In [1]:
```
##Importing Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
##reading data

df=pd.read_csv("zomato.csv")
df.head()
```

Out[2]:

	url	address	name	online_order	book_table	rate	votes	phone	location	rest_type
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1/5	775	080 42297555\r\n+91 9743772233	Banashankari	Casual Dining
1	https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1/5	787	080 41714161	Banashankari	Casual Dining
2	https://www.zomato.com/SanchurroBangalore?cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8/5	918	+91 9663487993	Banashankari	Cafe, Casual Dining
3	https://www.zomato.com/bangalore/addhuri-udupi...	1st Floor, Annakuteera, 3rd Stage, Banashankar...	Addhuri Udupi Bhojana	No	No	3.7/5	88	+91 9620009302	Banashankari	Quick Bites

	url	address	name	online_order	book_table	rate	votes	phone	location	rest_type
4	https://www.zomato.com/bangalore/grand-village...	10, 3rd Floor, Lakshmi Associates, Gandhi Baza...	Grand Village	No	No	3.8/5	166	+91 8026612447\r\n+91 9901210005	Basavanagudi	Casual Dining

In [3]:
```
## analysing data
```

In [4]:
```
df.shape
```

Out[4]:
```
(51717, 17)
```

In [5]:
```
df.columns
```

Out[5]:
```
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 [6]:
```
##remove unnecessary columns
df=df.drop(['url', 'address','phone','dish_liked','reviews_list', 'menu_item'],axis=1)
df.head()
```

Out[6]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	approx_cost(for two people)	listed_in(type)	listed_in(city)
0	Jalsa	Yes	Yes	4.1/5	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800	Buffet	Banashankari
1	Spice Elephant	Yes	No	4.1/5	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	800	Buffet	Banashankari
2	San Churro Cafe	Yes	No	3.8/5	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian	800	Buffet	Banashankari

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	approx_cost(for two people)	listed_in(type)	listed_in(city)
3	Addhuri Udupi Bhojana	No	No	3.7/5	88	Banashankari	Quick Bites	South Indian, North Indian	300	Buffet	Banashankari
4	Grand Village	No	No	3.8/5	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	600	Buffet	Banashankari

In [7]:
```
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
```

In [8]:
```
## remove duplicates
df.drop_duplicates(inplace=True)
df.shape
```

Out[8]:
```
(51609, 11)
```

In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 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                         43854 non-null  object
 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: int64(1), object(10)
memory usage: 4.7+ MB
```

In [10]:
```
df['rate'].unique()
```

Out[10]:
```
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 [11]:
```
def change(value):

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

In [12]:
```
df['rate'] = df['rate'].apply(change)
df['rate'].head()
```

Out[12]:
```
0    4.1
1    4.1
2    3.8
3    3.7
4    3.8
Name: rate, dtype: float64
```

In [13]:
```
##filling null values
df['rate'].fillna(df['rate'].mean(),inplace=True)
```

In [14]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 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
```

In [15]:
```
## others columns don't have so many null values so i have to remove all the rows which contains the null values
df.dropna(inplace=True)
df.head()
```

Out[15]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	approx_cost(for two people)	listed_in(type)	listed_in(city)
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800	Buffet	Banashankari

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	approx_cost(for two people)	listed_in(type)	listed_in(city)
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	800	Buffet	Banashankari
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian	800	Buffet	Banashankari
3	Addhuri Udupi Bhojana	No	No	3.7	88	Banashankari	Quick Bites	South Indian, North Indian	300	Buffet	Banashankari
4	Grand Village	No	No	3.8	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	600	Buffet	Banashankari

In [16]:
```
## rename the column
df.rename(columns = {'approx_cost(for two people)':'Cost2plates','listed_in(type)':'Type'},inplace=True)
df.head()
```

Out[16]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	Cost2plates	Type	listed_in(city)
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800	Buffet	Banashankari
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	800	Buffet	Banashankari
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian	800	Buffet	Banashankari
3	Addhuri Udupi Bhojana	No	No	3.7	88	Banashankari	Quick Bites	South Indian, North Indian	300	Buffet	Banashankari
4	Grand Village	No	No	3.8	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	600	Buffet	Banashankari

In [17]:
```
## location and listed_in(city) both are same values, so i have to remove one column

df= df.drop(['listed_in(city)'],axis=1)
df
```

Out[17]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	Cost2plates	Type

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	Cost2plates	Type
0	Jalsa	Yes	Yes	4.100000	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800	Buffet
1	Spice Elephant	Yes	No	4.100000	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	800	Buffet
2	San Churro Cafe	Yes	No	3.800000	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian	800	Buffet
3	Addhuri Udupi Bhojana	No	No	3.700000	88	Banashankari	Quick Bites	South Indian, North Indian	300	Buffet
4	Grand Village	No	No	3.800000	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	600	Buffet
...	...	...	...	...	...	...	...	...	...	...
51712	Best Brews - Four Points by Sheraton Bengaluru...	No	No	3.600000	27	Whitefield	Bar	Continental	1,500	Pubs and bars
51713	Vinod Bar And Restaurant	No	No	3.700142	0	Whitefield	Bar	Finger Food	600	Pubs and bars
51714	Plunge - Sheraton Grand Bengaluru Whitefield H...	No	No	3.700142	0	Whitefield	Bar	Finger Food	2,000	Pubs and bars
51715	Chime - Sheraton Grand Bengaluru Whitefield Ho...	No	Yes	4.300000	236	ITPL Main Road, Whitefield	Bar	Finger Food	2,500	Pubs and bars
51716	The Nest - The Den Bengaluru	No	No	3.400000	13	ITPL Main Road, Whitefield	Bar, Casual Dining	Finger Food, North Indian, Continental	1,500	Pubs and bars

51042 rows × 10 columns

In [18]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 51042 entries, 0 to 51716
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   name          51042 non-null  object 
 1   online_order  51042 non-null  object 
 2   book_table    51042 non-null  object 
 3   rate          51042 non-null  float64
 4   votes         51042 non-null  int64  
 5   location      51042 non-null  object 
 6   rest_type     51042 non-null  object 
 7   cuisines      51042 non-null  object 
 8   Cost2plates   51042 non-null  object 
 9   Type          51042 non-null  object 
dtypes: float64(1), int64(1), object(8)
memory usage: 4.3+ MB
```

In [19]:
```
df['Cost2plates'].unique()
```

Out[19]:
```
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)
```

In [20]:
```
df['Cost2plates']=df['Cost2plates'].str.replace(',','')
```

In [21]:
```
df['Cost2plates'].unique()
```

Out[21]:
```
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'], dtype=object)
```

In [22]:
```
df['Cost2plates']=df['Cost2plates'].astype(np.int64)
```

In [23]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 51042 entries, 0 to 51716
Data columns (total 10 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   name          51042 non-null  object 
 1   online_order  51042 non-null  object 
 2   book_table    51042 non-null  object 
 3   rate          51042 non-null  float64
 4   votes         51042 non-null  int64  
 5   location      51042 non-null  object 
 6   rest_type     51042 non-null  object 
 7   cuisines      51042 non-null  object 
 8   Cost2plates   51042 non-null  int64  
 9   Type          51042 non-null  object 
dtypes: float64(1), int64(2), object(7)
memory usage: 4.3+ MB
```

In [24]:
```
df
```

Out[24]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	Cost2plates	Type
0	Jalsa	Yes	Yes	4.100000	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800	Buffet
1	Spice Elephant	Yes	No	4.100000	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	800	Buffet
2	San Churro Cafe	Yes	No	3.800000	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian	800	Buffet
3	Addhuri Udupi Bhojana	No	No	3.700000	88	Banashankari	Quick Bites	South Indian, North Indian	300	Buffet

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	Cost2plates	Type
4	Grand Village	No	No	3.800000	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	600	Buffet
...	...	...	...	...	...	...	...	...	...	...
51712	Best Brews - Four Points by Sheraton Bengaluru...	No	No	3.600000	27	Whitefield	Bar	Continental	1500	Pubs and bars
51713	Vinod Bar And Restaurant	No	No	3.700142	0	Whitefield	Bar	Finger Food	600	Pubs and bars
51714	Plunge - Sheraton Grand Bengaluru Whitefield H...	No	No	3.700142	0	Whitefield	Bar	Finger Food	2000	Pubs and bars
51715	Chime - Sheraton Grand Bengaluru Whitefield Ho...	No	Yes	4.300000	236	ITPL Main Road, Whitefield	Bar	Finger Food	2500	Pubs and bars
51716	The Nest - The Den Bengaluru	No	No	3.400000	13	ITPL Main Road, Whitefield	Bar, Casual Dining	Finger Food, North Indian, Continental	1500	Pubs and bars

51042 rows × 10 columns

In [33]:
```
rest_types=df['rest_type'].value_counts()
rest_types
```

Out[33]:
```
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: rest_type, Length: 93, dtype: int64
```

In [34]:
```
rest_types_lessthan_1000 = rest_types[rest_types<1000]
rest_types_lessthan_1000
```

Out[34]:
```
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: rest_type, Length: 85, dtype: int64
```

In [35]:
```
def handle_rest_values(i):
    if (i in rest_types_lessthan_1000):
        return 'others'
    else:
        return i
```

In [36]:
```
df['rest_type']=df['rest_type'].apply(handle_rest_values)
```

In [37]:
```
df['rest_type'].value_counts()
```

Out[37]:
```
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: rest_type, dtype: int64
```

In [39]:
```
location1 = df['location'].value_counts()
location1
```

Out[39]:
```
BTM                    5056
HSR                    2494
Koramangala 5th Block  2479
JP Nagar               2218
Whitefield             2105
                       ...
West Bangalore            6
Yelahanka                 5
Jakkur                    3
Rajarajeshwari Nagar      2
Peenya                    1
Name: location, Length: 93, dtype: int64
```

In [40]:
```
location_lessthan_300 = location1[location1<300]
location_lessthan_300
```

Out[40]:
```
Koramangala 8th Block           294
Vasanth Nagar                   293
Jeevan Bhima Nagar              268
Wilson Garden                   246
Bommanahalli                    236
Koramangala 3rd Block           215
Thippasandra                    191
Kumaraswamy Layout              191
Nagawara                        187
Basaveshwara Nagar              187
Seshadripuram                   165
Hennur                          159
Majestic                        155
HBR Layout                      153
Infantry Road                   150
Race Course Road                139
City Market                     122
Yeshwantpur                     119
ITPL Main Road, Whitefield      113
Varthur Main Road, Whitefield   109
South Bangalore                 107
Koramangala 2nd Block           102
Kaggadasapura                   101
Hosur Road                       98
```

```
CV Raman Nagar                 89
Vijay Nagar                    78
RT Nagar                       78
Sanjay Nagar                   76
Sadashiv Nagar                 63
Sahakara Nagar                 53
Koramangala                    48
East Bangalore                 43
Jalahalli                      38
Magadi Road                    34
Rammurthy Nagar                32
Langford Town                  27
Sankey Road                    27
Old Madras Road                22
Mysore Road                    22
Kanakapura Road                19
KR Puram                       18
Uttarahalli                    17
Hebbal                         14
North Bangalore                14
Nagarbhavi                      9
Kengeri                         8
Central Bangalore               8
West Bangalore                  6
Yelahanka                       5
Jakkur                          3
Rajarajeshwari Nagar            2
Peenya                          1
Name: location, dtype: int64
```

In [41]:
```
def location_lessthan300(i):
    if (i in location_lessthan_300):
        return 'others'
    else:
        return i
```

In [42]:
```
df['location']=df['location'].apply(location_lessthan300)
```

In [43]:
```
df['location'].value_counts()
```

Out[43]:
```
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: location, dtype: int64
```

In [44]:
```
df.head()
```

Out[44]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	Cost2plates	Type
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800	Buffet
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	800	Buffet
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	others	Cafe, Mexican, Italian	800	Buffet
3	Addhuri Udupi Bhojana	No	No	3.7	88	Banashankari	Quick Bites	South Indian, North Indian	300	Buffet
4	Grand Village	No	No	3.8	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	600	Buffet

In [46]:
```
cuisines1=df['cuisines'].value_counts()
cuisines1
```

Out[46]:
```
North Indian                                2852
North Indian, Chinese                       2351
South Indian                                1820
Biryani                                      903
Bakery, Desserts                             898
                                            ...
North Indian, Chinese, Oriya, Mithai           1
Beverages, Burger                              1
North Indian, Mughlai, Lucknowi                1
Continental, Thai, North Indian, Chinese       1
North Indian, Chinese, Arabian, Momos          1
Name: cuisines, Length: 2704, dtype: int64
```

In [47]:
```
cuisines_lessthan_100=cuisines1[cuisines1<100]
cuisines_lessthan_100
```

Out[47]:
```
North Indian, Continental, Chinese          97
Juices                                      94
Fast Food, North Indian                     93
Bengali, North Indian                       93
Beverages, Juices                           90
                                            ..
North Indian, Chinese, Oriya, Mithai         1
Beverages, Burger                            1
North Indian, Mughlai, Lucknowi              1
```

```
Continental, Thai, North Indian, Chinese     1
North Indian, Chinese, Arabian, Momos        1
Name: cuisines, Length: 2635, dtype: int64
```

In [48]:
```
def cuisines_lessthan100(i):
    if (i in cuisines_lessthan_100):
        return 'others'
    else:
        return i
```

In [49]:
```
df['cuisines']=df['cuisines'].apply(cuisines_lessthan100)
df['cuisines'].value_counts()
```

Out[49]:
```
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: cuisines, Length: 70, dtype: int64
```

In [54]:
```
df['Type'].value_counts()

## Not require cleaning
```

Out[54]:
```
Delivery             25579
Dine-out             17562
Desserts              3559
Cafes                 1703
Drinks & nightlife    1084
Buffet                 869
Pubs and bars          686
Name: Type, dtype: int64
```

In [55]:
```
## i have done all the cleaning part
```

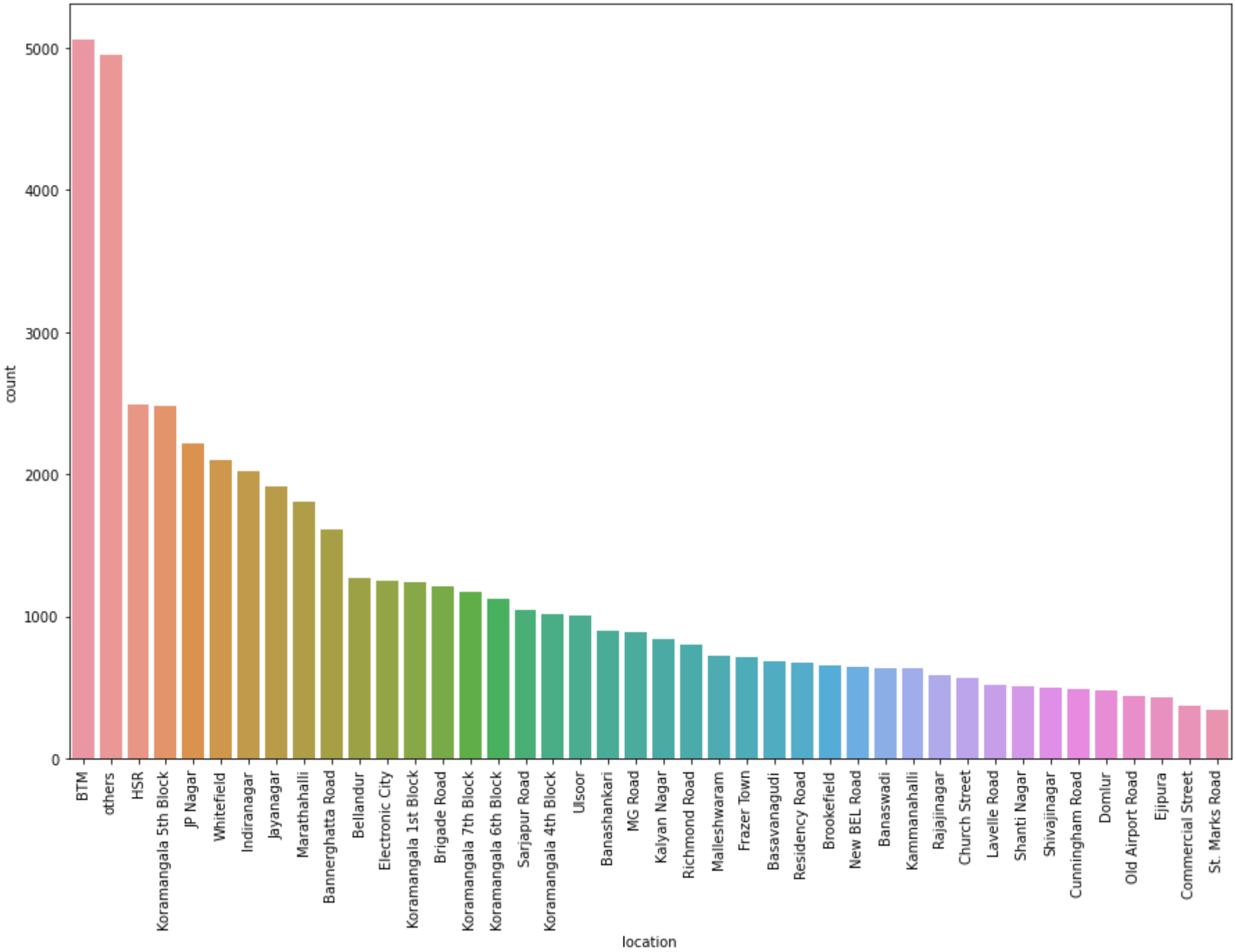
In [56]:
```
## data visualization
```

In [97]:
```
##Count Plot of Various Locations

plt.figure(figsize=(15,10))
sns.countplot(df['location'],order=df['location'].value_counts().index)
plt.xticks(rotation=90)
```

Out[97]:
```
C:\Users\Administrator\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyw
ord arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35, 36, 37, 38, 39, 40, 41]),
 [Text(0, 0, 'BTM'),
  Text(1, 0, 'others'),
  Text(2, 0, 'HSR'),
  Text(3, 0, 'Koramangala 5th Block'),
  Text(4, 0, 'JP Nagar'),
  Text(5, 0, 'Whitefield'),
  Text(6, 0, 'Indiranagar'),
  Text(7, 0, 'Jayanagar'),
  Text(8, 0, 'Marathahalli'),
  Text(9, 0, 'Bannerghatta Road'),
  Text(10, 0, 'Bellandur'),
  Text(11, 0, 'Electronic City'),
  Text(12, 0, 'Koramangala 1st Block'),
  Text(13, 0, 'Brigade Road'),
  Text(14, 0, 'Koramangala 7th Block'),
  Text(15, 0, 'Koramangala 6th Block'),
  Text(16, 0, 'Sarjapur Road'),
  Text(17, 0, 'Koramangala 4th Block'),
  Text(18, 0, 'Ulsoor'),
  Text(19, 0, 'Banashankari'),
  Text(20, 0, 'MG Road'),
  Text(21, 0, 'Kalyan Nagar'),
  Text(22, 0, 'Richmond Road'),
  Text(23, 0, 'Malleshwaram'),
  Text(24, 0, 'Frazer Town'),
```

```
Text(25, 0, 'Basavanagudi'),
Text(26, 0, 'Residency Road'),
Text(27, 0, 'Brookefield'),
Text(28, 0, 'New BEL Road'),
Text(29, 0, 'Banaswadi'),
Text(30, 0, 'Kammanahalli'),
Text(31, 0, 'Rajajinagar'),
Text(32, 0, 'Church Street'),
Text(33, 0, 'Lavelle Road'),
Text(34, 0, 'Shanti Nagar'),
Text(35, 0, 'Shivajinagar'),
Text(36, 0, 'Cunningham Road'),
Text(37, 0, 'Domlur'),
Text(38, 0, 'Old Airport Road'),
Text(39, 0, 'Ejipura'),
Text(40, 0, 'Commercial Street'),
Text(41, 0, 'St. Marks Road')])
```

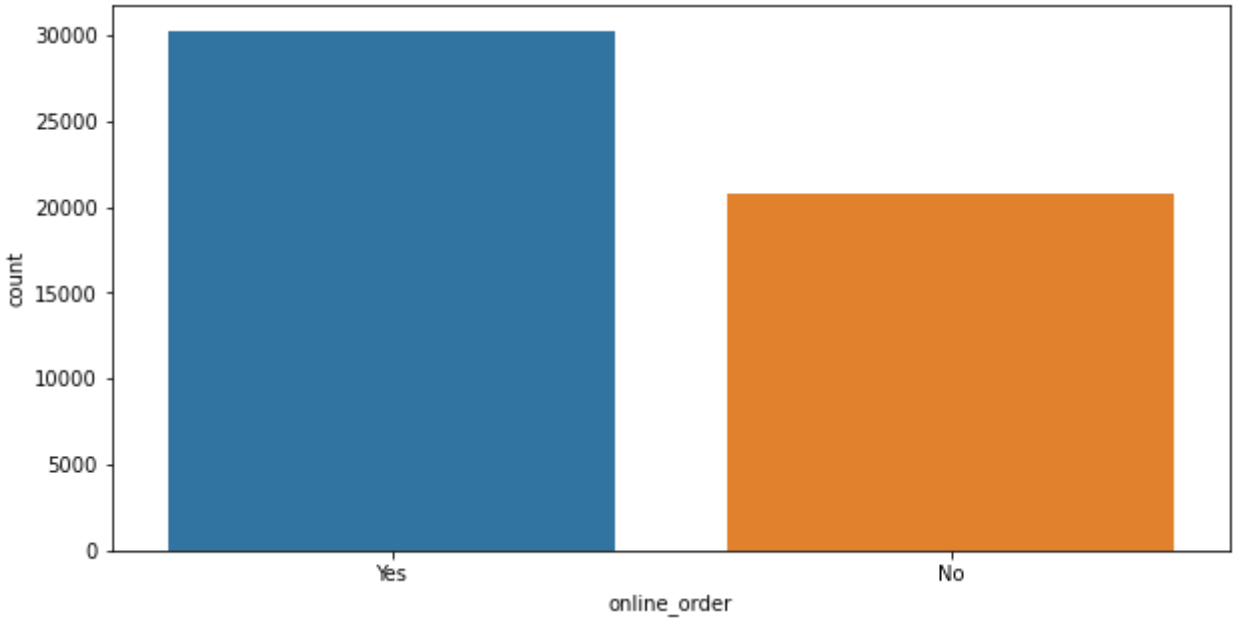

count
location
0
1000
2000
3000
4000
5000
BTM
others
HSR
Koramangala 5th Block
JP Nagar
Whitefield
Indiranagar
Jayanagar
Marathahalli
Bannerghatta Road
Bellandur
Electronic City
Koramangala 1st Block
Brigade Road
Koramangala 7th Block
Koramangala 6th Block
Sarjapur Road
Koramangala 4th Block
Ulsoor
Banashankari
MG Road
Kalyan Nagar
Richmond Road
Malleshwaram
Frazer Town
Basavanagudi
Residency Road
Brookefield
New BEL Road
Banaswadi
Kammanahalli
Rajajinagar
Church Street
Lavelle Road
Shanti Nagar
Shivajinagar
Cunningham Road
Domlur
Old Airport Road
Ejipura
Commercial Street
St. Marks Road

In [101…
```
## how many restaurants have online order and offline order facility?

plt.figure(figsize=(10,5))
sns.countplot(df['online_order'])
```

```
C:\Users\Administrator\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyw
ord arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[101…
```
<AxesSubplot:xlabel='online_order', ylabel='count'>
```

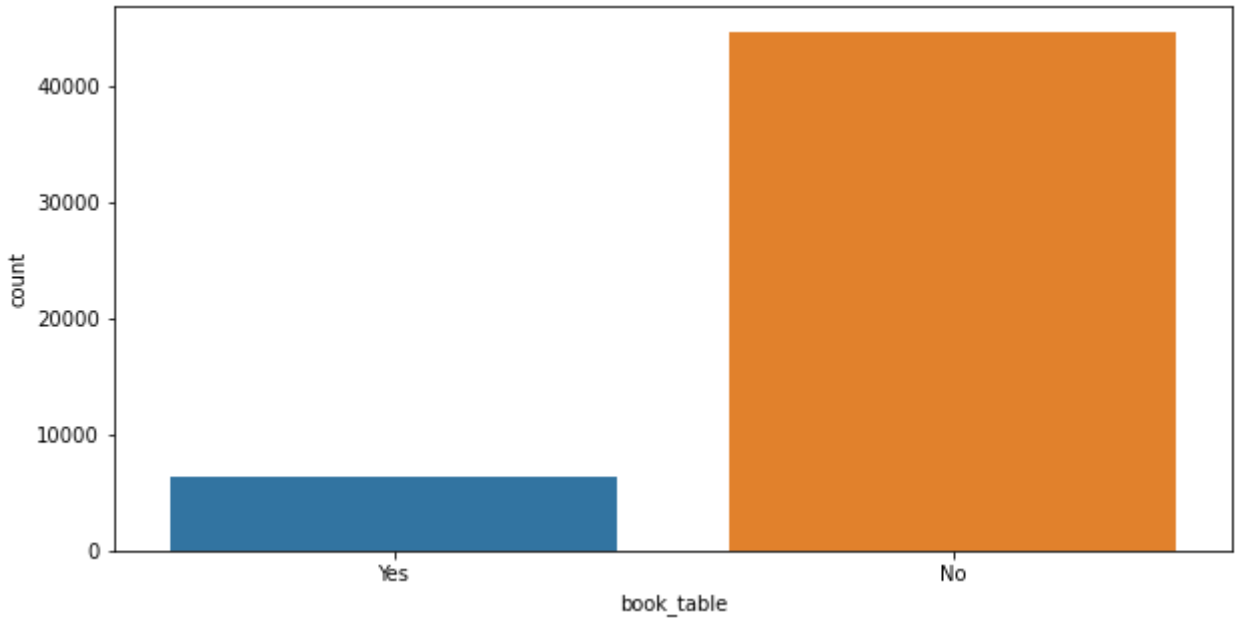


In [102…
```
## how many restaurants have booking table facility?

plt.figure(figsize=(10,5))
sns.countplot(df['book_table'])
```

```
C:\Users\Administrator\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyw
ord arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit
keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[102…
```
<AxesSubplot:xlabel='book_table', ylabel='count'>
```



In [107…

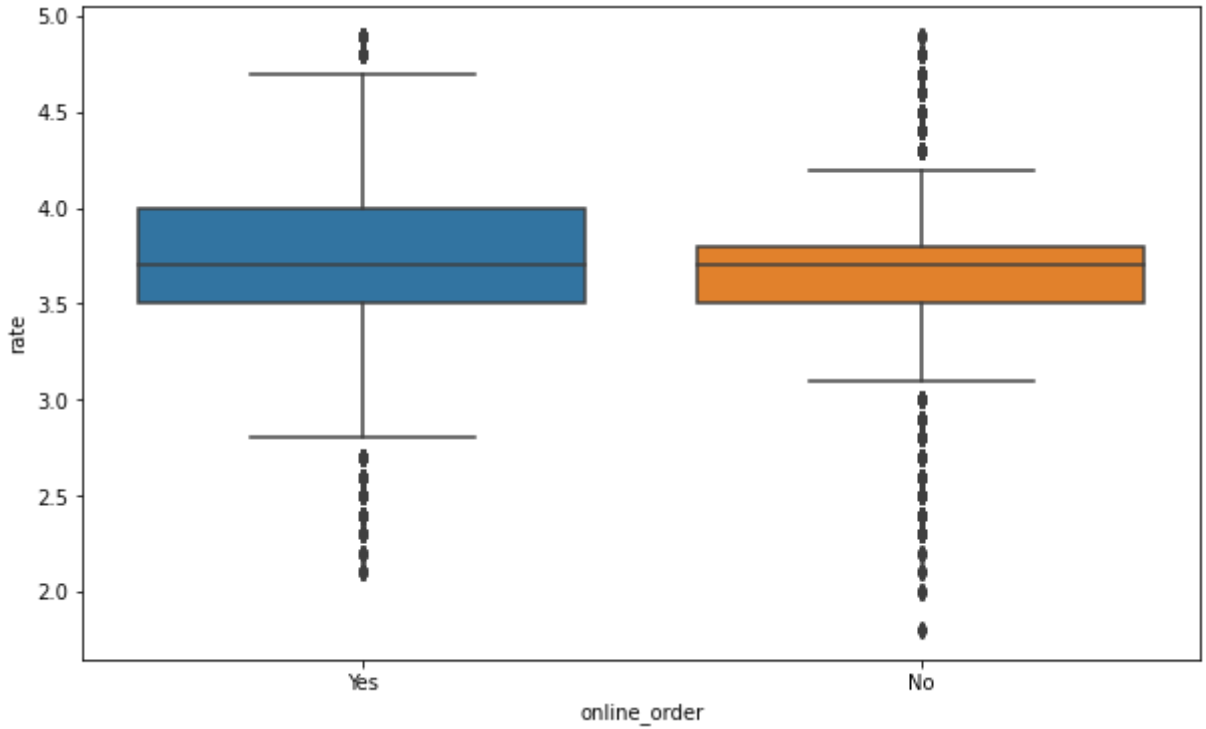
```
##Visualizing Online Order vs Rate

plt.figure(figsize = (10,6))
sns.boxplot(df['online_order'],df['rate'])
```

```
C:\Users\Administrator\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keywo
rd args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explic
it keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[107…

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



In [108…

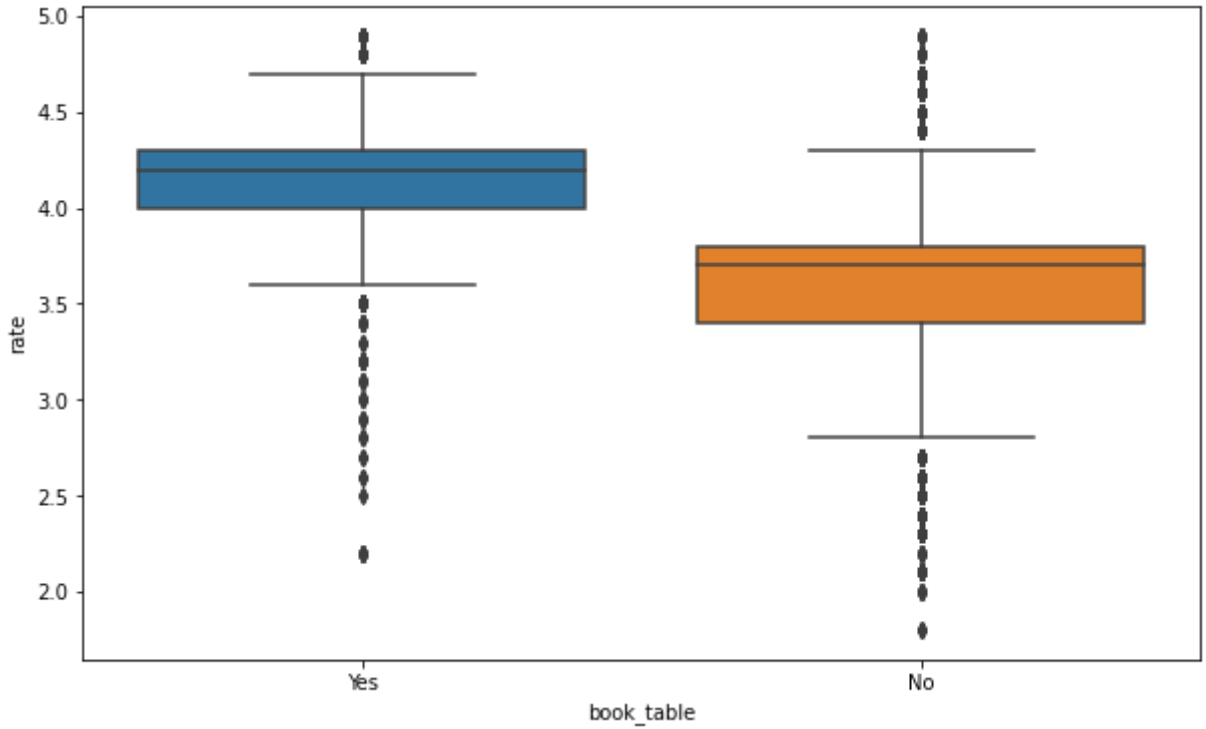
```
## Visualizing Book Table vs Rate

plt.figure(figsize = (10,6))
sns.boxplot(df['book_table'],df['rate'])
```

```
C:\Users\Administrator\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keywo
rd args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explic
it keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[108…

```
<AxesSubplot:xlabel='book_table', ylabel='rate'>
```



In [118…

```
##Visualizing Online Order Facility, Location Wise

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, index=['location'], columns=['online_order'], aggfunc=np.sum)
df1
```

Out[118…

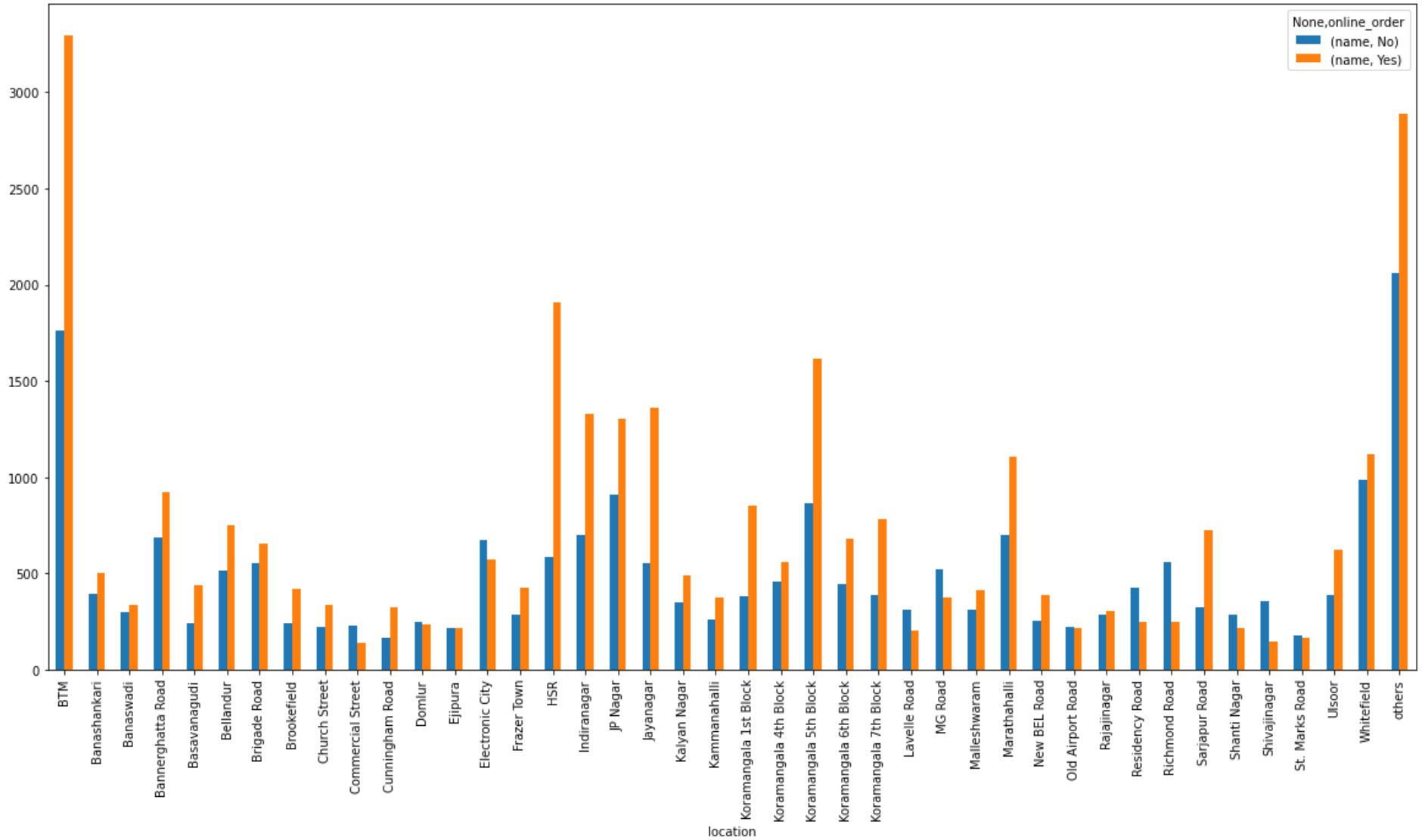
	name	
online_order	No	Yes
location		
BTM	1763	3293
Banashankari	397	505
Banaswadi	302	338

	name	
online_order	No	Yes
location		
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

	name	
online_order	No	Yes
location		
Koramangala 7th Block	389	785
Lavelle Road	315	203
MG Road	520	373
Malleshwaram	309	412
Marathahalli	701	1104
New BEL Road	255	389
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 [133…
```
df1.plot(kind='bar',figsize = (20,10))
```

Out[133…
```
<AxesSubplot:xlabel='location'>
```



In [136…

```
## Visualizing Book Table Facility, Location Wise

df2 = df.groupby(['location','book_table'])['name'].count()
df2.to_csv("location_book_table.csv")
df2 = pd.read_csv("location_book_table.csv")
df2 = pd.pivot_table(df2,index=['location'],columns=['book_table'],aggfunc=np.sum)
df2
```

Out[136...

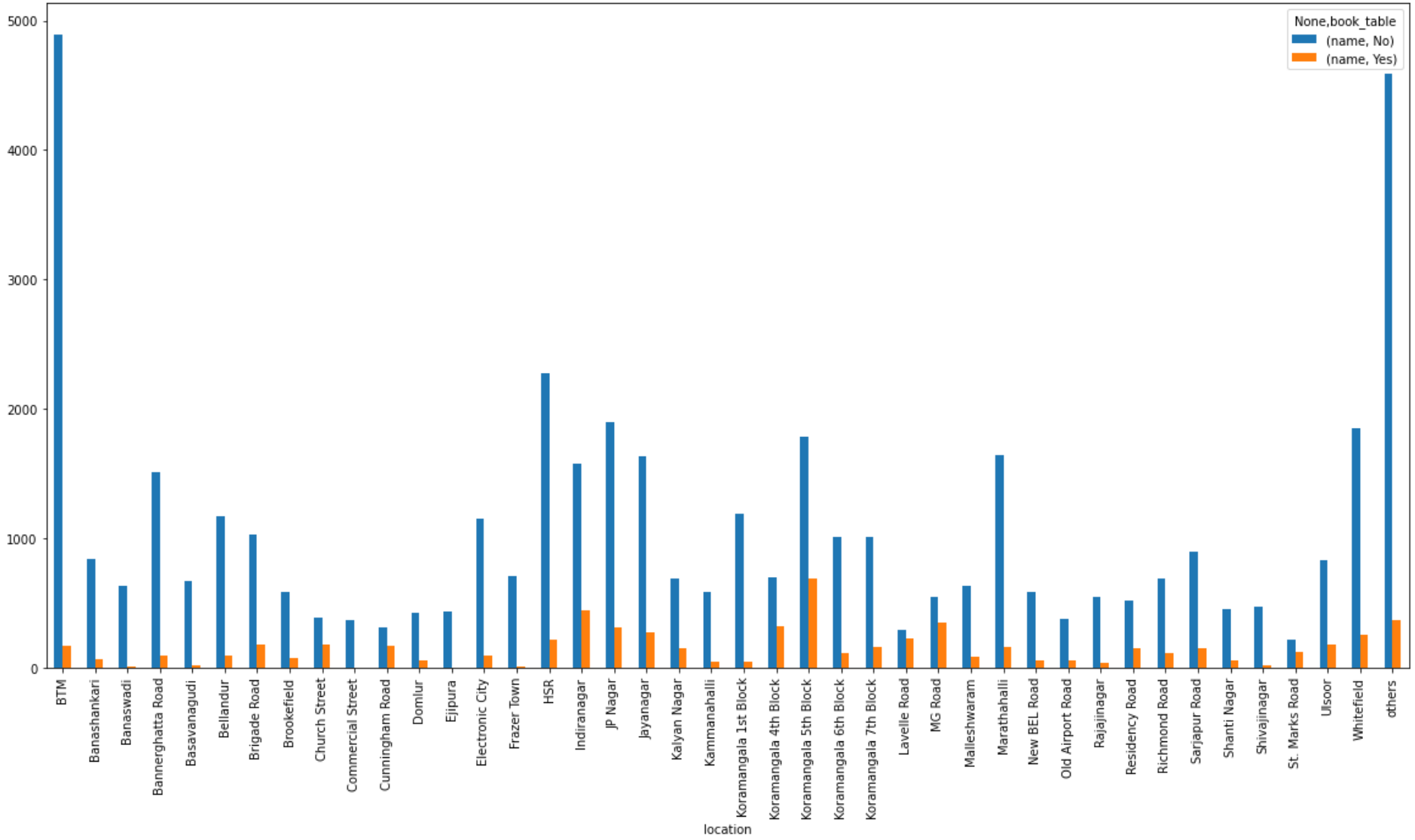
	name	
book_table	No	Yes
location		
BTM	4889.0	167.0
Banashankari	839.0	63.0
Banaswadi	632.0	8.0
Bannerghatta Road	1510.0	99.0
Basavanagudi	668.0	16.0
Bellandur	1170.0	98.0
Brigade Road	1034.0	176.0
Brookefield	582.0	74.0
Church Street	385.0	181.0
Commercial Street	370.0	NaN
Cunningham Road	315.0	175.0
Domlur	427.0	55.0
Ejipura	433.0	NaN
Electronic City	1148.0	98.0
Frazer Town	706.0	8.0
HSR	2277.0	217.0
Indiranagar	1578.0	448.0
JP Nagar	1903.0	315.0
Jayanagar	1637.0	279.0
Kalyan Nagar	692.0	149.0
Kammanahalli	590.0	49.0
Koramangala 1st Block	1186.0	50.0

	name	
book_table	No	Yes
location		
Koramangala 4th Block	695.0	322.0
Koramangala 5th Block	1787.0	692.0
Koramangala 6th Block	1015.0	112.0
Koramangala 7th Block	1012.0	162.0
Lavelle Road	290.0	228.0
MG Road	546.0	347.0
Malleshwaram	632.0	89.0
Marathahalli	1642.0	163.0
New BEL Road	588.0	56.0
Old Airport Road	378.0	59.0
Rajajinagar	550.0	41.0
Residency Road	522.0	149.0
Richmond Road	687.0	116.0
Sarjapur Road	893.0	154.0
Shanti Nagar	451.0	57.0
Shivajinagar	475.0	23.0
St. Marks Road	219.0	124.0
Ulsoor	834.0	177.0
Whitefield	1852.0	253.0
others	4587.0	367.0

In [137…

```
df2.plot(kind='bar',figsize=(20,10))
```

Out[137… <AxesSubplot:xlabel='location'>



In [139…

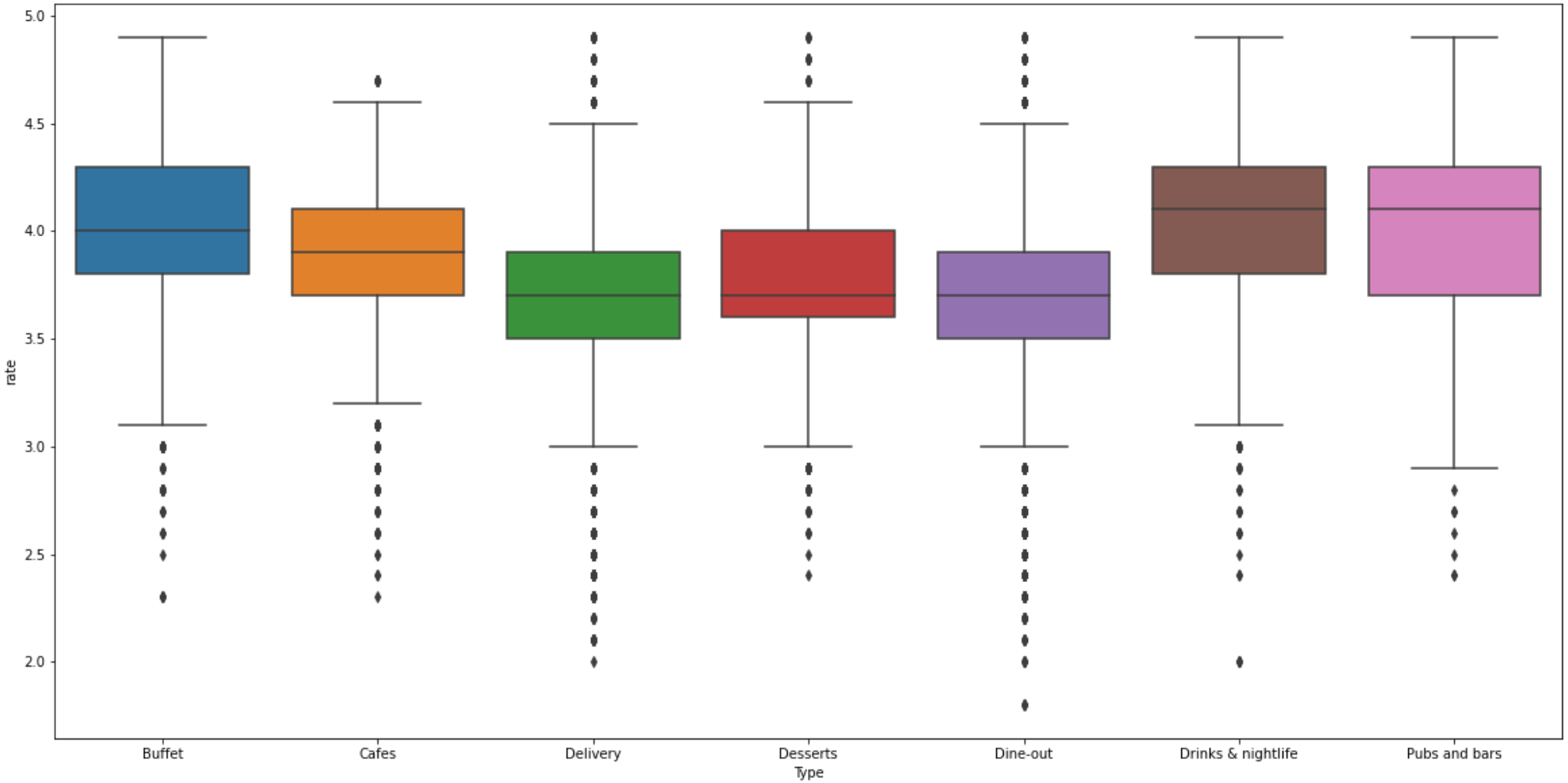
```
## Visualizing Types of Restaurents vs Rate

plt.figure(figsize=(20,10))
sns.boxplot(df['Type'],df['rate'])
```

```
C:\Users\Administrator\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keywo
rd args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explic
it keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[139…

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



In [142…

```
##Visualizing Type , Location Wise

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, index=['location'], columns=['Type'], aggfunc=np.sum)
df3
```

Out[142...

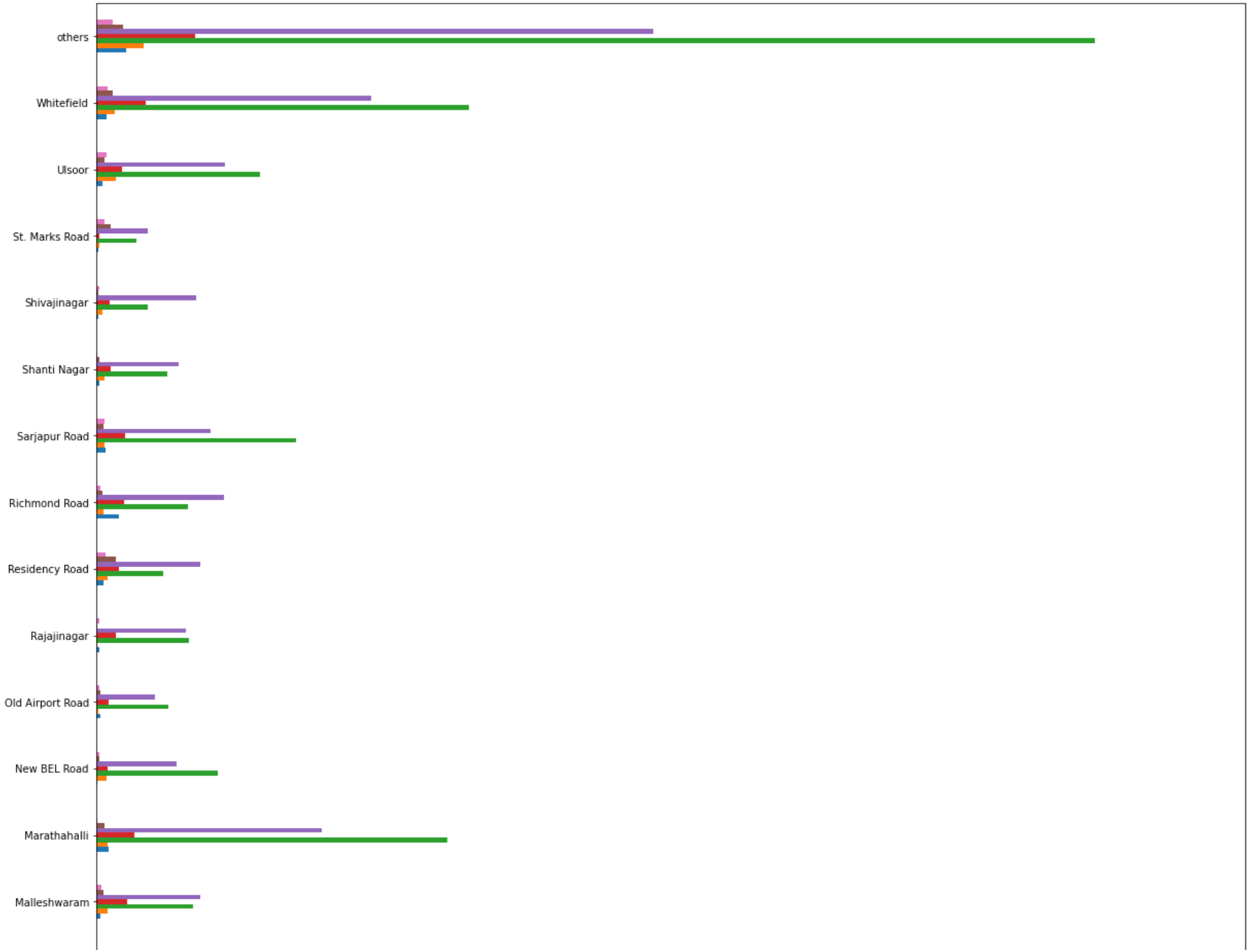
name							
Type	Buffet	Cafes	Delivery	Desserts	Dine-out	Drinks & nightlife	Pubs and bars
location							
BTM	21.0	83.0	3053.0	198.0	1660.0	22.0	19.0
Banashankari	7.0	36.0	418.0	71.0	356.0	14.0	NaN
Banaswadi	NaN	24.0	310.0	37.0	262.0	6.0	1.0
Bannerghatta Road	9.0	46.0	828.0	137.0	578.0	9.0	2.0
Basavanagudi	7.0	11.0	344.0	66.0	251.0	5.0	NaN
Bellandur	28.0	36.0	617.0	75.0	479.0	17.0	16.0
Brigade Road	25.0	46.0	497.0	108.0	455.0	57.0	22.0
Brookefield	6.0	17.0	339.0	45.0	245.0	4.0	NaN
Church Street	19.0	51.0	193.0	29.0	215.0	36.0	23.0
Commercial Street	NaN	13.0	121.0	77.0	159.0	NaN	NaN
Cunningham Road	29.0	34.0	194.0	26.0	184.0	16.0	7.0
Domlur	15.0	13.0	261.0	35.0	135.0	12.0	11.0
Ejipura	NaN	NaN	245.0	16.0	172.0	NaN	NaN
Electronic City	23.0	24.0	570.0	71.0	516.0	21.0	21.0
Frazer Town	1.0	11.0	470.0	56.0	172.0	2.0	2.0
HSR	19.0	49.0	1694.0	120.0	580.0	14.0	18.0
Indiranagar	38.0	97.0	1091.0	140.0	529.0	65.0	66.0
JP Nagar	45.0	76.0	1151.0	166.0	722.0	51.0	7.0
Jayanagar	27.0	77.0	1043.0	182.0	575.0	12.0	NaN
Kalyan Nagar	9.0	45.0	366.0	88.0	315.0	18.0	NaN
Kammanahalli	2.0	27.0	329.0	35.0	240.0	6.0	NaN
Koramangala 1st Block	3.0	26.0	716.0	70.0	398.0	7.0	16.0

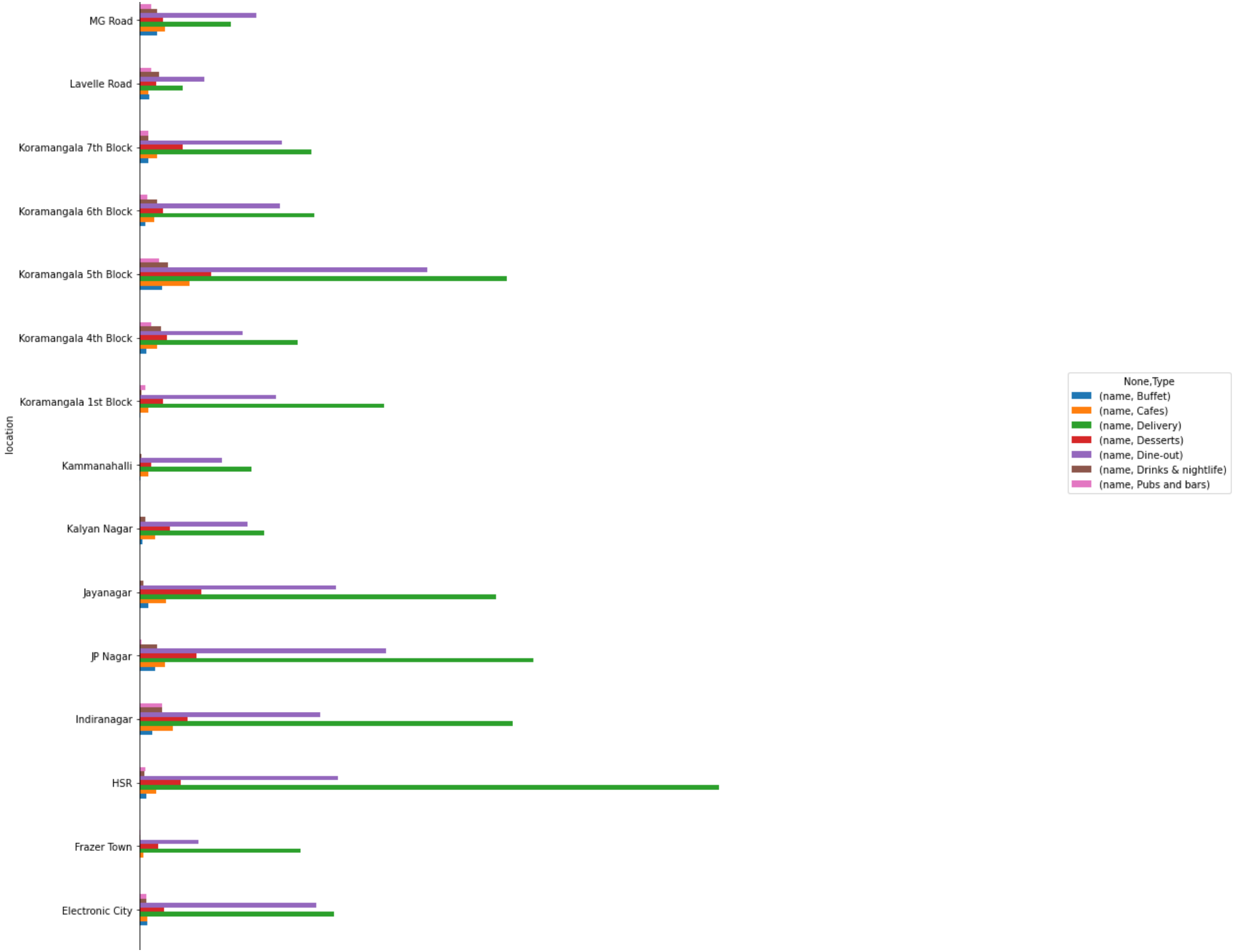
name Type location	Buffet	Cafes	Delivery	Desserts	Dine-out	Drinks & nightlife	Pubs and bars
Koramangala 4th Block	21.0	53.0	464.0	81.0	302.0	62.0	34.0
Koramangala 5th Block	65.0	146.0	1075.0	209.0	842.0	84.0	58.0
Koramangala 6th Block	18.0	43.0	511.0	70.0	411.0	51.0	23.0
Koramangala 7th Block	25.0	52.0	503.0	127.0	417.0	25.0	25.0
Lavelle Road	30.0	27.0	127.0	50.0	191.0	59.0	34.0
MG Road	51.0	76.0	266.0	68.0	343.0	53.0	36.0
Malleshwaram	11.0	31.0	269.0	85.0	291.0	20.0	14.0
Marathahalli	34.0	32.0	980.0	105.0	630.0	22.0	2.0
New BEL Road	4.0	29.0	338.0	33.0	224.0	8.0	8.0
Old Airport Road	12.0	5.0	200.0	35.0	164.0	12.0	9.0
Rajajinagar	10.0	4.0	258.0	55.0	251.0	3.0	10.0
Residency Road	20.0	31.0	187.0	63.0	289.0	55.0	26.0
Richmond Road	63.0	21.0	257.0	78.0	356.0	16.0	12.0
Sarjapur Road	25.0	22.0	558.0	82.0	319.0	19.0	22.0
Shanti Nagar	9.0	22.0	198.0	39.0	229.0	9.0	2.0
Shivajinagar	6.0	17.0	143.0	37.0	280.0	7.0	8.0
St. Marks Road	5.0	10.0	111.0	10.0	145.0	40.0	22.0
Ulsoor	16.0	56.0	456.0	71.0	359.0	23.0	30.0
Whitefield	28.0	51.0	1041.0	137.0	768.0	47.0	33.0
others	83.0	133.0	2787.0	276.0	1553.0	75.0	47.0

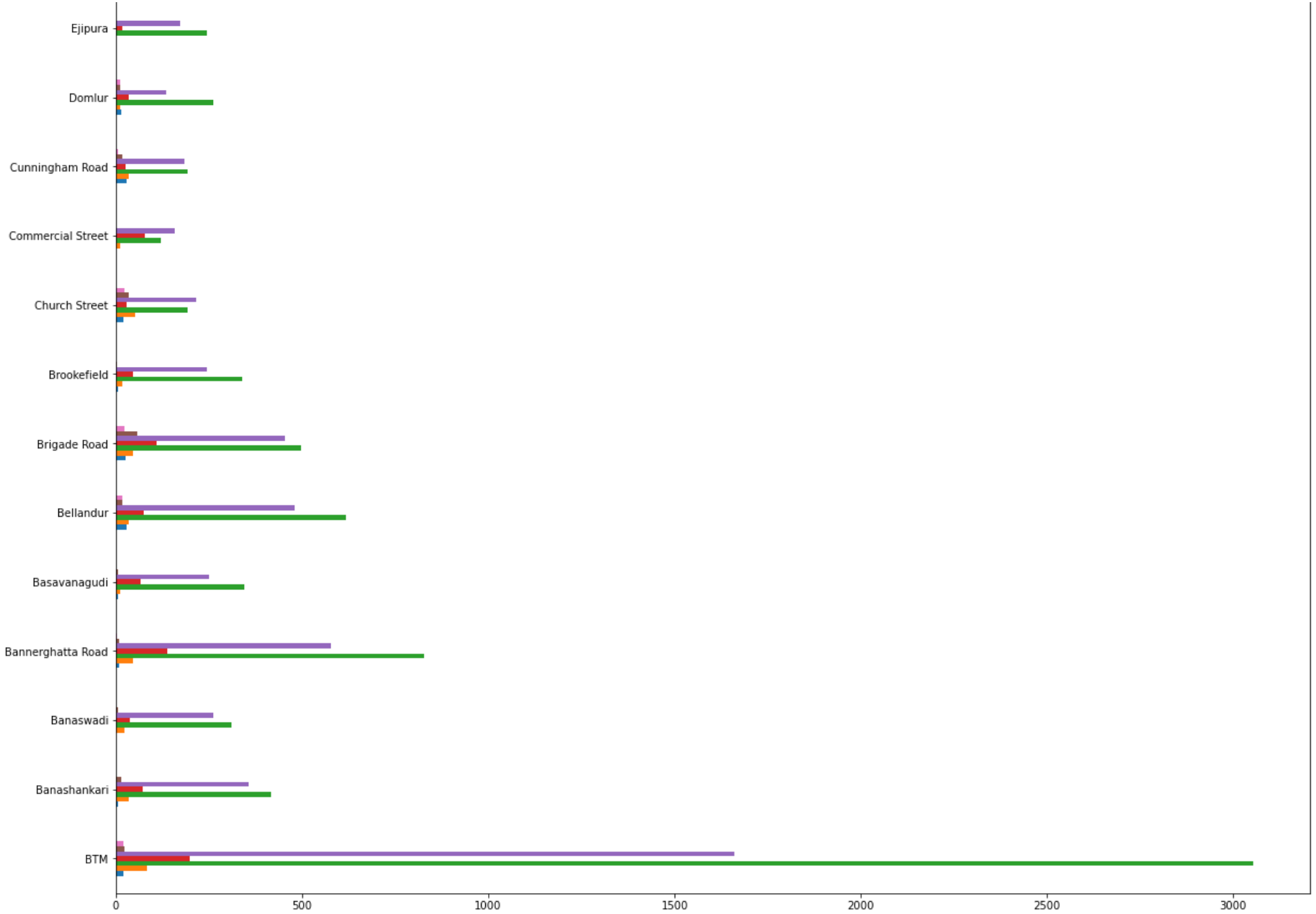
In [147…

```
df3.plot(kind = 'barh', figsize = (20,50))
```

Out[147… <AxesSubplot:ylabel='location'>




MG Road
Lavelle Road
Koramangala 7th Block
Koramangala 6th Block
Koramangala 5th Block
Koramangala 4th Block
Koramangala 1st Block
Kammanahalli
Kalyan Nagar
Jayanagar
JP Nagar
Indiranagar
HSR
Frazer Town
Electronic City
location
None,Type
(name, Buffet)
(name, Cafes)
(name, Delivery)
(name, Desserts)
(name, Dine-out)
(name, Drinks & nightlife)
(name, Pubs and bars)



In [161…

```
## No_of votes , location wise

df4 = df[['location','votes']]
```

```
df5 = df4.groupby(['location'])['votes'].sum()
df6 = df5.sort_values(ascending=False)
df6
```

Out[161…

```
location
Koramangala 5th Block    2214083
Indiranagar              1165909
Koramangala 4th Block     685156
Church Street             590306
JP Nagar                  586522
BTM                       573668
others                    563807
HSR                       498322
Lavelle Road              495777
Koramangala 7th Block     488225
Jayanagar                 487044
Whitefield                465734
Koramangala 6th Block     463503
Marathahalli              434235
MG Road                   428266
Brigade Road              426682
Sarjapur Road             398599
Residency Road            290513
Cunningham Road           287471
St. Marks Road            266099
Koramangala 1st Block     251681
Malleshwaram              238967
Bannerghatta Road         214989
Bellandur                 205308
Ulsoor                    180232
New BEL Road              175687
Kalyan Nagar              167992
Banashankari              162374
Old Airport Road          137832
Brookefield               118962
Richmond Road             118902
Electronic City           110774
Kammanahalli              105250
Frazer Town                97668
Domlur                     96721
Basavanagudi               94919
Rajajinagar                85274
Shanti Nagar               55298
```

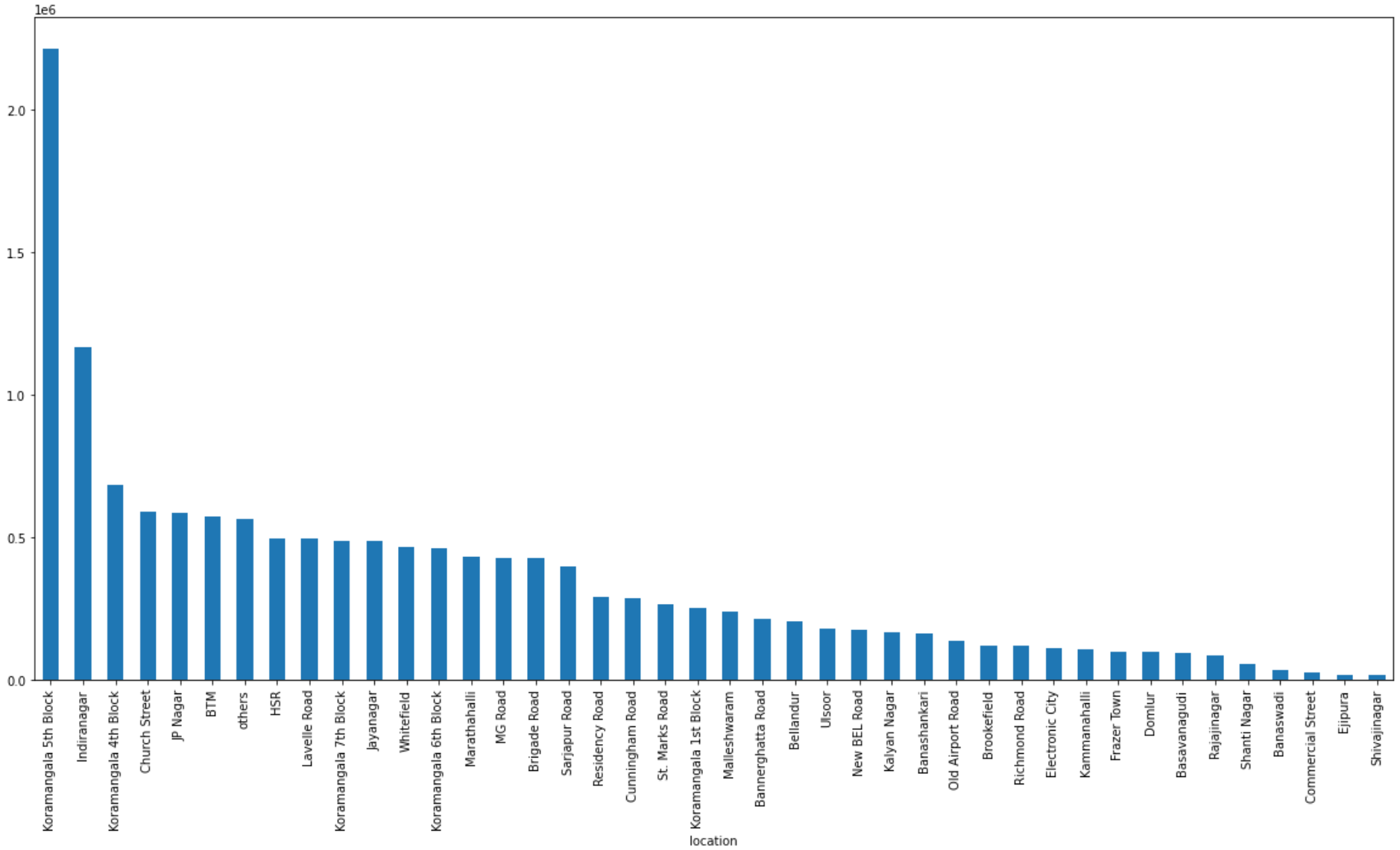
```
Banaswadi                 34845
Commercial Street         25563
Ejipura                   17015
Shivajinagar              15668
Name: votes, dtype: int64
```

In [162…

```
df6.plot(kind='bar', figsize=(20,10))
```

Out[162…

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



In [187…

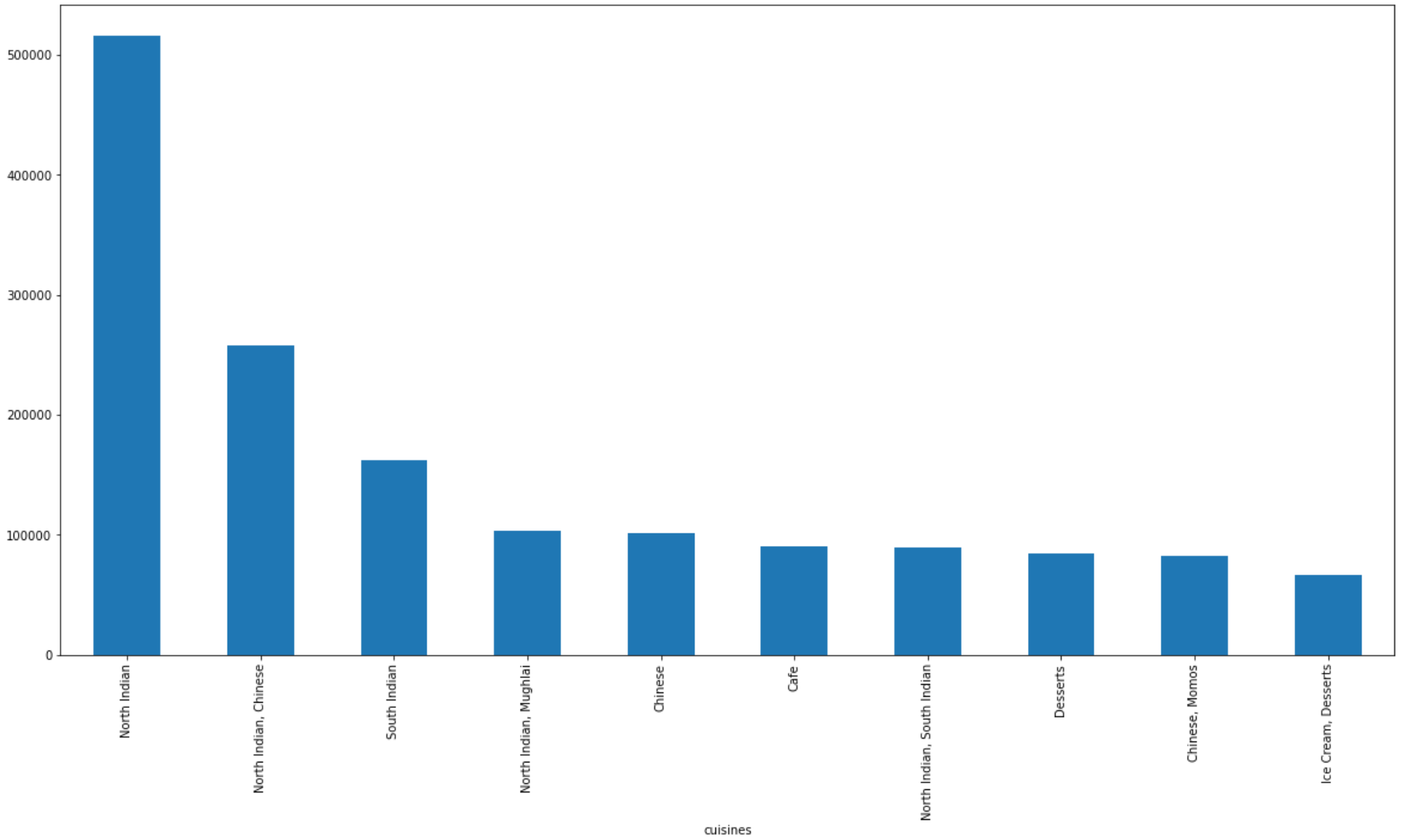
```
## find out Top 10 cuisines

df7 = df[['cuisines','votes']]
df8 = df7.groupby(['cuisines'])['votes'].sum()
df9 = df8.sort_values(ascending=False)
df9[1:11]
```

Out[187…
```
cuisines
North Indian                  516310
North Indian, Chinese         258225
South Indian                  161975
North Indian, Mughlai         103706
Chinese                       101728
Cafe                           89986
North Indian, South Indian     88925
Desserts                       84323
Chinese, Momos                 82796
Ice Cream, Desserts            66437
Name: votes, dtype: int64
```

In [191…
```
df9[1:11].plot(kind='bar' ,figsize=(20,10) )
```

Out[191…
```
<AxesSubplot:xlabel='cuisines'>
```


500000
400000
300000
200000
100000
0
North Indian
North Indian, Chinese
South Indian
North Indian, Mughlai
Chinese
Cafe
North Indian, South Indian
Desserts
Chinese, Momos
Ice Cream, Desserts
cuisines

In [ ]: