```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('dark_background')
```

```
!pip install opendatasets
```

Show hidden output

```
import opendatasets as od
import pandas as pd
od.download("https://www.kaggle.com/datasets/himanshupoddar/zomato-bangalore-restaurants")
```

Show hidden output

```
import pandas as pd
df = pd.read_csv("/content/zomato-bangalore-restaurants/zomato.csv")
```

```
df.head()
```

	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

```
df.shape
```

```
(51717, 17)
```

```
df.columns
```

```
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')
```

```
df = df.drop(['url', 'address', 'phone', 'menu_item', 'dish_liked', 'reviews_list'], axis = 1)
df.head()
```

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

```
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
```

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

```
(51609, 11)
```

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

```
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)
```

```
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()
```

	rate
0	4.1
1	4.1
2	3.8
3	3.7
4	3.8

```
df.rate.isnull().sum()
```

```
10019
```

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

```
<ipython-input-96-aaee158040fb>:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained ass
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[co


  df['rate'].fillna(df['rate'].mean(), inplace = True)
0
```

```
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
```

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

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	approx_cost(for two people)	listed_in(type)	listed_in(ci
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	800	Buffet	Banashan
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	800	Buffet	Banashan
	San						Cafe,	Cafe,			

```
df.rename(columns = {'approx_cost(for two people)':'cost2plates', 'listed_in(type)':'Type'}, inplace = True)
df.head()
```

	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

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

```
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)
```

```
df.head()
```

	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	Cafe, Casual Dining	Cafe, Mexican, Italian	800	Buffet
	Addhuri Udupi							South Indian, North		

Next steps: Generate code with df View recommended plots New interactive sheet

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

```
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',
       'Beverage Shop, Quick Bites', 'Confectionery',
       'Quick Bites, Beverage Shop', 'Dessert Parlor, Sweet Shop',
       'Bakery, Quick Bites', 'Sweet Shop, Quick Bites', 'Kiosk',
       'Food Truck', 'Quick Bites, Dessert Parlor',
       'Beverage Shop, Dessert Parlor', 'Takeaway', 'Pub, Casual Dining',
       'Casual Dining, Bar', 'Dessert Parlor, Beverage Shop',
       'Quick Bites, Bakery', 'Dessert Parlor, Quick Bites',
       'Microbrewery, Casual Dining', 'Lounge', 'Bar, Casual Dining',
       'Food Court', 'Cafe, Bakery', 'Dhaba', 'Quick Bites, Sweet Shop',
       'Microbrewery', 'Food Court, Quick Bites', 'Pub, Bar',
       'Casual Dining, Pub', 'Lounge, Bar', 'Food Court, Dessert Parlor',
       'Casual Dining, Sweet Shop', 'Food Court, Casual Dining',
       'Casual Dining, Microbrewery', 'Sweet Shop, Dessert Parlor',
       'Bakery, Beverage Shop', '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', 'Bar, Cafe',
       'Pub, Cafe', 'Casual Dining, Irani Cafee', 'Fine Dining, Lounge',
       'Bar, Quick Bites', 'Bakery, Kiosk', 'Pub, Microbrewery',
       'Microbrewery, Lounge', 'Fine Dining, Microbrewery',
       'Fine Dining, Bar', 'Mess, Quick Bites', 'Dessert Parlor, Kiosk',
       'Bhojanalya', 'Casual Dining, Quick Bites', 'Pop Up', 'Cafe, Bar',
       'Casual Dining, Lounge', 'Bakery, Sweet 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)
```

```
df['rest_type'].value_counts()
```

```
                              count
                 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

93 rows × 1 columns

dtype: int64
```

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

```
                                count
                  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

93 rows × 1 columns
```

```
rest_types_lessthan1000 = rest_types[rest_types<1000]
rest_types_lessthan1000
```

```
                                count
                  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

85 rows × 1 columns
```

```
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()
```

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

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

location	count
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

93 rows × 1 columns

```
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()
```

```
                       count
           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

dtype: int64
```

```
df['cuisines'].value_counts()
```

```
                                          count
                               cuisines
                           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

2704 rows × 1 columns

dtype: int64
```

```
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()
```

```
                                     count
                          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

70 rows × 1 columns

dtype: int64
```

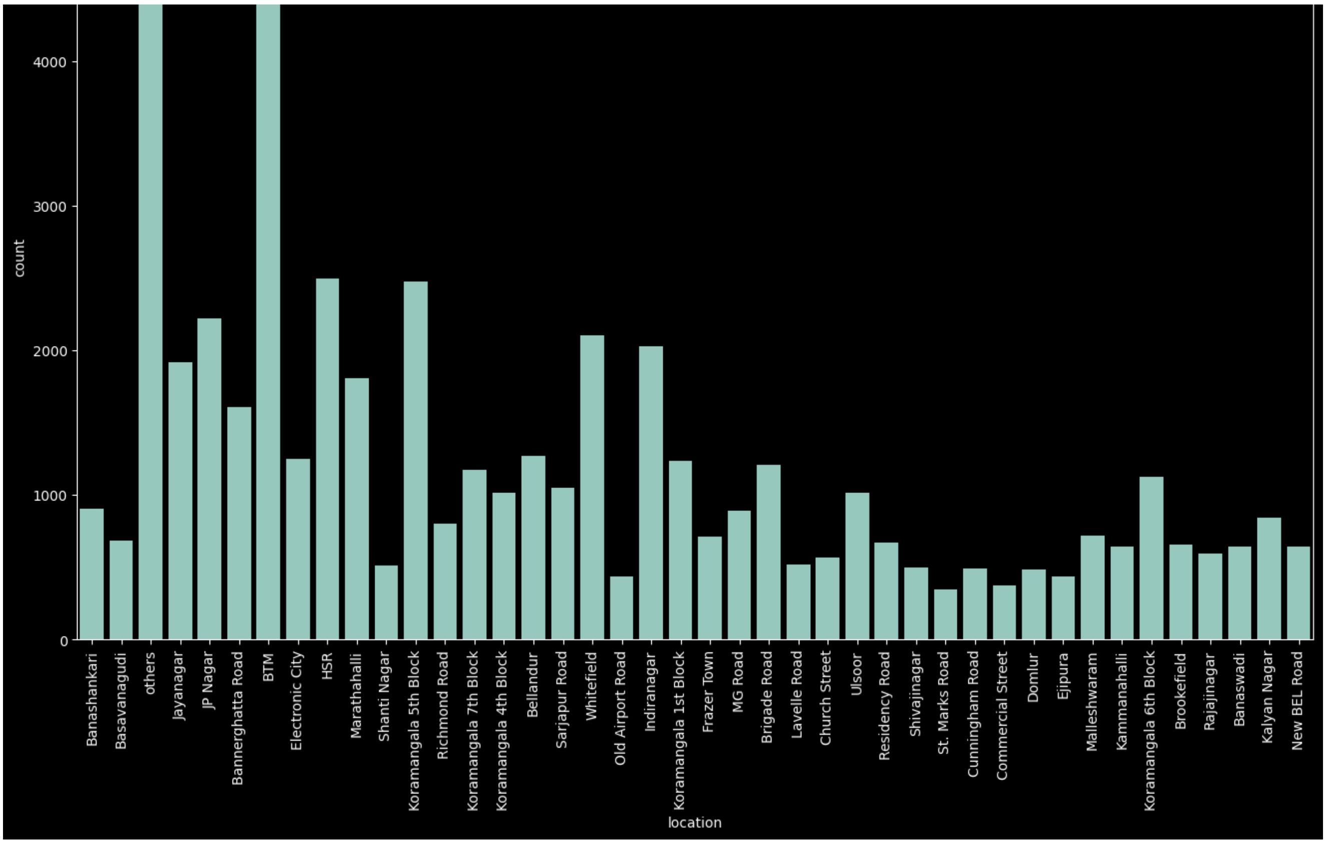
```
df.head()
```

```
             name  online_order  book_table  rate  votes      location   rest_type                 cuisines  cost2plates    Type
0           Jalsa           Yes         Yes   4.1    775  Banashankari      Casual   North Indian, Mughlai,          800  Buffet
                                                                            Dining                  Chinese
1  Spice Elephant           Yes          No   4.1    787  Banashankari      Casual                   others          800  Buffet
                                                                            Dining
2 San Churro Cafe           Yes          No   3.8    918  Banashankari      others                   others          800  Buffet
3   Addhuri Udupi            No          No   3.7     88  Banashankari Quick Bites     South Indian, North           300  Buffet
          Bhoiana                                                                                   Indian
```

Next steps: Generate code with `df` View recommended plots New interactive sheet

```
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(16, 10))
ax = sns.countplot(x="location", data=df)
plt.xticks(rotation=90)
```

```
([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, 'Banashankari'),
  Text(1, 0, 'Basavanagudi'),
  Text(2, 0, 'others'),
  Text(3, 0, 'Jayanagar'),
  Text(4, 0, 'JP Nagar'),
  Text(5, 0, 'Bannerghatta Road'),
  Text(6, 0, 'BTM'),
  Text(7, 0, 'Electronic City'),
  Text(8, 0, 'HSR'),
  Text(9, 0, 'Marathahalli'),
  Text(10, 0, 'Shanti Nagar'),
  Text(11, 0, 'Koramangala 5th Block'),
  Text(12, 0, 'Richmond Road'),
  Text(13, 0, 'Koramangala 7th Block'),
  Text(14, 0, 'Koramangala 4th Block'),
  Text(15, 0, 'Bellandur'),
  Text(16, 0, 'Sarjapur Road'),
  Text(17, 0, 'Whitefield'),
  Text(18, 0, 'Old Airport Road'),
  Text(19, 0, 'Indiranagar'),
  Text(20, 0, 'Koramangala 1st Block'),
  Text(21, 0, 'Frazer Town'),
  Text(22, 0, 'MG Road'),
  Text(23, 0, 'Brigade Road'),
  Text(24, 0, 'Lavelle Road'),
  Text(25, 0, 'Church Street'),
  Text(26, 0, 'Ulsoor'),
  Text(27, 0, 'Residency Road'),
  Text(28, 0, 'Shivajinagar'),
  Text(29, 0, 'St. Marks Road'),
  Text(30, 0, 'Cunningham Road'),
  Text(31, 0, 'Commercial Street'),
  Text(32, 0, 'Domlur'),
  Text(33, 0, 'Ejipura'),
  Text(34, 0, 'Malleshwaram'),
  Text(35, 0, 'Kammanahalli'),
  Text(36, 0, 'Koramangala 6th Block'),
  Text(37, 0, 'Brookefield'),
  Text(38, 0, 'Rajajinagar'),
  Text(39, 0, 'Banaswadi'),
  Text(40, 0, 'Kalyan Nagar'),
  Text(41, 0, 'New BEL Road')])
```


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

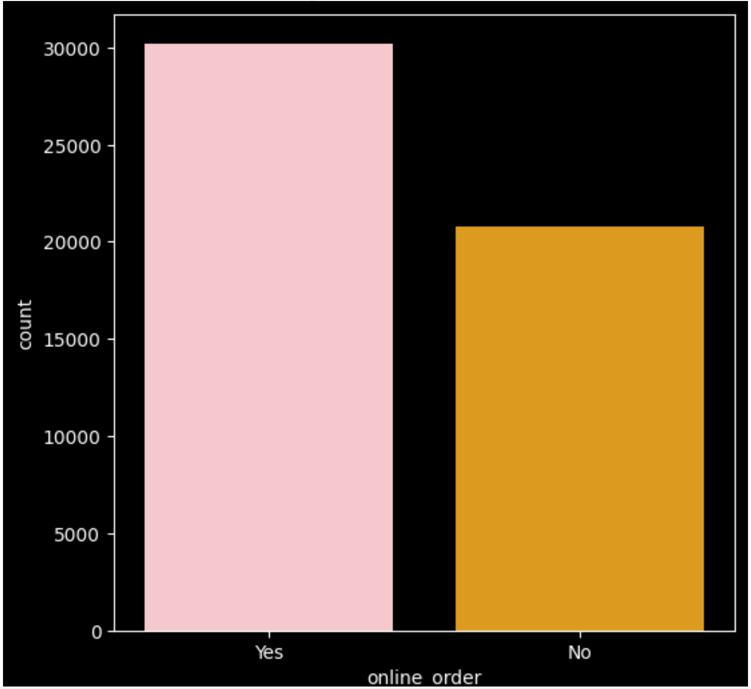
```
df.head()
```

	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	others	800	Buffet
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	others	others	800	Buffet
	Addhuri Udupi							South Indian, North		

Next steps: Generate code with df | View recommended plots | New interactive sheet

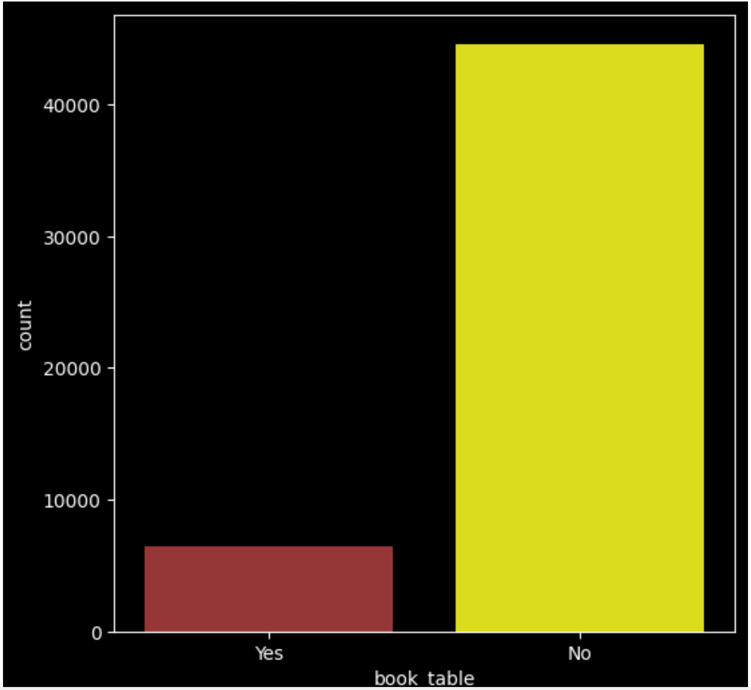
```
plt.figure(figsize = (6 , 6))
sns.countplot(x="online_order", data=df, palette=["pink", "orange"])
```

```
<ipython-input-141-f3914c1084ff>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.countplot(x="online_order", data=df, palette=["pink", "orange"])
<Axes: xlabel='online_order', ylabel='count'>
```



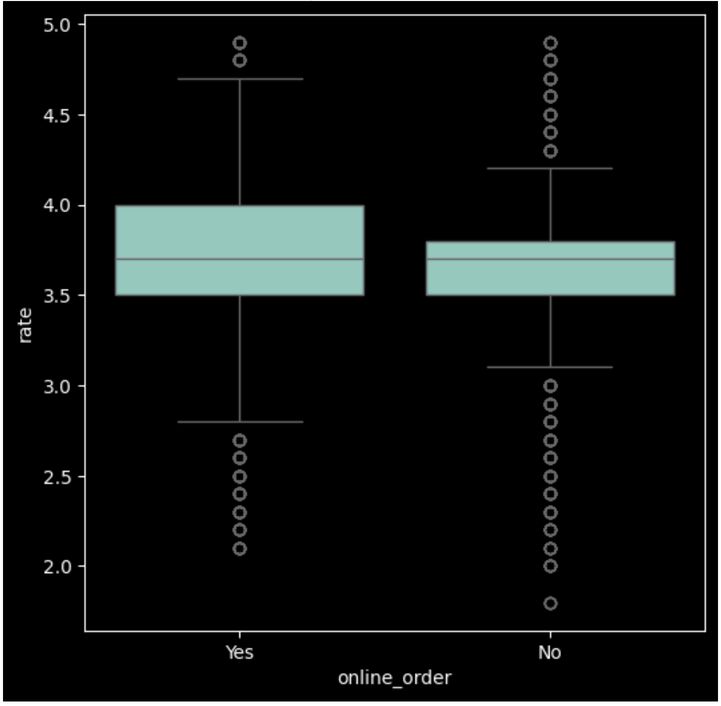
```
plt.figure(figsize = (6 , 6))
sns.countplot(x="book_table", data=df, palette=["brown", "yellow"])
```

```
<ipython-input-142-4c6ec46baddd>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.countplot(x="book_table", data=df, palette=["brown", "yellow"])
<Axes: xlabel='book_table', ylabel='count'>
```



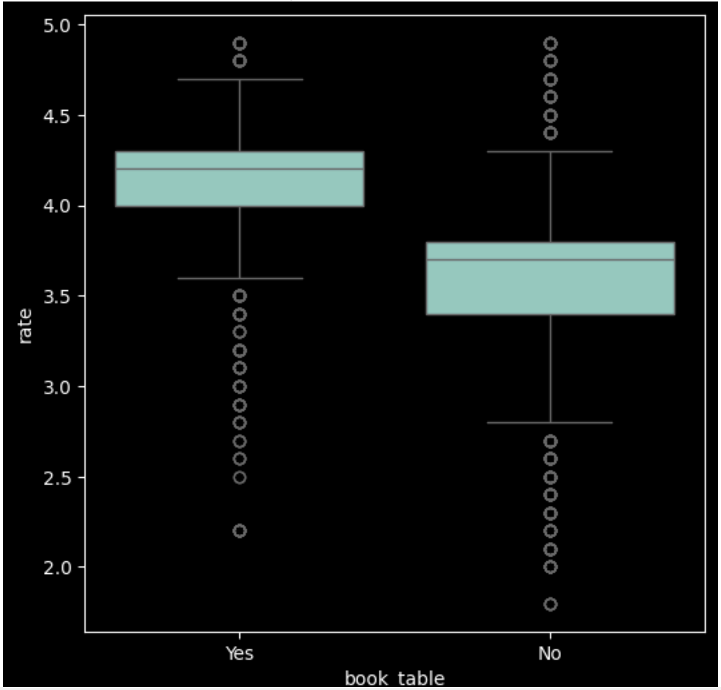
```
plt.figure(figsize = (6,6))
sns.boxplot(x = 'online_order', y = 'rate', data = df)
```

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



```
plt.figure(figsize = (6,6))
sns.boxplot(x = 'book_table', y = 'rate', data = df)
```

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



```
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_order'],fill_value=0,aggfunc=np.sum)
df1
```

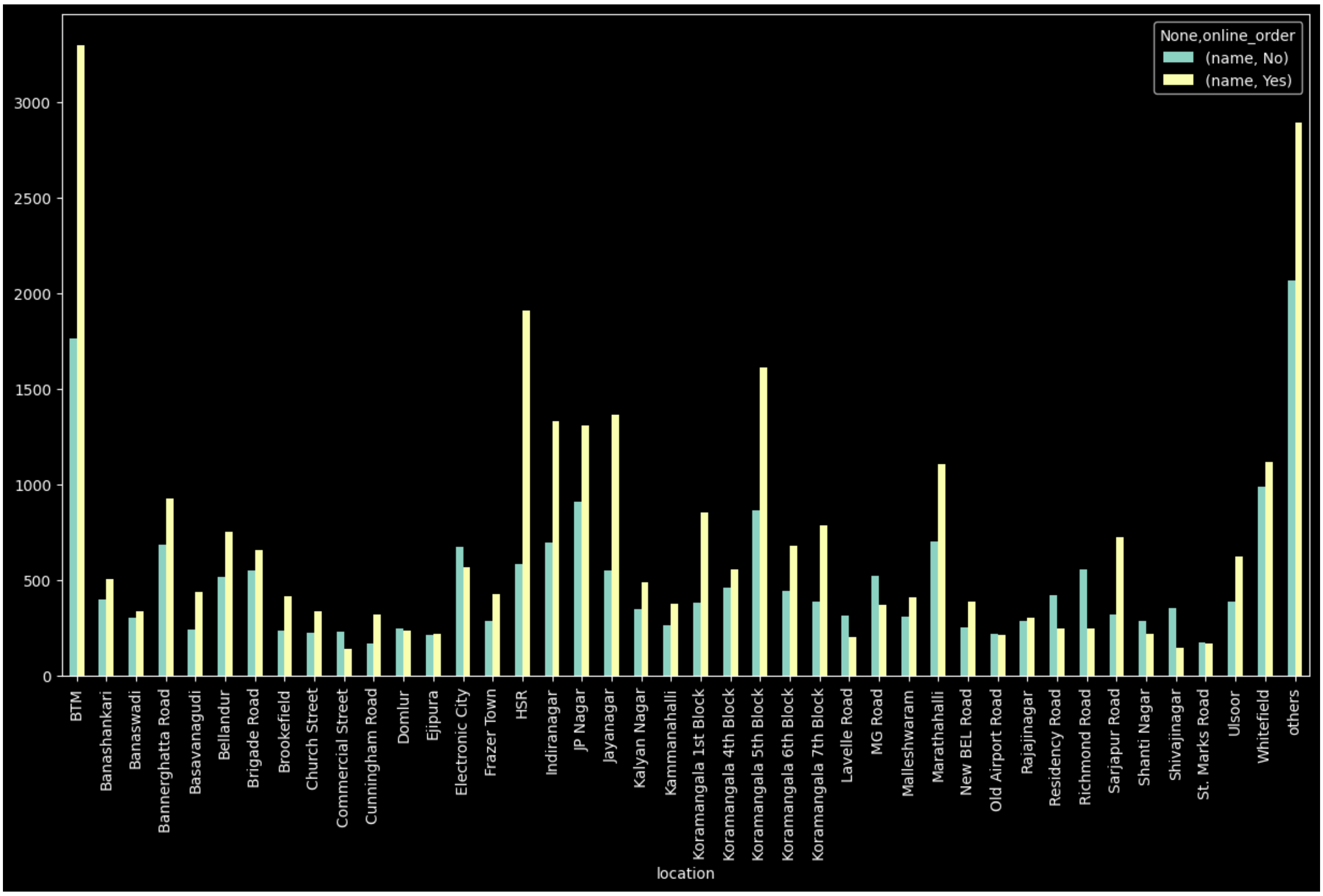
```
<ipython-input-145-b148e2592dd6>:4: FutureWarning: The provided callable <function sum at 0x7f0c61dfc670> is currently using DataFra
  df1 = pd.pivot_table(df1, values=None, index=['location'],columns=['online_order'],fill_value=0,aggfunc=np.sum)
```

```
                              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
             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
```

Next steps: Generate code with df1 View recommended plots New interactive sheet

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

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



```
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'],fill_value=0, aggfunc=np.sum)
df2
```

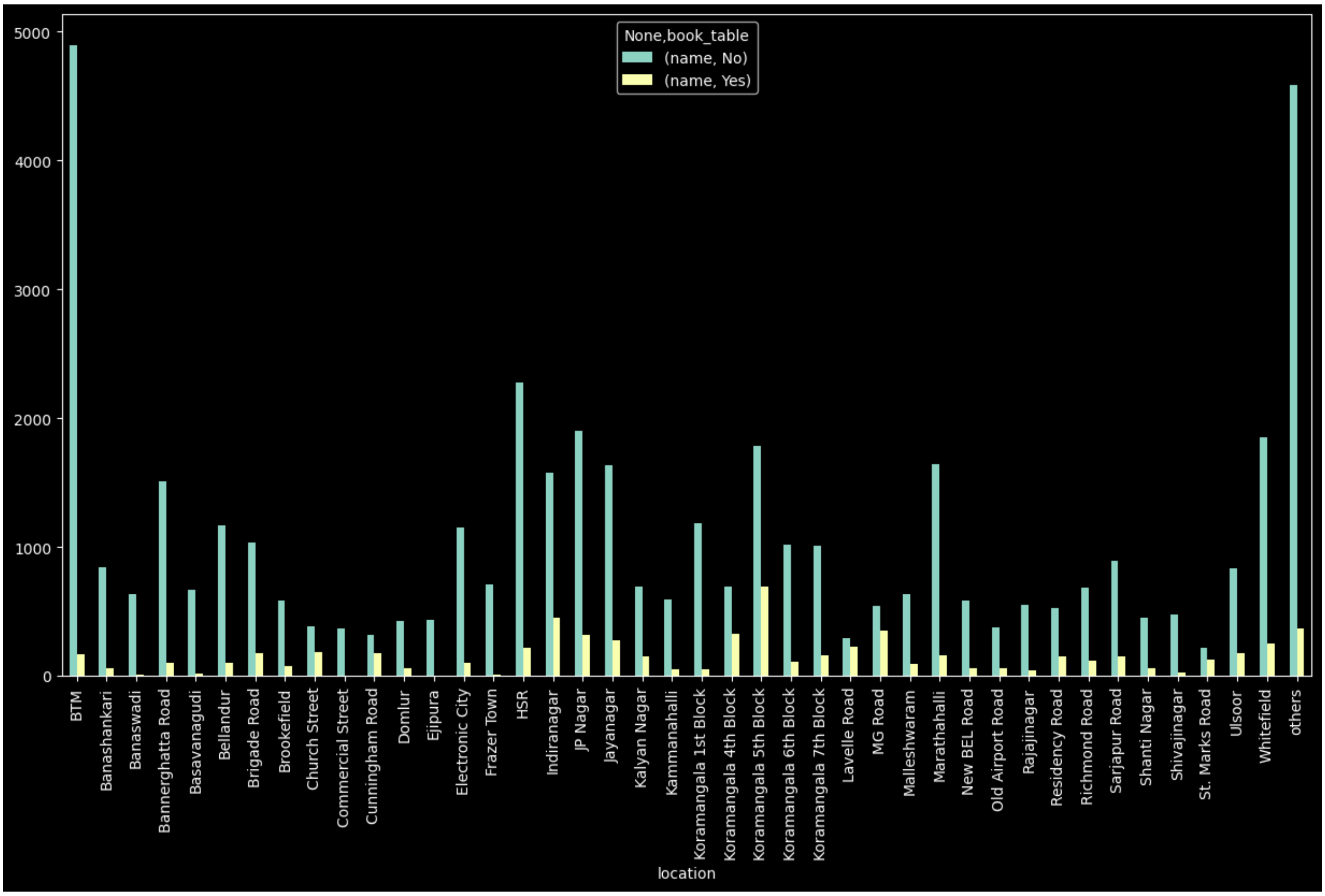
```
<ipython-input-147-3f242dc8ad86>:4: FutureWarning: The provided callable <function sum at 0x7f0c61dfc670> is currently using DataFra
  df2 = pd.pivot_table(df2, values=None, index=['location'], columns=['book_table'],fill_value=0, aggfunc=np.sum)
```

	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
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

Next steps: Generate code with df2 View recommended plots New interactive sheet

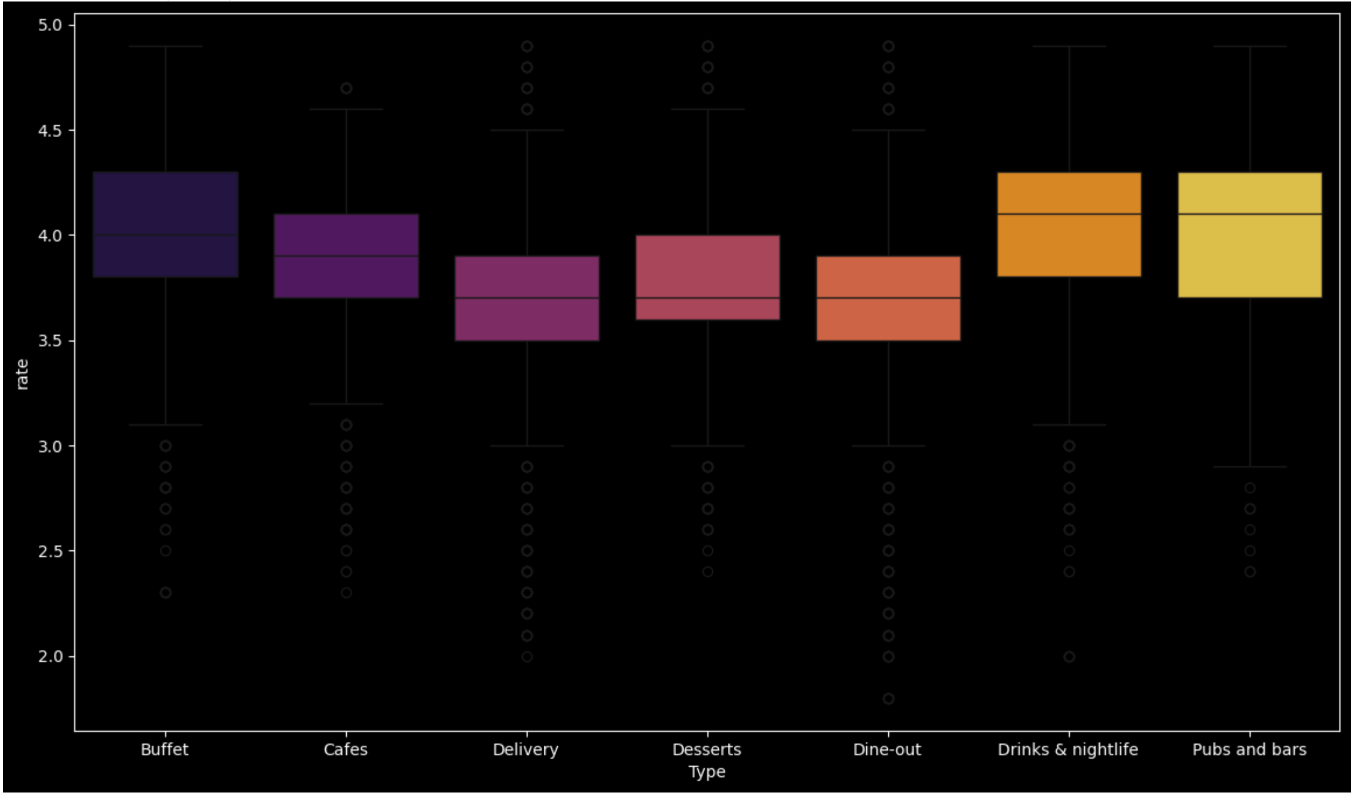
```
df2.plot(kind = 'bar', figsize = (15,8))
```

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



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

```
<ipython-input-150-633fd75d258e>:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `le

  sns.boxplot(x = 'Type', y = 'rate', data = df, palette = 'inferno')
<Axes: xlabel='Type', ylabel='rate'>
```



```
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'], fill_value=0, aggfunc=np.sum)
df3
```

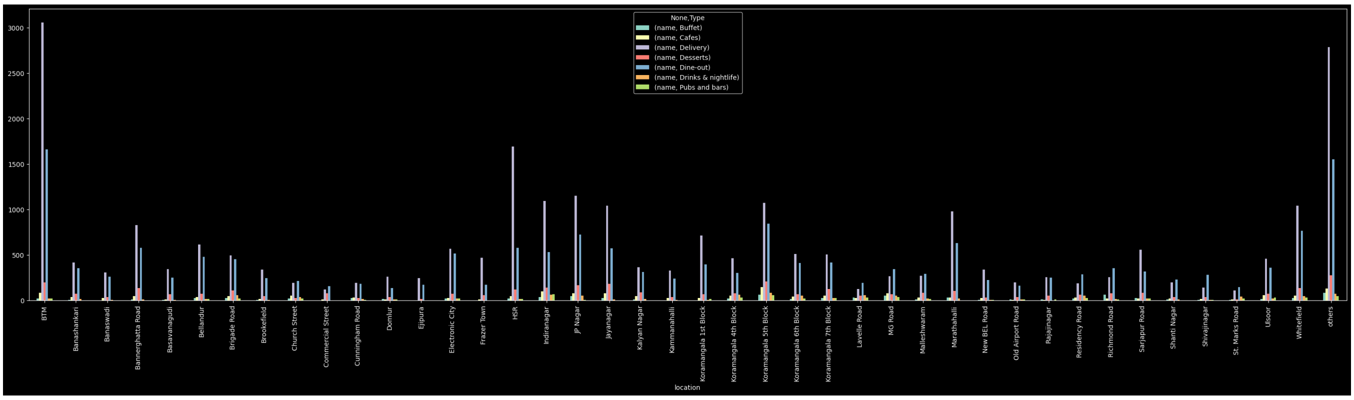
```
<ipython-input-151-842f7b0b8d59>:4: FutureWarning: The provided callable <function sum at 0x7f0c61dfc670> is currently using DataFra
  df3 = pd.pivot_table(df3, values=None, index=['location'], columns=['Type'], fill_value=0, aggfunc=np.sum)
```

	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
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

Next steps: Generate code with df3 View recommended plots New interactive sheet

```
df3.plot(kind = 'bar', figsize = (36,8))
```

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



```
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()
```

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

Next steps: Generate code with df5 View recommended plots New interactive sheet

```
plt.figure(figsize = (15,8))
sns.barplot(x=df5.index, y=df5['votes'])
plt.xticks(rotation = 90)
```