In [10]:
```
import pandas as pd
import numpy as np
import seaborn as sns
df = pd.read_csv('project-IBM data science\data analysis with python\kc_house_dat
df.head()
```

Out[10]:

	Unnamed: 0	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot
0	0	7129300520	20141013T000000	221900.0	3.0	1.00	1180	5650
1	1	6414100192	20141209T000000	538000.0	3.0	2.25	2570	7242
2	2	5631500400	20150225T000000	180000.0	2.0	1.00	770	10000
3	3	2487200875	20141209T000000	604000.0	4.0	3.00	1960	5000
4	4	1954400510	20150218T000000	510000.0	3.0	2.00	1680	8080

5 rows × 22 columns

In [17]:
```
df.dtypes
```

Out[17]:
```
Unnamed: 0         int64
id                 int64
date              object
price            float64
bedrooms         float64
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 [3]:

```
df.drop('id', axis=1, inplace=True)
df.drop('Unnamed: 0', axis=1, inplace=True)
df.describe()
```

Out[3]:

price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	
0e+04	21600.000000	21603.000000	21613.000000	2.161300e+04	21613.000000	21613.000000	21613.00
1e+05	3.372870	2.115736	2079.899736	1.510697e+04	1.494309	0.007542	0.23
2e+05	0.926657	0.768996	918.440897	4.142051e+04	0.539989	0.086517	0.76
0e+04	1.000000	0.500000	290.000000	5.200000e+02	1.000000	0.000000	0.00
0e+05	3.000000	1.750000	1427.000000	5.040000e+03	1.000000	0.000000	0.00
0e+05	3.000000	2.250000	1910.000000	7.618000e+03	1.500000	0.000000	0.00
0e+05	4.000000	2.500000	2550.000000	1.068800e+04	2.000000	0.000000	0.00
0e+06	33.000000	8.000000	13540.000000	1.651359e+06	3.500000	1.000000	4.00

In [7]:

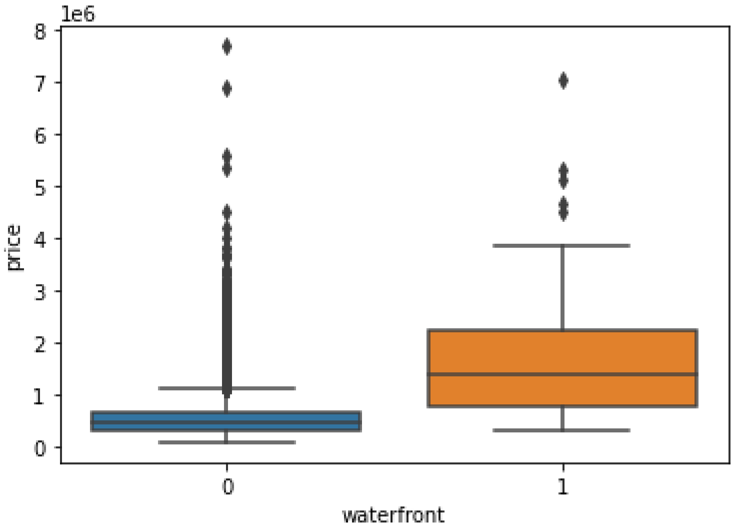
```
df['floors'].value_counts().to_frame()
```

Out[7]:

	floors
1.0	10680
2.0	8241
1.5	1910
3.0	613
2.5	161
3.5	8

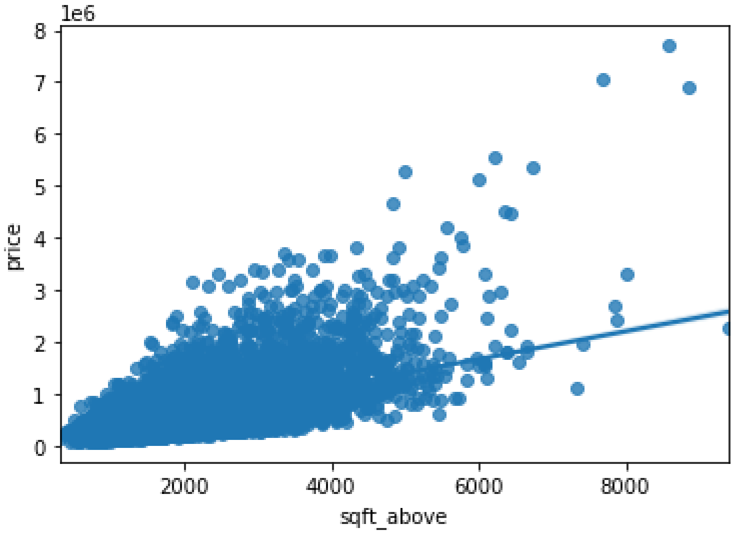
In [12]:
```
sns.boxplot(x='waterfront', y='price', data=df)
```

Out[12]:
```
<AxesSubplot:xlabel='waterfront', ylabel='price'>
```



In [14]:
```
sns.regplot(x='sqft_above', y='price', data=df)
```

Out[14]:
```
<AxesSubplot:xlabel='sqft_above', ylabel='price'>
```



In [16]:
```
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
```

In [19]:
```
X = df[['long']]
Y = df['price']
lm = LinearRegression()
lm
lm.fit(X,Y)
lm.score(X, Y)
U = df[['sqft_living']]
V = df['price']
lm.fit(U,V)
lm.score(U,V)
```

Out[19]:
```
0.4928532179037931
```

In [25]:
```
features =["floors", "waterfront","lat" ,"bedrooms" ,"sqft_basement" ,"view" ,"ba
lm = LinearRegression()
lm

X = df[['floors']]
Y = df['price']

lm.fit(X,Y)
lm.score(X,Y)
```

Out[25]:
```
0.06594310068341092
```

In [27]:
```
Input=[('scale',StandardScaler()),('polynomial', PolynomialFeatures(include_bias=
pipe=Pipeline(Input)
pipe
pipe.fit(X,Y)
pipe.score(X,Y)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-27-71cc21f57b41> in <module>
----> 1 Input=[('scale',StandardScaler()),('polynomial', PolynomialFeatures(inc
lude_bias=False)),('model',LinearRegression())]
      2 pipe=Pipeline(Input)
      3 pipe
      4 pipe.fit(X,Y)
      5 pipe.score(X,Y)

NameError: name 'StandardScaler' is not defined
```

In [32]:
```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Ridge
import matplotlib.pyplot as plt
```

In [29]:

```
features =["floors", "waterfront","lat" ,"bedrooms" ,"sqft_basement" ,"view" ,"ba
X = df[features ]
Y = df['price']

x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.15, random_


print("number of test samples :", x_test.shape[0])
print("number of training samples:",x_train.shape[0])
```

```
number of test samples : 3242
number of training samples: 18371
```

In [33]:

```
pr=PolynomialFeatures(degree=2)
x_train_pr=pr.fit_transform(x_train[['floors', 'waterfront','lat' ,'bedrooms' ,'s
x_test_pr=pr.fit_transform(x_test[['floors', 'waterfront','lat' ,'bedrooms' ,'sq

RidgeModel=Ridge(alpha=0.1)

RidgeModel.fit(x_train_pr, y_train)
RidgeModel.score(x_train_pr, y_train)
width = 12
height = 10
plt.figure(figsize=(width, height))

plt.plot(ALFA,Rsqu_test, label='validation data  ')
plt.plot(ALFA,Rsqu_train, 'r', label='training Data ')
plt.xlabel('alpha')
plt.ylabel('R^2')
plt.legend()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-33-bf737e8aeb84> in <module>
----> 1 pr=PolynomialFeatures(degree=2)
      2 x_train_pr=pr.fit_transform(x_train[['floors', 'waterfront','lat' ,'bed
rooms' ,'sqft_basement' ,'view' ,'bathrooms','sqft_living15','sqft_above','grad
e','sqft_living']])
      3 x_test_pr=pr.fit_transform(x_test[['floors', 'waterfront','lat' ,'bedro
oms' ,'sqft_basement' ,'view' ,'bathrooms','sqft_living15','sqft_above','grade'
,'sqft_living']])
      4 
      5 RidgeModel=Ridge(alpha=0.1)

NameError: name 'PolynomialFeatures' is not defined
```

In [ ]:

In [ ]:

In [ ]: