In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn import preprocessing, svm
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [2]:
```
df=pd.read_csv(r"C:\Users\pavan\Downloads\data.csv")
df
```

Out[2]:

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built
0	2014-05-02 00:00:00	3.130000e+05	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	1955
1	2014-05-02 00:00:00	2.384000e+06	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	1921
2	2014-05-02 00:00:00	3.420000e+05	3.0	2.00	1930	11947	1.0	0	0	4	1930	0	1966
3	2014-05-02 00:00:00	4.200000e+05	3.0	2.25	2000	8030	1.0	0	0	4	1000	1000	1963
4	2014-05-02 00:00:00	5.500000e+05	4.0	2.50	1940	10500	1.0	0	0	4	1140	800	1976
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4595	2014-07-09 00:00:00	3.081667e+05	3.0	1.75	1510	6360	1.0	0	0	4	1510	0	1954
4596	2014-07-09 00:00:00	5.343333e+05	3.0	2.50	1460	7573	2.0	0	0	3	1460	0	1983
4597	2014-07-09 00:00:00	4.169042e+05	3.0	2.50	3010	7014	2.0	0	0	3	3010	0	2009
4598	2014-07-10 00:00:00	2.034000e+05	4.0	2.00	2090	6630	1.0	0	0	3	1070	1020	1974
4599	2014-07-10 00:00:00	2.206000e+05	3.0	2.50	1490	8102	2.0	0	0	4	1490	0	1990

4600 rows × 18 columns

In [4]:
```
df=df[['sqft_living','sqft_above']]
df.columns=['living','above']
```

In [5]:
```
df.head(10)
```

Out[5]:

	living	above
0	1340	1340
1	3650	3370
2	1930	1930
3	2000	1000
4	1940	1140
5	880	880
6	1350	1350
7	2710	2710
8	2430	1570
9	1520	1520

In [6]:
```
df.describe()
```

Out[6]:

	living	above
count	4600.000000	4600.000000
mean	2139.346957	1827.265435
std	963.206916	862.168977
min	370.000000	370.000000
25%	1460.000000	1190.000000
50%	1980.000000	1590.000000
75%	2620.000000	2300.000000
max	13540.000000	9410.000000

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4600 entries, 0 to 4599
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   living  4600 non-null   int64
 1   above   4600 non-null   int64
dtypes: int64(2)
memory usage: 72.0 KB
```

In [8]:
```
df.fillna(method='ffill',inplace=True)
```

```
C:\Users\pavan\AppData\Local\Temp\ipykernel_1300\4116506308.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  df.fillna(method='ffill',inplace=True)
```

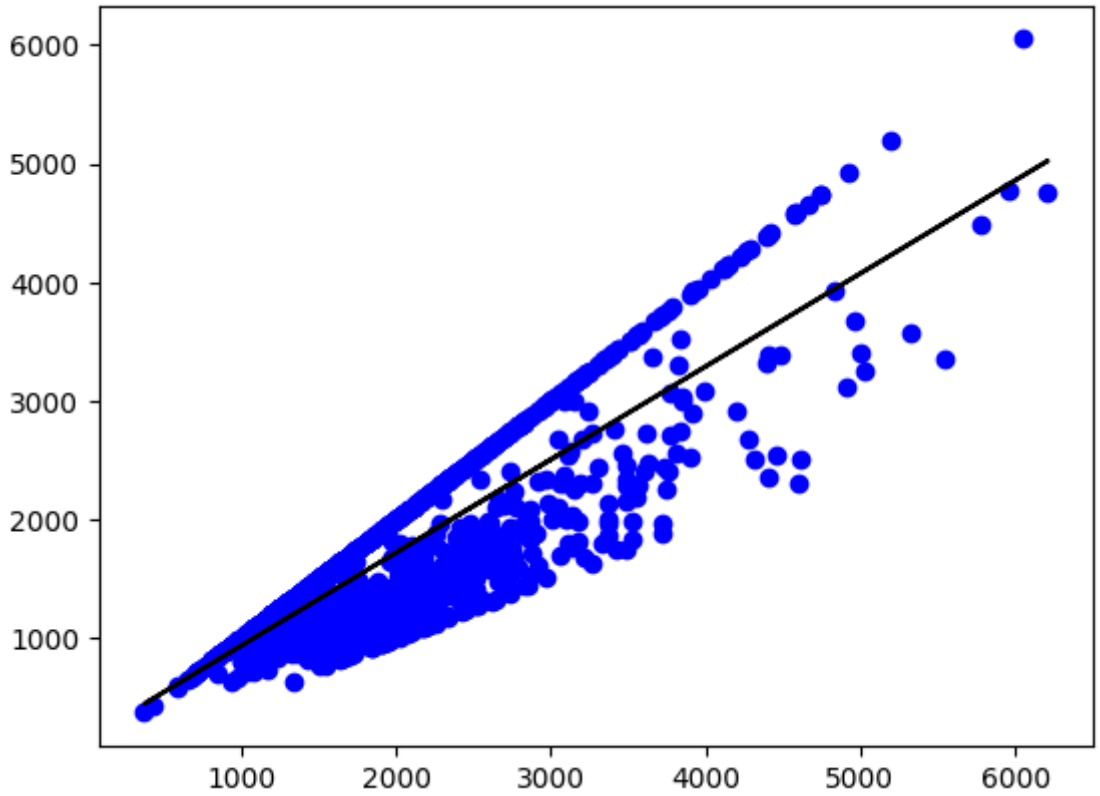
In [9]:
```
x=np.array(df['living']).reshape(-1,1)
y=np.array(df['above']).reshape(-1,1)
```

In [10]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
```

In [11]:
```
regr=LinearRegression()
regr.fit(x_train,y_train)
print(regr.score(x_test,y_test))
```

```
0.7335389803325467
```

In [12]:
```
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```

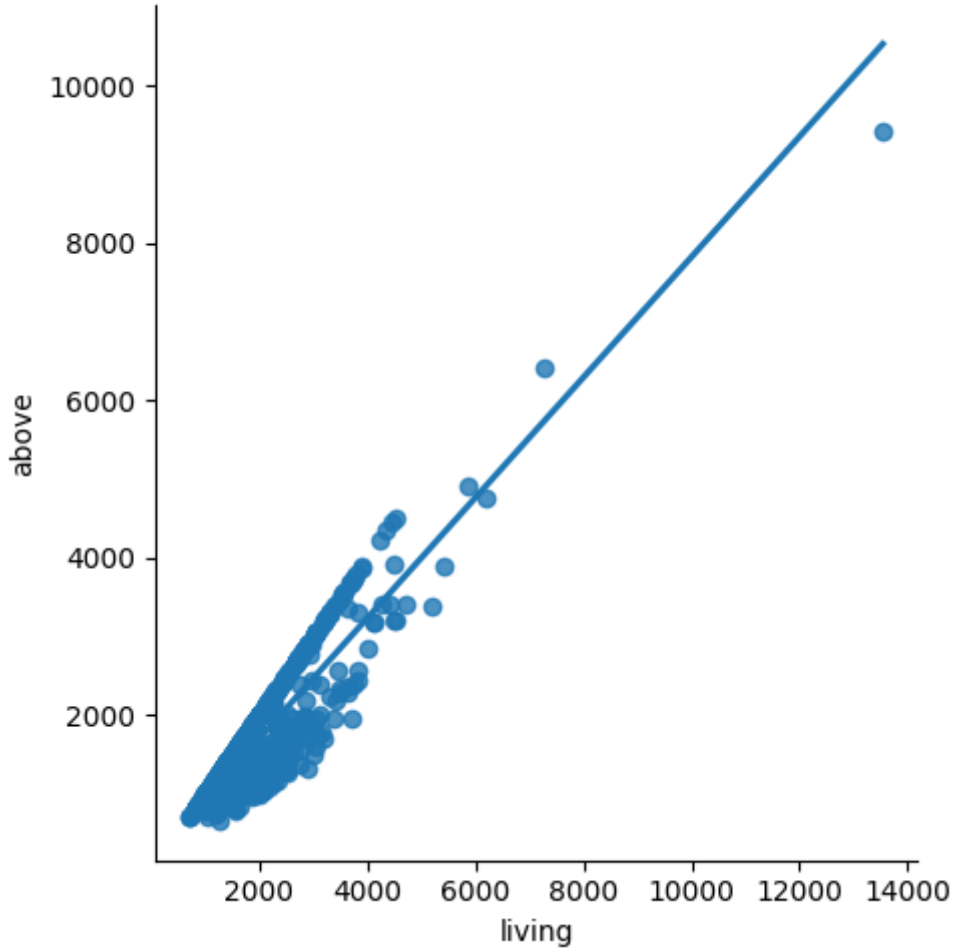


In [13]:
```
df500=df[:][:500]
```

In [16]:
```
df500=df[:][:500]
sns.lmplot(x="living",y="above",data=df500,order=1,ci=None)
df500.fillna(method='ffill',inplace=True)
x=np.array(df500['living']).reshape(-1,1)
y=np.array(df500['above']).reshape(-1,1)
df500.dropna(inplace=True)
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
regr=LinearRegression
regr.fit(x_train,y_train)
print("Regression:",regr.score(x_test,y_test))
y_pred=regr.predict(x_test)
plt.scatter(x_test,y_test,color='b')
plt.plot(x_test,y_pred,color='k')
plt.show()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[16], line 9
      7 x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
      8 regr=LinearRegression
----> 9 regr.fit(x_train,y_train)
     10 print("Regression:",regr.score(x_test,y_test))
     11 y_pred=regr.predict(x_test)

TypeError: LinearRegression.fit() missing 1 required positional argument: 'y'
```



In [17]:
```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
model=LinearRegression()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
r2=r2_score(y_test,y_pred)
print("R2 score:",r2)
```

```
R2 score: 0.715532733363012
```

In [ ]: