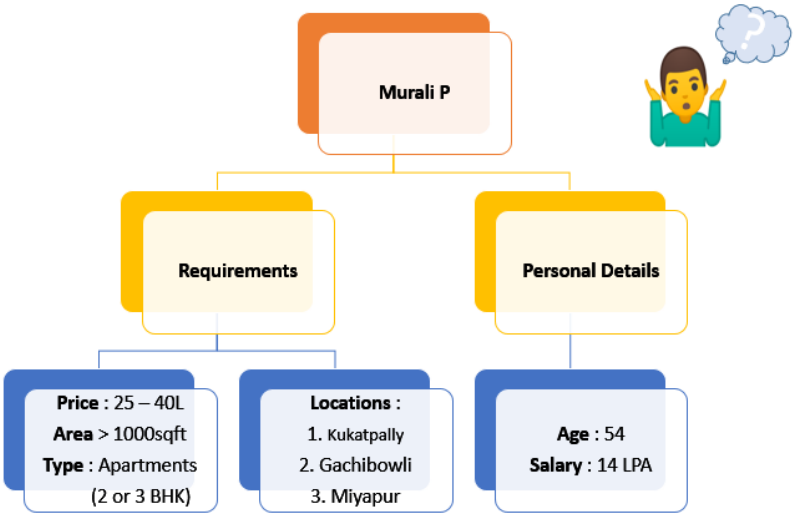
Bhargava Sai Reddy P · Follow
Oct 22, 2019 · 8 min read · Listen

Exploratory Data Analysis(EDA) on Residential Properties in Hyderabad

This article is a follow up of the previous one where I built a web scraper that extracted required data for a real estate website for analysis of residential properties in Hyderabad. In this, our primary focus would be on visualizing the data we have extracted and get meaningful insights of data so that we can come to the necessary conclusions.

Our ultimate goal is to find the residential properties with the requirements of my father which are shown stated below which states that there are four requirements i.e, price to be between 25 to 40 lakhs, the area should be greater than 1000 sqft, the property should be a 2 or 3 bhk apartment which is located in the areas kukatpally, gachibowli and miyapur.



We start with basic exploration and visualization of the data we have. For performing EDA we have univariate analysis, bivariate and multivariate analysis.

Firstly, we import the necessary libraries and see the basic details about the data like the number of rows, columns and the data types of each column. (rows represent the data points and column represents the features/attributes of data).

```
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

```
#Load the dataset into a dataframe(file containing data)

df=pd.read_excel('current.xlsx',index_col=0)
```

```
df.shape #returns the no. of rows and columns in our dataset
```

```
(25715, 7)
```

Let's take a look at the top and bottom rows of the dataset. To get the data types of each column we can use df.dtypes which returns the datatypes of each column. (df is the data frame we loaded)

```
df.head() #returns top 5 rows from our dataset
```

	title	location	price(L)	rate_persqft	area_insqft	building_status	agent_rating
0	Residential Plot	Maheshwaram	45.00	1111	4050	New	4.6
1	Residential Plot	Kondakal	34.00	2361	1440	New	3.7
2	Residential Plot	Bibinagar	14.19	944	1503	New	4.9
3	Residential Plot	Gachibowli	330.00	3333	9900	New	0.0
4	Residential Plot	Shadnagar	6.72	466	1440	New	4.7

```
df.tail(4) # Returns the last 4 rows from dataset
```

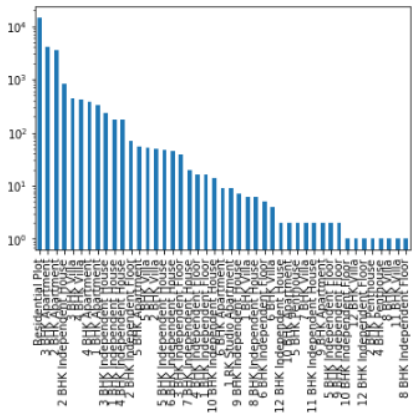
	title	location	price(L)	rate_persqft	area_insqft	building_status	agent_rating
25711	2 BHK Independent House	Kundanpally	25.74	3961	650	Ready to move	0.0
25712	3 BHK Apartment	Peeramcheru	81.00	4500	1800	Ready to move	0.0
25713	2 BHK Apartment	Nizampet	34.50	3833	900	Under Construction	0.0
25714	2 BHK Apartment	Chandanagar	40.00	4444	900	Under Construction	0.0

Now let's visualize every column we have,

```
df1=df['title'].value_counts() # returns the count of each unique data in the column.
df1.head()
```

```
Residential Plot           14642
3 BHK Apartment             4071
2 BHK Apartment             3532
2 BHK Independent House      832
3 BHK Villa                  440
Name: title, dtype: int64
```

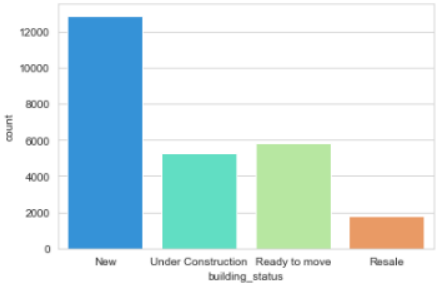
```
df1.plot.bar()
plt.yscale('log')
```



This bar plot above shows residential plots are the type of properties that are mostly found in Hyderabad which are about 14642 followed by 3 bhk apartments to be 4071 and similarly the count of other residential properties.

```
sns.set_style('whitegrid')
sns.countplot(x='building_status',data=df,palette='rainbow')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x22ffa86bef0>
```



The above plot depicts that there 4 categories in building_status which are New properties to be almost 12000, Under construction properties to be almost 5000, Ready to move properties to be 5800 and Resale type of properties to be 1900. To get the exact count of each category we can use value_counts().

```
d=df[(df['price(L)']>=25.00)&(df['price(L)']<=40.00)]
d.shape
```

```
(4789, 7)
```

when we consider the price range 25 to 40 L number of rows are reduced from 25715 to 4789 that means there are only 4789 properties in this price range.

Required locations: Kukatpally, Gachibowli, Miyapur

```
d1=d[(d['location']=='Gachibowli')|(d['location']=='Miyapur')|(d['location']=='Kukatpally')]
d1.shape
```

```
(85, 7)
```

When we consider the mentioned locations we are left with only 85 properties.

Requires only Apartment's

```
# Setting the title to index to extract required data.

d1.set_index(keys='title',inplace=True)
```

```
# Finding the properties that are Apartments
d3=d1.filter(like='Apartment',axis=0)
d3.head(5)
```

title	location	price(L)	rate_persqft	area_insqft	building_status	agent_rating
2 BHK Apartment	Miyapur	26.0	2600	1000	Ready to move	5.0
2 BHK Apartment	Miyapur	29.0	2504	1158	Under Construction	5.0
3 BHK Apartment	Miyapur	40.0	2439	1640	Under Construction	0.0
2 BHK Apartment	Miyapur	29.0	2406	1205	Under Construction	0.0
2 BHK Apartment	Miyapur	29.0	2412	1202	Under Construction	0.0

```
d3.reset_index(inplace=True) # Resetting the index
d3.shape
```

```
(69, 7)
```

To find the properties that are apartments we need to set the index as the title so that we can fetch the required data and then reset the index. We have only 69 properties that are apartments.

Do you want to know what is the average price of the property? If so we have a savior describe() function which returns the mean, median, std, min value, max value, 25th 50th & 75th percentiles of data etc.

```
d3.describe()
```

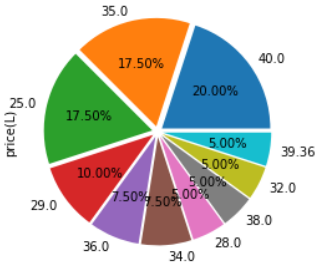
	price(L)	rate_persqft	area_insqft	agent_rating
count	69.000000	69.000000	69.000000	69.000000
mean	33.875217	3415.101449	1030.811594	2.621739
std	4.727725	809.044283	218.454617	2.219027
min	25.000000	2055.000000	558.000000	0.000000
25%	30.120000	3013.000000	925.000000	0.000000
50%	35.000000	3500.000000	1013.000000	4.100000
75%	37.000000	3733.000000	1140.000000	4.600000
max	40.000000	6333.000000	1640.000000	5.000000

Here count represents the number of rows and mean represents the average value of each column,25% says that 25% of data has a value less than that mentioned value there for example for column price 25% says that 25% of properties have their price less than 30.12 L.The minimum price of property is 25 L and maximum price is 40 L. And 75% of properties have their prices less than 37 L.

What say? Let's start visualizing??




```
<matplotlib.axes._subplots.AxesSubplot at 0x295dda0e4a8>
```



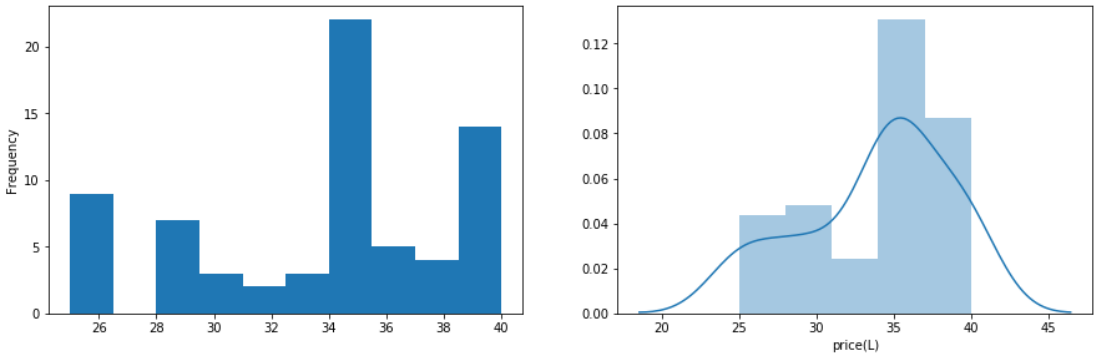
This a pie plot for the price column it shows that 20% of properties have their price as 40 L and 17.50% properties to have 25 L. Each pie in the plot represents different percentages of prices.

Let's see the distribution of prices.

```
f,ax=plt.subplots(figsize=(15,10)) # Plot size
ax2=plt.subplot(223)
d3['price(L)'].plot.hist() # Histogram for all prices.
ax2.plot

ax3=plt.subplot(224)
sns.distplot(d3['price(L)'],ax=ax3) # Distribution plot for all prices.
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x295dd41e2e8>
```



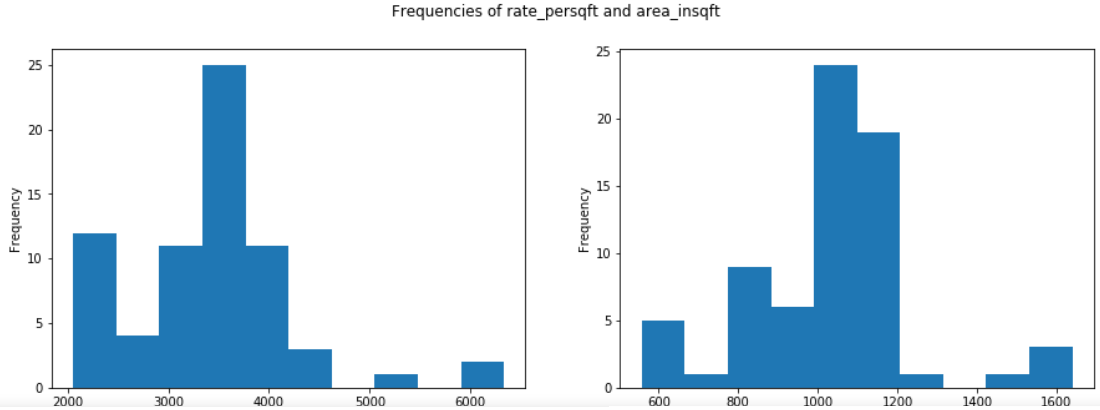
The left plot is a histogram in which x-axis represents the price and the y-axis represents the count/frequency of that price. The plot that is to the right is a distribution plot where the x-axis represents the price and the y-axis represents the probabilities of prices. For example, almost 23 properties are having a price range of 34 to 35 L and almost a probability of 0.12 for having a price range of 34 to 35 L.

In the same way, we can plot for other columns as well as shown below.

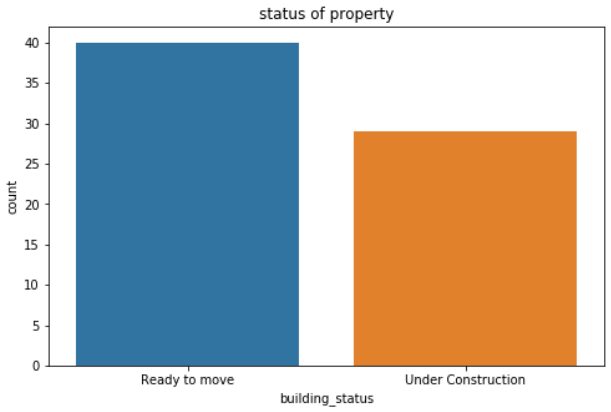
```
f,ax=plt.subplots(1,2,figsize=(15,5))
f.suptitle("Frequencies of rate_persqft and area_insqft") # Plot title
d3['rate_persqft'].plot('hist',ax=ax[0]) # Histogram for rate_persqft.

d3['area_insqft'].plot('hist',ax=ax[1]) # Histogram for area_insqft.
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x295ddce4080>
```



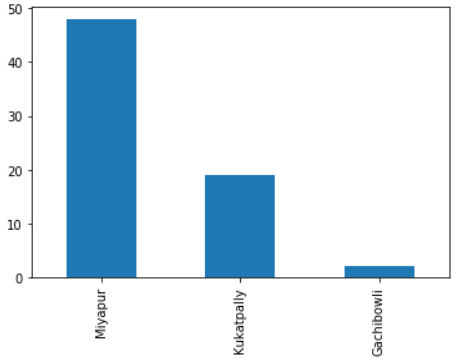
```
Text(0.5, 1.0, 'status of property')
```



Plot for building_status

```
d5=d3['location'].value_counts()
d5.plot.bar()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x295dd432b00>
```



Plot for locations

Oh! I guess we have forgotten to include two more requirements that are area and type of apartments. Well, no problem lets include them now and start further analysis and visualizations.

Client's required area in sqft

```
d4=d3[(d3['area_insqft']>=1000)]
d4.shape
```

```
(48, 7)
```

Client's required properties

```
d5=d4[(d4['title']=='2 BHK Apartment')|(d4['title']=='3 BHK Apartment')]
d5.shape
```

```
(46, 7)
```

Count of the properties

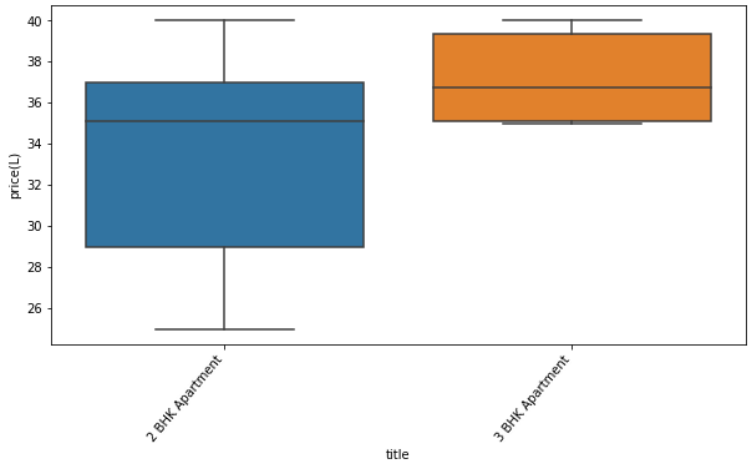
```
d5['title'].value_counts()
```

```
2 BHK Apartment    41
3 BHK Apartment     5
Name: title, dtype: int64
```

So after further adding our conditions we are only left with 46 properties.

Bivariate Analysis

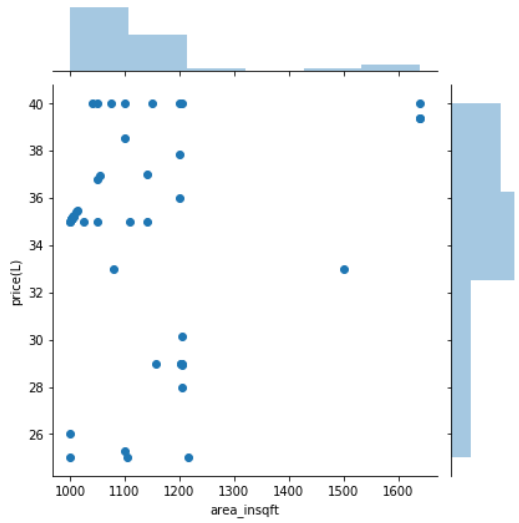
```
plt.show()
```



Plot for Title vs Price(L)

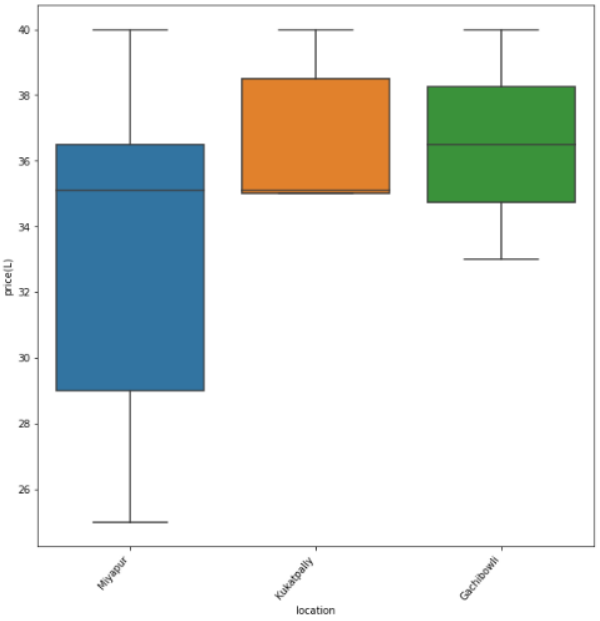
The above figure is a box blot in which the two lines on both ends are called whiskers, the bottom line is the minimum value where the topmost line is the maximum value for each category i.e, 2 bhk and 3 bhk and the middle line inside the colored area specify the median which is also the 50th percentile. The plot describes that for 2 bhk the minimum price is 25 L and maximum price is 40 L, 25% of 2 bhk's has their prices below 29 L whereas in 3 bhk the minimum starts at around 35 L and goes up to 40 L where 25% of 3 bhk's has their price below 37 L.

```
sns.jointplot(x="area_insqft", y="price(L)", data=d5)
<seaborn.axisgrid.JointGrid at 0x295ddc35fd0>
```



A Joint plot between area_insqft and price(L)

From the above plot we can infer the relation between area and price by seeing the histograms on top of each axis of the plot. Surprisingly there is no increase in price with an increase in the area of the property(these assumptions are only for the data that we have got after applying the conditions that are specified)

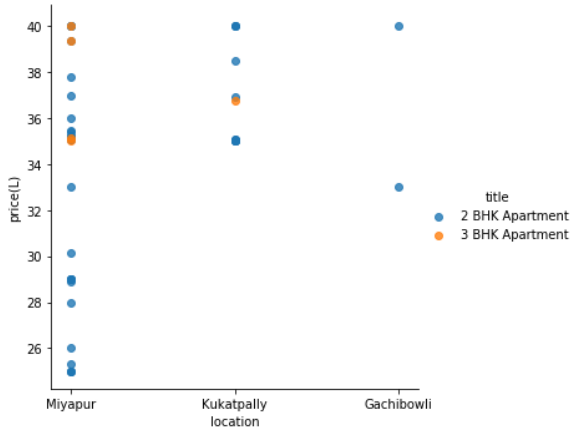


Boxplot between location and price(L)

Multivariate Analysis

```
sns.lmplot(x='location', y='price(L)', hue='title',
           data=d5,fit_reg=False)
```

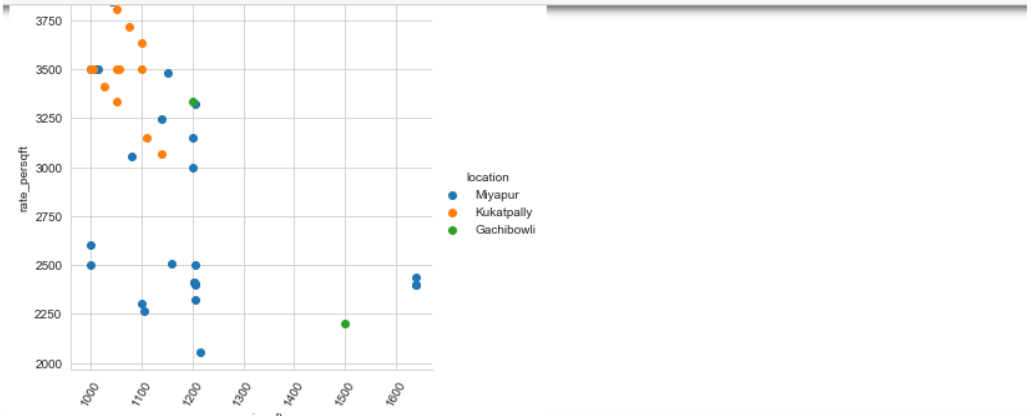
```
<seaborn.axisgrid.FacetGrid at 0x295dd7f4b00>
```



We can see from the above plot that miyapur has a wide range of prices from 25 L to 40 L in which most of them are 2 bhks. Whereas kukatpally has very few apartments whose price range starts from 35 L to 40 L. In gachibowli there are only two 2 bhk's whose prices are 33 L and 40 L.

This is a violin plot which describes that miyapur has both types of apartments that are ready to move and under construction as well where most of the ready to move apartments have their price between 35 L and 36 L. And gachibowli has only under construction apartments.

```
sns.set_style("whitegrid");
ax=sns.FacetGrid(d5,hue='location',height=5).map(plt.scatter,'area_insqft','rate_persqft').add_legend()
ax.set_xticklabels(rotation=60)
ax.set_xlabels()
plt.show()
```



This a scatter plot where the x-axis is area_insqft and the y-axis is rate_persqft. We can see that most of the apartments in miyapur has a rate per sqft between 2000 and 2500, whereas in kukatpally rate per sqft is mostly between 3000 and 3750.

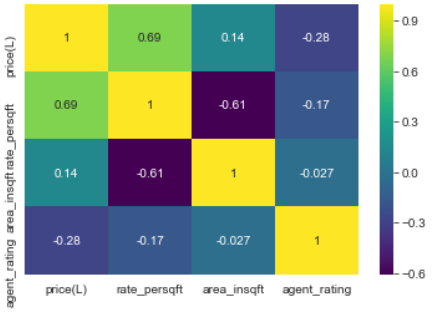
Correlation: It defines the relationship between two features whether they are positively correlated or negatively correlated. Result of a correlation is called the correlation coefficient (or “r”).It ranges from -1.0 to +1.0. The closer r is to +1 or -1, the more closely the two variables are related. If r is close to 0, it means there is no relationship between the variables. If r is positive, it means that as one variable gets larger the other gets larger. If r is negative it means that as one gets larger, the other gets smaller (often called an “inverse” correlation).

```
d5.corr()
```

	price(L)	rate_persqft	area_insqft	agent_rating
price(L)	1.000000	0.691705	0.144277	-0.278001
rate_persqft	0.691705	1.000000	-0.609346	-0.174956
area_insqft	0.144277	-0.609346	1.000000	-0.027115
agent_rating	-0.278001	-0.174956	-0.027115	1.000000

This is called a correlation matrix. We can even visualize it using seaborn.

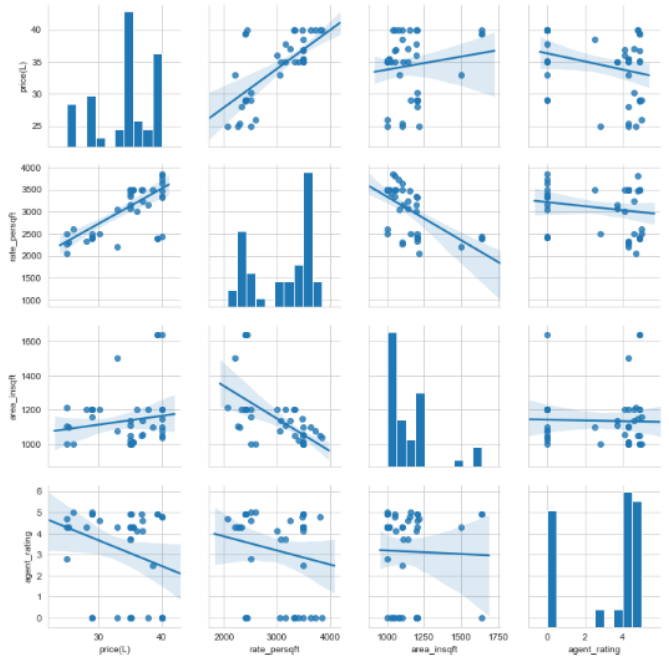
```
<matplotlib.axes._subplots.AxesSubplot at 0x295e1182cf8>
```



We can also use pair plots for correlation.

```
sns.pairplot(d5,kind='reg')
```

```
<seaborn.axisgrid.PairGrid at 0x295dd46a710>
```



For the price range considered the highest correlation is between rate_persqft and price, they both have a positive correlation. Negative correlation exists between area_insqft and rate_persqft. There's not much correlation between price and area_insqft.

Map Visualization

We can also use maps for visualization using a folium package.

```
import folium
from  geopy.geocoders import Nominatim
import warnings
warnings.filterwarnings("ignore")
from folium.plugins import HeatMap
```

importing necessary libraries

To get a map visualization we need latitude and longitude of the locations. Let us consider the three locations that were specified as one of the requirements(miyapur, kukatpally, gachibowli). And we will write a function that returns latitude and longitude of a location using the geopy module.

	title	location	price(L)	rate_persqft	area_insqft	building_status	agent_rating
0	2 BHK Apartment	Miyapur	26.00	2600	1000	Ready to move	5.0
5	2 BHK Apartment	Kukatpally	35.09	3499	1003	Ready to move	0.0
13	2 BHK Apartment	Gachibowli	33.00	2200	1500	Under Construction	4.3

```
lst=df['location'].tolist()
lst
```

```
['Miyapur', 'Kukatpally', 'Gachibowli']
```

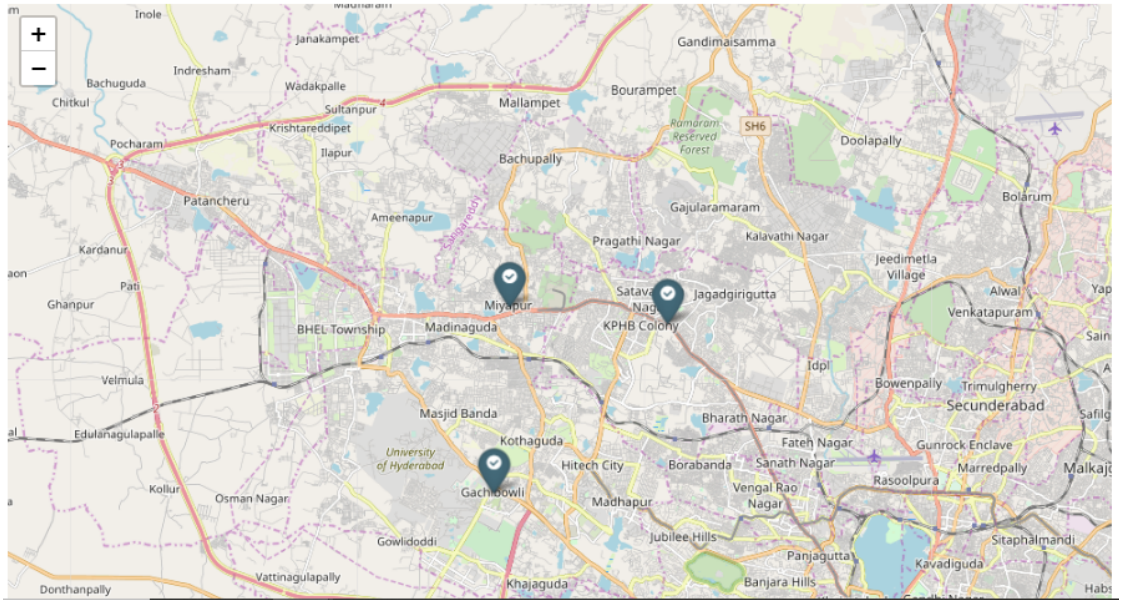
```
l_l1=[]
for i in lst:
    geolocator = Nominatim()
    Area =i
    City ="Hyderabad"
    loc = geolocator.geocode(Area+','+City)
    a=loc.latitude,loc.longitude
    l_l1.append(a)
```

```
l_l1
```

```
[(17.4981608, 78.3567628), (17.4930841, 78.4054408), (17.4436222, 78.3519638)]
```

From above we can see that we were able to extract latitude and longitude of a location. Now let's try plotting them.

```
map_hooray = folium.Map(location=[17.3850, 78.4867],zoom_start = 12)
for i in l_l1:
    folium.Marker(i,icon=folium.Icon(color='cadetblue',icon='check-circle', prefix='fa')).add_to(map_hooray)
map_hooray
```



Let's try plotting for all the locations we have in the dataset. we can extract all the latitudes and longitudes of locations in the same way as we have done above. First, extract all the lat n long of locations and then create two new columns in our original dataset latitude and longitude and store extracted values there and then do as follows,

```
by = d[['latitude', 'longitude']].groupby(['latitude', 'longitude']).sum().reset_index().values.tolist()
```

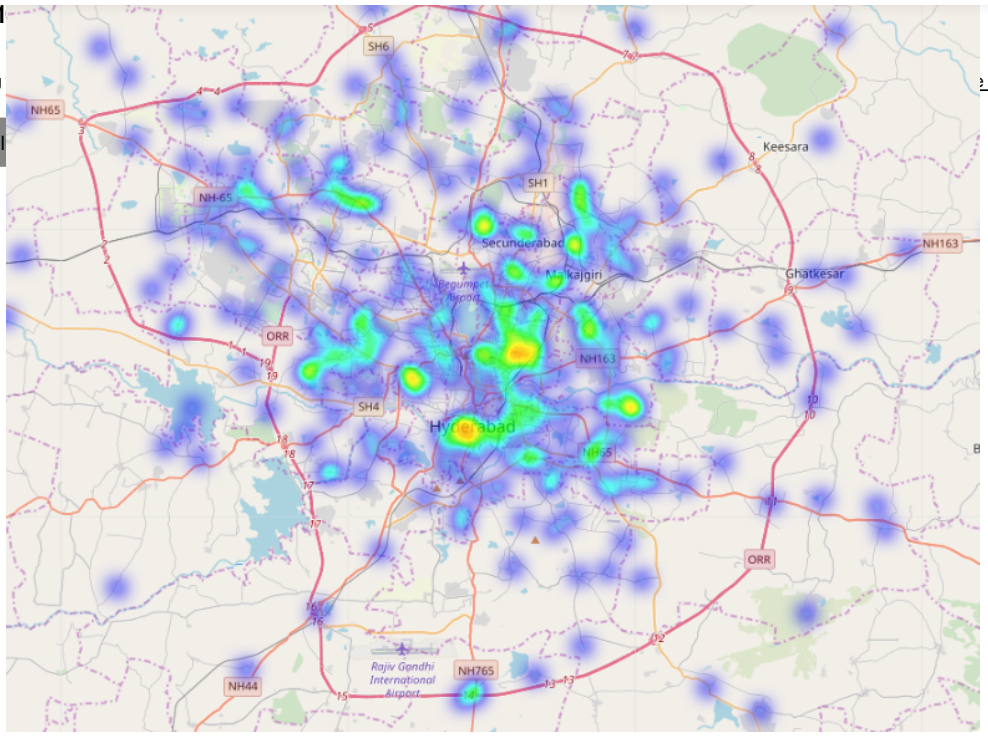
```
def generateBaseMap(default_location=[17.38878595,78.4610647345315], default_zoom_start=11):
    base_map = folium.Map(location=default_location, zoom_start=default_zoom_start)
    return base_map
```

```
def drawmap(data):

    base_map = generateBaseMap()
    x = HeatMap(data, radius=8, max_zoom=13).add_to(base_map)
    return x
```

```
ht = drawmap(by)
ht
```

```
<folium.plugins.heat_map.HeatMap at 0x1cc359fc2b0>
```



The above map shows us the intensity of properties in different locations of Hyderabad.

In conclusion, there is a total of 14642 residential plots, 8381 Apartments, 4071 3bhk apartments, 832 2bhk independent houses,440 3bhk villa etc. And in consideration to conditions give we finally have 46 properties that suit the requirements mentioned. Almost 20% of properties have a price of 40 L in price range 25 to 40 L and miyapur has more number of residential properties.

Bhargava Sai Reddy Putta - AMITY University Gurgaon - Hyderabad, Telangana, India | LinkedIn
View Bhargava Sai Reddy Putta's profile on LinkedIn, the world's largest professional community. Bhargava Sai Reddy's...
www.linkedin.com

Thanks for reading.