

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

```
In [2]: df= pd.read_csv(r'C:\Users\Administrator\zomato.csv')
```

```
In [3]: df.shape
```

```
Out[3]: (51717, 17)
```

```
In [4]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51717 entries, 0 to 51716
Data columns (total 17 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   url              51717 non-null   object 
 1   address          51717 non-null   object 
 2   name              51717 non-null   object 
 3   online_order      51717 non-null   object 
 4   book_table        51717 non-null   object 
 5   rate              43942 non-null   object 
 6   votes             51717 non-null   int64  
 7   phone             50509 non-null   object 
 8   location          51696 non-null   object 
 9   rest_type         51490 non-null   object 
 10  dish_liked        23639 non-null   object 
 11  cuisines          51672 non-null   object 
 12  approx_cost(for two people) 51371 non-null   object 
 13  reviews_list      51717 non-null   object 
 14  menu_item         51717 non-null   object 
 15  listed_in(type)   51717 non-null   object 
 16  listed_in(city)   51717 non-null   object 
dtypes: int64(1), object(16)
memory usage: 6.7+ MB
```

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

```
Out[5]: url          object
address       object
name          object
online_order   object
book_table     object
rate           object
votes          int64
phone          object
location        object
rest_type       object
dish_liked      object
cuisines        object
approx_cost(for two people) object
reviews_list    object
menu_item       object
listed_in(type) object
listed_in(city) object
dtype: object
```

```
In [6]: df.drop(['url', 'address', 'phone', 'location', 'dish_liked', 'reviews_list', 'menu_it
```

```
In [7]: df=df.rename(columns={"name":'Name', 'rate':'Ratings', 'votes':'Votes', 'rest_type':
```

In [8]: `df.sample(10)`

Out[8]:

		Name	Takes online orders?	Has table booking?	Ratings	Votes	Rest_Type	Cuisines	Cost	Type
46735	Mysore Hanumanthu Palav		Yes	No	3.9 /5	74	Casual Dining	South Indian, Fast Food	500	Dine-out
4035	Spicy Punjabi Pharatha		Yes	No	2.9/5	76	Quick Bites	North Indian, Chinese, Biryani	400	Delivery
5552	Ebony		Yes	Yes	4.3/5	3126	Casual Dining	North Indian, South Indian	1,500	Delivery
39165	Hotel Chandrika		Yes	No	3.8 /5	208	Casual Dining	South Indian, North Indian, Chinese, Street Food	400	Dine-out
22856	Sultan's Biryani		No	No	3.0/5	12	Delivery	Biryani, North Indian	700	Delivery
50645	Kitchen Online		No	No	3.3 /5	6	Takeaway, Delivery	North Indian	300	Delivery
18681	Ciclo Cafe		Yes	No	4.3/5	1273	Cafe, Casual Dining	Cafe, Italian, American	1,000	Dine-out
43890	Manhattan		No	No	NaN	0	Bar	American, Finger Food	1,000	Drinks & nightlife
48905	Punjabi Dawat		Yes	No	2.9 /5	254	Quick Bites	North Indian, Chinese	400	Delivery
305	Cafe Aira		Yes	No	4.1/5	216	Cafe	Cafe, Continental, Beverages, Desserts	500	Delivery



In [9]: `sum(df.duplicated())`

Out[9]: 124

In [10]: `df=df.drop_duplicates()`

```
In [11]: def name_clean(text):
    return re.sub(r"^[^a-zA-Z0-9 ]", "", text)
df['Name'] = df['Name'].apply(lambda x: name_clean(x))
```

```
In [12]: df["Ratings"] = df["Ratings"].replace("NEW", np.nan)
df['Ratings'] = df['Ratings'].replace('NaN', np.nan)
df['Ratings'] = df['Ratings'].replace('-', np.nan)
df['Ratings'] = df['Ratings'].replace('nan', np.nan)
def remove_5(value: str):
    if type(value) == str:
        value_new = value.split('/')[0]
        return value_new
    return value
df['Ratings'] = df['Ratings'].apply(remove_5)
df['Ratings'] = df['Ratings'].astype(float)
print(df['Ratings'].dtypes)
```

float64

```
In [13]: def cost(value):
    value = str(value)
    if "," in value:
        value = float(value.replace(",", ""))
        return value
    else:
        return float(value)
df['Cost'] = df['Cost'].apply(cost)
print(df['Cost'].head())
```

```
0    800.0
1    800.0
2    800.0
3    300.0
4    600.0
Name: Cost, dtype: float64
```

```
In [14]: print(df.isnull().sum())
print([features for features in df.columns if df[features].isnull().sum() > 0])
```

```
Name                  0
Takes online orders? 0
Has table booking?   0
Ratings              10003
Votes                0
Rest_Type             227
Cuisines              45
Cost                 344
Type                 0
City                 0
dtype: int64
['Ratings', 'Rest_Type', 'Cuisines', 'Cost']
```

In [15]: `df['Ratings'].unique()`

Out[15]: `array([4.1, 3.8, 3.7, 3.6, 4.6, 4. , 4.2, 3.9, 3.1, 3. , 3.2, 3.3, 2.8, 4.4, 4.3, nan, 2.9, 3.5, 2.6, 3.4, 4.5, 2.5, 2.7, 4.7, 2.4, 2.2, 2.3, 4.8, 4.9, 2.1, 2. , 1.8])`

In [16]: `def handelrate(value):
 if(value=="NEW" or value=="-"):
 return np.nan
 else:
 value=str(value).split("/")
 value=value[0]
 return float(value)
df['Ratings']=df['Ratings'].apply(handelrate)
df['Ratings'].head()`

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

In [17]: `df["Ratings"].fillna(df["Ratings"].mean(),inplace=True)
df["Ratings"].isnull().sum()`

Out[17]: `0`

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

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 51593 entries, 0 to 51716
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   Name             51593 non-null   object 
 1   Takes online orders? 51593 non-null   object 
 2   Has table booking? 51593 non-null   object 
 3   Ratings          51593 non-null   float64
 4   Votes            51593 non-null   int64  
 5   Rest_Type        51366 non-null   object 
 6   Cuisines         51548 non-null   object 
 7   Cost             51249 non-null   float64
 8   Type             51593 non-null   object 
 9   City             51593 non-null   object 
dtypes: float64(2), int64(1), object(7)
memory usage: 4.3+ MB
```

In [19]: df.dropna(inplace=True)
df.head(15)

Out[19]:

	Name	Takes online orders?	Has table booking?	Ratings	Votes	Rest_Type	Cuisines	Cost	Type
0	Jalsa	Yes	Yes	4.1	775	Casual Dining	North Indian, Mughlai, Chinese	800.0	Buffet Banquet
1	Spice Elephant	Yes	No	4.1	787	Casual Dining	Chinese, North Indian, Thai	800.0	Buffet Banquet
2	San Churro Cafe	Yes	No	3.8	918	Cafe, Casual Dining	Cafe, Mexican, Italian	800.0	Buffet Banquet
3	Addhuri Udupi Bhojana	No	No	3.7	88	Quick Bites	South Indian, North Indian	300.0	Buffet Banquet
4	Grand Village	No	No	3.8	166	Casual Dining	North Indian, Rajasthani	600.0	Buffet Banquet
5	Timepass Dinner	Yes	No	3.8	286	Casual Dining	North Indian	600.0	Buffet Banquet
6	Rosewood International Hotel Bar Restaurant	No	No	3.6	8	Casual Dining	North Indian, South Indian, Andhra, Chinese	800.0	Buffet Banquet
7	Onesta	Yes	Yes	4.6	2556	Casual Dining, Cafe	Pizza, Cafe, Italian	600.0	Cafes Banquet
8	Penthouse Cafe	Yes	No	4.0	324	Cafe	Cafe, Italian, Continental	700.0	Cafes Banquet
9	Smacznego	Yes	No	4.2	504	Cafe	Cafe, Mexican, Italian, Momos, Beverages	550.0	Cafes Banquet
10	Caf Down The Alley	Yes	No	4.1	402	Cafe	Cafe	500.0	Cafes Banquet
11	Cafe Shuffle	Yes	Yes	4.2	150	Cafe	Cafe, Italian, Continental	600.0	Cafes Banquet
12	The Coffee Shack	Yes	Yes	4.2	164	Cafe	Cafe, Chinese, Continental, Italian	500.0	Cafes Banquet
13	CafEleven	No	No	4.0	424	Cafe	Cafe, Continental	450.0	Cafes Banquet

	Name	Takes online orders?	Has table booking?	Ratings	Votes	Rest_Type	Cuisines	Cost	Type
14	San Churro Cafe	Yes	No	3.8	918	Cafe, Casual Dining	Cafe, Mexican, Italian	800.0	Cafes

In [20]: df.rename(columns={"approx_cost(for two people)": "Cost2Plates", "listed_in(type)": "Type"}, inplace=True)
df.head(15)

Out[20]:

	Name	Takes online orders?	Has table booking?	Ratings	Votes	Rest_Type	Cuisines	Cost	Type
0	Jalsa	Yes	Yes	4.1	775	Casual Dining	North Indian, Mughlai, Chinese	800.0	Buffet Banquet
1	Spice Elephant	Yes	No	4.1	787	Casual Dining	Chinese, North Indian, Thai	800.0	Buffet Banquet
2	San Churro Cafe	Yes	No	3.8	918	Cafe, Casual Dining	Cafe, Mexican, Italian	800.0	Buffet Banquet
3	Addhuri Udupi Bhojana	No	No	3.7	88	Quick Bites	South Indian, North Indian	300.0	Buffet Banquet
4	Grand Village	No	No	3.8	166	Casual Dining	North Indian, Rajasthani	600.0	Buffet Banquet
5	Timepass Dinner	Yes	No	3.8	286	Casual Dining	North Indian	600.0	Buffet Banquet
6	Rosewood International Hotel Bar Restaurant	No	No	3.6	8	Casual Dining	North Indian, South Indian, Andhra, Chinese	800.0	Buffet Banquet
7	Onesta	Yes	Yes	4.6	2556	Casual Dining, Cafe	Pizza, Cafe, Italian	600.0	Cafes Banquet
8	Penthouse Cafe	Yes	No	4.0	324	Cafe	Cafe, Italian, Continental	700.0	Cafes Banquet
9	Smacznego	Yes	No	4.2	504	Cafe	Cafe, Mexican, Italian, Momos, Beverages	550.0	Cafes Banquet
10	Caf Down The Alley	Yes	No	4.1	402	Cafe	Cafe	500.0	Cafes Banquet
11	Cafe Shuffle	Yes	Yes	4.2	150	Cafe	Cafe, Italian, Continental	600.0	Cafes Banquet
12	The Coffee Shack	Yes	Yes	4.2	164	Cafe	Cafe, Chinese, Continental, Italian	500.0	Cafes Banquet
13	CafEleven	No	No	4.0	424	Cafe	Cafe, Continental	450.0	Cafes Banquet

	Name	Takes online orders?	Has table booking?	Ratings	Votes	Rest_Type	Cuisines	Cost	Type
14	San Churro Cafe	Yes	No	3.8	918	Cafe, Casual Dining	Cafe, Mexican, Italian	800.0	Cafes

In [21]: `df.describe()`

Out[21]:

	Ratings	Votes	Cost
count	51026.000000	51026.000000	51026.000000
mean	3.701425	284.596911	555.872986
std	0.395259	806.038077	439.332026
min	1.800000	0.000000	40.000000
25%	3.500000	7.000000	300.000000
50%	3.700142	41.000000	400.000000
75%	3.900000	199.000000	700.000000
max	4.900000	16832.000000	6000.000000

In [22]: `df.columns.value_counts()`

Out[22]:

Name	1
Takes online orders?	1
Has table booking?	1
Ratings	1
Votes	1
Rest_Type	1
Cuisines	1
Cost	1
Type	1
City	1
dtype: int64	

In [23]: `df["Cost"].unique()`

Out[23]:

```
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 [24]: df["City"].unique()
```

```
Out[24]: 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)
```

```
In [25]: def handlecomma(value):  
    value=str(value)  
    if "," in value:  
        value=value.replace(",","",)  
        return float(value)  
    else:  
        return float(value)  
df["Cost2"]=df["Cost"].apply(handlecomma)  
df["Cost2"].unique()
```

```
Out[25]: 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 [26]: df.head(15)

Out[26]:

	Name	Takes online orders?	Has table booking?	Ratings	Votes	Rest_Type	Cuisines	Cost	Type
0	Jalsa	Yes	Yes	4.1	775	Casual Dining	North Indian, Mughlai, Chinese	800.0	Buffet Banasha
1	Spice Elephant	Yes	No	4.1	787	Casual Dining	Chinese, North Indian, Thai	800.0	Buffet Banasha
2	San Churro Cafe	Yes	No	3.8	918	Cafe, Casual Dining	Cafe, Mexican, Italian	800.0	Buffet Banasha
3	Addhuri Udupi Bhojana	No	No	3.7	88	Quick Bites	South Indian, North Indian	300.0	Buffet Banasha
4	Grand Village	No	No	3.8	166	Casual Dining	North Indian, Rajasthani	600.0	Buffet Banasha
5	Timepass Dinner	Yes	No	3.8	286	Casual Dining	North Indian	600.0	Buffet Banasha
6	Rosewood International Hotel Bar Restaurant	No	No	3.6	8	Casual Dining	North Indian, South Indian, Andhra, Chinese	800.0	Buffet Banasha
7	Onesta	Yes	Yes	4.6	2556	Casual Dining, Cafe	Pizza, Cafe, Italian	600.0	Cafes Banasha
8	Penthouse Cafe	Yes	No	4.0	324	Cafe	Cafe, Italian, Continental	700.0	Cafes Banasha
9	Smaczego	Yes	No	4.2	504	Cafe	Cafe, Mexican, Italian, Momos, Beverages	550.0	Cafes Banasha
10	Caf Down The Alley	Yes	No	4.1	402	Cafe	Cafe	500.0	Cafes Banasha
11	Cafe Shuffle	Yes	Yes	4.2	150	Cafe	Cafe, Italian, Continental	600.0	Cafes Banasha
12	The Coffee Shack	Yes	Yes	4.2	164	Cafe	Cafe, Chinese, Continental, Italian	500.0	Cafes Banasha
13	CafEleven	No	No	4.0	424	Cafe	Cafe, Continental	450.0	Cafes Banasha
14	San Churro Cafe	Yes	No	3.8	918	Cafe, Casual Dining	Cafe, Mexican, Italian	800.0	Cafes Banasha

```
In [27]: df["Rest_Type"].value_counts()
```

```
Out[27]: Quick Bites          19008
Casual Dining           10253
Cafe                   3682
Delivery               2574
Dessert Parlor          2229
...
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 [28]: rest_type=df["Rest_Type"].value_counts(ascending=False)
rest_type.head(16)
```

```
Out[28]: Quick Bites          19008
Casual Dining           10253
Cafe                   3682
Delivery               2574
Dessert Parlor          2229
Takeaway, Delivery       2008
Bakery                 1140
Casual Dining, Bar        1130
Beverage Shop            863
Bar                     686
Food Court              616
Sweet Shop               468
Bar, Casual Dining        411
Lounge                  395
Pub                      355
Fine Dining              345
Name: Rest_Type, dtype: int64
```

```
In [29]: rest_types_lessthan1000=rest_type[rest_type<1000]  
rest_types_lessthan1000
```

```
Out[29]: 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 [30]: 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[30]: Quick Bites      19008  
Casual Dining    10253  
others          9002  
Cafe            3682  
Delivery        2574  
Dessert Parlor   2229  
Takeaway, Delivery 2008  
Bakery          1140  
Casual Dining, Bar 1130  
Name: Rest_Type, dtype: int64
```

In [31]: df.head(16)

Out[31]:

	Name	Takes online orders?	Has table booking?	Ratings	Votes	Rest_Type	Cuisines	Cost	Type
0	Jalsa	Yes	Yes	4.1	775	Casual Dining	North Indian, Mughlai, Chinese	800.0	Buffet Banasha
1	Spice Elephant	Yes	No	4.1	787	Casual Dining	Chinese, North Indian, Thai	800.0	Buffet Banasha
2	San Churro Cafe	Yes	No	3.8	918	others	Cafe, Mexican, Italian	800.0	Buffet Banasha
3	Addhuri Udupi Bhojana	No	No	3.7	88	Quick Bites	South Indian, North Indian	300.0	Buffet Banasha
4	Grand Village	No	No	3.8	166	Casual Dining	North Indian, Rajasthani	600.0	Buffet Banasha
5	Timepass Dinner	Yes	No	3.8	286	Casual Dining	North Indian	600.0	Buffet Banasha
6	Rosewood International Hotel Bar Restaurant	No	No	3.6	8	Casual Dining	North Indian, South Indian, Andhra, Chinese	800.0	Buffet Banasha
7	Onesta	Yes	Yes	4.6	2556	others	Pizza, Cafe, Italian	600.0	Cafes Banasha
8	Penthouse Cafe	Yes	No	4.0	324	Cafe	Cafe, Italian, Continental	700.0	Cafes Banasha
9	Smaczego	Yes	No	4.2	504	Cafe	Cafe, Mexican, Italian, Momos, Beverages	550.0	Cafes Banasha
10	Caf Down The Alley	Yes	No	4.1	402	Cafe	Cafe	500.0	Cafes Banasha
11	Cafe Shuffle	Yes	Yes	4.2	150	Cafe	Cafe, Italian, Continental	600.0	Cafes Banasha
12	The Coffee Shack	Yes	Yes	4.2	164	Cafe	Cafe, Chinese, Continental, Italian	500.0	Cafes Banasha
13	CafEleven	No	No	4.0	424	Cafe	Cafe, Continental	450.0	Cafes Banasha
14	San Churro Cafe	Yes	No	3.8	918	others	Cafe, Mexican, Italian	800.0	Cafes Banasha

	Name	Takes online orders?	Has table booking?	Ratings	Votes	Rest_Type	Cuisines	Cost	Type	
15	Cafe Vivacity	Yes	No	3.8	90	Cafe	Cafe	650.0	Cafes	Banasha



```
In [32]: 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()
df.head(16)
```

Out[32]:

	Name	Takes online orders?	Has table booking?	Ratings	Votes	Rest_Type	Cuisines	Cost	Type	
0	Jalsa	Yes	Yes	4.1	775	Casual Dining	North Indian, Muglai, Chinese	800.0	Buffet	Banashan
1	Spice Elephant	Yes	No	4.1	787	Casual Dining	others	800.0	Buffet	Banashan
2	San Churro Cafe	Yes	No	3.8	918	others	others	800.0	Buffet	Banashan
3	Addhuri Udupi Bhojana	No	No	3.7	88	Quick Bites	South Indian, North Indian	300.0	Buffet	Banashan
4	Grand Village	No	No	3.8	166	Casual Dining	others	600.0	Buffet	Banashan
5	Timepass Dinner	Yes	No	3.8	286	Casual Dining	North Indian	600.0	Buffet	Banashan
6	Rosewood International Hotel Bar Restaurant	No	No	3.6	8	Casual Dining	others	800.0	Buffet	Banashan
7	Onesta	Yes	Yes	4.6	2556	others	others	600.0	Cafes	Banashan
8	Penthouse Cafe	Yes	No	4.0	324	Cafe	others	700.0	Cafes	Banashan
9	Smaczego	Yes	No	4.2	504	Cafe	others	550.0	Cafes	Banashan
10	Caf Down The Alley	Yes	No	4.1	402	Cafe	Cafe	500.0	Cafes	Banashan
11	Cafe Shuffle	Yes	Yes	4.2	150	Cafe	others	600.0	Cafes	Banashan
12	The Coffee Shack	Yes	Yes	4.2	164	Cafe	others	500.0	Cafes	Banashan
13	CafEleven	No	No	4.0	424	Cafe	Cafe, Continental	450.0	Cafes	Banashan
14	San Churro Cafe	Yes	No	3.8	918	others	others	800.0	Cafes	Banashan
15	Cafe Vivacity	Yes	No	3.8	90	Cafe	Cafe	650.0	Cafes	Banashan

```
In [33]: df["Type"].value_counts()
```

```
Out[33]: Delivery           25569  
Dine-out            17560  
Desserts             3555  
Cafes                 1703  
Drinks & nightlife    1084  
Buffet                  869  
Pubs and bars          686  
Name: Type, dtype: int64
```

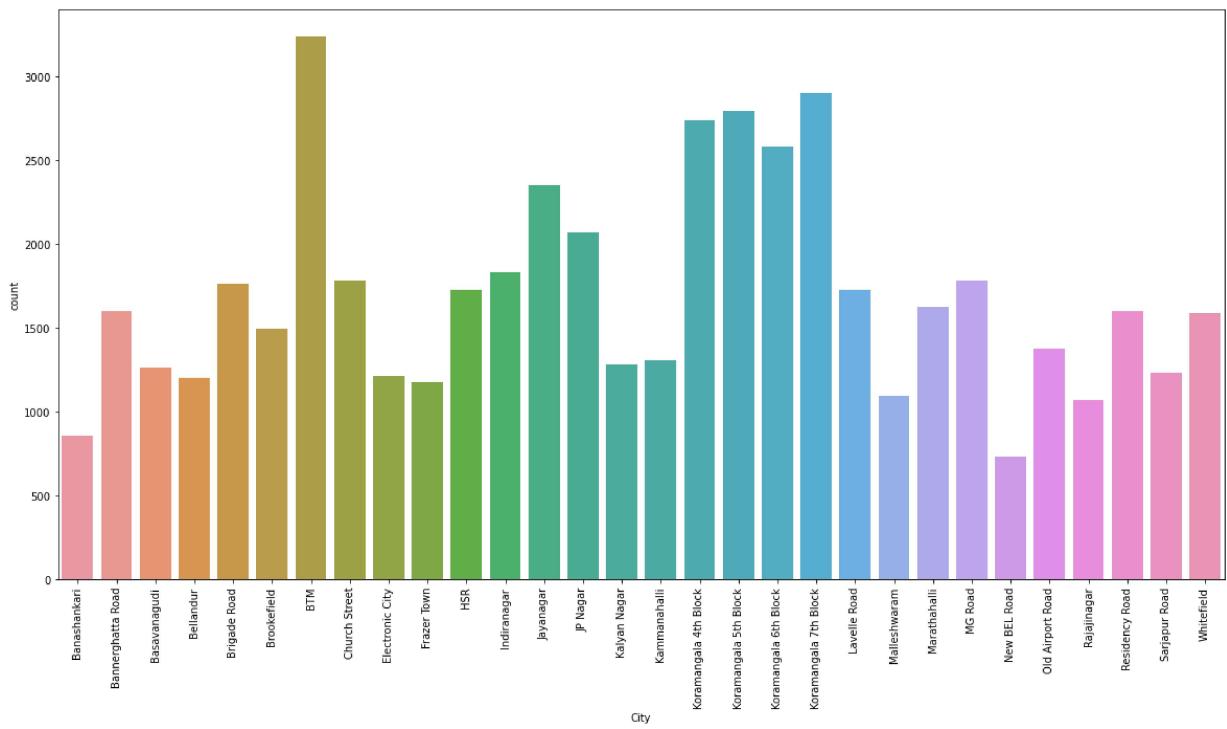
```
In [34]: df["City"].value_counts()
```

```
Out[34]: BTM                  3238  
Koramangala 7th Block      2900  
Koramangala 5th Block      2797  
Koramangala 4th Block      2740  
Koramangala 6th Block      2586  
Jayanagar                2350  
JP Nagar                  2070  
Indiranagar                1834  
Church Street                1785  
MG Road                    1780  
Brigade Road                1764  
HSR                        1727  
Lavelle Road                1726  
Marathahalli                1629  
Bannerghatta Road            1600  
Residency Road                1600  
Whitefield                  1589  
Brookefield                  1495  
Old Airport Road                1379  
Kammanahalli                1305  
Kalyan Nagar                  1285  
Basavanagudi                  1264  
Sarjapur Road                1234  
Electronic City                1214  
Bellandur                      1203  
Frazer Town                  1178  
Malleshwaram                  1092  
Rajajinagar                  1068  
Banasankari                  859  
New BEL Road                  735  
Name: City, dtype: int64
```

```
In [38]: plt.figure(figsize=(20,10))
graph=sns.countplot(df["City"]);
plt.xticks(rotation=90);
```

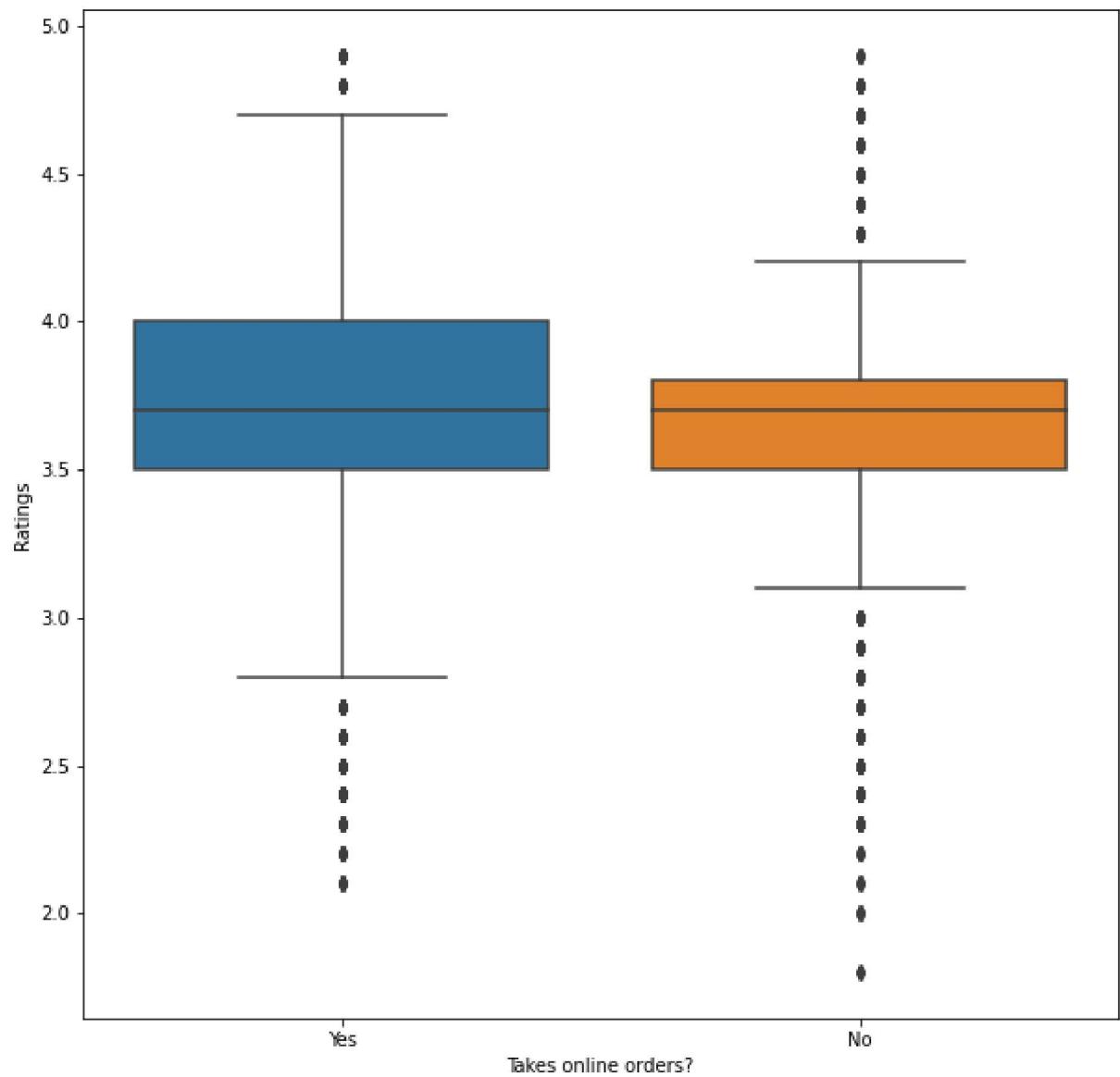
C:\ProgramData\Anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```



```
In [39]: plt.figure(figsize=(10,10))
sns.boxplot(x="Takes online orders?",y="Ratings",data=df)
```

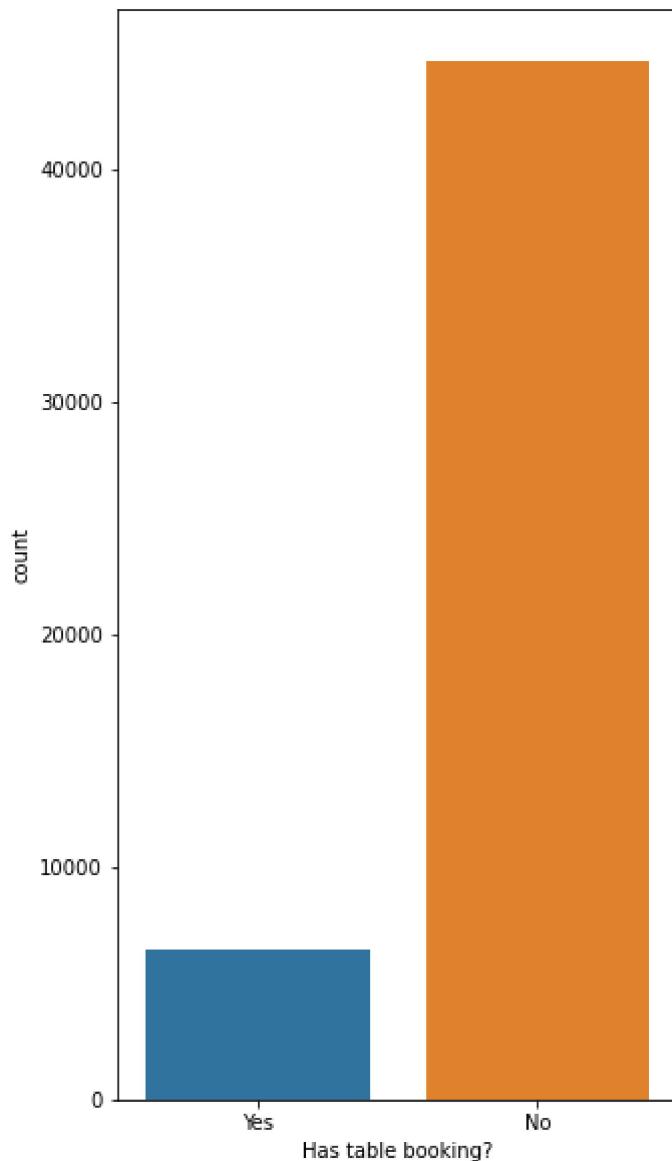
```
Out[39]: <AxesSubplot:xlabel='Takes online orders?', ylabel='Ratings'>
```



```
In [40]: plt.figure(figsize=(5,10))
sns.countplot(df["Has table booking?"])
```

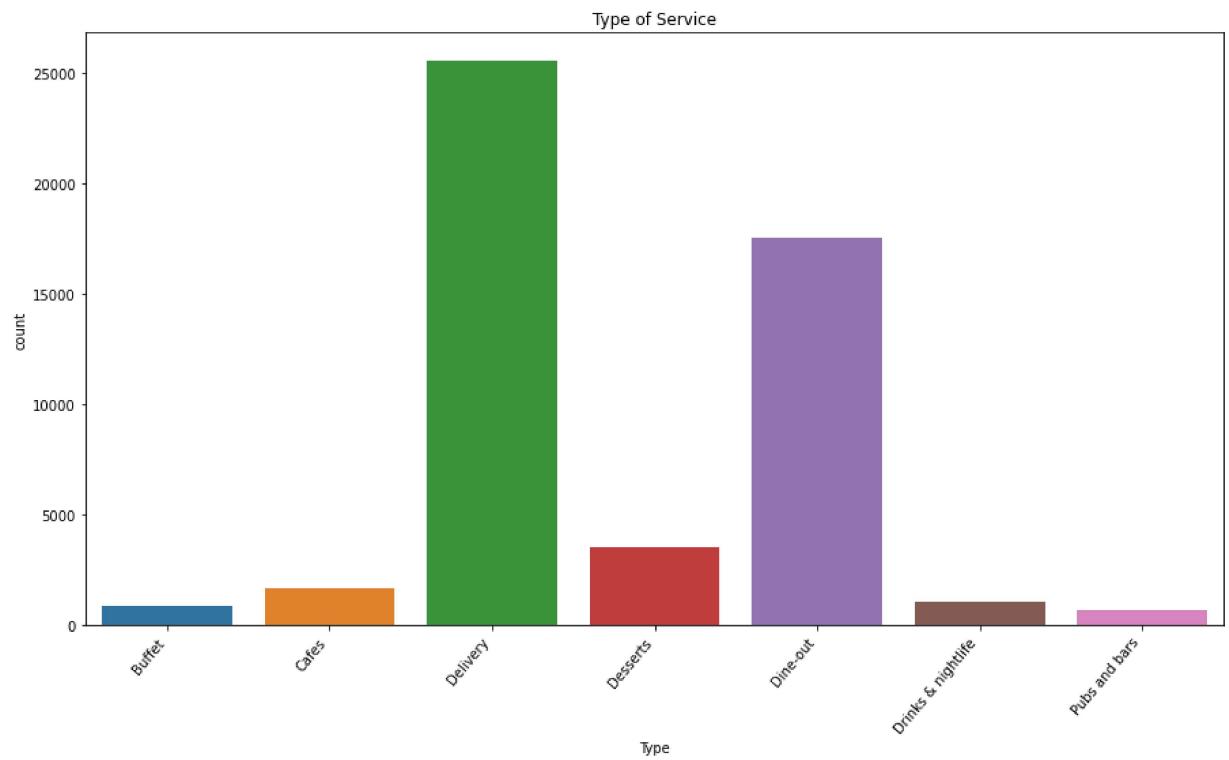
C:\ProgramData\Anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
warnings.warn(

```
Out[40]: <AxesSubplot:xlabel='Has table booking?', ylabel='count'>
```



```
In [52]: plt.figure(figsize=[15,8])
sns.countplot (x = df['Type'])
sns.countplot (x = df['Type']).set_xticklabels(sns.countplot(x = df['Type']).get_
plt.title('Type of Service')
```

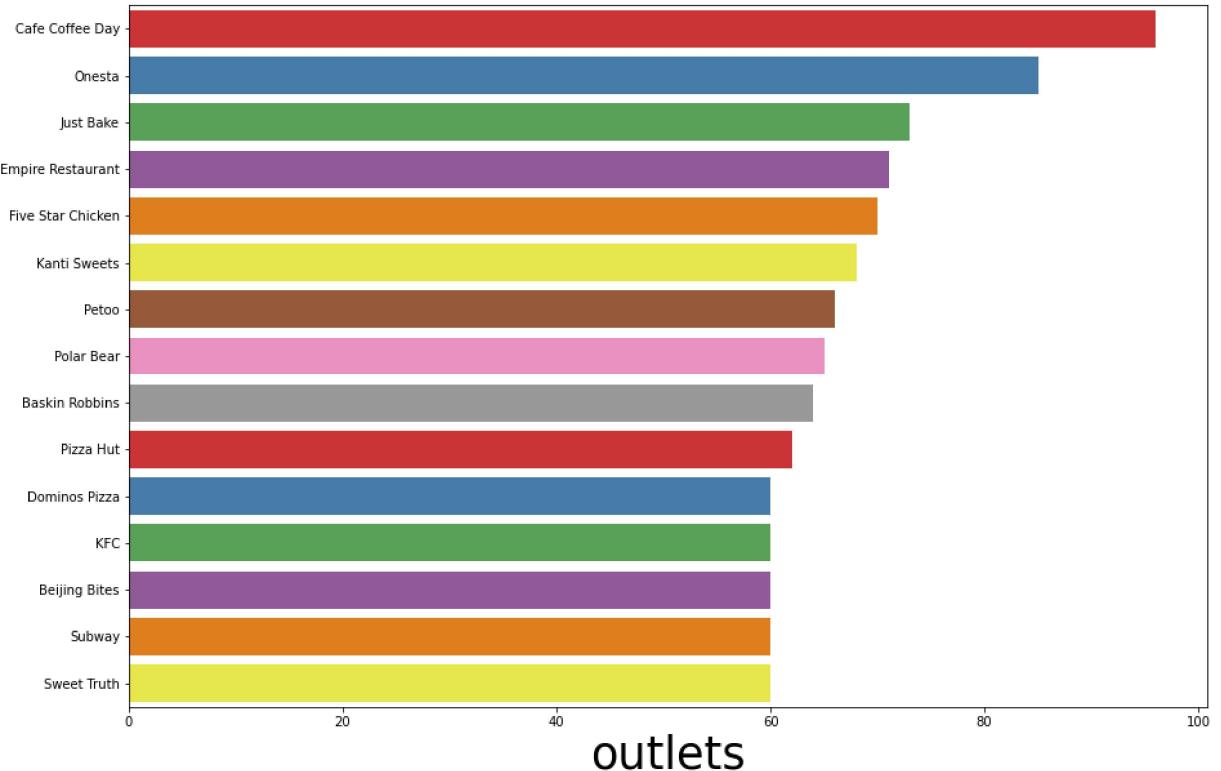
Out[52]: Text(0.5, 1.0, 'Type of Service')



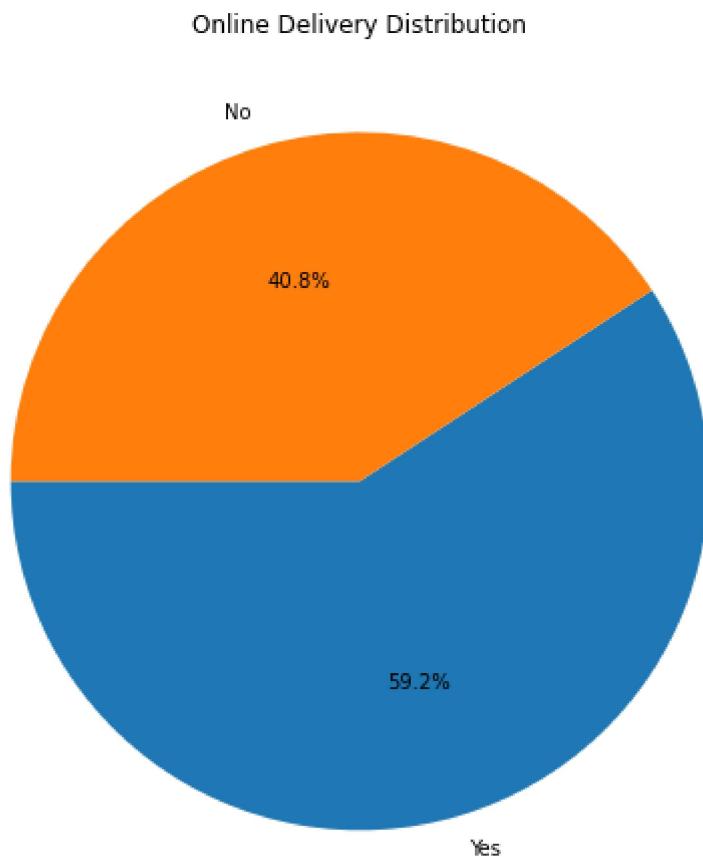
```
In [74]: plt.figure(figsize=[15,10])
chains=df['Name'].value_counts()[:15]
sns.barplot(x=chains,y=chains.index,palette='Set1')
plt.title('Famous Restaurant in Bangalore',size = 35 ,pad = 35)
plt.xlabel('outlets',size=35)
```

Out[74]: Text(0.5, 0, 'outlets')

Famous Restaurant in Bangalore



```
In [66]: plt.figure(figsize=(15,8))
plt.title('Online Delivery Distribution')
plt.pie(df['Takes online orders?'].value_counts()/9551*100, labels=df['Takes online orders?'])
```



```
In [80]: plt.figure(figsize=(12,6))
sns.scatterplot(x="Ratings", y="Rest_Type", hue='Cost2', data=df)

plt.xlabel("Ratings")
plt.ylabel("Rest_Type")
plt.title('Ratings vs Rest Type');
```



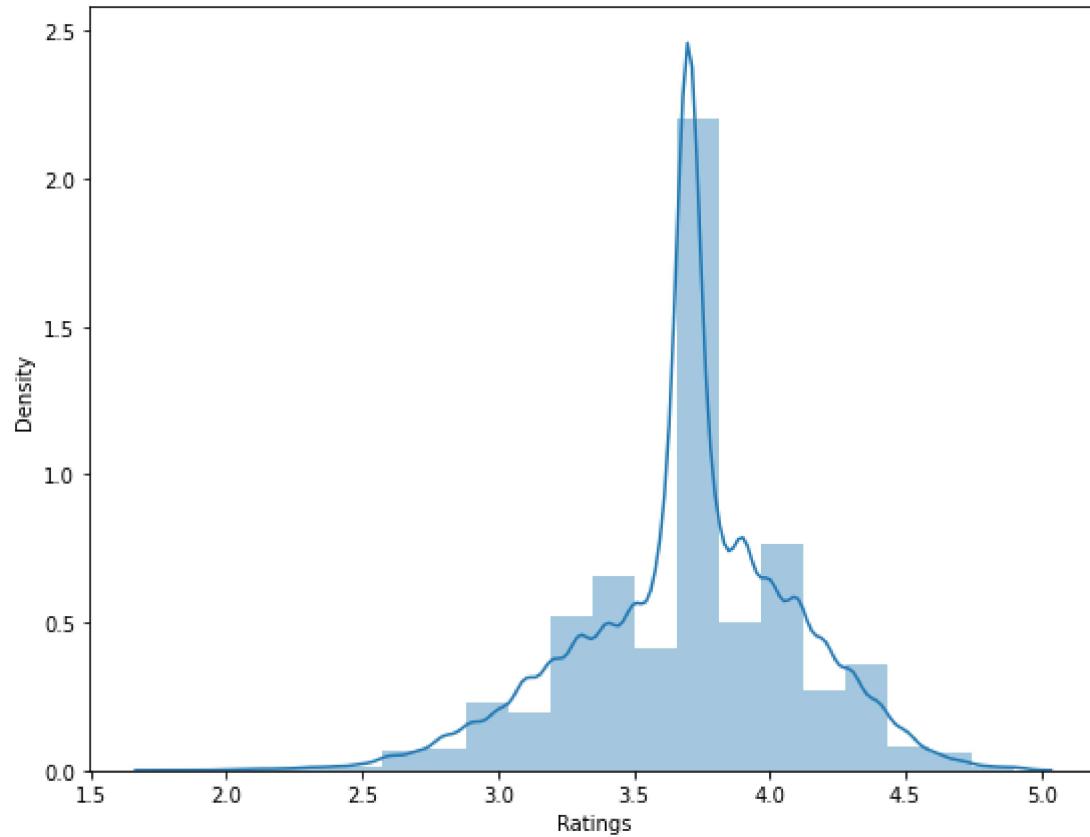
```
In [90]: plt.figure(figsize=(9,7))
```

```
sns.distplot(df['Ratings'],bins=20)
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

```
Out[90]: <AxesSubplot:xlabel='Ratings', ylabel='Density'>
```

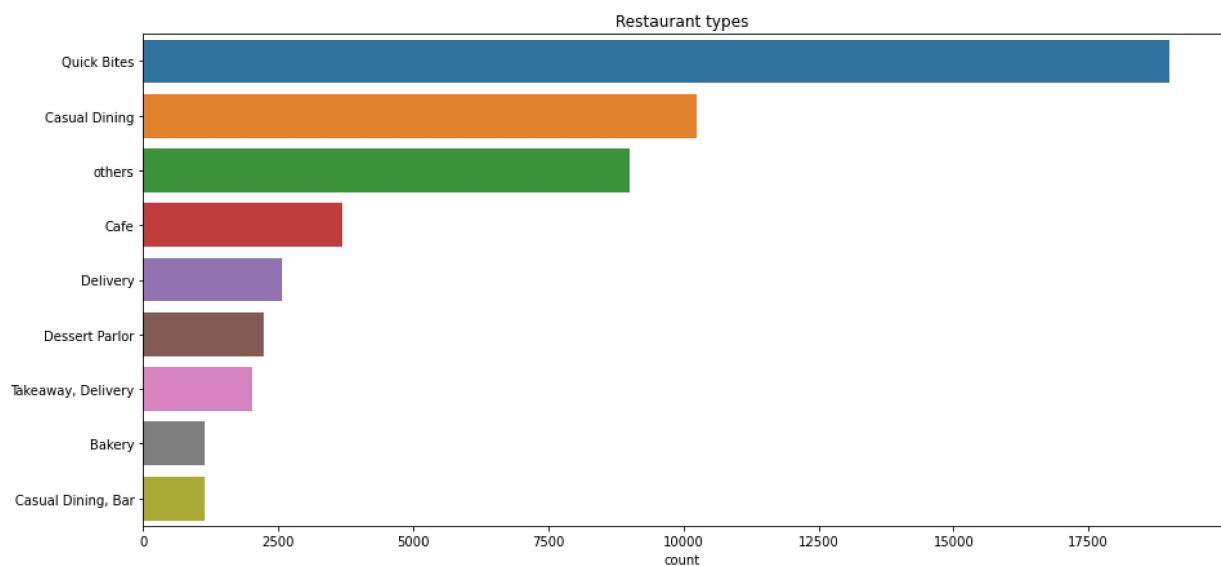


```
In [92]: plt.figure(figsize=(15,7))
rest=df['Rest_Type'].value_counts()[:20]
sns.barplot(rest,rest.index)
plt.title("Restaurant types")
plt.xlabel("count")
```

C:\ProgramData\Anaconda3\lib\site-packages\seaborn_decorators.py:36: FutureWarning:

Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without a n explicit keyword will result in an error or misinterpretation.

Out[92]: Text(0.5, 0, 'count')



In []: