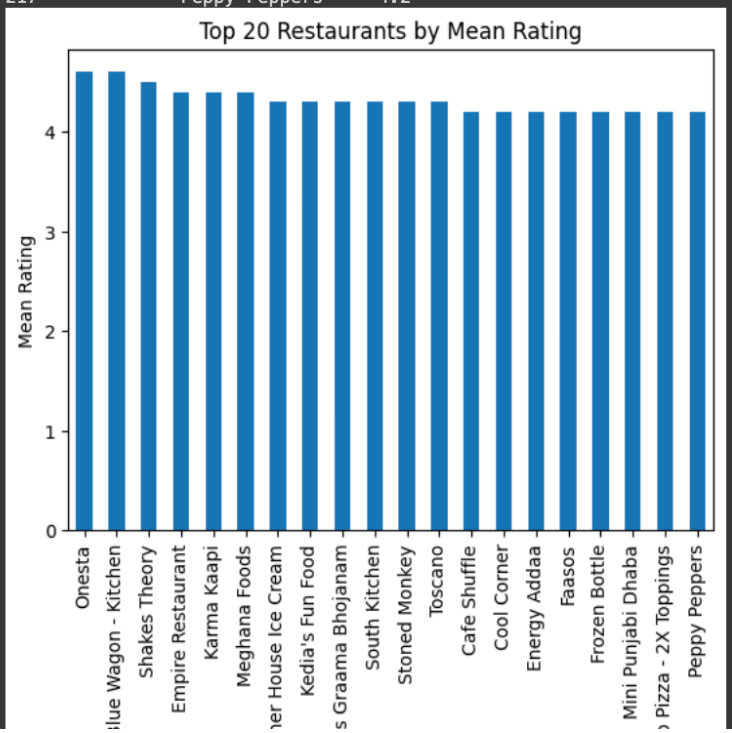
ZOMATO DATA OUTPUTS

```
plt.xlabel('Restaurant')
plt.ylabel('Mean Rating')
plt.title('Top 20 Restaurants by Mean Rating')
plt.show()

# Display the shape of the df_rate DataFrame
print(df_rate.shape)
```

	restaurant	rating
209	Onesta	4.6
297	The Blue Wagon - Kitchen	4.6
251	Shakes Theory	4.5
92	Empire Restaurant	4.4
153	Karma Kaapi	4.4
183	Meghana Foods	4.4
69	Corner House Ice Cream	4.3
155	Kedia's Fun Food	4.3
224	Prems Graama Bhojanam	4.3
262	South Kitchen	4.3
279	Stoned Monkey	4.3
312	Toscano	4.3
39	Cafe Shuffle	4.2
68	Cool Corner	4.2
93	Energy Addaa	4.2
94	Faasos	4.2
104	Frozen Bottle	4.2
186	Mini Punjabi Dhaba	4.2
188	Mojo Pizza - 2X Toppings	4.2
217	Peppy Peppers	4.2



```
sns.set_style(style='whitegrid')
sns.distplot(df_rate['rating'])
```

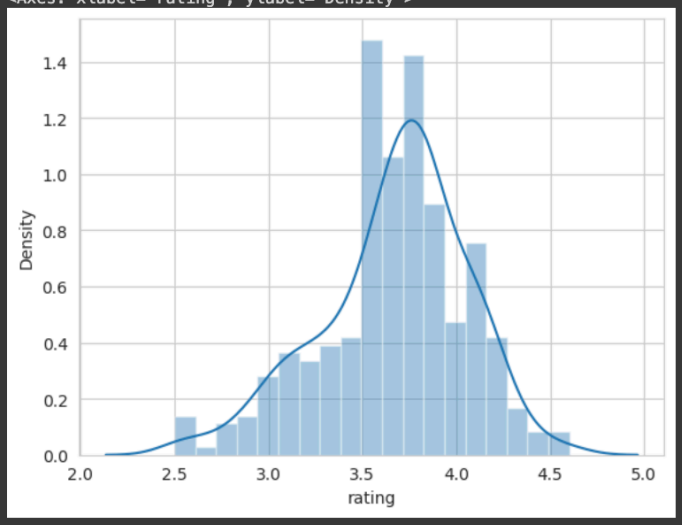
```
<ipython-input-22-fb1c7427c351>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df_rate['rating'])
<Axes: xlabel='rating', ylabel='Density'>
```



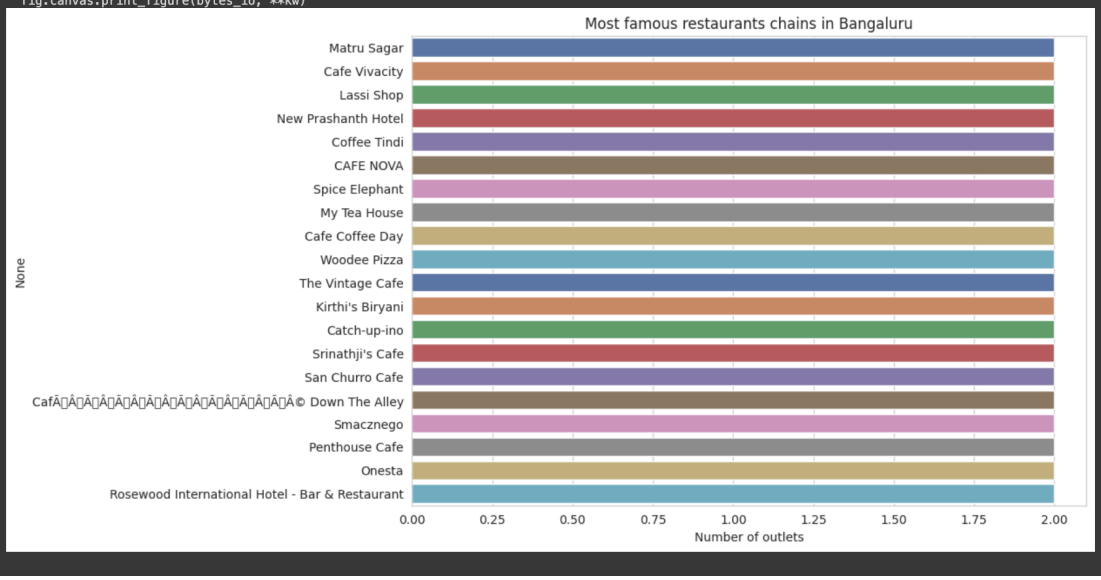

```
[23] plt.figure(figsize=(10,7))
chains=df['name'].value_counts()[0:20]
```

```
chains=df['name'].value_counts()[0:20]
sns.barplot(x=chains,y=chains.index,palette='deep')
plt.title("Most famous restaurants chains in Bangaluru")
plt.xlabel("Number of outlets")
```

```
<ipython-input-23-bd1da2a756ce>:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=chains,y=chains.index,palette='deep')
Text(0.5, 0, 'Number of outlets')
/usr/local/lib/python3.10/dist-packages/IPython/core/events.py:89: UserWarning: Glyph 131 (\x83) missing from current font.
  func(*args, **kwargs)
/usr/local/lib/python3.10/dist-packages/IPython/core/events.py:89: UserWarning: Glyph 130 (\x82) missing from current font.
  func(*args, **kwargs)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 131 (\x83) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.10/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 130 (\x82) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```

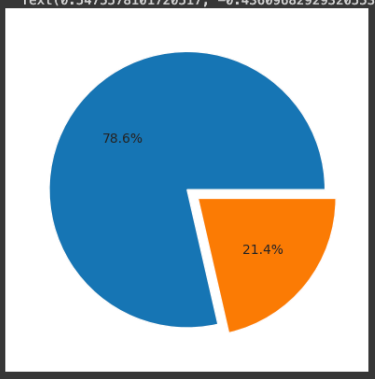


Number of outlets

NOT ACCEPTING ONLINE ORDER

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

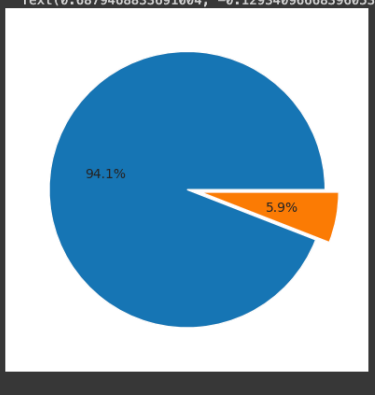
```
([<matplotlib.patches.Wedge at 0x7bca89256230>,
  <matplotlib.patches.Wedge at 0x7bca89256260>],
 [Text(-0.8604479874132245, 0.6852950174607515, ''),
  Text(0.9386705317235176, -0.7475945645026381, '')],
 [Text(-0.4693352658617588, 0.373797282251319, '78.6%'),
  Text(0.5475578101720517, -0.43609682929320553, '21.4%')])
```



What is the ratio b/w restaurants that provide and do not provide table booking ?

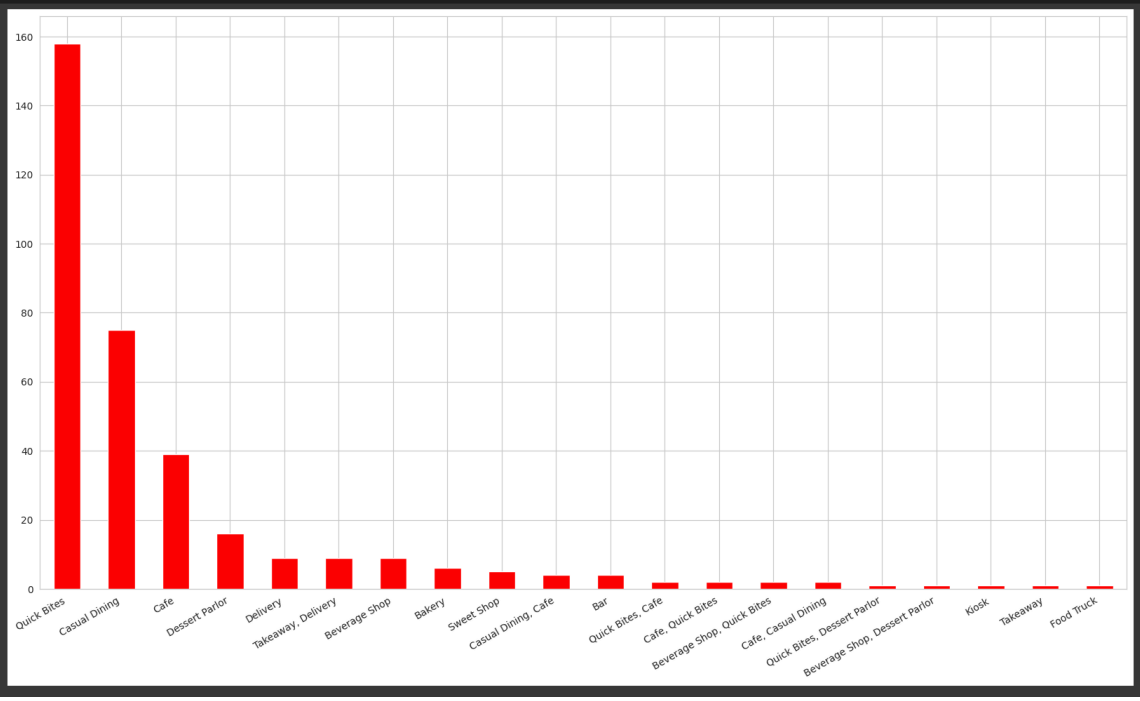
```
[25] x=df['book_table'].value_counts()
     labels=['not book','book']
     plt.pie(x,explode=[0.0,0.1],autopct='%1.1f%%')
```

```
([<matplotlib.patches.Wedge at 0x7bca8920d660>,
  <matplotlib.patches.Wedge at 0x7bca8920f4c0>],
 [Text(-1.0810593881514436, 0.20325009050336645, ''),
  Text(1.1793375143470295, -0.22172737145821808, '')],
 [Text(-0.5896687571735146, 0.11086368572910896, '94.1%'),
  Text(0.6879468833691004, -0.12934096668396053, '5.9%')])
```



How many types of restaurants we have?

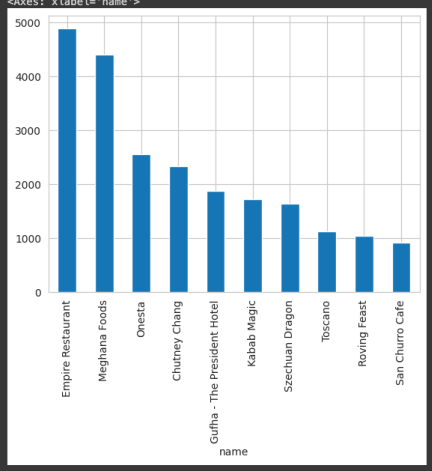
```
[26] df['rest_type'].isna().sum()
     df['rest_type'].dropna(inplace=True)
     df['rest_type'].isna().sum()
     len(df['rest_type'].unique())
     plt.figure(figsize=(20,12))
     df['rest_type'].value_counts().nlargest(20).plot.bar(color='red')
     plt.gcf().autofmt_xdate()
```



Highest Voted Restaurant

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

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



Total Restaurants At Different Locations Of Bengalore

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

```
location
Banashankari              [Jalsa, Spice Elephant, San Churro Cafe, Addhu...
Basavanagudi              [Grand Village, Timepass Dinner, Srinathji's C...
JP Nagar                                    [Re Malnad Nati Style Hotel]
Jayanagar                 [Empire Restaurant, Biryanis And More, Namma B...
Kumaraswamy Layout        [Kitchen Garden, Recipe, Tasty Bytes, Goa 0 Km...
Mysore Road               [Rosewood International Hotel - Bar & Restaura...
Rajarajeshwari Nagar                        [Beijing Bites, Firangi Bake]
South Bangalore                                              [As On Fire]
Uttarahalli               [Castle Rock, Potato House, Shree Mandarathi G...
Vijay Nagar               [Shree Cool Point, Kadalu Sea Food Restaurant,...
Name: name, dtype: object
```

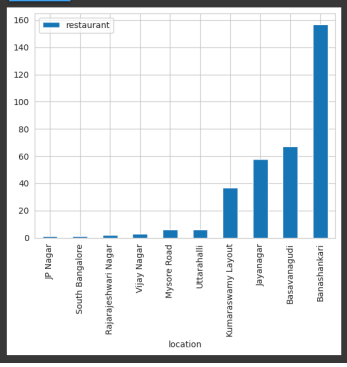
```
Uttarahalli            [Castle Rock, Potato House, Shree Mandarathi G...
Vijay Nagar            [Shree Cool Point, Kadalu Sea Food Restaurant,...
Name: name, dtype: object
```

```
restaurant=[]
location=[]
for key,location_df in df.groupby('location'):
    location.append(key)
    restaurant.append(len(location_df['name'].unique()))
df_total=pd.DataFrame(zip(location,restaurant))
df_total.columns=['location','restaurant']
df_total.set_index('location',inplace=True)
df_total.sort_values(by='restaurant').tail(10)
df_total.sort_values(by='restaurant').tail(10).plot.bar()
df_total.sort_values(by='restaurant').tail(10).index
!pip install plotly
import plotly.graph_objects as go
trace1 = go.Bar(
        x = df_total['restaurant'].nlargest(10).index,
        y = df_total['restaurant'].nlargest(10),
        name= 'Priority')
iplot([trace1])
df.isnull().sum()
```

```
Requirement already satisfied: plotly in /usr/local/lib/python3.10/dist-packages (5.15.0)
Requirement already satisfied: tenacity>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from plotly) (8.2.3)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from plotly) (23.2)
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-33-19480dad7c62> in <cell line: 18>()
     16         y = df_total['restaurant'].nlargest(10),
     17         name= 'Priority')
---> 18 iplot([trace1])
     19 df.isnull().sum()

NameError: name 'iplot' is not defined
```

EXPLAIN ERROR



Different Type Of Restaurants

```
#data is represented in form of percentage
(df['rest_type'].value_counts()/len(df))*100
```

```
Quick Bites                   44.507042
Casual Dining                 21.126761
Cafe                          10.985915
Dessert Parlor                 4.507042
Delivery                       2.535211
Takeaway, Delivery             2.535211
Beverage Shop                  2.535211
Bakery                         1.690141
Sweet Shop                     1.408451
Casual Dining, Cafe            1.126761
Bar                            1.126761
Quick Bites, Cafe              0.563380
Cafe, Quick Bites              0.563380
Beverage Shop, Quick Bites     0.563380
Cafe, Casual Dining            0.563380
Quick Bites, Dessert Parlor    0.281690
Beverage Shop, Dessert Parlor  0.281690
Kiosk                          0.281690
Takeaway                       0.281690
Food Truck                     0.281690
Fine Dining                    0.281690
Sweet Shop, Quick Bites        0.281690
Bakery, Quick Bites            0.281690
Dessert Parlor, Sweet Shop     0.281690
Pub                            0.281690
Bakery, Dessert Parlor         0.281690
Mess                           0.281690
Pub, Casual Dining             0.281690
Name: rest_type, dtype: float64
```

Total number of variety of restaurants ie north indian,south Indian

```
cuisines = df['cuisines'].value_counts()[:10]
sns.barplot(x=cuisines.index, y=cuisines)
plt.xlabel('Cuisine')
plt.ylabel('Count')
plt.title("Most popular cuisines of Bangalore")
plt.xticks(rotation=90)
plt.show()
```