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

In [2]:
```
df=pd.read_csv("House Price India.csv")
```

In [3]:
```
df
```

Out[3]:

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	condition of the house	...	Built Year	Renovation Year	Postal Code	Lattit
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4	5	...	1921	0	122003	52.8
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0	5	...	1909	0	122004	52.8
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0	3	...	1939	0	122004	52.8
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0	3	...	2001	0	122005	52.9
4	6762812919	42491	3	2.00	2710	4500	1.5	0	0	4	...	1929	0	122006	52.9
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
14615	6762830250	42734	2	1.50	1556	20000	1.0	0	0	4	...	1957	0	122066	52.6
14616	6762830339	42734	3	2.00	1680	7000	1.5	0	0	4	...	1968	0	122072	52.5
14617	6762830618	42734	2	1.00	1070	6120	1.0	0	0	3	...	1962	0	122056	52.7
14618	6762830709	42734	4	1.00	1030	6621	1.0	0	0	4	...	1955	0	122042	52.7
14619	6762831463	42734	3	1.00	900	4770	1.0	0	0	3	...	1969	2009	122018	52.5

14620 rows × 23 columns

In [4]:
```
#to find number of rows and columns
df.shape
```

Out[4]:
```
(14620, 23)
```

In [5]:
```
#general info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14620 entries, 0 to 14619
Data columns (total 23 columns):
 #   Column                                 Non-Null Count  Dtype  
---  ------                                 --------------  -----  
 0   id                                     14620 non-null  int64  
 1   Date                                   14620 non-null  int64  
 2   number of bedrooms                     14620 non-null  int64  
 3   number of bathrooms                    14620 non-null  float64
 4   living area                            14620 non-null  int64  
 5   lot area                               14620 non-null  int64  
 6   number of floors                       14620 non-null  float64
 7   waterfront present                     14620 non-null  int64  
 8   number of views                        14620 non-null  int64  
 9   condition of the house                 14620 non-null  int64  
 10  grade of the house                     14620 non-null  int64  
 11  Area of the house(excluding basement)  14620 non-null  int64  
 12  Area of the basement                   14620 non-null  int64  
 13  Built Year                             14620 non-null  int64  
 14  Renovation Year                        14620 non-null  int64  
 15  Postal Code                            14620 non-null  int64  
 16  Lattitude                              14620 non-null  float64
 17  Longitude                              14620 non-null  float64
 18  living_area_renov                      14620 non-null  int64  
 19  lot_area_renov                         14620 non-null  int64  
 20  Number of schools nearby               14620 non-null  int64  
 21  Distance from the airport              14620 non-null  int64  
 22  Price                                  14620 non-null  int64  
dtypes: float64(4), int64(19)
memory usage: 2.6 MB
```

In [6]:
```
# show top 5 rows
df.head()
```

Out[6]:

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	condition of the house	...	Built Year	Renovation Year	Postal Code	Lattitude
0	6762810145	42491	5	2.50	3650	9050	2.0	0	4	5	...	1921	0	122003	52.8645
1	6762810635	42491	4	2.50	2920	4000	1.5	0	0	5	...	1909	0	122004	52.8878
2	6762810998	42491	5	2.75	2910	9480	1.5	0	0	3	...	1939	0	122004	52.8852
3	6762812605	42491	4	2.50	3310	42998	2.0	0	0	3	...	2001	0	122005	52.9532
4	6762812919	42491	3	2.00	2710	4500	1.5	0	0	4	...	1929	0	122006	52.9047

5 rows × 23 columns

In [7]:
```
#extract all null values
df.isna().sum()
```

Out[7]:
```
id                                     0
Date                                   0
number of bedrooms                     0
number of bathrooms                    0
living area                            0
lot area                               0
number of floors                       0
waterfront present                     0
number of views                        0
condition of the house                 0
grade of the house                     0
Area of the house(excluding basement)  0
Area of the basement                   0
Built Year                             0
Renovation Year                        0
Postal Code                            0
Lattitude                              0
Longitude                              0
living_area_renov                      0
lot_area_renov                         0
Number of schools nearby               0
Distance from the airport              0
Price                                  0
dtype: int64
```

In [8]:
```
df.columns
```

Out[8]:
```
Index(['id', 'Date', 'number of bedrooms', 'number of bathrooms',
       'living area', 'lot area', 'number of floors', 'waterfront present',
       'number of views', 'condition of the house', 'grade of the house',
       'Area of the house(excluding basement)', 'Area of the basement',
       'Built Year', 'Renovation Year', 'Postal Code', 'Lattitude',
       'Longitude', 'living_area_renov', 'lot_area_renov',
       'Number of schools nearby', 'Distance from the airport', 'Price'],
      dtype='object')
```

In [9]:
```
df.dropna(inplace=True)
```

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

Out[10]:
```
id                                     0
Date                                   0
number of bedrooms                     0
number of bathrooms                    0
living area                            0
lot area                               0
number of floors                       0
waterfront present                     0
number of views                        0
condition of the house                 0
grade of the house                     0
Area of the house(excluding basement)  0
Area of the basement                   0
Built Year                             0
Renovation Year                        0
Postal Code                            0
Lattitude                              0
Longitude                              0
living_area_renov                      0
lot_area_renov                         0
Number of schools nearby               0
Distance from the airport              0
Price                                  0
dtype: int64
```

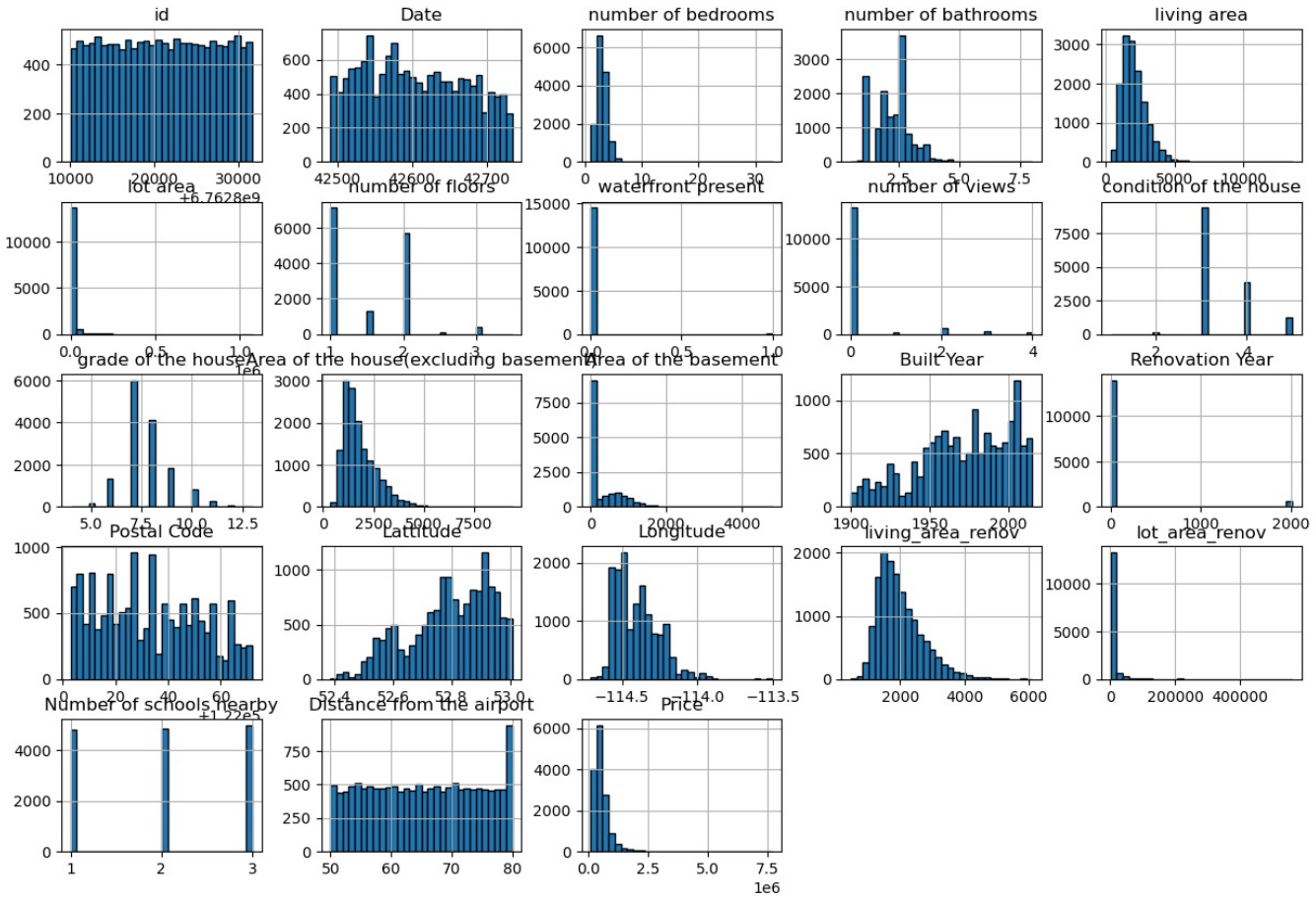
In [11]:
```
df.describe()
```

Out[11]:

	id	Date	number of bedrooms	number of bathrooms	living area	lot area	number of floors	waterfront present	number of views	cor tl
count	1.462000e+04	14620.000000	14620.000000	14620.000000	14620.000000	1.462000e+04	14620.000000	14620.000000	14620.000000	1462
mean	6.762821e+09	42604.538646	3.379343	2.129583	2098.262996	1.509328e+04	1.502360	0.007661	0.233105	
std	6.237575e+03	67.347991	0.938719	0.769934	928.275721	3.791962e+04	0.540239	0.087193	0.766259	
min	6.762810e+09	42491.000000	1.000000	0.500000	370.000000	5.200000e+02	1.000000	0.000000	0.000000	
25%	6.762815e+09	42546.000000	3.000000	1.750000	1440.000000	5.010750e+03	1.000000	0.000000	0.000000	
50%	6.762821e+09	42600.000000	3.000000	2.250000	1930.000000	7.620000e+03	1.500000	0.000000	0.000000	
75%	6.762826e+09	42662.000000	4.000000	2.500000	2570.000000	1.080000e+04	2.000000	0.000000	0.000000	
max	6.762832e+09	42734.000000	33.000000	8.000000	13540.000000	1.074218e+06	3.500000	1.000000	4.000000	

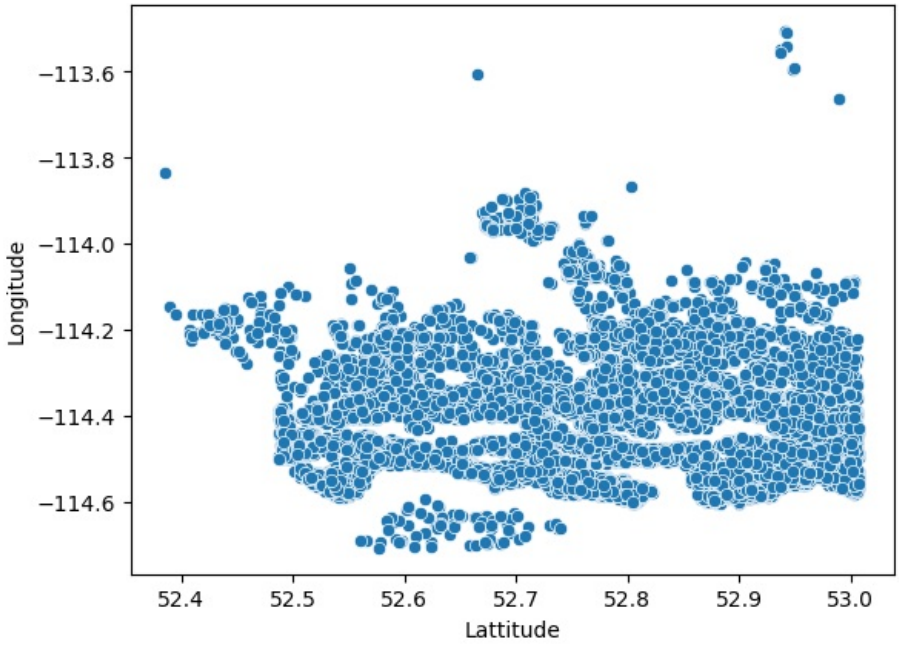
8 rows × 23 columns

In [12]:
```
df.hist(figsize=(15,10),bins=30,edgecolor="black")
plt.show()
```



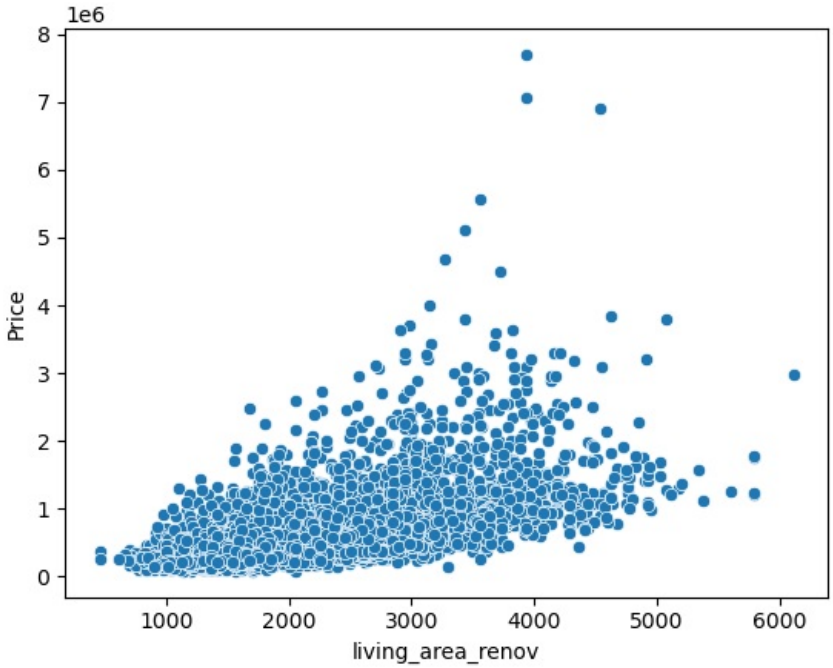
In [13]:
```
sns.scatterplot(x=df['Lattitude'],y=df['Longitude'])
```

Out[13]:
```
<Axes: xlabel='Lattitude', ylabel='Longitude'>
```



In [14]:
```
sns.scatterplot(x=df['living_area_renov'],y=df['Price'])
```

Out[14]:
```
<Axes: xlabel='living_area_renov', ylabel='Price'>
```



In [15]:
```
from sklearn import preprocessing
Label_encode=preprocessing.LabelEncoder()
```

In [16]:
```
#assign in new variables
```

```
df['living_area_renov']=Label_encode.fit_transform(df['living_area_renov'].values)
```

In [17]:
```
#check assigned values
m=df.groupby('living_area_renov')
m=m['living_area_renov']
m.first()
```

Out[17]:
```
living_area_renov
0        0
1        1
2        2
3        3
4        4
      ...
660    660
661    661
662    662
663    663
664    664
Name: living_area_renov, Length: 665, dtype: int64
```

In [18]:
```
#feature selection
columns=['id','Date','number of bedrooms','number of bathrooms','lot area','number of floors','waterfront prese
       'Area of the house(excluding basement)', 'Area of the basement','Built Year','Renovation Year','Postal C
,'living_area_renov', 'Number of schools nearby', 'Distance from the airport']
x=df[columns]
y=df['living area']
```

In [19]:
```
print(x)
```

```
                id   Date  number of bedrooms  number of bathrooms  lot area  \
0       6762810145  42491                   5                 2.50      9050
1       6762810635  42491                   4                 2.50      4000
2       6762810998  42491                   5                 2.75      9480
3       6762812605  42491                   4                 2.50     42998
4       6762812919  42491                   3                 2.00      4500
...            ...    ...                 ...                  ...       ...
14615   6762830250  42734                   2                 1.50     20000
14616   6762830339  42734                   3                 2.00      7000
14617   6762830618  42734                   2                 1.00      6120
14618   6762830709  42734                   4                 1.00      6621
14619   6762831463  42734                   3                 1.00      4770

       number of floors  waterfront present  number of views  \
0                   2.0                   0                4
1                   1.5                   0                0
2                   1.5                   0                0
3                   2.0                   0                0
4                   1.5                   0                0
...                 ...                 ...              ...
14615               1.0                   0                0
14616               1.5                   0                0
14617               1.0                   0                0
14618               1.0                   0                0
14619               1.0                   0                0

       condition of the house  grade of the house  ...  Postal Code  \
0                           5                  10  ...       122003
1                           5                   8  ...       122004
2                           3                   8  ...       122004
3                           3                   9  ...       122005
4                           4                   8  ...       122006
...                       ...                 ...  ...          ...
14615                       4                   7  ...       122066
14616                       4                   7  ...       122072
14617                       3                   6  ...       122056
14618                       4                   6  ...       122042
14619                       3                   6  ...       122018

       Lattitude  Longitude  living_area_renov  lot_area_renov  \
0        52.8645   -114.557                439            5400
1        52.8878   -114.470                347            4000
2        52.8852   -114.468                448            6600
3        52.9532   -114.321                501           42847
4        52.9047   -114.485                256            4500
...          ...        ...                ...             ...
14615    52.6191   -114.472                296           17286
14616    52.5075   -114.393                148            7480
14617    52.7289   -114.507                 53            6120
14618    52.7157   -114.411                113            6631
14619    52.5338   -114.552                 24            3480

       Number of schools nearby  Distance from the airport  living_area_renov  \
0                             2                         58                439
1                             2                         51                347
2                             1                         53                448
3                             3                         76                501
4                             1                         51                256
...                         ...                        ...                ...
14615                         3                         76                296
14616                         3                         59                148
14617                         2                         64                 53
14618                         3                         54                113
14619                         2                         55                 24

       Number of schools nearby  Distance from the airport
0                             2                         58
1                             2                         51
2                             1                         53
3                             3                         76
4                             1                         51
...                         ...                        ...
14615                         3                         76
14616                         3                         59
14617                         2                         64
14618                         3                         54
14619                         2                         55

[14620 rows x 24 columns]
```

In [20]:
```
print(y)
```

```
0        3650
1        2920
2        2910
3        3310
4        2710
         ...
14615    1556
14616    1680
14617    1070
14618    1030
14619     900
Name: living area, Length: 14620, dtype: int64
```

In [21]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x , y,train_size=0.7 ,test_size=0.3)
```

In [22]:
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
model= LinearRegression()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)
model.score(x_test,y_test)
```

Out[22]:
```
1.0
```

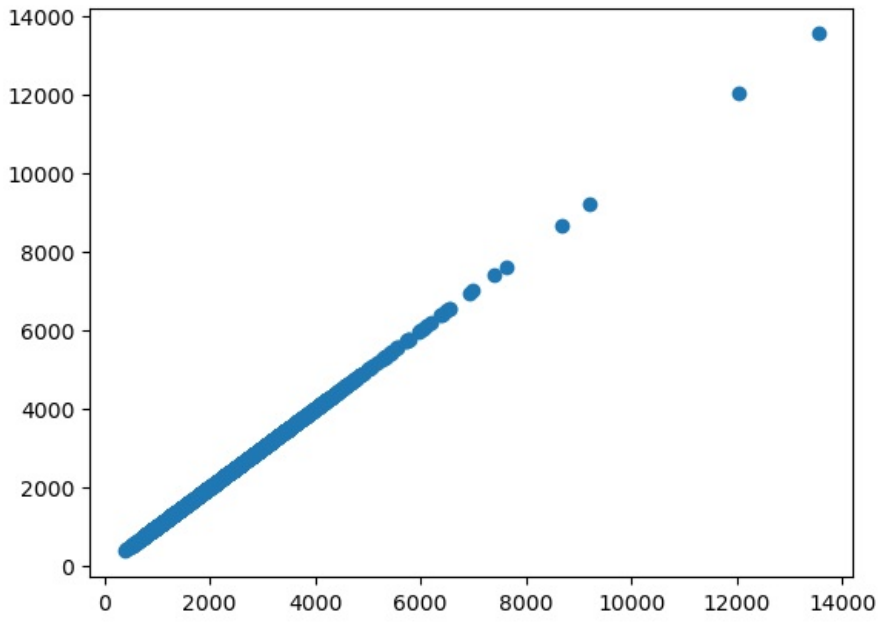
In [23]:
```
from sklearn.metrics import r2_score
score = r2_score(y_test,y_pred)
print("the accuracy of our model is{}%".format(round(score,2)*100))
```

```
the accuracy of our model is100.0%
```

In [24]:
```
predictions=model.predict(x_test)
```

In [25]:
```
plt.scatter(y_test,predictions)
```

Out[25]:
```
<matplotlib.collections.PathCollection at 0x1d01964e3d0>
```



In [ ]:

In [ ]:

In [ ]: