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

In [ ]:
```
#Download data set of bangalore resturants
bangalore=pd.read_csv('zomato.csv')
```

In [3]:
```
#View of the data set
bangalore.head(10)
```

Out[3]:

	url	address	name	online_order	book_tabl
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

	url	address	name	online_order	book_table
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
5	https://www.zomato.com/bangalore/timepass-dinn...	37, 5-1, 4th Floor, Bosco Court, Gandhi Bazaar...	Timepass Dinner	Yes	No
6	https://www.zomato.com/bangalore/rosewood-inte...	19/1, New Timberyard Layout, Beside Satellite ...	Rosewood International Hotel - Bar & Restaurant	No	No
7	https://www.zomato.com/bangalore/onesta-banash...	2469, 3rd Floor, 24th Cross, Opposite BDA Comp...	Onesta	Yes	Yes
8	https://www.zomato.com/bangalore/penthouse-caf...	1, 30th Main Road, 3rd Stage, Banashankari, Ba...	Penthouse Cafe	Yes	No
9	https://www.zomato.com/bangalore/smacznego-ban...	2470, 21 Main Road, 25th Cross, Banashankari, ...	Smacznego	Yes	No

In [4]:
```
#dropping columns which are not required
bangalore.drop(['url','address','phone','menu_item','listed_in(city)'],
axis=1,inplace =True)
```

In [5]:
```
bangalore.head()
```

Out[5]:

	name	online_order	book_table	rate	votes	location	rest_type	dish_liked	cuisines
0	Jalsa	Yes	Yes	4.1/5	775	Banashankari	Casual Dining	Pasta, Lunch Buffet, Masala Papad, Paneer Laja...	North Indian Mughlai Chines
1	Spice Elephant	Yes	No	4.1/5	787	Banashankari	Casual Dining	Momos, Lunch Buffet, Chocolate Nirvana, Thai G...	Chinese North Indian Tha
2	San Churro Cafe	Yes	No	3.8/5	918	Banashankari	Cafe, Casual Dining	Churros, Cannelloni, Minestrone Soup, Hot Choc...	Cafe Mexican Italia
3	Addhuri Udupi Bhojana	No	No	3.7/5	88	Banashankari	Quick Bites	Masala Dosa	South Indian North India
4	Grand Village	No	No	3.8/5	166	Basavanagudi	Casual Dining	Panipuri, Gol Gappe	North Indian Rajasthan

In [6]:
```
#informtion of the dataset
```

```
bangalore.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51717 entries, 0 to 51716
Data columns (total 12 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   dish_liked                   23639 non-null  object
 8   cuisines                     51672 non-null  object
 9   approx_cost(for two people)  51371 non-null  object
 10  reviews_list                 51717 non-null  object
 11  listed_in(type)              51717 non-null  object
dtypes: int64(1), object(11)
memory usage: 4.7+ MB
```

In [7]:
```
bangalore.describe()
```

Out[7]:

	votes
count	51717.000000
mean	283.697527
std	803.838853
min	0.000000
25%	7.000000
50%	41.000000
75%	198.000000
max	16832.000000

In [8]:
```
#calculating missing values in the data set
print(bangalore.isna().sum().sort_values(ascending= False))
```

```
dish_liked                     28078
rate                            7775
approx_cost(for two people)      346
rest_type                        227
cuisines                          45
location                          21
listed_in(type)                    0
reviews_list                       0
votes                              0
book_table                         0
online_order                       0
name                               0
dtype: int64
```

In [9]:
```
bangalore['approx_cost(for two people)']=bangalore['approx_cost(for two people)'].astype(str)
bangalore['approx_cost(for two people)']=bangalore['approx_cost(for two people)'].apply(lambda x:x.replace(',',''))
bangalore['approx_cost(for two people)']=bangalore['approx_cost(for two people)'].astype(float)
bangalore.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51717 entries, 0 to 51716
Data columns (total 12 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   dish_liked                   23639 non-null  object
 8   cuisines                     51672 non-null  object
```

```
 9   approx_cost(for two people)  51371 non-null  float64
 10  reviews_list                 51717 non-null  object
 11  listed_in(type)              51717 non-null  object
dtypes: float64(1), int64(1), object(10)
memory usage: 4.7+ MB
```

In [10]:
```
bangalore.rate = bangalore.rate.replace("NEW", np.nan)
bangalore.dropna(how ='any', inplace = True)
```

In [11]:
```
bangalore['rate']=bangalore['rate'].astype(str)
bangalore['rate']=bangalore['rate'].apply(lambda x:x.replace('/5',''))
bangalore['rate']=bangalore['rate'].astype(float)
bangalore.head()
```

Out[11]:

	name	online_order	book_table	rate	votes	location	rest_type	dish_liked	cuisines
0	Jalsa	Yes	Yes	4.1	775	Banashankari	Casual Dining	Pasta, Lunch Buffet, Masala Papad, Paneer Laja...	North Indian, Mughlai, Chinese
1	Spice Elephant	Yes	No	4.1	787	Banashankari	Casual Dining	Momos, Lunch Buffet, Chocolate Nirvana, Thai G...	Chinese, North Indian, Thai
2	San Churro Cafe	Yes	No	3.8	918	Banashankari	Cafe, Casual Dining	Churros, Cannelloni, Minestrone Soup, Hot Choc...	Cafe, Mexican, Italian
3	Addhuri Udupi Bhojana	No	No	3.7	88	Banashankari	Quick Bites	Masala Dosa	South Indian, North Indian

	name	online_order	book_table	rate	votes	location	rest_type	dish_liked	cuisines
4	Grand Village	No	No	3.8	166	Basavanagudi	Casual Dining	Panipuri, Gol Gappe	North Indian, Rajasthani

In [12]:
```
new_bangalore_rate=bangalore[['name','rate']].groupby(['rate'])
new_bangalore_rate=new_bangalore_rate.filter(lambda x:x.mean()>=4.5)
new_bangalore_rate=new_bangalore_rate.sort_values(by=['rate'])
new_bangalore_rate
```

Out[12]:

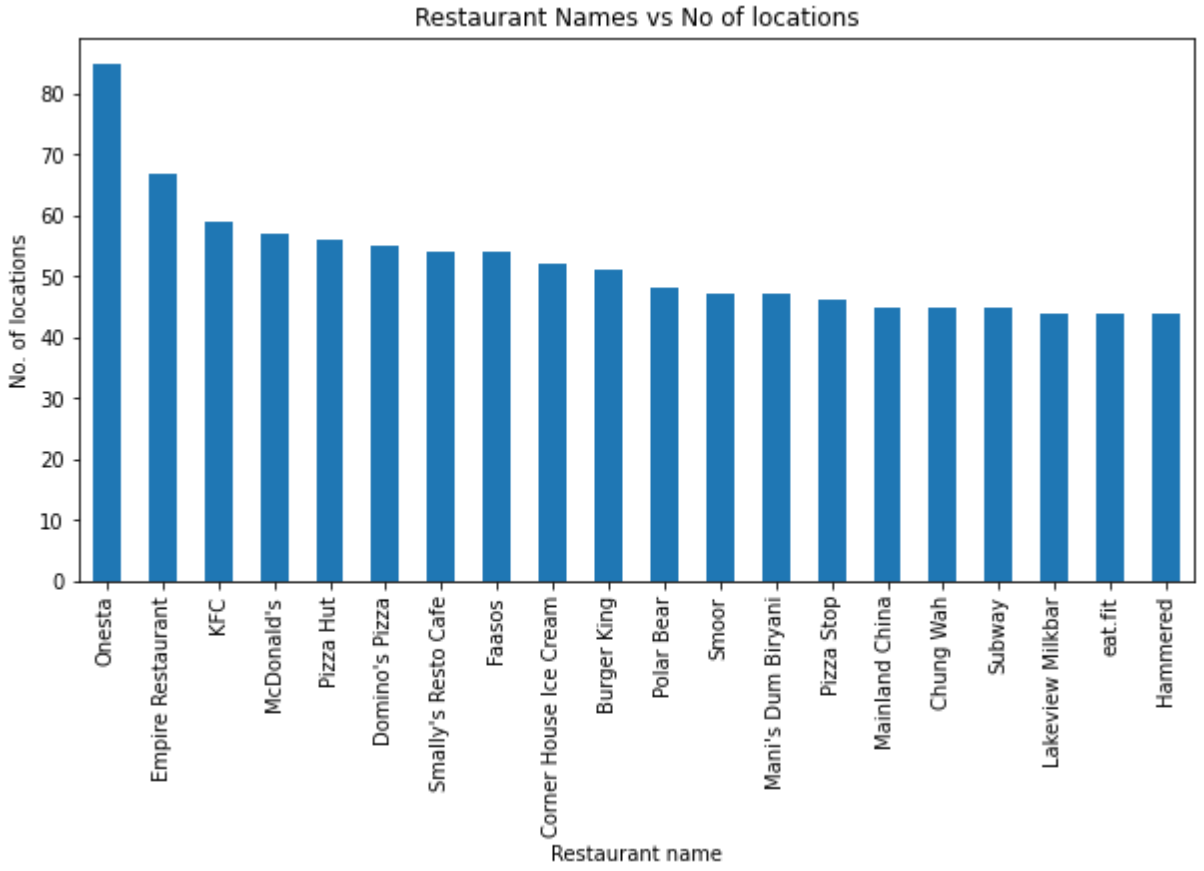
	name	rate
40498	The London Curry House - The Royale Senate Hotel	4.5
38456	Fabelle Chocolate Boutique - ITC Gardenia	4.5
26841	99 Pancakes	4.5
26860	Pin Me Down	4.5
26872	Stoner	4.5
...	...	...
4801	Byg Brewski Brewing Company	4.9
3921	Byg Brewski Brewing Company	4.9
10284	Belgian Waffle Factory	4.9
21770	AB's - Absolute Barbecues	4.9
49627	Byg Brewski Brewing Company	4.9

1236 rows × 2 columns

no of restaurant outlets in the city

In [13]:
```
# no of restaurant outlets in the city
plt.figure(figsize=(10,5))
ax=bangalore['name'].value_counts()[:20].plot(kind='bar')
plt.xlabel('Restaurant name')
plt.ylabel('No. of locations')
plt.title('Restaurant Names vs No of locations')
```

Out[13]:
```
Text(0.5, 1.0, 'Restaurant Names vs No of locations')
```



In [14]:
```
bangalore['online_order'].value_counts()
```

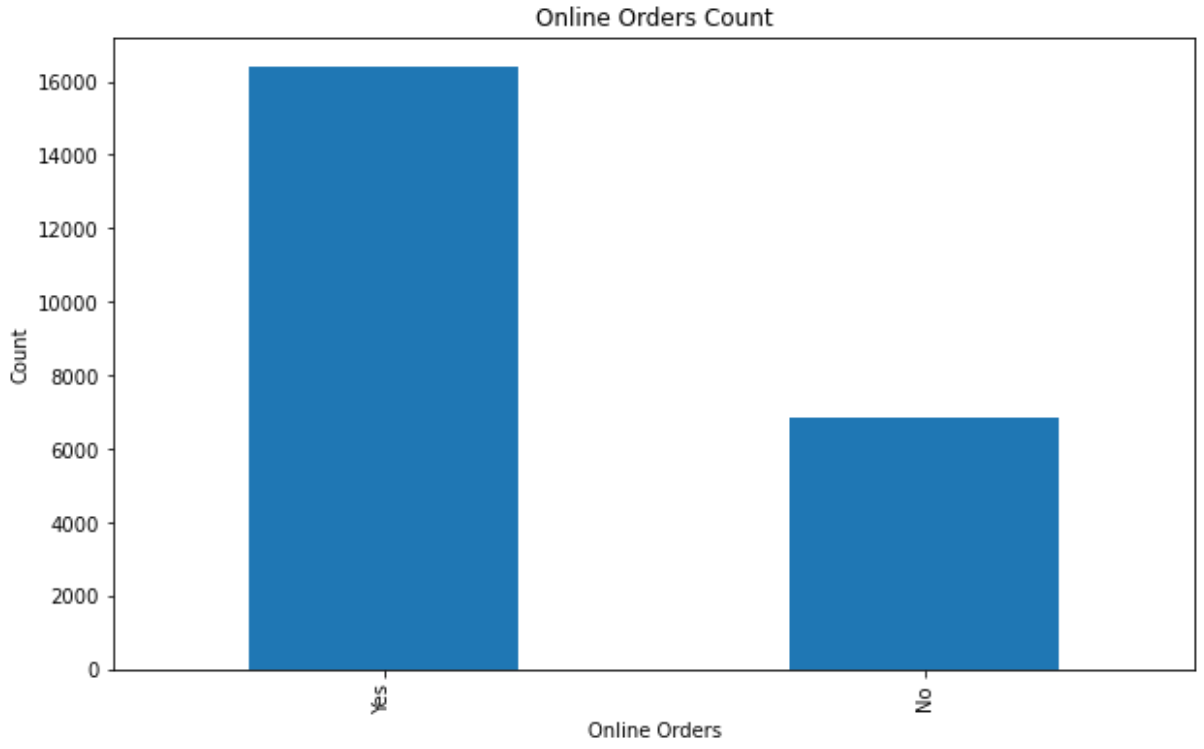
Out[14]:
```
Yes     16387
No       6872
```

```
Name: online_order, dtype: int64
```

number of online orders booked

In [15]:
```
plt.figure(figsize=(10,6))
ax=bangalore['online_order'].value_counts().plot(kind='bar')
plt.xlabel("Online Orders")
plt.ylabel("Count")
plt.title("Online Orders Count")
```

Out[15]:
```
Text(0.5, 1.0, 'Online Orders Count')
```



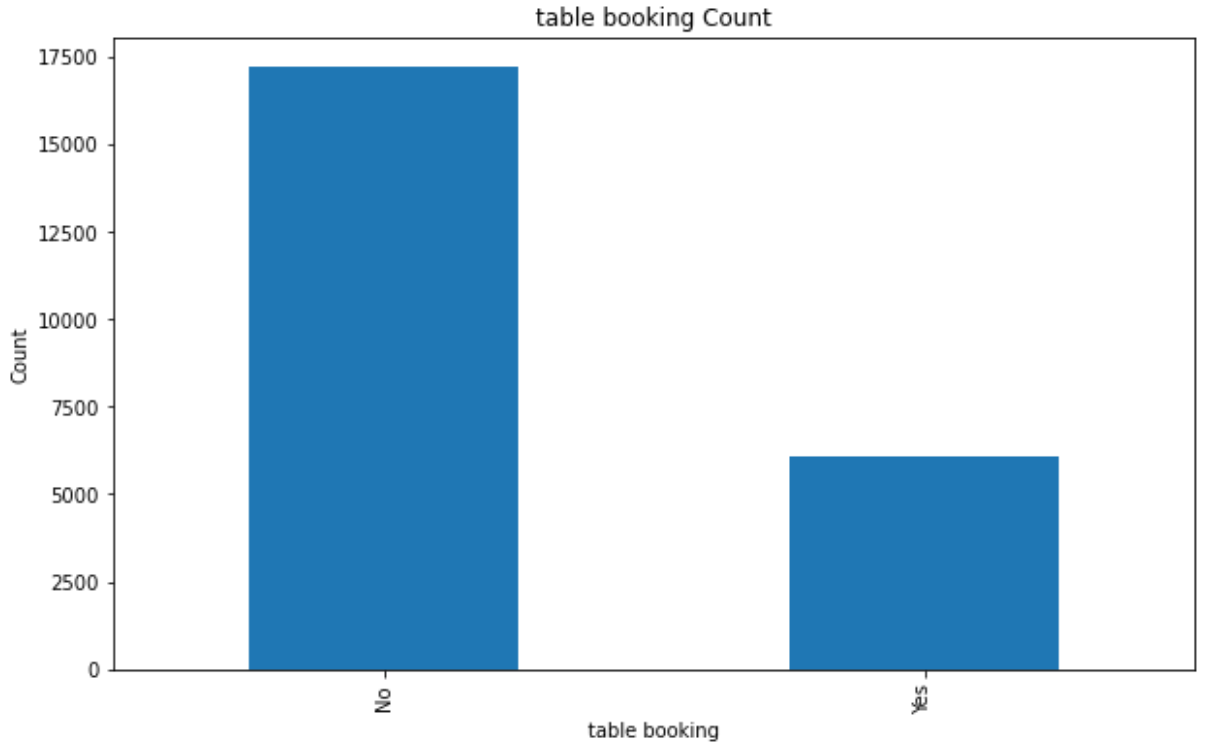
In [16]:
```
bangalore['book_table'].value_counts()
```

Out[16]:
```
No     17198
Yes     6061
Name: book_table, dtype: int64
```

number of table bookings

In [17]:
```
plt.figure(figsize=(10,6))
ax=bangalore['book_table'].value_counts().plot(kind='bar')
plt.xlabel('table booking')
plt.ylabel("Count")
plt.title("table booking Count")
```

Out[17]:
```
Text(0.5, 1.0, 'table booking Count')
```



In [18]:
```
bangalore[bangalore['book_table']=='No']['rate'].describe()
```

Out[18]:
```
count    17198.000000
mean         3.816130
std          0.429758
min          1.800000
25%          3.700000
50%          3.900000
75%          4.100000
max          4.900000
Name: rate, dtype: float64
```

In [ ]:
```
#This shows the average rating of restaurant without table booking opti
on is 3.8
```

In [19]:
```
bangalore[bangalore['book_table']=='Yes']['rate'].describe()
```

Out[19]:
```
count    6061.000000
mean        4.160370
std         0.291508
min         2.200000
25%         4.000000
50%         4.200000
75%         4.400000
max         4.900000
Name: rate, dtype: float64
```

In [ ]:
```
#This shows the average rating of resaturant with table booking is 4.1
```

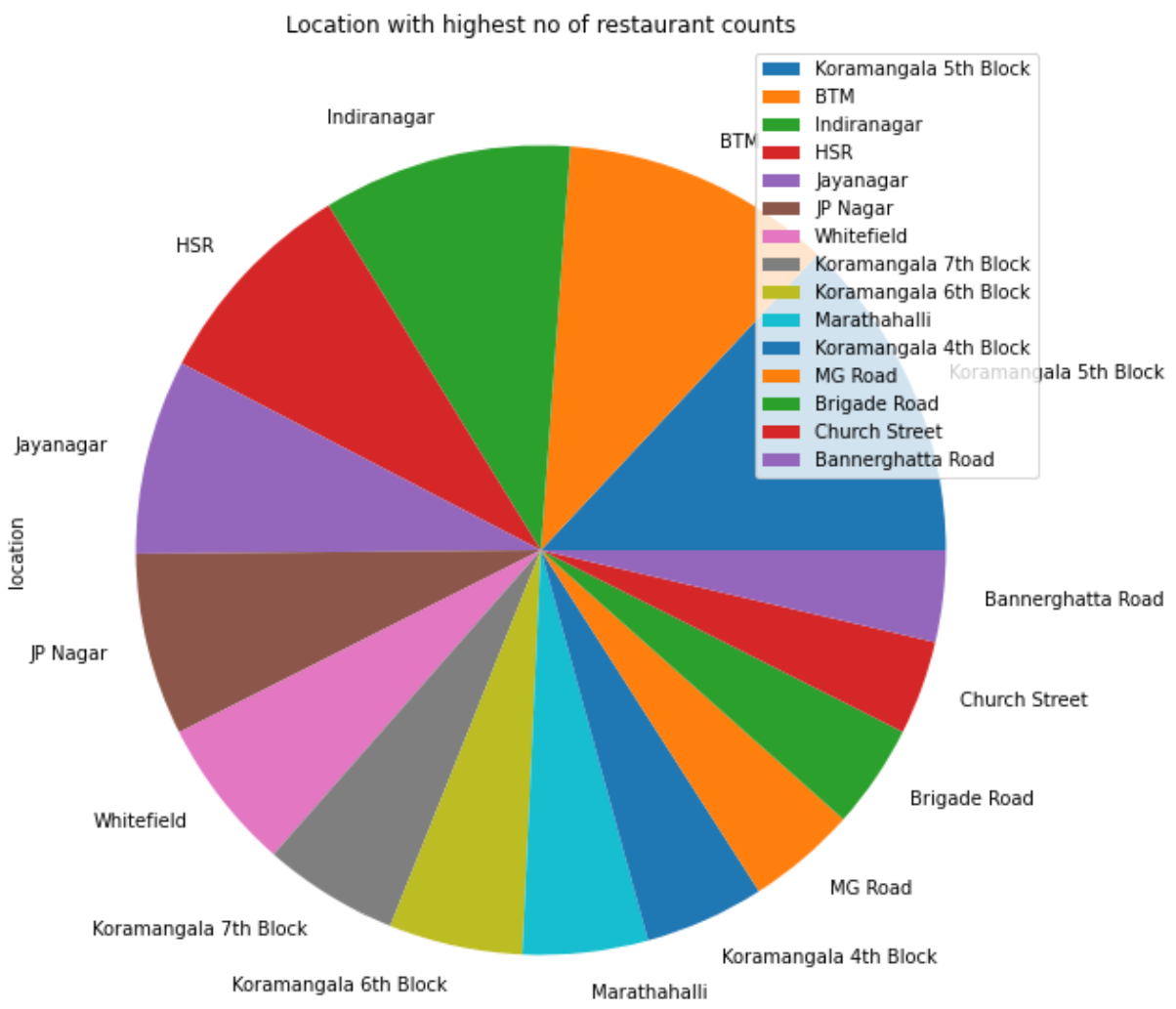
In [20]:
```
bangalore['location'].value_counts().head(15)
```

Out[20]:
```
Koramangala 5th Block    1776
BTM                      1474
Indiranagar              1355
HSR                      1170
Jayanagar                1060
JP Nagar                  993
Whitefield                832
```

```
Koramangala 7th Block     736
Koramangala 6th Block     728
Marathahalli              681
Koramangala 4th Block     652
MG Road                   596
Brigade Road              568
Church Street             512
Bannerghatta Road         498
Name: location, dtype: int64
```

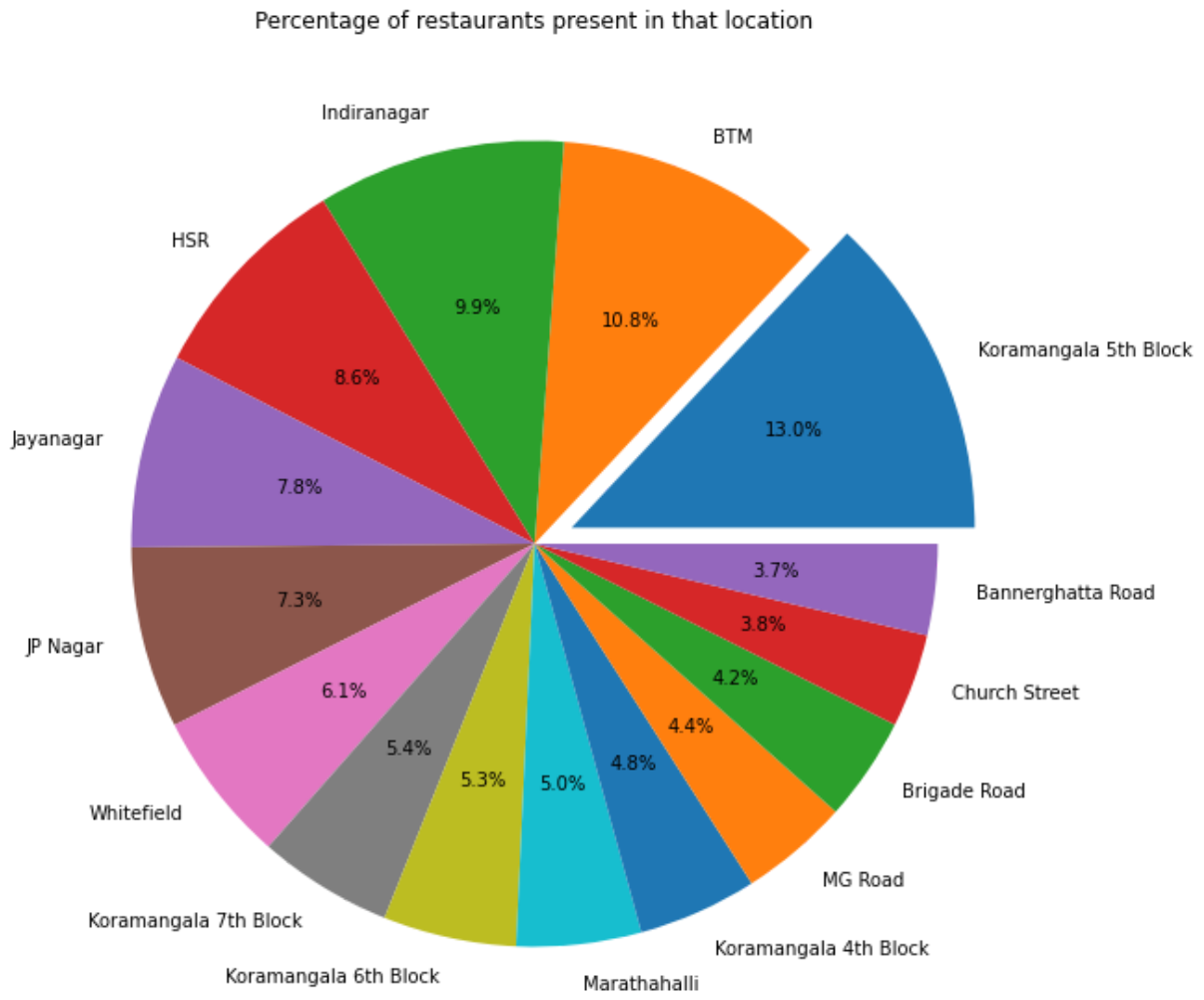
In [21]:
```
plt.figure(figsize=(10,10))
bangalore['location'].value_counts()[:15].plot(kind='pie')
plt.title('Location with highest no of restaurant counts')
plt.legend()
```

Out[21]:
```
<matplotlib.legend.Legend at 0x223cbe26610>
```



In [22]:
```
plt.figure(figsize=(10,10))
names=bangalore['location'].value_counts()[:15].index
values=bangalore['location'].value_counts()[:15].values
explode=(0.1,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
plt.pie(values,explode,autopct='%0.1f%%',labels=names)
```

```
plt.title("Percentage of restaurants present in that location")
plt.show()
```



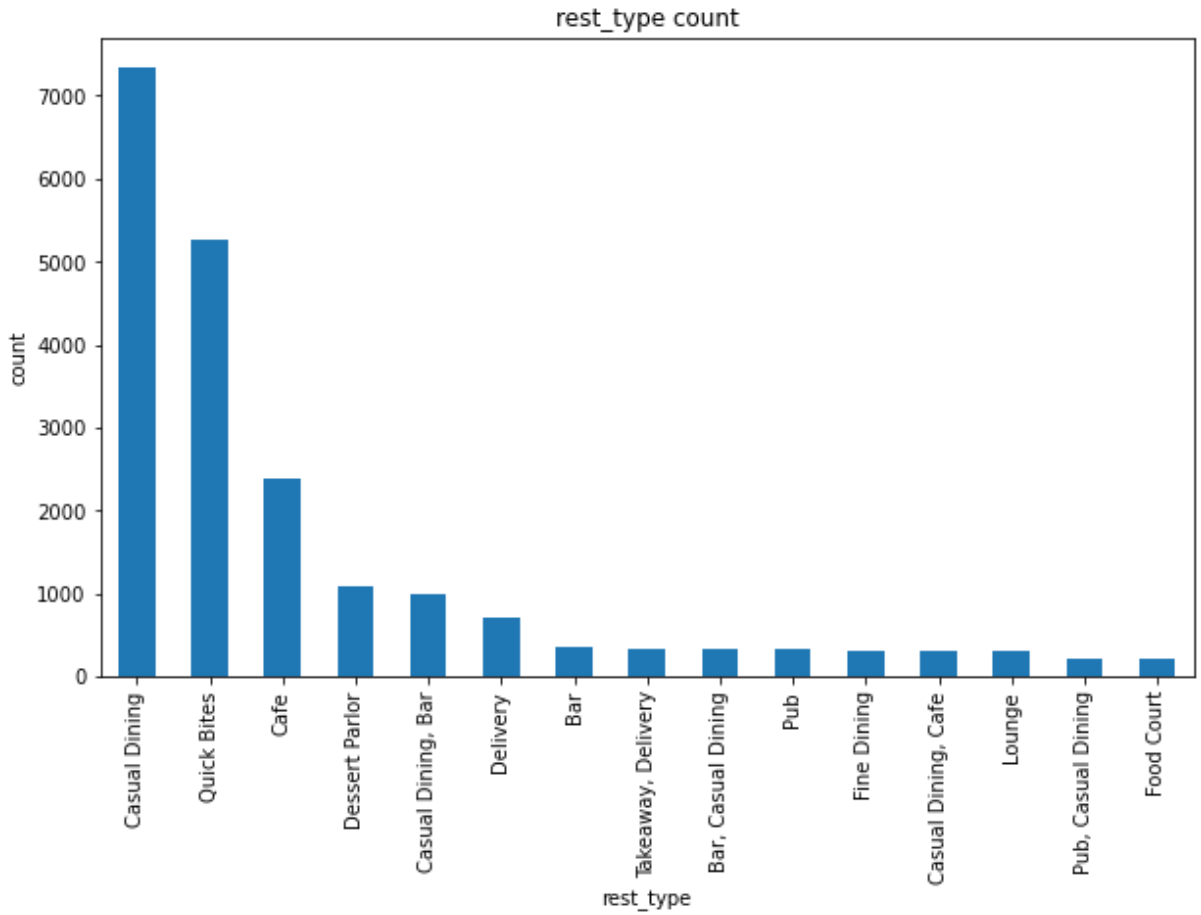
In [23]:
```
bangalore['rest_type'].value_counts().head(10)
```

Out[23]:
```
Casual Dining          7331
Quick Bites            5253
```

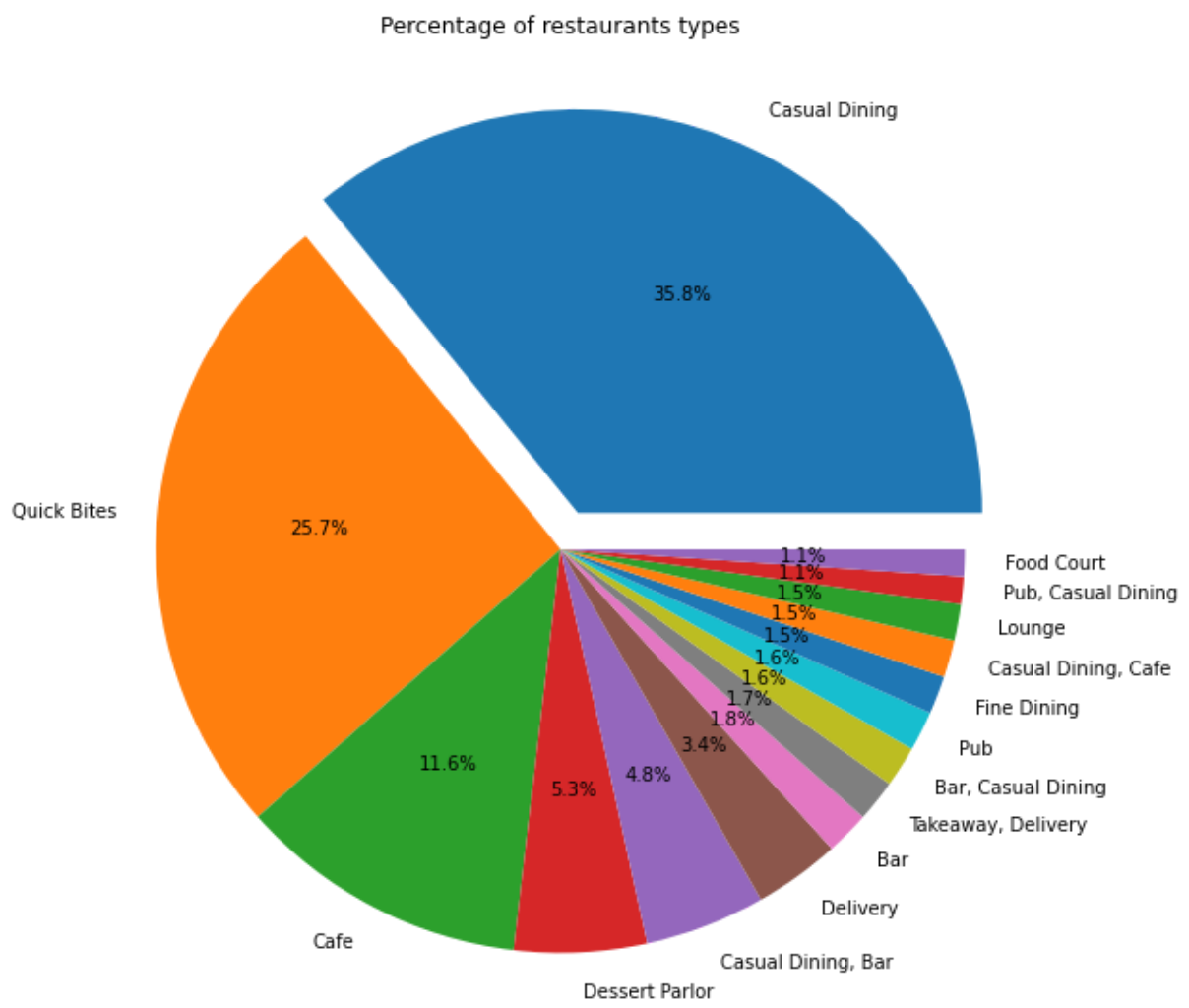
```
Cafe                    2375
Dessert Parlor          1083
Casual Dining, Bar       985
Delivery                 704
Bar                      358
Takeaway, Delivery       342
Bar, Casual Dining       336
Pub                      325
Name: rest_type, dtype: int64
```

In [24]:
```
plt.figure(figsize=(10,6))
ax=bangalore['rest_type'].value_counts()[:15].plot(kind='bar')
plt.xlabel('rest_type')
plt.ylabel('count')
plt.title('rest_type count')
```

Out[24]:
```
Text(0.5, 1.0, 'rest_type count')
```



In [25]:
```
plt.figure(figsize=(10,10))
names=bangalore['rest_type'].value_counts()[:15].index
values=bangalore['rest_type'].value_counts()[:15].values
explode=(0.1,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
plt.pie(values,explode=explode,autopct='%0.1f%%',labels=names)
plt.title("Percentage of restaurants types")
plt.show()
```



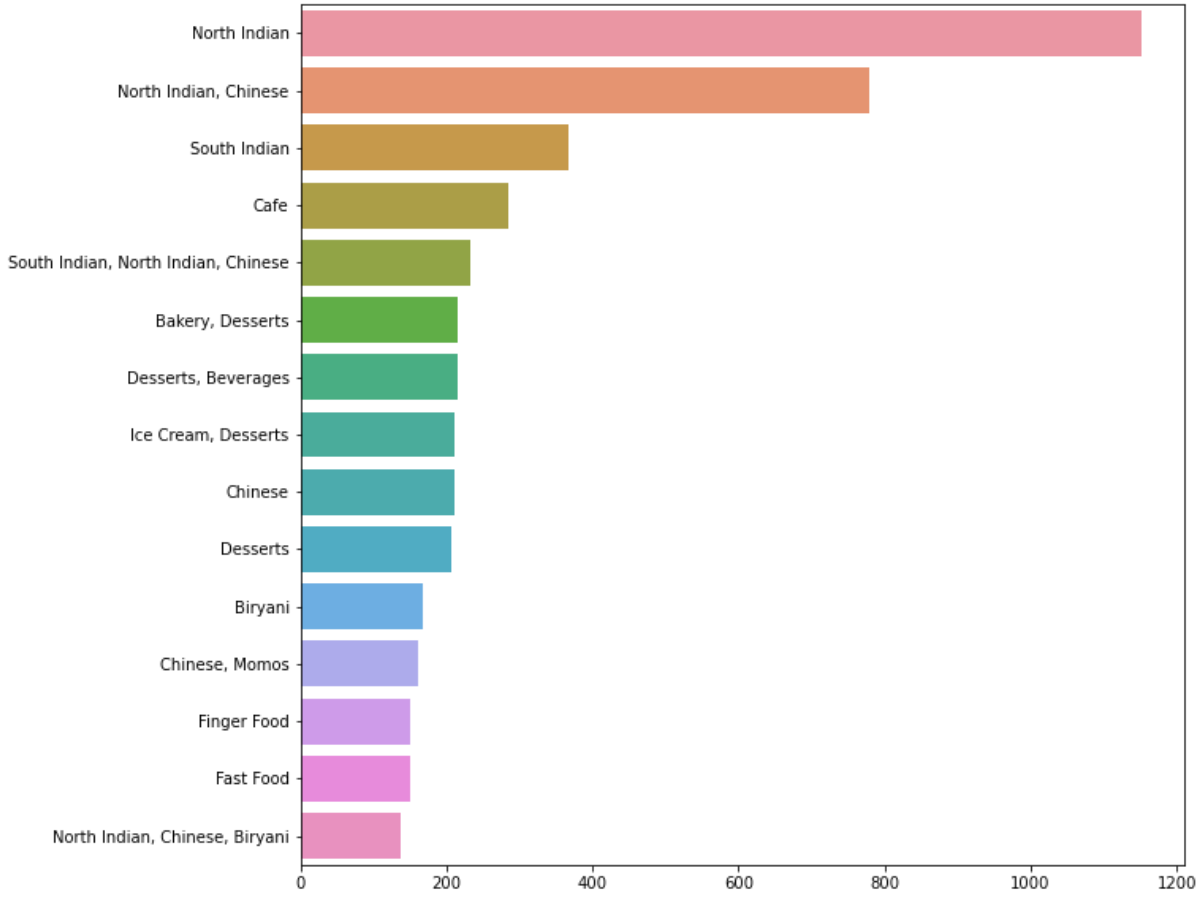
In [27]:
```
bangalore['cuisines'].value_counts()[:15]
```

Out[27]:
```
North Indian                              1152
North Indian, Chinese                      779
South Indian                               366
Cafe                                       285
South Indian, North Indian, Chinese        233
```

```
Bakery, Desserts                        216
Desserts, Beverages                     216
Ice Cream, Desserts                     212
Chinese                                 210
Desserts                                207
Biryani                                 167
Chinese, Momos                          162
Finger Food                             151
Fast Food                               150
North Indian, Chinese, Biryani          138
Name: cuisines, dtype: int64
```

In [38]:
```
plt.figure(figsize=(10,10))
sns.barplot(y=bangalore['cuisines'].value_counts()[:15].index, x=bangal
ore['cuisines'].value_counts()[:15].values,data=bangalore)
```

Out[38]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x223cc691a60>
```



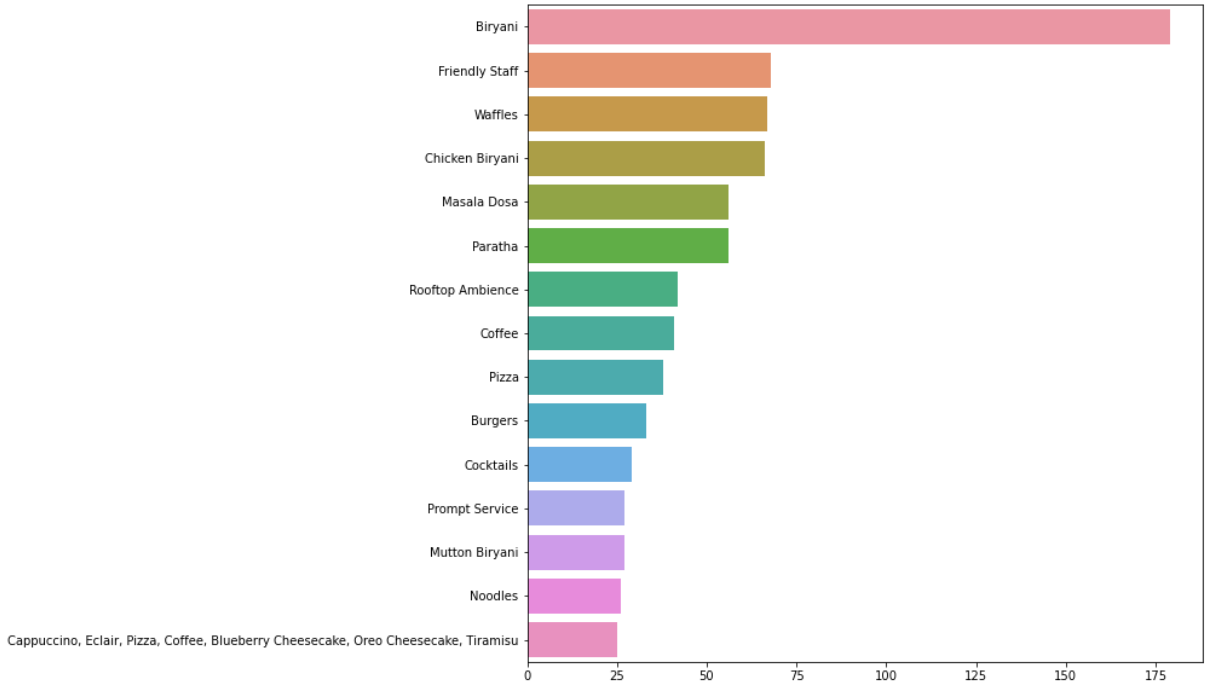
In [39]:
```
bangalore['dish_liked'].value_counts()[:15]
```

Out[39]:
```
Biryani
179
Friendly Staff
68
Waffles
67
Chicken Biryani
66
Masala Dosa
56
```

```
Paratha
56
Rooftop Ambience
42
Coffee
41
Pizza
38
Burgers
33
Cocktails
29
Prompt Service
27
Mutton Biryani
27
Noodles
26
Cappuccino, Eclair, Pizza, Coffee, Blueberry Cheesecake, Oreo Cheesecak
e, Tiramisu     25
Name: dish_liked, dtype: int64
```

In [45]:
```
plt.figure(figsize=(10,10))
sns.barplot(x=bangalore['dish_liked'].value_counts()[:15].values,y=bang
alore['dish_liked'].value_counts()[:15].index,data=bangalore)
```

Out[45]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x223ca7224c0>
```



In [40]:
```
bangalore_copy=bangalore.copy()
```

In [54]:
```
def Encode(bangalore_copy):
    for column in bangalore_copy.columns[~bangalore_copy.columns.isin(['rate', 'approx_cost(for two people)', 'votes', 'location'])]:
        bangalore_copy[column] = bangalore_copy[column].factorize()[0]
    return bangalore_copy

bangalore_copy = Encode(bangalore_copy)
```

In [55]:
```
bangalore_copy.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 23259 entries, 0 to 51715
Data columns (total 12 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
```

```
 0   name                         23259 non-null  int64
 1   online_order                 23259 non-null  int64
 2   book_table                   23259 non-null  int64
 3   rate                         23259 non-null  float64
 4   votes                        23259 non-null  int64
 5   location                     23259 non-null  object
 6   rest_type                    23259 non-null  int64
 7   dish_liked                   23259 non-null  int64
 8   cuisines                     23259 non-null  int64
 9   approx_cost(for two people)  23259 non-null  int64
 10  reviews_list                 23259 non-null  int64
 11  listed_in(type)              23259 non-null  int64
dtypes: float64(1), int64(10), object(1)
memory usage: 2.3+ MB
```

In [56]:
```
bangalore_copy.head()
```

Out[56]:

	name	online_order	book_table	rate	votes	location	rest_type	dish_liked	cuisines	ap
0	0	0	0	4.1	775	Banashankari	0	0	0	
1	1	0	1	4.1	787	Banashankari	0	1	1	
2	2	0	1	3.8	918	Banashankari	1	2	2	
3	3	1	1	3.7	88	Banashankari	2	3	3	
4	4	1	1	3.8	166	Basavanagudi	0	4	4	

In [ ]:
```
#Correlation between different variables
```

In [57]:
```
bangalore_copy.corr()
```

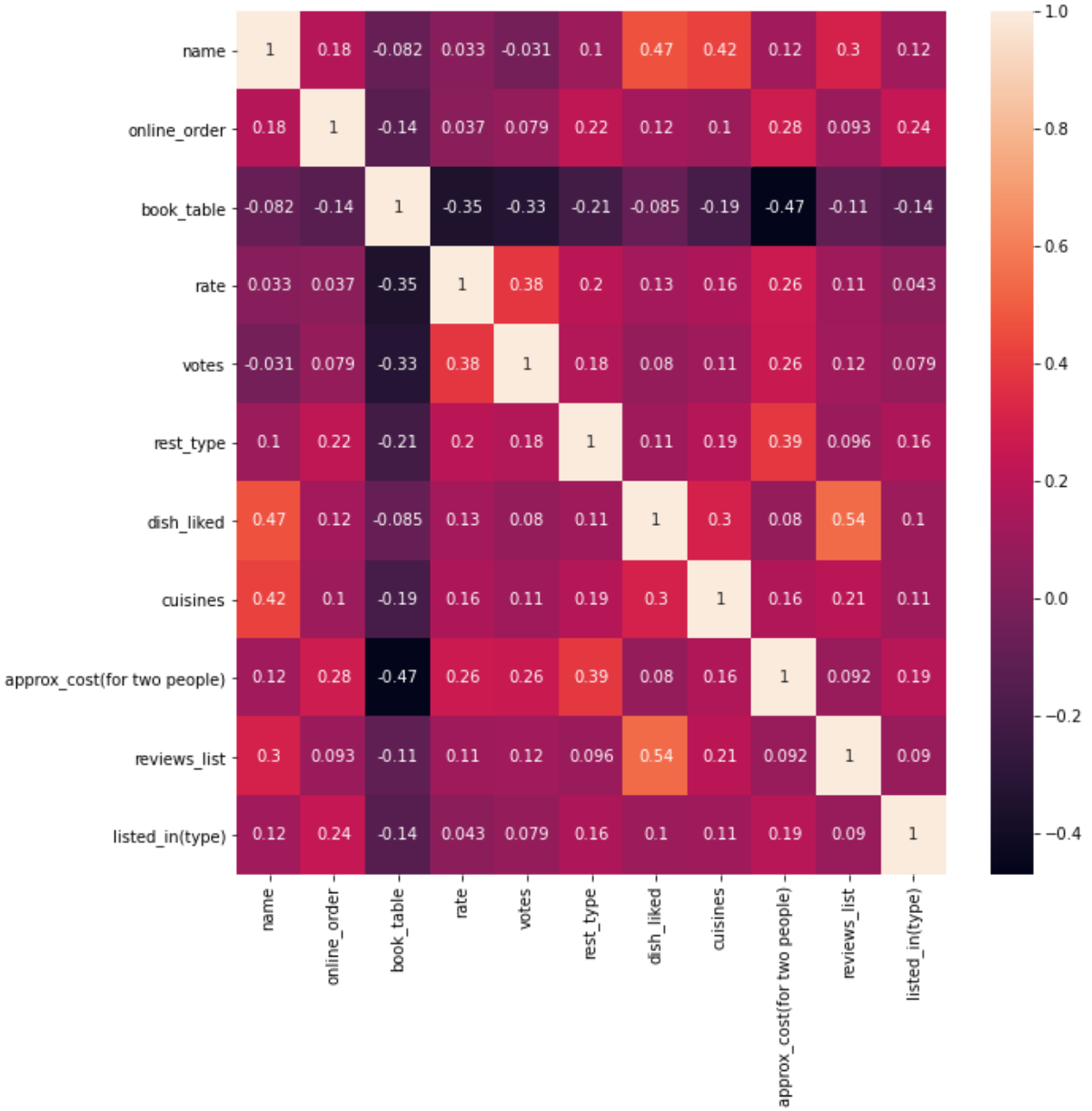
Out[57]:

	name	online_order	book_table	rate	votes	rest_type	dish_liked
name	1.000000	0.181296	-0.081610	0.032731	-0.031161	0.102107	0.467755

	name	online_order	book_table	rate	votes	rest_type	dish_liked
online_order	0.181296	1.000000	-0.135944	0.037093	0.079082	0.223510	0.123085
book_table	-0.081610	-0.135944	1.000000	-0.354657	-0.325880	-0.210772	-0.085114
rate	0.032731	0.037093	-0.354657	1.000000	0.381800	0.204149	0.131325
votes	-0.031161	0.079082	-0.325880	0.381800	1.000000	0.177859	0.079826
rest_type	0.102107	0.223510	-0.210772	0.204149	0.177859	1.000000	0.109708
dish_liked	0.467755	0.123085	-0.085114	0.131325	0.079826	0.109708	1.000000
cuisines	0.420979	0.100411	-0.194784	0.156985	0.108806	0.185605	0.300367
approx_cost(for two people)	0.120569	0.275346	-0.469933	0.264935	0.255292	0.386023	0.080273
reviews_list	0.303316	0.093369	-0.108163	0.107299	0.116729	0.095999	0.543024
listed_in(type)	0.115831	0.236674	-0.137728	0.042620	0.078958	0.157197	0.100231

In [59]:
```
plt.figure(figsize=(10,10))
sns.heatmap(bangalore_copy.corr(),annot= True)
```

Out[59]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x223cbe1d3a0>
```



In [ ]:
```
# The above heatmap shows following relations-
# 1. Dish_liked has maximium impact on reviews of a restaurant.
```

```
# 2. Rest_type has maximium impact on cost for two people.
# 3. Cost for two people has maximium positive impact on online orders.
# 4. Cost for two people has maximium negative impact on table booking
s.
```