Final Notebook

Please fill out:

- Student name: Andrew Levinton
- Student pace: self paced
- Scheduled project review date/time:
- Instructor name: Ahbineet Kulkarni
- Blog post URL:

Statsmodels debug

- This is because statsmodels was having version issues. this is a workaround
- The code below re-publishes the existing (but private) _centered function as a public attribute to the module already imported in RAM.

In [1]:
```
import  scipy.signal.signaltools

def _centered(arr, newsize):
    # Return the center newsize portion of the array.
    newsize = np.asarray(newsize)
    currsize = np.array(arr.shape)
    startind = (currsize - newsize) // 2
    endind = startind + newsize
    myslice = [slice(startind[k], endind[k]) for k in range(len(endind))]
    return arr[tuple(myslice)]

scipy.signal.signaltools._centered = _centered
```

Import necessary libraries

In [2]:
```
# raw data handling
import pandas as pd
import numpy as np
import datetime as dt

# data visualiztion
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import seaborn as sns

# regression modeling
import statsmodels.api as sm
from statsmodels.formula.api import ols
from statsmodels.stats.outliers_influence import variance_inflation_factor

import warnings # weird sns.distplot() warnings
warnings.filterwarnings("ignore")



plt.style.use('ggplot')
```

Define Functions

In [3]:
```
# Grabbing vifs

def get_vifs(data):
    # Get a list of the column names
    cols = data.columns
```

```python
    # Create an empty DataFrame to hold the VIF results
    vif_data = pd.DataFrame()

    # Loop through each column and calculate the VIF
    for i in range(len(cols)):
        vif = variance_inflation_factor(data[cols].values, i)
        vif_data = vif_data.append({'Variable': cols[i], 'VIF': vif}, ignore_index=True)

    # Print the VIF results
    return print(vif_data)


# get ols model and plot residual distribution
def get_OLS_model(name, X, y):
    model = sm.OLS(y, sm.add_constant(X))
    results = model.fit()
    model_residual = results.resid

    return print(results.summary()), plt.suptitle(f'Residual distribution for {name} model'), sns.distplot(model_resid



#get qq and histogram plots
def plot_hist_qq(df, target_col):
    """
    Creates a histogram and QQ-plot for a given dataframe and target column.

    Args:
        df (pandas.DataFrame): The dataframe to plot.
        target_col (str): The name of the target column.

    Returns:
        None
    """
    # Create subplots with 1 row and 2 columns
    fig, axs = plt.subplots(1, 2, figsize=(10, 5))

    # Plot histogram on the first subplot
    axs[0].hist(df[target_col], bins=30)
    axs[0].set_xlabel(target_col)
    axs[0].set_ylabel('Frequency')

    # Plot QQ-plot on the second subplot
    stats.probplot(df[target_col], plot=axs[1])
    axs[1].set_xlabel('Theoretical quantiles')
    axs[1].set_ylabel('Sample quantiles')

    # Adjust the layout and display the plot
    plt.tight_layout()
    plt.show()

# getting qqplots from stats model
def get_model_qqplots(data, y):
    # Set up the plot grid
    fig, axes = plt.subplots(nrows=5, ncols=6, figsize=(25, 18))

    # Loop through each variable in the DataFrame
    for i, var in enumerate(data.columns):
        # Fit a linear regression model
        X = sm.add_constant(data[var])
        model = sm.OLS(y, X).fit()

        # Calculate the residuals
        resid = model.resid

        # Create a QQ plot
        sm.qqplot(resid, line='s', ax=axes[i//6, i%6])
        axes[i//6, i%6].set_title(var)

    plt.tight_layout()
    plt.show()
```

Read in dataset, check length

In [4]:
```
cd data
```

```
C:\Users\alevi\Documents\Flatiron\dsc-data-science-env-config\Course_Folder\Phase_2\Housing_Linear_Model_Project\data
```

In [5]:
```
df = pd.read_csv('kc_house_data.csv')
len(df)
```

Out[5]:
```
30155
```

Dataset timeline

In [6]:
```
df['yr_built'].min(), df['yr_built'].max()
```

Out[6]:
```
(1900, 2022)
```

Checking dtypes

In [7]:
```
df.info()
```

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

Linear Model must meet the following assumptions:

Simple Linear Regression on select features

Assumption check:

- Is it linear?
- Is it normal?
    - histogram
    - QQ-plot
- Is it homoscedastic?

The process for building this linear model:

- Prep data for linear model regression: This involves dropping null values, dropping "bad data", as well as engineering features to assist in assuming linearization
- Key scores to look at:
- `R-Squared ( or the coefficient of determination)` - a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable. In other words, r-squared shows how well the data fit the regression model (the goodness of fit).
- `Correlation coefficients` - check to see what variables seem relatable to the target variable (price)
- `residual plots` - check how far data compares to the mean. Data should be normally distributed to avoid skewness of the mean
- `variance inflation factor` - level of statistical skew
- `Root mean squared erro` r - how far predictions fall from measured true values using Euclidean distance.
- `pvalues of independent variables` - measures how statistically significant the independent variables are

Data Preparation

Dropping nullls

In [8]:
```
df.dropna(inplace=True)
```

Recheck length

In [9]:
```
len(df)
```

Out[9]:
```
30111
```

Looking at Washington state

In [10]:
```
df['address'] = df['address'].str.lower()
```

In [11]:
```
df = df[df['address'].str.contains('washington')]
```

In [12]:
```
len(df)
```

Out[12]:
```
29208
```

Grabbing Zipcodes

In [13]:
```
df['zipcode'] = df['address'].apply(lambda x: x.split(',')[2].split(' ')[-1])
```

In [14]:
```
df['zipcode'] = df['zipcode'].astype(str)
```

In [15]:
```
df['zipcode'].unique()
```

Out[15]:
```
array(['98055', '98133', '98178', '98118', '98027', '98166', '98030',
       '98023', '98019', '98144', '98031', '98092', '98103', '98006',
       '98136', '98007', '98038', '98057', '98077', '98126', '98053',
       '98039', '98107', '98008', '98155', '98168', '98199', '98004',
       '98045', '98052', '98011', '98002', '98033', '98116', '98198',
       '98125', '98001', '98112', '98034', '98056', '98059', '98005',
       '98040', '98014', '98106', '98029', '98122', '98003', '98117',
       '98042', '98119', '98065', '98022', '98072', '98058', '98108',
```

```
'98115', '98074', '98105', '98024', '98146', '98109', '98102',
'98028', '98188', '98177', '98075', '98010', '98148', '98047',
'98032', '98070', '98051', '98288', '98354', '98272', '98296',
'98271', '98050', '63090', 'seattle', '98387', '15301', '98251',
'98223', '98338', '98224', '98372', '98663', '99202', '99403',
'98422', '99203', '99223', '98270'], dtype=object)
```

Categorizing waterfronts

In [16]:

```
duwamish = ['98168']
elliot_bay_zips= ['98119','98104','98129','98132','98127','98125','98195','98101','98134','98170','98139','98131','981
puget_sound = ['98071','98083','98013','98070','98031','98131','98063','98195','98207','98190']
lake_union = ['98109']
ship_canal = ['00000']
lake_washington = ['98072','98077']
lake_sammamish = ['98074','98075','98029']
other = ['00000']
river_slough_waterfronts = ['00000']

df['waterfront_loc'] = df['zipcode'].apply(lambda x: 'Duwamish' if x=='98168'\
                                           else 'Elliot Bay' if x in elliot_bay_zips\
                                           else 'Puget Sound' if x in puget_sound\
                                           else 'Lake Union' if x in lake_union\
                                           else 'ship canal' if x in ship_canal\
                                           else 'Lake Washington' if x in lake_washington\
                                           else 'Lake Sammamish' if x in lake_sammamish\
                                           else 'other')
```

In [17]:

```
df['waterfront_loc'].value_counts()
```

Out[17]:

```
other              25497
Lake Sammamish      1159
Elliot Bay           730
Puget Sound          721
Lake Washington      589
Duwamish             383
Lake Union           129
Name: waterfront_loc, dtype: int64
```

Filter by state of Washington Zipcodes (assuming seattle is its own zipcode)

In [18]:

```
df = df[df['zipcode'].str.startswith('98') | df['zipcode'].str.contains('seattle')]
```

One Hot Encoding Waterfronts

In [19]:

```
waterfront_dummies = pd.get_dummies(df['waterfront_loc'], prefix='water', drop_first=True)
```

In [20]:

```
waterfront_dummies
```

Out[20]:

	water_Elliot Bay	water_Lake Sammamish	water_Lake Union	water_Lake Washington	water_Puget Sound	water_other
0	0	0	0	0	0	1
1	0	0	0	0	0	1
2	0	0	0	0	0	1
3	0	0	0	0	0	1
4	0	0	0	0	0	1
...	...	...	...	...	...	...
30150	0	0	0	0	0	1
30151	0	0	0	0	0	1

	water_Elliot Bay	water_Lake Sammamish	water_Lake Union	water_Lake Washington	water_Puget Sound	water_other
30152	0	0	0	0	0	1
30153	0	0	0	0	0	1
30154	0	0	0	0	0	1

29200 rows × 6 columns

In [21]:
```
len(df)
```
Out[21]:
```
29200
```
In [22]:
```
len(df) == len(waterfront_dummies)
```
Out[22]:
```
True
```
In [23]:
```
df = pd.concat([df,waterfront_dummies], axis=1)
```

replacing seattle with seattle zipcode

In [24]:
```
df['zipcode'] = df['zipcode'].apply(lambda x: '98101' if x=='seattle' else x)
```

recheck zipcodes

In [25]:
```
df['zipcode'].unique()
```
Out[25]:
```
array(['98055', '98133', '98178', '98118', '98027', '98166', '98030',
       '98023', '98019', '98144', '98031', '98092', '98103', '98006',
       '98136', '98007', '98038', '98057', '98077', '98126', '98053',
       '98039', '98107', '98008', '98155', '98168', '98199', '98004',
       '98045', '98052', '98011', '98002', '98033', '98116', '98198',
       '98125', '98001', '98112', '98034', '98056', '98059', '98005',
       '98040', '98014', '98106', '98029', '98122', '98003', '98117',
       '98042', '98119', '98065', '98022', '98072', '98058', '98108',
       '98115', '98074', '98105', '98024', '98146', '98109', '98102',
       '98028', '98188', '98177', '98075', '98010', '98148', '98047',
       '98032', '98070', '98051', '98288', '98354', '98272', '98296',
       '98271', '98050', '98101', '98387', '98251', '98223', '98338',
       '98224', '98372', '98663', '98422', '98270'], dtype=object)
```
In [26]:
```
len(df['zipcode'].unique())
```
Out[26]:
```
89
```

Adding in Engineered Zipcode Data Generated from GreatSchools API

The csv file that is being imported was generated using an extensive process of requests and data aggregation of school ratings by zipcode. To view the process of retrieval and aggregation please visit the file Final_Exploratory_Data_Analysis.ipynb in the notebooks folder.

In [27]:
```
cd ..
```
```
C:\Users\alevi\Documents\Flatiron\dsc-data-science-env-config\Course_Folder\Phase_2\Housing_Linear_Model_Project
```
In [28]:
```
school_ratings = pd.read_csv('school_ratings_by_zipcode.csv')
school_ratings['zipcode'] = school_ratings['zipcode'].astype(str)
```

Assigning average school ratings to corresponding zipcodes

In [29]:

```
# Create a dictionary from the zipcode dataframe
zip_dict = school_ratings.set_index('zipcode')['avg_rating'].to_dict()

# Load your larger dataframe


# Assign the ratings from the zipcode dictionary to the large dataframe
df['school_rating'] = df['zipcode'].apply(lambda x: zip_dict.get(x, None))

# The above line applies the lambda function to each element of the 'zipcode' column of the large dataframe.
# If the zipcode is present in the zip_dict, its corresponding rating is assigned to the 'rating' column.
# If not, None is assigned.

# You can then save the updated large dataframe to a new csv file
df['school_rating'].isnull().sum()
```

Out[29]:

```
11315
```

Filling nulls with mean value

In [30]:

```
mean_val = df['school_rating'].mean()
```

In [31]:

```
df['school_rating'] = df['school_rating'].fillna(mean_val)
```

In [32]:

```
df['school_rating'].isnull().sum()
```

Out[32]:

```
0
```

Observing correlation matrix for possible features that can be used with the price

In [33]:

```
df.corr()['price'].abs().sort_values(ascending=False)
```

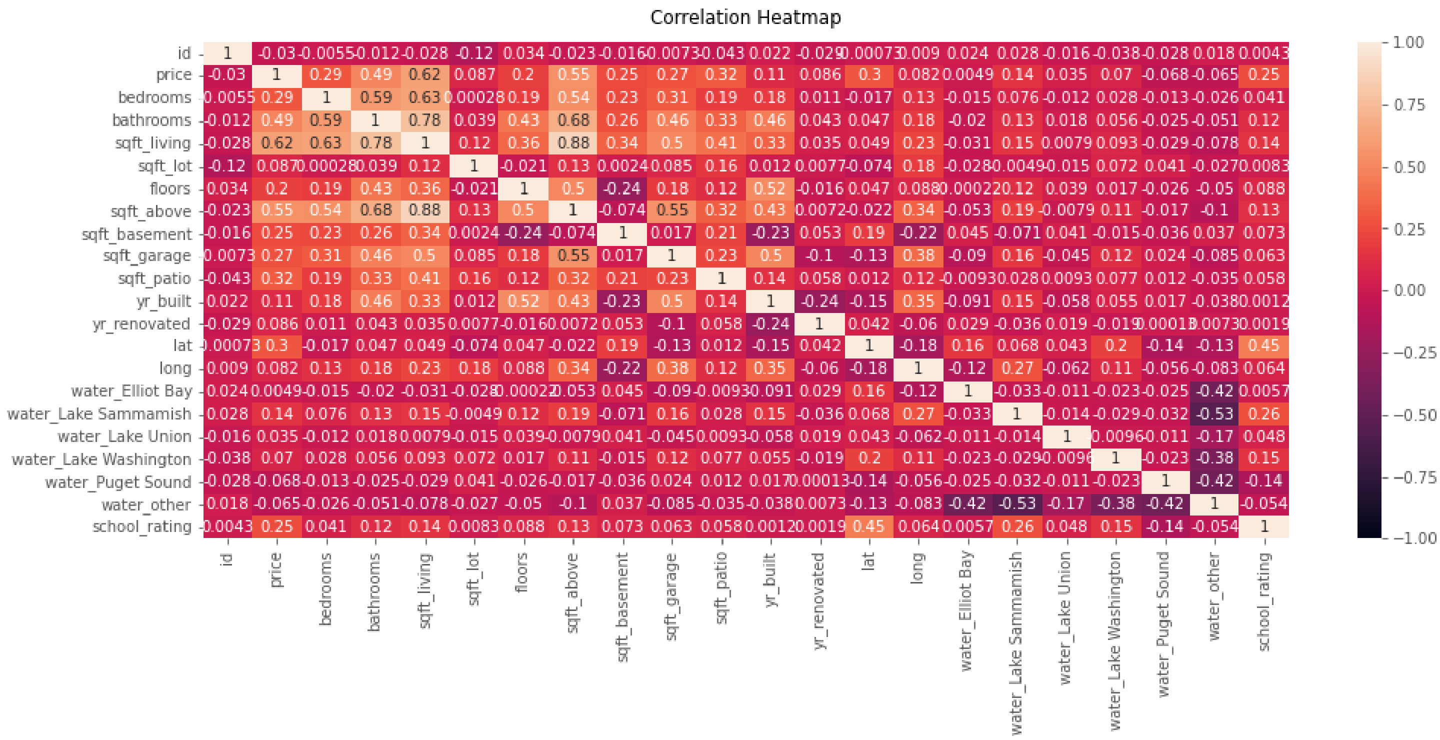
Out[33]:

```
price                    1.000000
sqft_living              0.616741
sqft_above               0.546108
bathrooms                0.488039
sqft_patio               0.317623
lat                      0.296212
bedrooms                 0.290994
sqft_garage              0.267477
school_rating            0.251165
sqft_basement            0.246548
floors                   0.199285
water_Lake Sammamish     0.141426
yr_built                 0.105877
sqft_lot                 0.086790
yr_renovated             0.085506
long                     0.081940
water_Lake Washington    0.070383
water_Puget Sound        0.068457
water_other              0.064781
water_Lake Union         0.035352
id                       0.030237
water_Elliot Bay         0.004859
Name: price, dtype: float64
```

In [34]:

```
# Increase the size of the heatmap.
plt.figure(figsize=(16, 6))
# Store heatmap object in a variable to easily access it when you want to include more features (such as title).
```

```python
# Set the range of values to be displayed on the colormap from -1 to 1, and set the annotation to True to display the
heatmap = sns.heatmap(df.corr(), vmin=-1, vmax=1, annot=True)
# Give a title to the heatmap. Pad defines the distance of the title from the top of the heatmap.
heatmap.set_title('Correlation Heatmap', fontdict={'fontsize':12}, pad=12);
```



Observations

- At first glance, it appears that sqft_living, sqft_above and bathrooms are the strongest correlated features to the price.
- Further investigation is needed to measure the validity of the variables. They may be correlated with the price due to skewness or other factors that can make the correlation a deceptively "good" feature.
- To investigate further, we will monitor the Variance Inflation Factor(VIF) to address the issue of multicollinearity.

Changing categorical variables to numerical columns - this needs to be done if we want to use them in a linear model

In [35]:

```python
#extracting grade as an integer
df['grade'] = df['grade'].apply(lambda x: int(str(x.split(' ')[0])))

# replacing conditions with values
cond_dict = {'Poor':1,'Fair':2,'Average':3,'Good':4,'Very Good':5}
df.condition.replace(to_replace=cond_dict,inplace=True)

#changing date to datetime object, get day and month
df['date'] = pd.to_datetime(df['date'])
df['month'] = df['date'].dt.month

df['day_of_year'] = df['date'].dt.dayofyear
```

Recheck dtypes

In [36]:

```python
df.dtypes
```

Out[36]:

```
id                        int64
date             datetime64[ns]
price                   float64
bedrooms                  int64
bathrooms               float64
sqft_living               int64
sqft_lot                  int64
```

```
floors                          float64
waterfront                       object
greenbelt                        object
nuisance                         object
view                             object
condition                         int64
grade                             int64
heat_source                      object
sewer_system                     object
sqft_above                        int64
sqft_basement                     int64
sqft_garage                       int64
sqft_patio                        int64
yr_built                          int64
yr_renovated                      int64
address                          object
lat                             float64
long                            float64
zipcode                          object
waterfront_loc                   object
water_Elliot Bay                  uint8
water_Lake Sammamish              uint8
water_Lake Union                  uint8
water_Lake Washington             uint8
water_Puget Sound                 uint8
water_other                       uint8
school_rating                   float64
month                             int64
day_of_year                       int64
dtype: object
```

Extracting Numerical Predictors by filtering dtypes

In [37]:
```
df.dtypes.unique()
```

Out[37]:
```
array([dtype('int64'), dtype('<M8[ns]'), dtype('float64'), dtype('O'),
       dtype('uint8')], dtype=object)
```

In [38]:
```
# categorizing dtypes
numerical_types = ['int64','float64']
numerical_predictors = list(df.select_dtypes(include=numerical_types))
numerical_predictors
```

Out[38]:
```
['id',
 'price',
 'bedrooms',
 'bathrooms',
 'sqft_living',
 'sqft_lot',
 'floors',
 'condition',
 'grade',
 'sqft_above',
 'sqft_basement',
 'sqft_garage',
 'sqft_patio',
 'yr_built',
 'yr_renovated',
 'lat',
 'long',
 'school_rating',
 'month',
 'day_of_year']
```

Create dataframe of numerical values

In [39]:
```
# df[numerical_predictors] selects only numerical columns
df_numerical = df[numerical_predictors]
```

In [40]:
```
df_numerical.columns
```
Out[40]:
```
Index(['id', 'price', 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot',
       'floors', 'condition', 'grade', 'sqft_above', 'sqft_basement',
       'sqft_garage', 'sqft_patio', 'yr_built', 'yr_renovated', 'lat', 'long',
       'school_rating', 'month', 'day_of_year'],
      dtype='object')
```
In [41]:
```
len(df_numerical)
```
Out[41]:
```
29200
```
In [42]:
```
len(waterfront_dummies)
```
Out[42]:
```
29200
```
Dropping price to isolate predictors

In [43]:
```
df_numerical = df_numerical.drop(['id','price'],axis=1)
```
In [44]:
```
df_numerical['floors'] = df['floors'].astype(float)
```
Calculating variance inflation factor [VIF]

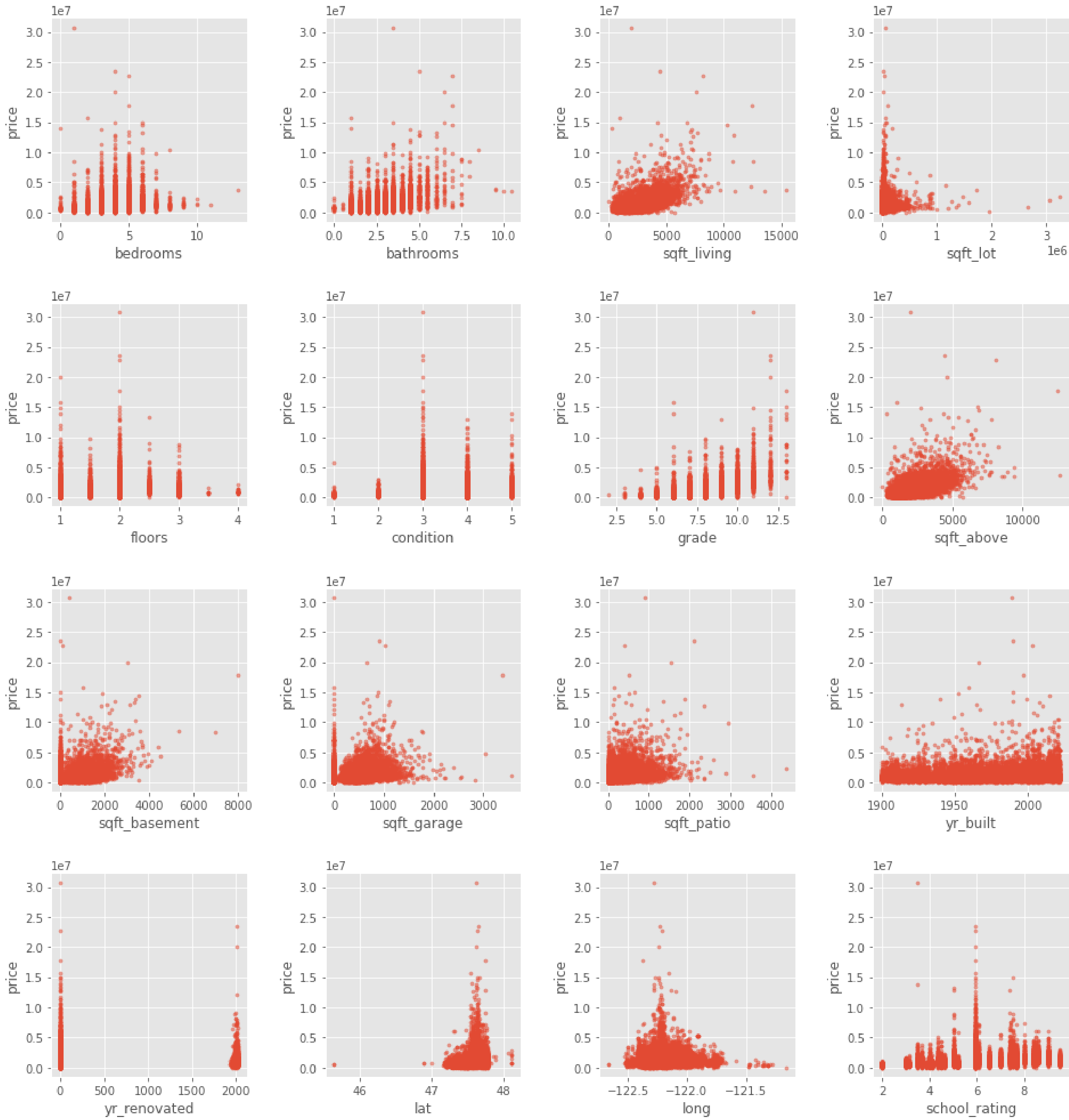
VIF levels:

- Good: VIF <= 5
- Moderate/Questionable: VIF >=5 and VIF <= 10
- Throw out: VIF >= 10

In [45]:
```
print(get_vifs(df_numerical))
```
```
         Variable            VIF
0        bedrooms      24.768622
1       bathrooms      26.263735
2     sqft_living     119.808110
3        sqft_lot       1.140594
4          floors      17.177547
5       condition      31.150197
6           grade     133.035571
7      sqft_above      92.874304
8   sqft_basement       7.075288
9     sqft_garage       4.675596
10     sqft_patio       2.240790
11       yr_built    9263.218882
12   yr_renovated       1.211647
13            lat  136585.268881
14           long  146658.438892
15  school_rating      22.635104
16          month     697.233857
17    day_of_year     612.219197
None
```
It appears at first glance that the data only yields a small set of independent variables that are not highly collinear with eachother. This will be looked at again after the removal of outliers, and the transformation of data.

In [46]:
```
# Specify the dependent variable and independent variables
y_col = 'price'
x_cols = [col for col in df_numerical.columns if col != y_col][:16]  # Use the first 15 independent variables
```

```python
# Create scatter plot matrix
fig, axs = plt.subplots(4, 4, figsize=(16, 16))
for i, x_var in enumerate(x_cols):
    row, col = divmod(i, 4)
    axs[row, col].scatter(df_numerical[x_var], df[y_col], alpha=0.5, s=10)
    axs[row, col].set_xlabel(x_var)
    axs[row, col].set_ylabel(y_col)

# Adjust plot layout
fig.subplots_adjust(top=0.93, hspace=0.4, wspace=0.4)

# Show the plot
plt.show()
```



Extracting Categorical String Predictors

In [47]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 29200 entries, 0 to 30154
Data columns (total 36 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   id                    29200 non-null  int64
 1   date                  29200 non-null  datetime64[ns]
 2   price                 29200 non-null  float64
 3   bedrooms              29200 non-null  int64
 4   bathrooms             29200 non-null  float64
 5   sqft_living           29200 non-null  int64
 6   sqft_lot              29200 non-null  int64
 7   floors                29200 non-null  float64
 8   waterfront            29200 non-null  object
 9   greenbelt             29200 non-null  object
 10  nuisance              29200 non-null  object
 11  view                  29200 non-null  object
 12  condition             29200 non-null  int64
 13  grade                 29200 non-null  int64
 14  heat_source           29200 non-null  object
 15  sewer_system          29200 non-null  object
 16  sqft_above            29200 non-null  int64
 17  sqft_basement         29200 non-null  int64
 18  sqft_garage           29200 non-null  int64
 19  sqft_patio            29200 non-null  int64
 20  yr_built              29200 non-null  int64
 21  yr_renovated          29200 non-null  int64
 22  address               29200 non-null  object
 23  lat                   29200 non-null  float64
 24  long                  29200 non-null  float64
 25  zipcode               29200 non-null  object
 26  waterfront_loc        29200 non-null  object
 27  water_Elliot Bay      29200 non-null  uint8
 28  water_Lake Sammamish  29200 non-null  uint8
 29  water_Lake Union      29200 non-null  uint8
 30  water_Lake Washington 29200 non-null  uint8
 31  water_Puget Sound     29200 non-null  uint8
 32  water_other           29200 non-null  uint8
 33  school_rating         29200 non-null  float64
 34  month                 29200 non-null  int64
 35  day_of_year           29200 non-null  int64
dtypes: datetime64[ns](1), float64(6), int64(14), object(9), uint8(6)
memory usage: 7.1+ MB
```

In [48]:
```
categorical_types = ['O']
categorical_predictors = list(df.select_dtypes(include=categorical_types))
categorical_predictors
```

Out[48]:
```
['waterfront',
 'greenbelt',
 'nuisance',
 'view',
 'heat_source',
 'sewer_system',
 'address',
 'zipcode',
 'waterfront_loc']
```

In [49]:
```
df_categorical = df[categorical_predictors]
```

In [50]:
```
df_categorical
```

Out[50]:

	waterfront	greenbelt	nuisance	view	heat_source	sewer_system	address	zipcode	waterfront_loc
0	NO	NO	NO	NONE	Gas	PUBLIC	2102 southeast 21st court, renton, washington ...	98055	other

	waterfront	greenbelt	nuisance	view	heat_source	sewer_system	address	zipcode	waterfront_loc
1	NO	NO	YES	AVERAGE	Oil	PUBLIC	11231 greenwood avenue north, seattle, washing...	98133	other
2	NO	NO	NO	AVERAGE	Gas	PUBLIC	8504 south 113th street, seattle, washington 9...	98178	other
3	NO	NO	NO	AVERAGE	Gas	PUBLIC	4079 letitia avenue south, seattle, washington...	98118	other
4	NO	NO	YES	NONE	Electricity	PUBLIC	2193 northwest talus drive, issaquah, washingt...	98027	other
...	...	...	...	...	...	...	...	...	...
30150	NO	NO	NO	NONE	Oil	PUBLIC	4673 eastern avenue north, seattle, washington...	98103	other
30151	NO	NO	NO	FAIR	Gas	PUBLIC	4131 44th avenue southwest, seattle, washingto...	98116	other
30152	NO	NO	YES	NONE	Gas	PUBLIC	910 martin luther king jr way, seattle, washin...	98122	other
30153	NO	NO	NO	NONE	Gas	PUBLIC	17127 114th avenue southeast, renton, washingt...	98055	other
30154	NO	NO	NO	NONE	Oil	PUBLIC	18615 7th avenue south, burien, washington 981...	98148	other

29200 rows × 9 columns

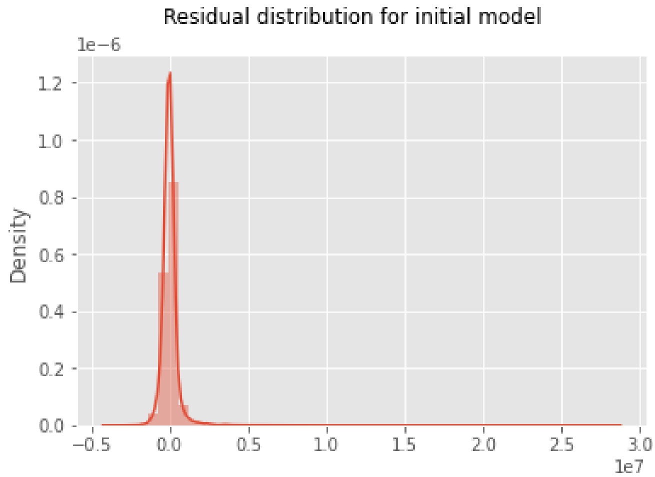
Model #1

In [51]:
```
model_data = df_numerical
```

In [52]:
```
get_OLS_model('initial',X = model_data, y = df['price'])
```

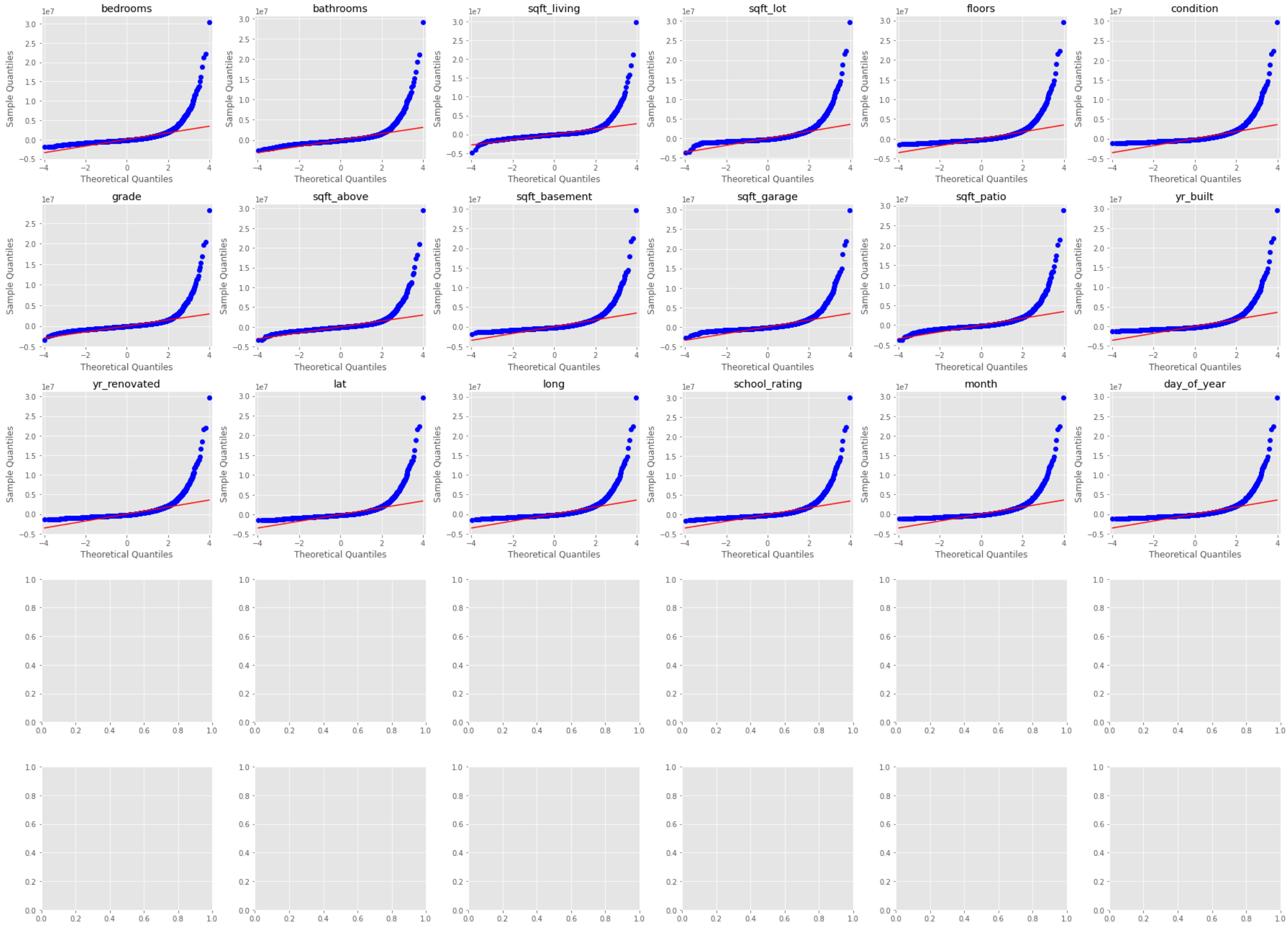
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.515
Model:                            OLS   Adj. R-squared:                  0.515
Method:                 Least Squares   F-statistic:                     1720.
Date:                Sat, 11 Mar 2023   Prob (F-statistic):               0.00
Time:                        10:24:58   Log-Likelihood:            -4.3106e+05
No. Observations:               29200   AIC:                         8.622e+05
Df Residuals:                   29181   BIC:                         8.623e+05
Df Model:                          18
Covariance Type:            nonrobust
=================================================================================
                    coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------
const          -6.06e+07      4e+06    -15.154      0.000   -6.84e+07   -5.28e+07
bedrooms      -1.126e+05   5087.667    -22.125      0.000   -1.23e+05   -1.03e+05
bathrooms      9.359e+04   7519.249     12.446      0.000    7.88e+04    1.08e+05
sqft_living     207.8333     17.053     12.187      0.000     174.408     241.258
sqft_lot          0.2621      0.062      4.197      0.000       0.140       0.385
floors         -1.49e+05   9560.059    -15.588      0.000   -1.68e+05    -1.3e+05
condition      5.268e+04   5772.321      9.126      0.000    4.14e+04     6.4e+04
grade          2.112e+05   5534.874     38.152      0.000       2e+05    2.22e+05
sqft_above      269.4581     17.407     15.480      0.000     235.340     303.576
sqft_basement    80.9114     12.880      6.282      0.000      55.666     106.157
sqft_garage    -169.2076     18.050     -9.374      0.000    -204.587    -133.829
sqft_patio      195.1326     16.668     11.707      0.000     162.463     227.802
yr_built      -2820.8412    190.261    -14.826      0.000   -3193.761   -2447.921
yr_renovated     70.5120      9.324      7.563      0.000      52.237      88.787
lat            1.243e+06   2.96e+04     41.954      0.000    1.19e+06     1.3e+06
long          -4.707e+04   3.06e+04     -1.539      0.124   -1.07e+05    1.29e+04
school_rating  2.247e+04   2840.631      7.910      0.000    1.69e+04     2.8e+04
month          1.828e+04   1.28e+04      1.430      0.153   -6782.700    4.33e+04
```

```
day_of_year   -1175.3379    419.594     -2.801      0.005   -1997.761    -352.914
==============================================================================
Omnibus:                    47069.210   Durbin-Watson:                   1.915
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         93858885.895
Skew:                          10.153   Prob(JB):                         0.00
Kurtosis:                     280.005   Cond. No.                     6.92e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.92e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```



Out[52]:
```
(None,
 Text(0.5, 0.98, 'Residual distribution for initial model'),
 <AxesSubplot:ylabel='Density'>,
 None)
```

In [53]:
```
get_model_qqplots(model_data, df['price'])
```



Observations

p_value > 0.05

- `longitude **`
- `month`
    - month was not anticipated as an effective predictor because it is not typical for the season to affect the sale price of a house

  Additional Observations:

    - The adjusted r-squared value is .514, indicating that his model can explain approximately 51.4% of the data.
    - Skew: A kurtosis value between -2 and +2 is good to prove normalcy. The skew score is 10.065, indicating that this model is heavily skewed. This will be addressed through transformations to normalize the data.

Possible Improvements to be made to model:

```
- dropping of variables that are not statistically significant (Pval > 0.05)
- addition of categorial variables(one hot encoded)
- location would possibly be the most interesting variable, mapped against the waterfront or view
variable
- transformation of data to satisfy normality assumption -ex: log transformation or square root
transformation
- removal of outliers: Outliers in this case will be considered to be any data falling greater than
  3 standard deviations outside the mean
```

Goals

```
- improve skewness - removal of outliers
- reduce homoscedacity - reduce value of VIFs
- increase rsquared to promote higher level explanation of data from model
```

Categorical data Exploratory Analysis and Engineering

The goal of this section will be to add in meaningful categorical data to the model, to be OneHotEncoded once prepped. For this, we first look at the categorical data.

In [54]:
```python
df_categorical.columns
```

Out[54]:
```
Index(['waterfront', 'greenbelt', 'nuisance', 'view', 'heat_source',
       'sewer_system', 'address', 'zipcode', 'waterfront_loc'],
      dtype='object')
```

Possible categorical variables of interest:

- `waterfront` - Whether the house is on a waterfront
  - Includes Duwamish, Elliott Bay, Puget Sound, Lake Union, Ship Canal, Lake Washington, Lake Sammamish, other lake, and river/slough waterfronts
- `greenbelt` - Whether the house is adjacent to a green belt
- `nuisance` - Whether the house has traffic noise or other recorded nuisances
- `view` - Quality of view from house
  - Includes views of Mt. Rainier, Olympics, Cascades, Territorial, Seattle Skyline, Puget Sound, Lake Washington, Lake Sammamish, small lake / river / creek, and other
- `heat_source` - Heat source for the house
- `sewer_system` - Sewer system for the house
- `address` - The street address

The grade and condition are already onehotencoded in the model and could be changed to a numerical variable, so this part of the analysis will focus on the string categorical variables.

The address appears to be the most interesting variable in the batch because it can be mapped against the waterfronts or the quality of view from the houses. For this, we will extrapolate features of the address to reduce and categorize the location.

In [55]:
```python
df['waterfront'].unique()
```

Out[55]:
```
array(['NO', 'YES'], dtype=object)
```

In [56]:
```python
# convert waterfront into numeric boolean
waterfront_bool_dict = {'YES':1,'NO':0,np.nan:0}
df_categorical.waterfront.replace(to_replace=waterfront_bool_dict,inplace=True)
```

In [57]:
```python
plt.scatter(x=df['waterfront'], y=df['price'])
```

Out[57]:
```
<matplotlib.collections.PathCollection at 0x25ab9a4d160>
```

In [58]:
```
df['nuisance'].unique()
```

Out[58]:
```
array(['NO', 'YES'], dtype=object)
```

In [59]:
```
# convert nuisance into numeric boolean
nuisance_bool_dict = {'YES':1,'NO':0,np.nan:0}
df_categorical.nuisance.replace(to_replace=nuisance_bool_dict,inplace=True)
```

In [60]:
```
plt.scatter(x=df['nuisance'], y=df['price'])
```

Out[60]:
```
<matplotlib.collections.PathCollection at 0x25ab505b610>
```

In [61]:
```
# convert nuisance into numeric boolean
greenbelt_bool_dict = {'YES':1,'NO':0,np.nan:0}
df_categorical.greenbelt.replace(to_replace=greenbelt_bool_dict,inplace=True)
```

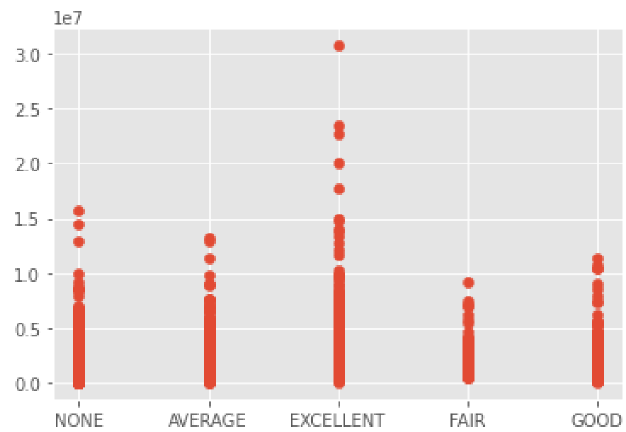
In [62]:
```
df['view'].unique()
```

Out[62]:
```
array(['NONE', 'AVERAGE', 'EXCELLENT', 'FAIR', 'GOOD'], dtype=object)
```

In [63]:
```
# convert view from string into categorical ordinal
view_dict = {'NONE':0,'FAIR':1,'AVERAGE':2,'GOOD':3,'EXCELLENT':4}
df_categorical.view.replace(to_replace=view_dict,inplace=True)
```

In [64]:
```
plt.scatter(x=df['view'], y=df['price'])
```

Out[64]:
```
<matplotlib.collections.PathCollection at 0x25ab240dca0>
```



In [65]:

```
df['heat_source'].unique()
```

Out[65]:

```
array(['Gas', 'Oil', 'Electricity', 'Gas/Solar', 'Electricity/Solar',
       'Other', 'Oil/Solar'], dtype=object)
```

In [66]:

```
heat_source_dummies = pd.get_dummies(df['heat_source'], prefix='heat_source',drop_first=True)
heat_source_dummies
```

Out[66]:

	heat_source_Electricity/Solar	heat_source_Gas	heat_source_Gas/Solar	heat_source_Oil	heat_source_Oil/Solar	heat_source_Other
0	0	1	0	0	0	0
1	0	0	0	1	0	0
2	0	1	0	0	0	0
3	0	1	0	0	0	0
4	0	0	0	0	0	0
...	...	...	...	...	...	...
30150	0	0	0	1	0	0
30151	0	1	0	0	0	0
30152	0	1	0	0	0	0
30153	0	1	0	0	0	0
30154	0	0	0	1	0	0

29200 rows × 6 columns

In [67]:

```
df['sewer_system'].unique()
```

Out[67]:

```
array(['PUBLIC', 'PRIVATE', 'PRIVATE RESTRICTED', 'PUBLIC RESTRICTED'],
      dtype=object)
```

In [68]:

```
sewer_dummies = pd.get_dummies(df['sewer_system'],prefix='sewer', drop_first=True)
sewer_dummies
```

Out[68]:

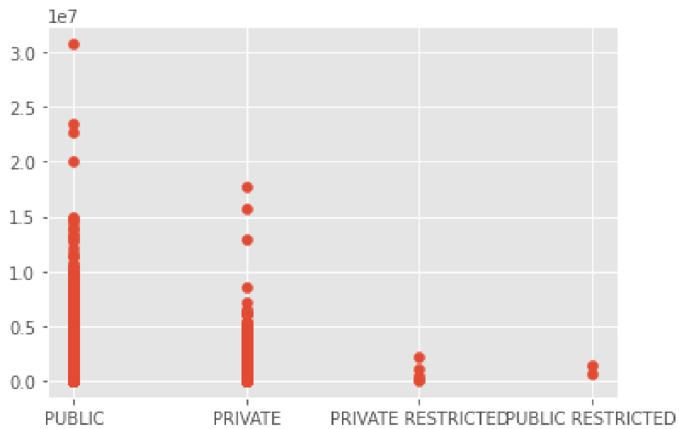
	sewer_PRIVATE RESTRICTED	sewer_PUBLIC	sewer_PUBLIC RESTRICTED
0	0	1	0
1	0	1	0
2	0	1	0
3	0	1	0
4	0	1	0
...	...	...	...

	sewer_PRIVATE RESTRICTED	sewer_PUBLIC	sewer_PUBLIC RESTRICTED
30150	0	1	0
30151	0	1	0
30152	0	1	0
30153	0	1	0
30154	0	1	0

29200 rows × 3 columns

In [69]:
```
plt.scatter(x=df['sewer_system'], y=df['price'])
```

Out[69]:
```
<matplotlib.collections.PathCollection at 0x25ab90c2f70>
```



Developing categorical dataframe

In [70]:
```
df_cat_pick = df_categorical[['waterfront','nuisance','view','greenbelt']]
```

Model #2

In [71]:
```
model_2_data = pd.concat([df_numerical,sewer_dummies,heat_source_dummies, df_cat_pick], axis = 1)
```

In [72]:
```
len(model_2_data) == len(waterfront_dummies)
```

Out[72]:
```
True
```

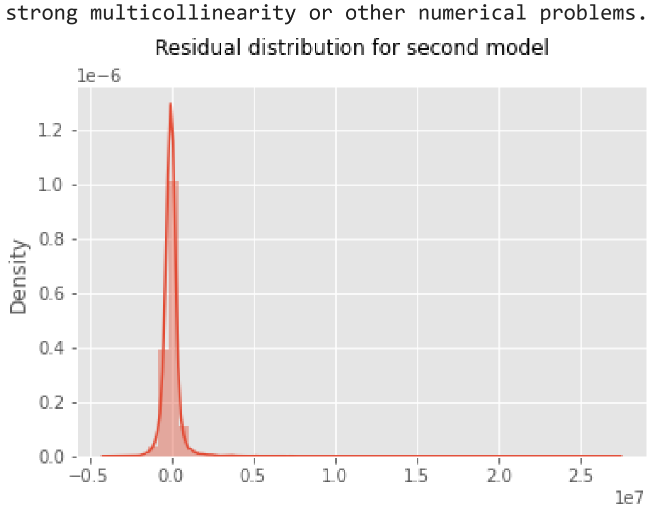
In [73]:
```
model_2_data.columns
```

Out[73]:
```
Index(['bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors',
       'condition', 'grade', 'sqft_above', 'sqft_basement', 'sqft_garage',
       'sqft_patio', 'yr_built', 'yr_renovated', 'lat', 'long',
       'school_rating', 'month', 'day_of_year', 'sewer_PRIVATE RESTRICTED',
       'sewer_PUBLIC', 'sewer_PUBLIC RESTRICTED',
       'heat_source_Electricity/Solar', 'heat_source_Gas',
       'heat_source_Gas/Solar', 'heat_source_Oil', 'heat_source_Oil/Solar',
       'heat_source_Other', 'waterfront', 'nuisance', 'view', 'greenbelt'],
      dtype='object')
```

In [74]:
```
get_OLS_model('second',model_2_data, df['price'])
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.556
Model:                            OLS   Adj. R-squared:                  0.555
Method:                 Least Squares   F-statistic:                     1177.
```

```
Date:                Sat, 11 Mar 2023   Prob (F-statistic):            0.00
Time:                        10:25:20   Log-Likelihood:         -4.2978e+05
No. Observations:               29200   AIC:                      8.596e+05
Df Residuals:                   29168   BIC:                      8.599e+05
Df Model:                          31
Covariance Type:            nonrobust
=================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------------------
const                        -5.484e+07   4.01e+06    -13.663      0.000   -6.27e+07    -4.7e+07
bedrooms                     -8.686e+04   4930.736    -17.617      0.000   -9.65e+04   -7.72e+04
bathrooms                      7.97e+04   7257.005     10.982      0.000    6.55e+04    9.39e+04
sqft_living                    161.2794     16.409      9.829      0.000     129.117     193.442
sqft_lot                         0.3719      0.063      5.925      0.000       0.249       0.495
floors                       -1.622e+05   9216.302    -17.600      0.000    -1.8e+05   -1.44e+05
condition                     5.753e+04   5591.576     10.289      0.000    4.66e+04    6.85e+04
grade                         1.943e+05   5359.515     36.262      0.000    1.84e+05    2.05e+05
sqft_above                     294.6078     16.761     17.577      0.000     261.755     327.461
sqft_basement                   70.5356     12.449      5.666      0.000      46.135      94.936
sqft_garage                   -107.0079     17.486     -6.120      0.000    -141.280     -72.735
sqft_patio                     131.4469     16.314      8.058      0.000      99.472     163.422
yr_built                     -2348.8234    185.775    -12.643      0.000   -2712.951   -1984.695
yr_renovated                    38.6745      8.992      4.301      0.000      21.050      56.299
lat                           1.313e+06   2.88e+04     45.638      0.000    1.26e+06    1.37e+06
long                          3.511e+04   3.05e+04      1.151      0.250   -2.47e+04    9.49e+04
school_rating                 2.287e+04   2722.759      8.398      0.000    1.75e+04    2.82e+04
month                         2.066e+04   1.22e+04      1.688      0.091   -3333.789    4.47e+04
day_of_year                  -1268.1044    401.778     -3.156      0.002   -2055.607    -480.601
sewer_PRIVATE RESTRICTED     -6.283e+04   2.68e+05     -0.235      0.814   -5.88e+05    4.62e+05
sewer_PUBLIC                  1.739e+05   1.16e+04     14.957      0.000    1.51e+05    1.97e+05
sewer_PUBLIC RESTRICTED      -3.257e+04   4.23e+05     -0.077      0.939   -8.61e+05    7.96e+05
heat_source_Electricity/Solar -3.714e+04  7.96e+04     -0.467      0.641   -1.93e+05    1.19e+05
heat_source_Gas              -3154.8068   9501.233     -0.332      0.740   -2.18e+04    1.55e+04
heat_source_Gas/Solar         1.181e+05   6.26e+04      1.887      0.059   -4594.164    2.41e+05
heat_source_Oil               -4.11e+04   1.45e+04     -2.828      0.005   -6.96e+04   -1.26e+04
heat_source_Oil/Solar        -1.393e+05   2.99e+05     -0.466      0.641   -7.25e+05    4.46e+05
heat_source_Other            -1.869e+04   1.34e+05     -0.139      0.889   -2.82e+05    2.44e+05
waterfront                    1.064e+06   2.99e+04     35.528      0.000       1e+06    1.12e+06
nuisance                       1.33e+04   9507.695      1.398      0.162   -5340.202    3.19e+04
view                          8.881e+04   4849.452     18.314      0.000    7.93e+04    9.83e+04
greenbelt                     4667.2139   2.23e+04      0.209      0.834   -3.91e+04    4.84e+04
==============================================================================
Omnibus:                    45743.713   Durbin-Watson:                   1.904
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         88257884.083
Skew:                           9.547   Prob(JB):                         0.00
Kurtosis:                     271.656   Cond. No.                     7.26e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7.26e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```




```
Out[74]: (None,
 Text(0.5, 0.98, 'Residual distribution for second model'),
```

```
 <AxesSubplot:ylabel='Density'>,
 None)
```

`heat_source`, `greenbelt` and `sewer_system` both have incredibly high p-values. These will be dropped from the final model if it holds.

Observations of Model 2

Model is still highly skewed although did present itself with some improvements. Next steps will be to normalize the data by transforming features that are skewed within the data, as well as remove outliers

- Jarque-Bera score is sky high and must come down for the model to hold any validity.
- Durbin Watson score is in the acceptable range of 1.50-2.50
- Rsquared has 'improved' but only at the expense of the the continued flaws mentioned before.

Eliminating Outliers

To normalize the distribution, outlier removal will be the first step. An outlier will be defined as three standard deviations away from the mean of the target variable.

In [75]:

```
outlier_thresh = df['price'].std()*3 # value of the prices at the third standard deviation
df_outlier_removed = df.loc[abs(df['price']) <= outlier_thresh] # slicing all data within the defined range

# assign y as the target variable
y = df_outlier_removed['price']
```

In [76]:

```
model_2_data_outlier_removed = model_2_data.loc[abs(df['price']) <= outlier_thresh]
```

In [77]:

```
df_outlier_removed
```

Out[77]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	greenbelt	...	waterfront_loc	water_El
0	7399300360	2022-05-24	675000.0	4	1.0	1180	7140	1.0	NO	NO	...	other	
1	8910500230	2021-12-13	920000.0	5	2.5	2770	6703	1.0	NO	NO	...	other	
2	1180000275	2021-09-29	311000.0	6	2.0	2880	6156	1.0	NO	NO	...	other	
3	1604601802	2021-12-14	775000.0	3	3.0	2160	1400	2.0	NO	NO	...	other	
4	8562780790	2021-08-24	592500.0	2	2.0	1120	758	2.0	NO	NO	...	other	
...	...	...	...	...	...	...	...	...	...	...	...	...	
30150	7834800180	2021-11-30	1555000.0	5	2.0	1910	4000	1.5	NO	NO	...	other	
30151	194000695	2021-06-16	1313000.0	3	2.0	2020	5800	2.0	NO	NO	...	other	
30152	7960100080	2022-05-27	800000.0	3	2.0	1620	3600	1.0	NO	NO	...	other	
30153	2781280080	2022-02-24	775000.0	3	2.5	2570	2889	2.0	NO	NO	...	other	
30154	9557800100	2022-04-29	500000.0	3	1.5	1200	11058	1.0	NO	NO	...	other	

28004 rows × 36 columns

In [78]:

```
waterfront_dummies = df_outlier_removed[['water_Elliot Bay','water_Lake Sammamish', 'water_Lake Washington','water_Pug
```

In [79]:

```
df_outlier_removed.columns
```

Out[79]:

```
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', 'zipcode', 'waterfront_loc',
       'water_Elliot Bay', 'water_Lake Sammamish', 'water_Lake Union',
       'water_Lake Washington', 'water_Puget Sound', 'water_other',
       'school_rating', 'month', 'day_of_year'],
      dtype='object')
```

New look at model with removed outliers

In [80]:

```
outlier_data = pd.concat([y,model_2_data_outlier_removed], axis=1)
```

In [81]:

```
outlier_data = outlier_data.drop('price', axis=1)
```

In [82]:

```
len(outlier_data)
```

Out[82]:

```
28004
```

In [83]:

```
outlier_data
```

Out[83]:

	bedrooms	bathrooms	sqft_living	sqft_lot	floors	condition	grade	sqft_above	sqft_basement	sqft_garage	...	heat_source_Electric
0	4	1.0	1180	7140	1.0	4	7	1180	0	0	...	
1	5	2.5	2770	6703	1.0	3	7	1570	1570	0	...	
2	6	2.0	2880	6156	1.0	3	7	1580	1580	0	...	
3	3	3.0	2160	1400	2.0	3	9	1090	1070	200	...	
4	2	2.0	1120	758	2.0	3	7	1120	550	550	...	
...	...	...	...	...	...	...	...	...	...	...	...	
30150	5	2.0	1910	4000	1.5	4	8	1600	1130	0	...	
30151	3	2.0	2020	5800	2.0	3	7	2020	0	0	...	
30152	3	2.0	1620	3600	1.0	3	7	940	920	240	...	
30153	3	2.5	2570	2889	2.0	3	8	1830	740	480	...	
30154	3	1.5	1200	11058	1.0	3	7	1200	0	420	...	

28004 rows × 31 columns

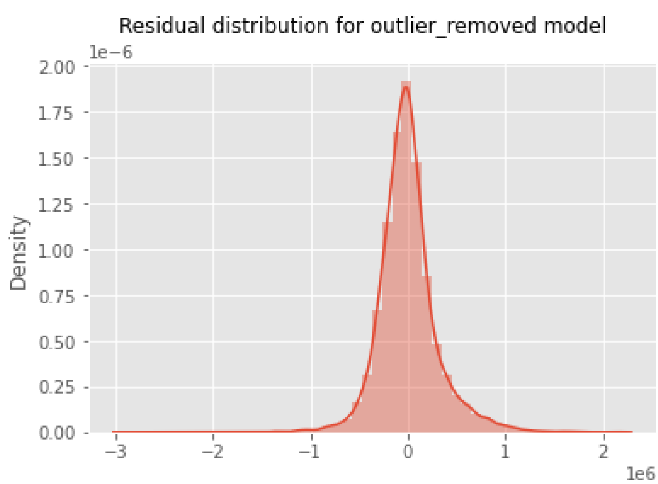
Model #3

In [84]:

```
get_OLS_model('outlier_removed', outlier_data,y)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.633
Model:                            OLS   Adj. R-squared:                  0.633
Method:                 Least Squares   F-statistic:                     1559.
Date:                Sat, 11 Mar 2023   Prob (F-statistic):               0.00
Time:                        10:25:37   Log-Likelihood:            -3.9305e+05
```

```
No. Observations:                28004   AIC:                         7.862e+05
Df Residuals:                    27972   BIC:                         7.864e+05
Df Model:                           31
Covariance Type:             nonrobust
===================================================================================================
                                      coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------------
const                           -2.636e+07   2.07e+06    -12.734      0.000   -3.04e+07   -2.23e+07
bedrooms                        -9759.7401   2609.082     -3.741      0.000   -1.49e+04   -4645.812
bathrooms                        3.304e+04   3848.208      8.586      0.000    2.55e+04    4.06e+04
sqft_living                       138.2445      8.838     15.643      0.000     120.922     155.567
sqft_lot                            0.3592      0.036     10.002      0.000       0.289       0.430
floors                          -2.954e+04   4853.999     -6.087      0.000   -3.91e+04      -2e+04
condition                        5.835e+04   2878.028     20.275      0.000    5.27e+04     6.4e+04
grade                            1.395e+05   2855.796     48.855      0.000    1.34e+05    1.45e+05
sqft_above                         95.3023      9.083     10.492      0.000      77.499     113.106
sqft_basement                       8.4125      6.627      1.269      0.204      -4.577      21.402
sqft_garage                       -22.1680      9.206     -2.408      0.016     -40.213      -4.123
sqft_patio                         56.4334      8.752      6.448      0.000      39.278      73.588
yr_built                        -1966.2467     96.948    -20.281      0.000   -2156.270   -1776.223
yr_renovated                       32.6548      4.756      6.866      0.000      23.332      41.977
lat                              1.121e+06   1.47e+04     76.353      0.000    1.09e+06    1.15e+06
long                             1.966e+05   1.57e+04     12.526      0.000    1.66e+05    2.27e+05
school_rating                    4.123e+04   1399.288     29.468      0.000    3.85e+04     4.4e+04
month                            1.741e+04   6310.185      2.759      0.006    5039.033    2.98e+04
day_of_year                     -1051.5291    207.122     -5.077      0.000   -1457.497    -645.561
sewer_PRIVATE RESTRICTED         1.571e+05   1.35e+05      1.161      0.246   -1.08e+05    4.22e+05
sewer_PUBLIC                     6.024e+04   6023.531     10.001      0.000    4.84e+04     7.2e+04
sewer_PUBLIC RESTRICTED          3.156e+04   2.13e+05      0.148      0.882   -3.87e+05     4.5e+05
heat_source_Electricity/Solar   -2.933e+04   4.06e+04     -0.723      0.470   -1.09e+05    5.02e+04
heat_source_Gas                  2.871e+04   4874.881      5.888      0.000    1.92e+04    3.83e+04
heat_source_Gas/Solar            1.578e+05   3.32e+04      4.746      0.000    9.26e+04    2.23e+05
heat_source_Oil                 -1.927e+04   7423.084     -2.596      0.009   -3.38e+04   -4719.812
heat_source_Oil/Solar           -1.627e+04   1.51e+05     -0.108      0.914   -3.12e+05     2.8e+05
heat_source_Other                8.418e+04   6.95e+04      1.210      0.226   -5.21e+04     2.2e+05
waterfront                       1.201e+05   1.79e+04      6.706      0.000     8.5e+04    1.55e+05
nuisance                        -2.754e+04   4931.454     -5.584      0.000   -3.72e+04   -1.79e+04
view                             6.125e+04   2614.263     23.429      0.000    5.61e+04    6.64e+04
greenbelt                        9.391e+04   1.17e+04      8.023      0.000     7.1e+04    1.17e+05
==============================================================================
Omnibus:                     4005.357   Durbin-Watson:                   1.997
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            20991.018
Skew:                           0.590   Prob(JB):                         0.00
Kurtosis:                       7.074   Cond. No.                     6.61e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.61e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```




```
Out[84]: (None,
          Text(0.5, 0.98, 'Residual distribution for outlier_removed model'),
          <AxesSubplot:ylabel='Density'>,
          None)
```

In [85]:
```
outlier_data.columns
```

Out[85]:
```
Index(['bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors',
       'condition', 'grade', 'sqft_above', 'sqft_basement', 'sqft_garage',
       'sqft_patio', 'yr_built', 'yr_renovated', 'lat', 'long',
       'school_rating', 'month', 'day_of_year', 'sewer_PRIVATE RESTRICTED',
       'sewer_PUBLIC', 'sewer_PUBLIC RESTRICTED',
       'heat_source_Electricity/Solar', 'heat_source_Gas',
       'heat_source_Gas/Solar', 'heat_source_Oil', 'heat_source_Oil/Solar',
       'heat_source_Other', 'waterfront', 'nuisance', 'view', 'greenbelt'],
      dtype='object')
```

Observations of model 3

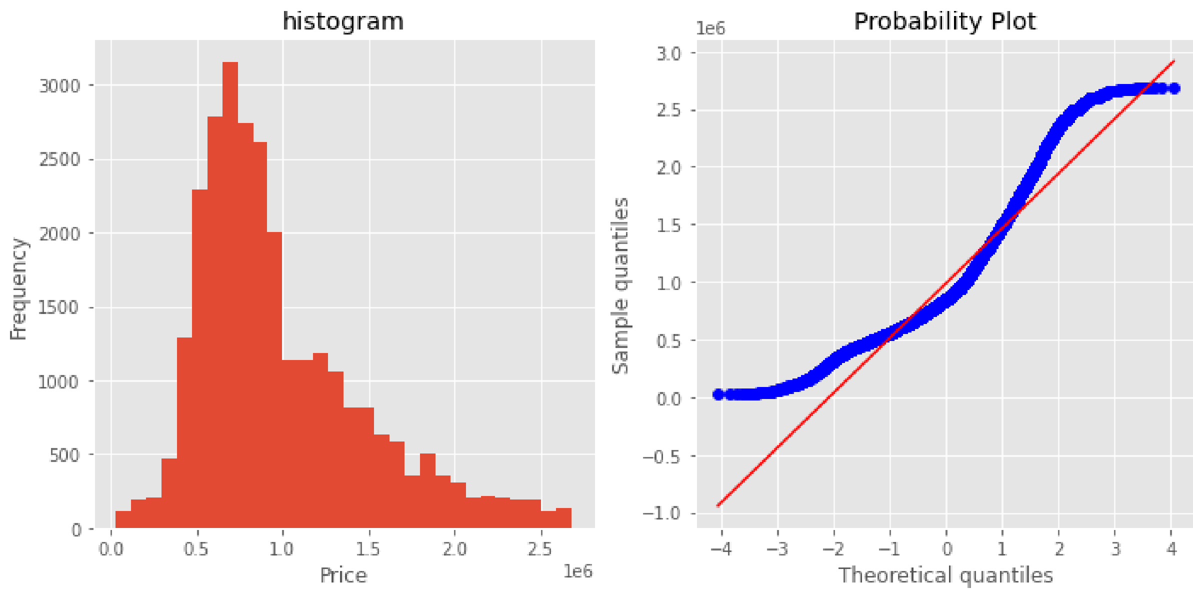
pvalue > 0.05

- `sqft_basement`
- `sqft_garage`
- `sewer_PRIVATE RESTRICTED`
- `sewer_PUBLIC RESTRICTED`
- `heat_source_Electricity/Solar`
- `heat_source_Oil/Solar`
- `heat_source_Other`

- Adjusted rsquared indicates that the model explains 62.2% of the data.
- Skewness has improved dramatically to an acceptable range between -2 and 2. The removal of outliers has made this possible.
- Durbin-Watson score is still in the acceptable ranges of 1.5-2.5
- Jarque-Bera score is still very high but has been brought down by a significant factor. Still not perfect but trending in the right direction.
- Multicollinearity is possibly present in the model and likely so given the initial VIFs before the first model was built. VIFS should be revisited again to see if those variables are worth keeping.

Looking at transformations for the price.

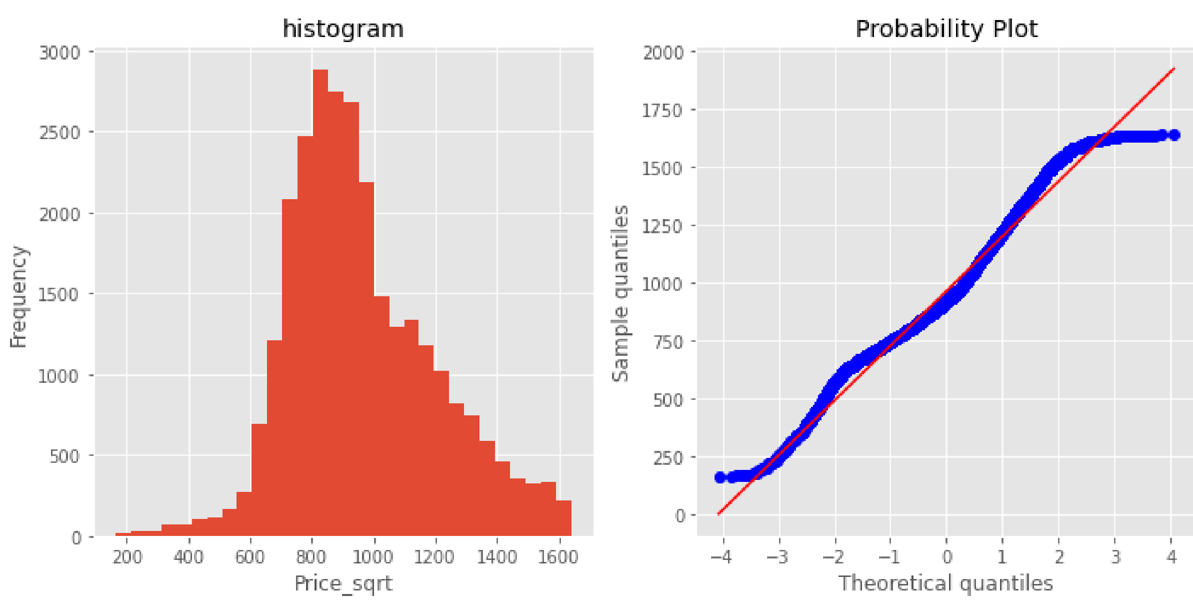
In [86]:
```
import scipy.stats as stats
fig, axs = plt.subplots(1, 2, figsize=(10, 5))

# Plot histogram on the first subplot
axs[0].hist(y, bins=30)
axs[0].set_xlabel('Price')
axs[0].set_ylabel('Frequency')
axs[0].set_title('histogram')
# Plot QQ-plot on the second subplot
stats.probplot(y, plot=axs[1])
axs[1].set_xlabel('Theoretical quantiles')
axs[1].set_ylabel('Sample quantiles')

# Adjust the layout and display the plot
plt.tight_layout()
plt.show()
```



Issue above is the data shows linearization everywhere but both tails of the data. Catching the lower tail will be the goal for the next test of transformation. For this, we will try a root transformation.

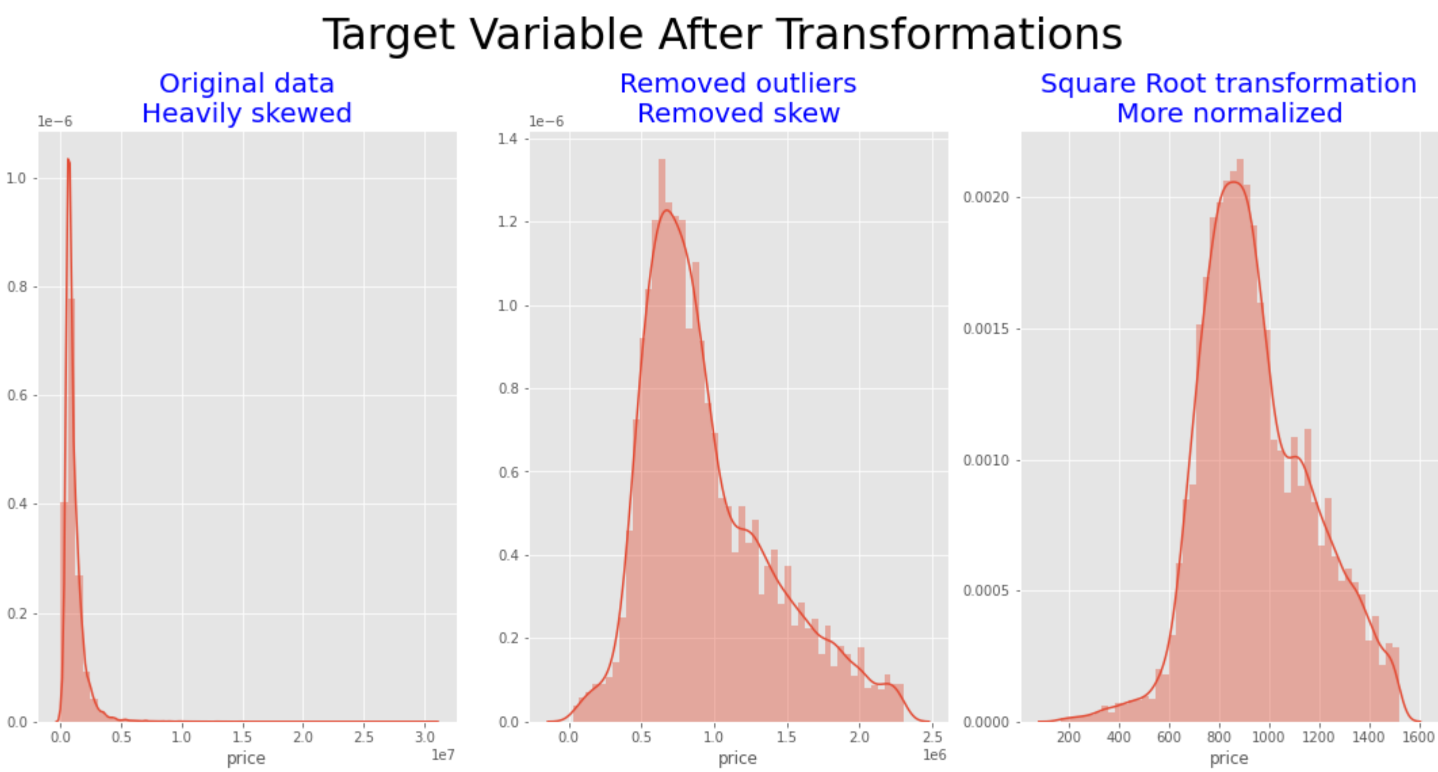
In [87]:

```
import matplotlib.pyplot as plt
import scipy.stats as stats

# Create subplots with 1 row and 2 columns
fig, axs = plt.subplots(1, 2, figsize=(10, 5))
y_sqrt = y**0.5
# Plot histogram on the first subplot
axs[0].hist(y_sqrt, bins=30)
axs[0].set_xlabel('Price_sqrt')
axs[0].set_ylabel('Frequency')
axs[0].set_title('histogram')

# Plot QQ-plot on the second subplot
stats.probplot(y_sqrt, plot=axs[1])
axs[1].set_xlabel('Theoretical quantiles')
axs[1].set_ylabel('Sample quantiles')

# Adjust the layout and display the plot
plt.tight_layout()
plt.show()
```



In [88]:

```
def plot_dist(ax, data, title):
    sns.distplot(data, ax=ax)
    ax.set_title(title, fontsize=20, color='b')
    ax.set_ylabel("")

# Create subplots
fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(15, 8))

# Plot the original data
plot_dist(ax1, df['price'], "Original data\nHeavily skewed")

# Remove outliers from the data
q1 = df['price'].quantile(0.25)
q3 = df['price'].quantile(0.75)
iqr = q3 - q1
lower_bound = q1 - 1.5*iqr
upper_bound = q3 + 1.5*iqr
df_outliers_removed = df[(df['price'] > lower_bound) & (df['price'] < upper_bound)]
y = df_outliers_removed['price']

# Plot the data with outliers removed
plot_dist(ax2, y, "Removed outliers\nRemoved skew")

# Apply square root transformation to the data
y_sqrt = np.sqrt(y)

# Plot the transformed data
plot_dist(ax3, y_sqrt, "Square Root transformation\nMore normalized")

# Set the overall title of the figure
fig.suptitle("Target Variable After Transformations", fontsize=32)

# Adjust the layout of the subplots
fig.tight_layout()

# Show the figure
plt.show()
```

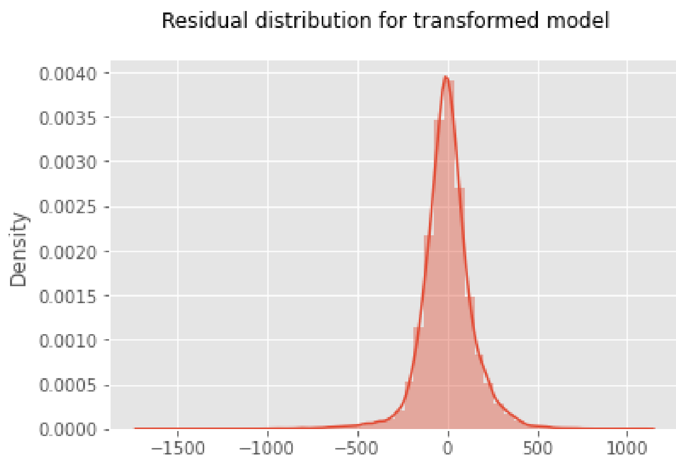


Checking model with transformed target variable - square root transformation

In [146…

```
get_OLS_model('transformed', outlier_data, y_sqrt)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.640
Model:                            OLS   Adj. R-squared:                  0.640
Method:                 Least Squares   F-statistic:                     1605.
Date:                Fri, 10 Mar 2023   Prob (F-statistic):               0.00
Time:                        17:27:23   Log-Likelihood:            -1.7892e+05
No. Observations:               28004   AIC:                         3.579e+05
Df Residuals:                   27972   BIC:                         3.582e+05
Df Model:                          31
Covariance Type:            nonrobust
===================================================================================================
                                      coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------------
const                           -1.321e+04    989.306    -13.348      0.000   -1.51e+04   -1.13e+04
bedrooms                           -1.6560      1.247     -1.328      0.184      -4.100       0.788
bathrooms                          18.9610      1.839     10.311      0.000      15.357      22.565
sqft_living                         0.0574      0.004     13.585      0.000       0.049       0.066
sqft_lot                            0.0002   1.72e-05     10.647      0.000       0.000       0.000
floors                             -7.4553      2.320     -3.214      0.001     -12.002      -2.909
condition                          30.6156      1.375     22.261      0.000      27.920      33.311
grade                              65.7100      1.365     48.150      0.000      63.035      68.385
sqft_above                          0.0445      0.004     10.243      0.000       0.036       0.053
sqft_basement                       0.0092      0.003      2.893      0.004       0.003       0.015
sqft_garage                        -0.0089      0.004     -2.013      0.044      -0.017      -0.000
sqft_patio                          0.0287      0.004      6.868      0.000       0.021       0.037
yr_built                           -0.8548      0.046    -18.450      0.000      -0.946      -0.764
yr_renovated                        0.0154      0.002      6.768      0.000       0.011       0.020
lat                               575.8772      7.015     82.087      0.000     562.127     589.628
long                              102.1319      7.499     13.619      0.000      87.433     116.830
school_rating                      20.9074      0.669     31.267      0.000      19.597      22.218
month                               9.2201      3.015      3.058      0.002       3.310      15.130
day_of_year                        -0.5355      0.099     -5.410      0.000      -0.729      -0.341
sewer_PRIVATE RESTRICTED          -21.5905     64.646     -0.334      0.738    -148.299     105.118
sewer_PUBLIC                       28.1201      2.878      9.769      0.000      22.478      33.762
sewer_PUBLIC RESTRICTED            30.1586    101.994      0.296      0.767    -169.755     230.073
heat_source_Electricity/Solar     -35.1174     19.383     -1.812      0.070     -73.109       2.874
heat_source_Gas                    17.0425      2.330      7.316      0.000      12.477      21.609
heat_source_Gas/Solar              64.6741     15.885      4.071      0.000      33.538      95.810
heat_source_Oil                    -3.3639      3.547     -0.948      0.343     -10.317       3.589
heat_source_Oil/Solar              10.5192     72.126      0.146      0.884    -130.851     151.890
heat_source_Other                  33.1838     33.233      0.999      0.318     -31.955      98.323
waterfront                         61.4871      8.556      7.186      0.000      44.717      78.257
nuisance                          -15.6720      2.357     -6.650      0.000     -20.291     -11.053
view                               27.9663      1.249     22.386      0.000      25.518      30.415
greenbelt                          43.5181      5.593      7.781      0.000      32.556      54.481
==============================================================================
Omnibus:                     3994.455   Durbin-Watson:                   2.002
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            38628.578
Skew:                          -0.371   Prob(JB):                         0.00
Kurtosis:                       8.706   Cond. No.                     6.61e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.61e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```



Out[146…

```
(None,
 Text(0.5, 0.98, 'Residual distribution for transformed model'),
 <AxesSubplot:ylabel='Density'>,
 None)
```

y_log vs y_sqrt

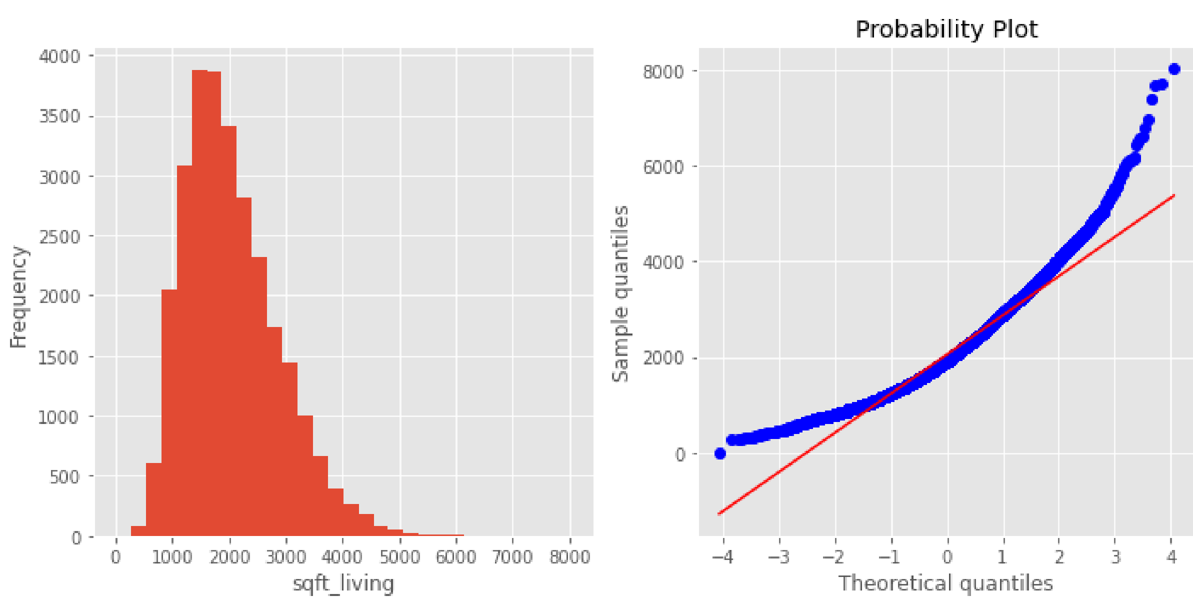
The model with the square root transformation appears to be less skewed and possesses a higher rsquared value, lending the ability of the model to explain more of the data. For these reasons we will use y_sqrt as our dependent variable for now until y_log appears to outweight the benefit of y_sqrt.

Jarque-Beras score is significantly better as well with the y_sqrt variable so I'll go with it for now.

Checking distribution of predictor

In [147…

```
plot_hist_qq(outlier_data, 'sqft_living')
```



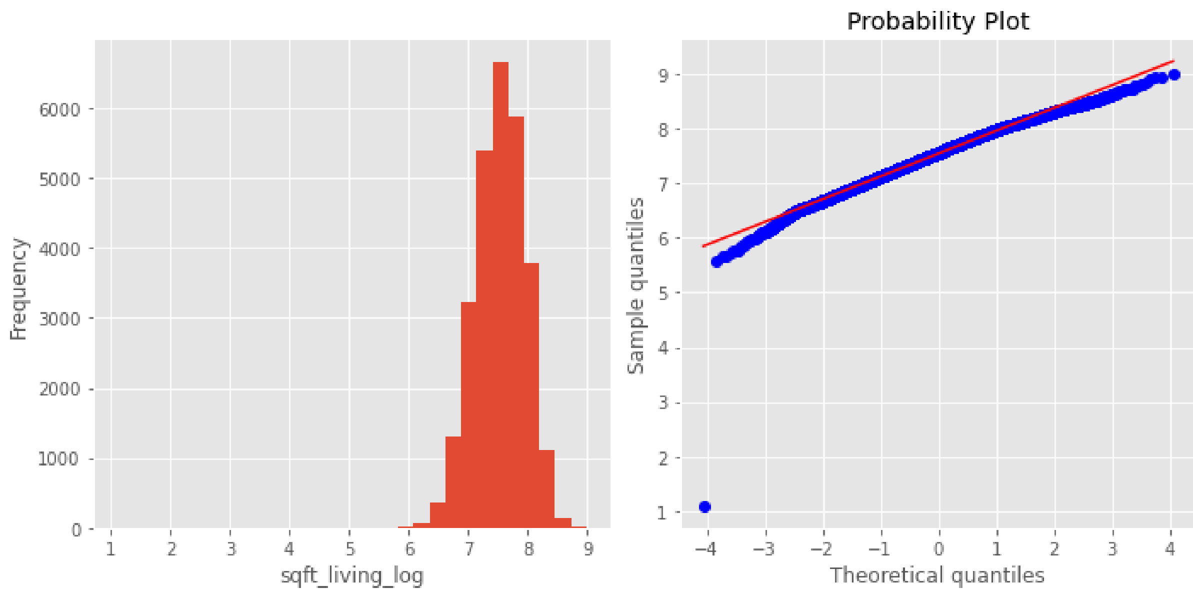
Data is clearly skewed right and follows an exponential pattern similar to price. For this, we will use a logarithmic transformation.

In [148…

```
outlier_data['sqft_living_log'] = np.log(outlier_data['sqft_living'])
```

In [149…

```
plot_hist_qq(outlier_data, 'sqft_living_log')
```



In [150…

```
outlier_data = outlier_data.drop('sqft_living', axis=1)
```

In [151…

```
outlier_data
```

Out[151…

	bedrooms	bathrooms	sqft_lot	floors	condition	grade	sqft_above	sqft_basement	sqft_garage	sqft_patio	...	heat_source_Gas	he
0	4	1.0	7140	1.0	4	7	1180	0	0	40	...	1	
1	5	2.5	6703	1.0	3	7	1570	1570	0	240	...	0	
2	6	2.0	6156	1.0	3	7	1580	1580	0	0	...	1	
3	3	3.0	1400	2.0	3	9	1090	1070	200	270	...	1	
4	2	2.0	758	2.0	3	7	1120	550	550	30	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
30150	5	2.0	4000	1.5	4	8	1600	1130	0	210	...	0	
30151	3	2.0	5800	2.0	3	7	2020	0	0	520	...	1	
30152	3	2.0	3600	1.0	3	7	940	920	240	110	...	1	
30153	3	2.5	2889	2.0	3	8	1830	740	480	100	...	1	
30154	3	1.5	11058	1.0	3	7	1200	0	420	0	...	0	

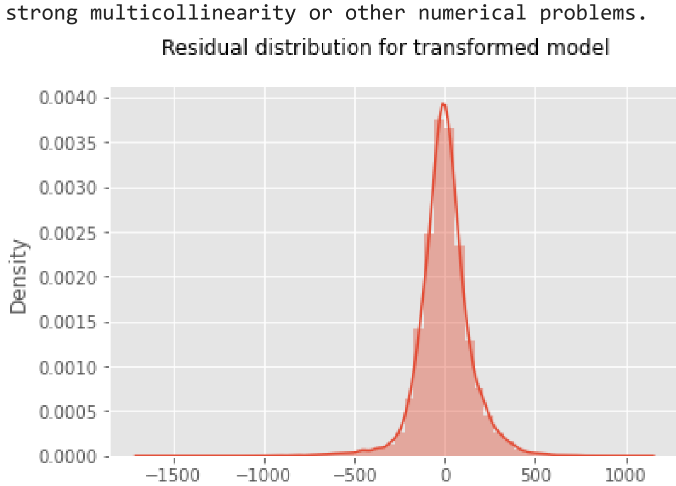
28004 rows × 31 columns

In [152…

```
get_OLS_model('transformed', outlier_data, y_sqrt)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.638
Model:                            OLS   Adj. R-squared:                  0.638
Method:                 Least Squares   F-statistic:                     1591.
Date:                Fri, 10 Mar 2023   Prob (F-statistic):               0.00
Time:                        17:29:11   Log-Likelihood:            -1.7900e+05
No. Observations:               28004   AIC:                         3.581e+05
Df Residuals:                   27972   BIC:                         3.583e+05
Df Model:                          31
Covariance Type:            nonrobust
===================================================================================================
                                      coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------------
const                            -1.35e+04    996.223    -13.555      0.000   -1.55e+04   -1.16e+04
bedrooms                           -0.2917      1.284     -0.227      0.820      -2.809       2.226
```

```
bathrooms                        22.9976      1.828     12.581      0.000      19.415      26.580
sqft_lot                          0.0002   1.72e-05     10.834      0.000       0.000       0.000
floors                          -12.1468      2.296     -5.290      0.000     -16.648      -7.646
condition                        31.9052      1.382     23.089      0.000      29.197      34.614
grade                            66.9617      1.374     48.752      0.000      64.270      69.654
sqft_above                        0.0843      0.003     26.836      0.000       0.078       0.090
sqft_basement                     0.0330      0.003     12.313      0.000       0.028       0.038
sqft_garage                      -0.0172      0.004     -3.936      0.000      -0.026      -0.009
sqft_patio                        0.0320      0.004      7.635      0.000       0.024       0.040
yr_built                         -0.8065      0.046    -17.386      0.000      -0.897      -0.716
yr_renovated                      0.0160      0.002      7.013      0.000       0.012       0.020
lat                             577.1969      7.039     81.999      0.000     563.400     590.994
long                            102.7232      7.529     13.643      0.000      87.965     117.481
school_rating                    20.8403      0.671     31.075      0.000      19.526      22.155
month                             9.0914      3.024      3.006      0.003       3.164      15.019
day_of_year                      -0.5309      0.099     -5.349      0.000      -0.725      -0.336
sewer_PRIVATE RESTRICTED        -10.6824     64.837     -0.165      0.869    -137.766     116.402
sewer_PUBLIC                     27.2432      2.886      9.440      0.000      21.587      32.900
sewer_PUBLIC RESTRICTED          20.3019    102.283      0.198      0.843    -180.177     220.781
heat_source_Electricity/Solar   -35.2264     19.438     -1.812      0.070     -73.327       2.874
heat_source_Gas                  16.2656      2.344      6.938      0.000      11.671      20.861
heat_source_Gas/Solar            65.8071     15.931      4.131      0.000      34.582      97.032
heat_source_Oil                  -6.5076      3.553     -1.832      0.067     -13.472       0.457
heat_source_Oil/Solar            12.7877     72.335      0.177      0.860    -128.993     154.568
heat_source_Other                35.1208     33.328      1.054      0.292     -30.204     100.446
waterfront                       61.7469      8.580      7.196      0.000      44.929      78.565
nuisance                        -15.4541      2.364     -6.538      0.000     -20.087     -10.821
view                             28.8273      1.251     23.043      0.000      26.375      31.279
greenbelt                        45.4650      5.607      8.108      0.000      34.475      56.455
sqft_living_log                  30.7669      6.226      4.942      0.000      18.564      42.970
==============================================================================
Omnibus:                     3939.729   Durbin-Watson:                   2.002
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            38097.523
Skew:                          -0.358   Prob(JB):                         0.00
Kurtosis:                       8.669   Cond. No.                     6.64e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.64e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```

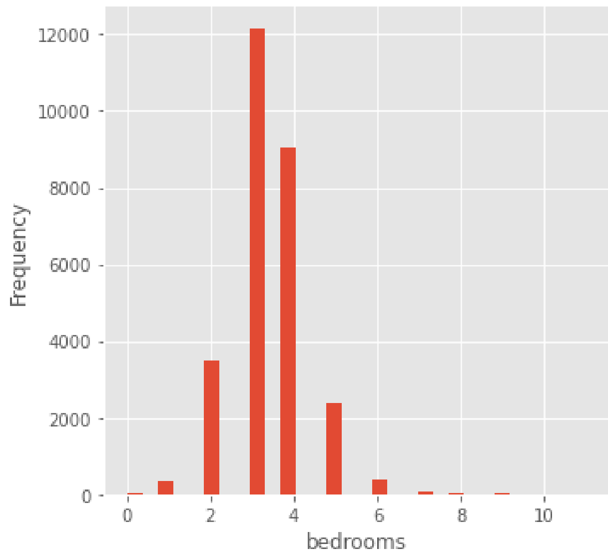


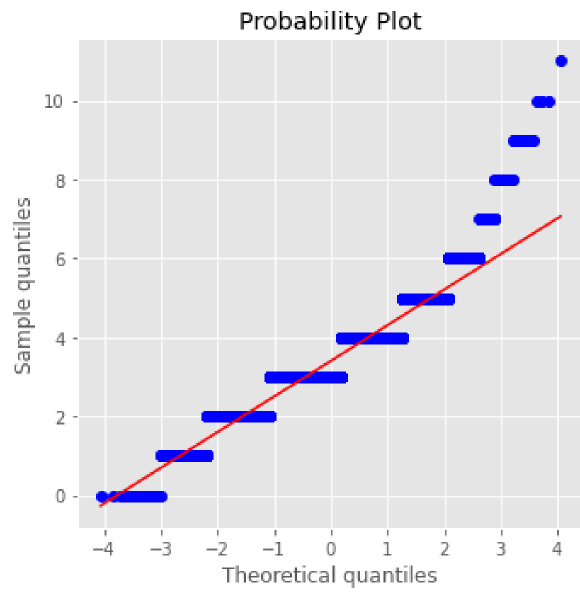
Out[152…

```
(None,
 Text(0.5, 0.98, 'Residual distribution for transformed model'),
 <AxesSubplot:ylabel='Density'>,
 None)
```

In [153…

```
plot_hist_qq(outlier_data, 'bedrooms')
```





pval > 0.05

- bedrooms - will be dropped from the current model

In [154…

```
outlier_data = outlier_data.drop(['bedrooms'], axis=1)
```

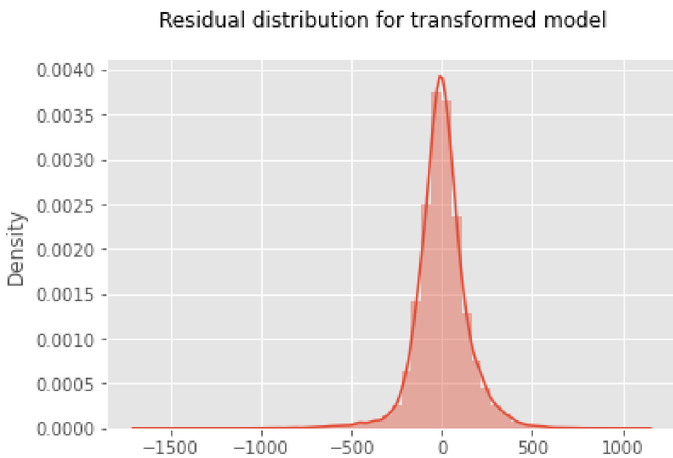
Rerun model

In [155…

```
get_OLS_model('transformed', outlier_data, y_sqrt)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.638
Model:                            OLS   Adj. R-squared:                  0.638
Method:                 Least Squares   F-statistic:                     1644.
Date:                Fri, 10 Mar 2023   Prob (F-statistic):               0.00
Time:                        17:29:33   Log-Likelihood:            -1.7900e+05
No. Observations:               28004   AIC:                         3.581e+05
Df Residuals:                   27973   BIC:                         3.583e+05
Df Model:                          30
Covariance Type:            nonrobust
===================================================================================================
                                     coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------------------
const                           -1.35e+04    996.206    -13.555      0.000   -1.55e+04   -1.16e+04
bathrooms                         22.9062      1.783     12.847      0.000      19.411      26.401
sqft_lot                           0.0002   1.72e-05     10.854      0.000       0.000       0.000
floors                           -12.1136      2.292     -5.286      0.000     -16.605      -7.622
condition                         31.8985      1.382     23.090      0.000      29.191      34.606
grade                             67.0021      1.362     49.198      0.000      64.333      69.671
sqft_above                         0.0843      0.003     26.841      0.000       0.078       0.090
sqft_basement                      0.0330      0.003     12.318      0.000       0.028       0.038
sqft_garage                       -0.0172      0.004     -3.933      0.000      -0.026      -0.009
sqft_patio                         0.0320      0.004      7.658      0.000       0.024       0.040
yr_built                          -0.8058      0.046    -17.413      0.000      -0.897      -0.715
yr_renovated                       0.0160      0.002      7.028      0.000       0.012       0.020
lat                              577.2484      7.035     82.050      0.000     563.459     591.038
long                             102.7379      7.529     13.646      0.000      87.981     117.495
school_rating                     20.8466      0.670     31.110      0.000      19.533      22.160
month                              9.0900      3.024      3.006      0.003       3.163      15.017
day_of_year                       -0.5309      0.099     -5.349      0.000      -0.725      -0.336
sewer_PRIVATE RESTRICTED         -10.6996     64.836     -0.165      0.869    -137.781     116.382
sewer_PUBLIC                      27.2052      2.881      9.443      0.000      21.558      32.852
sewer_PUBLIC RESTRICTED           20.1599    102.279      0.197      0.844    -180.312     220.632
heat_source_Electricity/Solar    -35.2728     19.437     -1.815      0.070     -73.370       2.825
heat_source_Gas                   16.2630      2.344      6.937      0.000      11.668      20.858
heat_source_Gas/Solar             65.8436     15.930      4.133      0.000      34.621      97.066
heat_source_Oil                   -6.5198      3.553     -1.835      0.066     -13.483       0.443
```

```
heat_source_Oil/Solar          12.9767     72.329      0.179      0.858    -128.792     154.746
heat_source_Other              35.1673     33.327      1.055      0.291     -30.155     100.490
waterfront                     61.8135      8.575      7.208      0.000      45.005      78.621
nuisance                      -15.4565      2.364     -6.539      0.000     -20.089     -10.824
view                           28.8507      1.247     23.142      0.000      26.407      31.294
greenbelt                      45.4778      5.607      8.111      0.000      34.488      56.468
sqft_living_log                30.3501      5.949      5.102      0.000      18.690      42.010
==============================================================================
Omnibus:                     3939.335   Durbin-Watson:                   2.002
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            38093.563
Skew:                          -0.358   Prob(JB):                         0.00
Kurtosis:                       8.669   Cond. No.                     6.64e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.64e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```



Out[155…

```
(None,
 Text(0.5, 0.98, 'Residual distribution for transformed model'),
 <AxesSubplot:ylabel='Density'>,
 None)
```

Dropping sewer/heat source data

In [156…

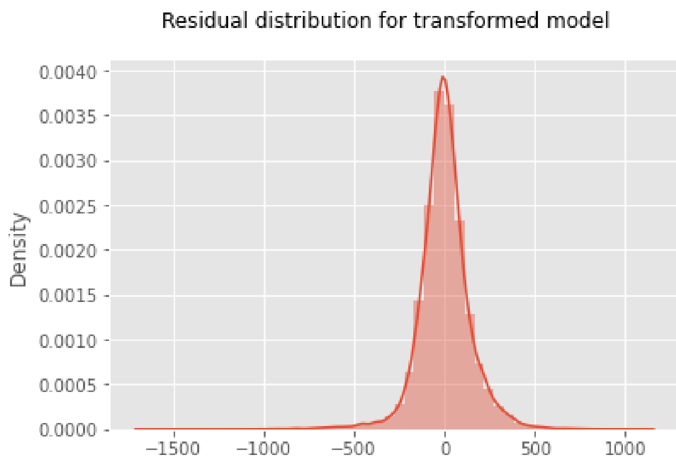
```python
new_outlier_data = outlier_data.drop(['sewer_PRIVATE RESTRICTED','sewer_PUBLIC RESTRICTED', 'heat_source_Oil', 'heat_s
```

In [157…

```python
get_OLS_model('transformed', new_outlier_data, y_sqrt)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.638
Model:                            OLS   Adj. R-squared:                  0.638
Method:                 Least Squares   F-statistic:                     2055.
Date:                Fri, 10 Mar 2023   Prob (F-statistic):               0.00
Time:                        17:29:42   Log-Likelihood:            -1.7901e+05
No. Observations:               28004   AIC:                         3.581e+05
Df Residuals:                   27979   BIC:                         3.583e+05
Df Model:                          24
Covariance Type:            nonrobust
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
const                  -1.35e+04    996.005    -13.550      0.000   -1.54e+04   -1.15e+04
bathrooms                23.2745      1.769     13.157      0.000      19.807      26.742
sqft_lot                  0.0002   1.72e-05     10.955      0.000       0.000       0.000
floors                  -11.9917      2.286     -5.247      0.000     -16.471      -7.512
condition                32.1613      1.370     23.480      0.000      29.477      34.846
grade                    66.9431      1.361     49.190      0.000      64.276      69.611
sqft_above                0.0840      0.003     26.796      0.000       0.078       0.090
sqft_basement             0.0327      0.003     12.226      0.000       0.027       0.038
sqft_garage              -0.0173      0.004     -3.974      0.000      -0.026      -0.009
sqft_patio                0.0325      0.004      7.790      0.000       0.024       0.041
```

```
yr_built                 -0.7915      0.046    -17.346      0.000      -0.881      -0.702
yr_renovated              0.0163      0.002      7.202      0.000       0.012       0.021
lat                     577.3020      7.035     82.059      0.000     563.513     591.091
long                    103.0530      7.526     13.693      0.000      88.302     117.804
school_rating            20.8322      0.670     31.095      0.000      19.519      22.145
month                     9.1006      3.023      3.010      0.003       3.175      15.027
day_of_year              -0.5313      0.099     -5.353      0.000      -0.726      -0.337
sewer_PUBLIC             26.8222      2.874      9.333      0.000      21.189      32.455
heat_source_Gas          18.4700      2.064      8.948      0.000      14.424      22.516
heat_source_Gas/Solar    68.0729     15.891      4.284      0.000      36.926      99.220
waterfront               62.6105      8.564      7.311      0.000      45.824      79.397
nuisance                -15.4416      2.364     -6.533      0.000     -20.074     -10.809
view                     28.8205      1.246     23.130      0.000      26.378      31.263
greenbelt                45.4525      5.607      8.107      0.000      34.463      56.442
sqft_living_log          29.9818      5.945      5.043      0.000      18.329      41.634
==============================================================================
Omnibus:                     3940.818   Durbin-Watson:                   2.002
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            38067.661
Skew:                          -0.359   Prob(JB):                         0.00
Kurtosis:                       8.667   Cond. No.                     6.63e+07
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.63e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
```



Out[157…

```
(None,
 Text(0.5, 0.98, 'Residual distribution for transformed model'),
 <AxesSubplot:ylabel='Density'>,
 None)
```

Observations

- pval > 0.05

`bedrooms` - dropped from the current model

- all variables are statistically significant (pvalue < 0.05)
- Durbin-Watson Score continues to be "fine" but not improve a whole lot.
- Jarque-Bera Score continues to improve but still must come down
- skewness is now an afterthought as its at a very low -0.347 Overall no real improvement of the model happens here, we will try adding in new variables to improve as well as revisit VIFs to likely drop all that were originally at extremely high levels.

Next steps to improve the model:

1. revisit VIFs to see if any variables(now that outliers are removed and data has been transformed) should now be dropped from the model.
2. New predictors will be engineered to be added to the model. The next focus will be on the zipcodes in an attempt to narrow down the data with location-dependent price points. Possible data to be looked at are:

- waterfronts

- views
- school districts: rating, and school taxes
- tax brackets

Jarque-Beras score and skew level continue to improve but there is still some work to do.

Rechecking VIFs

In [159...

```
from statsmodels.stats.outliers_influence import variance_inflation_factor

# Load your data into a pandas DataFrame
data = new_outlier_data

# Get a list of the column names
cols = data.columns

# Create an empty DataFrame to hold the VIF results
vif_data = pd.DataFrame()

# Loop through each column and calculate the VIF
for i in range(len(cols)):
    vif = variance_inflation_factor(data[cols].values, i)
    vif_data = vif_data.append({'Variable': cols[i], 'VIF': vif}, ignore_index=True)

# Print the VIF results
print(vif_data)
```

```
                Variable            VIF
0              bathrooms      24.288846
1               sqft_lot       1.299746
2                 floors      17.348050
3              condition      31.675593
4                  grade     139.069100
5             sqft_above      48.182817
6          sqft_basement       4.898548
7            sqft_garage       4.596102
8             sqft_patio       2.243068
9               yr_built    9648.237288
10          yr_renovated       1.206333
11                   lat  134915.436533
12                  long  146175.174281
13         school_rating      22.303406
14                 month     699.107278
15           day_of_year     614.257403
16          sewer_PUBLIC       8.789766
17       heat_source_Gas       3.863875
18  heat_source_Gas/Solar       1.015124
19            waterfront       1.202711
20              nuisance       1.268682
21                  view       1.425945
22             greenbelt       1.061947
23       sqft_living_log    2675.582034
```

Scaling data

In [160...

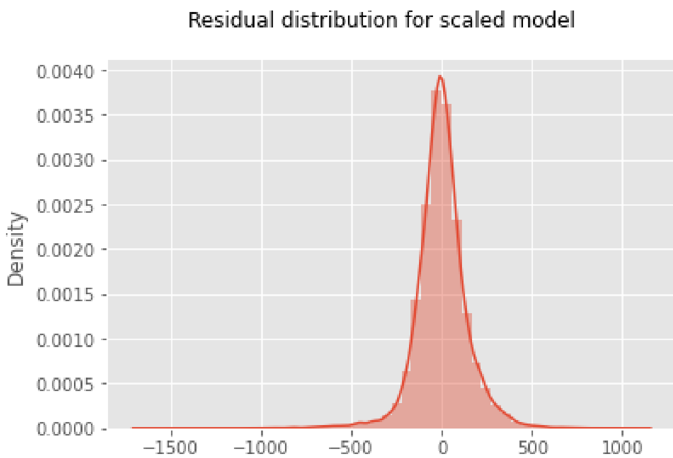
```
scaledX = (new_outlier_data - np.mean(new_outlier_data)) / np.std(new_outlier_data)
```

In [161...

```
get_OLS_model('scaled',scaledX, y_sqrt)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.638
Model:                            OLS   Adj. R-squared:                  0.638
Method:                 Least Squares   F-statistic:                     2055.
Date:                Fri, 10 Mar 2023   Prob (F-statistic):               0.00
Time:                        17:31:22   Log-Likelihood:            -1.7901e+05
No. Observations:               28004   AIC:                         3.581e+05
Df Residuals:                   27979   BIC:                         3.583e+05
```

```
Df Model:                          24
Covariance Type:            nonrobust
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
const                   963.8260      0.864   1115.774      0.000     962.133     965.519
bathrooms                19.0895      1.451     13.157      0.000      16.246      21.933
sqft_lot                 10.3831      0.948     10.955      0.000       8.525      12.241
floors                   -6.5607      1.250     -5.247      0.000      -9.012      -4.110
condition                22.8152      0.972     23.480      0.000      20.911      24.720
grade                    69.7631      1.418     49.190      0.000      66.983      72.543
sqft_above               65.2076      2.433     26.796      0.000      60.438      69.977
sqft_basement            18.1476      1.484     12.226      0.000      15.238      21.057
sqft_garage              -4.7843      1.204     -3.974      0.000      -7.144      -2.424
sqft_patio                7.5481      0.969      7.790      0.000       5.649       9.447
yr_built                -24.9442      1.438    -17.346      0.000     -27.763     -22.126
yr_renovated              6.6952      0.930      7.202      0.000       4.873       8.517
lat                      86.1110      1.049     82.059      0.000      84.054      88.168
long                     14.8860      1.087     13.693      0.000      12.755      17.017
school_rating            30.8462      0.992     31.095      0.000      28.902      32.791
month                    28.2020      9.369      3.010      0.003       9.838      46.566
day_of_year             -50.1549      9.369     -5.353      0.000     -68.518     -31.792
sewer_PUBLIC              9.5675      1.025      9.333      0.000       7.558      11.577
heat_source_Gas           8.5882      0.960      8.948      0.000       6.707      10.469
heat_source_Gas/Solar     3.7226      0.869      4.284      0.000       2.019       5.426
waterfront                6.8867      0.942      7.311      0.000       5.040       8.733
nuisance                 -5.7981      0.887     -6.533      0.000      -7.538      -4.059
view                     22.6720      0.980     23.130      0.000      20.751      24.593
greenbelt                 7.1597      0.883      8.107      0.000       5.429       8.891
sqft_living_log          12.5658      2.492      5.043      0.000       7.682      17.450
==============================================================================
Omnibus:                     3940.818   Durbin-Watson:                   2.002
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            38067.661
Skew:                          -0.359   Prob(JB):                         0.00
Kurtosis:                       8.667   Cond. No.                         33.1
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```



Out[161…

```
(None,
 Text(0.5, 0.98, 'Residual distribution for scaled model'),
 <AxesSubplot:ylabel='Density'>,
 None)
```

In [162…

```
get_vifs(scaledX)
```

```
                Variable        VIF
0              bathrooms   2.821152
1               sqft_lot   1.203865
2                 floors   2.095425
3              condition   1.265329
4                  grade   2.695549
5             sqft_above   7.936066
6          sqft_basement   2.952938
7            sqft_garage   1.942777
```

```
8             sqft_patio    1.258357
9               yr_built    2.771267
10          yr_renovated    1.158071
11                   lat    1.475757
12                  long    1.583765
13         school_rating    1.318808
14                 month  117.641746
15           day_of_year  117.630036
16          sewer_PUBLIC    1.408264
17       heat_source_Gas    1.234423
18  heat_source_Gas/Solar    1.012082
19            waterfront    1.189211
20              nuisance    1.055506
21                  view    1.287633
22             greenbelt    1.045362
23       sqft_living_log    8.320085
```

Adding waterfront dummies to the model

In [163…

```
water_data = pd.concat([scaledX,waterfront_dummies], axis=1)
```

In [164…

```
water_data.columns
```

Out[164…

```
Index(['bathrooms', 'sqft_lot', 'floors', 'condition', 'grade', 'sqft_above',
       'sqft_basement', 'sqft_garage', 'sqft_patio', 'yr_built',
       'yr_renovated', 'lat', 'long', 'school_rating', 'month', 'day_of_year',
       'sewer_PUBLIC', 'heat_source_Gas', 'heat_source_Gas/Solar',
       'waterfront', 'nuisance', 'view', 'greenbelt', 'sqft_living_log',
       'water_Elliot Bay', 'water_Lake Sammamish', 'water_Lake Washington',
       'water_Puget Sound', 'water_other'],
      dtype='object')
```

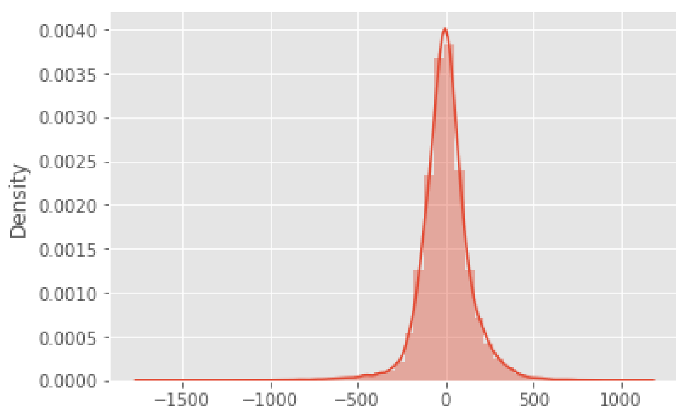
In [165…

```
get_OLS_model('waterfront',water_data,y_sqrt)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.643
Model:                            OLS   Adj. R-squared:                  0.643
Method:                 Least Squares   F-statistic:                     1739.
Date:                Fri, 10 Mar 2023   Prob (F-statistic):               0.00
Time:                        17:35:13   Log-Likelihood:            -1.7881e+05
No. Observations:               28004   AIC:                         3.577e+05
Df Residuals:                   27974   BIC:                         3.579e+05
Df Model:                          29
Covariance Type:            nonrobust
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
const                   976.6862      6.723    145.275      0.000     963.509     989.864
bathrooms                18.7417      1.441     13.005      0.000      15.917      21.566
sqft_lot                 10.8990      0.943     11.559      0.000       9.051      12.747
floors                   -6.8945      1.242     -5.549      0.000      -9.330      -4.459
condition                23.3661      0.965     24.203      0.000      21.474      25.258
grade                    67.2257      1.416     47.469      0.000      64.450      70.002
sqft_above               64.3625      2.418     26.618      0.000      59.623      69.102
sqft_basement            17.7966      1.476     12.061      0.000      14.904      20.689
sqft_garage              -4.1468      1.199     -3.458      0.001      -6.498      -1.796
sqft_patio                8.1893      0.963      8.505      0.000       6.302      10.077
yr_built                -23.7499      1.432    -16.583      0.000     -26.557     -20.943
yr_renovated              7.0150      0.923      7.598      0.000       5.205       8.825
lat                      88.8548      1.088     81.671      0.000      86.722      90.987
long                     10.8223      1.114      9.719      0.000       8.640      13.005
school_rating            27.9965      1.052     26.616      0.000      25.935      30.058
month                    28.2488      9.304      3.036      0.002      10.013      46.485
day_of_year             -50.2341      9.303     -5.400      0.000     -68.469     -31.999
sewer_PUBLIC              6.6492      1.052      6.322      0.000       4.588       8.711
heat_source_Gas           8.7973      0.954      9.224      0.000       6.928      10.667
heat_source_Gas/Solar     3.5964      0.863      4.167      0.000       1.905       5.288
waterfront                6.9197      0.938      7.379      0.000       5.082       8.758
nuisance                 -5.7070      0.882     -6.469      0.000      -7.436      -3.978
```

```
view                        22.6447      0.974     23.243      0.000      20.735      24.554
greenbelt                    7.1693      0.877      8.172      0.000       5.450       8.889
sqft_living_log             14.2442      2.476      5.753      0.000       9.391      19.097
water_Elliot Bay           -39.0850      8.589     -4.551      0.000     -55.920     -22.250
water_Lake Sammamish        63.0784      8.636      7.305      0.000      46.152      80.005
water_Lake Washington      -77.2032      9.562     -8.074      0.000     -95.945     -58.462
water_Puget Sound          -27.5858      8.540     -3.230      0.001     -44.325     -10.846
water_other                -13.8228      6.784     -2.037      0.042     -27.120      -0.525
==============================================================================
Omnibus:                     4051.157   Durbin-Watson:                   1.999
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            41380.974
Skew:                          -0.360   Prob(JB):                         0.00
Kurtosis:                       8.911   Cond. No.                         43.5
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Residual distribution for waterfront model



Out[165…

```
(None,
 Text(0.5, 0.98, 'Residual distribution for waterfront model'),
 <AxesSubplot:ylabel='Density'>,
 None)
```

Elliot Bay and Puget Sound present high pvalues indicating a lack of statistical significance. These will be dropped from the model.

In [166…

```
water_data = water_data.drop(['water_other'], axis=1)
```

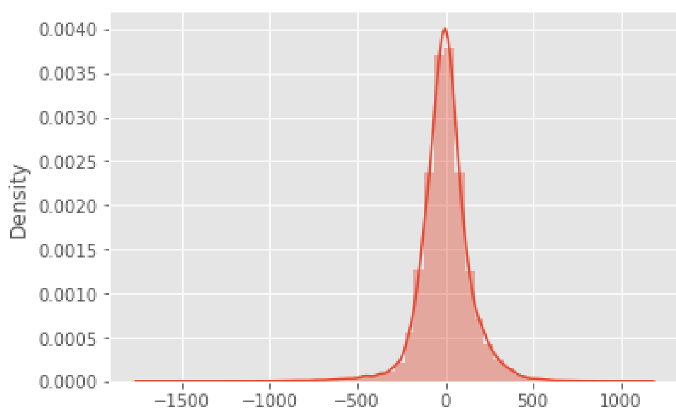
In [167…

```
get_OLS_model('waterfront',water_data,y_sqrt)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.643
Model:                            OLS   Adj. R-squared:                  0.643
Method:                 Least Squares   F-statistic:                     1800.
Date:                Fri, 10 Mar 2023   Prob (F-statistic):               0.00
Time:                        17:35:26   Log-Likelihood:            -1.7881e+05
No. Observations:               28004   AIC:                         3.577e+05
Df Residuals:                   27975   BIC:                         3.579e+05
Df Model:                          28
Covariance Type:            nonrobust
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
const                   963.1152      0.913   1054.524      0.000     961.325     964.905
bathrooms                18.7388      1.441     13.002      0.000      15.914      21.564
sqft_lot                 10.8676      0.943     11.526      0.000       9.020      12.716
floors                   -6.8505      1.242     -5.514      0.000      -9.286      -4.415
condition                23.3259      0.965     24.165      0.000      21.434      25.218
grade                    67.2016      1.416     47.451      0.000      64.426      69.978
sqft_above               64.4402      2.418     26.652      0.000      59.701      69.179
sqft_basement            17.8321      1.476     12.085      0.000      14.940      20.724
sqft_garage              -4.1655      1.199     -3.473      0.001      -6.516      -1.815
sqft_patio                8.1537      0.963      8.469      0.000       6.267      10.041
yr_built                -23.8822      1.431    -16.691      0.000     -26.687     -21.078
yr_renovated              7.0119      0.923      7.594      0.000       5.202       8.822
```

```
lat                       89.0352      1.084     82.104      0.000      86.910      91.161
long                      10.6828      1.112      9.611      0.000       8.504      12.861
school_rating             27.4121      1.012     27.085      0.000      25.428      29.396
month                     28.2761      9.304      3.039      0.002      10.039      46.513
day_of_year              -50.2469      9.304     -5.401      0.000     -68.483     -32.011
sewer_PUBLIC               6.5177      1.050      6.208      0.000       4.460       8.575
heat_source_Gas            8.8023      0.954      9.229      0.000       6.933      10.672
heat_source_Gas/Solar      3.6083      0.863      4.181      0.000       1.917       5.300
waterfront                 6.8824      0.938      7.341      0.000       5.045       8.720
nuisance                  -5.6511      0.882     -6.408      0.000      -7.380      -3.923
view                      22.6665      0.974     23.265      0.000      20.757      24.576
greenbelt                  7.1836      0.877      8.188      0.000       5.464       8.903
sqft_living_log           14.2023      2.476      5.736      0.000       9.349      19.056
water_Elliot Bay         -25.8213      5.603     -4.608      0.000     -36.804     -14.839
water_Lake Sammamish      77.5858      4.886     15.878      0.000      68.008      87.163
water_Lake Washington    -63.3047      6.701     -9.448      0.000     -76.438     -50.171
water_Puget Sound        -14.4146      5.581     -2.583      0.010     -25.354      -3.475
==============================================================================
Omnibus:                     4051.884   Durbin-Watson:                   1.999
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            41320.495
Skew:                          -0.361   Prob(JB):                         0.00
Kurtosis:                       8.907   Cond. No.                         33.1
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Residual distribution for waterfront model



Out[167…

```
(None,
 Text(0.5, 0.98, 'Residual distribution for waterfront model'),
 <AxesSubplot:ylabel='Density'>,
 None)
```

Recheck VIFs

In [168…

```
get_vifs(water_data)
```

```
                Variable         VIF
0              bathrooms    2.822800
1               sqft_lot    1.208054
2                 floors    2.097623
3              condition    1.266312
4                  grade    2.725312
5             sqft_above    7.944928
6          sqft_basement    2.958500
7            sqft_garage    1.954332
8             sqft_patio    1.259752
9               yr_built    2.782233
10          yr_renovated    1.158737
11                   lat    1.594028
12                  long    1.678379
13         school_rating    1.391194
14                 month  117.653026
15           day_of_year  117.640224
16          sewer_PUBLIC    1.495934
17       heat_source_Gas    1.236235
```

```
18  heat_source_Gas/Solar    1.012365
19             waterfront    1.194632
20               nuisance    1.056843
21                   view    1.290019
22              greenbelt    1.046040
23        sqft_living_log    8.332512
24        water_Elliot Bay   1.045550
25    water_Lake Sammamish   1.183889
26  water_Lake Washington    1.161711
27        water_Puget Sound  1.055570
```

Month and day_of_year present with high variance inflation factors indicating possible collinearity. These will be dropped.

In [169…

```
water_data = water_data.drop(['month','day_of_year','sqft_living_log'], axis =1)
```

In [170…

```
get_vifs(water_data)
```

```
                 Variable       VIF
0               bathrooms  2.613181
1                sqft_lot  1.206485
2                  floors  2.085134
3               condition  1.240220
4                   grade  2.686084
5              sqft_above  3.274387
6           sqft_basement  1.792147
7             sqft_garage  1.936027
8              sqft_patio  1.258711
9                yr_built  2.764872
10           yr_renovated  1.156089
11                    lat  1.590733
12                   long  1.671893
13          school_rating  1.390790
14           sewer_PUBLIC  1.495842
15        heat_source_Gas  1.227726
16  heat_source_Gas/Solar  1.012330
17             waterfront  1.194429
18               nuisance  1.056327
19                   view  1.289824
20              greenbelt  1.046010
21       water_Elliot Bay  1.045322
22   water_Lake Sammamish  1.182965
23  water_Lake Washington  1.161392
24      water_Puget Sound  1.054799
```

All VIFs are now below 3 aside from sqft_above, meaning the issue of collinearity is now for the most part solved.

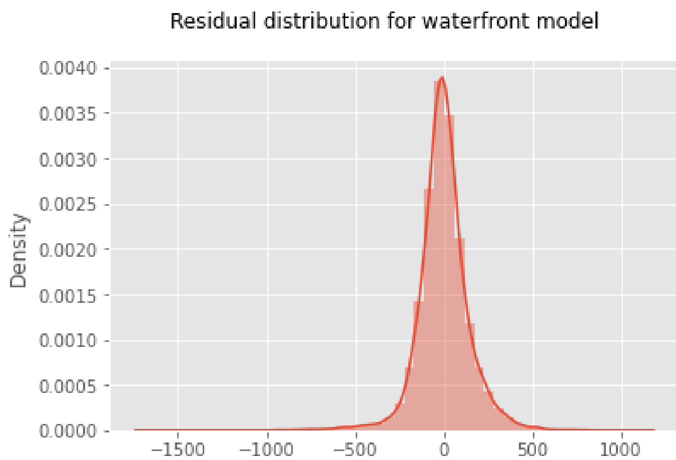
Final model

In [171…

```
get_OLS_model('waterfront',water_data,y_sqrt)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.634
Model:                            OLS   Adj. R-squared:                  0.634
Method:                 Least Squares   F-statistic:                     1940.
Date:                Fri, 10 Mar 2023   Prob (F-statistic):               0.00
Time:                        17:36:58   Log-Likelihood:            -1.7916e+05
No. Observations:               28004   AIC:                         3.584e+05
Df Residuals:                   27978   BIC:                         3.586e+05
Df Model:                          25
Covariance Type:            nonrobust
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
const                   963.2348      0.925   1041.678      0.000     961.422     965.047
bathrooms                20.2542      1.404     14.427      0.000      17.502      23.006
sqft_lot                 10.4141      0.954     10.916      0.000       8.544      12.284
floors                   -7.0520      1.254     -5.623      0.000      -9.510      -4.594
condition                24.0354      0.967     24.850      0.000      22.140      25.931
```

```
grade                     68.4592      1.424     48.090      0.000      65.669      71.249
sqft_above                74.7971      1.572     47.593      0.000      71.717      77.878
sqft_basement             23.0438      1.163     19.817      0.000      20.765      25.323
sqft_garage               -4.5841      1.209     -3.793      0.000      -6.953      -2.215
sqft_patio                 8.2143      0.974      8.430      0.000       6.304      10.124
yr_built                 -23.5442      1.444    -16.303      0.000     -26.375     -20.714
yr_renovated               7.2267      0.934      7.739      0.000       5.396       9.057
lat                       88.8161      1.097     80.977      0.000      86.666      90.966
long                      11.3439      1.123     10.099      0.000       9.142      13.545
school_rating             27.3538      1.025     26.698      0.000      25.346      29.362
sewer_PUBLIC               6.5841      1.063      6.194      0.000       4.501       8.668
heat_source_Gas            9.2782      0.962      9.641      0.000       7.392      11.164
heat_source_Gas/Solar      3.6079      0.874      4.129      0.000       1.895       5.321
waterfront                 6.7153      0.949      7.075      0.000       4.855       8.576
nuisance                  -6.0733      0.893     -6.803      0.000      -7.823      -4.324
view                      22.4724      0.986     22.783      0.000      20.539      24.406
greenbelt                  7.1611      0.888      8.062      0.000       5.420       8.902
water_Elliot Bay         -25.2860      5.672     -4.458      0.000     -36.404     -14.168
water_Lake Sammamish      75.7294      4.946     15.313      0.000      66.036      85.423
water_Lake Washington    -64.3982      6.783     -9.494      0.000     -77.694     -51.103
water_Puget Sound        -16.0049      5.649     -2.833      0.005     -27.077      -4.933
==============================================================================
Omnibus:                     3781.180   Durbin-Watson:                   2.004
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            36246.490
Skew:                          -0.323   Prob(JB):                         0.00
Kurtosis:                       8.536   Cond. No.                         15.9
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```



Out[171…

```
(None,
 Text(0.5, 0.98, 'Residual distribution for waterfront model'),
 <AxesSubplot:ylabel='Density'>,
 None)
```

In [172…

```
water_data.columns
```

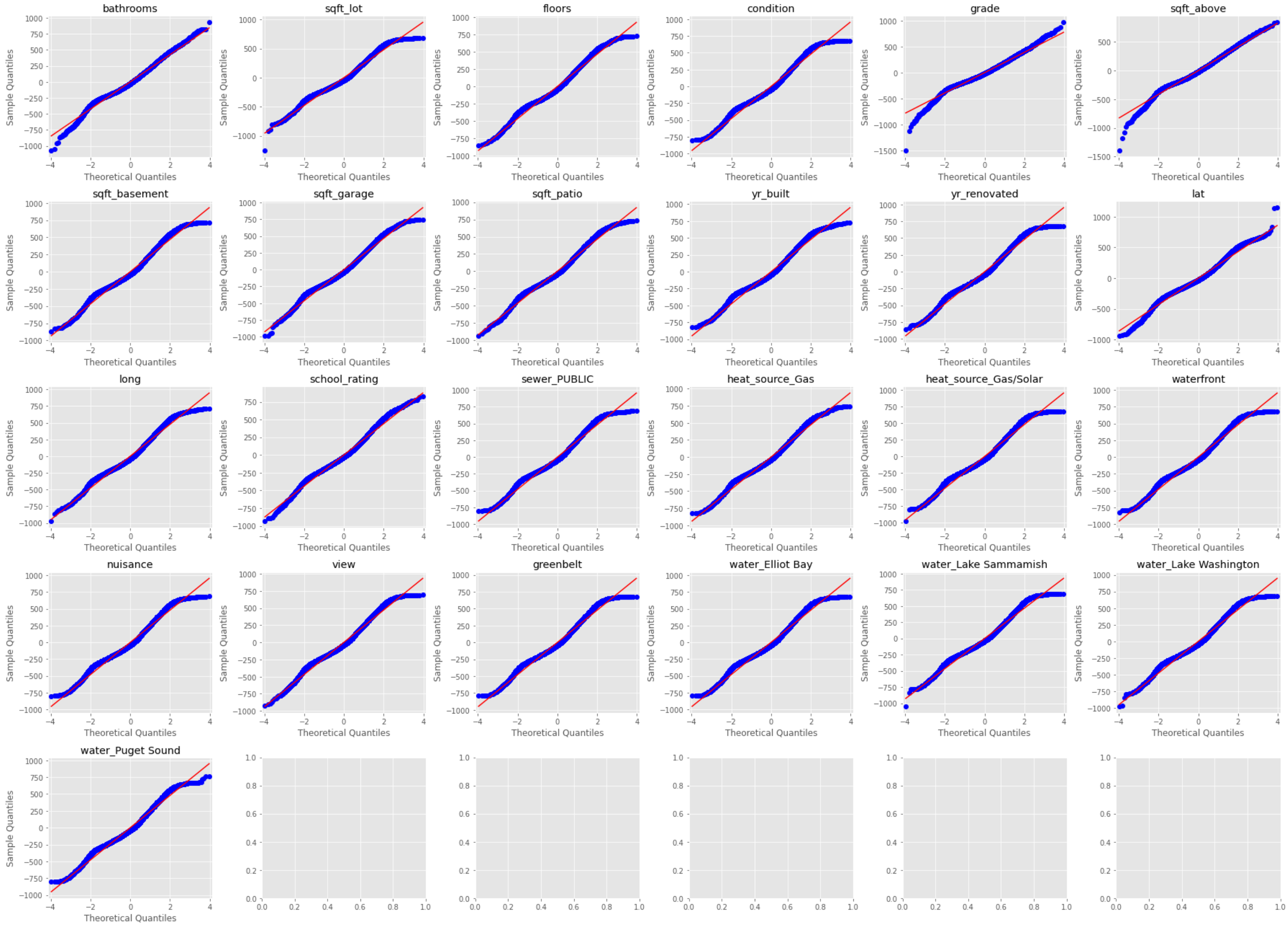
Out[172…

```
Index(['bathrooms', 'sqft_lot', 'floors', 'condition', 'grade', 'sqft_above',
       'sqft_basement', 'sqft_garage', 'sqft_patio', 'yr_built',
       'yr_renovated', 'lat', 'long', 'school_rating', 'sewer_PUBLIC',
       'heat_source_Gas', 'heat_source_Gas/Solar', 'waterfront', 'nuisance',
       'view', 'greenbelt', 'water_Elliot Bay', 'water_Lake Sammamish',
       'water_Lake Washington', 'water_Puget Sound'],
      dtype='object')
```

Constructing QQplots for all independent variables within the model

In [173…

```
get_model_qqplots(water_data, y_sqrt)
```



In [174…

```
model = sm.OLS(y_sqrt, sm.add_constant(water_data))
results = model.fit()
model_residual = results.resid
model_params = results.params

print(results.params)
```

```
const                     963.234796
bathrooms                  20.254186
sqft_lot                   10.414107
floors                     -7.052045
condition                  24.035449
grade                      68.459181
sqft_above                 74.797094
sqft_basement              23.043827
sqft_garage                -4.584058
sqft_patio                  8.214320
yr_built                  -23.544185
yr_renovated                7.226726
lat                        88.816082
long                       11.343899
school_rating              27.353843
sewer_PUBLIC                6.584108
heat_source_Gas             9.278196
heat_source_Gas/Solar       3.607930
waterfront                  6.715330
nuisance                   -6.073275
view                       22.472439
greenbelt                   7.161059
water_Elliot Bay          -25.286029
water_Lake Sammamish       75.729413
water_Lake Washington     -64.398218
```

```
water_Puget Sound        -16.004882
dtype: float64
```

We have a linear model with the dependent variable (price) square root transformed, and the following independent variables and their corresponding coefficients:

- const: 963.234796
- bathrooms: 20.254186
- sqft_lot: 10.414107
- floors: -7.052045
- condition: 24.035449
- grade: 68.459181
- sqft_above: 74.797094
- sqft_basement: 23.043827
- sqft_garage: -4.584058
- sqft_patio: 8.214320
- yr_built: -23.544185
- yr_renovated: 7.226726
- lat: 88.816082
- long: 11.343899
- school_rating: 27.353843
- sewer_PUBLIC: 6.584108
- heat_source_Gas: 9.278196
- heat_source_Gas/Solar: 3.607930
- waterfront: 6.715330
- nuisance: -6.073275
- view: 22.472439
- greenbelt: 7.161059
- water_Elliot Bay: -25.286029
- water_Lake Sammamish: 75.729413
- water_Lake Washington: -64.398218
- water_Puget Sound: -16.004882

A positive coefficient indicates that as the corresponding independent variable increases, the square root of the price of the house also increases, while a negative coefficient indicates that as the corresponding independent variable increases, the square root of the price of the house decreases.

In this model, we see that the most important variable in predicting the square root of house prices is the latitude of the house, with a coefficient of 100.368386. This suggests that houses located further north tend to have higher prices. The next most important variable is water proximity, with Water_Lake Sammamish variable having a very high coefficient of 75.729, suggesting that houses located near this lake tend to have much higher prices than other houses. On the other hand, the Water_Lake Washington variable has a negative coefficient, indicating that houses located near this lake tend to have lower prices than other houses.

Other important variables include the grade of the house, the square footage of the house above ground, and the condition of the house, all with coefficients greater than 20. The number of bathrooms, square footage of the basement, and the size of the view from the house are also important, with coefficients greater than 15.

On the other hand, variables such as the square footage of the garage and the presence of a nuisance nearby have negative coefficients, indicating that houses with larger garages or located near nuisances tend to have lower prices. The year the house was built and the longitude of the house also have negative coefficients, suggesting that older houses and houses located further west tend to have lower prices.

Overall, these results suggest that there are many factors that contribute to the price of a house, and that location, house size and quality, and the presence of nearby amenities all play important roles in determining the square root of house prices.

The constant coefficient in a linear regression model represents the expected value of the dependent variable (in this case, the square root of the price) when all the independent variables are equal to zero. Therefore, as the constant coefficient is 963.234796, we would expect the square root of the price to be around 963 when all the independent variables are zero. However, it's important to note that in the context of the model, there may not be any real-world scenarios where all the independent variables are actually zero. The constant term is mainly used as a baseline reference point for the other predictors in the model.

- As the number of bathrooms increases by one standard deviation, the square root price increases by 20.254186.
- As the size of the lot increases by one standard deviation, the square root price increases by 10.414107.
- As the number of floors increases by one standard deviation, the square root price decreases by 7.052045.
- As the condition of the house increases by one standard deviation, the square root price increases by 24.035449.
- As the grade of the house increases by one standard deviation, the square root price increases by 68.459181.
- As the size of the above ground living area increases by one standard deviation, the square root price increases by 74.797094.
- As the size of the basement living area increases by one standard deviation, the square root price increases by 23.043827.
- As the size of the garage increases by one standard deviation, the square root price decreases by 4.584058.
- As the size of the patio increases by one standard deviation, the square root price increases by 8.214320.
- As the age of the house (yr_built) increases by one standard deviation, the square root price decreases by 23.544185.
- As the year of renovation (yr_renovated) increases by one standard deviation, the square root price increases by 7.226726.
- As the latitude of the house increases by one standard deviation, the square root price increases by 88.816082.
- As the longitude of the house increases by one standard deviation, the square root price increases by 11.343899.
- As the school rating increases by one standard deviation, the square root price increases by 27.353843.
- As the house has a public sewer system (sewer_PUBLIC) instead of a private one, the square root price increases by 6.584108.
- As the heat source for the house switches from something other than gas to gas, the square root price increases by 9.278196.
- As the heat source for the house switches from something other than gas/solar to gas/solar, the square root price increases by 3.607930.
- As the house is on a waterfront property, the square root price increases by 6.715330.
- As the house experiences a nuisance (as defined by the model), the square root price decreases by 6.073275.
- As the view from the house improves by one standard deviation, the square root price increases by 22.472439.
- As the house is adjacent to a greenbelt, the square root price increases by 7.161059.
- As the house is located closer to Elliot Bay (in Seattle), the square root price decreases by 25.286029.
- As the house is located closer to Lake Sammamish, the square root price increases by 75.729413.
- As the house is located closer to Lake Washington, the square root price decreases by 64.398218.
- As the house is located closer to Puget Sound, the square root price decreases by 16.004882.

In [ ]: