

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns

%matplotlib inline
```

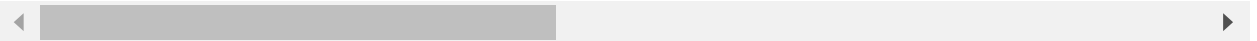
```
In [2]: house = pd.read_csv('C:\\Users\\bittu\\Desktop\\home_data.csv' , encoding="ISO-8859-1")
```

```
In [3]: house.head()
```

Out[3]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront
0	7129300520	20141013T000000	221900	3	1.00	1180	5650	1.0	
1	6414100192	20141209T000000	538000	3	2.25	2570	7242	2.0	
2	5631500400	20150225T000000	180000	2	1.00	770	10000	1.0	
3	2487200875	20141209T000000	604000	4	3.00	1960	5000	1.0	
4	1954400510	20150218T000000	510000	3	2.00	1680	8080	1.0	

5 rows × 21 columns

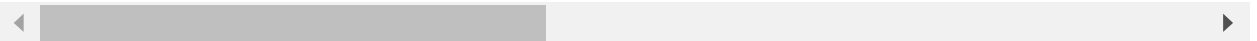


```
In [4]: house.tail()
```

Out[4]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront
21608	263000018	20140521T000000	360000	3	2.50	1530	1131	3.0	
21609	6600060120	20150223T000000	400000	4	2.50	2310	5813	2.0	
21610	1523300141	20140623T000000	402101	2	0.75	1020	1350	2.0	
21611	291310100	20150116T000000	400000	3	2.50	1600	2388	2.0	
21612	1523300157	20141015T000000	325000	2	0.75	1020	1076	2.0	

5 rows × 21 columns



In [5]: `house.info()`

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 21 columns):
id                21613 non-null int64
date              21613 non-null object
price             21613 non-null int64
bedrooms          21613 non-null int64
bathrooms         21613 non-null float64
sqft_living       21613 non-null int64
sqft_lot          21613 non-null int64
floors            21613 non-null float64
waterfront        21613 non-null int64
view              21613 non-null int64
condition         21613 non-null int64
grade             21613 non-null int64
sqft_above        21613 non-null int64
sqft_basement     21613 non-null int64
yr_built          21613 non-null int64
yr_renovated      21613 non-null int64
zipcode           21613 non-null int64
lat               21613 non-null float64
long              21613 non-null float64
sqft_living15     21613 non-null int64
sqft_lot15        21613 non-null int64
dtypes: float64(4), int64(16), object(1)
memory usage: 3.5+ MB

```

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

Out[6]:

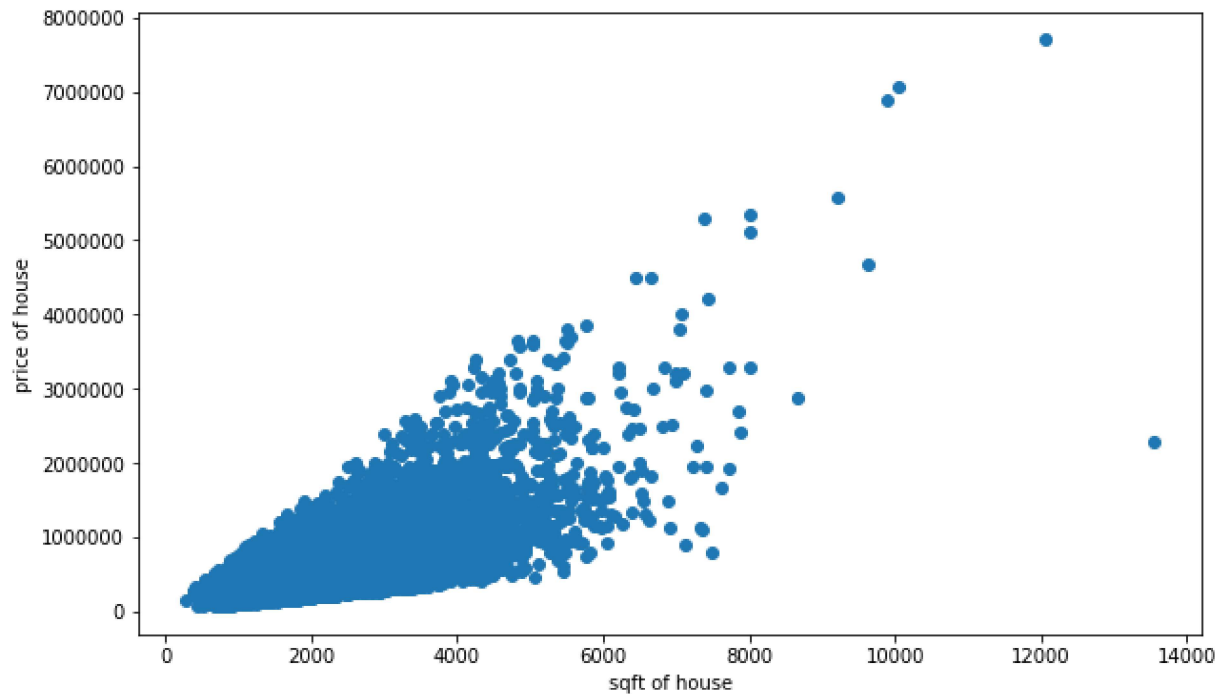
	id	price	bedrooms	bathrooms	sqft_living	sqft_lot	
count	2.161300e+04	2.161300e+04	21613.000000	21613.000000	21613.000000	2.161300e+04	21613
mean	4.580302e+09	5.400881e+05	3.370842	2.114757	2079.899736	1.510697e+04	1
std	2.876566e+09	3.671272e+05	0.930062	0.770163	918.440897	4.142051e+04	0
min	1.000102e+06	7.500000e+04	0.000000	0.000000	290.000000	5.200000e+02	1
25%	2.123049e+09	3.219500e+05	3.000000	1.750000	1427.000000	5.040000e+03	1
50%	3.904930e+09	4.500000e+05	3.000000	2.250000	1910.000000	7.618000e+03	1
75%	7.308900e+09	6.450000e+05	4.000000	2.500000	2550.000000	1.068800e+04	2
max	9.900000e+09	7.700000e+06	33.000000	8.000000	13540.000000	1.651359e+06	3

```
In [7]: house.columns
```

```
Out[7]: 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')
```

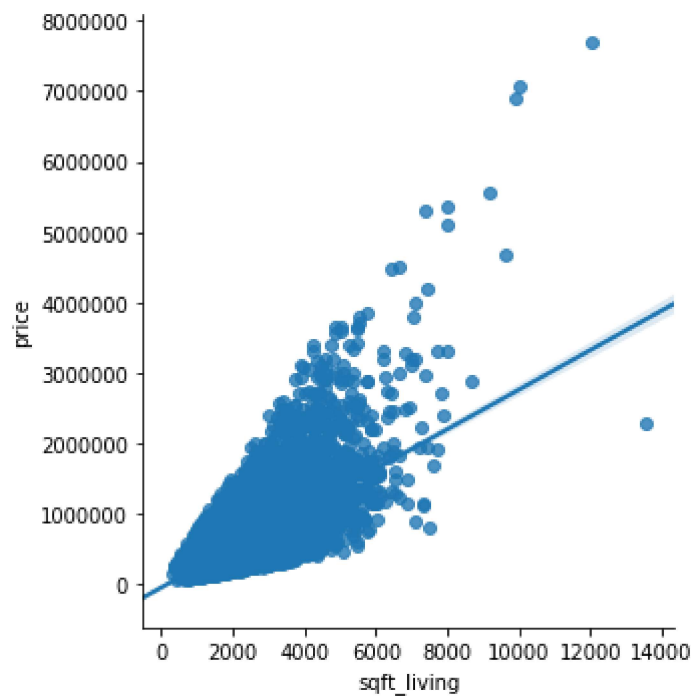
```
In [8]: plt.figure(figsize=(10,6))  
plt.scatter(house.sqft_living,house.price)  
plt.xlabel('sqft of house')  
plt.ylabel('price of house')
```

```
Out[8]: Text(0, 0.5, 'price of house')
```



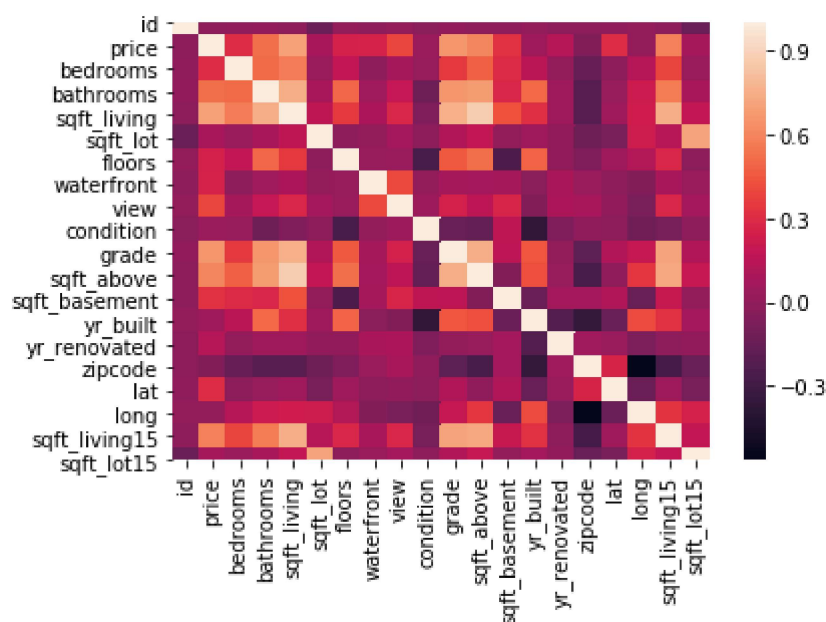
```
In [9]: sns.lmplot('sqft_living', 'price', data=house)
```

```
Out[9]: <seaborn.axisgrid.FacetGrid at 0x1c9a11a24c8>
```



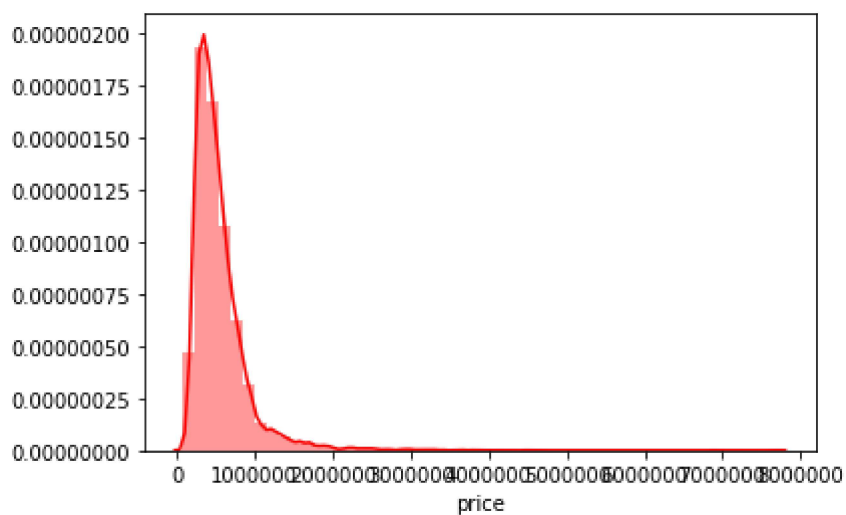
```
In [11]: sns.heatmap(house.corr())
```

```
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x1c9a1077bc8>
```



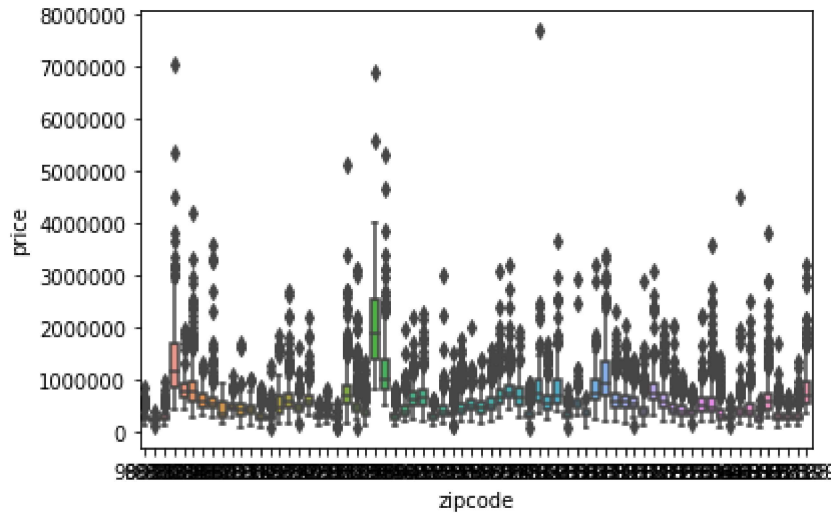
```
In [13]: sns.distplot(house['price'],color='red')
```

```
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x1c9a1b1d988>
```



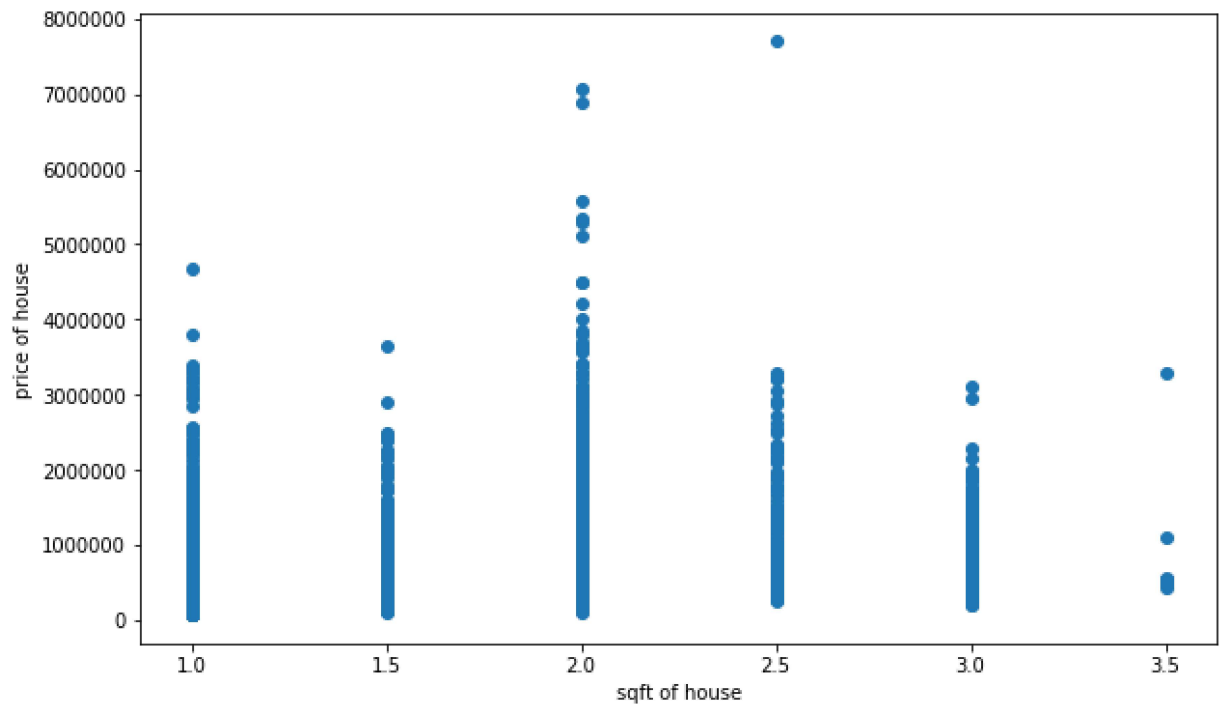
```
In [14]: sns.boxplot(x='zipcode',y='price',data=house)
```

```
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x1c9a1bfc5c8>
```



```
In [15]: plt.figure(figsize=(10,6))
plt.scatter(house.floors,house.price)
plt.xlabel('sqft of house')
plt.ylabel('price of house')
```

```
Out[15]: Text(0, 0.5, 'price of house')
```



```
In [ ]:
```