No	3.8/5	166	Basavanagudi	Casual Dining	North Indian, Rajasthani

In [10]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51717 entries, 0 to 51716
Data columns (total 11 columns):
 #   Column                       Non-Null Count  Dtype 
---  ------                       --------------  ----- 
 0   name                         51717 non-null  object
 1   online_order                 51717 non-null  object
 2   book_table                   51717 non-null  object
 3   rate                         43942 non-null  object
 4   votes                        51717 non-null  int64 
 5   location                     51696 non-null  object
 6   rest_type                    51490 non-null  object
 7   cuisines                     51672 non-null  object
 8   approx_cost(for two people)  51371 non-null  object
 9   listed_in(type)              51717 non-null  object
 10  listed_in(city)              51717 non-null  object
dtypes: int64(1), object(10)
memory usage: 4.3+ MB
```

Dropping Duplicates

In [12]:

```
df.drop_duplicates(inplace=True)
df.shape
```

Out[12]:

```
(51609, 11)
```

Cleaning Rate Column

```
In [14]: df['rate'].unique()
```

```
Out[14]: 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)
```

Removing "NEW" , "-" and "/5" from Rate Column

```
In [16]: def handlerate(value):
             if(value=='NEW' or value=='-'):
                 return np.nan
             else:
                 value = str(value).split('/')
                 value = value[0]
                 return float(value)

         df['rate'] = df['rate'].apply(handlerate)
         df['rate'].head()
```

```
Out[16]: 0    4.1
         1    4.1
         2    3.8
         3    3.7
         4    3.8
         Name: rate, dtype: float64
```

Filling Null values i Rate Column with Mean

```
In [18]: df['rate'].fillna(df['rate'].mean(), inplace = True)
         df['rate'].isnull().sum()
```

```
C:\Users\chitt\AppData\Local\Temp\ipykernel_2808\1687008678.py:1: FutureWarning:
A value is trying to be set on a copy of a DataFrame or Series through chained as
signment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work becau
se the intermediate object on which we are setting values always behaves as a cop
y.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.meth
od({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to pe
rform the operation inplace on the original object.


  df['rate'].fillna(df['rate'].mean(), inplace = True)
```

```
Out[18]: 0
```

```
In [19]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 51609 entries, 0 to 51716
Data columns (total 11 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   name                         51609 non-null  object 
 1   online_order                 51609 non-null  object 
 2   book_table                   51609 non-null  object 
 3   rate                         51609 non-null  float64
 4   votes                        51609 non-null  int64  
 5   location                     51588 non-null  object 
 6   rest_type                    51382 non-null  object 
 7   cuisines                     51564 non-null  object 
 8   approx_cost(for two people)  51265 non-null  object 
 9   listed_in(type)              51609 non-null  object 
 10  listed_in(city)              51609 non-null  object 
dtypes: float64(1), int64(1), object(9)
memory usage: 4.7+ MB
```

Dropping Null Values

In [21]:

```
df.dropna(inplace= True)
df.head()
```

Out[21]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	Chinese, North Indian, Thai
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian
3	Addhuri Udupi Bhojana	No	No	3.7	88	Banashankari	Quick Bites	South Indian, North Indian
4	Grand Village	No	No	3.8	166	Basavanagudi	Casual Dining	North Indian, Rajasthani

In [37]:

```
df.rename(columns = {'approx_cost(for two people)':'Cost2plates', 'listed_in(typ
df.head()
```

Out[37]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	Chinese, North Indian, Thai
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian
3	Addhuri Udupi Bhojana	No	No	3.7	88	Banashankari	Quick Bites	South Indian, North Indian
4	Grand Village	No	No	3.8	166	Basavanagudi	Casual Dining	North Indian, Rajasthani

In [39]:

```
df['location'].unique()
```

Out[39]:

```
array(['Banashankari', 'Basavanagudi', 'Mysore Road', 'Jayanagar',
       'Kumaraswamy Layout', 'Rajarajeshwari Nagar', 'Vijay Nagar',
       'Uttarahalli', 'JP Nagar', 'South Bangalore', 'City Market',
       'Nagarbhavi', 'Bannerghatta Road', 'BTM', 'Kanakapura Road',
       'Bommanahalli', 'CV Raman Nagar', 'Electronic City', 'HSR',
       'Marathahalli', 'Wilson Garden', 'Shanti Nagar',
       'Koramangala 5th Block', 'Koramangala 8th Block', 'Richmond Road',
       'Koramangala 7th Block', 'Jalahalli', 'Koramangala 4th Block',
       'Bellandur', 'Sarjapur Road', 'Whitefield', 'East Bangalore',
       'Old Airport Road', 'Indiranagar', 'Koramangala 1st Block',
       'Frazer Town', 'RT Nagar', 'MG Road', 'Brigade Road',
       'Lavelle Road', 'Church Street', 'Ulsoor', 'Residency Road',
       'Shivajinagar', 'Infantry Road', 'St. Marks Road',
       'Cunningham Road', 'Race Course Road', 'Commercial Street',
       'Vasanth Nagar', 'HBR Layout', 'Domlur', 'Ejipura',
       'Jeevan Bhima Nagar', 'Old Madras Road', 'Malleshwaram',
       'Seshadripuram', 'Kammanahalli', 'Koramangala 6th Block',
       'Majestic', 'Langford Town', 'Central Bangalore', 'Sanjay Nagar',
       'Brookefield', 'ITPL Main Road, Whitefield',
       'Varthur Main Road, Whitefield', 'KR Puram',
       'Koramangala 2nd Block', 'Koramangala 3rd Block', 'Koramangala',
       'Hosur Road', 'Rajajinagar', 'Banaswadi', 'North Bangalore',
       'Nagawara', 'Hennur', 'Kalyan Nagar', 'New BEL Road', 'Jakkur',
       'Rammurthy Nagar', 'Thippasandra', 'Kaggadasapura', 'Hebbal',
       'Kengeri', 'Sankey Road', 'Sadashiv Nagar', 'Basaveshwara Nagar',
       'Yeshwantpur', 'West Bangalore', 'Magadi Road', 'Yelahanka',
       'Sahakara Nagar', 'Peenya'], dtype=object)
```

In [41]:

```
df['listed_in(city)'].unique()
```

Out[41]:
```
array(['Banashankari', 'Bannerghatta Road', 'Basavanagudi', 'Bellandur',
       'Brigade Road', 'Brookefield', 'BTM', 'Church Street',
       'Electronic City', 'Frazer Town', 'HSR', 'Indiranagar',
       'Jayanagar', 'JP Nagar', 'Kalyan Nagar', 'Kammanahalli',
       'Koramangala 4th Block', 'Koramangala 5th Block',
       'Koramangala 6th Block', 'Koramangala 7th Block', 'Lavelle Road',
       'Malleshwaram', 'Marathahalli', 'MG Road', 'New BEL Road',
       'Old Airport Road', 'Rajajinagar', 'Residency Road',
       'Sarjapur Road', 'Whitefield'], dtype=object)
```

listed in(city) and location, both are there, lets keep only one.

In [44]:
```python
df = df.drop(['listed_in(city)'], axis =1)
```

In [48]:
```python
df['Cost2plates'].unique()
```

Out[48]:
```
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', '50', '190', '1,700', '1,400', '180',
       '1,350', '2,200', '2,000', '1,800', '1,900', '330', '2,500',
       '2,100', '3,000', '2,800', '3,400', '40', '1,250', '3,500',
       '4,000', '2,400', '2,600', '120', '1,450', '469', '70', '3,200',
       '60', '560', '240', '360', '6,000', '1,050', '2,300', '4,100',
       '5,000', '3,700', '1,650', '2,700', '4,500', '140'], dtype=object)
```

Removig, from Cost2Plates Column

In [51]:
```python
def handlecomma(value):
    value = str(value)
    if ',' in value:
        value = value.replace(',', '')
        return float(value)
    else:
        return float(value)

df['Cost2plates'] = df['Cost2plates'].apply(handlecomma)
df['Cost2plates'].unique()
```

Out[51]:
```
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.,   50.,  190., 1700., 1400.,  180., 1350., 2200.,
       2000., 1800., 1900.,  330., 2500., 2100., 3000., 2800., 3400.,
         40., 1250., 3500., 4000., 2400., 2600.,  120., 1450.,  469.,
         70., 3200.,   60.,  560.,  240.,  360., 6000., 1050., 2300.,
       4100., 5000., 3700., 1650., 2700., 4500.,  140.])
```

In [53]:
```python
df.head()
```

Out[53]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	Chinese, North Indian, Thai
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian
3	Addhuri Udupi Bhojana	No	No	3.7	88	Banashankari	Quick Bites	South Indian, North Indian
4	Grand Village	No	No	3.8	166	Basavanagudi	Casual Dining	North Indian, Rajasthani

Cleaning Rest Type Column

In [56]:
```
rest_types = df['rest_type'].value_counts(ascending = False)
rest_types
```

Out[56]:
```
rest_type
Quick Bites                   19010
Casual Dining                 10253
Cafe                           3682
Delivery                       2574
Dessert Parlor                 2242
                              ...  
Dessert Parlor, Kiosk             2
Food Court, Beverage Shop         2
Dessert Parlor, Food Court        2
Quick Bites, Kiosk                1
Sweet Shop, Dessert Parlor        1
Name: count, Length: 93, dtype: int64
```

In [58]:
```
rest_types_lessthan1000 = rest_types[rest_types<1000]
rest_types_lessthan1000
```

Out[58]:

```
rest_type
Beverage Shop                  863
Bar                            686
Food Court                     616
Sweet Shop                     468
Bar, Casual Dining             411
                              ...
Dessert Parlor, Kiosk            2
Food Court, Beverage Shop        2
Dessert Parlor, Food Court       2
Quick Bites, Kiosk               1
Sweet Shop, Dessert Parlor       1
Name: count, Length: 85, dtype: int64
```

Making Rest Types less than 1000 in frequency as others

In [61]:

```
def handle_rest_type(value):
    if(value in rest_types_lessthan1000):
        return 'others'
    else:
        return value

df['rest_type'] = df['rest_type'].apply(handle_rest_type)
df['rest_type'].value_counts()
```

Out[61]:

```
rest_type
Quick Bites           19010
Casual Dining         10253
others                 9003
Cafe                   3682
Delivery               2574
Dessert Parlor         2242
Takeaway, Delivery     2008
Bakery                 1140
Casual Dining, Bar     1130
Name: count, dtype: int64
```

Cleaning Location Column

In [68]:

```
location = df['location'].value_counts(ascending = False)

location_lessthan300 = location[location<300]

def handle_location(value):
    if(value in location_lessthan300):
        return 'others'
    else:
        return value

df['location'] = df['location'].apply(handle_location)
df['location'].value_counts()
```

Out[68]:

```
location
BTM                      5056
others                   4954
HSR                      2494
Koramangala 5th Block    2479
JP Nagar                 2218
Whitefield               2105
Indiranagar              2026
Jayanagar                1916
Marathahalli             1805
Bannerghatta Road        1609
Bellandur                1268
Electronic City          1246
Koramangala 1st Block    1236
Brigade Road             1210
Koramangala 7th Block    1174
Koramangala 6th Block    1127
Sarjapur Road            1047
Koramangala 4th Block    1017
Ulsoor                   1011
Banashankari              902
MG Road                   893
Kalyan Nagar              841
Richmond Road             803
Malleshwaram              721
Frazer Town               714
Basavanagudi              684
Residency Road            671
Brookefield               656
New BEL Road              644
Banaswadi                 640
Kammanahalli              639
Rajajinagar               591
Church Street             566
Lavelle Road              518
Shanti Nagar              508
Shivajinagar              498
Cunningham Road           490
Domlur                    482
Old Airport Road          437
Ejipura                   433
Commercial Street         370
St. Marks Road            343
Name: count, dtype: int64
```

Cleaning Cuisines Column

In [71]:

```python
cuisines = df['cuisines'].value_counts(ascending = False)

cuisines_lessthan100 = cuisines[cuisines<100]

def handle_cuisines(value):
    if(value in cuisines_lessthan100):
        return 'others'
    else:
        return value
```

```
df['cuisines'] = df['cuisines'].apply(handle_cuisines)
df['cuisines'].value_counts()
```

Out[71]:

```
cuisines
others                             26159
North Indian                        2852
North Indian, Chinese               2351
South Indian                        1820
Biryani                              903
                                   ...
South Indian, Chinese, North Indian  105
North Indian, Mughlai, Chinese       104
South Indian, Fast Food              104
Italian, Pizza                       102
North Indian, Chinese, Seafood       102
Name: count, Length: 70, dtype: int64
```

In [73]:

```
df.head()
```

Out[73]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	C
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	others	
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	others	others	
3	Addhuri Udupi Bhojana	No	No	3.7	88	Banashankari	Quick Bites	South Indian, North Indian	
4	Grand Village	No	No	3.8	166	Basavanagudi	Casual Dining	others	

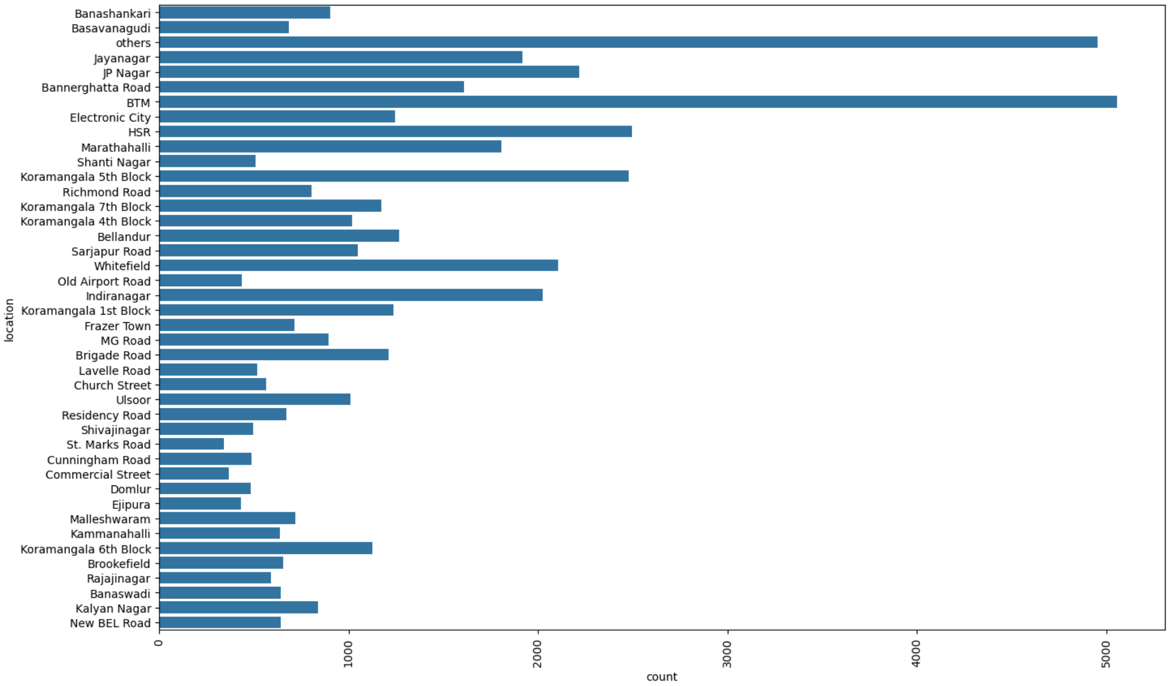
Data is Clean, Lets jump to Visualization

Count Plot of Various Locations

In [77]:

```
plt.figure(figsize = (16,10))
ax = sns.countplot(df['location'])
plt.xticks(rotation=90)
```

```
Out[77]: (array([   0., 1000., 2000., 3000., 4000., 5000., 6000.]),
          [Text(0.0, 0, '0'),
           Text(1000.0, 0, '1000'),
           Text(2000.0, 0, '2000'),
           Text(3000.0, 0, '3000'),
           Text(4000.0, 0, '4000'),
           Text(5000.0, 0, '5000'),
           Text(6000.0, 0, '6000')])
```



Visualizing Online Order

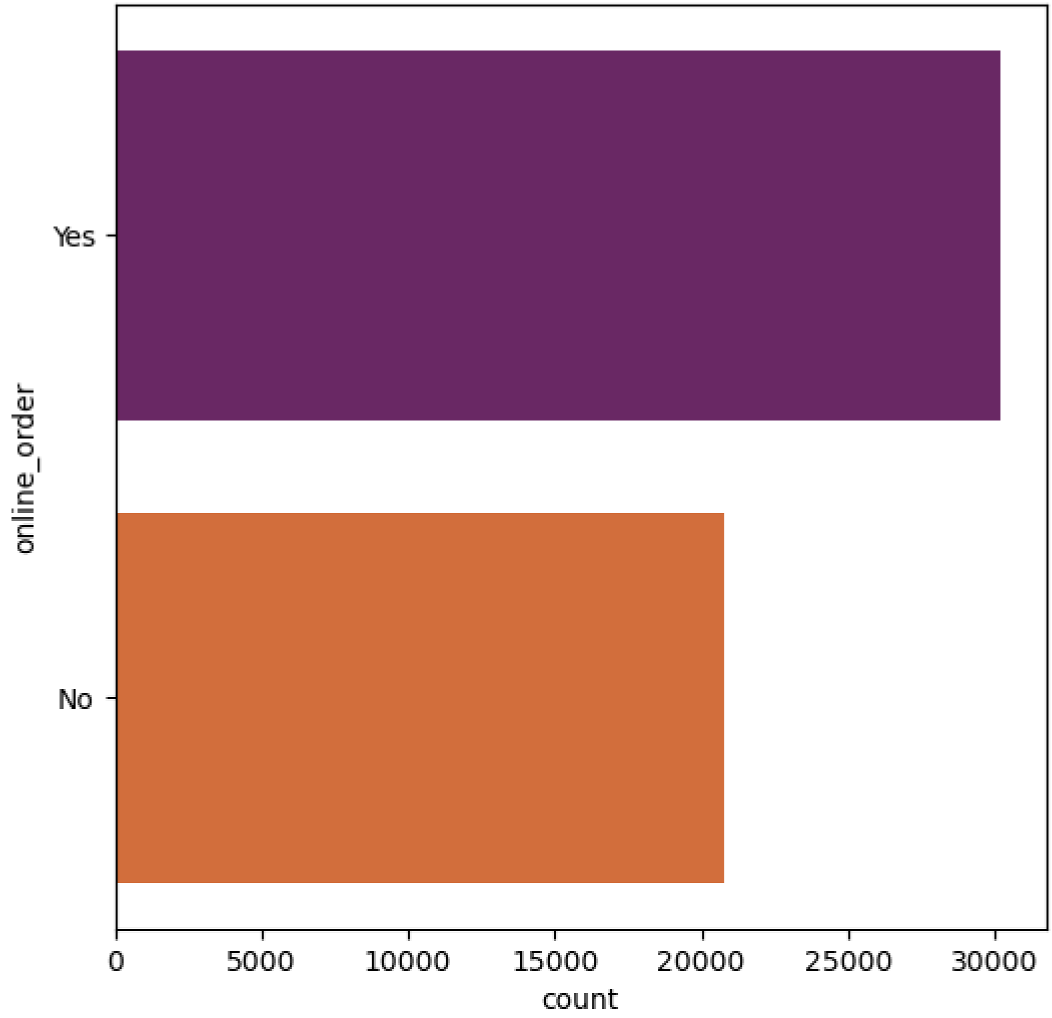
```
In [80]: plt.figure(figsize = (6,6))
         sns.countplot(df['online_order'], palette = 'inferno')
```

```
C:\Users\chitt\AppData\Local\Temp\ipykernel_2808\2251546332.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.countplot(df['online_order'], palette = 'inferno')
```

```
Out[80]: <Axes: xlabel='count', ylabel='online_order'>
```



Visualizing Book Table

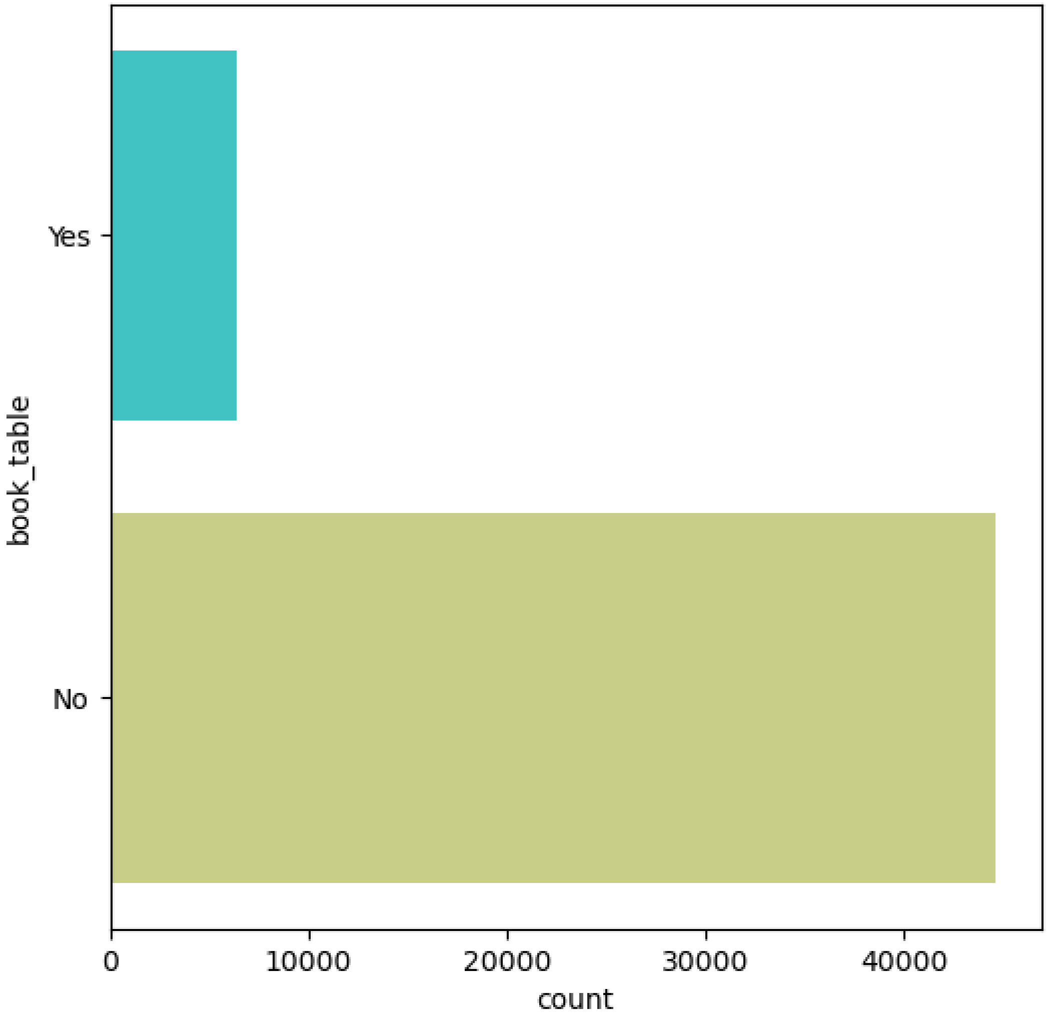
```
In [85]: plt.figure(figsize = (6,6))
         sns.countplot(df['book_table'], palette = 'rainbow')
```

```
C:\Users\chitt\AppData\Local\Temp\ipykernel_2808\222089420.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.countplot(df['book_table'], palette = 'rainbow')
```

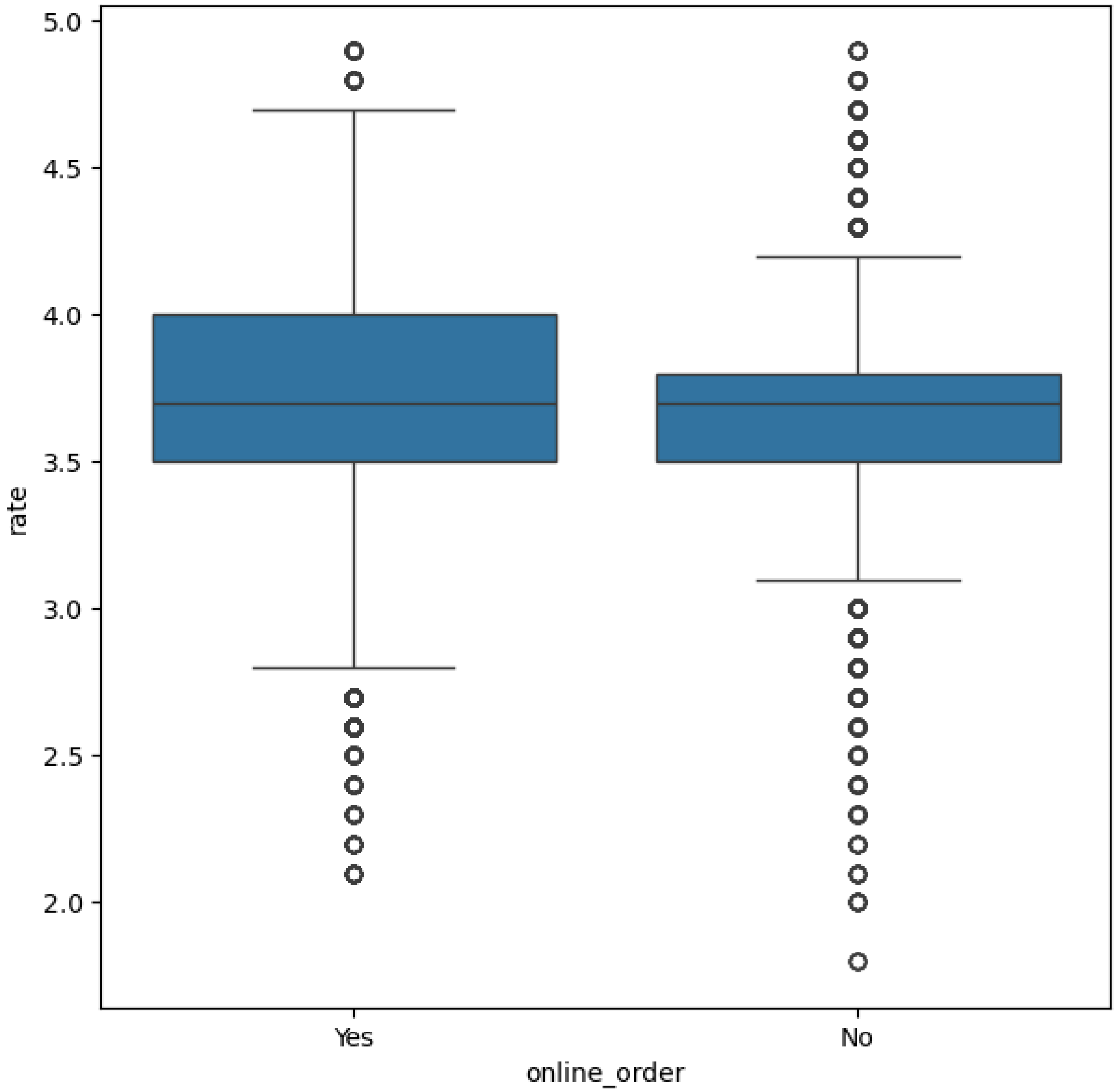
```
Out[85]: <Axes: xlabel='count', ylabel='book_table'>
```



Visualizing Online Order v/s Rate

In [88]:
```
plt.figure(figsize = (7,7))
sns.boxplot(x= 'online_order', y = 'rate', data = df)
```

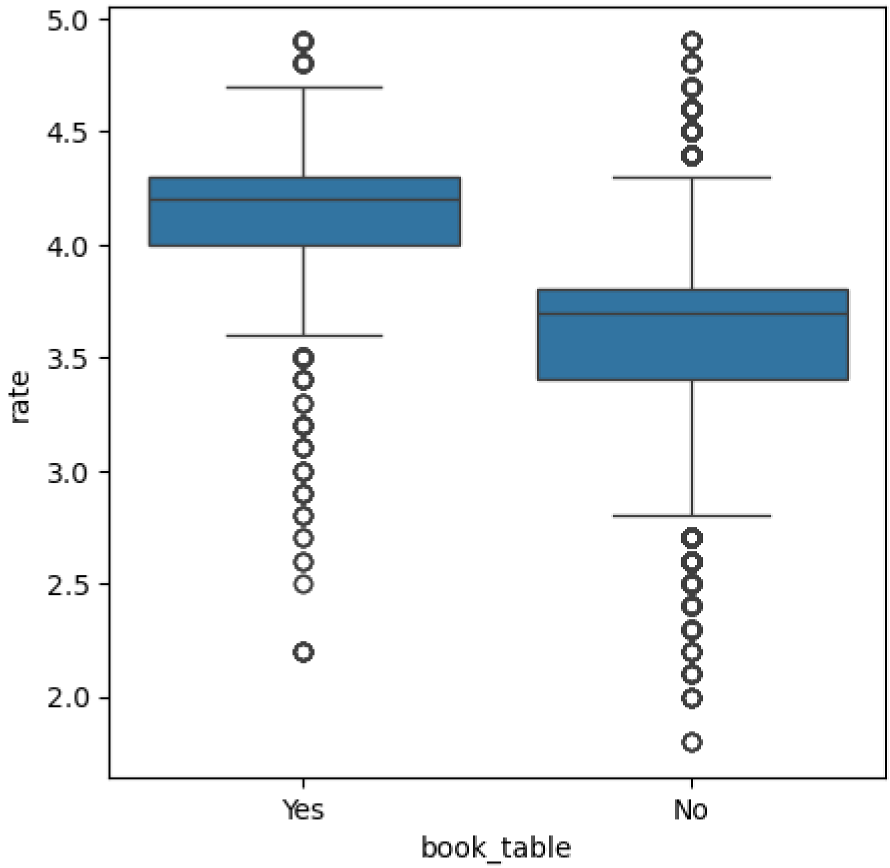
Out[88]:
```
<Axes: xlabel='online_order', ylabel='rate'>
```



Visualizing Book Table v/s Rate

```
In [91]: plt.figure(figsize=(5,5))
         sns.boxplot(x='book_table', y= 'rate', data = df)
```

```
Out[91]: <Axes: xlabel='book_table', ylabel='rate'>
```



Visualizing Online Order Facility, Location Wise

In [94]:

```
df1 = df.groupby(['location','online_order'])['name'].count()
df1.to_csv('location_online.csv')
df1 = pd.read_csv('location_online.csv')
df1 = pd.pivot_table(df1, values=None, index=['location'], columns=['online_orde
df1
```

```
C:\Users\chitt\AppData\Local\Temp\ipykernel_2808\2546502282.py:4: FutureWarning:
The provided callable <function sum at 0x00000164D55E9BC0> is currently using Dat
aFrameGroupBy.sum. In a future version of pandas, the provided callable will be u
sed directly. To keep current behavior pass the string "sum" instead.
  df1 = pd.pivot_table(df1, values=None, index=['location'], columns=['online_ord
er'], fill_value=0, aggfunc=np.sum)
```

Out[94]:

	name	
online_order	**No**	**Yes**
location		
BTM	1763	3293
Banashankari	397	505
Banaswadi	302	338
Bannerghatta Road	685	924
Basavanagudi	243	441
Bellandur	517	751
Brigade Road	552	658
Brookefield	239	417
Church Street	226	340
Commercial Street	228	142
Cunningham Road	168	322
Domlur	247	235
Ejipura	214	219
Electronic City	676	570
Frazer Town	287	427
HSR	584	1910
Indiranagar	697	1329
JP Nagar	911	1307
Jayanagar	552	1364
Kalyan Nagar	350	491
Kammanahalli	264	375
Koramangala 1st Block	384	852
Koramangala 4th Block	459	558
Koramangala 5th Block	866	1613
Koramangala 6th Block	445	682
Koramangala 7th Block	389	785
Lavelle Road	315	203
MG Road	520	373
Malleshwaram	309	412
Marathahalli	701	1104
New BEL Road	255	389

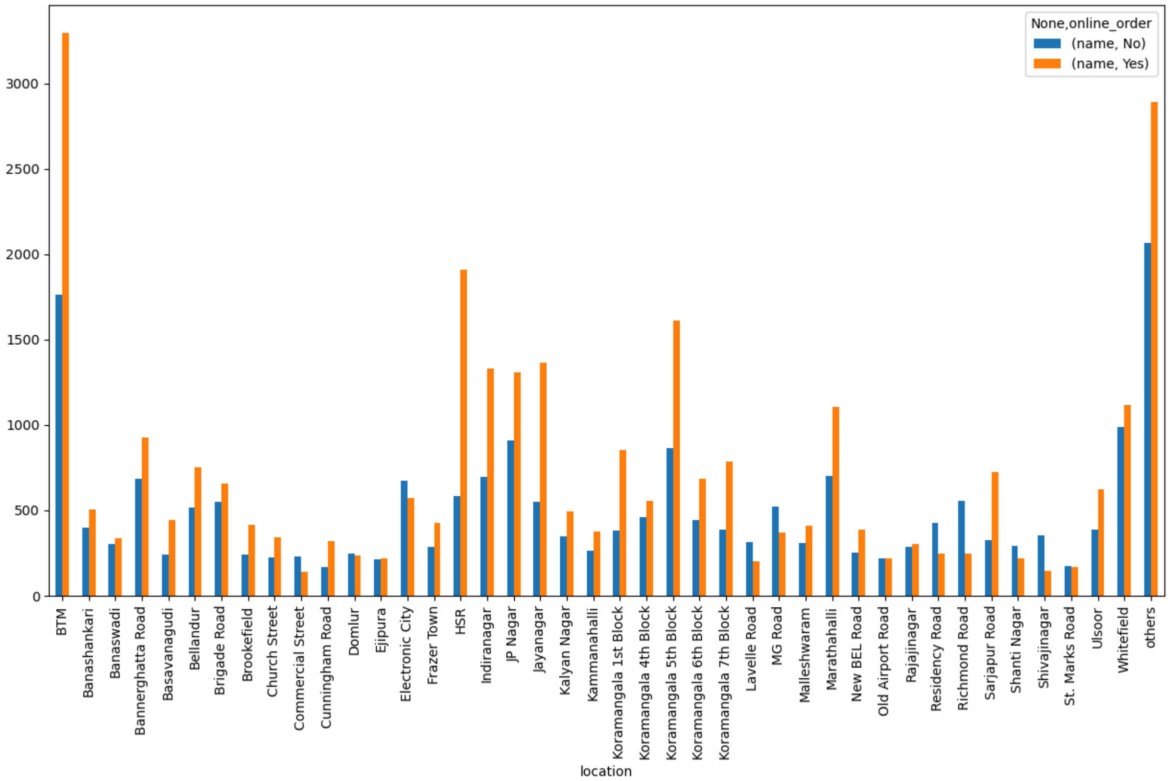
	name	
online_order	**No**	**Yes**
location		
Old Airport Road	221	216
Rajajinagar	286	305
Residency Road	424	247
Richmond Road	557	246
Sarjapur Road	323	724
Shanti Nagar	289	219
Shivajinagar	354	144
St. Marks Road	176	167
Ulsoor	389	622
Whitefield	986	1119
others	2064	2890

In [96]:

```python
df1.plot(kind = 'bar', figsize = (15,8))
```

Out[96]:

```
<Axes: xlabel='location'>
```



Visualizing Book Table Facility, Location Wise

In [99]:

```python
df2 = df.groupby(['location','book_table'])['name'].count()
df2.to_csv('location_booktable.csv')
df2 = pd.read_csv('location_booktable.csv')
```

```
df2 = pd.pivot_table(df2, values=None, index=['location'], columns=['book_table'
df2
```

```
C:\Users\chitt\AppData\Local\Temp\ipykernel_2808\814380054.py:4: FutureWarning: T
he provided callable <function sum at 0x00000164D55E9BC0> is currently using Data
FrameGroupBy.sum. In a future version of pandas, the provided callable will be us
ed directly. To keep current behavior pass the string "sum" instead.
  df2 = pd.pivot_table(df2, values=None, index=['location'], columns=['book_tabl
e'], fill_value=0, aggfunc=np.sum)
```

Out[99]:

	name	
book_table	**No**	**Yes**
location		
BTM	4889	167
Banashankari	839	63
Banaswadi	632	8
Bannerghatta Road	1510	99
Basavanagudi	668	16
Bellandur	1170	98
Brigade Road	1034	176
Brookefield	582	74
Church Street	385	181
Commercial Street	370	0
Cunningham Road	315	175
Domlur	427	55
Ejipura	433	0
Electronic City	1148	98
Frazer Town	706	8
HSR	2277	217
Indiranagar	1578	448
JP Nagar	1903	315
Jayanagar	1637	279
Kalyan Nagar	692	149
Kammanahalli	590	49
Koramangala 1st Block	1186	50
Koramangala 4th Block	695	322
Koramangala 5th Block	1787	692
Koramangala 6th Block	1015	112
Koramangala 7th Block	1012	162
Lavelle Road	290	228
MG Road	546	347
Malleshwaram	632	89
Marathahalli	1642	163
New BEL Road	588	56

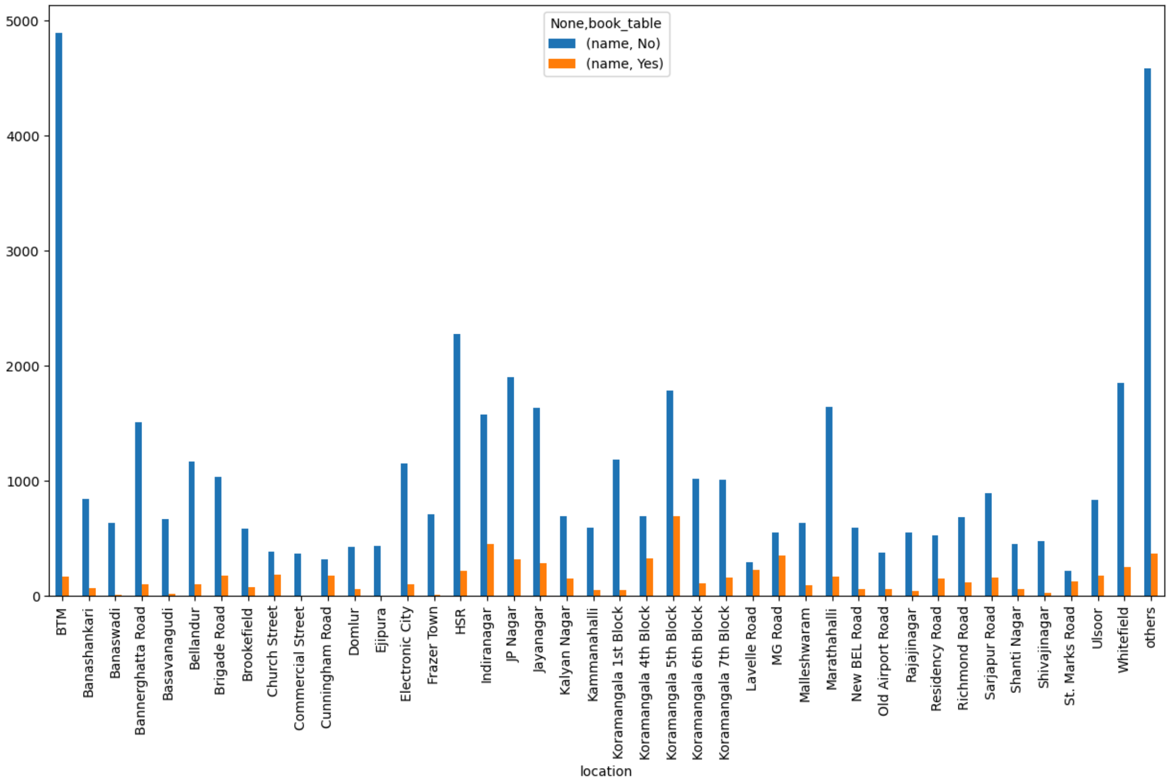
	name	
book_table	**No**	**Yes**
location		
Old Airport Road	378	59
Rajajinagar	550	41
Residency Road	522	149
Richmond Road	687	116
Sarjapur Road	893	154
Shanti Nagar	451	57
Shivajinagar	475	23
St. Marks Road	219	124
Ulsoor	834	177
Whitefield	1852	253
others	4587	367

In [101…

```
df2.plot(kind = 'bar', figsize = (15,8))
```

Out[101…

```
<Axes: xlabel='location'>
```



Visualizing Types of Resturant vs Rate

In [104…

```
plt.figure(figsize= (14,8))
sns.boxplot(x = 'Type', y = 'rate', data= df, palette = 'inferno')
```

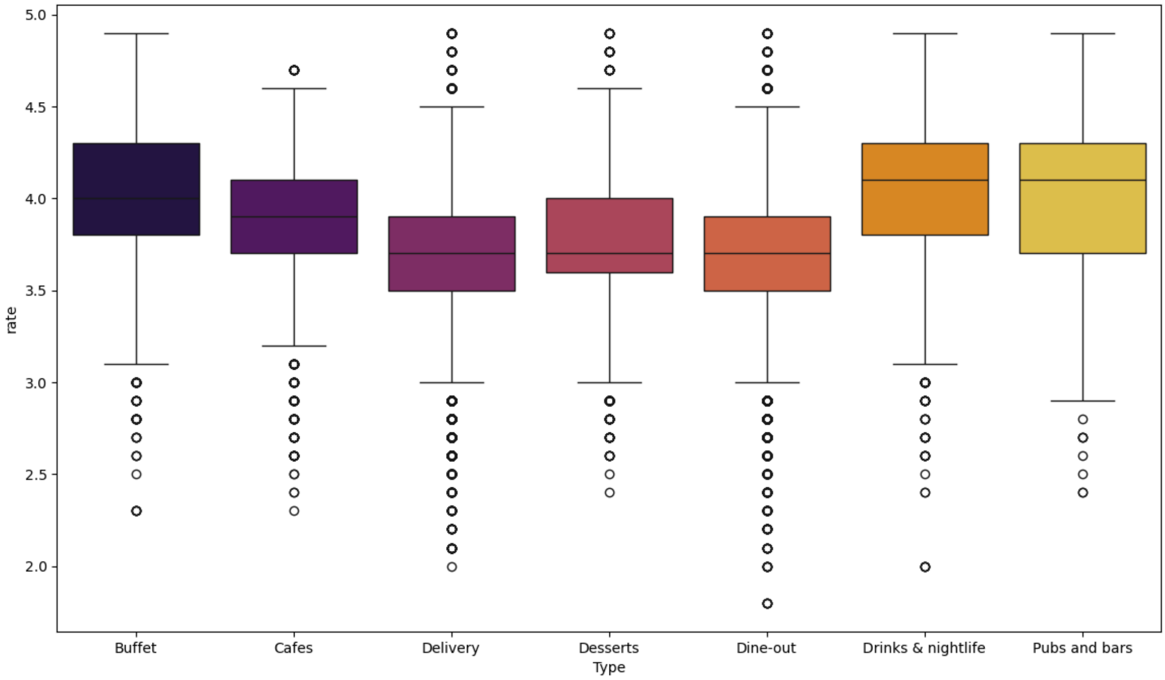
```
C:\Users\chitt\AppData\Local\Temp\ipykernel_2808\4050412383.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v
0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effe
ct.

  sns.boxplot(x = 'Type', y = 'rate', data= df, palette = 'inferno')
```

Out[104…

```
<Axes: xlabel='Type', ylabel='rate'>
```



Grouping Types of Resturents, Location wise

In [107…

```
df3 = df.groupby(['location','Type'])['name'].count()
df3.to_csv('location_Type.csv')
df3 = pd.read_csv('location_Type.csv')
df3 = pd.pivot_table(df3, values=None, index=['location'], columns=['Type'], fil
df3
```

```
C:\Users\chitt\AppData\Local\Temp\ipykernel_2808\1140243432.py:4: FutureWarning:
The provided callable <function sum at 0x00000164D55E9BC0> is currently using Dat
aFrameGroupBy.sum. In a future version of pandas, the provided callable will be u
sed directly. To keep current behavior pass the string "sum" instead.
  df3 = pd.pivot_table(df3, values=None, index=['location'], columns=['Type'], fi
ll_value=0, aggfunc=np.sum)
```

Out[107…

							name
Type	**Buffet**	**Cafes**	**Delivery**	**Desserts**	**Dine-out**	**Drinks & nightlife**	**Pubs and bars**
location							
BTM	21	83	3053	198	1660	22	19
Banashankari	7	36	418	71	356	14	0
Banaswadi	0	24	310	37	262	6	1
Bannerghatta Road	9	46	828	137	578	9	2
Basavanagudi	7	11	344	66	251	5	0
Bellandur	28	36	617	75	479	17	16
Brigade Road	25	46	497	108	455	57	22
Brookefield	6	17	339	45	245	4	0
Church Street	19	51	193	29	215	36	23
Commercial Street	0	13	121	77	159	0	0
Cunningham Road	29	34	194	26	184	16	7
Domlur	15	13	261	35	135	12	11
Ejipura	0	0	245	16	172	0	0
Electronic City	23	24	570	71	516	21	21
Frazer Town	1	11	470	56	172	2	2
HSR	19	49	1694	120	580	14	18
Indiranagar	38	97	1091	140	529	65	66
JP Nagar	45	76	1151	166	722	51	7
Jayanagar	27	77	1043	182	575	12	0
Kalyan Nagar	9	45	366	88	315	18	0
Kammanahalli	2	27	329	35	240	6	0
Koramangala 1st Block	3	26	716	70	398	7	16
Koramangala 4th Block	21	53	464	81	302	62	34
Koramangala 5th Block	65	146	1075	209	842	84	58
Koramangala 6th Block	18	43	511	70	411	51	23

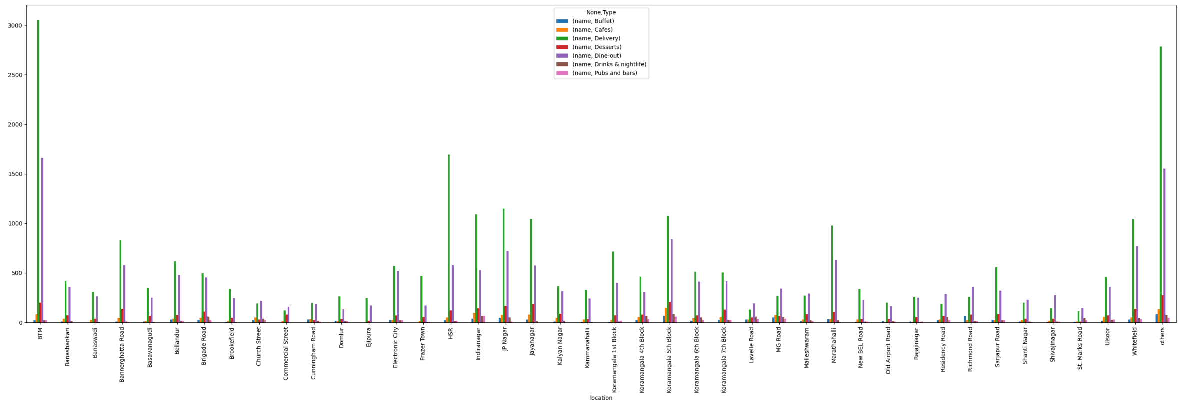
	name						
Type	Buffet	Cafes	Delivery	Desserts	Dine-out	Drinks & nightlife	Pubs and bars
location							
Koramangala 7th Block	25	52	503	127	417	25	25
Lavelle Road	30	27	127	50	191	59	34
MG Road	51	76	266	68	343	53	36
Malleshwaram	11	31	269	85	291	20	14
Marathahalli	34	32	980	105	630	22	2
New BEL Road	4	29	338	33	224	8	8
Old Airport Road	12	5	200	35	164	12	9
Rajajinagar	10	4	258	55	251	3	10
Residency Road	20	31	187	63	289	55	26
Richmond Road	63	21	257	78	356	16	12
Sarjapur Road	25	22	558	82	319	19	22
Shanti Nagar	9	22	198	39	229	9	2
Shivajinagar	6	17	143	37	280	7	8
St. Marks Road	5	10	111	10	145	40	22
Ulsoor	16	56	456	71	359	23	30
Whitefield	28	51	1041	137	768	47	33
others	83	133	2787	276	1553	75	47

In [111…

```
df3.plot(kind = 'bar', figsize = (36,10))
```

Out[111…

```
<Axes: xlabel='location'>
```



No. of Votes, Location Wise

In [114…

```
df4 = df[['location', 'votes']]
df4.drop_duplicates()
df5 = df4.groupby(['location'])['votes'].sum()
df5 = df5.to_frame()
df5 = df5.sort_values('votes', ascending=False)
df5.head()
```

Out[114…

	votes
location	
Koramangala 5th Block	2214083
Indiranagar	1165909
Koramangala 4th Block	685156
Church Street	590306
JP Nagar	586522

In [116…

```
df.head()
```

Out[116…

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	C
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	others	
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	others	others	
3	Addhuri Udupi Bhojana	No	No	3.7	88	Banashankari	Quick Bites	South Indian, North Indian	
4	Grand Village	No	No	3.8	166	Basavanagudi	Casual Dining	others	

Visualizing Top Cuisines

In [119…

```
df6 = df[['cuisines', 'votes']]
df6.drop_duplicates()
df7 = df6.groupby(['cuisines'])['votes'].sum()
df7 = df7.to_frame()
df7 = df7.sort_values('votes', ascending=False)
df7.head()
```

Out[119…

	votes
cuisines	
others	11542182
North Indian	516310
North Indian, Chinese	258225
South Indian	161975
North Indian, Mughlai	103706

In [121…

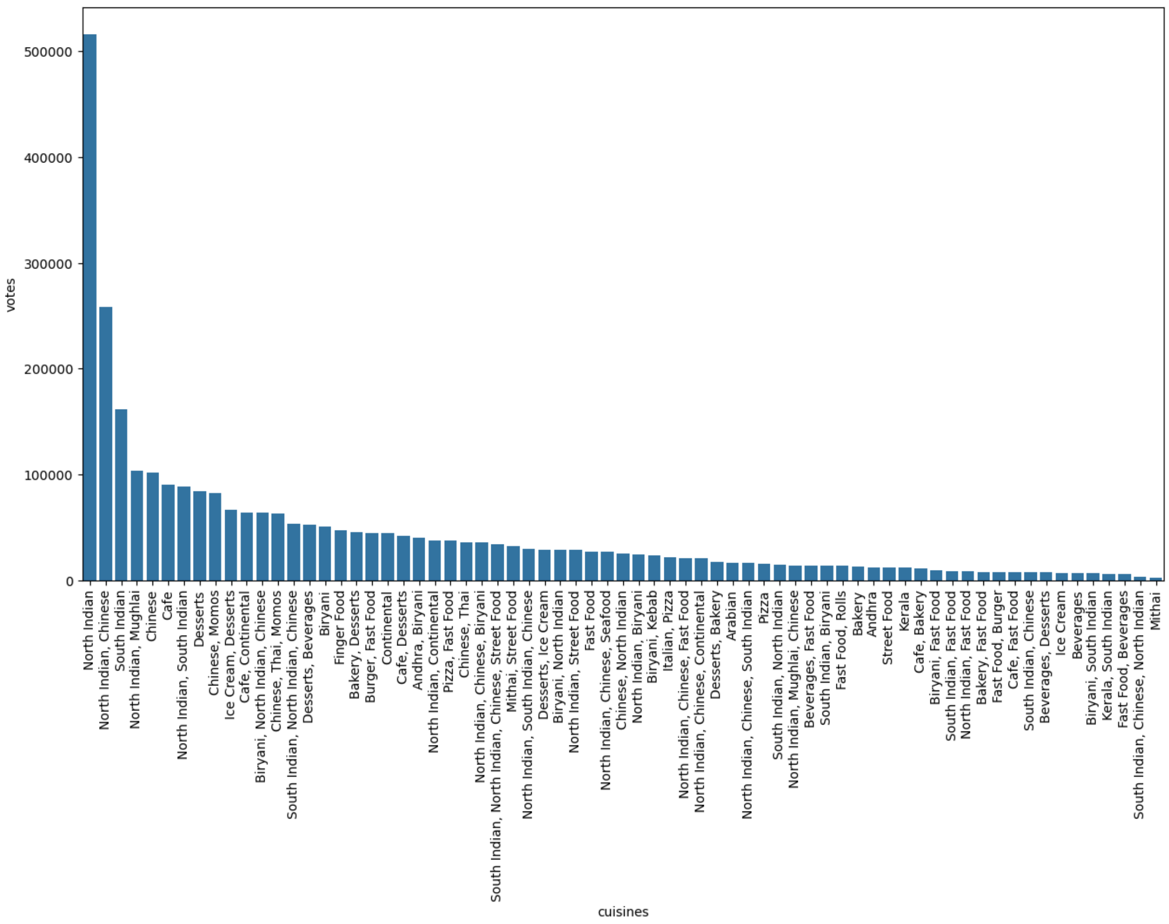
```
df7 = df7.iloc[1:, :]
df7.head()
```

Out[121…

	votes
cuisines	
North Indian	516310
North Indian, Chinese	258225
South Indian	161975
North Indian, Mughlai	103706
Chinese	101728

In [125…

```
plt.figure(figsize=(15, 8))
sns.barplot(x=df7.index, y=df7['votes'])
plt.xticks(rotation=90)
plt.show()
```



Completed

```
In [ ]:
```