In [12]:

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

In [13]:

```
df = pd.read_excel(r'C:\Users\NITU\Downloads\zomato.xlsx')
```

In [14]:

```
df.head()
```

Out[14]:

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

In [15]:

```
df.shape
```

Out[15]:

```
(51717, 17)
```

In [16]:

```
df.columns
```

Out[16]:

```
Index(['url', 'address', 'name', 'online_order', 'book_table', 'rate', 'vote
s',
       '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 [17]:

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

Out[17]:

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

In [18]:

```
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  float64
 9   listed_in(type)              51717 non-null  object 
 10  listed_in(city)              51717 non-null  object 
dtypes: float64(1), int64(1), object(9)
memory usage: 4.3+ MB
```

In [19]:

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

Out[19]:

```
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 [20]:

```
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)
```

In [21]:

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

Out[21]:

```
0    4.1/5
1    4.1/5
2    3.8/5
3    3.7/5
4    3.8/5
Name: rate, dtype: object
```

In [22]:

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

Out[22]:

```
7775
```

In [23]:

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

Out[23]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	cost2pl
0	Jalsa	Yes	Yes	4.1/5	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	8
1	Spice Elephant	Yes	No	4.1/5	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	8
2	San Churro Cafe	Yes	No	3.8/5	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian	8
3	Addhuri Udupi Bhojana	No	No	3.7/5	88	Banashankari	Quick Bites	South Indian, North Indian	3
4	Grand Village	No	No	3.8/5	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	6

In [24]:

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

Out[24]:

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

In [25]:

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

Out[25]:

```
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 [26]:

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

In [27]:

```
df.head()
```

Out[27]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	cost2pl
0	Jalsa	Yes	Yes	4.1/5	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	8
1	Spice Elephant	Yes	No	4.1/5	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	8
2	San Churro Cafe	Yes	No	3.8/5	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian	8
3	Addhuri Udupi Bhojana	No	No	3.7/5	88	Banashankari	Quick Bites	South Indian, North Indian	3
4	Grand Village	No	No	3.8/5	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	6

In [28]:

```
df['cost2plates'].unique()
```

Out[28]:

```
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.,   nan, 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 [29]:

```
df.head()
```

Out[29]:

	name	online_order	book_table	rate	votes	location	rest_type	cuisines	cost2pl
0	Jalsa	Yes	Yes	4.1/5	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	8
1	Spice Elephant	Yes	No	4.1/5	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	8
2	San Churro Cafe	Yes	No	3.8/5	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian	8
3	Addhuri Udupi Bhojana	No	No	3.7/5	88	Banashankari	Quick Bites	South Indian, North Indian	3
4	Grand Village	No	No	3.8/5	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	6

In [30]:

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

Out[30]:

```
Quick Bites                    19132
Casual Dining                  10330
Cafe                            3732
Delivery                        2604
Dessert Parlor                  2263
                               ...  
Dessert Parlor, Kiosk              2
Food Court, Beverage Shop          2
Dessert Parlor, Food Court         2
Sweet Shop, Dessert Parlor         1
Quick Bites, Kiosk                 1
Name: rest_type, Length: 93, dtype: int64
```

In [31]:

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

In [32]:

```
rest_types
```

Out[32]:

```
Quick Bites                   19132
Casual Dining                 10330
Cafe                           3732
Delivery                       2604
Dessert Parlor                 2263
                              ...  
Dessert Parlor, Kiosk             2
Food Court, Beverage Shop         2
Dessert Parlor, Food Court        2
Sweet Shop, Dessert Parlor        1
Quick Bites, Kiosk                1
Name: rest_type, Length: 93, dtype: int64
```

In [33]:

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

Out[33]:

```
Beverage Shop                 867
Bar                           697
Food Court                    624
Sweet Shop                    468
Bar, Casual Dining            425
                             ... 
Dessert Parlor, Kiosk           2
Food Court, Beverage Shop       2
Dessert Parlor, Food Court      2
Sweet Shop, Dessert Parlor      1
Quick Bites, Kiosk              1
Name: rest_type, Length: 85, dtype: int64
```

In [34]:

```
 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()
```

In [35]:

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

Out[35]:

```
Quick Bites                      19132
Casual Dining                    10330
Cafe                              3732
Delivery                          2604
Dessert Parlor                    2263
                                 ...  
Dessert Parlor, Kiosk                2
Food Court, Beverage Shop            2
Dessert Parlor, Food Court           2
Sweet Shop, Dessert Parlor           1
Quick Bites, Kiosk                   1
Name: rest_type, Length: 93, dtype: int64
```

In [36]:

```
df['rest_type'].isna().sum()
```

Out[36]:

```
227
```

In [37]:

```
df['rest_type'].dropna(inplace =True)
```

In [38]:

```
df['rest_type'].isna().sum()
```

Out[38]:

```
227
```

In [39]:

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

Out[39]:

```
94
```

In [40]:

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

Out[40]:

```
location
BTM                    [Sankranthi Veg Restaurant, Hearts Unlock Cafe...
Banashankari           [Jalsa, Spice Elephant, San Churro Cafe, Addhu...
Banaswadi              [Cafe Nibras, The Sanctuary, Crunch Pizzas, 9 ...
Bannerghatta Road      [Deja Vu Resto Bar, Fattoush, Empire Restauran...
Basavanagudi           [Grand Village, Timepass Dinner, Srinathji's C...
                                             ...
West Bangalore         [FreshMenu, Fit Dish Fetish, Garden City Mobil...
Whitefield             [Imperio Cafe, Night Diaries, LocalHost, AB's ...
Wilson Garden          [Tree Top, Sahana's (Nati Style), Karavali Kol...
Yelahanka              [Prashanth Naati Corner, Red Chillies Curries ...
Yeshwantpur            [Chef's Bank, New Agarwal Bhavan, Fishing Boat...
Name: name, Length: 93, dtype: object
```

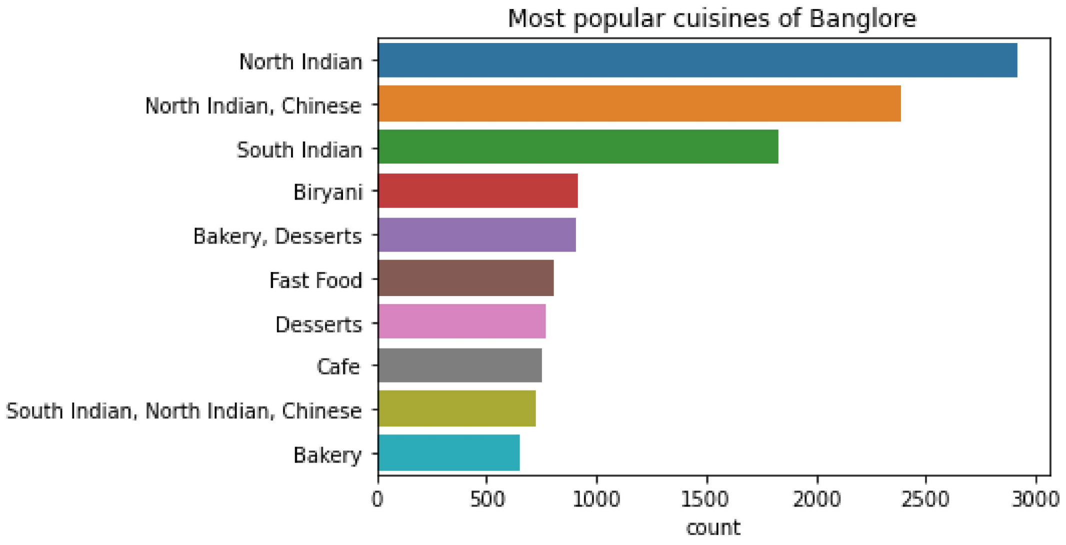
In [41]:

```
cuisines=df['cuisines'].value_counts()[:10]
sns.barplot(cuisines,cuisines.index)
plt.xlabel('count')
plt.title("Most popular cuisines of Banglore")
```

```
C:\Users\NITU\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureW
arning: Pass the following variables as keyword args: x, y. From version 0.1
2, the only valid positional argument will be `data`, and passing other argu
ments without an explicit keyword will result in an error or misinterpretati
on.
  warnings.warn(
```

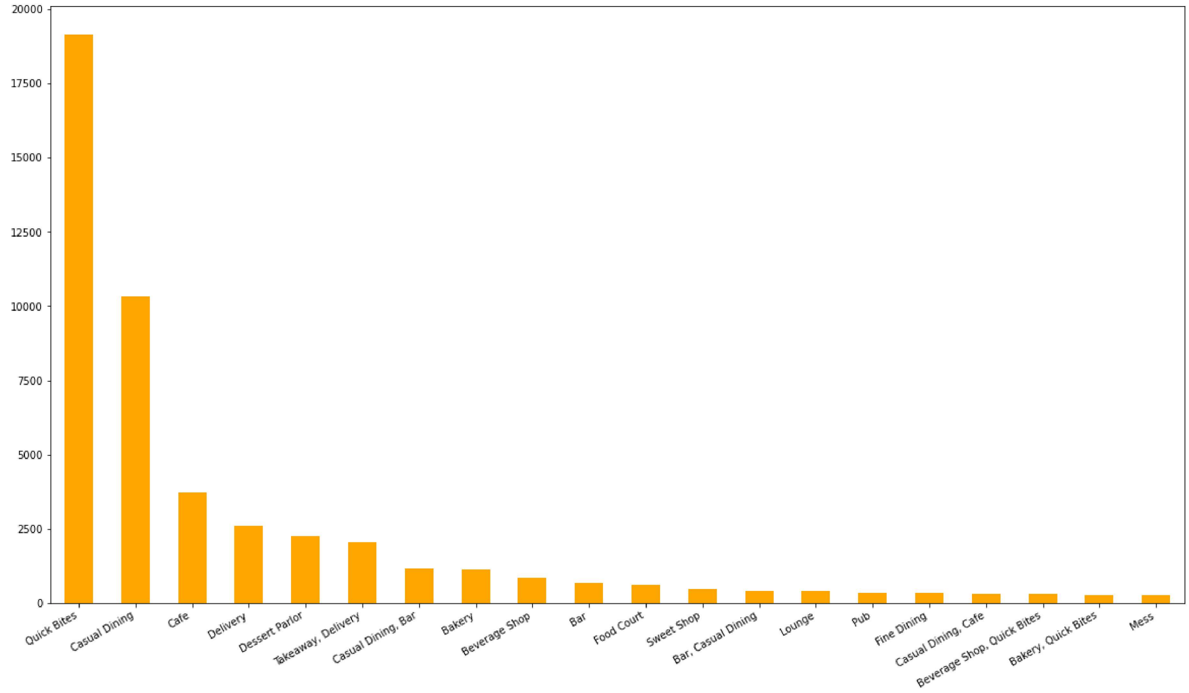
Out[41]:

```
Text(0.5, 1.0, 'Most popular cuisines of Banglore')
```



In [42]:

```
plt.figure(figsize=(20,12))
df['rest_type'].value_counts().nlargest(20).plot.bar(color='orange')
plt.gcf().autofmt_xdate()
```



In [43]:

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

```
  Text(90, 0, 'Yelahanka'),
  Text(91, 0, 'Sahakara Nagar'),
  Text(92, 0, 'Peenya')])
```

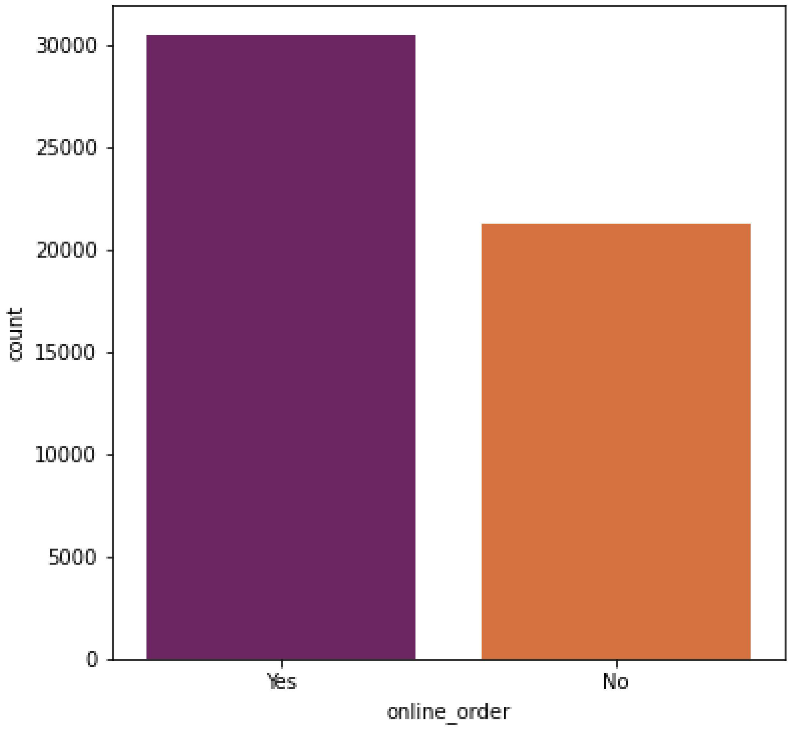
In [44]:

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

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

Out[44]:

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



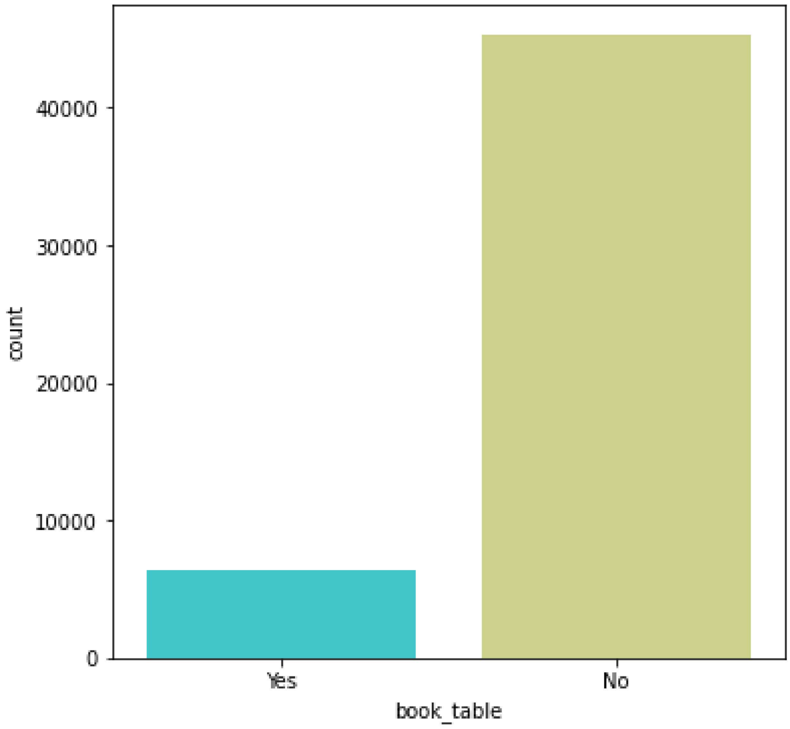
In [45]:

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

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

Out[45]:

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

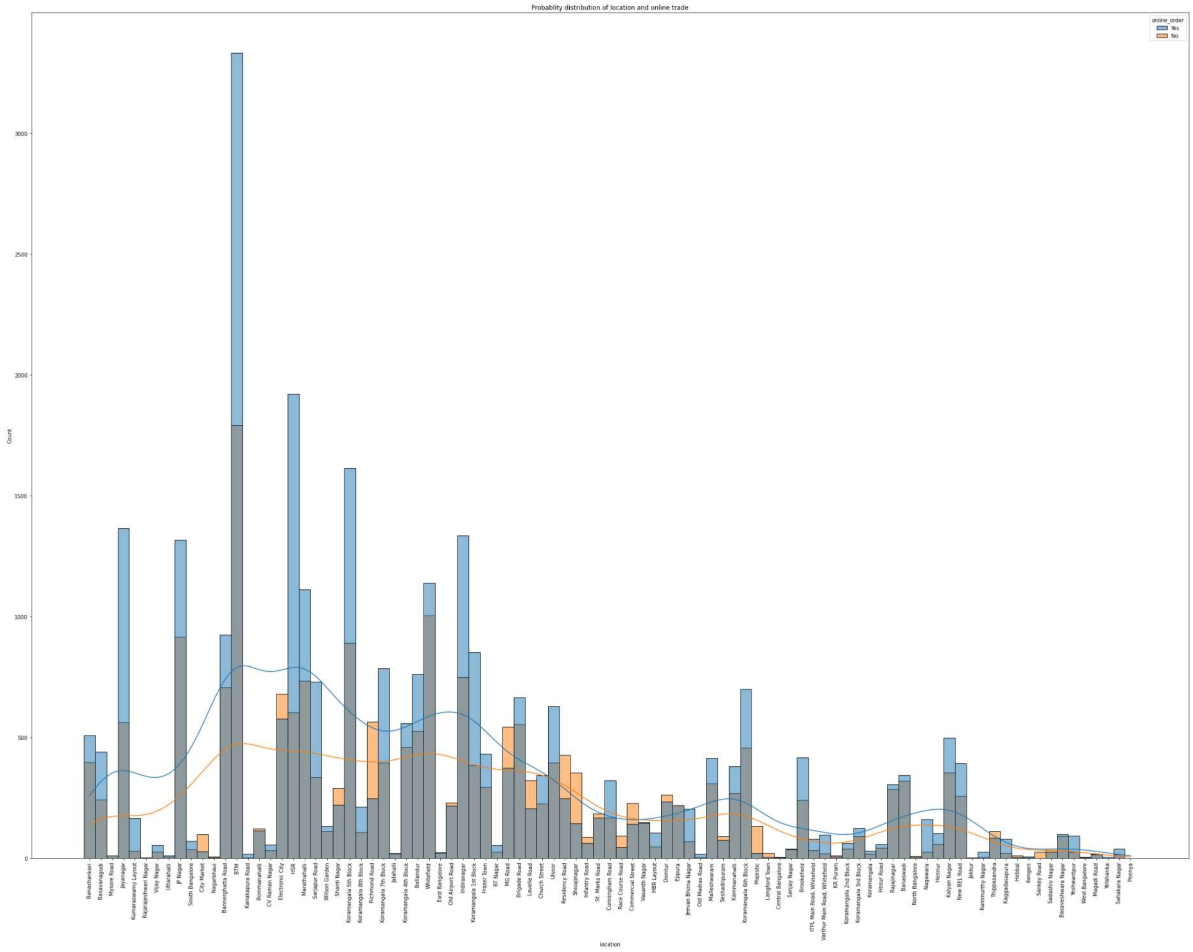


In [46]:

```
plt.figure(figsize=(40,30))
sns.histplot(df,x='location',hue='online_order',kde=True,bins=3)
plt.xticks(rotation=89)
plt.title('Probablity distribution of location and online trade')
```

Out[46]:

```
Text(0.5, 1.0, 'Probablity distribution of location and online trade')
```



In [47]:

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

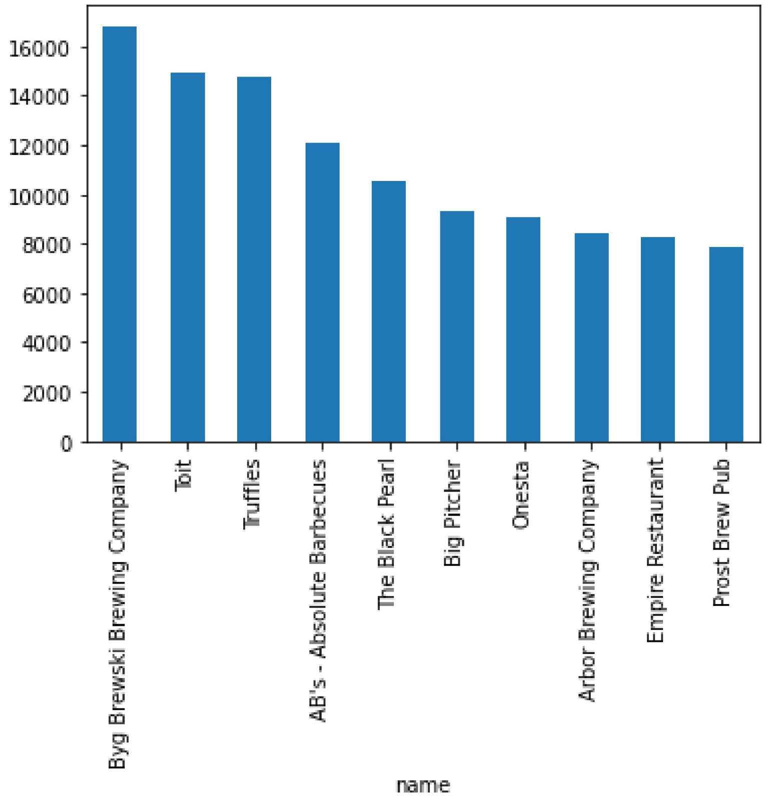
```
  Text(56, 0, 'Seshadripuram'),
  Text(57, 0, 'Kammanahalli'),
  Text(58, 0, 'Koramangala 6th Block'),
  Text(59, 0, 'Majestic'),
  Text(60, 0, 'Langford Town'),
  Text(61, 0, 'Central Bangalore'),
  Text(62, 0, 'Sanjay Nagar'),
  Text(63, 0, 'Brookefield'),
  Text(64, 0, 'ITPL Main Road, Whitefield'),
  Text(65, 0, 'Varthur Main Road, Whitefield'),
  Text(66, 0, 'KR Puram'),
  Text(67, 0, 'Koramangala 2nd Block'),
  Text(68, 0, 'Koramangala 3rd Block'),
  Text(69, 0, 'Koramangala'),
  Text(70, 0, 'Hosur Road'),
  Text(71, 0, 'Rajajinagar'),
  Text(72, 0, 'Banaswadi'),
  Text(73, 0, 'North Bangalore'),
  Text(74, 0, 'Nagawara'),
  Text(75, 0, 'Hennur'),
```

In [48]:

```
df.groupby('name')['votes'].max().nlargest(10).plot.bar()
```

Out[48]:

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

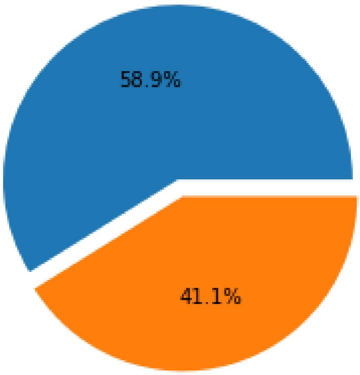


In [49]:

```
x = df['online_order'].value_counts()
labels =['accepted','not accepted']
plt.pie(x,explode=[0.0,0.1],autopct='%1.1f%%')
```

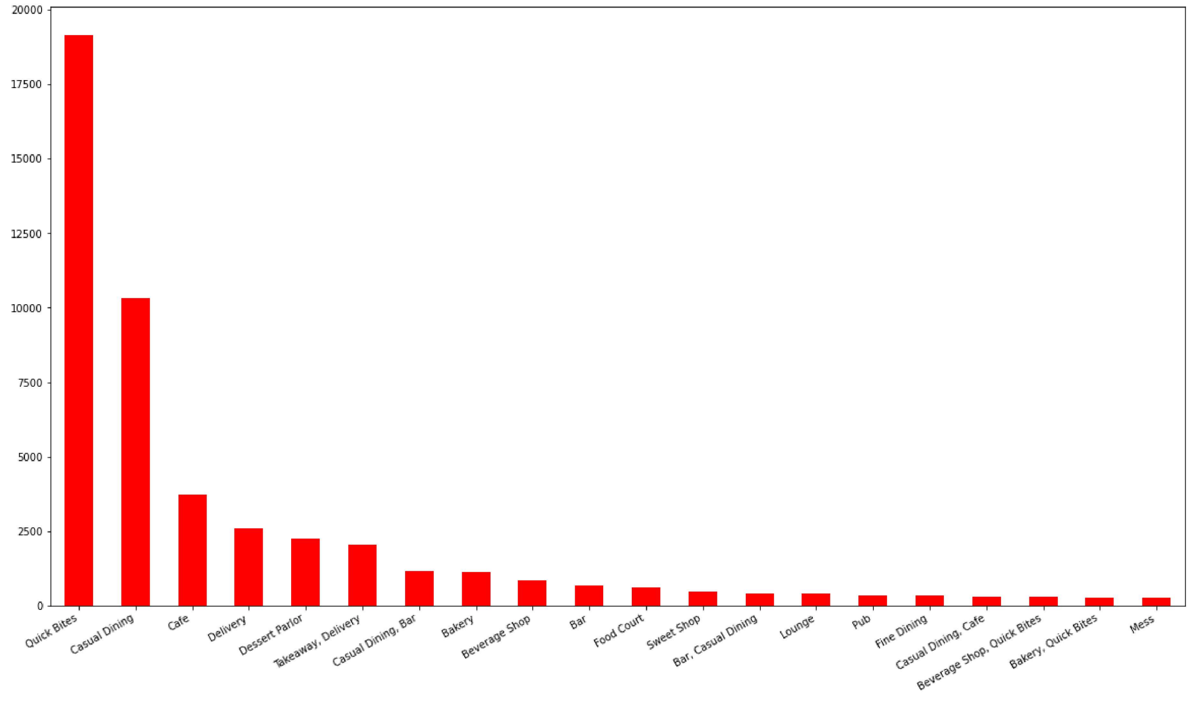
Out[49]:

```
([<matplotlib.patches.Wedge at 0x220bb5edfa0>,
  <matplotlib.patches.Wedge at 0x220bb5f8700>],
 [Text(-0.30245810454915784, 1.0576006311422714, ''),
  Text(0.3299542958718086, -1.1537461430642961, '')],
 [Text(-0.16497714793590426, 0.5768730715321481, '58.9%'),
  Text(0.192473339258555, -0.6730185834541726, '41.1%')])
```



In [50]:

```
plt.figure(figsize=(20,12))
df['rest_type'].value_counts().nlargest(20).plot.bar(color='red')
plt.gcf().autofmt_xdate()
```

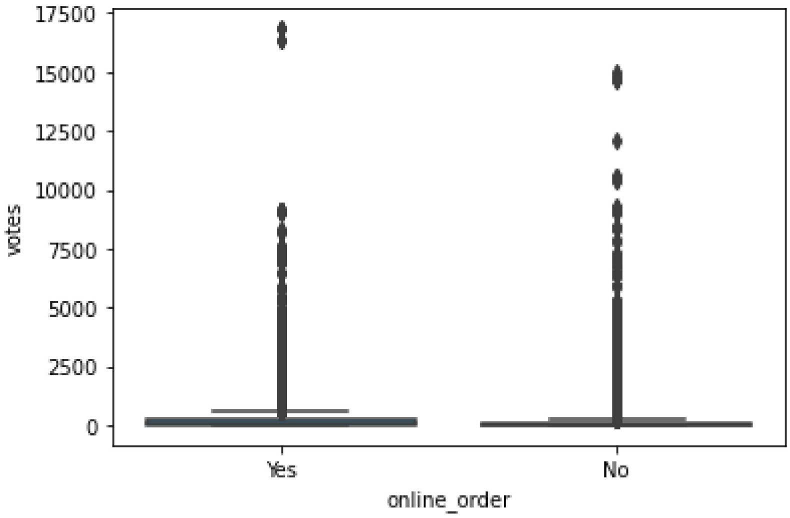


In [51]:

```
sns.boxplot(x='online_order',y='votes',data=df)
```

Out[51]:

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

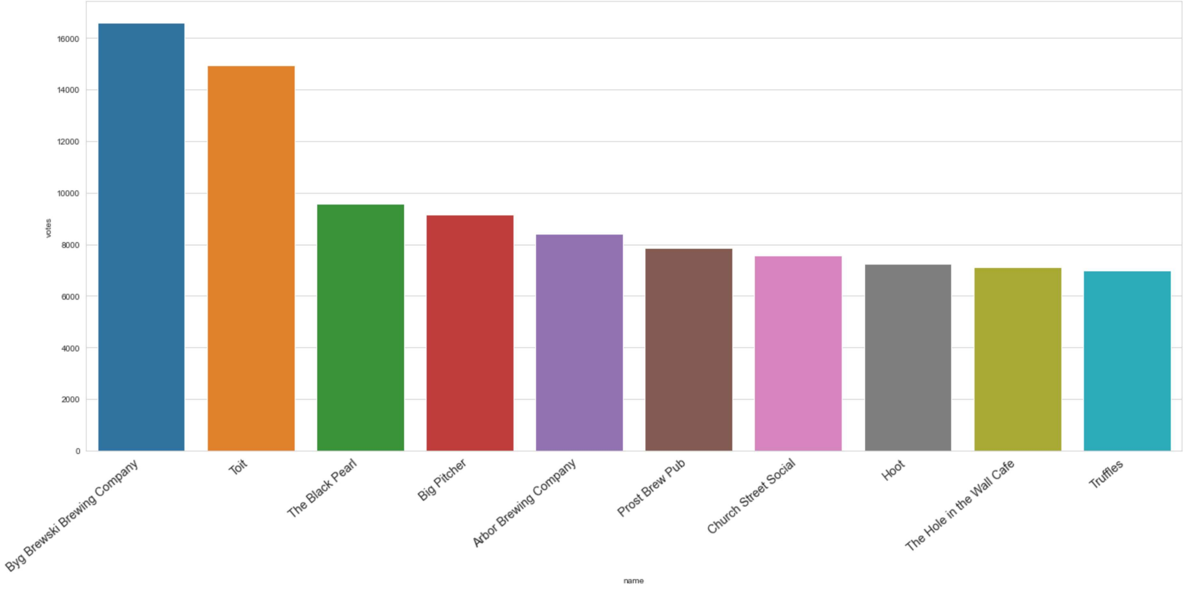


In [52]:

```
high_vote=df.groupby("name")[["votes"]].mean().sort_values(by="votes",ascending=False).head
```

In [53]:

```
fig=plt.figure(figsize=(20,10))
sns.set_style(style="whitegrid")
sns.barplot(x="name",y="votes",data=high_vote)
plt.xticks(rotation=40,horizontalalignment="right",fontsize="x-large")
plt.tight_layout()
plt.show()
```



In [ ]: