Final Capstone Regression Project

October 2, 2022

1 Capstone Regression Project

by: Rio Kinslow

1.1 Business Understanding

Building a Multi-Variate Linear Regression Model using King County,WA House Prices Dataset

For this multiple linear regression project I will be using the kc_house_data.csv dataset. I will obtain the data using the pandas package and retrieve valuable information pertaining to the dataset using its associated modules. I will then scrub the dataset, going column per column, and inspecting for null values and dropping unnecessary columns that we won't be using in our linear regression. There will be some renaming of columns and also creation of dummies that will aid in the process. The columns with a vast number of null values will be filled in with the median, whereas the columns with not many null values will be filled with 0's. During the exploration phase of this project, we will be creating visualizations using the matplotlib library and also seaborn. I will be creating barplots, scatterplots, bargraph and matrices. These visualizations will help us derive particular features that may be of interest to us as we move along. The trends and correlations we observe will help drive our linear regression moving forward.

1.2 Data Understanding

After completing this initial phase of the project, I will dive right into the moduling phase of the project which encompasses building boxplots to deal with outliers. But, first I will need to deal with the categorical and continuous features for my model I will be using. For the categorical features I want, I will be using dummy datasets, whereas for the continuous features, I will then perform the linear regression looking at valuable information such as the rˆ2 score, & significant coefficient value ,as well as the average predicted price and the average actual price for that particular model. I will conduct two models. For each, I will also going to test for model accuracy and looking at the significant features we used in the model that were below a p-value of 0.05.

1.3 Data Preparation

1.3.1 Loading the Data

[1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
import seaborn as sns
import statsmodels.formula.api as smf
import scipy.stats as stats
import statsmodels.stats.api as sms

import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
sns.set(style='dark')
plt.style.use('seaborn')
```

[2]:

```
# loading in dataset and displaying head and tail of dataset

df = pd.read_csv('./data/kc_house_data.csv')
display(df.head())
df.tail()
```

```
           id        date     price  bedrooms  bathrooms  sqft_living  \
0  7399300360   5/24/2022  675000.0         4        1.0         1180
1  8910500230  12/13/2021  920000.0         5        2.5         2770
2  1180000275   9/29/2021  311000.0         6        2.0         2880
3  1604601802  12/14/2021  775000.0         3        3.0         2160
4  8562780790   8/24/2021  592500.0         2        2.0         1120

   sqft_lot  floors waterfront greenbelt  ... sewer_system sqft_above  \
0      7140     1.0         NO        NO  ...       PUBLIC       1180
1      6703     1.0         NO        NO  ...       PUBLIC       1570
2      6156     1.0         NO        NO  ...       PUBLIC       1580
3      1400     2.0         NO        NO  ...       PUBLIC       1090
4       758     2.0         NO        NO  ...       PUBLIC       1120

  sqft_basement sqft_garage sqft_patio yr_built  yr_renovated  \
0             0           0         40     1969             0
1          1570           0        240     1950             0
2          1580           0          0     1956             0
3          1070         200        270     2010             0
4           550         550         30     2012             0

                                              address        lat       long
0  2102 Southeast 21st Court, Renton, Washington ...  47.461975 -122.19052
1  11231 Greenwood Avenue North, Seattle, Washing...  47.711525 -122.35591
2  8504 South 113th Street, Seattle, Washington 9...  47.502045 -122.22520
3  4079 Letitia Avenue South, Seattle, Washington...  47.566110 -122.29020
4  2193 Northwest Talus Drive, Issaquah, Washingt...  47.532470 -122.07188

[5 rows x 25 columns]
```

```
[2]:                id        date      price  bedrooms  bathrooms  sqft_living  \
     30150  7834800180  11/30/2021  1555000.0         5        2.0         1910
     30151   194000695   6/16/2021  1313000.0         3        2.0         2020
     30152  7960100080   5/27/2022   800000.0         3        2.0         1620
     30153  2781280080   2/24/2022   775000.0         3        2.5         2570
     30154  9557800100   4/29/2022   500000.0         3        1.5         1200

            sqft_lot  floors waterfront greenbelt  … sewer_system sqft_above  \
     30150      4000     1.5         NO        NO  …       PUBLIC       1600
     30151      5800     2.0         NO        NO  …       PUBLIC       2020
     30152      3600     1.0         NO        NO  …       PUBLIC        940
     30153      2889     2.0         NO        NO  …       PUBLIC       1830
     30154     11058     1.0         NO        NO  …       PUBLIC       1200

           sqft_basement sqft_garage sqft_patio yr_built  yr_renovated  \
     30150          1130           0        210     1921             0
     30151             0           0        520     2011             0
     30152           920         240        110     1995             0
     30153           740         480        100     2006             0
     30154             0         420          0     1965             0

                                                     address        lat        long
     30150  4673 Eastern Avenue North, Seattle, Washington…  47.664740 -122.32940
     30151  4131 44th Avenue Southwest, Seattle, Washingto…  47.565610 -122.38851
     30152  910 Martin Luther King Jr Way, Seattle, Washin…  47.610395 -122.29585
     30153  17127 114th Avenue Southeast, Renton, Washingt…  47.449490 -122.18908
     30154  18615 7th Avenue South, Burien, Washington 981…  47.435840 -122.32634

     [5 rows x 25 columns]
```

1.4 Data Exploration

```
[3]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30155 entries, 0 to 30154
Data columns (total 25 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   id              30155 non-null  int64
 1   date            30155 non-null  object
 2   price           30155 non-null  float64
 3   bedrooms        30155 non-null  int64
 4   bathrooms       30155 non-null  float64
 5   sqft_living     30155 non-null  int64
 6   sqft_lot        30155 non-null  int64
 7   floors         30155 non-null  float64
 8   waterfront     30155 non-null  object
 9   greenbelt      30155 non-null  object
 10  nuisance       30155 non-null  object
 11  view           30155 non-null  object
 12  condition      30155 non-null  object
 13  grade          30155 non-null  object
 14  heat_source    30123 non-null  object
 15  sewer_system   30141 non-null  object
 16  sqft_above     30155 non-null  int64
 17  sqft_basement  30155 non-null  int64
 18  sqft_garage    30155 non-null  int64
 19  sqft_patio     30155 non-null  int64
 20  yr_built       30155 non-null  int64
 21  yr_renovated   30155 non-null  int64
 22  address        30155 non-null  object
 23  lat            30155 non-null  float64
 24  long           30155 non-null  float64
dtypes: float64(5), int64(10), object(10)
memory usage: 5.8+ MB
```

[4]:

```
# shape of the dataset

df.shape
```

[4]:

```
(30155, 25)
```

[5]:

```
# columns of the dataset as a list

df.columns
```

[5]:

```
Index(['id', 'date', 'price', 'bedrooms', 'bathrooms', 'sqft_living',
       'sqft_lot', 'floors', 'waterfront', 'greenbelt', 'nuisance', 'view',
       'condition', 'grade', 'heat_source', 'sewer_system', 'sqft_above',
       'sqft_basement', 'sqft_garage', 'sqft_patio', 'yr_built',
       'yr_renovated', 'address', 'lat', 'long'],
      dtype='object')
```

[6]:

```
# description of the dataset

df.describe()
```

[6]:

```
                 id         price      bedrooms     bathrooms   sqft_living  \
count  3.015500e+04  3.015500e+04  30155.000000  30155.000000  30155.000000
mean   4.538104e+09  1.108536e+06      3.413530      2.334737   2112.424739
std    2.882587e+09  8.963857e+05      0.981612      0.889556    974.044318
min    1.000055e+06  2.736000e+04      0.000000      0.000000      3.000000
```

```
25%    2.064175e+09  6.480000e+05      3.000000      2.000000   1420.000000
50%    3.874011e+09  8.600000e+05      3.000000      2.500000   1920.000000
75%    7.287100e+09  1.300000e+06      4.000000      3.000000   2619.500000
max    9.904000e+09  3.075000e+07     13.000000     10.500000  15360.000000

           sqft_lot        floors    sqft_above  sqft_basement   sqft_garage  \
count  3.015500e+04  30155.000000  30155.000000   30155.000000  30155.000000
mean   1.672360e+04      1.543492   1809.826098     476.039396    330.211142
std    6.038260e+04      0.567717    878.306131     579.631302    285.770536
min    4.020000e+02      1.000000      2.000000       0.000000      0.000000
25%    4.850000e+03      1.000000   1180.000000       0.000000      0.000000
50%    7.480000e+03      1.500000   1560.000000       0.000000    400.000000
75%    1.057900e+04      2.000000   2270.000000     940.000000    510.000000
max    3.253932e+06      4.000000  12660.000000    8020.000000   3580.000000

         sqft_patio      yr_built   yr_renovated           lat          long
count  30155.000000  30155.000000   30155.000000  30155.000000  30155.000000
mean     217.412038   1975.163953      90.922301     47.328076   -121.317397
std      245.302792     32.067362     416.473038      1.434005      5.725475
min        0.000000   1900.000000       0.000000     21.274240   -157.791480
25%       40.000000   1953.000000       0.000000     47.405320   -122.326045
50%      150.000000   1977.000000       0.000000     47.551380   -122.225585
75%      320.000000   2003.000000       0.000000     47.669913   -122.116205
max     4370.000000   2022.000000    2022.000000     64.824070    -70.074340
```

[7]:
```
df.head()
```

[7]:
```
           id        date     price  bedrooms  bathrooms  sqft_living  \
0  7399300360   5/24/2022  675000.0         4        1.0         1180
1  8910500230  12/13/2021  920000.0         5        2.5         2770
2  1180000275   9/29/2021  311000.0         6        2.0         2880
3  1604601802  12/14/2021  775000.0         3        3.0         2160
4  8562780790   8/24/2021  592500.0         2        2.0         1120

   sqft_lot  floors waterfront greenbelt  … sewer_system sqft_above  \
0      7140     1.0         NO        NO  …       PUBLIC       1180
1      6703     1.0         NO        NO  …       PUBLIC       1570
2      6156     1.0         NO        NO  …       PUBLIC       1580
3      1400     2.0         NO        NO  …       PUBLIC       1090
4       758     2.0         NO        NO  …       PUBLIC       1120

  sqft_basement sqft_garage sqft_patio yr_built  yr_renovated  \
0             0           0         40     1969             0
1          1570           0        240     1950             0
2          1580           0          0     1956             0
3          1070         200        270     2010             0
4           550         550         30     2012             0
```

```
                                             address        lat       long
0  2102 Southeast 21st Court, Renton, Washington …  47.461975 -122.19052
1  11231 Greenwood Avenue North, Seattle, Washing…  47.711525 -122.35591
2  8504 South 113th Street, Seattle, Washington 9…  47.502045 -122.22520
3  4079 Letitia Avenue South, Seattle, Washington…  47.566110 -122.29020
4  2193 Northwest Talus Drive, Issaquah, Washingt…  47.532470 -122.07188

[5 rows x 25 columns]
```

1.4.1 Data Cleaning

[8]:
```
df.drop(labels='id' , axis=1)
```

[8]:
```
             date      price  bedrooms  bathrooms  sqft_living  sqft_lot  \
0       5/24/2022   675000.0         4        1.0         1180      7140
1      12/13/2021   920000.0         5        2.5         2770      6703
2       9/29/2021   311000.0         6        2.0         2880      6156
3      12/14/2021   775000.0         3        3.0         2160      1400
4       8/24/2021   592500.0         2        2.0         1120       758
…           …        …       …        …          …       …
30150  11/30/2021  1555000.0         5        2.0         1910      4000
30151   6/16/2021  1313000.0         3        2.0         2020      5800
30152   5/27/2022   800000.0         3        2.0         1620      3600
30153   2/24/2022   775000.0         3        2.5         2570      2889
30154   4/29/2022   500000.0         3        1.5         1200     11058

       floors waterfront greenbelt nuisance  … sewer_system sqft_above  \
0         1.0         NO        NO       NO  …       PUBLIC       1180
1         1.0         NO        NO      YES  …       PUBLIC       1570
2         1.0         NO        NO       NO  …       PUBLIC       1580
3         2.0         NO        NO       NO  …       PUBLIC       1090
4         2.0         NO        NO      YES  …       PUBLIC       1120
…        …        …       …      …  …         …        …
30150     1.5         NO        NO       NO  …       PUBLIC       1600
30151     2.0         NO        NO       NO  …       PUBLIC       2020
30152     1.0         NO        NO      YES  …       PUBLIC        940
30153     2.0         NO        NO       NO  …       PUBLIC       1830
30154     1.0         NO        NO       NO  …       PUBLIC       1200

      sqft_basement sqft_garage sqft_patio  yr_built  yr_renovated  \
0                 0           0         40      1969             0
1              1570           0        240      1950             0
2              1580           0          0      1956             0
3              1070         200        270      2010             0
4               550         550         30      2012             0
…             …         …        …       …           …
```

```
30150          1130           0        210      1921             0
30151             0           0        520      2011             0
30152           920         240        110      1995             0
30153           740         480        100      2006             0
30154             0         420          0      1965             0

                                                 address       lat       long
0      2102 Southeast 21st Court, Renton, Washington …  47.461975 -122.19052
1      11231 Greenwood Avenue North, Seattle, Washing…  47.711525 -122.35591
2      8504 South 113th Street, Seattle, Washington 9…  47.502045 -122.22520
3      4079 Letitia Avenue South, Seattle, Washington…  47.566110 -122.29020
4      2193 Northwest Talus Drive, Issaquah, Washingt…  47.532470 -122.07188
…                                                   …        …          …
30150  4673 Eastern Avenue North, Seattle, Washington…  47.664740 -122.32940
30151  4131 44th Avenue Southwest, Seattle, Washingto…  47.565610 -122.38851
30152  910 Martin Luther King Jr Way, Seattle, Washin…  47.610395 -122.29585
30153  17127 114th Avenue Southeast, Renton, Washingt…  47.449490 -122.18908
30154  18615 7th Avenue South, Burien, Washington 981…  47.435840 -122.32634

[30155 rows x 24 columns]
```

[9]:
```
df = df.drop(labels='id' , axis=1)
```

[10]:
```
df.dtypes
```

[10]:
```
date              object
price            float64
bedrooms           int64
bathrooms        float64
sqft_living        int64
sqft_lot           int64
floors           float64
waterfront        object
greenbelt         object
nuisance          object
view              object
condition         object
grade             object
heat_source       object
sewer_system      object
sqft_above         int64
sqft_basement      int64
sqft_garage        int64
sqft_patio         int64
yr_built           int64
yr_renovated       int64
address           object
```

```
lat                float64
long               float64
dtype: object
```

[11]:
```
df['sale_yr'] = df.date.map(lambda x: '{}'.format(x[-4:]))
df.head(5)
```

[11]:
```
         date     price  bedrooms  bathrooms  sqft_living  sqft_lot  floors  \
0   5/24/2022  675000.0         4        1.0         1180      7140     1.0
1  12/13/2021  920000.0         5        2.5         2770      6703     1.0
2   9/29/2021  311000.0         6        2.0         2880      6156     1.0
3  12/14/2021  775000.0         3        3.0         2160      1400     2.0
4   8/24/2021  592500.0         2        2.0         1120       758     2.0

  waterfront greenbelt nuisance  … sqft_above sqft_basement sqft_garage  \
0         NO        NO       NO  …       1180             0           0
1         NO        NO      YES  …       1570          1570           0
2         NO        NO       NO  …       1580          1580           0
3         NO        NO       NO  …       1090          1070         200
4         NO        NO      YES  …       1120           550         550

  sqft_patio yr_built  yr_renovated  \
0         40     1969             0
1        240     1950             0
2          0     1956             0
3        270     2010             0
4         30     2012             0

                                             address        lat        long  \
0  2102 Southeast 21st Court, Renton, Washington …  47.461975 -122.19052
1  11231 Greenwood Avenue North, Seattle, Washing…  47.711525 -122.35591
2  8504 South 113th Street, Seattle, Washington 9…  47.502045 -122.22520
3  4079 Letitia Avenue South, Seattle, Washington…  47.566110 -122.29020
4  2193 Northwest Talus Drive, Issaquah, Washingt…  47.532470 -122.07188

  sale_yr
0    2022
1    2021
2    2021
3    2021
4    2021

[5 rows x 25 columns]
```

[12]:
```
df['sale_yr'] = df['sale_yr'].astype('int')
```

[13]:
```
df.dtypes
```

[13]:
```
date              object
price            float64
bedrooms           int64
bathrooms        float64
sqft_living        int64
sqft_lot           int64
floors           float64
waterfront        object
greenbelt         object
nuisance          object
view              object
condition         object
grade             object
heat_source       object
sewer_system      object
sqft_above         int64
sqft_basement      int64
sqft_garage        int64
sqft_patio         int64
yr_built           int64
yr_renovated       int64
address           object
lat              float64
long             float64
sale_yr            int64
dtype: object
```

[14]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30155 entries, 0 to 30154
Data columns (total 25 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   date           30155 non-null  object
 1   price          30155 non-null  float64
 2   bedrooms       30155 non-null  int64
 3   bathrooms      30155 non-null  float64
 4   sqft_living    30155 non-null  int64
 5   sqft_lot       30155 non-null  int64
 6   floors         30155 non-null  float64
 7   waterfront     30155 non-null  object
 8   greenbelt      30155 non-null  object
 9   nuisance       30155 non-null  object
 10  view           30155 non-null  object
 11  condition      30155 non-null  object
 12  grade          30155 non-null  object
 13  heat_source    30123 non-null  object
 14  sewer_system   30141 non-null  object
 15  sqft_above     30155 non-null  int64
 16  sqft_basement  30155 non-null  int64
 17  sqft_garage    30155 non-null  int64
 18  sqft_patio     30155 non-null  int64
 19  yr_built       30155 non-null  int64
 20  yr_renovated   30155 non-null  int64
 21  address        30155 non-null  object
 22  lat            30155 non-null  float64
 23  long           30155 non-null  float64
 24  sale_yr        30155 non-null  int64
dtypes: float64(5), int64(10), object(10)
memory usage: 5.8+ MB
```

[15]:

```
df['yr_old'] = np.where(df['yr_renovated'] != 0,df['sale_yr'].apply(lambda x:␣
 ↪x) - df['yr_renovated'],
                           df['sale_yr'].apply(lambda x: x) - df['yr_built'])
```

[16]:

```
df.head()
```

[16]:

```
         date     price  bedrooms  bathrooms  sqft_living  sqft_lot  floors  \
0   5/24/2022  675000.0         4        1.0         1180      7140     1.0
1  12/13/2021  920000.0         5        2.5         2770      6703     1.0
2   9/29/2021  311000.0         6        2.0         2880      6156     1.0
3  12/14/2021  775000.0         3        3.0         2160      1400     2.0
4   8/24/2021  592500.0         2        2.0         1120       758     2.0

  waterfront greenbelt nuisance  … sqft_basement sqft_garage sqft_patio  \
0         NO        NO       NO  …             0           0         40
1         NO        NO      YES  …          1570           0        240
2         NO        NO       NO  …          1580           0          0
3         NO        NO       NO  …          1070         200        270
4         NO        NO      YES  …           550         550         30

  yr_built yr_renovated                                            address  \
0     1969            0  2102 Southeast 21st Court, Renton, Washington …
1     1950            0  11231 Greenwood Avenue North, Seattle, Washing…
2     1956            0  8504 South 113th Street, Seattle, Washington 9…
3     2010            0  4079 Letitia Avenue South, Seattle, Washington…
4     2012            0  2193 Northwest Talus Drive, Issaquah, Washingt…

         lat        long  sale_yr  yr_old
0  47.461975 -122.19052     2022      53
1  47.711525 -122.35591     2021      71
2  47.502045 -122.22520     2021      65
3  47.566110 -122.29020     2021      11
```

```
4  47.532470 -122.07188     2021       9

[5 rows x 26 columns]
```

Adding the Zipcodes that is in the range of King County

```
[17]: df.address[0:5]
```

```
[17]: 0    2102 Southeast 21st Court, Renton, Washington …
      1    11231 Greenwood Avenue North, Seattle, Washing…
      2    8504 South 113th Street, Seattle, Washington 9…
      3    4079 Letitia Avenue South, Seattle, Washington…
      4    2193 Northwest Talus Drive, Issaquah, Washingt…
      Name: address, dtype: object
```

```
[18]: #zipcodes started at 98.....
      # it looks like every column has the same format and ending...
      # when working with strings, keep in mind that if the strings are not of equal␣
       ↪length
      df.address[1000][-20:-15]
```

```
[18]: '98019'
```

```
[19]: df.address[0].split(',')
```

```
[19]: ['2102 Southeast 21st Court', ' Renton', ' Washington 98055', ' United States']
```

```
[20]: df.address[0].split(',')[2][-5:]
```

```
[20]: '98055'
```

```
[21]: df['zipcode'] = df.address.apply(lambda x: x[-20:-15])
```

```
[22]: df['zipcode'].value_counts()
```

```
[22]: 98042    992
      98038    858
      98115    761
      98103    761
      98117    748
              …
      62204      1
      68862      1
      85207      1
      99202      1
      34470      1
      Name: zipcode, Length: 399, dtype: int64
```

[23]:

```
df['zipcode'] = df['zipcode'].astype(str)
```

[24]:

```
Zip_list = ['98042', '98038', '98103', '98115', '98117', '98023', '98133',␣
 ↪'98058',
       '98034', '98001', '98092', '98118', '98106', '98059', '98031', '98033',
       '98052', '98056', '98155', '98125', '98022', '98107', '98126', '98146',
       '98144', '98122', '98045', '98003', '98198', '98006']
```

[25]:

```
Filtered_df = df[df['zipcode'].isin(Zip_list)]
```

[26]:

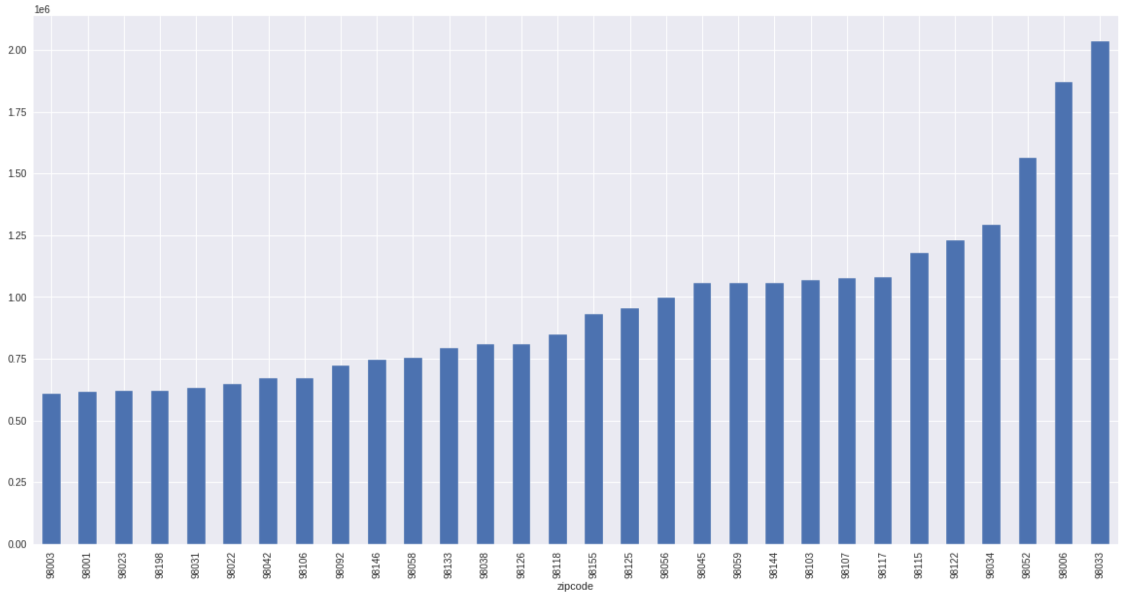
```
Filtered_df
```

[26]:

```
             date      price  bedrooms  bathrooms  sqft_living  sqft_lot  \
1      12/13/2021   920000.0         5        2.5         2770      6703
3      12/14/2021   775000.0         3        3.0         2160      1400
5       7/20/2021   625000.0         2        1.0         1190      5688
8       3/17/2022   780000.0         4        2.5         2340      8125
10       6/1/2022  1025000.0         3        1.5         2570      6379
…           …        …       …        …           …       …
30145  12/27/2021   705000.0         3        2.5         2260     50965
30147   2/28/2022   665000.0         3        2.5         2100      7210
30149   10/7/2021   719000.0         3        2.5         1270      1141
30150  11/30/2021  1555000.0         5        2.0         1910      4000
30152   5/27/2022   800000.0         3        2.0         1620      3600

       floors waterfront greenbelt nuisance  … sqft_garage sqft_patio  \
1         1.0         NO        NO      YES  …           0        240
3         2.0         NO        NO       NO  …         200        270
5         1.0         NO        NO      YES  …         300          0
8         2.0         NO        NO       NO  …         440         70
10        1.5         NO        NO      YES  …           0        250
…       …        …       …      …  …          …        …
30145     2.0         NO        NO       NO  …         480        200
30147     2.0         NO        NO       NO  …         440         40
30149     2.0         NO        NO       NO  …         200         60
30150     1.5         NO        NO       NO  …           0        210
30152     1.0         NO        NO      YES  …         240        110

      yr_built yr_renovated  \
1         1950            0
3         2010            0
5         1948            0
8         1989            0
10        1912            0
…        …          …
30145     1998            0
30147     1979            0
```

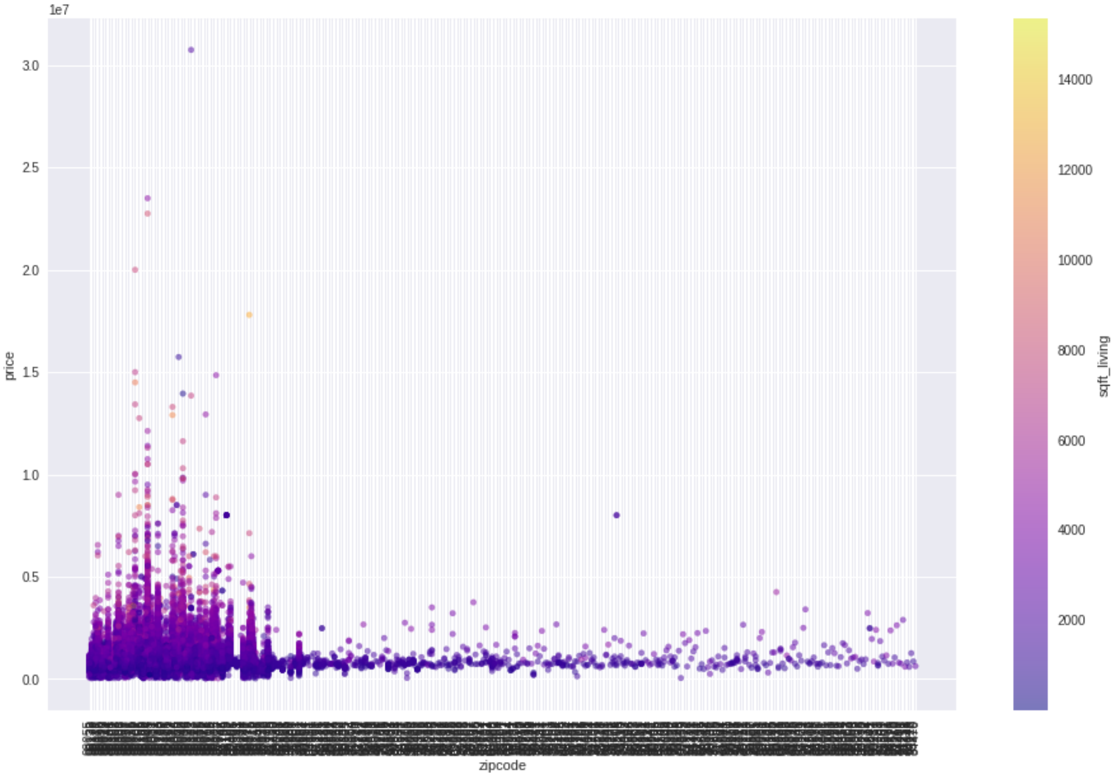
```
30149     2007            0
30150     1921            0
30152     1995            0

                                                address        lat  \
1      11231 Greenwood Avenue North, Seattle, Washing…  47.711525
3      4079 Letitia Avenue South, Seattle, Washington…  47.566110
5      1602 North 185th Street, Shoreline, Washington…  47.763470
8      2721 Southwest 343rd Place, Federal Way, Washi…  47.293770
10     3408 Beacon Avenue South, Seattle, Washington …  47.572760
…                                                  …         …
30145  46533 Southeast 156th Place, North Bend, Washi…  47.457410
30147  5218 South 302nd Place, Auburn, Washington 980…  47.331160
30149  8359 11th Avenue Northwest, Seattle, Washingto…  47.690440
30150  4673 Eastern Avenue North, Seattle, Washington…  47.664740
30152  910 Martin Luther King Jr Way, Seattle, Washin…  47.610395

             long  sale_yr  yr_old  zipcode
1     -122.355910     2021      71    98133
3     -122.290200     2021      11    98118
5     -122.340155     2021      73    98133
8     -122.369320     2022      33    98023
10    -122.308200     2022     110    98144
…               …      …     …      …
30145 -121.719630     2021      23    98045
30147 -122.268565     2022      43    98001
30149 -122.370620     2021      14    98117
30150 -122.329400     2021     100    98103
30152 -122.295850     2022      27    98122

[17570 rows x 27 columns]
```

```
[27]: plt.figure(figsize=(20,10))
      zip_graph =Filtered_df.groupby(Filtered_df.zipcode).price.mean().
       ↪sort_values(ascending=True)
      zip_graph.plot(kind='bar');
```



[28]:

```
# plotting the sqft_living and zipcode  and coding it according to price

df.plot(kind='scatter', x='zipcode', y='price',
        alpha=.5, figsize=(16,10), c='sqft_living', cmap='plasma', sharey=True,␣
 ↪sharex=False);

plt.xticks(rotation =90);
```



Note: House prices are clustered according to zipcode. Many factors and variables, tied into the zipcode, may influence the price either positively or negatively and we must be mindful of that.

1.5 Dropping missing values

[29]:
```
# remove missing values in these columns, make change permanent using␣
 ↪`inplace=True`
df.dropna(subset=['heat_source','sewer_system'], axis=0, inplace=True)
```

[30]:
```
df.isna().sum()/df.shape[0]
```

[30]:
```
date                 0.0
price                0.0
bedrooms             0.0
bathrooms            0.0
sqft_living          0.0
sqft_lot             0.0
floors               0.0
waterfront           0.0
greenbelt            0.0
nuisance             0.0
view                 0.0
```

```
condition        0.0
grade            0.0
heat_source      0.0
sewer_system     0.0
sqft_above       0.0
sqft_basement    0.0
sqft_garage      0.0
sqft_patio       0.0
yr_built         0.0
yr_renovated     0.0
address          0.0
lat              0.0
long             0.0
sale_yr          0.0
yr_old           0.0
zipcode          0.0
dtype: float64
```

[31]:

```
# quantity of null values for each column

df.isnull().sum().sort_values(ascending=False)
```

[31]:

```
date             0
sewer_system     0
yr_old           0
sale_yr          0
long             0
lat              0
address          0
yr_renovated     0
yr_built         0
sqft_patio       0
sqft_garage      0
sqft_basement    0
sqft_above       0
heat_source      0
price            0
grade            0
condition        0
view             0
nuisance         0
greenbelt        0
waterfront       0
floors           0
sqft_lot         0
sqft_living      0
bathrooms        0
```

```
bedrooms         0
zipcode          0
dtype: int64
```

[32]:

```
# unique values for sqft_basement column

df.sqft_basement.unique()
```

[32]:

```
array([   0, 1570, 1580, 1070,  550, 1560, 1100, 1310,  430,  660,  700,
        810,  860, 1250, 1220,  340, 1040, 1650, 2030,  930, 1030,  940,
       1400,  680,  300, 1230,  190,  830,  640, 1150,  990, 1740, 1810,
       1170, 1630, 1060,  470,  950,  500,  650,  780,  380,  530, 1240,
       1110, 2960, 1020,  600, 1380,  460, 1610, 1010, 1440,  670, 1500,
       1120,  750,  160,  390, 1280, 1530, 1090,  560,  720, 1200,  980,
        440,  630, 1360,  800,  610, 2070, 1450,  870,  250,  260,  320,
       1290,  740, 1340, 1300,  580,  730,  770,  900,  880,  400, 1410,
       1140,  669,  570,  710, 2590, 3140,  590, 1080, 1480, 1600,  920,
       1270,  840,  790,  850, 1330, 1430,  220,  410, 1180,  910,  382,
       2060, 1160, 1640,  450,  760,  420,  290, 2830, 1210,  960,  520,
        330,  350,  620,  310, 1460,  820, 1130, 1596,  510, 1510, 1490,
       2620,  480, 1550, 1800, 1390, 1000, 1370, 2460, 5350, 1690, 1870,
       1050,   80,  970,  690, 2740,  270, 1470, 1910, 1260, 1720,  962,
        525, 1620, 1840,  370,  360, 1860, 1420, 1590, 1540,  695, 2280,
       2640,  890, 1670, 1056, 1700,  280, 1970, 2800, 1770,  540, 2580,
       1940,  490, 2120, 1350,  130, 1850,  452,  200,  150, 2220, 1960,
       1950, 1520, 1190, 2147, 1780, 1320, 2210, 2200, 1930, 1820, 4520,
       1900, 2240,  100, 1760, 2540,  782,  602, 1495,  672,  170, 2750,
        576, 1392, 1730, 2050,  938,  230, 1880,  240,  180, 2720,  835,
        928, 1423,  943, 2380, 2770, 1920,  120, 3560,  110, 2020, 1790,
       2420, 2550, 2320, 1473, 1076, 1660, 1131, 1225, 3810, 1680,  552,
        968, 4000, 3150, 2170,  909, 2440,  210, 2010, 2510,  762, 3910,
       2190, 1710, 2390,  140, 2080, 1128, 2310, 2100,  474, 1890,  786,
       1750, 1466, 6970, 1830, 2230, 2110, 2360, 2130, 1333, 3320,  986,
        924, 3090,  387, 2262, 2610,  379, 1221, 2520, 3120, 1079, 1012,
        675, 3310, 1749,  429, 1605, 3750, 2300, 2560,  953,  608, 2090,
        404,  475,  472, 3060, 3960, 2450, 2330, 2660, 3220, 2480, 1508,
        768, 1174,  438, 2430, 2700, 1708, 1353, 2205, 3050, 3080, 3000,
       2177, 3710,   70, 2760, 1990, 2000, 2990,  454, 2140,  838,  892,
       1158,  988, 2340, 2870, 3160, 1906, 2160, 1168, 1003, 2040,  637,
        755,  476, 3410,  469,  532, 2500, 1166,  325,  374, 2680,  776,
        543,  736, 2569,  375,    1,  896, 1657, 1471, 2290,  508,  694,
        728, 3180,  733, 2250,  652, 2400,  766,  417,  775, 1832, 2670,
       1164, 1408, 1972,  888,   60, 1980,  442, 8020, 4420, 1156,  555,
        615, 1502,  557, 2470, 2176, 1874, 3280, 3110,  416,  265,  994,
        708, 2180,  535, 3700, 3640, 3500,  599, 1289, 1548, 2150, 3350,
        459, 2526, 3530, 2260,  493, 1302, 2270, 2350, 1963, 1686, 3200,
       3660,  512,  662,  946, 1179, 1412, 2570, 2940,  572, 1541, 1836,
        781,  902,  626, 4130,   90,  471,  933, 1245, 1118, 3390, 3590,
        746, 1704,  632, 3600, 3010, 1429,  504,  369, 2044, 2710, 2780,
        858,  812,  315,  627,  432, 1279, 1812, 1365])
```

[33]:

```
# remove missing values in these columns, make change permanent using␣
 ↪'inplace=True'
df.dropna(subset=['heat_source','sewer_system'], axis=0, inplace=True)
```

[34]:

```
#check percentage of missing data in columns
# sum of na values for each column, they should all be 0
df.isna().sum()/df.shape[0]
```

[34]:

```
date             0.0
price            0.0
bedrooms         0.0
bathrooms        0.0
sqft_living      0.0
sqft_lot         0.0
floors           0.0
waterfront       0.0
greenbelt        0.0
nuisance         0.0
view             0.0
condition        0.0
grade            0.0
heat_source      0.0
sewer_system     0.0
sqft_above       0.0
sqft_basement    0.0
sqft_garage      0.0
sqft_patio       0.0
yr_built         0.0
yr_renovated     0.0
address          0.0
lat              0.0
long             0.0
sale_yr          0.0
yr_old           0.0
zipcode          0.0
dtype: float64
```

[35]:

```
df.shape
```

[35]:

```
(30111, 27)
```

1.6 Exploring Data

[36]:
```
# displaying head and tail of final dataset

display(df.head())
display(df.tail())
```

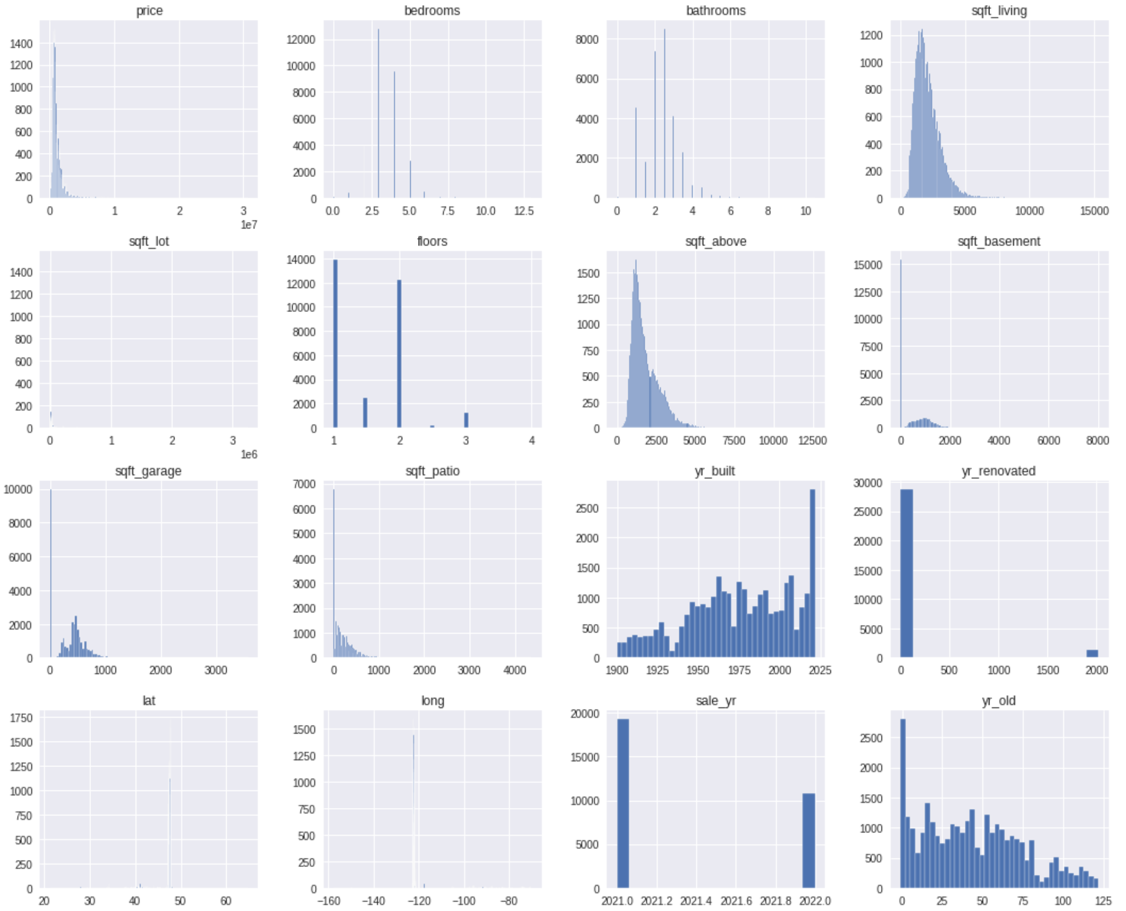
```
         date     price  bedrooms  bathrooms  sqft_living  sqft_lot  floors  \
0   5/24/2022  675000.0         4        1.0         1180      7140     1.0   
1  12/13/2021  920000.0         5        2.5         2770      6703     1.0   
2   9/29/2021  311000.0         6        2.0         2880      6156     1.0   
3  12/14/2021  775000.0         3        3.0         2160      1400     2.0   
4   8/24/2021  592500.0         2        2.0         1120       758     2.0   

  waterfront greenbelt nuisance  ... sqft_garage sqft_patio yr_built  \
0         NO        NO       NO  ...           0         40     1969   
1         NO        NO      YES  ...           0        240     1950   
2         NO        NO       NO  ...           0          0     1956   
3         NO        NO       NO  ...         200        270     2010   
4         NO        NO      YES  ...         550         30     2012   

  yr_renovated                                            address        lat  \
0            0  2102 Southeast 21st Court, Renton, Washington ...  47.461975   
1            0  11231 Greenwood Avenue North, Seattle, Washing...  47.711525   
2            0  8504 South 113th Street, Seattle, Washington 9...  47.502045   
3            0  4079 Letitia Avenue South, Seattle, Washington...  47.566110   
4            0  2193 Northwest Talus Drive, Issaquah, Washingt...  47.532470   

        long  sale_yr  yr_old  zipcode  
0 -122.19052     2022      53    98055  
1 -122.35591     2021      71    98133  
2 -122.22520     2021      65    98178  
3 -122.29020     2021      11    98118  
4 -122.07188     2021       9    98027  

[5 rows x 27 columns]
```

```
             date      price  bedrooms  bathrooms  sqft_living  sqft_lot  \
30150  11/30/2021  1555000.0         5        2.0         1910      4000   
30151   6/16/2021  1313000.0         3        2.0         2020      5800   
30152   5/27/2022   800000.0         3        2.0         1620      3600   
30153   2/24/2022   775000.0         3        2.5         2570      2889   
30154   4/29/2022   500000.0         3        1.5         1200     11058   

       floors waterfront greenbelt nuisance  ... sqft_garage sqft_patio  \
30150     1.5         NO        NO       NO  ...           0        210   
30151     2.0         NO        NO       NO  ...           0        520   
```

```
30152     1.0          NO        NO      YES  ...         240        110
30153     2.0          NO        NO       NO  ...         480        100
30154     1.0          NO        NO       NO  ...         420          0

      yr_built yr_renovated  \
30150     1921            0
30151     2011            0
30152     1995            0
30153     2006            0
30154     1965            0

                                                 address        lat  \
30150  4673 Eastern Avenue North, Seattle, Washington...  47.664740
30151  4131 44th Avenue Southwest, Seattle, Washingto...  47.565610
30152  910 Martin Luther King Jr Way, Seattle, Washin...  47.610395
30153  17127 114th Avenue Southeast, Renton, Washingt...  47.449490
30154  18615 7th Avenue South, Burien, Washington 981...  47.435840

           long  sale_yr  yr_old  zipcode
30150 -122.32940     2021     100    98103
30151 -122.38851     2021      10    98116
30152 -122.29585     2022      27    98122
30153 -122.18908     2022      16    98055
30154 -122.32634     2022      57    98148

[5 rows x 27 columns]
```

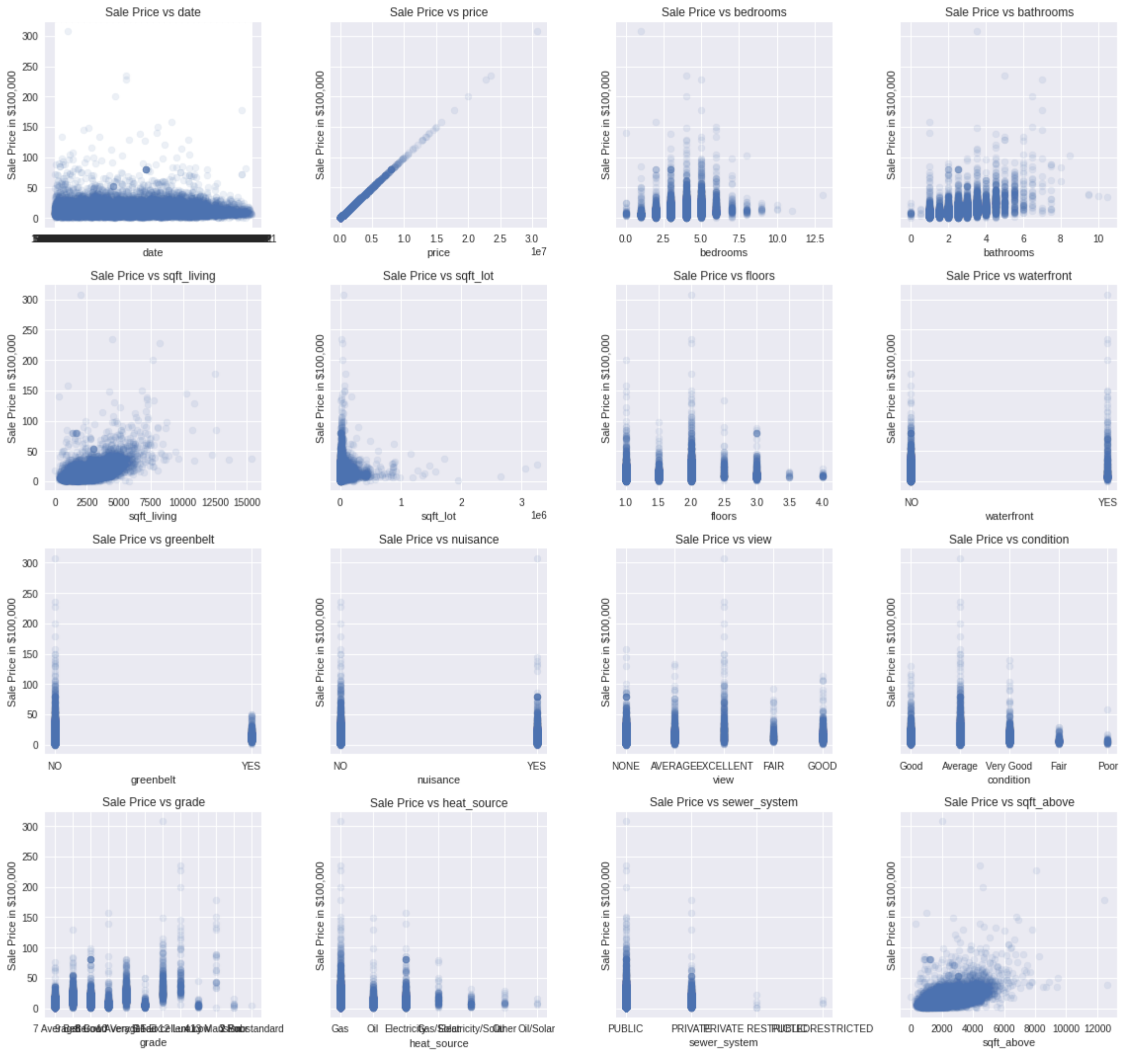
[37]:

```
# histograms across all columns

df.hist(figsize=(18,15), bins='auto');
```



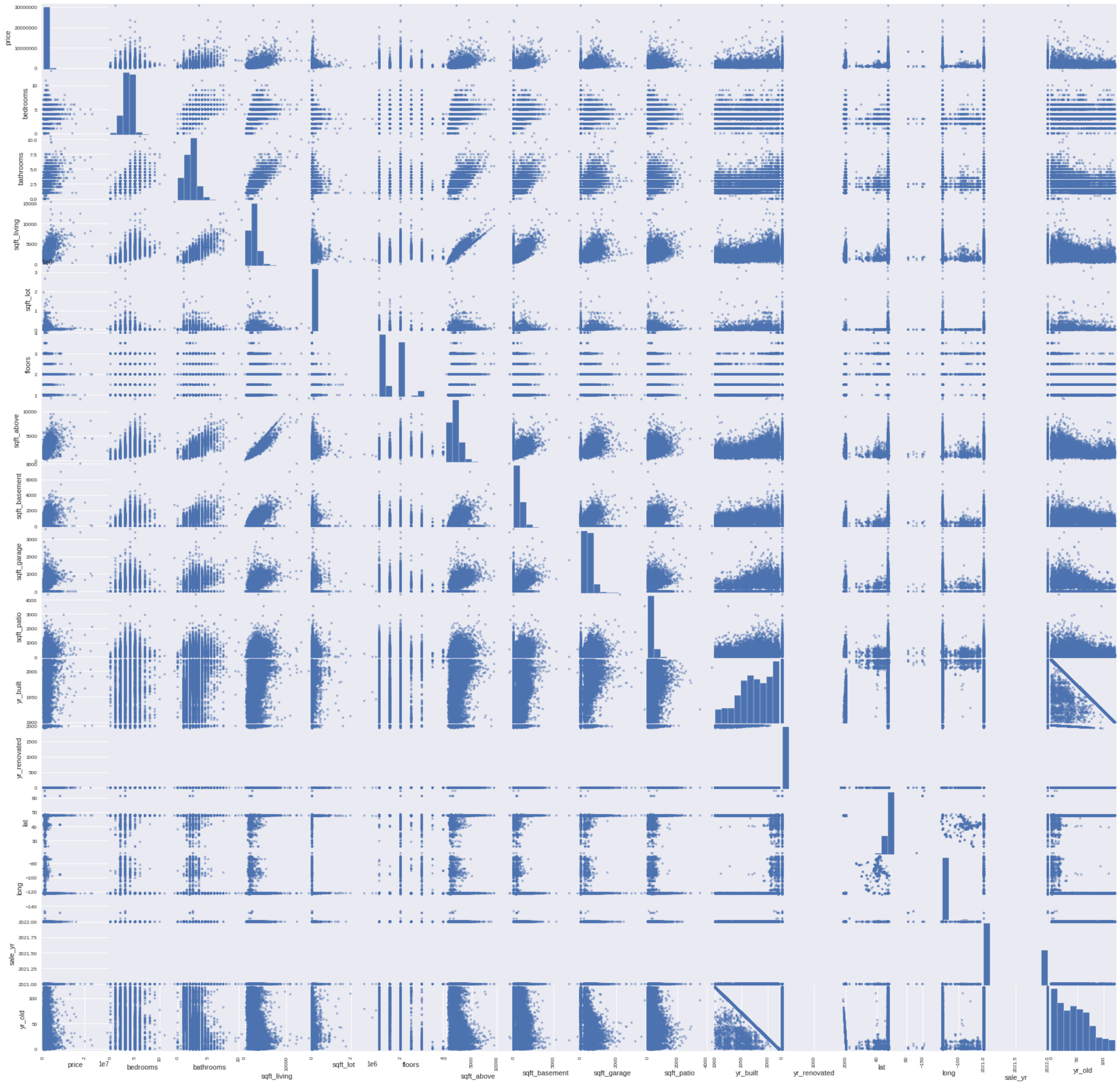
[38]:

```python
# scatterplots across all columns

fig, axes = plt.subplots(nrows=4, ncols=4, figsize=(16,15), sharey=True)

for ax, column in zip(axes.flatten(), df.columns):
    ax.scatter(df[column], df['price'] / 100_000, label=column, alpha=.1)
    ax.set_title(f'Sale Price vs {column}')
    ax.set_xlabel(column)
    ax.set_ylabel('Sale Price in $100,000')

fig.tight_layout()
```



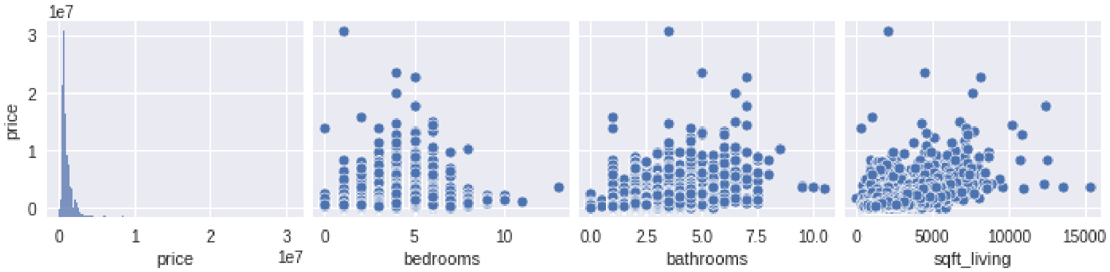
Scatter Matrix:

[39]:
```
# scatter matrix plotting every feature against each other

pd.plotting.scatter_matrix(df, figsize = [30,30]);
plt.show()
```



[40]:

```
# pairplot of certain features from the dataset vs. price

sns.pairplot(data=df, x_vars=['price','bedrooms','bathrooms','sqft_living'],␣
 ↪y_vars=['price']);
```



1.6.1 Exploring Main Columns

```
[41]: df.columns
```

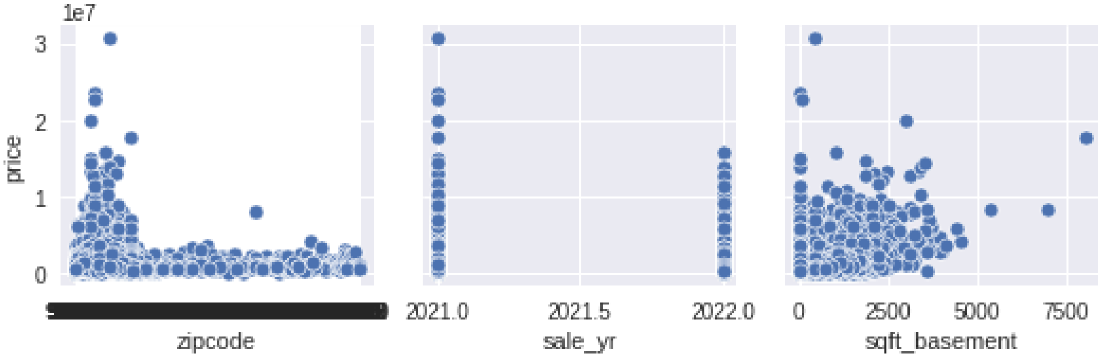
```
[41]: Index(['date', 'price', 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot',
             'floors', 'waterfront', 'greenbelt', 'nuisance', 'view', 'condition',
             'grade', 'heat_source', 'sewer_system', 'sqft_above', 'sqft_basement',
             'sqft_garage', 'sqft_patio', 'yr_built', 'yr_renovated', 'address',
             'lat', 'long', 'sale_yr', 'yr_old', 'zipcode'],
            dtype='object')
```

```
[42]: df.head()
```

```
[42]:          date     price  bedrooms  bathrooms  sqft_living  sqft_lot  floors  \
      0   5/24/2022  675000.0         4        1.0         1180      7140     1.0
      1  12/13/2021  920000.0         5        2.5         2770      6703     1.0
      2   9/29/2021  311000.0         6        2.0         2880      6156     1.0
      3  12/14/2021  775000.0         3        3.0         2160      1400     2.0
      4   8/24/2021  592500.0         2        2.0         1120       758     2.0

        waterfront greenbelt nuisance  … sqft_garage sqft_patio yr_built  \
      0         NO        NO       NO  …           0         40     1969
      1         NO        NO      YES  …           0        240     1950
      2         NO        NO       NO  …           0          0     1956
      3         NO        NO       NO  …         200        270     2010
      4         NO        NO      YES  …         550         30     2012

        yr_renovated                                            address        lat  \
      0            0  2102 Southeast 21st Court, Renton, Washington …  47.461975
      1            0  11231 Greenwood Avenue North, Seattle, Washing…  47.711525
      2            0  8504 South 113th Street, Seattle, Washington 9…  47.502045
      3            0  4079 Letitia Avenue South, Seattle, Washington…  47.566110
      4            0  2193 Northwest Talus Drive, Issaquah, Washingt…  47.532470

              long  sale_yr  yr_old  zipcode
      0 -122.19052     2022      53    98055
      1 -122.35591     2021      71    98133
      2 -122.22520     2021      65    98178
      3 -122.29020     2021      11    98118
      4 -122.07188     2021       9    98027

      [5 rows x 27 columns]
```

```
[43]: # pairplot of certain features from the dataset vs. price
```

```
sns.pairplot(data=df, x_vars=['zipcode','sale_yr','sqft_basement'],␣
 ↪y_vars=['price']);
```

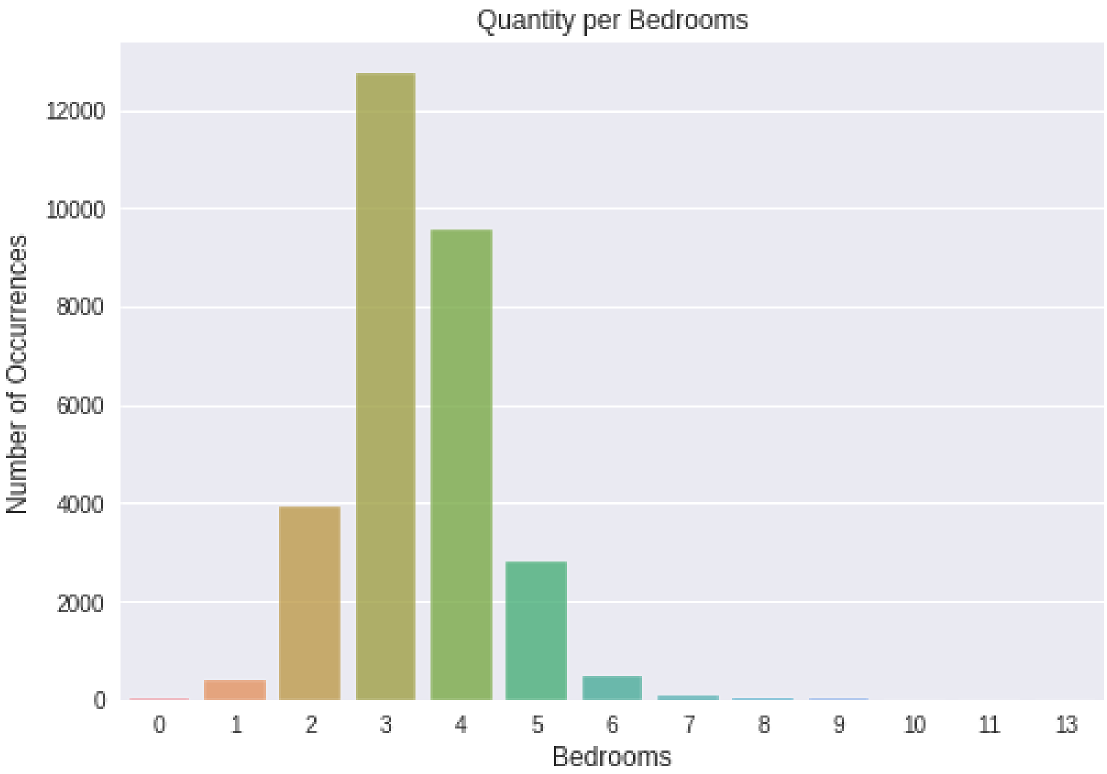


Bedrooms column

[44]:
```
# value counts for bedrooms in sorting them in descending order

df.bedrooms.value_counts().sort_values(ascending=False)
```

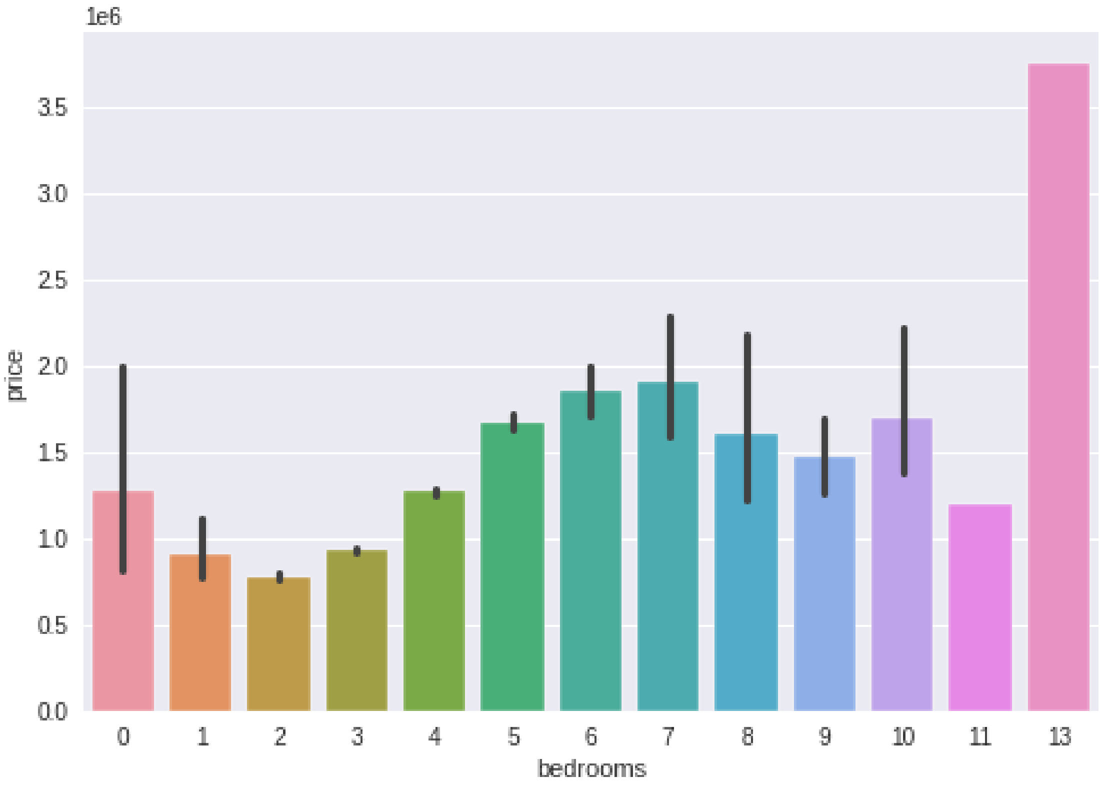
[44]:
```
3     12746
4      9591
2      3925
5      2794
6       498
1       381
7        80
0        39
8        38
9        14
10        3
11        1
13        1
Name: bedrooms, dtype: int64
```

[45]:
```
# barplot of bedrooms vs. number of occurrences

bedrooms  = df['bedrooms'].value_counts()
sns.barplot(bedrooms.index, bedrooms.values, alpha=0.8)
plt.title('Quantity per Bedrooms')
plt.ylabel('Number of Occurrences', fontsize=12)
plt.xlabel('Bedrooms', fontsize=12)
plt.show()
```



[46]:

```
# barplot of bedrooms vs. price

sns.barplot(x="bedrooms", y="price", data=df);
```



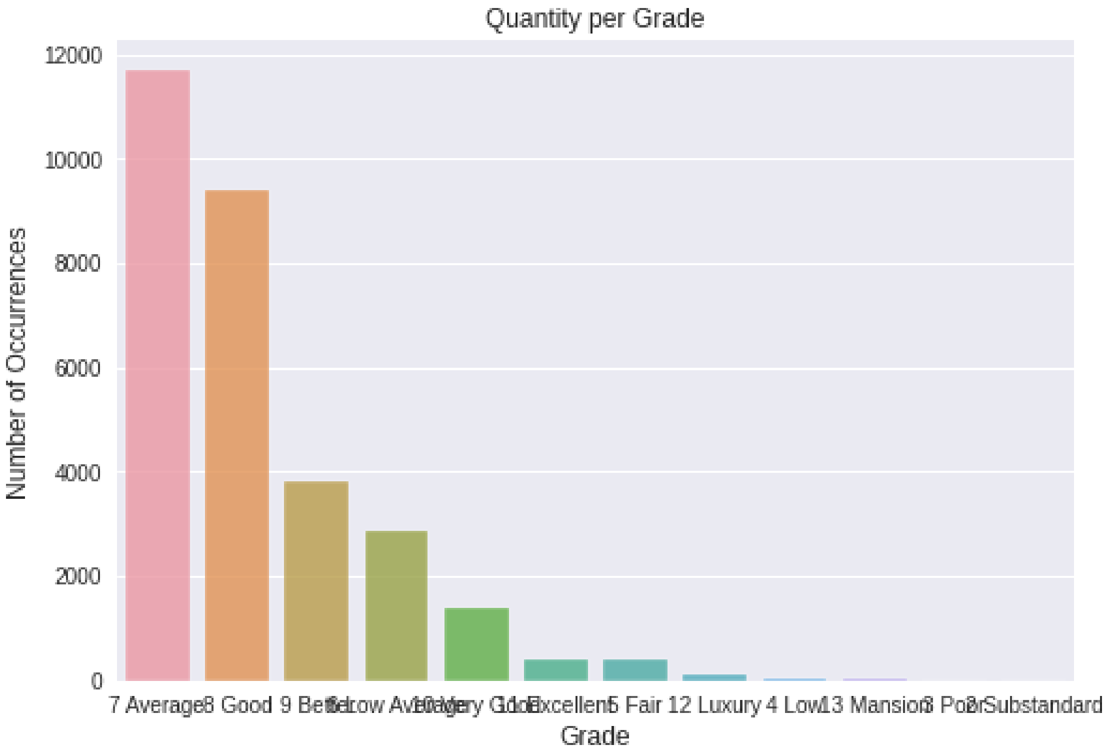
Grade Column

[47]:
```
# value counts for grades and sorting them in descending order

df.grade.value_counts().sort_values(ascending=False)
```

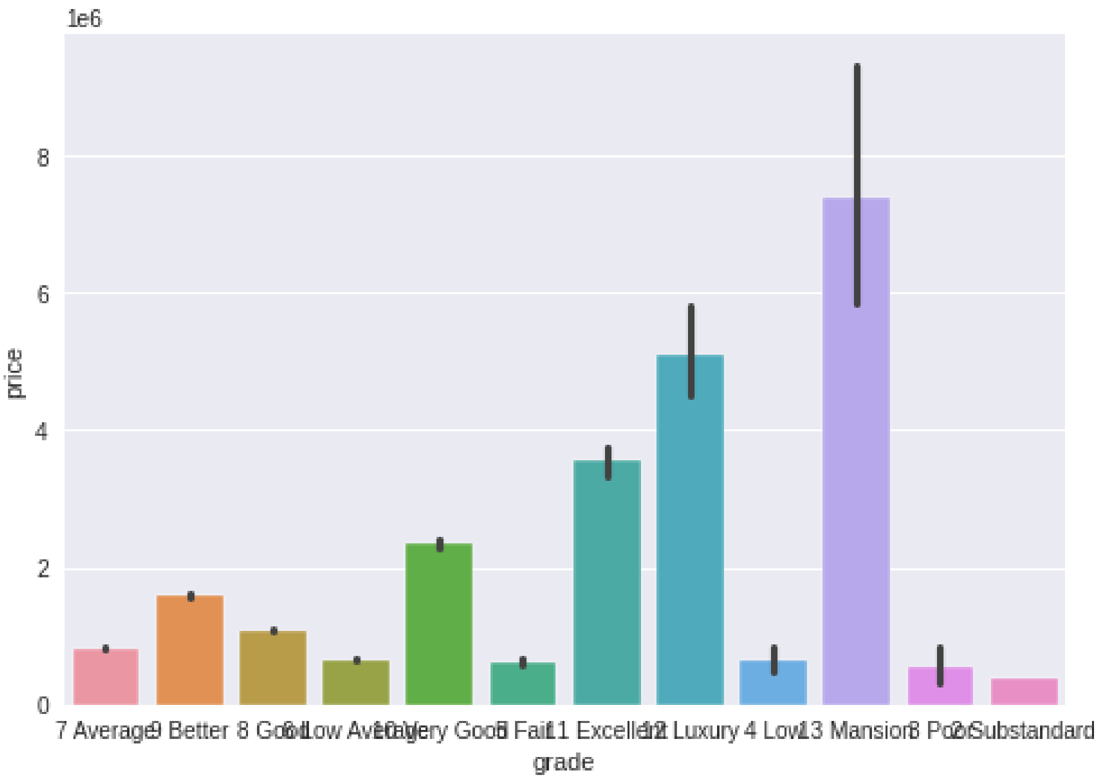
[47]:
```
7 Average        11693
8 Good            9400
9 Better          3804
6 Low Average     2852
10 Very Good      1369
11 Excellent       406
5 Fair             385
12 Luxury          122
4 Low               46
13 Mansion          24
3 Poor               9
2 Substandard        1
Name: grade, dtype: int64
```

[48]:
```
# bar graph of grade vs. number of occurrences
```

```
grades  = df['grade'].value_counts()
sns.barplot(grades.index, grades.values, alpha=0.8)
plt.title('Quantity per Grade')
plt.ylabel('Number of Occurrences', fontsize=12)
plt.xlabel('Grade', fontsize=12)
plt.show()
```



[49]:
```
# barplot of grade vs. price

sns.barplot(x="grade", y="price", data=df);
```



Bathrooms column

```
[50]: # value counts for bathrooms and sorting them in descending order

      df.bathrooms.value_counts().sort_values(ascending=False)
```

```
[50]: 2.5     8471
      2.0     7343
      1.0     4556
      3.0     4116
      3.5     2264
      1.5     1807
      4.0      645
      4.5      531
      5.0      145
      5.5      102
      6.0       45
      6.5       25
      0.0       25
      7.0       12
      7.5       12
      0.5        5
```

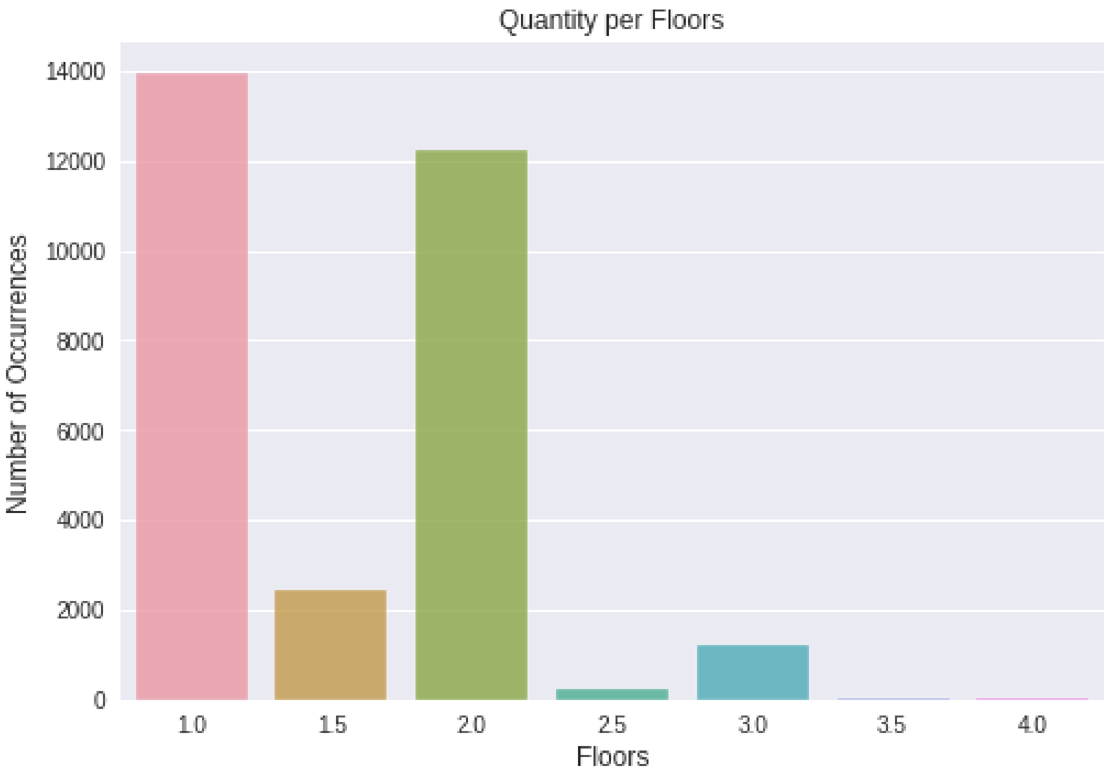
```
9.5       2
8.0       2
10.5      1
10.0      1
8.5       1
Name: bathrooms, dtype: int64
```

[51]:
```
# value counts for floors and sorting them in descending order

df.floors.value_counts().sort_values(ascending=False)
```

[51]:
```
1.0    13943
2.0    12246
1.5     2434
3.0     1221
2.5      222
4.0       30
3.5       15
Name: floors, dtype: int64
```

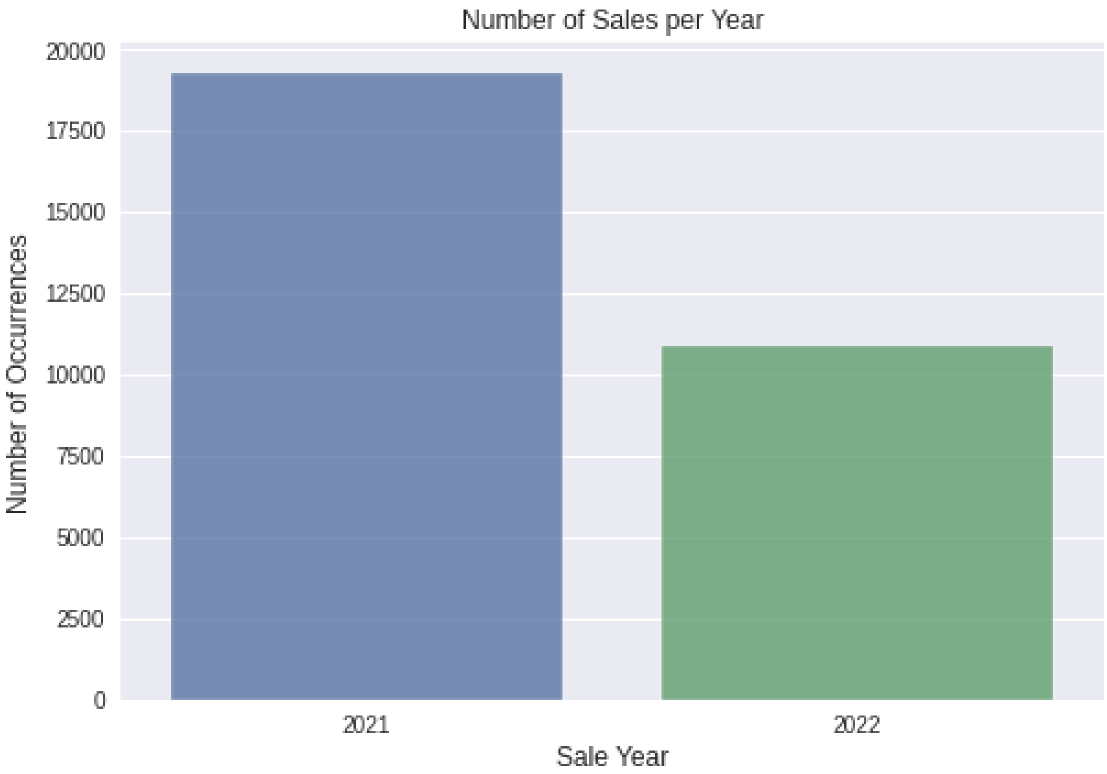
Sale_Yr column:

[52]:
```
# barplot of floors vs. number of occurrences

floors  = df['floors'].value_counts()
sns.barplot(floors.index, floors.values, alpha=0.8)
plt.title('Quantity per Floors')
plt.ylabel('Number of Occurrences', fontsize=12)
plt.xlabel('Floors', fontsize=12)
plt.show()
```

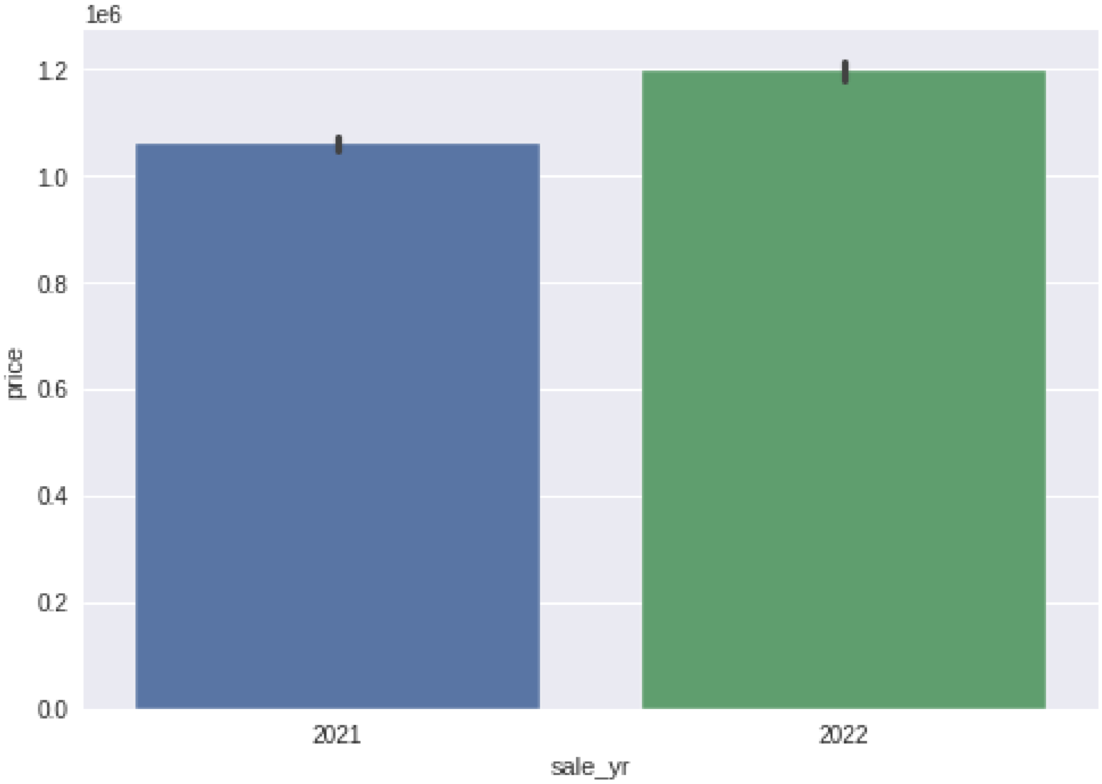


[53]:

```
# barplot of sale_yr vs. number of occurrences

sale_yr  = df['sale_yr'].value_counts()
sns.barplot(sale_yr.index, sale_yr.values, alpha=0.8)
plt.title('Number of Sales per Year')
plt.ylabel('Number of Occurrences', fontsize=12)
plt.xlabel('Sale Year', fontsize=12)
plt.show()
```



[54]:
```
# barplot of sale_yr vs. price

sns.barplot(x="sale_yr", y="price", data=df);
```

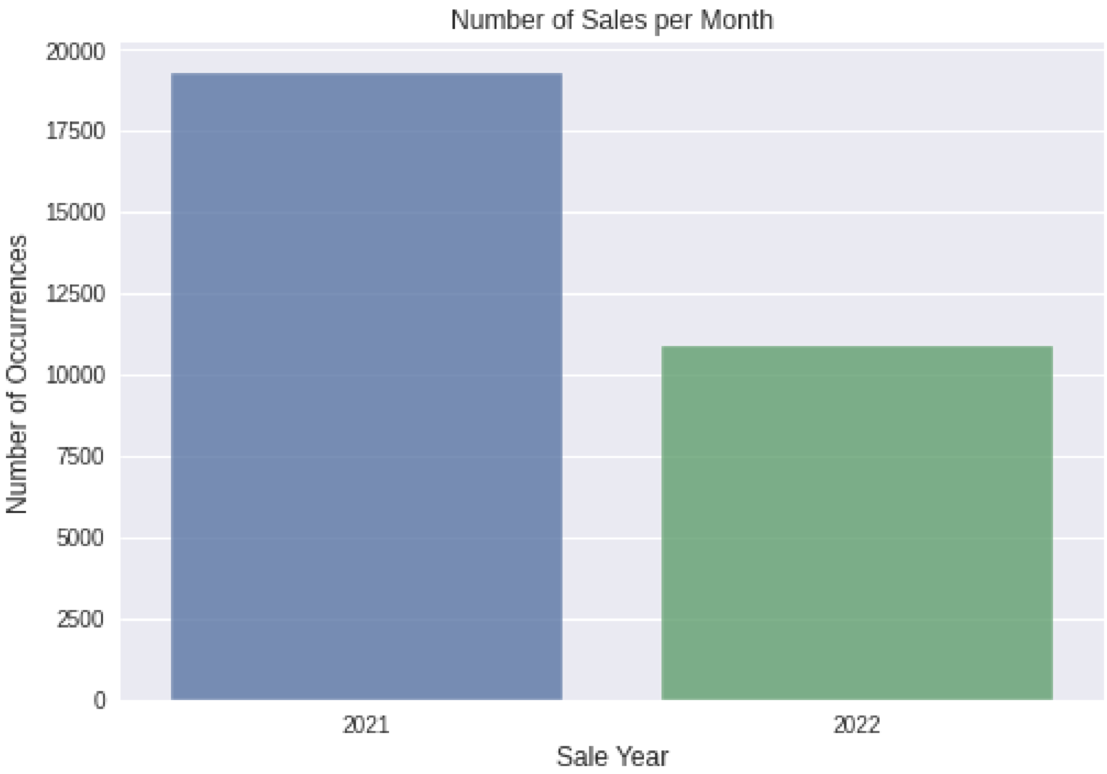


Sale Year Column:

```
[55]: # value counts of sale_yr column and sorting them in descending order

      df.sale_yr.value_counts().sort_values(ascending=False)
```

```
[55]: 2021    19261
      2022    10850
      Name: sale_yr, dtype: int64
```

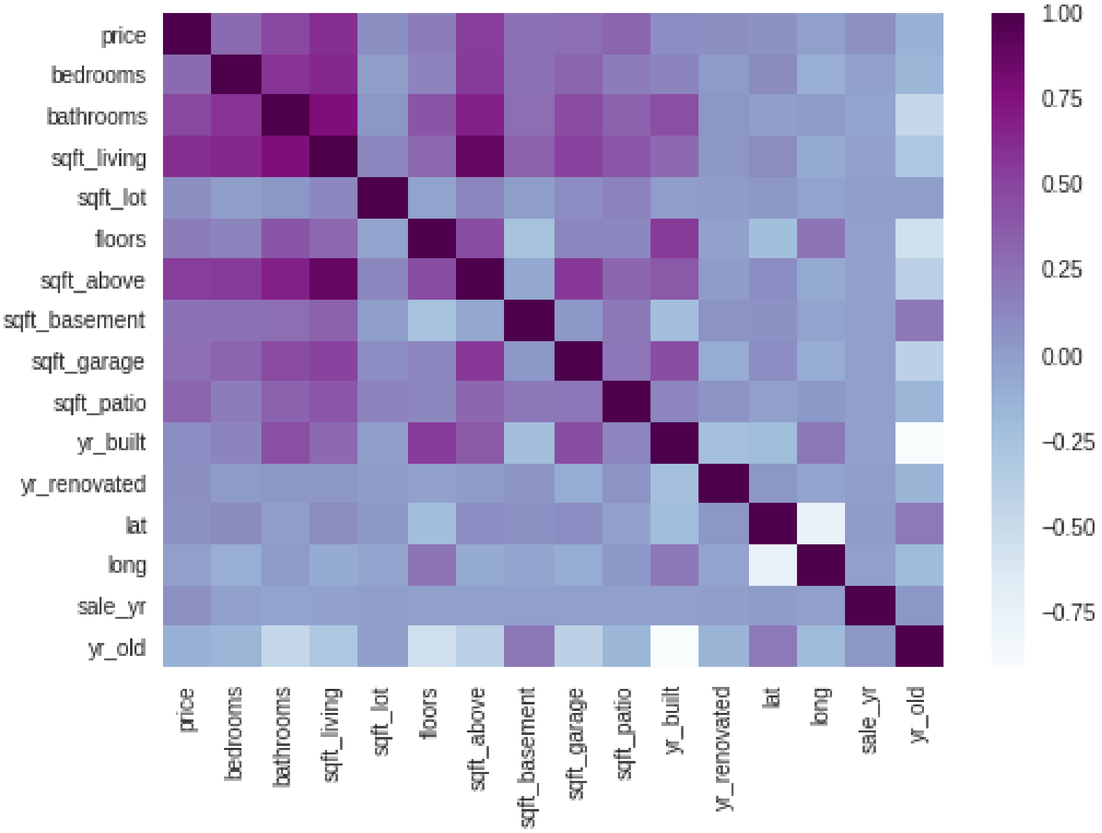
```
[56]: # barplot of sale_month vs. number of occurrences

      sale_month  = df['sale_yr'].value_counts()
      sns.barplot(sale_month.index, sale_month.values, alpha=0.8)
      plt.title('Number of Sales per Month')
      plt.ylabel('Number of Occurrences', fontsize=12)
      plt.xlabel('Sale Year', fontsize=12)
      plt.show()
```



1.6.2 Correlation Visualizations

[57]:

```
# correlational heatmap comparing all features of the dataset

sns.heatmap(df.corr(), cmap="BuPu");
```



```
[58]: # correlational values comparing all features
      df.corr()
```

```
[58]:                   price  bedrooms  bathrooms  sqft_living  sqft_lot    floors  \
      price          1.000000  0.288954   0.480337     0.608616  0.086550  0.180589
      bedrooms       0.288954  1.000000   0.588035     0.637048  0.006215  0.146871
      bathrooms      0.480337  0.588035   1.000000     0.772226  0.038028  0.404291
      sqft_living    0.608616  0.637048   0.772226     1.000000  0.122271  0.303911
      sqft_lot       0.086550  0.006215   0.038028     0.122271  1.000000 -0.031555
      floors         0.180589  0.146871   0.404291     0.303911 -0.031555  1.000000
      sqft_above     0.538631  0.546221   0.674239     0.883733  0.131756  0.448245
      sqft_basement  0.245005  0.237957   0.260684     0.338387  0.004457 -0.248466
      sqft_garage    0.263674  0.318110   0.456264     0.510967  0.089318  0.132363
      sqft_patio     0.313789  0.183660   0.327982     0.396530  0.154575  0.125016
      yr_built       0.095796  0.145497   0.443379     0.291242  0.001897  0.544314
      yr_renovated   0.085023  0.015369   0.041574     0.039089  0.009390 -0.025041
      lat            0.063430  0.108883  -0.005481     0.102205  0.030041 -0.218174
      long          -0.022278 -0.106791   0.017684    -0.087625 -0.034408  0.233589
      sale_yr        0.073904 -0.027387  -0.042125    -0.029198 -0.004733 -0.017305
      yr_old        -0.126909 -0.156650  -0.471854    -0.312269 -0.003427 -0.552862
```

```
               sqft_above  sqft_basement  sqft_garage  sqft_patio  yr_built  \
price            0.538631       0.245005     0.263674    0.313789  0.095796
bedrooms         0.546221       0.237957     0.318110    0.183660  0.145497
bathrooms        0.674239       0.260684     0.456264    0.327982  0.443379
sqft_living      0.883733       0.338387     0.510967    0.396530  0.291242
sqft_lot         0.131756       0.004457     0.089318    0.154575  0.001897
floors           0.448245      -0.248466     0.132363    0.125016  0.544314
sqft_above       1.000000      -0.067306     0.559972    0.312593  0.387253
sqft_basement   -0.067306       1.000000     0.025766    0.210305 -0.230783
sqft_garage      0.559972       0.025766     1.000000    0.216512  0.447720
sqft_patio       0.312593       0.210305     0.216512    1.000000  0.138112
yr_built         0.387253      -0.230783     0.447720    0.138112  1.000000
yr_renovated     0.011036       0.054032    -0.098301    0.056183 -0.239466
lat              0.092317       0.059664     0.092092   -0.019666 -0.207133
long            -0.082722      -0.045104    -0.096639    0.025675  0.209842
sale_yr         -0.023131      -0.009571    -0.012821   -0.016531 -0.023375
yr_old          -0.397502       0.211054    -0.409075   -0.157426 -0.912768

               yr_renovated        lat       long    sale_yr     yr_old
price              0.085023   0.063430  -0.022278   0.073904  -0.126909
bedrooms           0.015369   0.108883  -0.106791  -0.027387  -0.156650
bathrooms          0.041574  -0.005481   0.017684  -0.042125  -0.471854
sqft_living        0.039089   0.102205  -0.087625  -0.029198  -0.312269
sqft_lot           0.009390   0.030041  -0.034408  -0.004733  -0.003427
floors            -0.025041  -0.218174   0.233589  -0.017305  -0.552862
sqft_above         0.011036   0.092317  -0.082722  -0.023131  -0.397502
sqft_basement      0.054032   0.059664  -0.045104  -0.009571   0.211054
sqft_garage       -0.098301   0.092092  -0.096639  -0.012821  -0.409075
sqft_patio         0.056183  -0.019666   0.025675  -0.016531  -0.157426
yr_built          -0.239466  -0.207133   0.209842  -0.023375  -0.912768
yr_renovated       1.000000   0.036880  -0.035598  -0.001741  -0.144820
lat                0.036880   1.000000  -0.760532   0.010180   0.197614
long              -0.035598  -0.760532   1.000000  -0.011211  -0.200985
sale_yr           -0.001741   0.010180  -0.011211   1.000000   0.041987
yr_old            -0.144820   0.197614  -0.200985   0.041987   1.000000
```

[59]:

```
# correlational map with levels of precision

corr = df.corr()
corr.style.background_gradient(cmap='viridis').set_precision(3)
```

[59]:

```
<pandas.io.formats.style.Styler at 0x7f0f02ec6280>
```

Note: The Grade given by King county seems to be very influential after looking at the correlation visualizations.

1.7 Modeling:

Dealing with the Outliers:

[60]:

```
# looking at the head of the dataframe for a final check of the model
df.head()
```

[60]:

```
         date     price  bedrooms  bathrooms  sqft_living  sqft_lot  floors  \
0   5/24/2022  675000.0         4        1.0         1180      7140     1.0
1  12/13/2021  920000.0         5        2.5         2770      6703     1.0
2   9/29/2021  311000.0         6        2.0         2880      6156     1.0
3  12/14/2021  775000.0         3        3.0         2160      1400     2.0
4   8/24/2021  592500.0         2        2.0         1120       758     2.0

  waterfront greenbelt nuisance  … sqft_garage sqft_patio yr_built  \
0         NO        NO       NO  …           0         40     1969
1         NO        NO      YES  …           0        240     1950
2         NO        NO       NO  …           0          0     1956
3         NO        NO       NO  …         200        270     2010
4         NO        NO      YES  …         550         30     2012

  yr_renovated                                            address        lat  \
0            0  2102 Southeast 21st Court, Renton, Washington …  47.461975
1            0  11231 Greenwood Avenue North, Seattle, Washing…  47.711525
2            0  8504 South 113th Street, Seattle, Washington 9…  47.502045
3            0  4079 Letitia Avenue South, Seattle, Washington…  47.566110
4            0  2193 Northwest Talus Drive, Issaquah, Washingt…  47.532470

        long  sale_yr  yr_old  zipcode
0 -122.19052     2022      53    98055
1 -122.35591     2021      71    98133
2 -122.22520     2021      65    98178
3 -122.29020     2021      11    98118
4 -122.07188     2021       9    98027

[5 rows x 27 columns]
```

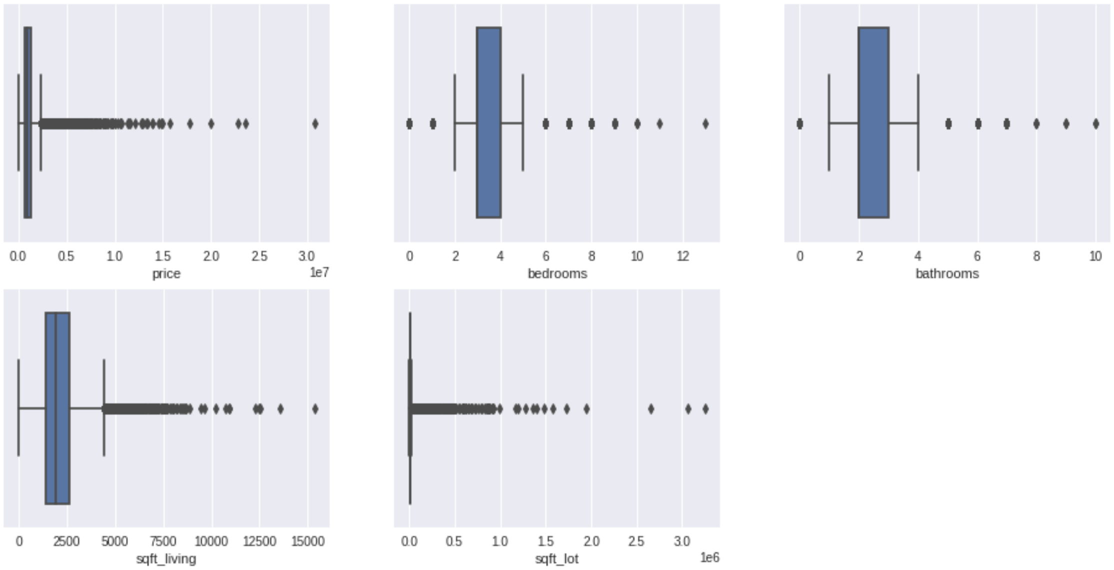
[61]:

```
# info of final dataset

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 30111 entries, 0 to 30154
Data columns (total 27 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   date           30111 non-null  object
 1   price          30111 non-null  float64
 2   bedrooms       30111 non-null  int64
 3   bathrooms      30111 non-null  float64
 4   sqft_living    30111 non-null  int64
 5   sqft_lot       30111 non-null  int64
 6   floors         30111 non-null  float64
 7   waterfront     30111 non-null  object
 8   greenbelt      30111 non-null  object
 9   nuisance       30111 non-null  object
 10  view           30111 non-null  object
 11  condition      30111 non-null  object
 12  grade          30111 non-null  object
 13  heat_source    30111 non-null  object
 14  sewer_system   30111 non-null  object
 15  sqft_above     30111 non-null  int64
 16  sqft_basement  30111 non-null  int64
 17  sqft_garage    30111 non-null  int64
 18  sqft_patio     30111 non-null  int64
 19  yr_built       30111 non-null  int64
 20  yr_renovated   30111 non-null  int64
 21  address        30111 non-null  object
 22  lat            30111 non-null  float64
 23  long           30111 non-null  float64
 24  sale_yr        30111 non-null  int64
 25  yr_old         30111 non-null  int64
 26  zipcode        30111 non-null  object
dtypes: float64(5), int64(11), object(11)
memory usage: 7.4+ MB
```

[62]:

```python
# boxplots on certain features that contain a great deal of outliers

plt.figure(figsize=(16,12))
plt.subplot(331)
sns.boxplot(df.price)
plt.subplot(332)
sns.boxplot(df.bedrooms)
plt.subplot(333)
sns.boxplot(df.bathrooms.astype('int'))
plt.subplot(334)
sns.boxplot(df.sqft_living)
plt.subplot(335)
sns.boxplot(df.sqft_lot);
```



[63]:
```
# create a filter that has only the numerical columns of the dataset
pred_cols = [x for x in df.columns if x not in␣
 ↪['selldate','price','waterfront','greenbelt','nuisance','view','condition','grade','heat_so
pred_cols
```

[63]:
```
['date',
 'bedrooms',
 'bathrooms',
 'sqft_living',
 'sqft_lot',
 'floors',
 'sqft_above',
 'sqft_basement',
 'sqft_garage',
 'sqft_patio',
 'yr_built',
 'yr_renovated',
 'lat',
 'long',
 'sale_yr',
 'yr_old']
```

[64]:
```
pred_cols = ['bedrooms',
 'bathrooms',
 'sqft_living',
 'sqft_lot',
 'floors',
 'sqft_above',
 'sqft_basement',
 'sqft_garage',
 'sqft_patio',
 'yr_built',
 'sale_yr',
 'yr_old',
 'zipcode']
```

```
[65]: Filtered_df
```

```
[65]:             date      price  bedrooms  bathrooms  sqft_living  sqft_lot  \
1       12/13/2021   920000.0         5        2.5         2770      6703
3       12/14/2021   775000.0         3        3.0         2160      1400
5        7/20/2021   625000.0         2        1.0         1190      5688
8        3/17/2022   780000.0         4        2.5         2340      8125
10        6/1/2022  1025000.0         3        1.5         2570      6379
…            …          …        …        …          …           …
30145   12/27/2021   705000.0         3        2.5         2260     50965
30147    2/28/2022   665000.0         3        2.5         2100      7210
30149    10/7/2021   719000.0         3        2.5         1270      1141
30150   11/30/2021  1555000.0         5        2.0         1910      4000
30152    5/27/2022   800000.0         3        2.0         1620      3600

       floors waterfront greenbelt nuisance  … sqft_garage sqft_patio  \
1         1.0         NO        NO      YES  …           0        240
3         2.0         NO        NO       NO  …         200        270
5         1.0         NO        NO      YES  …         300          0
8         2.0         NO        NO       NO  …         440         70
10        1.5         NO        NO      YES  …           0        250
…       …        …        …      …  …         …        …
30145     2.0         NO        NO       NO  …         480        200
30147     2.0         NO        NO       NO  …         440         40
30149     2.0         NO        NO       NO  …         200         60
30150     1.5         NO        NO       NO  …           0        210
30152     1.0         NO        NO      YES  …         240        110

      yr_built yr_renovated  \
1         1950            0
3         2010            0
5         1948            0
8         1989            0
10        1912            0
…        …          …
30145     1998            0
30147     1979            0
30149     2007            0
30150     1921            0
```

```
30152     1995            0

                                                     address        lat  \
1      11231 Greenwood Avenue North, Seattle, Washing…  47.711525
3      4079 Letitia Avenue South, Seattle, Washington…  47.566110
5      1602 North 185th Street, Shoreline, Washington…  47.763470
8      2721 Southwest 343rd Place, Federal Way, Washi…  47.293770
10     3408 Beacon Avenue South, Seattle, Washington …  47.572760
…                                                 …          …
30145  46533 Southeast 156th Place, North Bend, Washi…  47.457410
30147  5218 South 302nd Place, Auburn, Washington 980…  47.331160
30149  8359 11th Avenue Northwest, Seattle, Washingto…  47.690440
30150  4673 Eastern Avenue North, Seattle, Washington…  47.664740
30152  910 Martin Luther King Jr Way, Seattle, Washin…  47.610395

             long  sale_yr  yr_old  zipcode
1     -122.355910     2021      71    98133
3     -122.290200     2021      11    98118
5     -122.340155     2021      73    98133
8     -122.369320     2022      33    98023
10    -122.308200     2022     110    98144
…             …      …       …       …
30145 -121.719630     2021      23    98045
30147 -122.268565     2022      43    98001
30149 -122.370620     2021      14    98117
30150 -122.329400     2021     100    98103
30152 -122.295850     2022      27    98122

[17570 rows x 27 columns]
```

[66]:
```python
# apply the filter we created to our dataset, assign the model features to␣
 ↪'preds' and assign price to 'target'.
preds = Filtered_df[pred_cols]
target = Filtered_df.price
preds = pd.get_dummies(preds, columns=['zipcode'], drop_first=True)
```

[67]:
```python
# create baseline model predictor df and target
y= target
X= preds

model = sm.OLS(y, sm.add_constant(X))
results= model.fit()
```

[68]:
```python
results.summary()
```

[68]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.531
Model:                            OLS   Adj. R-squared:                  0.530
Method:                 Least Squares   F-statistic:                     483.9
Date:                Sun, 02 Oct 2022   Prob (F-statistic):               0.00
Time:                        02:27:44   Log-Likelihood:            -2.5349e+05
No. Observations:               17570   AIC:                         5.071e+05
Df Residuals:                   17528   BIC:                         5.074e+05
Df Model:                          41
Covariance Type:            nonrobust
================================================================================
=
                    coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
-
const         -2.675e+08   1.42e+07    -18.821      0.000   -2.95e+08
-2.4e+08
bedrooms      -6.881e+04   4872.019    -14.123      0.000   -7.84e+04
-5.93e+04
bathrooms      5.768e+04   7096.661      8.127      0.000    4.38e+04
7.16e+04
sqft_living     223.2195     16.628     13.424      0.000     190.626
255.813
sqft_lot          0.6769      0.060     11.205      0.000       0.558
0.795
floors        -6.176e+04   9722.290     -6.353      0.000   -8.08e+04
-4.27e+04
sqft_above      183.0324     16.859     10.856      0.000     149.986
216.078
sqft_basement    38.5500     12.662      3.045      0.002      13.732
63.368
sqft_garage      72.1712     17.928      4.026      0.000      37.030
107.312
sqft_patio      202.9465     16.617     12.213      0.000     170.375
235.518
yr_built      -1854.8044    277.819     -6.676      0.000   -2399.358
-1310.251
sale_yr        1.342e+05   7036.368     19.068      0.000     1.2e+05
1.48e+05
yr_old        -1139.6758    277.625     -4.105      0.000   -1683.849
-595.503
zipcode_98003  3.316e+04   2.78e+04      1.191      0.234   -2.14e+04
8.77e+04
zipcode_98006  9.378e+05   2.68e+04     34.968      0.000    8.85e+05
9.9e+05
```

```
zipcode_98022 -1.027e+04   2.78e+04     -0.370      0.712   -6.47e+04
4.42e+04
zipcode_98023 -1.504e+04   2.47e+04     -0.609      0.543   -6.35e+04
3.34e+04
zipcode_98031  4.687e+04   2.64e+04      1.774      0.076   -4929.354
9.87e+04
zipcode_98033  1.213e+06   2.56e+04     47.337      0.000    1.16e+06
1.26e+06
zipcode_98034  6.546e+05   2.48e+04     26.435      0.000    6.06e+05
7.03e+05
zipcode_98038  9.052e+04   2.36e+04      3.834      0.000    4.42e+04
1.37e+05
zipcode_98042  2442.6679   2.29e+04      0.107      0.915   -4.24e+04
4.73e+04
zipcode_98045  2.466e+05   2.75e+04      8.983      0.000    1.93e+05
3e+05
zipcode_98052  8.075e+05    2.6e+04     31.044      0.000    7.57e+05
8.59e+05
zipcode_98056  3.547e+05   2.66e+04     13.347      0.000    3.03e+05
4.07e+05
zipcode_98058  1.048e+05   2.48e+04      4.232      0.000    5.63e+04
1.53e+05
zipcode_98059  2.945e+05   2.58e+04     11.417      0.000    2.44e+05
3.45e+05
zipcode_98092 -4.479e+04   2.56e+04     -1.751      0.080   -9.49e+04
5335.920
zipcode_98103  6.227e+05   2.56e+04     24.317      0.000    5.73e+05
6.73e+05
zipcode_98106  2.679e+05   2.68e+04      9.986      0.000    2.15e+05
3.21e+05
zipcode_98107  6.258e+05   2.88e+04     21.747      0.000    5.69e+05
6.82e+05
zipcode_98115  6.402e+05   2.52e+04     25.446      0.000    5.91e+05
6.9e+05
zipcode_98117  6.002e+05   2.55e+04     23.561      0.000     5.5e+05
6.5e+05
zipcode_98118  3.675e+05   2.63e+04     13.959      0.000    3.16e+05
4.19e+05
zipcode_98122  7.218e+05   2.94e+04     24.585      0.000    6.64e+05
7.79e+05
zipcode_98125  4.444e+05   2.74e+04     16.203      0.000    3.91e+05
4.98e+05
zipcode_98126  3.805e+05   2.86e+04     13.284      0.000    3.24e+05
4.37e+05
zipcode_98133  3.403e+05   2.53e+04     13.456      0.000    2.91e+05
3.9e+05
zipcode_98144   5.57e+05   2.92e+04     19.090      0.000       5e+05
```

```
6.14e+05
zipcode_98146   2.86e+05   2.84e+04     10.073      0.000     2.3e+05
3.42e+05
zipcode_98155  3.917e+05   2.69e+04     14.582      0.000    3.39e+05
4.44e+05
zipcode_98198  9.302e+04   2.82e+04      3.297      0.001    3.77e+04
1.48e+05
==============================================================================
Omnibus:                    39918.472   Durbin-Watson:                   1.865
Prob(Omnibus):                  0.000   Jarque-Bera (JB):        959358946.422
Skew:                          21.171   Prob(JB):                         0.00
Kurtosis:                    1146.966   Cond. No.                     2.54e+08
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 2.54e+08. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

Note:the r^2 value which gives the accuracy of the model. its at 1146.96. Cleaning the data, feature engineering and including the dummified categorical variables should improve the r2.

1.8 Model Itiration

[69]:

```
# getting rid of those outliers to drive our linear regression

Filtered_df2 = Filtered_df[Filtered_df.price < 4000000]
Filtered_df = Filtered_df[Filtered_df.bedrooms < 5]
Filtered_df = Filtered_df[Filtered_df.bathrooms < 4]
Filtered_df = Filtered_df[Filtered_df.sqft_living < 8000]
Filtered_df = Filtered_df[Filtered_df.sqft_lot < 500000]
```

[70]:

```
# creating copies of the dataframe which will be used in the trial linear␣
 ↪regressions

trial_df1 = df.copy()
trial_df2 = df.copy()
trial_df3 = df.copy()
```

[71]:

```
df.dtypes
```

[71]:

```
date              object
price            float64
bedrooms           int64
bathrooms        float64
```

```
sqft_living        int64
sqft_lot           int64
floors           float64
waterfront        object
greenbelt         object
nuisance          object
view              object
condition         object
grade             object
heat_source       object
sewer_system      object
sqft_above         int64
sqft_basement      int64
sqft_garage        int64
sqft_patio         int64
yr_built           int64
yr_renovated       int64
address           object
lat              float64
long             float64
sale_yr            int64
yr_old             int64
zipcode           object
dtype: object
```

[72]:

```
# Take a look at the value_counts for our categorical variables. Consider how␣
 ↪some of the entries might be reformatted.
# for example, condition can be altered to take on values of above average,␣
 ↪average and below average...
df[['waterfront','greenbelt','nuisance','view','condition','grade']].
 ↪value_counts()
```

[72]:

```
waterfront  greenbelt  nuisance  view     condition  grade
NO          NO         NO        NONE     Average    8 Good                4863
                                                     7 Average             4448
                                          Good       7 Average             2931
                                          Average    9 Better              2083
                                          Good       8 Good                1447
                                                                           …
                       YES       AVERAGE  Poor       7 Average                1
                                                     6 Low Average            1
YES         NO         NO        AVERAGE  Very Good  9 Better                 1
NO          NO         YES       AVERAGE  Poor       5 Fair                   1
YES         YES        NO        AVERAGE  Good       12 Luxury                1
Length: 470, dtype: int64
```

[73]:
```
#use pd.get_dummies to dummify categorical variables
cat_columns =␣
 ↪['waterfront','greenbelt','nuisance','view','condition','grade','heat_source','sewer_system
dummy_df = pd.get_dummies(data=df, columns=cat_columns, drop_first=True)
```

[74]:
```
dummy_df.columns
```

[74]:
```
Index(['date', 'price', 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot',
       'floors', 'sqft_above', 'sqft_basement', 'sqft_garage', 'sqft_patio',
       'yr_built', 'yr_renovated', 'address', 'lat', 'long', 'sale_yr',
       'yr_old', 'zipcode', 'waterfront_YES', 'greenbelt_YES', 'nuisance_YES',
       'view_EXCELLENT', 'view_FAIR', 'view_GOOD', 'view_NONE',
       'condition_Fair', 'condition_Good', 'condition_Poor',
       'condition_Very Good', 'grade_11 Excellent', 'grade_12 Luxury',
       'grade_13 Mansion', 'grade_2 Substandard', 'grade_3 Poor',
       'grade_4 Low', 'grade_5 Fair', 'grade_6 Low Average', 'grade_7 Average',
       'grade_8 Good', 'grade_9 Better', 'heat_source_Electricity/Solar',
       'heat_source_Gas', 'heat_source_Gas/Solar', 'heat_source_Oil',
       'heat_source_Oil/Solar', 'heat_source_Other',
       'sewer_system_PRIVATE RESTRICTED', 'sewer_system_PUBLIC',
       'sewer_system_PUBLIC RESTRICTED'],
      dtype='object')
```

1.9 Baseline Model

1.9.1 Model Trial 1

[75]:
```
# dealing with all the categorical features from the dataset

trial_df1.grade = trial_df1.grade.astype('category')
trial_df1.zipcode = trial_df1.zipcode.astype('category')
```

[76]:
```
trial_df1.columns
```

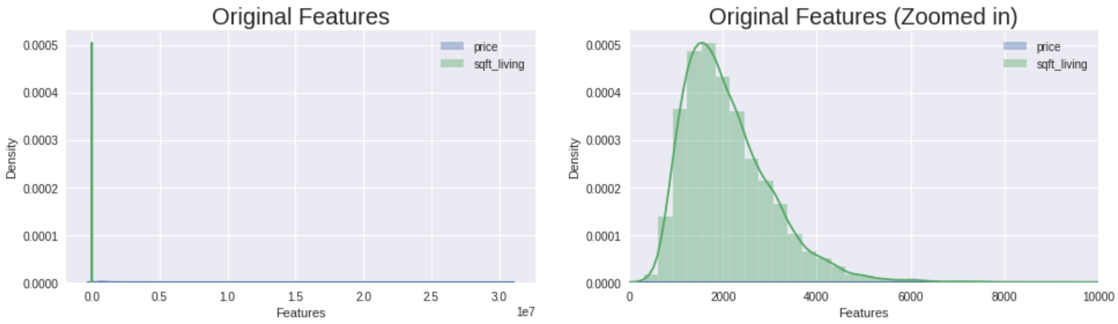
[76]:
```
Index(['date', 'price', 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot',
       'floors', 'waterfront', 'greenbelt', 'nuisance', 'view', 'condition',
       'grade', 'heat_source', 'sewer_system', 'sqft_above', 'sqft_basement',
       'sqft_garage', 'sqft_patio', 'yr_built', 'yr_renovated', 'address',
       'lat', 'long', 'sale_yr', 'yr_old', 'zipcode'],
      dtype='object')
```

[77]:
```
# making dummies for all the categorical features
grade = pd.get_dummies(trial_df1.grade, prefix='grade', drop_first=True)
zipcode = pd.get_dummies(trial_df1.grade, prefix='zipcode', drop_first=True)
```

[78]:
```
# adding dummies to the dataset and removing the original features
```

```
trial_df1 = trial_df1.join([grade, zipcode])
trial_df1.drop(['grade', 'zipcode'], axis=1, inplace=True)
```

[79]:
```
# displots on the continuous features from the dataset

plt.figure(figsize=(16,4))
plt.subplot(121)
sns.distplot(trial_df2.price, label='price')
sns.distplot(trial_df2.sqft_living, label='sqft_living')
plt.title('Original Features', fontdict={'fontsize': 20})
plt.xlabel('Features')
plt.legend()

plt.subplot(122)
sns.distplot(trial_df2.price, label='price')
sns.distplot(trial_df2.sqft_living, label='sqft_living')
plt.title('Original Features (Zoomed in)', fontdict={'fontsize': 20})
plt.xlabel('Features')
plt.xlim(0, 10000)
plt.legend()
plt.show()
```



[80]:
```
trial_df1['price_1'] = ( trial_df1['price'] - trial_df1['price'].min() ) / ( ␣
 ↪trial_df1['price'].max() - trial_df1['price'].min() )
trial_df1['sqft_living_1'] = ( trial_df1['sqft_living'] -␣
 ↪trial_df1['sqft_living'].min() ) / (trial_df1['sqft_living'].max() -␣
 ↪trial_df1['sqft_living'].min() )
```

[81]:
```
trial_df1
```

[81]:
```
          date     price  bedrooms  bathrooms  sqft_living  sqft_lot  \
0    5/24/2022  675000.0         4        1.0         1180      7140
1   12/13/2021  920000.0         5        2.5         2770      6703
2    9/29/2021  311000.0         6        2.0         2880      6156
3   12/14/2021  775000.0         3        3.0         2160      1400
```

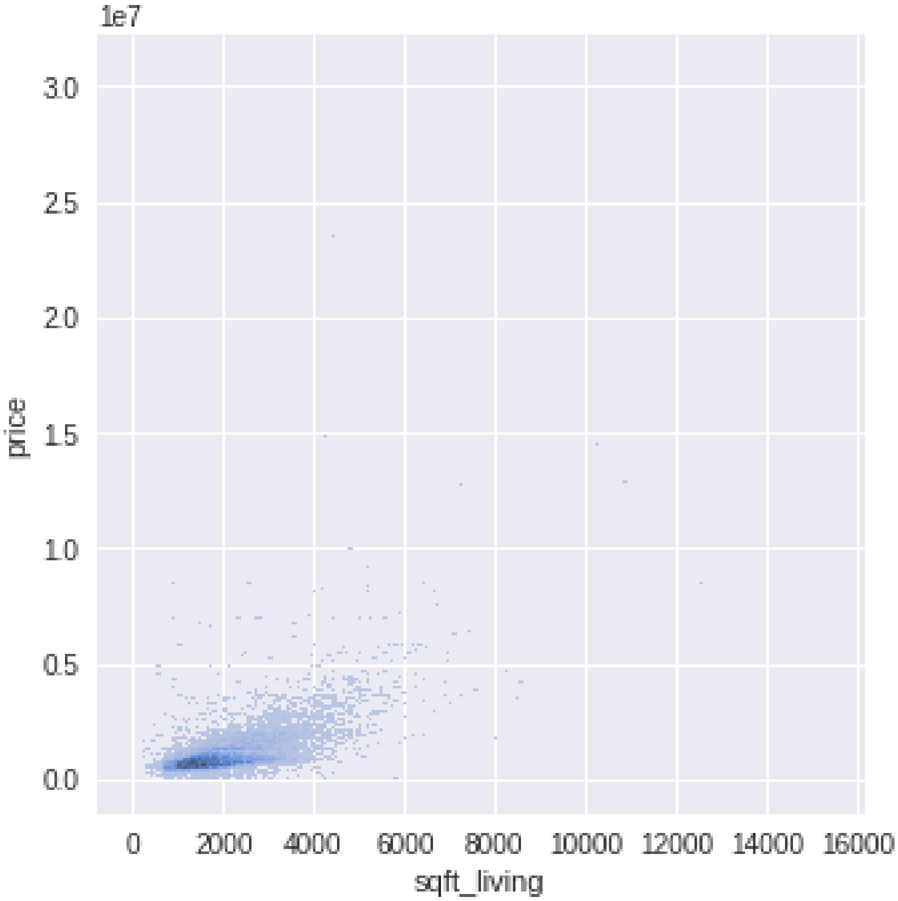
```
4       8/24/2021   592500.0         2        2.0         1120       758
…           …        …       …        …          …       …
30150  11/30/2021  1555000.0         5        2.0         1910      4000
30151   6/16/2021  1313000.0         3        2.0         2020      5800
30152   5/27/2022   800000.0         3        2.0         1620      3600
30153   2/24/2022   775000.0         3        2.5         2570      2889
30154   4/29/2022   500000.0         3        1.5         1200     11058

       floors waterfront greenbelt nuisance  … zipcode_2 Substandard  \
0          1.0         NO        NO       NO  …                     0
1          1.0         NO        NO      YES  …                     0
2          1.0         NO        NO       NO  …                     0
3          2.0         NO        NO       NO  …                     0
4          2.0         NO        NO      YES  …                     0
…       …        …       …      …  …                   …
30150      1.5         NO        NO       NO  …                     0
30151      2.0         NO        NO       NO  …                     0
30152      1.0         NO        NO      YES  …                     0
30153      2.0         NO        NO       NO  …                     0
30154      1.0         NO        NO       NO  …                     0

      zipcode_3 Poor zipcode_4 Low zipcode_5 Fair  zipcode_6 Low Average  \
0                  0             0              0                      0
1                  0             0              0                      0
2                  0             0              0                      0
3                  0             0              0                      0
4                  0             0              0                      0
…              …           …            …                    …
30150              0             0              0                      0
30151              0             0              0                      0
30152              0             0              0                      0
30153              0             0              0                      0
30154              0             0              0                      0

       zipcode_7 Average  zipcode_8 Good  zipcode_9 Better   price_1  \
0                      1               0                 0  0.021080
1                      1               0                 0  0.029055
2                      1               0                 0  0.009232
3                      0               0                 1  0.024335
4                      1               0                 0  0.018395
…                  …             …               …       …
30150                  0               1                 0  0.049724
30151                  1               0                 0  0.041847
30152                  1               0                 0  0.025149
30153                  0               1                 0  0.024335
30154                  1               0                 0  0.015384
```

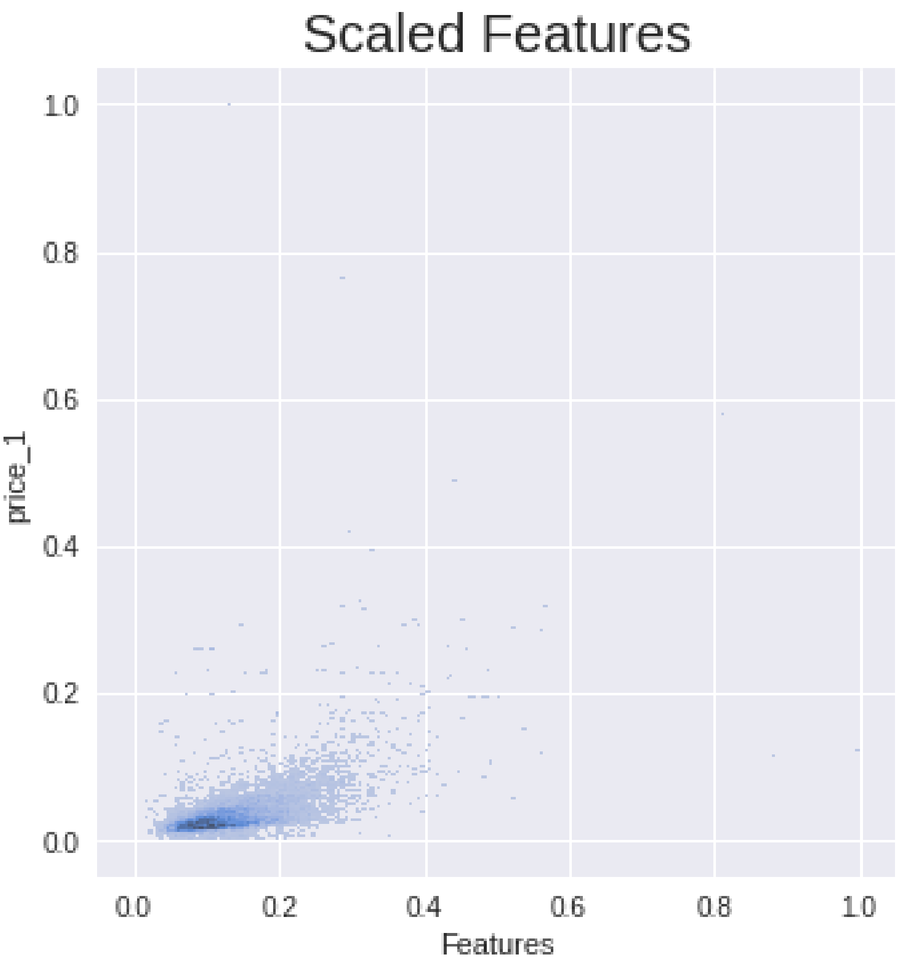
```
       sqft_living_1
0           0.076643
1           0.180178
2           0.187341
3           0.140457
4           0.072736
…              …
30150       0.124178
30151       0.131341
30152       0.105294
30153       0.167155
30154       0.077945

[30111 rows x 49 columns]
```

[82]:

```
# performing min-max scaling on continuous features

plt.figure(figsize=(10.5,6))
sns.displot(data=trial_df1, x='sqft_living', y='price', kind='hist')
sns.displot(data=trial_df1, x='sqft_living_1', y='price_1', kind='hist')

plt.title('Scaled Features', fontdict={'fontsize': 20})
plt.xlabel('Features')
plt.show();
```

```
<Figure size 756x432 with 0 Axes>
```


1e7
3.0
2.5
2.0
1.5
1.0
0.5
0.0
price
0
2000
4000
6000
8000
10000
12000
14000
16000
sqft_living



Note: Correlation between square footage of living space and price of the home is fairly high compared to the other features. It is clear that larger homes mandate higher asking prices. Selling homes on the larger-end of the spectrum are guaranteed to generate the most revenue.

1.9.2 Model Trial 2

[83]:
```
# dealing with all the categorical features from the dataset
#bathroom,grade, zipcode

trial_df2.bathrooms = trial_df2.bathrooms.astype('int').astype('category')
trial_df2.grade = trial_df2.grade.astype('category')
trial_df2.zipcode = trial_df2.zipcode.astype('category')
```

[84]:
```
# making dummies for all the categorical features

bathrooms = pd.get_dummies(trial_df2.bathrooms, prefix='bathrooms',␣
 ↪drop_first=True)
```

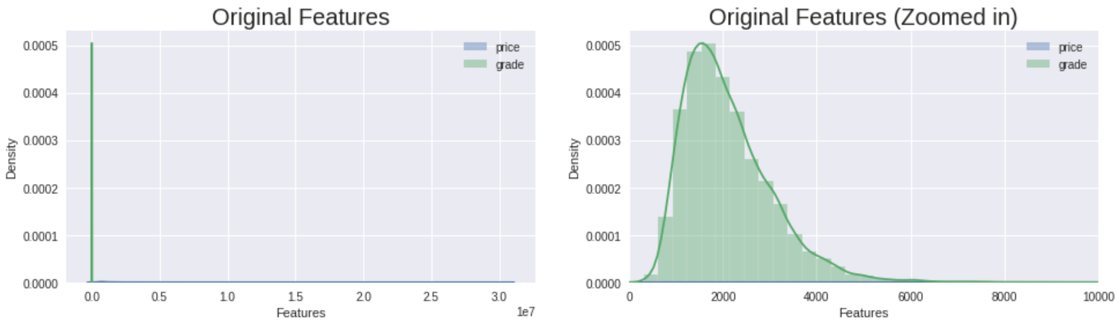
```
grade = pd.get_dummies(trial_df2.grade, prefix='grade', drop_first=True)
zipcode = pd.get_dummies(trial_df2.zipcode, prefix='zipcode', drop_first=True)
```

[85]:

```
# adding dummies to the dataset and removing the original features

trial_df2 = trial_df2.join([bathrooms, grade, zipcode])
trial_df2.drop(['bathrooms', 'grade', 'zipcode'], axis=1, inplace=True)
```

[86]:

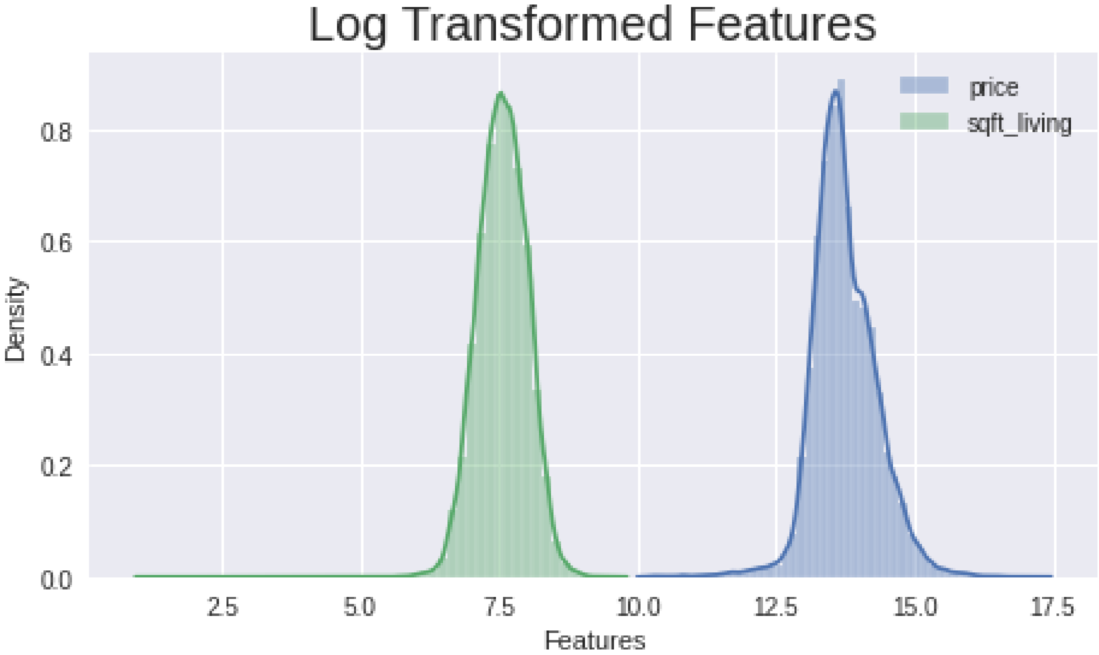
```
# displots on the continuous features from the dataset

plt.figure(figsize=(16,4))
plt.subplot(121)
sns.distplot(trial_df3.price, label='price')
sns.distplot(trial_df3.sqft_living, label='grade')
plt.title('Original Features', fontdict={'fontsize': 20})
plt.xlabel('Features')
plt.legend()

plt.subplot(122)
sns.distplot(trial_df2.price, label='price')
sns.distplot(trial_df2.sqft_living, label='grade')
plt.title('Original Features (Zoomed in)', fontdict={'fontsize': 20})
plt.xlabel('Features')
plt.xlim(0, 10000)
plt.legend()
plt.show()
```



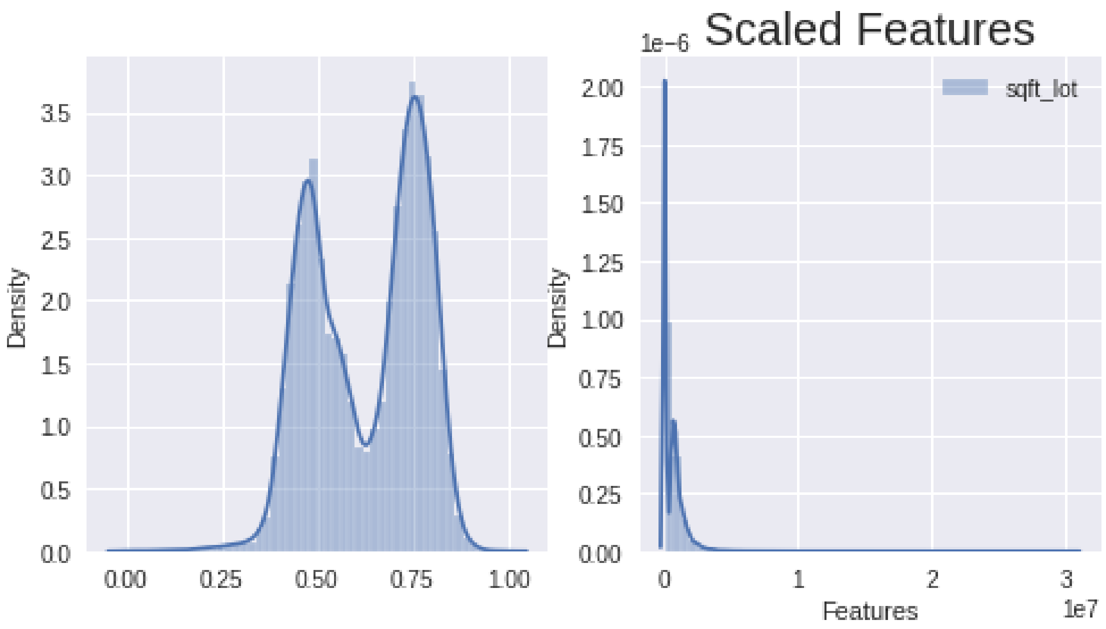
Note: It is very influential in the price of the home. In general, as the grade increases, the price increases as well. This highlights the positive linear correlation between the two.

Sidenote: The grade distribution follows a normal curve, which suggests that they are being issued in a forthright and diligent manner. If interested it would be engaging to see what goes into the grading component of the homes. But that's a project for another time.

[87]:
```
# logarithmic transformation on the continuous features price versus sqft_living

price = np.log(trial_df2.price)
sqft_living = np.log(trial_df2.sqft_living)

plt.figure(figsize=(7.5,4))
sns.distplot(price, label='price')
sns.distplot(sqft_living, label='sqft_living')

plt.title('Log Transformed Features', fontdict={'fontsize': 20})
plt.xlabel('Features')
plt.legend()
plt.show()
```



[88]:
```
# performing min-max scaling on continuous features price versus sqft_lot

trial_df2['price_2'] = ( price - min(price) ) / ( max(price) - min(price) )
trial_df2['sqft_living_2'] = ( sqft_living - min(sqft_living) ) / (␣
 ↪max(sqft_living) - min(sqft_living) )

test = trial_df2[['sqft_living_2', 'price_2']] #Mini dataframe
test_1 = trial_df2[['sqft_lot','price']]

fig, (ax1,ax2) = plt.subplots(ncols=2 , figsize=(8,4))
sns.distplot(test, label='sqft_living', ax=ax1)
sns.distplot(test_1, label='sqft_lot', ax=ax2)
#sns.displot(trial_df3.sqft_lot, trial_df3.price, label='sqft_lot')

plt.title('Scaled Features', fontdict={'fontsize': 20})
plt.xlabel('Features')
plt.legend()
plt.show()
```



1.9.3 Final Model

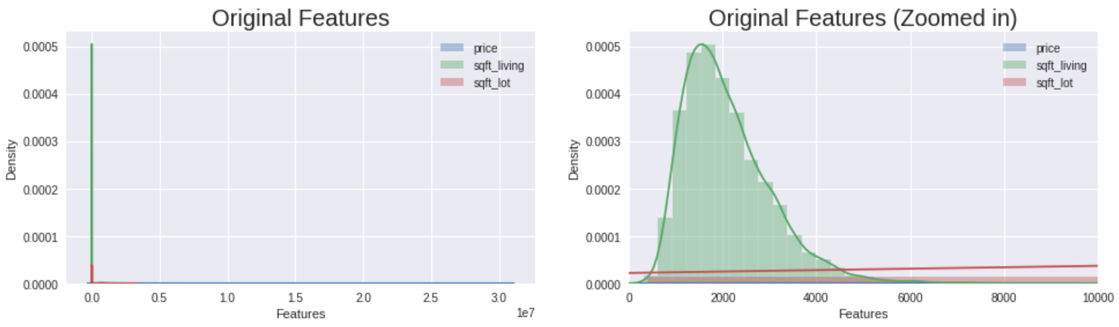
[89]:
```
df.head()
```

[89]:
```
         date     price  bedrooms  bathrooms  sqft_living  sqft_lot  floors  \
0   5/24/2022  675000.0         4        1.0         1180      7140     1.0
1  12/13/2021  920000.0         5        2.5         2770      6703     1.0
2   9/29/2021  311000.0         6        2.0         2880      6156     1.0
3  12/14/2021  775000.0         3        3.0         2160      1400     2.0
4   8/24/2021  592500.0         2        2.0         1120       758     2.0

  waterfront greenbelt nuisance  … sqft_garage sqft_patio yr_built  \
0         NO        NO       NO  …           0         40     1969
1         NO        NO      YES  …           0        240     1950
2         NO        NO       NO  …           0          0     1956
3         NO        NO       NO  …         200        270     2010
4         NO        NO      YES  …         550         30     2012
```

```
  yr_renovated                                                   address       lat  \
0            0  2102 Southeast 21st Court, Renton, Washington …  47.461975
1            0  11231 Greenwood Avenue North, Seattle, Washing…  47.711525
2            0  8504 South 113th Street, Seattle, Washington 9…  47.502045
3            0  4079 Letitia Avenue South, Seattle, Washington…  47.566110
4            0  2193 Northwest Talus Drive, Issaquah, Washingt…  47.532470

       long  sale_yr  yr_old  zipcode
0 -122.19052     2022      53    98055
1 -122.35591     2021      71    98133
2 -122.22520     2021      65    98178
3 -122.29020     2021      11    98118
4 -122.07188     2021       9    98027

[5 rows x 27 columns]
```

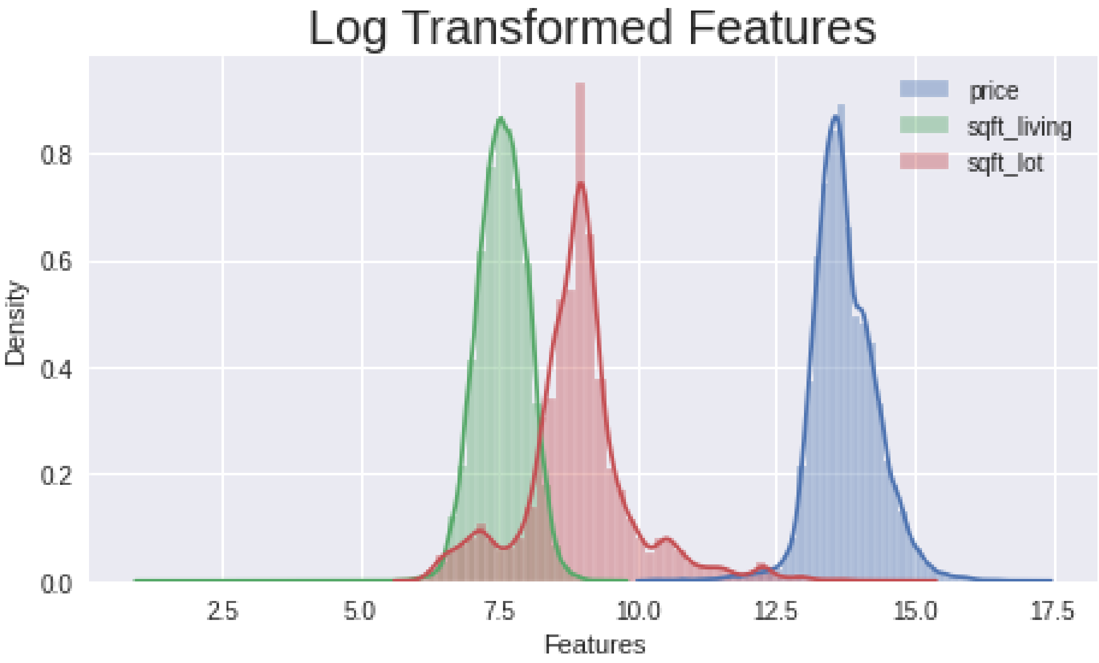
[90]:

```
# displots on the continuous features from the dataset
#sqft_living, sqft_lot and price

plt.figure(figsize=(16,4))
plt.subplot(121)
sns.distplot(trial_df1.price, label='price')
sns.distplot(trial_df1.sqft_living, label='sqft_living')
sns.distplot(trial_df1.sqft_lot, label='sqft_lot')
plt.title('Original Features', fontdict={'fontsize': 20})
plt.xlabel('Features')
plt.legend()

plt.subplot(122)
sns.distplot(trial_df1.price, label='price')
sns.distplot(trial_df1.sqft_living, label='sqft_living')
sns.distplot(trial_df1.sqft_lot, label='sqft_lot')
plt.title('Original Features (Zoomed in)', fontdict={'fontsize': 20})
plt.xlabel('Features')
plt.xlim(0, 10000)
plt.legend()
plt.show()
```



[91]:

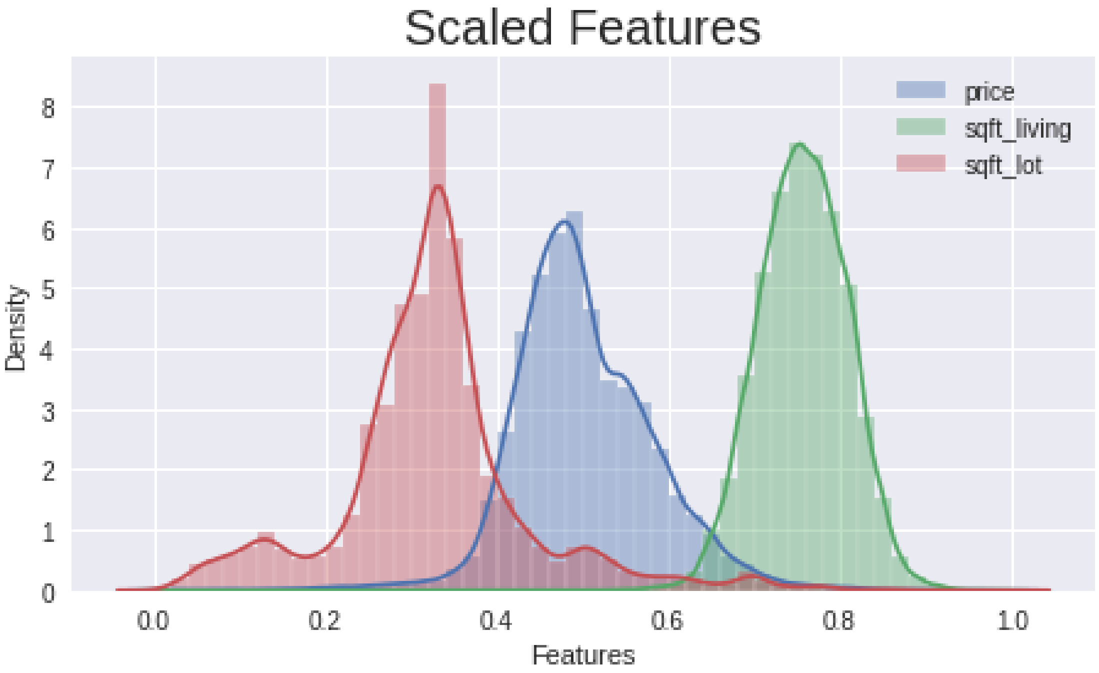
```
# logarithmic transformation on the continuous features

price = np.log(trial_df1.price)
sqft_living = np.log(trial_df1.sqft_living)
sqft_lot = np.log(trial_df1.sqft_lot)

plt.figure(figsize=(7.5,4))
sns.distplot(price, label='price')
sns.distplot(sqft_living, label='sqft_living')
sns.distplot(sqft_lot, label='sqft_lot')

plt.title('Log Transformed Features', fontdict={'fontsize': 20})
plt.xlabel('Features')
plt.legend()
plt.show()
```



[92]:

```
# performing min-max scaling on continuous features

trial_df1['price'] = ( price - min(price) ) / ( max(price) - min(price) )
trial_df1['sqft_living'] = ( sqft_living - min(sqft_living) ) / (␣
 ↪max(sqft_living) - min(sqft_living) )
trial_df1['sqft_lot'] = ( sqft_lot - min(sqft_lot) ) / ( max(sqft_lot) -␣
 ↪min(sqft_lot) )
```

```
plt.figure(figsize=(7.5,4))
sns.distplot(trial_df1.price, label='price')
sns.distplot(trial_df1.sqft_living, label='sqft_living')
sns.distplot(trial_df1.sqft_lot, label='sqft_lot')
plt.title('Scaled Features', fontdict={'fontsize': 20})
plt.xlabel('Features')
plt.legend()
plt.show()
```



[93]:
```
# create a filter that has only the numerical columns of the dataset
pred_cols = [x for x in df.columns if x not in␣
 ↪['selldate','price','waterfront','greenbelt','nuisance','view','condition','grade','heat_so
pred_cols
```

[93]:
```
['date',
 'bedrooms',
 'bathrooms',
 'sqft_living',
 'sqft_lot',
 'floors',
 'sqft_above',
 'sqft_basement',
 'sqft_garage',
 'sqft_patio',
 'yr_built',
 'yr_renovated',
 'lat',
 'long',
 'sale_yr',
 'yr_old']
```

[94]:

```
pred_cols = ['bedrooms',
 'bathrooms',
 'sqft_living',
 'sqft_lot',
 'floors',
 'sqft_above',
 'sqft_basement',
 'sqft_garage',
 'sqft_patio',
 'yr_built',
 'sale_yr',
 'yr_old']
```

[95]:

```
Filtered_df2['Mean_sqft_living'] = Filtered_df2['sqft_living'] -␣
 ↪Filtered_df2['sqft_living'].mean()
Filtered_df2['Mean_sqft_lot'] = Filtered_df2['sqft_lot'] -␣
 ↪Filtered_df2['sqft_lot'].mean()
Filtered_df2['Mean_sqft_above'] = Filtered_df2['sqft_above'] -␣
 ↪Filtered_df2['sqft_above'].mean()
Filtered_df2['Mean_sqft_basement'] = Filtered_df2 ['sqft_basement'] -␣
 ↪Filtered_df2['sqft_basement'].mean()
```

[96]:

```
pred_cols_test =  ['bedrooms',
 'bathrooms',
 'Mean_sqft_living',
 'Mean_sqft_lot',
 'floors',
 'Mean_sqft_above',
 'Mean_sqft_basement',
 'sqft_garage',
 'sqft_patio',
 'yr_built',
 'sale_yr',
 'yr_old',
 'zipcode']
```

[97]:

```
Filtered_df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 17498 entries, 1 to 30152
```

```
Data columns (total 31 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   date                17498 non-null  object
 1   price               17498 non-null  float64
 2   bedrooms            17498 non-null  int64
 3   bathrooms           17498 non-null  float64
 4   sqft_living         17498 non-null  int64
 5   sqft_lot            17498 non-null  int64
 6   floors              17498 non-null  float64
 7   waterfront          17498 non-null  object
 8   greenbelt           17498 non-null  object
 9   nuisance            17498 non-null  object
 10  view                17498 non-null  object
 11  condition           17498 non-null  object
 12  grade               17498 non-null  object
 13  heat_source         17489 non-null  object
 14  sewer_system        17490 non-null  object
 15  sqft_above          17498 non-null  int64
 16  sqft_basement       17498 non-null  int64
 17  sqft_garage         17498 non-null  int64
 18  sqft_patio          17498 non-null  int64
 19  yr_built            17498 non-null  int64
 20  yr_renovated        17498 non-null  int64
 21  address             17498 non-null  object
 22  lat                 17498 non-null  float64
 23  long                17498 non-null  float64
 24  sale_yr             17498 non-null  int64
 25  yr_old              17498 non-null  int64
 26  zipcode             17498 non-null  object
 27  Mean_sqft_living    17498 non-null  float64
 28  Mean_sqft_lot       17498 non-null  float64
 29  Mean_sqft_above     17498 non-null  float64
 30  Mean_sqft_basement  17498 non-null  float64
dtypes: float64(9), int64(11), object(11)
memory usage: 4.3+ MB
```

[98]:

```
# apply the filter we created to our dataset, assign the model features to␣
↪'preds' and assign price to 'target'.
preds2 = Filtered_df2[pred_cols_test]
target2 = Filtered_df2.price
preds2 = pd.get_dummies(preds2, columns=['zipcode'], drop_first=True)
```

[99]:

```
# create baseline model predictor df and target
y2= target2
X2= preds2
```

```
model2 = sm.OLS(y2, sm.add_constant(X2))
results2 = model2.fit()
```

[100]:

```
results2.summary()
```

[100]:

```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.689
Model:                            OLS   Adj. R-squared:                  0.688
Method:                 Least Squares   F-statistic:                     942.3
Date:                Sun, 02 Oct 2022   Prob (F-statistic):               0.00
Time:                        02:27:54   Log-Likelihood:            -2.4500e+05
No. Observations:               17498   AIC:                         4.901e+05
Df Residuals:                   17456   BIC:                         4.904e+05
Df Model:                          41
Covariance Type:            nonrobust
================================================================================
======
                         coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
------
const              -2.651e+08   9.31e+06    -28.479      0.000   -2.83e+08
-2.47e+08
bedrooms            -3.76e+04   3208.410    -11.720      0.000   -4.39e+04
-3.13e+04
bathrooms           4.955e+04   4654.851     10.645      0.000    4.04e+04
5.87e+04
Mean_sqft_living     229.0350     10.958     20.900      0.000     207.555
250.515
Mean_sqft_lot          0.6347      0.040     16.058      0.000       0.557
0.712
floors             -5.328e+04   6375.611     -8.356      0.000   -6.58e+04
-4.08e+04
Mean_sqft_above      133.5156     11.097     12.032      0.000     111.765
155.267
Mean_sqft_basement     7.3212      8.325      0.879      0.379      -8.997
23.639
sqft_garage           79.4208     11.800      6.731      0.000      56.292
102.550
sqft_patio           128.2852     10.957     11.708      0.000     106.808
149.763
yr_built           -1864.3002    182.179    -10.233      0.000   -2221.389
-1507.211
sale_yr             1.333e+05   4607.518     28.933      0.000    1.24e+05
```

```
1.42e+05
yr_old             -1262.6780    181.967     -6.939      0.000   -1619.352
-906.004
zipcode_98003       3579.3787   1.82e+04      0.196      0.844   -3.21e+04
3.93e+04
zipcode_98006       8.907e+05   1.76e+04     50.511      0.000    8.56e+05
9.25e+05
zipcode_98022       4072.6313   1.82e+04      0.224      0.823   -3.15e+04
3.97e+04
zipcode_98023      -1.106e+04   1.61e+04     -0.685      0.493   -4.27e+04
2.06e+04
zipcode_98031       4.293e+04   1.73e+04      2.486      0.013    9084.274
7.68e+04
zipcode_98033       1.142e+06   1.69e+04     67.580      0.000    1.11e+06
1.18e+06
zipcode_98034       6.081e+05   1.62e+04     37.463      0.000    5.76e+05
6.4e+05
zipcode_98038       1.015e+05   1.54e+04      6.576      0.000    7.12e+04
1.32e+05
zipcode_98042       4471.8438   1.49e+04      0.299      0.765   -2.48e+04
3.38e+04
zipcode_98045       2.669e+05   1.79e+04     14.876      0.000    2.32e+05
3.02e+05
zipcode_98052       8.084e+05    1.7e+04     47.490      0.000    7.75e+05
8.42e+05
zipcode_98056       3.176e+05   1.74e+04     18.251      0.000    2.84e+05
3.52e+05
zipcode_98058       1.074e+05   1.62e+04      6.639      0.000    7.57e+04
1.39e+05
zipcode_98059       2.805e+05   1.69e+04     16.629      0.000    2.47e+05
3.14e+05
zipcode_98092      -3.048e+04   1.67e+04     -1.824      0.068   -6.32e+04
2281.534
zipcode_98103       6.216e+05   1.67e+04     37.120      0.000    5.89e+05
6.54e+05
zipcode_98106       2.604e+05   1.75e+04     14.845      0.000    2.26e+05
2.95e+05
zipcode_98107       6.203e+05   1.88e+04     32.945      0.000    5.83e+05
6.57e+05
zipcode_98115       6.388e+05   1.65e+04     38.814      0.000    6.07e+05
6.71e+05
zipcode_98117       6.029e+05   1.67e+04     36.197      0.000     5.7e+05
6.36e+05
zipcode_98118       3.653e+05   1.72e+04     21.222      0.000    3.32e+05
3.99e+05
zipcode_98122        6.25e+05   1.92e+04     32.485      0.000    5.87e+05
6.63e+05
```

```
zipcode_98125       4.408e+05   1.79e+04     24.570      0.000    4.06e+05
4.76e+05
zipcode_98126       3.794e+05   1.87e+04     20.263      0.000    3.43e+05
4.16e+05
zipcode_98133       3.325e+05   1.65e+04     20.115      0.000       3e+05
3.65e+05
zipcode_98144        5.41e+05   1.91e+04     28.288      0.000    5.04e+05
5.78e+05
zipcode_98146       2.631e+05   1.86e+04     14.158      0.000    2.27e+05
3e+05
zipcode_98155       3.771e+05   1.76e+04     21.459      0.000    3.43e+05
4.12e+05
zipcode_98198       9.053e+04   1.84e+04      4.911      0.000    5.44e+04
1.27e+05
==============================================================================
Omnibus:                     5950.012   Durbin-Watson:                   1.940
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            93449.777
Skew:                           1.209   Prob(JB):                         0.00
Kurtosis:                      14.060   Cond. No.                     2.47e+08
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 2.47e+08. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

Note the r2 value which gives the accuracy of the model. its at 14.060, therefor there is an increase of r^2 score at 0.158.

1.10 Regression Results

After building the multiple linear regression, I arrived at in increase of the variance of the price, for the first model at r^2 score of .531 and second model at r^2 score of 0.689 , after performing the final model it validates that the model accuracy of an increase of r^2 score of .0158. In which the model accuracy increases to 69%. When using signigicant features using those p-value below 0.05 - Mean_sqft_basement = 0.379, zipcode_98003 = 0.844 , zipcode_98022 = .0823 , zipcode_98023 = 0.493 , zipcode_98031= 0.013 ,zipcode_98042=0.0765 , zipcode_98092 = 0.068

In using all the data of columns choosen for final model , the regression coefficient matrix for bedrooms = -3.76e+04 indicates that the value decreases than the bathrooms = 4.955e+04 tends to increase , The coefficient values that signifies how much the mean of following , Mean_sqft_living = 229.0350 , Mean_sqft_lot = 0.6347, Mean_sqft_above = 133.5156 , Mean_sqft_basement = 7.3212, it changes the model constant. Coefficients tell you about these changes and p-values tell you if these coefficients are significantly different from zero.

1.11 Conclusion

After having done for the First Model linear regression without extracting any of the features it is evident that the model was in less status. The accuracy was on the below and it was nowhere close to predicting the house price at an accurate level or precision. After controlling for the features in the final model and only allowing for sqft_living, sqft_lot grade, and zipcode (which was one feature I never considered using until it came up as a significant feature in the final model), the r^2 score from .0531 in first Model to .0689 in the final model and the model accuracy was up to 69%.

I can conclude from looking at all this that the final model are more significant (sqft_living, sqft_above, zipcode) to best predict house prices. The model without controlling for significant features did slightly better than the one that did, but the results we obtained don't seem to be that different from one another. Both models did extremely well.

1.12 Recommendations

1.Make sure to focus a great deal on the living space (sqft) of the house when taking price into account. These two are very much positively correlated. This means that as living space square footage increases, so does the price. If there is one sole feature that will drive the price of a particular house up, it would have to be the sqare footage.

2.Location, location, and location! Pay particular attention to the locality of the house. Particular zipcodes are associated with quite expensive homes and vice-versa. Although we didn't dive much into it in this project, it would be interesting to see the what the ratings of the schools are in these areas and the median salaries for people living in these regions.

3.The grade of the home had a significant impact on the price of the homes as well. I'm not too sure what goes into the grading system that King County uses. It would be interesting to see what the variables are that are taken into account when grading a particular home. The grading system seems to be fairly distributed in terms of homes per particular grade.

4.If there was a need to include a fourth feature when looking at house prices it would have to be bathrooms. I found it quite odd, to say the least, that bathrooms drove the price up more than did bedrooms. I would've assumed it would be the other way around, but after performing correlation analysis, it proved to be bathrooms first and bedrooms second.

1.13 Level Up: Project Enhancements

After completing the minimum project requirements, you could consider the following enhancements if you have time:

- Consider applying a linear or non-linear transformation to your features and/or target
- Investigate the linear regression assumptions for your final model
- Identify and remove outliers, then redo the analysis
- Compile the data cleaning code into a function

NOTE: I was not able to have more time on this^^, but it's absoltely a good project in the future.