In [203]:

```
import pandas as pd
import numpy as np

import seaborn as sns
import matplotlib.pyplot as plt
import plotly.graph_objects as go
%matplotlib inline
```

In [133]:

```
Bangalore = pd.read_csv (r'C:\Users\Hp\Downloads\Bangalore.csv')
BangaloreCopy = pd.read_csv (r'C:\Users\Hp\Downloads\Bangalore.csv')
Chennai=pd.read_csv (r'C:\Users\Hp\Downloads\Chennai.csv')
Kolkata=pd.read_csv (r'C:\Users\Hp\Downloads\Kolkata.csv')
Hyderabad=pd.read_csv (r'C:\Users\Hp\Downloads\Hyderabad.csv')
Delhi=pd.read_csv (r'C:\Users\Hp\Downloads\Delhi.csv')
Mumbai=pd.read_csv (r'C:\Users\Hp\Downloads\Mumbai.csv')
```

In [134]:

```
Bangalore.head(10)
```

Out[134]:

	Price	Area	Location	No. of Bedrooms	Resale	MaintenanceStaff	Gymnasium	SwimmingPool	LandscapedGardens	Jo
0	30000000	3340	JP Nagar Phase 1	4	0	1	1	1	1	
1	7888000	1045	Dasarahalli on Tumkur Road	2	0	0	1	1	1	
2	4866000	1179	Kannur on Thanisandra Main Road	2	0	0	1	1	1	
3	8358000	1675	Doddanekundi	3	0	0	0	0	0	
4	6845000	1670	Kengeri	3	0	1	1	1	1	
5	6797000	1220	Horamavu	2	0	0	1	1	1	
6	20000000	2502	Thanisandra	4	0	0	1	1	1	
7	7105000	1438	Ramamurthy Nagar	3	0	0	1	0	0	
8	8405000	1405	Whitefield Hope Farm Junction	3	0	0	1	1	1	
9	3506000	660	Electronic City Phase 1	1	0	1	1	1	1	

10 rows × 40 columns

In [135]:

```
Bangalore['City'] = "Bangalore"
```

In [136]:

```
Bangalore.head()
```

Out[136]:

	Price	Area	Location	No. of Bedrooms	Resale	MaintenanceStaff	Gymnasium	SwimmingPool	LandscapedGardens	Jo

	Price	Area	Location	No. of Bedrooms	Resale	MaintenanceStaff	Gymnasium	SwimmingPool	LandscapedGardens	Jo
0	30000000	3340	JP Nagar Phase 1	4	0	1	1	1	1	
1	7888000	1045	Dasarahalli on Tumkur Road	2	0	0	1	1	1	
2	4866000	1179	Kannur on Thanisandra Main Road	2	0	0	1	1	1	
3	8358000	1675	Doddanekundi	3	0	0	0	0	0	
4	6845000	1670	Kengeri	3	0	1	1	1	1	

5 rows × 41 columns

In [137]:

```
Bangalore.shape
```

Out[137]:

```
(6207, 41)
```

In [ ]:

```

```

In [138]:

```
Bangalore.drop_duplicates()
```

Out[138]:

	Price	Area	Location	No. of Bedrooms	Resale	MaintenanceStaff	Gymnasium	SwimmingPool	LandscapedGardens
0	30000000	3340	JP Nagar Phase 1	4	0	1	1	1	1
1	7888000	1045	Dasarahalli on Tumkur Road	2	0	0	1	1	1
2	4866000	1179	Kannur on Thanisandra Main Road	2	0	0	1	1	1
3	8358000	1675	Doddanekundi	3	0	0	0	0	0
4	6845000	1670	Kengeri	3	0	1	1	1	1
...	...	...	...	...	...	...	...	...	...
6202	5364000	590	Chandapura	1	0	9	9	9	9
6203	8716000	1179	Kasavanahalli	2	0	9	9	9	9
6204	7373000	1143	Kasavanahalli	2	0	9	9	9	9
6205	4985000	1680	Kasavanahalli	3	0	9	9	9	9
6206	10900000	1162	Kasavanahalli	2	0	9	9	9	9

5521 rows × 41 columns

In [139]:

```
'''Bangalore.Area.dropna()'''
```

Out[139]:

```
'Bangalore.Area.dropna()'
```

In [140]:

```
Bangalore.shape
```

Out[140]:

```
(6207, 41)
```

In [141]:

```
Bangalore.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6207 entries, 0 to 6206
Data columns (total 41 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   Price                6207 non-null   int64 
 1   Area                 6207 non-null   int64 
 2   Location             6207 non-null   object
 3   No. of Bedrooms      6207 non-null   int64 
 4   Resale               6207 non-null   int64 
 5   MaintenanceStaff     6207 non-null   int64 
 6   Gymnasium            6207 non-null   int64 
 7   SwimmingPool         6207 non-null   int64 
 8   LandscapedGardens    6207 non-null   int64 
 9   JoggingTrack         6207 non-null   int64 
 10  RainWaterHarvesting  6207 non-null   int64 
 11  IndoorGames          6207 non-null   int64 
 12  ShoppingMall         6207 non-null   int64 
 13  Intercom             6207 non-null   int64 
 14  SportsFacility       6207 non-null   int64 
 15  ATM                  6207 non-null   int64 
 16  ClubHouse            6207 non-null   int64 
 17  School               6207 non-null   int64 
 18  24X7Security         6207 non-null   int64 
 19  PowerBackup          6207 non-null   int64 
 20  CarParking           6207 non-null   int64 
 21  StaffQuarter         6207 non-null   int64 
 22  Cafeteria            6207 non-null   int64 
 23  MultipurposeRoom     6207 non-null   int64 
 24  Hospital             6207 non-null   int64 
 25  WashingMachine       6207 non-null   int64 
 26  Gasconnection        6207 non-null   int64 
 27  AC                   6207 non-null   int64 
 28  Wifi                 6207 non-null   int64 
 29  Children'splayarea   6207 non-null   int64 
 30  LiftAvailable        6207 non-null   int64 
 31  BED                  6207 non-null   int64 
 32  VaastuCompliant      6207 non-null   int64 
 33  Microwave            6207 non-null   int64 
 34  GolfCourse           6207 non-null   int64 
 35  TV                   6207 non-null   int64 
 36  DiningTable          6207 non-null   int64 
 37  Sofa                 6207 non-null   int64 
 38  Wardrobe             6207 non-null   int64 
 39  Refrigerator         6207 non-null   int64 
 40  City                 6207 non-null   object
dtypes: int64(39), object(2)
memory usage: 1.9+ MB
```

In [ ]:

```

```

In [142]:

```
'''location=Bangalore.Location.value_counts(dropna=False)
location'''
```

Out[142]:

```
'location=Bangalore.Location.value_counts(dropna=False)\nlocation'
```

In [143]:

```
'''df[df.population > 1000000000]'''
```

Out[143]:

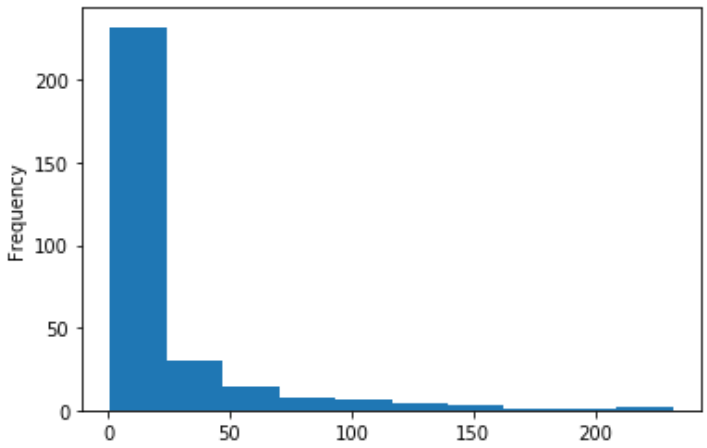
```
'df[df.population > 1000000000]'
```

In [144]:

```
location.plot(kind='hist')
```

Out[144]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2ad73a56548>
```



In [ ]:

```

```

In [145]:

```
Bangalore.describe()
```

Out[145]:

	Price	Area	No. of Bedrooms	Resale	MaintenanceStaff	Gymnasium	SwimmingPool	LandscapedGard
count	6.207000e+03	6207.000000	6207.000000	6207.000000	6207.000000	6207.000000	6207.000000	6207.000
mean	1.058510e+07	1526.094248	2.556952	0.078782	6.208797	6.461576	6.436121	6.382
std	1.410943e+07	764.845609	0.694300	0.269420	4.126883	3.752421	3.792567	3.875
min	2.000000e+06	415.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000
25%	5.000000e+06	1110.000000	2.000000	0.000000	0.000000	1.000000	1.000000	1.000
50%	7.368000e+06	1340.000000	3.000000	0.000000	9.000000	9.000000	9.000000	9.000
75%	1.070000e+07	1662.500000	3.000000	0.000000	9.000000	9.000000	9.000000	9.000
max	3.000000e+08	9900.000000	7.000000	1.000000	9.000000	9.000000	9.000000	9.000

8 rows × 39 columns

In [146]:

```
Bangalore.head()
```

Out[146]:

	Price	Area	Location	No. of Bedrooms	Resale	MaintenanceStaff	Gymnasium	SwimmingPool	LandscapedGardens	Jo
0	30000000	3340	JP Nagar Phase 1	4	0	1	1	1	1	

	Price	Area	Location	No. of Bedrooms	Resale	MaintenanceStaff	Gymnasium	SwimmingPool	LandscapedGardens	Jo
1	7888000	1045	Dasarahalli on Tumkur Road	2	0	0	1	1	1	
2	4866000	1179	Kannur on Thanisandra Main Road	2	0	0	1	1	1	
3	8358000	1675	Doddanekundi	3	0	0	0	0	0	
4	6845000	1670	Kengeri	3	0	1	1	1	1	

5 rows × 41 columns

In [152]:

```
Bangalore.set_index(['Location'],inplace=True)
```

In [153]:

```
Bangalore
```

Out[153]:

Location	Price	Area	No. of Bedrooms	Resale	MaintenanceStaff	Gymnasium	SwimmingPool	LandscapedGardens	Joggi
JP Nagar Phase 1	30000000	3340	4	0	1	1	1	1	
Dasarahalli on Tumkur Road	7888000	1045	2	0	0	1	1	1	
Kannur on Thanisandra Main Road	4866000	1179	2	0	0	1	1	1	
Doddanekundi	8358000	1675	3	0	0	0	0	0	
Kengeri	6845000	1670	3	0	1	1	1	1	
...	...	...	...	...	...	...	...	...	
Chandapura	5364000	590	1	0	9	9	9	9	
Kasavanahalli	8716000	1179	2	0	9	9	9	9	
Kasavanahalli	7373000	1143	2	0	9	9	9	9	
Kasavanahalli	4985000	1680	3	0	9	9	9	9	
Kasavanahalli	10900000	1162	2	0	9	9	9	9	

6207 rows × 40 columns

In [154]:

```
Bangalore.sort_index(inplace=True)
Bangalore
```

Out[154]:

Location	Price	Area	No. of Bedrooms	Resale	MaintenanceStaff	Gymnasium	SwimmingPool	LandscapedGardens	JoggingTr
5th Phase	8500000	1430	3	0	9	9	9	9	
5th Phase	16000000	1185	2	0	9	9	9	9	
5th Phase	13200000	1180	2	0	9	9	9	9	

Location	Price	Area	No. of Bedrooms	Resale	MaintenanceStaff	Gymnasium	SwimmingPool	LandscapedGardens	JoggingTr
5th Phase	4800000	1180	2	0	9	9	9	9	
5th Phase	7498999	1180	2	0	9	9	9	9	
...	...	...	...	...	...	...	...	...	
sarjapura attibele road	6580000	926	3	0	9	9	9	9	
sarjapura attibele road	17600000	1100	2	0	9	9	9	9	
sarjapura attibele road	3800000	1468	3	0	9	9	9	9	
sarjapura attibele road	10000000	1017	2	0	9	9	9	9	
sarjapura attibele road	5750000	943	2	0	9	9	9	9	

6207 rows × 40 columns

Since for a set of houses, nothing was mentioned about certain amenities, '9' was used to mark such values, which could indicate the absense of information about the apartment but these values dont't ascertain the absence of such a feature in real life.

We will be dropping these values

In [174]:

```
Bangalore.replace(9, np.nan, inplace=True)
```

In [175]:

```
Bangalore=Bangalore.dropna()
```

In [176]:

```
Bangalore.isnull().sum()
```

Out[176]:

```
Price                 0
Area                  0
No. of Bedrooms       0
Resale                0
MaintenanceStaff      0
Gymnasium             0
SwimmingPool          0
LandscapedGardens     0
JoggingTrack          0
RainWaterHarvesting   0
IndoorGames           0
ShoppingMall          0
Intercom              0
SportsFacility        0
ATM                   0
ClubHouse             0
School                0
24X7Security          0
PowerBackup           0
CarParking            0
StaffQuarter          0
Cafeteria             0
MultipurposeRoom      0
```

```
Hospital              0
WashingMachine        0
Gasconnection         0
AC                    0
Wifi                  0
Children'splayarea    0
LiftAvailable         0
BED                   0
VaastuCompliant       0
Microwave             0
GolfCourse            0
TV                    0
DiningTable           0
Sofa                  0
Wardrobe              0
Refrigerator          0
City                  0
dtype: int64
```

In [155]:

```
Bangalore.index.value_counts().to_frame()
```

Out[155]:

	Location
Electronic City Phase 2	**232**
RR Nagar	**217**
Begur	**186**
Varthur	**168**
Kumaraswamy Layout	**154**
...	**...**
Sindhi Colony	**1**
Sanjeevini Nagar	**1**
Richards Town	**1**
Kodipalya	**1**
K P Main Road	**1**

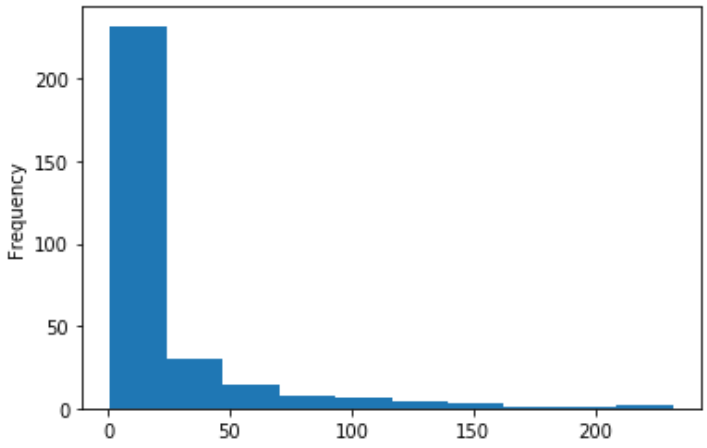
302 rows × 1 columns

In [167]:

```
Bangalore.index.value_counts().plot(kind='hist')
```

Out[167]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2ad7b78ba88>
```



In [170]:

```
Bangalore.sort_values(by=['Price','Area'])
```

Out[170]:

Location	Price	Area	No. of Bedrooms	Resale	MaintenanceStaff	Gymnasium	SwimmingPool	LandscapedGardens	Jogg
Sarjapur Road	2000000	2730	4	0	9	9	9	9	
Anekal City	2096000	645	1	0	0	1	1	1	
Electronic City Phase 2	2100000	1477	3	0	9	9	9	9	
Banashankari	2100000	2814	4	1	9	9	9	9	
Chandapura	2150000	873	2	0	9	9	9	9	
...	...	...	...	...	...	...	...	...	
Kumbalgodu	200000000	1070	2	0	9	9	9	9	
Hebbal	202700000	9900	5	0	0	1	1	0	
Manyata Tech Park Nagawara	260000000	731	2	0	9	9	9	9	
Anekal City	270000000	1225	3	0	9	9	9	9	
Bommasandra	300000000	785	2	0	9	9	9	9	

6207 rows × 40 columns

Modifing prices to lakh

In [178]:

```
Bangalore['Price'] = Bangalore['Price']/100000
```

```
G:\anaconda\lib\site-packages\ipykernel_launcher.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_g
uide/indexing.html#returning-a-view-versus-a-copy
  """Entry point for launching an IPython kernel.
```

In [180]:

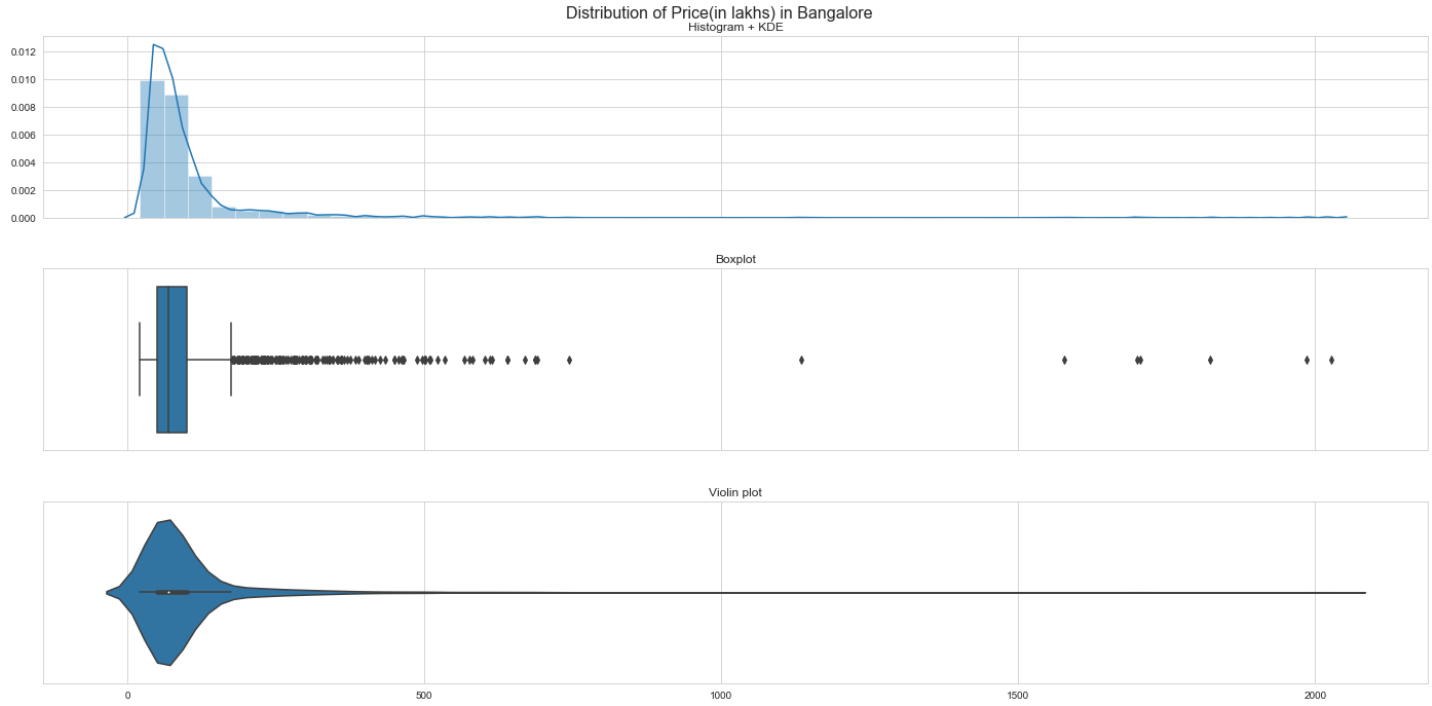
```
sns.set_style("whitegrid")
```

In [188]:

```
def triple_plot(x, title):
    fig, ax = plt.subplots(3,1,figsize=(20,10),sharex=True)
    sns.distplot(x, ax=ax[0])
    ax[0].set(xlabel=None)
    ax[0].set_title('Histogram + KDE')
    sns.boxplot(x, ax=ax[1])
    ax[1].set(xlabel=None)
    ax[1].set_title('Boxplot')
    sns.violinplot(x, ax=ax[2])
    ax[2].set(xlabel=None)
    ax[2].set_title('Violin plot')
    fig.suptitle(title, fontsize=16)
    plt.tight_layout(pad=3.0)
    plt.show()
```

In [192]:

```
triple_plot(Bangalore['Price'],'Distribution of Price(in lakhs) in Bangalore')
```



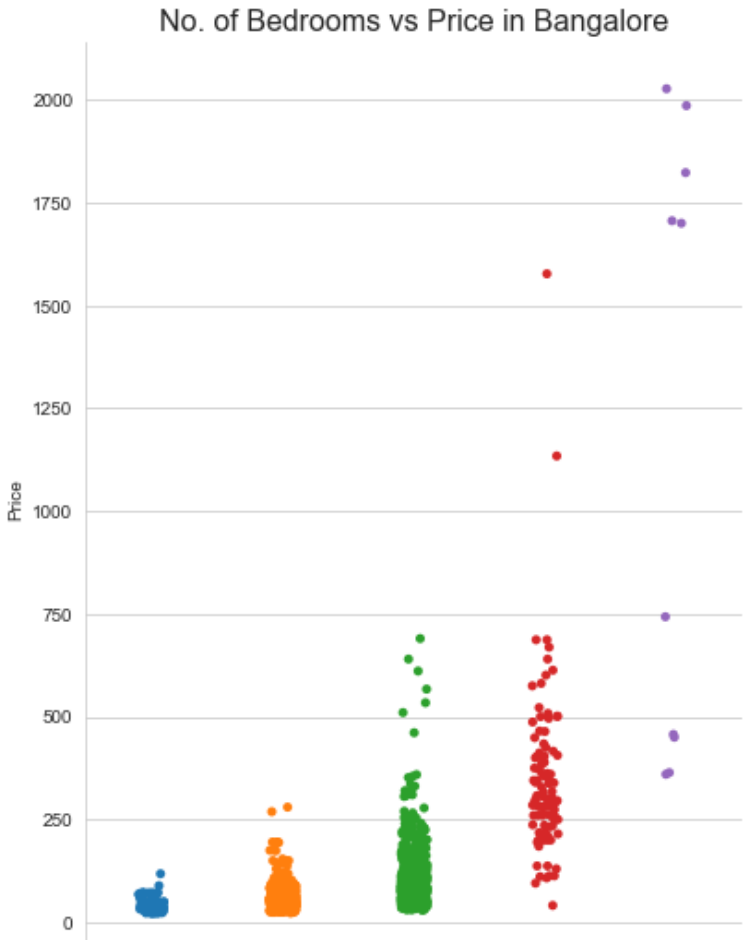
def cat_plot(data,title,p): sns.catplot(x="No. of Bedrooms", y="Price", data=data,palette = p) plt.title('No. of Bedrooms vs Price in '+ title,size=16) plt.gcf().set_size_inches(6,8) plt.show()

In [194]:

```
def cat_plot(data,title):
    sns.catplot(x="No. of Bedrooms", y="Price", data=data)
    plt.title('No. of Bedrooms vs Price in '+ title,size=16)
    plt.gcf().set_size_inches(6,8)
    plt.show()
```

In [196]:

```
cat_plot(Bangalore,'Bangalore')
```

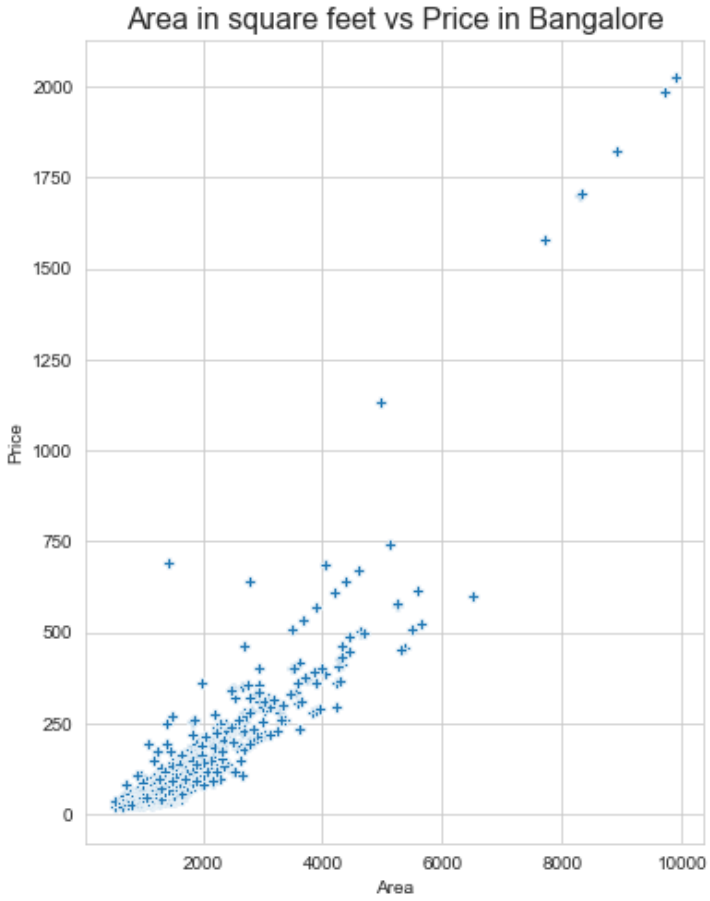


In [197]:

```
def scatter_plot(data,title):
    sns.scatterplot(x="Area", y="Price", data=data,marker="P")
    plt.title('Area in square feet vs Price in '+ title,size=16)
    plt.gcf().set_size_inches(6,8)
    plt.show()
```

In [198]:

```
scatter_plot(Bangalore,'Bangalore')
```



In [206]:

```
def pie_chart(df,link,addAll = True):
    df = df.iloc [:,5:-2]
    fig = go.Figure()
    for column in df.columns.to_list():
        val = df[column].value_counts().rename_axis('unique_values').reset_index(name='v
al_count')
        labels = val['unique_values']
        values = val['val_count']
        fig.add_trace(
            go.Pie(
                labels=labels,
                values=values,

            )
        )
        button_all = dict(label = 'All',
                      method = 'update',
                      args = [{'visible': df.columns.isin(df.columns),
                               'title': 'All',
                               'showlegend':True}])


    def create_layout_button(column):
```

```
        return dict(label = column,
                    method = 'update',
                    args = [{'visible': df.columns.isin([column]),
                             'title': column,
                             'showlegend': True}])
    fig.add_layout_image(
    dict(
        source=link,
        xref="paper", yref="paper",
        x=0.5, y=0.95,
        sizex=0.9, sizey=0.6,
        xanchor="center", yanchor="bottom"
    )
    )
    fig.update_layout(
        updatemenus=[go.layout.Updatemenu(
            active = 0,
            buttons = ([button_all] * addAll) + list(df.columns.map(lambda column: creat
e_layout_button(column)))
            )
        ])

    fig.show()
```

In [208]:

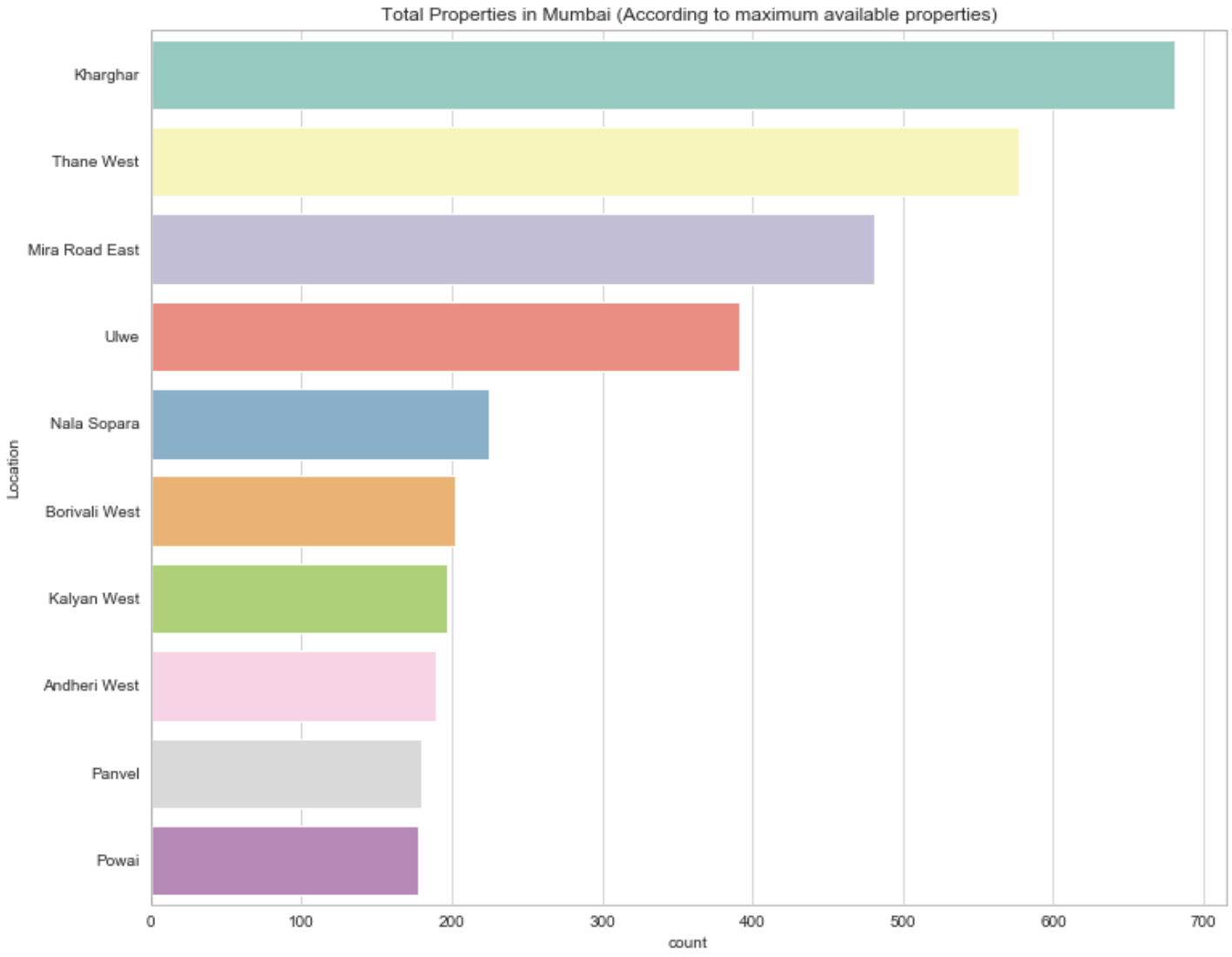
```
pie_chart(Mumbai,"https://i.imgur.com/OEr0Lw2.png")
```

In [214]:

```
#df = mumbai.Location.value_counts().index[:10]
plt.figure(figsize=(12,10))
plt.title('Total Properties in Mumbai (According to maximum available properties)')
sns.countplot(y='Location',data=Mumbai,order= Mumbai.Location.value_counts().index[:10],
palette = "Set3")
```

Out[214]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2ad00b7ddc8>
```

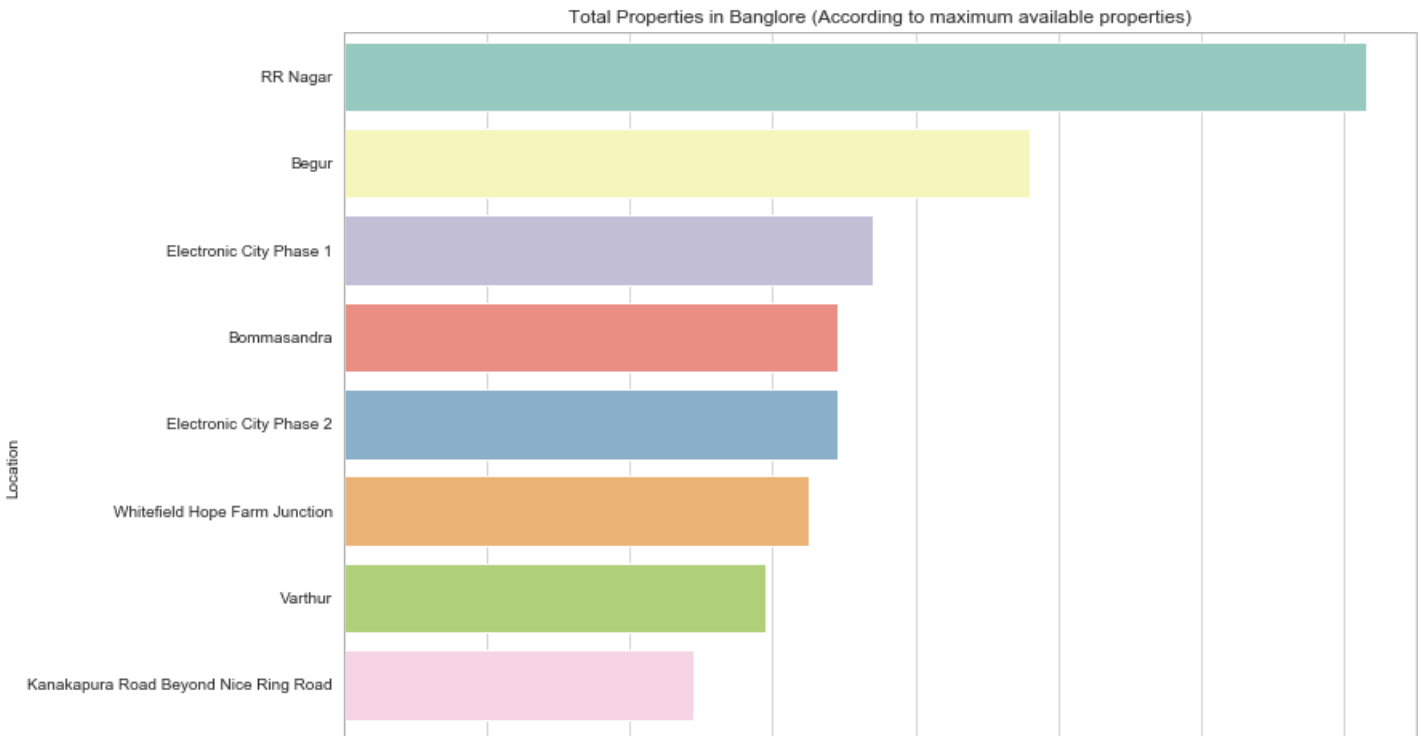


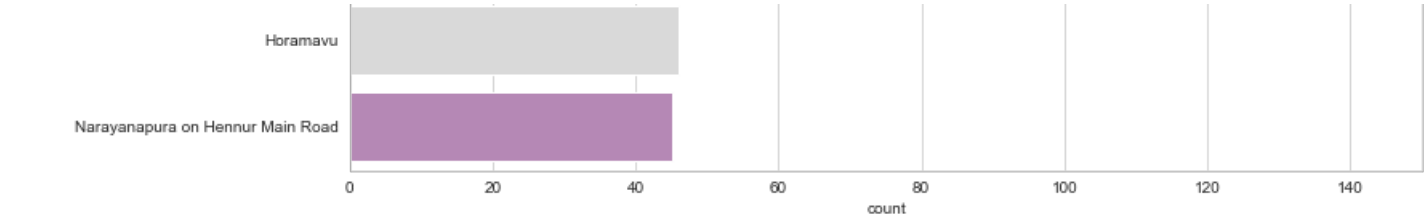
In [218]:

```
Bangalore.reset_index(inplace=True)
plt.figure(figsize=(12,10))
plt.title('Total Properties in Banglore (According to maximum available properties)')
sns.countplot(y='Location',data=Bangalore,order= Bangalore.Location.value_counts().index[
:10],palette = "Set3")
```

Out[218]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2ad00b20288>
```





Getting the most expensive place in Banglore and Mumbai

In [222]:

```
max_price_b = Bangalore[Bangalore['Price'] == Bangalore['Price'].max()]
max_price_b
```

Out[222]:

	Price	Area	Location	No. of Bedrooms	Resale	MaintenanceStaff	Gymnasium	SwimmingPool	LandscapedGardens	Jog
2083	420000000	1050	MG Road	2	0	9	9	9	9	

1 rows × 40 columns

In [223]:

```
max_price_b = Mumbai[Mumbai['Price'] == Mumbai['Price'].max()]
max_price_b
```

Out[223]:

	Price	Area	Location	No. of Bedrooms	Resale	MaintenanceStaff	Gymnasium	SwimmingPool	LandscapedGardens	Jog
2083	420000000	1050	MG Road	2	0	9	9	9	9	

1 rows × 40 columns

In [226]:

```
Bangalore_mod =Bangalore[['Price','Area','No. of Bedrooms','Gymnasium','SwimmingPool','IndoorGames','JoggingTrack']]
```

In [227]:

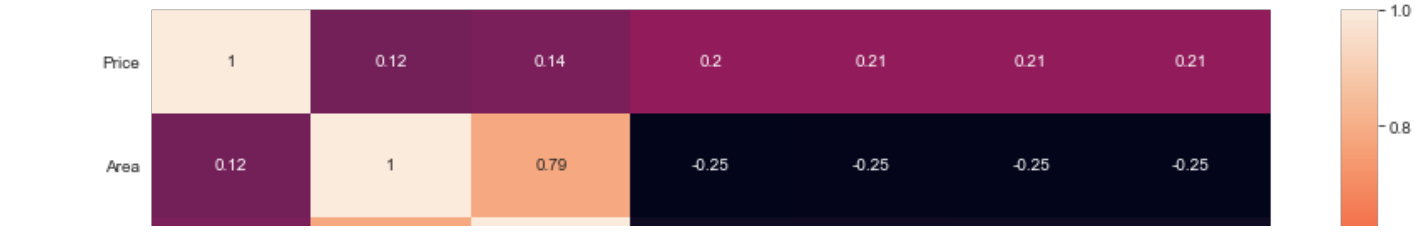
```
Mumbai_mod = Mumbai[['Price','Area','No. of Bedrooms','Gymnasium','SwimmingPool','IndoorGames','JoggingTrack']]
```

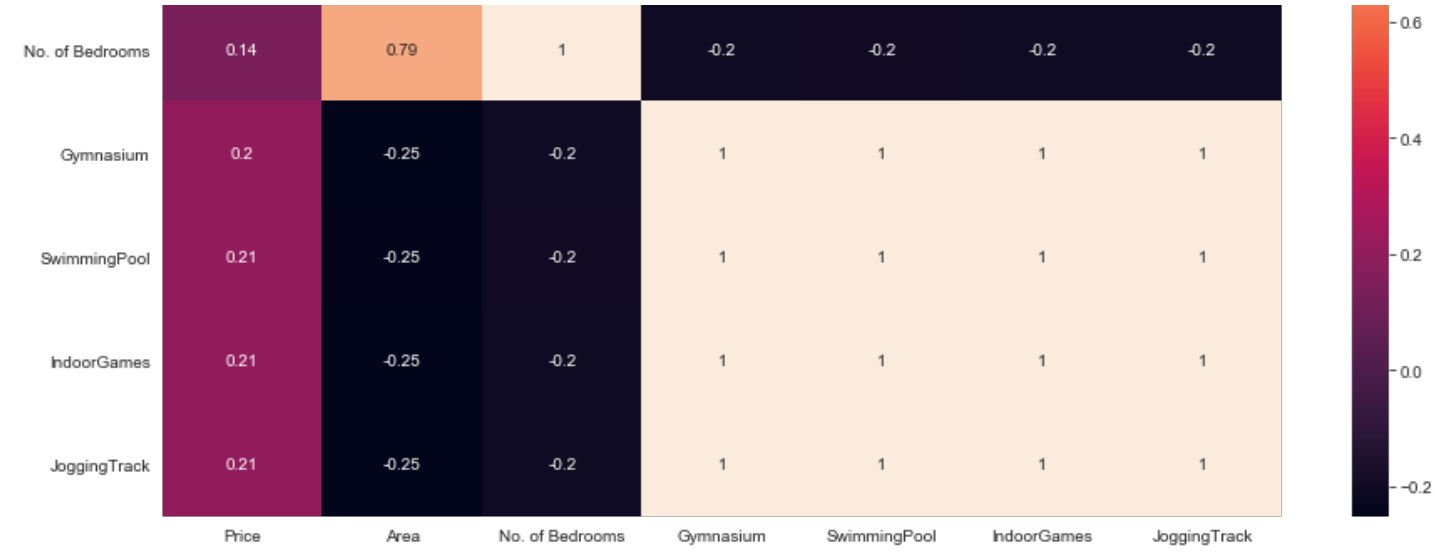
In [228]:

```
frames = [Bangalore_mod,Mumbai_mod]
merged = pd.concat(frames)
plt.figure(figsize=(15,8))
sns.heatmap(merged.corr(),annot=True)
```

Out[228]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2ad003b2b48>
```




No. of Bedrooms
Gymnasium
SwimmingPool
IndoorGames
JoggingTrack
0.14
0.79
1
-0.2
-0.2
-0.2
-0.2
0.2
-0.25
-0.2
1
1
1
1
0.21
-0.25
-0.2
1
1
1
1
0.21
-0.25
-0.2
1
1
1
1
0.21
-0.25
-0.2
1
1
1
1
Price
Area
No. of Bedrooms
Gymnasium
SwimmingPool
IndoorGames
JoggingTrack
0.6
0.4
0.2
0.0
−0.2

In [ ]: