import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read\_excel("Linear Regression.xlsx")

data.head()

Out[3]:

price sqft\_living bedrooms bathrooms floors

0 221900 1180 3 1.00 1.0

1 538000 2570 3 2.25 2.0

2 180000 770 2 1.00 1.0

3 604000 1960 4 3.00 1.0

4 510000 1680 3 2.00 1.0

data.describe()

Out[4]:

price sqft\_living bedrooms bathrooms floors

count 2.161300e+04 21613.000000 21613.000000 21613.000000 21613.000000

mean 5.400881e+05 2079.899736 3.370842 2.114757 1.494309

std 3.671272e+05 918.440897 0.930062 0.770163 0.539989

min 7.500000e+04 290.000000 0.000000 0.000000 1.000000

25% 3.219500e+05 1427.000000 3.000000 1.750000 1.000000

50% 4.500000e+05 1910.000000 3.000000 2.250000 1.500000

75% 6.450000e+05 2550.000000 4.000000 2.500000 2.000000

max 7.700000e+06 13540.000000 33.000000 8.000000 3.500000

data.hist()

Out[5]:

array([[<matplotlib.axes.\_subplots.AxesSubplot object at 0x000002AF951FE730>,

<matplotlib.axes.\_subplots.AxesSubplot object at 0x000002AF9583EBE0>],

[<matplotlib.axes.\_subplots.AxesSubplot object at 0x000002AF955AC0D0>,

<matplotlib.axes.\_subplots.AxesSubplot object at 0x000002AF9548B4F0>],

[<matplotlib.axes.\_subplots.AxesSubplot object at 0x000002AF952BB940>,

<matplotlib.axes.\_subplots.AxesSubplot object at 0x000002AF95192DF0>]],

dtype=object)

data.hist()

Out[6]:

array([[<matplotlib.axes.\_subplots.AxesSubplot object at 0x000002AF95AD69D0>,

<matplotlib.axes.\_subplots.AxesSubplot object at 0x000002AF95AECBE0>],

[<matplotlib.axes.\_subplots.AxesSubplot object at 0x000002AF95B190D0>,

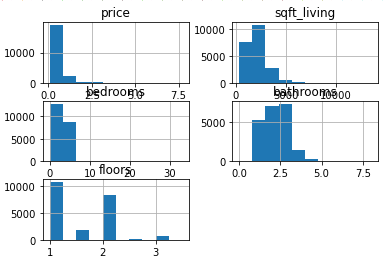
<matplotlib.axes.\_subplots.AxesSubplot object at 0x000002AF95B51460>],

[<matplotlib.axes.\_subplots.AxesSubplot object at 0x000002AF95B7E8B0>,

<matplotlib.axes.\_subplots.AxesSubplot object at 0x000002AF95BABD60>]],

dtype=object)

￼



data.corr()

Out[7]:

price sqft\_living bedrooms bathrooms floors

price 1.000000 0.702035 0.308350 0.525138 0.256794

sqft\_living 0.702035 1.000000 0.576671 0.754665 0.353949

bedrooms 0.308350 0.576671 1.000000 0.515884 0.175429

bathrooms 0.525138 0.754665 0.515884 1.000000 0.500653

floors 0.256794 0.353949 0.175429 0.500653 1.000000

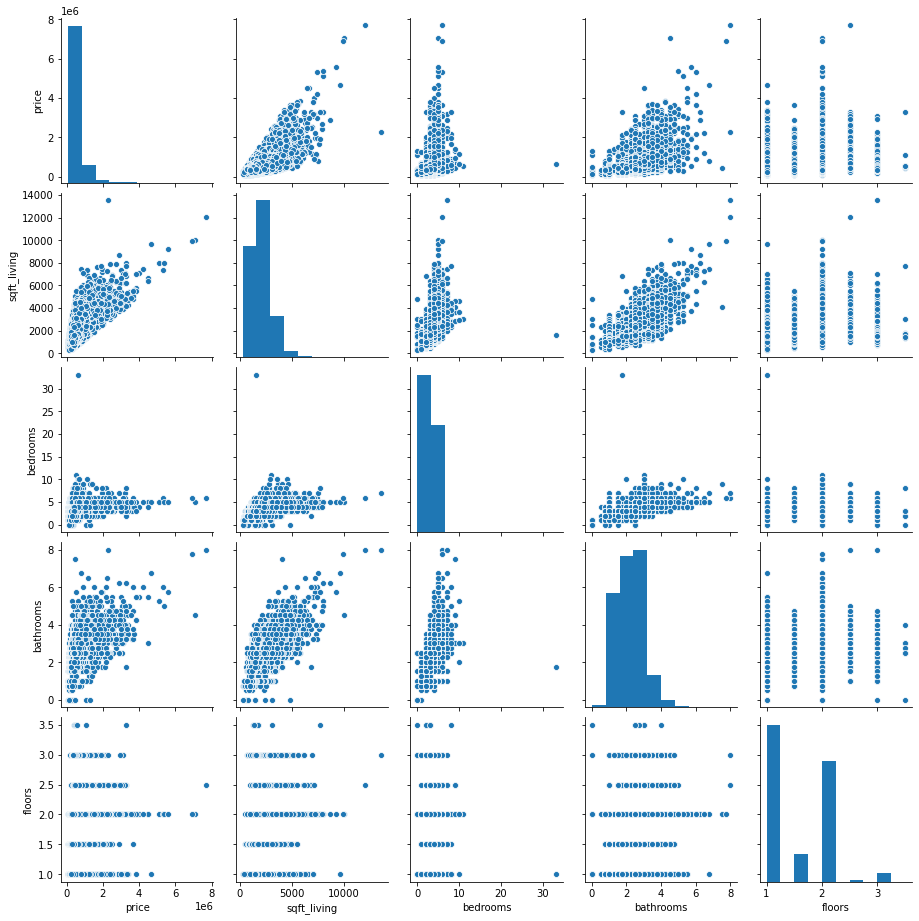
sns.pairplot(data)

Out[8]: <seaborn.axisgrid.PairGrid at 0x2af95772340>

￼

sns.pairplot(data)

Out[9]: <seaborn.axisgrid.PairGrid at 0x2af973d77c0>



data.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 21613 entries, 0 to 21612

Data columns (total 5 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 price 21613 non-null int64

1 sqft\_living 21613 non-null int64

2 bedrooms 21613 non-null int64

3 bathrooms 21613 non-null float64

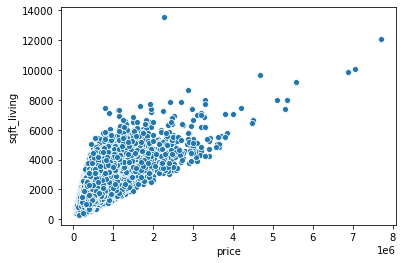
4 floors 21613 non-null float64

dtypes: float64(2), int64(3)

memory usage: 844.4 KB

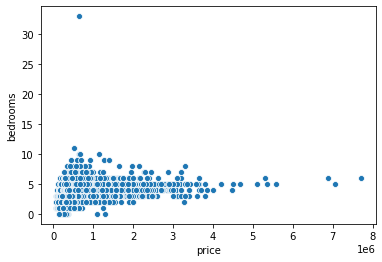
sns.scatterplot(data['price'],data['sqft\_living'])

Out[11]: <matplotlib.axes.\_subplots.AxesSubplot at 0x2af992042e0>



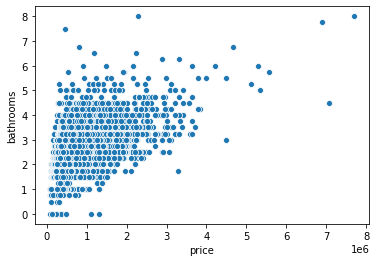
sns.scatterplot(data['price'],data['bedrooms'])

Out[12]: <matplotlib.axes.\_subplots.AxesSubplot at 0x2af995efdf0>



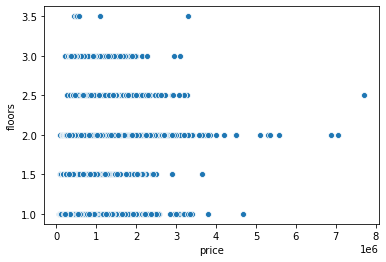
sns.scatterplot(data['price'],data['bathrooms'])

Out[13]: <matplotlib.axes.\_subplots.AxesSubplot at 0x2af9561bc40>



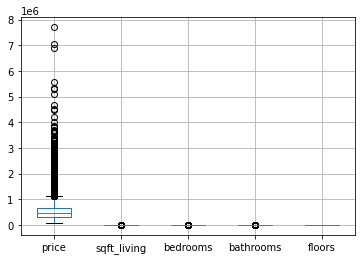
sns.scatterplot(data['price'],data['floors'])

Out[14]: <matplotlib.axes.\_subplots.AxesSubplot at 0x2af99666040>



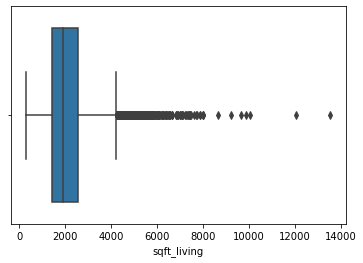
data.boxplot()

Out[15]: <matplotlib.axes.\_subplots.AxesSubplot at 0x2af99726dc0>



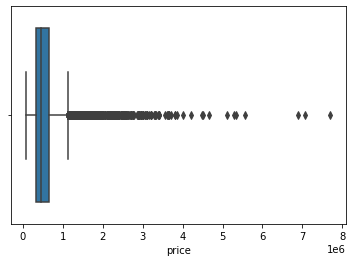
sns.boxplot(data["sqft\_living"])

Out[16]: <matplotlib.axes.\_subplots.AxesSubplot at 0x2af9965ef70>



sns.boxplot(data["price"])

Out[17]: <matplotlib.axes.\_subplots.AxesSubplot at 0x2af9b370d90>



# Price vs bathroom

x = data.drop(['price','sqft\_living','bedrooms','floors'], axis=1)

y = data.drop(['sqft\_living','bedrooms','bathrooms','floors'], axis=1)

from sklearn.model\_selection import train\_test\_split

train\_x,test\_x,train\_y,test\_y = train\_test\_split(x,y,random\_state = 42,test\_size=0.25)

from sklearn.linear\_model import LinearRegression

lr = LinearRegression()

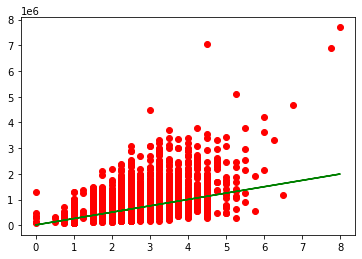
lr.fit(train\_x, train\_y)

Out[21]: LinearRegression()

plt.scatter(train\_x, train\_y, color = 'red')

plt.plot(train\_x, lr.predict(train\_x), color = 'green')

plt.show()



ypred = lr.predict(test\_x)

ypred

Out[24]:

array([[571353.12624725],

[756916.93399872],

[633207.72883108],

...,

[633207.72883108],

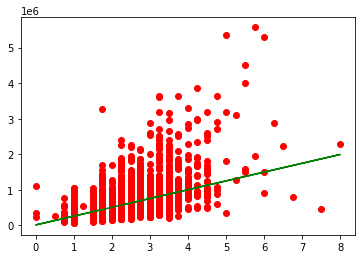
[571353.12624725],

[447643.92107961]])

plt.scatter(test\_x, test\_y, color = 'red')

plt.plot(test\_x, lr.predict(test\_x), color = 'green')

plt.show()



from sklearn.metrics import r2\_score

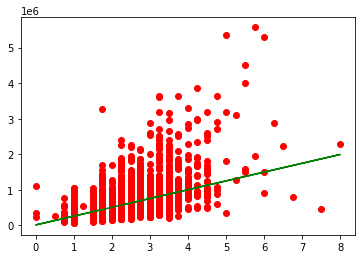
print(r2\_score(test\_y, ypred))

0.2741517408809929

plt.scatter(test\_x, test\_y, color = 'red')

plt.plot(test\_x, lr.predict(test\_x), color = 'green')

plt.show()



unseen\_pred=lr.predict(np.array([[3234]]))

unseen\_pred

Out[29]: array([[8.00165801e+08]])

x.head()

Out[30]:

bathrooms

0 1.00

1 2.25

2 1.00

3 3.00

4 2.00

y.head()

Out[31]:

price

0 221900

1 538000

2 180000

3 604000

4 510000

unseen\_pred=lr.predict(np.array([[323486]]))

unseen\_pred

Out[33]: array([[8.00364065e+10]])

**#Price vs Sq.Ft.ipynb**

x = data.drop(['price','bedrooms','bathrooms','floors'], axis=1)

y = data.drop(['sqft\_living','bedrooms','bathrooms','floors'], axis=1)

x.head()

Out[36]:

sqft\_living

0 1180

1 2570

2 770

3 1960

4 1680

y.head()

Out[37]:

price

0 221900

1 538000

2 180000

3 604000

4 510000

train\_x,test\_x,train\_y,test\_y = train\_test\_split(x,y,random\_state = 24,test\_size=0.35)

lr\_sqft=LinearRegression()

lr\_sqft.fit(train\_x,train\_y)

Out[40]: LinearRegression()

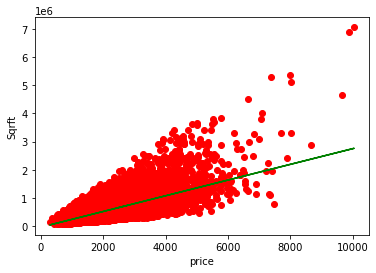
plt.scatter(train\_x, train\_y, color = 'red')

plt.plot(train\_x, lr.predict(train\_x), color = 'green')

plt.xlabel('Sqrft')

plt.ylabel('price')

plt.show()



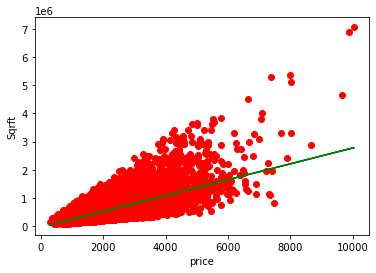
plt.scatter(train\_x, train\_y, color = 'red')

plt.plot(train\_x, lr.predict(train\_x), color = 'green')

plt.xlabel('price')

plt.ylabel('Sqrft')

plt.show()



ypred=lr\_sqft(test\_x)

Traceback (most recent call last):

File "<ipython-input-43-53d6c5d002d7>", line 1, in <module>

ypred=lr\_sqft(test\_x)

TypeError: 'LinearRegression' object is not callable

ypred=lr\_sqft.predict(test\_x)

r2\_score(test\_y, ypred)

Out[45]: 0.5035790021964279

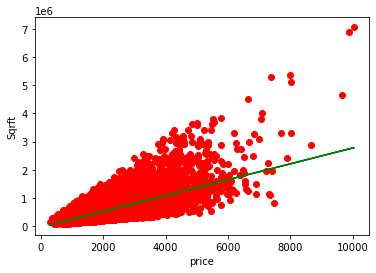
plt.scatter(test\_x, test\_y, color = 'red')

plt.plot(test\_x, lr.predict(test\_x), color = 'green')

plt.xlabel('price')

plt.ylabel('Sqrft')

plt.show()



**# to Check to more accuracy**

train\_x,test\_x,train\_y,test\_y = train\_test\_split(x,y,random\_state = 24,test\_size=0.20)

lr\_sqft1.fit(train\_x,train\_y)

Out[51]: LinearRegression()

plt.scatter(train\_x, train\_y, color = 'red')

plt.plot(train\_x, lr.predict(train\_x), color = 'green')

plt.xlabel('price')

plt.ylabel('Sqrft')

plt.show()

ypred=lr\_sqft1.predict(test\_x)

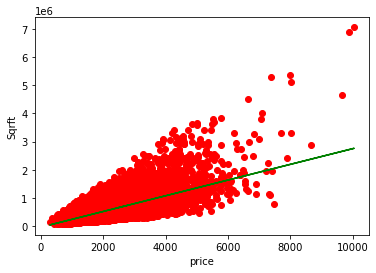
plt.scatter(train\_x, train\_y, color = 'red')

plt.plot(train\_x, lr\_sqft1.predict(train\_x), color = 'green')

plt.xlabel('price')

plt.ylabel('Sqrft')

plt.show()



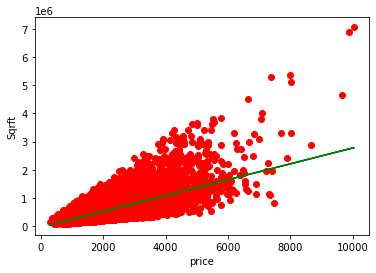
plt.scatter(train\_x, train\_y, color = 'red')

plt.plot(train\_x, lr\_sqft.predict(train\_x), color = 'green')

plt.xlabel('price')

plt.ylabel('Sqrft')

plt.show()



ypred=lr\_sqft1.predict(test\_x)

r2\_score(test\_y, ypred)

Out[57]: 0.5099988522992103

**# Price with BedRoom**

x = data.drop(['price','sqft\_living','bathrooms','floors'], axis=1)

y = data.drop(['sqft\_living','bedrooms','bathrooms','floors'], axis=1)

train\_x,test\_x,train\_y,test\_y = train\_test\_split(x,y,random\_state = 42,test\_size=0.25)

lr\_dedroom=LinearRegression()

lr\_dedroom.fit(train\_x,train\_y)

Out[62]: LinearRegression()

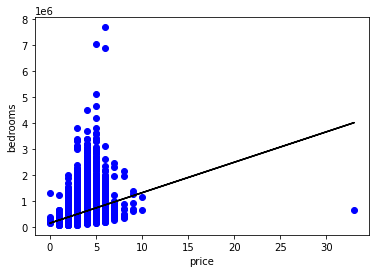
plt.scatter(train\_x, train\_y, color = 'blue')

plt.plot(train\_x, lr\_dedroom.predict(train\_x), color = 'black')

plt.xlabel('price')

plt.ylabel('bedrooms')

plt.show()



ypred = lr\_dedroom.predict(test\_x)

r2\_score(test\_y,ypred)

Out[65]: 0.10025622434499692

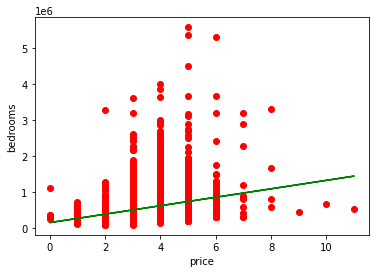
plt.scatter(test\_x, test\_y, color = 'red')

plt.plot(test\_x, lr\_dedroom.predict(test\_x), color = 'green')

plt.xlabel('price')

plt.ylabel('bedrooms')

plt.show()



**# Proce vs Floor**

x = data.drop(['price','sqft\_living','bedrooms','bathrooms'], axis=1)

y = data.drop(['sqft\_living','bedrooms','bathrooms','floors'], axis=1)

train\_x,test\_x,train\_y,test\_y = train\_test\_split(x,y,random\_state = 42,test\_size=0.25)

lr\_floor=LinearRegression()

lr\_floor.fit(train\_x,train\_y)

Out[71]: LinearRegression()

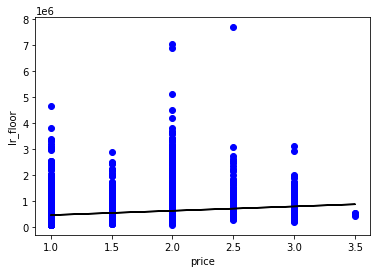
plt.scatter(train\_x, train\_y, color = 'blue')

plt.plot(train\_x, lr\_floor.predict(train\_x), color = 'black')

plt.xlabel('price')

plt.ylabel('lr\_floor')

plt.show()



floor\_predict=lr\_floor.predict(test\_x)

r2\_score(test\_y,floor\_predict)

Out[75]: 0.07175399691766815

plt.scatter(test\_x, test\_y, color = 'red')

plt.plot(test\_x, lr\_floor.predict(test\_x), color = 'green')

plt.xlabel('price')

plt.ylabel('lr\_floor')

plt.show()

