index

October 2, 2022

1 King County Housing Linear Regression Project by Mike Van Eaton

1.1 Business Understanding

House appraisers have software available to them to help predict the value of homes recently put on the market. There are many features of houses that are both objective and subjective. The data is gathered by realtors who are trying to get the most value for their clients house. This is a problem for a predictive model for the house appraisers. House appraisers verify the realtors descriptions and quality of subjective features. A quality predictivice model will help both for buyer and seller feel good about their purchase and for the realtor and appraiser who depend on each other for accurate and valid valuation. This project looks at many mulitple listing service (MLS) features and distance to neighborhood locations such as schools, transit, coffee shops to increase the models explanatin to the spread of house prices in the Washington states King County region.

1.2 Data Understanding

The data in this project comes from Washington state's records of housing sales for the years 2020-2011 , a list of zip codes representing 693 cities in Washington state, and Washington state's King County GIS data hub from reports created on 9/22/2022. The housing authority maintains current records for all King County realestate transactions where the price of the sale of the house is recorded. The geographic information system (GIS) data hub contains all data that includes a location for King County. This data set contains approximately 28,000 points after cleaning. The data includes typical MLS quantitative features as size of living area, lot size, and the number of bedrooms and bathrooms. Also included are qualitative features such as nuisances, views, condition, and quality grade. Additional potential features are comparing distances to schools, parks, and coffee shops, etc. to the sale price. The MLS features are used as they are typical descriptors for a property price that potential home buyer will see. Nearby neighborhood features were selected for their potential subconcious consideration when thinking about house location and the price of the house. One issue with the distances gathered for this analysis is they are not driving direction distances but rather straight line distances. Both sets of data also have potetial data entry errors due to human error.

1.3 Data Preparation

1.3.1 Loading the Data

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import seaborn as sns
import scipy.stats as stats
import statsmodels.api as sm
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
#%matplotlib nbagg
#plt.style.use('seaborn')

kcdf = pd.read_csv('data/kc_house_data.csv')
WAzips = pd.read_csv('data/wa zip.csv')
kcwaste = pd.read_csv('data/
 ↪Solid_Waste_Facilities_Location___sw_facilities_point.csv')
kcpoints = pd.read_csv('data/
 ↪Common_Points_of_Interest_for_King_County____common_interest_point.csv')
kcinspect = pd.read_csv('data/
 ↪Restaurant_Inspections___restaurant_inspections_point.csv')
```

1.4 King County Housing Data

[2]:
```
# Columns for the king county data frame
kcdf.columns
```

[2]:
```
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')
```

[3]:
```
# Drop ['id'] column, unneeded for project.
kcdf.drop('id', axis = 1, inplace = True)
# Rename ['date'] to ['selldate'] for clarity.
kcdf.rename(columns = {'date':'selldate'}, inplace = True)
```

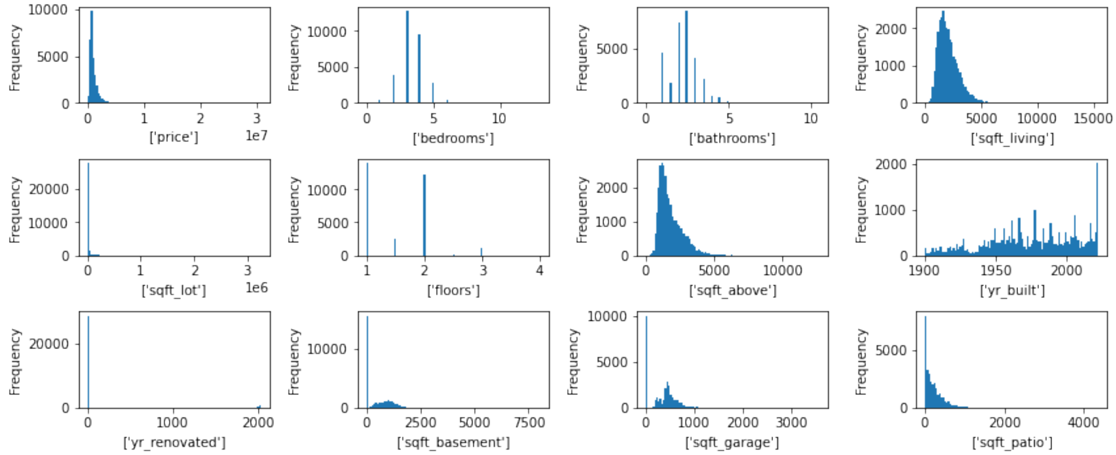
[4]:
```
kcdf.info() # checking for consistant column entries, data type
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30155 entries, 0 to 30154
Data columns (total 24 columns):
```

```
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   selldate       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
dtypes: float64(5), int64(9), object(10)
memory usage: 5.5+ MB
```

[5]:

```
#Checking features for outliers for MLS numeric features

numeric_features = kcdf[['price','bedrooms', 'bathrooms',␣
 ↪'sqft_living','sqft_lot', 'floors', 'sqft_above',
                         'yr_built','yr_renovated','sqft_basement',␣
 ↪'sqft_garage', 'sqft_patio'
                        ]].columns

fig, axes = plt.subplots(3,4, figsize=(12,5))
axe = axes.ravel()
for index, feature in enumerate(numeric_features):
    kcdf[feature].plot.hist(ax=axe[index],bins = 100).set_xlabel([feature])
    fig.tight_layout(pad=1)
```



[6]:
```
#['selldate'] to datetime .dtype
kcdf['selldate'] = pd.to_datetime(kcdf['selldate'])
kcdf['selldate'].dtype
```

[6]:
```
dtype('<M8[ns]')
```

[7]:
```
#['yr_built'] to datetime .dtype
kcdf['yr_built'] = pd.to_datetime(kcdf.yr_built, format = '%Y').dt.year
kcdf['yr_built'].dtype
```

[7]:
```
dtype('int64')
```

[8]:
```
# Calculate ['age'] of house from  ['yr_built'] or ['yr_renovated'] to␣
 ↪['selldate']
kcdf['age'] = np.where( kcdf['yr_renovated'] != 0,kcdf['selldate'].apply(
                lambda x: x.year) - kcdf['yr_renovated'],
                 kcdf['selldate'].apply(lambda x: x.year) - kcdf['yr_built']
                )
kcdf['age'].head(5)
```

[8]:
```
0    53
1    71
2    65
3    11
4     9
Name: age, dtype: int64
```

[9]:
```
#['view'] and ['grade'] descriptions to compare description syntax
kcdf['view'].value_counts(), kcdf['grade'].value_counts()
```

[9]:
```
(NONE         26589
 AVERAGE       1915
 GOOD           878
 EXCELLENT      553
 FAIR           220
 Name: view, dtype: int64,
 7 Average         11697
 8 Good             9410
 9 Better           3806
 6 Low Average      2858
 10 Very Good       1371
 11 Excellent        406
 5 Fair              393
 12 Luxury           122
 4 Low                51
 13 Mansion           24
 3 Poor               13
 1 Cabin               2
 2 Substandard         2
 Name: grade, dtype: int64)
```

[10]:
```
# Split GRADE into value and description columns, delete grade combined value,␣
↪make value int64

kcgrade = kcdf['grade'].str.split(pat = ' ', expand = True)
kcdf.insert(loc = 13, column = 'grade val', value = kcgrade[0])
kcdf.insert(loc = 14, column = 'grade desc', value = kcgrade[1])
kcdf.drop('grade', axis = 1, inplace = True)

kcdf['grade val']=kcdf['grade val'].astype('int64')
kcdf['grade val'].dtype

# Split ADDRESS into value and description columns, delete grade combined value
kcaddress = kcdf['address'].str.split(pat = ',', expand = True)
kcaddressstatezip=kcaddress[2].str.split(pat = ' ', expand = True)
kcdf.insert(loc = 23, column = 'street', value = kcaddress[0])
kcdf.insert(loc = 24, column = 'city', value = kcaddress[1])
kcdf.insert(loc = 25, column = 'state', value = kcaddressstatezip[1])
kcdf.insert(loc = 26, column = 'zip', value = kcaddressstatezip[2])
kcdf.drop('address', axis = 1, inplace = True)

# Change spelling for [view] features to match other feature spellings
kcdf['state'] = kcdf['state'].str.replace('Washington','WA')
kcdf['view'] = kcdf['view'].str.replace('NONE','None')
kcdf['view'] = kcdf['view'].str.replace('AVERAGE','Average')
kcdf['view'] = kcdf['view'].str.replace('GOOD','Good')
kcdf['view'] = kcdf['view'].str.replace('EXCELLENT','Excellent')
```

```
kcdf['view'] = kcdf['view'].str.replace('FAIR','Fair')


kcdf.head(5)
```

[10]:

```
     selldate     price  bedrooms  bathrooms  sqft_living  sqft_lot  floors  \
0  2022-05-24  675000.0         4        1.0         1180      7140     1.0
1  2021-12-13  920000.0         5        2.5         2770      6703     1.0
2  2021-09-29  311000.0         6        2.0         2880      6156     1.0
3  2021-12-14  775000.0         3        3.0         2160      1400     2.0
4  2021-08-24  592500.0         2        2.0         1120       758     2.0

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

                          street       city state    zip        lat       long  \
0     2102 Southeast 21st Court     Renton    WA  98055  47.461975 -122.19052
1  11231 Greenwood Avenue North    Seattle    WA  98133  47.711525 -122.35591
2        8504 South 113th Street    Seattle    WA  98178  47.502045 -122.22520
3      4079 Letitia Avenue South    Seattle    WA  98118  47.566110 -122.29020
4     2193 Northwest Talus Drive   Issaquah    WA  98027  47.532470 -122.07188

   age
0   53
1   71
2   65
3   11
4    9

[5 rows x 29 columns]
```

[11]:

```
# Check for missing data
kcdf.isna().sum()
```

[11]:

```
selldate         0
price            0
bedrooms         0
bathrooms        0
sqft_living      0
sqft_lot         0
floors           0
waterfront       0
greenbelt        0
```

```
nuisance          0
view              0
condition         0
grade val         0
grade desc        0
heat_source      32
sewer_system     14
sqft_above        0
sqft_basement     0
sqft_garage       0
sqft_patio        0
yr_built          0
yr_renovated      0
street            0
city              0
state             0
zip              35
lat               0
long              0
age               0
dtype: int64
```

[12]:
```
#Drop missing data. Very few missing compared to size of data set.
kcdf = kcdf.dropna()
kcdf.isna().sum()
```

[12]:
```
selldate         0
price            0
bedrooms         0
bathrooms        0
sqft_living      0
sqft_lot         0
floors           0
waterfront       0
greenbelt        0
nuisance         0
view             0
condition        0
grade val        0
grade desc       0
heat_source      0
sewer_system     0
sqft_above       0
sqft_basement    0
sqft_garage      0
sqft_patio       0
yr_built         0
```

```
yr_renovated     0
street           0
city             0
state            0
zip              0
lat              0
long             0
age              0
dtype: int64
```

1.5 King County Zip Code List

[13]:
```
# Check for column names for filtering.
WAzips.head(3)
```

[13]:

	zip	Zipcode name	City	State	County Name
0	98520	ABERDEEN, WA	ABERDEEN	WA	GRAYS HARBOR
1	98220	ACME, WA	ACME	WA	WHATCOM
2	99101	ADDY, WA	ADDY	WA	STEVENS

[14]:
```
# Filter data for WA ["State"] and KING for ["County Name"]
kczip = WAzips[(WAzips["State"] == "WA") & (WAzips["County Name"] == "KING")]
kczip['County Name'].value_counts()
```

[14]:
```
KING    115
Name: County Name, dtype: int64
```

[15]:
```
kczip.dtypes
```

[15]:
```
zip             object
Zipcode name    object
City            object
State           object
County Name     object
dtype: object
```

1.6 Merge King County Housing List and King County Zip Codes to clear bad entries

[16]:
```
# Filter Housing data only zip codes in King county using zip code list by␣
↪merging the data frames
#kcdfm - king county data frame merge
kcdfm=pd.merge(kcdf,kczip, left_on='zip',right_on='zip')
kcdfm.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 29142 entries, 0 to 29141
```

```
Data columns (total 33 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   selldate        29142 non-null  datetime64[ns]
 1   price           29142 non-null  float64
 2   bedrooms        29142 non-null  int64
 3   bathrooms       29142 non-null  float64
 4   sqft_living     29142 non-null  int64
 5   sqft_lot        29142 non-null  int64
 6   floors          29142 non-null  float64
 7   waterfront      29142 non-null  object
 8   greenbelt       29142 non-null  object
 9   nuisance        29142 non-null  object
 10  view            29142 non-null  object
 11  condition       29142 non-null  object
 12  grade val       29142 non-null  int64
 13  grade desc      29142 non-null  object
 14  heat_source     29142 non-null  object
 15  sewer_system    29142 non-null  object
 16  sqft_above      29142 non-null  int64
 17  sqft_basement   29142 non-null  int64
 18  sqft_garage     29142 non-null  int64
 19  sqft_patio      29142 non-null  int64
 20  yr_built        29142 non-null  int64
 21  yr_renovated    29142 non-null  int64
 22  street          29142 non-null  object
 23  city            29142 non-null  object
 24  state           29142 non-null  object
 25  zip             29142 non-null  object
 26  lat             29142 non-null  float64
 27  long            29142 non-null  float64
 28  age             29142 non-null  int64
 29  Zipcode name    29142 non-null  object
 30  City            29142 non-null  object
 31  State           29142 non-null  object
 32  County Name     29142 non-null  object
dtypes: datetime64[ns](1), float64(5), int64(11), object(16)
memory usage: 7.6+ MB
```

[17]:
```
# Remove any rows with the house price outside of three standard deviations␣
 ↪(outlier removal)
#king county data frame merge outliers (removed)
kcdfmo = kcdfm[np.abs(stats.zscore(kcdfm['price'])) < 3]
kcdfmo
```

[17]:
```
        selldate      price  bedrooms  bathrooms  sqft_living  sqft_lot  \
0     2022-05-24   675000.0         4        1.0         1180      7140
```

```
1      2022-03-02   750000.0         3        2.0         1830      7969
2      2022-03-29   728000.0         4        2.0         2170      7520
3      2022-03-24   565000.0         4        2.0         1400     10364
4      2021-12-28   645000.0         3        2.0         1520      8250
…          …           …        …        …          …         …
29137  2022-05-17   395000.0         1        1.0          620     10400
29138  2021-07-09   328000.0         2        1.5          980      5000
29139  2022-01-26   600000.0         3        2.5         3150    989234
29140  2022-02-08  2451000.0         4        3.5         4050    204296
29141  2021-09-15   750000.0         3        1.0         1530     33250

       floors waterfront greenbelt nuisance  …       city state   zip  \
0         1.0         NO        NO       NO  …     Renton    WA  98055
1         1.0         NO        NO       NO  …     Renton    WA  98055
2         1.0         NO        NO       NO  …     Renton    WA  98055
3         1.5         NO        NO       NO  …     Renton    WA  98055
4         1.0         NO        NO       NO  …     Renton    WA  98055
…        …        …         …        …   …        …  …     …
29137     1.5         NO        NO      YES  …  Skykomish    WA  98288
29138     2.0         NO        NO       NO  …  Skykomish    WA  98288
29139     1.5        YES        NO      YES  …  Skykomish    WA  98288
29140     2.0         NO        NO       NO  …    Preston    WA  98050
29141     1.5         NO        NO       NO  …   Issaquah    WA  98050

             lat       long  age   Zipcode name       City  State  County Name
0      47.461975 -122.19052   53    RENTON, WA     RENTON     WA         KING
1      47.466730 -122.21400   14    RENTON, WA     RENTON     WA         KING
2      47.463930 -122.18974   49    RENTON, WA     RENTON     WA         KING
3      47.448450 -122.21243   51    RENTON, WA     RENTON     WA         KING
4      47.460870 -122.18869   40    RENTON, WA     RENTON     WA         KING
…            …          …  …             …      …      …        …
29137  47.712560 -121.31959   41  SKYKOMISH, WA  SKYKOMISH     WA         KING
29138  47.707580 -121.35905   18  SKYKOMISH, WA  SKYKOMISH     WA         KING
29139  47.714420 -121.27639   39  SKYKOMISH, WA  SKYKOMISH     WA         KING
29140  47.557160 -121.94932   37    PRESTON, WA    PRESTON     WA         KING
29141  47.523720 -121.93144  117    PRESTON, WA    PRESTON     WA         KING

[28733 rows x 33 columns]
```

[18]:
```python
# Additionally remove rows of outliers beyond three standard deviations in the
 ↪['sqft_living']
# kcdfmosq - king county data frame outlier square foot living space (removed)
kcdfmosq = kcdfmo[np.abs(stats.zscore(kcdfmo['sqft_living'])) < 3]
kcdfmosq
```

[18]:
```
         selldate      price  bedrooms  bathrooms  sqft_living  sqft_lot  \
0      2022-05-24   675000.0         4        1.0         1180      7140
```

```
1      2022-03-02   750000.0         3        2.0           1830       7969
2      2022-03-29   728000.0         4        2.0           2170       7520
3      2022-03-24   565000.0         4        2.0           1400      10364
4      2021-12-28   645000.0         3        2.0           1520       8250
…          …         …       …        …          …          …
29137  2022-05-17   395000.0         1        1.0            620      10400
29138  2021-07-09   328000.0         2        1.5            980       5000
29139  2022-01-26   600000.0         3        2.5           3150     989234
29140  2022-02-08  2451000.0         4        3.5           4050     204296
29141  2021-09-15   750000.0         3        1.0           1530      33250

       floors waterfront greenbelt nuisance  …       city state    zip  \
0         1.0         NO        NO       NO  …     Renton    WA  98055
1         1.0         NO        NO       NO  …     Renton    WA  98055
2         1.0         NO        NO       NO  …     Renton    WA  98055
3         1.5         NO        NO       NO  …     Renton    WA  98055
4         1.0         NO        NO       NO  …     Renton    WA  98055
…         …        …       …       …  …          …  …    …
29137     1.5         NO        NO      YES  …  Skykomish    WA  98288
29138     2.0         NO        NO       NO  …  Skykomish    WA  98288
29139     1.5        YES        NO      YES  …  Skykomish    WA  98288
29140     2.0         NO        NO       NO  …    Preston    WA  98050
29141     1.5         NO        NO       NO  …   Issaquah    WA  98050

             lat        long  age   Zipcode name       City  State  County Name
0      47.461975 -122.19052   53     RENTON, WA     RENTON     WA         KING
1      47.466730 -122.21400   14     RENTON, WA     RENTON     WA         KING
2      47.463930 -122.18974   49     RENTON, WA     RENTON     WA         KING
3      47.448450 -122.21243   51     RENTON, WA     RENTON     WA         KING
4      47.460870 -122.18869   40     RENTON, WA     RENTON     WA         KING
…            …         …   …            …         …    …          …
29137  47.712560 -121.31959   41  SKYKOMISH, WA  SKYKOMISH     WA         KING
29138  47.707580 -121.35905   18  SKYKOMISH, WA  SKYKOMISH     WA         KING
29139  47.714420 -121.27639   39  SKYKOMISH, WA  SKYKOMISH     WA         KING
29140  47.557160 -121.94932   37    PRESTON, WA    PRESTON     WA         KING
29141  47.523720 -121.93144  117    PRESTON, WA    PRESTON     WA         KING

[28474 rows x 33 columns]
```

1.7 King County Schools

1.7.1 Elementary Schools

[19]:
```
# Filter GIS point of interes for elementary schools.  Code 660 is for␣
↪elementary schools.
kcEschools =kcpoints[kcpoints['CODE'] == 660]
kcEschools.head()
```

[19]:
```
             X          Y  OBJECTID  FEATURE_ID     ESITE  CODE  \
3  -122.264083  47.319432         4           7      33.0   660
4  -122.261359  47.333845         5     6600283  692199.0   660
5  -122.259132  47.468914         6     6600241   21158.0   660
9  -122.138595  47.661808        10         612  568273.0   660
10 -122.325166  47.484421        11        1268     152.0   660

                                NAME            ABB_NAME            ADDRESS  \
3   Evergreen Heights Elementary School  Evergreen Heights    5602 S 316th St
4       Meredith Hill Elementary School               Hill  5830 S 300th St
5            Tukwila Elementary School             Tukwila    5939 S 149th St
9       Benjamin Rush Elementary School               Rush  6101 152nd Ave NE
10         Cedarhurst Elementary School         Cedarhurst     611 S 132nd St

    ZIPCODE
3   98001.0
4   98001.0
5   98168.0
9   98052.0
10  98168.0
```

[20]:
```python
# Refine list for only Name, longitude, and latitude
kcEschools_reduced = kcEschools[['NAME','X','Y']]
kcEschools_reduced.head()
```

[20]:
```
                                NAME           X          Y
3   Evergreen Heights Elementary School -122.264083  47.319432
4       Meredith Hill Elementary School -122.261359  47.333845
5            Tukwila Elementary School -122.259132  47.468914
9       Benjamin Rush Elementary School -122.138595  47.661808
10         Cedarhurst Elementary School -122.325166  47.484421
```

[21]:
```python
#create list of tuples: (latitude , longitude)
kcEschool_loc = np.array(list(zip(kcEschools_reduced.Y,kcEschools_reduced.X)))
#kcEschool_loc
```

1.7.2 Middle Schools

[22]:
```python
# Filter GIS point of interes for middle schools.  Code 661 is for elementary␣
↪schools.
kcMschools =kcpoints[kcpoints['CODE'] == 661]
kcMschools.head()
```

[22]:
```
              X          Y  OBJECTID  FEATURE_ID     ESITE  CODE  \
14  -122.220144  47.274526        15     6600534      57.0   661
15  -122.229658  47.385693        16     6600644  616921.0   661
254 -122.454360  47.428631       255         770   21170.0   661
```

```
257 -122.294872  47.682589       258          962    12297.0    661
315 -122.119788  47.691700       316          633    12148.0    661

                         NAME         ABB_NAME              ADDRESS   ZIPCODE
14     Mt. Baker Middle School        Mt. Baker        620 37th St SE  98002.0
15    Mill Creek Middle School       Mill Creek     620 Central Ave N  98032.0
254     McMurray Middle School         McMurray   9329 SW Cemetery Rd  98070.0
257     Eckstein Middle School         Eckstein       3003 NE 75th St  98115.0
315      Redmond Middle School   Redmond Middle    10055 166th Ave NE  98052.0
```

[23]:

```
# Refine list for only Name, latitude, and longitude
kcMschools_reduced = kcMschools[['NAME','X','Y']]
kcMschools_reduced.head()
```

[23]:

```
                         NAME           X          Y
14     Mt. Baker Middle School -122.220144  47.274526
15    Mill Creek Middle School -122.229658  47.385693
254     McMurray Middle School -122.454360  47.428631
257     Eckstein Middle School -122.294872  47.682589
315      Redmond Middle School -122.119788  47.691700
```

[24]:

```
#create list of tuples: (latitude , longitude)
kcMschool_loc = np.array(list(zip(kcMschools_reduced.Y,kcMschools_reduced.X)))
#kcMschool_loc
```

[25]:

```
#columns = ['store_id', 'email', 'sales_channel', 'category']
#df['metadata'] = df[columns].to_dict(orient='records')
```

1.7.3 High Schools

[26]:

```
# Filter GIS point of interes for high schools.  Code 662 is for elementary␣
↪schools.
kcHschools =kcpoints[kcpoints['CODE'] == 662]
kcHschools.head()
```

[26]:

```
              X          Y  OBJECTID  FEATURE_ID     ESITE  CODE  \
313 -122.207627  47.373404       314         379    8601.0   662
324 -122.197680  47.604405       325         112      82.0   662
328 -122.198624  47.715496       329         507   42811.0   662
330 -122.294727  47.708042       331        1093   12282.0   662
398 -122.152348  47.501630       399         756  579312.0   662

                          NAME         ABB_NAME                ADDRESS   ZIPCODE
313  Kent-Meridian High School    Kent-Meridian        10020 SE 256th St  98030.0
324        Bellevue High School         Bellevue   10416 SE Wolverine Way  98004.0
328         Juanita High School          Juanita        10601 NE 132nd St  98034.0
330     Nathan Hale High School             Hale        10750 30th Ave NE  98125.0
```

```
398          Hazen High School          Hazen     1101 Hoquiam Ave NE  98059.0
```

[27]:
```
# Refine list for only Name, latitude, and longitude
kcHschools_reduced = kcHschools[['NAME','X','Y']]
kcHschools_reduced.head()
```

[27]:
```
                          NAME           X          Y
313  Kent-Meridian High School -122.207627  47.373404
324       Bellevue High School -122.197680  47.604405
328         Juanita High School -122.198624  47.715496
330     Nathan Hale High School -122.294727  47.708042
398           Hazen High School -122.152348  47.501630
```

[28]:
```
#create list of tuples: (latitude , longitude)
kcHschool_loc = np.array(list(zip(kcHschools_reduced.Y,kcHschools_reduced.X)))
#kcHschool_loc
```

1.8 King County Solid Waste

[29]:
```
kcwaste.head(5)
```

[29]:
```
            X          Y  OBJECTID  TransSiteID               SITEADDR  \
0 -122.178413  47.483646         1            4          3021 NE 4th St
1 -122.267774  47.433836         2            5       18800 Orilla Rd S
2 -122.259761  47.285164         3            6   35315 West Valley Hwy
3 -121.954398  47.205286         4            7     1650 Battersby St E
4 -122.499497  47.435395         5            8   18900 Westside Hwy SE

          SITETYPE      CITY  SITENAME        OWNER IsActive  \
0  Transfer Station    Renton    Renton  King County      Yes
1  Transfer Station   Tukwila  Bow Lake  King County      Yes
2  Transfer Station    Algona    Algona  King County      Yes
3  Transfer Station  Enumclaw  Enumclaw  King County      Yes
4  Transfer Station    Vashon    Vashon  King County      Yes

  IsClosedLandfill ClosedLandfillName
0              NaN                NaN
1              Yes  Bow Lake Landfill
2              NaN                NaN
3              Yes  Enumclaw Landfill
4              Yes    Vashon Landfill
```

[30]:
```
# Refine list for only Name, latitude, and longitude
kcwaste_reduced = kcwaste[['SITENAME','X','Y']]
kcwaste_reduced
```

[30]:

```
                               SITENAME           X          Y
0                                Renton -122.178413  47.483646
1                              Bow Lake -122.267774  47.433836
2                                Algona -122.259761  47.285164
3                              Enumclaw -121.954398  47.205286
4                                Vashon -122.499497  47.435395
5                             Shoreline -122.331846  47.749687
6                              Houghton -122.183508  47.662026
7                              Factoria -122.159177  47.582221
8                           Cedar Falls -121.761446  47.449135
9         Cedar Hills Regional Landfill -122.047540  47.462462
10                       Third & Lander -122.330945  47.578159
11            Eastmont Recycling Center -122.336158  47.535907
12                            Skykomish -121.339824  47.712538
13                          Black River -122.251727  47.477391
14                      Duvall Landfill -122.033430  47.752261
15         Puyallup/Kit Corner Landfill -122.304859  47.284225
16                      Hobart Landfill -121.975811  47.388563
17                   South Park Landfill -122.330981  47.528657
18                   Snoqualmie Drop Box -121.414934  47.412290
19                            Argo Yard -122.331761  47.559284
20        Cascade Recycling Center (CRC) -122.151749  47.765991
21            Recycling Northwest (RNW) -122.237240  47.310793
22                North Transfer Station -122.340749  47.648727
23                South Transfer Station -122.330213  47.530407
24   Northwest Container Services, Inc. -122.324570  47.559806
```

[31]:

```
#create list of tuples: (latitude , longitude)
waste_loc = np.array(list(zip(kcwaste_reduced.Y,kcwaste_reduced.X)))
#waste_loc
```

1.9 King County Churches

[32]:

```
# Filter GIS point of interes for churches.  Code 800 is for churches.
kcchurch = kcpoints[kcpoints['CODE'] == 800]
```

[33]:

```
# Refine list for only Name, latitude, and longitude
kcchurch_reduced = kcchurch[['NAME','X','Y']]
kcchurch_reduced.head()
```

[33]:

```
                                   NAME           X          Y
4194                      UNITY CHURCH -122.340705  47.620133
4195  CROWNHILL UNITED METHODIST CHURCH -122.373708  47.690945
4196       ST. MARGARET CATHOLIC CHURCH -122.375265  47.648951
4197                             CHURCH -122.384288  47.648659
4198              GRACE FELLOWSHIP CHURCH -122.362299  47.674390
```

[34]:
```python
#create list of tuples: (latitude , longitude)
church_loc = np.array(list(zip(kcchurch_reduced.Y,kcchurch_reduced.X)))
```

1.10 King County Parks

[35]:
```python
# Filter GIS point of interes for parks.  Code 600 is for parks.
kcparks = kcpoints[kcpoints['CODE'] == 600]
```

[36]:
```python
# Refine list for only Name, latitude, and longitude
kcparks_reduced = kcparks[['NAME','X','Y']]
kcparks_reduced.head()
```

[36]:

	NAME	X	Y
19	Island Center Forest Natural Area	-122.472494	47.438270
27	Counterbalance Park	-122.356324	47.625681
31	Bear Creek Park - Redmond	-122.108619	47.672285
32	McCormick Park - Bellevue	-122.186478	47.502681
34	Richmond Beach Center	-122.385011	47.771882

[37]:
```python
#create list of tuples: (latitude , longitude)
parks_loc = np.array(list(zip(kcparks_reduced.Y,kcparks_reduced.X)))
```

1.11 King County Transit Stations

[38]:
```python
# Filter GIS point of interes for transit stations.  Code 510 is for transit␣
↪stations.
kctransit = kcpoints[kcpoints['CODE'] == 510]
```

[39]:
```python
# Refine list for only Name, latitude, and longitude
kctransit_reduced = kctransit[['NAME','X','Y']]
kctransit_reduced.head()
```

[39]:

	NAME	X	Y
42	U District Link Light Rail Station	-122.313981	47.660104
151	Mercer Island P&R during construction	-122.231997	47.588452
160	Capitol Hill Link Light Rail Station	-122.320200	47.619062
259	Kenmore Air Harbor Inc	-122.257891	47.757110
261	SODO Link Light Rail Station	-122.327331	47.581797

[40]:
```python
#create list of tuples: (latitude , longitude)
transit_loc = np.array(list(zip(kctransit_reduced.Y,kctransit_reduced.X)))
```

1.12 King County Starbucks

[41]:
```
kcinspect.head()
```

[41]:
```
            X          Y  OBJECTID  FEATURE_ID             NAME  \
0 -122.296415  47.662311         1           2  #807 TUTTA BELLA
1 -122.296415  47.662311         2           3  #807 TUTTA BELLA
2 -122.334587  47.648180         3           4         +MAS CAFE
3 -122.334587  47.648180         4           5         +MAS CAFE
4 -122.331727  47.629021         5           7       100 LB CLAM

  PROGRAM_IDENTIFIER      SEAT_CAP RISK                           ADDRESS  \
0   #807 TUTTA BELLA  Seating 0-12  III                2746 NE 45TH ST
1   #807 TUTTA BELLA  Seating 0-12  III                2746 NE 45TH ST
2          +MAS CAFE  Seating 0-12  III                 1906 N 34TH ST
3          +MAS CAFE  Seating 0-12  III                 1906 N 34TH ST
4        100 LB CLAM  Seating 0-12  III  1001 FAIRVIEW AVE N Unit 1700A

            PHONE  …  RESULT_INSPECTION  CLOSE_BUS_INSPECTION VIOLATIONTYPE  \
0  (206) 722-6400  …       Satisfactory                 False           NaN
1  (206) 722-6400  …       Satisfactory                 False           NaN
2  (206) 491-4694  …       Satisfactory                 False           NaN
3  (206) 491-4694  …       Satisfactory                 False           NaN
4  (206) 369-2978  …         Incomplete                 False           NaN

  VIOLATIONDESCR VIOLATIONPOINTS RECORD_ID FACILITY_NAME  CHAIN_NAME  \
0            NaN               0       NaN           NaN         NaN
1            NaN               0       NaN           NaN         NaN
2            NaN               0       NaN           NaN         NaN
3            NaN               0       NaN           NaN         NaN
4            NaN               0       NaN           NaN         NaN

  CHAIN_ESTABLISHMENT  SITE_ADDRESS
0                 NaN           NaN
1                 NaN           NaN
2                 NaN           NaN
3                 NaN           NaN
4                 NaN           NaN

[5 rows x 28 columns]
```

[42]:
```
#Number of Starbucks in King County
len(kcinspect.loc[kcinspect['NAME'].str.contains('starbucks',case=False, regex␣
 ↪=True)])
```

[42]: 3958

[43]:
```
# Filter file for only Starbucks' informatin
kcstar = kcinspect.loc[kcinspect['NAME'].str.contains('starbucks',case=False,␣
↪regex =True)]
```

[44]:
```
# Refine list for only Name, latitude, and longitude
kcstar_reduced = kcstar[['NAME','X','Y']]
kcstar_reduced.head()
```

[44]:

	NAME	X	Y
26685	BON APPETIT CAFE @ Starbucks Center 5th FL	-122.335918	47.580901
26686	BON APPETIT CAFE @ Starbucks Center 5th FL	-122.335918	47.580901
26687	BON APPETIT CAFE @ Starbucks Center 5th FL	-122.335918	47.580901
26688	BON APPETIT CAFE @ Starbucks Center 5th FL	-122.335918	47.580901
26689	BON APPETIT CAFE @ Starbucks Center 5th FL	-122.335918	47.580901

[45]:
```
#create list of tuples: (latitude , longitude)
star_loc = np.array(list(zip(kcstar_reduced.Y,kcstar_reduced.X)))
```

1.13 King County House Data Coordinates

[46]:
```
#create list of tuples: (latitude , longitude) for the housing data
loc_coord = np.array(list(zip(kcdfmo.lat,kcdfmo.long)))
loc_coord
```

[46]:
```
array([[  47.461975, -122.19052 ],
       [  47.46673 , -122.214   ],
       [  47.46393 , -122.18974 ],
       …,
       [  47.71442 , -121.27639 ],
       [  47.55716 , -121.94932 ],
       [  47.52372 , -121.93144 ]])
```

1.13.1 Baseline Model 1a - compare highest correlated feature with price

A quick look at the correlated values to price to see how good a basemodel with one feature will do.

[47]:
```
#MLS numeric data most correlated to price
kcdfm.corr()['price'].sort_values(ascending=False)
```

[47]:
```
price           1.000000
sqft_living     0.616624
grade val       0.577933
sqft_above      0.545979
bathrooms       0.487963
sqft_patio      0.317627
lat             0.297603
```

```
bedrooms          0.290732
sqft_garage       0.267402
sqft_basement     0.246252
floors            0.199810
yr_built          0.106065
sqft_lot          0.086826
yr_renovated      0.085597
long              0.082432
age              -0.138162
Name: price, dtype: float64
```

[48]:
```
X1 = kcdfm["sqft_living"]
y1 = kcdfm["price"]
```

[49]:
```
model1 = sm.OLS(endog=y1, exog=sm.add_constant(X1))
model1
```

[49]:
```
<statsmodels.regression.linear_model.OLS at 0x7f7f384e8490>
```

[50]:
```
results1 = model1.fit()
results1
```

[50]:
```
<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x7f7f384e8b20>
```

[51]:
```
results1.summary()
```

[51]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.380
Model:                            OLS   Adj. R-squared:                  0.380
Method:                 Least Squares   F-statistic:                 1.788e+04
Date:                Sun, 02 Oct 2022   Prob (F-statistic):               0.00
Time:                        05:56:12   Log-Likelihood:            -4.3379e+05
No. Observations:               29142   AIC:                         8.676e+05
Df Residuals:                   29140   BIC:                         8.676e+05
Df Model:                           1
Covariance Type:            nonrobust
===============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
const       -9.219e+04   9919.150     -9.294      0.000   -1.12e+05   -7.28e+04
sqft_living   565.5280      4.230    133.705      0.000     557.238     573.818
==============================================================================
Omnibus:                    42176.216   Durbin-Watson:                   1.297
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          49934507.561
```

```
Skew:                           8.236   Prob(JB):                         0.00
Kurtosis:                     205.120   Cond. No.                     5.63e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 5.63e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

1.13.2 Model 1a Conclusion

This model has a very low rsquared value only explaining 38% of the price variance. The coefficient represents a house with zero living area costs about $-\$92,000$. Each increas in 1 square foot increases the value by \$560. The p value shows that this is statiscally relavent.

1.13.3 Base Model 1B - compare highest correlated feature with price outliers removed

This model removes the price outliers to see any quick improvement to the overall model.

[52]:
```
#MLS numeric data most correlated to price
kcdfmo.corr()['price'].sort_values(ascending=False)
```

[52]:
```
price            1.000000
sqft_living      0.638320
grade val        0.620132
sqft_above       0.561526
bathrooms        0.499907
lat              0.387137
bedrooms         0.338844
sqft_patio       0.294154
sqft_garage      0.278429
floors           0.242799
sqft_basement    0.220932
long             0.121130
yr_built         0.115352
sqft_lot         0.093967
yr_renovated     0.080101
age             -0.147850
Name: price, dtype: float64
```

[53]:
```
X1b = kcdfmo["sqft_living"]
y1b = kcdfmo["price"]
```

[54]:
```
model1b = sm.OLS(endog=y1b, exog=sm.add_constant(X1b))
model1b
```

[54]: <statsmodels.regression.linear_model.OLS at 0x7f7f345c9850>

[55]:
```
results1b = model1b.fit()
results1b
```

[55]: <statsmodels.regression.linear_model.RegressionResultsWrapper at 0x7f7f345c9490>

[56]:
```
results1b.summary()
```

[56]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.407
Model:                            OLS   Adj. R-squared:                  0.407
Method:                 Least Squares   F-statistic:                 1.976e+04
Date:                Sun, 02 Oct 2022   Prob (F-statistic):               0.00
Time:                        05:56:13   Log-Likelihood:            -4.1577e+05
No. Observations:               28733   AIC:                         8.316e+05
Df Residuals:                   28731   BIC:                         8.316e+05
Df Model:                           1
Covariance Type:            nonrobust
===============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
const        1.504e+05   6936.151     21.684      0.000    1.37e+05    1.64e+05
sqft_living   427.4331      3.041    140.557      0.000     421.473     433.394
==============================================================================
Omnibus:                     6045.409   Durbin-Watson:                   1.044
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            19303.716
Skew:                           1.072   Prob(JB):                         0.00
Kurtosis:                       6.395   Cond. No.                     5.76e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 5.76e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

Model 1b Conclusion This model has the price outliers removed. It has a bit higher rsquared value only explaining 41% of the price variance. The coefficient represents a house with zero living area costs about $150,000 and an increase of $427 a square foot added to the house. The p value shows that this is statiscally relavent.

1.13.4 Model 2 - use all numeric features with price outliers removed

This model uses the Model 1b with all numeric features added in to see any quick improvement to the model.

[57]:
```
kcdfmo_features = kcdfmo.drop(['selldate','lat','long','street', 'Zipcode␣
↪name','city', 'state','State', 'County␣
↪Name','yr_built','yr_renovated','City'],axis=1)
kcdfmo_features.columns
```

[57]:
```
Index(['price', 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors',
       'waterfront', 'greenbelt', 'nuisance', 'view', 'condition', 'grade val',
       'grade desc', 'heat_source', 'sewer_system', 'sqft_above',
       'sqft_basement', 'sqft_garage', 'sqft_patio', 'zip', 'age'],
      dtype='object')
```

[58]:
```
numeric = kcdfmo_features.select_dtypes('number')
numeric
```

[58]:
```
           price  bedrooms  bathrooms  sqft_living  sqft_lot  floors  \
0       675000.0         4        1.0         1180      7140     1.0
1       750000.0         3        2.0         1830      7969     1.0
2       728000.0         4        2.0         2170      7520     1.0
3       565000.0         4        2.0         1400     10364     1.5
4       645000.0         3        2.0         1520      8250     1.0
…            …       …         …          …        …      …
29137   395000.0         1        1.0          620     10400     1.5
29138   328000.0         2        1.5          980      5000     2.0
29139   600000.0         3        2.5         3150    989234     1.5
29140  2451000.0         4        3.5         4050    204296     2.0
29141   750000.0         3        1.0         1530     33250     1.5

       grade val  sqft_above  sqft_basement  sqft_garage  sqft_patio  age
0              7        1180              0            0          40   53
1              7         930            930          240          90   14
2              7        1240           1240          490          60   49
3              6        1400              0          330         330   51
4              8        1190            590          420         200   40
…          …         …            …           …          …  …
29137          6         620              0            0         100   41
29138          7         980              0            0         260   18
29139          7        2150           1390            0        2360   39
29140          9        2280           1770          750        1250   37
29141          6        1530            110            0         360  117

[28733 rows x 12 columns]
```

[59]:
```
X2 = kcdfmo[numeric.columns].drop(['price'],axis=1)
y2 = kcdfmo["price"]
```

[60]:
```
model2 = sm.OLS(endog=y2, exog=sm.add_constant(X2))
results2 = model2.fit()
results2.summary()
```

[60]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.507
Model:                            OLS   Adj. R-squared:                  0.507
Method:                 Least Squares   F-statistic:                     2682.
Date:                Sun, 02 Oct 2022   Prob (F-statistic):               0.00
Time:                        05:56:13   Log-Likelihood:            -4.1314e+05
No. Observations:               28733   AIC:                         8.263e+05
Df Residuals:                   28721   BIC:                         8.264e+05
Df Model:                          11
Covariance Type:            nonrobust
================================================================================
=
                    coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
-
const         -1.416e+06    2.7e+04    -52.508      0.000   -1.47e+06
-1.36e+06
bedrooms      -5.082e+04   3544.991    -14.334      0.000   -5.78e+04
-4.39e+04
bathrooms      7.398e+04   5185.552     14.266      0.000    6.38e+04
8.41e+04
sqft_living     172.3532     11.868     14.522      0.000     149.091
195.616
sqft_lot          0.1699      0.043      3.971      0.000       0.086
0.254
floors        -4.008e+04   6466.238     -6.199      0.000   -5.28e+04
-2.74e+04
grade val      2.363e+05   3717.959     63.569      0.000    2.29e+05
2.44e+05
sqft_above      117.3480     12.069      9.723      0.000      93.692
141.004
sqft_basement    59.2530      8.761      6.763      0.000      42.081
76.425
sqft_garage    -186.6787     11.930    -15.648      0.000    -210.061
-163.296
sqft_patio       75.7462     11.619      6.519      0.000      52.972
```

```
98.521
age              3816.9153    115.783     32.966      0.000    3589.975
4043.856
==============================================================================
Omnibus:                     6426.221   Durbin-Watson:                   1.247
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            26092.207
Skew:                           1.060   Prob(JB):                         0.00
Kurtosis:                       7.159   Cond. No.                     6.79e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 6.79e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

Model 2a Conclusion This model has the price outliers removed and the other numeric features added in. It has a bit higher rsquared value only explaining 51% of the price variance. The coefficient represents a house with zero living area costs about $-\$1,420,000$ and an increase of \$172 a square foot, $\$74,000$ per bathroom, and about $\$3,800$ for each year in the age of the home added to the house. There are some features that add negative value. The p value shows that all features are statiscally relavent.

1.13.5 Model 2b - use all numeric features with all columns' outliers removed

This model removes all outliers from all columns.

[61]:
```python
#MLS numeric data most correlated to price with all outliers removed
kcdfmAo = numeric[(np.abs(stats.zscore(numeric)) < 3).all(axis=1)]
kcdfmAo.corr()['price'].sort_values(ascending=False)
```

[61]:
```
price            1.000000
sqft_living      0.578097
grade val        0.563996
sqft_above       0.485585
bathrooms        0.443254
bedrooms         0.318784
sqft_patio       0.241885
sqft_garage      0.224421
floors           0.223609
sqft_basement    0.201274
sqft_lot         0.063679
age             -0.112120
Name: price, dtype: float64
```

[62]:
```python
len(kcdfmAo)
```

[62]: 26634

[63]:
```
X2b = kcdfmAo[numeric.columns].drop(['price'],axis=1)
y2b = kcdfmAo["price"]
```

[64]:
```
model2b = sm.OLS(endog=y2b, exog=sm.add_constant(X2b))
model2b
```

[64]: <statsmodels.regression.linear_model.OLS at 0x7f7f347ed940>

[65]:
```
results2b = model2b.fit()
results2b
```

[65]: <statsmodels.regression.linear_model.RegressionResultsWrapper at 0x7f7f347ed9a0>

[66]:
```
results2b.summary()
```

[66]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.446
Model:                            OLS   Adj. R-squared:                  0.446
Method:                 Least Squares   F-statistic:                     1949.
Date:                Sun, 02 Oct 2022   Prob (F-statistic):               0.00
Time:                        05:56:14   Log-Likelihood:            -3.7910e+05
No. Observations:               26634   AIC:                         7.582e+05
Df Residuals:                   26622   BIC:                         7.583e+05
Df Model:                          11
Covariance Type:            nonrobust
===============================================================================
=
                    coef    std err          t      P>|t|      [0.025
0.975]
-------------------------------------------------------------------------------
-
const         -1.278e+06   2.59e+04    -49.287      0.000   -1.33e+06
-1.23e+06
bedrooms      -3.953e+04   3450.146    -11.456      0.000   -4.63e+04
-3.28e+04
bathrooms      6.716e+04   4959.340     13.542      0.000    5.74e+04
7.69e+04
sqft_living     160.3684     11.331     14.153      0.000     138.159
182.578
sqft_lot         -0.3102      0.148     -2.094      0.036      -0.601
-0.020
floors        -3.123e+04   6101.094     -5.118      0.000   -4.32e+04
```

```
-1.93e+04
grade val        2.18e+05   3589.093     60.734      0.000    2.11e+05
2.25e+05
sqft_above        97.6455     11.518      8.478      0.000      75.069
120.222
sqft_basement     53.2036      8.191      6.495      0.000      37.148
69.259
sqft_garage     -151.9938     11.601    -13.101      0.000    -174.733
-129.254
sqft_patio        85.7194     12.817      6.688      0.000      60.597
110.842
age             3690.2676    105.847     34.864      0.000    3482.803
3897.733
==============================================================================
Omnibus:                     3468.163   Durbin-Watson:                   1.207
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             8081.234
Skew:                           0.769   Prob(JB):                         0.00
Kurtosis:                       5.218   Cond. No.                     2.25e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 2.25e+05. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

Model 2b Conclusion This model has all column outliers removed. It has a much lower rsquared value compared to the previous model 2a, only explaining 45% of the price variance. The coefficient represents a house with zero living area costs about $-\$1,280,000$ and an increase of \$160 a square foot, \$67,000 per bathroom, and about \$3,700 for each year in the age of the home added to the house. There are some features that add negative value. The p value shows that all features are statiscally relavent.

1.13.6 Model 3, use numeric and categorical except [lat], [long], [Zipcode], [name], [State], [County Name]

This model removes the outliers from the price column, has all of the numeric columns and adds categoric columns with one-hot encoding.

[67]:
```python
#Get only categoric features from kcdfmo (price outliers removed)
categoric = kcdfmo_features.select_dtypes('object')
categoric
```

[67]:
```
      waterfront greenbelt nuisance     view  condition grade desc  \
0             NO        NO       NO     None       Good    Average
1             NO        NO       NO     None    Average    Average
```

```
2          NO      NO      NO     None     Average    Average
3          NO      NO      NO     None        Good        Low
4          NO      NO      NO     None     Average       Good
…          …       …       …      …        …          …
29137      NO      NO     YES     None     Average        Low
29138      NO      NO      NO     None     Average    Average
29139     YES      NO     YES  Average     Average    Average
29140      NO      NO      NO     Good   Very Good     Better
29141      NO      NO      NO     None   Very Good        Low

      heat_source sewer_system    zip
0             Gas       PUBLIC  98055
1             Gas       PUBLIC  98055
2             Gas       PUBLIC  98055
3     Electricity       PUBLIC  98055
4             Gas       PUBLIC  98055
…             …            …    …
29137 Electricity      PRIVATE  98288
29138 Electricity       PUBLIC  98288
29139 Electricity      PRIVATE  98288
29140 Electricity      PRIVATE  98050
29141         Oil      PRIVATE  98050

[28733 rows x 9 columns]
```

[68]:

```python
#Get only numeric features from kcdfmo (price outliers removed)
num = kcdfmo_features.select_dtypes('number')
num
```

[68]:

```
           price  bedrooms  bathrooms  sqft_living  sqft_lot  floors  \
0       675000.0         4        1.0         1180      7140     1.0
1       750000.0         3        2.0         1830      7969     1.0
2       728000.0         4        2.0         2170      7520     1.0
3       565000.0         4        2.0         1400     10364     1.5
4       645000.0         3        2.0         1520      8250     1.0
…          …        …         …           …          …         …
29137   395000.0         1        1.0          620     10400     1.5
29138   328000.0         2        1.5          980      5000     2.0
29139   600000.0         3        2.5         3150    989234     1.5
29140  2451000.0         4        3.5         4050    204296     2.0
29141   750000.0         3        1.0         1530     33250     1.5

      grade val  sqft_above  sqft_basement  sqft_garage  sqft_patio  age
0             7        1180              0            0          40   53
1             7         930            930          240          90   14
2             7        1240           1240          490          60   49
3             6        1400              0          330         330   51
```

```
4                8        1190          590         420         200   40
…            …        …           …           …           …  …
29137            6         620            0           0         100   41
29138            7         980            0           0         260   18
29139            7        2150         1390           0        2360   39
29140            9        2280         1770         750        1250   37
29141            6        1530          110           0         360  117

[28733 rows x 12 columns]
```

[69]:

```python
# One-hot encode categoric features (create dummy columns)
# catwd - categoric features with dummies
# numcatwd - concatenate numeric and categoric dummy columns
catwd = pd.get_dummies(categoric, drop_first=True)
catwd.columns
numcatwd = pd.concat([num,catwd], axis=1)
numcatwd
```

[69]:

```
           price  bedrooms  bathrooms  sqft_living  sqft_lot  floors  \
0       675000.0         4        1.0         1180      7140     1.0
1       750000.0         3        2.0         1830      7969     1.0
2       728000.0         4        2.0         2170      7520     1.0
3       565000.0         4        2.0         1400     10364     1.5
4       645000.0         3        2.0         1520      8250     1.0
…            …      …        …          …        …      …
29137   395000.0         1        1.0          620     10400     1.5
29138   328000.0         2        1.5          980      5000     2.0
29139   600000.0         3        2.5         3150    989234     1.5
29140  2451000.0         4        3.5         4050    204296     2.0
29141   750000.0         3        1.0         1530     33250     1.5

       grade val  sqft_above  sqft_basement  sqft_garage  …  zip_98148  \
0              7        1180              0            0  …          0
1              7         930            930          240  …          0
2              7        1240           1240          490  …          0
3              6        1400              0          330  …          0
4              8        1190            590          420  …          0
…            …        …            …            …  …        …
29137          6         620              0            0  …          0
29138          7         980              0            0  …          0
29139          7        2150           1390            0  …          0
29140          9        2280           1770          750  …          0
29141          6        1530            110            0  …          0

       zip_98155  zip_98166  zip_98168  zip_98177  zip_98178  zip_98188  \
0              0          0          0          0          0          0
1              0          0          0          0          0          0
```

```
2           0        0        0        0        0        0
3           0        0        0        0        0        0
4           0        0        0        0        0        0
…        …       …       …       …       …       …
29137       0        0        0        0        0        0
29138       0        0        0        0        0        0
29139       0        0        0        0        0        0
29140       0        0        0        0        0        0
29141       0        0        0        0        0        0

       zip_98198  zip_98199  zip_98288
0              0          0          0
1              0          0          0
2              0          0          0
3              0          0          0
4              0          0          0
…            …          …          …
29137          0          0          1
29138          0          0          1
29139          0          0          1
29140          0          0          0
29141          0          0          0

[28733 rows x 116 columns]
```

[70]:
```python
X3 = numcatwd[numcatwd.drop(['price'],axis=1).columns]
y3 = numcatwd["price"]
```

[71]:
```python
model3 = sm.OLS(endog=y3, exog=sm.add_constant(X3))
results3 = model3.fit()
results3.summary()
```

[71]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.750
Model:                            OLS   Adj. R-squared:                  0.749
Method:                 Least Squares   F-statistic:                     748.4
Date:                Sun, 02 Oct 2022   Prob (F-statistic):               0.00
Time:                        05:56:14   Log-Likelihood:            -4.0335e+05
No. Observations:               28733   AIC:                         8.069e+05
Df Residuals:                   28617   BIC:                         8.079e+05
Df Model:                         115
Covariance Type:            nonrobust
================================================================================
===================
```

```
                                   coef    std err          t      P>|t|
[0.025      0.975]
--------------------------------------------------------------------------------
-------------------
const                        3.896e+05   1.63e+05      2.391      0.017
7.02e+04    7.09e+05
bedrooms                    -1034.7857   2642.199     -0.392      0.695
-6213.620    4144.049
bathrooms                    3.086e+04   3799.891      8.122      0.000
2.34e+04    3.83e+04
sqft_living                   116.6925      8.692     13.425      0.000
99.656     133.729
sqft_lot                        0.5161      0.033     15.649      0.000
0.451       0.581
floors                      -6.996e+04   5158.105    -13.564      0.000
-8.01e+04   -5.99e+04
grade val                   -4.381e+04   2.32e+04     -1.889      0.059
-8.93e+04    1643.025
sqft_above                    164.3339      8.876     18.514      0.000
146.937     181.731
sqft_basement                  28.0405      6.636      4.226      0.000
15.034      41.047
sqft_garage                     2.4488      9.319      0.263      0.793
-15.817      20.714
sqft_patio                     29.0779      8.601      3.381      0.001
12.220      45.936
age                          -191.0020     99.535     -1.919      0.055
-386.096       4.092
waterfront_YES               2.102e+05   1.78e+04     11.786      0.000
1.75e+05    2.45e+05
greenbelt_YES                3.964e+04   1.17e+04      3.375      0.001
1.66e+04    6.27e+04
nuisance_YES                -4.998e+04   4941.874    -10.113      0.000
-5.97e+04   -4.03e+04
view_Excellent               3.841e+05   1.78e+04     21.572      0.000
3.49e+05    4.19e+05
view_Fair                    6.564e+04   2.27e+04      2.889      0.004
2.11e+04     1.1e+05
view_Good                    9.245e+04   1.29e+04      7.189      0.000
6.72e+04    1.18e+05
view_None                   -9.584e+04   7743.840    -12.376      0.000
-1.11e+05   -8.07e+04
condition_Fair              -7.393e+04   2.08e+04     -3.553      0.000
-1.15e+05   -3.31e+04
condition_Good               5.775e+04   4625.216     12.486      0.000
4.87e+04    6.68e+04
condition_Poor              -9.332e+04   4.09e+04     -2.282      0.023
```

```
-1.73e+05   -1.31e+04
condition_Very Good              1.237e+05   6495.594     19.046      0.000
1.11e+05    1.36e+05
grade desc_Better                3.325e+05   4.69e+04      7.089      0.000
2.41e+05    4.24e+05
grade desc_Excellent             8.058e+05   9.49e+04      8.491      0.000
6.2e+05    9.92e+05
grade desc_Fair                 -3.682e+04   4.89e+04     -0.753      0.451
-1.33e+05     5.9e+04
grade desc_Good                  1.046e+05   2.37e+04      4.414      0.000
5.81e+04    1.51e+05
grade desc_Low                  -2.827e+04   2.48e+04     -1.141      0.254
-7.68e+04    2.03e+04
grade desc_Luxury                8.761e+05   1.23e+05      7.125      0.000
6.35e+05    1.12e+06
grade desc_Mansion               1.382e+05   2.06e+05      0.669      0.503
-2.66e+05    5.43e+05
grade desc_Poor                 -1.429e+04   1.37e+05     -0.104      0.917
-2.84e+05    2.55e+05
grade desc_Substandard          -1.258e+05   3.27e+05     -0.385      0.700
-7.66e+05    5.14e+05
grade desc_Very                   5.91e+05   7.04e+04      8.395      0.000
4.53e+05    7.29e+05
heat_source_Electricity/Solar   -2.841e+04   4.05e+04     -0.701      0.483
-1.08e+05     5.1e+04
heat_source_Gas                  1.718e+04   4934.309      3.481      0.000
7506.417    2.68e+04
heat_source_Gas/Solar            1.302e+05   3.27e+04      3.983      0.000
6.61e+04    1.94e+05
heat_source_Oil                  4989.8017   7509.947      0.664      0.506
-9730.046    1.97e+04
heat_source_Oil/Solar             1.18e+05   1.52e+05      0.778      0.437
-1.79e+05    4.15e+05
heat_source_Other                1.169e+05   6.85e+04      1.708      0.088
-1.73e+04    2.51e+05
sewer_system_PRIVATE RESTRICTED -3.597e+05   1.39e+05     -2.593      0.010
-6.32e+05   -8.79e+04
sewer_system_PUBLIC              1.545e+04   6593.880      2.343      0.019
2523.221    2.84e+04
sewer_system_PUBLIC RESTRICTED   3.965e+04   2.14e+05      0.185      0.853
-3.81e+05     4.6e+05
zip_98002                        2.086e+04   2.02e+04      1.035      0.301
-1.86e+04    6.04e+04
zip_98003                       -1.236e+04    1.9e+04     -0.650      0.516
-4.96e+04    2.49e+04
zip_98004                        1.493e+06   2.49e+04     60.001      0.000
1.44e+06    1.54e+06
```

zip_98005	1.043e+06	2.67e+04	39.074	0.000
9.9e+05 1.09e+06				
zip_98006	7.748e+05	1.86e+04	41.675	0.000
7.38e+05 8.11e+05				
zip_98007	7.192e+05	2.74e+04	26.202	0.000
6.65e+05 7.73e+05				
zip_98008	7.404e+05	1.98e+04	37.298	0.000
7.01e+05 7.79e+05				
zip_98010	3027.0564	2.11e+04	0.143	0.886
-3.84e+04 4.44e+04				
zip_98011	4.681e+05	2.25e+04	20.828	0.000
4.24e+05 5.12e+05				
zip_98014	2.117e+05	2.73e+04	7.753	0.000
1.58e+05 2.65e+05				
zip_98019	2.292e+05	2.29e+04	9.991	0.000
1.84e+05 2.74e+05				
zip_98022	-2.282e+04	1.89e+04	-1.210	0.226
-5.98e+04 1.41e+04				
zip_98023	-4.084e+04	1.68e+04	-2.428	0.015
-7.38e+04 -7876.807				
zip_98024	3.586e+05	3.22e+04	11.133	0.000
2.95e+05 4.22e+05				
zip_98027	4.608e+05	2e+04	23.043	0.000
4.22e+05 5e+05				
zip_98028	3.958e+05	2.04e+04	19.416	0.000
3.56e+05 4.36e+05				
zip_98029	6.334e+05	2.12e+04	29.815	0.000
5.92e+05 6.75e+05				
zip_98030	9024.5580	1.98e+04	0.455	0.649
-2.98e+04 4.79e+04				
zip_98031	4.331e+04	1.8e+04	2.405	0.016
8018.853 7.86e+04				
zip_98032	4.265e+04	2.56e+04	1.669	0.095
-7437.222 9.27e+04				
zip_98033	1.09e+06	1.77e+04	61.683	0.000
1.06e+06 1.12e+06				
zip_98034	5.943e+05	1.7e+04	35.034	0.000
5.61e+05 6.28e+05				
zip_98038	1.025e+05	1.6e+04	6.400	0.000
7.11e+04 1.34e+05				
zip_98039	2.127e+06	6.2e+04	34.328	0.000
2.01e+06 2.25e+06				
zip_98040	1.114e+06	2.2e+04	50.701	0.000
1.07e+06 1.16e+06				
zip_98042	1.414e+04	1.55e+04	0.911	0.362
-1.63e+04 4.46e+04				
zip_98045	2.436e+05	1.87e+04	13.032	0.000

2.07e+05	2.8e+05				
zip_98047		5.536e+04	3.67e+04	1.509	0.131
-1.66e+04	1.27e+05				
zip_98050		5.835e+05	2.15e+05	2.717	0.007
1.63e+05	1e+06				
zip_98051		4.382e+04	3.96e+04	1.106	0.269
-3.38e+04	1.21e+05				
zip_98052		7.751e+05	1.78e+04	43.520	0.000
7.4e+05	8.1e+05				
zip_98053		6.101e+05	1.99e+04	30.587	0.000
5.71e+05	6.49e+05				
zip_98055		9.695e+04	2.39e+04	4.053	0.000
5.01e+04	1.44e+05				
zip_98056		2.77e+05	1.82e+04	15.226	0.000
2.41e+05	3.13e+05				
zip_98057		1.119e+05	2.99e+04	3.741	0.000
5.33e+04	1.71e+05				
zip_98058		9.568e+04	1.69e+04	5.674	0.000
6.26e+04	1.29e+05				
zip_98059		2.599e+05	1.76e+04	14.782	0.000
2.25e+05	2.94e+05				
zip_98065		3.606e+05	2.17e+04	16.641	0.000
3.18e+05	4.03e+05				
zip_98070		2.34e+05	2.58e+04	9.067	0.000
1.83e+05	2.85e+05				
zip_98072		5.284e+05	2.07e+04	25.521	0.000
4.88e+05	5.69e+05				
zip_98074		6.845e+05	1.94e+04	35.326	0.000
6.47e+05	7.22e+05				
zip_98075		6.932e+05	1.97e+04	35.154	0.000
6.55e+05	7.32e+05				
zip_98077		5.19e+05	2.36e+04	21.974	0.000
4.73e+05	5.65e+05				
zip_98092		-6.468e+04	1.73e+04	-3.730	0.000
-9.87e+04	-3.07e+04				
zip_98102		8.182e+05	2.91e+04	28.133	0.000
7.61e+05	8.75e+05				
zip_98103		6.312e+05	1.73e+04	36.563	0.000
5.97e+05	6.65e+05				
zip_98105		7.384e+05	2.17e+04	34.020	0.000
6.96e+05	7.81e+05				
zip_98106		2.519e+05	1.83e+04	13.772	0.000
2.16e+05	2.88e+05				
zip_98107		6.19e+05	1.95e+04	31.812	0.000
5.81e+05	6.57e+05				
zip_98108		2.78e+05	2.17e+04	12.833	0.000
2.36e+05	3.2e+05				

zip_98109	7.941e+05	3.04e+04	26.134	0.000	7.35e+05	8.54e+05
zip_98112	8.848e+05	2.26e+04	39.071	0.000	8.4e+05	9.29e+05
zip_98115	6.427e+05	1.71e+04	37.520	0.000	6.09e+05	6.76e+05
zip_98116	5.326e+05	2.03e+04	26.245	0.000	4.93e+05	5.72e+05
zip_98117	6.028e+05	1.72e+04	34.953	0.000	5.69e+05	6.37e+05
zip_98118	3.503e+05	1.78e+04	19.628	0.000	3.15e+05	3.85e+05
zip_98119	7.756e+05	2.4e+04	32.355	0.000	7.29e+05	8.23e+05
zip_98122	5.788e+05	1.99e+04	29.083	0.000	5.4e+05	6.18e+05
zip_98125	4.242e+05	1.87e+04	22.670	0.000	3.88e+05	4.61e+05
zip_98126	3.46e+05	1.95e+04	17.767	0.000	3.08e+05	3.84e+05
zip_98133	3.436e+05	1.73e+04	19.841	0.000	3.1e+05	3.77e+05
zip_98136	4.799e+05	2.19e+04	21.886	0.000	4.37e+05	5.23e+05
zip_98144	5.089e+05	1.97e+04	25.790	0.000	4.7e+05	5.48e+05
zip_98146	2.516e+05	1.94e+04	12.970	0.000	2.14e+05	2.9e+05
zip_98148	1.253e+05	3.36e+04	3.729	0.000	5.94e+04	1.91e+05
zip_98155	3.864e+05	1.84e+04	21.006	0.000	3.5e+05	4.22e+05
zip_98166	1.958e+05	2.06e+04	9.487	0.000	1.55e+05	2.36e+05
zip_98168	1.29e+05	2e+04	6.468	0.000	8.99e+04	1.68e+05
zip_98177	4.723e+05	2.18e+04	21.638	0.000	4.29e+05	5.15e+05
zip_98178	1.601e+05	2e+04	8.021	0.000	1.21e+05	1.99e+05
zip_98188	1.032e+05	2.48e+04	4.167	0.000	5.47e+04	1.52e+05
zip_98198	8.019e+04	1.92e+04	4.179	0.000	4.26e+04	1.18e+05
zip_98199	7.219e+05	2.01e+04	35.859	0.000	6.82e+05	7.61e+05
zip_98288	-1.644e+04	7.71e+04	-0.213	0.831		

```
-1.68e+05    1.35e+05
==============================================================================
Omnibus:                     6254.679   Durbin-Watson:                   1.931
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            97617.748
Skew:                           0.621   Prob(JB):                         0.00
Kurtosis:                      11.944   Cond. No.                     1.29e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 1.29e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
"""
```

Model 3 Conclusion This model has the price outliers removed with all of the nuemric and categoric featuresd. It has a much higher rsquared value compared to model 2a, explaining 75% of the price variance. The coefficient represents a house with zero living area costs about $390,000 and an increase of $120 a square foot, and $31,000 per bathroom added to the house. There are some features that add negative value. The p value shows that most of the features are statiscally relavent.

2 Distances From House to Points of Interests Effect on Price Variance

[72]:
```python
#loc_coord = np.array(list(zip(kcdfmo.lat,kcdfmo.long)))
#loc_coord
```

[73]:
```python
# Install geopy to calculate distances using latitude and longitude
!pip install geopy
```

```
Requirement already satisfied: geopy in /opt/conda/lib/python3.9/site-packages
(2.2.0)
Requirement already satisfied: geographiclib<2,>=1.49 in
/opt/conda/lib/python3.9/site-packages (from geopy) (1.52)
```

[74]:
```python
# Import geopy to use to calculate distance between latitude and longitude
import geopy.distance
```

2.0.1 Elementary Schools

[75]:
```python
# Compare all house coordinates to each feature and keep the smallest distance.
kcEschool_prox = []

for houseloc in loc_coord:
    sortlist=[]
```

```
    for schoolloc in kcEschool_loc:
        sortlist.append(geopy.distance.great_circle(houseloc, schoolloc).miles)
    kcEschool_prox.append(min(sortlist))
```

[76]:
```
# Check list of distances
#kcEschool_prox
```

[77]:
```
# Check range of distance for sensibility
min(kcEschool_prox), max(kcEschool_prox)
```

[77]: (0.020390431570705116, 32.29794313499601)

[78]:
```
# Add distance to the location to both the kcdfmo_feature df and the df with␣
 ↪dummies in it.
numcatwd['kcEschool_prox_mi'] = kcEschool_prox
kcdfmo_features['kcEschool_prox_mi'] = kcEschool_prox
```

2.0.2 Middle Schools

[79]:
```
# Compare all house coordinates to each feature and keep the smallest distance.
kcMschool_prox = []

for houseloc in loc_coord:
    sortlist=[]
    for schoolloc in kcMschool_loc:
        sortlist.append(geopy.distance.great_circle(houseloc, schoolloc).miles)
    kcMschool_prox.append(min(sortlist))
```

[80]:
```
# Check list of distances
#kcMschool_prox
```

[81]:
```
# Check range of distance for sensibility
min(kcMschool_prox), max(kcMschool_prox)
```

[81]: (0.023731571267947817, 30.449474540105175)

[82]:
```
# Add distance to the location to both the kcdfmo_feature df and the df with␣
 ↪dummies in it.
numcatwd['kcMschool_prox_mi'] = kcMschool_prox
kcdfmo_features['kcMschool_prox_mi'] = kcMschool_prox
```

2.0.3 High Schools

[83]:
```
# Compare all house coordinates to each feature and keep the smallest distance.
kcHschool_prox = []
```

```
for houseloc in loc_coord:
    sortlist=[]
    for schoolloc in kcHschool_loc:
        sortlist.append(geopy.distance.great_circle(houseloc, schoolloc).miles)
    kcHschool_prox.append(min(sortlist))
```

[84]:
```
# Check list of distances
#kcHschool_prox
```

[85]:
```
# Check range of distance for sensibility
min(kcHschool_prox), max(kcHschool_prox)
```

[85]:
```
(0.033460971302705166, 33.06807144741129)
```

[86]:
```
# Add distance to the location to both the kcdfmo_feature df and the df with␣
 ↪dummies in it.
numcatwd['kcHschool_prox_mi'] = kcHschool_prox
kcdfmo_features['kcHschool_prox_mi'] = kcHschool_prox
```

2.0.4 Solid Waste Disposal Sites (Landfills)

[87]:
```
# Compare all house coordinates to each feature and keep the smallest distance.
waste_prox = []

for houseloc in loc_coord:
    sortlist=[]
    for wasteloc in waste_loc:
        sortlist.append(geopy.distance.great_circle(houseloc, wasteloc).miles)
    waste_prox.append(min(sortlist))
```

[88]:
```
# Check list of distances
#waste_prox
```

[89]:
```
# Check range of distance for sensibility
min(waste_prox), max(waste_prox)
```

[89]:
```
(0.1161633399445429, 11.87994545536181)
```

[90]:
```
# Add distance to the location to both the kcdfmo_feature df and the df with␣
 ↪dummies in it.
numcatwd['waste_prox_mi'] = waste_prox
kcdfmo_features['waste_prox_mi'] = waste_prox
#numcatwd
```

2.0.5 Churches

[91]:
```
# Compare all house coordinates to each feature and keep the smallest distance.
church_prox = []

for houseloc in loc_coord:
    sortlist=[]
    for churchloc in waste_loc:
        sortlist.append(geopy.distance.great_circle(houseloc, churchloc).miles)
    church_prox.append(min(sortlist))
```

[92]:
```
# Check list of distances
#church_prox
```

[93]:
```
# Check range of distance for sensibility
min(church_prox), max(church_prox)
```

[93]: (0.1161633399445429, 11.87994545536181)

[94]:
```
# Add distance to the location to both the kcdfmo_feature df and the df with␣
↪dummies in it.
numcatwd['church_prox_mi'] = church_prox
kcdfmo_features['church_prox_mi'] = church_prox
#numcatwd
```

2.0.6 Parks

[95]:
```
# Compare all house coordinates to each feature and keep the smallest distance.
parks_prox = []

for houseloc in loc_coord:
    sortlist=[]
    for parkloc in parks_loc:
        sortlist.append(geopy.distance.great_circle(houseloc, parkloc).miles)
    parks_prox.append(min(sortlist))
```

[96]:
```
# Check list of distances
#parks_prox
```

[97]:
```
# Check range of distance for sensibility
min(parks_prox), max(parks_prox)
```

[97]: (0.009437956565269303, 4.196430433740691)

[98]:
```
# Add distance to the location to both the kcdfmo_feature df and the df with␣
↪dummies in it.
numcatwd['parks_prox_mi'] = parks_prox
```

```
kcdfmo_features['parks_prox_mi'] = parks_prox
#numcatwd
```

2.0.7 Transit Stations

[99]:
```
# Compare all house coordinates to each feature and keep the smallest distance.
transit_prox = []

for houseloc in loc_coord:
    sortlist=[]
    for transitloc in transit_loc:
        sortlist.append(geopy.distance.great_circle(houseloc, transitloc).miles)
    transit_prox.append(min(sortlist))
```

[100]:
```
# Check list of distances
#transit_prox
```

[101]:
```
# Check range of distance for sensibility
min(transit_prox), max(transit_prox)
```

[101]:
```
(0.05095511002476106, 46.83608645130408)
```

[102]:
```
# Add distance to the location to both the kcdfmo_feature df and the df with␣
↪dummies in it.
numcatwd['transit_prox_mi'] = transit_prox
kcdfmo_features['transit_prox_mi'] = transit_prox
#numcatwd
```

2.0.8 Starbucks

[103]:
```
# Compare all house coordinates to each feature and keep the smallest distance.
star_prox = []

for houseloc in loc_coord:
    sortlist=[]
    for starloc in star_loc:
        sortlist.append(geopy.distance.great_circle(houseloc, starloc).miles)
    star_prox.append(min(sortlist))
```

[104]:
```
# Check list of distances
#star_prox
```

[105]:
```
# Check range of distance for sensibility
min(star_prox), max(star_prox)
```

[105]:
```
(0.014098543413358052, 33.12862811739588)
```

[106]:
```
# Add distance to the location to both the kcdfmo_feature df and the df with␣
↪dummies in it.
numcatwd['star_prox_mi'] = star_prox
kcdfmo_features['star_prox_mi'] = star_prox
numcatwd
```

[106]:
```
            price  bedrooms  bathrooms  sqft_living  sqft_lot  floors  \
0        675000.0         4        1.0         1180      7140     1.0
1        750000.0         3        2.0         1830      7969     1.0
2        728000.0         4        2.0         2170      7520     1.0
3        565000.0         4        2.0         1400     10364     1.5
4        645000.0         3        2.0         1520      8250     1.0
…             …       …        …           …         …       …
29137    395000.0         1        1.0          620     10400     1.5
29138    328000.0         2        1.5          980      5000     2.0
29139    600000.0         3        2.5         3150    989234     1.5
29140   2451000.0         4        3.5         4050    204296     2.0
29141    750000.0         3        1.0         1530     33250     1.5

       grade val  sqft_above  sqft_basement  sqft_garage  …  zip_98199  \
0              7        1180              0            0  …          0
1              7         930            930          240  …          0
2              7        1240           1240          490  …          0
3              6        1400              0          330  …          0
4              8        1190            590          420  …          0
…             …          …              …          …  …        …
29137          6         620              0            0  …          0
29138          7         980              0            0  …          0
29139          7        2150           1390            0  …          0
29140          9        2280           1770          750  …          0
29141          6        1530            110            0  …          0

       zip_98288  kcEschool_prox_mi  kcMschool_prox_mi  kcHschool_prox_mi  \
0              0           0.340336           0.476492           1.241054
1              0           0.587415           0.944820           1.042188
2              0           0.423767           0.593220           1.269118
3              0           0.695935           0.912647           2.205839
4              0           0.225745           0.511211           1.132772
…             …                …                …                …
29137          1          26.201981          24.729601          26.615296
29138          1          24.514366          23.151868          24.854750
29139          1          27.940436          26.343423          28.442678
29140          0           2.033666           2.244347           4.415076
29141          0           2.331958           3.400030           4.539513

       waste_prox_mi  church_prox_mi  parks_prox_mi  transit_prox_mi  \
0           1.600536        1.600536       0.584268         2.298407
```

```
1            1.909783         1.909783       0.084149          1.260441
2            1.461359         1.461359       0.545286          2.341829
3            2.714450         2.714450       0.705959          1.548849
4            1.645249         1.645249       0.517346          2.383377
…               …              …             …              …
29137        0.940667         0.940667       1.312926         39.781199
29138        0.957244         0.957244       0.618282         37.916433
29139        2.951827         2.951827       3.324140         41.768004
29140        7.988979         7.988979       0.662403          9.530592
29141        6.876939         6.876939       0.105228         10.942480

       star_prox_mi
0          1.148006
1          0.306163
2          1.115312
3          0.704123
4          1.261119
…              …
29137     26.983483
29138     25.292599
29139     28.724867
29140      2.593044
29141      2.680708

[28733 rows x 124 columns]
```

2.0.9 Model 4 - numeric and categoric as dummies with location distances

This model removes the price outliers, adds the numeric features, endcodes categoric features and adds distances to schools, parks, churches, landfills, transit stations and Starbucks cafes.

[107]:
```
# Valus for X and y with numeric, encoded categoric, and distances to locations.
X4 = numcatwd[numcatwd.drop(['price'],axis=1).columns]
y4 = numcatwd["price"]

model4 = sm.OLS(endog=y4, exog=sm.add_constant(X4))
results4 = model4.fit()
results4.summary()
```

[107]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.752
Model:                            OLS   Adj. R-squared:                  0.750
Method:                 Least Squares   F-statistic:                     709.3
Date:                Sun, 02 Oct 2022   Prob (F-statistic):               0.00
```

```
Time:                        06:28:39   Log-Likelihood:            -4.0329e+05
No. Observations:               28733   AIC:                         8.068e+05
Df Residuals:                   28610   BIC:                         8.078e+05
Df Model:                         122
Covariance Type:            nonrobust
================================================================================
===================
                                     coef    std err          t      P>|t|
[0.025      0.975]
--------------------------------------------------------------------------------
-------------------
const                           3.986e+05   1.63e+05      2.447      0.014
7.93e+04    7.18e+05
bedrooms                       -1010.8143   2639.143     -0.383      0.702
-6183.657    4162.029
bathrooms                       3.022e+04   3793.150      7.967      0.000
2.28e+04    3.77e+04
sqft_living                      117.1472      8.675     13.504      0.000
100.144     134.150
sqft_lot                           0.5432      0.033     16.227      0.000
0.478       0.609
floors                          -7.01e+04   5159.864    -13.586      0.000
-8.02e+04      -6e+04
grade val                      -4.194e+04   2.31e+04     -1.812      0.070
-8.73e+04    3426.061
sqft_above                       165.5163      8.864     18.674      0.000
148.143     182.889
sqft_basement                     28.2330      6.624      4.262      0.000
15.250      41.216
sqft_garage                        2.4403      9.303      0.262      0.793
-15.794      20.675
sqft_patio                        31.1228      8.594      3.622      0.000
14.279      47.967
age                             -207.2143     99.504     -2.082      0.037
-402.247     -12.182
waterfront_YES                  2.191e+05   1.79e+04     12.265      0.000
1.84e+05    2.54e+05
greenbelt_YES                   3.371e+04   1.17e+04      2.870      0.004
1.07e+04    5.67e+04
nuisance_YES                   -5.025e+04   4938.999    -10.174      0.000
-5.99e+04   -4.06e+04
view_Excellent                  3.778e+05   1.78e+04     21.228      0.000
3.43e+05    4.13e+05
view_Fair                       6.237e+04   2.27e+04      2.749      0.006
1.79e+04    1.07e+05
view_Good                        9.15e+04   1.28e+04      7.124      0.000
6.63e+04    1.17e+05
```

view_None	-9.502e+04	7734.106	-12.285	0.000	-1.1e+05	-7.99e+04
condition_Fair	-7.565e+04	2.08e+04	-3.643	0.000	-1.16e+05	-3.49e+04
condition_Good	5.781e+04	4617.768	12.518	0.000	4.88e+04	6.69e+04
condition_Poor	-8.948e+04	4.08e+04	-2.192	0.028	-1.7e+05	-9451.396
condition_Very Good	1.233e+05	6487.286	19.008	0.000	1.11e+05	1.36e+05
grade desc_Better	3.255e+05	4.68e+04	6.953	0.000	2.34e+05	4.17e+05
grade desc_Excellent	7.892e+05	9.47e+04	8.330	0.000	6.04e+05	9.75e+05
grade desc_Fair	-3.097e+04	4.88e+04	-0.635	0.525	-1.27e+05	6.46e+04
grade desc_Good	1.013e+05	2.36e+04	4.283	0.000	5.49e+04	1.48e+05
grade desc_Low	-2.554e+04	2.47e+04	-1.033	0.302	-7.4e+04	2.29e+04
grade desc_Luxury	8.529e+05	1.23e+05	6.947	0.000	6.12e+05	1.09e+06
grade desc_Mansion	1.048e+05	2.06e+05	0.508	0.611	-2.99e+05	5.09e+05
grade desc_Poor	5784.2906	1.37e+05	0.042	0.966	-2.63e+05	2.75e+05
grade desc_Substandard	-2.135e+04	3.26e+05	-0.065	0.948	-6.61e+05	6.18e+05
grade desc_Very	5.801e+05	7.03e+04	8.255	0.000	4.42e+05	7.18e+05
heat_source_Electricity/Solar	-1.982e+04	4.05e+04	-0.490	0.624	-9.91e+04	5.95e+04
heat_source_Gas	1.684e+04	4936.302	3.411	0.001	7162.951	2.65e+04
heat_source_Gas/Solar	1.273e+05	3.26e+04	3.902	0.000	6.33e+04	1.91e+05
heat_source_Oil	4370.0633	7500.291	0.583	0.560	-1.03e+04	1.91e+04
heat_source_Oil/Solar	1.142e+05	1.51e+05	0.754	0.451	-1.83e+05	4.11e+05
heat_source_Other	1.379e+05	6.84e+04	2.016	0.044	3852.237	2.72e+05
sewer_system_PRIVATE RESTRICTED	-3.715e+05	1.38e+05	-2.684	0.007	-6.43e+05	-1e+05
sewer_system_PUBLIC	2197.5082	6898.410	0.319	0.750	-1.13e+04	1.57e+04
sewer_system_PUBLIC RESTRICTED	1.823e+04	2.14e+05	0.085	0.932		

```
-4.01e+05    4.38e+05
zip_98002                    3448.7485   2.03e+04      0.170      0.865
-3.63e+04    4.32e+04
zip_98003                   -3.216e+04   1.92e+04     -1.673      0.094
-6.98e+04    5512.676
zip_98004                    1.475e+06   2.51e+04     58.837      0.000
1.43e+06    1.52e+06
zip_98005                     1.03e+06   2.68e+04     38.498      0.000
9.78e+05    1.08e+06
zip_98006                    7.559e+05   1.88e+04     40.255      0.000
7.19e+05    7.93e+05
zip_98007                    7.054e+05   2.75e+04     25.610      0.000
6.51e+05    7.59e+05
zip_98008                     7.17e+05   2.03e+04     35.397      0.000
6.77e+05    7.57e+05
zip_98010                    3.283e+04   2.81e+04      1.170      0.242
-2.22e+04    8.78e+04
zip_98011                    4.447e+05   2.27e+04     19.569      0.000
4e+05    4.89e+05
zip_98014                    1.426e+05    3.9e+04      3.660      0.000
6.63e+04    2.19e+05
zip_98019                    2.626e+05   3.15e+04      8.344      0.000
2.01e+05    3.24e+05
zip_98022                    5.429e+04   3.06e+04      1.774      0.076
-5679.725    1.14e+05
zip_98023                   -7.992e+04   1.74e+04     -4.592      0.000
-1.14e+05   -4.58e+04
zip_98024                    2.991e+05   4.14e+04      7.218      0.000
2.18e+05     3.8e+05
zip_98027                    4.779e+05   2.27e+04     21.033      0.000
4.33e+05    5.22e+05
zip_98028                    3.524e+05    2.1e+04     16.764      0.000
3.11e+05    3.94e+05
zip_98029                    5.827e+05   2.64e+04     22.091      0.000
5.31e+05    6.34e+05
zip_98030                   -4.389e+04   2.11e+04     -2.081      0.037
-8.52e+04   -2560.691
zip_98031                    2147.9431   1.89e+04      0.113      0.910
-3.49e+04    3.92e+04
zip_98032                    -1.34e+04   2.63e+04     -0.510      0.610
-6.5e+04    3.81e+04
zip_98033                    1.069e+06   1.79e+04     59.650      0.000
1.03e+06     1.1e+06
zip_98034                    5.565e+05   1.76e+04     31.647      0.000
5.22e+05    5.91e+05
zip_98038                    1.356e+05   2.35e+04      5.763      0.000
8.95e+04    1.82e+05
```

zip_98039	2.094e+06	6.2e+04	33.768	0.000	1.97e+06	2.22e+06
zip_98040	1.075e+06	2.27e+04	47.360	0.000	1.03e+06	1.12e+06
zip_98042	-1.076e+04	1.95e+04	-0.552	0.581	-4.9e+04	2.75e+04
zip_98045	3.479e+05	4.36e+04	7.984	0.000	2.63e+05	4.33e+05
zip_98047	5.42e+04	3.68e+04	1.472	0.141	-1.8e+04	1.26e+05
zip_98050	5.463e+05	2.15e+05	2.536	0.011	1.24e+05	9.69e+05
zip_98051	1.495e+05	4.74e+04	3.157	0.002	5.67e+04	2.42e+05
zip_98052	7.527e+05	1.85e+04	40.766	0.000	7.17e+05	7.89e+05
zip_98053	5.841e+05	2.45e+04	23.872	0.000	5.36e+05	6.32e+05
zip_98055	7.031e+04	2.42e+04	2.900	0.004	2.28e+04	1.18e+05
zip_98056	2.462e+05	1.84e+04	13.345	0.000	2.1e+05	2.82e+05
zip_98057	1.05e+05	3.01e+04	3.485	0.000	4.59e+04	1.64e+05
zip_98058	9.217e+04	1.81e+04	5.101	0.000	5.68e+04	1.28e+05
zip_98059	2.598e+05	1.82e+04	14.279	0.000	2.24e+05	2.95e+05
zip_98065	3.666e+05	3.63e+04	10.093	0.000	2.95e+05	4.38e+05
zip_98070	3.091e+05	2.97e+04	10.412	0.000	2.51e+05	3.67e+05
zip_98072	5.22e+05	2.08e+04	25.088	0.000	4.81e+05	5.63e+05
zip_98074	6.555e+05	2.32e+04	28.310	0.000	6.1e+05	7.01e+05
zip_98075	6.59e+05	2.36e+04	27.958	0.000	6.13e+05	7.05e+05
zip_98077	5.309e+05	2.53e+04	20.970	0.000	4.81e+05	5.81e+05
zip_98092	-6.659e+04	1.81e+04	-3.686	0.000	-1.02e+05	-3.12e+04
zip_98102	7.992e+05	2.93e+04	27.237	0.000	7.42e+05	8.57e+05
zip_98103	6.092e+05	1.74e+04	34.936	0.000	5.75e+05	6.43e+05
zip_98105	6.958e+05	2.21e+04	31.506	0.000		

```
6.52e+05    7.39e+05
zip_98106                       2.451e+05   1.84e+04     13.293      0.000
2.09e+05    2.81e+05
zip_98107                       6.015e+05   2.02e+04     29.777      0.000
5.62e+05    6.41e+05
zip_98108                       2.571e+05   2.19e+04     11.729      0.000
2.14e+05       3e+05
zip_98109                       7.728e+05   3.06e+04     25.246      0.000
7.13e+05    8.33e+05
zip_98112                       8.609e+05   2.29e+04     37.668      0.000
8.16e+05    9.06e+05
zip_98115                       6.024e+05   1.77e+04     34.092      0.000
5.68e+05    6.37e+05
zip_98116                       4.951e+05   2.08e+04     23.807      0.000
4.54e+05    5.36e+05
zip_98117                        5.81e+05   1.78e+04     32.620      0.000
5.46e+05    6.16e+05
zip_98118                       3.166e+05   1.81e+04     17.445      0.000
2.81e+05    3.52e+05
zip_98119                       7.524e+05   2.42e+04     31.045      0.000
7.05e+05       8e+05
zip_98122                       5.529e+05   2.03e+04     27.209      0.000
5.13e+05    5.93e+05
zip_98125                       3.968e+05   1.91e+04     20.735      0.000
3.59e+05    4.34e+05
zip_98126                       3.254e+05   1.97e+04     16.511      0.000
2.87e+05    3.64e+05
zip_98133                       3.272e+05   1.76e+04     18.605      0.000
2.93e+05    3.62e+05
zip_98136                       4.406e+05   2.23e+04     19.786      0.000
3.97e+05    4.84e+05
zip_98144                       4.869e+05      2e+04     24.332      0.000
4.48e+05    5.26e+05
zip_98146                       2.326e+05   1.95e+04     11.929      0.000
1.94e+05    2.71e+05
zip_98148                       9.018e+04   3.39e+04      2.658      0.008
2.37e+04    1.57e+05
zip_98155                       3.696e+05   1.86e+04     19.921      0.000
3.33e+05    4.06e+05
zip_98166                       1.489e+05   2.15e+04      6.918      0.000
1.07e+05    1.91e+05
zip_98168                       1.103e+05   2.01e+04      5.499      0.000
7.1e+04     1.5e+05
zip_98177                       4.465e+05   2.22e+04     20.137      0.000
4.03e+05     4.9e+05
zip_98178                       1.559e+05      2e+04      7.816      0.000
1.17e+05    1.95e+05
```

```
zip_98188                          9.78e+04   2.48e+04      3.938      0.000
4.91e+04    1.46e+05
zip_98198                         4.075e+04   1.98e+04      2.062      0.039
2022.479    7.95e+04
zip_98199                         6.769e+05   2.13e+04     31.732      0.000
6.35e+05    7.19e+05
zip_98288                         6.439e+05   1.56e+05      4.121      0.000
3.38e+05     9.5e+05
kcEschool_prox_mi                -1.723e+04   5600.703     -3.077      0.002
-2.82e+04   -6255.010
kcMschool_prox_mi                 7325.1887   3436.158      2.132      0.033
590.158    1.41e+04
kcHschool_prox_mi                 1.806e+04   3306.755      5.462      0.000
1.16e+04    2.45e+04
waste_prox_mi                     6390.2824   1120.269      5.704      0.000
4194.503    8586.062
church_prox_mi                    6390.2824   1120.269      5.704      0.000
4194.503    8586.062
parks_prox_mi                    -1.374e+04   8262.992     -1.662      0.096
-2.99e+04    2460.431
transit_prox_mi                  -8442.3438   2246.172     -3.759      0.000
-1.28e+04   -4039.742
star_prox_mi                      -2.12e+04   3977.523     -5.329      0.000
-2.9e+04   -1.34e+04
==============================================================================
Omnibus:                     6211.588   Durbin-Watson:                   1.930
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            97777.116
Skew:                           0.610   Prob(JB):                         0.00
Kurtosis:                      11.954   Cond. No.                     1.22e+16
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The smallest eigenvalue is 7.59e-19. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
"""
```

Model 4 Conclusion This model has the price outliers removed with all of the nuemric, encoded categoric features and distances to select locations. The rsquared value remains the same, still explaining 75% of the price variance. The coefficient represents a house with zero living area costs about $\$400,000$ and an increase of \$120 a square foot, $\$30,000$ per bathroom, and $-\$210$ per year in the age of the house added to the house. There are some features that add negative value. The p value shows that most of the features are statiscally relavent.

2.1 LINE checks

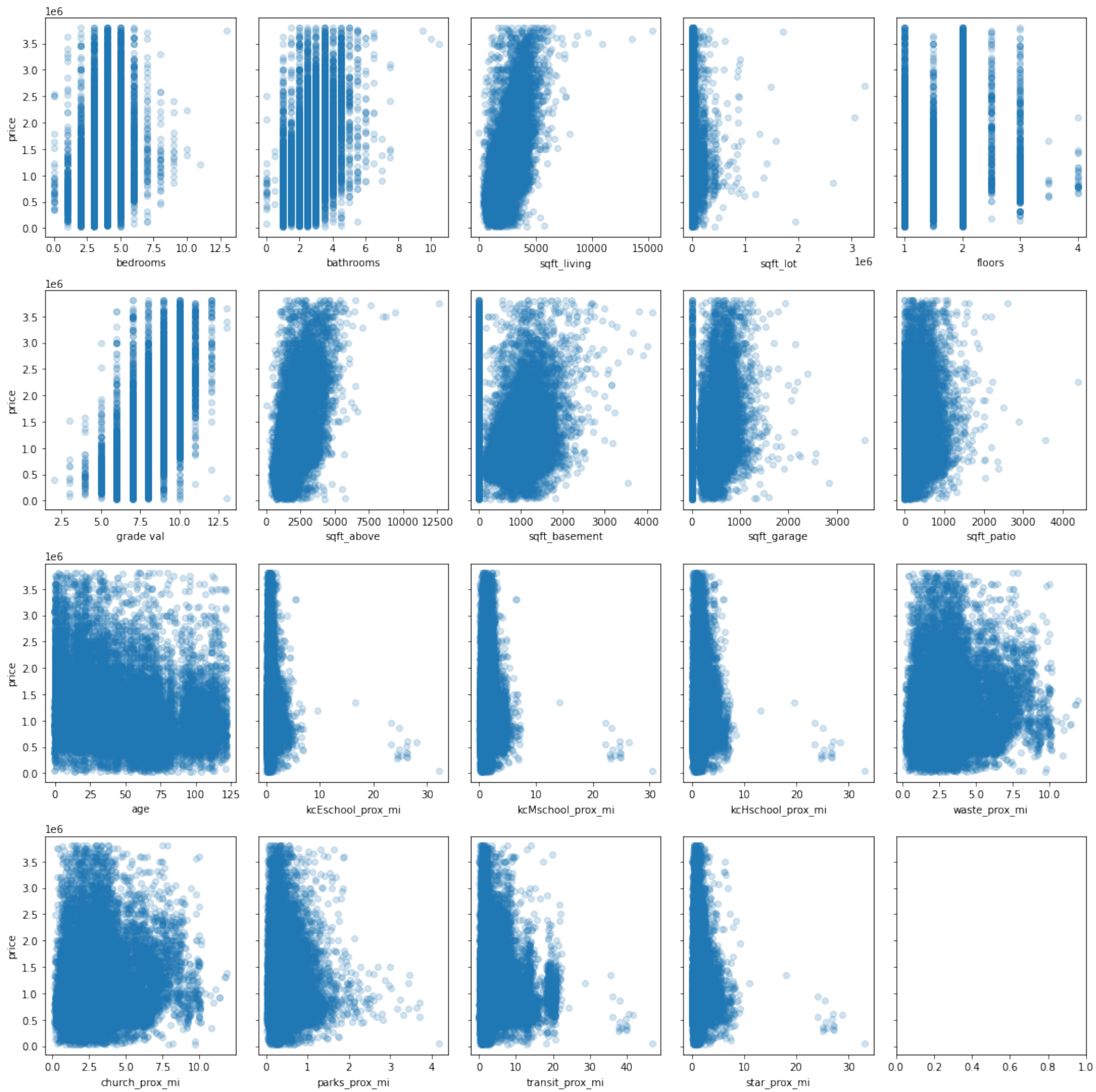
Check linear regression assumptions

2.1.1 Linearity - log tranformations to improve relationship

[108]:
```
# housing data with price outliers removed
kcdfmo_features.columns
```

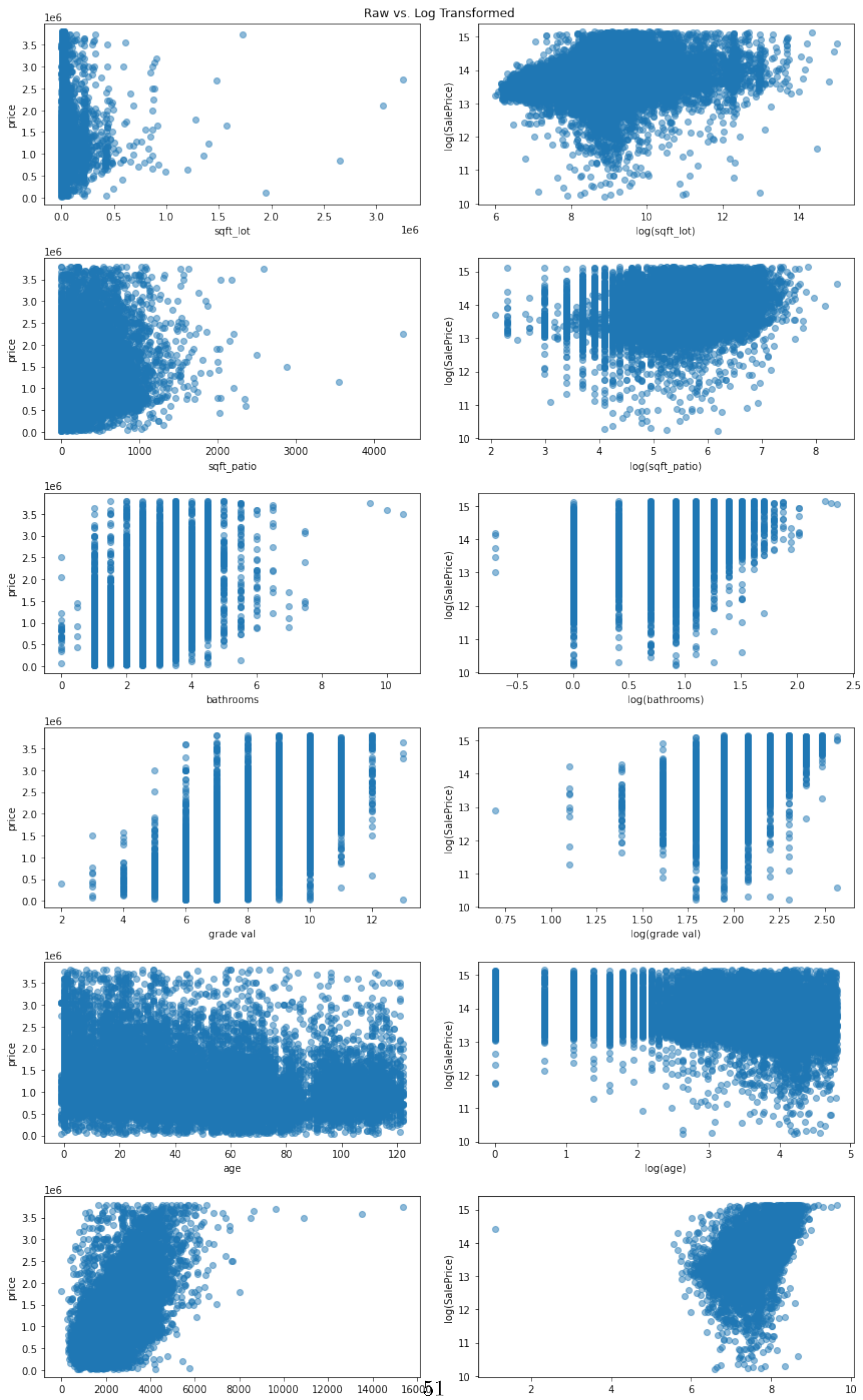
[108]:
```
Index(['price', 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors',
       'waterfront', 'greenbelt', 'nuisance', 'view', 'condition', 'grade val',
       'grade desc', 'heat_source', 'sewer_system', 'sqft_above',
       'sqft_basement', 'sqft_garage', 'sqft_patio', 'zip', 'age',
       'kcEschool_prox_mi', 'kcMschool_prox_mi', 'kcHschool_prox_mi',
       'waste_prox_mi', 'church_prox_mi', 'parks_prox_mi', 'transit_prox_mi',
       'star_prox_mi'],
      dtype='object')
```

[109]:
```
# Create scatter plots of each numeric feature compared to price.  Determine if␣
↪any are suitable for log transformation
# to check for linearity.
y = kcdfmo_features["price"]
X = kcdfmo_features[kcdfmo_features.select_dtypes('number').
↪drop(['price'],axis=1).columns]

fig, axes = plt.subplots(nrows=4, ncols=5, figsize=(15,15), sharey=True)

for i, column in enumerate(X.columns):
    # Locate applicable axes
    row = i // 5
    col = i % 5
    ax = axes[row][col]

    # Plot feature vs. y and label axes
    ax.scatter(X[column], y, alpha=0.2)
    ax.set_xlabel(column)
    if col == 0:
        ax.set_ylabel("price")

fig.tight_layout()
```



[110]:
```
# Shapes of the "sqft_lot", "sqft_patio", "bathrooms",'grade␣
↪val','age','sqft_living' graphs to be chceked for linearity
candidates = ["sqft_lot", "sqft_patio", "bathrooms",'grade␣
↪val','age','sqft_living']

fig, axes = plt.subplots(ncols=2, nrows=len(candidates), figsize=(12,20))

for i, column in enumerate(candidates):
    # Plot raw version
    left_ax = axes[i][0]
    left_ax.scatter(kcdfmo_features[column], y, alpha=0.5)
    left_ax.set_xlabel(column)
    left_ax.set_ylabel("price")
```

```
    # Plot log transformed version
    right_ax = axes[i][1]
    right_ax.scatter(np.log(kcdfmo_features[column]), np.log(y), alpha=0.5)
    right_ax.set_xlabel(f"log({column})")
    right_ax.set_ylabel("log(SalePrice)")

fig.suptitle("Raw vs. Log Transformed")

fig.tight_layout()
```


Raw vs. Log Transformed
price
sqft_lot
log(SalePrice)
log(sqft_lot)
price
sqft_patio
log(SalePrice)
log(sqft_patio)
price
bathrooms
log(SalePrice)
log(bathrooms)
price
grade val
log(SalePrice)
log(grade val)
price
age
log(SalePrice)
log(age)
price
sqft_living
log(SalePrice)
log(sqft_living)

log[price]

[111]:
```
# price changed to log of price to use in the log plots to check for linearity
y_log = np.log(y)
y_log.name = "log_price"
```

log transformation [sqft living]

[112]:
```
# Compare rsquared value for regular and transformed feature. If the rsquared␣
↪value is greater for the log transformed
# feature, then the feature is not linear and removed from the final model.
Xliving = kcdfmo_features['sqft_living']

Xliving_log = kcdfmo_features['sqft_living'].copy()
Xliving_log = np.log(Xliving_log)
Xliving_log.name = "sqft_living"

modelliving = sm.OLS(endog=y, exog=sm.add_constant(Xliving))
resultsliving = modelliving.fit()

modelliving_log = sm.OLS(endog=y_log, exog=sm.add_constant(Xliving_log))
resultsliving_log = modelliving_log.fit()

print('Linear Rsqrd value is', resultsliving.rsquared_adj ,
      'and log transformed Rsqrd is', resultsliving_log.rsquared_adj)
```

```
Linear Rsqrd value is 0.40743212410009355 and log transformed Rsqrd is
0.32675159877684856
```

log transformation [bathrooms]

[113]:
```
XBR = kcdfmo_features['bathrooms']

XBR_log = kcdfmo_features['bathrooms'].copy()
XBR_log = np.log(XBR+1)
XBR_log.name = "bathrooms"

modelBR_log = sm.OLS(endog=y_log, exog=sm.add_constant(XBR_log))
resultsBR_log = modelBR_log.fit()

modelBR = sm.OLS(endog=y, exog=sm.add_constant(XBR))
resultsBR = modelBR.fit()

print('Linear Rsqrd value is', resultsBR.rsquared_adj ,
      'and log transformed Rsqrd is', resultsBR_log.rsquared_adj)
```

Linear Rsqrd value is 0.2498813546769596 and log transformed Rsqrd is 0.22271660374570723

log transformation [sqft lot]

[114]:
```
Xlot = kcdfmo_features['sqft_lot']

Xlot_log = kcdfmo_features['sqft_lot'].copy()
Xlot_log = np.log(Xlot)
Xlot_log.name = "sqft_lot"

modellot = sm.OLS(endog=y, exog=sm.add_constant(Xlot))
resultslot = modellot.fit()

modellot_log = sm.OLS(endog=y_log, exog=sm.add_constant(Xlot_log))
resultslot_log = modellot_log.fit()

print('Linear Rsqrd value is', resultslot.rsquared_adj ,
      'and log transformed Rsqrd is', resultslot_log.rsquared_adj)
```

Linear Rsqrd value is 0.008795210587623115 and log transformed Rsqrd is 0.0198398489945798

log transformation [age]

[115]:
```
Xage = kcdfmo_features['age']

Xage_log = kcdfmo_features['age'].copy()
Xage_log = np.log(Xage_log+1.1)
Xage_log.name = "age"
Xage_log

modelage = sm.OLS(endog=y, exog=sm.add_constant(Xage))
resultsage = modelage.fit()

modelage_log = sm.OLS(endog=y_log, exog=sm.add_constant(Xage_log))
resultsage_log = modelage_log.fit()

print('Linear Rsqrd value is', resultsage.rsquared_adj ,
      'and log transformed Rsqrd is', resultsage_log.rsquared_adj)
```

Linear Rsqrd value is 0.021825472295047055 and log transformed Rsqrd is 0.021747497999161203

log transformation [sqft patio]

[116]:
```
Xpatio = kcdfmo_features['sqft_patio']

Xpatio_log = kcdfmo_features['sqft_patio'].copy()
```

```
y_log = np.log(y)
Xpatio_log.name = "sqft_patio"
Xpatio_log

modelpatio = sm.OLS(endog=y, exog=sm.add_constant(Xpatio))
resultspatio = modelpatio.fit()

modelpatio_log = sm.OLS(endog=y_log, exog=sm.add_constant(Xpatio_log))
resultspatio_log = modelpatio_log.fit()

print('Linear Rsqrd value is', resultspatio.rsquared_adj ,
      'and log transformed Rsqrd is', resultspatio_log.rsquared_adj)
```

```
Linear Rsqrd value is 0.08649474679350078 and log transformed Rsqrd is
0.075780138745012
```

log transformation [grade val]

[117]:
```
Xgradv = kcdfmo_features['grade val']

Xgradv_log = kcdfmo_features['grade val'].copy()
y_log = np.log(y)
Xgradv_log.name = "grade val"
Xgradv_log

modelgradv = sm.OLS(endog=y, exog=sm.add_constant(Xgradv))
resultsgradv = modelgradv.fit()

modelgradv_log = sm.OLS(endog=y_log, exog=sm.add_constant(Xgradv_log))
resultsgradv_log = modelgradv_log.fit()

print('Linear Rsqrd value is', resultsgradv.rsquared_adj ,
      'and log transformed Rsqrd is', resultsgradv_log.rsquared_adj)
```

```
Linear Rsqrd value is 0.38454274353656337 and log transformed Rsqrd is
0.34899023100492566
```

Conclusion The only seleceted value that showed a slight improvement with the log transformation is sqft_lot.

2.1.2 Independance - Check for Colinearity

[118]:
```
# Numeric features before one-hot encoding with location columns
numwd = kcdfmo_features.select_dtypes('number')
numwd
```

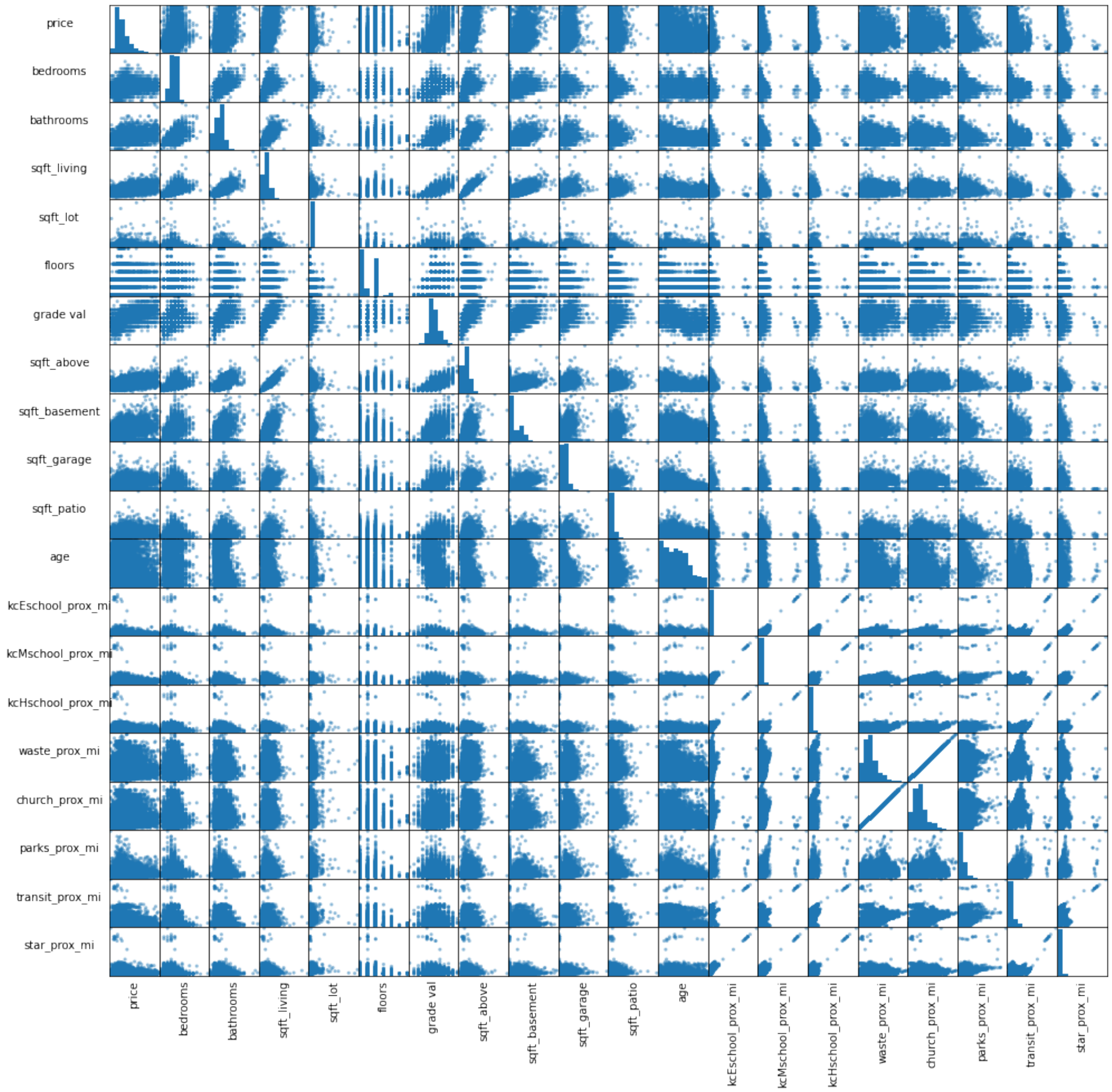
[118]:
```
          price  bedrooms  bathrooms  sqft_living  sqft_lot  floors  \
0      675000.0         4        1.0         1180      7140     1.0
```

```
1       750000.0          3        2.0          1830      7969      1.0
2       728000.0          4        2.0          2170      7520      1.0
3       565000.0          4        2.0          1400     10364      1.5
4       645000.0          3        2.0          1520      8250      1.0
…            …       …        …          …       …     …
29137   395000.0          1        1.0           620     10400      1.5
29138   328000.0          2        1.5           980      5000      2.0
29139   600000.0          3        2.5          3150    989234      1.5
29140  2451000.0          4        3.5          4050    204296      2.0
29141   750000.0          3        1.0          1530     33250      1.5

       grade val  sqft_above  sqft_basement  sqft_garage  sqft_patio  age  \
0              7        1180              0            0          40   53
1              7         930            930          240          90   14
2              7        1240           1240          490          60   49
3              6        1400              0          330         330   51
4              8        1190            590          420         200   40
…            …         …            …            …           …  …
29137          6         620              0            0         100   41
29138          7         980              0            0         260   18
29139          7        2150           1390            0        2360   39
29140          9        2280           1770          750        1250   37
29141          6        1530            110            0         360  117

       kcEschool_prox_mi  kcMschool_prox_mi  kcHschool_prox_mi  waste_prox_mi  \
0               0.340336           0.476492           1.241054       1.600536
1               0.587415           0.944820           1.042188       1.909783
2               0.423767           0.593220           1.269118       1.461359
3               0.695935           0.912647           2.205839       2.714450
4               0.225745           0.511211           1.132772       1.645249
…                    …                …                  …              …
29137          26.201981          24.729601          26.615296       0.940667
29138          24.514366          23.151868          24.854750       0.957244
29139          27.940436          26.343423          28.442678       2.951827
29140           2.033666           2.244347           4.415076       7.988979
29141           2.331958           3.400030           4.539513       6.876939

       church_prox_mi  parks_prox_mi  transit_prox_mi  star_prox_mi
0            1.600536       0.584268         2.298407      1.148006
1            1.909783       0.084149         1.260441      0.306163
2            1.461359       0.545286         2.341829      1.115312
3            2.714450       0.705959         1.548849      0.704123
4            1.645249       0.517346         2.383377      1.261119
…                 …            …               …            …
29137        0.940667       1.312926        39.781199     26.983483
29138        0.957244       0.618282        37.916433     25.292599
29139        2.951827       3.324140        41.768004     28.724867
```

```
29140        7.988979       0.662403         9.530592      2.593044
29141        6.876939       0.105228        10.942480      2.680708

[28733 rows x 20 columns]
```

[119]:

```python
sm = pd.plotting.scatter_matrix(numwd, figsize=[16, 16]);

# Rotates the text
[s.xaxis.label.set_rotation(90) for s in sm.reshape(-1)]
[s.yaxis.label.set_rotation(0) for s in sm.reshape(-1)]

#May need to offset label when rotating to prevent overlap of figure
[s.get_yaxis().set_label_coords(-1,0.5) for s in sm.reshape(-1)]

#Hide all ticks
[s.set_xticks(()) for s in sm.reshape(-1)]
[s.set_yticks(()) for s in sm.reshape(-1)]

plt.show()
```



[120]:
```
# creates a grid of scatter plots to see if there is any visual similarities␣
↪showing collinearity
numwd.corr()
```

[120]:

	price	bedrooms	bathrooms	sqft_living	sqft_lot	\
price	1.000000	0.338844	0.499907	0.638320	0.093967	
bedrooms	0.338844	1.000000	0.586554	0.637716	0.000055	
bathrooms	0.499907	0.586554	1.000000	0.761008	0.037066	
sqft_living	0.638320	0.637716	0.761008	1.000000	0.122382	
sqft_lot	0.093967	0.000055	0.037066	0.122382	1.000000	
floors	0.242799	0.188930	0.428891	0.359265	-0.021562	
grade val	0.620132	0.372245	0.625728	0.714901	0.058273	
sqft_above	0.561526	0.533788	0.653069	0.871328	0.131000	

sqft_basement	0.220932	0.223717	0.231375	0.298444	0.000762
sqft_garage	0.278429	0.295640	0.445255	0.490724	0.085516
sqft_patio	0.294154	0.174748	0.305389	0.380260	0.160135
age	-0.147850	-0.185714	-0.487674	-0.351148	-0.013511
kcEschool_prox_mi	0.002813	-0.018929	0.025380	0.066784	0.226781
kcMschool_prox_mi	0.033477	0.019280	0.068450	0.120564	0.208074
kcHschool_prox_mi	0.069501	0.019831	0.079194	0.141859	0.234680
waste_prox_mi	0.090734	0.093262	0.148759	0.192150	0.125995
church_prox_mi	0.090734	0.093262	0.148759	0.192150	0.125995
parks_prox_mi	0.001045	0.057124	0.093320	0.157874	0.270861
transit_prox_mi	-0.013417	0.045744	0.108096	0.159418	0.215240
star_prox_mi	-0.015442	0.010374	0.030927	0.105861	0.286632

	floors	grade val	sqft_above	sqft_basement	\
price	0.242799	0.620132	0.561526	0.220932	
bedrooms	0.188930	0.372245	0.533788	0.223717	
bathrooms	0.428891	0.625728	0.653069	0.231375	
sqft_living	0.359265	0.714901	0.871328	0.298444	
sqft_lot	-0.021562	0.058273	0.131000	0.000762	
floors	1.000000	0.477052	0.514269	-0.259659	
grade val	0.477052	1.000000	0.698390	0.100396	
sqft_above	0.514269	0.698390	1.000000	-0.130612	
sqft_basement	-0.259659	0.100396	-0.130612	1.000000	
sqft_garage	0.174709	0.504171	0.543259	-0.011878	
sqft_patio	0.117802	0.323891	0.293621	0.193133	
age	-0.533243	-0.493828	-0.442833	0.225486	
kcEschool_prox_mi	0.017875	0.033916	0.087140	-0.046124	
kcMschool_prox_mi	0.035221	0.077720	0.150940	-0.062655	
kcHschool_prox_mi	0.024720	0.108841	0.177237	-0.072491	
waste_prox_mi	0.087243	0.179676	0.250889	-0.112317	
church_prox_mi	0.087243	0.179676	0.250889	-0.112317	
parks_prox_mi	0.016197	0.099615	0.219161	-0.126110	
transit_prox_mi	0.098335	0.105340	0.258434	-0.204316	
star_prox_mi	-0.013434	0.039100	0.143626	-0.078753	

	sqft_garage	sqft_patio	age	kcEschool_prox_mi	\
price	0.278429	0.294154	-0.147850	0.002813	
bedrooms	0.295640	0.174748	-0.185714	-0.018929	
bathrooms	0.445255	0.305389	-0.487674	0.025380	
sqft_living	0.490724	0.380260	-0.351148	0.066784	
sqft_lot	0.085516	0.160135	-0.013511	0.226781	
floors	0.174709	0.117802	-0.533243	0.017875	
grade val	0.504171	0.323891	-0.493828	0.033916	
sqft_above	0.543259	0.293621	-0.442833	0.087140	
sqft_basement	-0.011878	0.193133	0.225486	-0.046124	
sqft_garage	1.000000	0.211407	-0.453354	0.069252	
sqft_patio	0.211407	1.000000	-0.146439	0.132981	

age	-0.453354	-0.146439	1.000000	-0.087443
kcEschool_prox_mi	0.069252	0.132981	-0.087443	1.000000
kcMschool_prox_mi	0.122184	0.126566	-0.128297	0.703401
kcHschool_prox_mi	0.153952	0.152746	-0.144474	0.690197
waste_prox_mi	0.265262	0.091312	-0.216875	0.101103
church_prox_mi	0.265262	0.091312	-0.216875	0.101103
parks_prox_mi	0.236210	0.128032	-0.190199	0.359166
transit_prox_mi	0.241224	0.110707	-0.267417	0.450416
star_prox_mi	0.129268	0.148054	-0.106110	0.779534

	kcMschool_prox_mi	kcHschool_prox_mi	waste_prox_mi	\
price	0.033477	0.069501	0.090734	
bedrooms	0.019280	0.019831	0.093262	
bathrooms	0.068450	0.079194	0.148759	
sqft_living	0.120564	0.141859	0.192150	
sqft_lot	0.208074	0.234680	0.125995	
floors	0.035221	0.024720	0.087243	
grade val	0.077720	0.108841	0.179676	
sqft_above	0.150940	0.177237	0.250889	
sqft_basement	-0.062655	-0.072491	-0.112317	
sqft_garage	0.122184	0.153952	0.265262	
sqft_patio	0.126566	0.152746	0.091312	
age	-0.128297	-0.144474	-0.216875	
kcEschool_prox_mi	0.703401	0.690197	0.101103	
kcMschool_prox_mi	1.000000	0.587465	0.190771	
kcHschool_prox_mi	0.587465	1.000000	0.248463	
waste_prox_mi	0.190771	0.248463	1.000000	
church_prox_mi	0.190771	0.248463	1.000000	
parks_prox_mi	0.357631	0.398232	0.297026	
transit_prox_mi	0.504618	0.610373	0.394429	
star_prox_mi	0.674813	0.679227	0.196927	

	church_prox_mi	parks_prox_mi	transit_prox_mi	\
price	0.090734	0.001045	-0.013417	
bedrooms	0.093262	0.057124	0.045744	
bathrooms	0.148759	0.093320	0.108096	
sqft_living	0.192150	0.157874	0.159418	
sqft_lot	0.125995	0.270861	0.215240	
floors	0.087243	0.016197	0.098335	
grade val	0.179676	0.099615	0.105340	
sqft_above	0.250889	0.219161	0.258434	
sqft_basement	-0.112317	-0.126110	-0.204316	
sqft_garage	0.265262	0.236210	0.241224	
sqft_patio	0.091312	0.128032	0.110707	
age	-0.216875	-0.190199	-0.267417	
kcEschool_prox_mi	0.101103	0.359166	0.450416	
kcMschool_prox_mi	0.190771	0.357631	0.504618	

```
kcHschool_prox_mi        0.248463      0.398232       0.610373
waste_prox_mi            1.000000      0.297026       0.394429
church_prox_mi           1.000000      0.297026       0.394429
parks_prox_mi            0.297026      1.000000       0.436286
transit_prox_mi          0.394429      0.436286       1.000000
star_prox_mi             0.196927      0.456406       0.481454

                   star_prox_mi
price                 -0.015442
bedrooms               0.010374
bathrooms              0.030927
sqft_living            0.105861
sqft_lot               0.286632
floors                -0.013434
grade val              0.039100
sqft_above             0.143626
sqft_basement         -0.078753
sqft_garage            0.129268
sqft_patio             0.148054
age                   -0.106110
kcEschool_prox_mi      0.779534
kcMschool_prox_mi      0.674813
kcHschool_prox_mi      0.679227
waste_prox_mi          0.196927
church_prox_mi         0.196927
parks_prox_mi          0.456406
transit_prox_mi        0.481454
star_prox_mi           1.000000
```

[121]:
```python
# Identify any correlation values between features greater than .75 (1 being
↪100% correlated)
abs(numwd.corr()) > 0.75
```

[121]:
```
                   price  bedrooms  bathrooms  sqft_living  sqft_lot  floors  \
price               True     False      False        False     False   False
bedrooms           False      True      False        False     False   False
bathrooms          False     False       True         True     False   False
sqft_living        False     False       True         True     False   False
sqft_lot           False     False      False        False      True   False
floors             False     False      False        False     False    True
grade val          False     False      False        False     False   False
sqft_above         False     False      False         True     False   False
sqft_basement      False     False      False        False     False   False
sqft_garage        False     False      False        False     False   False
sqft_patio         False     False      False        False     False   False
age                False     False      False        False     False   False
kcEschool_prox_mi  False     False      False        False     False   False
```

```
kcMschool_prox_mi  False     False      False        False     False   False
kcHschool_prox_mi  False     False      False        False     False   False
waste_prox_mi      False     False      False        False     False   False
church_prox_mi     False     False      False        False     False   False
parks_prox_mi      False     False      False        False     False   False
transit_prox_mi    False     False      False        False     False   False
star_prox_mi       False     False      False        False     False   False

                   grade val  sqft_above  sqft_basement  sqft_garage  \
price                  False       False          False        False
bedrooms               False       False          False        False
bathrooms              False       False          False        False
sqft_living            False        True          False        False
sqft_lot               False       False          False        False
floors                 False       False          False        False
grade val               True       False          False        False
sqft_above             False        True          False        False
sqft_basement          False       False           True        False
sqft_garage            False       False          False         True
sqft_patio             False       False          False        False
age                    False       False          False        False
kcEschool_prox_mi      False       False          False        False
kcMschool_prox_mi      False       False          False        False
kcHschool_prox_mi      False       False          False        False
waste_prox_mi          False       False          False        False
church_prox_mi         False       False          False        False
parks_prox_mi          False       False          False        False
transit_prox_mi        False       False          False        False
star_prox_mi           False       False          False        False

                   sqft_patio    age  kcEschool_prox_mi  kcMschool_prox_mi  \
price                   False  False              False              False
bedrooms                False  False              False              False
bathrooms               False  False              False              False
sqft_living             False  False              False              False
sqft_lot                False  False              False              False
floors                  False  False              False              False
grade val               False  False              False              False
sqft_above              False  False              False              False
sqft_basement           False  False              False              False
sqft_garage             False  False              False              False
sqft_patio               True  False              False              False
age                     False   True              False              False
kcEschool_prox_mi       False  False               True              False
kcMschool_prox_mi       False  False              False               True
kcHschool_prox_mi       False  False              False              False
waste_prox_mi           False  False              False              False
```

```
church_prox_mi          False  False              False              False
parks_prox_mi           False  False              False              False
transit_prox_mi         False  False              False              False
star_prox_mi            False  False               True              False

                  kcHschool_prox_mi  waste_prox_mi  church_prox_mi  \
price                         False          False           False
bedrooms                      False          False           False
bathrooms                     False          False           False
sqft_living                   False          False           False
sqft_lot                      False          False           False
floors                        False          False           False
grade val                     False          False           False
sqft_above                    False          False           False
sqft_basement                 False          False           False
sqft_garage                   False          False           False
sqft_patio                    False          False           False
age                           False          False           False
kcEschool_prox_mi             False          False           False
kcMschool_prox_mi             False          False           False
kcHschool_prox_mi              True          False           False
waste_prox_mi                 False           True            True
church_prox_mi                False           True            True
parks_prox_mi                 False          False           False
transit_prox_mi               False          False           False
star_prox_mi                  False          False           False

                  parks_prox_mi  transit_prox_mi  star_prox_mi
price                     False            False         False
bedrooms                  False            False         False
bathrooms                 False            False         False
sqft_living               False            False         False
sqft_lot                  False            False         False
floors                    False            False         False
grade val                 False            False         False
sqft_above                False            False         False
sqft_basement             False            False         False
sqft_garage               False            False         False
sqft_patio                False            False         False
age                       False            False         False
kcEschool_prox_mi         False            False          True
kcMschool_prox_mi         False            False         False
kcHschool_prox_mi         False            False         False
waste_prox_mi             False            False         False
church_prox_mi            False            False         False
parks_prox_mi              True            False         False
transit_prox_mi           False             True         False
```

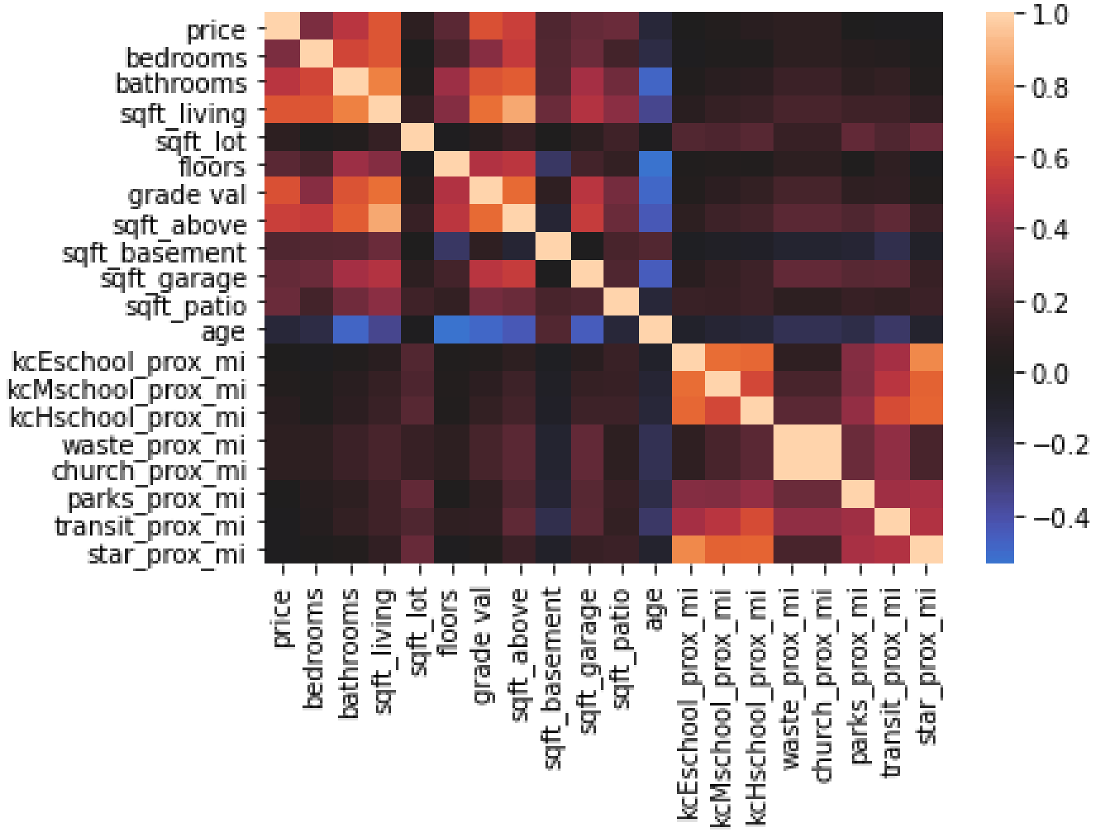
```
star_prox_mi                  False            False          True
```

[122]:
```
# Report any pair comboniation where the correlation value is greater than .75.
# A pair correlation greater than .75 indicates the two values are not␣
↪independent from each other.
# Removing one from the pair will take away the collinearity of the pair.
df = numwd.corr().abs().stack().reset_index().sort_values(0, ascending=False)

df['pairs'] = list(zip(df.level_0, df.level_1))

df.set_index(['pairs'], inplace = True)

df.drop(columns=['level_1', 'level_0'], inplace = True)

# cc for correlation coefficient
df.columns = ['cc']

df.drop_duplicates(inplace=True)

df[(df.cc>.75) & (df.cc<1)]
```

[122]:

pairs	cc
(sqft_living, sqft_above)	0.871328
(kcEschool_prox_mi, star_prox_mi)	0.779534
(sqft_living, bathrooms)	0.761008

[123]:
```
# Visual representation of the independance between each combination of␣
↪features.
sns.heatmap(numwd.corr(), center=0);
```



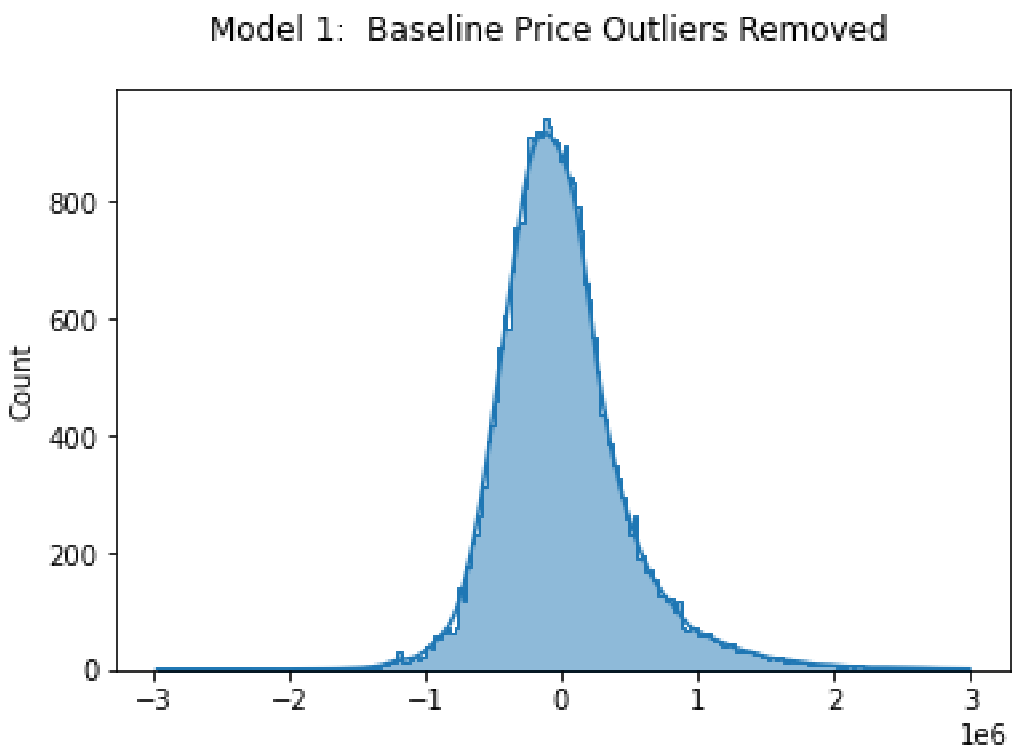
Conclusion There are three pair of features that appear to have enough correlation that they are not independant of each other. Removing ['sqft_above'], ['star_prox_mi'], and ['bathrooms'] will remove the collinearity.

2.1.3 Normality - check for normal distribution

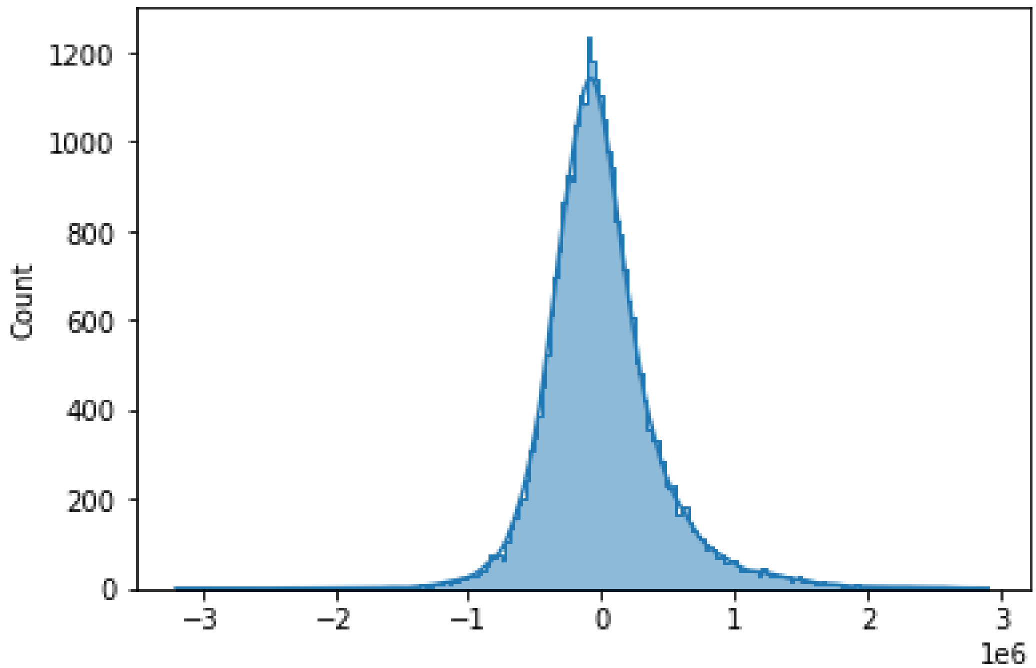
[124]:
```
from statsmodels.stats.stattools import jarque_bera
```

[125]:
```
#fig, axes = plt.subplots(nrows=5, ncols=3, figsize=(15,15), sharey=True)

fig, ax1 = plt.subplots()
sns.histplot(results1b.resid, bins=200, element="step", kde=True, ax=ax1)
ax.set_xlabel("Model Residuals")
fig.suptitle("Model 1:  Baseline Price Outliers Removed")

fig, ax2 = plt.subplots()
sns.histplot(results2.resid, bins=200, element="step", kde=True, ax=ax2)
ax.set_xlabel("Model Residuals")
fig.suptitle("Model 2:  Numeric Features")
```

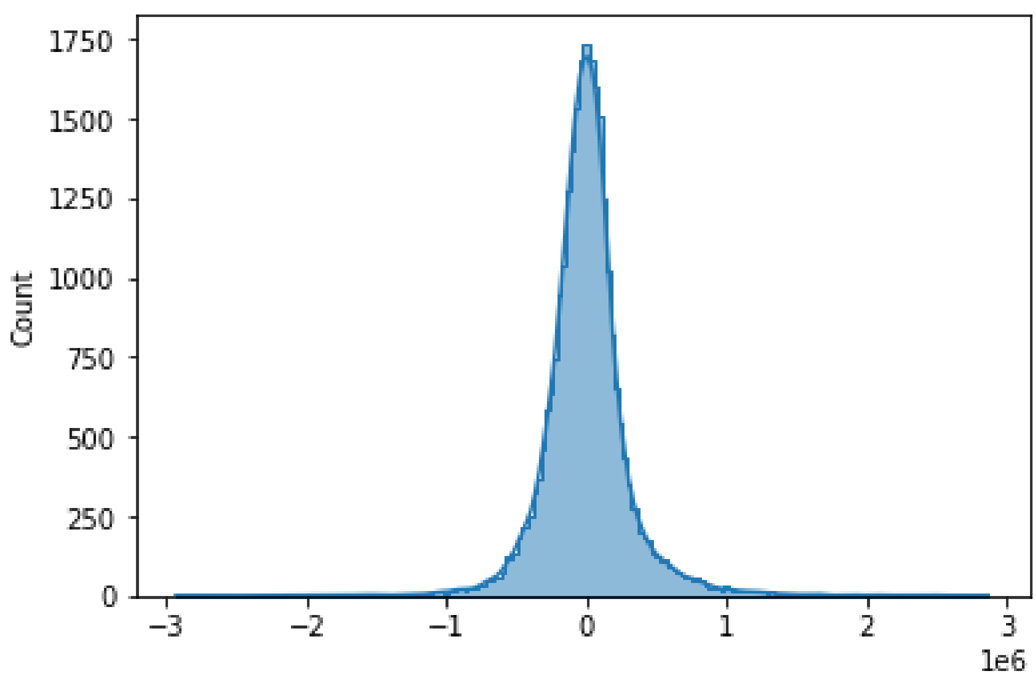
```
fig, ax3 = plt.subplots()
sns.histplot(results3.resid, bins=200, element="step", kde=True, ax=ax3)
ax.set_xlabel("Model Residuals")
fig.suptitle("Model 3:  All Features")

fig, ax4 = plt.subplots()
sns.histplot(results4.resid, bins=200, element="step", kde=True, ax=ax4)
ax.set_xlabel("Model Residuals")
fig.suptitle("Model 4:  All Features with Locations' Distances");
```

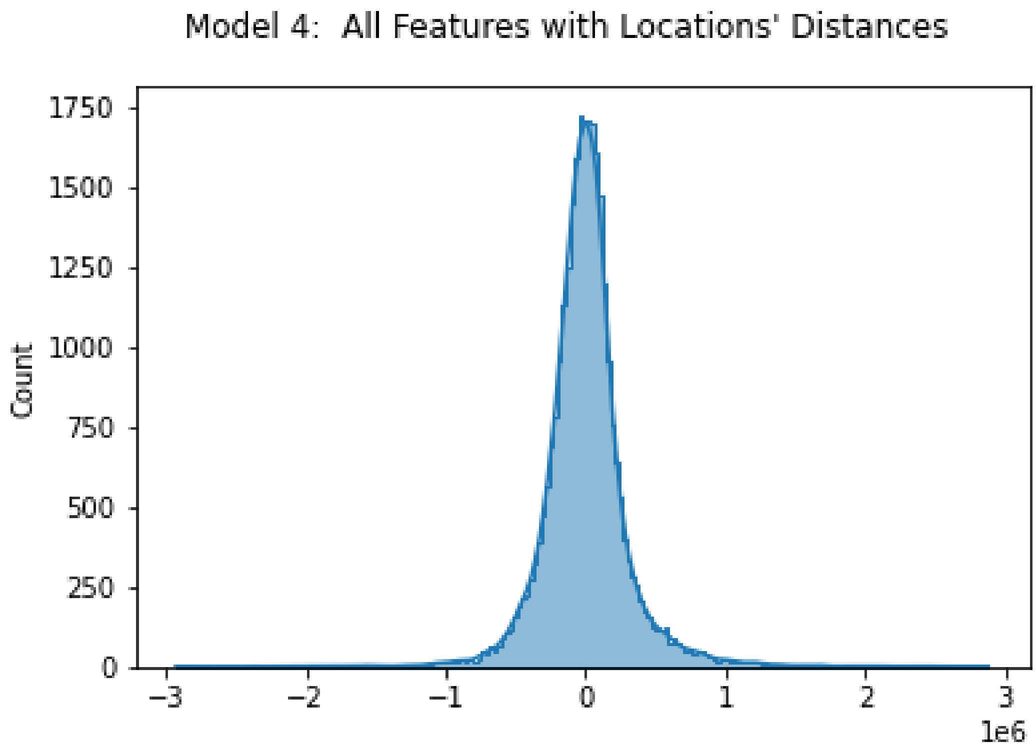


Model 2: Numeric Features

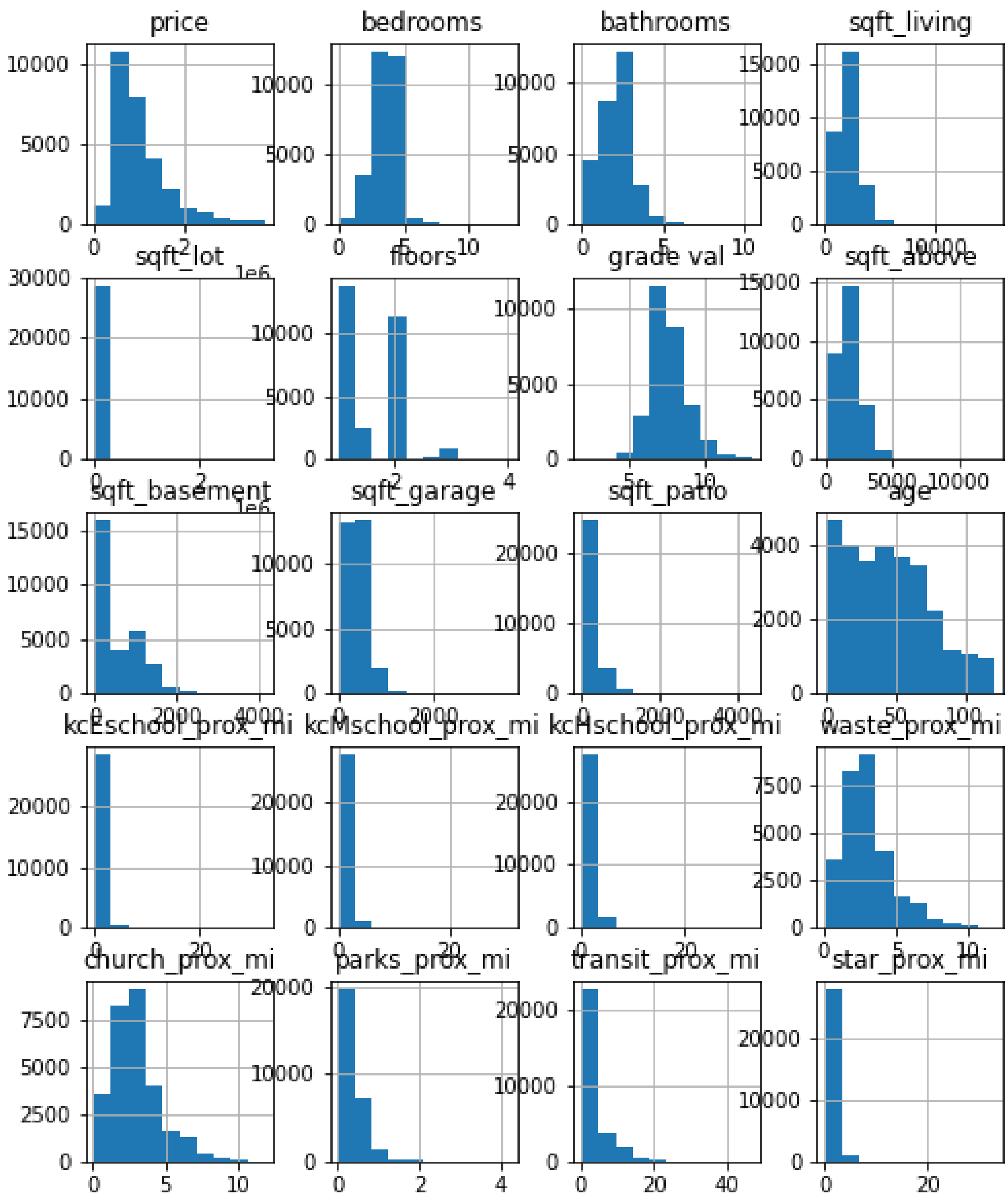


Model 3: All Features

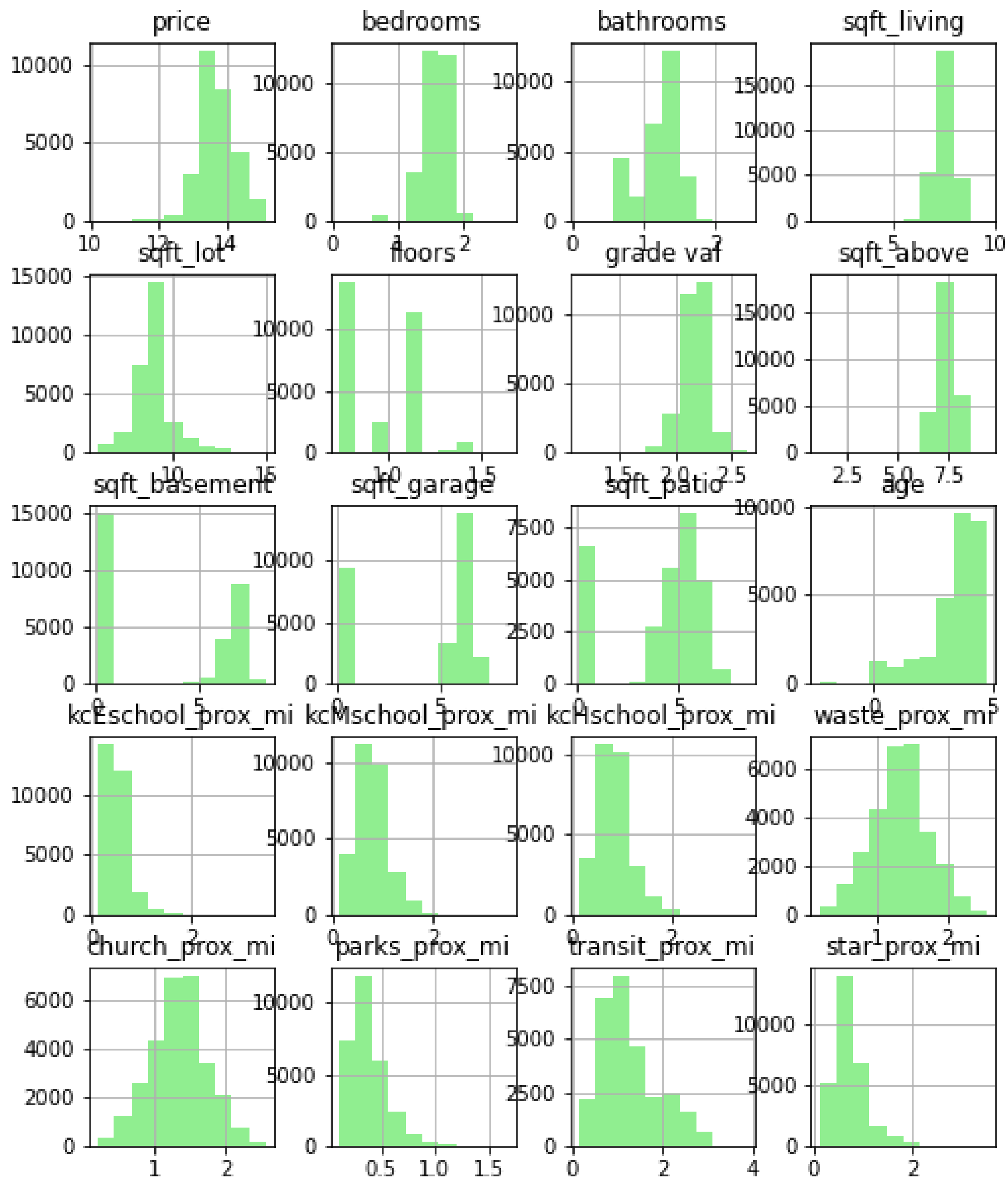





```
[126]: numcatwd[numwd.columns]\
           .hist(figsize=(8, 10));
```

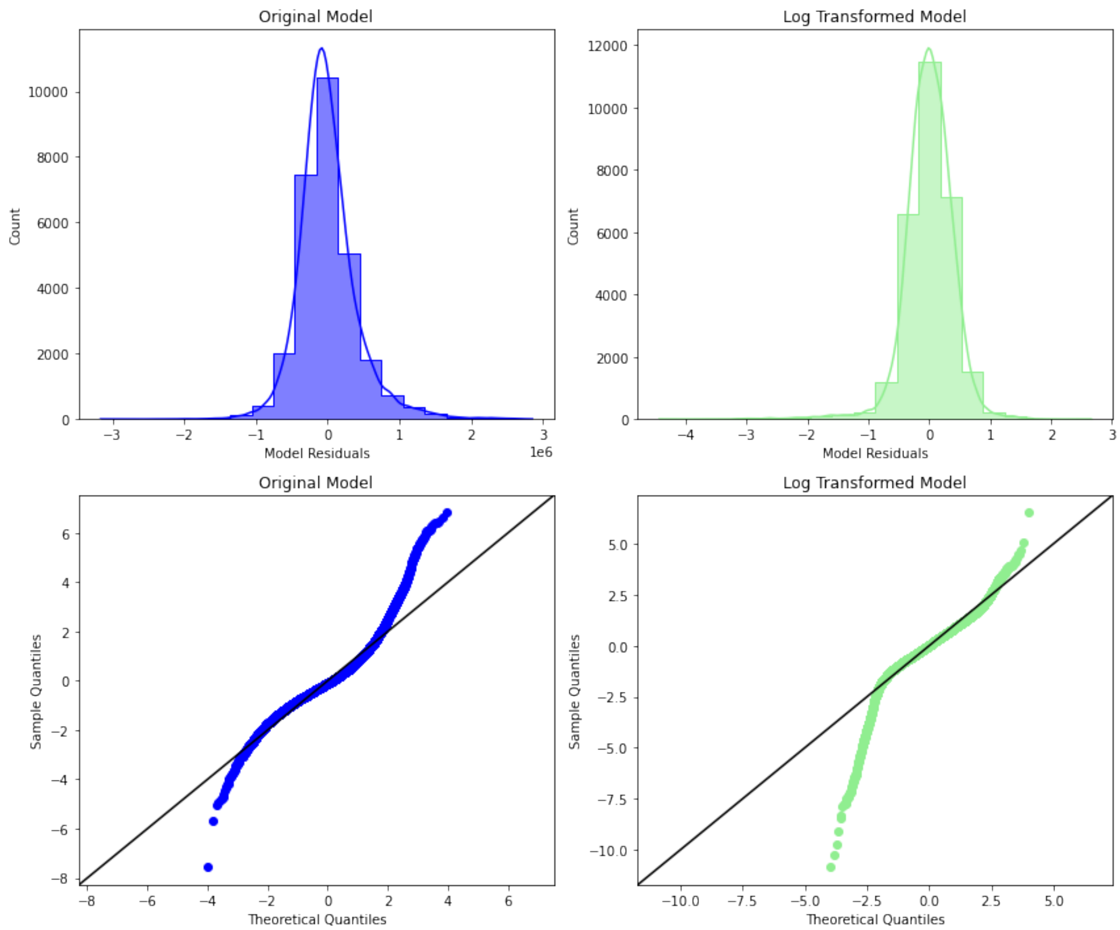


[127]:
```
np.log(numcatwd[numwd.columns]+1.1)\
     .hist(figsize=(8, 10), color="lightgreen");
```




```
[128]: import pandas as pd
       import numpy as np
       import statsmodels.api as sm

       y = kcdfmo_features["price"]
       X = kcdfmo_features[['bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'grade␣
        ↪val',
```

```
       'sqft_above','floors',
       'sqft_basement', 'sqft_garage', 'sqft_patio', 'age',
       'kcEschool_prox_mi', 'kcMschool_prox_mi', 'kcHschool_prox_mi',
       'waste_prox_mi', 'church_prox_mi', 'parks_prox_mi', 'transit_prox_mi',
       'star_prox_mi']]
model = sm.OLS(y, sm.add_constant(X))
results = model.fit()

# Build log transformed model
y_log = np.log(kcdfmo_features["price"])
X_log = pd.concat([np.log(kcdfmo_features[[ 'sqft_living', 'sqft_lot', 'grade␣
 ↪val',
       'sqft_above','sqft_basement', 'sqft_patio',
       'kcEschool_prox_mi', 'kcMschool_prox_mi', 'kcHschool_prox_mi',
       'waste_prox_mi', 'church_prox_mi', 'parks_prox_mi', 'transit_prox_mi',
       'star_prox_mi']]+1.1),
        kcdfmo_features[[ 'bedrooms','floors', 'age', 'bathrooms',␣
 ↪'sqft_garage']]], axis=1)
log_model = sm.OLS(y_log, sm.add_constant(X_log))
log_results = log_model.fit()

# Set up plot and properties of two models
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12,10))
resids = [results.resid, log_results.resid]
labels = ["Original Model", "Log Transformed Model"]
colors = ["blue", "lightgreen"]

# Plot histograms
for index, ax in enumerate(axes[0]):
    sns.histplot(resids[index], bins=20, element="step", kde=True,␣
 ↪color=colors[index], ax=ax)
    ax.set_xlabel("Model Residuals")
    ax.set_title(labels[index])

# Plot Q-Q plots
for index, ax in enumerate(axes[1]):
    sm.graphics.qqplot(resids[index], dist=stats.norm, line='45', fit=True,␣
 ↪ax=ax)
    scatter = ax.lines[0]
    line = ax.lines[1]
    scatter.set_markeredgecolor(colors[index])
    scatter.set_markerfacecolor(colors[index])
    line.set_color("black")
    ax.set_title(labels[index])

fig.tight_layout()
```



Conlusion The normality has a normal but narrow ditribution. This is seen both in the plots and the QQ plot with the S-like line.

2.1.4 Equal Variance

[129]:
```
# Check the Baseline model for variance with the Goldfeld-Quandt test.
from statsmodels.stats.diagnostic import het_goldfeldquandt
```

[130]:
```
# Using the X and y value from the baseline model with price outliers removed.
het_goldfeldquandt(y1b, X1b.values.reshape(-1,1), alternative='two-sided')
```

[130]:
```
(0.8800666205888453, 1.9551515481137923e-14, 'two-sided')
```

Conclusion The Goldfeld_Quandt test between price and sqft_living shows that it is heteroskedastic.

2.1.5 Final Model

Price outliers removed; all numeric features, encoded categoric features, and distances to locations added; nonlinear and dependant features removed.

[131]:
```
# Removed collinera features ('sqft_above','bathrooms','star_prox_mi') and a␣
↪non-linear feature ('sqft_lot').
import pandas as pd
import numpy as np

final_features = numcatwd.
↪drop(['price','sqft_lot','sqft_above','bathrooms','star_prox_mi'],axis=1).
↪columns
#final_features
#numcatwd
```

[132]:
```
# Features defined to determine final model
Xfinal = numcatwd[final_features]
yfinal = numcatwd["price"]
```

[133]:
```
# Final model with LINE adjustments
modelfinal = sm.OLS(endog=yfinal, exog=sm.add_constant(Xfinal))
resultsfinal = modelfinal.fit()
resultsfinal.summary()
```

[133]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.746
Model:                            OLS   Adj. R-squared:                  0.744
Method:                 Least Squares   F-statistic:                     710.4
Date:                Sun, 02 Oct 2022   Prob (F-statistic):               0.00
Time:                        06:29:34   Log-Likelihood:            -4.0363e+05
No. Observations:               28733   AIC:                         8.075e+05
Df Residuals:                   28614   BIC:                         8.085e+05
Df Model:                         118
Covariance Type:            nonrobust
================================================================================
===================
                                      coef    std err          t      P>|t|
[0.025      0.975]
--------------------------------------------------------------------------------
-------------------
const                            4.131e+05   1.65e+05      2.509      0.012
9.04e+04    7.36e+05
bedrooms                         2467.2544   2596.285      0.950      0.342
-2621.586    7556.095
```

sqft_living	271.7567	4.193	64.818	0.000	263.539	279.974
floors	-4.234e+04	5025.551	-8.425	0.000	-5.22e+04	-3.25e+04
grade val	-4.477e+04	2.34e+04	-1.913	0.056	-9.06e+04	1098.719
sqft_basement	-61.1925	4.379	-13.973	0.000	-69.777	-52.608
sqft_garage	49.1307	9.101	5.398	0.000	31.292	66.969
sqft_patio	37.2279	8.686	4.286	0.000	20.203	54.252
age	7.0298	94.910	0.074	0.941	-178.999	193.058
waterfront_YES	2.271e+05	1.81e+04	12.564	0.000	1.92e+05	2.62e+05
greenbelt_YES	3.157e+04	1.19e+04	2.661	0.008	8319.283	5.48e+04
nuisance_YES	-4.772e+04	4995.187	-9.554	0.000	-5.75e+04	-3.79e+04
view_Excellent	3.66e+05	1.8e+04	20.331	0.000	3.31e+05	4.01e+05
view_Fair	5.802e+04	2.3e+04	2.527	0.012	1.3e+04	1.03e+05
view_Good	9.097e+04	1.3e+04	7.003	0.000	6.55e+04	1.16e+05
view_None	-9.603e+04	7825.650	-12.271	0.000	-1.11e+05	-8.07e+04
condition_Fair	-5.522e+04	2.1e+04	-2.630	0.009	-9.64e+04	-1.41e+04
condition_Good	4.801e+04	4634.060	10.361	0.000	3.89e+04	5.71e+04
condition_Poor	-8.27e+04	4.13e+04	-2.002	0.045	-1.64e+05	-1730.275
condition_Very Good	1.086e+05	6445.193	16.855	0.000	9.6e+04	1.21e+05
grade desc_Better	3.399e+05	4.74e+04	7.178	0.000	2.47e+05	4.33e+05
grade desc_Excellent	8.326e+05	9.58e+04	8.690	0.000	6.45e+05	1.02e+06
grade desc_Fair	-3.602e+04	4.93e+04	-0.730	0.465	-1.33e+05	6.07e+04
grade desc_Good	1.075e+05	2.39e+04	4.497	0.000	6.07e+04	1.54e+05
grade desc_Low	-3.118e+04	2.5e+04	-1.247	0.212	-8.02e+04	1.78e+04
grade desc_Luxury	9.366e+05	1.24e+05	7.547	0.000		

```
6.93e+05    1.18e+06
grade desc_Mansion                 2.019e+05   2.09e+05      0.968      0.333
-2.07e+05    6.11e+05
grade desc_Poor                   -8826.9306   1.39e+05     -0.064      0.949
-2.81e+05    2.63e+05
grade desc_Substandard            -9.379e+04    3.3e+05     -0.284      0.776
-7.4e+05    5.53e+05
grade desc_Very                    6.099e+05   7.11e+04      8.582      0.000
4.71e+05    7.49e+05
heat_source_Electricity/Solar     -2.652e+04   4.09e+04     -0.648      0.517
-1.07e+05    5.37e+04
heat_source_Gas                    1.763e+04   4988.186      3.534      0.000
7853.381    2.74e+04
heat_source_Gas/Solar              1.334e+05    3.3e+04      4.043      0.000
6.87e+04    1.98e+05
heat_source_Oil                    1.376e+04   7511.079      1.832      0.067
-959.871    2.85e+04
heat_source_Oil/Solar              1.182e+05   1.53e+05      0.771      0.441
-1.82e+05    4.19e+05
heat_source_Other                  1.582e+05   6.92e+04      2.287      0.022
2.26e+04    2.94e+05
sewer_system_PRIVATE RESTRICTED -4.021e+05    1.4e+05     -2.871      0.004
-6.77e+05   -1.28e+05
sewer_system_PUBLIC               -9700.2216   6814.359     -1.423      0.155
-2.31e+04    3656.242
sewer_system_PUBLIC RESTRICTED     2.383e+04   2.17e+05      0.110      0.912
-4.01e+05    4.48e+05
zip_98002                          2.167e+04   2.04e+04      1.061      0.289
-1.84e+04    6.17e+04
zip_98003                          -2.15e+04   1.93e+04     -1.112      0.266
-5.94e+04    1.64e+04
zip_98004                          1.487e+06   2.53e+04     58.708      0.000
1.44e+06    1.54e+06
zip_98005                          1.052e+06    2.7e+04     38.977      0.000
9.99e+05     1.1e+06
zip_98006                          7.608e+05   1.89e+04     40.285      0.000
7.24e+05    7.98e+05
zip_98007                          7.233e+05   2.78e+04     26.049      0.000
6.69e+05    7.78e+05
zip_98008                          7.334e+05   2.04e+04     36.037      0.000
6.94e+05    7.73e+05
zip_98010                          2.038e+04   2.84e+04      0.719      0.472
-3.52e+04     7.6e+04
zip_98011                          4.596e+05   2.29e+04     20.029      0.000
4.15e+05    5.05e+05
zip_98014                          2.063e+05    3.8e+04      5.427      0.000
1.32e+05    2.81e+05
```

zip_98019	3.125e+05	3.07e+04	10.177	0.000	2.52e+05	3.73e+05
zip_98022	1.176e+05	3e+04	3.924	0.000	5.89e+04	1.76e+05
zip_98023	-6.554e+04	1.74e+04	-3.771	0.000	-9.96e+04	-3.15e+04
zip_98024	3.47e+05	4.14e+04	8.371	0.000	2.66e+05	4.28e+05
zip_98027	4.875e+05	2.28e+04	21.355	0.000	4.43e+05	5.32e+05
zip_98028	3.627e+05	2.12e+04	17.085	0.000	3.21e+05	4.04e+05
zip_98029	6.06e+05	2.58e+04	23.474	0.000	5.55e+05	6.57e+05
zip_98030	-2.052e+04	2.1e+04	-0.975	0.329	-6.18e+04	2.07e+04
zip_98031	1.474e+04	1.9e+04	0.776	0.438	-2.25e+04	5.19e+04
zip_98032	2059.2970	2.66e+04	0.077	0.938	-5e+04	5.42e+04
zip_98033	1.088e+06	1.8e+04	60.492	0.000	1.05e+06	1.12e+06
zip_98034	5.687e+05	1.77e+04	32.087	0.000	5.34e+05	6.03e+05
zip_98038	1.721e+05	2.28e+04	7.559	0.000	1.27e+05	2.17e+05
zip_98039	2.084e+06	6.28e+04	33.203	0.000	1.96e+06	2.21e+06
zip_98040	1.077e+06	2.29e+04	47.131	0.000	1.03e+06	1.12e+06
zip_98042	4720.8961	1.94e+04	0.243	0.808	-3.33e+04	4.27e+04
zip_98045	3.975e+05	4.24e+04	9.364	0.000	3.14e+05	4.81e+05
zip_98047	6.498e+04	3.72e+04	1.748	0.081	-7890.616	1.38e+05
zip_98050	5.388e+05	2.18e+05	2.473	0.013	1.12e+05	9.66e+05
zip_98051	2.025e+05	4.75e+04	4.265	0.000	1.09e+05	2.96e+05
zip_98052	7.647e+05	1.85e+04	41.296	0.000	7.28e+05	8.01e+05
zip_98053	6.172e+05	2.41e+04	25.663	0.000	5.7e+05	6.64e+05
zip_98055	8.942e+04	2.43e+04	3.685	0.000	4.19e+04	1.37e+05
zip_98056	2.567e+05	1.86e+04	13.828	0.000		

					2.2e+05	2.93e+05
zip_98057	1.256e+05	3.03e+04	4.140	0.000	6.62e+04	1.85e+05
zip_98058	1.073e+05	1.81e+04	5.943	0.000	7.19e+04	1.43e+05
zip_98059	2.761e+05	1.82e+04	15.156	0.000	2.4e+05	3.12e+05
zip_98065	4.121e+05	3.54e+04	11.654	0.000	3.43e+05	4.81e+05
zip_98070	2.651e+05	2.8e+04	9.475	0.000	2.1e+05	3.2e+05
zip_98072	5.309e+05	2.1e+04	25.276	0.000	4.9e+05	5.72e+05
zip_98074	6.693e+05	2.31e+04	29.017	0.000	6.24e+05	7.14e+05
zip_98075	6.734e+05	2.35e+04	28.655	0.000	6.27e+05	7.2e+05
zip_98077	5.273e+05	2.56e+04	20.594	0.000	4.77e+05	5.77e+05
zip_98092	-5.794e+04	1.83e+04	-3.173	0.002	-9.37e+04	-2.21e+04
zip_98102	8.135e+05	2.96e+04	27.498	0.000	7.56e+05	8.72e+05
zip_98103	6.265e+05	1.75e+04	35.781	0.000	5.92e+05	6.61e+05
zip_98105	7.171e+05	2.21e+04	32.423	0.000	6.74e+05	7.6e+05
zip_98106	2.593e+05	1.86e+04	13.975	0.000	2.23e+05	2.96e+05
zip_98107	6.198e+05	2.01e+04	30.769	0.000	5.8e+05	6.59e+05
zip_98108	2.634e+05	2.21e+04	11.916	0.000	2.2e+05	3.07e+05
zip_98109	7.933e+05	3.08e+04	25.739	0.000	7.33e+05	8.54e+05
zip_98112	8.711e+05	2.31e+04	37.722	0.000	8.26e+05	9.16e+05
zip_98115	6.159e+05	1.78e+04	34.630	0.000	5.81e+05	6.51e+05
zip_98116	5.139e+05	2.09e+04	24.566	0.000	4.73e+05	5.55e+05
zip_98117	5.9e+05	1.79e+04	32.914	0.000	5.55e+05	6.25e+05
zip_98118	3.251e+05	1.83e+04	17.728	0.000	2.89e+05	3.61e+05
zip_98119	7.697e+05	2.43e+04	31.635	0.000	7.22e+05	8.17e+05

zip_98122	5.68e+05	2.05e+04	27.768	0.000	5.28e+05	6.08e+05
zip_98125	4.12e+05	1.92e+04	21.468	0.000	3.74e+05	4.5e+05
zip_98126	3.399e+05	1.98e+04	17.153	0.000	3.01e+05	3.79e+05
zip_98133	3.416e+05	1.76e+04	19.465	0.000	3.07e+05	3.76e+05
zip_98136	4.648e+05	2.24e+04	20.759	0.000	4.21e+05	5.09e+05
zip_98144	4.975e+05	2.02e+04	24.639	0.000	4.58e+05	5.37e+05
zip_98146	2.354e+05	1.97e+04	11.937	0.000	1.97e+05	2.74e+05
zip_98148	1.059e+05	3.41e+04	3.101	0.002	3.89e+04	1.73e+05
zip_98155	3.85e+05	1.86e+04	20.689	0.000	3.49e+05	4.22e+05
zip_98166	1.639e+05	2.15e+04	7.609	0.000	1.22e+05	2.06e+05
zip_98168	1.106e+05	2.03e+04	5.449	0.000	7.08e+04	1.5e+05
zip_98177	4.601e+05	2.22e+04	20.759	0.000	4.17e+05	5.04e+05
zip_98178	1.529e+05	2.02e+04	7.577	0.000	1.13e+05	1.92e+05
zip_98188	1.047e+05	2.51e+04	4.173	0.000	5.55e+04	1.54e+05
zip_98198	4.106e+04	2e+04	2.055	0.040	1905.867	8.02e+04
zip_98199	6.911e+05	2.12e+04	32.656	0.000	6.5e+05	7.33e+05
zip_98288	3.625e+05	1.57e+05	2.307	0.021	5.45e+04	6.71e+05
kcEschool_prox_mi	-2.098e+04	5502.004	-3.814	0.000	-3.18e+04	-1.02e+04
kcMschool_prox_mi	7150.7390	3431.856	2.084	0.037	424.139	1.39e+04
kcHschool_prox_mi	1.574e+04	3258.692	4.830	0.000	9351.498	2.21e+04
waste_prox_mi	6729.2184	1132.802	5.940	0.000	4508.874	8949.563
church_prox_mi	6729.2184	1132.802	5.940	0.000	4508.874	8949.563
parks_prox_mi	-6946.1197	8182.788	-0.849	0.396	-2.3e+04	9092.529
transit_prox_mi	-1.068e+04	2200.320	-4.854	0.000		

```
-1.5e+04   -6367.007
==============================================================================
Omnibus:                     6625.724   Durbin-Watson:                   1.932
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            95985.309
Skew:                           0.710   Prob(JB):                         0.00
Kurtosis:                      11.841   Cond. No.                     3.52e+16
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The smallest eigenvalue is 1.31e-22. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
"""
```

Model Final Conclusion This final model has the price outliers removed with all of the nuemric, encoded categoric features and distances to select locations. Also, nonlinear and dependent featrues are removed. The rsquared value explains 74% of the price variance. The coefficient represents a house with zero living area costs about $410,000 and an increase of $270 a square foot of living space, $49 a square foot of garage, and $37 a sqare foot for patio space added to the house. There are some features that add negative value. The p value shows that most of the features are statiscally relavent.

2.2 Regression Results

In this analysis the best baseline model had an r-squared value that describes 41% of housing price variance. This was based on only one feature with a correlation of 63% to the prices. After adding numeric features on the MLS sheet the r-squared value increased to explaining 51% of housing price variance. The highest r-squared value described the housing price variance at 75% when the categoric features were introduced with one-hot encoding. The linearity was checked with log transformation. The features that were chosen all had a decrease in r-squared values except one that only increased by 1%. The indepednece checked showed three pair of features that were collinear. The normality of the model is normal but narrow. The narrow distribution is verified with the QQ graph. The variance of the data is not equally dispersed. The final r-squared value describes the housing price variance at 74% once the three collinear features and the one slightly non-linear feature were removed.

The approach to this analysis was to use everything in large chunks to see what made the most difference. This analysis shows that MLS categoric features make the most improvement with and increase of 25% to the model's predictability. The MLS numeric feature had a 10% increase above a base model. The distances to locations had a negligible effect on the model.

Next steps are to look at each feature within the categoric and numeric features to looking at p-values and find the minimum number of features with maximum predictability. Knowing the features that have the most impact will help both pairs, seller and buyer and realestate agents and appraiser, to be insynch with the sell price of a home.