In [3]:
```
import os
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [3]:
```
os. getcwd()
```

Out[3]:
```
'C:\\Users\\hangl\\Downloads'
```

In [4]:
```
dt = pd.read_csv('C:\\Users\\hangl\\Downloads\House Prices dataset.csv')
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4600 entries, 0 to 4599
Data columns (total 18 columns):
date             4600 non-null object
price            4600 non-null float64
bedrooms         4600 non-null int64
bathrooms        4600 non-null float64
sqft_living      4600 non-null int64
sqft_lot         4600 non-null int64
floors           4600 non-null float64
waterfront       4600 non-null int64
view             4600 non-null int64
condition        4600 non-null int64
sqft_above       4600 non-null int64
sqft_basement    4600 non-null int64
yr_built         4600 non-null int64
yr_renovated     4600 non-null int64
street           4600 non-null object
city             4600 non-null object
statezip         4600 non-null object
country          4600 non-null object
dtypes: float64(3), int64(10), object(5)
memory usage: 647.0+ KB
```

In [6]:
```
dt.head()
```

Out[6]:

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_l
0	5/2/2014 0:00	313000.0	3	1.50	1340	7912	1.5	0	0	3	1340	0	1
1	5/2/2014 0:00	2384000.0	5	2.50	3650	9050	2.0	0	4	5	3370	280	1
2	5/2/2014 0:00	342000.0	3	2.00	1930	11947	1.0	0	0	4	1930	0	1
3	5/2/2014 0:00	420000.0	3	2.25	2000	8030	1.0	0	0	4	1000	1000	1
4	5/2/2014 0:00	550000.0	4	2.50	1940	10500	1.0	0	0	4	1140	800	1

In [278]:
```
dt.shape
```

Out[278]: (4600, 18)

In [279]:
```
# Checking missing values
null_cols = []
for col in dt.columns:
    if dt[col].isnull().sum() > 0 :
        print("Column",col, "has", dt[col].isnull().sum(),"null values")
    else:
        print("None")
```

```
None
None
None
None
None
None
None
None
None
None
None
None
None
None
None
None
None
None
None
None
None
```

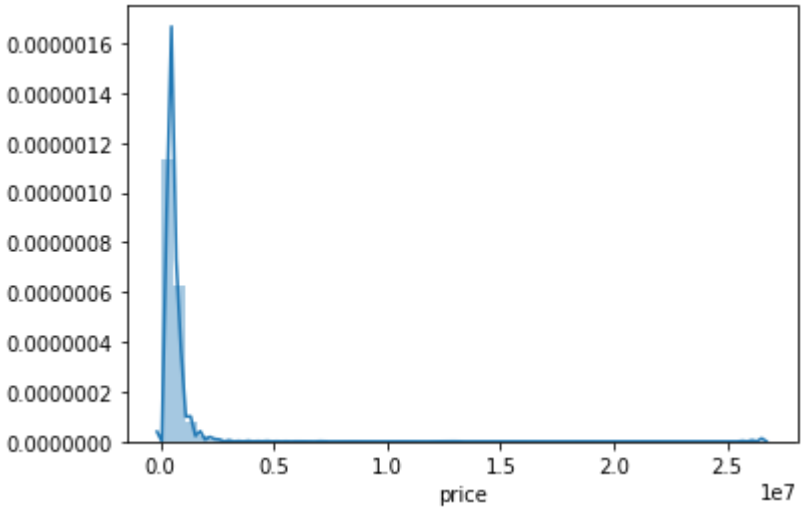
In [280]:
```
#describe of data set
dt.describe()
```

Out[280]:

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft
count	4.600000e+03	4600.000000	4600.000000	4600.000000	4.600000e+03	4600.000000	4600.000000	4600.000000	4600.000000	4600
mean	5.519630e+05	3.400870	2.160815	2139.346957	1.485252e+04	1.512065	0.007174	0.240652	3.451739	1827
std	5.638347e+05	0.908848	0.783781	963.206916	3.588444e+04	0.538288	0.084404	0.778405	0.677230	862
min	0.000000e+00	0.000000	0.000000	370.000000	6.380000e+02	1.000000	0.000000	0.000000	1.000000	370
25%	3.228750e+05	3.000000	1.750000	1460.000000	5.000750e+03	1.000000	0.000000	0.000000	3.000000	1190
50%	4.609435e+05	3.000000	2.250000	1980.000000	7.683000e+03	1.500000	0.000000	0.000000	3.000000	1590
75%	6.549625e+05	4.000000	2.500000	2620.000000	1.100125e+04	2.000000	0.000000	0.000000	4.000000	2300
max	2.659000e+07	9.000000	8.000000	13540.000000	1.074218e+06	3.500000	1.000000	4.000000	5.000000	9410

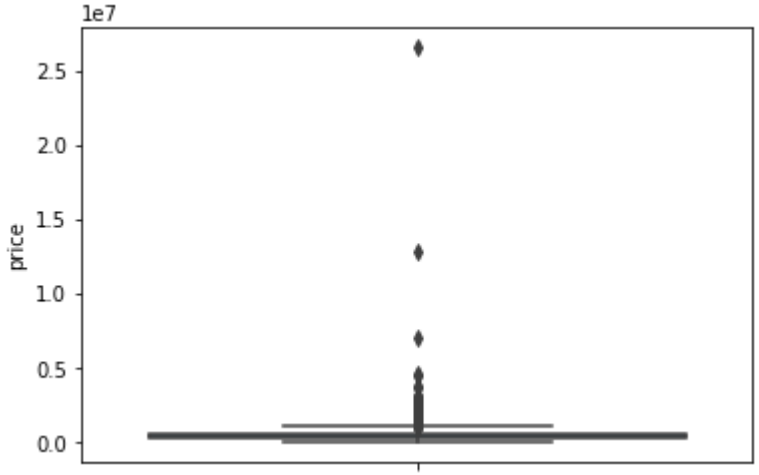
In [281]:
```
#histogram
sns.distplot(dt['price'])
```

Out[281]: <matplotlib.axes._subplots.AxesSubplot at 0x272443590c8>



In [282]:
```
#Box plot of price
sns.boxplot(y=dt["price"])
```

Out[282]: <matplotlib.axes._subplots.AxesSubplot at 0x2724429ad48>



In [283]:
```
dt['price'].value_counts()
```

Out[283]:
```
0.0          49
300000.0     42
400000.0     31
450000.0     29
440000.0     29
             ..
226500.0      1
257200.0      1
415500.0      1
1255000.0     1
256000.0      1
Name: price, Length: 1741, dtype: int64
```

In [284]:
```
dt.sort_values(by='price', ascending=False)
```

Out[284]:

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement
4350	7/3/2014 0:00	26590000.0	3	2.00	1180	7793	1.0	0	0	4	1180	0
4346	6/23/2014 0:00	12899000.0	3	2.50	2190	11394	1.0	0	0	3	1550	640
2286	6/11/2014 0:00	7062500.0	5	4.50	10040	37325	2.0	1	2	3	7680	2360
2654	6/17/2014 0:00	4668000.0	5	6.75	9640	13068	1.0	1	4	3	4820	4820
2761	6/18/2014 0:00	4489000.0	4	3.00	6430	27517	2.0	0	0	3	6430	0
...	...	...	...	...	...	...	...	...	...	...	...	...
4472	6/9/2014 0:00	0.0	4	3.75	4060	19290	2.0	0	0	3	4060	0
4567	7/2/2014 0:00	0.0	4	2.50	4080	18362	2.0	0	2	4	4080	0
4354	5/5/2014 0:00	0.0	3	1.75	1490	10125	1.0	0	0	4	1490	0
4454	6/3/2014 0:00	0.0	5	2.50	2090	4698	2.0	0	0	3	2090	0
4382	5/12/2014 0:00	0.0	5	4.50	4630	6324	2.0	0	0	3	3210	1420

4600 rows × 18 columns

In [7]:
```
# Droping house with price = 0
dt_new = dt[dt.price != 0]
```

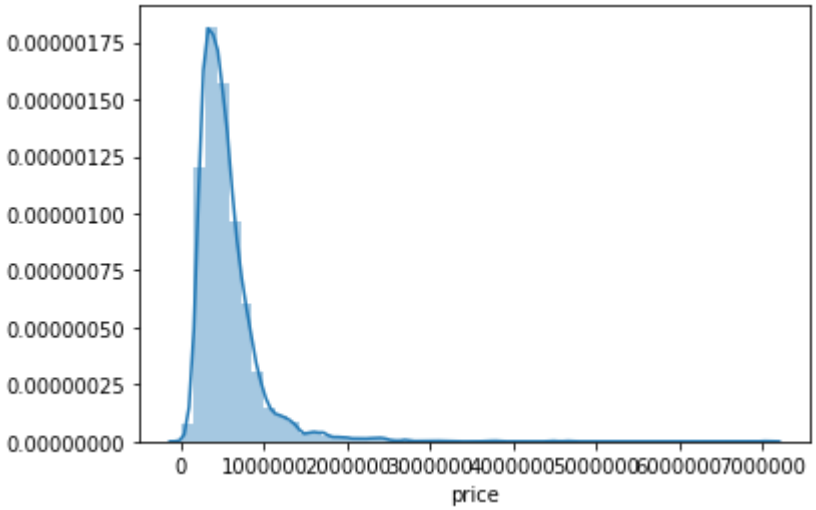
In [8]:
```
# Droping outliers
dt_new = dt_new[dt_new.price != 26590000]
dt_new = dt_new[dt_new.price != 12899000]
```

In [9]:
```
dt_new.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4549 entries, 0 to 4599
Data columns (total 18 columns):
date             4549 non-null object
price            4549 non-null float64
bedrooms         4549 non-null int64
bathrooms        4549 non-null float64
sqft_living      4549 non-null int64
sqft_lot         4549 non-null int64
floors           4549 non-null float64
waterfront       4549 non-null int64
view             4549 non-null int64
condition        4549 non-null int64
sqft_above       4549 non-null int64
sqft_basement    4549 non-null int64
yr_built         4549 non-null int64
yr_renovated     4549 non-null int64
street           4549 non-null object
city             4549 non-null object
statezip         4549 non-null object
country          4549 non-null object
dtypes: float64(3), int64(10), object(5)
memory usage: 675.2+ KB
```

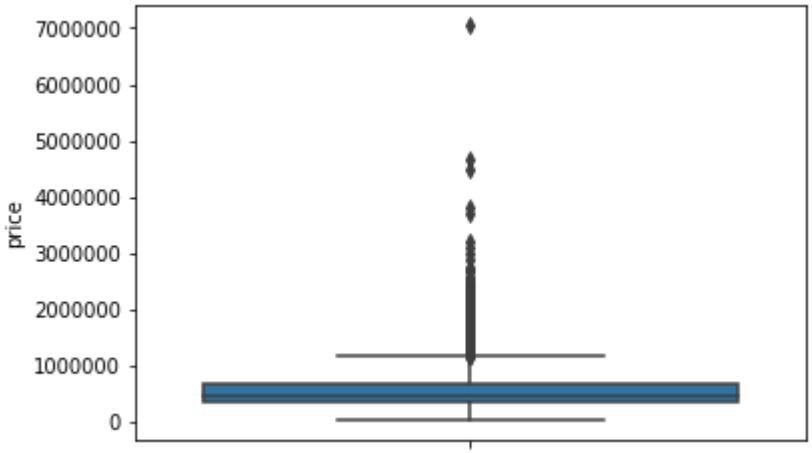
In [10]:
```
sns.distplot(dt_new['price'])
```

Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x20edfe150c8>



In [11]:
```
#Box plot of price
sns.boxplot(y=dt_new["price"])
```

Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x20ee16ee6c8>

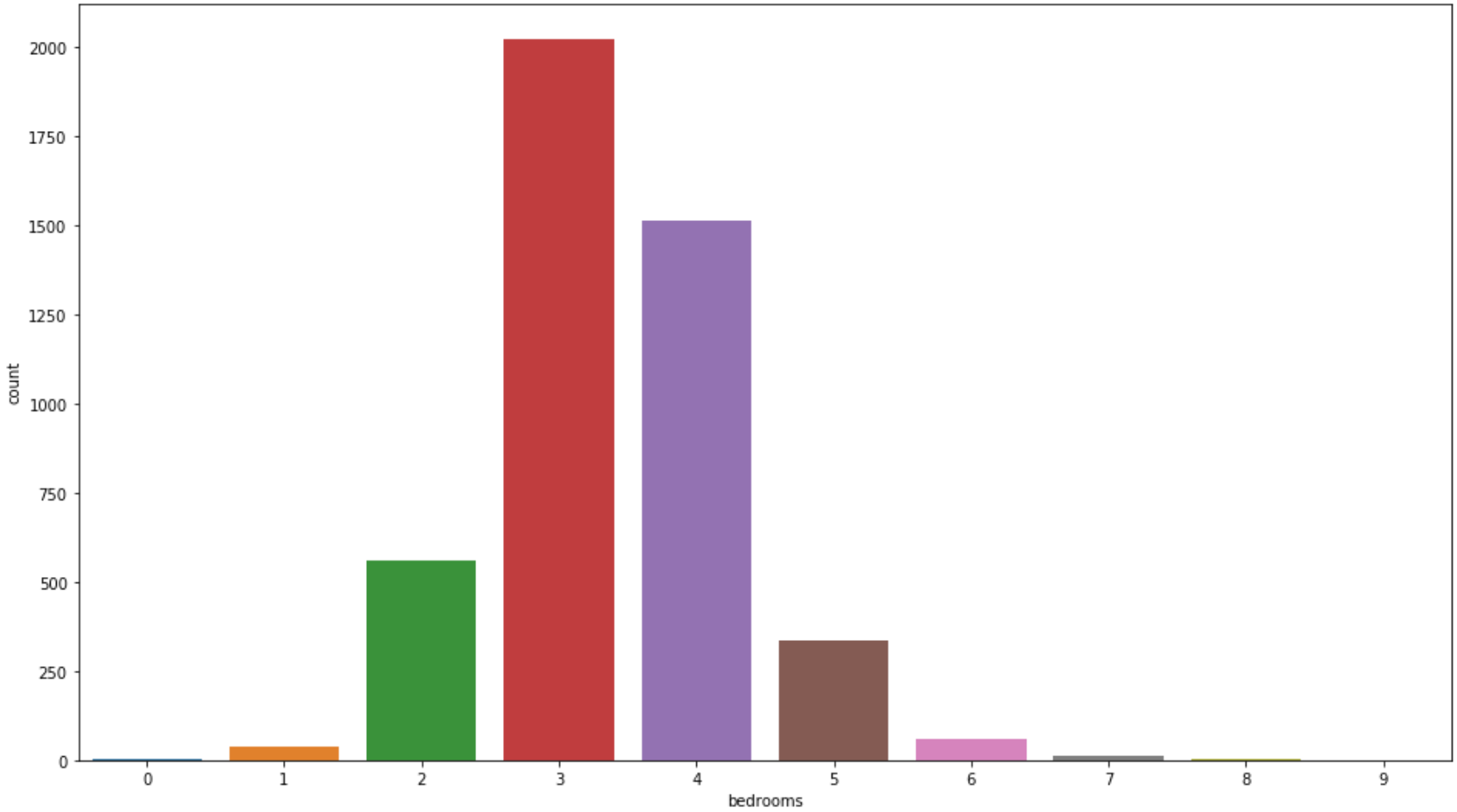


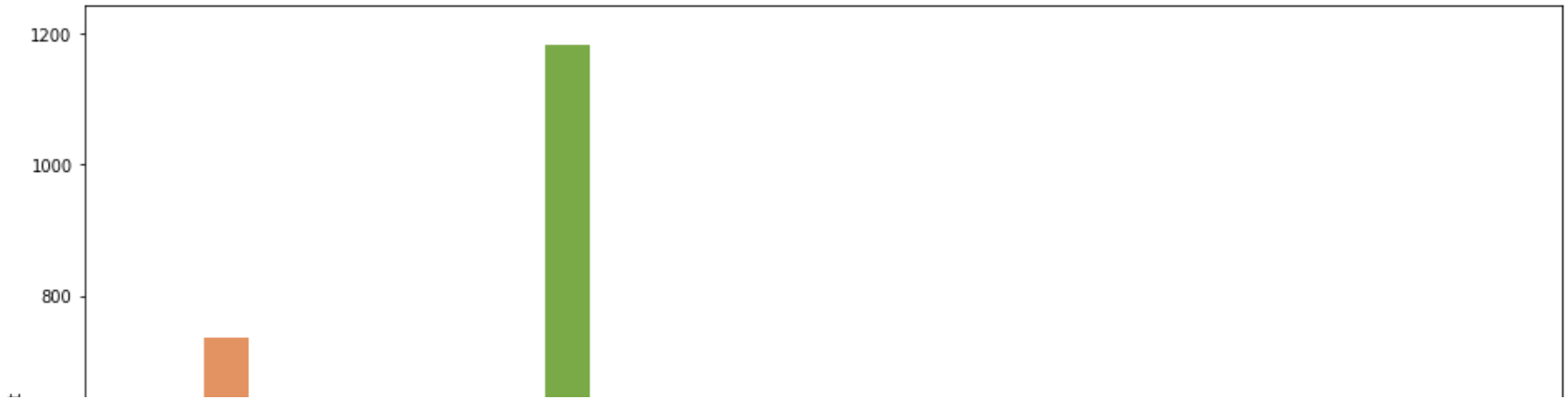
In [12]:
```
# lets check the number of observation in different catergorical variables
feature = ['bedrooms', 'bathrooms', 'floors', 'waterfront', 'view', 'condition']
for col in feature:
    print(dt_new[col].value_counts())
    print("\n")
```

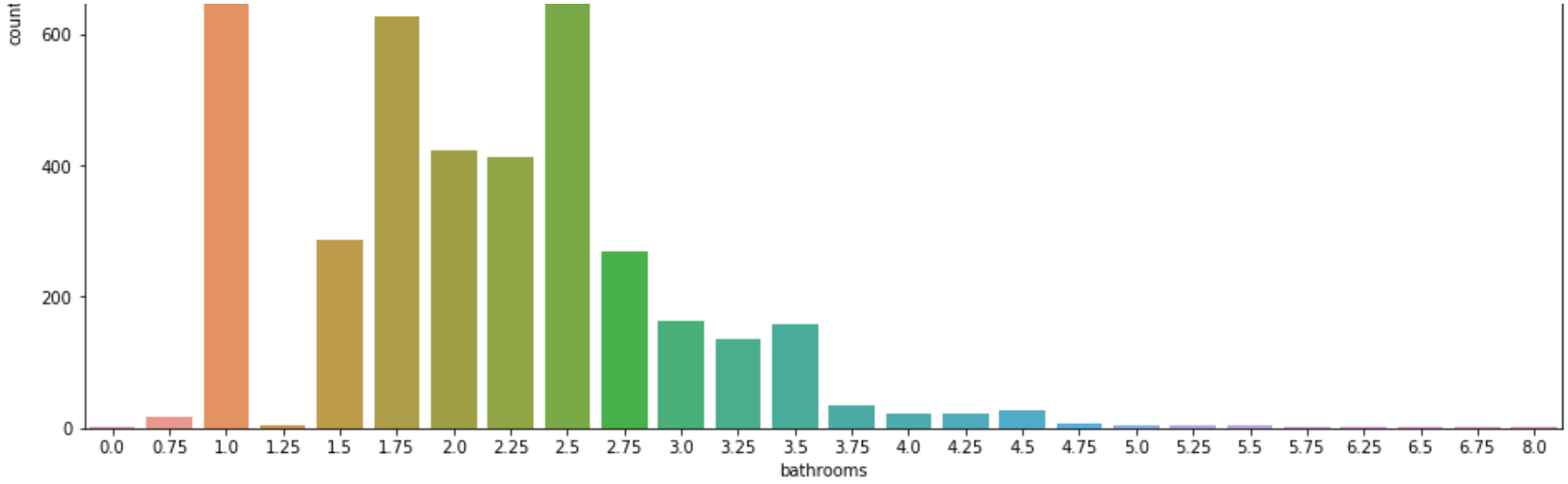
```
3    2023
4    1512
2     561
5     338
6      59
1      37
7      14
8       2
0       2
9       1
Name: bedrooms, dtype: int64


2.50    1183
1.00     736
1.75     628
2.00     424
2.25     413
1.50     287
2.75     270
3.00     164
3.50     159
3.25     135
3.75      34
4.50      26
4.25      22
4.00      21
0.75      17
4.75       7
5.00       5
5.50       4
5.25       4
1.25       3
0.00       2
6.25       1
8.00       1
6.50       1
5.75       1
6.75       1
Name: bathrooms, dtype: int64


1.0    2149
```

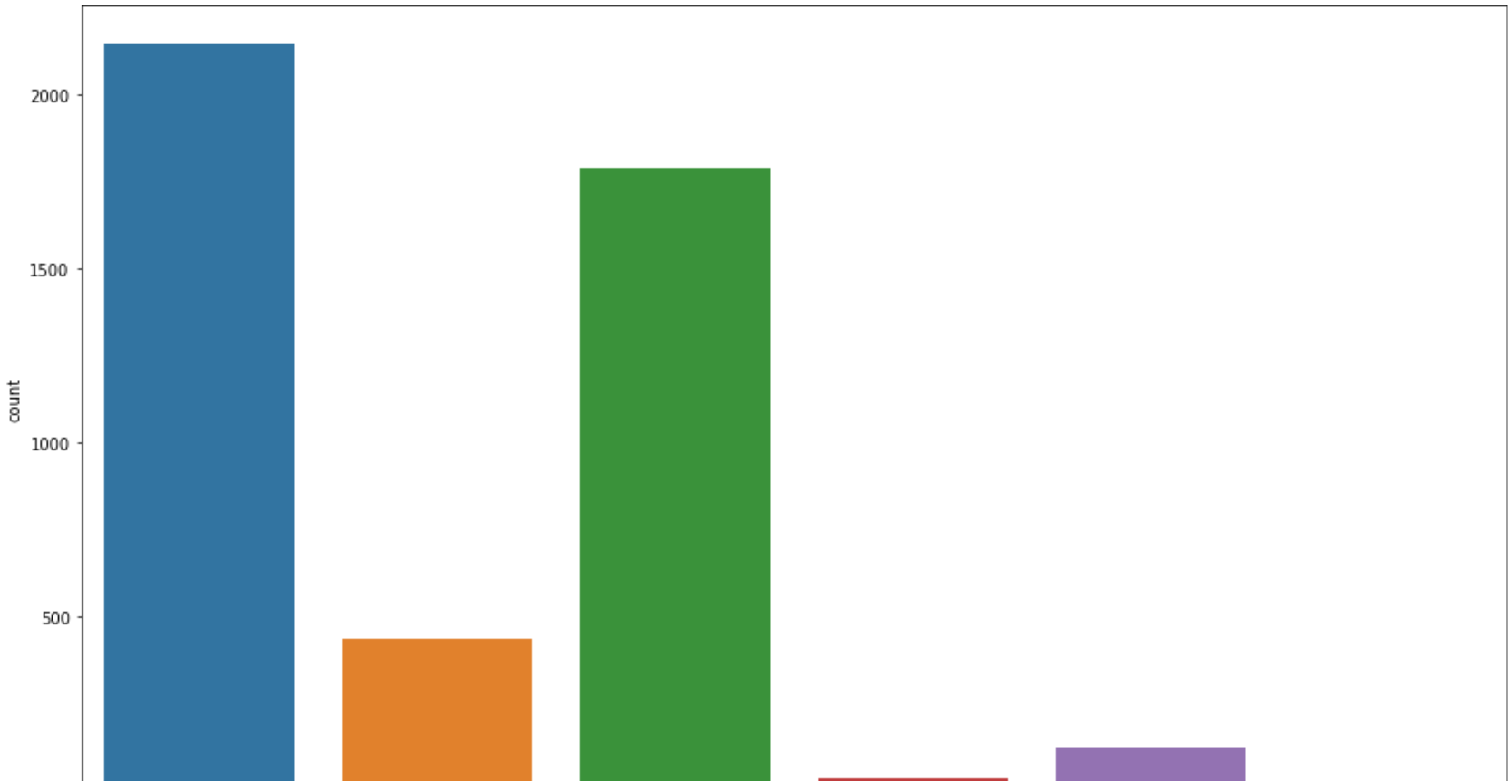
```
2.0    1791
1.5     439
3.0     127
2.5      41
3.5       2
Name: floors, dtype: int64


0    4519
1      30
Name: waterfront, dtype: int64


0    4101
2     201
3     115
1      69
4      63
Name: view, dtype: int64


3    2850
4    1237
5     425
2      31
1       6
Name: condition, dtype: int64
```

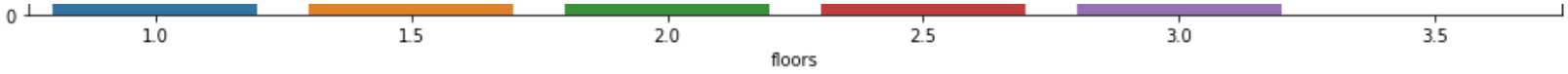
In [13]:
```
# Visualize the count of obserations in each categorical variable
feature = ['bedrooms', 'bathrooms', 'floors', 'waterfront', 'view', 'condition']

plt.figure(figsize = (16,96))
for idx,col in enumerate(feature):
    plt.subplot(9,1,idx+1)
    ax=sns.countplot(dt_new[col])
```

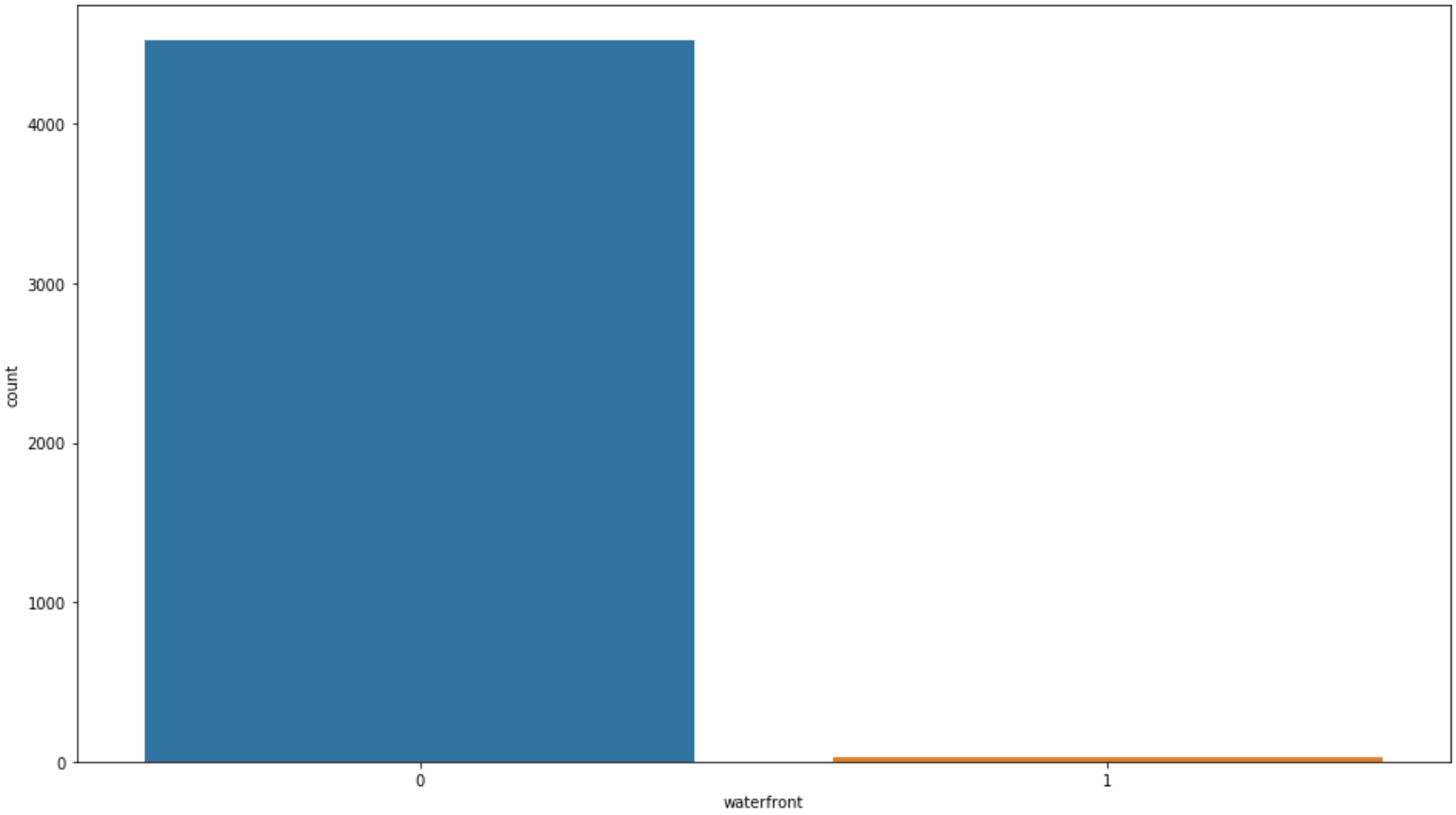

2000
1750
1500
1250
1000
750
500
250
0
count
0
1
2
3
4
5
6
7
8
9
bedrooms


1200
1000
800

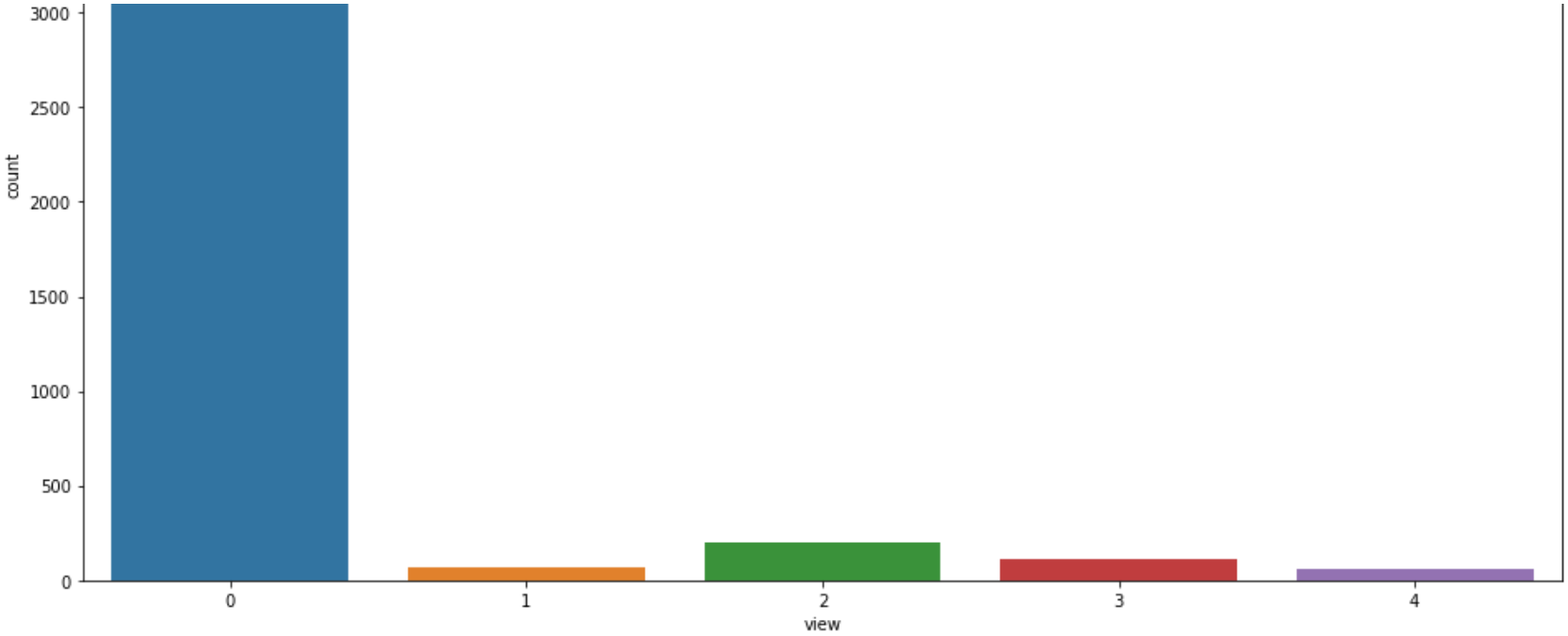

count
600
400
200
0
0.0
0.75
1.0
1.25
1.5
1.75
2.0
2.25
2.5
2.75
3.0
3.25
3.5
3.75
4.0
4.25
4.5
4.75
5.0
5.25
5.5
5.75
6.25
6.5
6.75
8.0
bathrooms

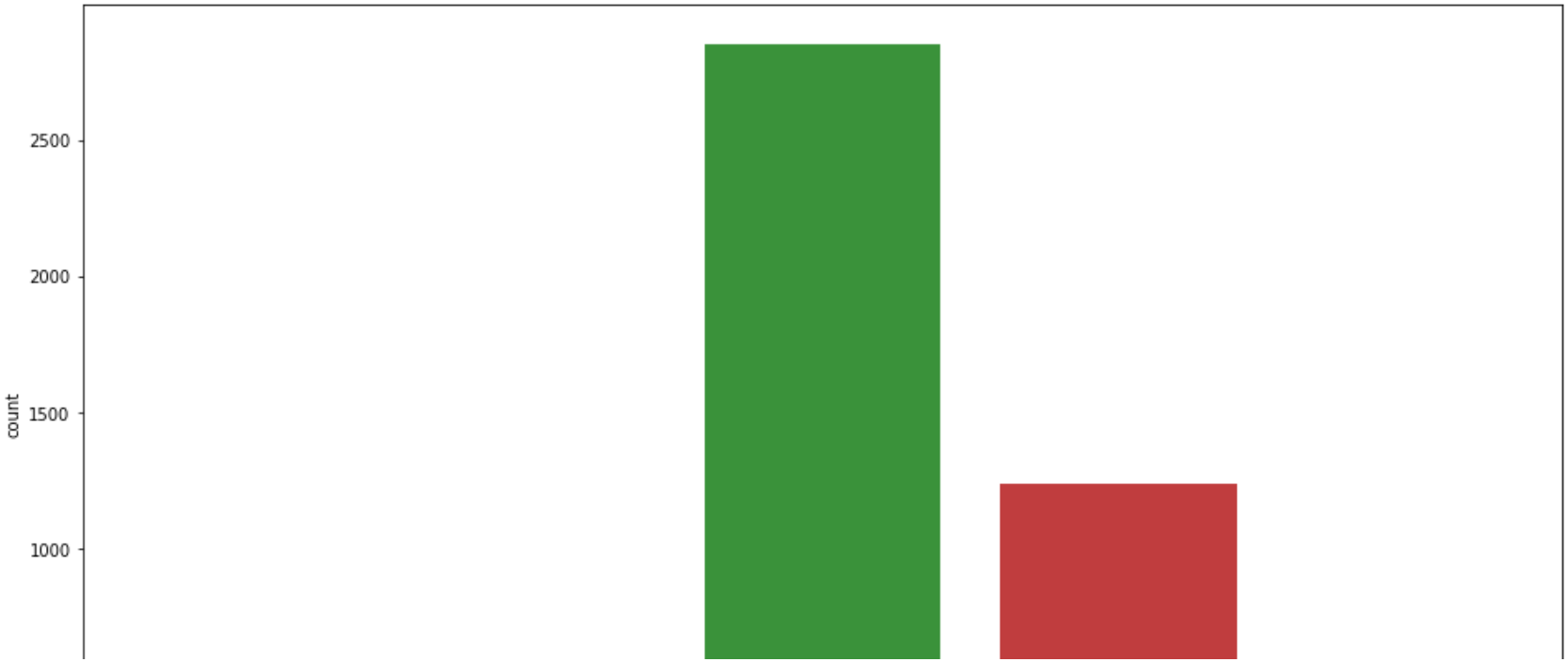

2000
1500
1000
500
count

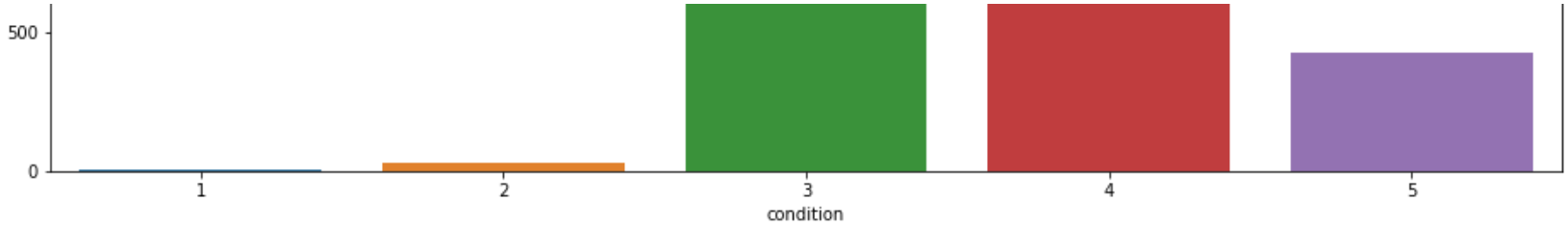

0
1.0
1.5
2.0
2.5
3.0
3.5
floors


4000
3000
2000
1000
0
count
0
1
waterfront


4000
3500


3000
2500
2000
1500
1000
500
0
count
0
1
2
3
4
view


2500
2000
1500
1000
count



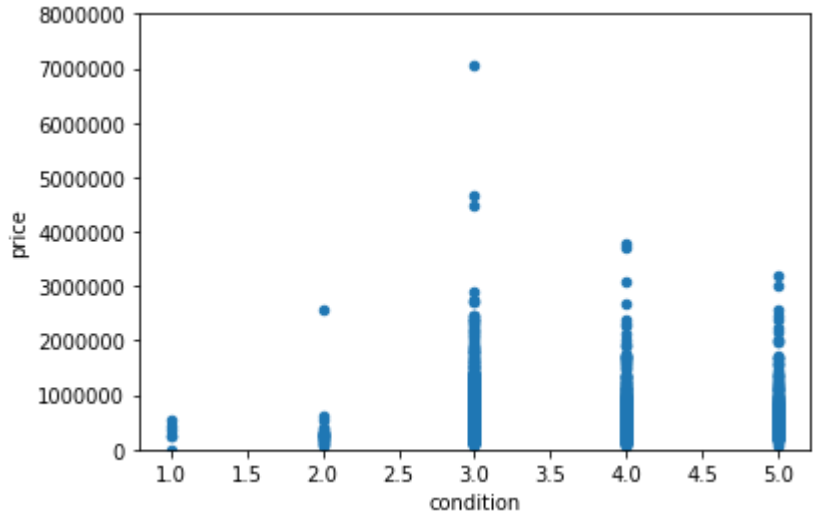
In [29]:
```
dt_new = dt_new[dt_new.bedrooms != 0]
```

In [30]:
```
dt_new.info()
```

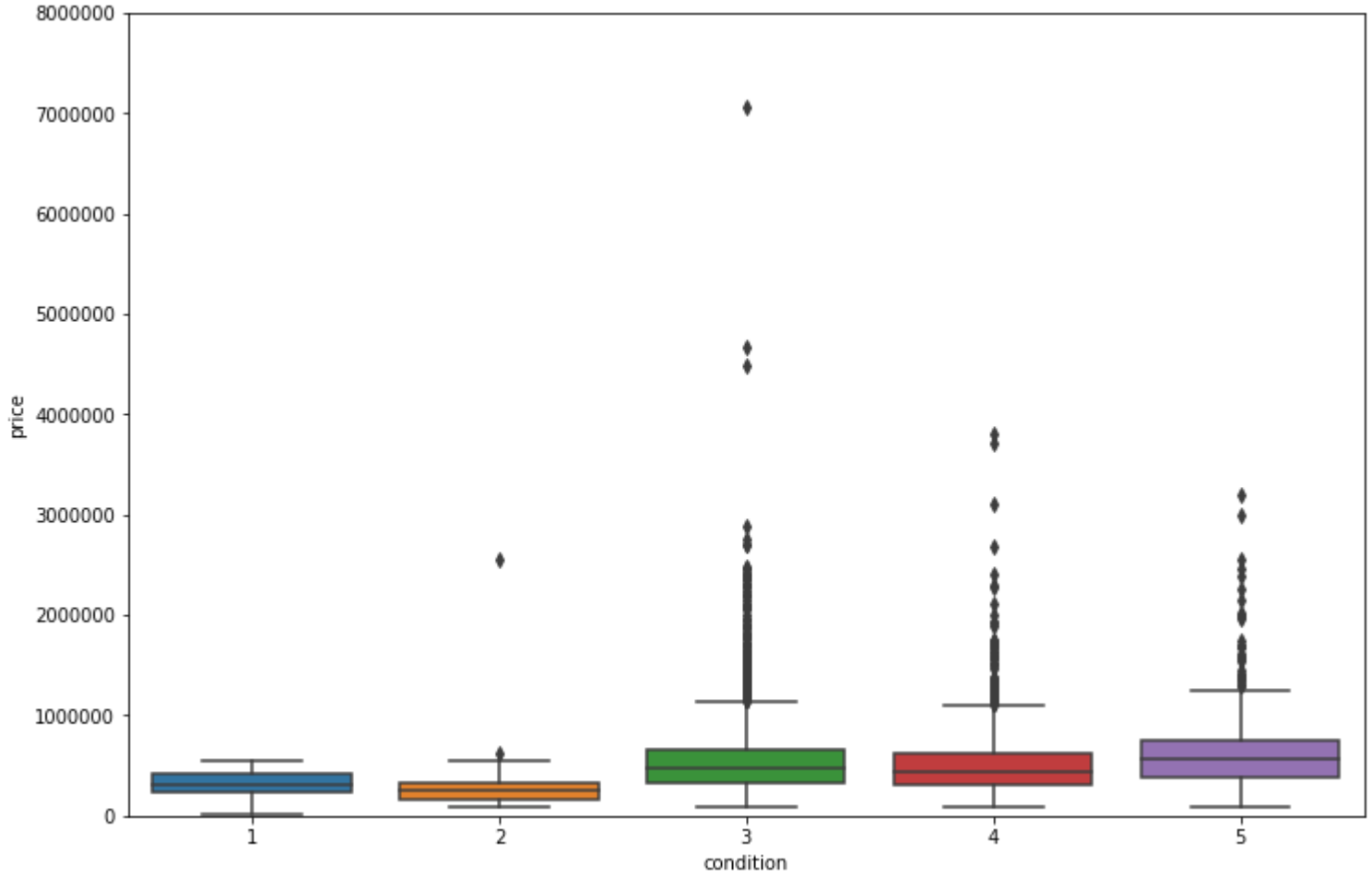
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4547 entries, 0 to 4599
Data columns (total 18 columns):
date             4547 non-null object
price            4547 non-null float64
bedrooms         4547 non-null int64
bathrooms        4547 non-null float64
sqft_living      4547 non-null int64
sqft_lot         4547 non-null int64
floors           4547 non-null float64
waterfront       4547 non-null int64
view             4547 non-null int64
condition        4547 non-null int64
sqft_above       4547 non-null int64
sqft_basement    4547 non-null int64
yr_built         4547 non-null int64
yr_renovated     4547 non-null int64
street           4547 non-null object
city             4547 non-null object
statezip         4547 non-null object
country          4547 non-null object
dtypes: float64(3), int64(10), object(5)
memory usage: 674.9+ KB
```

In [31]:
```
# scatter plot condition and sale price
dt_new.plot.scatter(x='condition', y='price', ylim=(0,8000000))
```

Out[31]: <matplotlib.axes._subplots.AxesSubplot at 0x20ee26ecb48>

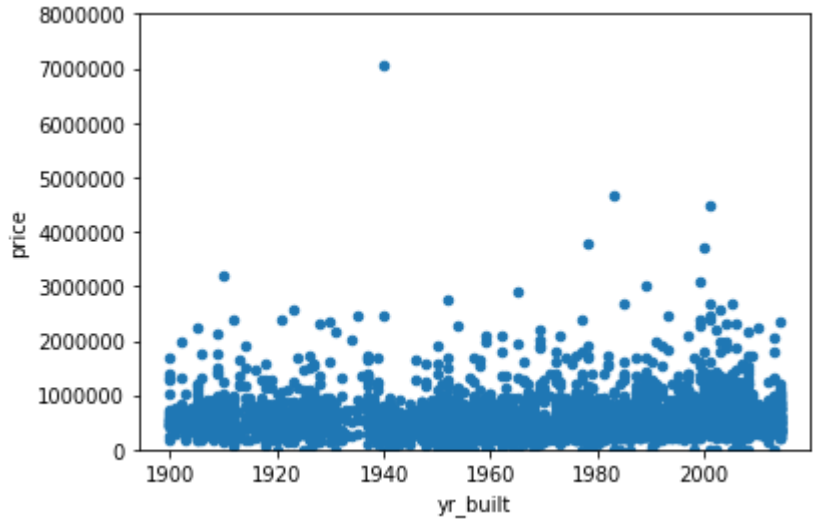


In [15]:
```
# Price and condition
var = 'condition'
f, ax = plt.subplots(figsize=(12, 8))
fig = sns.boxplot(x=var, y="price", data=dt_new)
fig.axis(ymin=0, ymax=8000000);
```



In [32]:
```
#Year built and sale price
dt.plot.scatter(x='yr_built', y='price', ylim=(0,8000000))
```

Out[32]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x20ee27a43c8>
```

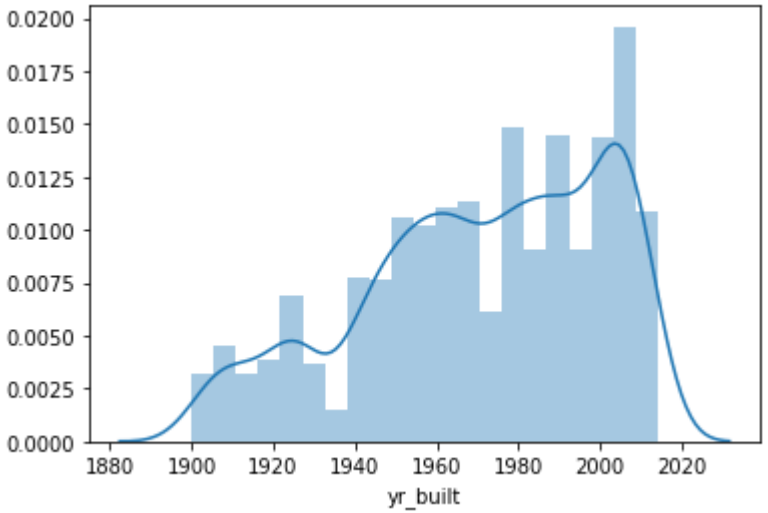


In [33]:
```
#Top 15 Age that have most observations
dt_new['yr_built'].value_counts().sort_values(ascending = False).head(15)
```

Out[33]:
```
2006    109
2005    103
2004     92
2007     92
1978     90
2003     89
2008     88
1967     82
1977     79
2014     78
1968     76
1987     74
1989     72
1959     67
1990     66
Name: yr_built, dtype: int64
```

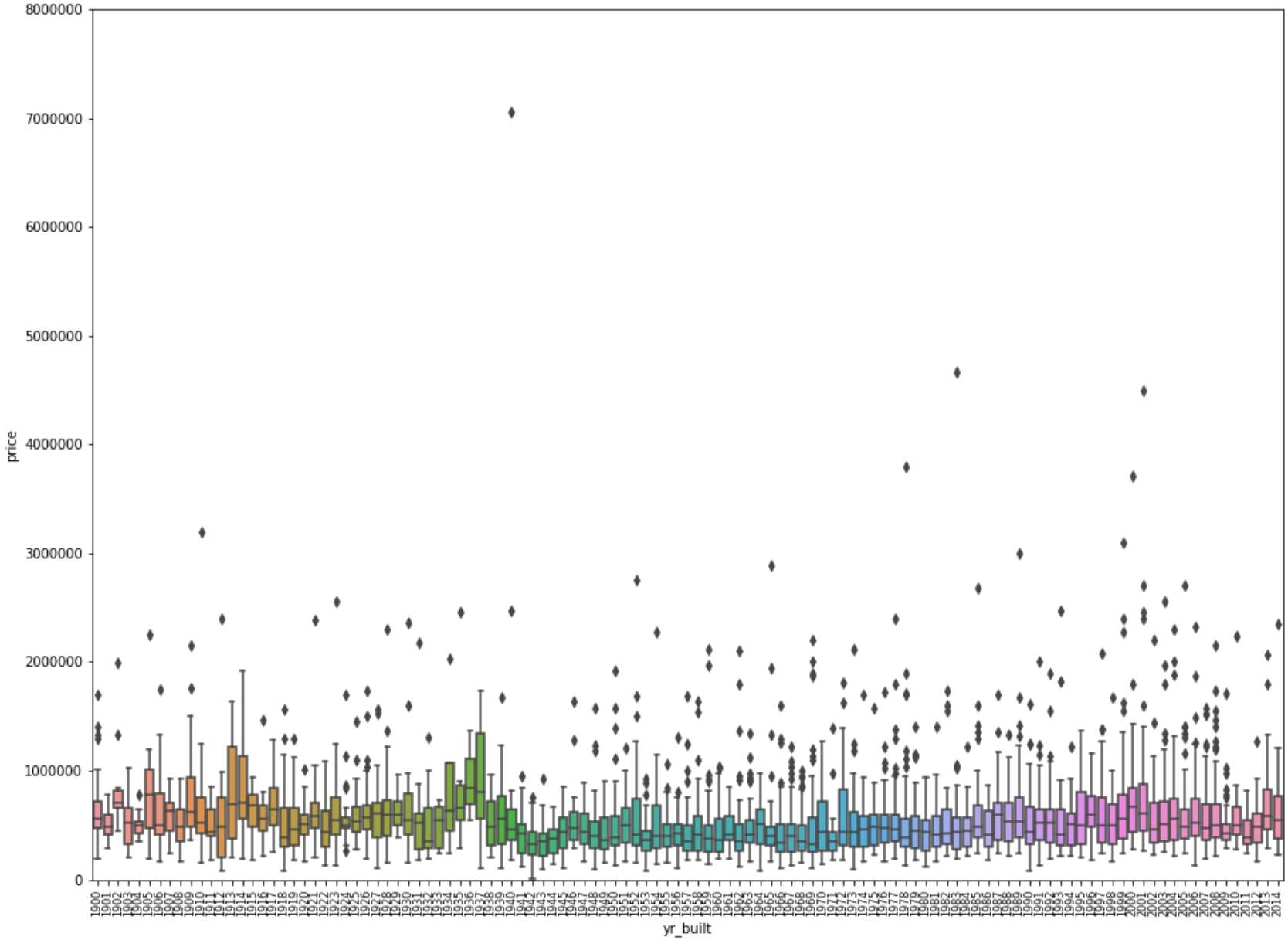
In [34]:
```
#histogram
sns.distplot(dt_new['yr_built'])
```

Out[34]: <matplotlib.axes._subplots.AxesSubplot at 0x20ee2ed3a08>



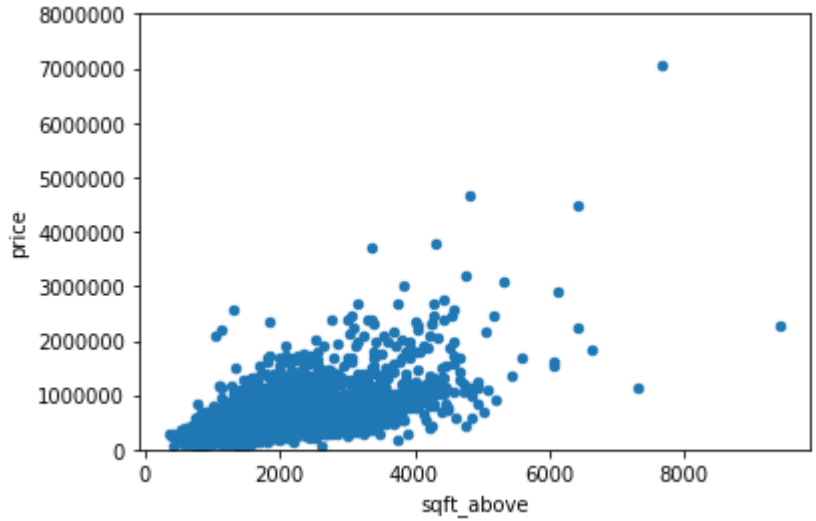
In [35]:
```
# price and year built
f, ax = plt.subplots(figsize=(16, 12))
fig = sns.boxplot(x='yr_built', y="price", data=dt_new)
fig.axis(ymin=0, ymax=8000000)
plt.xticks(rotation=90, fontsize = 8)
```

```
Out[35]: (array([  0,   1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,
                  13,  14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,  25,
                  26,  27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,  38,
                  39,  40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  51,
                  52,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,
                  65,  66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,
                  78,  79,  80,  81,  82,  83,  84,  85,  86,  87,  88,  89,  90,
                  91,  92,  93,  94,  95,  96,  97,  98,  99, 100, 101, 102, 103,
                 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114]),
          <a list of 115 Text xticklabel objects>)
```


price
8000000
7000000
6000000
5000000
4000000
3000000
2000000
1000000
0
yr_built

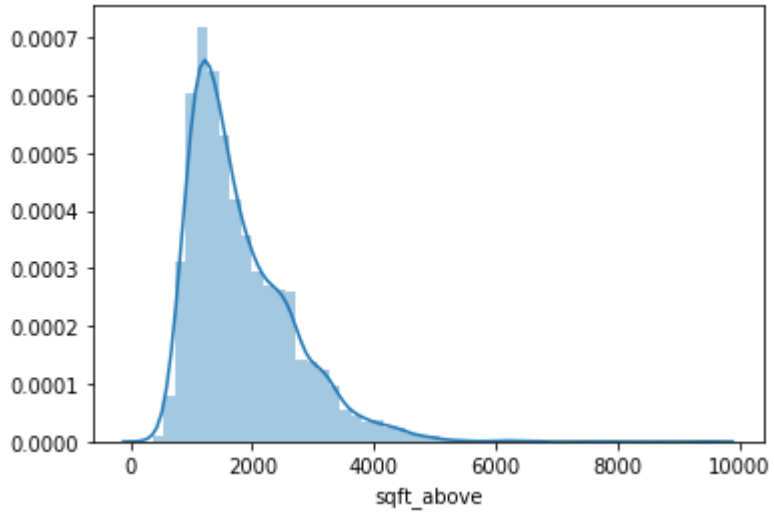
In [36]:
```
dt_new.plot.scatter(x='sqft_above', y='price', ylim=(0,8000000))
```

Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x20ee492f888>



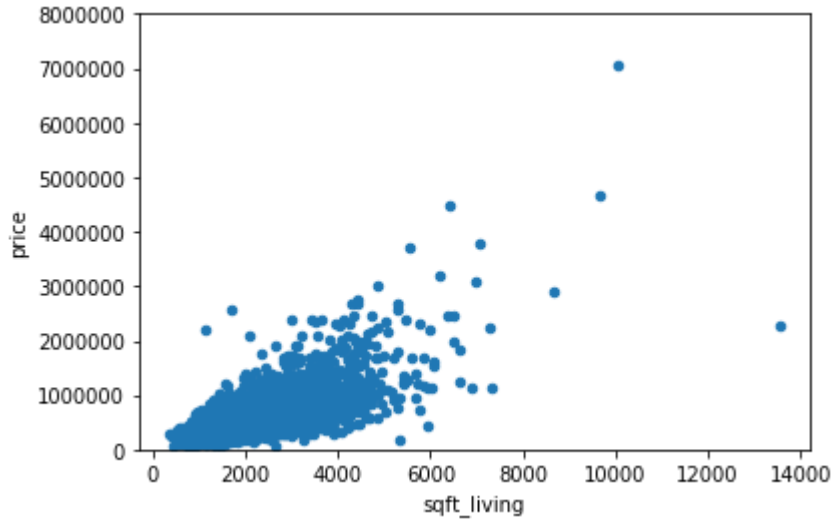
In [37]:
```
sns.distplot(dt_new['sqft_above'])
```

Out[37]: <matplotlib.axes._subplots.AxesSubplot at 0x20ee4e5dec8>



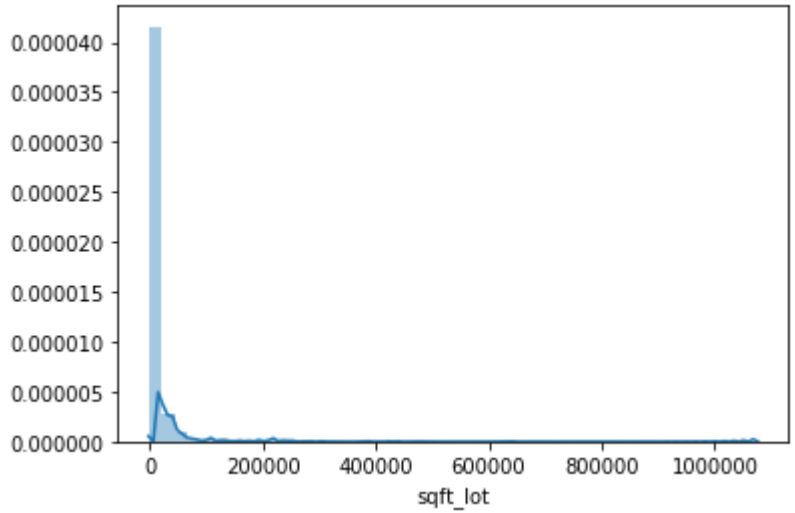
In [38]:
```
dt_new.plot.scatter(x='sqft_living', y='price', ylim=(0,8000000))
```

Out[38]: <matplotlib.axes._subplots.AxesSubplot at 0x20ee4a0a908>



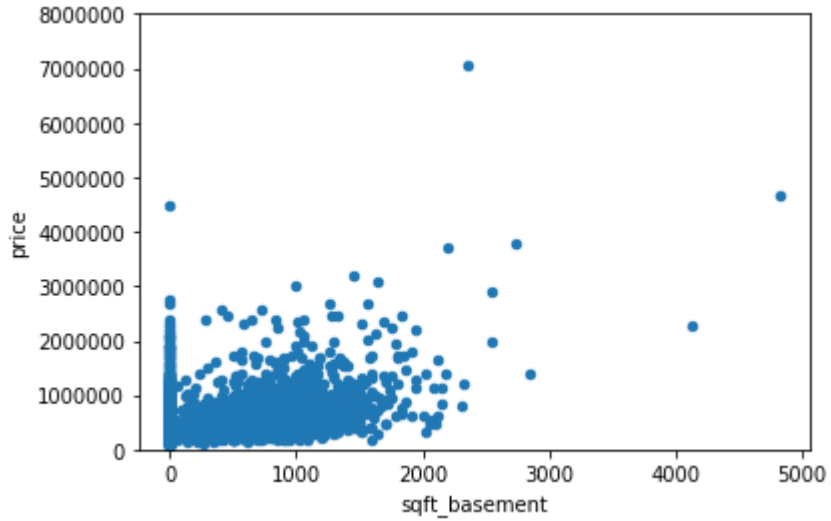
In [53]:
```
sns.distplot(dt_new['sqft_lot'])
```

Out[53]: <matplotlib.axes._subplots.AxesSubplot at 0x20ee4d0f148>



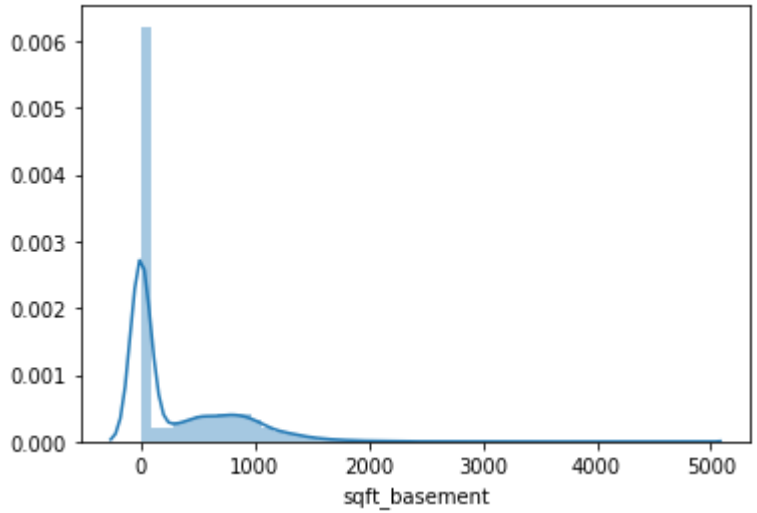
In [40]:
```
dt_new.plot.scatter(x='sqft_basement', y='price', ylim=(0,8000000))
```

Out[40]: <matplotlib.axes._subplots.AxesSubplot at 0x20ee264ee48>

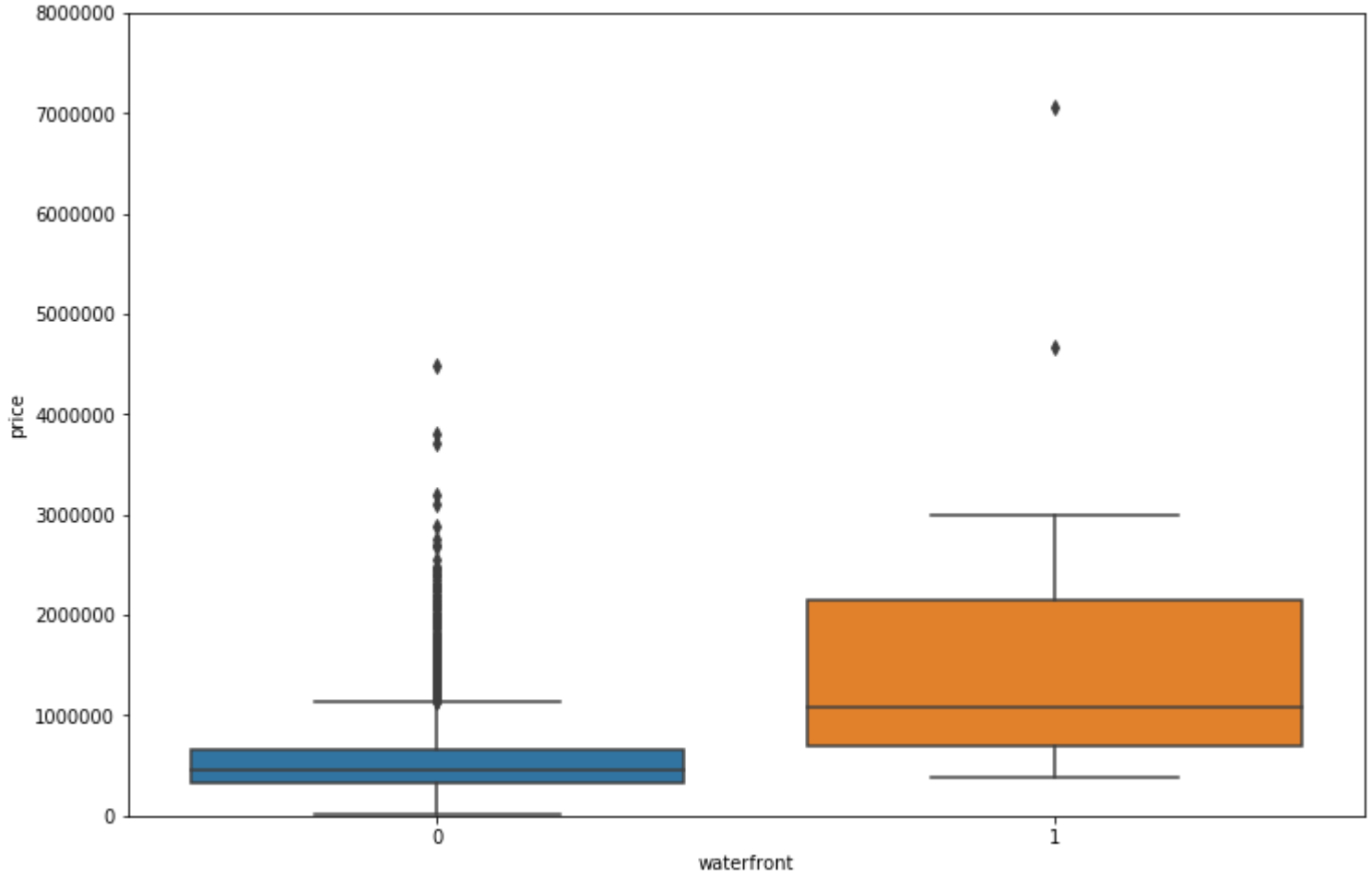


In [41]:
```
sns.distplot(dt_new['sqft_basement'])
```

Out[41]: <matplotlib.axes._subplots.AxesSubplot at 0x20ee26a7708>

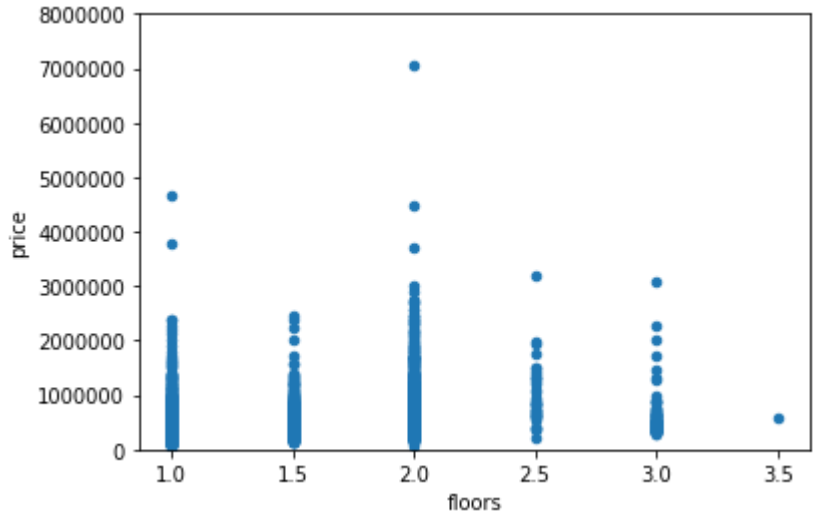


In [42]:
```
# Price and waterfront
var = 'waterfront'
f, ax = plt.subplots(figsize=(12, 8))
fig = sns.boxplot(x=var, y="price", data=dt_new)
fig.axis(ymin=0, ymax=8000000);
```

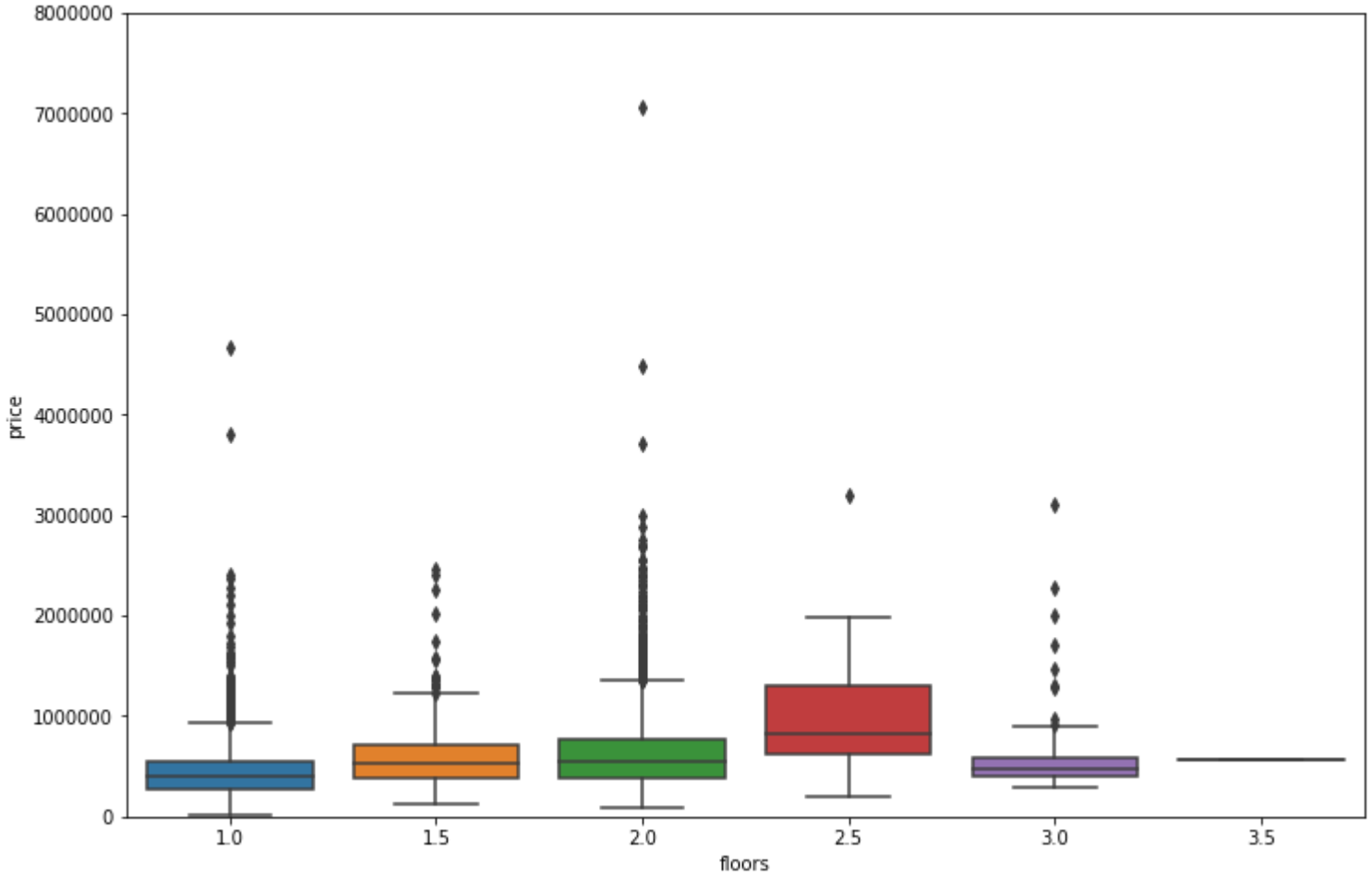


In [43]:
```
dt_new.plot.scatter(x='floors', y='price', ylim=(0,8000000))
```

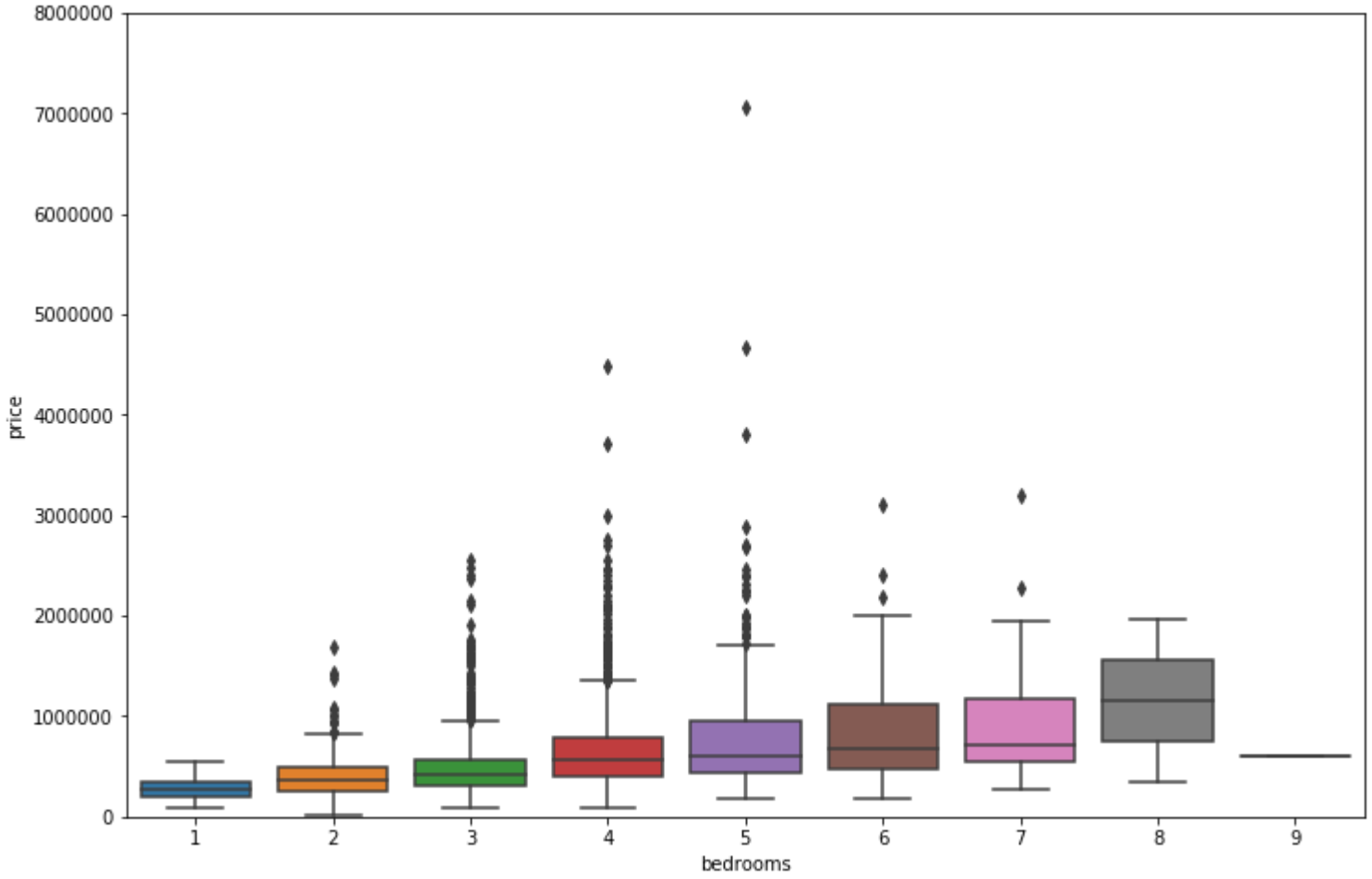
Out[43]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x20ee4c96888>
```



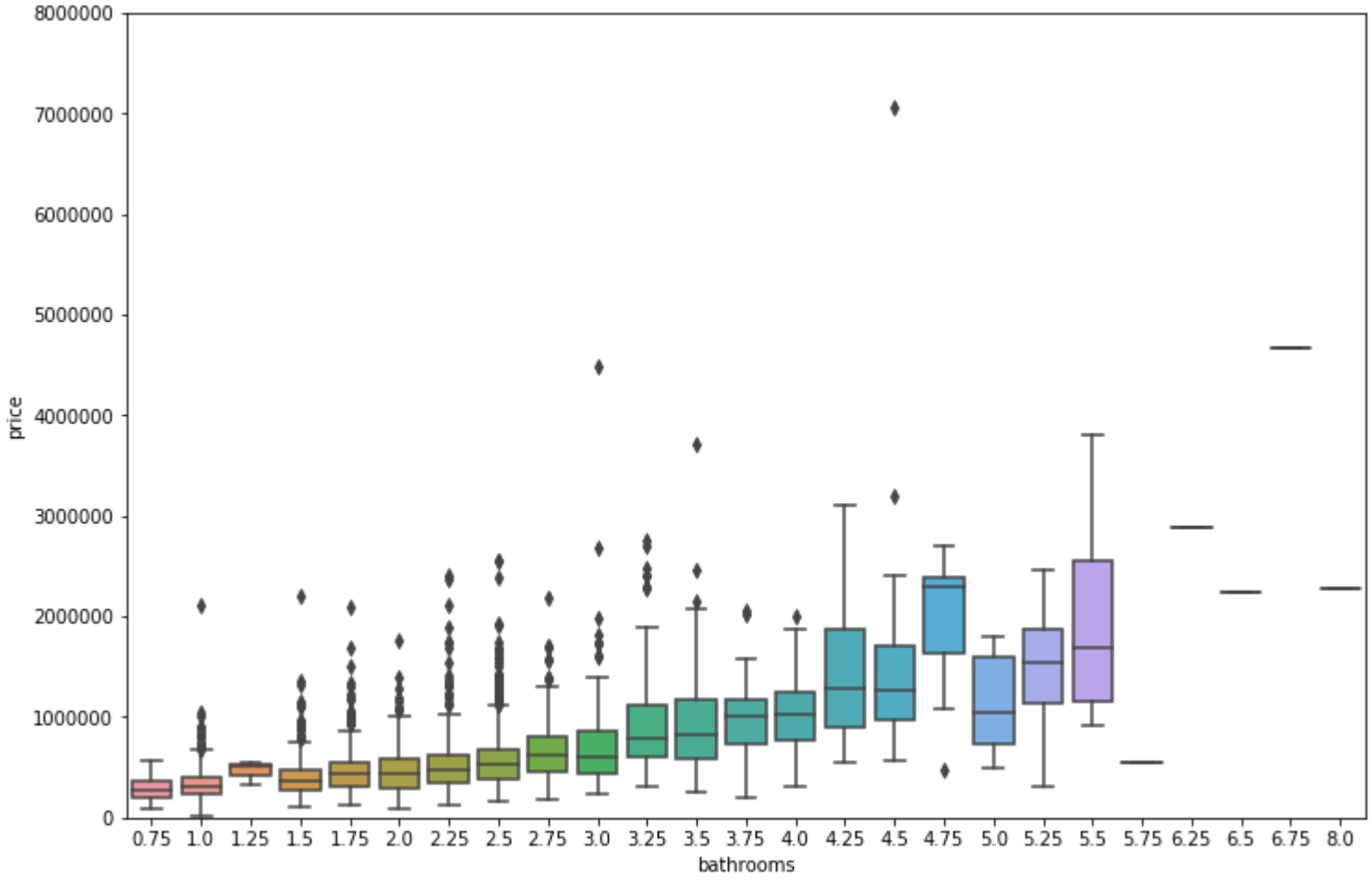
In [44]:
```
# Price and waterfront
var = 'floors'
f, ax = plt.subplots(figsize=(12, 8))
fig = sns.boxplot(x=var, y="price", data=dt_new)
fig.axis(ymin=0, ymax=8000000);
```



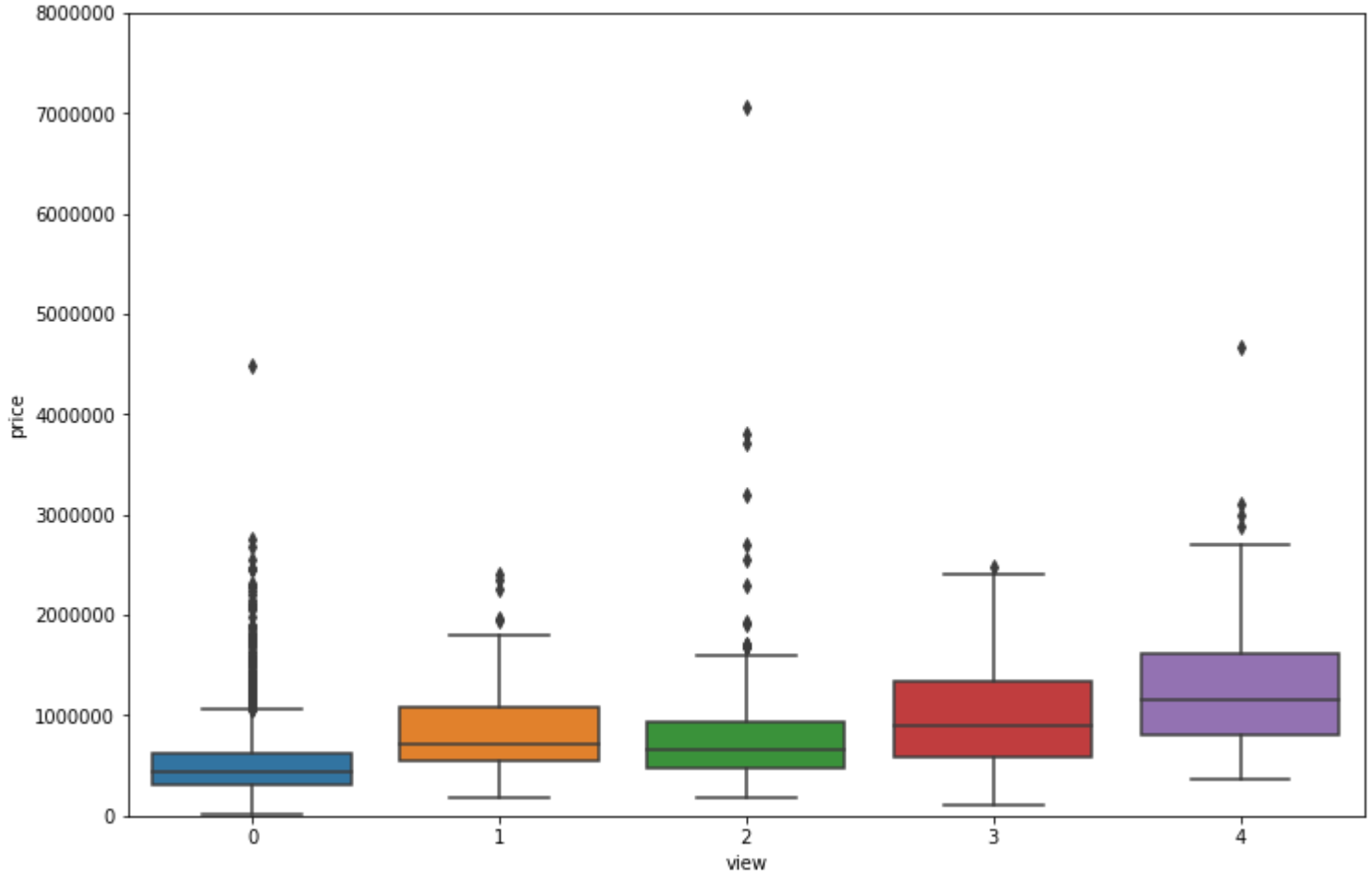
In [45]:
```
# Price and bedrooms
var = 'bedrooms'
f, ax = plt.subplots(figsize=(12, 8))
fig = sns.boxplot(x=var, y="price", data=dt_new)
fig.axis(ymin=0, ymax=8000000);
```



In [46]:
```
# Price and bathrooms
var = 'bathrooms'
f, ax = plt.subplots(figsize=(12, 8))
fig = sns.boxplot(x=var, y="price", data=dt_new)
fig.axis(ymin=0, ymax=8000000);
```

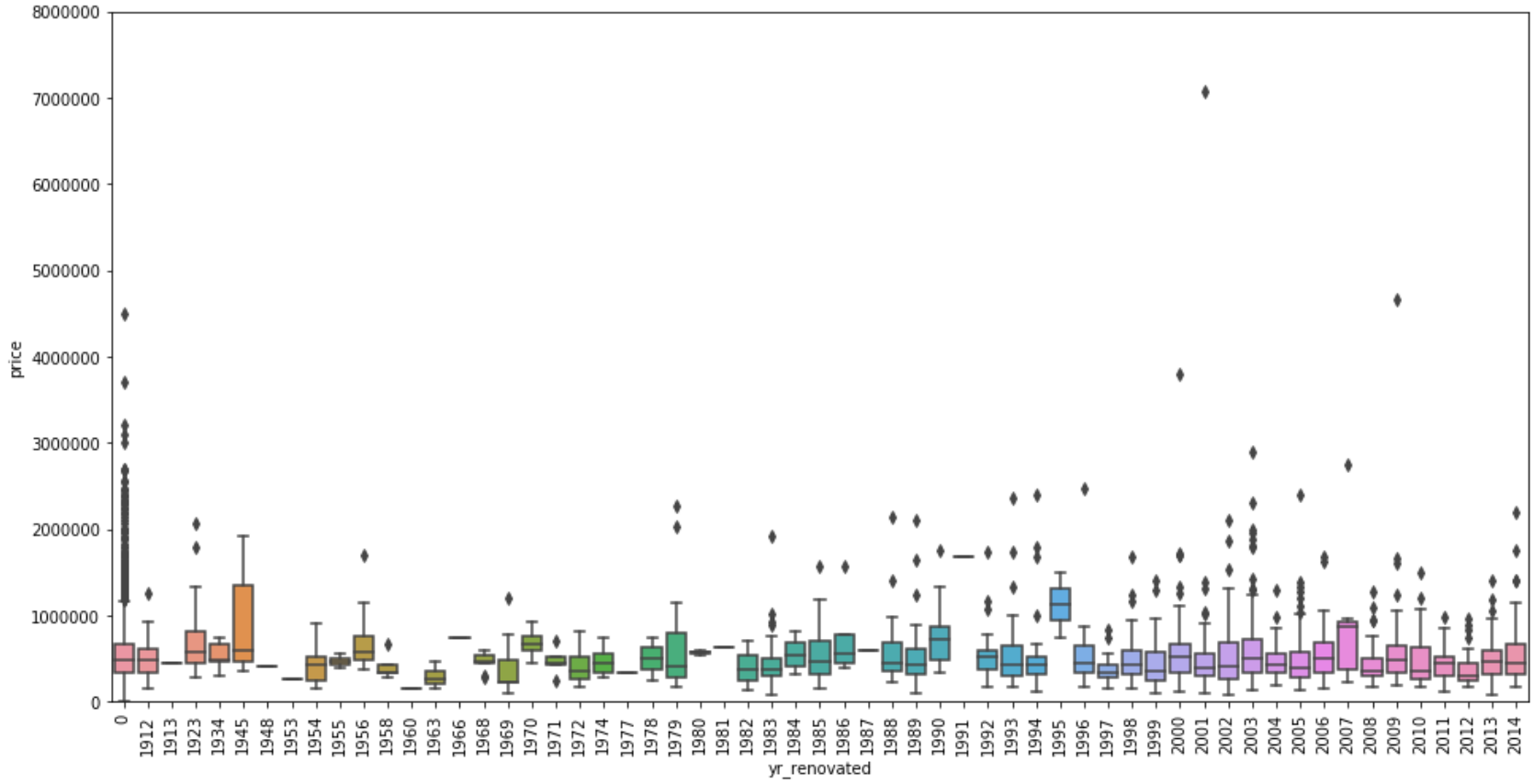


In [47]:
```
# Price and floors
var = 'view'
f, ax = plt.subplots(figsize=(12, 8))
fig = sns.boxplot(x=var, y="price", data=dt_new)
fig.axis(ymin=0, ymax=8000000);
```



In [48]:
```
# price and year renovated
f, ax = plt.subplots(figsize=(16, 8))
fig = sns.boxplot(x='yr_renovated', y="price", data=dt_new)
fig.axis(ymin=0, ymax=8000000)
plt.xticks(rotation=90)
```

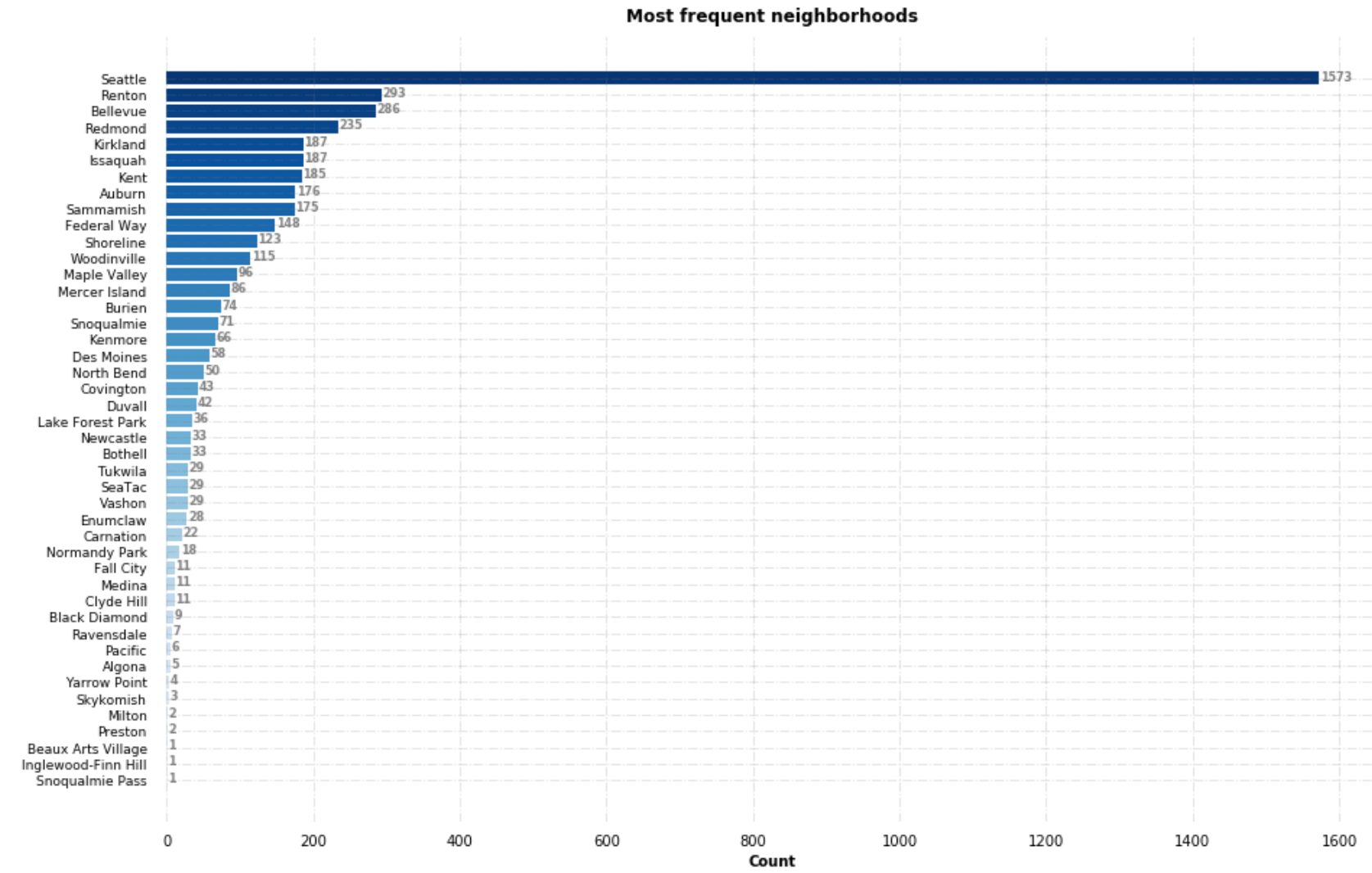
Out[48]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
        34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50,
        51, 52, 53, 54, 55, 56, 57, 58]),
 <a list of 59 Text xticklabel objects>)
```



In [49]:
```
# count number of city
dt["city"].nunique()
```

Out[49]: 44

In [50]:
```
# Visualize price by city
# Figure Size
fig, ax = plt.subplots(figsize=(15,10))

# Horizontal Bar Plot
title_cnt=dt.city.value_counts().sort_values(ascending=False).reset_index()
mn= ax.barh(title_cnt.iloc[:,0], title_cnt.iloc[:,1], color=sns.color_palette('Blues_r',len(title_cnt)))

# Remove axes splines
for s in ['top','bottom','left','right']:
    ax.spines[s].set_visible(False)

# Remove x,y Ticks
ax.xaxis.set_ticks_position('none')
ax.yaxis.set_ticks_position('none')

# Add padding between axes and labels
ax.xaxis.set_tick_params(pad=5)
ax.yaxis.set_tick_params(pad=10)

# Add x,y gridlines
ax.grid(b=True, color='grey', linestyle='-.', linewidth=1, alpha=0.2)

# Show top values
ax.invert_yaxis()

# Add Plot Title
ax.set_title('Most frequent neighborhoods',weight='bold',
             loc='center', pad=10, fontsize=12)
ax.set_xlabel('Count', weight='bold')


# Add annotation to bars
for i in ax.patches:
    ax.text(i.get_width()+1, i.get_y()+0.5, str(round((i.get_width()), 2)),
            fontsize=8, fontweight='bold', color='grey')
plt.yticks(fontsize=9)


plt.show()
# Show Plot
plt.show()
```


Most frequent neighborhoods
Seattle 1573
Renton 293
Bellevue 286
Redmond 235
Kirkland 187
Issaquah 187
Kent 185
Auburn 176
Sammamish 175
Federal Way 148
Shoreline 123
Woodinville 115
Maple Valley 96
Mercer Island 86
Burien 74
Snoqualmie 71
Kenmore 66
Des Moines 58
North Bend 50
Covington 43
Duvall 42
Lake Forest Park 36
Newcastle 33
Bothell 33
Tukwila 29
SeaTac 29
Vashon 29
Enumclaw 28
Carnation 22
Normandy Park 18
Fall City 11
Medina 11
Clyde Hill 11
Black Diamond 9
Ravensdale 7
Pacific 6
Algona 5
Yarrow Point 4
Skykomish 3
Milton 2
Preston 2
Beaux Arts Village 1
Inglewood-Finn Hill 1
Snoqualmie Pass 1
0 200 400 600 800 1000 1200 1400 1600
Count

In [52]:
```
# checking mean price by city
c= dt.groupby('city').mean().sort_values(by = ['price'], ascending = True)
print(c)
```

city	price	bedrooms	bathrooms	sqft_living	\
Algona	2.072880e+05	3.200000	1.900000	1608.600000	
Pacific	2.252333e+05	3.333333	1.958333	1520.833333	
Skykomish	2.330000e+05	2.666667	1.666667	1356.666667	
SeaTac	2.452906e+05	3.275862	1.724138	1678.517241	
Milton	2.850000e+05	3.000000	1.750000	1255.000000	
Federal Way	2.898877e+05	3.500000	2.108108	2054.114865	
Covington	2.962304e+05	3.325581	1.970930	1792.558140	
Auburn	2.993404e+05	3.420455	2.092330	2019.579545	
Des Moines	3.049925e+05	3.241379	1.862069	1812.620690	
Enumclaw	3.076146e+05	3.178571	1.758929	1922.500000	
Tukwila	3.082901e+05	3.068966	1.663793	1706.206897	
Maple Valley	3.364749e+05	3.489583	2.346354	2086.041667	
Black Diamond	3.396056e+05	3.222222	1.750000	1863.333333	
Burien	3.489472e+05	3.337838	1.746622	1815.337838	
Renton	3.770410e+05	3.481229	2.122014	2114.761092	
North Bend	3.995657e+05	3.300000	2.200000	1995.400000	
Duvall	4.039941e+05	3.380952	2.267857	2161.547619	
Shoreline	4.203924e+05	3.308943	1.800813	1774.837398	
Inglewood-Finn Hill	4.250000e+05	4.000000	2.000000	1520.000000	
Kent	4.394924e+05	3.459459	2.139189	1981.270270	
Kenmore	4.474940e+05	3.560606	2.196970	2110.530303	
Lake Forest Park	4.484750e+05	3.750000	2.270833	2283.055556	
Vashon	4.725569e+05	2.758621	1.853448	1889.689655	
Bothell	4.814419e+05	3.606061	2.431818	2319.393939	
Normandy Park	5.067931e+05	3.444444	2.013889	2093.277778	
Carnation	5.087520e+05	3.090909	2.170455	2392.454545	
Ravensdale	5.140714e+05	3.428571	2.035714	2612.857143	
Snoqualmie Pass	5.250000e+05	3.000000	2.750000	2100.000000	
Snoqualmie	5.363053e+05	3.577465	2.633803	2716.056338	
Preston	5.624500e+05	2.500000	2.375000	2280.000000	
Seattle	5.798375e+05	3.169739	1.962174	1828.623649	
Issaquah	5.961637e+05	3.561497	2.593583	2458.844920	
Woodinville	6.095650e+05	3.556522	2.378261	2663.008696	
Kirkland	6.515836e+05	3.540107	2.327540	2259.481283	
Newcastle	6.660467e+05	3.757576	2.515152	2689.090909	
Redmond	6.676495e+05	3.421277	2.380851	2491.761702	
Sammamish	6.869176e+05	3.725714	2.578571	2830.120000	
Fall City	6.926818e+05	3.818182	2.568182	2865.909091	
Beaux Arts Village	7.450000e+05	3.000000	1.750000	1490.000000	
Bellevue	8.471807e+05	3.804196	2.453671	2694.527972	
Mercer Island	1.123818e+06	4.116279	2.834302	3275.313953	

Yarrow Point	1.194838e+06	3.750000	1.937500	2472.500000
Clyde Hill	1.321945e+06	4.181818	2.613636	3620.909091
Medina	2.046559e+06	4.090909	3.045455	4103.636364

city	sqft_lot	floors	waterfront	view	condition	\
Algona	11494.200000	1.400000	0.000000	0.000000	3.200000	
Pacific	9159.166667	1.333333	0.000000	0.000000	3.166667	
Skykomish	36042.000000	1.666667	0.000000	0.000000	3.333333	
SeaTac	11469.931034	1.189655	0.000000	0.172414	3.206897	
Milton	10150.000000	1.000000	0.000000	0.000000	4.000000	
Federal Way	11455.209459	1.344595	0.006757	0.175676	3.432432	
Covington	12122.976744	1.348837	0.000000	0.000000	3.651163	
Auburn	24999.221591	1.448864	0.000000	0.125000	3.409091	
Des Moines	10517.706897	1.250000	0.034483	0.689655	3.482759	
Enumclaw	93430.392857	1.303571	0.000000	0.607143	3.928571	
Tukwila	11494.758621	1.379310	0.000000	0.068966	3.034483	
Maple Valley	13730.354167	1.796875	0.000000	0.041667	3.166667	
Black Diamond	25006.666667	1.388889	0.000000	0.000000	3.222222	
Burien	12158.554054	1.209459	0.040541	0.540541	3.554054	
Renton	14555.914676	1.482935	0.003413	0.136519	3.433447	
North Bend	32053.680000	1.570000	0.000000	0.220000	3.320000	
Duvall	45446.357143	1.642857	0.000000	0.000000	3.071429	
Shoreline	9102.365854	1.219512	0.000000	0.227642	3.642276	
Inglewood-Finn Hill	7983.000000	1.000000	0.000000	0.000000	5.000000	
Kent	15156.924324	1.454054	0.000000	0.032432	3.518919	
Kenmore	12902.696970	1.484848	0.000000	0.106061	3.378788	
Lake Forest Park	12138.472222	1.250000	0.000000	0.333333	3.416667	
Vashon	83760.517241	1.517241	0.241379	1.241379	3.586207	
Bothell	9060.181818	1.500000	0.000000	0.000000	3.151515	
Normandy Park	13441.277778	1.055556	0.055556	0.777778	3.888889	
Carnation	64873.772727	1.522727	0.000000	0.318182	3.090909	
Ravensdale	132017.142857	1.714286	0.000000	0.000000	3.142857	
Snoqualmie Pass	10362.000000	2.000000	0.000000	0.000000	3.000000	
Snoqualmie	17616.535211	1.873239	0.000000	0.281690	3.028169	
Preston	116130.000000	1.750000	0.000000	0.000000	4.500000	
Seattle	5326.625556	1.530833	0.002543	0.297521	3.523204	
Issaquah	24724.074866	1.799465	0.005348	0.090909	3.310160	
Woodinville	42377.417391	1.547826	0.000000	0.026087	3.382609	
Kirkland	10317.705882	1.459893	0.005348	0.165775	3.491979	
Newcastle	10934.939394	1.727273	0.000000	0.000000	3.696970	
Redmond	23936.191489	1.563830	0.008511	0.055319	3.238298	
Sammamish	16127.331429	1.768571	0.017143	0.194286	3.165714	

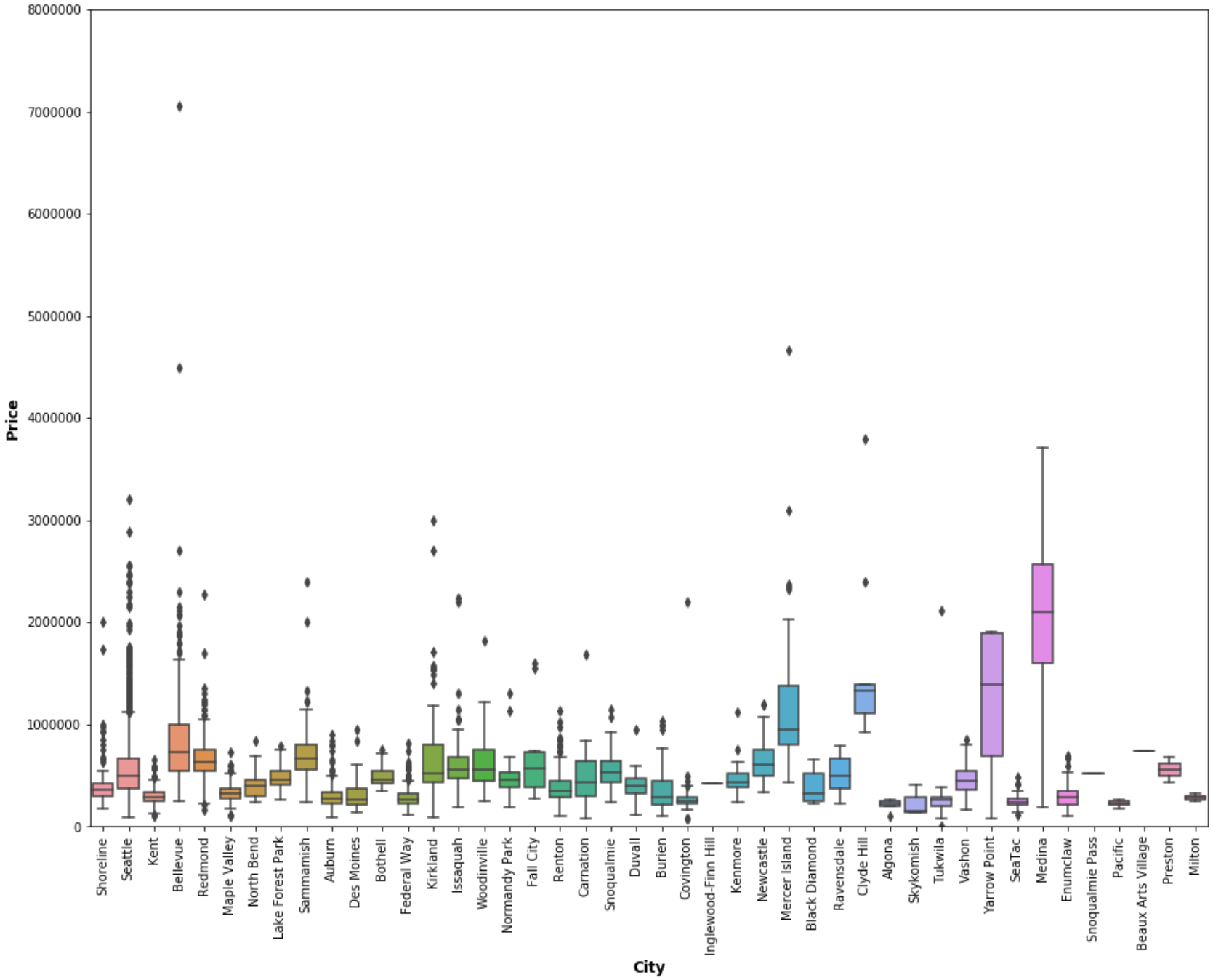
Fall City	91681.363636	1.363636	0.000000	0.272727	3.272727
Beaux Arts Village	9800.000000	1.000000	0.000000	0.000000	4.000000
Bellevue	13741.178322	1.398601	0.006993	0.391608	3.590909
Mercer Island	14654.302326	1.598837	0.058140	0.848837	3.790698
Yarrow Point	13373.000000	2.000000	0.000000	0.500000	3.750000
Clyde Hill	19633.909091	1.181818	0.000000	0.818182	3.545455
Medina	20626.181818	1.636364	0.000000	0.454545	3.454545

city	sqft_above	sqft_basement	yr_built	yr_renovated
Algona	1608.600000	0.000000	1983.200000	800.400000
Pacific	1520.833333	0.000000	1993.333333	335.666667
Skykomish	1356.666667	0.000000	1963.333333	667.666667
SeaTac	1435.275862	243.241379	1965.448276	896.344828
Milton	1255.000000	0.000000	1983.000000	0.000000
Federal Way	1775.108108	279.006757	1978.189189	688.324324
Covington	1648.837209	143.720930	1984.651163	558.116279
Auburn	1862.250000	157.329545	1980.795455	725.761364
Des Moines	1509.517241	303.103448	1964.689655	1137.051724
Enumclaw	1823.928571	98.571429	1968.250000	355.250000
Tukwila	1417.931034	288.275862	1956.413793	1241.655172
Maple Valley	2049.270833	36.770833	1996.447917	457.239583
Black Diamond	1807.777778	55.555556	1978.666667	1107.333333
Burien	1468.445946	346.891892	1957.351351	1183.945946
Renton	1890.699659	224.061433	1981.266212	733.716724
North Bend	1879.200000	116.200000	1983.280000	909.700000
Duvall	2010.833333	150.714286	1991.333333	520.166667
Shoreline	1489.552846	285.284553	1959.764228	1215.959350
Inglewood-Finn Hill	1520.000000	0.000000	1967.000000	0.000000
Kent	1770.135135	211.135135	1982.108108	497.135135
Kenmore	1792.045455	318.484848	1979.363636	695.393939
Lake Forest Park	1798.611111	484.444444	1965.500000	1276.611111
Vashon	1751.241379	138.448276	1964.965517	894.827586
Bothell	1957.878788	361.515152	1982.696970	727.151515
Normandy Park	1717.722222	375.555556	1960.722222	1107.722222
Carnation	2205.181818	187.272727	1982.818182	727.318182
Ravensdale	2612.857143	0.000000	1986.000000	571.571429
Snoqualmie Pass	1510.000000	590.000000	1998.000000	2006.000000
Snoqualmie	2607.323944	108.732394	1999.633803	444.126761
Preston	2205.000000	75.000000	1949.500000	994.000000
Seattle	1437.729816	390.893833	1952.338843	992.223776
Issaquah	2122.695187	336.149733	1992.716578	405.802139
Woodinville	2405.217391	257.791304	1982.704348	782.704348

Kirkland	1955.149733	304.331551	1979.385027	662.919786
Newcastle	2599.696970	89.393939	1989.424242	237.696970
Redmond	2288.655319	203.106383	1989.502128	526.676596
Sammamish	2664.062857	166.057143	1991.668571	696.685714
Fall City	2611.818182	254.090909	1972.636364	909.727273
Beaux Arts Village	1140.000000	350.000000	1947.000000	1988.000000
Bellevue	2182.604895	511.923077	1973.814685	823.517483
Mercer Island	2623.151163	652.162791	1972.720930	672.651163
Yarrow Point	2472.500000	0.000000	1965.750000	992.000000
Clyde Hill	2522.727273	1098.181818	1964.454545	1092.000000
Medina	3420.000000	683.636364	1972.454545	1089.000000

In [354]:
```
var = 'city'
f, ax = plt.subplots(figsize=(16, 12))
fig = sns.boxplot(x=var, y="price", data=dt_new)
fig.axis(ymin=0, ymax=8000000);
xt = plt.xticks(rotation=90)
# Add Plot Title
ax.set_xlabel('City', weight='bold', fontsize = 12)
ax.set_ylabel('Price', weight='bold', fontsize = 12)
```

Out[354]: Text(0, 0.5, 'Price')



In [148]:
```
# Price and State Zip
dt['statezip'].nunique()
```

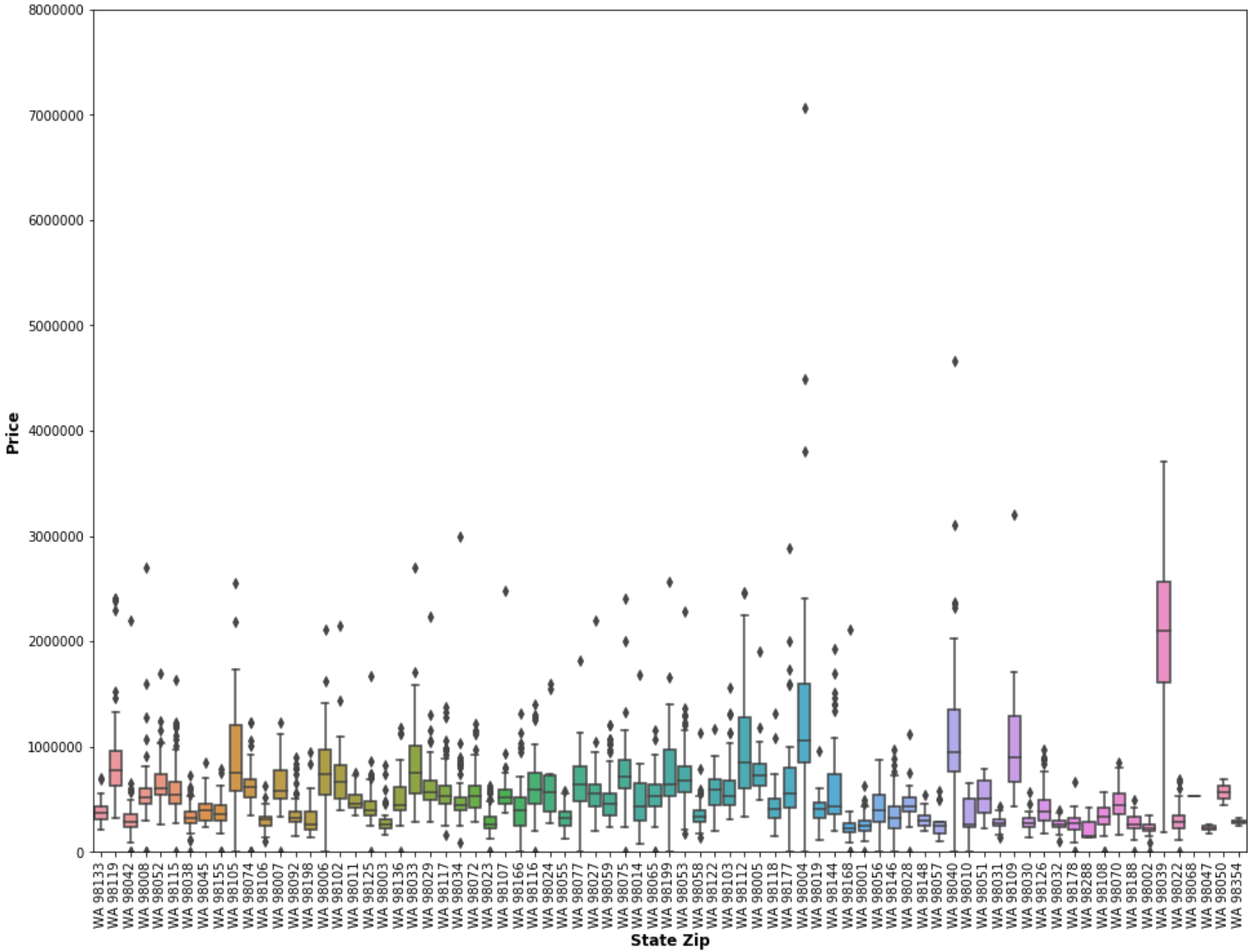
Out[148]:
```
77
```

In [149]:
```
dt['statezip'].value_counts()
```

Out[149]:
```
WA 98103    148
WA 98052    135
WA 98117    132
WA 98115    130
WA 98006    110
           ...
WA 98047      6
WA 98288      3
WA 98354      2
WA 98050      2
WA 98068      1
Name: statezip, Length: 77, dtype: int64
```

In [150]:
```
var = 'statezip'
f, ax = plt.subplots(figsize=(16, 12))
fig = sns.boxplot(x=var, y="price", data=dt)
fig.axis(ymin=0, ymax=8000000);
xt = plt.xticks(rotation=90)
# Add Plot Title
ax.set_xlabel('State Zip', weight='bold', fontsize = 12)
ax.set_ylabel('Price', weight='bold', fontsize = 12)
```

Out[150]: Text(0, 0.5, 'Price')



In [336]:
```
# Feature Engineering
# Getting only year in sale date
import datetime
dt_new['date'] = pd.to_datetime(dt_new['date'])
```

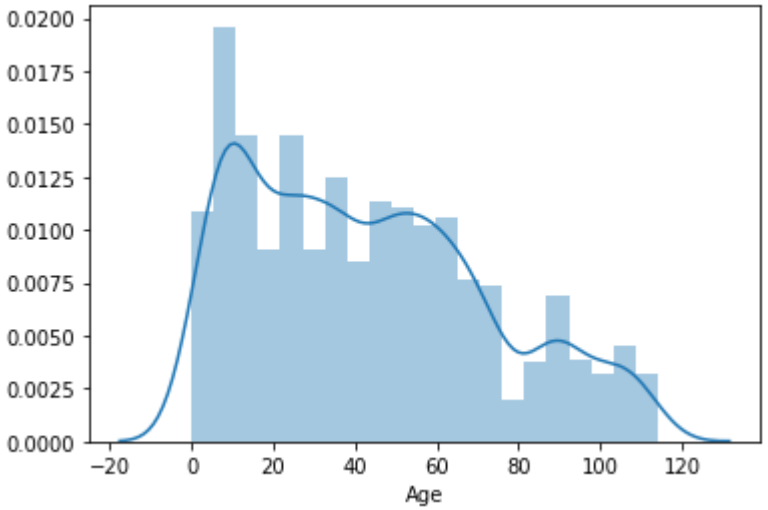
In [337]:
```
#Generating house age
dt_new['Age'] = 2014 - dt_new['yr_built']
```

In [338]:
```
#Top 15 Age that have most observations
dt_new['Age'].value_counts().sort_values(ascending = False).head(15)
```

Out[338]:
```
8     109
9     103
7      92
10     92
36     90
11     89
6      88
47     82
37     79
0      78
46     76
27     74
25     72
55     67
24     66
Name: Age, dtype: int64
```

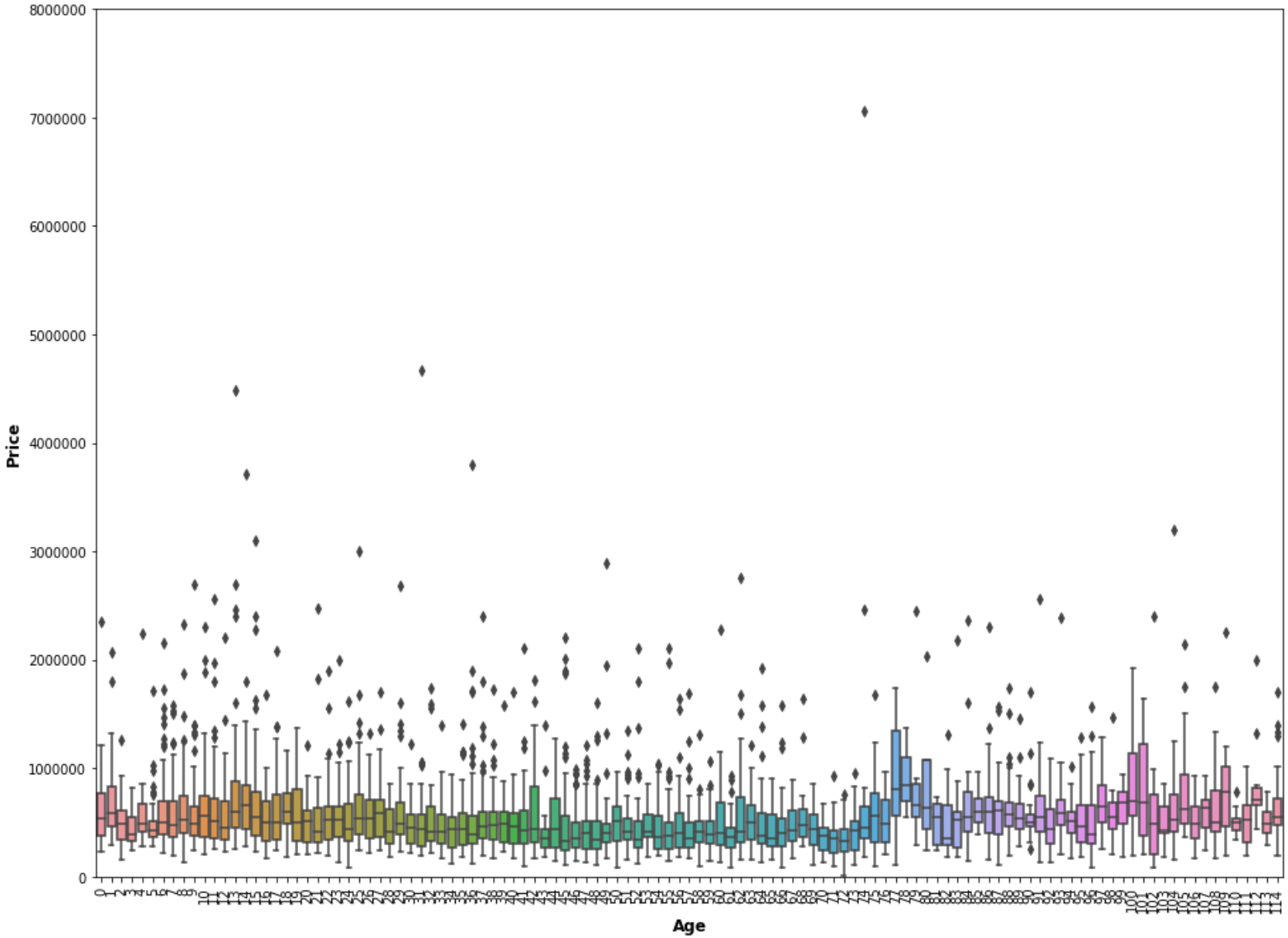
In [339]:
```
# Histogram of house's age
sns.distplot(dt_new['Age'])
```

Out[339]: <matplotlib.axes._subplots.AxesSubplot at 0x272446325c8>



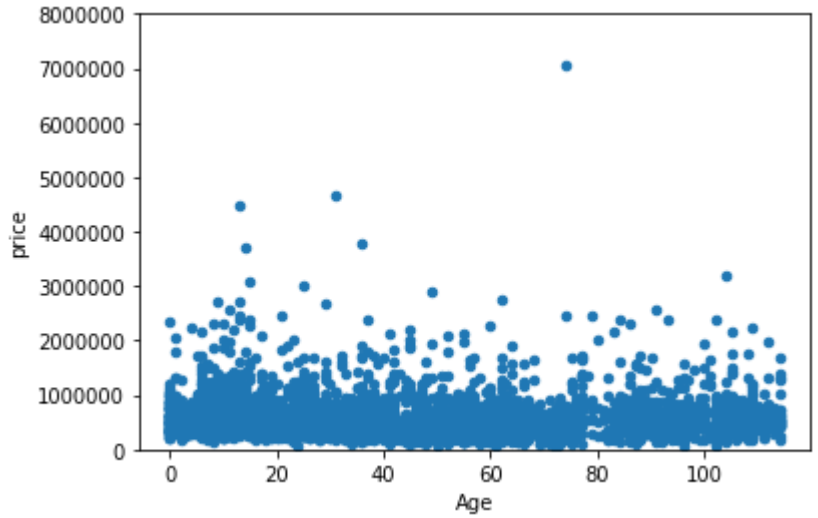
In [340]:
```
var = 'Age'
f, ax = plt.subplots(figsize=(16, 12))
fig = sns.boxplot(x=var, y="price", data=dt_new)
fig.axis(ymin=0, ymax=8000000);
xt = plt.xticks(rotation=90)
# Add Plot Title
ax.set_xlabel('Age', weight='bold', fontsize = 12)
ax.set_ylabel('Price', weight='bold', fontsize = 12)
```

Out[340]: Text(0, 0.5, 'Price')



In [371]:
```
dt_new.plot.scatter(x='Age', y='price', ylim=(0,8000000))
```

Out[371]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2724a8da208>
```



In [356]:
```
dt_new.groupby('Age').mean().sort_values(by = ['price'], ascending = True)
```

Out[356]:

Age	price	bedrooms	bathrooms	sqft_lot	floors	waterfront	condition	sqft_above	sqft_basement	Renovated
71	346578.680000	2.742857	1.364286	8366.542857	1.228571	0.028571	3.685714	1094.114286	223.428571	0.485714
72	346766.112319	2.759259	1.449074	7925.555556	1.148148	0.000000	3.648148	1131.111111	255.000000	0.796296
70	362486.753471	3.041667	1.354167	6539.166667	1.208333	0.000000	3.541667	1142.916667	279.166667	0.000000
61	404973.873873	3.243243	1.547297	18444.324324	1.162162	0.000000	3.675676	1481.081081	186.756757	0.459459
46	411965.309211	3.631579	1.914474	13129.671053	1.092105	0.013158	3.789474	1567.500000	373.684211	0.302632
...	...	...	...	...	...	...	...	...	...	...
100	864360.000000	3.200000	2.100000	5360.500000	1.600000	0.000000	3.600000	1765.000000	322.000000	1.000000
80	881750.000000	3.500000	2.062500	12920.500000	1.625000	0.000000	3.500000	1970.000000	622.500000	1.000000
79	914333.333333	3.333333	2.125000	7823.500000	1.666667	0.000000	3.833333	2130.000000	561.666667	0.500000
78	918500.000000	3.000000	2.500000	9683.666667	1.500000	0.000000	3.666667	1636.666667	570.000000	0.000000
77	927218.750000	3.550000	2.412500	15558.550000	1.625000	0.100000	3.550000	1958.800000	372.500000	0.600000

115 rows × 10 columns

In [345]:
```
def test_func(dt_new):
    if dt_new['yr_renovated'] > 0:
        return 1
    else:
        return 0

dt_new['Renovated'] = dt_new.apply(test_func, axis=1)
```

In [346]:
```
dt_new['Renovated'].value_counts()
```

Out[346]:
```
0    2705
1    1842
Name: Renovated, dtype: int64
```

In [347]:
```
dt_new.groupby('Renovated').mean().sort_values(by = ['price'], ascending = True)
```

Out[347]:

Renovated	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	s
1	530788.440553	3.330619	1.952633	1989.990228	13896.461455	1.358035	0.008143	0.257872	3.300217	1653.820304	
0	561714.525500	3.441035	2.294362	2228.325323	15477.309057	1.616636	0.005545	0.218484	3.551201	1935.670980	

In [352]:
```
dt_new = dt_new.drop(['sqft_living', 'view', 'country', 'street', 'date', 'statezip', 'yr_built', 'yr_renovated'], axis = 1)
```

In [353]:
```
dt_new.info()
```

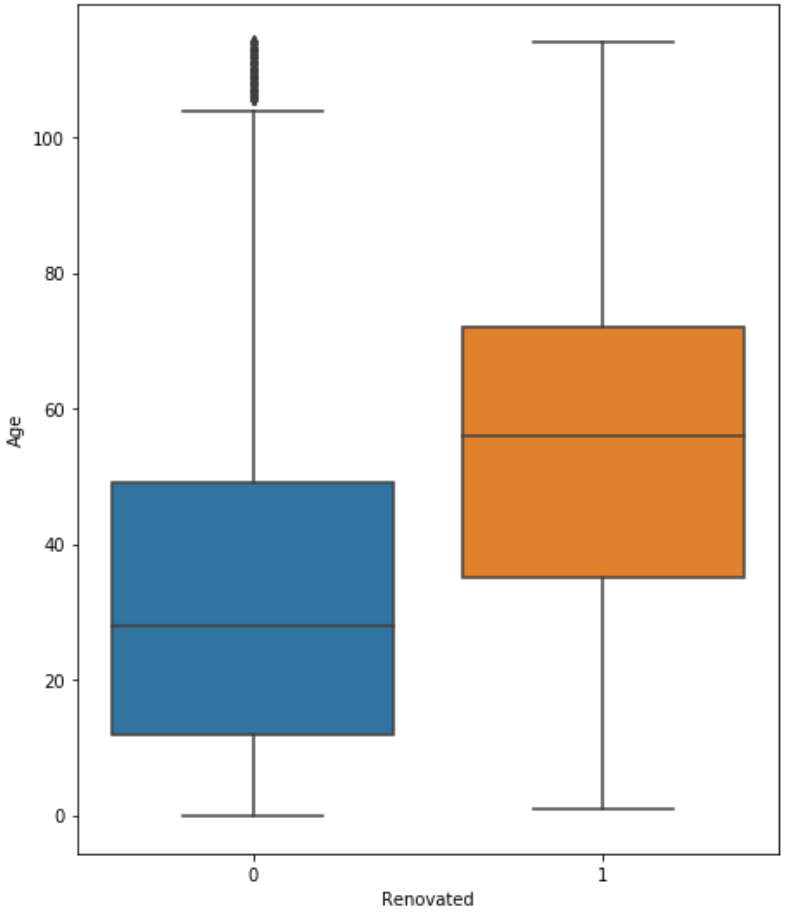
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4547 entries, 0 to 4599
Data columns (total 12 columns):
price            4547 non-null float64
bedrooms         4547 non-null int64
bathrooms        4547 non-null float64
sqft_lot         4547 non-null int64
floors           4547 non-null float64
waterfront       4547 non-null int64
condition        4547 non-null int64
sqft_above       4547 non-null int64
sqft_basement    4547 non-null int64
city             4547 non-null object
Age              4547 non-null int64
Renovated        4547 non-null int64
dtypes: float64(3), int64(8), object(1)
memory usage: 621.8+ KB
```

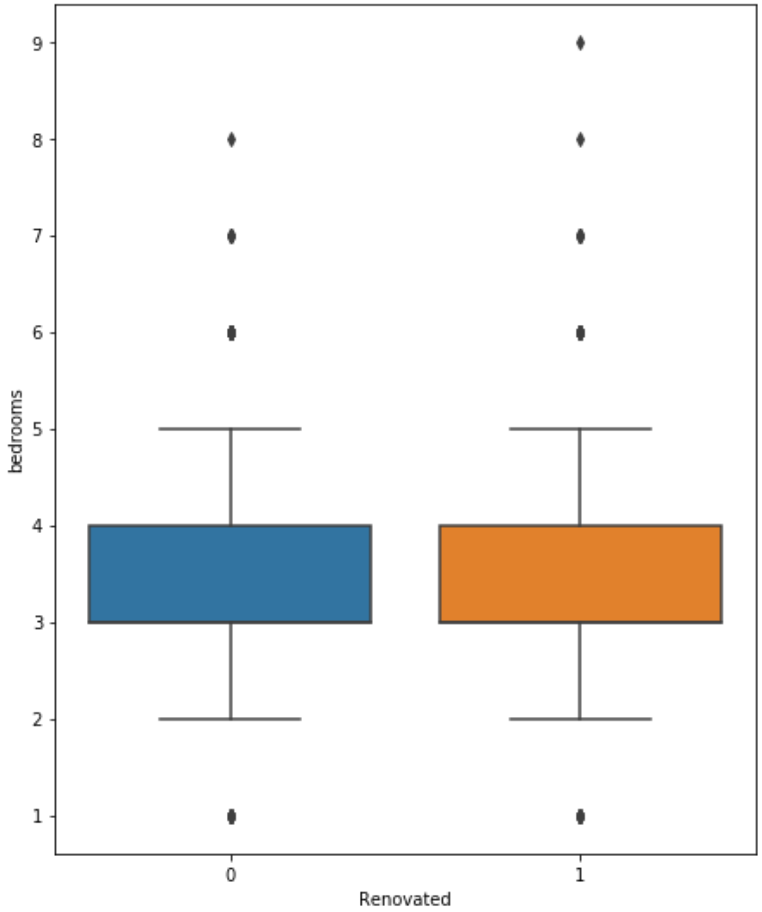
In [357]:
```
dt_new.describe()
```

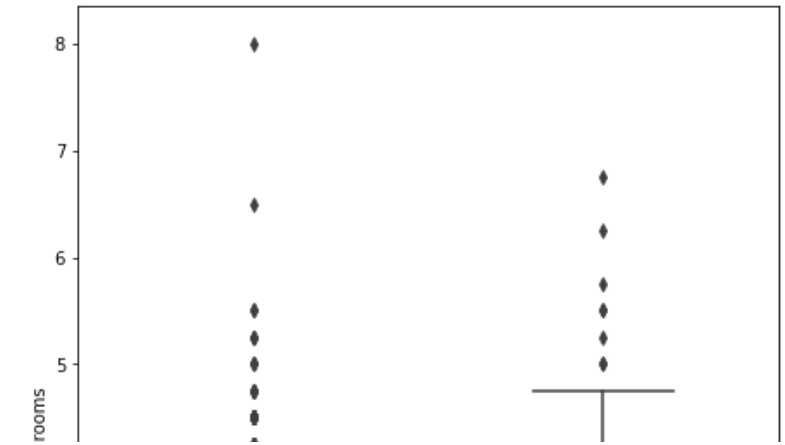
Out[357]:

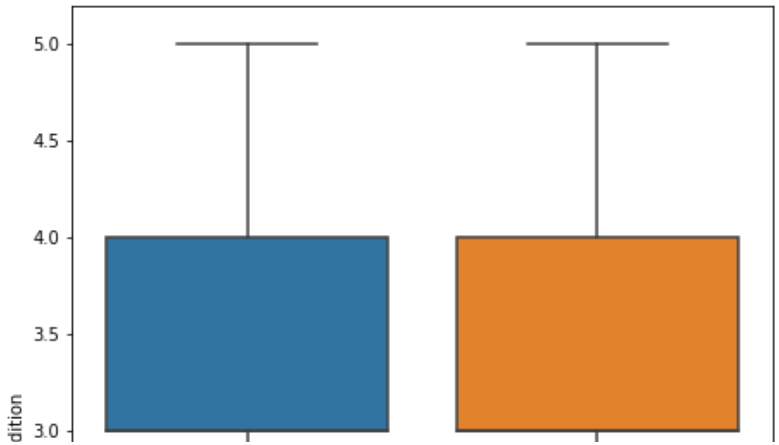
	price	bedrooms	bathrooms	sqft_lot	floors	waterfront	condition	sqft_above	sqft_basement	
count	4.547000e+03	4547.000000	4547.000000	4.547000e+03	4547.000000	4547.000000	4547.000000	4547.000000	4547.000000	454
mean	5.491863e+05	3.396305	2.155927	1.483690e+04	1.511876	0.006598	3.449527	1821.492633	310.282604	4
std	3.680563e+05	0.902148	0.775356	3.597887e+04	0.537805	0.080967	0.675309	853.417666	462.096242	2
min	7.800000e+03	1.000000	0.750000	6.380000e+02	1.000000	0.000000	1.000000	370.000000	0.000000	
25%	3.260500e+05	3.000000	1.750000	5.000000e+03	1.000000	0.000000	3.000000	1190.000000	0.000000	1
50%	4.650000e+05	3.000000	2.250000	7.680000e+03	1.500000	0.000000	3.000000	1590.000000	0.000000	3
75%	6.570500e+05	4.000000	2.500000	1.096600e+04	2.000000	0.000000	4.000000	2300.000000	600.000000	6
max	7.062500e+06	9.000000	8.000000	1.074218e+06	3.500000	1.000000	5.000000	9410.000000	4820.000000	11

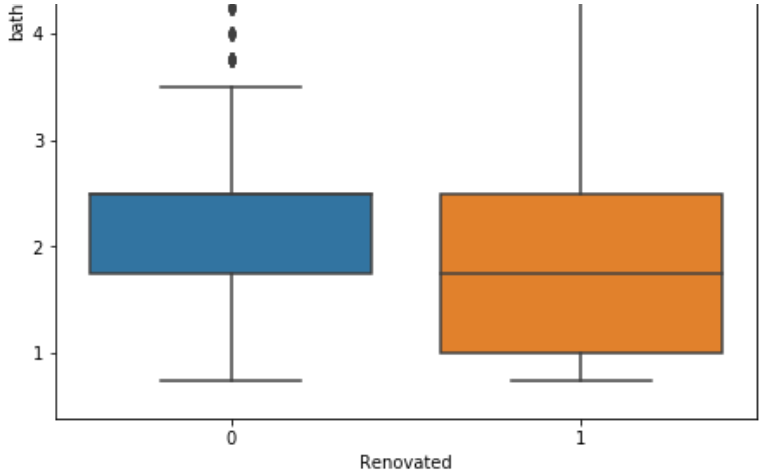
In [348]:
```
# Differences between renovated house and not renovated house
condition = ['Age', 'bedrooms', 'bathrooms', 'condition', 'price', 'floors']
var = 'Renovated'
plt.figure(figsize = (16,96))
for idx,col in enumerate(condition):
    plt.subplot(9,2,idx+1)
    fig = sns.boxplot(x=var, y=col, data=dt_new)
```

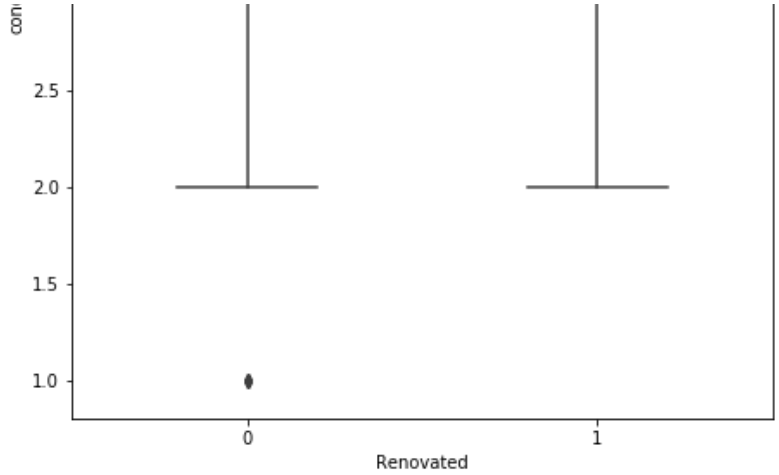

Age
Renovated
0
1
0
20
40
60
80
100

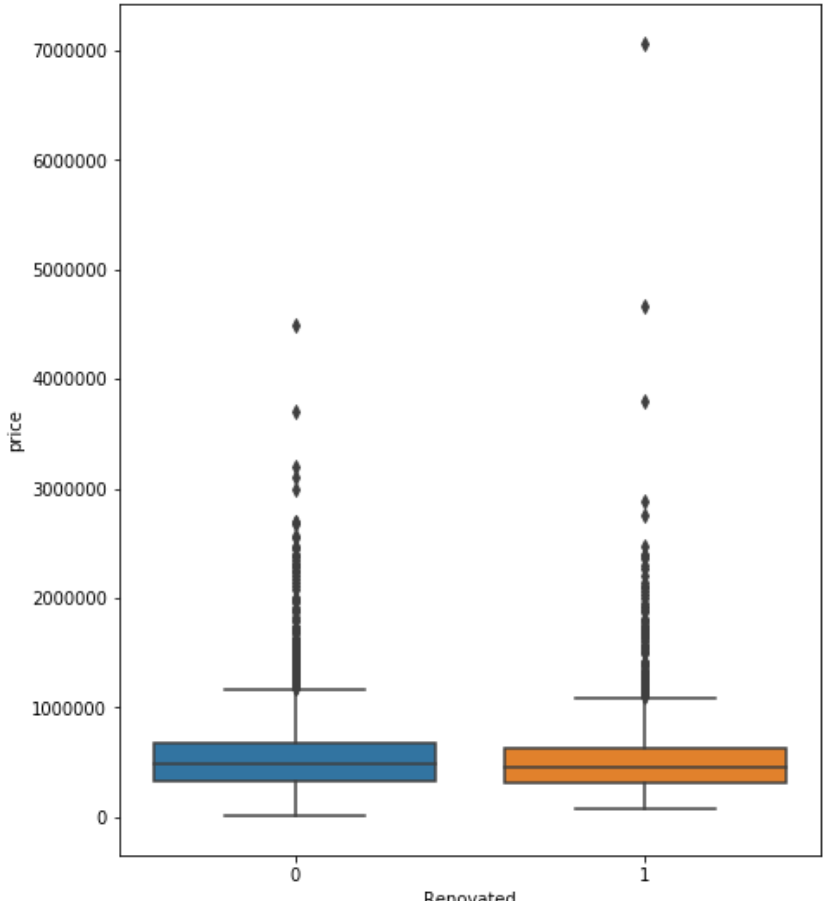

bedrooms
Renovated
0
1
1
2
3
4
5
6
7
8
9

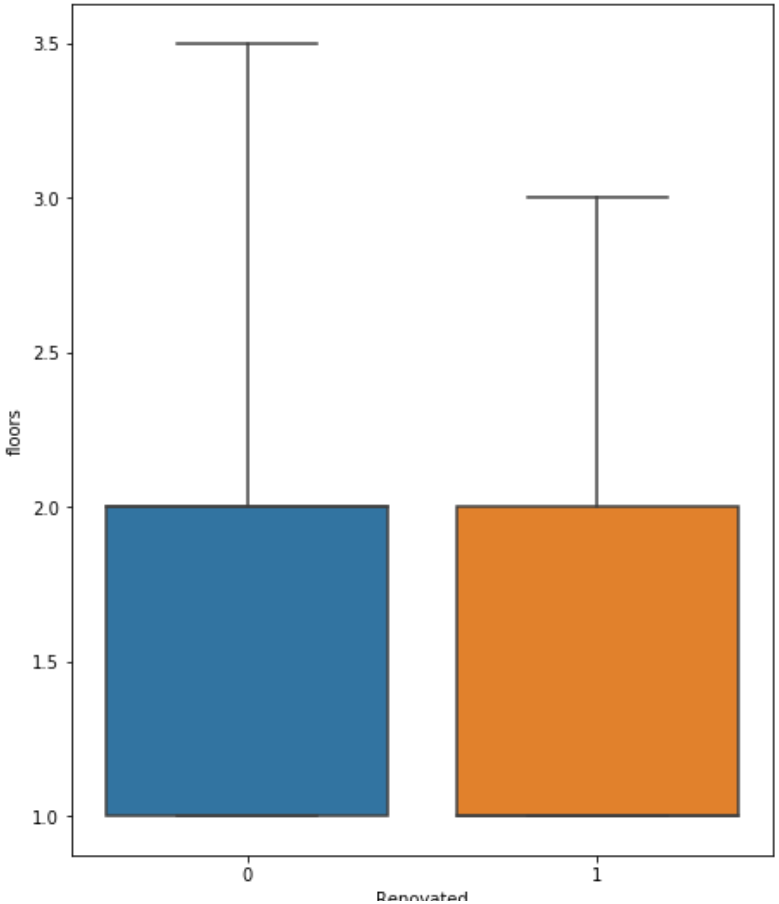

rooms
5
6
7
8


dition
3.0
3.5
4.0
4.5
5.0


bath
4
3
2
1
0
1
Renovated


2.5
2.0
1.5
1.0
0
1
Renovated


price
7000000
6000000
5000000
4000000
3000000
2000000
1000000
0
0
1


floors
3.5
3.0
2.5
2.0
1.5
1.0
0
1

In [349]:
```
# Checking correlation between price and other numeric variables
dt_new.corr()
```

Out[349]:

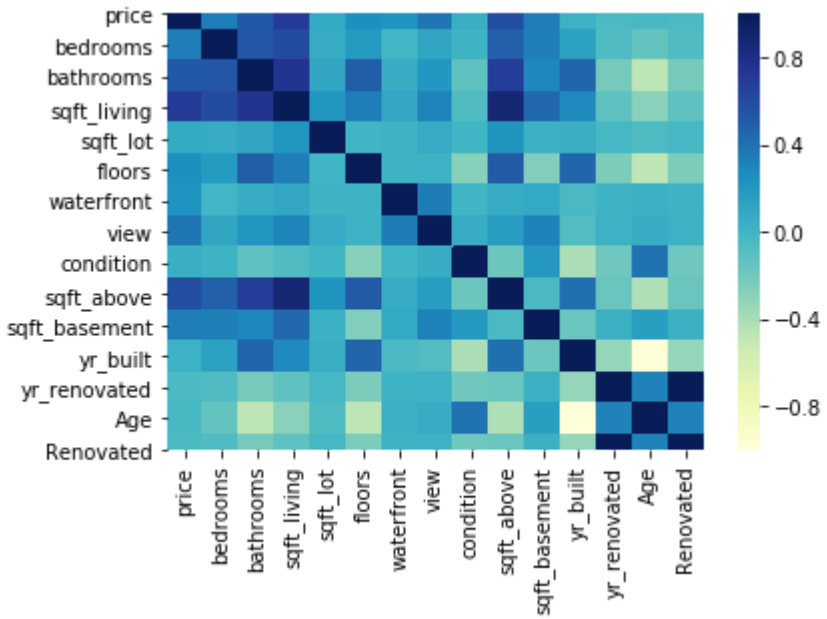
	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_bas
price	1.000000	0.336208	0.525774	0.697116	0.082391	0.254818	0.231987	0.378502	0.052363	0.596494	0.3
bedrooms	0.336208	1.000000	0.545706	0.601616	0.071407	0.180719	-0.005689	0.117111	0.022008	0.492009	0.3
bathrooms	0.525774	0.545706	1.000000	0.761598	0.109576	0.494137	0.063325	0.207257	-0.121713	0.692561	0.2
sqft_living	0.697116	0.601616	0.761598	1.000000	0.213296	0.342951	0.107926	0.309498	-0.061869	0.875418	0.4
sqft_lot	0.082391	0.071407	0.109576	0.213296	1.000000	0.004347	0.017405	0.072721	0.000963	0.219330	0.0
floors	0.254818	0.180719	0.494137	0.342951	0.004347	1.000000	0.015881	0.030114	-0.273631	0.521908	-0.2
waterfront	0.231987	-0.005689	0.063325	0.107926	0.017405	0.015881	1.000000	0.347812	0.006092	0.072692	0.0
view	0.378502	0.117111	0.207257	0.309498	0.072721	0.030114	0.347812	1.000000	0.062916	0.174451	0.3
condition	0.052363	0.022008	-0.121713	-0.061869	0.000963	-0.273631	0.006092	0.062916	1.000000	-0.176065	0.1
sqft_above	0.596494	0.492009	0.692561	0.875418	0.219330	0.521908	0.072692	0.174451	-0.176065	1.000000	-0.0
sqft_basement	0.339593	0.335121	0.295483	0.450649	0.035902	-0.254862	0.088875	0.317676	0.197256	-0.036996	1.0
yr_built	0.025440	0.143058	0.465991	0.284716	0.049181	0.467000	-0.031996	-0.066666	-0.399055	0.406559	-0.1
yr_renovated	-0.041614	-0.060383	-0.217316	-0.123006	-0.021113	-0.237685	0.015881	0.025363	-0.183943	-0.163028	0.0
Age	-0.025440	-0.143058	-0.465991	-0.284716	-0.049181	-0.467000	0.031996	0.066666	0.399055	-0.406559	0.1
Renovated	-0.041254	-0.060091	-0.216388	-0.122484	-0.021572	-0.236079	0.015754	0.025271	-0.182472	-0.162147	0.0

In [350]:
```
dt_new.corr()['price'].sort_values(ascending= False)
```

Out[350]:
```
price            1.000000
sqft_living      0.697116
sqft_above       0.596494
bathrooms        0.525774
view             0.378502
sqft_basement    0.339593
bedrooms         0.336208
floors           0.254818
waterfront       0.231987
sqft_lot         0.082391
condition        0.052363
yr_built         0.025440
Age             -0.025440
Renovated       -0.041254
yr_renovated    -0.041614
Name: price, dtype: float64
```

In [351]:
```
# Visualizing Heatmap
sns.heatmap(dt_new.corr(), cmap="YlGnBu", center= 0)
```

Out[351]: <matplotlib.axes._subplots.AxesSubplot at 0x272421a1208>



In [230]:
```
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder
```

In [69]:
```
features =[]
X = dt[features]
Y = dt['price']

x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.20, random_state=1)


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

```
number of test samples : 920
number of training samples: 3680
```

In [358]:
```
dum_df = pd.get_dummies(dt_new, columns=["city"])
```

In [368]:
```
dum_df.corr()
```

Out[368]:

	price	bedrooms	bathrooms	sqft_lot	floors	waterfront	condition	sqft_above	sqft_basement	Age	...
price	1.000000	0.336208	0.525774	0.082391	0.254818	0.231987	0.052363	0.596494	0.339593	-0.025440	...
bedrooms	0.336208	1.000000	0.545706	0.071407	0.180719	-0.005689	0.022008	0.492009	0.335121	-0.143058	...
bathrooms	0.525774	0.545706	1.000000	0.109576	0.494137	0.063325	-0.121713	0.692561	0.295483	-0.465991	...
sqft_lot	0.082391	0.071407	0.109576	1.000000	0.004347	0.017405	0.000963	0.219330	0.035902	-0.049181	...
floors	0.254818	0.180719	0.494137	0.004347	1.000000	0.015881	-0.273631	0.521908	-0.254862	-0.467000	...
waterfront	0.231987	-0.005689	0.063325	0.017405	0.015881	1.000000	0.006092	0.072692	0.088875	0.031996	...
condition	0.052363	0.022008	-0.121713	0.000963	-0.273631	0.006092	1.000000	-0.176065	0.197256	0.399055	...
sqft_above	0.596494	0.492009	0.692561	0.219330	0.521908	0.072692	-0.176065	1.000000	-0.036996	-0.406559	...
sqft_basement	0.339593	0.335121	0.295483	0.035902	-0.254862	0.088875	0.197256	-0.036996	1.000000	0.162225	...
Age	-0.025440	-0.143058	-0.465991	-0.049181	-0.467000	0.031996	0.399055	-0.406559	0.162225	1.000000	...
Renovated	-0.041254	-0.060091	-0.216388	-0.021572	-0.236079	0.015754	-0.182472	-0.162147	0.046234	0.322132	...
city_Algona	-0.030824	-0.007220	-0.010953	-0.003083	-0.006903	-0.002704	-0.012261	-0.008278	-0.022281	-0.013838	...
city_Auburn	-0.134897	0.004622	-0.017013	0.056942	-0.022489	-0.016305	-0.014675	0.010738	-0.068265	-0.069194	...
city_Beaux Arts Village	0.007892	-0.006516	-0.007766	-0.002077	-0.014118	-0.001209	0.012091	-0.011845	0.001275	0.011852	...
city_Bellevue	0.218331	0.115062	0.092404	-0.007529	-0.055773	0.001647	0.057734	0.099430	0.117064	-0.024155	...
city_Black Diamond	-0.019066	-0.012627	-0.020289	0.005476	-0.015565	-0.003421	-0.020177	-0.003763	-0.022514	-0.012996	...
city_Bothell	-0.015739	0.019882	0.030427	-0.013730	-0.001888	-0.006968	-0.037736	0.013666	0.009481	-0.034216	...
city_Burien	-0.065675	-0.016667	-0.068687	-0.009052	-0.069962	0.054945	0.014700	-0.051288	0.003773	0.054783	...
city_Carnation	-0.007661	-0.023607	0.001307	0.096982	0.001407	-0.005682	-0.037032	0.031352	-0.018563	-0.028187	...
city_Clyde Hill	0.129061	0.046261	0.027743	0.006957	-0.033190	-0.003629	-0.007663	0.033381	0.084898	0.010153	...
city_Covington	-0.064517	-0.011838	-0.023868	-0.007274	-0.032058	-0.007869	0.027645	-0.022853	-0.034092	-0.044566	...
city_Des Moines	-0.075424	-0.019522	-0.043085	-0.013647	-0.055355	0.039152	0.005594	-0.041557	-0.001766	0.023281	...
city_Duvall	-0.038094	-0.001643	0.013940	0.082155	0.023518	-0.007869	-0.054066	0.021424	-0.033346	-0.066655	...
city_Enumclaw	-0.048340	-0.014916	-0.042314	0.161309	-0.028323	-0.006299	0.058768	-0.001041	-0.034803	0.010311	...
city_Fall City	0.019201	0.023031	0.026186	0.105190	-0.013575	-0.004013	-0.012894	0.045609	-0.005989	-0.003062	...

	price	bedrooms	bathrooms	sqft_lot	floors	waterfront	condition	sqft_above	sqft_basement	Age	...
city_Federal Way	-0.124919	0.022945	-0.010267	-0.016725	-0.056380	0.000670	-0.005897	-0.008347	-0.010018	-0.046134	...
city_Inglewood-Finn Hill	-0.005005	0.009926	-0.002983	-0.002826	-0.014118	-0.001209	0.034056	-0.005240	-0.009960	0.001886	...
city_Issaquah	0.028161	0.036049	0.115287	0.056392	0.110034	-0.003116	-0.043760	0.072419	0.009991	-0.152528	...
city_Kenmore	-0.031024	0.021032	0.009236	-0.006499	-0.004382	-0.009814	-0.011577	-0.003218	0.003415	-0.036222	...
city_Kent	-0.139216	0.014236	-0.003298	0.001811	-0.020129	-0.016689	0.019450	-0.010908	-0.045099	-0.078581	...
city_Kirkland	0.057623	0.033015	0.045843	-0.026016	-0.020020	-0.003198	0.013020	0.032438	-0.002667	-0.059832	...
city_Lake Forest Park	-0.017531	0.029788	0.010532	-0.006009	-0.049390	-0.007074	-0.008635	-0.008888	0.038070	0.016357	...
city_Maple Valley	-0.083026	0.012531	0.035576	-0.004107	0.076839	-0.011905	-0.060813	0.038919	-0.086350	-0.125863	...
city_Medina	0.200366	0.037920	0.056502	0.007925	0.011400	-0.004013	0.000366	0.092249	0.039792	-0.002761	...
city_Mercer Island	0.231789	0.109016	0.115036	-0.000262	0.020057	0.070611	0.073766	0.125730	0.097908	-0.007361	...
city_Milton	-0.015059	-0.009216	-0.010984	-0.002733	-0.019968	-0.001710	0.017101	-0.013926	-0.014087	-0.008608	...
city_Newcastle	0.027150	0.034244	0.039618	-0.009274	0.034248	-0.006968	0.031333	0.077975	-0.040876	-0.053541	...
city_Normandy Park	-0.002094	0.001050	-0.012321	-0.002082	-0.051612	0.039518	0.039268	-0.005559	0.004564	0.020625	...
city_North Bend	-0.042870	-0.011258	0.005994	0.050463	0.011397	-0.008593	-0.020227	0.007131	-0.044292	-0.044262	...
city_Pacific	-0.031997	-0.002538	-0.009264	-0.005737	-0.012069	-0.002962	-0.015227	-0.012807	-0.024410	-0.027536	...
city_Preston	0.000756	-0.020844	0.005928	0.059065	0.009289	-0.001710	0.032635	0.009428	-0.010682	0.015002	...
city_Ravensdale	-0.003747	0.001405	-0.006089	0.127902	0.014780	-0.003200	-0.017834	0.036415	-0.026369	-0.020071	...
city_Redmond	0.073280	0.010223	0.070634	0.058802	0.021697	0.005608	-0.072513	0.124577	-0.053592	-0.146450	...
city_Renton	-0.120472	0.022588	-0.012315	-0.001942	-0.017472	-0.021310	-0.005073	0.018734	-0.047923	-0.091030	...
city_Sammamish	0.082613	0.069500	0.098542	0.007076	0.094522	0.012448	-0.081953	0.190373	-0.064921	-0.137707	...
city_SeaTac	-0.063127	-0.015883	-0.045747	-0.007088	-0.046175	-0.006415	-0.031587	-0.035181	-0.014200	0.016057	...
city_Seattle	0.052763	-0.180718	-0.182033	-0.191175	0.027559	-0.035974	0.075607	-0.326085	0.122975	0.446473	...
city_Shoreline	-0.058354	-0.016149	-0.076376	-0.026579	-0.090655	-0.013589	0.047597	-0.064862	-0.009021	0.061743	...

	price	bedrooms	bathrooms	sqft_lot	floors	waterfront	condition	sqft_above	sqft_basement	Age	...
city_Skykomish	-0.022076	-0.020784	-0.016215	0.015145	0.007396	-0.002094	-0.004422	-0.013996	-0.017255	0.006433	...
city_Snoqualmie	0.000899	0.019255	0.070174	0.010692	0.082568	-0.010116	-0.077310	0.112974	-0.058830	-0.119841	...
city_Snoqualmie Pass	-0.000975	-0.006516	0.011365	-0.001845	0.013463	-0.001209	-0.009874	-0.005414	0.008979	-0.013561	...
city_Tukwila	-0.052443	-0.029073	-0.050858	-0.007443	-0.019751	-0.006529	-0.049245	-0.037890	-0.003816	0.038685	...
city_Vashon	-0.016682	-0.056637	-0.031258	0.153495	0.000799	0.232343	0.016217	-0.006596	-0.029796	0.015666	...
city_Woodinville	0.028640	0.026232	0.045315	0.122315	0.012154	-0.013069	-0.019262	0.111174	-0.021696	-0.064388	...
city_Yarrow Point	0.052058	0.011635	-0.008360	-0.001207	0.026935	-0.002418	0.013204	0.022638	-0.019927	0.005020	...

55 rows × 55 columns

In [370]:
```
dum_df.corr()['city_Skykomish'].sort_values(ascending= False)
```

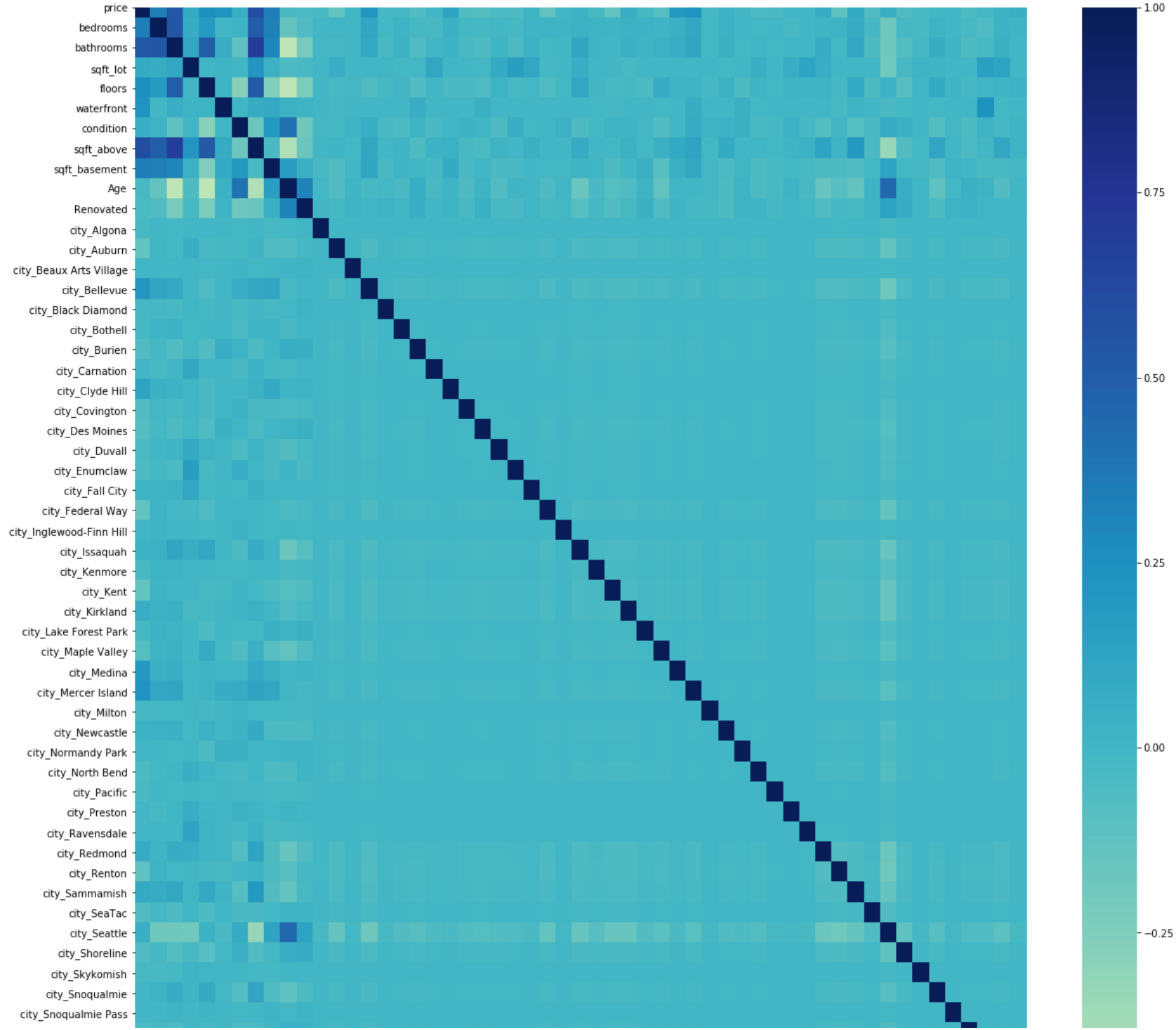
Out[370]:

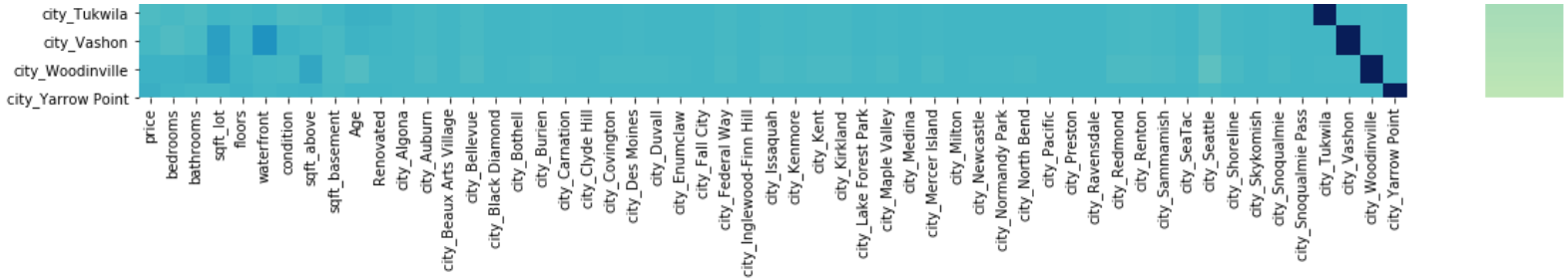
city_Skykomish	1.000000
sqft_lot	0.015145
floors	0.007396
Age	0.006433
city_Inglewood-Finn Hill	-0.000381
city_Snoqualmie Pass	-0.000381
city_Beaux Arts Village	-0.000381
city_Preston	-0.000539
city_Milton	-0.000539
city_Yarrow Point	-0.000762
city_Algona	-0.000853
city_Pacific	-0.000934
city_Ravensdale	-0.001009
city_Black Diamond	-0.001079
city_Clyde Hill	-0.001144
city_Fall City	-0.001265
city_Medina	-0.001265
city_Normandy Park	-0.001574
city_Carnation	-0.001792
city_Enumclaw	-0.001986
city_SeaTac	-0.002023
city_Vashon	-0.002059
city_Tukwila	-0.002059
waterfront	-0.002094
city_Newcastle	-0.002197
city_Bothell	-0.002197
city_Lake Forest Park	-0.002230
city_Covington	-0.002481
city_Duvall	-0.002481
city_North Bend	-0.002709
city_Des Moines	-0.002921
city_Kenmore	-0.003094
city_Snoqualmie	-0.003190
city_Burien	-0.003259
city_Mercer Island	-0.003482
city_Maple Valley	-0.003753
Renovated	-0.003756
city_Woodinville	-0.004120
city_Shoreline	-0.004284
condition	-0.004422
city_Federal Way	-0.004663
city_Sammamish	-0.005079
city_Auburn	-0.005141

```
city_Kent                  -0.005262
city_Issaquah              -0.005306
city_Kirkland              -0.005321
city_Redmond               -0.005985
city_Bellevue              -0.006595
city_Renton                -0.006719
sqft_above                 -0.013996
bathrooms                  -0.016215
sqft_basement              -0.017255
city_Seattle               -0.018560
bedrooms                   -0.020784
price                      -0.022076
Name: city_Skykomish, dtype: float64
```

In [367]:
```
f, ax = plt.subplots(figsize=(20, 20))
sns.heatmap(dum_df.corr(), cmap="YlGnBu", center= 0)
```

Out[367]: <matplotlib.axes._subplots.AxesSubplot at 0x272496436c8>


price
bedrooms
bathrooms
sqft_lot
floors
waterfront
condition
sqft_above
sqft_basement
Age
Renovated
city_Algona
city_Auburn
city_Beaux Arts Village
city_Bellevue
city_Black Diamond
city_Bothell
city_Burien
city_Carnation
city_Clyde Hill
city_Covington
city_Des Moines
city_Duvall
city_Enumclaw
city_Fall City
city_Federal Way
city_Inglewood-Finn Hill
city_Issaquah
city_Kenmore
city_Kent
city_Kirkland
city_Lake Forest Park
city_Maple Valley
city_Medina
city_Mercer Island
city_Milton
city_Newcastle
city_Normandy Park
city_North Bend
city_Pacific
city_Preston
city_Ravensdale
city_Redmond
city_Renton
city_Sammamish
city_SeaTac
city_Seattle
city_Shoreline
city_Skykomish
city_Snoqualmie
city_Snoqualmie Pass
1.00
0.75
0.50
0.25
0.00
−0.25



In [361]:
```
dum_df.to_csv('new_house_data.csv')
```

In [362]:
```
dt_new.to_csv('new_house_data_no_dummy.csv')
```

In [68]:
```
dt.columns.values.tolist()
```

Out[68]:
```
['date',
 'price',
 'bedrooms',
 'bathrooms',
 'sqft_living',
 'sqft_lot',
 'floors',
 'waterfront',
 'view',
 'condition',
 'sqft_above',
 'sqft_basement',
 'yr_built',
 'yr_renovated',
 'street',
 'city',
 'statezip',
 'country',
 'Age',
 'Renovated',
 'date',
 'price',
 'bedrooms',
 'bathrooms',
 'sqft_living',
 'sqft_lot',
 'floors',
 'waterfront',
 'view',
 'condition',
 'sqft_above',
 'sqft_basement',
 'yr_built',
 'yr_renovated',
 'street',
 'statezip',
 'country',
 'Age',
 'Renovated',
 'city_Algona',
 'city_Auburn',
 'city_Beaux Arts Village',
 'city_Bellevue',
```

```
'city_Black Diamond',
'city_Bothell',
'city_Burien',
'city_Carnation',
'city_Clyde Hill',
'city_Covington',
'city_Des Moines',
'city_Duvall',
'city_Enumclaw',
'city_Fall City',
'city_Federal Way',
'city_Inglewood-Finn Hill',
'city_Issaquah',
'city_Kenmore',
'city_Kent',
'city_Kirkland',
'city_Lake Forest Park',
'city_Maple Valley',
'city_Medina',
'city_Mercer Island',
'city_Milton',
'city_Newcastle',
'city_Normandy Park',
'city_North Bend',
'city_Pacific',
'city_Preston',
'city_Ravensdale',
'city_Redmond',
'city_Renton',
'city_Sammamish',
'city_SeaTac',
'city_Seattle',
'city_Shoreline',
'city_Skykomish',
'city_Snoqualmie',
'city_Snoqualmie Pass',
'city_Tukwila',
'city_Vashon',
'city_Woodinville',
'city_Yarrow Point']
```