In [59]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sklearn.preprocessing import StandardScaler
```

In [62]:
```
file_path="C:\\Users\\Venkatesh\\TechnoHacks internship\\Data Files\\kc_house_
df=pd.read_csv(file_path)
df
```

Out[62]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floo
0	7129300520	20141013T000000	221900.0	3	1.00	1180	5650	1
1	6414100192	20141209T000000	538000.0	3	2.25	2570	7242	2
2	5631500400	20150225T000000	180000.0	2	1.00	770	10000	1
3	2487200875	20141209T000000	604000.0	4	3.00	1960	5000	1
4	1954400510	20150218T000000	510000.0	3	2.00	1680	8080	1
...	...	...	...	...	...	...	...	
21608	263000018	20140521T000000	360000.0	3	2.50	1530	1131	3
21609	6600060120	20150223T000000	400000.0	4	2.50	2310	5813	2
21610	1523300141	20140623T000000	402101.0	2	0.75	1020	1350	2
21611	291310100	20150116T000000	400000.0	3	2.50	1600	2388	2
21612	1523300157	20141015T000000	325000.0	2	0.75	1020	1076	2

21613 rows × 21 columns

In [63]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   id             21613 non-null  int64  
 1   date           21613 non-null  object 
 2   price          21613 non-null  float64
 3   bedrooms       21613 non-null  int64  
 4   bathrooms      21613 non-null  float64
 5   sqft_living    21613 non-null  int64  
 6   sqft_lot       21613 non-null  int64  
 7   floors         21613 non-null  float64
 8   waterfront     21613 non-null  int64  
 9   view           21613 non-null  int64  
 10  condition      21613 non-null  int64  
 11  grade          21613 non-null  int64  
 12  sqft_above     21613 non-null  int64  
 13  sqft_basement  21613 non-null  int64  
 14  yr_built       21613 non-null  int64  
 15  yr_renovated   21613 non-null  int64  
 16  zipcode        21613 non-null  int64  
 17  lat            21613 non-null  float64
 18  long           21613 non-null  float64
 19  sqft_living15  21613 non-null  int64  
 20  sqft_lot15     21613 non-null  int64  
dtypes: float64(5), int64(15), object(1)
memory usage: 3.5+ MB
```

In [64]:
```
df.head(6)
```

Out[64]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors
0	7129300520	20141013T000000	221900.0	3	1.00	1180	5650	1.0
1	6414100192	20141209T000000	538000.0	3	2.25	2570	7242	2.0
2	5631500400	20150225T000000	180000.0	2	1.00	770	10000	1.0
3	2487200875	20141209T000000	604000.0	4	3.00	1960	5000	1.0
4	1954400510	20150218T000000	510000.0	3	2.00	1680	8080	1.0
5	7237550310	20140512T000000	1225000.0	4	4.50	5420	101930	1.0

6 rows × 21 columns

In [65]:
```
df.tail(6)
```

Out[65]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floo
21607	2997800021	20150219T000000	475000.0	3	2.50	1310	1294	2
21608	263000018	20140521T000000	360000.0	3	2.50	1530	1131	3
21609	6600060120	20150223T000000	400000.0	4	2.50	2310	5813	2
21610	1523300141	20140623T000000	402101.0	2	0.75	1020	1350	2
21611	291310100	20150116T000000	400000.0	3	2.50	1600	2388	2
21612	1523300157	20141015T000000	325000.0	2	0.75	1020	1076	2

6 rows × 21 columns

In [66]:
```
# chechking the number of rows and columns in the data
df.shape
```

Out[66]:
```
(21613, 21)
```

In [67]:
```
# To get the column names in the data
df.columns
```

Out[67]:
```
Index(['id', 'date', 'price', 'bedrooms', 'bathrooms', 'sqft_living',
       'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade',
       'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated', 'zipcode',
       'lat', 'long', 'sqft_living15', 'sqft_lot15'],
      dtype='object')
```

In [68]:
```
#To get the datatype of every column from the data
df.dtypes
```

Out[68]:
```
id                int64
date             object
price           float64
bedrooms          int64
bathrooms       float64
sqft_living       int64
sqft_lot          int64
floors          float64
waterfront        int64
view              int64
condition         int64
grade             int64
sqft_above        int64
sqft_basement     int64
yr_built          int64
yr_renovated      int64
zipcode           int64
lat             float64
long            float64
sqft_living15     int64
sqft_lot15        int64
dtype: object
```

In [69]:
```
df.select_dtypes('object')
```

Out[69]:

	date
0	20141013T000000
1	20141209T000000
2	20150225T000000
3	20141209T000000
4	20150218T000000
...	...
21608	20140521T000000
21609	20150223T000000
21610	20140623T000000
21611	20150116T000000
21612	20141015T000000

21613 rows × 1 columns

In [70]:
```
df.select_dtypes('int64')
```

Out[70]:

	id	bedrooms	sqft_living	sqft_lot	waterfront	view	condition	grade	sqft_abov
0	7129300520	3	1180	5650	0	0	3	7	118
1	6414100192	3	2570	7242	0	0	3	7	217
2	5631500400	2	770	10000	0	0	3	6	77
3	2487200875	4	1960	5000	0	0	5	7	105
4	1954400510	3	1680	8080	0	0	3	8	168
...	...	...	...	...	...	...	...	...	
21608	263000018	3	1530	1131	0	0	3	8	153
21609	6600060120	4	2310	5813	0	0	3	8	231
21610	1523300141	2	1020	1350	0	0	3	7	102
21611	291310100	3	1600	2388	0	0	3	8	160
21612	1523300157	2	1020	1076	0	0	3	7	102

21613 rows × 15 columns

In [71]:
```
# To check the missing values
df.isnull().sum()
```

Out[71]:
```
id               0
date             0
price            0
bedrooms         0
bathrooms        0
sqft_living      0
sqft_lot         0
floors           0
waterfront       0
view             0
condition        0
grade            0
sqft_above       0
sqft_basement    0
yr_built         0
yr_renovated     0
zipcode          0
lat              0
long             0
sqft_living15    0
sqft_lot15       0
dtype: int64
```

In [72]:
```
df.isnull().values.any()
```

Out[72]:
```
False
```

In [73]:
```
# statistical measures of the dataset
df.describe()
```

Out[73]:

	id	price	bedrooms	bathrooms	sqft_living	sqft_lot	
count	2.161300e+04	2.161300e+04	21613.000000	21613.000000	21613.000000	2.161300e+04	216
mean	4.580302e+09	5.400881e+05	3.370842	2.114757	2079.899736	1.510697e+04	
std	2.876566e+09	3.671272e+05	0.930062	0.770163	918.440897	4.142051e+04	
min	1.000102e+06	7.500000e+04	0.000000	0.000000	290.000000	5.200000e+02	
25%	2.123049e+09	3.219500e+05	3.000000	1.750000	1427.000000	5.040000e+03	
50%	3.904930e+09	4.500000e+05	3.000000	2.250000	1910.000000	7.618000e+03	
75%	7.308900e+09	6.450000e+05	4.000000	2.500000	2550.000000	1.068800e+04	
max	9.900000e+09	7.700000e+06	33.000000	8.000000	13540.000000	1.651359e+06	

In [74]:
```
b_room=df['bedrooms']
b_room
```

Out[74]:
```
0         3
1         3
2         2
3         4
4         3
        ..
21608     3
21609     4
21610     2
21611     3
21612     2
Name: bedrooms, Length: 21613, dtype: int64
```

In [75]:
```
b_room.value_counts()
```

Out[75]:
```
bedrooms
3     9824
4     6882
2     2760
5     1601
6      272
1      199
7       38
0       13
8       13
9        6
10       3
11       1
33       1
Name: count, dtype: int64
```

In [76]:
```
b_room_mean=round(np.mean(df['bedrooms']),2)
b_room_median=round(np.median(df['bedrooms']),2)
b_room_min=round(np.min(df['bedrooms']),2)
b_room_max=round(np.max(df['bedrooms']),2)
b_room_std=round(np.std(df['bedrooms']),2)

list1=[b_room_mean,b_room_median,b_room_min,b_room_max,b_room_std]
index=['Mean','Median','Min','Max','Std']
pd.DataFrame(list1,columns=['bedrooms'],index=index)
```

Out[76]:

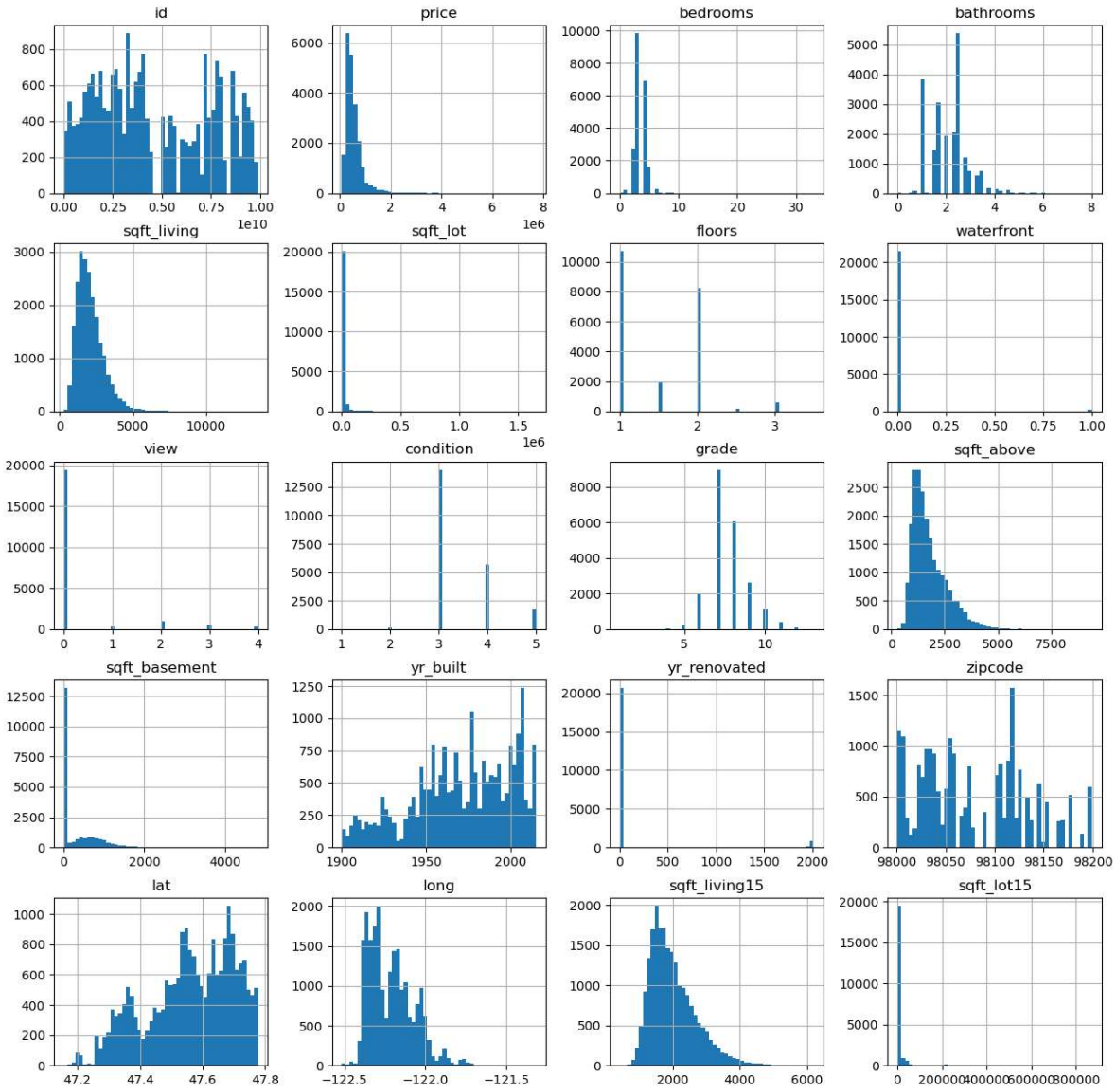
	bedrooms
Mean	3.37
Median	3.00
Min	0.00
Max	33.00
Std	0.93

In [78]:
```
df.describe(include='all').T
```

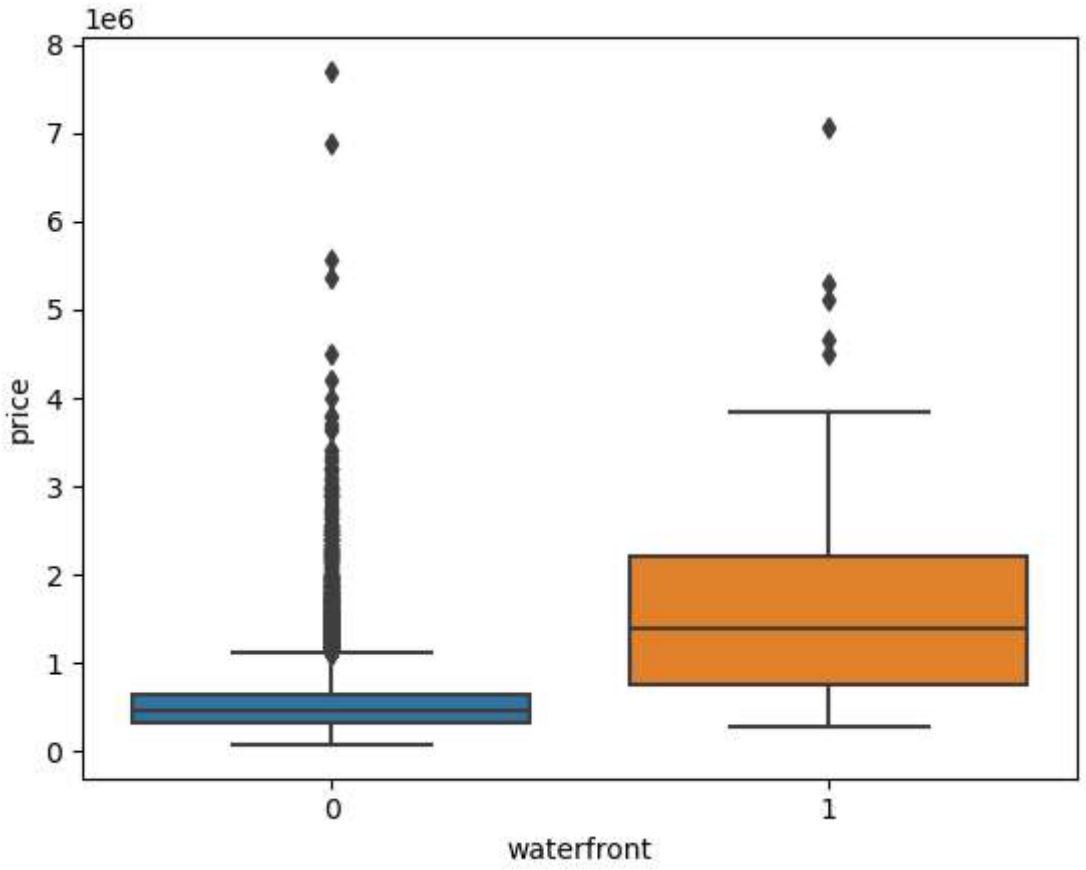
Out[78]:

	count	unique	top	freq	mean	std
id	21613.0	NaN	NaN	NaN	4580301520.864988	2876565571.312049
date	21613	372	20140623T000000	142	NaN	NaN
price	21613.0	NaN	NaN	NaN	540088.141767	367127.196483
bedrooms	21613.0	NaN	NaN	NaN	3.370842	0.930062
bathrooms	21613.0	NaN	NaN	NaN	2.114757	0.770163
sqft_living	21613.0	NaN	NaN	NaN	2079.899736	918.440897
sqft_lot	21613.0	NaN	NaN	NaN	15106.967566	41420.511515
floors	21613.0	NaN	NaN	NaN	1.494309	0.539989
waterfront	21613.0	NaN	NaN	NaN	0.007542	0.086517
view	21613.0	NaN	NaN	NaN	0.234303	0.766318
condition	21613.0	NaN	NaN	NaN	3.40943	0.650743
grade	21613.0	NaN	NaN	NaN	7.656873	1.175459
sqft_above	21613.0	NaN	NaN	NaN	1788.390691	828.090978
sqft_basement	21613.0	NaN	NaN	NaN	291.509045	442.575043
yr_built	21613.0	NaN	NaN	NaN	1971.005136	29.373411
yr_renovated	21613.0	NaN	NaN	NaN	84.402258	401.67924
zipcode	21613.0	NaN	NaN	NaN	98077.939805	53.505026
lat	21613.0	NaN	NaN	NaN	47.560053	0.138564
long	21613.0	NaN	NaN	NaN	-122.213896	0.140828
sqft_living15	21613.0	NaN	NaN	NaN	1986.552492	685.391304
sqft_lot15	21613.0	NaN	NaN	NaN	12768.455652	27304.179631

In [79]:
```
df.hist(bins=50,figsize=(15,15))
plt.show()
```

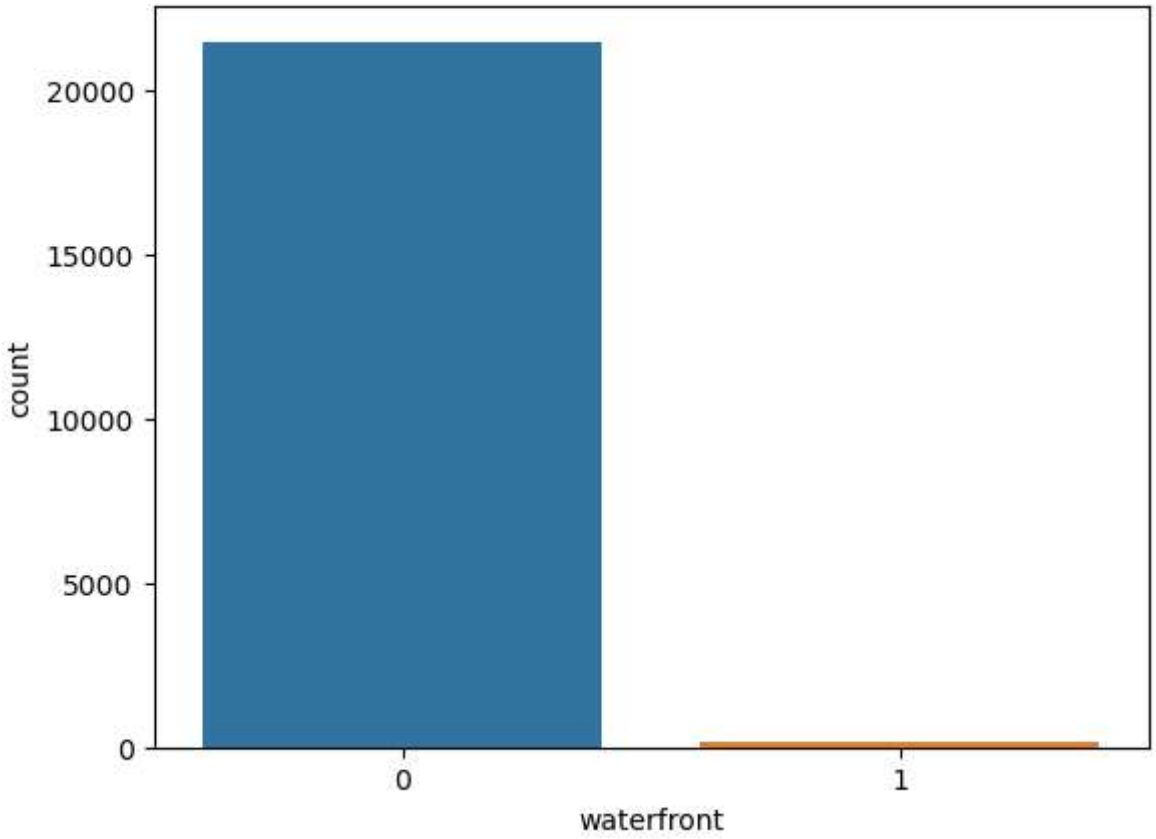


In [80]:
```
sns.boxplot(data=df,x=df['waterfront'],y=df['price'])
plt.show()
```

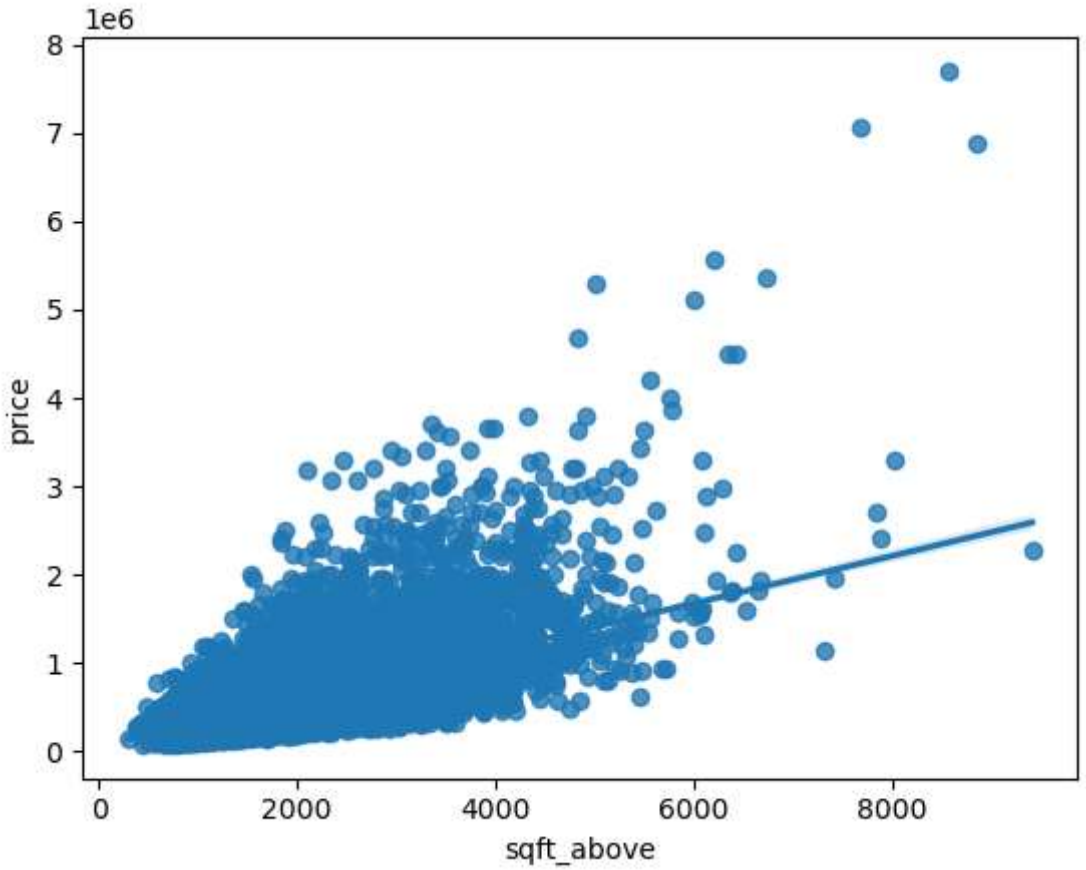


In [81]:
```
sns.countplot(data=df,x=df['waterfront'])
```

Out[81]:
```
<Axes: xlabel='waterfront', ylabel='count'>
```

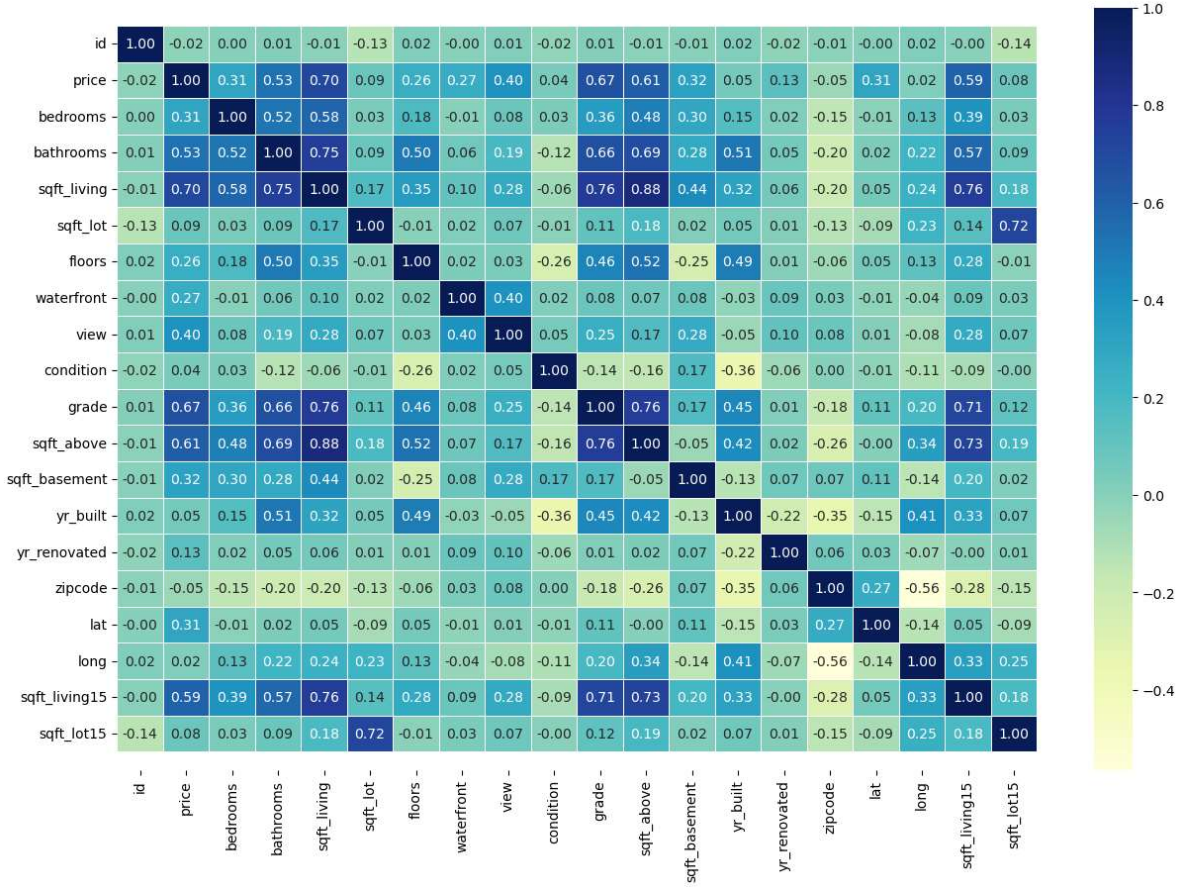


In [83]:
```
sns.regplot(data=df,x=df['sqft_above'],y=df['price'])
plt.show()
```



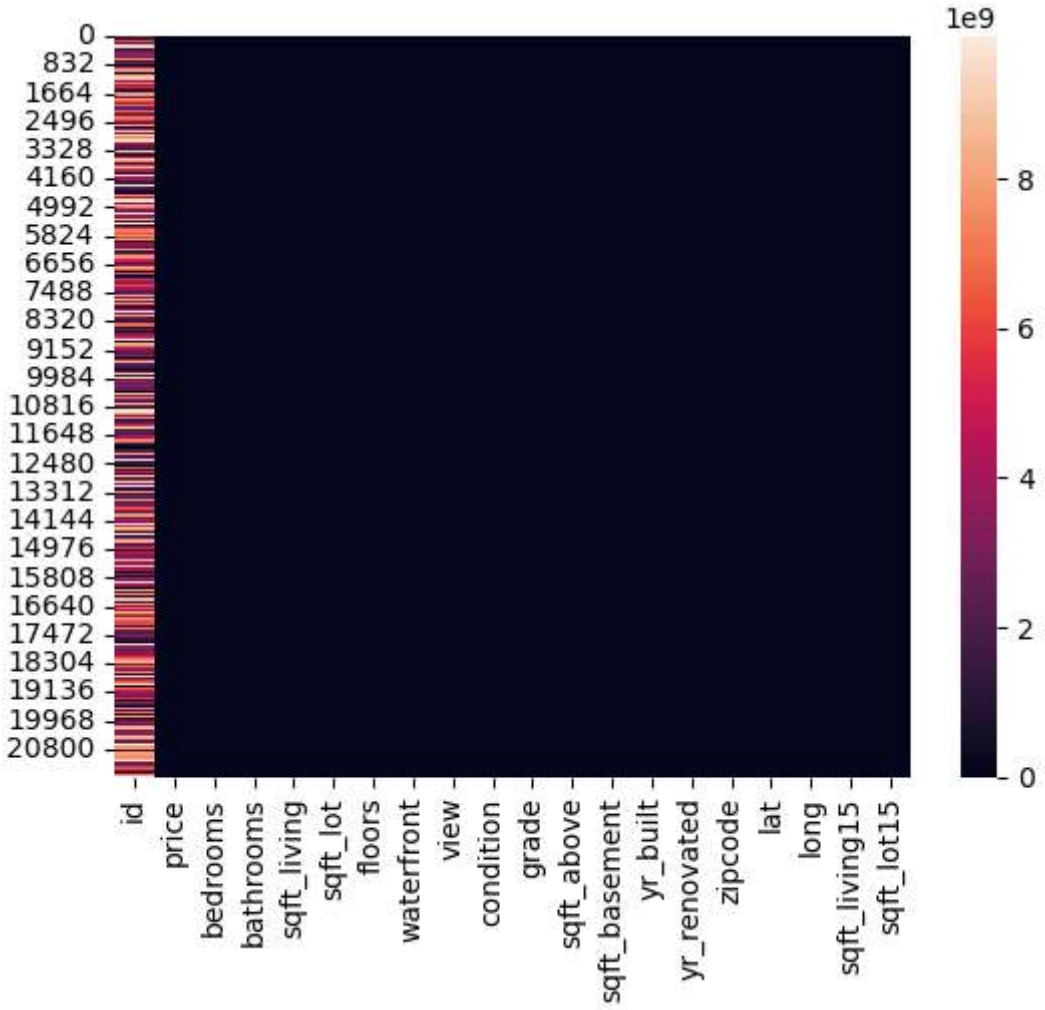
In [86]:
```
df1 = df.drop(columns = 'date')
```

In [88]:
```
corr_matrix = df1.corr()
fig,ax = plt.subplots(figsize=(15,10))
ax = sns.heatmap(corr_matrix,
                 annot=True,
                 linewidths=0.5,
                 fmt=".2f",
                 cmap="YlGnBu");
bottom,top = ax.get_ylim()
ax.set_ylim(bottom + 0.5, top - 0.5)
plt.show()
```



In [89]:
```
sns.heatmap(df1)
```

Out[89]: <Axes: >



In [ ]: