In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import cufflinks as cf
import plotly.express as px
%matplotlib inline

from plotly.offline import download_plotlyjs,init_notebook_mode,plot, iplot
init_notebook_mode(connected=True)
cf.go_offline()
```

In [3]:
```
df=pd.read_csv(r"C:\Users\HP\Downloads\zomato.csv\zomato.csv")
```

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

Out[4]:

	url	address	name	online_order	book_table	rat
0	https://www.zomato.com/bangalore/jalsa-banasha...	942, 21st Main Road, 2nd Stage, Banashankari, ...	Jalsa	Yes	Yes	4.1
1	https://www.zomato.com/bangalore/spice-elephan...	2nd Floor, 80 Feet Road, Near Big Bazaar, 6th ...	Spice Elephant	Yes	No	4.1
2	https://www.zomato.com/SanchurroBangalore?cont...	1112, Next to KIMS Medical College, 17th Cross...	San Churro Cafe	Yes	No	3.8
3	https://www.zomato.com/bangalore/addhuri-udupi...	1st Floor, Annakuteera, 3rd Stage, Banashankar...	Addhuri Udupi Bhojana	No	No	3.7
4	https://www.zomato.com/bangalore/grand-village...	10, 3rd Floor, Lakshmi Associates, Gandhi Baza...	Grand Village	No	No	3.8

In [8]:
```
print("The number of rows:{}".format(df.shape[0]))
print("The number of columns:{}".format(df.shape[1]))
```

```
The number of rows:51717
The number of columns:17
```

In [6]:
```
df.shape[0]
```

Out[6]:
```
51717
```

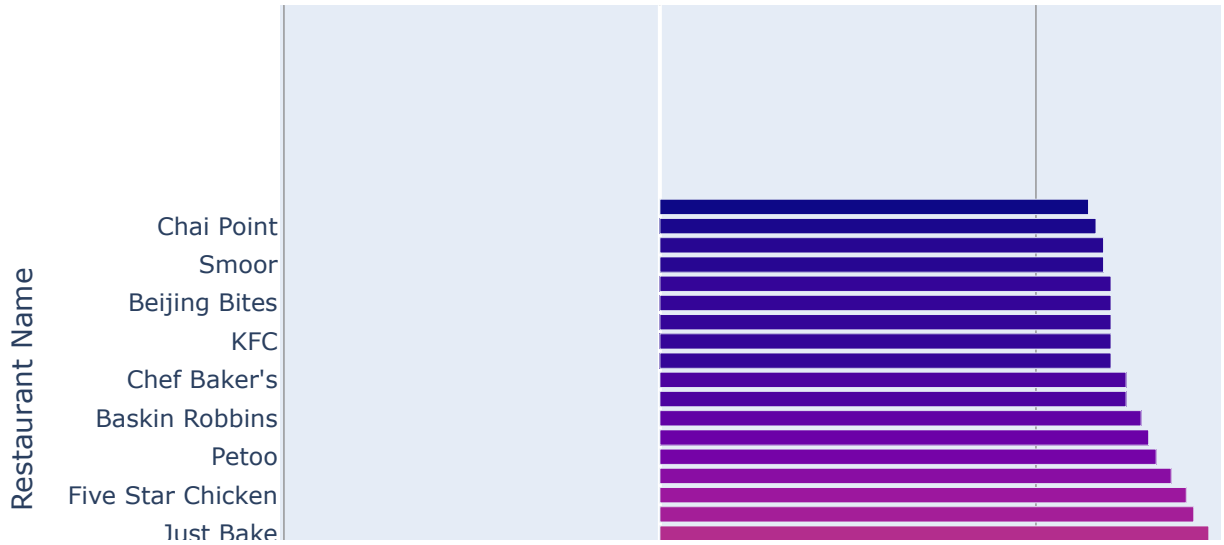
In [10]:
```
df.isnull().sum()
```

Out[10]:
```
url                                0
address                            0
name                               0
online_order                       0
book_table                         0
rate                            7775
votes                              0
phone                           1208
location                          21
rest_type                        227
dish_liked                     28078
cuisines                          45
approx_cost(for two people)      346
reviews_list                       0
menu_item                          0
listed_in(type)                    0
listed_in(city)                    0
dtype: int64
```

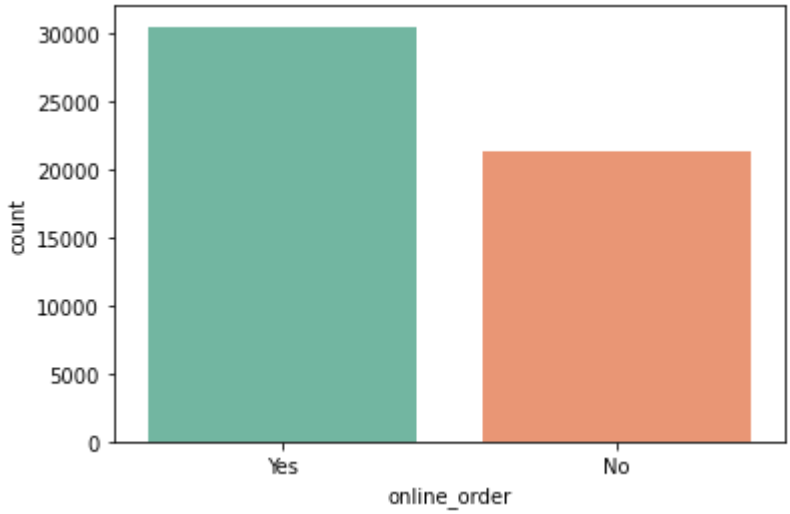
In [11]:
```
df["Rating_from_5"]=df['rate'].dropna().apply(lambda x : float(x.split('/')[0]) i
```

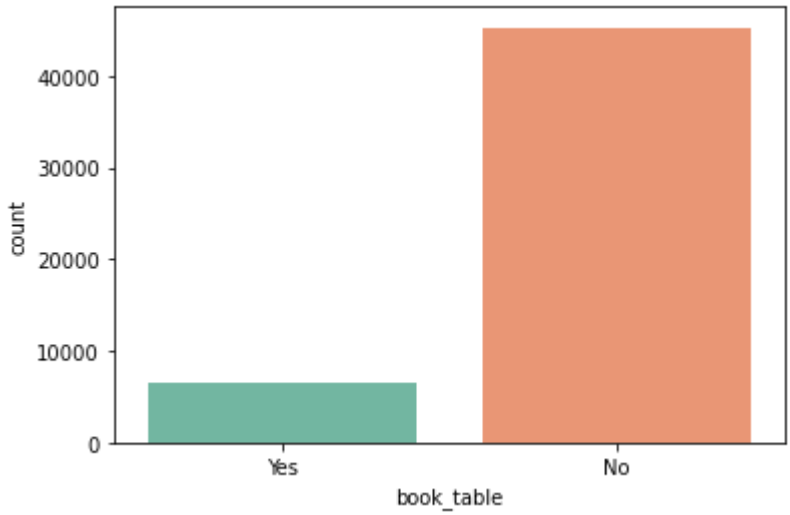
In [13]:
```
import plotly
```

In [15]:
```
top_chains=df['name'].value_counts()[:20]
fig=px.bar(top_chains, x='name',y=top_chains.index,labels={'index':"Restaurant Na
fig.update(layout_coloraxis_showscale=False)
```



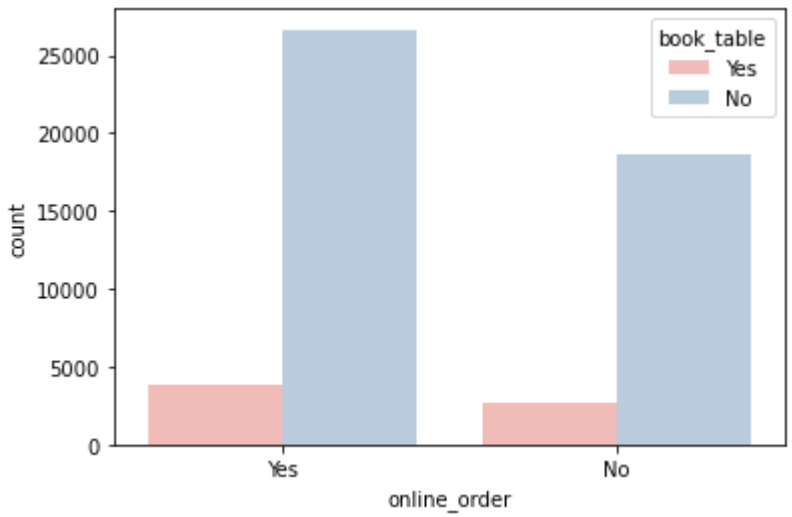
In [20]:
```
flag=["online_order", "book_table"]
for val in flag:
    sns.countplot(x=val, data=df,palette ="Set2")
    plt.show()
```



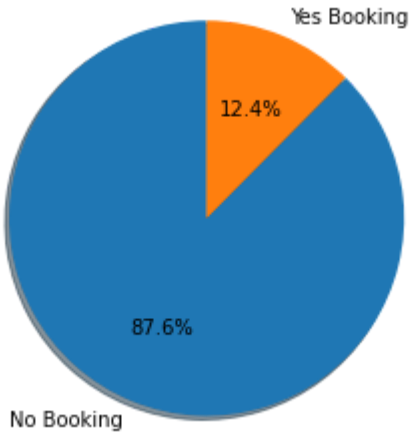


In [21]:
```
sns.countplot(x='online_order', data=df, hue='book_table', palette = "Pastel1")
```

Out[21]:
```
<AxesSubplot:xlabel='online_order', ylabel='count'>
```



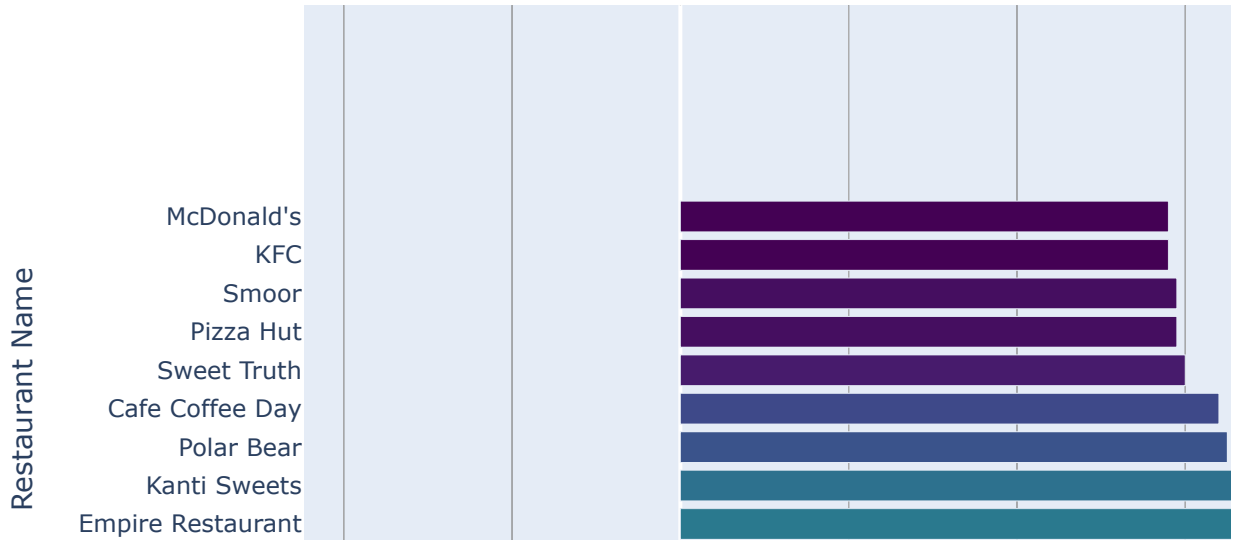
In [25]:
```
# Percentage division of offline restaurants which accept pre booking
b=df.groupby(['online_order','book_table']).book_table.value_counts().sort_values
sizes=[]
for i in range (2):
    sizes.append(b.loc['No'][i])
labels=['No Booking', 'Yes Booking']
fig1, ax1 = plt.subplots()
ax1.pie(sizes, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=90)
ax1.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

plt.show()
```



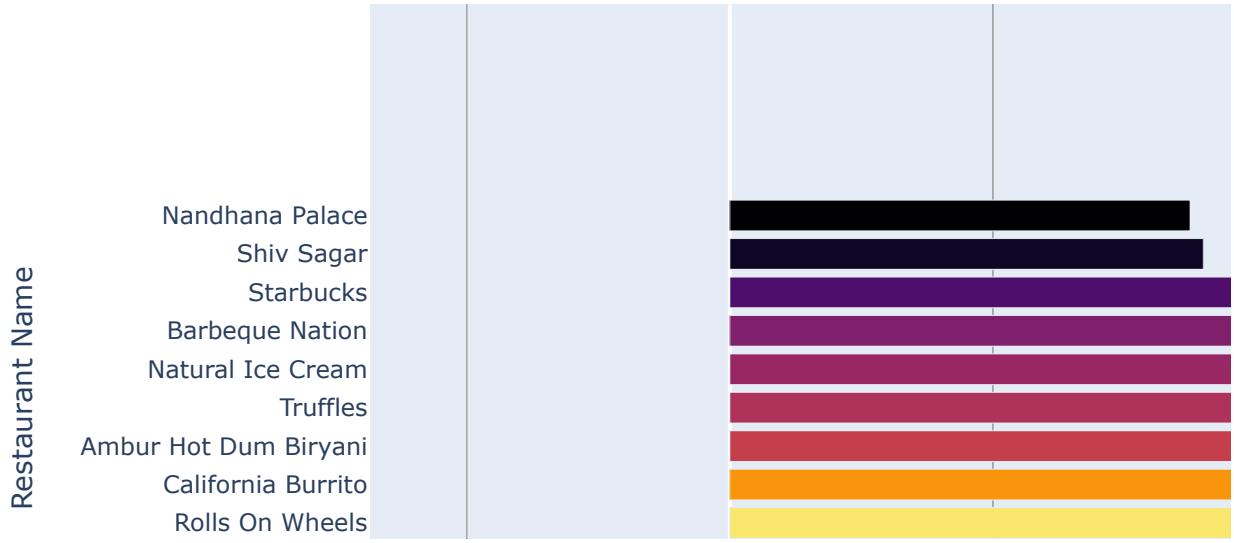
Top 10 online and 10 offline restaurants

In [36]:
```
#Top 10 online restaurants
a=df.groupby(['online_order','name']).name.count().sort_values(ascending=False)
top_online=a.loc['Yes'][:10]
top_offline=a.loc['No'][:10]
```

In [42]:
```
fig=px.bar(top_online, x='name',y=top_online.index,labels={'index':"Restaurant Na
fig.update(layout_coloraxis_showscale=False)
```



In [44]:
```
# Top 10 offline restaurants
fig=px.bar(top_offline, x='name',y=top_offline.index,labels={'index':"Restaurant
fig.update(layout_coloraxis_showscale=False)
```

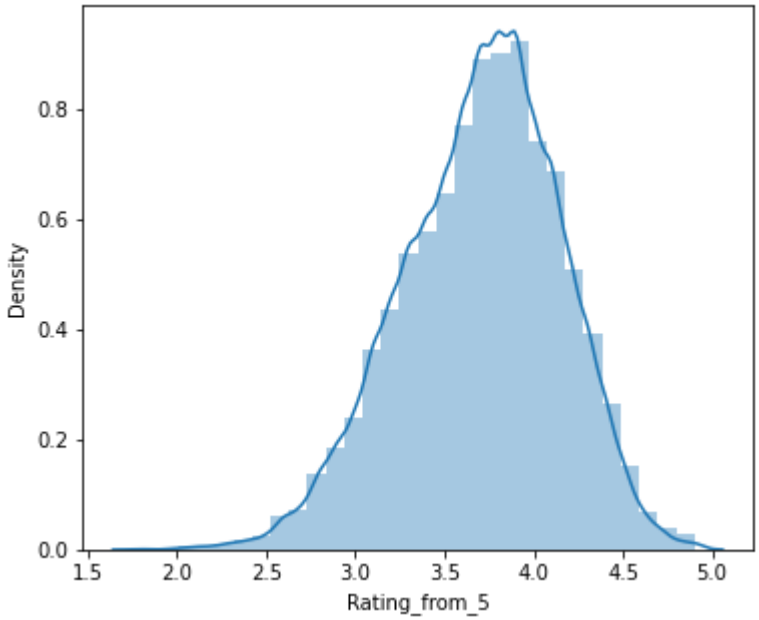


In [5]:
```
#No. of votes for online and offline orders
px.violin(df,y="votes", x="online_order", color="online_order", box=True,points='
```

In [55]:
```
# Rating distribution out of 5
plt.figure(figsize=(6,5))
sns.distplot(df['Rating_from_5'],bins=30)
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWa
rning:

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

Out[55]:
```
<AxesSubplot:xlabel='Rating_from_5', ylabel='Density'>
```



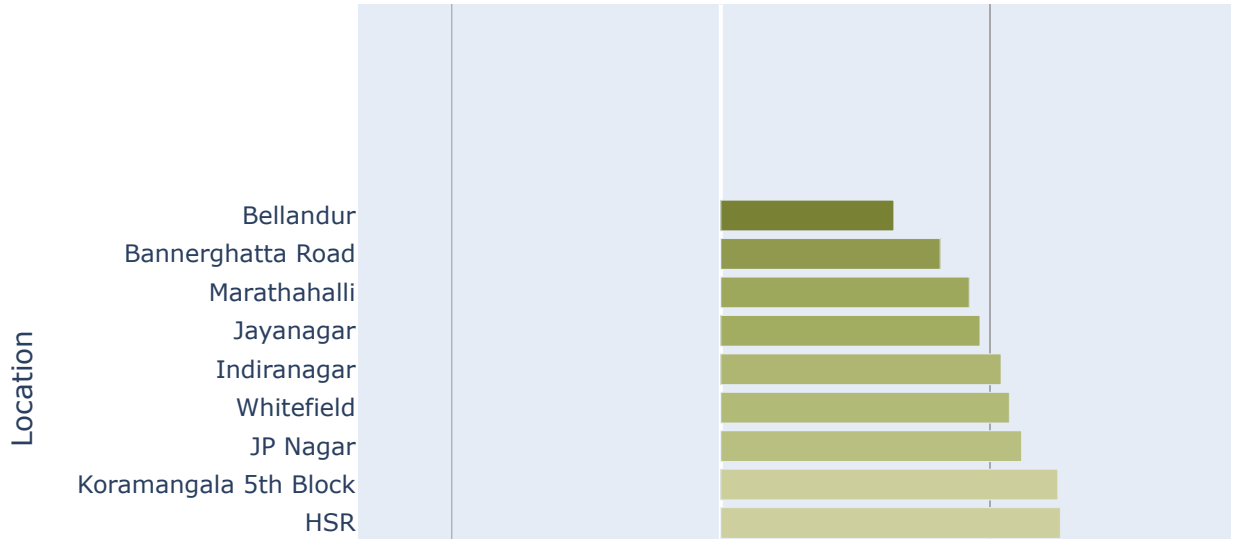
In [ ]:
```
#maximum restaurants have  arating between 3.5 to 4
```

In [56]:
```
c=b=df.groupby(['online_order'])["Rating_from_5"].mean()
```

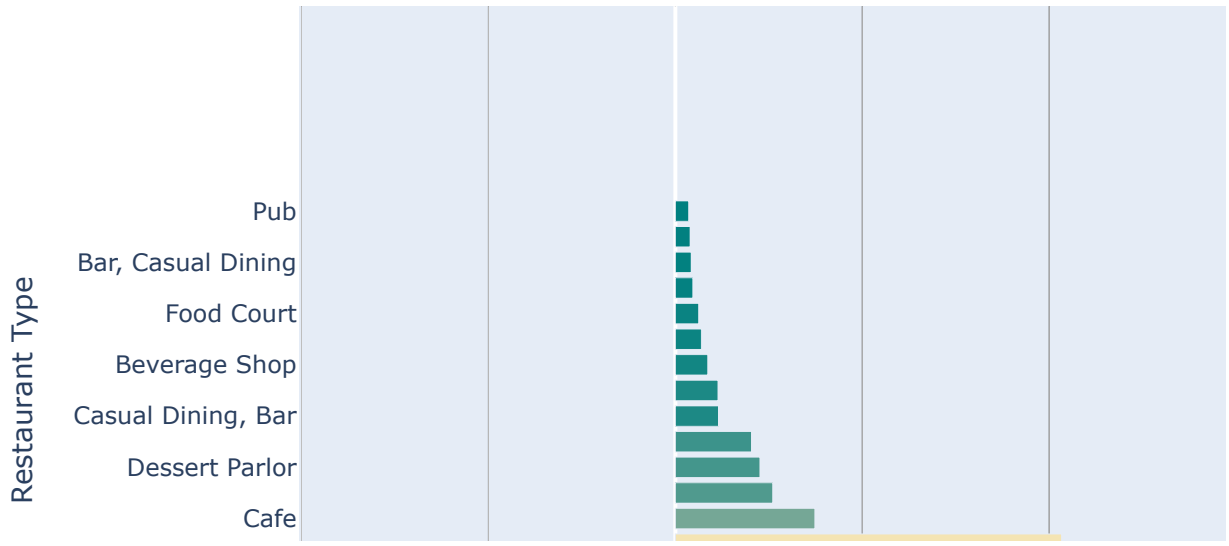
In [62]:
```
print("The average rating of restaurants delivering online order is {:.2f} and th
```

```
The average rating of restaurants delivering online order is 3.72 and the avera
ge rating of restaurants delivering offline order is 3.66
```

In [184]:
```
#Top 10 locations in the city
top_locations=df['location'].value_counts()[:10]
fig=px.bar(top_locations, x='location',y=top_locations.index,labels={'index':"Loc
fig.update(layout_coloraxis_showscale=False)
```



In [186]:
```
#Top 15 restaurant types
top_rest_type=df['rest_type'].value_counts()[:15]
fig=px.bar(top_rest_type, x='rest_type',y=top_rest_type.index,labels={'index':"Re
fig.update(layout_coloraxis_showscale=False)
```



In [138]:
```
df1=df.groupby(['location','cuisines']).agg({'name':'count'})
df1.reset_index(inplace=True)
df1 = df1.drop_duplicates(subset = ["name"])
df1=df1.sort_values(by="name",ascending = False)
df1.head(10)
```

Out[138]:

	location	cuisines	name
240	BTM	North Indian, Chinese	379
225	BTM	North Indian	340
7554	Whitefield	North Indian	188
2672	HSR	North Indian	173
1187	Bellandur	North Indian	161
3466	JP Nagar	North Indian	157
5595	Marathahalli	North Indian	155
4331	Koramangala 1st Block	North Indian	150
5609	Marathahalli	North Indian, Chinese	145
797	Bannerghatta Road	North Indian, Chinese	130

In [163]:
```
df2=df.groupby(['rest_type','cuisines']).agg({'name':'count'})
df3=pd.DataFrame(df2)
df3.reset_index(inplace=True)
df3 = df3.drop_duplicates(subset = ["name"])
df3=df3.sort_values(by="name",ascending = False)
df3.head(10)
```

Out[163]:

	rest_type	cuisines	name
3360	Quick Bites	South Indian	1532
3076	Quick Bites	North Indian	1480
3109	Quick Bites	North Indian, Chinese	1259
282	Cafe	Cafe	737
1097	Casual Dining	North Indian, Chinese	712
1057	Casual Dining	North Indian	695
2816	Quick Bites	Fast Food	681
2583	Quick Bites	Biryani	678
3	Bakery	Bakery, Desserts	561
3431	Quick Bites	South Indian, North Indian, Chinese	504

In [165]:
```
df["listed_in(type)"].unique
```

Out[165]:
```
<bound method Series.unique of 0              Buffet
1              Buffet
2              Buffet
3              Buffet
4              Buffet
              ...
51712    Pubs and bars
51713    Pubs and bars
51714    Pubs and bars
51715    Pubs and bars
51716    Pubs and bars
Name: listed_in(type), Length: 51717, dtype: object>
```

In [167]:
```
df4=df.groupby(['listed_in(type)']).agg({'Rating_from_5':'mean'})
df4
```

Out[167]:

listed_in(type)	Rating_from_5
Buffet	3.982105
Cafes	3.872477
Delivery	3.653257
Desserts	3.777013
Dine-out	3.680826
Drinks & nightlife	4.017062
Pubs and bars	4.022933

In [171]:
```
df5=df.groupby(['rest_type']).agg({'Rating_from_5':'mean'})
df5.dropna()
```

Out[171]:

rest_type	Rating_from_5
Bakery	3.607955
Bakery, Beverage Shop	3.200000
Bakery, Cafe	4.009722
Bakery, Dessert Parlor	3.687402
Bakery, Food Court	3.100000
...	...
Quick Bites, Sweet Shop	3.570667
Sweet Shop	3.626012
Sweet Shop, Quick Bites	3.610526
Takeaway	3.407229
Takeaway, Delivery	3.513622

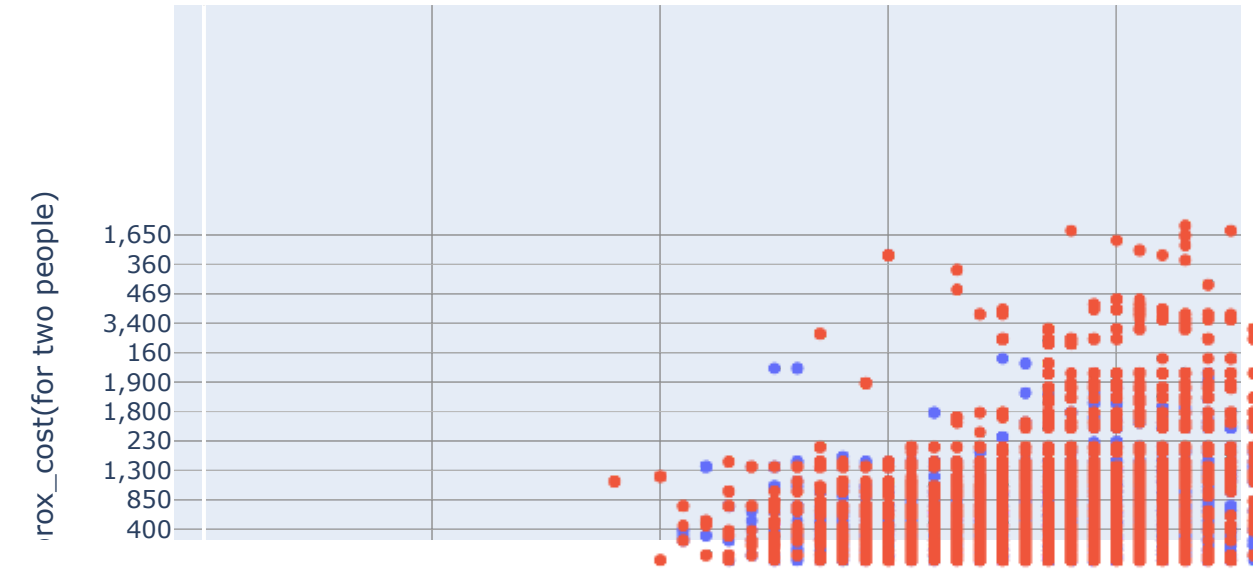
87 rows × 1 columns

In [173]:
```
df.dtypes
```

Out[173]:
```
url                            object
address                        object
name                           object
online_order                   object
book_table                     object
rate                           object
votes                           int64
phone                          object
location                       object
rest_type                      object
dish_liked                     object
cuisines                       object
approx_cost(for two people)    object
reviews_list                   object
menu_item                      object
listed_in(type)                object
listed_in(city)                object
Rating_from_5                 float64
dtype: object
```

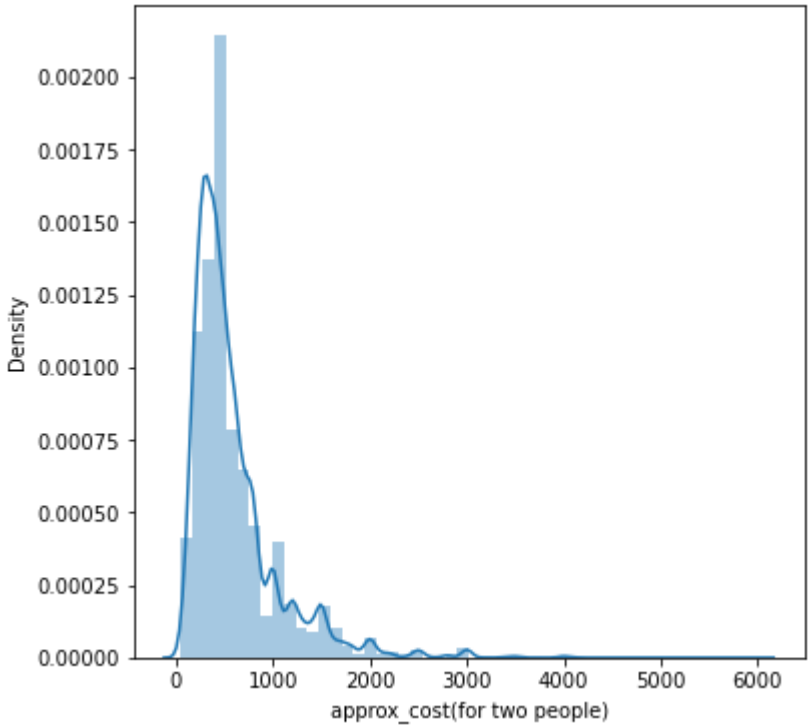
In [178]:
```
cost_dist=df[['rate','approx_cost(for two people)','online_order']].dropna()
cost_dist['rate']=cost_dist['rate'].apply(lambda x: float(x.split('/')[0]) if len
cost_dist['approx_cost(for two people)']=cost_dist['approx_cost(for two people)']
```

In [182]:
```
#Scatter plot for Rating and cost for two people(in Rupees)
px.scatter(df, x="Rating_from_5", y="approx_cost(for two people)", color="online_
```



In [180]:
```
#Distribution of cost for two people in Rupees
plt.figure(figsize=(6,6))
sns.distplot(cost_dist['approx_cost(for two people)'])
plt.show()
```

```
C:\Users\HP\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWa
rning:

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



In [ ]:
```
#An average good meal for two people costs less than $12
```

In [137]:
```
df1.head(10)
```

Out[137]:

	location	cuisines	name
240	BTM	North Indian, Chinese	379
225	BTM	North Indian	340
7554	Whitefield	North Indian	188
2672	HSR	North Indian	173
1187	Bellandur	North Indian	161
3466	JP Nagar	North Indian	157
5595	Marathahalli	North Indian	155
4331	Koramangala 1st Block	North Indian	150
5609	Marathahalli	North Indian, Chinese	145
797	Bannerghatta Road	North Indian, Chinese	130

dfvd

In [ ]: