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 [7]:

```
df=pd.read_csv(r"C:\Users\Svijayalakshmi\Downloads\data.csv",low_memory=False)
df
```

Out[7]:

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront
0	2014-05-02 00:00:00	3.130000e+05	3.0	1.50	1340	7912	1.5	0
1	2014-05-02 00:00:00	2.384000e+06	5.0	2.50	3650	9050	2.0	0
2	2014-05-02 00:00:00	3.420000e+05	3.0	2.00	1930	11947	1.0	0
3	2014-05-02 00:00:00	4.200000e+05	3.0	2.25	2000	8030	1.0	0
4	2014-05-02 00:00:00	5.500000e+05	4.0	2.50	1940	10500	1.0	0
...	...	...	...	...	...	...	...	...
4595	2014-07-09 00:00:00	3.081667e+05	3.0	1.75	1510	6360	1.0	0
4596	2014-07-09 00:00:00	5.343333e+05	3.0	2.50	1460	7573	2.0	0
4597	2014-07-09 00:00:00	4.169042e+05	3.0	2.50	3010	7014	2.0	0
4598	2014-07-10 00:00:00	2.034000e+05	4.0	2.00	2090	6630	1.0	0
4599	2014-07-10 00:00:00	2.206000e+05	3.0	2.50	1490	8102	2.0	0

4600 rows × 18 columns

In [8]:

```
df=df[['sqft_living','sqft_basement']]
df.columns=['living','basement']
```

In [9]:

```
df.head(10)
```

Out[9]:

	living	basement
0	1340	0
1	3650	280
2	1930	0
3	2000	1000
4	1940	800
5	880	0
6	1350	0
7	2710	0
8	2430	860
9	1520	0

In [10]:

```
df.describe()
```

Out[10]:

	living	basement
count	4600.000000	4600.000000
mean	2139.346957	312.081522
std	963.206916	464.137228
min	370.000000	0.000000
25%	1460.000000	0.000000
50%	1980.000000	0.000000
75%	2620.000000	610.000000
max	13540.000000	4820.000000

In [11]:

```
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   basement  4600 non-null   int64
dtypes: int64(2)
memory usage: 72.0 KB
```

In [12]:

```
df.fillna(method='ffill',inplace=True)
```

```
C:\Users\Svijayalakshmi\AppData\Local\Temp\ipykernel_14112\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-doc
s/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 [14]:

```
x=np.array(df['living']).reshape(-1,1)
y=np.array(df['basement']).reshape(-1,1)
```

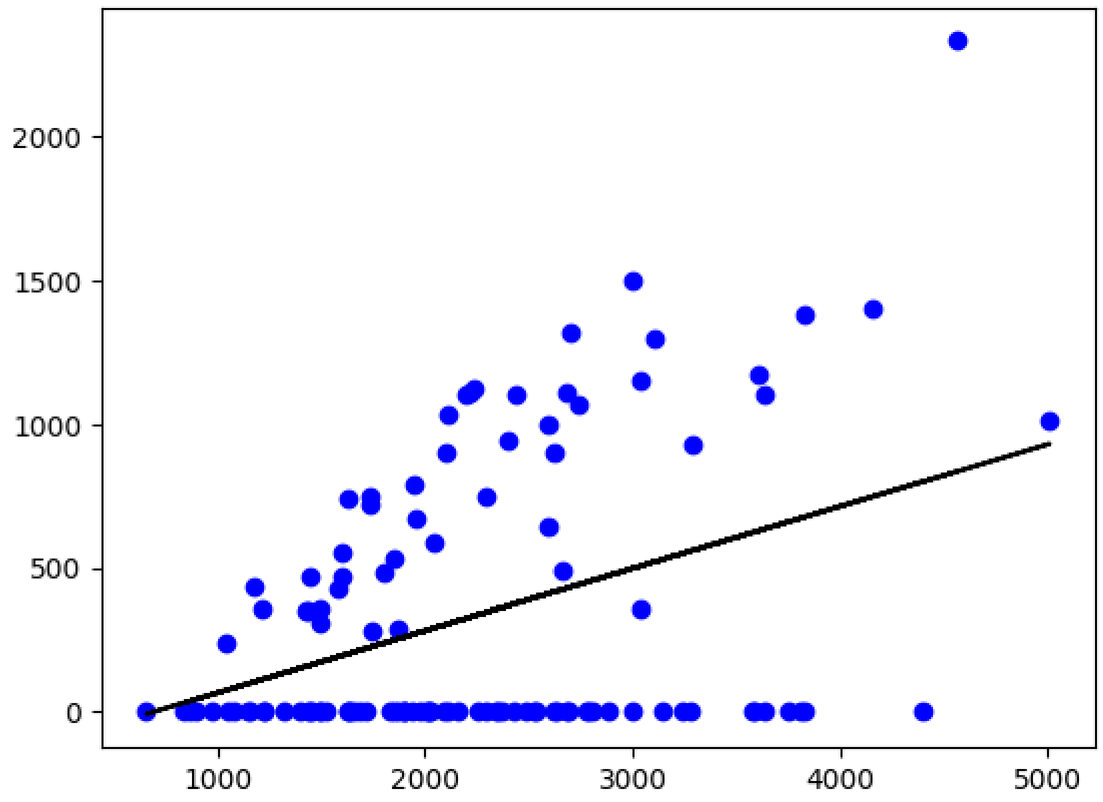
In [15]:

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.025)
```

In [16]:

```
regr=LinearRegression()
regr.fit(x_train,y_train)
print(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()
```

```
0.14489353017494044
```

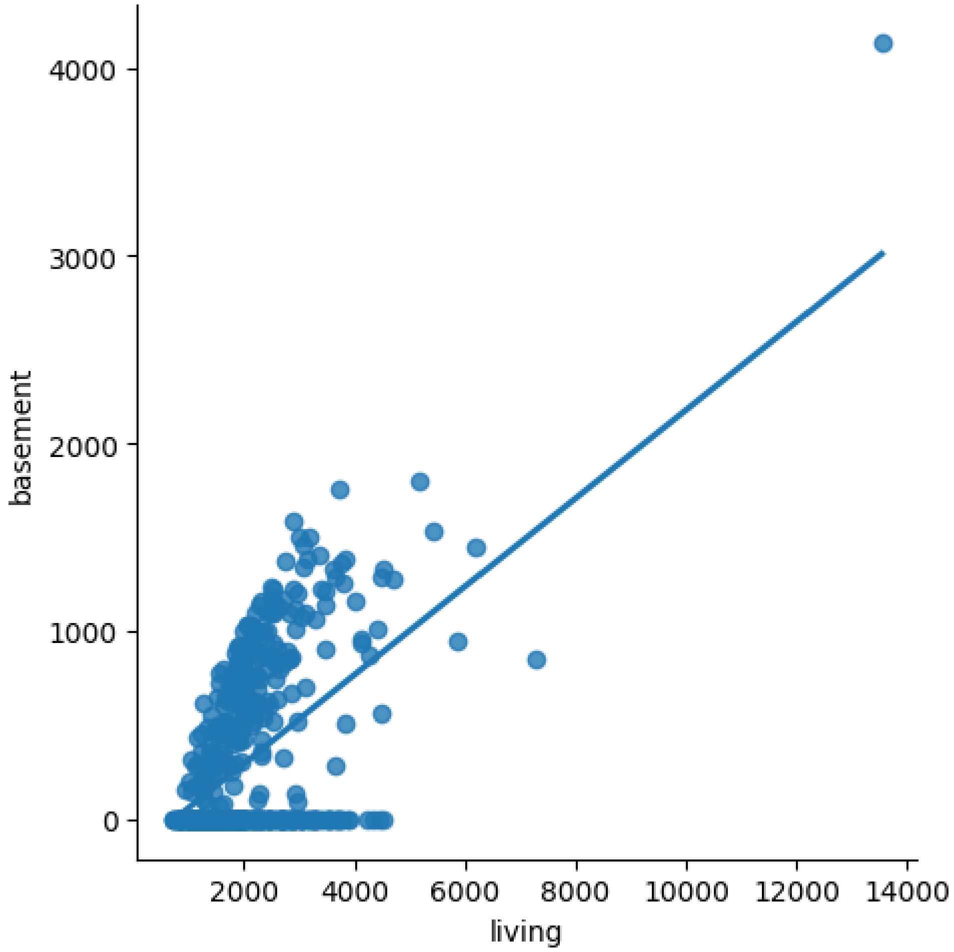


In [18]:

```
df500=df[:][:500]
sns.lmplot(x="living",y="basement",data=df500,order=1,ci=None)
```

Out[18]:

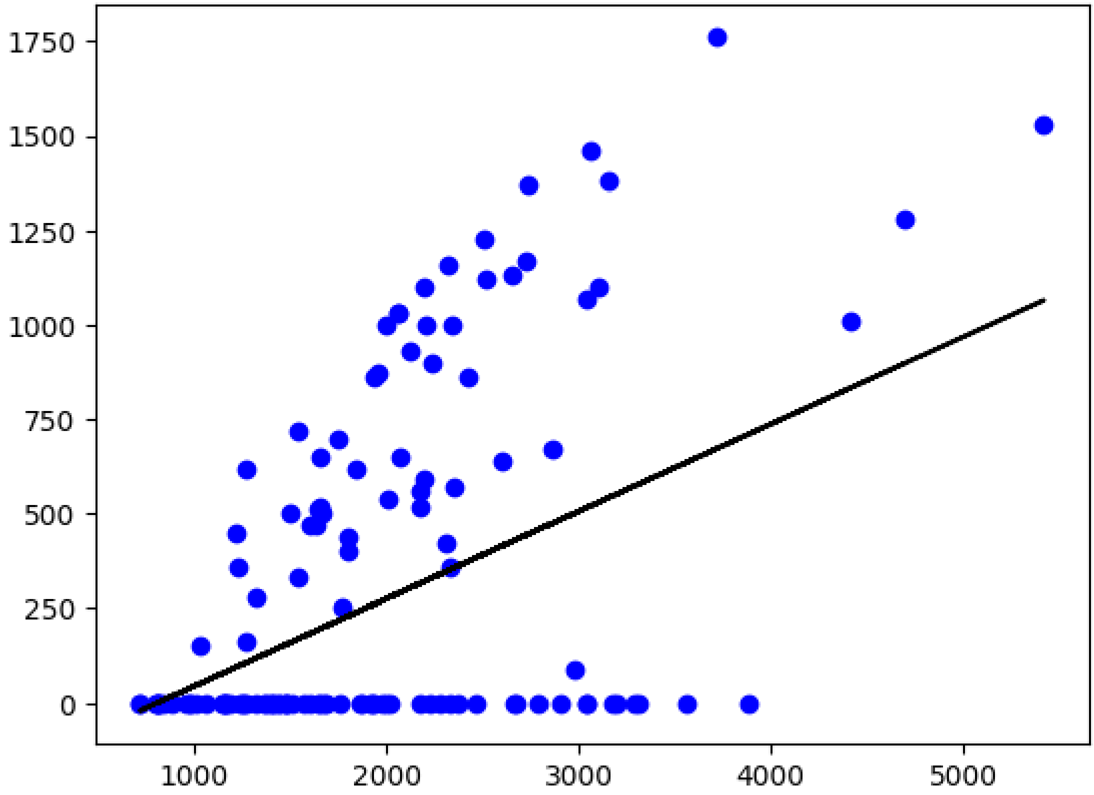
```
<seaborn.axisgrid.FacetGrid at 0x2ce9b6f9790>
```



In [19]:

```
df500.fillna(method='ffill',inplace=True)
x=np.array(df500['living']).reshape(-1,1)
y=np.array(df500['basement']).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()
```

```
regression: 0.2224330491600338
```



In [20]:

```
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: 0.2224330491600338
```

In [ ]: