



zomato

Zomato EDA & Market Analysis

About Dataset:

Zomato is an Indian multinational restaurant aggregator and food delivery company founded by Deepinder Goyal and Pankaj Chaddah in 2008. Zomato provides information, menus and user-reviews of restaurants as well as food delivery options from partner restaurants in select cities.

This dataset containing information of food restraunts in banglaore who are working with Zomato. The data was scraped from Zomato in two phase. After going through the structure of the website I found.

for each neighborhood there are 6-7 category of restaurants viz. Buffet, Cafes, Delivery, Desserts, Dine-out, Drinks & nightlife, Pubs and bars. So, here we are trying to find the best restaurants for customer depends on their need.

```
In [1]: import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

import warnings
warnings.filterwarnings("ignore")
```

```
In [2]: rc = {'figure.dpi': 100, 'axes.labelsize': 7,
           'axes.facecolor': '#FAFAD2', 'grid.color': 'Red','figure.figsize':(12,8),
           'figure.facecolor': '#FAFAD2'}
sns.set_theme(context='notebook',
              style='dark',
              palette='deep',
              font='Comic Sans Ms',
              font_scale=1.5,
              color_codes='red',
              rc=rc)

plt.rcParams['legend.fontsize'] = 10
plt.rcParams['font.size'] = 15
plt.rcParams['font.family'] = 'Comic Sans Ms'
plt.rcParams['axes.facecolor'] = '#FAFAD2'
plt.rcParams["axes.labelcolor"] = '#FAFAD2'
```

```
In [3]: pd.set_option("display.max_columns",None)
```

```
In [4]: df = pd.read_csv('zomato.csv')
df.head()
```

Out[4]:

	url	address	name	online_order	book_table	rate	votes	phone	location	rest_type	dish_li
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1/5	775	080 42297555\n+91 9743772233	Banashankari	Casual Dining	Pa Lu Bu Ma Paç Par La
1	https://www.zomato.com/bangalore/spice-elephant...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1/5	787	080 41714161	Banashankari	Casual Dining	Mon Lu Bu Choco Nirv Thai
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	Chur Cannell Minestr Soup, Ch
3	https://www.zomato.com/bangalore/addhuri-udipi...	1st Floor, Annakuteera, 3rd Stage, Banashankar...	Addhuri Udupi Bhojana	No	No	3.7/5	88	+91 9620009302	Banashankari	Quick Bites	Ma D
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\n+91 9901210005	Basavanagudi	Casual Dining	Pani Gol Ga

```
In [5]: df = df.drop(['url', 'address', 'phone', 'menu_item', 'reviews_list','dish_liked'], axis = 1)
```

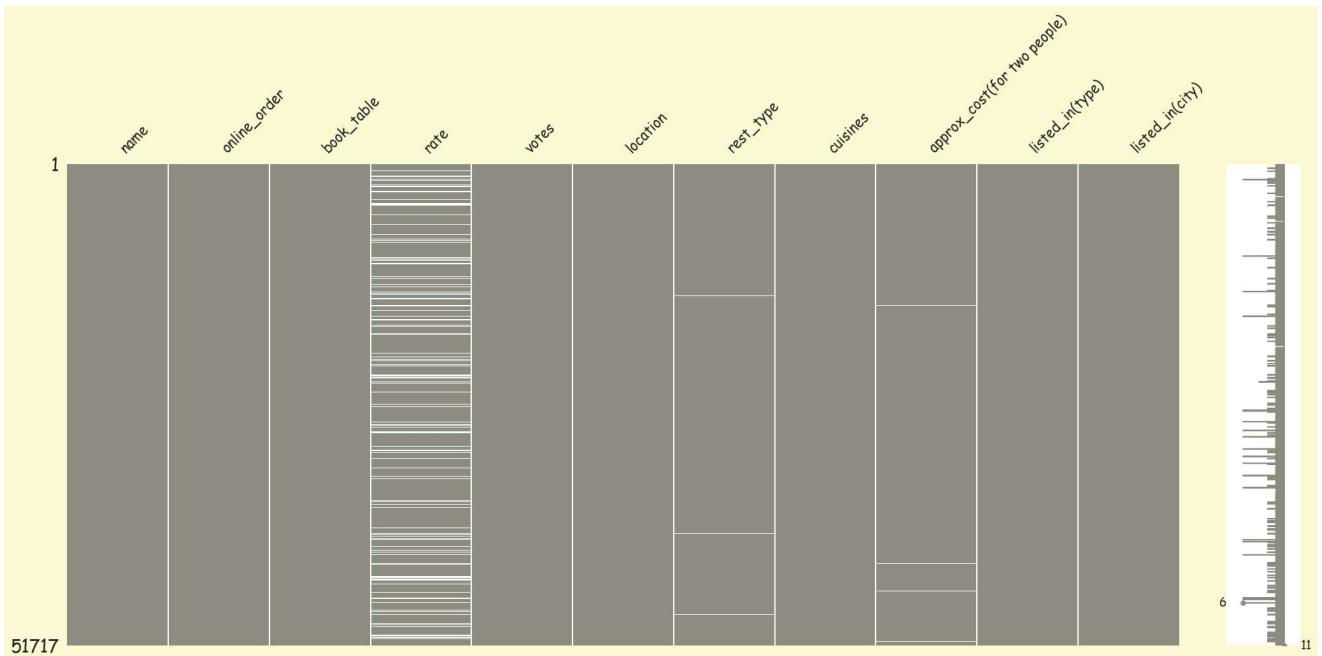
```
In [6]: df.shape
```

```
Out[6]: (51717, 11)
```

```
In [7]: (df.isnull().sum()/df.shape[0])*100
```

```
Out[7]: name          0.000000
online_order      0.000000
book_table        0.000000
rate            15.033741
votes           0.000000
location         0.040606
rest_type        0.438927
cuisines         0.087012
approx_cost(for two people) 0.669026
listed_in(type)    0.000000
listed_in(city)     0.000000
dtype: float64
```

```
In [8]: import missingno as msno  
msno.matrix(df,color=(0.55, 0.55, 0.55));
```



```
In [9]: df.dropna(inplace=True)
```

Information About the data set

```
In [10]: # --- Print Dataset Detail ---  
print('\033[7m' + ': Dataset Details :')  
print('\033[0m\033[43m*' * 47 + '\033[0m')  
df.info(memory_usage = False)
```

```
: Dataset Details :  
*****  
<class 'pandas.core.frame.DataFrame'>  
Int64Index: 43533 entries, 0 to 51716  
Data columns (total 11 columns):  
 #   Column           Non-Null Count Dtype  
---  --  
 0   name            43533 non-null  object  
 1   online_order    43533 non-null  object  
 2   book_table      43533 non-null  object  
 3   rate            43533 non-null  object  
 4   votes           43533 non-null  int64  
 5   location         43533 non-null  object  
 6   rest_type        43533 non-null  object  
 7   cuisines         43533 non-null  object  
 8   approx_cost(for two people) 43533 non-null  object  
 9   listed_in(type) 43533 non-null  object  
 10  listed_in(city) 43533 non-null  object  
dtypes: int64(1), object(10)
```

```
In [11]: (df.online_order.value_counts()/df.shape[0])*100
```

```
Out[11]: Yes     64.783038  
No      35.216962  
Name: online_order, dtype: float64
```

```
In [12]: print(f"\033[031m\033[1m")  
print(df['rate'].unique())
```

```
['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'  
'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']
```

{ Cleaning the Rating }

```
In [13]: def rate_clean(value):
    if(value=='NEW' or value=='-'):
        return np.nan
    else:
        value = str(value).split('/')
        value = value[0]
        return float(value)
```

```
In [14]: print(f"\033[031m\033[1m")
df['rate'] = df['rate'].apply(rate_clean)
print(df['rate'].unique())
```

```
[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 [15]: df.dropna(inplace=True)
```

{ Giving appropriate names to the Features }

```
In [16]: df.rename(columns = {'name':'Restaurant Name','online_order':'Online Order','book_table':'Book Table'}, inplace = True)
df.rename(columns = {'rate':'Ratings','votes':'Votes','location':'Location','rest_type':'Type Of Restaurant','cuisines':'Cuisines',
df.rename(columns = {'approx_cost(for two people)':'Cost For Two', 'listed_in(type)':'Catagory','listed_in(city)' : "Listed in Ci
```

```
In [17]: df['Restaurant Name'] = df['Restaurant Name'].str.replace('Ã', '')
df['Restaurant Name'] = df['Restaurant Name'].str.replace('®', '')
df['Restaurant Name'] = df['Restaurant Name'].str.replace('Ã', '')
df['Restaurant Name'] = df['Restaurant Name'].str.replace('•', '')
df['Restaurant Name'] = df['Restaurant Name'].str.replace('•', '')
```

{ Unique Location List in Bangalore }

```
In [18]: print(f"\033[034m\033[1m")
print("Unique Location List :")
print(f"\033[031m\033[1m")
print(df['Location'].unique())
```

Unique Location List :

```
['Banashankari' 'Basavanagudi' 'Mysore Road' 'Jayanagar'
 'Kumaraswamy Layout' 'Rajarajeshwari Nagar' 'Vijay Nagar' 'Uttarahalli'
 'JP Nagar' 'South Bangalore' 'City Market' 'Bannerghatta Road' 'BTM'
 'Kanakapura Road' 'Bommanahalli' 'Electronic City' 'Wilson Garden'
 'Shanti Nagar' 'Koramangala 5th Block' 'Richmond Road' 'HSR'
 'Koramangala 7th Block' 'Bellandur' 'Sarjapur Road' 'Marathahalli'
 'Whitefield' 'East Bangalore' 'Old Airport Road' 'Indiranagar'
 'Koramangala 1st Block' 'Frazer Town' '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' 'Domlur' 'Koramangala 8th Block'
 'Ejipura' 'Jeevan Bhima Nagar' 'Old Madras Road' 'Seshadripuram'
 'Kammanahalli' 'Koramangala 6th Block' 'Majestic' 'Langford Town'
 'Central Bangalore' 'Brookefield' 'ITPL Main Road, Whitefield'
 'Varthur Main Road, Whitefield' 'Koramangala 2nd Block'
 'Koramangala 3rd Block' 'Koramangala 4th Block' 'Koramangala'
 'Hosur Road' 'RT Nagar' 'Banaswadi' 'North Bangalore' 'Nagawara' 'Hennur'
 'Kalyan Nagar' 'HBR Layout' 'Rammurthy Nagar' 'Thippasandra'
 'CV Raman Nagar' 'Kaggadasapura' 'Kengeri' 'Sankey Road' 'Malleshwaram'
 'Sanjay Nagar' 'Sadashiv Nagar' 'Basaveshwara Nagar' 'Rajajinagar'
 'Yeshwantpur' 'New BEL Road' 'West Bangalore' 'Magadi Road' 'Yelahanka'
 'Sahakara Nagar' 'Jalahalli' 'Hebbal' 'Nagarbhavi' 'Peenya' 'KR Puram']
```

{ Number of Restaurant Categories }

```
In [19]: print(f"\033[034m\033[1m")
print(df["Catagory"].unique())
print(f"\033[035m\033[1m")
print("Categories Count :")
df["Catagory"].value_counts()
```

```
['Buffet' 'Cafes' 'Delivery' 'Desserts' 'Dine-out' 'Drinks & nightlife'
 'Pubs and bars']
```

Categories Count :

```
Out[19]: Delivery      20452
Dine-out      14066
Desserts      2709
Cafes         1511
Drinks & nightlife 1045
Buffet        848
Pubs and bars 632
Name: Catagory, dtype: int64
```

```
In [20]: df.isnull().sum()
```

```
Out[20]: Restaurant Name      0
Online Order      0
Book Table       0
Ratings          0
Votes            0
Location         0
Type Of Restaurant 0
Cuisines         0
Cost For Two     0
Catagory         0
Listed in City   0
dtype: int64
```

```
In [21]: df.shape
```

```
Out[21]: (41263, 11)
```

```
In [22]: df.dtypes
```

```
Out[22]: Restaurant Name      object
Online Order      object
Book Table       object
Ratings          float64
Votes            int64
Location         object
Type Of Restaurant  object
Cuisines         object
Cost For Two     object
Catagory         object
Listed in City   object
dtype: object
```

Converting Cost of Two to Cost per Head

```
In [23]: #Changing Cost per head from cost of 2
df['Cost For Two']=df['Cost For Two'].str.replace(',','')
df['Cost For Two']=df['Cost For Two'].astype(float)
df['Cost For Two']=df['Cost For Two'].apply(lambda x: x/2)
df.rename(columns = {'Cost For Two':'Cost Per Head'}, inplace = True)
```

```
In [24]: ty=df['Type Of Restaurant'].value_counts(ascending = False)
np.mean(ty)
```

```
Out[24]: 474.28735632183907
```

```
In [25]: typeLessThan10=ty[ty<10]
typeLessThan10
```

```
Out[25]: Bar, Quick Bites      9
Microbrewery, Lounge          9
Food Court, Dessert Parlor   8
Casual Dining, Sweet Shop    8
Bar, Lounge                   6
Fine Dining, Microbrewery    5
Bar, Pub                      5
Food Court, Casual Dining    5
Casual Dining, Quick Bites    4
Club, Casual Dining          4
Quick Bites, Mess             4
Quick Bites, Meat Shop        4
Bhojanalya                   3
Cafe, Food Court              2
Bakery, Food Court            2
Food Court, Beverage Shop    2
Dessert Parlor, Food Court   2
Dessert Parlor, Kiosk          2
Bakery, Beverage Shop         1
Quick Bites, Kiosk             1
Name: Type Of Restaurant, dtype: int64
```

```
In [26]: def cleanTypeOfRestaurant(value):
    if(value in typeLessThan10):
        return 'others'
    else:
        return value

df['Type Of Restaurant']=df['Type Of Restaurant'].apply(cleanTypeOfRestaurant)
df['Type Of Restaurant'].value_counts()
```

```
Out[26]: Quick Bites           13885
Casual Dining                  9614
Cafe                           3369
Dessert Parlor                 1850
Delivery                        1667
...
Casual Dining, Irani Cafee     15
Fine Dining, Lounge             14
Microbrewery, Bar               13
Dessert Parlor, Sweet Shop     12
Casual Dining, Lounge           10
Name: Type Of Restaurant, Length: 68, dtype: int64
```

```
In [27]: location = df['Location'].value_counts(ascending = False)
location_lessthan10 = location[location<10]
def cleanlocation(value):
    if(value in location_lessthan10):
        return 'others'
    else:
        return value
df['Location'] = df['Location'].apply(cleanlocation)
df['Location'].value_counts()
```

```
Out[27]: BTM                    3879
Koramangala 5th Block          2297
HSR                     1993
Indiranagar                1800
JP Nagar                   1710
...
Old Madras Road                22
Kanakapura Road                19
Mysore Road                   17
North Bangalore                10
KR Puram                      10
Name: Location, Length: 84, dtype: int64
```

```
In [28]: cuisines = df['Cuisines'].value_counts(ascending = False)
cuisines_lessthan1 = cuisines[cuisines<1]

def cleancuisines(value):
    if(value in cuisines_lessthan1):
        return 'others'
    else:
        return value

df['Cuisines'] = df['Cuisines'].apply(cleancuisines)
df['Cuisines'].value_counts()
```

```
Out[28]: North Indian           2108
North Indian, Chinese          1953
South Indian                  1232
Cafe                          620
Bakery, Desserts              613
...
North Indian, Mughlai, Lucknowi      1
Continental, Thai, North Indian, Chinese 1
North Indian, Bengali, Chinese, Beverages 1
Biryani, North Indian, Chinese, Kebab   1
North Indian, Chinese, Arabian, Momos   1
Name: Cuisines, Length: 2367, dtype: int64
```

```
In [29]: print(df[['Catagory']].nunique())
print(df['Catagory'].unique())
```

```
Catagory    7
dtype: int64
['Buffet' 'Cafes' 'Delivery' 'Desserts' 'Dine-out' 'Drinks & nightlife'
 'Pubs and bars']
```

(Top 10 Unique Restaurant Names by Numbers)

```
In [30]: print("\u033[031m\033[1m")
print("Unique Restaurant Names :", df['Restaurant Name'].nunique())
df['Restaurant Name'].value_counts().nlargest(10).to_frame().style.background_gradient(cmap='copper').set_precision(2)
```

Unique Restaurant Names : 6574

Out[30]:

Restaurant Name	
Cafe Coffee Day	86
Onesta	85
Empire Restaurant	69
Kanti Sweets	68
Five Star Chicken	68
Just Bake	67
Baskin Robbins	62
Pizza Hut	60
Petoo	60
KFC	60

List of top 10 Restaurant Name in Bangalore is shown.

(Top 10 Unique Types Of Restaurant by Numbers)

```
In [31]: print(f"\033[031m\033[1m")
print("Unique Type Of Restaurant :", df['Type Of Restaurant'].nunique())
df['Type Of Restaurant'].value_counts().nlargest(10).to_frame().style.background_gradient(cmap='copper').set_precision(2)
```

Unique Type Of Restaurant : 68

Out[31]:

Type Of Restaurant	
Quick Bites	13885
Casual Dining	9614
Cafe	3369
Dessert Parlor	1850
Delivery	1667
Takeaway, Delivery	1279
Casual Dining, Bar	1092
Bakery	704
Bar	640
Beverage Shop	640

List of top 10 Type Of Restaurant in Bangalore is shown. Quick Bites has the highest number in Bangalore

(Different Catagory)

```
In [32]: print(f"\033[031m\033[1m")
print("Unique Catagory :", df['Catagory'].nunique())
df['Catagory'].value_counts().nlargest(10).to_frame().style.background_gradient(cmap='copper').set_precision(2)
```

Unique Catagory : 7

Out[32]:

Catagory	
Delivery	20452
Dine-out	14066
Desserts	2709
Cafes	1511
Drinks & nightlife	1045
Buffet	848
Pubs and bars	632

List of top 7 Catagory in Bangalore is shown. Delivery has the highest number in Bangalore

(Top 10 Unique Cuisines by Numbers)

```
In [33]: print(f"\033[031m\033[1m")
print("Unique Cuisines :", df['Cuisines'].nunique())
df['Cuisines'].value_counts().nlargest(10).to_frame().style.background_gradient(cmap='copper').set_precision(2)
```

Unique Cuisines : 2367

Out[33]:

Cuisines	
North Indian	2108
North Indian, Chinese	1953
South Indian	1232
Cafe	620
Bakery, Desserts	613
Biryani	600
South Indian, North Indian, Chinese	561
Desserts	545
Fast Food	514
Chinese	410

List of top 10 Cuisines in Bangalore is shown. North Indian food is the most liked Cuisines in Bangalore

{ WordCloud for Cuisines }

In [34]:

```
from wordcloud import WordCloud, STOPWORDS

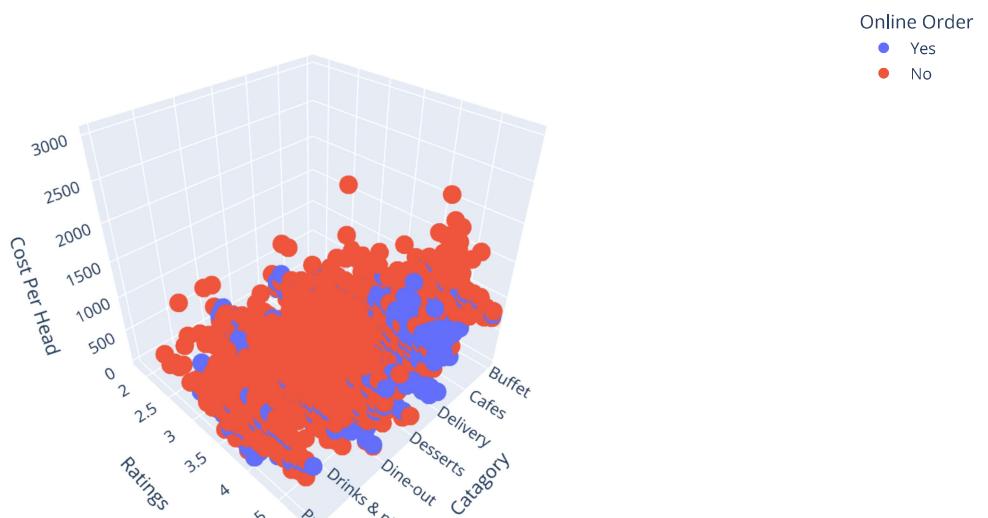
text = " ".join(Company for Company in df["Cuisines"])
font = "Quicksand-Bold.ttf"
word_cloud = WordCloud(width = 2300,
                       height = 800,
                       colormap = 'jet',
                       background_color = "white").generate(text)
plt.figure(figsize = (50, 8))
plt.imshow(word_cloud, interpolation = "gaussian")
plt.axis("off")
plt.show()
```



{ 3D Scatter Plot }

In [35]:

```
# 3D Scatter Plot
import plotly.express as px
fig = px.scatter_3d(df, x='Category', z='Cost Per Head', y='Ratings', color='Online Order')
fig.show()
```



{ Top 10 Listed in City by Numbers }

```
In [36]: print(f"\033[0;1m\033[1m")
print("Unique Listed in City :", df['Listed in City'].nunique())
df['Listed in City'].value_counts().nlargest(10).to_frame().style.background_gradient(cmap='copper').set_precision(2)
```

Unique Listed in City : 30

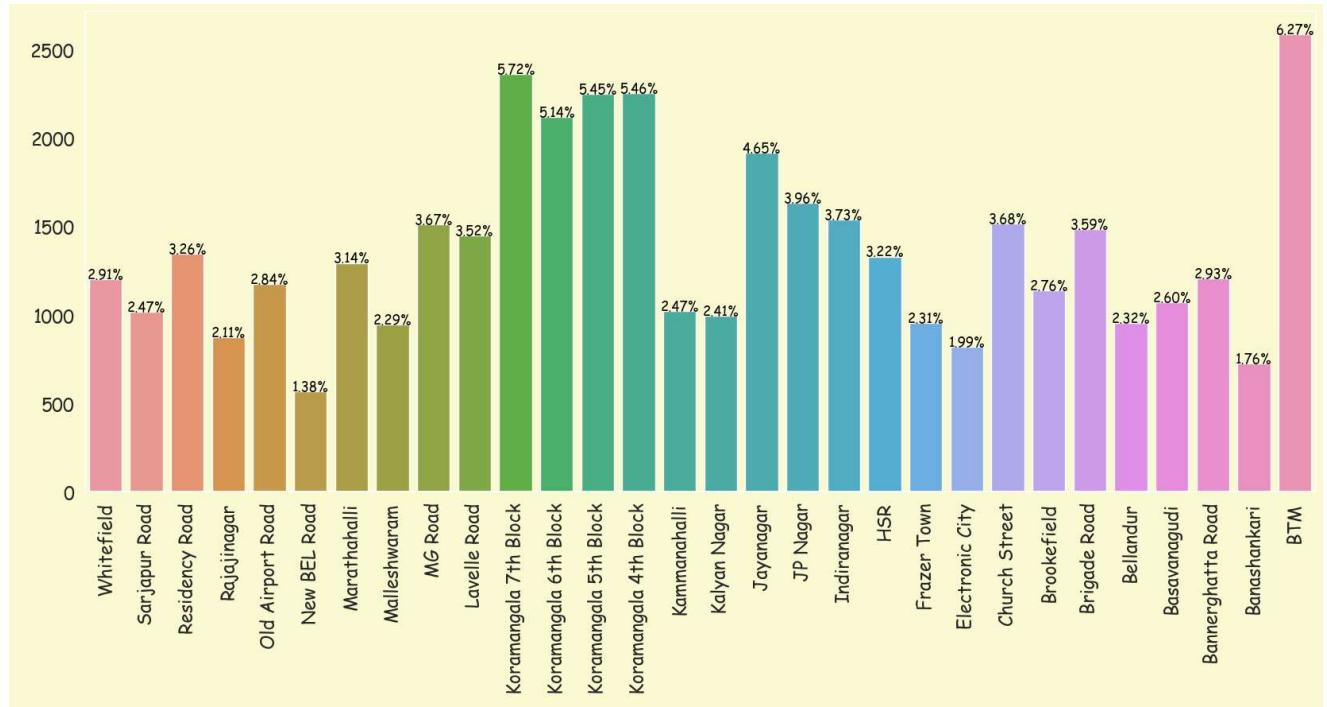
Out[36]:

Listed in City	
BTM	2586
Koramangala 7th Block	2361
Koramangala 4th Block	2254
Koramangala 5th Block	2250
Koramangala 6th Block	2121
Jayanagar	1918
JP Nagar	1633
Indiranagar	1538
Church Street	1518
MG Road	1514

List of top 10 Listed in City in Bangalore is shown

(Percentage of Restaurant Listed in City)

```
In [37]: plt.figure(figsize = (20, 8))
s = sns.countplot(x = df['Listed in City'].sort_values(ascending=False), data = df)
sizes=[]
for p in s.patches:
    height = p.get_height()
    sizes.append(height)
    s.text(p.get_x() + p.get_width()/2.,
           height + 3,
           '{:1.2f}%'.format(height/len(df)*100),
           ha="center", fontsize=12, color = "Black")
plt.xticks(rotation=90)
```



In this graph the percentage of restaurants in Bangalore is shown

Here the top 2 Location for the Maximum percentage Restaurant in Bangalore are:

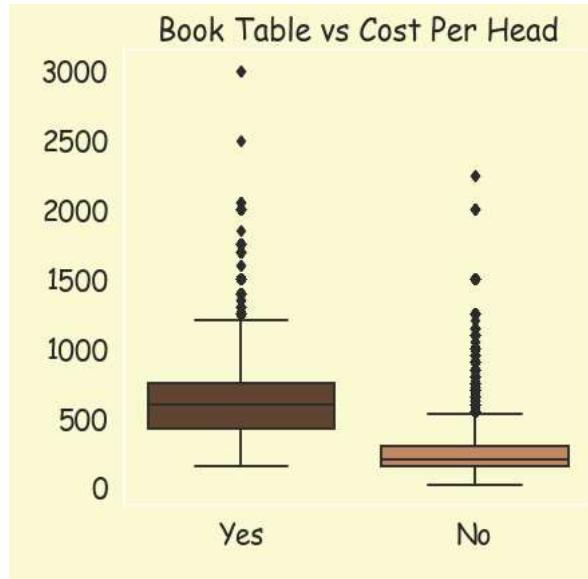
- BTM has highest percentage (6.27 %) of restaurants in Bangalore
- Karamangala 7th Block has second highest percentage (5.72 %) of restaurants in Bangalore

Here the Top 2 Location for the Minimum percentage Restaurant in Bangalore are:

- New BEL Road has least percentage (1.38 %) of restaurants in Bangalore
- Banashankari has second least percentage (1.76 %) of restaurants in Bangalore

(Book Table vs Cost Per Head)

```
In [38]: plt.figure(figsize=(5,5),dpi=90)
plt.title('Book Table vs Cost Per Head')
sns.boxplot(x='Book Table',y='Cost Per Head',data=df,palette='copper');
```



Book Table vs Cost Per Head

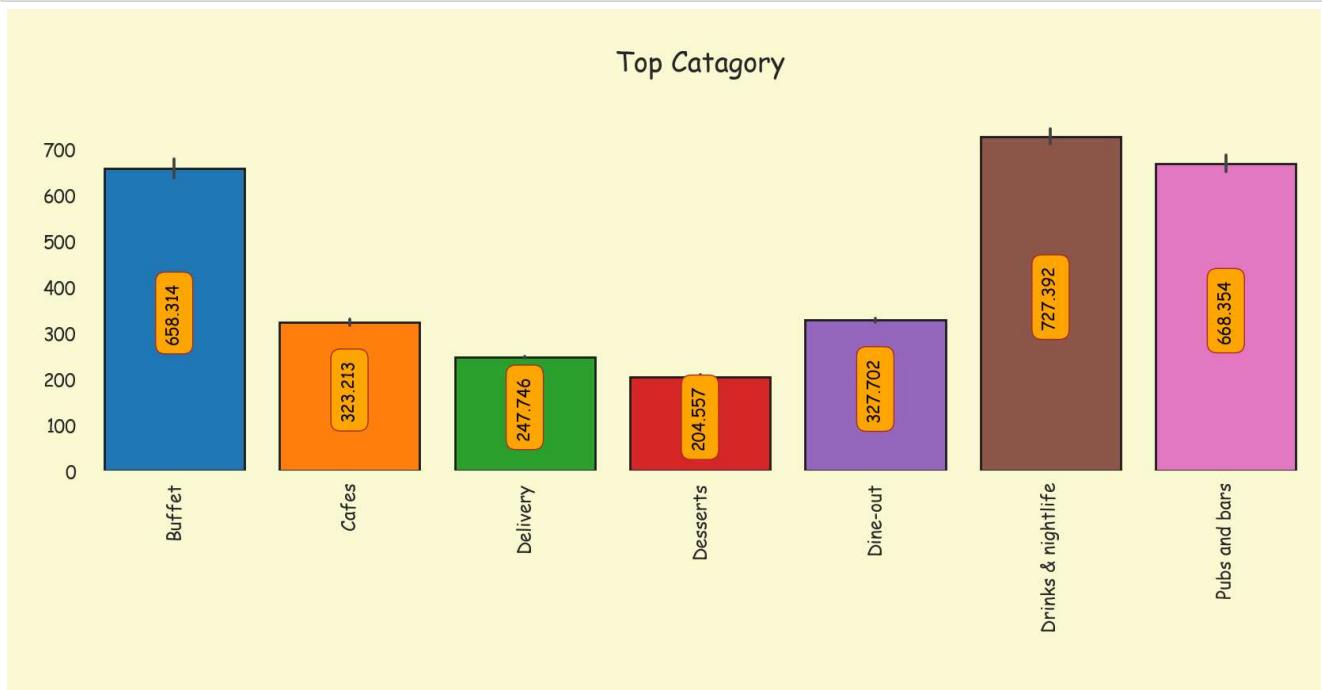
The Restaurants which accepts booking have the highest Cost Per Head as compared to those which do not accept booking.

If Restaurants want increase Cost Per Head sales they should start taking bookings.

(Barplot for Category vs Cost per Head)

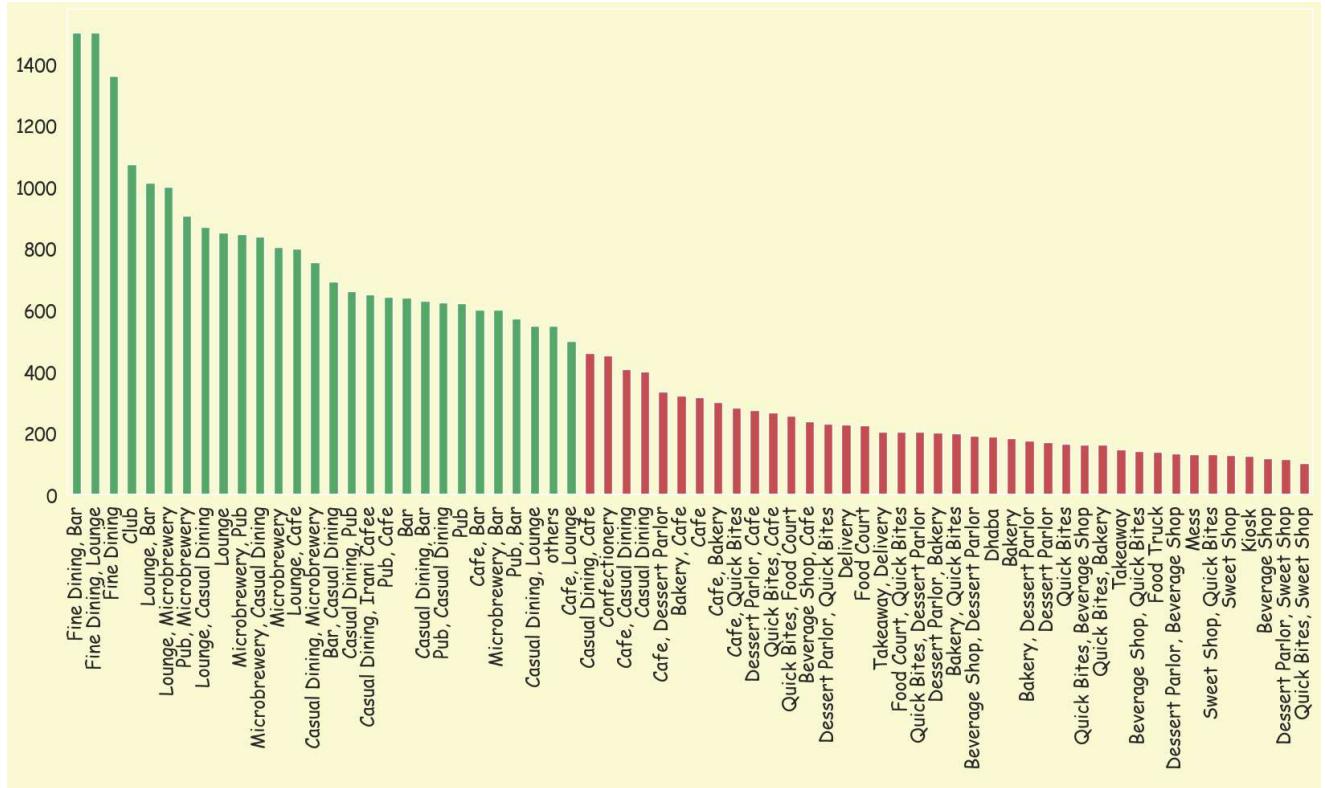
```
In [39]: plt.subplots(figsize=(20,6))
p = sns.barplot(x=df["Catagory"],y=df["Cost Per Head"],palette='tab10', saturation=1, edgecolor = "#1c1c1c", linewidth = 2)
p.axes.set_title("\nTop Catagory\n", fontsize=25)
plt.xlabel("Cost Per Head" , fontsize = 20)
plt.ylabel("\nCatagory" , fontsize = 20)
# plt.yscale("Log")
plt.xticks(rotation = 90)
for container in p.containers:
    p.bar_label(container,label_type = "center",padding = 6,size = 15,color = "black",rotation = 90,
    bbox={"boxstyle": "round", "pad": 0.6, "facecolor": "orange", "edgecolor": "Brown", "alpha": 1})

sns.despine(left=True, bottom=True)
plt.show()
```



{ Barplot for Type Of Restaurant Vs Mean Cost Per Head }

```
In [40]: plt.figure(figsize = (20,8))
df2=df.groupby('Type Of Restaurant')['Cost Per Head'].mean().sort_values(ascending=False)
color = [('r' if i < 500 else 'g') for i in df2]
df2.plot.bar(color=color);
```



```
In [41]: df[['Restaurant Name','Online Order','Book Table','Location','Type Of Restaurant','Cuisines','Catagory','Listed in City']].nunique
```

```
Out[41]:
```

	Count
Restaurant Name	6574
Online Order	2
Book Table	2
Location	84
Type Of Restaurant	68
Cuisines	2367
Catagory	7
Listed in City	30

dtype: int64

(Restaurant Category by total sum of (Ratings & Cost per Head))

```
In [42]: df.reset_index().groupby('Catagory')[['Ratings', 'Cost Per Head']].agg('sum').style.background_gradient(cmap='copper').set_precision(2)
```

```
Out[42]:
```

Catagory	Ratings	Cost Per Head
	Buffet	3376.20
Cafes	5850.80	488375.00
Delivery	74753.80	5066908.00
Desserts	10237.50	554145.00
Dine-out	51798.40	4609450.00
Drinks & nightlife	4197.10	760125.00
Pubs and bars	2542.30	422400.00

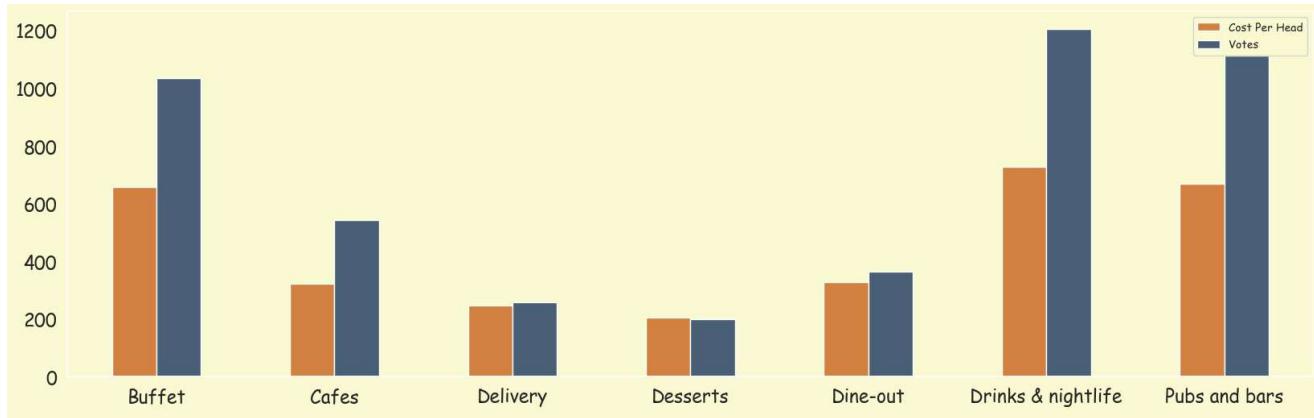
(Restaurant Category by Mean of (Ratings & Cost per Head))

```
In [43]: df.groupby('Catagory')[['Ratings', 'Cost Per Head']].agg('mean').style.background_gradient(cmap='copper').set_precision(2)
```

Out[43]:

Catagory	Ratings	Cost Per Head
Buffet	3.98	658.31
Cafes	3.87	323.21
Delivery	3.66	247.75
Desserts	3.78	204.56
Dine-out	3.68	327.70
Drinks & nightlife	4.02	727.39
Pubs and bars	4.02	668.35

```
In [44]: color = ['#D28140', '#495F75']
df.groupby(["Catagory"])[["Cost Per Head", 'Votes']].mean().plot(kind="bar", figsize=(20,6), color=color);
plt.xticks(rotation=0);
```

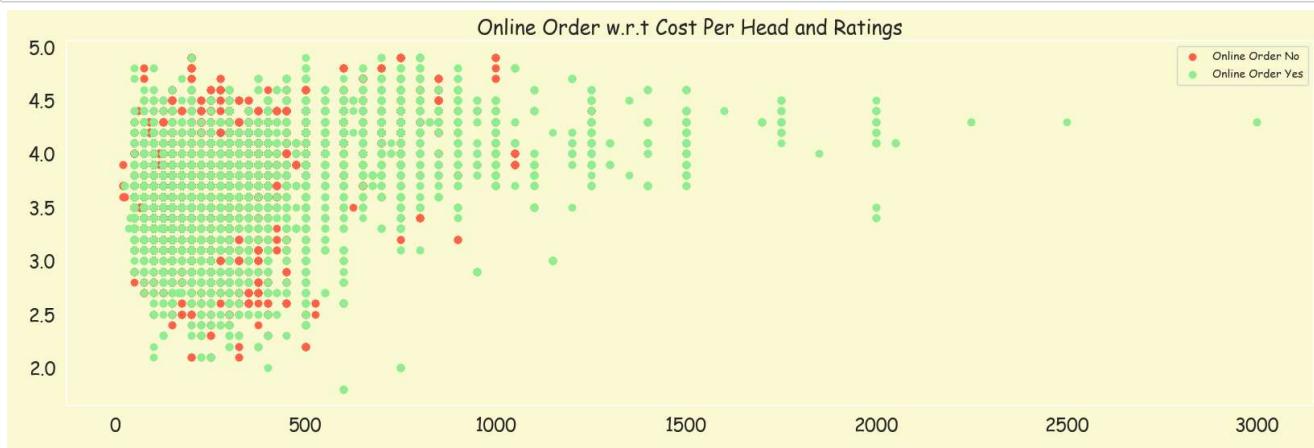


```
In [45]: plt.figure(figsize=(20,6))

plt.scatter(df['Cost Per Head'][df['Online Order']=="Yes"],
            df.Ratings[df['Online Order']=="Yes"],
            c="tomato")

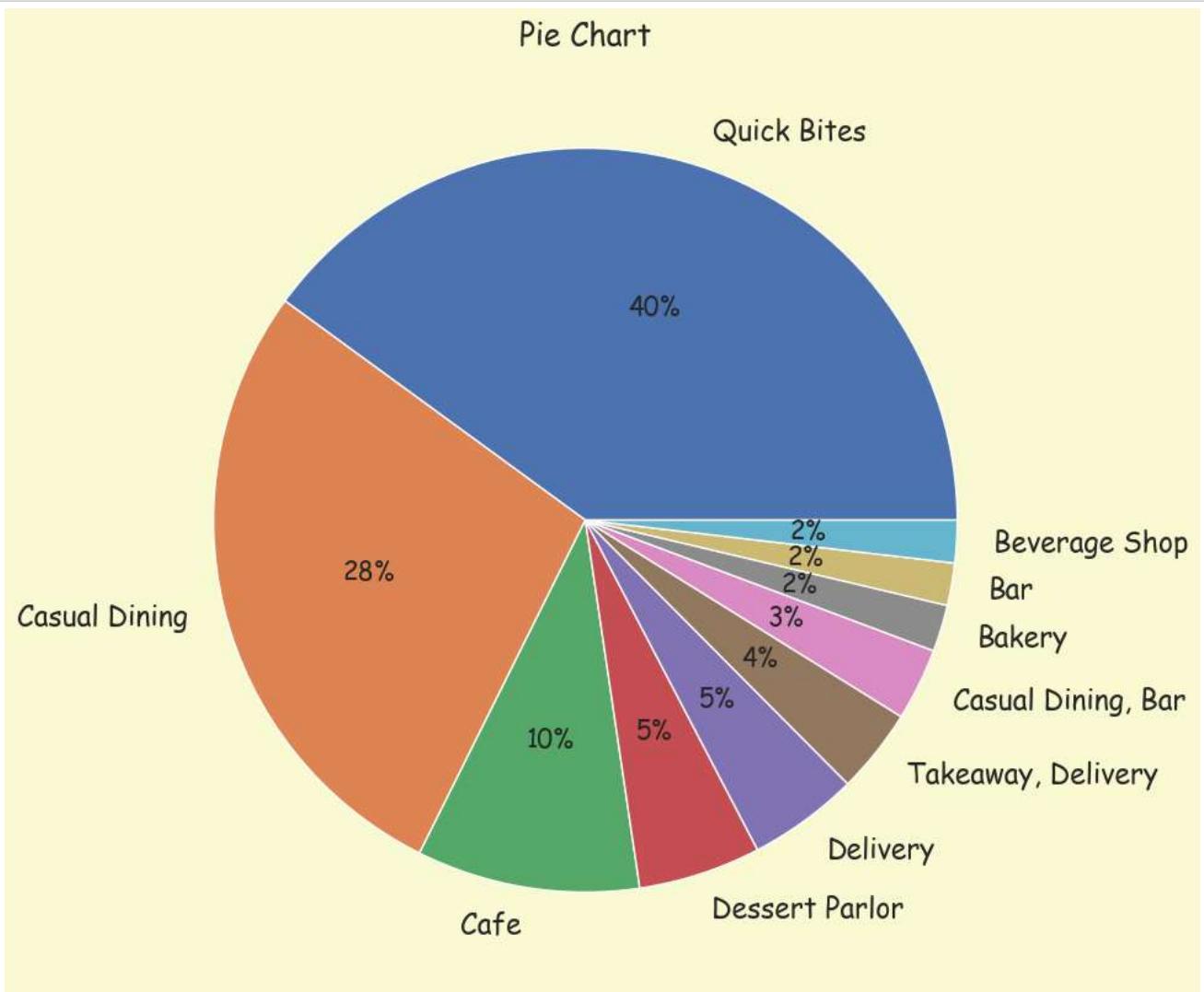
plt.scatter(df['Cost Per Head'][df['Online Order']=="No"],
            df['Ratings'][df['Online Order']=="No"],
            c="lightgreen")

plt.title("Online Order w.r.t Cost Per Head and Ratings")
plt.xlabel("Cost Per Head")
plt.legend(["Online Order No", "Online Order Yes"])
plt.ylabel("Ratings");
```

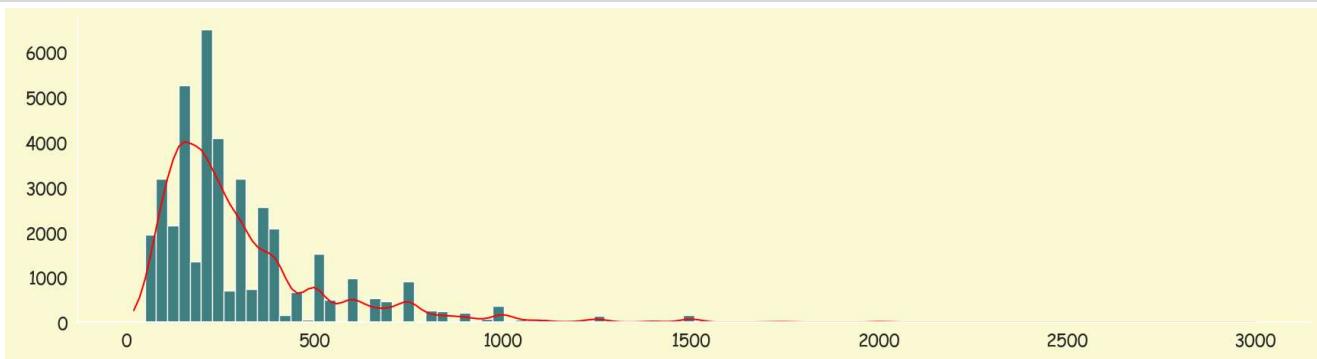


Top 10 Type Of Restaurant with Percentage

```
In [46]: df[["Type Of Restaurant"]].value_counts()[:10].plot.pie(figsize = (10, 10),  
                                         autopct = '%1.0f%%')  
plt.title("Pie Chart")  
plt.xticks(rotation = 90)  
plt.show()
```

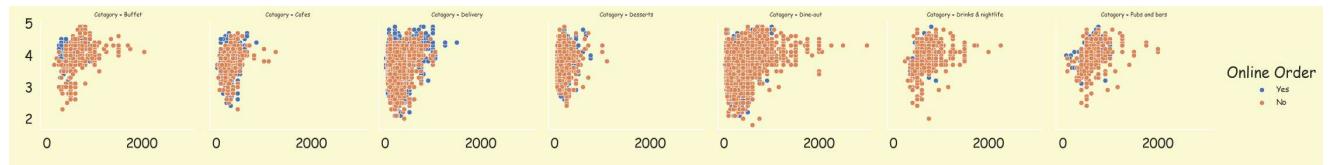


```
In [47]: sns.displot(data=df, x="Cost Per Head", kde=True, bins = 100,color = "red", facecolor = "#3F7F7F",height = 5, aspect = 3.5);
```



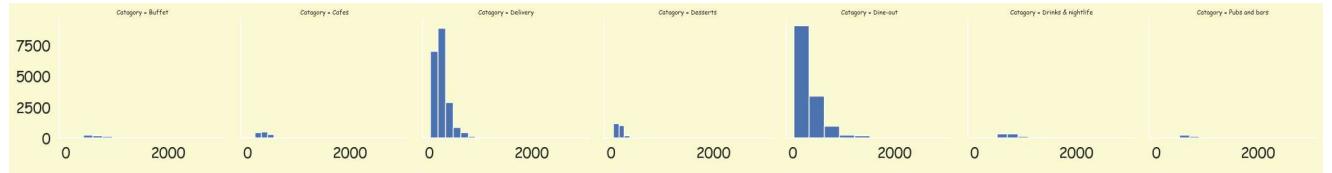
{ Scatterplot Plot for Cost Per Head Vs Catagory }

```
In [48]: g = sns.FacetGrid(df, col="Category", hue="Online Order")
g.map_dataframe(sns.scatterplot, x="Cost Per Head", y="Ratings")
g.add_legend()
plt.show()
```



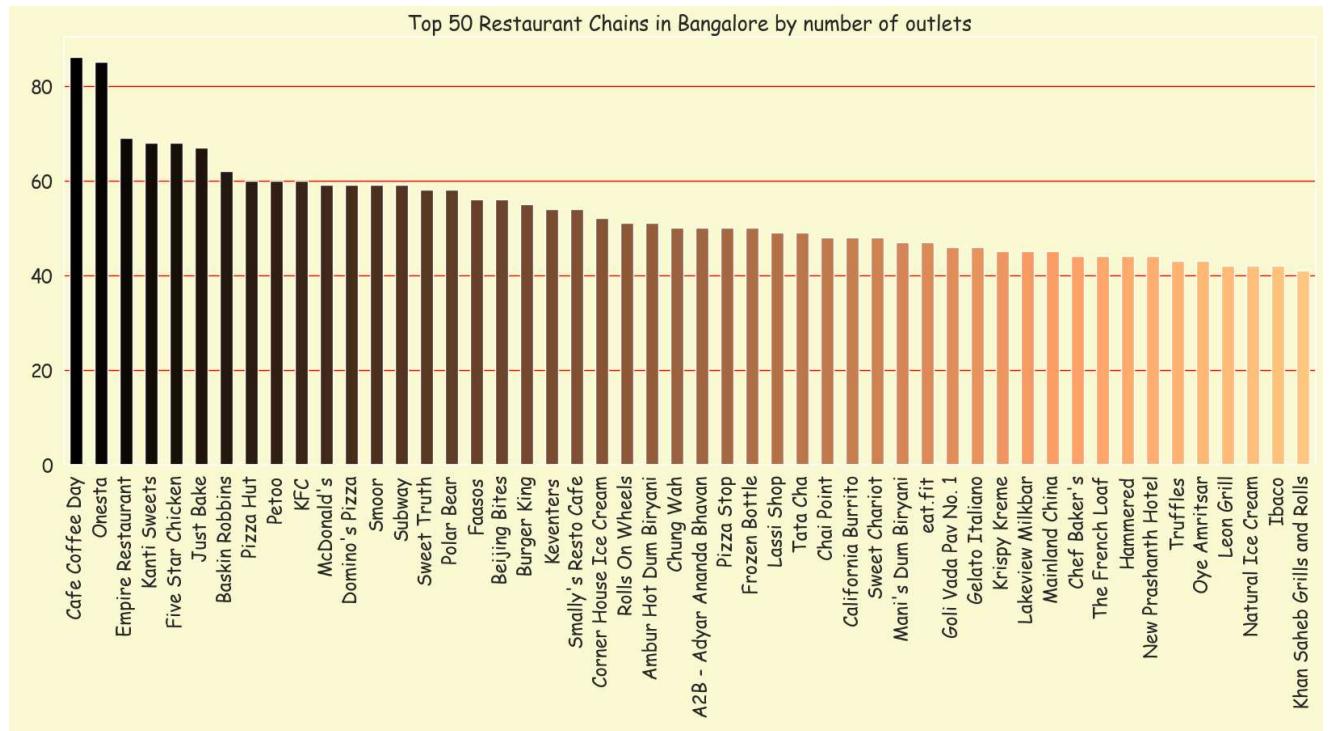
{ HistPlot for Cost Per Head Vs Catagory }

```
In [49]: g = sns.FacetGrid(df, col="Category")
g = g.map(plt.hist, "Cost Per Head")
```



{ Top 50 Restaurant Chains in Bangalore by number of outlets }

```
In [50]: plt.figure(figsize=(20,7))
df['Restaurant Name'].value_counts()[:50].plot(kind = 'bar', color = plt.cm.copper(np.linspace(0, 1, 50)))
plt.title('Top 50 Restaurant Chains in Bangalore by number of outlets')
plt.xlabel('Restaurant name')
plt.ylabel("No of Outlets")
plt.grid(axis = 'y')
plt.xticks(rotation = 90);
```



{ Mean Cost Per Head spend by Customer in Location Listed in City }

```
In [51]: df_Listed= df.groupby(['Listed in City'])['Cost Per Head'].mean().reset_index()
df_Listed.sort_values(ascending=False,by='Cost Per Head').style.background_gradient(cmap='copper').set_precision(2)
```

Out[51]:

	Listed in City	Cost Per Head
7	Church Street	424.00
5	Brigade Road	419.47
20	Lavelle Road	415.68
21	MG Road	415.36
27	Residency Road	409.74
11	Indiranagar	352.42
25	Old Airport Road	327.40
29	Whitefield	325.60
22	Malleshwaram	309.18
9	Frazer Town	305.15
4	Bellandur	298.63
28	Sarjapur Road	289.99
6	Brookefield	284.13
16	Koramangala 4th Block	282.73
17	Koramangala 5th Block	282.20
8	Electronic City	281.98
19	Koramangala 7th Block	279.95
18	Koramangala 6th Block	279.56
23	Marathahalli	268.61
14	Kalyan Nagar	266.21
0	BTM	265.27
26	Rajajinagar	264.85
10	HSR	260.25
15	Kammanahalli	260.20
24	New BEL Road	251.63
13	Jayanagar	245.75
12	JP Nagar	242.13
2	Bannerghatta Road	241.89
3	Basavanagudi	234.26
1	Banashankari	213.34

{ Maximum Cost Per Head spend by Customer in Location Listed in City }

```
In [52]: df_Listed= df.groupby(['Listed in City'])['Cost Per Head'].max().reset_index()
df_Listed.sort_values(ascending=False,by='Cost Per Head').style.background_gradient(cmap='copper').set_precision(2)
```

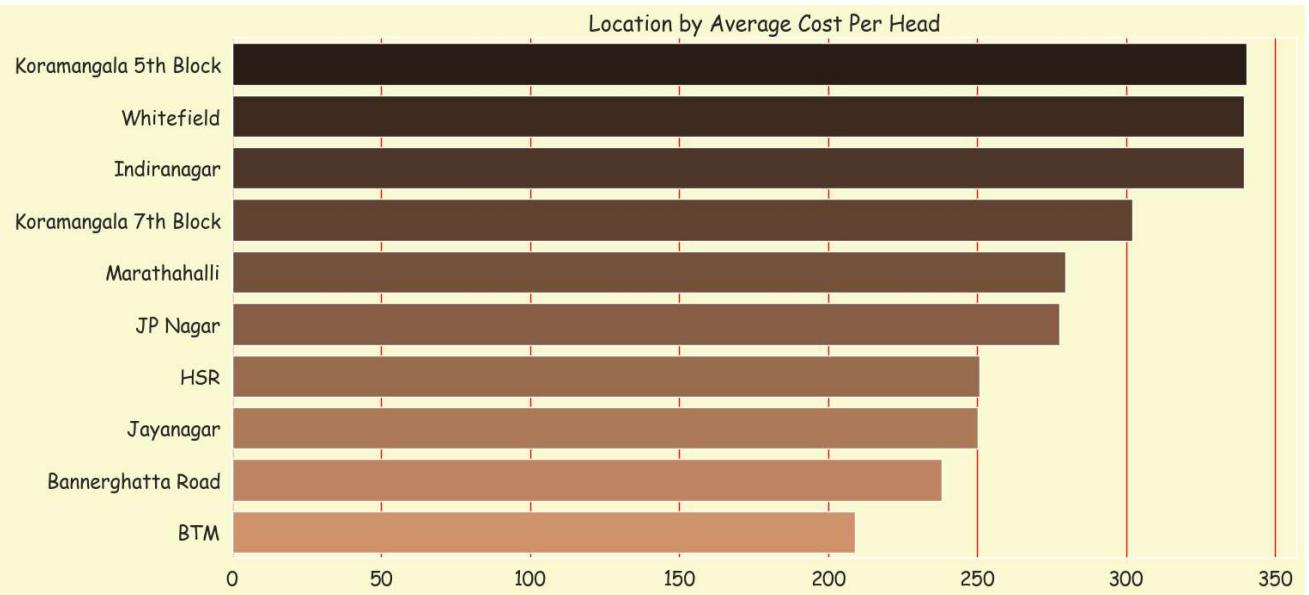
Out[52]:

	Listed in City	Cost Per Head
11	Indiranagar	3000.00
25	Old Airport Road	3000.00
22	Malleshwaram	2500.00
23	Marathahalli	2250.00
20	Lavelle Road	2050.00
27	Residency Road	2000.00
5	Brigade Road	2000.00
21	MG Road	2000.00
7	Church Street	2000.00
29	Whitefield	1500.00
16	Koramangala 4th Block	1500.00
9	Frazer Town	1500.00
8	Electronic City	1500.00
6	Brookefield	1500.00
26	Rajajinagar	1300.00
18	Koramangala 6th Block	1250.00
28	Sarjapur Road	1250.00
19	Koramangala 7th Block	1250.00
0	BTM	1250.00
17	Koramangala 5th Block	1250.00
4	Bellandur	1250.00
3	Basavanagudi	1100.00
24	New BEL Road	1000.00
10	HSR	1000.00
15	Kammanahalli	1000.00
14	Kalyan Nagar	900.00
13	Jayanagar	850.00
12	JP Nagar	850.00
2	Bannerghatta Road	850.00
1	Banashankari	750.00

BarPlot of Location by Average Cost Per Head

```
In [53]: colors=sns.color_palette('copper')[0:5]
res_chain = df['Cost Per Head'].groupby(df['Location'],sort=True)
dict_2={}
for i,j in df['Location'].value_counts()[:10].to_dict().items():
    dict_2[i]=round(res_chain.get_group(i).mean(),2)
cost_df2 = pd.DataFrame(list(dict_2.items()),columns=['Location',"Average Cost Per Head"])

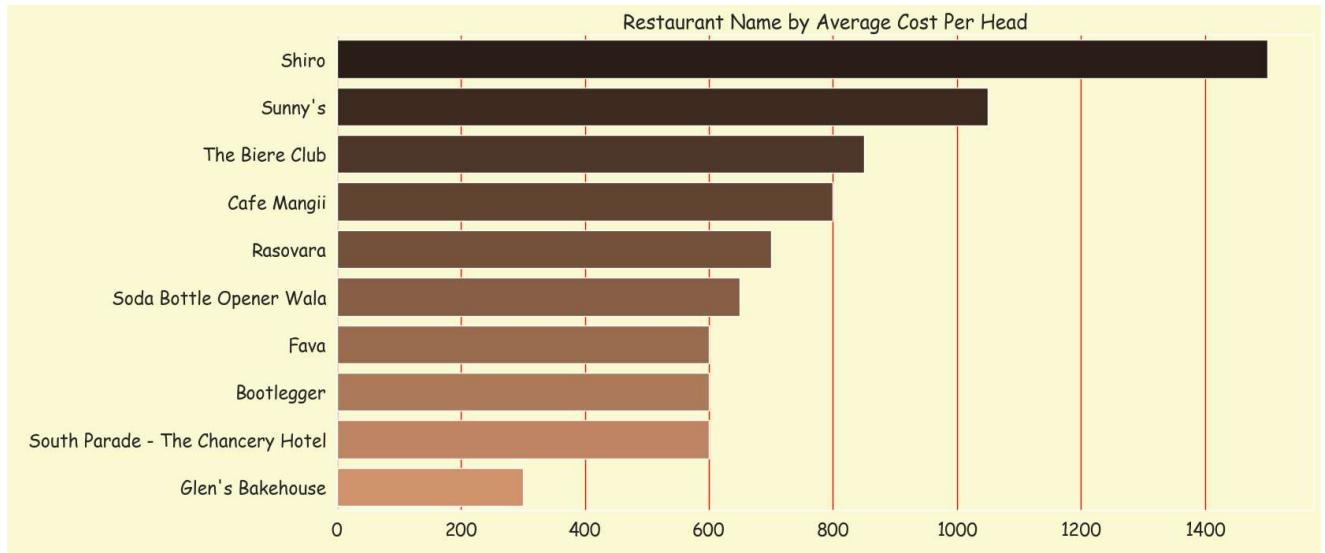
fig , ax = plt.subplots(figsize=(16,8))
sns.barplot(data = cost_df2.sort_values(by=['Average Cost Per Head'],ascending=False),
            x = 'Average Cost Per Head',y = 'Location',palette=sns.blend_palette(colors, n_colors=10))
plt.title('Location by Average Cost Per Head')
plt.grid(axis = 'x')
```



BarPlot of Restaurant Name (Lavelle Road) by Average Cost Per Head

```
In [54]: Low_cost_perhead = df[df['Location'] == "Lavelle Road"]
colors=sns.color_palette('copper')[0:5]
res_chain = Low_cost_perhead['Cost Per Head'].groupby(Low_cost_perhead['Restaurant Name'],sort=True)
dict_2={}
for i,j in Low_cost_perhead['Restaurant Name'].value_counts()[:10].to_dict().items():
    dict_2[i]=round(res_chain.get_group(i).mean(),2)
cost_df2 = pd.DataFrame(list(dict_2.items()),columns=['Restaurant Name','Average Rating'])

fig , ax = plt.subplots(figsize=(16,8))
sns.barplot(data = cost_df2.sort_values(by=['Average Rating'],ascending=False),
             x = 'Average Rating',y = 'Restaurant Name',palette=sns.blend_palette(colors, n_colors=10))
plt.title('Restaurant Name by Average Cost Per Head')
plt.grid(axis = 'x')
```



{ WordCloud for Restaurant Name (Le Rock) }

```
In [55]: Le_Rock = df[df['Restaurant Name'] == 'Le Rock']
from wordcloud import WordCloud, STOPWORDS

text = " ".join(Company for Company in Le_Rock["Catagory"])
#font = "Quicksand-Bold.ttf"
word_cloud = WordCloud(width = 2300,
                       height = 800,
                       colormap = 'jet',
                       background_color = "white").generate(text)
plt.figure(figsize = (50, 8))
plt.imshow(word_cloud, interpolation = "gaussian")
plt.axis("off")
plt.show()
```



{ WordCloud for Restaurant Name (Le Rock) for Cuisines }

```
In [56]: from wordcloud import WordCloud, STOPWORDS

text = " ".join(Company for Company in Le_Rock["Cuisines"])
#font = "Quicksand-Bold.ttf"
word_cloud = WordCloud(width = 2300,
                       height = 800,
                       colormap = 'jet',
                       background_color = "white").generate(text)
plt.figure(figsize = (50, 8))
plt.imshow(word_cloud, interpolation = "gaussian")
plt.axis("off")
plt.show()
```



{ WordCloud for Restaurant Name (Biergarten) for Cuisines }

```
In [57]: Biergarten = df[df['Restaurant Name'] == 'Biergarten']
from wordcloud import WordCloud, STOPWORDS

text = " ".join(Company for Company in Biergarten["Cuisines"])
#font = "Quicksand-Bold.ttf"
word_cloud = WordCloud(width = 2300,
                       height = 800,
                       colormap = 'jet',
                       background_color = "white").generate(text)
plt.figure(figsize = (50, 8))
plt.imshow(word_cloud, interpolation = "gaussian")
plt.axis("off")
plt.show()
```

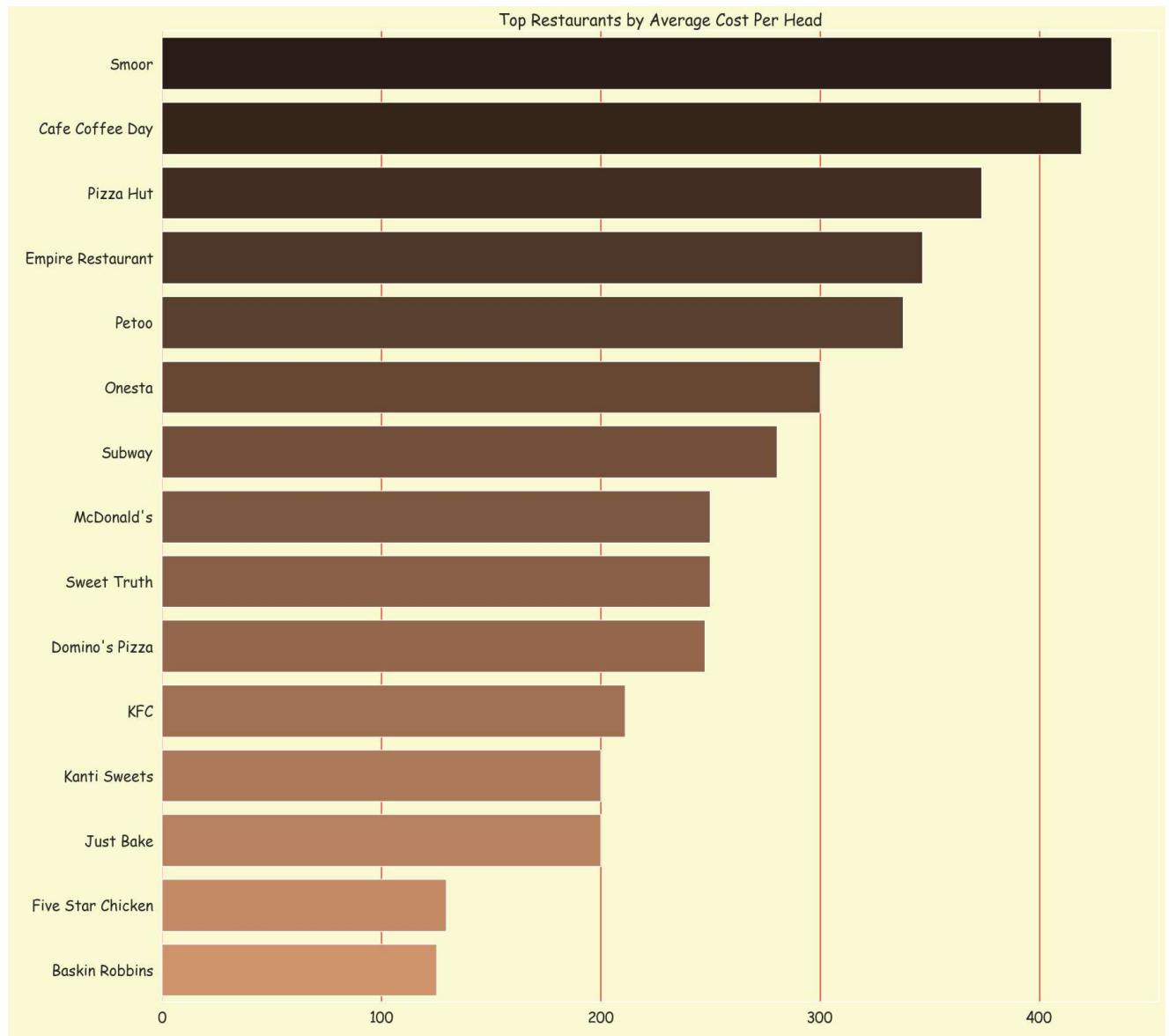


{ Top Restaurants by Average Cost Per Head }

```
In [58]: colors=sns.color_palette('copper')[0:5]
res_chain = df['Cost Per Head'].groupby(df['Restaurant Name'],sort=True)
dict_2={}
for i,j in df['Restaurant Name'].value_counts()[:15].to_dict().items():
    dict_2[i]=round(res_chain.get_group(i).mean(),2)
cost_df2 = pd.DataFrame(list(dict_2.items()),columns=['Restaurant Name',"Average Cost Per Head"])


```

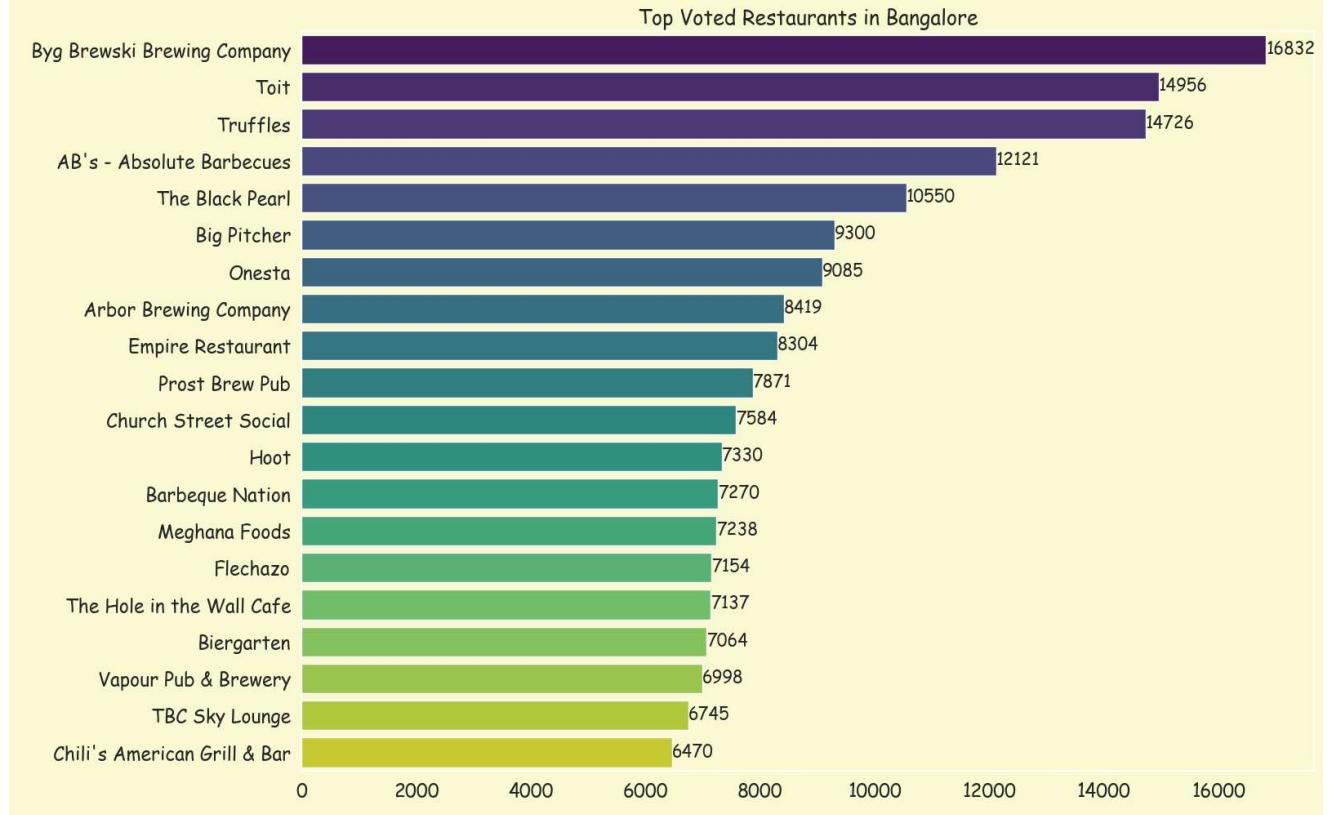
```
fig , ax = plt.subplots(figsize=(20,20))
sns.barplot(data = cost_df2.sort_values(by=['Average Cost Per Head'],ascending=False),
            x = 'Average Cost Per Head',y = 'Restaurant Name',palette=sns.blend_palette(colors, n_colors=15))
plt.title('Top Restaurants by Average Cost Per Head')
plt.grid(axis = 'x')
```



{ Top Voted Restaurants in Bangalore }

```
In [59]: vote_dataset = df.copy(deep=True)
vote_dataset = vote_dataset.sort_values(by=['Votes'], ascending=False)
vote_dataset = vote_dataset.drop_duplicates(subset=['Restaurant Name'], keep='first').reset_index()
vote_dataset = vote_dataset.head(20)

plt.figure(figsize=(16, 12))
vote_plt = sns.barplot(x='Votes', y='Restaurant Name', palette='viridis', data=vote_dataset)
vote_plt.bar_label(vote_plt.containers[0], label=vote_dataset['Votes'])
plt.title('Top Voted Restaurants in Bangalore')
plt.show()
```



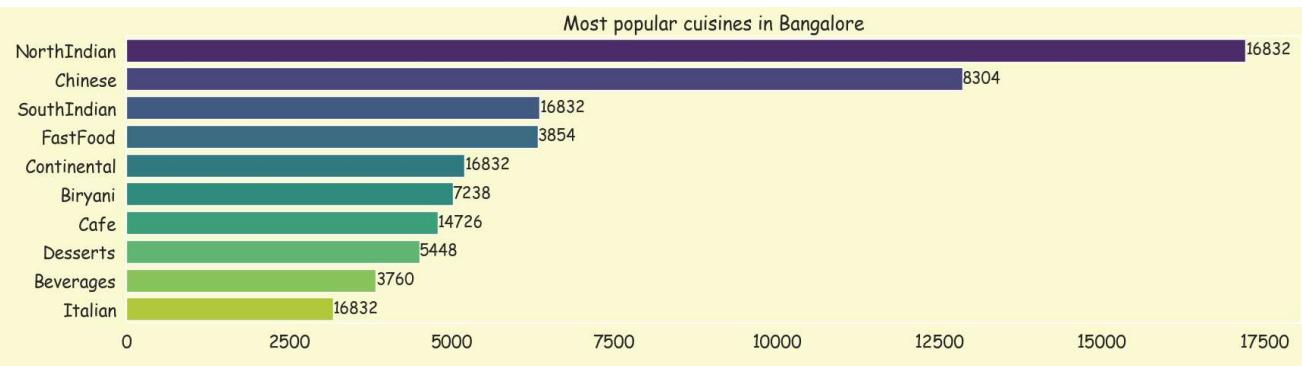
```
In [60]: cuisine_dataset = df.copy(deep=True)
cuisine_dataset['Cuisines'] = cuisine_dataset['Cuisines'].apply(lambda x:x.split(','))
cuisine_dataset = cuisine_dataset.explode('Cuisines')
cuisine_dataset['Cuisines'] = cuisine_dataset['Cuisines'].apply(lambda x:x.replace(' ', ''))
cuisine_dataset = cuisine_dataset.groupby('Cuisines').aggregate({'Votes': list, 'Location': list})
cuisine_dataset['count'] = cuisine_dataset['Votes'].apply(lambda x:len(x))
cuisine_dataset['top_votes'] = cuisine_dataset['Votes'].apply(lambda x:max(x))
cuisine_dataset = cuisine_dataset.sort_values(by=['count', 'top_votes'], ascending=False).reset_index()
cuisine_dataset = cuisine_dataset.head(10)
cuisine_dataset
```

Out[60]:

	Cuisines	Votes	Location	count	top_votes
0	NorthIndian	[775, 787, 88, 166, 286, 8, 62, 148, 506, 415,...	[Banashankari, Banashankari, Banashankari, Bas...	17212	16832
1	Chinese	[775, 787, 8, 164, 133, 144, 13, 148, 506, 91,...	[Banashankari, Banashankari, Mysore Road, Bana...	12859	8304
2	SouthIndian	[88, 8, 75, 4884, 17, 28, 618, 34, 558, 28, 39...	[Banashankari, Mysore Road, Banashankari, Jaya...	6357	16832
3	FastFood	[133, 180, 28, 62, 4, 23, 148, 506, 415, 91, 1...	[Banashankari, Banashankari, Banashankari, Ban...	6324	3854
4	Continental	[324, 150, 164, 424, 133, 13, 36, 1047, 627, 8...	[Banashankari, Banashankari, Banashankari, Ban...	5197	16832
5	Biryani	[415, 230, 804, 618, 627, 104, 163, 6, 520, 28...	[Banashankari, Banashankari, Kumaraswamy Layou...	5018	7238
6	Cafe	[918, 2556, 324, 504, 402, 150, 164, 424, 918,...	[Banashankari, Banashankari, Banashankari, Ban...	4785	14726
7	Desserts	[35, 345, 17, 39, 146, 66, 7, 71, 13, 8, 25, 4...	[Banashankari, Banashankari, Banashankari, Ban...	4506	5448
8	Beverages	[504, 180, 4, 28, 17, 71, 146, 71, 175, 438, 2...	[Banashankari, Banashankari, Banashankari, Vij...	3836	3760
9	Italian	[918, 2556, 324, 504, 150, 164, 918, 144, 93, ...	[Banashankari, Banashankari, Banashankari, Ban...	3177	16832

(Most popular cuisines in Bangalore)

```
In [61]: plt.figure(figsize=(20, 5))
cuisine_plt = sns.barplot(x='count', y='Cuisines', data=cuisine_dataset, palette='viridis')
cuisine_plt.bar_label(cuisine_plt.containers[0], labels=cuisine_dataset['top_votes'])
plt.title('Most popular cuisines in Bangalore')
plt.show()
```



```
In [62]: fig = plt.figure(figsize=(20,20))
sns.boxplot(data=df, y="Location", x="Cost Per Head",color="#4C72B0")
plt.grid(True, color = "grey", linewidth = "0.5", alpha=0.5, axis="both")
plt.ylabel("")
plt.show()
```



```
In [63]: from IPython.core.display import HTML

def value_counts_all(df, columns):
    pd.set_option('display.max_rows', 50)
    table_list = []
    for col in columns:
        table_list.append(pd.DataFrame(df[col].value_counts()))
    return HTML(
        f"<table><tr> {''.join(['<td>' + table._repr_html_() + '</td>' for table in table_list])}</tr></table>")
value_counts_all(df, ["Location","Listed in City","Restaurant Name","Cuisines","Catagory"])

Out[63]:
```

		Listed in City							
		BTM	2586						
		Koramangala 7th Block	2361						
		Koramangala 4th Block	2254						
		Koramangala 5th Block	2250						
		Koramangala 6th Block	2121						
		Jayanagar	1918			Restaurant Name		Cuisines	
		JP Nagar	1633	Cafe	Coffee Day	86	North Indian	2108	
		Indiranagar	1538	Onesta	Empire Restaurant	85	North Indian, Chinese	1953	
		Church Street	1518	Bakery, Desserts		1232	South Indian	1232	
		MG Road	1514	Cafe		620	Cafe	620	
		Brigade Road	1483	Bakery, Desserts		613	Delivery	20452	
		Lavelle Road	1451	Desserts		...	Dine-out	14066	
		Residency Road	1345	North Indian, Mughlai, Lucknowi		...	Cafes	1511	
		HSR	1330	Continental, Thai, North Indian, Chinese		1	Drinks & nightlife	1045	
		Marathahalli	1294	North Indian, Bengali, Chinese, Beverages		1	Buffet	848	
		Bannerghatta Road	1208	Biryani, North Indian, Chinese, Kebab		1	Pubs and bars	632	
		Whitefield	1201	North Indian, Chinese, Arabian, Momos		1			
		Old Airport Road	1173			2367 rows × 1 columns			
		Brookefield	1139						
		Basavanagudi	1072						
		Kammanahalli	1020						
		Sarjapur Road	1018						
		Kalyan Nagar	993						
		Bellandur	956						
		Frazer Town	953						
		Malleshwaram	946						
		Rajajinagar	872						
		Electronic City	820						
		Banashankari	727						
		New BEL Road	569						

(Top 15 Best Liked cuisines in Bangalore)

```
In [64]: print('\033[7m'+'.: Cuisines :')
print('\033[0m\033[43m*' * 47 +'\033[0m')
df['Cuisines'].value_counts()[0:15]

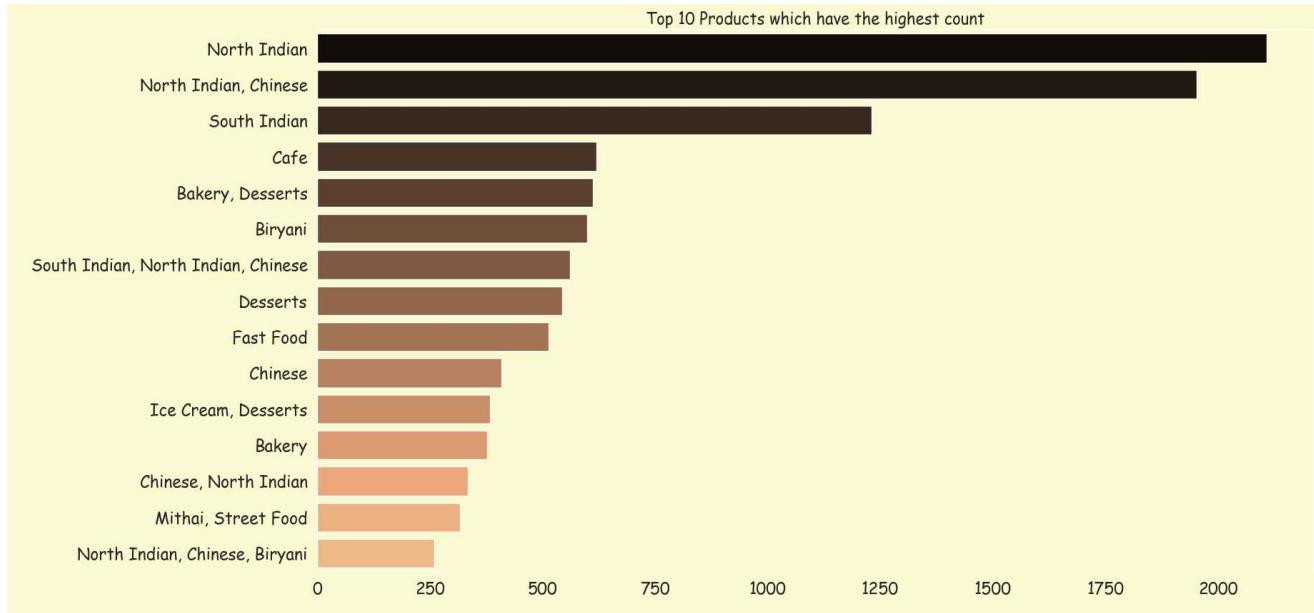
.: Cuisines :
*****
```

```
Out[64]: North Indian          2108
North Indian, Chinese      1953
South Indian             1232
Cafe                      620
Bakery, Desserts        613
Biryani                  600
South Indian, North Indian, Chinese 561
Desserts                 545
Fast Food                 514
Chinese                   410
Ice Cream, Desserts     384
Bakery                   377
Chinese, North Indian    335
Mithai, Street Food      317
North Indian, Chinese, Biryani 260
Name: Cuisines, dtype: int64
```

```
In [65]: # Top 15 Cuisines which have the highest Order

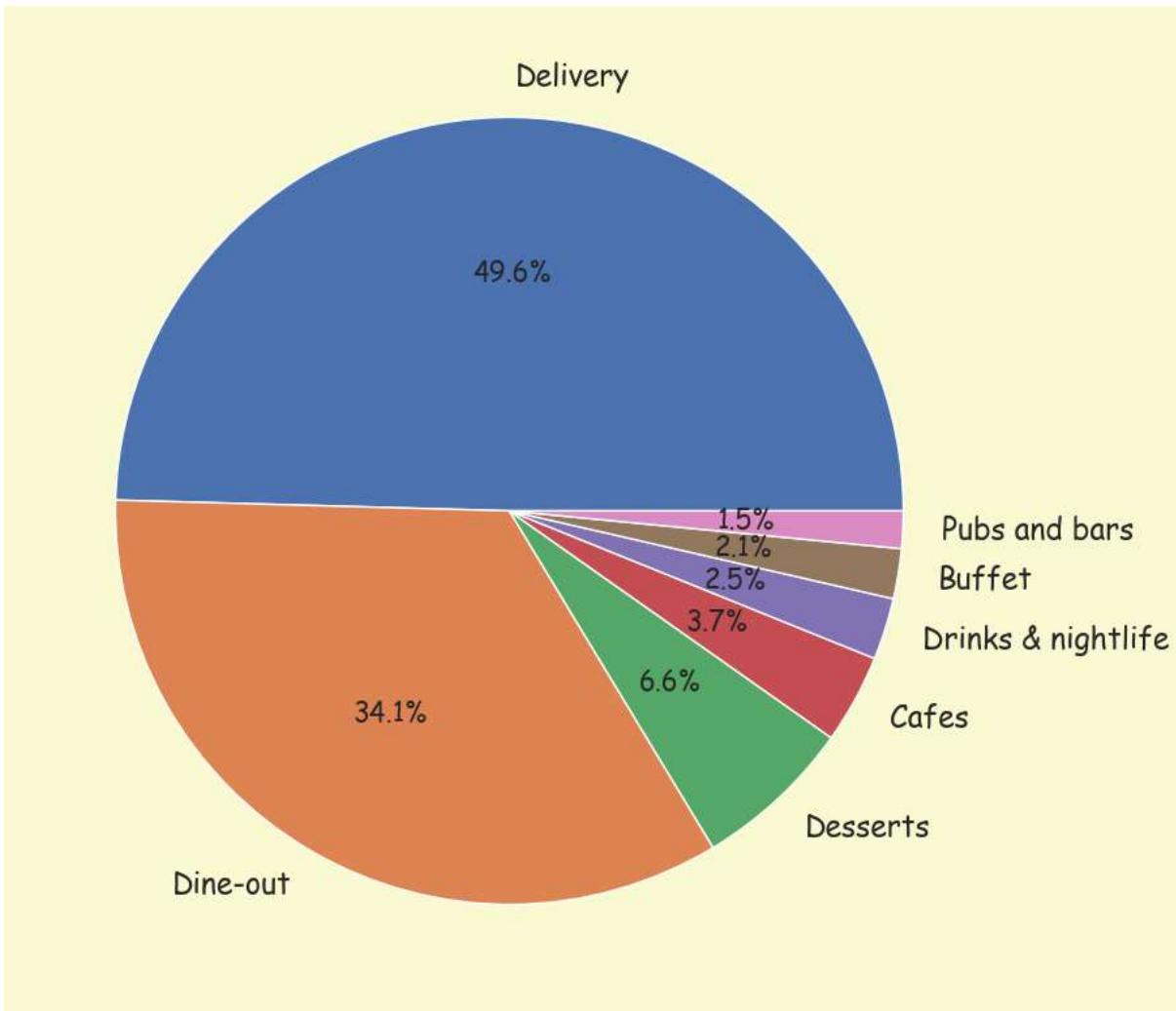
plt.figure(figsize=(18, 10))
sns.countplot(y='Cuisines', data=df, order=df['Cuisines'].value_counts().head(15).index, palette='copper')
plt.title('Top 10 Products which have the highest count', fontsize=16)
plt.xlabel('Count', fontsize=12)
plt.ylabel('Product Name', fontsize=12)

plt.show()
```



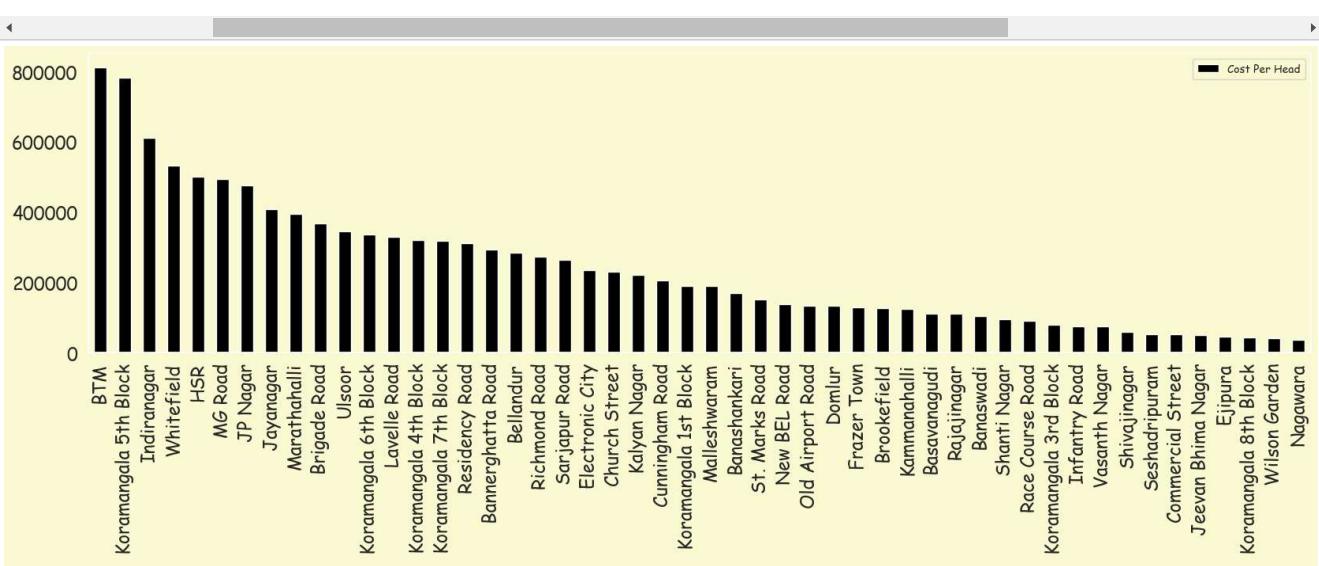
{ PieChat for Category of Food Bangalore }

```
In [66]: plt.figure(figsize=(20,10))
plt.pie(df['Catagory'].value_counts(),labels=df["Catagory"].value_counts().keys(),autopct="%0.1f%%")
plt.show()
```



Top 50 Location in Bangalore with total Collection (Cost per Head)

```
In [67]: ['Cost Per Head']].sort_values(by="Cost Per Head",ascending=False).nlargest(n=50, columns=['Cost Per Head']).plot.bar(color=plt.c
```



```
In [68]: top_sp=pd.DataFrame(df.groupby(['Location']).sum()['Cost Per Head'])
top = top_sp.sort_values('Cost Per Head',ascending=False)[:10]
top.head(10).style.background_gradient(cmap='copper').set_precision(2)
```

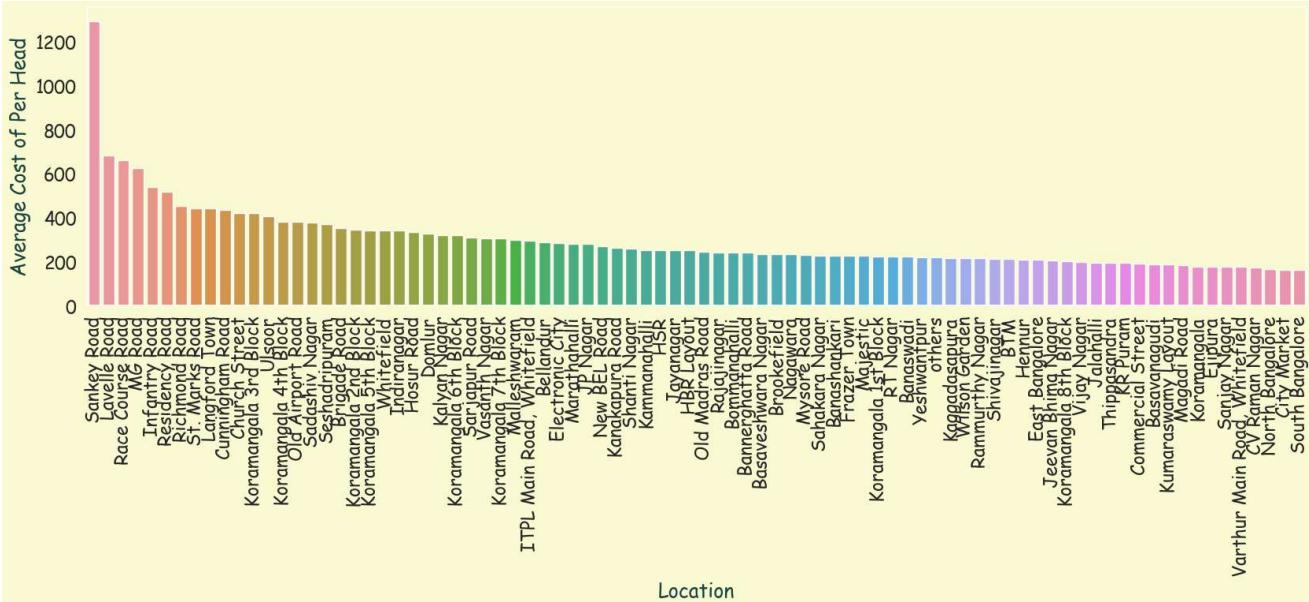
Out[68]:

Cost Per Head	
Location	
BTM	810780.00
Koramangala 5th Block	781800.00
Indiranagar	611520.00
Whitefield	532750.00
HSR	499735.00
MG Road	493450.00
JP Nagar	474875.00
Jayanagar	408825.00
Marathahalli	394175.00
Brigade Road	367870.00

Top 50 Location in Bangalore with Average Cost per Head

```
In [69]: # In which Location the Cost Per Head is highest
regions= df.groupby(['Location']).mean().sort_values('Cost Per Head', ascending=False)
regions.reset_index(inplace=True)

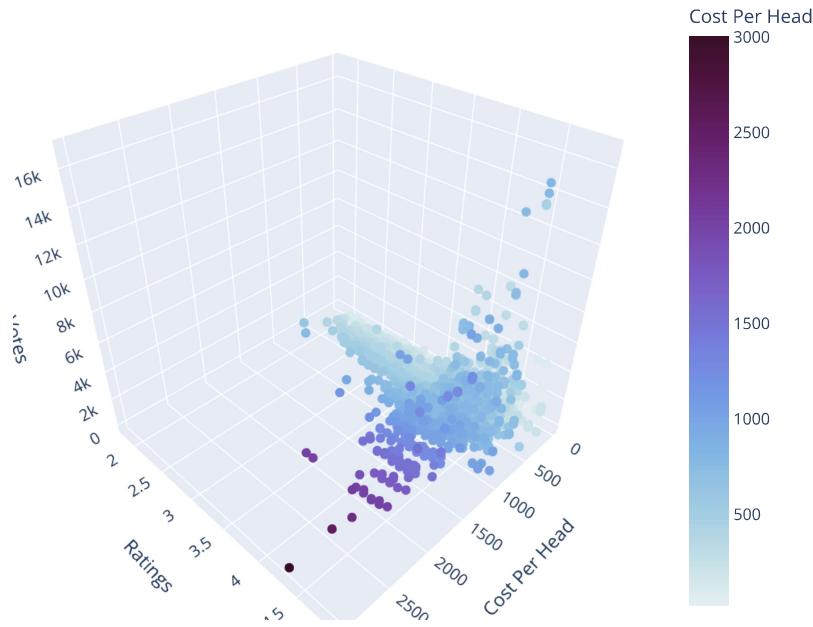
fig, ax = plt.subplots(figsize=(20, 5))
sns.barplot(x='Location', y='Cost Per Head', data=regions)
plt.ylabel("Average Cost of Per Head", fontsize=18,color="#1a4441",font='Comic Sans MS');
plt.xlabel("Location",fontsize=18,color="#1a4441",font='Comic Sans MS');
plt.xticks(rotation=90)
plt.show();
```



```
In [70]: import plotly.graph_objects as go
import plotly.express as px
from PIL import Image, ImageEnhance; Image.MAX_IMAGE_PIXELS = 5_000_000_000;
import plotly
import PIL

fig = px.scatter_3d(df, x='Cost Per Head', y='Ratings', z='Votes', color="Cost Per Head",
                     opacity=1, width=700, height=650,
                     color_continuous_scale="dense",
                     title="Cost Per Head")
fig.update_traces(marker_size=4)
fig.show()
```

Cost Per Head



Summary of Top 15 Cuisines in Bangalore

```
In [71]: Summary_df = pd.DataFrame({'Cuisines':df['Cuisines'].value_counts().sort_index().index,
                                'Ratings':df.groupby(['Cuisines'])['Ratings'].mean().sort_index().values,
                                'Votes':df.groupby(['Cuisines'])['Votes'].mean().sort_index().values,
                                'Cost Per Head Mean': df.groupby(['Cuisines'])['Cost Per Head'].mean().sort_index().values,
                                'Cost Per Head Sum': df.groupby(['Cuisines'])['Cost Per Head'].sum().sort_index().values,
                                })
Summary_df.sort_values(by='Cost Per Head Mean', ascending=False)[:15].style.background_gradient(cmap='copper').set_precision(2)
```

Out[71]:

	Cuisines	Ratings	Votes	Cost Per Head Mean	Cost Per Head Sum
1237	French, Italian	4.30	127.00	3000.00	6000.00
1808	North Indian, Continental, Japanese, Chinese, South Indian	4.10	309.00	2050.00	8200.00
2277	Steak, Mediterranean, Grill	3.44	12.00	2000.00	10000.00
1856	North Indian, Kashmiri, Mughlai	4.00	134.00	1850.00	1850.00
1485	Mangalorean, Konkan, Seafood, Kerala	4.50	678.60	1750.00	8750.00
1846	North Indian, Italian, Asian, Continental, Mediterranean	4.20	593.25	1750.00	7000.00
2344	Thai, Japanese, Chinese	4.15	246.00	1750.00	3500.00
1818	North Indian, European, Thai	4.30	618.10	1700.00	17000.00
1849	North Indian, Italian, Chinese, Japanese	4.30	429.00	1700.00	5100.00
1607	North Indian, Asian, South Indian, Italian, European	4.20	471.00	1500.00	6000.00
187	Asian, Japanese, Thai	4.40	877.22	1500.00	13500.00
40	American, North Indian, European, Tex-Mex	4.10	116.00	1500.00	9000.00
1729	North Indian, Chinese, Mediterranean	3.80	619.73	1500.00	16500.00
1000	Continental, North Indian, Italian, Chinese	4.00	344.00	1500.00	3000.00
1088	European, Mediterranean, Salad	4.10	94.50	1500.00	3000.00

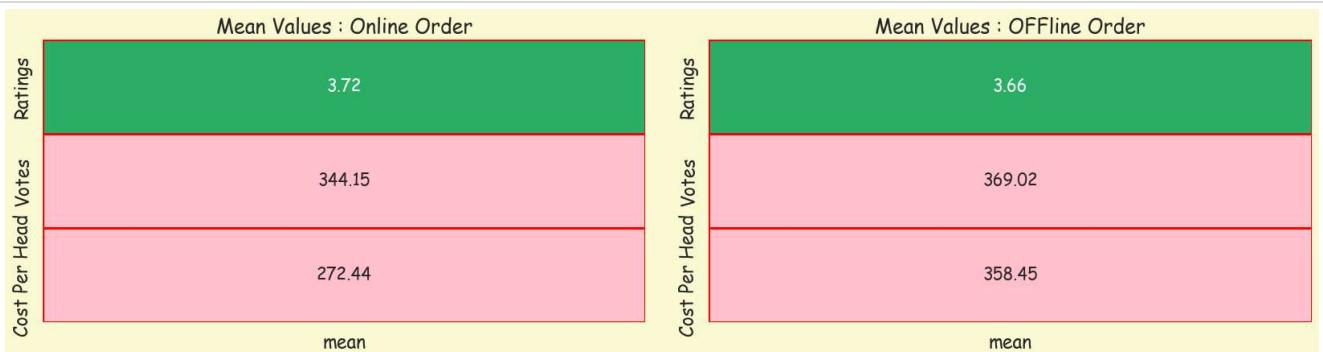
```
In [72]: yes = df[df['Online Order'] == 'Yes'].describe().T
no = df[df['Online Order'] == 'No'].describe().T

colors = ['#2BAE66','pink']

fig,ax = plt.subplots(nrows = 1,ncols = 2,figsize = (17,5))
plt.subplot(1,2,1)
sns.heatmap(yes[['mean']],annot = True,cmap = colors,linewidths = 0.8,linecolor = 'Red',cbar = False,fmt = '.2f')
plt.title('Mean Values : Online Order');

plt.subplot(1,2,2)
sns.heatmap(no[['mean']],annot = True,cmap = colors,linewidths = 0.8,linecolor = 'Red',cbar = False,fmt = '.2f')
plt.title('Mean Values : Offline Order');

fig.tight_layout(pad = 2)
```



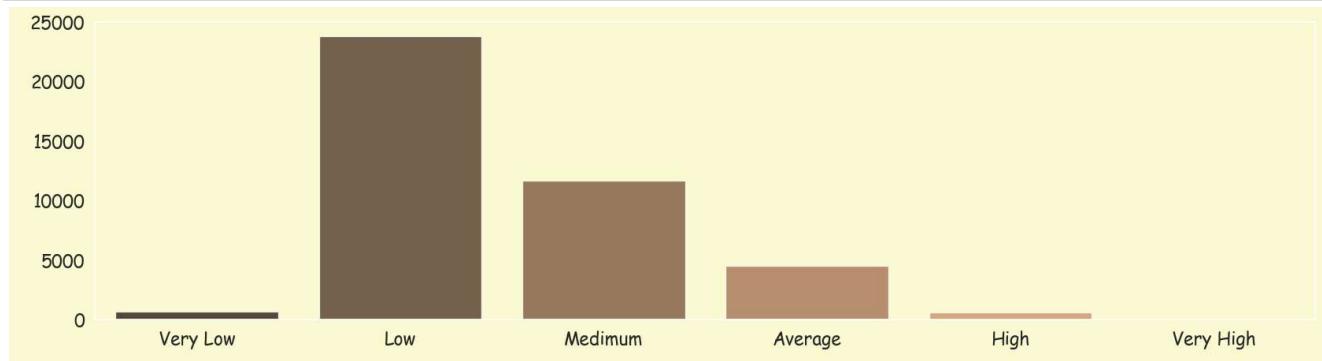
{ Creating New Feature Cost Per Head Range }

```
In [73]: # Cost Per Head Range
df['Cost_Per_Head_Range'] = pd.cut(df['Cost Per Head'],
bins=[0,50,250,500,1000,2000,3000],
labels=['Very Low','Low','Medium','Average','High','Very High'])
df['Cost_Per_Head_Range'].value_counts(ascending = False).to_frame().style.background_gradient(cmap='copper').set_precision(2)
```

Out[73]:

Cost_Per_Head_Range	
Low	23810
Medium	11666
Average	4518
Very Low	652
High	608
Very High	9

```
In [74]: plt.figure(figsize=(20,5))
sns.countplot(df['Cost_Per_Head_Range'], palette="copper", alpha=.8);
```



(Creating New Feature Ratings_Grade)

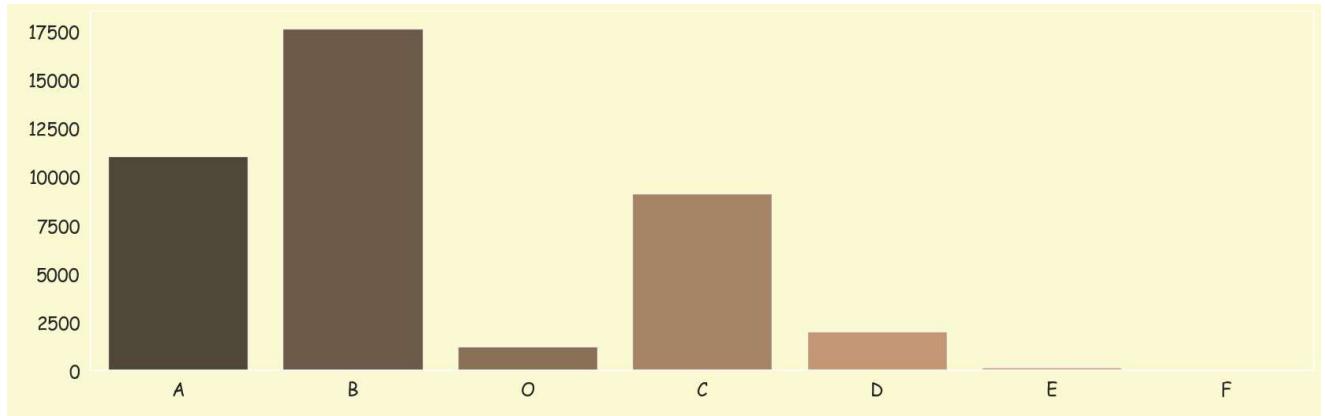
```
In [75]: def Grade(Ratings):
    if Ratings >= 4.5:
        return 'O'
    if Ratings >= 4:
        return 'A'
    if Ratings >= 3.5:
        return 'B'
    if Ratings >= 3:
        return 'C'
    if Ratings >= 2.5:
        return 'D'
    if Ratings >= 2.0:
        return 'E'
    return 'F'

df['Ratings_Grade']=df.apply(lambda x: Grade(x['Ratings']),axis=1)
df.head().style.background_gradient(cmap='copper').set_precision(2)
```

Out[75]:

	Restaurant Name	Online Order	Book Table	Ratings	Votes	Location	Type Of Restaurant	Cuisines	Cost Per Head	Catagory	Listed in City	Cost_Per_Head_Range	Ratings_Grade
0	Jalsa	Yes	Yes	4.10	775	Banashankari	Casual Dining	North Indian, Mughlai, Chinese	400.00	Buffet	Banashankari	Medium	A
1	Spice Elephant	Yes	No	4.10	787	Banashankari	Casual Dining	Chinese, North Indian, Thai	400.00	Buffet	Banashankari	Medium	A
2	San Churro Cafe	Yes	No	3.80	918	Banashankari	Cafe, Casual Dining	Cafe, Mexican, Italian	400.00	Buffet	Banashankari	Medium	B
3	Addhuri Udupi Bhojana	No	No	3.70	88	Banashankari	Quick Bites	South Indian, North Indian	150.00	Buffet	Banashankari	Low	B
4	Grand Village	No	No	3.80	166	Basavanagudi	Casual Dining	North Indian, Rajasthani	300.00	Buffet	Banashankari	Medium	B

```
In [76]: plt.figure(figsize=(20,6))
sns.countplot(df['Ratings_Grade'], palette="copper", alpha=.8);
```



```
In [77]: pd.set_option('display.float_format', '{:.2f}'.format)
```

```
# calculate the total views per event
df.groupby('Location')['Cost Per Head'].agg(['count', 'mean', 'sum']).sort_values(by='sum', ascending=False).head(10).style.background
```

```
Out[77]:
```

Location	count	mean	sum
BTM	3879	209.02	810780.00
Koramangala 5th Block	2297	340.36	781800.00
Indiranagar	1800	339.73	611520.00
Whitefield	1568	339.76	532750.00
HSR	1993	250.75	499735.00
MG Road	793	622.26	493450.00
JP Nagar	1710	277.70	474875.00
Jayanagar	1634	250.20	408825.00
Marathahalli	1410	279.56	394175.00
Brigade Road	1052	349.69	367870.00

```
In [78]: pd.set_option('display.float_format', '{:.2f}'.format)
```

```
# calculate the total views per event
df.groupby('Catagory')['Cost Per Head'].agg(['count', 'mean', 'sum']).sort_values('sum').head(10).style.background_gradient(cmap=
```

```
Out[78]:
```

Catagory	count	mean	sum
Pubs and bars	632	668.35	422400.00
Cafes	1511	323.21	488375.00
Desserts	2709	204.56	554145.00
Buffet	848	658.31	558250.00
Drinks & nightlife	1045	727.39	760125.00
Dine-out	14066	327.70	4609450.00
Delivery	20452	247.75	5066908.00

```
In [79]: df2=df.groupby('Restaurant Name')[['Catagory','Cuisines','Cost_Per_Head_Range','Listed in City','Online Order','Location','Type Of Restaurant','Ratings_Grade']]
df2[:10].style.background_gradient(cmap='copper').set_precision(2)
```

Out[79]:

	Restaurant Name	Catagory	Cuisines	Cost_Per_Head_Range	Listed in City	Online Order	Location	Type Of Restaurant	Ratings_Grade
0	Chung Wah	2	7	2	27	2	14	3	4
1	Sri Krishna Sagar	2	7	1	16	2	9	3	2
2	New Shanthi Sagar	3	6	2	12	2	10	3	3
3	Imperial Restaurant	2	6	1	17	1	7	1	2
4	Lassi Shop	2	6	2	29	2	21	4	3
5	Meghana Foods	2	5	1	17	1	4	1	2
6	Subway	2	5	2	30	2	22	3	5
7	Andhra Ruchulu	2	5	2	13	2	6	4	3
8	Smoke - The Sizzler House	2	5	2	16	2	6	2	4
9	A2B - Adyar Ananda Bhavan	2	5	2	26	2	19	2	4

{ Top 10 Most Liked Cuisines in Bangalore }

```
In [80]: print(f"\u033[034m\u033[1m")
sport=df['Cuisines'].value_counts()[:10]
print(sport)
```

```
North Indian          2108
North Indian, Chinese 1953
South Indian          1232
Cafe                  620
Bakery, Desserts      613
Biryani                600
South Indian, North Indian, Chinese 561
Desserts              545
Fast Food              514
Chinese                410
Name: Cuisines, dtype: int64
```

```
In [81]: Dine_out=df[df['Catagory'] == 'Dine-out']['Location'].value_counts().sort_values(ascending=False).head()
Dine_out
```

```
Out[81]: BTM            1195
Koramangala 5th Block  790
Whitefield            590
JP Nagar              555
Jayanagar              486
Name: Location, dtype: int64
```

```
In [82]: Delivery=df[df['Catagory'] == 'Delivery']['Location'].value_counts().sort_values(ascending=False).head()
Delivery
```

```
Out[82]: BTM            2438
HSR              1351
Koramangala 5th Block  966
Indiranagar          963
Jayanagar            904
Name: Location, dtype: int64
```

```
In [83]: Buffet=df[df['Catagory'] == 'Buffet']['Location'].value_counts().sort_values(ascending=False).head()
Buffet
```

```
Out[83]: Richmond Road    63
Koramangala 5th Block  60
MG Road              51
JP Nagar              42
Indiranagar           36
Name: Location, dtype: int64
```

```
In [84]: City=df.groupby(['Listed in City', 'Online Order'])['Catagory'].count().reset_index().head(10)  
City
```

Out[84]:

	Listed in City	Online Order	Catagory
0	BTM	No	732
1	BTM	Yes	1854
2	Banashankari	No	212
3	Banashankari	Yes	515
4	Bannerghatta Road	No	348
5	Bannerghatta Road	Yes	860
6	Basavanagudi	No	337
7	Basavanagudi	Yes	735
8	Bellandur	No	251
9	Bellandur	Yes	705

```
In [85]: df.groupby('Cuisines')['Cost Per Head'].count().nlargest(20).reset_index().head(10).style.background_gradient(cmap='copper').set_
```

Out[85]:

	Cuisines	Cost Per Head
0	North Indian	2108
1	North Indian, Chinese	1953
2	South Indian	1232
3	Cafe	620
4	Bakery, Desserts	613
5	Biryani	600
6	South Indian, North Indian, Chinese	561
7	Desserts	545
8	Fast Food	514
9	Chinese	410

```
In [86]: Online= df.groupby(['Online Order','Catagory'])[['Cost Per Head','Ratings']].mean().reset_index()  
Online.style.background_gradient(cmap='copper').set_precision(2)
```

Out[86]:

	Online Order	Catagory	Cost Per Head	Ratings
0	No	Buffet	706.10	3.91
1	No	Cafes	326.66	3.80
2	No	Delivery	235.93	3.53
3	No	Desserts	194.34	3.68
4	No	Dine-out	371.50	3.64
5	No	Drinks & nightlife	734.85	3.97
6	No	Pubs and bars	680.48	3.96
7	Yes	Buffet	589.66	4.09
8	Yes	Cafes	320.52	3.93
9	Yes	Delivery	250.56	3.69
10	Yes	Desserts	213.36	3.87
11	Yes	Dine-out	289.52	3.72
12	Yes	Drinks & nightlife	701.39	4.17
13	Yes	Pubs and bars	639.05	4.18

```
In [87]: pd.crosstab(df['Location'],df['Online Order']).style.background_gradient(cmap='copper').set_precision(2)
```

Bommanahalli	53	91
Brigade Road	442	610
Brookefield	159	384
CV Raman Nagar	18	43
Church Street	212	334
City Market	48	27
Commercial Street	146	124
Cunningham Road	156	319
Domlur	188	218
East Bangalore	9	19
Ejipura	84	172
Electronic City	364	470
Frazer Town	196	370

```
In [88]: x=pd.DataFrame(pd.pivot_table(df,index=['Online Order','Catagory'],aggfunc='count')['Cost Per Head'])  
x.style.background_gradient(cmap='copper').set_precision(2)
```

Out[88]:

Cost Per Head		
Online Order	Catagory	
	Buffet	500
	Cafes	663
	Delivery	3934
No	Desserts	1254
	Dine-out	6551
	Drinks & nightlife	812
	Pubs and bars	447
	Buffet	348
	Cafes	848
	Delivery	16518
Yes	Desserts	1455
	Dine-out	7515
	Drinks & nightlife	233
	Pubs and bars	185

```
In [89]: df.pivot_table(index="Restaurant Name")[:10].sort_values(by="Cost Per Head",ascending=False).style.background_gradient(cmap='copper')
```

Out[89]:

Restaurant Name	Cost Per Head	Ratings	Votes
12th Main - Grand Mercure	1000.00	4.10	354.10
1131 Bar + Kitchen	750.00	4.50	2823.00
1522 - The Pub	700.00	4.21	1646.88
1441 Pizzeria	400.00	4.10	132.80
100°C	225.00	3.70	41.00
#L-81 Cafe	200.00	3.90	48.00
#refuel	200.00	3.70	37.00
1000 B.C	150.00	3.20	49.00
11 to 11 Express Biriyani	150.00	3.50	22.00
#FeelTheROLL	100.00	3.40	7.00

```
In [90]: df.pivot_table(index="Cuisines")[:10].sort_values(by="Cost Per Head", ascending=False).style.background_gradient(cmap='copper').se
```

Out[90]:

Cuisines	Cost Per Head	Ratings	Votes
American, BBQ	1250.00	4.50	5276.58
American, Burger, Italian, Steak	850.00	3.90	1338.00
American, Cafe, Continental	850.00	4.40	710.71
American, Asian, European, North Indian	700.00	4.30	3868.75
American, Asian, Continental, North Indian, South Indian, Chinese	600.00	4.10	326.67
African, Burger	500.00	4.59	457.82
African, Burger, Desserts, Beverages, Fast Food	500.00	4.60	595.00
American	358.93	3.56	49.64
American, Burger, Fast Food	200.00	4.21	289.00
American, Burger, Momos, Bengali	200.00	3.50	9.00

```
In [91]: df.groupby('Location')[['Cost Per Head']].mean().sort_values(by='Cost Per Head', ascending=False).reset_index().head(10).style.back
```

Out[91]:

	Location	Cost Per Head
0	Sankey Road	1291.35
1	Lavelle Road	682.69
2	Race Course Road	660.74
3	MG Road	622.26
4	Infantry Road	536.43
5	Residency Road	515.02
6	Richmond Road	451.61
7	St. Marks Road	441.84
8	Langford Town	441.67
9	Cunningham Road	433.58

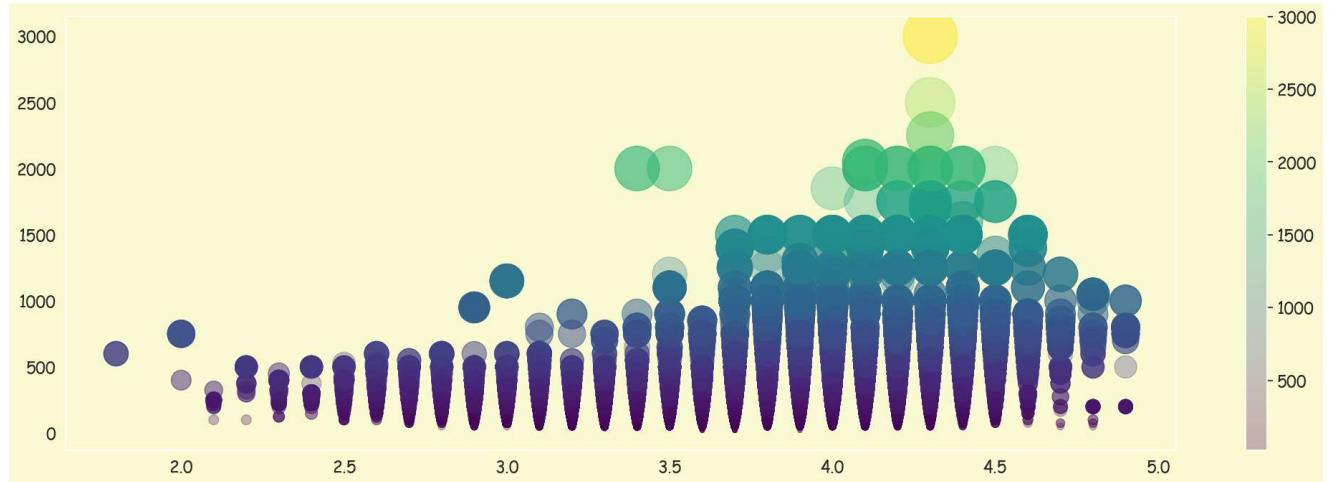
```
In [92]: df.groupby('Location')[['Cost Per Head']].sum().sort_values(by='Cost Per Head', ascending=False).reset_index().head(10).style.back
```

Out[92]:

	Location	Cost Per Head
0	BTM	810780.00
1	Koramangala 5th Block	781800.00
2	Indiranagar	611520.00
3	Whitefield	532750.00
4	HSR	499735.00
5	MG Road	493450.00
6	JP Nagar	474875.00
7	Jayanagar	408825.00
8	Marathahalli	394175.00
9	Brigade Road	367870.00

```
In [93]: x = df["Ratings"]
y = df["Cost Per Head"]
colors = df["Cost Per Head"]
sizes = df["Cost Per Head"]

plt.figure(figsize = (25, 8))
plt.ticklabel_format(style = 'plain')
plt.scatter(x, y, c = colors, s = sizes, alpha = 0.3, cmap = 'viridis')
plt.colorbar();
```

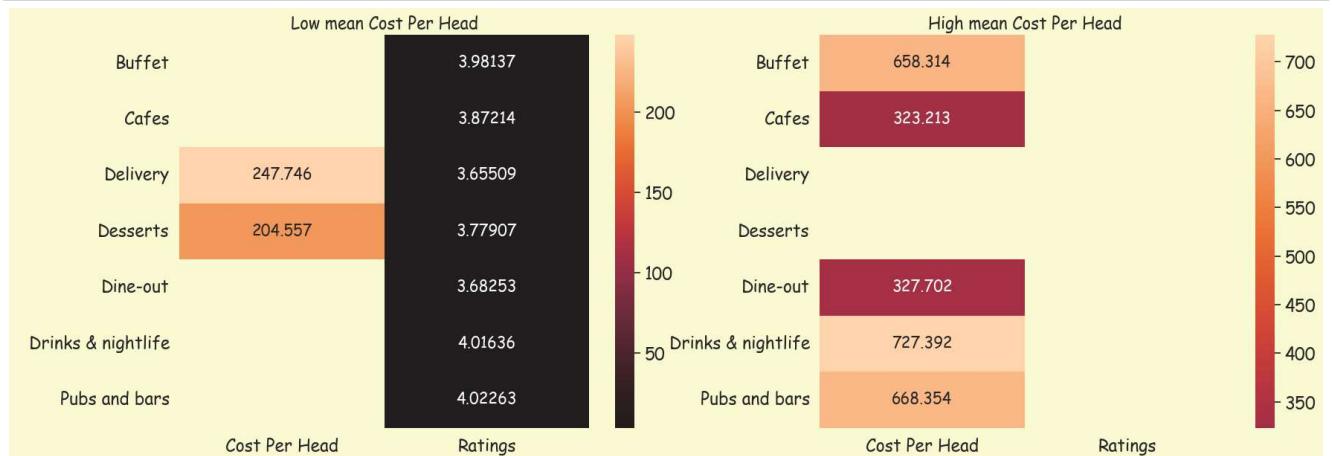


(Low mean Cost Per Head & High mean Cost Per Head)

```
In [94]: category = df.groupby('Catagory')[['Cost Per Head', 'Ratings']].agg('mean')

low = category[category < 300]
high = category[category > 300]

fig = plt.figure(figsize=(20,7), dpi=100)
fig.subplots_adjust(wspace=0.25)
ax1 = fig.add_subplot(1,2,1)
ax2 = fig.add_subplot(1,2,2)
sns.heatmap(low, annot=True, fmt=' .6g', center=0, ax=ax1)
sns.heatmap(high, annot=True, fmt=' .6g', center=0, ax=ax2)
ax1.set_title('Low mean Cost Per Head', fontdict={'fontsize': 16})
ax2.set_title('High mean Cost Per Head', fontdict={'fontsize': 16})
plt.show()
```

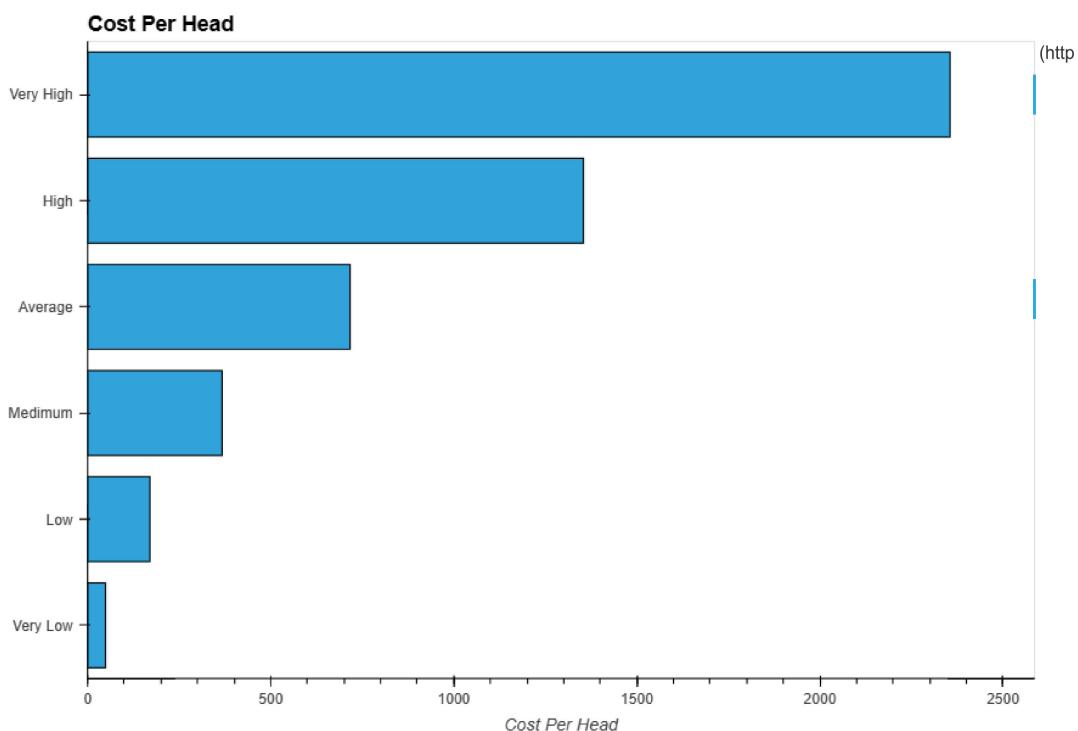


```
In [95]: category = df.groupby('Catagory')[['Cost Per Head', 'Ratings', 'Votes']].agg('mean')
plt.figure(figsize=(20,5), dpi=100)
sns.heatmap(category, annot=True, fmt=".4g", center=0)
plt.title('Total mean Cost Per Head')
plt.show()
```



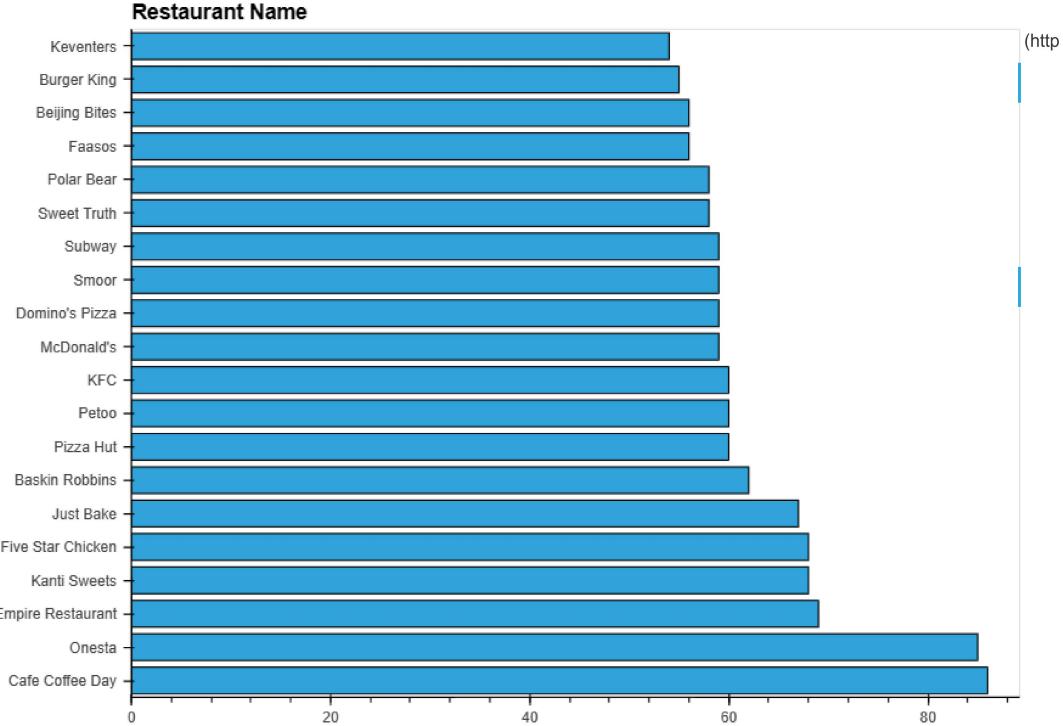
```
In [96]: import hvplot.pandas
df.groupby('Cost_Per_Head_Range')['Cost_Per_Head'].mean().sort_values(ascending=True)[:20].hvplot.barh(height=550, width=800)
```

Out[96]:



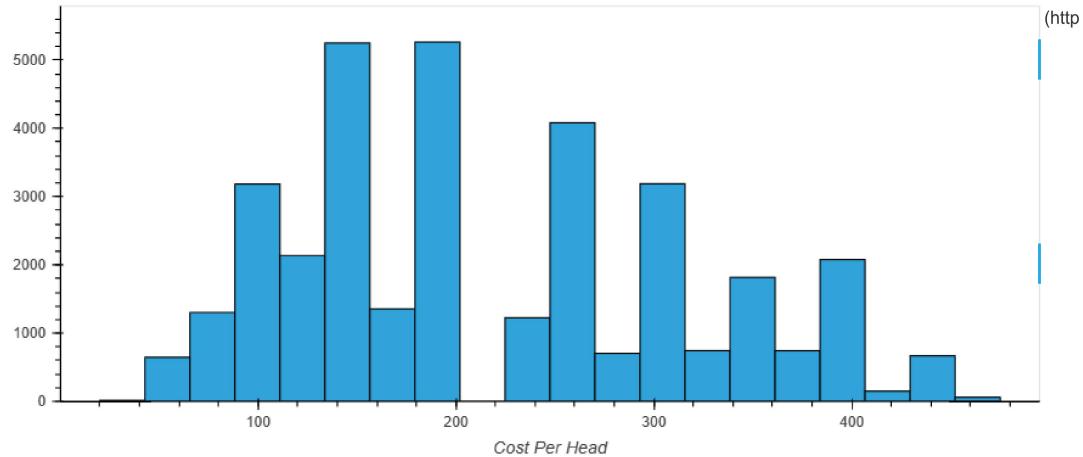
```
In [97]: import hvplot.pandas  
df['Restaurant Name'].value_counts()[:20].hvplot.barr(height=550, width=800)
```

Out[97]:



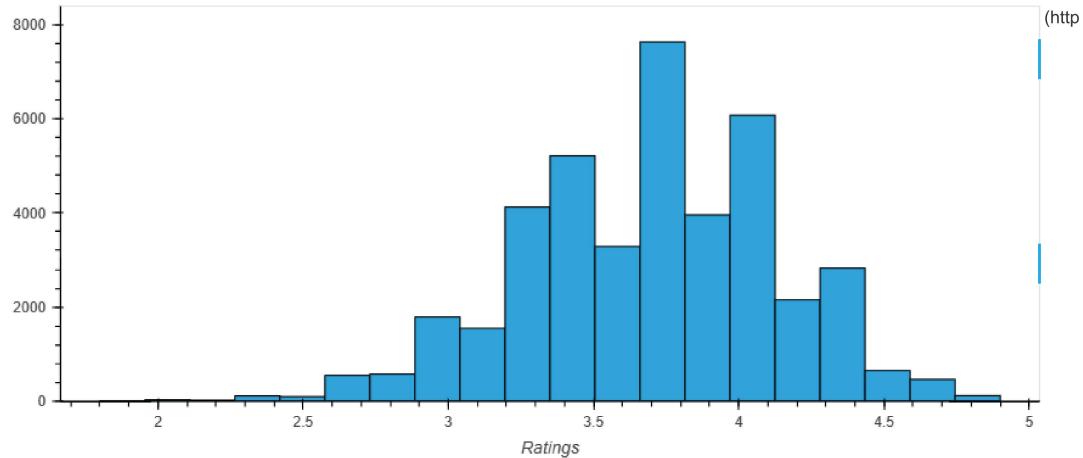
```
In [98]: import hvplot.pandas  
df[df['Cost Per Head'] < 500].hvplot.hist(y='Cost Per Head', bins=20, height=350, width=800)
```

Out[98]:



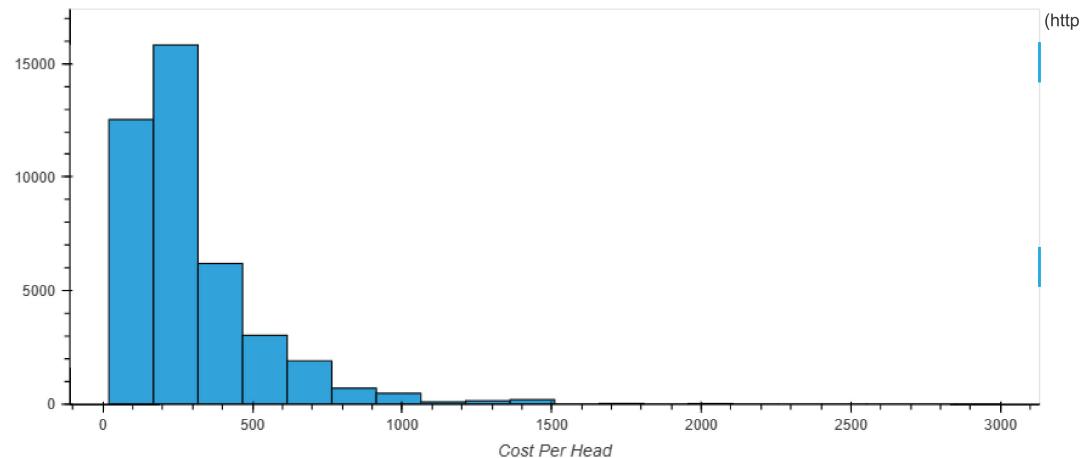
```
In [99]: import hvplot.pandas  
df.hvplot.hist(y='Ratings', height=350, width=800)
```

Out[99]:



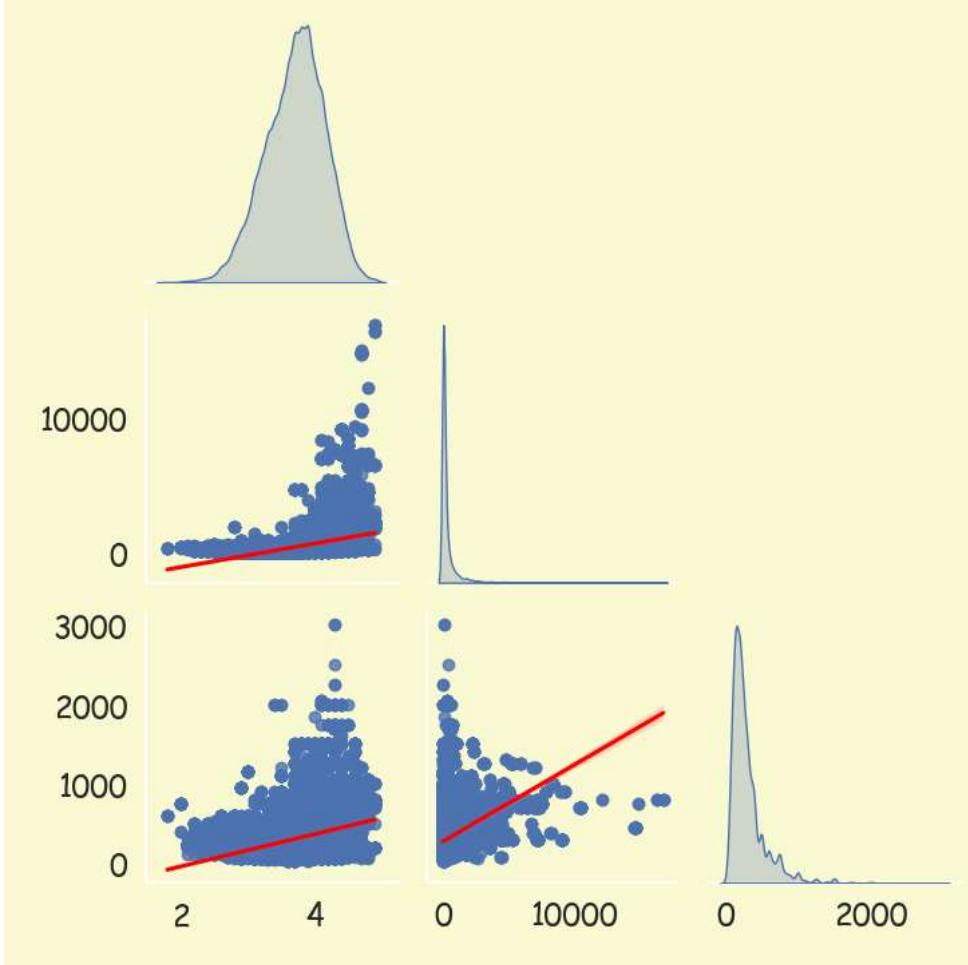
```
In [100]: import hvplot.pandas  
df.hvplot.hist(y='Cost Per Head', height=350, width=800)
```

Out[100]:



{ Pairplot }

```
In [101]: sns.pairplot(df,kind="reg",diag_kind='kde',plot_kws={'line_kws':{'color':'red'}},corner=True)
plt.tight_layout()
plt.show()
```



{ BarPlot of Catagory }

```
In [102]: plt.figure(figsize = (20, 6))
ax = df.Catagory.value_counts().plot(kind = 'bar', color = 'olivedrab')

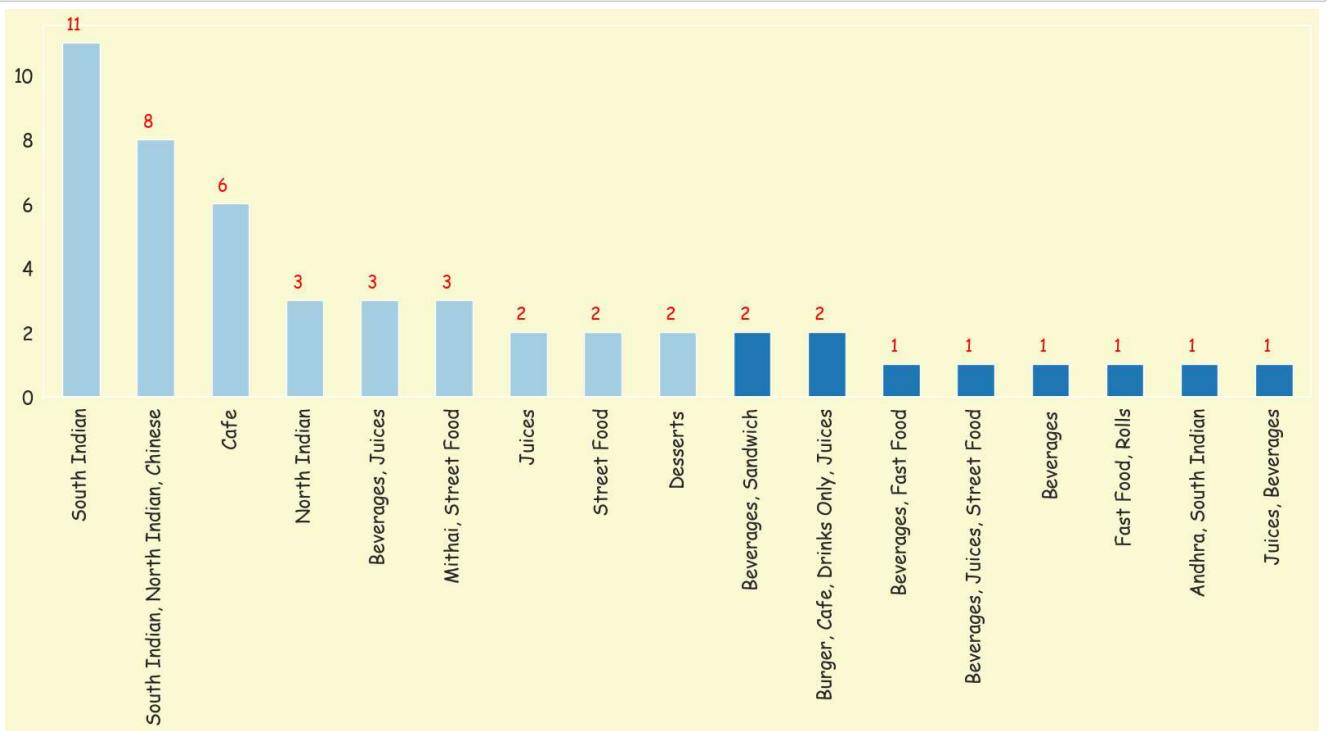
for p in ax.patches:
    ax.annotate(int(p.get_height()), (p.get_x() + 0.25, p.get_height() + 1), ha = 'center', va = 'bottom', color = 'black')
```



{ BarPlot of Cuisines }

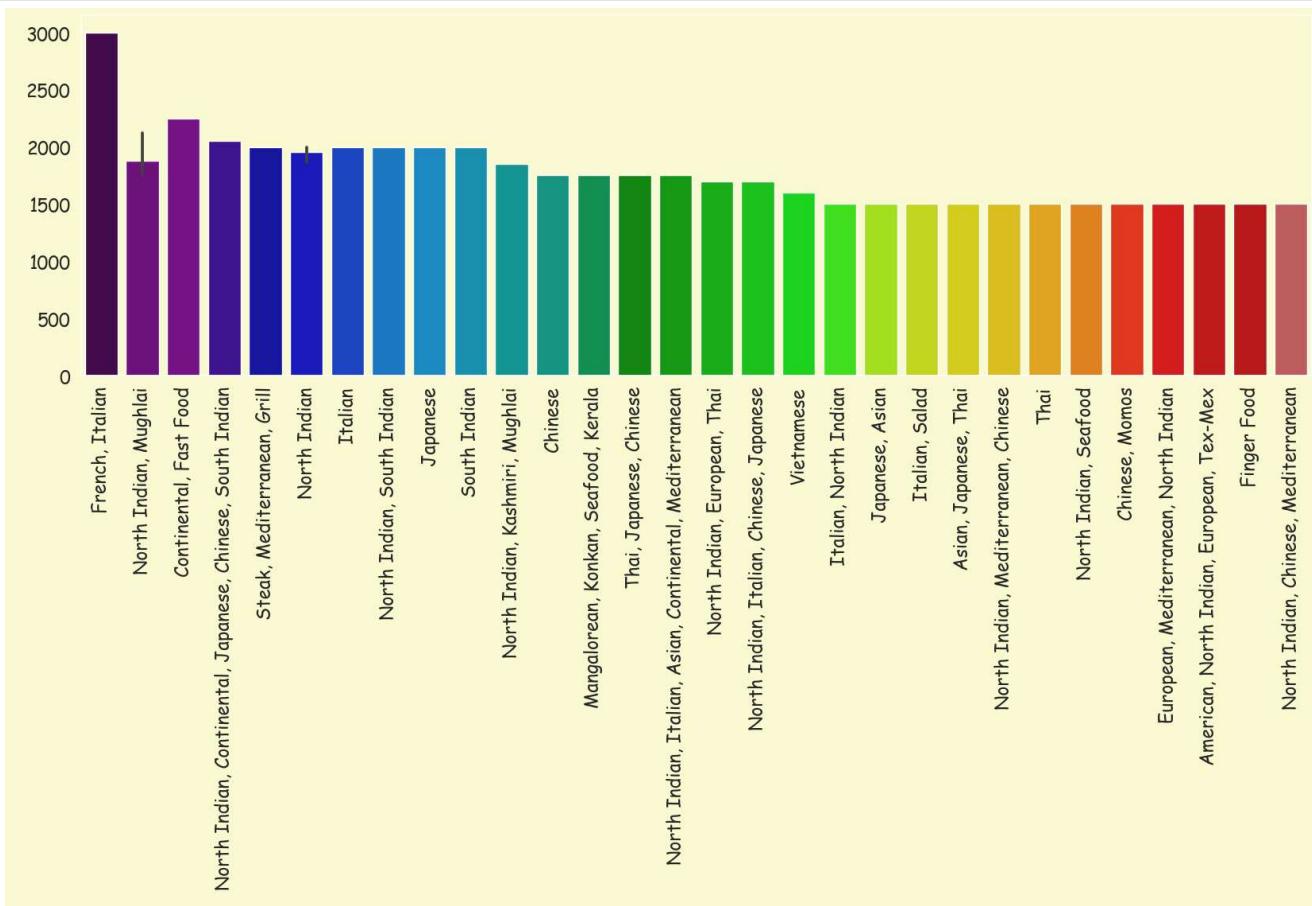
```
In [114]: plt.figure(figsize = (20, 6))
ax = df.Cuisines[0:50].value_counts().plot(kind = 'bar', color = plt.cm.Paired(np.linspace(0, 1, 100)))

for p in ax.patches:
    ax.annotate(int(p.get_height()), (p.get_x() + 0.15, p.get_height() + .25), ha = 'center', va = 'bottom', color = 'Red')
```



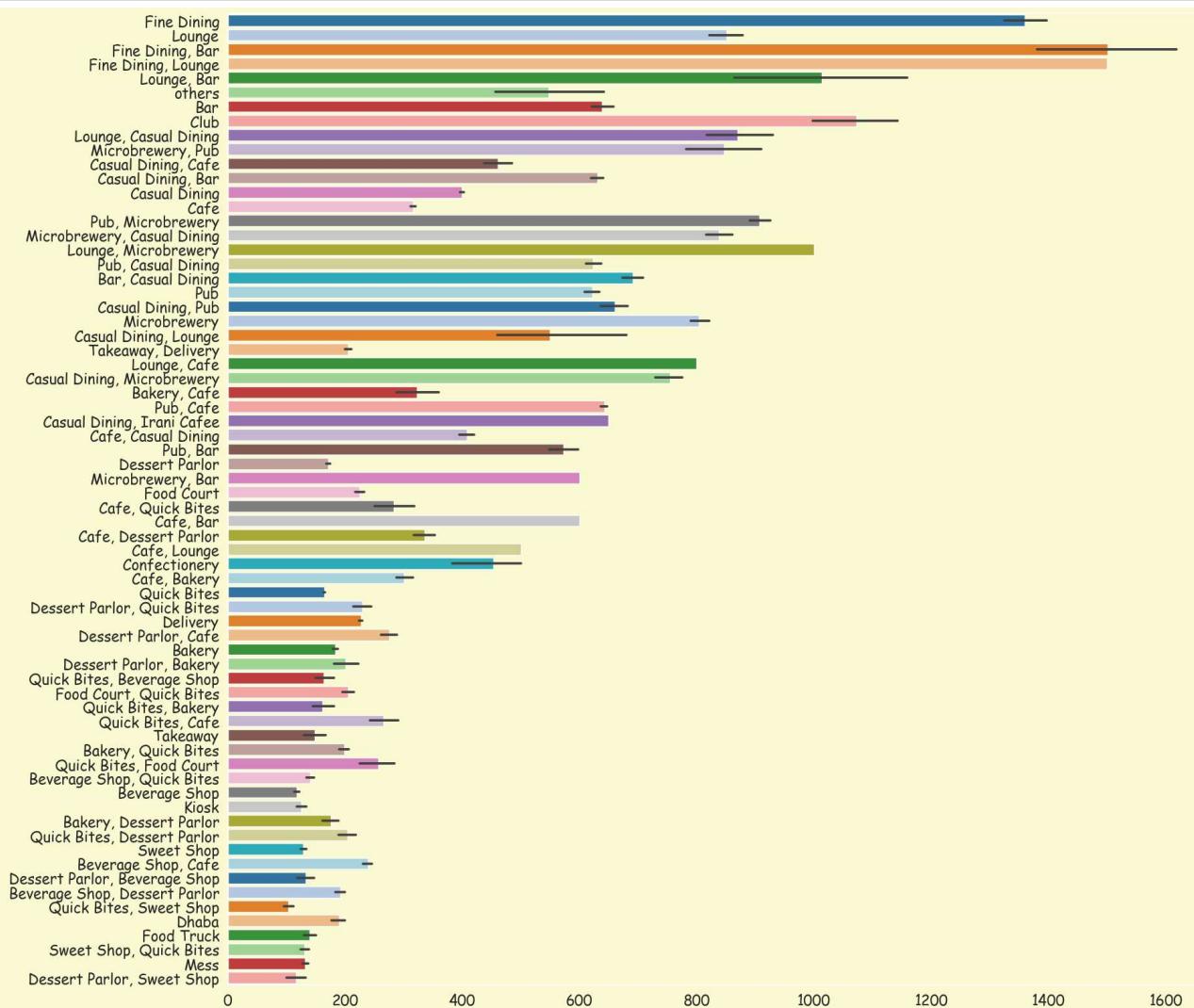
(BarPlot of Cuisines Vs Cost Per Head)

```
In [104]: plt.figure(figsize = (20, 6))
plt.xticks(rotation = 90)
plt.ticklabel_format(style = 'plain')
df.sort_values("Cost Per Head", axis = 0, ascending = False, inplace = True)
sns.barplot(x = df["Cuisines"][1:100],
            y = df["Cost Per Head"][1:100],
            palette = "nipy_spectral");
```



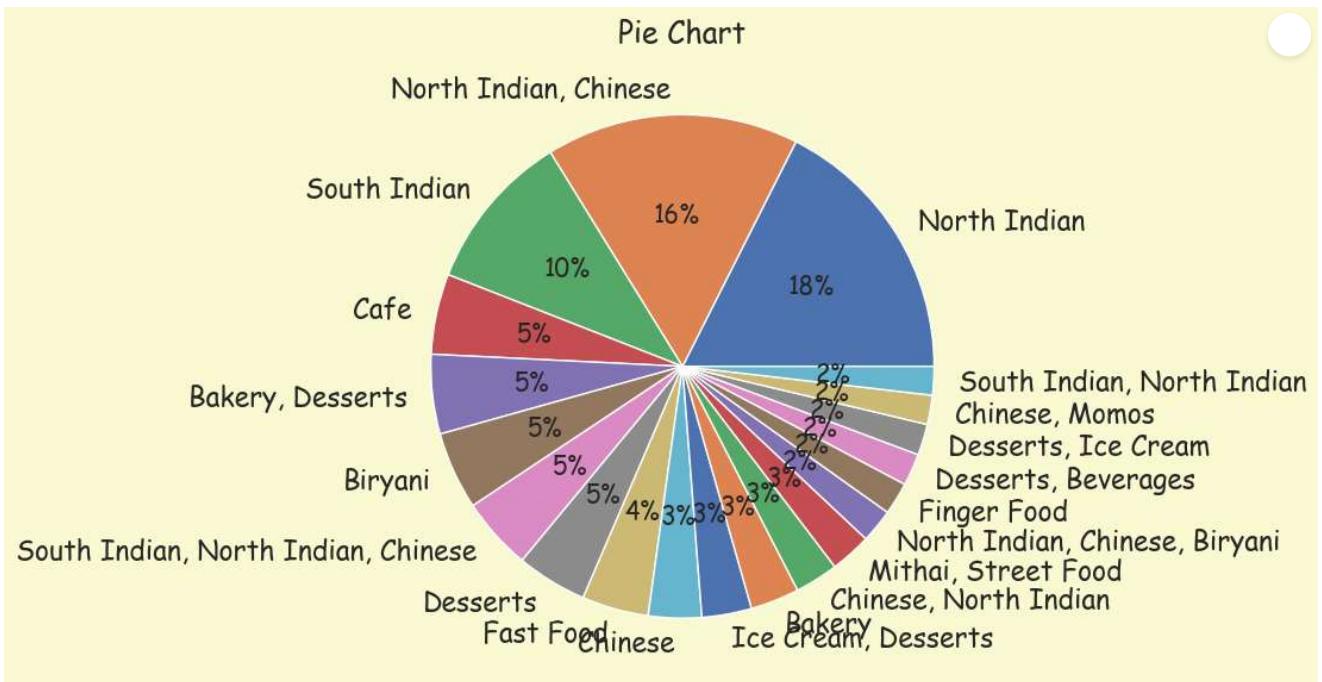
(BarPlot of Cost Per Head Vs Type Of Restaurant)

```
In [105]: plt.figure(figsize = (20, 20))
plt.ticklabel_format(style = 'plain')
sns.barplot(y = df["Type Of Restaurant"], x = df["Cost Per Head"], palette = "tab20");
```



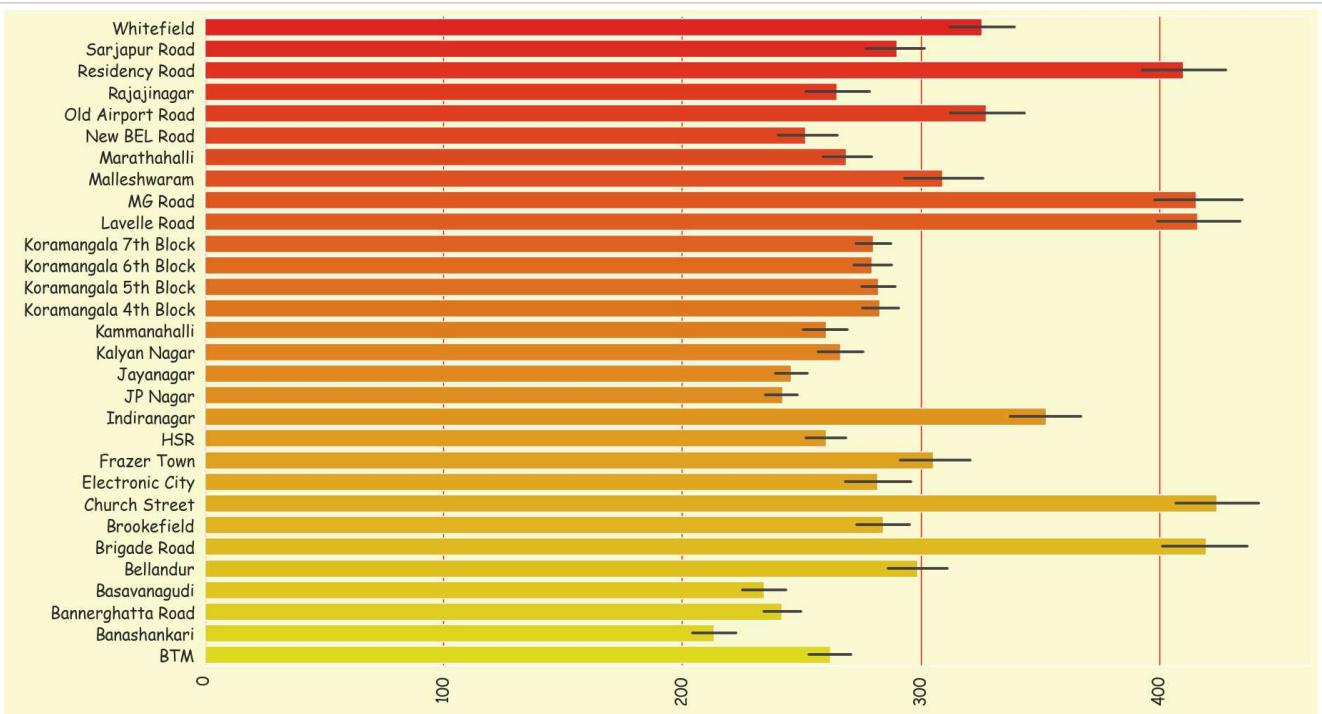
[PieChat of Cuisines Liked in Bangalore]

```
In [106]: df[["Cuisines"]].value_counts()[:20].plot.pie(figsize = (7, 7),
                                                autopct = '%1.0f%%')
plt.title("Pie Chart")
plt.xticks(rotation = 90)
plt.show()
```

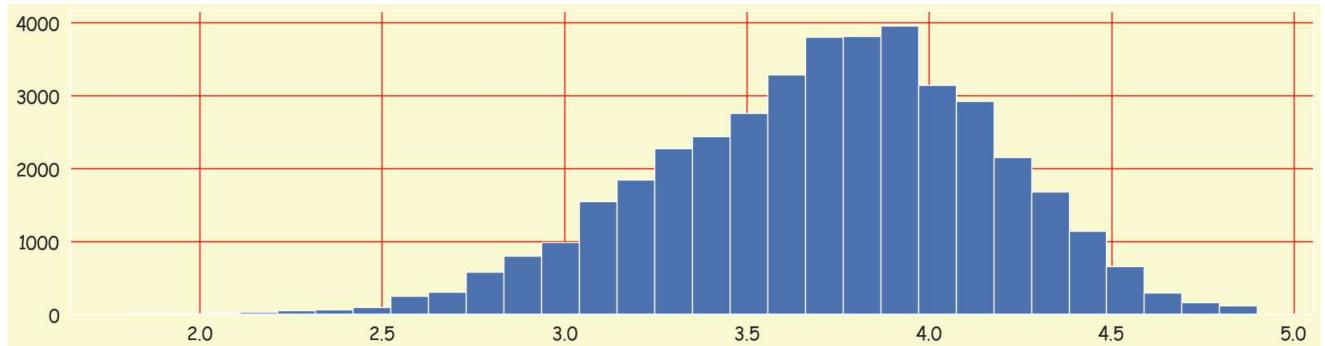


Barplot of Cost Per Head Vs Listed in City

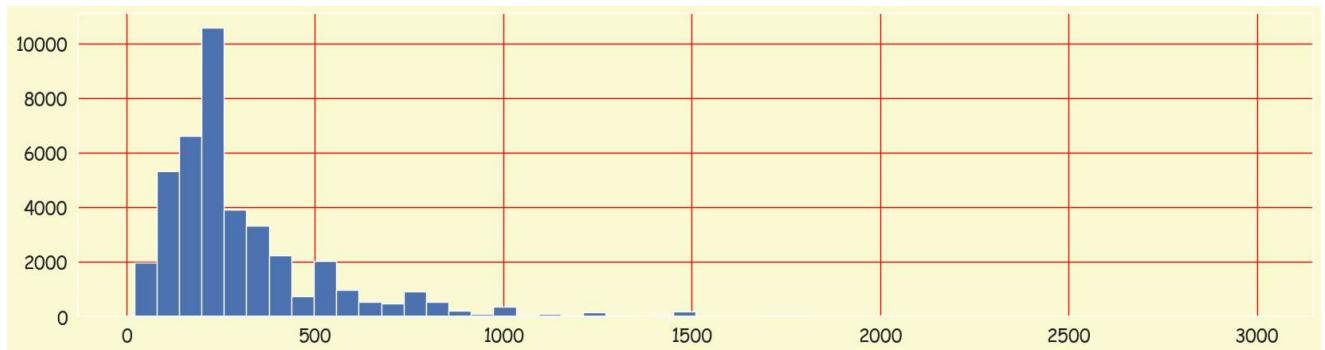
```
In [107]: plt.figure(figsize = (20, 12))
plt.grid()
plt.xticks(rotation = 90)
plt.ticklabel_format(style = 'plain')
df.sort_values("Listed in City", axis = 0,
               ascending = False,
               inplace = True)
sns.barplot(x = df["Cost Per Head"][:40000],
            y = df["Listed in City"],
            palette = "autumn");
```



```
In [108]: n_bins = 30  
plt.figure(figsize = (20, 5))  
df["Ratings"].hist(bins = n_bins);
```



```
In [109]: n_bins = 50  
plt.figure(figsize = (20, 5))  
df["Cost Per Head"].hist(bins = n_bins);
```



```
In [110]: df['Ratings_Grade'].value_counts()
```

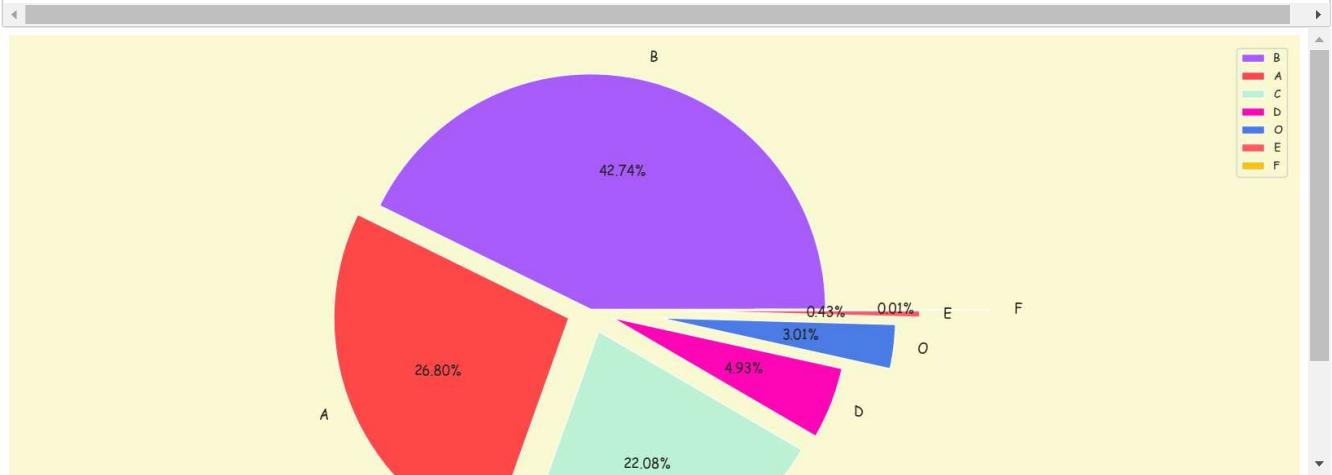
```
Out[110]: B    17636  
A    11057  
C    9110  
D    2033  
O    1244  
E    178  
F      5  
Name: Ratings_Grade, dtype: int64
```

(PieChat of Ratings_Grade)

```
In [111]: plt.figure(figsize = (20, 9))

counts = df["Ratings_Grade"].value_counts()
explode = (0, 0.1, 0.1, 0.1, 0.3, 0.4, 0.7)
colors = ['#A85CF9', '#FF4949', '#BDF2D5', '#FF06B7', '#4B7BE5', '#FF5D5D', '#FAC213', '#37E2D5', '#6D8B74', '#E9D5CA', '#F77E21']

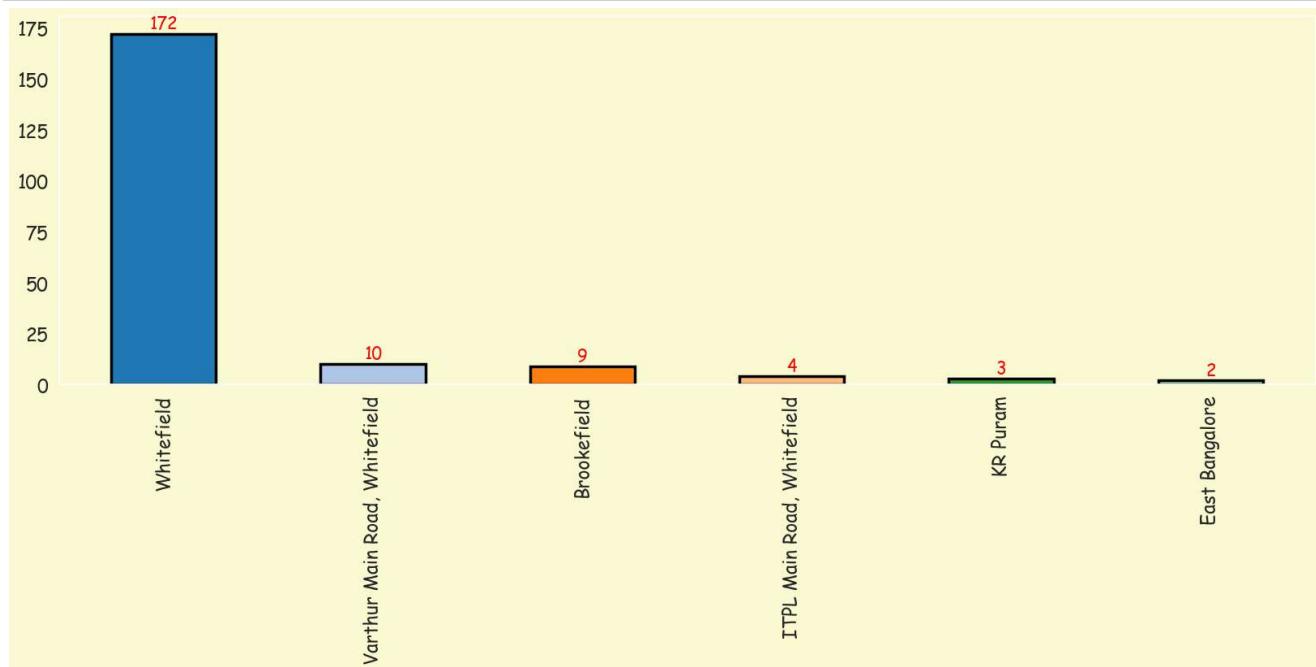
counts.plot(kind = 'pie', fontsize = 12, colors = colors, explode = explode, autopct = '%1.2f%%')
plt.axis('equal')
plt.legend(labels = counts.index, loc = "best")
plt.show()
```



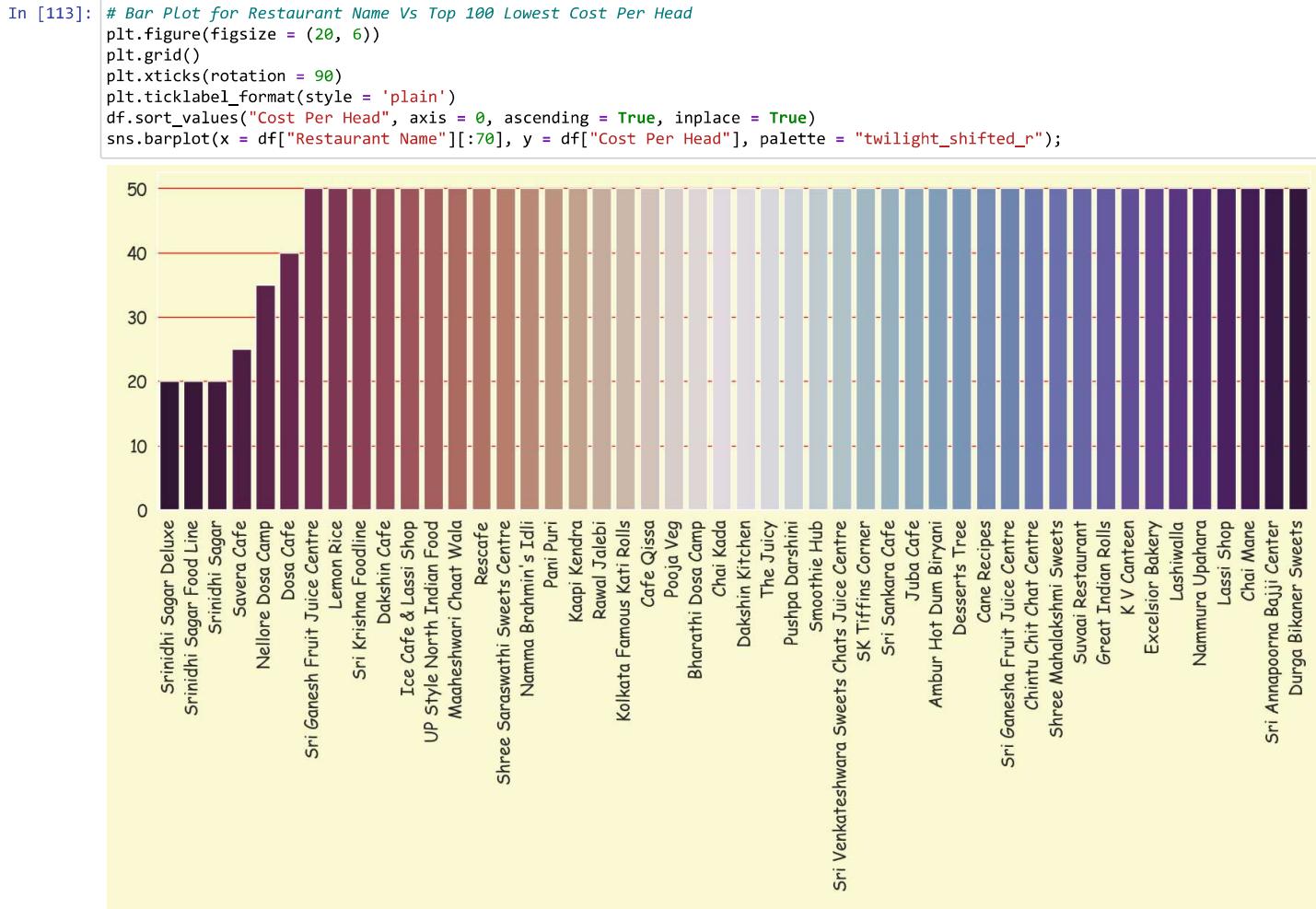
{ Barplot of Location }

```
In [112]: plt.figure(figsize = (20, 6))
col_map = plt.get_cmap('tab20')
ax = df["Location"][:200].value_counts().plot(kind = 'bar',
                                                color = col_map.colors,
                                                rot = 0, edgecolor = 'Black',
                                                linewidth = 2.5)

plt.xticks(rotation=90)
for p in ax.patches:
    ax.annotate(int(p.get_height()), (p.get_x() + 0.25, p.get_height() + .25), ha = 'center', va = 'bottom', color = 'Red')
```



{ Barplot of Restaurant Name Vs Cost Per Head }



Work in Progress

In []: