In [121…

```
import pandas as pd
import numpy as np

import matplotlib as plt
import matplotlib.pyplot as plt
import seaborn as sns

import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor
%matplotlib inline
```

In [122…

```
data = pd.read_csv('Bengaluru_House_Data.csv')
```

In [123…

```
type(data)
```

Out[123…

```
pandas.core.frame.DataFrame
```

In [124…

```
data.to_string
```

Out[124…

```
<bound method DataFrame.to_string of                   area_type   availability
location  \
0      Super built-up  Area         19-Dec  Electronic City Phase II
1                Plot  Area  Ready To Move          Chikka Tirupathi
2            Built-up  Area  Ready To Move               Uttarahalli
3      Super built-up  Area  Ready To Move        Lingadheeranahalli
4      Super built-up  Area  Ready To Move                  Kothanur
...                      ...            ...                       ...
13315        Built-up  Area  Ready To Move                Whitefield
13316  Super built-up  Area  Ready To Move              Richards Town
13317        Built-up  Area  Ready To Move     Raja Rajeshwari Nagar
13318  Super built-up  Area         18-Jun           Padmanabhanagar
13319  Super built-up  Area  Ready To Move              Doddathoguru

            size  society total_sqft  bath  balcony   price
0          2 BHK  Coomee        1056   2.0      1.0   39.07
1      4 Bedroom  Theanmp       2600   5.0      3.0  120.00
2          3 BHK      NaN       1440   2.0      3.0   62.00
3          3 BHK  Soiewre       1521   3.0      1.0   95.00
4          2 BHK      NaN       1200   2.0      1.0   51.00
...          ...      ...        ...   ...      ...     ...
13315  5 Bedroom  ArsiaEx       3453   4.0      0.0  231.00
13316      4 BHK      NaN       3600   5.0      NaN  400.00
13317      2 BHK  Mahla T       1141   2.0      1.0   60.00
13318      4 BHK  SollyCl       4689   4.0      1.0  488.00
13319      1 BHK      NaN        550   1.0      1.0   17.00

[13320 rows x 9 columns]>
```

In [125…

```
type(data)
```

Out[125…

```
pandas.core.frame.DataFrame
```

In [126…

```
print(data.shape)
```

```
(13320, 9)
```

In [127…

```
print(data.describe(include='all'))
```

```
                  area_type   availability    location   size  society  \
count                 13320          13320       13319  13304     7818
unique                    4             81        1305     31     2688
top     Super built-up  Area  Ready To Move  Whitefield  2 BHK  GrrvaGr
freq                   8790          10581         540   5199       80
mean                    NaN            NaN         NaN    NaN      NaN
std                     NaN            NaN         NaN    NaN      NaN
min                     NaN            NaN         NaN    NaN      NaN
25%                     NaN            NaN         NaN    NaN      NaN
50%                     NaN            NaN         NaN    NaN      NaN
75%                     NaN            NaN         NaN    NaN      NaN
max                     NaN            NaN         NaN    NaN      NaN

       total_sqft          bath       balcony         price
count       13320  13247.000000  12711.000000  13320.000000
unique       2117           NaN           NaN           NaN
top          1200           NaN           NaN           NaN
freq          843           NaN           NaN           NaN
mean          NaN      2.692610      1.584376    112.565627
std           NaN      1.341458      0.817263    148.971674
min           NaN      1.000000      0.000000      8.000000
25%           NaN      2.000000      1.000000     50.000000
50%           NaN      2.000000      2.000000     72.000000
75%           NaN      3.000000      2.000000    120.000000
max           NaN     40.000000      3.000000   3600.000000
```

In [128…

```
data.columns
```

Out[128…

```
Index(['area_type', 'availability', 'location', 'size', 'society',
       'total_sqft', 'bath', 'balcony', 'price'],
      dtype='object')
```

In [129…

```
data.price
```

Out[129…

```
0         39.07
1        120.00
2         62.00
3         95.00
4         51.00
          ...
13315    231.00
13316    400.00
13317     60.00
13318    488.00
13319     17.00
Name: price, Length: 13320, dtype: float64
```

Data Exploration

In [130…

```
data.head(10) #head of the data
```

Out[130…

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440	2.0	3.0	62.00
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	NaN	1200	2.0	1.0	51.00
5	Super built-up Area	Ready To Move	Whitefield	2 BHK	DuenaTa	1170	2.0	1.0	38.00
6	Super built-up Area	18-May	Old Airport Road	4 BHK	Jaades	2732	4.0	NaN	204.00
7	Super built-up Area	Ready To Move	Rajaji Nagar	4 BHK	Brway G	3300	4.0	NaN	600.00
8	Super built-up Area	Ready To Move	Marathahalli	3 BHK	NaN	1310	3.0	1.0	63.25
9	Plot Area	Ready To Move	Gandhi Bazar	6 Bedroom	NaN	1020	6.0	NaN	370.00

In [131…

```
data.tail() #bottom of the data
```

Out[131…

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
13315	Built-up Area	Ready To Move	Whitefield	5 Bedroom	ArsiaEx	3453	4.0	0.0	231.0
13316	Super built-up Area	Ready To Move	Richards Town	4 BHK	NaN	3600	5.0	NaN	400.0
13317	Built-up Area	Ready To Move	Raja Rajeshwari Nagar	2 BHK	Mahla T	1141	2.0	1.0	60.0
13318	Super built-up Area	18-Jun	Padmanabhanagar	4 BHK	SollyCl	4689	4.0	1.0	488.0
13319	Super built-up Area	Ready To Move	Doddathoguru	1 BHK	NaN	550	1.0	1.0	17.0

In [132…

```
data.count
```

Out[132…

```
<bound method DataFrame.count of                   area_type   availability
location  \
0      Super built-up  Area         19-Dec  Electronic City Phase II
1                Plot  Area  Ready To Move          Chikka Tirupathi
2            Built-up  Area  Ready To Move               Uttarahalli
```

```
3      Super built-up  Area  Ready To Move        Lingadheeranahalli
4      Super built-up  Area  Ready To Move                  Kothanur
...                      ...           ...                       ...
13315        Built-up  Area  Ready To Move               Whitefield
13316  Super built-up  Area  Ready To Move             Richards Town
13317        Built-up  Area  Ready To Move     Raja Rajeshwari Nagar
13318  Super built-up  Area         18-Jun           Padmanabhanagar
13319  Super built-up  Area  Ready To Move              Doddathoguru

            size  society total_sqft  bath  balcony   price
0          2 BHK  Coomee        1056   2.0      1.0   39.07
1      4 Bedroom  Theanmp       2600   5.0      3.0  120.00
2          3 BHK      NaN       1440   2.0      3.0   62.00
3          3 BHK  Soiewre       1521   3.0      1.0   95.00
4          2 BHK      NaN       1200   2.0      1.0   51.00
...          ...      ...        ...   ...      ...     ...
13315  5 Bedroom  ArsiaEx       3453   4.0      0.0  231.00
13316      4 BHK      NaN       3600   5.0      NaN  400.00
13317      2 BHK  Mahla T       1141   2.0      1.0   60.00
13318      4 BHK  SollyCl       4689   4.0      1.0  488.00
13319      1 BHK      NaN        550   1.0      1.0   17.00

[13320 rows x 9 columns]>
```

Finding the missing values

In [133…

```
pd.isnull(data).head(10) #false meaning no missing values in the dataset #true means
```

Out[133…

	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	True	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	True	False	False	False	False
5	False	False	False	False	False	False	False	False	False
6	False	False	False	False	False	False	False	True	False
7	False	False	False	False	False	False	False	True	False
8	False	False	False	False	True	False	False	False	False
9	False	False	False	False	True	False	False	True	False

In [134…

```
pd.isnull(data).any()
```

Out[134…

```
area_type       False
availability    False
location         True
size             True
society          True
total_sqft      False
bath             True
balcony          True
price           False
dtype: bool
```

In [135…

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13320 entries, 0 to 13319
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   area_type     13320 non-null  object 
 1   availability  13320 non-null  object 
 2   location      13319 non-null  object 
 3   size          13304 non-null  object 
 4   society       7818 non-null   object 
 5   total_sqft    13320 non-null  object 
 6   bath          13247 non-null  float64
 7   balcony       12711 non-null  float64
 8   price         13320 non-null  float64
dtypes: float64(3), object(6)
memory usage: 936.7+ KB
```

In [136…

```
data.dropna(subset=['price'], inplace=True)
# Clean null values in the 'price' column
data
```

Out[136…

	area_type	availability	location	size	society	total_sqft	bath	balcony	pric
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.(
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.(
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440	2.0	3.0	62.(
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.(
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	NaN	1200	2.0	1.0	51.(
...	...	...	...	...	...	...	...	...	
13315	Built-up Area	Ready To Move	Whitefield	5 Bedroom	ArsiaEx	3453	4.0	0.0	231.(
13316	Super built-up Area	Ready To Move	Richards Town	4 BHK	NaN	3600	5.0	NaN	400.(
13317	Built-up Area	Ready To Move	Raja Rajeshwari Nagar	2 BHK	Mahla T	1141	2.0	1.0	60.(
13318	Super built-up Area	18-Jun	Padmanabhanagar	4 BHK	SollyCl	4689	4.0	1.0	488.(
13319	Super built-up Area	Ready To Move	Doddathoguru	1 BHK	NaN	550	1.0	1.0	17.(

13320 rows × 9 columns

Data visualising

In [137…

```
pip install --upgrade matplotlib
```

```
Requirement already satisfied: matplotlib in c:\users\sajid\anaconda3\lib\site-packa
ges (3.7.1)
Requirement already satisfied: numpy>=1.20 in c:\users\sajid\anaconda3\lib\site-pack
ages (from matplotlib) (1.22.4)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\sajid\anaconda3\lib\sit
e-packages (from matplotlib) (4.25.0)
Requirement already satisfied: importlib-resources>=3.2.0 in c:\users\sajid\anaconda
3\lib\site-packages (from matplotlib) (5.12.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\sajid\anaconda3\lib\sit
e-packages (from matplotlib) (1.3.1)
Requirement already satisfied: packaging>=20.0 in c:\users\sajid\anaconda3\lib\site-
packages (from matplotlib) (21.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\sajid\anaconda3\lib\site
-packages (from matplotlib) (3.0.4)
Requirement already satisfied: cycler>=0.10 in c:\users\sajid\anaconda3\lib\site-pac
kages (from matplotlib) (0.10.0)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\sajid\anaconda3\lib\site
-packages (from matplotlib) (1.0.7)
Requirement already satisfied: pillow>=6.2.0 in c:\users\sajid\anaconda3\lib\site-pa
ckages (from matplotlib) (8.4.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\sajid\anaconda3\lib
\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: six in c:\users\sajid\anaconda3\lib\site-packages (fr
om cycler>=0.10->matplotlib) (1.16.0)
Requirement already satisfied: zipp>=3.1.0 in c:\users\sajid\anaconda3\lib\site-pack
ages (from importlib-resources>=3.2.0->matplotlib) (3.6.0)
Note: you may need to restart the kernel to use updated packages.
```

In [138…

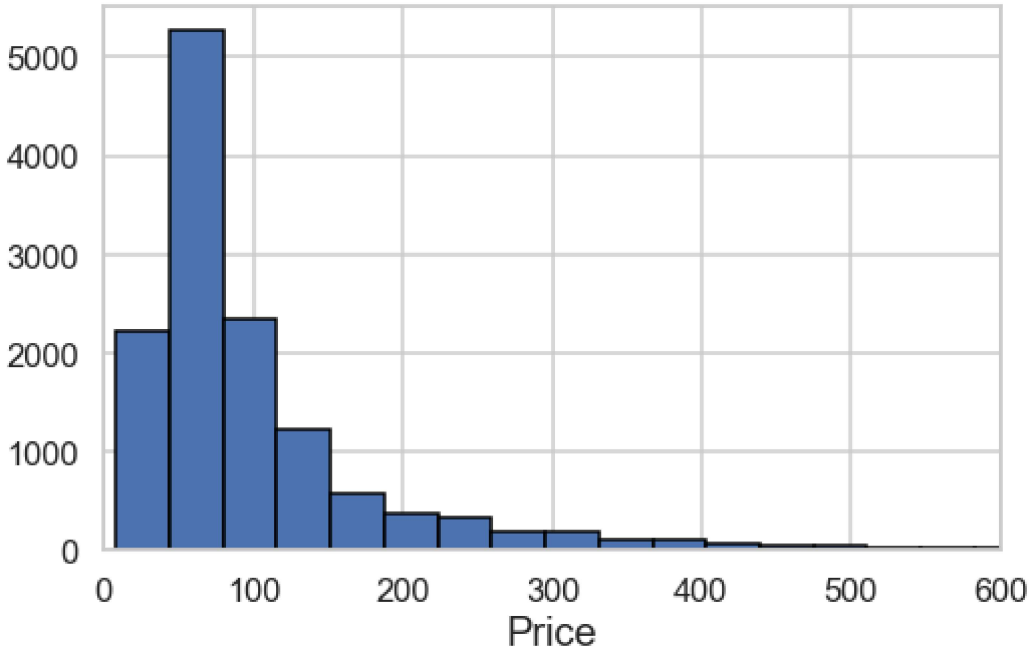
```
data = data[data['price'] != 0]

# Print the updated DataFrame
print(data)
```

```
                area_type   availability                  location  \
0      Super built-up  Area         19-Dec  Electronic City Phase II
1            Plot  Area  Ready To Move          Chikka Tirupathi
2         Built-up  Area  Ready To Move               Uttarahalli
3      Super built-up  Area  Ready To Move        Lingadheeranahalli
4      Super built-up  Area  Ready To Move                  Kothanur
...                  ...            ...                       ...
13315        Built-up  Area  Ready To Move                Whitefield
13316  Super built-up  Area  Ready To Move              Richards Town
13317        Built-up  Area  Ready To Move     Raja Rajeshwari Nagar
13318  Super built-up  Area         18-Jun           Padmanabhanagar
13319  Super built-up  Area  Ready To Move              Doddathoguru

            size  society total_sqft  bath  balcony   price
0          2 BHK  Coomee        1056   2.0      1.0   39.07
1      4 Bedroom  Theanmp       2600   5.0      3.0  120.00
2          3 BHK      NaN       1440   2.0      3.0   62.00
3          3 BHK  Soiewre       1521   3.0      1.0   95.00
4          2 BHK      NaN       1200   2.0      1.0   51.00
...          ...      ...        ...   ...      ...     ...
13315  5 Bedroom  ArsiaEx       3453   4.0      0.0  231.00
13316      4 BHK      NaN       3600   5.0      NaN  400.00
```

```
13317      2 BHK  Mahla T       1141   2.0      1.0   60.00
13318      4 BHK  SollyCl       4689   4.0      1.0  488.00
13319      1 BHK      NaN        550   1.0      1.0   17.00

[13320 rows x 9 columns]
```

In [139…

```
plt.figure(figsize=(8,5))
plt.hist(data['price'],bins=100,ec='black')
plt.xlim(0,600)

plt.xlabel('Price',size=20)
plt.show()
```



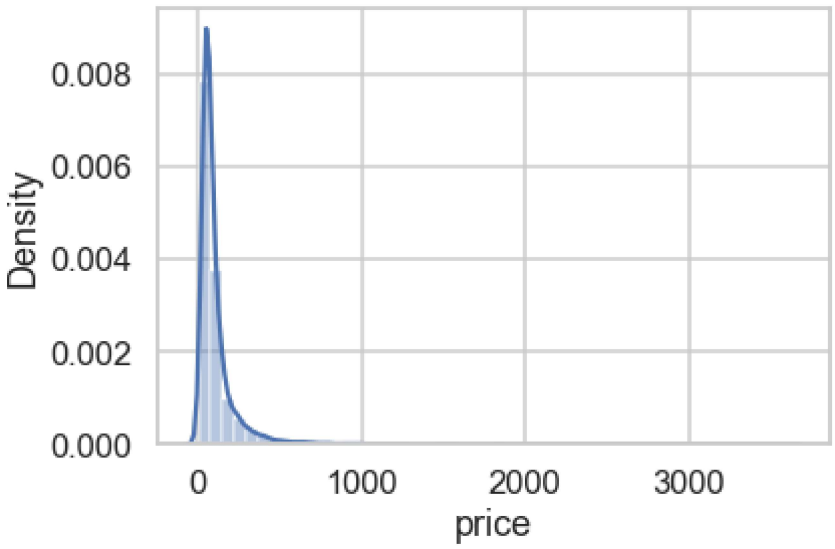
In [140…

```
sns.distplot(data['price'],bins=50)
```

```
C:\Users\sajid\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarn
ing: `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 similar f
lexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[140…

```
<Axes: xlabel='price', ylabel='Density'>
```



In [141…

```
data.head(10)
```

Out[141…

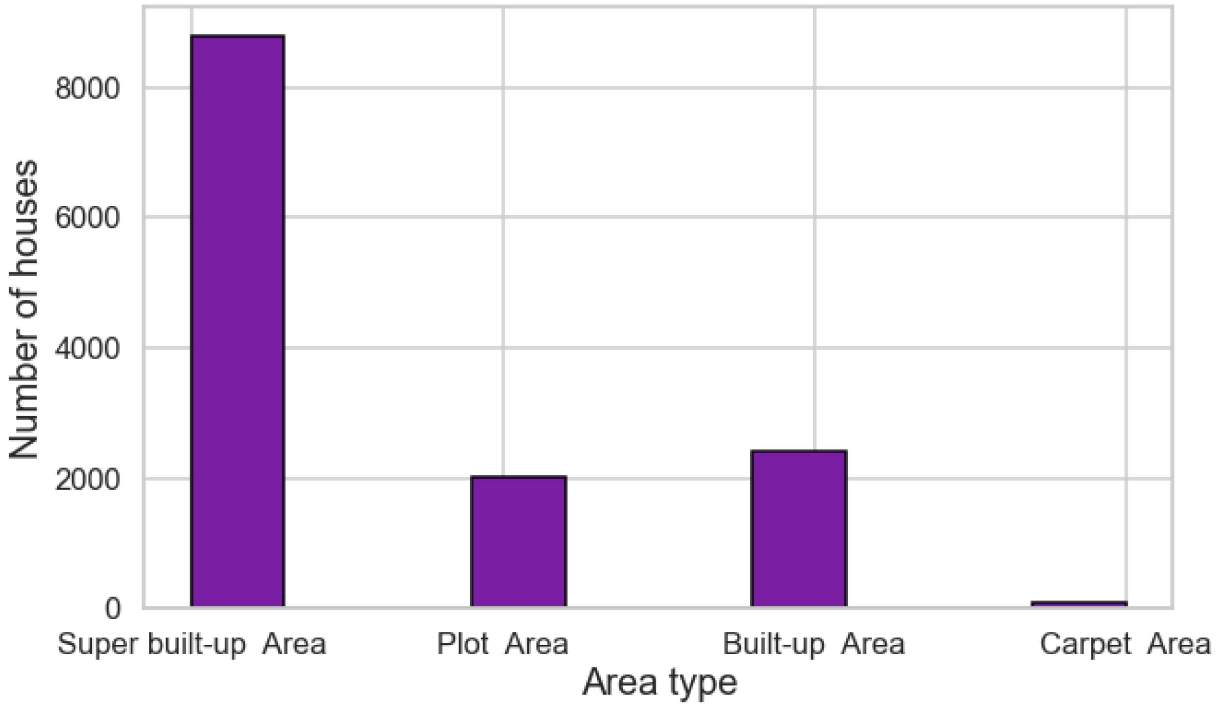
	area_type	availability	location	size	society	total_sqft	bath	balcony	price
0	Super built-up Area	19-Dec	Electronic City Phase II	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Plot Area	Ready To Move	Chikka Tirupathi	4 Bedroom	Theanmp	2600	5.0	3.0	120.00
2	Built-up Area	Ready To Move	Uttarahalli	3 BHK	NaN	1440	2.0	3.0	62.00
3	Super built-up Area	Ready To Move	Lingadheeranahalli	3 BHK	Soiewre	1521	3.0	1.0	95.00
4	Super built-up Area	Ready To Move	Kothanur	2 BHK	NaN	1200	2.0	1.0	51.00
5	Super built-up Area	Ready To Move	Whitefield	2 BHK	DuenaTa	1170	2.0	1.0	38.00
6	Super built-up Area	18-May	Old Airport Road	4 BHK	Jaades	2732	4.0	NaN	204.00
7	Super built-up Area	Ready To Move	Rajaji Nagar	4 BHK	Brway G	3300	4.0	NaN	600.00
8	Super built-up Area	Ready To Move	Marathahalli	3 BHK	NaN	1310	3.0	1.0	63.25
9	Plot Area	Ready To Move	Gandhi Bazar	6 Bedroom	NaN	1020	6.0	NaN	370.00

In [142…

```
plt.figure(figsize=(10,6))
plt.hist(data['area_type'],ec='black',color='#7b1fa2')
plt.xlabel('Area type',size=20)
plt.ylabel('Number of houses',size=20)
```

Out[142…

```
Text(0, 0.5, 'Number of houses')
```



In [143…

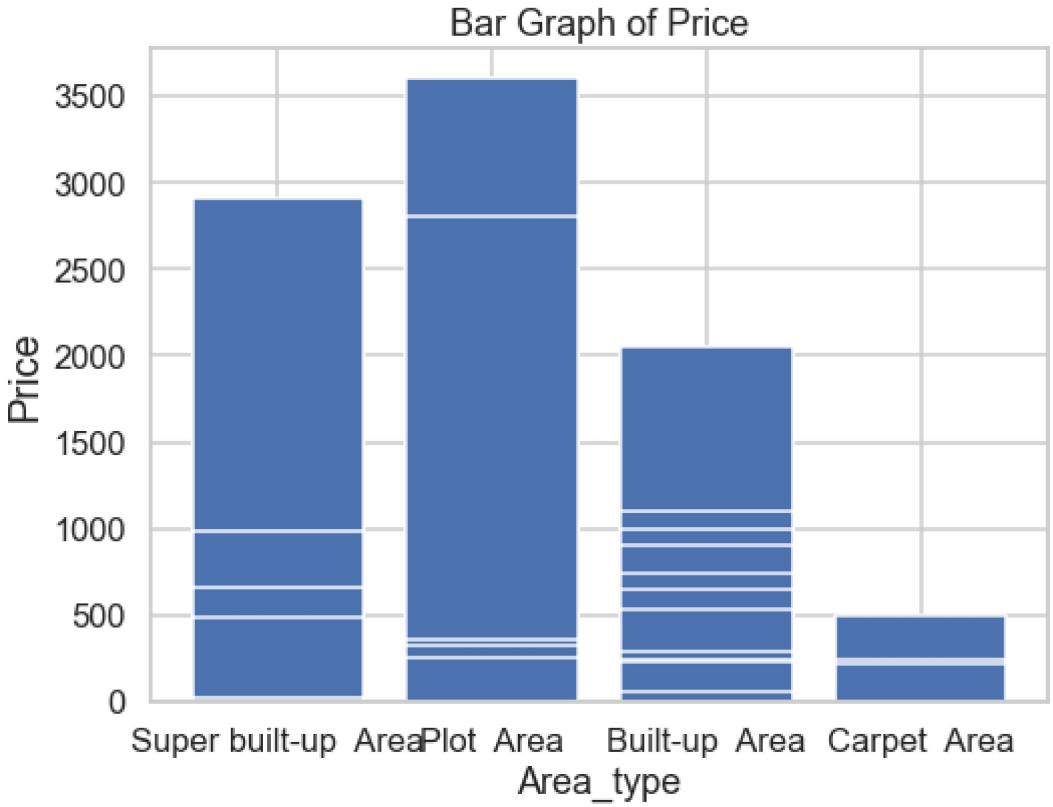
```
data['area_type'].value_counts()
```

Out[143…

```
Super built-up  Area    8790
Built-up  Area          2418
Plot  Area              2025
Carpet  Area              87
Name: area_type, dtype: int64
```

In [144…

```
plt.figure(figsize=(8,6))
plt.bar(data['area_type'],data['price'])
plt.xlabel('Area_type')
plt.ylabel('Price',size=20)
plt.title('Bar Graph of Price')

plt.show()
```



Descriptive statistics

In [145...
```
data['price'].min()
```

Out[145...
```
8.0
```

In [146...
```
data['price'].max()
```

Out[146...
```
3600.0
```

In [147...
```
data['price'].mean()
```

Out[147...
```
112.56562650150138
```

In [148...
```
data['price'].median()
```

Out[148...
```
72.0
```

In [149...
```
data.describe()
```

Out[149...

	bath	balcony	price
count	13247.000000	12711.000000	13320.000000
mean	2.692610	1.584376	112.565627
std	1.341458	0.817263	148.971674
min	1.000000	0.000000	8.000000

	bath	balcony	price
25%	2.000000	1.000000	50.000000
50%	2.000000	2.000000	72.000000
75%	3.000000	2.000000	120.000000
max	40.000000	3.000000	3600.000000

Correlation

$$\rho_{XY} = corr(X, Y)$$

$$-1.0 \leq \rho_{XY} \leq +1.0$$

In [150…

```
data['price'] = pd.to_numeric(data['price'], errors='coerce')
data['total_sqft'] = pd.to_numeric(data['total_sqft'], errors='coerce')

# Calculating the correlation
correlation_1 = data['price'].corr(data['total_sqft'])
print(correlation_1)
```

```
0.5729036676981671
```

In [151…

```
data['price'] = pd.to_numeric(data['price'], errors='coerce')
data['balcony'] = pd.to_numeric(data['balcony'], errors='coerce')

# Calculating the correlation
correlation_2 = data['price'].corr(data['balcony'])
print(correlation_1)
```

```
0.5729036676981671
```

In [152…

```
#we dont have any negative correlation in the dataset
#the correlation of itself is equal to one
data.corr()
```

Out[152…

	total_sqft	bath	balcony	price
total_sqft	1.000000	0.387206	0.155187	0.572904
bath	0.387206	1.000000	0.204201	0.456345
balcony	0.155187	0.204201	1.000000	0.120355
price	0.572904	0.456345	0.120355	1.000000

Plotting the heat map

In [153…

```
mask = np.zeros_like(data.corr())
triangle_indices = np.triu_indices_from(mask)
mask[triangle_indices] = True
mask
```

Out[153…

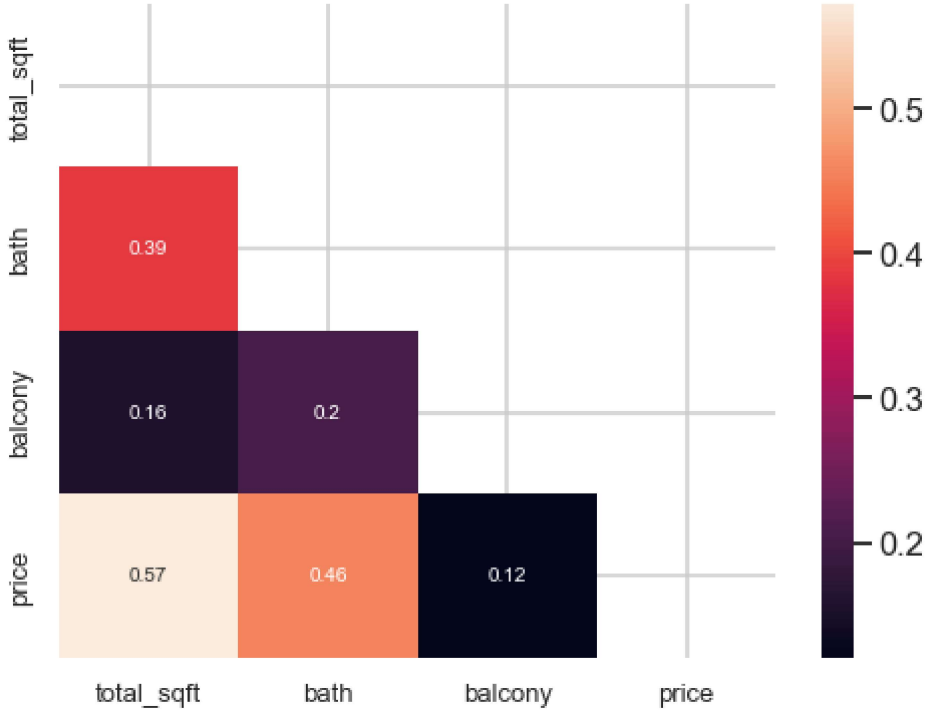
```
array([[1., 1., 1., 1.],
       [0., 1., 1., 1.],
       [0., 0., 1., 1.],
       [0., 0., 0., 1.]])
```

In [154…

```
plt.figure(figsize=(8,6))
sns.heatmap(data.corr(),annot=True,annot_kws={"size":14})
plt.yticks(fontsize=13)
plt.xticks(fontsize=13)
plt.show()
```

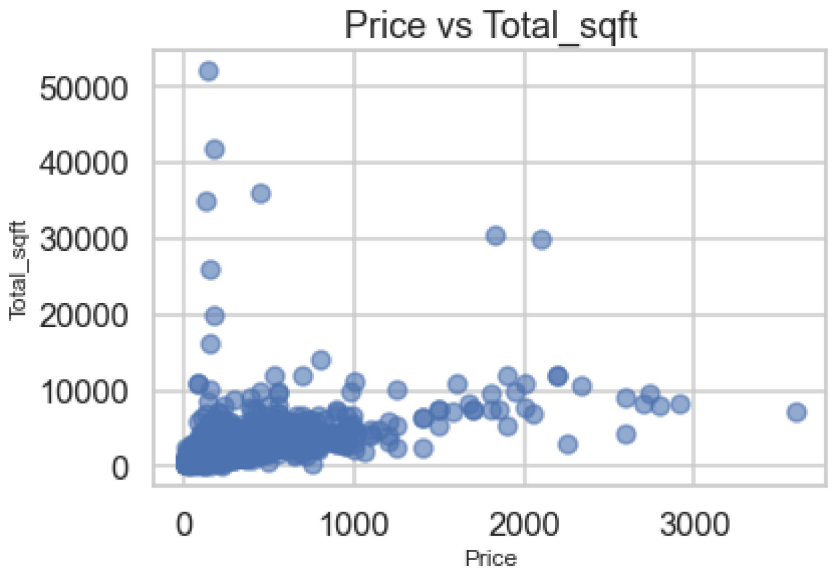
In [155…

```
plt.figure(figsize=(8,6))
sns.heatmap(data.corr(),mask = mask,annot=True)
plt.yticks(fontsize=13)
plt.xticks(fontsize=13)
plt.show()
```



In [156…

```
plt.scatter(x=data['price'],y=data['total_sqft'],alpha=0.6,s=80)
plt.xlabel('Price',fontsize=12)
plt.ylabel('Total_sqft',fontsize=12)
plt.title(f'Price vs Total_sqft')
plt.show()
```

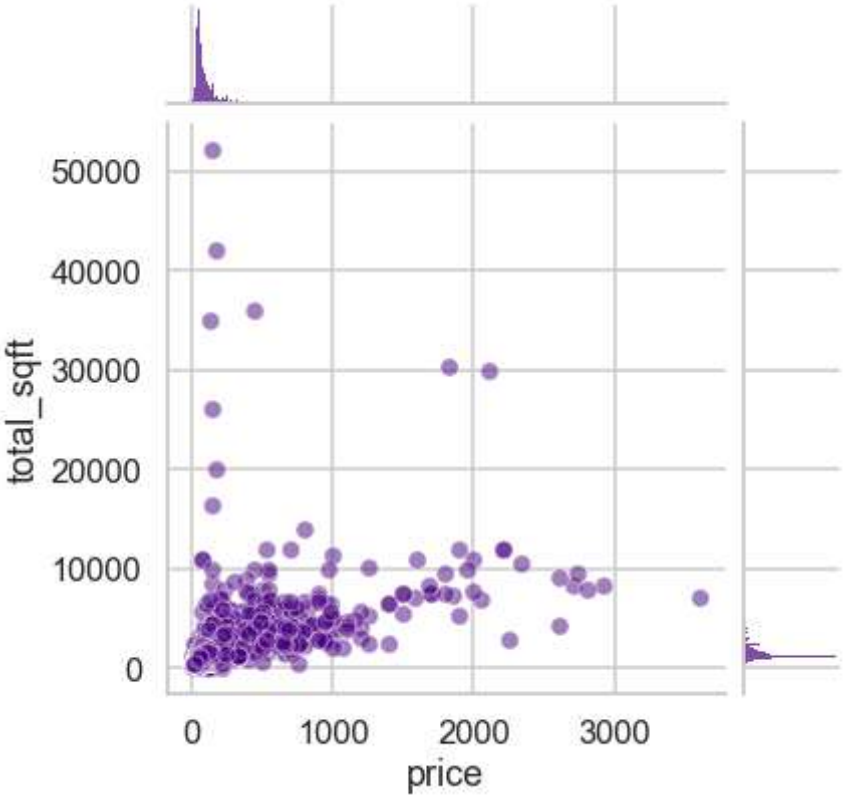


In [157…

```
sns.set()
sns.set_style('whitegrid')
sns.set_context('talk')
sns.jointplot(x=data['price'],y=data['total_sqft'],color='indigo', joint_kws={'alpha
```

Out[157…

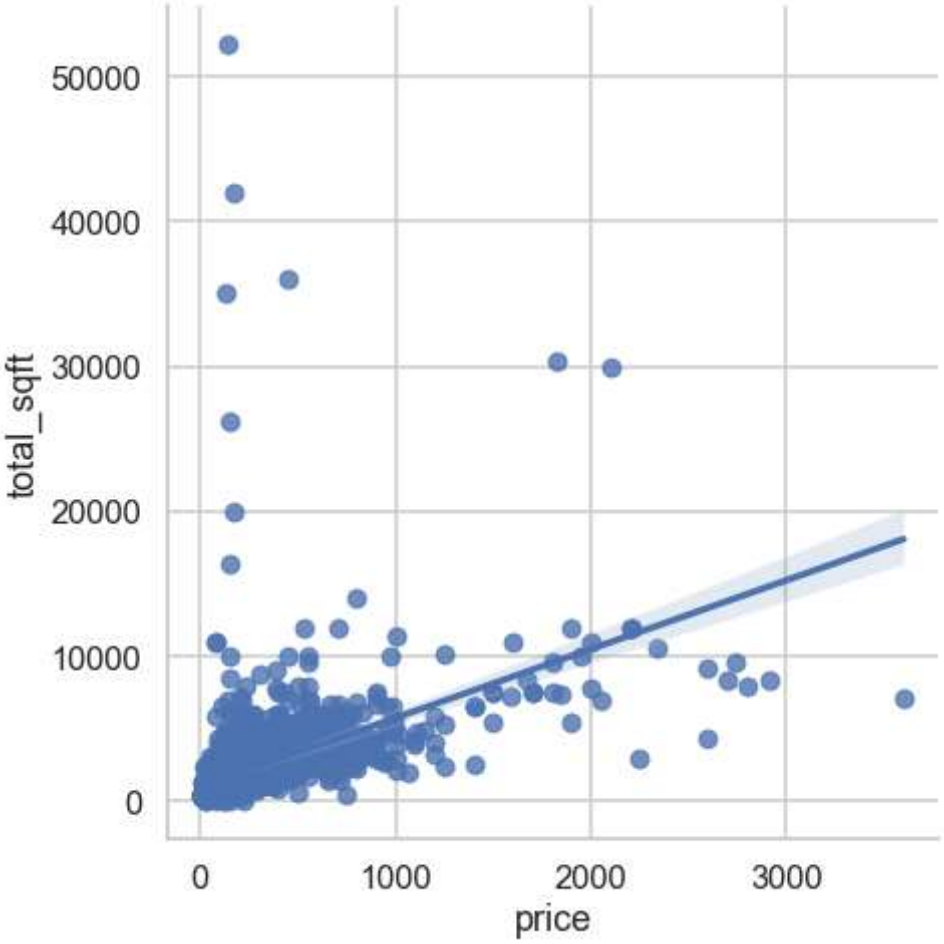
```
<seaborn.axisgrid.JointGrid at 0x2d3de1ea430>
```



In [158…

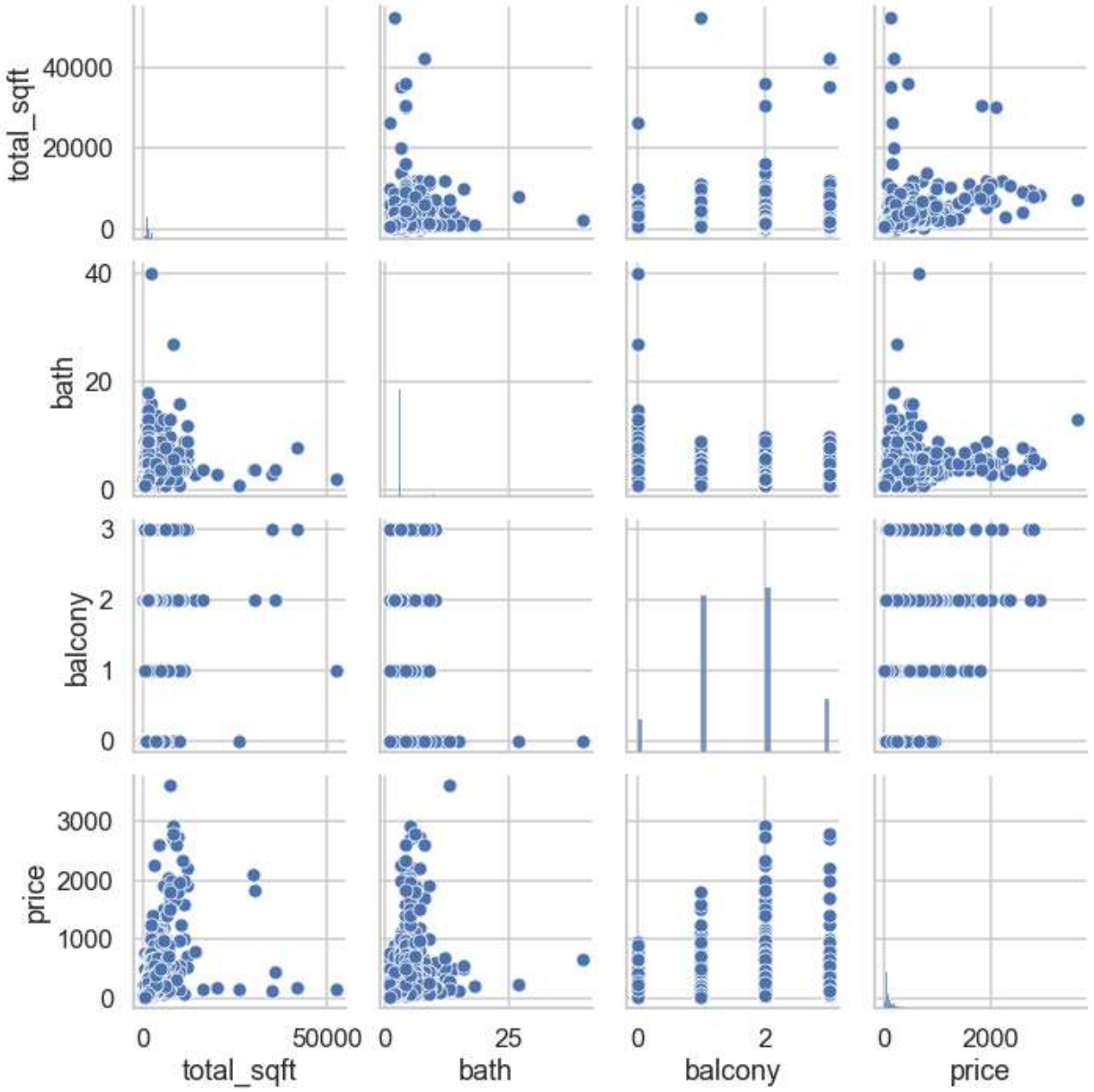
```
sns.lmplot(x='price',y='total_sqft',data=data,size=7)
plt.show()
```

```
C:\Users\sajid\anaconda3\lib\site-packages\seaborn\regression.py:581: UserWarning: T
he `size` parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```



In [159…

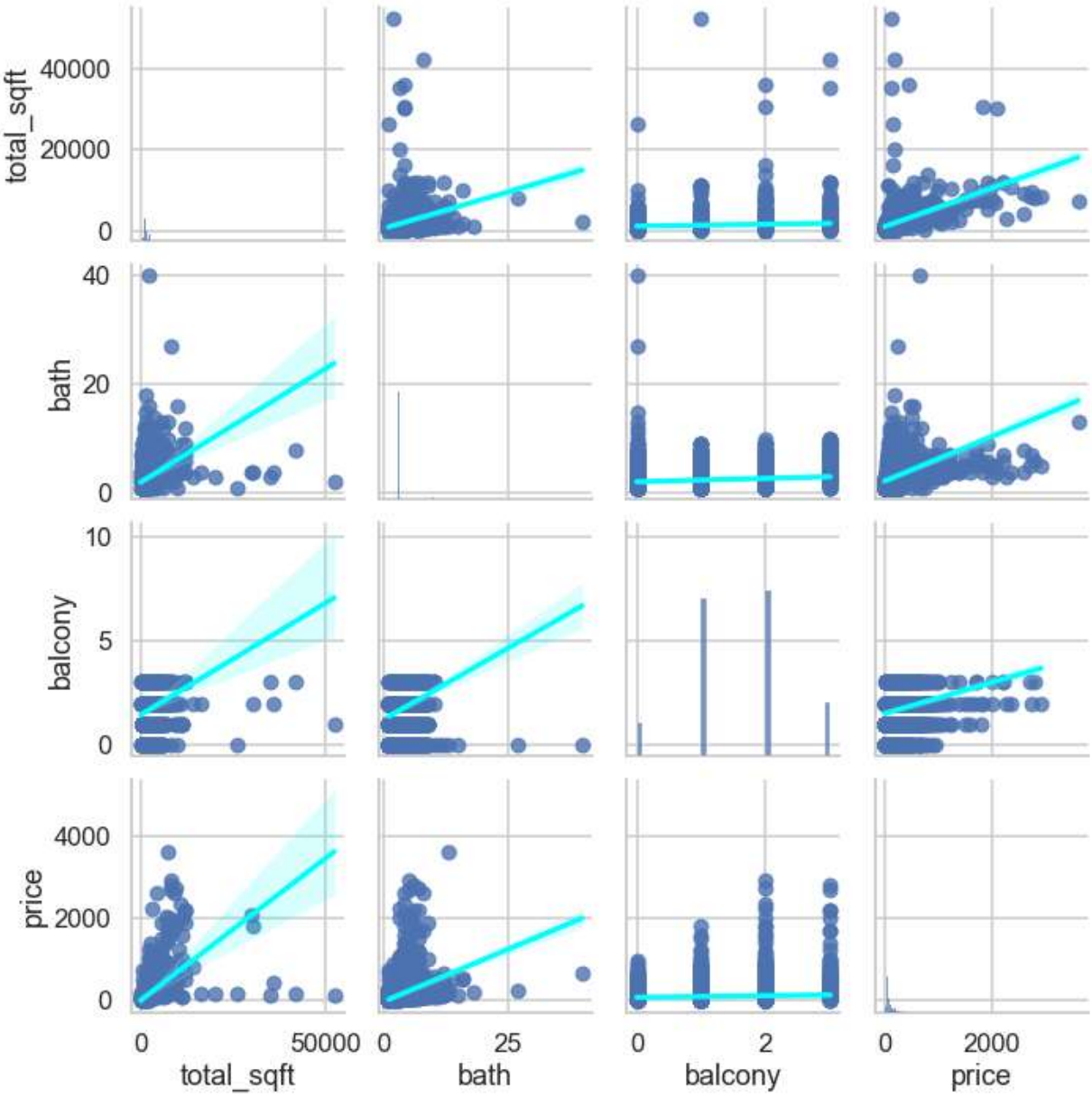
```
%%time
sns.pairplot(data)
plt.show()
```




```
Wall time: 17.4 s
```

In [160…

```
sns.pairplot(data, kind='reg', plot_kws={'line_kws': {'color': 'cyan'}})
plt.show()
```



Training our model by multivariable regression

Train Test split

In [161…

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [162…

```
#Dropping the multiple columns

data.drop(['area_type', 'availability','location','size','society'], axis=1, inplace
```

In [163…

```
prices = data['price']
features = data.drop('price',axis=1)

X_train, X_test, y_train, y_test = train_test_split(features,prices,
```

```
                                                    test_size=0.2,random_state=10)
len(X_train)/len(features)
```

Out[163…
```
0.8
```

In [164…
```
# % number of test dataset
X_test.shape[0]/features.shape[0]
```

Out[164…
```
0.2
```

In [165…
```
non_numeric_columns = X_train.select_dtypes(include='object').columns
print(non_numeric_columns)
```

```
Index([], dtype='object')
```

In [166…
```
# Drop samples with missing values from X_train and y_train
X_train.dropna(inplace=True)
y_train = y_train[X_train.index]
```

In [167…
```
from sklearn.impute import SimpleImputer

# Create an instance of SimpleImputer with the desired strategy
imputer = SimpleImputer(strategy='mean')

# Fit and transform the imputer on X_train
X_train_imputed = imputer.fit_transform(X_train)

# Fit the linear regression model on the imputed dataset
regr.fit(X_train_imputed, y_train)
```

Out[167…
```
▾ LinearRegression

LinearRegression()
```

In [168…
```
from sklearn.linear_model import LinearRegression
import pandas as pd

# Create an instance of LinearRegression
regr = LinearRegression()

# Fit the model to the training data
regr.fit(X_train, y_train)

# Print the intercept
print('Intercept:', regr.intercept_)

# Create a DataFrame to display the coefficient values
coef_df = pd.DataFrame(data=regr.coef_, index=X_train.columns, columns=['Coefficient
print(coef_df)

#r^2 value
# Create an instance of SimpleImputer with the desired strategy
imputer = SimpleImputer(strategy='mean')

# Fit and transform the imputer on X_train
X_train_imputed = imputer.fit_transform(X_train)

# Transform X_test using the fitted imputer
```

```
X_test_imputed = imputer.transform(X_test)

# Fit the linear regression model on the imputed training dataset
regr.fit(X_train_imputed, y_train)

# Calculate R-squared for the imputed test dataset
test_r2 = regr.score(X_test_imputed, y_test)
print('Test data R-squared:', test_r2)

# Calculate R-squared for the imputed training dataset
train_r2 = regr.score(X_train_imputed, y_train)
print('Training data R-squared:', train_r2)
```

```
Intercept: -57.1131452508743
            Coefficient
total_sqft     0.071284
bath          24.187042
balcony       -4.839946
Test data R-squared: 0.19055998660980233
Training data R-squared: 0.44013359461613166
```

Data Transformation

In [169…

```
#finding the skew value
data['price'].skew()
```

Out[169…

```
8.064468821273252
```

In [170…

```
#applying the data transformation to the entire data series of price
y_log = np.log(data['price'])
y_log.head()
```

Out[170…

```
0    3.665355
1    4.787492
2    4.127134
3    4.553877
4    3.931826
Name: price, dtype: float64
```

In [171…

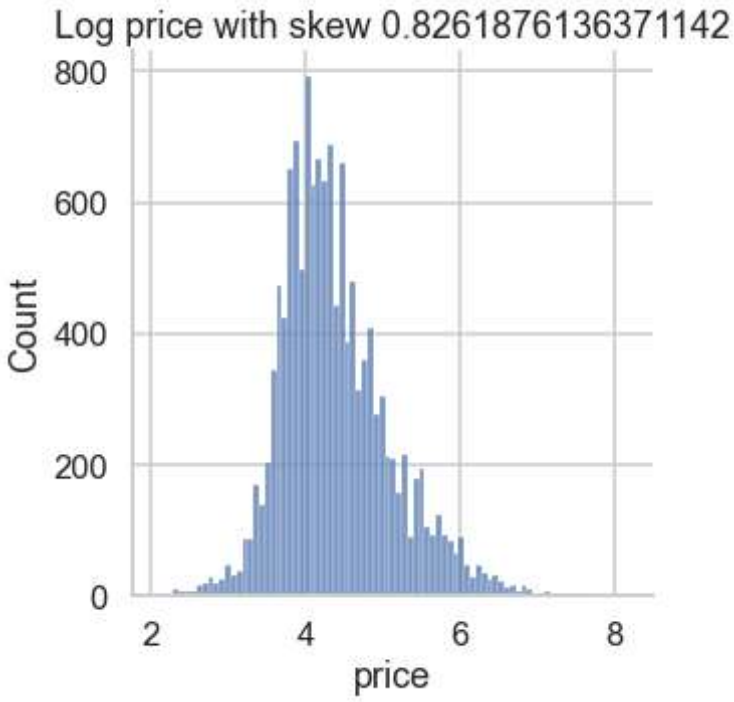
```
y_log.skew()
```

Out[171…

```
0.8261876136371142
```

In [172…

```
sns.displot(y_log)
plt.title(f'Log price with skew {y_log.skew()}')
plt.show()
```



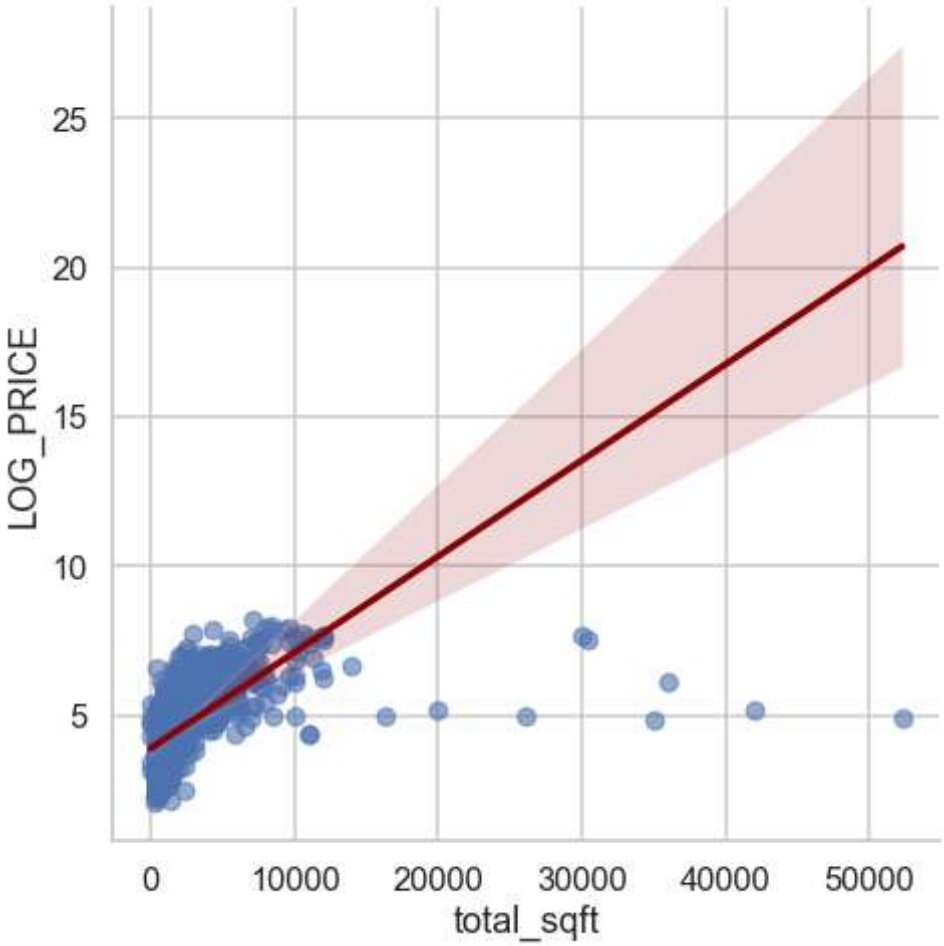
In [173…

```
transformed_data = features
transformed_data['LOG_PRICE'] = y_log



sns.lmplot(x='total_sqft',y='LOG_PRICE',data=transformed_data,size=7,scatter_kws={'a
```

```
C:\Users\sajid\anaconda3\lib\site-packages\seaborn\regression.py:581: UserWarning: T
he `size` parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```

Out[173…

```
<seaborn.axisgrid.FacetGrid at 0x2d3ee3ab9a0>
```



p values & Evaluating coefficients

In [ ]:

In [174…

```
X_incl_const = sm.add_constant(X_train)
model = sm.OLS(y_train, X_incl_const)
results = model.fit()

results.params
```

```
C:\Users\sajid\anaconda3\lib\site-packages\statsmodels\tsa\tsatools.py:142: FutureWa
rning: In a future version of pandas all arguments of concat except for the argument
'objs' will be keyword-only
  x = pd.concat(x[::order], 1)
```

Out[174…

```
const        -57.113145
total_sqft     0.071284
bath          24.187042
balcony       -4.839946
dtype: float64
```

In [175…

```
print(X_incl_const)
```

```
      const  total_sqft  bath  balcony
3875    1.0      1149.0   2.0      2.0
7004    1.0      1255.0   2.0      3.0
6286    1.0      1200.0   2.0      1.0
7746    1.0      1100.0   2.0      3.0
4701    1.0      1645.0   3.0      2.0
...     ...         ...   ...      ...
```

```
11633    1.0      1590.0   2.0      2.0
1344     1.0      1655.0   3.0      2.0
12815    1.0       900.0   2.0      1.0
7293     1.0      1230.0   2.0      2.0
1289     1.0      1239.0   2.0      1.0

[10021 rows x 4 columns]
```

In [176…
```
results.pvalues
```

Out[176…
```
const          1.122834e-88
total_sqft     0.000000e+00
bath          2.405531e-143
balcony        1.081299e-04
dtype: float64
```

In [177…
```
pd.DataFrame({'coef': results.params, 'p-value': round(results.pvalues, 5)})
```

Out[177…

	coef	p-value
const	-57.113145	0.00000
total_sqft	0.071284	0.00000
bath	24.187042	0.00000
balcony	-4.839946	0.00011

Testing Multicollinearity

In [178…
```
variance_inflation_factor(exog=X_incl_const.values, exog_idx=1)
```

Out[178…
```
1.2716858253457326
```

In [179…
```
len(X_incl_const.columns)
X_incl_const.shape[1]
```

Out[179…
```
4
```

In [180…
```
#vif calculate for every single feature in our dataframe.
for i in range(X_incl_const.shape[1]):
    print(variance_inflation_factor(exog=X_incl_const.values, exog_idx=i))
print('All done')
```

```
8.498180156516922
1.2716858253457326
1.2976011349043304
1.0596205824478515
All done
```

In [181…
```
vif = [variance_inflation_factor(exog=X_incl_const.values, exog_idx=i) for i in rang


print(vif)

pd.DataFrame({'coef_name': X_incl_const.columns,'vif':vif})
```

```
[8.498180156516922, 1.2716858253457326, 1.2976011349043304, 1.0596205824478515]
```

Out[181...

	coef_name	vif
0	const	8.498180
1	total_sqft	1.271686
2	bath	1.297601
3	balcony	1.059621

In [182...

```
X_incl_const = sm.add_constant(X_train)
X_incl_const = X_incl_const.drop({'bath','total_sqft'}, axis = 1)

model = sm.OLS(y_train, X_incl_const)
results = model.fit()

org_coef = pd.DataFrame({'coef': results.params, 'p-value': round(results.pvalues, 5


print('BIC is',results.bic)
print('r-squared is', results.rsquared)
```

```
BIC is 125875.99908756578
r-squared is 0.012807183614004125
C:\Users\sajid\anaconda3\lib\site-packages\statsmodels\tsa\tsatools.py:142: FutureWa
rning: In a future version of pandas all arguments of concat except for the argument
'objs' will be keyword-only
  x = pd.concat(x[::order], 1)
```