

## Contents ⚙️

6.2 Visualizing the Comparison
6.3 Home History
6.3.1 Age of Home
6.3.2 Year Modified
7 Distance Data
7.1 Getting Infrastructure Data
7.1.1 Downtown
7.1.2 Hospitals
7.1.3 Police Stations
7.1.4 Schools
7.2 Finding Distance for Infrastructure
7.3 Exploring Distance Data
7.3.1 Downtown
7.3.2 Hospitals
7.3.3 Police Stations
7.3.4 Schools
8 Preparing for Modeling
8.0.1 Dropping View and Waterfront Properties from the Model
9 Zipcode
10 Seeing the Correlation
11 Building a model
11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2 Second Model: Transforming the Price with Log
12 Improving the Model with Adding Features
12.1 Third Model: Bedrooms, Floors, Closest School
12.2 Fourth Model: Square Foot Basement, Closest School
12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
12.4 Sixth Model: Condition, Waterfront and View
12.5 Seventh Model: Adding Zipcodes
12.6 Eighth Model: Latitude and Closest Hospital Distance
13 Conclusion
14 Recommendations:
14.0.1 Visualizing Schools

▼	<h1>1 Seattle Section 8 Housing Project</h1>
▼	<h2>2 Introduction</h2>
▼	<h3>2.1 Premise</h3> <p>Public/low income housing has become a somewhat controversial topic in Seattle. While many can agree there are two Seattles, or about the existence of a "homelessness issue," ideas about how to solve for this problem range from increased taxes on real estate investors, increased police budgets, to "defensive gentrification" (building buildings to stop people from sleeping there) to increasing unemployment rates and back-to-work programs. In the city's political history, candidates have proposed solutions to homelessness either larger or smaller, depending on their intended audience. In 2015, when this dataset is from, 45 homeless individuals were found dead in Seattle, which was a direct result of their homelessness, and the mayor declared a state of emergency. (CITATION: <a href="https://www.capitolhillseattle.com/2015/11/45-dead-in-2015-mayor-declares-seattle-homelessness-in-a-state-of-emergency/">https://www.capitolhillseattle.com/2015/11/45-dead-in-2015-mayor-declares-seattle-homelessness-in-a-state-of-emergency/</a> (<a href="https://www.capitolhillseattle.com/2015/11/45-dead-in-2015-mayor-declares-seattle-homelessness-in-a-state-of-emergency/">https://www.capitolhillseattle.com/2015/11/45-dead-in-2015-mayor-declares-seattle-homelessness-in-a-state-of-emergency/</a>))</p> <p>Hypothetical Premise: The City of Seattle city government, in partnership with Washington's state senate has a new pilot program called Section 8 Housing. There is a 100 million dollar budget in this pilot program to buy real estate and convert it into public housing all over the city. This project has 3 main goals:      1) to increase the availability of low-income housing      2) to increase property values      3) to decrease class disparity.</p> <p>Instead of making parts of the city into the projects, this program will spread public housing all over the city. This evidence based practice results in benefits for both the tenants and the properties. Socially, it reduces stigma and increases acceptance of low-income individuals.</p>

## Contents

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

well.

Each property bought for this project will undergo a renovation to living, and there is a separate budget set aside specifically for the project, I'm not responsible for balancing the cost of renovating the project.

I am instead tasked with using machine learning to build a model to recommend where to build more public housing, under the caveat that the properties will be renovated. I am looking for properties that will have increased property values in the future. Certain considerations will need to be taken into account for the recommendation, such as the spacing between these properties.

While other cities such as Dallas, Texas and Gary Indiana have been successful from the public with success, this would be on a much larger scale. <https://www.fastcompany.com/90618596/the-radical-way-cities-are-building-affordable-housing> (<https://www.fastcompany.com/90618596/the-radical-way-cities-are-building-affordable-housing>)

(CITATION FOR OVERVIEW ON SECTION 8 HOUSING:

[https://www.hud.gov/program\\_offices/housing/mfh/rfp/s8bkinfo](https://www.hud.gov/program_offices/housing/mfh/rfp/s8bkinfo) ([https://www.hud.gov/program\\_offices/housing/mfh/rfp/s8bkinfo](https://www.hud.gov/program_offices/housing/mfh/rfp/s8bkinfo))

FURTHER READING: On mixed income housing in NYC, where

<https://case.edu/socialwork/nimc/sites/case.edu.nimc/files/2020-05/Schwartz%20NewYork%202020.pdf> (<https://case.edu/socialwork/nimc/sites/case.edu.nimc/files/2020-05/Schwartz%20NewYork%202020.pdf>)

▼ Why should low-income housing be interspersed throughout neighborhoods? (CITATION)

"Proponents of mixed-income housing see it as a tool to address what has been termed the *culture of poverty*. This phrase describes the physical concentration of poor households in multifamily projects and the problems for the residents, including joblessness, drug abuse, ar-

## Contents

6.2 Visualizing the Comparison
▼ 6.3 Home History
6.3.1 Age of Home
6.3.2 Year Modified
▼ 7 Distance Data
▼ 7.1 Getting Infrastructure Data
7.1.1 Downtown
7.1.2 Hospitals
7.1.3 Police Stations
7.1.4 Schools
7.2 Finding Distance for Infrastructure
▼ 7.3 Exploring Distance Data
7.3.1 Downtown
7.3.2 Hospitals
7.3.3 Police Stations
7.3.4 Schools
▼ 8 Preparing for Modeling
8.0.1 Dropping View and Waterfront Properties from the Model
9 Zipcode
10 Seeing the Correlation
▼ 11 Building a model
11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2 Second Model: Transforming the Price with Logarithms
▼ 12 Improving the Model with Adding Features
12.1 Third Model: Bedrooms, Floors, Closest School
12.2 Fourth Model: Square Foot Basement, Closest Hospital
12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4 Sixth Model: Condition, Waterfront and View
12.5 Seventh Model: Adding Zipcodes
12.6 Eighth Model: Latitude and Closest Hospital Distance
13 Conclusion
▼ 14 Recommendations:
14.0.1 Visualizing Schools

According to Brophy and Smith 6 Cityscape this theory, a mixture of lower income and higher income households will reduce the social pathology caused by concentration. Anticipated benefits of mixed-income housing include the following:

- The behavior patterns of some lower income residents will be similar to those of their higher income neighbors. The quality of the living environment and the housing quality alone, leads to upward mobility.
- Nonworking low-income tenants will find their way into the work force numbers because of the social norms of their new environment (they work/school every day) and the informal networking with employees.
- The crime rate will fall because the higher income households have a more positive attitude and better enforced set of ground rules for the community.
- Low-income households will have the benefit of better schools, enhanced safety, and enhanced safety, enabling them to move themselves and their children to better locations based on their current economic condition.

Many of these anticipated results are subtle and difficult to measure quantitatively. One of the most important quantifiable result as the effect of mixed incomes on the project's analysis of the effects of mixed-income housing on the behavior of residents. It is also important to consider the reasons behind these intended benefits. First, if low-income households are subsidized to live in developments that are in locations with good schools, access to jobs, and access to transportation, there is some likelihood that the beneficiaries will move to these locations and result in the anticipated outcomes described above.<sup>7</sup>

<https://www.huduser.gov/periodicals/cityscape/vol3num2/success.pdf>  
<https://www.huduser.gov/periodicals/cityscape/vol3num2/success.pdf>

## 2.2 The Stakeholder's Interests:

The city of Seattle has tasked me with ensuring their making good on their promise to provide a safe and healthy environment for all residents with an assured outcome. Since this project is using public funds, it is important to document the process and results of the analysis and the model, furthermore, I will be透明地 sharing the code and results with the public.

## Contents ⚙️

6.2 Visualizing the Comparison
6.3 Home History
6.3.1 Age of Home
6.3.2 Year Modified
7 Distance Data
7.1 Getting Infrastructure Data
7.1.1 Downtown
7.1.2 Hospitals
7.1.3 Police Stations
7.1.4 Schools
7.2 Finding Distance for Infrastructure
7.3 Exploring Distance Data
7.3.1 Downtown
7.3.2 Hospitals
7.3.3 Police Stations
7.3.4 Schools
8 Preparing for Modeling
8.0.1 Dropping View and Waterfront Properties for ML
9 Zipcode
10 Seeing the Correlation
11 Building a model
11.1 Baseline Model: Grade, Bathrooms, Lat, Long
11.2 Second Model: Transforming the Price with Log
12 Improving the Model with Adding Features
12.1 Third Model: Bedrooms, Floors, Closest School
12.2 Fourth Model: Square Foot Basement, Closest
12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
12.4 Sixth Model: Condition, Waterfront and View
12.5 Seventh Model: Adding Zipcodes
12.6 Eighth Model: Latitude and Closest Hospital Distance
13 Conclusion
14 Recommendations:
14.0.1 Visualizing Schools

understandable to civic workers.

The city is also especially interested in seeing the range of housing prices. Often in statistics, we focus on the median or mean, however for this project, it is important to be collecting data on the size of the ranges.

## 2.3 Why Use Machine Learning?

It is colloquial knowledge that there is class disparity in Seattle. However, the gap is between neighborhoods specifically, home by home, property by property. I've been tasked to research. Seattle has decided that the best approach is to use machine learning to combat this disparity. This is backed by research, science and evidence.

Using machine learning, we can rely on the data itself to ensure that we have evidence-backed solutions to this problem.

Machine Learning is being implemented here to reduce the political bias in the solution. However, ML can only be as free of bias as the implementation allows.

### 2.3.1 Why I Was Hired to Oversee the Project

In order to counter the mechanisms were ML can propagate racism, Seattle chose me specifically as a Data Scientist. (CITATION: <https://www.nytimes.com/2021/03/15/technology/artificial-intelligence-bias.html> (<https://www.nytimes.com/2021/03/15/technology/artificial-intelligence-bias.html>)) Seattle chose me specifically as a Data Scientist because I have experience working with homeless folks, at a housing-forward non-profit.

## 3 Loading and Inspecting the Data

## Contents ⚙️

6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Zipcode
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Lat
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest
12.3	Fifth Model: Square Foot Lot of Closest 15 Nei
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital D
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

```
In [1]: 1 # importing the libraries to use, the basics, the libraries
2 # geopy for distance data collection and selenium for scrap-
3
4 import pandas as pd
5 import matplotlib.pyplot as plt
6 import seaborn as sns
7 import numpy as np
8
9 import statsmodels.api as sm
10 from statsmodels.stats.outliers_influence import variance_infl
11 import scipy.stats as stats
12 from sklearn.preprocessing import StandardScaler
13
14 ## geopy must be installed first
15 from geopy import distance
16
17 ## for scraping distances
18 from selenium import webdriver
19 from webdriver_manager.chrome import ChromeDriverManager
20 from selenium.webdriver.common.by import By
21 from selenium.webdriver.common.keys import Keys
22
23 import requests, json
24
25 import folium
26
27 %matplotlib inline
28
29 palette = sns.color_palette("colorblind")
30 sns.set_style('white')
```

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

```
In [2]: 1 # loading the main CSV
2
3 data = pd.read_csv('Data/kc_house_data.csv')
```

```
In [3]: 1 # inspecting the data
2
3 data.head()
```

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot
0	7129300520	10/13/2014	221900.0	3	1.00	1180	5650
1	6414100192	12/9/2014	538000.0	3	2.25	2570	7242
2	5631500400	2/25/2015	180000.0	2	1.00	770	10000
3	2487200875	12/9/2014	604000.0	4	3.00	1960	5000
4	1954400510	2/18/2015	510000.0	3	2.00	1680	8080
5 rows × 21 columns							

## Contents ⚙️

6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
▼ 7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
▼ 7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Now
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest School
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighbors
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

In [4]:

```
1 data.info()
2
3 # taking a look at the raw data
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21597 entries, 0 to 21596
Data columns (total 21 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   id               21597 non-null   int64  
 1   date              21597 non-null   object  
 2   price              21597 non-null   float64 
 3   bedrooms           21597 non-null   int64  
 4   bathrooms           21597 non-null   float64 
 5   sqft_living         21597 non-null   int64  
 6   sqft_lot             21597 non-null   int64  
 7   floors              21597 non-null   float64 
 8   waterfront          19221 non-null   object  
 9   view                21534 non-null   object  
 10  condition            21597 non-null   object  
 11  grade                21597 non-null   object  
 12  sqft_above            21597 non-null   int64  
 13  sqft_basement         21597 non-null   object  
 14  yr_built              21597 non-null   int64  
 15  yr_renovated          17755 non-null   float64 
 16  zipcode              21597 non-null   int64  
 17  lat                  21597 non-null   float64 
 18  long                 21597 non-null   float64 
 19  sqft_living15          21597 non-null   int64  
 20  sqft_lot15              21597 non-null   int64  
dtypes: float64(6), int64(9), object(6)
memory usage: 3.5+ MB
```

Of note: waterfront, and year renovated have significant data missing. Both of these columns have missing only a bit of data and those null values can probably be filled in. This makes the most sense given the context.

## Contents ⚙

- 6.2 Visualizing the Comparison
- ▼ 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- ▼ 7 Distance Data
  - ▼ 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - ▼ 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- ▼ 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- ▼ 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- ▼ 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- ▼ 14 Recommendations:
  - 14.0.1 Visualizing Schools

```
In [5]: 1 data.isna().sum()
```

id	0
date	0
price	0
bedrooms	0
bathrooms	0
sqft_living	0
sqft_lot	0
floors	0
waterfront	2376
view	63
condition	0
grade	0
sqft_above	0
sqft_basement	0
yr_built	0
yr_renovated	3842
zipcode	0
lat	0
long	0
sqft_living15	0
sqft_lot15	0
	dtype: int64

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [6]: 1 data.describe()

	<b>id</b>	<b>price</b>	<b>bedrooms</b>	<b>bathrooms</b>	<b>sqft_living</b>	
<b>count</b>	2.159700e+04	2.159700e+04	21597.000000	21597.000000	21597.000000	2.159700e+04
<b>mean</b>	4.580474e+09	5.402966e+05	3.373200	2.115826	2080.321850	1.50
<b>std</b>	2.876736e+09	3.673681e+05	0.926299	0.768984	918.106125	4.14
<b>min</b>	1.000102e+06	7.800000e+04	1.000000	0.500000	370.000000	5.20
<b>25%</b>	2.123049e+09	3.220000e+05	3.000000	1.750000	1430.000000	5.04
<b>50%</b>	3.904930e+09	4.500000e+05	3.000000	2.250000	1910.000000	7.618
<b>75%</b>	7.308900e+09	6.450000e+05	4.000000	2.500000	2550.000000	1.06
<b>max</b>	9.900000e+09	7.700000e+06	33.000000	8.000000	13540.000000	1.65

Of note--the average house has 3 bedrooms, 2 bathrooms, which we are already looking to make public. The average house was last sold 5 years ago, which is older than the standard property is old, which information on renovations must be explored in greater detail.

In [7]: 1 data.columns

2

3 # looking to see what columns are in the dataset

```
Index(['id', 'date', 'price', 'bedrooms', 'bathrooms', 'sqft_living',
       'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade',
       'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated', 'zipcode',
       'lat', 'long', 'sqft_living15', 'sqft_lot15'],
      dtype='object')
```

## Contents ⚙

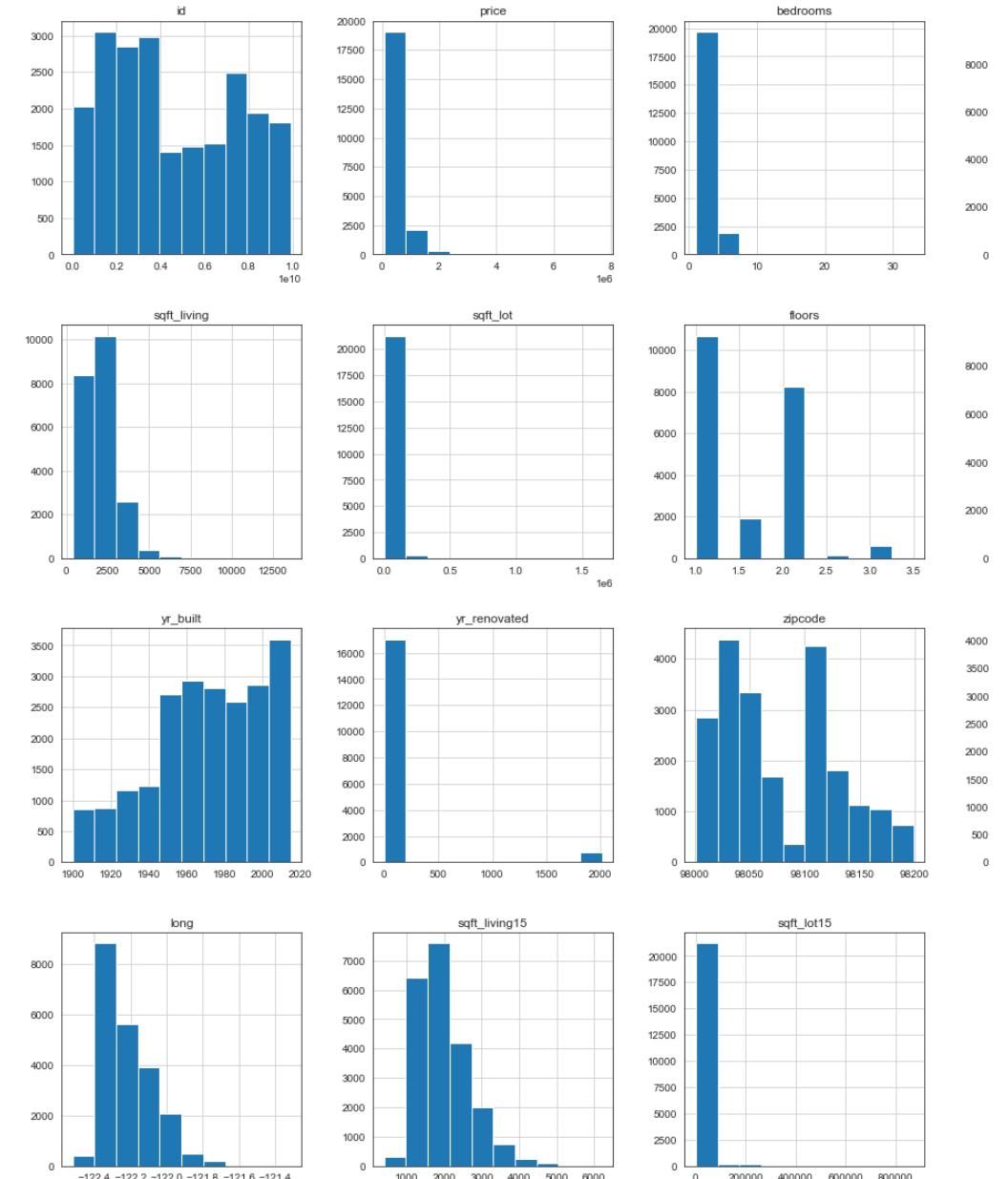
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [8]:

```
1 sns.set_style('white')
2
3 data.hist(figsize=(20,20));
4
5 # let's look at the distributions of some of these columns
```

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, and Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools



## 4 Cleaning and Preparing the Data

## Contents ⚙️

Comparing Renovated Homes with Unrenovated

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
- 7.3 Exploring Distance Data
  - 7.3.1 Downtown
  - 7.3.2 Hospitals
  - 7.3.3 Police Stations
  - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, and Longitude
    - 11.2 Second Model: Transforming the Price with Logarithms
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest Police Station
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

## 4.0.1 Duplicates

```
In [9]: 1 data['id'].value_counts().head()
2 # let's take a look to see if there are duplicates
```

795000620	3
1825069031	2
2019200220	2
7129304540	2
1781500435	2

Name: id, dtype: int64

Some of these properties are listed three times, that will throw off our model. Let's remove the duplicates.

```
In [10]: 1 data.drop_duplicates(subset='id', inplace=True)
2
3 # this will get rid of duplicates
```

## 4.0.2 Price

## Contents ⚙️

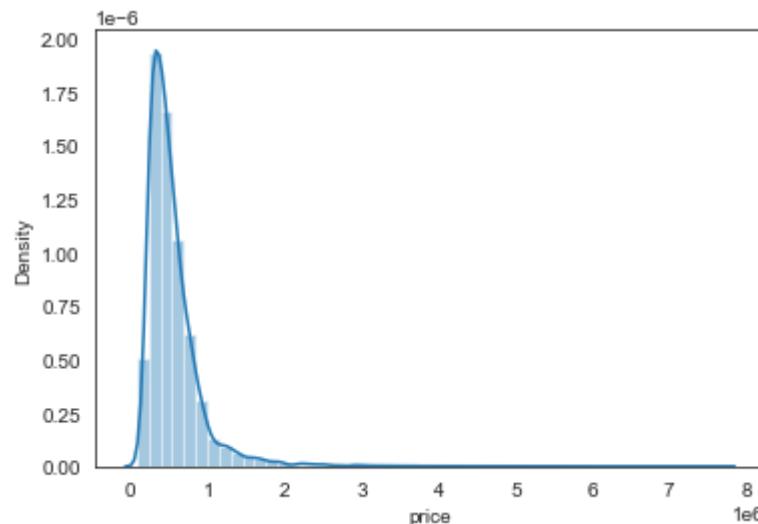
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

In [11]:

```
1 sns.distplot( a=data["price"]);
2
3 # this will show a histogram for the price distribution
```

/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/yaml/2551: FutureWarning: `distplot` is a deprecated function and will be removed. Please adapt your code to use either `displot` (a figure-level function) or `histplot` (an axes-level function for histograms).

```
warnings.warn(msg, FutureWarning)
```



Observations: The distribution of price in this dataset has a very long tail, with many inexpensive houses, and the data is leptokurtic (taller and skinnier) than a normal distribution.

Since this project is specifically about buying a lot of properties, I will focus on properties that are priced at 2.5 million and below, which is where most of the data is. This will prevent all the calculations from being dragged into the extreme price brackets for the city, and for the tenants of the new housing developments.

## Contents ⚙️

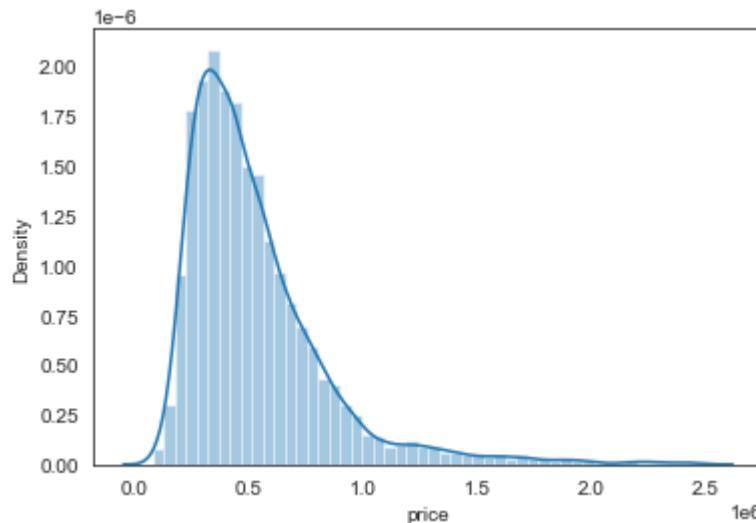
6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Now
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest School
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

```
In [12]: 1 data = data.loc[data['price'] < 2500000.0]
2
3 # this will get rid of all properties in the data set above
```

```
In [13]: 1 sns.distplot( a=data["price"])
2
3 # this will show what the distribution of prices looks like
4 # that remains
```

/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/  
y:2551: FutureWarning: `distplot` is a deprecated function and will be rem  
n. Please adapt your code to use either `displot` (a figure-level function  
y) or `histplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)

<AxesSubplot:xlabel='price', ylabel='Density'>



## 4.0.3 Sqft\_living

## Contents ⚙️

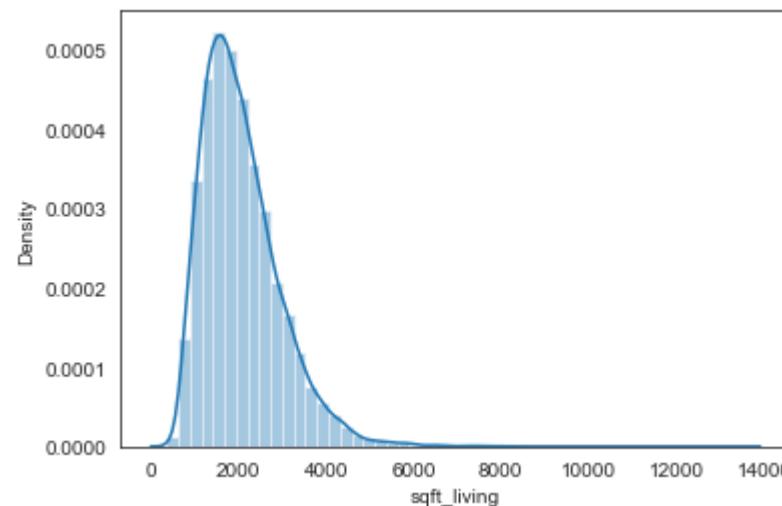
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

Square footage of living space in the home-from the data.

In [14]:

```
1 sns.distplot( a=data["sqft_living"]);
2
3 # Taking a look at the distribution of this variable
```

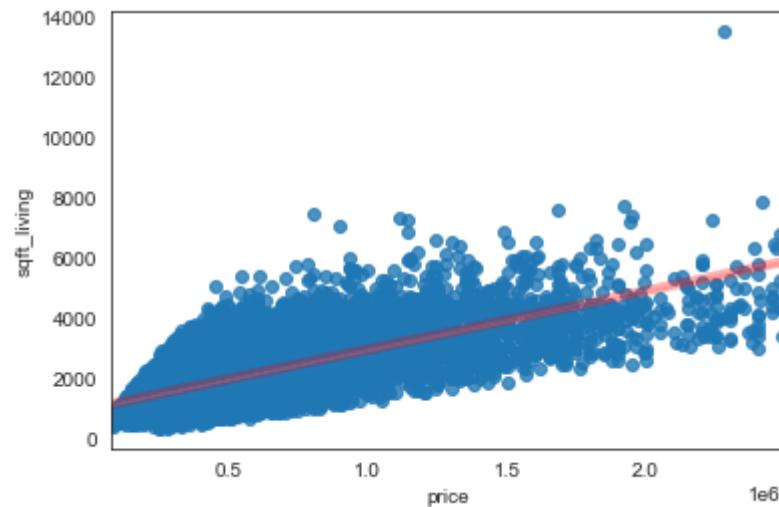
/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/  
y:2551: FutureWarning: `distplot` is a deprecated function and will be rem  
n. Please adapt your code to use either `displot` (a figure-level function  
y) or `histplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)



## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

```
In [15]: 1 sns.regplot(data = data, x = 'price', y = 'sqft_living',
2                         line_kws={"color":"r","alpha":0.3,"lw":5})
3 plt.show()
4
5 # Visualizing how sqft_living affects price
```



I'm going to get rid of the outliers in the data set, the homes we're interested in are not public housing will not be these outliers.

```
In [16]: 1 data = data.loc[data['sqft_living'] < 8000]
```

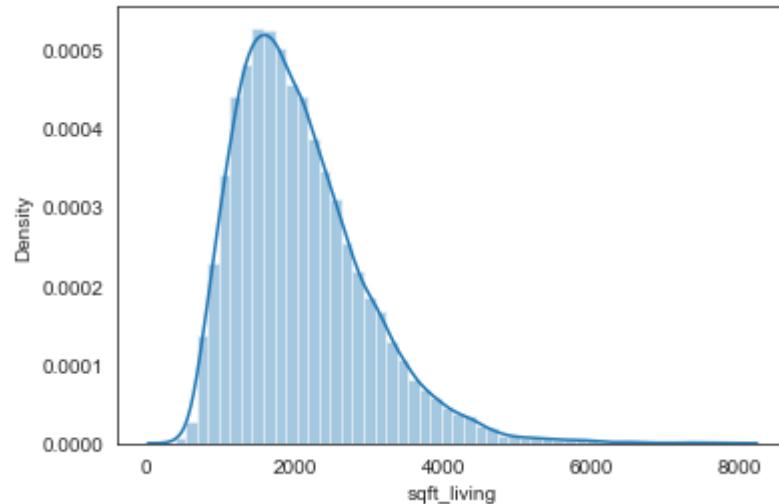
## Contents ⚙️

- .. Comparing Renovated Homes with Unrenovated
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [17]:

```
1 sns.distplot( a=data["sqft_living"]);
```

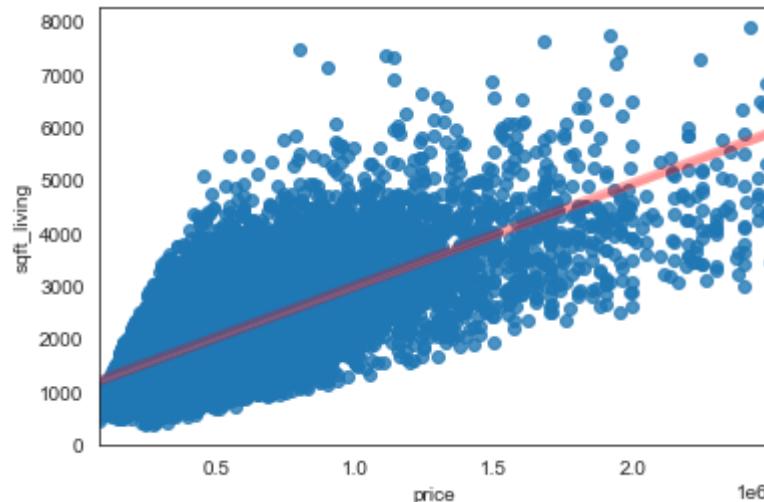
```
/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/
y:2551: FutureWarning: `distplot` is a deprecated function and will be remo
n. Please adapt your code to use either `displot` (a figure-level function
y) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
```



## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Logarithm
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

```
In [18]: 1 sns.regplot(data = data, x = 'price', y = 'sqft_living',
2                         line_kws={"color":"r","alpha":0.3,"lw":5})
3 plt.show()
```



## 4.0.4 Sqft\_above

Square footage of house apart from basement-from the data.

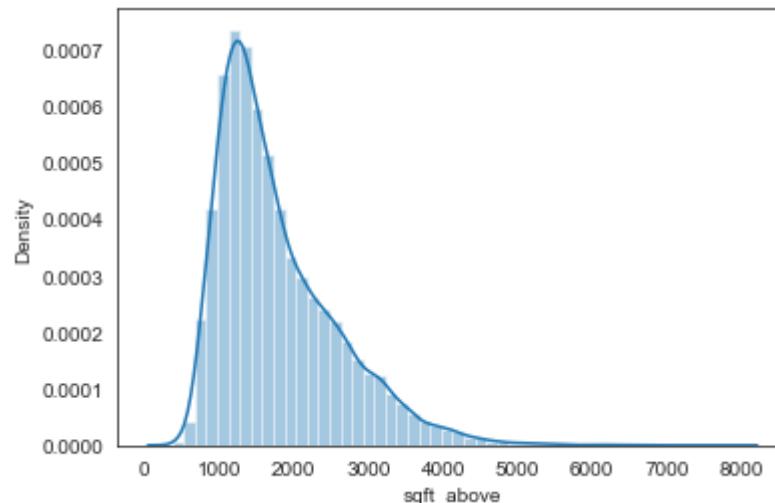
## Contents ⚙️

- .. Comparing Renovated Homes with Unrenovated
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [19]:

```
1 sns.distplot( a=data["sqft_above"]);
```

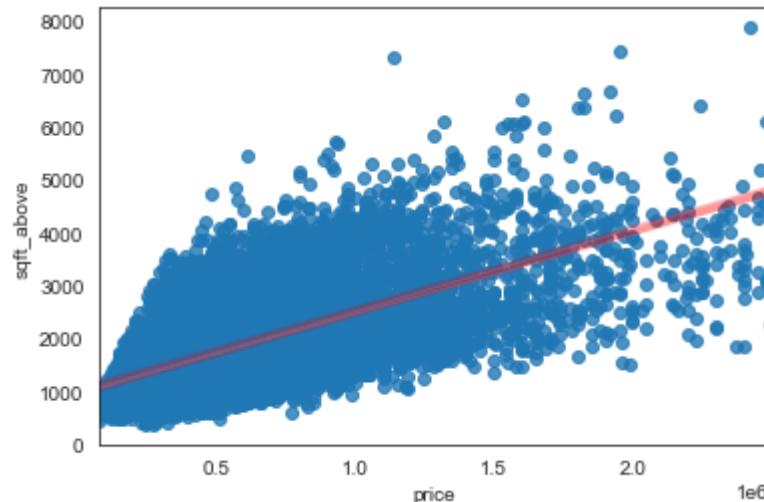
```
/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/
y:2551: FutureWarning: `distplot` is a deprecated function and will be remo-
n. Please adapt your code to use either `displot` (a figure-level function)
y) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
```



## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

```
In [20]: 1 sns.regplot(data = data, x = 'price', y = 'sqft_above',  
2                      line_kws={"color":"r","alpha":0.3,"lw":5})  
3 plt.show()
```



## 4.0.5 Sqft\_lot

Square footage of the lot-from the data.

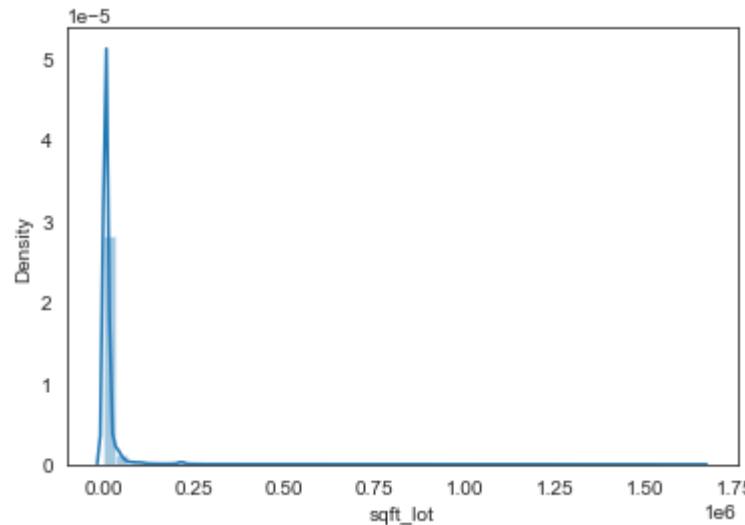
## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [21]:

```
1 sns.distplot( a=data[ "sqft_lot" ]);
```

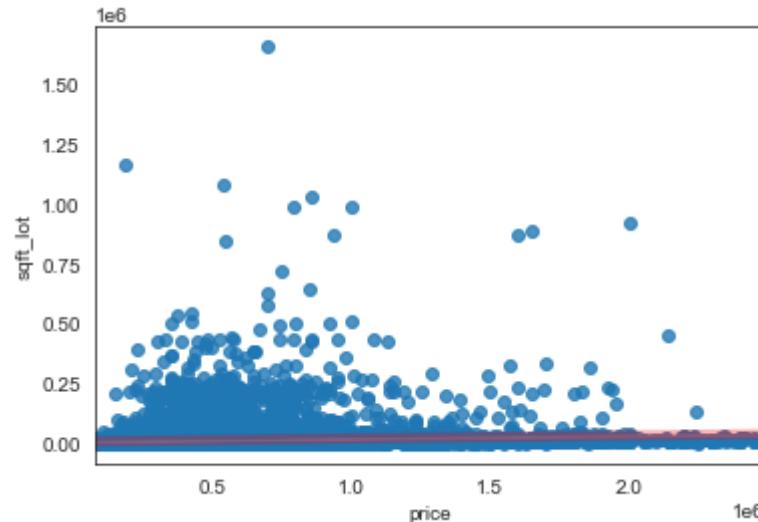
```
/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/
y:2551: FutureWarning: `distplot` is a deprecated function and will be remo
n. Please adapt your code to use either `displot` (a figure-level function
y) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
```



## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

```
In [22]: 1 sns.regplot(data = data, x = 'price', y = 'sqft_lot',  
2                      line_kws={"color":"r","alpha":0.3,"lw":5})  
3 plt.show()
```



## Contents ⚙️

- 6.2 Visualizing the Comparison
- ▼ 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- ▼ 7 Distance Data
  - ▼ 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - ▼ 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- ▼ 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - ▼ 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - ▼ 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - ▼ 14 Recommendations:
    - 14.0.1 Visualizing Schools

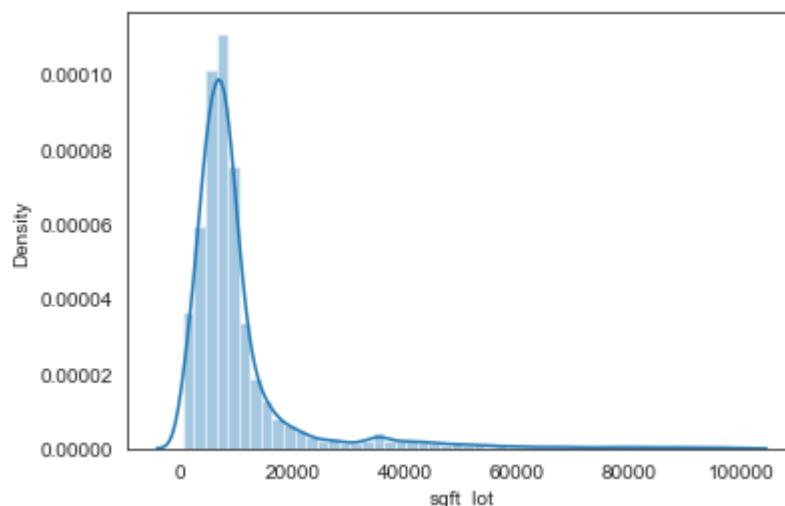
```
In [23]: 1 data['sqft_lot'].describe()
```

count	2.131700e+04
mean	1.508035e+04
std	4.156528e+04
min	5.200000e+02
25%	5.030000e+03
50%	7.590000e+03
75%	1.060000e+04
max	1.651359e+06
Name:	sqft_lot, dtype: float64

```
In [24]: 1 data = data.loc[data['sqft_lot'] < 100000]
```

```
In [25]: 1 sns.distplot( a=data["sqft_lot"]);
```

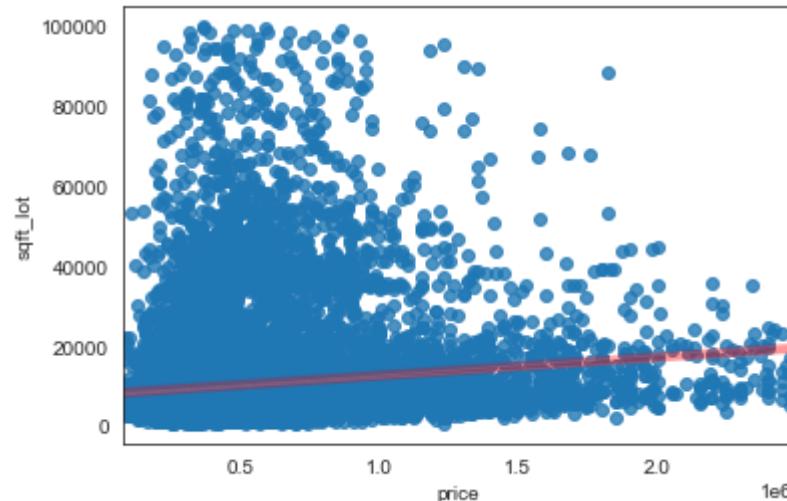
/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/sns/\_decorators.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed. Please adapt your code to use either `displot` (a figure-level function) or `histplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)



## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Logarithmic Scale
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

```
In [26]: 1 sns.regplot(data = data, x = 'price', y = 'sqft_lot',
2                         line_kws={"color":"r","alpha":0.3,"lw":5})
3 plt.show()
```



## 4.0.6 Sqft\_basement

```
In [27]: 1 data['sqft_basement'].value_counts().head()
```

Sqft_Basement	Count
0.0	12374
?	437
600.0	214
700.0	204
500.0	204

Name: sqft\_basement, dtype: int64

## Contents ⚙️

- 6.2 Visualizing the Comparison
- ▼ 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- ▼ 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - ▼ 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- ▼ 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - ▼ 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
- ▼ 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- ▼ 14 Recommendations:
  - 14.0.1 Visualizing Schools

```
In [28]: 1 data['sqft_basement'].describe()

count      20855
unique     288
top        0.0
freq      12374
Name: sqft_basement, dtype: object
```

```
In [29]: 1 data.loc[data['sqft_basement'] == '?', 'sqft_basement'] = 0.1
2
3 #? doesn't make sense as a value. I will replace it with the mean.
```

## Contents ⚙️

6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Modeling
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest School
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighbors
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

In [30]:

1

2 data.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 20855 entries, 0 to 21596
Data columns (total 21 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   id               20855 non-null   int64  
 1   date              20855 non-null   object  
 2   price              20855 non-null   float64 
 3   bedrooms            20855 non-null   int64  
 4   bathrooms            20855 non-null   float64 
 5   sqft_living          20855 non-null   int64  
 6   sqft_lot              20855 non-null   int64  
 7   floors              20855 non-null   float64 
 8   waterfront            18559 non-null   object  
 9   view                  20795 non-null   object  
 10  condition             20855 non-null   object  
 11  grade                  20855 non-null   object  
 12  sqft_above             20855 non-null   int64  
 13  sqft_basement          20855 non-null   object  
 14  yr_built              20855 non-null   int64  
 15  yr_renovated           17129 non-null   float64 
 16  zipcode                20855 non-null   int64  
 17  lat                     20855 non-null   float64 
 18  long                    20855 non-null   float64 
 19  sqft_living15          20855 non-null   int64  
 20  sqft_lot15              20855 non-null   int64  
dtypes: float64(6), int64(9), object(6)
memory usage: 3.5+ MB
```

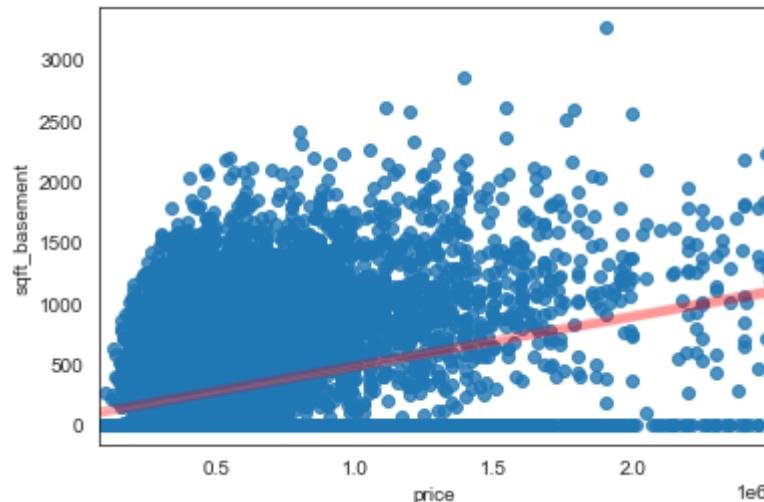
In [31]:

1 data['sqft\_basement'] = data['sqft\_basement'].astype(float)

## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

```
In [32]: 1 sns.regplot(data = data, x = 'price', y = 'sqft_basement',  
2                      line_kws={"color":"r","alpha":0.3,"lw":5})  
3 plt.show()
```

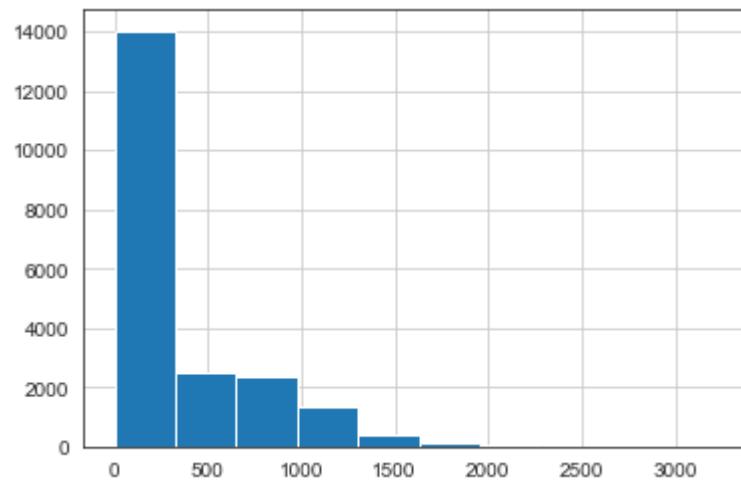


## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

In [33]: 1 data['sqft\_basement'].hist()

<AxesSubplot:>



I'm going to turn this into a binary variable: 0 if the property has no basement (sqft\_basement at 0,) and 1 if the property has a basement. There are many more highly priced homes without a basement, and over 50% of homes don't have a basement.

In [34]: 1 data.loc[data['sqft\_basement'] != 0, 'sqft\_basement'] = 1

In [35]: 1 data['sqft\_basement'].value\_counts().head()

```
0.0    12811
1.0     8044
Name: sqft_basement, dtype: int64
```

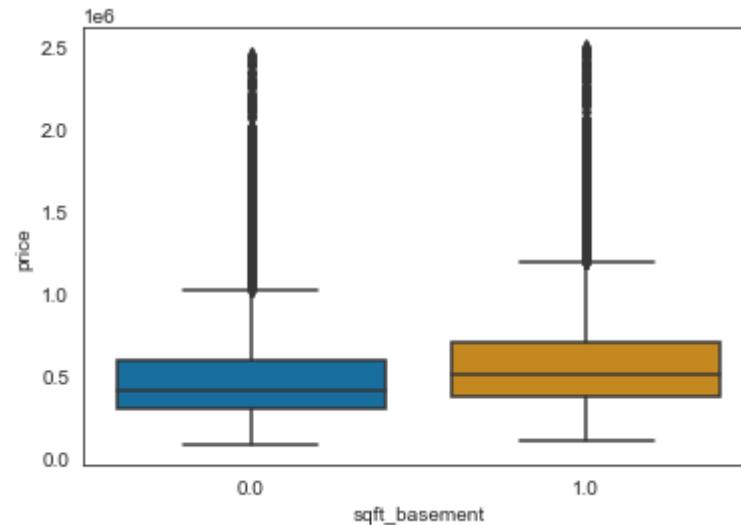
## Contents ⚙️

6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
▼ 7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
▼ 7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Modeling
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest School
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighbors
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

In [36]:

```
1 sns.boxplot(data = data, x="sqft_basement", y="price", palette="pale
```

<AxesSubplot:xlabel='sqft\_basement', ylabel='price'>



Observations: the minimum quartile price, average price and maximum price are all higher in properties with basements, as to be expected.



## Square Foot Living 15 and Square Lot 15

It is noteworthy to have the square footage of the living space and the square footage of the lot be 15. These following two metrics are especially interesting. Traditional public housing, in big blocky complexes, would have larger square footages than in mixed income public housing or private housing.

### FURTHER WORK:

None of the properties in this data set are public housing properties. To compare how the relationships to the closest 15 neighboring properties differ between both public and non-public housing, I'd need to collect more data.

## 4.0.7 Sqft\_living15

### Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest Neighborhood
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

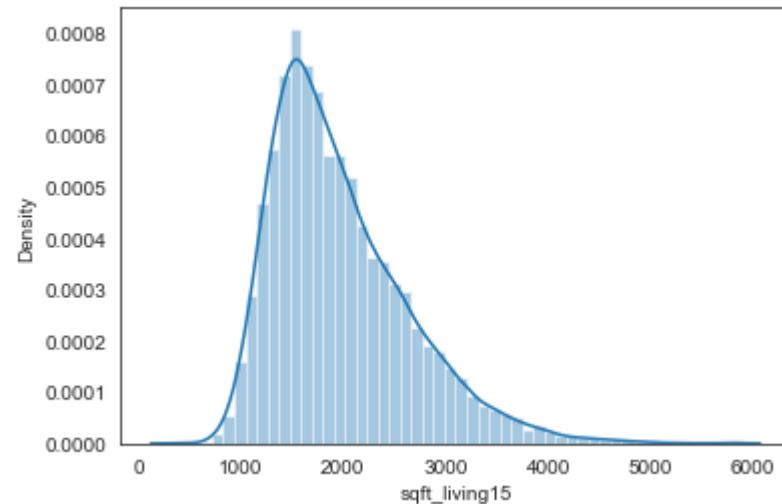
The square footage of interior housing living space for the nearest 15 neighborhoods.

In [37]:

```
1 sns.distplot( a=data["sqft_living15"]);
```

/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/sns/\_decorators.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function) or `histplot` (an axes-level function for histograms).

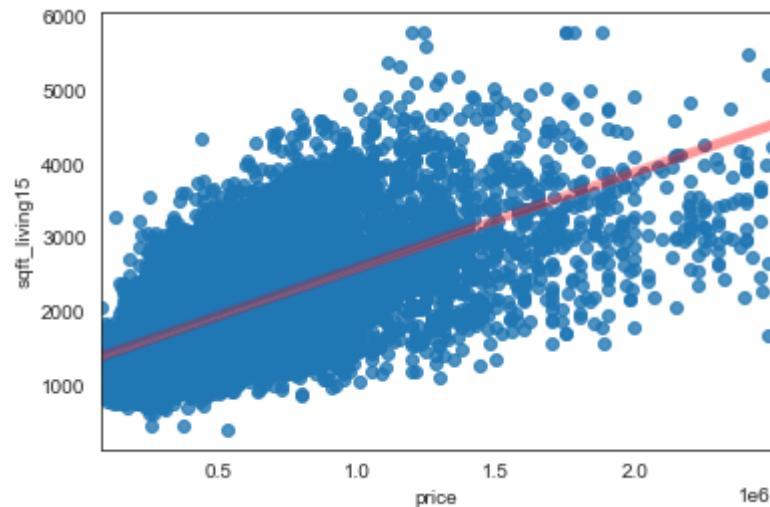
```
warnings.warn(msg, FutureWarning)
```



## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

```
In [38]: 1 sns.regplot(data = data, x = 'price', y = 'sqft_living15',  
2                      line_kws={"color":"r","alpha":0.3,"lw":5})  
3 plt.show()
```

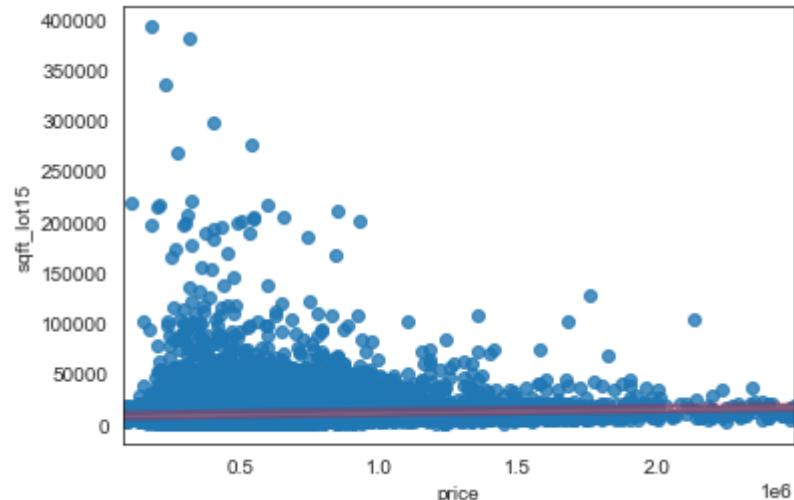


## 4.0.8 Sqft\_lot15

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Logarithmic Scale
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

```
In [39]: 1 sns.regplot(data = data, x = 'price', y = 'sqft_lot15',
2                      line_kws={"color":"r","alpha":0.3,"lw":5})
3 plt.show()
```



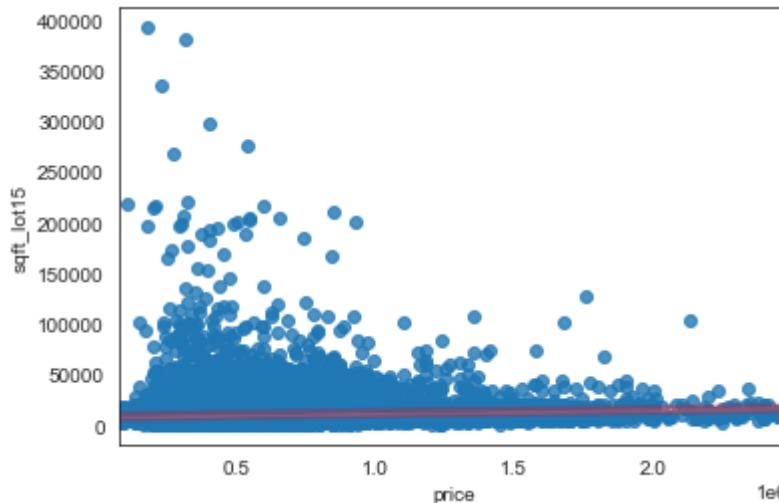
I'm going to get rid of the upper valued outliers here, which would be where the 15 closest neighbors have a *lot* of square footage. The neighborhoods will throw off the data. I want to be conservative.

## Contents ⚙️

- [6.2 Visualizing the Comparison](#)
- ▼ [6.3 Home History](#)
  - [6.3.1 Age of Home](#)
  - [6.3.2 Year Modified](#)
- ▼ [7 Distance Data](#)
  - ▼ [7.1 Getting Infrastructure Data](#)
    - [7.1.1 Downtown](#)
    - [7.1.2 Hospitals](#)
    - [7.1.3 Police Stations](#)
    - [7.1.4 Schools](#)
  - [7.2 Finding Distance for Infrastructure](#)
  - ▼ [7.3 Exploring Distance Data](#)
    - [7.3.1 Downtown](#)
    - [7.3.2 Hospitals](#)
    - [7.3.3 Police Stations](#)
    - [7.3.4 Schools](#)
- ▼ [8 Preparing for Modeling](#)
  - [8.0.1 Dropping View and Waterfront Properties for Modeling](#)
  - [9 Zipcode](#)
  - [10 Seeing the Correlation](#)
  - ▼ [11 Building a model](#)
    - [11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude](#)
    - [11.2 Second Model: Transforming the Price with Log](#)
- ▼ [12 Improving the Model with Adding Features](#)
  - [12.1 Third Model: Bedrooms, Floors, Closest School](#)
  - [12.2 Fourth Model: Square Foot Basement, Closest Hospital](#)
  - [12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods](#)
  - [12.4 Sixth Model: Condition, Waterfront and View](#)
  - [12.5 Seventh Model: Adding Zipcodes](#)
  - [12.6 Eighth Model: Latitude and Closest Hospital Distance](#)
- [13 Conclusion](#)
- ▼ [14 Recommendations:](#)
  - [14.0.1 Visualizing Schools](#)

```
In [40]: 1 data = data[data['sqft_lot15'] < 400000]
```

```
In [41]: 1 sns.regplot(data = data, x = 'price', y = 'sqft_lot15',
2                      line_kws={"color":"r","alpha":0.3,"lw":5})
3 plt.show()
```



## 4.0.9 Waterfront

```
In [42]: 1 data['waterfront'].value_counts()
```

waterfront	count
NO	18447
YES	112

Name: waterfront, dtype: int64

```
In [43]: 1 data['waterfront'].isna().sum()
```

2296

## Contents ⚙️

6.2 Visualizing the Comparison
▼ 6.3 Home History
6.3.1 Age of Home
6.3.2 Year Modified
▼ 7 Distance Data
▼ 7.1 Getting Infrastructure Data
7.1.1 Downtown
7.1.2 Hospitals
7.1.3 Police Stations
7.1.4 Schools
7.2 Finding Distance for Infrastructure
▼ 7.3 Exploring Distance Data
7.3.1 Downtown
7.3.2 Hospitals
7.3.3 Police Stations
7.3.4 Schools
▼ 8 Preparing for Modeling
8.0.1 Dropping View and Waterfront Properties for Now
9 Zipcode
10 Seeing the Correlation
▼ 11 Building a model
11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2 Second Model: Transforming the Price with Log
▼ 12 Improving the Model with Adding Features
12.1 Third Model: Bedrooms, Floors, Closest School
12.2 Fourth Model: Square Foot Basement, Closest School
12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4 Sixth Model: Condition, Waterfront and View
12.5 Seventh Model: Adding Zipcodes
12.6 Eighth Model: Latitude and Closest Hospital Distance
13 Conclusion
▼ 14 Recommendations:
14.0.1 Visualizing Schools

```
In [44]: 1 data['waterfront'].fillna(value = "NO", inplace = True)
```

```
In [45]: 1 data['waterfront'].replace(['NO', 'YES'],
2                                     [0, 1], inplace=True)
3 #this will make the value numeric and therefore able to be used by the model
```

Because the average value is "no", I'm choosing to fill null values with "no". A waterfront view would be a boon to the property, the risk involved in representing a property with a waterfront to have no waterfront, a selected for public housing has a surprise positive effect for the folks who live there. By contrast, if I filled these values with "yes" and they didn't have a waterfront, this would falsely inflate the assigned value of that property, and were selected because of that false inflated value, that would affect folks who lived there.

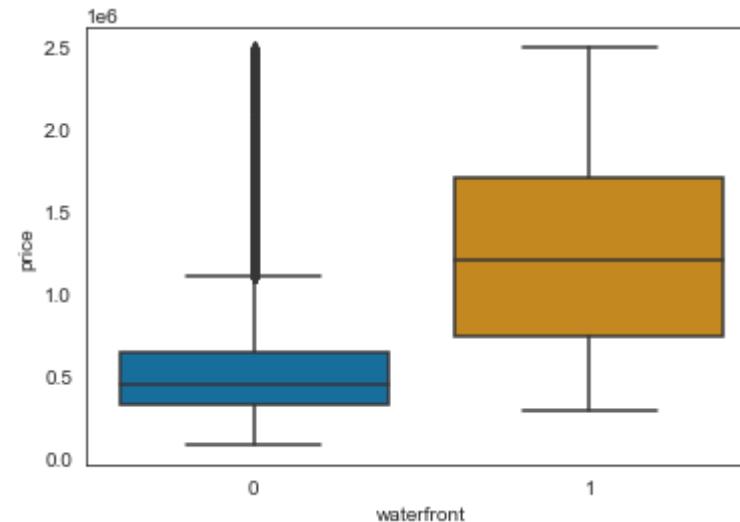
## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Lat, Long
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

In [46]:

```
1 sns.boxplot(data = data, x = 'waterfront', y = 'price', palette='magma')
```

&lt;AxesSubplot:xlabel='waterfront', ylabel='price'&gt;



This is a clear visualization that represents:

1) homes with a waterfront have a much wider spread for their average price. 2) the upper limits of how pricey those homes can be. But in general, homes with a waterfront are more expensive. 2) there are plenty outlying examples of homes with a waterfront that are more expensive.



## 4.0.10 Bedrooms

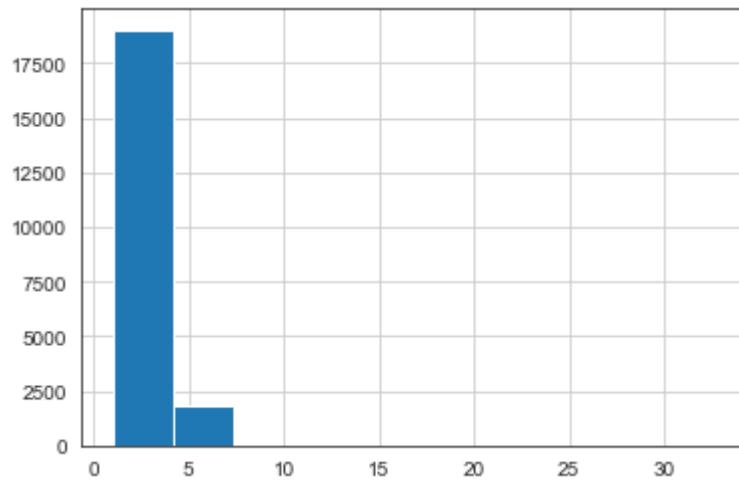
## Contents ⚙

- 6.2 Visualizing the Comparison
- ▼ 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- ▼ 7 Distance Data
  - ▼ 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - ▼ 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- ▼ 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- ▼ 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- ▼ 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- ▼ 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [47]:

```
1 data['bedrooms'].hist()  
2 # taking a look at the distribution of bedrooms in this dataset
```

&lt;AxesSubplot:&gt;



## Contents ⚙

- 6.2 Visualizing the Comparison
- ▼ 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- ▼ 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- ▼ 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- ▼ 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- ▼ 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- ▼ 14 Recommendations:
  - 14.0.1 Visualizing Schools

```
In [48]: 1 data['bedrooms'].value_counts()
```

3	9512
4	6649
2	2685
5	1517
6	248
1	186
7	35
8	12
9	6
10	3
11	1
33	1

Name: bedrooms, dtype: int64

Since properties where there are more than 9 bedrooms may be public housing anyway, and because most of the data is outside the range of 1-9, we can drop these properties by dropping properties that have more than 9 bedrooms.

```
In [49]: 1 data = data.loc[data['bedrooms'] <= 9]
```

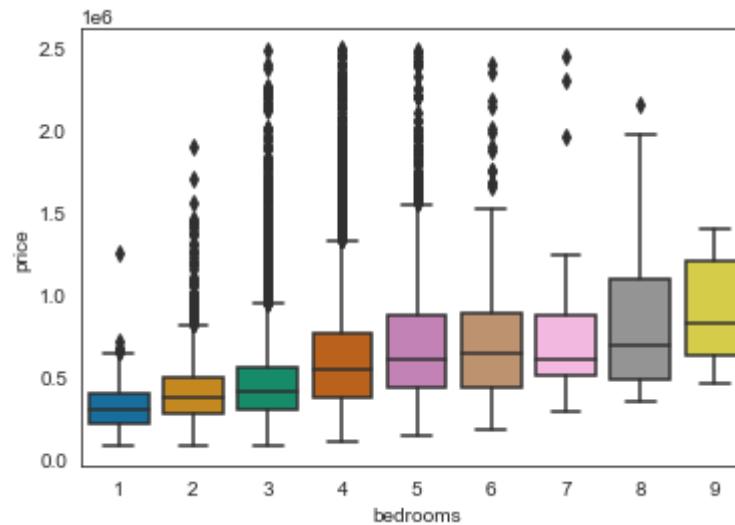
## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Logarithmic Scale
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [50]:

```
1 sns.boxplot(data = data, x="bedrooms", y="price", palette =
2 # we can see from the figure that as bedrooms increase, price
3 # increases, and also we can get a better sense for the distribution
4 # outliers by bedroom.
```

&lt;AxesSubplot:xlabel='bedrooms', ylabel='price'&gt;



This plot shows 1) that homes show a steady increase in minimum price as the number of bedrooms increase 2) that there are very many outliers in homes that have more than 7 bedrooms, but even with those outliers, there is also a steady increase in median price as the number of bedrooms increase as well.

## 5 Transforming Categorical Data

### 5.0.1 Condition

CITATION: <https://info.kingcounty.gov/assessor/esales/Glossary.aspx>

## Contents ⚙️

6.2	Visualizing the Comparison
6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties from the Model
9	Zipcode
10	Seeing the Correlation
11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest School
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
14	Recommendations:
14.0.1	Visualizing Schools

(<https://info.kingcounty.gov/assessor/esales/Glossary.aspx?type=r#Condition>)

Relative to age and grade. Coded 1-5.

1 = Poor- Worn out. Repair and overhaul needed on painted surfaces, plumbing, heating and numerous functional inadequacies. Excessive maintenance and abuse, limited value-in-use, approaching abandonment; reconstruction; reuse or change in occupancy is imminent. Effect of the scale regardless of the actual chronological age.

2 = Fair- Badly worn. Much repair needed. Many items need repair. Deferred maintenance obvious, inadequate building utility and system. Shorten the life expectancy and increasing the effective age.

3 = Average- Some evidence of deferred maintenance and normal wear and tear in that a few minor repairs are needed, along with some repair. Components still functional and contributing toward an extended life expectancy. Effective age and utility is standard for like properties of its class.

4 = Good- No obvious maintenance required but neither is everything perfect. Utility and system are above the standard and the overall effective age will be longer than typical property.

5 = Very Good- All items well maintained, many having been overhauled as they have shown signs of wear, increasing the life expectancy. Long effective age with little deterioration or obsolescence evident with no loss of utility.

Based on the description as given by the data set, this is a good model for transforming continuous data into categorical data. A house in "very good" condition could get into "good" condition in the future if the property doesn't receive proper maintenance. I think a house in "fair" condition could be renovated and made into a property with "good" condition.

Based on the description as given by the data set, this is a good model for transforming continuous data into categorical data. A house in "very good" condition could get into "good" condition in the future if the property doesn't receive proper maintenance. I think a house in "fair" condition could be renovated and made into a property with "good" condition.

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

```
In [51]: 1 data['condition'].replace(['Poor', 'Fair', 'Average', 'Good',
2                               [0, 1, 2, 3, 4], inplace=True)
3
4 # making these values continuous
```

---

```
In [52]: 1 data['condition'].value_counts()
2
3 # looking at the spread of the data
```

Condition	Count
2	13519
3	5502
4	1653
1	149
0	27

Name: condition, dtype: int64

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

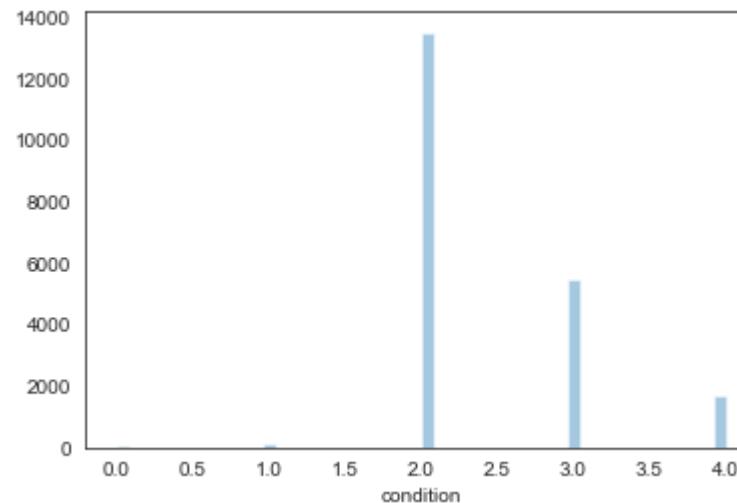
In [53]:

```
1 sns.distplot( a=data["condition"], hist=True, kde=False, rug=False)
2
3 # visualizing the spread of the data
```

/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/sns/\_decorators.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed. Please adapt your code to use either `displot` (a figure-level function) or `histplot` (an axes-level function for histograms).

```
warnings.warn(msg, FutureWarning)
```

&lt;AxesSubplot: xlabel='condition'&gt;



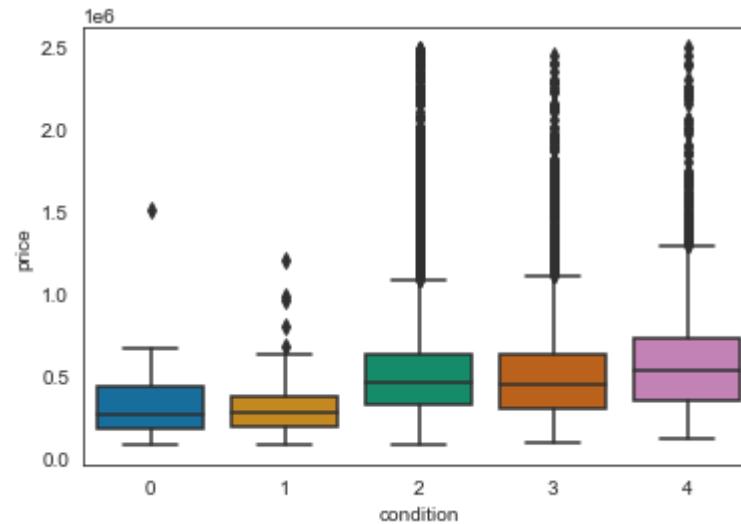
Observations: there is relatively no representation of homes with "fair" in this dataset. A majority of homes are in fair condition, with even less in "good" condition and even less in "very good" condition.

## Contents ⚙️

6.2	Visualizing the Comparison
6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Modeling
9	Zipcode
10	Seeing the Correlation
11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest Hospital
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
14	Recommendations:
14.0.1	Visualizing Schools

In [54]:

```
1 sns.boxplot(data = data, x = 'condition', y = 'price', palette='viridis')
2 plt.show();
```



Observations: The price of homes in poor condition and fair condition is lower than homes in better condition. The minimum price for homes in better condition increases in value, as well as the upper limit price for homes in better condition.



## 5.0.2 Grade

CITATION: <https://info.kingcounty.gov/assessor/esales/Glossary.aspx?type=r#Condition>  
<https://info.kingcounty.gov/assessor/esales/Glossary.aspx?type=r#Condition>

1-3 Falls short of minimum building standards. Normally cabin or mobile home.

4 Generally older, low quality construction. Does not meet code.

5 Low construction costs and workmanship. Small, simple design.

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

- 6 Lowest grade currently meeting building code. Low quality materials and designs.
- 7 Average grade of construction and design. Commonly seen in lower divisions.
- 8 Just above average in construction and design. Usually better interior and exterior finish work.
- 9 Better architectural design with extra interior and exterior design features.
- 10 Homes of this quality generally have high quality features. Fine finishes and more design quality is seen in the floor plans. Generally have a large kitchen island.
- 11 Custom design and higher quality finish work with added amenities like bathroom fixtures and more luxurious options.
- 12 Custom design and excellent builders. All materials are of the highest quality and conveniences are present.
- 13 Generally custom designed and built. Mansion level. Large amounts of cabinet work, wood trim, marble, entry ways etc.

In [55]:

```
1 data['grade'].value_counts()
```

7 Average	8762
8 Good	5928
9 Better	2505
6 Low Average	1967
10 Very Good	1046
11 Excellent	328
5 Fair	227
12 Luxury	54
4 Low	27
13 Mansion	5
3 Poor	1

Name: grade, dtype: int64

## Contents ⚙️

- 6 Comparing Renovated Homes with Unrenovated
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

```
In [56]: 1 data['grade'].replace(['3 Poor', '4 Low', '5 Fair', '6 Low',
2                               '7 Average', '8 Good', '9 Better', '10
3                               '11 Excellent', '12 Luxury', '13 Man
4                               [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
```

```
In [57]: 1 data['grade'].value_counts()
```

grade	count
4	8762
5	5928
6	2505
3	1967
7	1046
8	328
2	227
9	54
1	27
10	5
0	1

Name: grade, dtype: int64

For our purposes, it doesn't make sense to keep any Luxury or Man properties in the data set, since those kinds of properties would need to be renovated. This would better use of the space, and would effectively be making these properties unlivable. It also doesn't make sense to keep the single property in the poor condition category, since we assume that property will be on the list, and take it out of the data set. Therefore, I'll be dropping those rows.

Likewise it doesn't make sense to keep homes in this data set that are labeled as 'Man' or 'Excellent'. These homes would need essential building done, such as foundation work and water. While we are looking to do renovations on properties, there is a clear distinction between building and renovating. Therefore I'll be dropping those rows.

## FURTHER WORK:

## Contents ⚙️

<a href="#">6 Comparing Renovated Homes with Unrenovated</a>
6.2 Visualizing the Comparison
▼ 6.3 Home History
6.3.1 Age of Home
6.3.2 Year Modified
▼ 7 Distance Data
7.1 Getting Infrastructure Data
7.1.1 Downtown
7.1.2 Hospitals
7.1.3 Police Stations
7.1.4 Schools
7.2 Finding Distance for Infrastructure
7.3 Exploring Distance Data
7.3.1 Downtown
7.3.2 Hospitals
7.3.3 Police Stations
7.3.4 Schools
▼ 8 Preparing for Modeling
8.0.1 Dropping View and Waterfront Properties for Now
9 Zipcode
10 Seeing the Correlation
▼ 11 Building a model
11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2 Second Model: Transforming the Price with Log
▼ 12 Improving the Model with Adding Features
12.1 Third Model: Bedrooms, Floors, Closest School
12.2 Fourth Model: Square Foot Basement, Closest School
12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
12.4 Sixth Model: Condition, Waterfront and View
12.5 Seventh Model: Adding Zipcodes
12.6 Eighth Model: Latitude and Closest Hospital Distance
13 Conclusion
▼ 14 Recommendations:
14.0.1 Visualizing Schools

I would like to come back to this variable in future works and do further analysis on it. I am curious if homes currently at a grade of 6 or above meet building codes that are in place. It may be beneficial to come back and raise the minimum grade from the current data set to 6, to lower the cost of renovations needed to make these homes up to code. All homes not currently up to code would need to undergo renovations, should they be selected for the Section 8 Expansion.

In [58]:

```
1 poor = data.loc[data['grade'] == 0]
2
3 # labeling this single property for the city to buy, bulldoze, and
4 # make into a small public park
```

In [59]:

```
1 data = data.loc[data['grade'] < 9]
2 data = data.loc[data['grade'] >= 3]
```

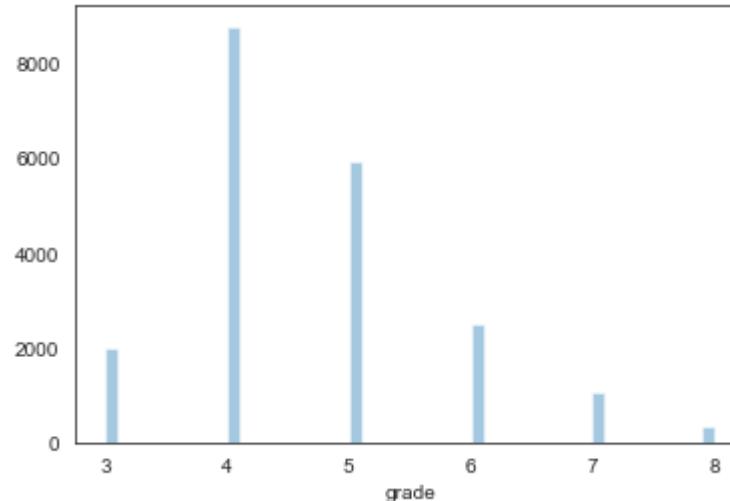
## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [60]:

```
1 sns.distplot( a=data["grade"], hist=True, kde=False, rug=False)
```

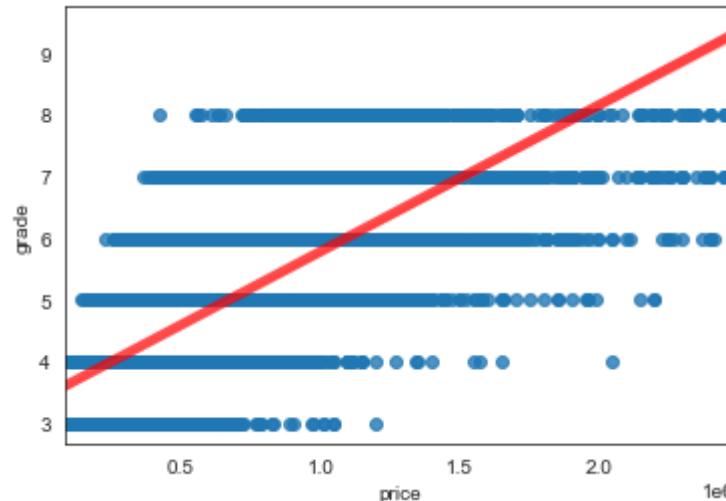
/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/sns/\_decorators.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function) or `histplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)



## Contents ⚙️

6.2	Visualizing the Comparison
6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Modeling
9	Zipcode
10	Seeing the Correlation
11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest School
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighbors
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
14	Recommendations:
14.0.1	Visualizing Schools

```
In [61]: 1 sns.regplot(data = data, x = 'price', y = 'grade',
2                      line_kws={"color":"r","alpha":0.7,"lw":5})
3 plt.show()
```



Observations: As grade improves, so does price, this is what we'd expect.

### 5.0.3 View

FROM THE DATASET: Includes views of Mt. Rainier, Olympics, City, Seattle Skyline, Puget Sound, Lake Washington, Lake Sammamish, and other. Represents quality of view from house.

Since most properties have "none" listed as the view, and this can start with, I am going to convert it to a binary column. Homes with views will have a value of 1, all other homes will have a value of 0.

## Contents ⚙️

6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Modeling
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest School
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

```
In [62]: 1 data['view'].value_counts()

          NONE      18569
          AVERAGE    883
          GOOD       455
          FAIR       316
          EXCELLENT   253
Name: view, dtype: int64
```

```
In [63]: 1 data['view'].replace(['NONE', 'FAIR', 'AVERAGE', 'GOOD', 'EXCELLENT'],
                           [0, 0, 0, 1, 1], inplace=True)

          2
          3
          4 # replacing the values to binary values
```

```
In [64]: 1 data['view'].fillna(value=0, inplace=True)

          2
          3 # filling in n/a values with 0
```

```
In [65]: 1 data['view'].value_counts()

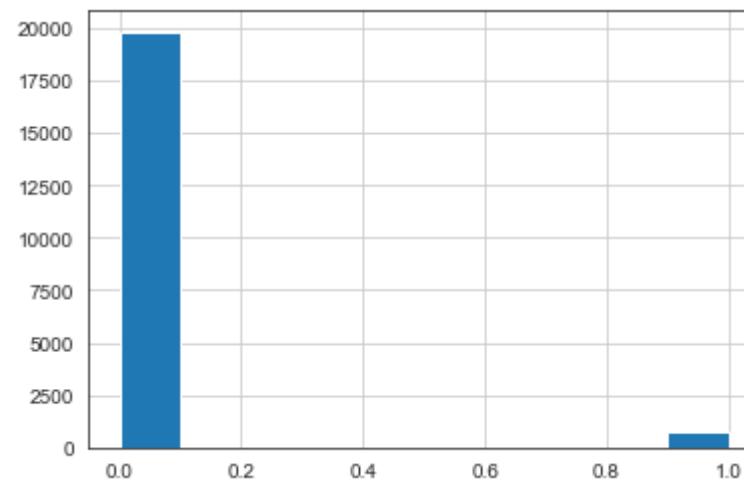
          2
          3 # taking a look at the distribution now

          0.0      19828
          1.0      708
Name: view, dtype: int64
```

## Contents ⚙

- 6.2 Visualizing the Comparison
- ▼ 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- ▼ 7 Distance Data
  - ▼ 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - ▼ 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- ▼ 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- ▼ 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- ▼ 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- ▼ 14 Recommendations:
  - 14.0.1 Visualizing Schools

```
In [66]: 1 data['view'].hist()  
2  
3 # it's easier to see visually  
<AxesSubplot:>
```

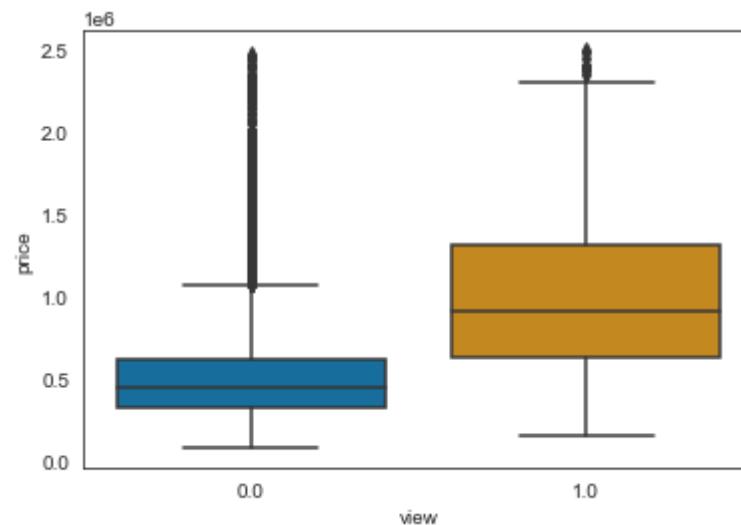


## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [67]:

```
1 sns.boxplot(data = data, x = 'view', y = 'price', palette =
2 plt.show();
```



Observations: We can see here that having a view has a clear impact on the price of the home, and there is a much larger spread in homes with a view than without. Likewise, we can see that homes without a view are generally less expensive than homes with a view but there are very many outlying cases where homes with a view are more expensive than homes without a view.

Contents  

- 6.2 Visualizing the Comparison
  - ▼ 6.3 Home History
    - 6.3.1 Age of Home
    - 6.3.2 Year Modified
  - ▼ 7 Distance Data
    - ▼ 7.1 Getting Infrastructure Data
      - 7.1.1 Downtown
      - 7.1.2 Hospitals
      - 7.1.3 Police Stations
      - 7.1.4 Schools
    - 7.2 Finding Distance for Infrastructure
  - ▼ 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
  - ▼ 8 Preparing for Modeling
    - 8.0.1 Dropping View and Waterfront Properties for Analysis
  - 9 Zipcode
  - 10 Seeing the Correlation
  - ▼ 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Logarithmic Function
  - ▼ 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - ▼ 14 Recommendations:
    - 14.0.1 Visualizing Schools

## 5.1 Year Renovated

Because we will be renovating each property selected for this pilot, it is important to understand the effects renovating a property will have.

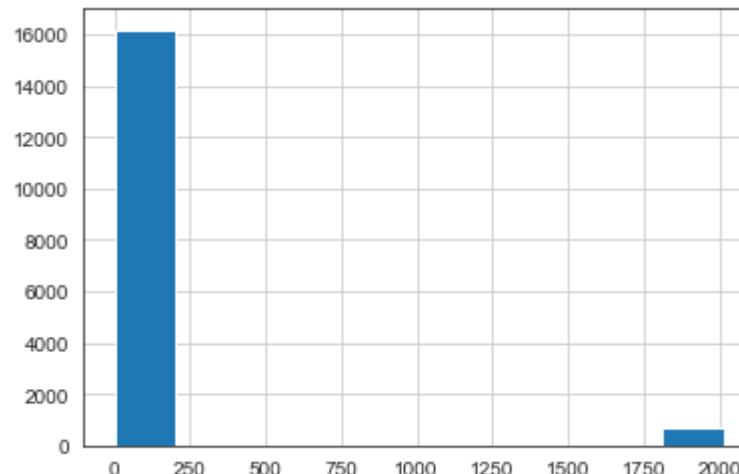
## Contents ⚙️

- [6.1 Comparing Renovated Homes with Unrenovated](#)
- [6.2 Visualizing the Comparison](#)
- ▼ [6.3 Home History](#)
  - [6.3.1 Age of Home](#)
  - [6.3.2 Year Modified](#)
- ▼ [7 Distance Data](#)
  - ▼ [7.1 Getting Infrastructure Data](#)
    - [7.1.1 Downtown](#)
    - [7.1.2 Hospitals](#)
    - [7.1.3 Police Stations](#)
    - [7.1.4 Schools](#)
  - [7.2 Finding Distance for Infrastructure](#)
  - ▼ [7.3 Exploring Distance Data](#)
    - [7.3.1 Downtown](#)
    - [7.3.2 Hospitals](#)
    - [7.3.3 Police Stations](#)
    - [7.3.4 Schools](#)
- ▼ [8 Preparing for Modeling](#)
  - [8.0.1 Dropping View and Waterfront Properties for Modeling](#)
  - [9 Zipcode](#)
  - [10 Seeing the Correlation](#)
- ▼ [11 Building a model](#)
  - [11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude](#)
  - [11.2 Second Model: Transforming the Price with Log](#)
- ▼ [12 Improving the Model with Adding Features](#)
  - [12.1 Third Model: Bedrooms, Floors, Closest School](#)
  - [12.2 Fourth Model: Square Foot Basement, Closest Hospital](#)
  - [12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods](#)
  - [12.4 Sixth Model: Condition, Waterfront and View](#)
  - [12.5 Seventh Model: Adding Zipcodes](#)
  - [12.6 Eighth Model: Latitude and Closest Hospital Distance](#)
- [13 Conclusion](#)
- ▼ [14 Recommendations:](#)
  - [14.0.1 Visualizing Schools](#)

In [69]:

```
1 data['yr_renovated'].hist()
2 # taking a look at the distributions of homes renovated vs unrenovated
```

&lt;AxesSubplot:&gt;



This plot doesn't tell us much except that the majority of homes in Seattle have never been renovated.

In [70]:

```
1 data['yr_renovated'].mean()
```

82.43672236046925

## Contents ⚙️

6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Modeling
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest School
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

```
In [71]: 1 data['yr_renovated'].isna().sum()/data.shape[0]
2
3 # 17% of the data in this column is missing, and in this case
4
5 0.17812621737436696
```

```
In [72]: 1 data['yr_renovated'].fillna(value = 0, inplace = True)
```

I'm filling the NA values with 0, as the vast majority of homes have been renovated in this dataset.

## 6 Analyzing Property Timelines

Because I'm now finished cleaning and preparing the data, I can start building a model. The first step is to compare renovated homes with non-renovated homes.

### 6.1 Comparing Renovated Homes with Non-Renovated Homes

## Contents ⚙️

- 6.1 Comparing Renovated Homes with Unrenovated
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

```
In [73]: 1 # the mean price for a home not renovated in this data set .  
2  
3 no_reno = data.loc[data['yr_renovated'] == 0]  
4 no_reno_stats = no_reno.mean()  
5 no_reno_stats
```

id	4.648427e+09
price	5.178077e+05
bedrooms	3.373759e+00
bathrooms	2.102563e+00
sqft_living	2.047173e+03
sqft_lot	1.018782e+04
floors	1.496295e+00
waterfront	4.082867e-03
view	3.165482e-02
condition	2.419174e+00
grade	4.651394e+00
sqft_above	1.762171e+03
sqft_basement	3.855537e-01
yr_built	1.972364e+03
yr_renovated	0.000000e+00
zipcode	9.807812e+04
lat	4.756143e+01
long	-1.222163e+02
sqft_living15	1.972658e+03
sqft_lot15	9.898066e+03
dtype:	float64

## Contents ⚙️

6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Modeling
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest School
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighbors
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

In [74]:

```
1 yes_reno = data.loc[data['yr_renovated'] != 0].copy()
2 yes_reno_stats = yes_reno.mean()
3 yes_reno_stats
```

id	4.514145e+09
price	7.200811e+05
bedrooms	3.440459e+00
bathrooms	2.268651e+00
sqft_living	2.248066e+03
sqft_lot	1.052812e+04
floors	1.502869e+00
waterfront	3.299857e-02
view	1.147776e-01
condition	2.216643e+00
grade	4.714491e+00
sqft_above	1.816657e+03
sqft_basement	5.007174e-01
yr_built	1.939235e+03
yr_renovated	1.996222e+03
zipcode	9.809764e+04
lat	4.758326e+01
long	-1.222680e+02
sqft_living15	1.947396e+03
sqft_lot15	9.874834e+03
dtype:	float64

## FURTHER WORK:

I'd like to come back to these statistics and do a much further analysis to see if there are any differences between the average renovated home vs non-renovated home. I would like to look at other variables besides price. IE does the average renovated home have more square footage or neighbors with more living space than non-renovated counterparts? I would also like to only be comparing the prices.

I'd also like to come back to the data and isolate non-renovated homes that have been renovated in the last 10 years from all other non-renovated homes and compare their prices. This will help me identify groups. However, in my initial research, I have only had the resolution to look at the year renovated, so I can't really do this.

## Contents ⚙️

6.2 Visualizing the Comparison
▼ 6.3 Home History
6.3.1 Age of Home
6.3.2 Year Modified
▼ 7 Distance Data
▼ 7.1 Getting Infrastructure Data
7.1.1 Downtown
7.1.2 Hospitals
7.1.3 Police Stations
7.1.4 Schools
7.2 Finding Distance for Infrastructure
▼ 7.3 Exploring Distance Data
7.3.1 Downtown
7.3.2 Hospitals
7.3.3 Police Stations
7.3.4 Schools
▼ 8 Preparing for Modeling
8.0.1 Dropping View and Waterfront Properties for Modeling
9 Zipcode
10 Seeing the Correlation
▼ 11 Building a model
11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2 Second Model: Transforming the Price with Logarithms
▼ 12 Improving the Model with Adding Features
12.1 Third Model: Bedrooms, Floors, Closest School
12.2 Fourth Model: Square Foot Basement, Closest School
12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4 Sixth Model: Condition, Waterfront and View
12.5 Seventh Model: Adding Zipcodes
12.6 Eighth Model: Latitude and Closest Hospital Distance
13 Conclusion
▼ 14 Recommendations:
14.0.1 Visualizing Schools

inspection of renovated VS non-renovated.

```
In [75]: 1 reno_differences = yes_reno_stats - no_reno_stats
          2 reno_difference = reno_differences['price']
          3 reno_difference
```

202273.36808995338

Of note: this is the exact calculated difference in average price between homes that have been renovated vs. homes that have not been renovated, ignoring other factors such as bedrooms or living space for example, this statistic does indicate that renovated homes cost \$201,941 more than non-renovated homes.

```
In [76]: 1 average_yes_reno_price = yes_reno_stats['price']
          2 average_yes_reno_price
```

720081.1162123386

```
In [77]: 1 budget = 100000000
          2 baseline_yes_reno = budget / average_yes_reno_price
          3 baseline_yes_reno
```

138.87324323404678

The average non renovated home in the remaining data after cleaning is priced at \$719,794. If we were to hypothetically spend our entire budget, we'd be able to buy 138 properties with our budget.

## Contents ⚙️

<a href="#">6.2 Visualizing the Comparison</a>
<a href="#">6.3 Home History</a>
<a href="#">6.3.1 Age of Home</a>
<a href="#">6.3.2 Year Modified</a>
<a href="#">7 Distance Data</a>
<a href="#">7.1 Getting Infrastructure Data</a>
<a href="#">7.1.1 Downtown</a>
<a href="#">7.1.2 Hospitals</a>
<a href="#">7.1.3 Police Stations</a>
<a href="#">7.1.4 Schools</a>
<a href="#">7.2 Finding Distance for Infrastructure</a>
<a href="#">7.3 Exploring Distance Data</a>
<a href="#">7.3.1 Downtown</a>
<a href="#">7.3.2 Hospitals</a>
<a href="#">7.3.3 Police Stations</a>
<a href="#">7.3.4 Schools</a>
<a href="#">8 Preparing for Modeling</a>
<a href="#">8.0.1 Dropping View and Waterfront Properties for Modeling</a>
<a href="#">9 Zipcode</a>
<a href="#">10 Seeing the Correlation</a>
<a href="#">11 Building a model</a>
<a href="#">11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude</a>
<a href="#">11.2 Second Model: Transforming the Price with Log</a>
<a href="#">12 Improving the Model with Adding Features</a>
<a href="#">12.1 Third Model: Bedrooms, Floors, Closest School</a>
<a href="#">12.2 Fourth Model: Square Foot Basement, Closest School</a>
<a href="#">12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods</a>
<a href="#">12.4 Sixth Model: Condition, Waterfront and View</a>
<a href="#">12.5 Seventh Model: Adding Zipcodes</a>
<a href="#">12.6 Eighth Model: Latitude and Closest Hospital Distance</a>
<a href="#">13 Conclusion</a>
<a href="#">14 Recommendations:</a>
<a href="#">14.0.1 Visualizing Schools</a>

In [78]:

```
1 average_no_reno_price = no_reno_stats['price']
2 average_no_reno_price
```

517807.7481223852

In [79]:

```
1 baseline_no_reno = budget / average_no_reno_price
2 baseline_no_reno
```

193.12186880673084

In [80]:

```
1 baseline_no_reno - baseline_yes_reno
```

54.24862557268406

The average non renovated home in the remaining data after cleaning is priced at \$517,853. That may sound pretty close to an average price of \$193,121. This means we can fit into the laid out budget for the project about 193 times.

For clarity, if we were only buying average priced non-renovated homes, we would need to buy 54 more homes than if we were only buying average price renovated homes.

To properly compare visually, I will take each home that has been renovated and subtract the average price of a non-renovated home from the sale price. This will recenter the visualization's bars to show how above or below each home's price is compared to the average un-renovated home.

## Contents ⚙️

6.2	Visualizing the Comparison
6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Modeling
9	Zipcode
10	Seeing the Correlation
11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Logarithms
12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest School
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
14	Recommendations:
14.0.1	Visualizing Schools

```
In [81]: 1 # make a function where the input is each price for each property
2 # been renovated and the output is that price minus the average
3 # renovated home
4
5 def find_price_adjusted(price):
6     price = price - average_no_reno_price
7     return price
8
9 yes_reno['price'] = yes_reno['price'].map(find_price_adjusted)
```

To organize the data, and get a sense of timeline, I'm going to be properties into chronological bins. Every 5 years will show very broad decade.

### FURTHER WORK:

I'd like to come back to this visualization with more time and resolution. I would like to have bins every four years, in correlation with the mayoral history of Seattle.

## Contents ⚙️

- [Comparing renovated homes with unrenovated](#)
- [6.2 Visualizing the Comparison](#)
- [6.3 Home History](#)
  - [6.3.1 Age of Home](#)
  - [6.3.2 Year Modified](#)
- [7 Distance Data](#)
  - [7.1 Getting Infrastructure Data](#)
    - [7.1.1 Downtown](#)
    - [7.1.2 Hospitals](#)
    - [7.1.3 Police Stations](#)
    - [7.1.4 Schools](#)
  - [7.2 Finding Distance for Infrastructure](#)
  - [7.3 Exploring Distance Data](#)
    - [7.3.1 Downtown](#)
    - [7.3.2 Hospitals](#)
    - [7.3.3 Police Stations](#)
    - [7.3.4 Schools](#)
- [8 Preparing for Modeling](#)
  - [8.0.1 Dropping View and Waterfront Properties for Modeling](#)
  - [9 Zipcode](#)
  - [10 Seeing the Correlation](#)
- [11 Building a model](#)
  - [11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude](#)
  - [11.2 Second Model: Transforming the Price with Log](#)
- [12 Improving the Model with Adding Features](#)
  - [12.1 Third Model: Bedrooms, Floors, Closest School](#)
  - [12.2 Fourth Model: Square Foot Basement, Closest Hospital](#)
  - [12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods](#)
  - [12.4 Sixth Model: Condition, Waterfront and View](#)
  - [12.5 Seventh Model: Adding Zipcodes](#)
  - [12.6 Eighth Model: Latitude and Closest Hospital Distance](#)
- [13 Conclusion](#)
- [14 Recommendations:](#)
  - [14.0.1 Visualizing Schools](#)

```
In [82]: 1 # aggregating the data by half-decade
2
3 bins = [ 1935, 1940, 1945, 1950, 1955, 1960, 1965, 1970, 1975, 1980,
4           1985, 1990, 1995, 2000, 2005, 2010, 2015]
5
6 labels = ['1940', '1945', '1950', '1955', '1960', '1965', '1970',
7           '1975', '1980', '1985', '1990', '1995', '2000', '2005', '2010', '2015']
8
9 cut_data, bin_edges = pd.cut(yes_reno['yr_renovated'], bins=bins,
10                               labels=labels, retbins=True)
11 bin_edges
```

array([1935, 1940, 1945, 1950, 1955, 1960, 1965, 1970, 1975, 1980, 1985,
 1990, 1995, 2000, 2005, 2010, 2015])

```
In [83]: 1 yes_reno.head()
2
3 #taking a look to make sure aggriagation worked correctly
```

	id	date	price	bedrooms	bathrooms	sqft_living	size_sqft
1	6414100192	12/9/2014	20192.251878	3	2.25	2570	710
35	9547205180	6/13/2014	178192.251878	3	2.50	2300	3000
95	1483300570	9/8/2014	387192.251878	4	2.50	3300	10000
103	2450000295	10/7/2014	572192.251878	3	2.50	2920	8000
125	4389200955	3/2/2015	932192.251878	4	2.75	2750	11000

5 rows × 21 columns

## Contents ⚙

6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Now
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest School
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighbors
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

In [84]:

1 yes\_reno['yr\_renovated'] = cut\_data

2

3 # replacing the yr\_renovated with the binned data

Since I have homes grouped by 5 year bins, it's important that for calculating average price instead of a sum value of all the properties in that bin, we take into account the different numbers of properties in each bin. Getting the average price will give us something comparable.

## Contents ⚙️

- [Comparing renovated homes with unrenovated](#)
- [6.2 Visualizing the Comparison](#)
- [6.3 Home History](#)
  - [6.3.1 Age of Home](#)
  - [6.3.2 Year Modified](#)
- [7 Distance Data](#)
  - [7.1 Getting Infrastructure Data](#)
    - [7.1.1 Downtown](#)
    - [7.1.2 Hospitals](#)
    - [7.1.3 Police Stations](#)
    - [7.1.4 Schools](#)
  - [7.2 Finding Distance for Infrastructure](#)
  - [7.3 Exploring Distance Data](#)
    - [7.3.1 Downtown](#)
    - [7.3.2 Hospitals](#)
    - [7.3.3 Police Stations](#)
    - [7.3.4 Schools](#)
- [8 Preparing for Modeling](#)
  - [8.0.1 Dropping View and Waterfront Properties for Modeling](#)
  - [9 Zipcode](#)
  - [10 Seeing the Correlation](#)
  - [11 Building a model](#)
    - [11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude](#)
    - [11.2 Second Model: Transforming the Price with Log](#)
  - [12 Improving the Model with Adding Features](#)
    - [12.1 Third Model: Bedrooms, Floors, Closest School](#)
    - [12.2 Fourth Model: Square Foot Basement, Closest Hospital](#)
    - [12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods](#)
    - [12.4 Sixth Model: Condition, Waterfront and View](#)
    - [12.5 Seventh Model: Adding Zipcodes](#)
    - [12.6 Eighth Model: Latitude and Closest Hospital Distance](#)
  - [13 Conclusion](#)
  - [14 Recommendations:](#)
    - [14.0.1 Visualizing Schools](#)

In [85]:

```
1 # making sure I get the average per bin, and resetting the index
2
3 yes_reno_averages = yes_reno.groupby(by='yr_renovated').mean()
4 yes_reno_averages = yes_reno_averages.reset_index()
5 yes_reno_averages
```

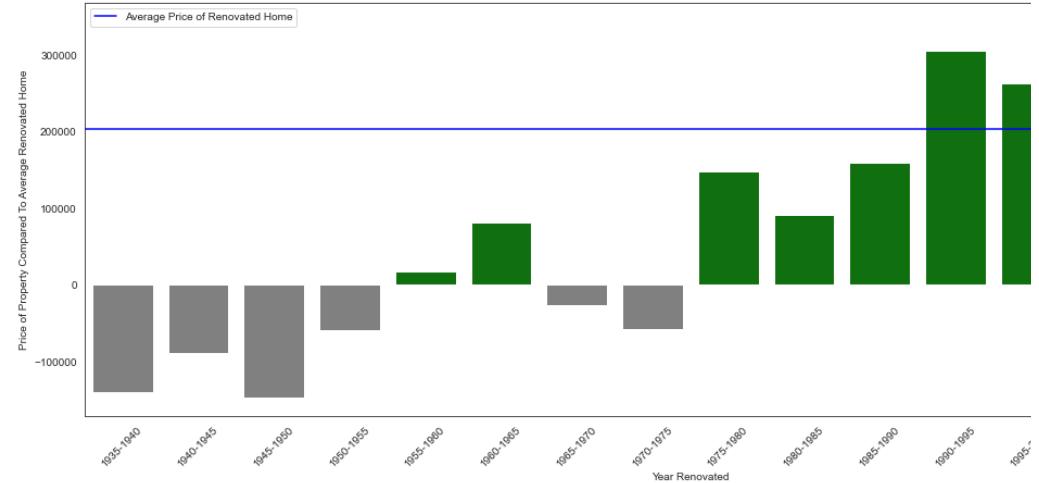
	yr_renovated	id	price	bedrooms	bathrooms	sqft_living
0	1940	1.451350e+09	-139407.748122	2.000000	1.000000	1150.0000
1	1945	4.448326e+09	-88557.748122	2.750000	1.687500	1607.5000
2	1950	3.606523e+09	-146841.081456	1.333333	1.166667	963.3333
3	1955	4.577260e+09	-59474.414789	3.166667	1.458333	1586.6666
4	1960	5.478101e+09	17870.251878	3.100000	1.500000	1814.5000
5	1965	3.869926e+09	81296.098031	3.846154	2.038462	2057.6923
6	1970	4.685418e+09	-26341.081456	3.380952	1.904762	1993.8095
7	1975	5.433291e+09	-58104.081456	2.933333	1.683333	1822.6666
8	1980	4.851001e+09	147021.797332	2.909091	1.931818	1958.1818
9	1985	4.466804e+09	90343.251878	3.320755	2.037736	2063.5849
10	1990	4.876861e+09	158870.585211	3.291667	2.093750	2166.2500
11	1995	4.523660e+09	305449.735749	3.451613	2.282258	2379.0161
12	2000	4.254875e+09	261863.451878	3.400000	2.523333	2478.4933
13	2005	4.306589e+09	343149.008634	3.630631	2.495495	2466.2522
14	2010	4.364097e+09	308062.113580	3.595745	2.500000	2468.5531
15	2015	4.627709e+09	132147.432329	3.563910	2.313910	2121.8270

## 6.2 Visualizing the Comparison

## Contents ⚙️

- .. Comparing Renovated Homes With Unrenovated
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

```
In [86]: 1 # color coding the bars to be green if the average price of
2 # is higher than the average non-renovated home, and grey if
3 # visual intuition
4
5 values = yes_reno_averages['price']
6 colors = ['grey' if (val < 0) else 'green' for val in values]
7
8 # making the graph
9 plt.figure(figsize = (20,7))
10 ax = sns.barplot(data = yes_reno_averages, x = 'yr_renovated',
11                   y ='price', palette = colors)
12
13 # making the labels for the bins
14 ax.set_xticklabels(['1935-1940', '1940-1945', '1945-1950',
15                     '1955-1960', '1960-1965', '1965-1970',
16                     '1975-1980', '1980-1985', '1985-1990',
17                     '1995-2000', '2000-2005', '2005-2010'],
18
19 # rotating my labels
20 plt.xticks(rotation=45);
21
22 # making the blue line where the average price for renovated
23 plt.axhline(reno_difference, color = 'blue', label = "Average Price")
24 plt.legend()
25
26 # making the labels and titles for the axes
27 ax.set(xlabel = 'Year Renovated',
28         ylabel = 'Price of Property Compared To Average',
29         title = "Average Price of Renovated Home Compared To Non-Renovated")
```

**KEY**

X axis = average price of non-renovated home

Green Bar = above average price of non-renovated home

Blue Line = average price of renovated home

This plot's 0 on the Y axis has been adjusted so that everything is relative to the average non-renovated home. The green bars represent where their price (grouped by every 5 years,) is above the average non-renovated home. The blue line represents the average price of all homes that have been renovated. Properties above the blue line represent renovated properties that are valued above the average renovated property.

**Observations:**

- 1) all properties renovated after 1975 are more valuable than the average non-renovated property.
- 2) properties renovated in the years between 1990 to 2010, are even more valuable compared to other renovated properties.

**Inferences:****Contents**

6.2 Visualizing the Comparison
6.3 Home History
6.3.1 Age of Home
6.3.2 Year Modified
7 Distance Data
7.1 Getting Infrastructure Data
7.1.1 Downtown
7.1.2 Hospitals
7.1.3 Police Stations
7.1.4 Schools
7.2 Finding Distance for Infrastructure
7.3 Exploring Distance Data
7.3.1 Downtown
7.3.2 Hospitals
7.3.3 Police Stations
7.3.4 Schools
8 Preparing for Modeling
8.0.1 Dropping View and Waterfront Properties for Modeling
9 Zipcode
10 Seeing the Correlation
11 Building a model
11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2 Second Model: Transforming the Price with Log Function
12 Improving the Model with Adding Features
12.1 Third Model: Bedrooms, Floors, Closest School
12.2 Fourth Model: Square Foot Basement, Closest Hospital
12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4 Sixth Model: Condition, Waterfront and View
12.5 Seventh Model: Adding Zipcodes
12.6 Eighth Model: Latitude and Closest Hospital Distance
13 Conclusion
14 Recommendations:
14.0.1 Visualizing Schools

## Contents ⚙️

- 6.1 Comparing Renovated Homes with Unrenovated
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
- 7.3 Exploring Distance Data
  - 7.3.1 Downtown
  - 7.3.2 Hospitals
  - 7.3.3 Police Stations
  - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

## Renovating properties increases their value, that's to be expected. We can now infer that if we renovate properties that have already been

## 6.3 Home History

### 6.3.1 Age of Home

```
In [87]: 1 def age_of_home(year_built):
2     age = 2014 - year_built
3
4
5 data['age_of_home'] = data['yr_built'].map(age_of_home)
6
7 # showing the age of the home rather than the year it was built
8 # a different perspective on the data
```

```
In [88]: 1 data.head()
```

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot
0	7129300520	10/13/2014	221900.0	3	1.00	1180	5650
1	6414100192	12/9/2014	538000.0	3	2.25	2570	7242
2	5631500400	2/25/2015	180000.0	2	1.00	770	10000
3	2487200875	12/9/2014	604000.0	4	3.00	1960	5000
4	1954400510	2/18/2015	510000.0	3	2.00	1680	8080

5 rows × 22 columns

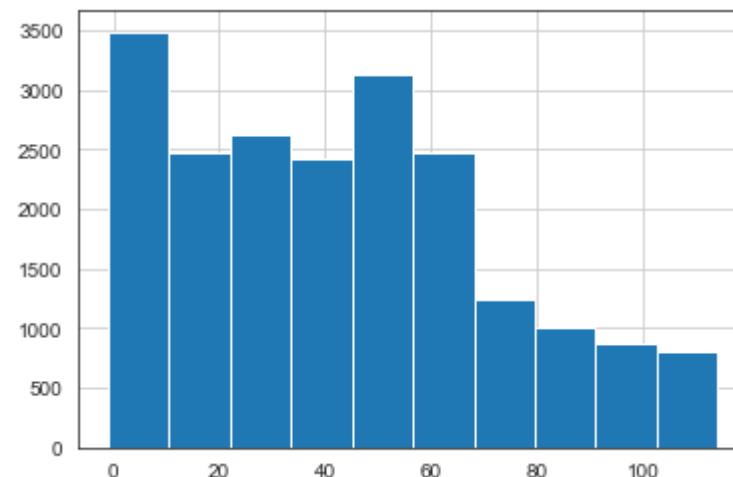
## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

In [89]:

```
1 data['age_of_home'].hist()
2
3 # seeing the distributions of how old the home is
```

&lt;AxesSubplot:&gt;



## 6.3.2 Year Modified

I'm going to make a new column for year modified. This column will represent the amount of years since last renovation or the date of construction, whichever happened most recently. This column helps liken newer non-renovated homes, and older non-renovated homes with homes that have been renovated a while ago.

## Contents ⚙️

- 6.1 Comparing Renovated Homes with Unrenovated
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

```
In [90]: 1 data['yr_mod'] = data[['yr_built','yr_renovated']].max(axis=1)
2
3 # this will select the most recent value between either year
```

```
In [91]: 1 data.head()
```

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot
0	7129300520	10/13/2014	221900.0	3	1.00	1180	5650
1	6414100192	12/9/2014	538000.0	3	2.25	2570	7242
2	5631500400	2/25/2015	180000.0	2	1.00	770	10000
3	2487200875	12/9/2014	604000.0	4	3.00	1960	5000
4	1954400510	2/18/2015	510000.0	3	2.00	1680	8080

5 rows × 23 columns

```
In [92]: 1 data['yr_mod'].value_counts().head()
```

2014.0	626
2005.0	465
2006.0	452
2003.0	440
2004.0	435

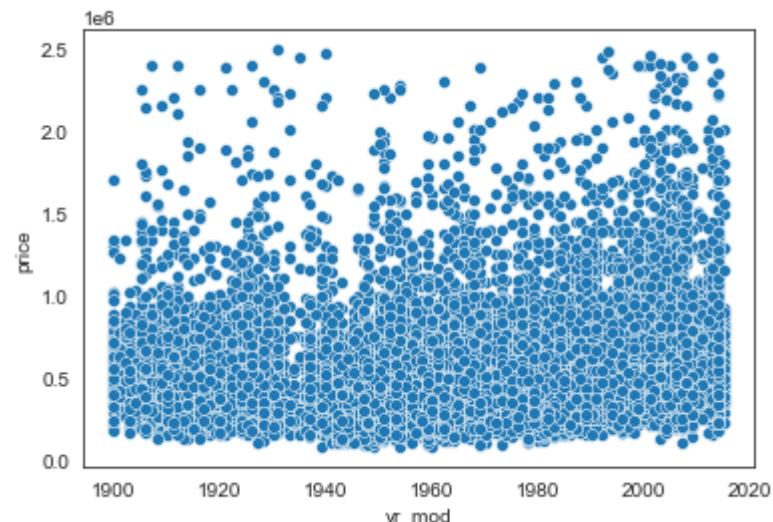
Name: yr\_mod, dtype: int64

## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [93]:

```
1 sns.scatterplot(data = data, x = 'yr_mod', y = 'price')
2 plt.show()
```



## Contents ⚙️

- 6.1 Comparing Renovated Homes with Unrenovated
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

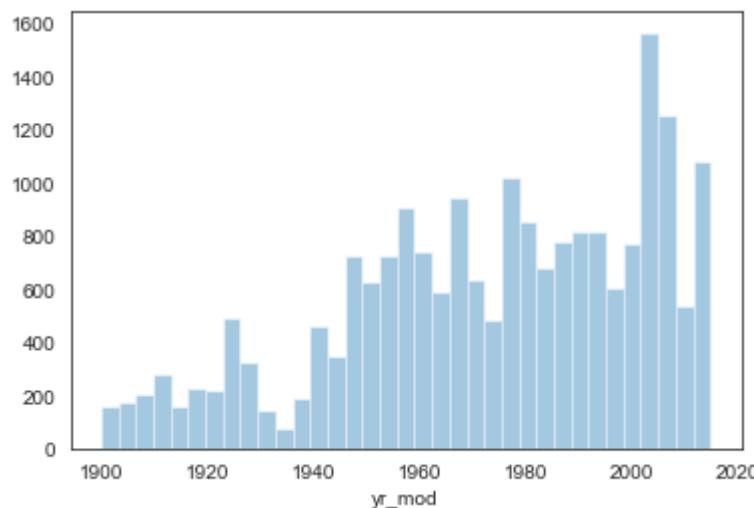
In [94]:

```
1 sns.distplot( a=data["yr_mod"], hist=True, kde=False, rug=False)
```

/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/sns/\_decorators.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function) or `histplot` (an axes-level function for histograms).

```
warnings.warn(msg, FutureWarning)
```

&lt;AxesSubplot:xlabel='yr\_mod'&gt;



## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
- 7.3 Exploring Distance Data
  - 7.3.1 Downtown
  - 7.3.2 Hospitals
  - 7.3.3 Police Stations
  - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

## 7 Distance Data

Now that I have limited the amount of properties I'll be looking up for, it's time to gather some more data.

Using Google's Geocoding api and scraping with Selenium to me from downtown, police stations, hospitals and schools.

### ASSUMPTIONS:

I have not had the resources yet to cross reference if the infrastructure have changed since 2015. I'm assuming in this data that the location of Police Stations and Schools in 2022 is the same as it was in 2015.

### FURTHER WORK:

I'd also like to include data about firehouses, which I have a source for "Data" but didn't have the resources to currently explore. As the scope increases, I'd like to include distances away from public parks, like universities, and public transportation.

I'd also like to do more research about the history of these infrastructure.

## 7.1 Getting Infrastructure Data

```
In [95]: 1 # for the api
           2
           3 with open('.secrets/google.json') as f:
           4     creds = json.load(f)
```

Making a function that intakes a list of addresses and outputs a list containing the coordinates for that address using google's API.

## Contents ⚙️

...	Comparing Neighborhood Homes with Surrounding
6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties fo
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Lat
11.2	Second Model: Transforming the Price with Loc
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest
12.3	Fifth Model: Square Foot Lot of Closest 15 Nei
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital D
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

```
In [96]: 1 # making an empty dictionary to contain the address as the key
          2 # of latitude and longitude as the values.
          3
          4 def get_coordinates(addresses):
          5     for address in addresses:
          6
          7         google_query = f"https://maps.googleapis.com/maps/api/geocode/json?address={address}"
          8
          9         results = requests.get(google_query)
         10        results_json = results.json()
         11        lat = results_json['results'][0]['geometry']['location']['lat']
         12        long = results_json['results'][0]['geometry']['location']['long']
         13
         14        coordinates[f'{address}']=(lat, long)
```

## 7.1.1 Downtown

```
In [97]: 1 downtown = (47.6050, -122.3344)
```

Making a function that intakes the dataframe, and a specific location. Then it finds the coordinates for that location, and adds a new row to the DF, column name as the property name at that location, this column lists the distance each property is away from the specified location.

## 7.1.2 Hospitals

### CITATION

<https://data-seattlecitygis.opendata.arcgis.com/datasets/hospitals/locations>  
<https://data-seattlecitygis.opendata.arcgis.com/datasets/hospitals/showTable=true>

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, and Longitude
  - 11.2 Second Model: Transforming the Price with Logarithms
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

```
In [98]: 1 # loading the data
2
3 hospital_data = pd.read_csv('Data/Hospitals.csv')
```

```
In [99]: 1 hospital_data.head()
2
3 # taking a look at the data in this csv
```

	X	Y	OBJECTID	FACILITY	ADDRESS	SE_ANNO_CAD_DATA
0	-122.336888	47.714248	1	UW Medical Center - Northwest	1550 N 115th ST	NaN
1	-122.379555	47.667365	2	Swedish Medical Center - Ballard Campus	5300 Tallman Ave NW	NaN
2	-122.341419	47.457800	3	Highline Medical Center	16251 Sylvester Rd SW	NaN
3	-122.214121	47.442273	4	Valley Medical Center	400 S 43rd St	NaN
4	-122.327279	47.292612	5	St. Francis Hospital	34515 9th Ave S	NaN

## Contents ⚙️

6.2 Visualizing the Comparison
6.3 Home History
6.3.1 Age of Home
6.3.2 Year Modified
7 Distance Data
7.1 Getting Infrastructure Data
7.1.1 Downtown
7.1.2 Hospitals
7.1.3 Police Stations
7.1.4 Schools
7.2 Finding Distance for Infrastructure
7.3 Exploring Distance Data
7.3.1 Downtown
7.3.2 Hospitals
7.3.3 Police Stations
7.3.4 Schools
8 Preparing for Modeling
8.0.1 Dropping View and Waterfront Properties for Modeling
9 Zipcode
10 Seeing the Correlation
11 Building a model
11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2 Second Model: Transforming the Price with Logarithm
12 Improving the Model with Adding Features
12.1 Third Model: Bedrooms, Floors, Closest School
12.2 Fourth Model: Square Foot Basement, Closest School
12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4 Sixth Model: Condition, Waterfront and View
12.5 Seventh Model: Adding Zipcodes
12.6 Eighth Model: Latitude and Closest Hospital Distance
13 Conclusion
14 Recommendations:
14.0.1 Visualizing Schools

```
In [100]: 1 hospital_addresses = hospital_data['ADDRESS'] + hospital_da
2
3 # getting proper addresses for the Google API
```

---

```
In [101]: 1 hospital_addresses.head()
2
3 # seeing the addresses
```

0	1550 N 115th STSeattle, WA, 98133-8401
1	5300 Tallman Ave NWSeattle, WA 98107
2	16251 Sylvester Rd SWBurien, WA, 98166-3017
3	400 S 43rd StRenton, WA, 98055-5714
4	34515 9th Ave SFederal Way, WA, 98003-6761

dtype: object

---

```
In [102]: 1 hospital_names = hospital_data['FACILITY']
2
3 # making a list of hospital names for later
```

## 7.1.3 Police Stations

### CITATION

<http://www.seattle.gov/police/about-us/police-locations>  
 (<http://www.seattle.gov/police/about-us/police-locations>)

## Contents ⚙

6.2	Visualizing the Comparison
6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Modeling
9	Zipcode
10	Seeing the Correlation
11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest School
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
14	Recommendations:
14.0.1	Visualizing Schools

```
In [103]: # I didn't bother to scrape for police data, there are only
# scraping would be more work than hardcoding at this point
police_stations = {'north_ps' : "10049 College Way N. Seattle, WA 98103",
'west_ps' : "810 Virginia Street, Seattle, WA 98101",
'east_ps' : "1519 12th Avenue Seattle, WA 98122",
'south_ps' : "3001 S. Myrtle Seattle, WA 98108",
'southwest_ps' : "2300 S.W. Webster Seattle, WA 98106"}
ps_addresses = police_stations.values()
ps_names = police_stations.keys()
```

## 7.1.4 Schools

For the purposes of this project, I'm only including public schools.

## Scraping for School Data

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, and Longitude
    - 11.2 Second Model: Transforming the Price with Logarithms
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

```
In [104]: 1 # first telling selenium where the webdriver is on my local
2 driver = webdriver.Chrome('/Users/b0ihazard/Desktop/chromedriver')
3
4 # telling selenium where the url is that I want to scrape from
5 # that up
6 school_url = "https://www.seattleschools.org/schools/"
7 go_article = driver.get(school_url)
8
9 # making a list of school names and addresses as selenium objects
10 grab_school_names = driver.find_elements(By.CSS_SELECTOR, "h3")
11 grab_school_addresses = driver.find_elements(By.CSS_SELECTOR, "p")
12
13 # making corresponding lists of addresses and names for text
14 school_addresses = []
15 school_names = []
16
17 # reformatting each raw address to a string and putting it on
18 for raw_address in grab_school_addresses:
19     address = raw_address.text.split("Main Office")[0]
20     school_addresses.append(address)
21
22 # reformatting each raw name to a string and putting it on
23 for raw_name in grab_school_names:
24     school_name = raw_name.text
25     school_names.append(school_name)
26
27 # closing the webdriver after use
28 driver.close()
```

## Contents ⚙️

6.2	Visualizing the Comparison
6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Modeling
9	Zipcode
10	Seeing the Correlation
11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest School
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
14	Recommendations:
14.0.1	Visualizing Schools

```
In [105]: 1 # taking a look at each school just to make sure formatting is correct
2
3 number = 0
4
5 for address in school_addresses:
6     number = number + 1
7     print(number, address)
```

1 6110 28th Ave. NW  
Seattle, WA 98107

2 3928 S Graham St.  
Seattle, WA 98118

3 8601 Rainier Ave. S  
Seattle, WA 98118

4 3010 59th Ave. SW  
Seattle, WA 98116

5 3701 SW 104th St.  
Seattle, WA 98146

6 1301 E Yesler Way  
Seattle, WA 98122

7 1418 NW 65th St.  
Seattle, WA 98117

```
In [106]: 1 school_addresses.remove(school_addresses[65])
```

This isn't an address so we must remove it from the data, but I have to take the accompanying name, Middle College from the list as well.

## Contents ⚙️

6.2 Visualizing the Comparison
▼ 6.3 Home History
6.3.1 Age of Home
6.3.2 Year Modified
▼ 7 Distance Data
▼ 7.1 Getting Infrastructure Data
7.1.1 Downtown
7.1.2 Hospitals
7.1.3 Police Stations
7.1.4 Schools
7.2 Finding Distance for Infrastructure
▼ 7.3 Exploring Distance Data
7.3.1 Downtown
7.3.2 Hospitals
7.3.3 Police Stations
7.3.4 Schools
▼ 8 Preparing for Modeling
8.0.1 Dropping View and Waterfront Properties for Modeling
9 Zipcode
10 Seeing the Correlation
▼ 11 Building a model
11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2 Second Model: Transforming the Price with Logarithm
▼ 12 Improving the Model with Adding Features
12.1 Third Model: Bedrooms, Floors, Closest School
12.2 Fourth Model: Square Foot Basement, Closest School
12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4 Sixth Model: Condition, Waterfront and View
12.5 Seventh Model: Adding Zipcodes
12.6 Eighth Model: Latitude and Closest Hospital Distance
13 Conclusion
▼ 14 Recommendations:
14.0.1 Visualizing Schools

```
In [107]: 1 school_names[65]
'Middle College'
```

```
In [108]: 1 school_names.remove('Middle College')
```

## 7.2 Finding Distance for Infrastructure

Getting a dictionary of coordinates based on the collected addresses  
Police Stations, Schools

```
In [109]: 1 coordinates ={}
```

```
In [110]: 1 get_coordinates(hospital_addresses)
```

```
In [111]: 1 get_coordinates(ps_addresses)
```

```
In [112]: 1 get_coordinates(school_addresses)
```

```
In [113]: 1 coordinates['downtown'] = downtown
```

Getting a list of all the names of these points of interests for the city

## Contents ⚙️

6.2 Visualizing the Comparison
6.3 Home History
6.3.1 Age of Home
6.3.2 Year Modified
7 Distance Data
7.1 Getting Infrastructure Data
7.1.1 Downtown
7.1.2 Hospitals
7.1.3 Police Stations
7.1.4 Schools
7.2 Finding Distance for Infrastructure
7.3 Exploring Distance Data
7.3.1 Downtown
7.3.2 Hospitals
7.3.3 Police Stations
7.3.4 Schools
8 Preparing for Modeling
8.0.1 Dropping View and Waterfront Properties for Modeling
9 Zipcode
10 Seeing the Correlation
11 Building a model
11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2 Second Model: Transforming the Price with Logarithm
12 Improving the Model with Adding Features
12.1 Third Model: Bedrooms, Floors, Closest School
12.2 Fourth Model: Square Foot Basement, Closest School
12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4 Sixth Model: Condition, Waterfront and View
12.5 Seventh Model: Adding Zipcodes
12.6 Eighth Model: Latitude and Closest Hospital Distance
13 Conclusion
14 Recommendations:
14.0.1 Visualizing Schools

```
In [114]: 1 all_names = []
2
3 for hospital in hospital_names:
4     all_names.append(hospital)
5 for ps_name in ps_names:
6     all_names.append(ps_name)
7 for school_name in school_names:
8     all_names.append(school_name)
9
```

---

```
In [115]: 1 all_names.append('downtown')
```

---

```
In [116]: 1 len(all_names) == len(coordinates)
2
3 # to make sure that each name has a set of coordinates and
4
5 True
```

---

```
In [117]: 1 counter = 0
2 columns = {}
3 for name in all_names:
4     columns[name] = counter
5     counter +=1
6
7 # making a dictionary of column names with an index for reference
```

---

Making a function that will calculate how far away each point of interest is from each point of interest a column, and making the rows how far away each point of interest is from each property. Each column should be labeled properly by its name.

## Contents ⚙️

6.1	Comparing Neighborhood Homes with Surrounding
6.2	Visualizing the Comparison
6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties fo
9	Zipcode
10	Seeing the Correlation
11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Lat
11.2	Second Model: Transforming the Price with Log
12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest
12.3	Fifth Model: Square Foot Lot of Closest 15 Nei
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital D
13	Conclusion
14	Recommendations:
14.0.1	Visualizing Schools

```
In [118]: 1 # intake: dataframe, location as a tuple of coordinates, l.
           2 # output: column on dataframe that measures distance from l
           3
           4 def distance_from (data, coordinates, all_names):
           5     counter = 0
           6     for coordinate_value in coordinates.values():
           7         distance_from_coordinates = []
           8         col_name = all_names[counter]
           9         for x in data.index:
          10             property_coordinates = (data.loc[x]['lat'], data.loc[x]['lon'])
          11             dist = distance.distance(property_coordinates, coordinate_value)
          12             distance_from_coordinates.append(dist)
          13         data[col_name] = distance_from_coordinates
          14         counter += 1
          15
          16 #     return f"added distance from {coordinate}, {col_name}"
```

```
In [119]: 1 distance_from(data, coordinates, all_names)
           2
           3 # implementing this function on all of our data does take some time
```

▼ Finding the closest possible Hospital, Police Station and School for each property.

### FURTHER WORK:

This is now a much larger amount of information, I've added over 1000 new columns to the original dataset. In further work, I would like to isolate each point of interest and compare how individual points of interest affect property values relative to other individual points of interest. However, given the parameters of this project, this is a good place to stop.

## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

must simplify this data significantly. Therefore, I'll be reducing the only three columns, to show the closest possible hospital, police station, and school. This will avoid the multicollinearity problem with having too much correlated data.

In [120]:

```
1 hospitals = hospital_names
```

## Contents ⚙️

6.2	Visualizing the Comparison
6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Modeling
9	Zipcode
10	Seeing the Correlation
11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest School
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
14	Recommendations:
14.0.1	Visualizing Schools

```
In [121]: 1 # making a dataframe just for the hospital information for
2 # the last columns in this dataframe will default to the di...
3
4 data_h = data[hospitals]
5 data_h.head()
6 data_h['closest_hospital_distance'] = data[['UW Medical Cen...
7 'Swedish Medical Center - Ballard Campus',
8 'Highline Medical Center',
9 'Valley Medical Center',
10 'St. Francis Hospital',
11 'Snoqualmie Valley Hospital',
12 'Swedish Medical Center - First Hill',
13 'MultiCare Auburn Medical Center',
14 'Overlake Hospital Medical Center',
15 "Seattle Children's Hospital",
16 'Virginia Mason Medical Center',
17 'EvergreenHealth - Kirkland',
18 'Swedish Medical Center - Cherry Hill',
19 'Seattle VA Medical Center',
20 'Swedish Medical Center - Issaquah Campus',
21 'Kaiser Permanente Capitol Hill Campus',
22 'St. Elizabeth Hospital',
23 'Harborview Medical Center',
24 'University of Washington Medical Center']].min(axis=1)
```

<ipython-input-121-b04b706e5bbc>:6: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-views-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-views-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-views-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-views-versus-a-copy))

## Contents ⚙️

6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
▼ 7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
▼ 7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Now
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest School
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

```
In [122]: 1 making a dataframe just for school information for future
2 the last columns in this dataframe will default to the di
3
4 data_s = data[school_names]
5 data_s['closest_school_distance'] = data[['Adams Elementary
6 Aki Kurose Middle School',
7 Alan T. Sugiyama High School',
8 Alki Elementary',
9 Arbor Heights Elementary',
10 Bailey Gatzert Elementary',
11 Ballard High School',
12 Beacon Hill International Elementary',
13 Benjamin Franklin Day Elementary',
14 Broadview-Thomson K-8',
15 Bryant Elementary',
16 Cascade Parent Partnership Program',
17 Cascadia Elementary',
18 Catharine Blaine K-8',
19 Cedar Park Elementary',
20 Chief Sealth International High School',
21 Cleveland High School',
22 Coe Elementary',
23 Concord International Elementary',
24 Daniel Bagley Elementary',
25 Dearborn Park International Elementary',
26 Decatur Elementary',
27 Denny International Middle School',
28 Dunlap Elementary',
29 Eckstein Middle School',
30 Emerson Elementary',
31 Fairmount Park Elementary',
32 Franklin High School',
```

## Contents ⚙️

- 6.2 Visualizing the Comparison
- ▼ 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- ▼ 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- ▼ 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
- ▼ 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, and Longitude
  - 11.2 Second Model: Transforming the Price with Logarithms
- ▼ 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- ▼ 14 Recommendations:
  - 14.0.1 Visualizing Schools

```
33'Garfield High School',
34'Gatewood Elementary',
35'Genesee Hill Elementary',
36'Graham Hill Elementary',
37'Green Lake Elementary',
38'Greenwood Elementary',
39'Hamilton International Middle School',
40'Hawthorne Elementary',
41'Hazel Wolf K-8',
42'Highland Park Elementary',
43'Ingraham High School',
44'Interagency Academy',
45'Jane Addams Middle School',
46'John Hay Elementary',
47'John Muir Elementary',
48'John Rogers Elementary',
49'John Stanford International Elementary',
50'Kimball Elementary',
51'Lafayette Elementary',
52'Laurelhurst Elementary',
53'Lawton Elementary',
54'Leschi Elementary',
55'Licton Springs K-8',
56'Lincoln High School',
57'Louisa Boren K-8',
58'Lowell Elementary',
59'Loyal Heights Elementary',
60'Madison Middle School',
61'Madrona Elementary',
62'Magnolia Elementary',
63'Maple Elementary',
64'Martin Luther King, Jr. Elementary',
```

## Contents ⚙️

- [6.2 Visualizing the Comparison](#)
- ▼ [6.3 Home History](#)
  - [6.3.1 Age of Home](#)
  - [6.3.2 Year Modified](#)
- ▼ [7 Distance Data](#)
  - ▼ [7.1 Getting Infrastructure Data](#)
    - [7.1.1 Downtown](#)
    - [7.1.2 Hospitals](#)
    - [7.1.3 Police Stations](#)
    - [7.1.4 Schools](#)
  - [7.2 Finding Distance for Infrastructure](#)
  - ▼ [7.3 Exploring Distance Data](#)
    - [7.3.1 Downtown](#)
    - [7.3.2 Hospitals](#)
    - [7.3.3 Police Stations](#)
    - [7.3.4 Schools](#)
- ▼ [8 Preparing for Modeling](#)
  - [8.0.1 Dropping View and Waterfront Properties for Modeling](#)
  - [9 Zipcode](#)
  - [10 Seeing the Correlation](#)
- ▼ [11 Building a model](#)
  - [11.1 Baseline Model: Grade, Bathrooms, Latitude, and Longitude](#)
  - [11.2 Second Model: Transforming the Price with Logarithms](#)
- ▼ [12 Improving the Model with Adding Features](#)
  - [12.1 Third Model: Bedrooms, Floors, Closest School](#)
  - [12.2 Fourth Model: Square Foot Basement, Closest Hospital](#)
  - [12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods](#)
  - [12.4 Sixth Model: Condition, Waterfront and View](#)
  - [12.5 Seventh Model: Adding Zipcodes](#)
  - [12.6 Eighth Model: Latitude and Closest Hospital Distance](#)
- [13 Conclusion](#)
- ▼ [14 Recommendations:](#)
  - [14.0.1 Visualizing Schools](#)

```
65 McClure Middle School',
66 McDonald International Elementary',
67 McGilvra Elementary',
68 Meany Middle School',
69 Mercer International Middle School',
70 Montlake Elementary',
71 Nathan Hale High School',
72 North Beach Elementary',
73 Northgate Elementary',
74 Nova',
75 Olympic Hills Elementary',
76 Olympic View Elementary',
77 Orca K-8',
78 Pathfinder K-8',
79 Queen Anne Elementary',
80 Rainier Beach High School',
81 Rainier View Elementary',
82 Rising Star Elementary',
83 Robert Eagle Staff Middle School',
84 Roosevelt High School',
85 Roxhill Elementary',
86 Sacajawea Elementary',
87 Salmon Bay K-8',
88 Sand Point Elementary',
89 Sanislo Elementary',
90 Seattle World School',
91 Skills Center',
92 South Shore PK-8',
93 Stevens Elementary',
94 The Center School',
95 Thornton Creek Elementary',
96 Thurgood Marshall Elementary',
```

## Contents ⚙️

- [6.2 Visualizing the Comparison](#)
- ▼ [6.3 Home History](#)
  - [6.3.1 Age of Home](#)
  - [6.3.2 Year Modified](#)
- ▼ [7 Distance Data](#)
  - ▼ [7.1 Getting Infrastructure Data](#)
    - [7.1.1 Downtown](#)
    - [7.1.2 Hospitals](#)
    - [7.1.3 Police Stations](#)
    - [7.1.4 Schools](#)
  - [7.2 Finding Distance for Infrastructure](#)
  - ▼ [7.3 Exploring Distance Data](#)
    - [7.3.1 Downtown](#)
    - [7.3.2 Hospitals](#)
    - [7.3.3 Police Stations](#)
    - [7.3.4 Schools](#)
- ▼ [8 Preparing for Modeling](#)
  - [8.0.1 Dropping View and Waterfront Properties for Modeling](#)
  - [9 Zipcode](#)
  - [10 Seeing the Correlation](#)
  - ▼ [11 Building a model](#)
    - [11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude](#)
    - [11.2 Second Model: Transforming the Price with Log](#)
  - ▼ [12 Improving the Model with Adding Features](#)
    - [12.1 Third Model: Bedrooms, Floors, Closest School](#)
    - [12.2 Fourth Model: Square Foot Basement, Closest Hospital](#)
    - [12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods](#)
    - [12.4 Sixth Model: Condition, Waterfront and View](#)
    - [12.5 Seventh Model: Adding Zipcodes](#)
    - [12.6 Eighth Model: Latitude and Closest Hospital Distance](#)
  - [13 Conclusion](#)
  - ▼ [14 Recommendations:](#)
    - [14.0.1 Visualizing Schools](#)

```
97 TOPS K-8',
98 View Ridge Elementary',
99 Viewlands Elementary',
100 Washington Middle School',
101 Wedgwood Elementary',
102 West Seattle Elementary',
103 West Seattle High School',
104 West Woodland Elementary',
105 Whitman Middle School',
106 Whittier Elementary',
107 Wing Luke Elementary']].min(axis=1)
```

<ipython-input-122-d282df538c6c>:5: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-ndarrays](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-ndarrays)  
data\_s['closest\_school\_distance'] = data[['Adams Elementary',

In [123]: 1 # data.to\_csv('Housing\_Project\_Full')

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

```
In [124]: 1 # making a dataframe just for police stations for future work
2 # the last columns in this dataframe will default to the distance
3
4
5 data_ps = data[ps_names]
6 data_ps['closest_police_station'] = data[['north_ps', 'west_ps',
7 'south_ps', 'southeast_ps']]
```

<ipython-input-124-643df7c19b3e>:6: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-views-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-views-versus-a-copy) ([https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-views-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-views-versus-a-copy))

## Simplifying the Data To Only Show the Closest Distance

```
In [125]: 1 to_drop = hospitals, school_names, ps_names
2 for column in to_drop:
3     data = data.drop(columns = column)
4
5 # dropping all distance data from main dataframe
```

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [126]: 1 data.head()

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot
0	7129300520	10/13/2014	221900.0	3	1.00	1180	5650
1	6414100192	12/9/2014	538000.0	3	2.25	2570	7242
2	5631500400	2/25/2015	180000.0	2	1.00	770	10000
3	2487200875	12/9/2014	604000.0	4	3.00	1960	5000
4	1954400510	2/18/2015	510000.0	3	2.00	1680	8080

5 rows × 24 columns

In [127]: `_h['closest_hospital_distance'], data_s['closest_school_distance']  
to_add:  
data.join(column)`

*lified distance data back into main dataframe*

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
- 7.3 Exploring Distance Data
  - 7.3.1 Downtown
  - 7.3.2 Hospitals
  - 7.3.3 Police Stations
  - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Logarithms
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [128]: 1 data.head()

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot
0	7129300520	10/13/2014	221900.0	3	1.00	1180	5650
1	6414100192	12/9/2014	538000.0	3	2.25	2570	7242
2	5631500400	2/25/2015	180000.0	2	1.00	770	10000
3	2487200875	12/9/2014	604000.0	4	3.00	1960	5000
4	1954400510	2/18/2015	510000.0	3	2.00	1680	8080

5 rows × 27 columns

## 7.3 Exploring Distance Data

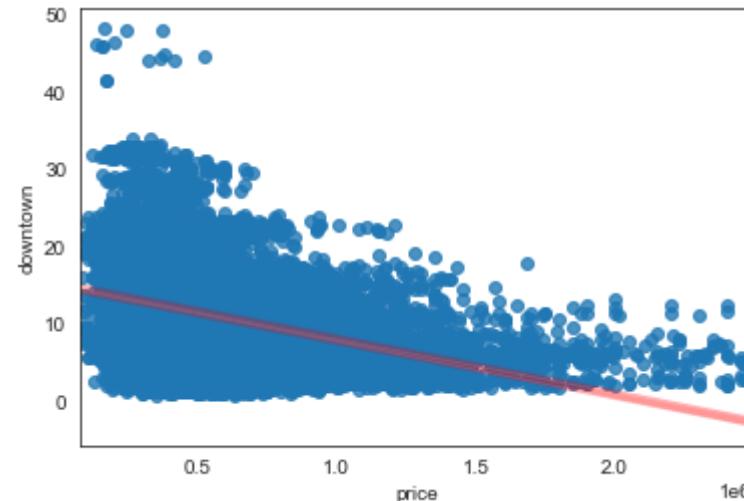
### 7.3.1 Downtown

I'm going to make a cut-off for houses more than 25 miles away from downtown Seattle. The goal of this project is to increase affordable housing in Seattle projects.

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Logarithm
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

```
In [129]: 1 sns.regplot(data = data, x = 'price', y = 'downtown',
2                         line_kws={"color":"r","alpha":0.3,"lw":5})
3 plt.show()
```



```
In [130]: 1 data = data.loc[data['downtown'] < 25]
```

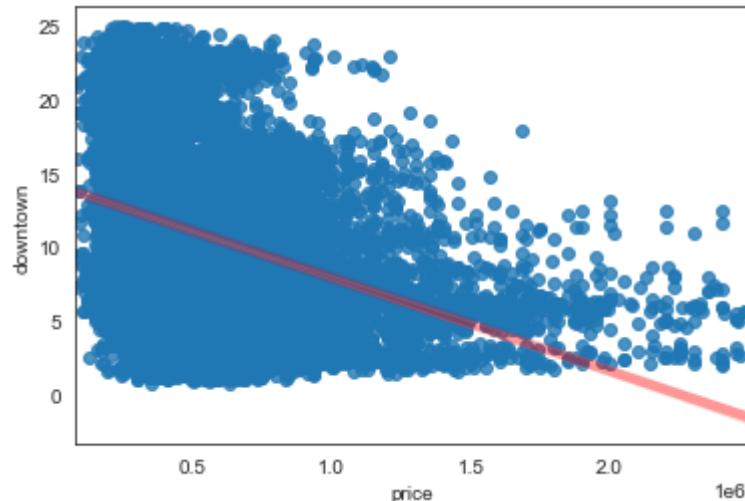
```
In [131]: 1 data['downtown'].max()
```

24.934920914948457

## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

```
In [132]: 1 sns.regplot(data = data, x = 'price', y = 'downtown',  
2                      line_kws={"color":"r","alpha":0.3,"lw":5})  
3 plt.show()
```



We can see some clustering in this graph, showing us that prices from downtown increases in the maximum prices for homes.

## Contents ⚙️

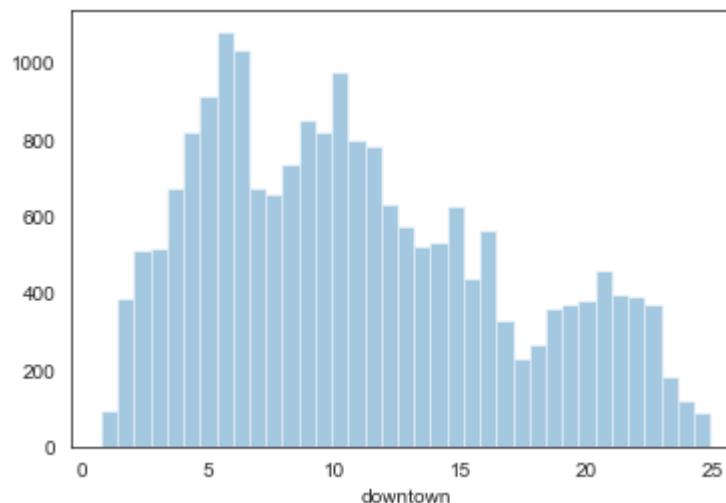
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [133]:

```
1 sns.distplot( a=data[ "downtown" ], hist=True, kde=False, rug:
```

/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/y:2551: FutureWarning: `distplot` is a deprecated function and will be removed. Please adapt your code to use either `displot` (a figure-level function) or `histplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)

&lt;AxesSubplot:xlabel='downtown'&gt;



## Contents ⚙️

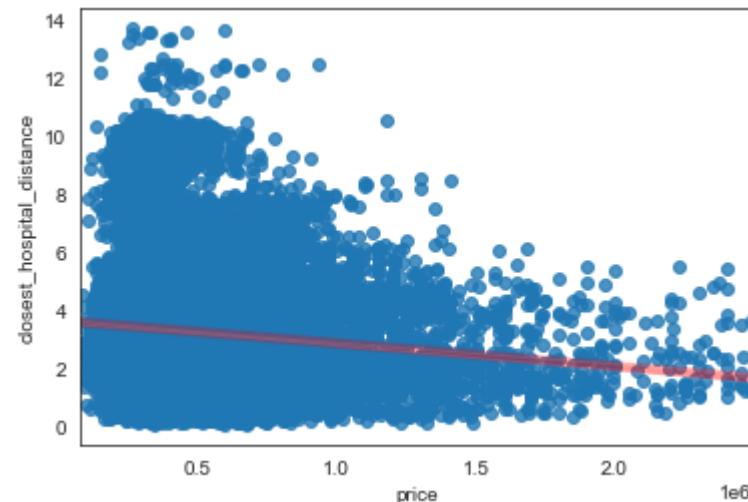
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
- 7.3 Exploring Distance Data
  - 7.3.1 Downtown
  - 7.3.2 Hospitals
  - 7.3.3 Police Stations
  - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

```
In [134]: 1 data['downtown'].describe()
```

	downtown
count	20109.000000
mean	10.923908
std	5.919272
min	0.719565
25%	5.979404
50%	10.079701
75%	15.018175
max	24.934921
Name:	downtown, dtype: float64

## 7.3.2 Hospitals

```
In [135]: 1 sns.regplot(data = data, x = 'price', y = 'closest_hospital_distance', line_kws={"color": "r", "alpha": 0.3, "lw": 5})
2
3 plt.show()
```



Observations: while there are plenty of low priced homes close to hospitals, the limit of how expensive a home can be does look to go down the further it gets from a hospital.

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

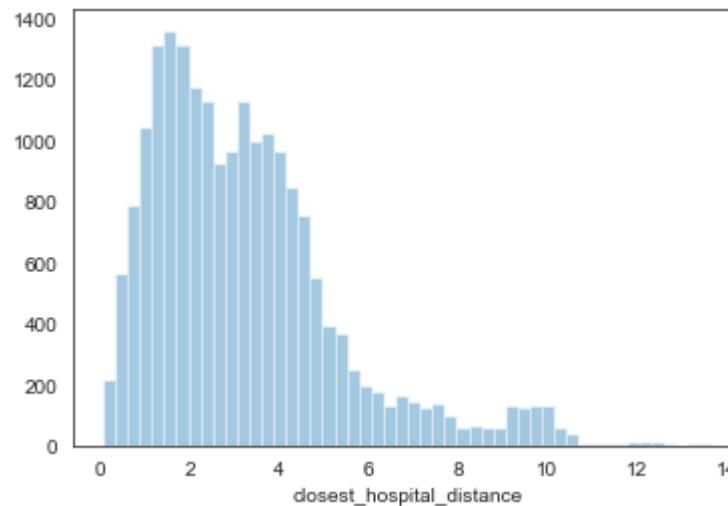
hospital it is.

In [136]:

```
1 sns.distplot( a=data["closest_hospital_distance"], hist=True)
```

```
/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/
y:2551: FutureWarning: `distplot` is a deprecated function and will be remo-
n. Please adapt your code to use either `displot` (a figure-level function
y) or `histplot` (an axes-level function for histograms).
warnings.warn(msg, FutureWarning)
```

```
<AxesSubplot:xlabel='closest_hospital_distance'>
```

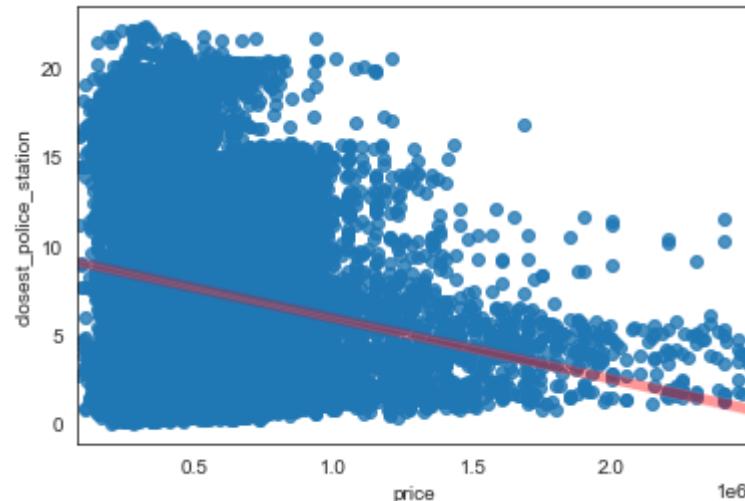


### 7.3.3 Police Stations

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Logarithmic Scale
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

```
In [137]: 1 sns.regplot(data = data, x = 'price', y = 'closest_police_station',
2                         line_kws={"color":"r","alpha":0.3,"lw":5})
3 plt.show()
```



Observations: while there are plenty of low priced homes close to a police station, the upper limit of how expensive a home can be does look to go down as the distance from a police station it is, similar to hospitals.

## Contents ⚙️

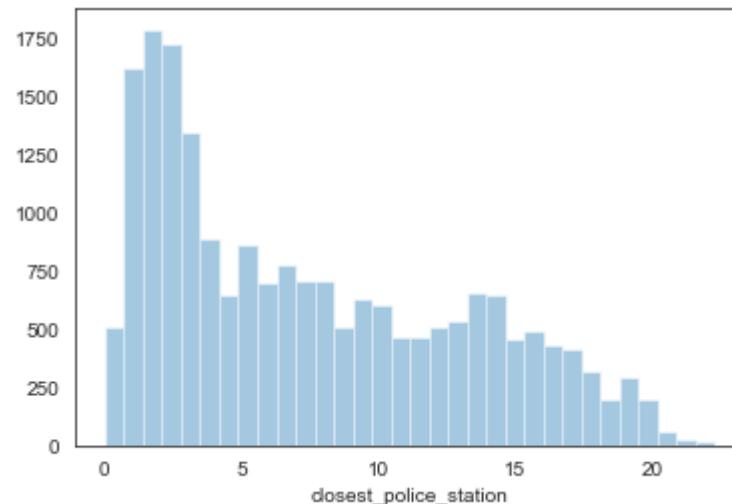
- 6.1 Comparing Renovated Homes with Unrenovated
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [138]:

```
1 sns.distplot( a=data["closest_police_station"], hist=True, kde=False)
```

/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/sns/\_decorators.py:2551: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function) or `histplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)

<AxesSubplot:xlabel='closest\_police\_station'>

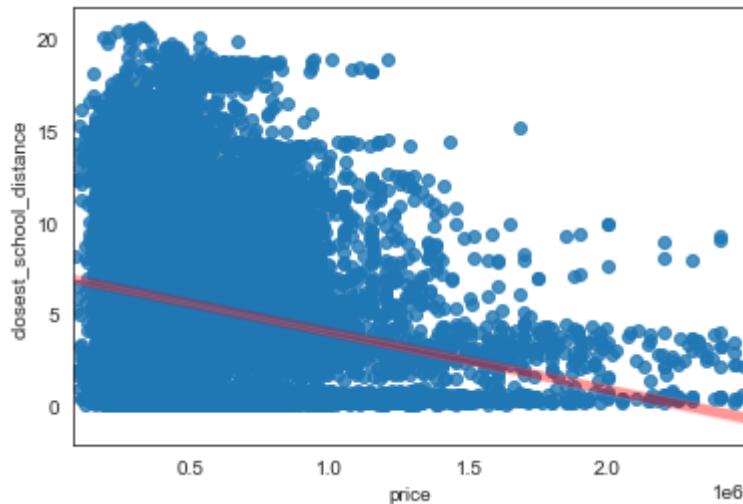


### 7.3.4 Schools

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Logarithms
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

```
In [139]: 1 sns.regplot(data = data, x = 'price', y = 'closest_school_distance',
2                      line_kws={"color":"r","alpha":0.3,"lw":5})
3 plt.show()
```



Observations: while there are plenty of low priced homes close to schools, the limit of how expensive a home can be does look to go down the further away from a school it is, similar to hospitals and police stations.

## Contents ⚙️

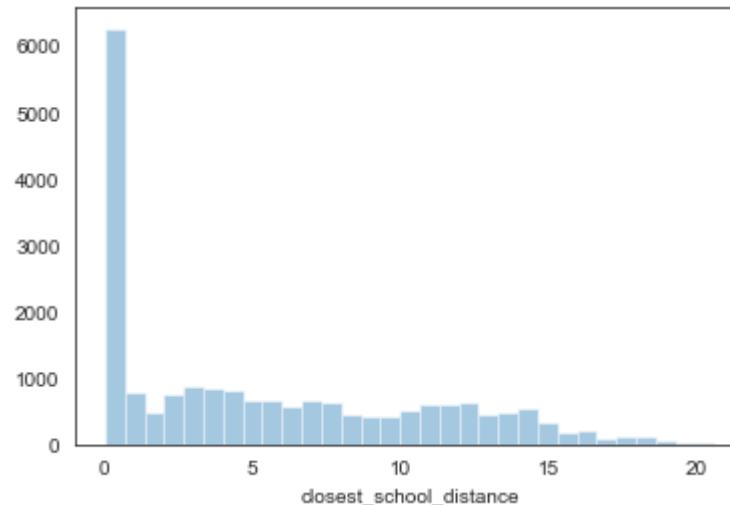
- 6.1 Comparing Renovated Homes with Unrenovated
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [140]:

```
1 sns.distplot( a=data["closest_school_distance"], hist=True,
```

/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/  
y:2551: FutureWarning: `distplot` is a deprecated function and will be rem  
n. Please adapt your code to use either `displot` (a figure-level function  
y) or `histplot` (an axes-level function for histograms).  
warnings.warn(msg, FutureWarning)

<AxesSubplot:xlabel='closest\_school\_distance'>



## Contents ⚙️

- 6.1 Comparing Renovated Homes with Unrenovated
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

# 8 Preparing for Modeling

## 8.0.1 Dropping View and Waterfront Properties Modeling

Since we are using a linear regression model, I must drop values for these features.

```
In [141]: 1 data['view'].value_counts()
```

	0.0	19419
	1.0	690
	Name:	view, dtype: int64

```
In [142]: 1 data['waterfront'].value_counts()
```

	0	20005
	1	104
	Name:	waterfront, dtype: int64

As a majority of data does not have a view or a waterfront, in order to have enough data to manipulate and have findings I must drop properties. Most of the values are at 1.

```
In [143]: 1 data = data.loc[data['view'] == 0]
2 data = data.loc[data['waterfront'] == 0]
```

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Lat, Long
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

In [144]:

```
1 data.hist(figsize=(20,20));
```



## Contents ⚙️

6.2 Visualizing the Comparison
▼ 6.3 Home History
6.3.1 Age of Home
6.3.2 Year Modified
▼ 7 Distance Data
▼ 7.1 Getting Infrastructure Data
7.1.1 Downtown
7.1.2 Hospitals
7.1.3 Police Stations
7.1.4 Schools
7.2 Finding Distance for Infrastructure
▼ 7.3 Exploring Distance Data
7.3.1 Downtown
7.3.2 Hospitals
7.3.3 Police Stations
7.3.4 Schools
▼ 8 Preparing for Modeling
8.0.1 Dropping View and Waterfront Properties for Modeling
9 Zipcode
10 Seeing the Correlation
▼ 11 Building a model
11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2 Second Model: Transforming the Price with Logarithm
▼ 12 Improving the Model with Adding Features
12.1 Third Model: Bedrooms, Floors, Closest School
12.2 Fourth Model: Square Foot Basement, Closest Hospital
12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4 Sixth Model: Condition, Waterfront and View
12.5 Seventh Model: Adding Zipcodes
12.6 Eighth Model: Latitude and Closest Hospital Distance
13 Conclusion
▼ 14 Recommendations:
14.0.1 Visualizing Schools

## 9 Zipcode

Using this kind of model, I'll need to pick one zipcode as the default and compare the price of a property going up or down depending on its distance from that zipcode.

Seeing as this project aims to specifically research class disparity, I'll use my default zipcode, it's the wealthiest zipcode in Seattle.

CITATION: <https://www.zipdatamaps.com/economics/income/agi/rank-of-zipcodes-in-metro-seattle-tacoma>  
[\(https://www.zipdatamaps.com/economics/income/agi/metro/wealthiest-zipcodes-in-metro-seattle-tacoma\)](https://www.zipdatamaps.com/economics/income/agi/metro/wealthiest-zipcodes-in-metro-seattle-tacoma) (Source: US Internal Revenue Service - 2019)

By comparing all other zipcodes to the wealthiest zipcode, we can see how much a zipcode can affect property values. I'm assuming 1) that the infrastructure is working well in this zipcode, and is reflected partially in the property value, and also assuming 2) that this zipcode was also the wealthiest zipcode in the area.

FURTHER WORK: I'd like to, in the future, expand this model to include more functionality, so the user can select which zipcode they'd like to compare each and every zipcode in relative scale to each other.

In [145]:

```
1 zip_dummies = pd.get_dummies(data.zipcode, drop_first=False)
2
3 # Concatenate the dummies to original dataframe
4 zipcode_data = pd.concat([data, zip_dummies], axis='columns')
5
6 ## drop zipcode 98039 instead of "first"
```

## Contents ⚙️

6.2 Visualizing the Comparison
▼ 6.3 Home History
6.3.1 Age of Home
6.3.2 Year Modified
▼ 7 Distance Data
▼ 7.1 Getting Infrastructure Data
7.1.1 Downtown
7.1.2 Hospitals
7.1.3 Police Stations
7.1.4 Schools
7.2 Finding Distance for Infrastructure
▼ 7.3 Exploring Distance Data
7.3.1 Downtown
7.3.2 Hospitals
7.3.3 Police Stations
7.3.4 Schools
▼ 8 Preparing for Modeling
8.0.1 Dropping View and Waterfront Properties for
9 Zipcode
10 Seeing the Correlation
▼ 11 Building a model
11.1 Baseline Model: Grade, Bathrooms, Lat
11.2 Second Model: Transforming the Price with Log
▼ 12 Improving the Model with Adding Features
12.1 Third Model: Bedrooms, Floors, Closest School
12.2 Fourth Model: Square Foot Basement, Closest
12.3 Fifth Model: Square Foot Lot of Closest 15 Nei
12.4 Sixth Model: Condition, Waterfront and View
12.5 Seventh Model: Adding Zipcodes
12.6 Eighth Model: Latitude and Closest Hospital D
13 Conclusion
▼ 14 Recommendations:
14.0.1 Visualizing Schools

```
In [146]: 1 zipcode_data= zipcode_data.drop(columns = 'zipcode_98039')
```

```
In [147]: 1 zipcode_data.shape
```

(19414, 94)

```
In [148]: 1 data.shape
```

(19414, 27)

I'll be using a combination of data and zipcode\_data when appropriate except one has each zipcode as a separate column and both of them is useful in making the correlation matrix.

Use csvs when only working on models for speed, and api usage

```
In [149]: 1 # data = pd.read_csv('Data/Housing_Project_No_Dummies',  
2 # index_col="Unnamed: 0").reset_index()  
3 # zipcode_data = pd.read_csv('Data/Housing_Project_With_Zipcodes',  
4 # index_col="Unnamed: 0").reset_index()
```

```
In [150]: 1 # data.to_csv('Housing_Project_No_Dummies')  
2 # zipcode_data.to_csv('Housing_Project_With_Zipcodes')
```

## 10 Seeing the Correlation

The next step is to avoid multicollinearity by looking at the correlation matrix. If two columns have a correlation of over 70 percent, one will be chosen based on the reasons described below.

## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

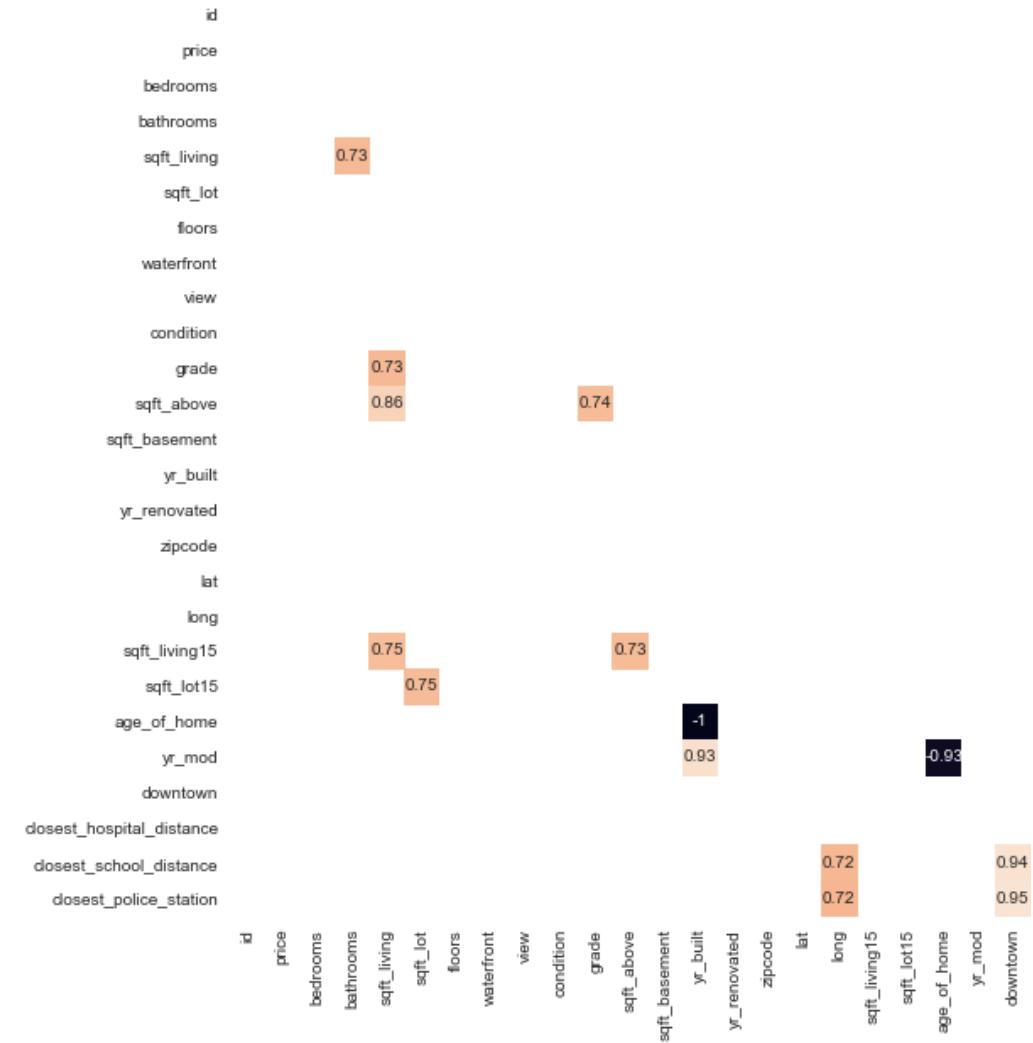
In [151]:

```
1 corr = data.corr()
2 # building the matrix
3 trimask = np.triu(np.ones_like(corr, dtype = 'bool'))
4 # defining the filter to only show half and avoid repeating
5 plt.figure(figsize=(12,10))
6 sns.heatmap(corr, mask = trimask | (np.abs(corr) <= 0.70),
7 # visualizing the correlation in seaborn and setting it to white
8 # correlation of over 70 percent
```

&lt;AxesSubplot:&gt;

## Contents ⚙️

- 6.2 Visualizing the Comparison
- ▼ 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- ▼ 7 Distance Data
  - ▼ 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - ▼ 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- ▼ 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
  - ▼ 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - ▼ 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - ▼ 14 Recommendations:
    - 14.0.1 Visualizing Schools



Choosing which features to drop: sqft\_living has an over 70% correlation with sqft\_living15. This makes sense, the closest 15 neighbors to a home have a similar size of living space. sqft\_living also has a correlation with sqft\_lot. This makes sense, as these features are very similar. sqft\_lot is closely correlated with sqft\_lot15 for similar reasons. sqft\_living also is interestingly correlated with sqft\_lot15. Because this feature is so correlated to others, I'll be dropping it.

## Contents ⚙️

- [Comparing Renovated Homes with Unrenovated](#)
- [6.2 Visualizing the Comparison](#)
- ▼ [6.3 Home History](#)
  - [6.3.1 Age of Home](#)
  - [6.3.2 Year Modified](#)
- ▼ [7 Distance Data](#)
  - ▼ [7.1 Getting Infrastructure Data](#)
    - [7.1.1 Downtown](#)
    - [7.1.2 Hospitals](#)
    - [7.1.3 Police Stations](#)
    - [7.1.4 Schools](#)
  - [7.2 Finding Distance for Infrastructure](#)
  - ▼ [7.3 Exploring Distance Data](#)
    - [7.3.1 Downtown](#)
    - [7.3.2 Hospitals](#)
    - [7.3.3 Police Stations](#)
    - [7.3.4 Schools](#)
- ▼ [8 Preparing for Modeling](#)
  - [8.0.1 Dropping View and Waterfront Properties from Data](#)
  - [9 Zipcode](#)
  - [10 Seeing the Correlation](#)
- ▼ [11 Building a model](#)
  - [11.1 Baseline Model: Grade, Bathrooms, Latitude, and Longitude](#)
  - [11.2 Second Model: Transforming the Price with Logarithms](#)
- ▼ [12 Improving the Model with Adding Features](#)
  - [12.1 Third Model: Bedrooms, Floors, Closest School](#)
  - [12.2 Fourth Model: Square Foot Basement, Closest Police Station](#)
  - [12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors](#)
  - [12.4 Sixth Model: Condition, Waterfront and View](#)
  - [12.5 Seventh Model: Adding Zipcodes](#)
  - [12.6 Eighth Model: Latitude and Closest Hospital Distance](#)
- [13 Conclusion](#)
- ▼ [14 Recommendations:](#)
  - [14.0.1 Visualizing Schools](#)

For only keeping one of those variables, Keeping sqft\_15 makes about the closest 15 neighbors, not about the property itself, which is better since this metric is more closely related to the theory of machine learning. However, sqft\_above is also too closely correlated with grade. Among all the features, grade is the feature that we have most control of with respect to the property. Therefore I'll be getting rid of sqft\_above to keep grade. yr\_built, yr\_renovated and yr\_mod are all correlating with each other which makes sense. Of the three features, I'm keeping yr\_renovated since it's more closely related to our interests.

Closest school is almost entirely predicted by closest police station. I'll be dropping closest police station as closest school has a more complex relationship with it. Downtown is shows multicollinearity with closest police station and closest school so I'll be dropping downtown also.

Additionally, I'll be dropping zipcode since I've made those into discrete categories.

```
In [152]: 1 zipcode_data = zipcode_data.drop(columns = ['sqft_living', 'sqft_lot', 'zipcode', 'yr_mod', 'downtown', 'closest_police_station'])
2
3
4
5 data = data.drop(columns = ['sqft_living', 'sqft_living15', 'zipcode', 'yr_built', 'yr_mod', 'sqft_above', 'closest_police_station'])
6
7
8
9 # dropping these features from both data sets
```

## Contents ⚙

- [Comparing Neighborhoods with Correlation](#)
- [6.2 Visualizing the Comparison](#)
- [6.3 Home History](#)
  - [6.3.1 Age of Home](#)
  - [6.3.2 Year Modified](#)
- [7 Distance Data](#)
  - [7.1 Getting Infrastructure Data](#)
    - [7.1.1 Downtown](#)
    - [7.1.2 Hospitals](#)
    - [7.1.3 Police Stations](#)
    - [7.1.4 Schools](#)
  - [7.2 Finding Distance for Infrastructure](#)
  - [7.3 Exploring Distance Data](#)
    - [7.3.1 Downtown](#)
    - [7.3.2 Hospitals](#)
    - [7.3.3 Police Stations](#)
    - [7.3.4 Schools](#)
- [8 Preparing for Modeling](#)
  - [8.0.1 Dropping View and Waterfront Properties for Modeling](#)
  - [9 Zipcode](#)
  - [10 Seeing the Correlation](#)
- [11 Building a model](#)
  - [11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude](#)
  - [11.2 Second Model: Transforming the Price with Log](#)
- [12 Improving the Model with Adding Features](#)
  - [12.1 Third Model: Bedrooms, Floors, Closest School](#)
  - [12.2 Fourth Model: Square Foot Basement, Closest Hospital](#)
  - [12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods](#)
  - [12.4 Sixth Model: Condition, Waterfront and View](#)
  - [12.5 Seventh Model: Adding Zipcodes](#)
  - [12.6 Eighth Model: Latitude and Closest Hospital Distance](#)
- [13 Conclusion](#)
- [14 Recommendations:](#)
  - [14.0.1 Visualizing Schools](#)

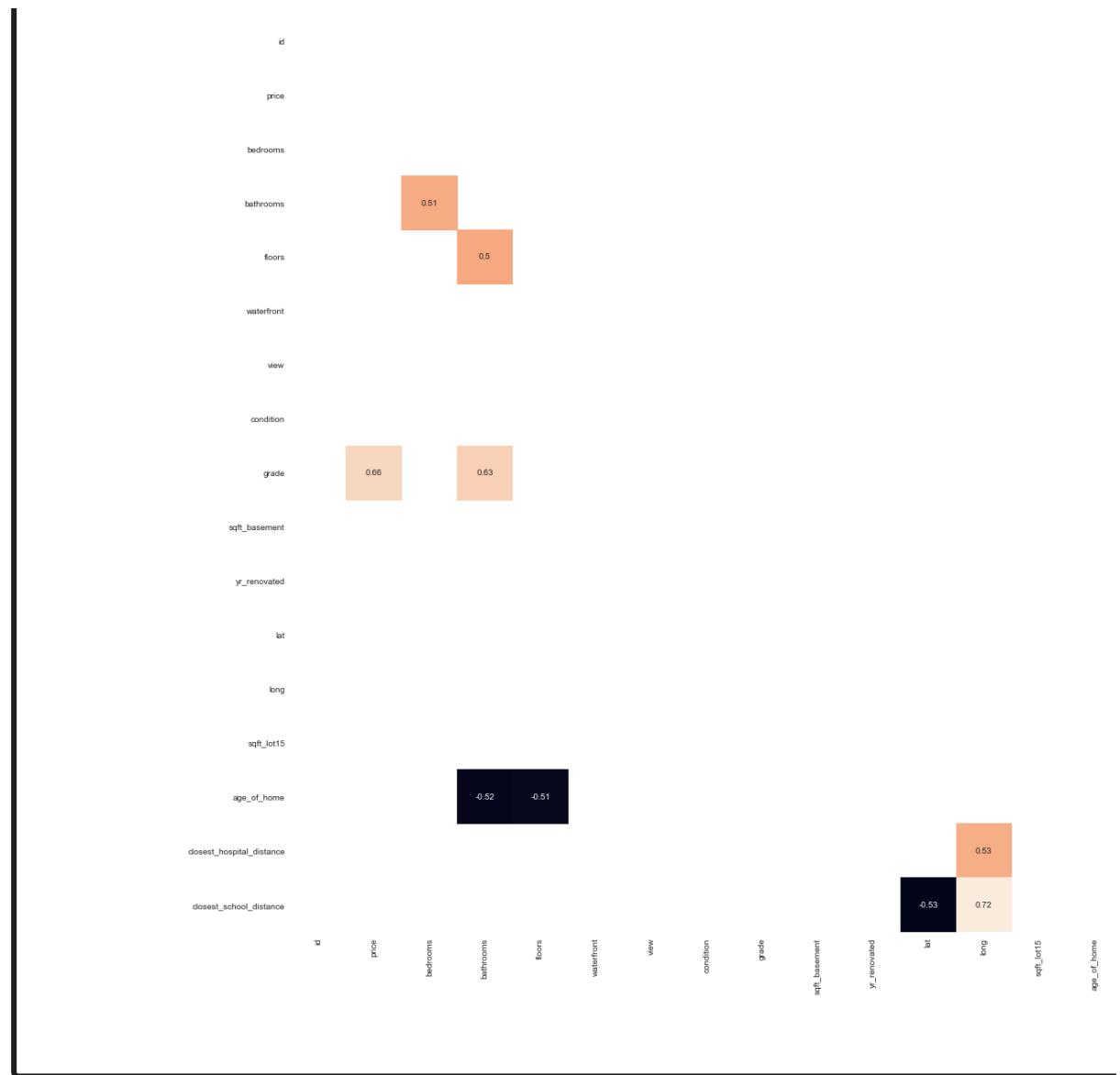
In [153]:

```
1 corr = data.corr()
2 trimask = np.triu(np.ones_like(corr, dtype = 'bool'))
3 plt.figure(figsize=(25,20))
4 sns.heatmap(corr, mask = trimask | (np.abs(corr) <= 0.5), annot=True)
5
6 # taking a look to see the results and to make sure there's
7 # close to 70 percent
```

&lt;AxesSubplot:&gt;

## Contents ⚙️

- 6.2 Visualizing the Comparison
- ▼ 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- ▼ 7 Distance Data
  - ▼ 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - ▼ 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- ▼ 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
  - ▼ 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Lat, Long
    - 11.2 Second Model: Transforming the Price with Log
  - ▼ 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - ▼ 14 Recommendations:
    - 14.0.1 Visualizing Schools



## Contents ⚙️

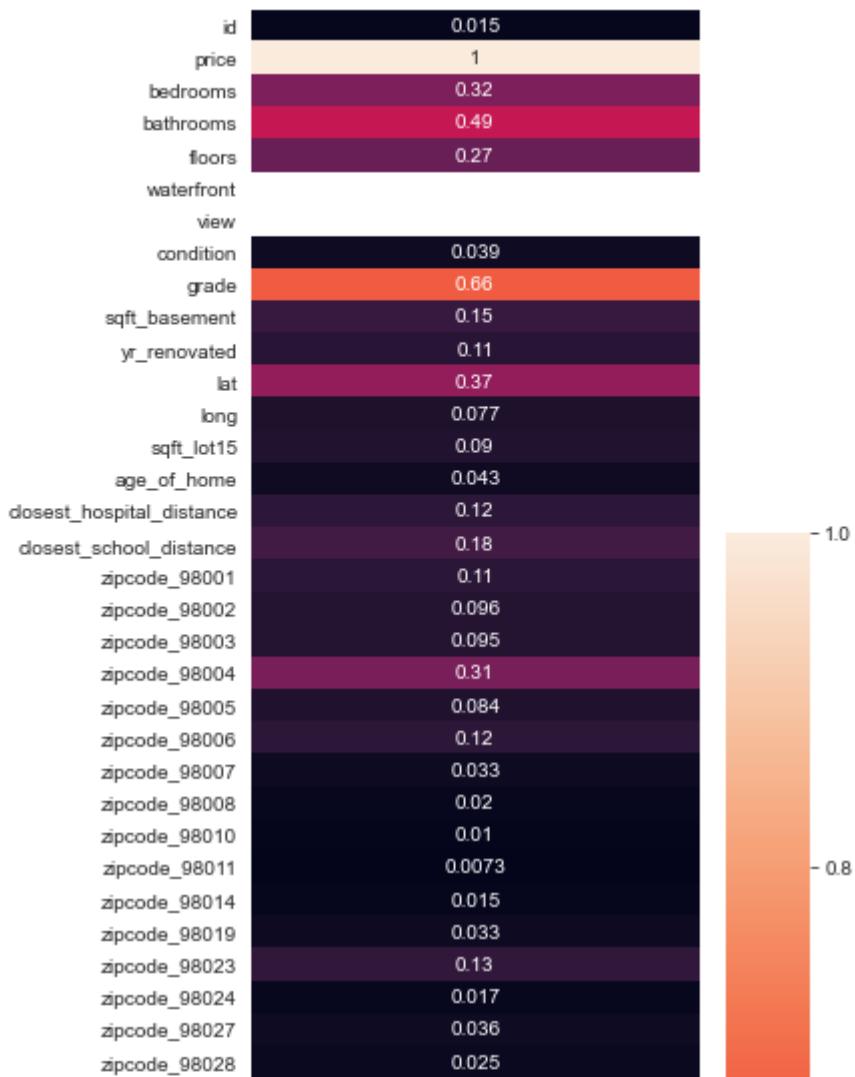
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Logarithm
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

In [154]:

```

1 sns.set_style('white')
2
3 plt.figure(figsize=(5, 25))
4 sns.heatmap(abs(zipcode_data.corr())[['price']], annot=True)
5
6 # now we can see the correlation with price and get a sense
7 # zipcode data will be

```



## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

## Seattle\_Section\_8\_Project - Jupyter Notebook



## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Logarithms
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

zipcode_98177	0.019
zipcode_98178	0.091
zipcode_98188	0.065
zipcode_98198	0.097
zipcode_98199	0.088

price

```
In [155]: 1 order_and_values = np.abs(data.corr()['price']).sort_values()
2 # saving this series of features and correlations as an object
```

```
In [156]: 1 order = list(order_and_values.index)
2 # saving only the index of that series
```

```
In [157]: 1 order.remove('price')
2 order.remove('id')
3 # removing price and id from order
```

```
In [158]: 1 features_to_remove = []
2 # making an empty list of features to remove
3 order = [feature for feature in order if feature not in features_to_remove]
4 # making sure the order doesn't include any of those features
```

## 11 Building a model

I'm going to build two helper functions here, one to tell me which features have p-values above 0.05 and are therefore unable to reject the null hypothesis (so they're not doing a good job of predicting.)

I'm also going to build a simple function to tell me the next three most important features in the order of features.

Further work:

## Contents ⚙️

6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
▼ 7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
▼ 7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Now
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest Hospital
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

Currently these two helper functions are sketched out but with more time and resources, I'd like to make the model building more automated. This will be helpful for the civic workers tasked with using the model, and also higher up in the organization. I wanted to keep these preliminary models however, I wanted to keep a track of what was included in each model, which is why some of the code is repetitive.

### Bugs:

Currently, the list `features_to_remove` never gets cleared of information. It needs to be emptied before each iteration of the model. As things are right now, it's working but it's not efficient, needing to remove that many features, so this is working as well. But with more time and resources, this functionality would be improved.

```
In [159]: 1 def high_p_vals(model):
2     p_vals = model.pvalues
3     labels = list(model.pvalues.index)
4     p_dict = dict(zip(labels, p_vals))
5     for key, value in p_dict.items():
6         if value > 0.05:
7             features_to_remove.append(key)
8     else:
9         print(f'{key} has an acceptable p_value')
10    return f'drop {features_to_remove} for the next model iteration'
```

```
In [160]: 1 def next_features(order, features_to_remove):
2     print (order[0: 3])
3     del order[0: 3]
```

## 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude

## Contents ⚙️

6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
▼ 7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
▼ 7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Now
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Lat
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighbors
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

```
In [161]: 1 next_features(order, features_to_remove)
2
3 # finding out what features to include
4
5 ['grade', 'bathrooms', 'lat']

In [162]: 1 df_baseline = zipcode_data.loc[:, ['price', 'grade', 'bathrooms']]
2 df_baseline.info()
3
4 # making a new DF with those features
5
6
7 <class 'pandas.core.frame.DataFrame'>
8 Int64Index: 19414 entries, 0 to 21596
9 Data columns (total 4 columns):
10 #   Column      Non-Null Count  Dtype  
11 ---  --          --          --    
12 0   price       19414 non-null  float64
13 1   grade        19414 non-null  int64  
14 2   bathrooms    19414 non-null  float64
15 3   lat          19414 non-null  float64
16 dtypes: float64(3), int64(1)
17 memory usage: 758.4 KB

In [163]: 1 y = zipcode_data['price']
2 X = df_baseline.drop(['price'], axis = 1)
3
4 # assigning my X and y variables
```

## Contents ⚙️

6.2	Visualizing the Comparison
6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Now
9	Zipcode
10	Seeing the Correlation
11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Zipcode
11.2	Second Model: Transforming the Price with Log
12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest Hospital
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
14	Recommendations:
14.0.1	Visualizing Schools

```
In [164]: 1 baseline_model = sm.OLS(y, sm.add_constant(X)).fit()
2 baseline_model.summary()
3
4 # building the model and getting a summary
```

### OLS Regression Results

Dep. Variable:	price	R-squared:	0.544			
Model:	OLS	Adj. R-squared:	0.544			
Method:	Least Squares	F-statistic:	7709.			
Date:	Tue, 12 Apr 2022	Prob (F-statistic):	0.00			
Time:	17:54:48	Log-Likelihood:	-2.6330e+05			
No. Observations:	19414	AIC:	5.266e+05			
Df Residuals:	19410	BIC:	5.266e+05			
Df Model:	3					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	-3.169e+07	4.77e+05	-66.366	0.000	-3.26e+07	-3.08e+07
grade	1.414e+05	1660.306	85.164	0.000	1.38e+05	1.45e+05
bathrooms	5.583e+04	2385.026	23.407	0.000	5.12e+04	6.05e+04
lat	6.607e+05	1e+04	65.753	0.000	6.41e+05	6.8e+05
Omnibus:	9369.097	Durbin-Watson:	1.958			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	88342.127			
Skew:	2.108	Prob(JB):	0.00			
Kurtosis:	12.562	Cond. No.	1.69e+04			

### Notes:

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

[2] The condition number is large, 1.69e+04. This might indicate that there is strong multicollinearity or other numerical problems.

## Contents ⚙️

6.2 Visualizing the Comparison
6.3 Home History
6.3.1 Age of Home
6.3.2 Year Modified
7 Distance Data
7.1 Getting Infrastructure Data
7.1.1 Downtown
7.1.2 Hospitals
7.1.3 Police Stations
7.1.4 Schools
7.2 Finding Distance for Infrastructure
7.3 Exploring Distance Data
7.3.1 Downtown
7.3.2 Hospitals
7.3.3 Police Stations
7.3.4 Schools
8 Preparing for Modeling
8.0.1 Dropping View and Waterfront Properties for Now
9 Zipcode
10 Seeing the Correlation
11 Building a model
11.1 Baseline Model: Grade, Bathrooms, Latitude
11.2 Second Model: Transforming the Price with Log
12 Improving the Model with Adding Features
12.1 Third Model: Bedrooms, Floors, Closest School
12.2 Fourth Model: Square Foot Basement, Closest
12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
12.4 Sixth Model: Condition, Waterfront and View
12.5 Seventh Model: Adding Zipcodes
12.6 Eighth Model: Latitude and Closest Hospital Distance
13 Conclusion
14 Recommendations:
14.0.1 Visualizing Schools

Observations: the model is currently predicting at above 50% accuracy. It's already a turning point, it's getting it right more than it's getting it wrong. However, it's way too high for only having three features, this must be a sign that the model is overfitting.

In [165]: `high_p_vals(baseline_model)`

```
const has an acceptable p_value
grade has an acceptable p_value
bathrooms has an acceptable p_value
lat has an acceptable p_value
```

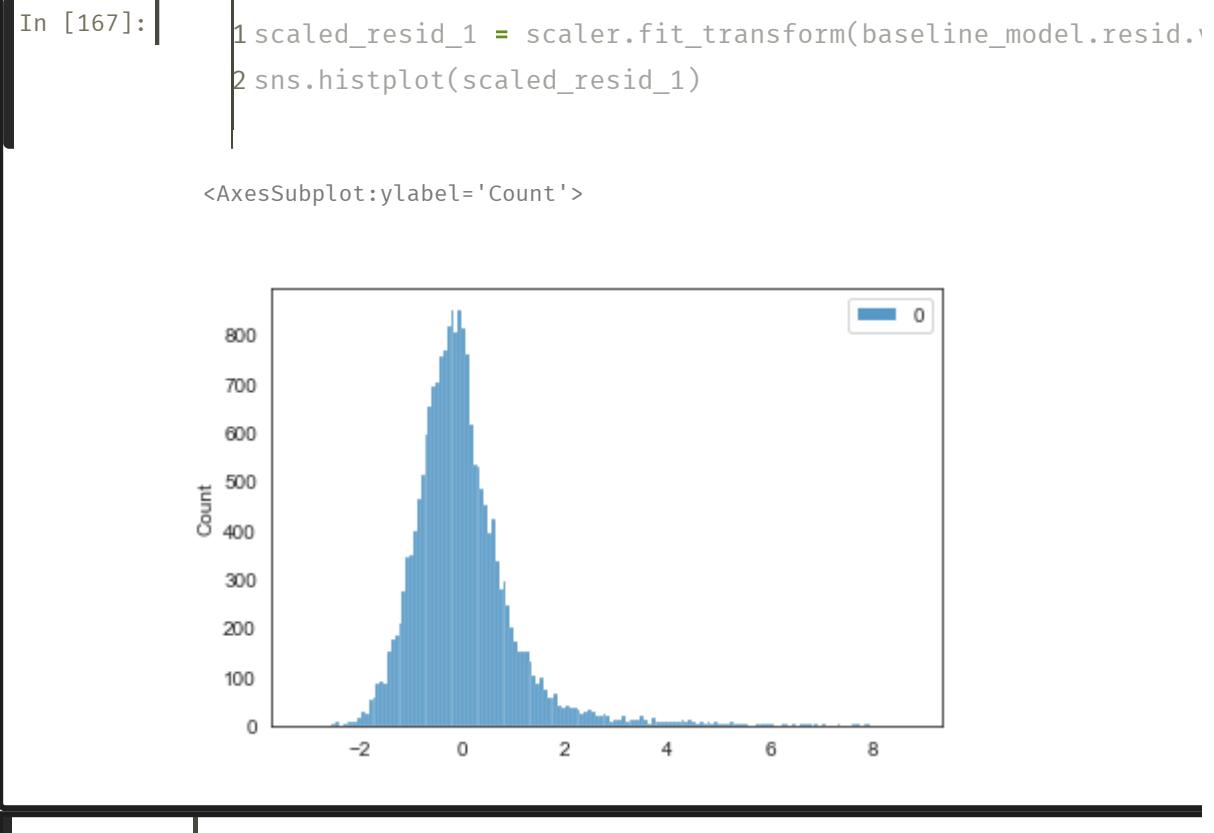
'drop [] for the next model due to high p\_value(s)'

All the P values look good to continue.

In [166]: `scaler = StandardScaler()`

## Contents ⚙

- 6.2 Visualizing the Comparison
- ▼ 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- ▼ 7 Distance Data
  - ▼ 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - ▼ 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- ▼ 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- ▼ 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- ▼ 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- ▼ 14 Recommendations:
  - 14.0.1 Visualizing Schools

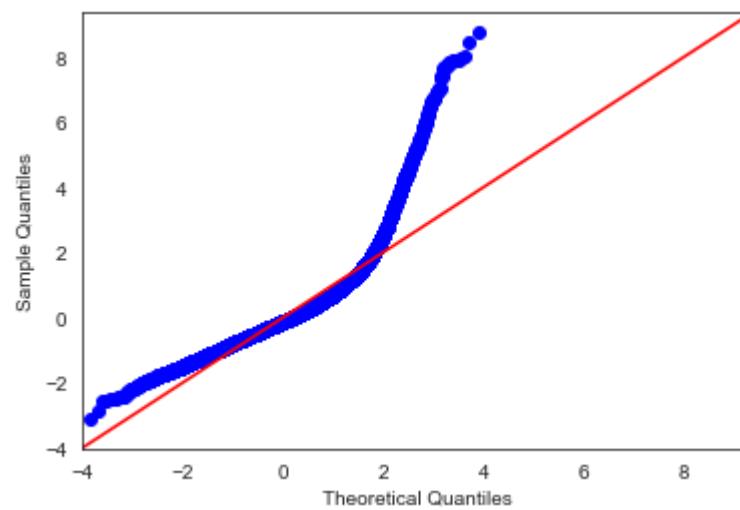
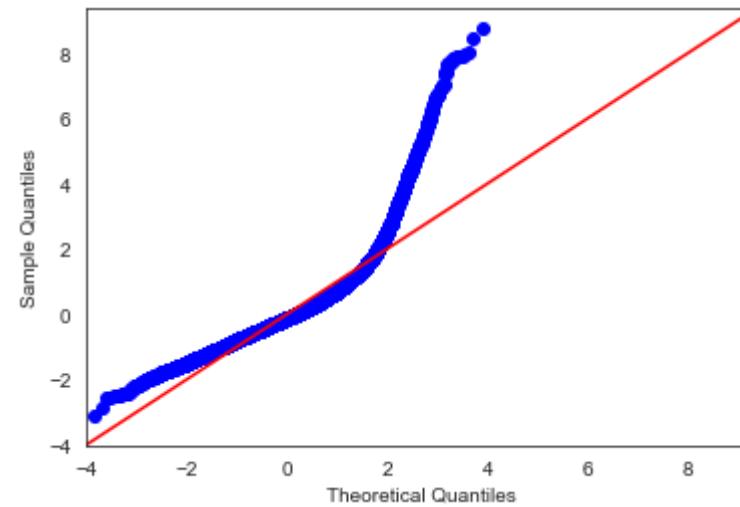


## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [168]:

```
1 sm.qqplot(baseline_model.resid, dist = stats.norm, line = "s")
```



## Contents ⚙️

6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
▼ 7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
▼ 7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Now
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest Hospital
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

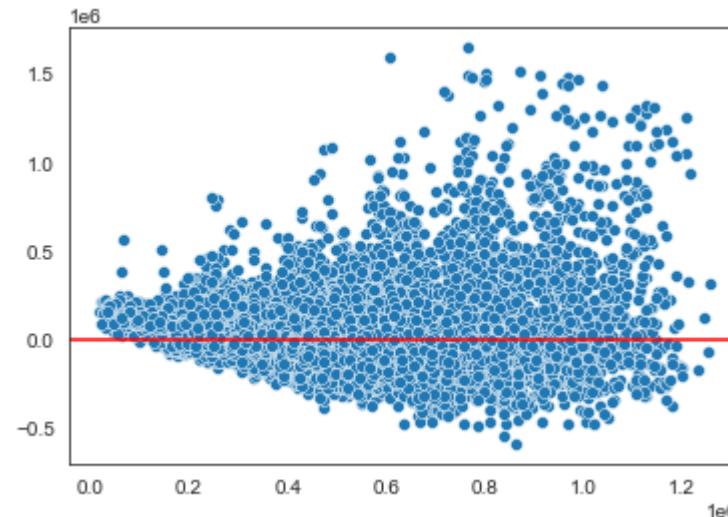
Observations: the blue line curves up sharply from the red line at larger deviations from the mean, representing the long tail we saw in the distribution.

In [169]:

```
1 sns.scatterplot(baseline_model.predict(sm.add_constant(X)),  
2 plt.axhline(0, color = 'red')
```

```
/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/  
6: FutureWarning: Pass the following variables as keyword args: x, y. From  
valid positional argument will be `data`, and passing other arguments with  
will result in an error or misinterpretation.  
warnings.warn(
```

<matplotlib.lines.Line2D at 0x7ff364ab0ac0>



## Contents

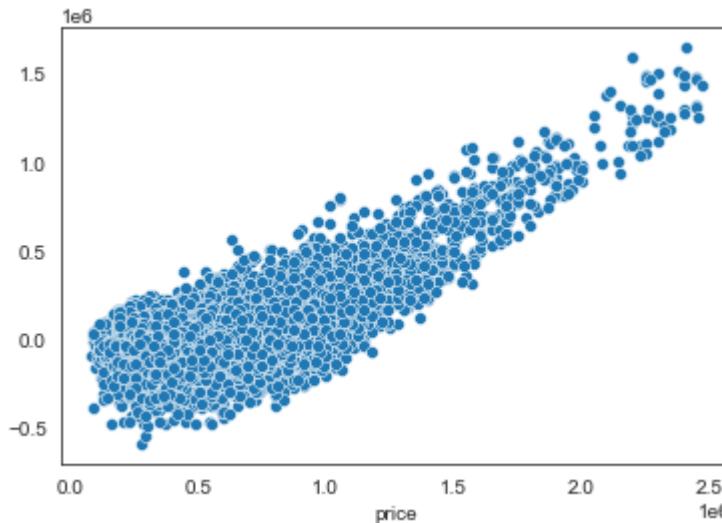
- [6.2 Visualizing the Comparison](#)
- [6.3 Home History](#)
  - [6.3.1 Age of Home](#)
  - [6.3.2 Year Modified](#)
- [7 Distance Data](#)
  - [7.1 Getting Infrastructure Data](#)
    - [7.1.1 Downtown](#)
    - [7.1.2 Hospitals](#)
    - [7.1.3 Police Stations](#)
    - [7.1.4 Schools](#)
  - [7.2 Finding Distance for Infrastructure](#)
  - [7.3 Exploring Distance Data](#)
    - [7.3.1 Downtown](#)
    - [7.3.2 Hospitals](#)
    - [7.3.3 Police Stations](#)
    - [7.3.4 Schools](#)
- [8 Preparing for Modeling](#)
  - [8.0.1 Dropping View and Waterfront Properties for Modeling](#)
- [9 Zipcode](#)
- [10 Seeing the Correlation](#)
- [11 Building a model](#)
  - [11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude](#)
  - [11.2 Second Model: Transforming the Price with Log](#)
- [12 Improving the Model with Adding Features](#)
  - [12.1 Third Model: Bedrooms, Floors, Closest School](#)
  - [12.2 Fourth Model: Square Foot Basement, Closest School](#)
  - [12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods](#)
  - [12.4 Sixth Model: Condition, Waterfront and View](#)
  - [12.5 Seventh Model: Adding Zipcodes](#)
  - [12.6 Eighth Model: Latitude and Closest Hospital Distance](#)
- [13 Conclusion](#)
- [14 Recommendations:](#)
  - [14.0.1 Visualizing Schools](#)

Observations: There is a cone shape on the left side of this plot, indicating heteroscedasticity. This is to say that there isn't homogeneity of variance for the next model.

In [170]:

```
1 sns.scatterplot(y, baseline_model.resid);
```

```
/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/
6: FutureWarning: Pass the following variables as keyword args: x, y. From
valid positional argument will be `data`, and passing other arguments with
will result in an error or misinterpretation.
warnings.warn(
```



Observations: This is a different way to see the homogeneity of the variance. In this visual doesn't tell us much.



## 11.2 Second Model: Transforming the Price with Log

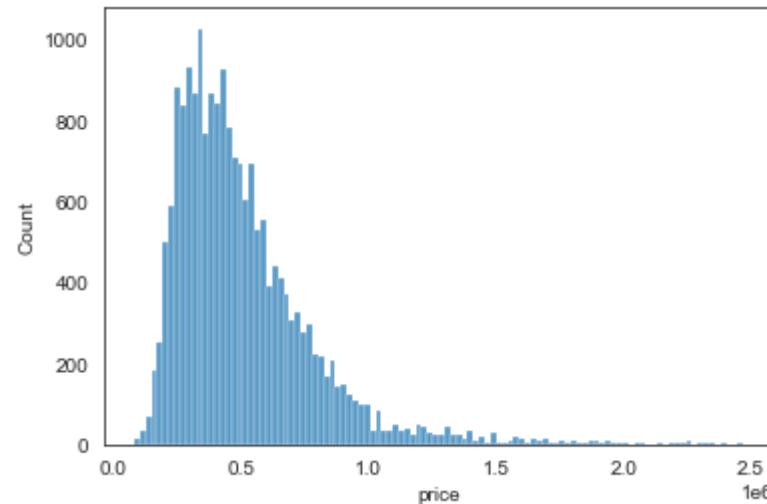
## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [171]:

```
1 sns.histplot(zipcode_data['price'])
```

<AxesSubplot:xlabel='price', ylabel='Count'>



Price of home does not have a normal distribution in this data set. There are more lower priced homes and a much longer tail on the right hand side than the lack of upward limit on home price.

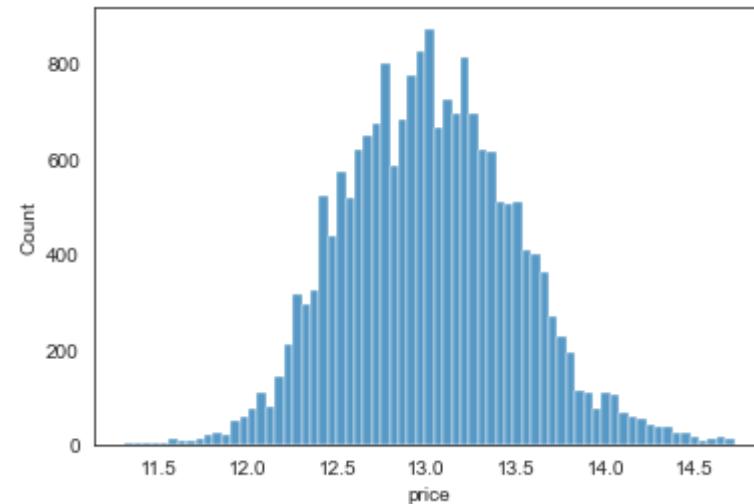
## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [172]:

```
1 sns.histplot(np.log(zipcode_data['price']))
```

&lt;AxesSubplot:xlabel='price', ylabel='Count'&gt;



The log of the price however, does have a much more normal distribution than the raw price. This will be important for all future models.

**NOTE:** I'll now be interpreting the price by percentage instead of raw amount.

## Contents ⚙️

- Comparing Neighborhood Homes with One Another
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

```
In [173]: 1 zipcode_data['price_log'] = np.log(zipcode_data['price'])
2
3 # making a new column in the data for the log of the price
```

```
In [174]: 1 price_log = zipcode_data['price_log']
```

```
In [175]: 1 features_2 = zipcode_data.loc[:,['price_log', 'grade', 'bathrooms']]
2 features_2.info()
3
4 # making the dataframe for the second model
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 19414 entries, 0 to 21596
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype  
--- 
 0   price_log    19414 non-null   float64
 1   grade        19414 non-null   int64  
 2   bathrooms    19414 non-null   float64
 3   lat          19414 non-null   float64
dtypes: float64(3), int64(1)
memory usage: 758.4 KB
```

```
In [176]: 1 y = features_2['price_log']
2 X = features_2.drop(['price_log'], axis = 1)
3
4 # labeling the axes
```

## Contents ⚙️

6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Now
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Zipcode
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest Hospital
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

```
In [177]: 1 model_2 = sm.OLS(y, sm.add_constant(X)).fit()
2 model_2.summary()
3
4 # making the model
```

### OLS Regression Results

Dep. Variable: price\_log R-squared: 0.631  
 Model: OLS Adj. R-squared: 0.630  
 Method: Least Squares F-statistic: 1.104e+04  
 Date: Tue, 12 Apr 2022 Prob (F-statistic): 0.00  
 Time: 17:54:49 Log-Likelihood: -3753.2  
 No. Observations: 19414 AIC: 7514.  
 Df Residuals: 19410 BIC: 7546.  
 Df Model: 3  
 Covariance Type: nonrobust

	coef	std err	t	P> t	[0.025	0.975]
const	-61.9788	0.746	-83.051	0.000	-63.442	-60.516
grade	0.2313	0.003	89.137	0.000	0.226	0.236
bathrooms	0.1188	0.004	31.870	0.000	0.111	0.126
lat	1.5491	0.016	98.632	0.000	1.518	1.580
Omnibus:	441.832	Durbin-Watson:	1.973			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	590.661			
Skew:	0.279	Prob(JB):	5.49e-129			
Kurtosis:	3.647	Cond. No.	1.69e+04			

### Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified

[2] The condition number is large, 1.69e+04. This might indicate that there is strong multicollinearity or other numerical problems.

## Contents ⚙

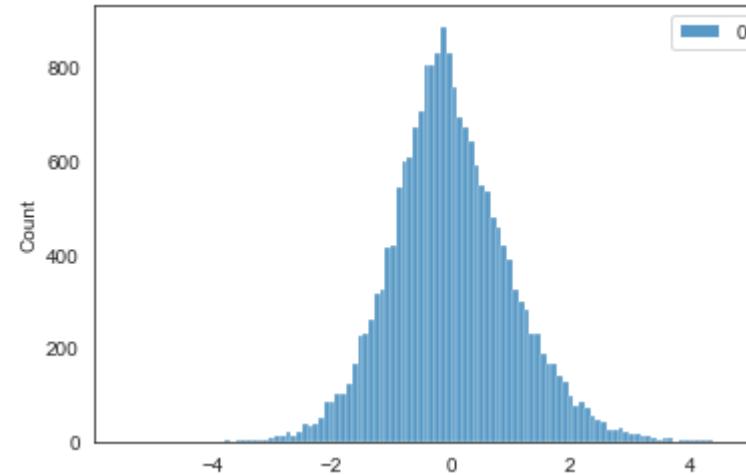
6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
▼ 7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
▼ 7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Now
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest Hospital
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

Observations: the kurtosis is much lower now, and we've improved our model by about 10%.

In [178]:

```
1 scaled_resid_2 = scaler.fit_transform(model_2.resid.values)
2 sns.histplot(scaled_resid_2)
```

<AxesSubplot:ylabel='Count'>



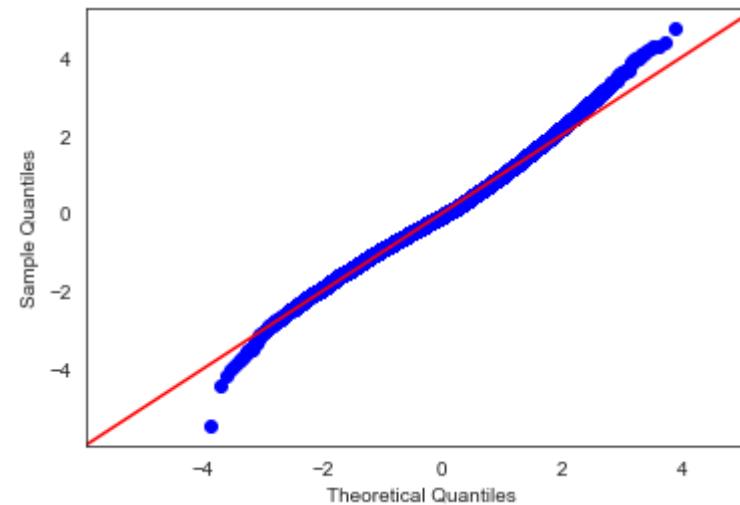
Observations: these residuals fit a normal distribution much better than the previous one.

## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

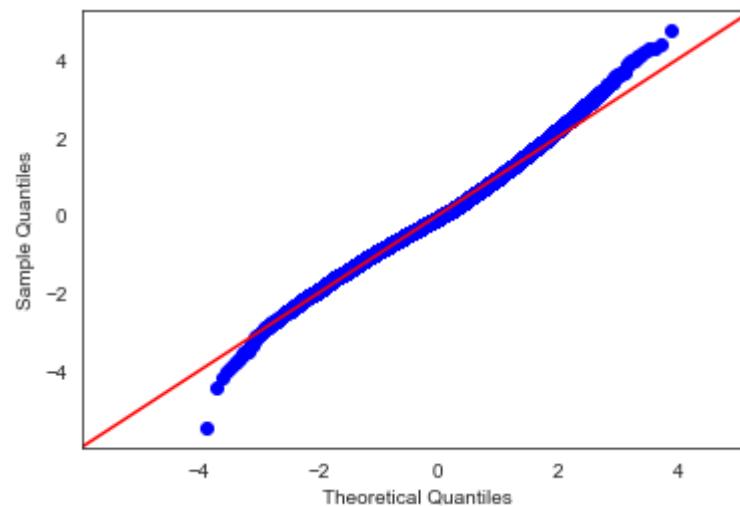
In [179]:

```
1 sm.qqplot(model_2.resid, dist = stats.norm, line = "45", fi
```



## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools



The blue line is much closer to the red line, representing the more normal distribution.

## Contents ⚙️

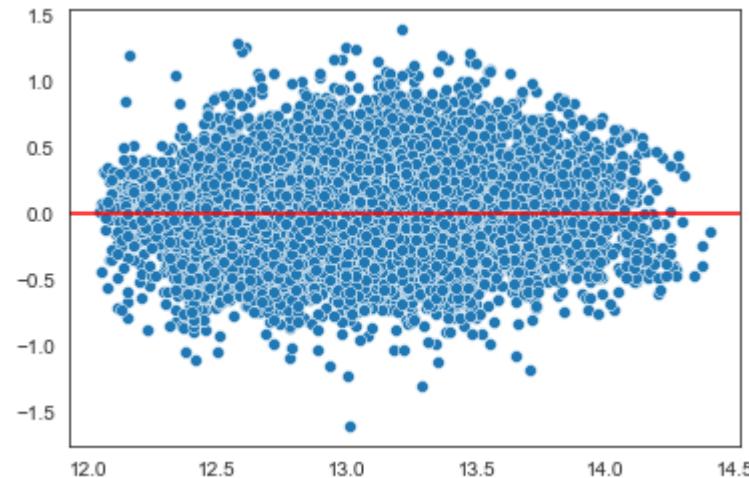
6.2	Visualizing the Comparison
6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Now
9	Zipcode
10	Seeing the Correlation
11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest Hospital
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
14	Recommendations:
14.0.1	Visualizing Schools

In [180]:

```
1 sns.scatterplot(model_2.predict(sm.add_constant(X)), model_2.y)
2 plt.axhline(0, color = 'red')
```

/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/sklearn/utils/\_param\_validation.py:6: FutureWarning: Pass the following variables as keyword args: x, y. From version 1.4, valid positional argument will be `data`, and passing other arguments with `x` will result in an error or misinterpretation.  
warnings.warn(  
 "Pass the following variables as keyword args: x, y. From version 1.4, valid positional argument will be `data`, and passing other arguments with `x` will result in an error or misinterpretation.", FutureWarning)

&lt;matplotlib.lines.Line2D at 0x7ff364b2ab20&gt;



Observations: the cone shape is gone now on the left side, representing a pattern that the model isn't picking up.

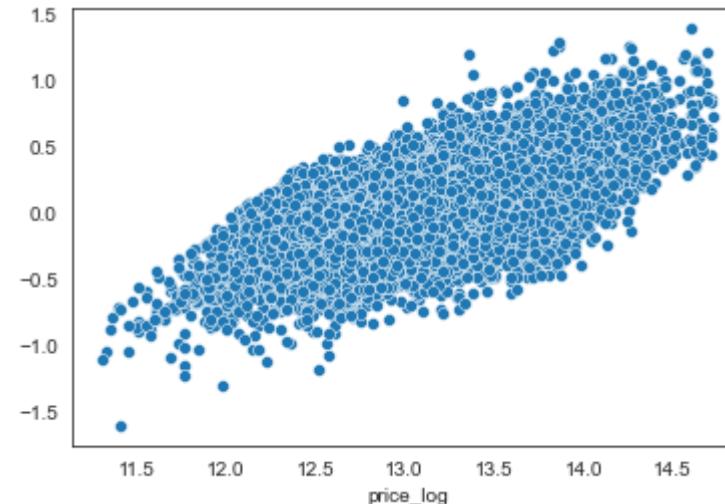
## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [181]:

```
1 sns.scatterplot(y, model_2.resid);
```

```
/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/
6: FutureWarning: Pass the following variables as keyword args: x, y. From
valid positional argument will be `data`, and passing other arguments with
will result in an error or misinterpretation.
warnings.warn()
```



Observations: the points are now a bit more scattered, which is what we want.

## 12 Improving the Model with Adding Features

### 12.1 Third Model: Bedrooms, Floors, Closest School Distance

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, and Longitude
    - 11.2 Second Model: Transforming the Price with Logarithms
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

```
In [182]: 1 next_features(order, features_to_remove)

['bedrooms', 'floors', 'closest_school_distance']

In [183]: 1 features_3 = ['bedrooms', 'floors', 'closest_school_distance']

In [184]: 1 X = pd.concat([X, zipcode_data[features_3]], axis = 1)

In [185]: 1 X.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 19414 entries, 0 to 21596
Data columns (total 6 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   grade            19414 non-null   int64  
 1   bathrooms        19414 non-null   float64 
 2   lat              19414 non-null   float64 
 3   bedrooms         19414 non-null   int64  
 4   floors            19414 non-null   float64 
 5   closest_school_distance  19414 non-null   float64 
dtypes: float64(4), int64(2)
memory usage: 1.0 MB
```

## Contents ⚙️

6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
▼ 7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
▼ 7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Now
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest School
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighbors
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

```
In [186]: 1 model_3 = sm.OLS(y, sm.add_constant(X)).fit()
2 model_3.summary()
```

### OLS Regression Results

Dep. Variable:	price_log	R-squared:	0.653			
Model:	OLS	Adj. R-squared:	0.653			
Method:	Least Squares	F-statistic:	6079.			
Date:	Tue, 12 Apr 2022	Prob (F-statistic):	0.00			
Time:	17:54:50	Log-Likelihood:	-3152.5			
No. Observations:	19414	AIC:	6319.			
Df Residuals:	19407	BIC:	6374.			
Df Model:	6					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	-50.0830	0.872	-57.451	0.000	-51.792	-48.374
grade	0.2437	0.003	93.661	0.000	0.239	0.249
bathrooms	0.1078	0.004	25.545	0.000	0.100	0.116
lat	1.2971	0.018	70.756	0.000	1.261	1.333
bedrooms	0.0554	0.003	20.328	0.000	0.050	0.061
floors	-0.0427	0.005	-9.442	0.000	-0.052	-0.034
closest_school_distance	-0.0127	0.000	-26.354	0.000	-0.014	-0.012
Omnibus:	292.743	Durbin-Watson:	1.979			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	433.905			
Skew:	0.172	Prob(JB):	6.01e-95			
Kurtosis:	3.647	Cond. No.	2.06e+04			

## Contents ⚙️

6.2 Visualizing the Comparison
6.3 Home History
6.3.1 Age of Home
6.3.2 Year Modified
7 Distance Data
7.1 Getting Infrastructure Data
7.1.1 Downtown
7.1.2 Hospitals
7.1.3 Police Stations
7.1.4 Schools
7.2 Finding Distance for Infrastructure
7.3 Exploring Distance Data
7.3.1 Downtown
7.3.2 Hospitals
7.3.3 Police Stations
7.3.4 Schools
8 Preparing for Modeling
8.0.1 Dropping View and Waterfront Properties for Model
9 Zipcode
10 Seeing the Correlation
11 Building a model
11.1 Baseline Model: Grade, Bathrooms, Latitude, and Zipcode
11.2 Second Model: Transforming the Price with Logarithms
12 Improving the Model with Adding Features
12.1 Third Model: Bedrooms, Floors, Closest School, and Zipcode
12.2 Fourth Model: Square Foot Basement, Closest School, and Zipcode
12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4 Sixth Model: Condition, Waterfront and View
12.5 Seventh Model: Adding Zipcodes
12.6 Eighth Model: Latitude and Closest Hospital Distance
13 Conclusion
14 Recommendations:
14.0.1 Visualizing Schools

### Notes:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 2.06e+04. This might indicate that there is strong multicollinearity or other numerical problems.

Observations: adding these three features didn't make the accuracy go up by 3%.

### FURTHER WORK:

I have not looked too deeply into floors as I have with some other features. In this model, it's predicting that as floors decrease, the price rises. This is counterintuitive, and bears further investigation.

In [187]:

```
1 high_p_vals(model_3)
```

```
const has an acceptable p_value
grade has an acceptable p_value
bathrooms has an acceptable p_value
lat has an acceptable p_value
bedrooms has an acceptable p_value
floors has an acceptable p_value
closest_school_distance has an acceptable p_value
```

```
'drop [] for the next model due to high p_value(s)'
```

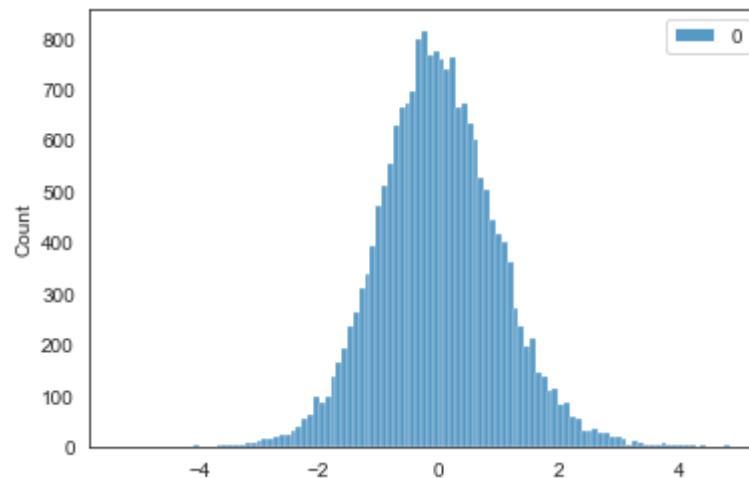
## Contents ⚙

- 6.2 Visualizing the Comparison
- ▼ 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- ▼ 7 Distance Data
  - ▼ 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - ▼ 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- ▼ 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- ▼ 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- ▼ 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- ▼ 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [188]:

```
1 scaled_resid_3 = scaler.fit_transform(model_3.resid.values)
2 sns.histplot(scaled_resid_3)
```

&lt;AxesSubplot:ylabel='Count'&gt;

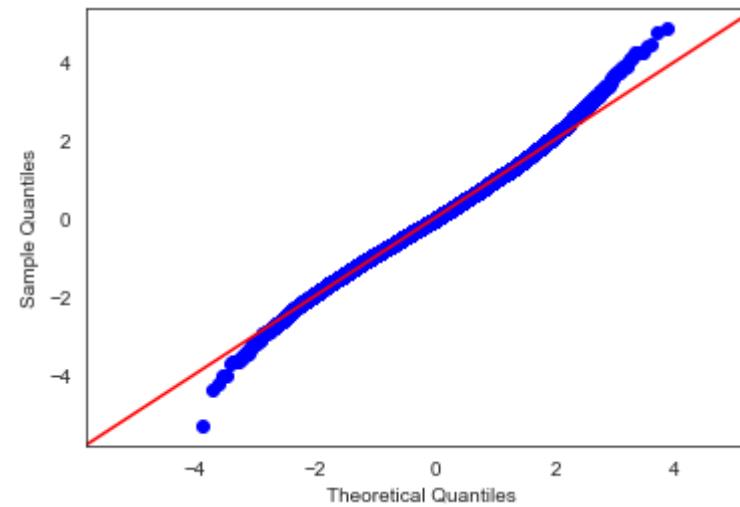


## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

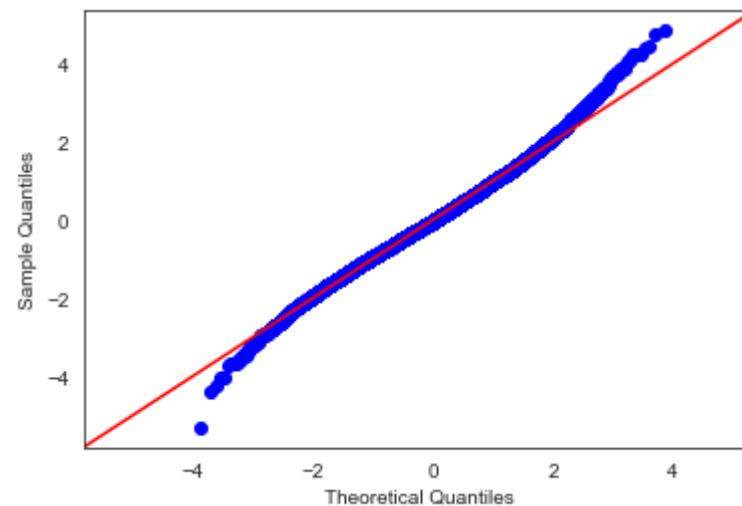
In [189]:

```
1 sm.qqplot(model_3.resid, dist = stats.norm, line = "45", fi
```



## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools



## Contents ⚙

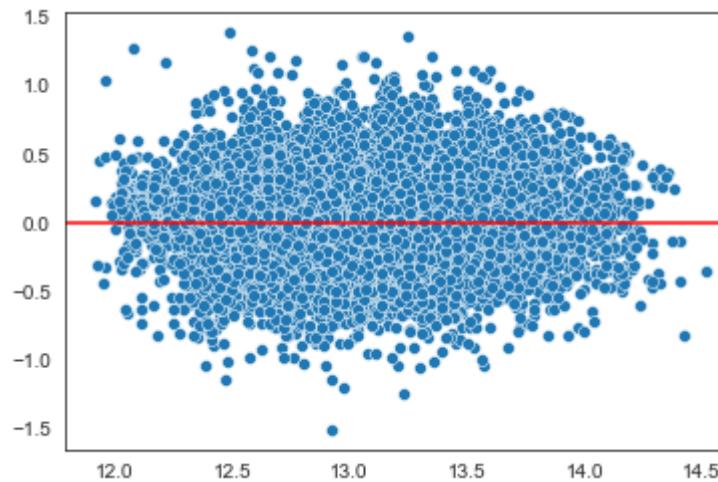
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [190]:

```
1 sns.scatterplot(model_3.predict(sm.add_constant(X)), model_3.y)
```

```
/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/
6: FutureWarning: Pass the following variables as keyword args: x, y. From
valid positional argument will be `data`, and passing other arguments with
will result in an error or misinterpretation.
warnings.warn(
```

&lt;matplotlib.lines.Line2D at 0x7ff36599f490&gt;



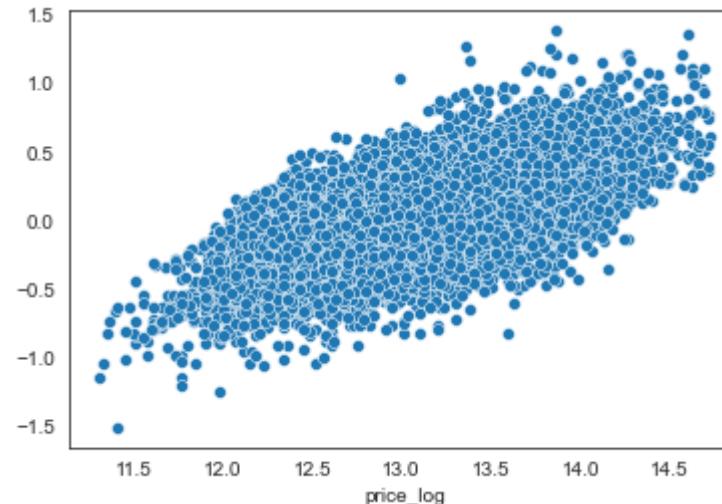
## Contents ⚙️

- 6.1 Comparing Renovated Homes with Unrenovated
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

In [191]:

```
1 sns.scatterplot(y, model_3.resid);
```

```
/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/
6: FutureWarning: Pass the following variables as keyword args: x, y. From
valid positional argument will be `data`, and passing other arguments with
will result in an error or misinterpretation.
warnings.warn(
```



Observations: these plots show that the model is functioning properly. The lack of cone shapes in the scatterplots and the blue line being straight indicate a good fit.

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, and Longitude
    - 11.2 Second Model: Transforming the Price with Logarithms
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

on the QQ plot.

## 12.2 Fourth Model: Square Foot Basement, Closest Hospital Distance and Year Renovated

```
In [192]: 1 next_features(order, features_to_remove)
```

```
['sqft_basement', 'closest_hospital_distance', 'yr_renovated']
```

```
In [193]: 1 features_4 = ['sqft_basement', 'closest_hospital_distance',
```

```
In [194]: 1 X = pd.concat([X, zipcode_data[features_4]], axis = 1)
2 X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
Int64Index: 19414 entries, 0 to 21596
```

```
Data columns (total 9 columns):
```

#	Column	Non-Null Count	Dtype
0	grade	19414	non-null
1	bathrooms	19414	non-null
2	lat	19414	non-null
3	bedrooms	19414	non-null
4	floors	19414	non-null
5	closest_school_distance	19414	non-null
6	sqft_basement	19414	non-null
7	closest_hospital_distance	19414	non-null
8	yr_renovated	19414	non-null

```
dtypes: float64(7), int64(2)
```

```
memory usage: 1.5 MB
```

## Contents ⚙️

6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Now
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Zipcode
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest Hospital
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

```
In [195]: 1 model_4 = sm.OLS(y, sm.add_constant(X)).fit()
2 model_4.summary()
```

### OLS Regression Results

```
Dep. Variable: price_log R-squared: 0.661
Model: OLS Adj. R-squared: 0.661
Method: Least Squares F-statistic: 4213.
Date: Tue, 12 Apr 2022 Prob (F-statistic): 0.00
Time: 17:54:50 Log-Likelihood: -2903.6
No. Observations: 19414 AIC: 5827.
Df Residuals: 19404 BIC: 5906.
Df Model: 9
Covariance Type: nonrobust
coef std err t P>|t| [0.025 0.975]
const -51.2004 0.863 -59.347 0.000 -52.891 -49.509
grade 0.2446 0.007 35.185 0.000 0.240 0.250
```

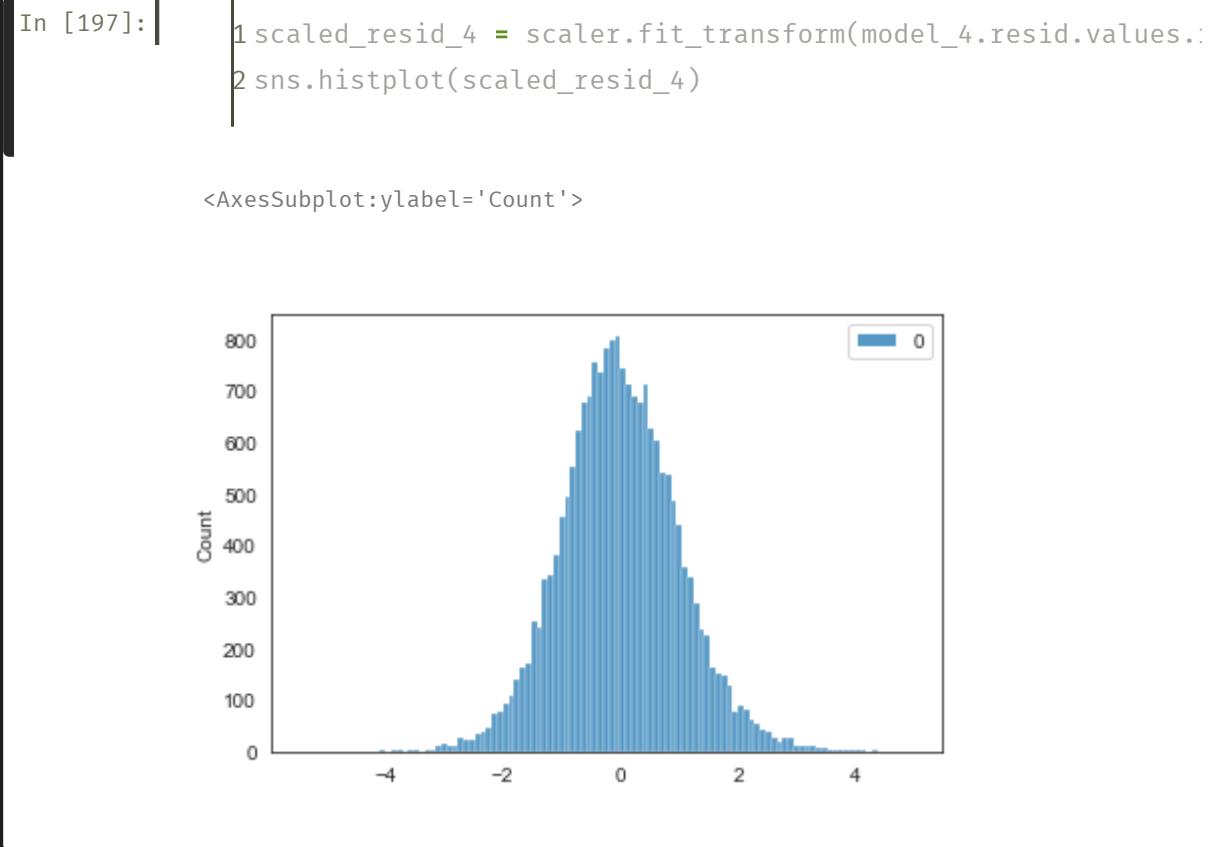
```
In [196]: 1 high_p_vals(model_4)
```

```
const has an acceptable p_value
grade has an acceptable p_value
bathrooms has an acceptable p_value
lat has an acceptable p_value
bedrooms has an acceptable p_value
floors has an acceptable p_value
closest_school_distance has an acceptable p_value
sqft_basement has an acceptable p_value
closest_hospital_distance has an acceptable p_value
yr_renovated has an acceptable p_value
```

```
'drop [] for the next model due to high p_value(s)'
```

## Contents ⚙

- 6.2 Visualizing the Comparison
- ▼ 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- ▼ 7 Distance Data
  - ▼ 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - ▼ 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- ▼ 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- ▼ 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- ▼ 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- ▼ 14 Recommendations:
  - 14.0.1 Visualizing Schools

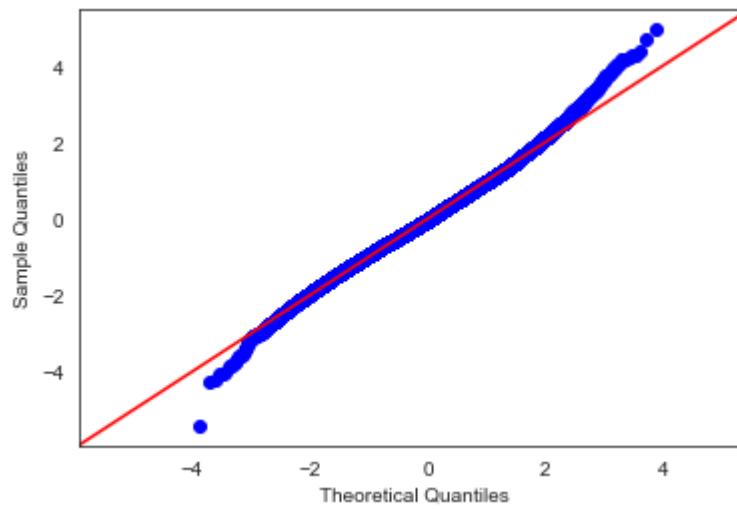
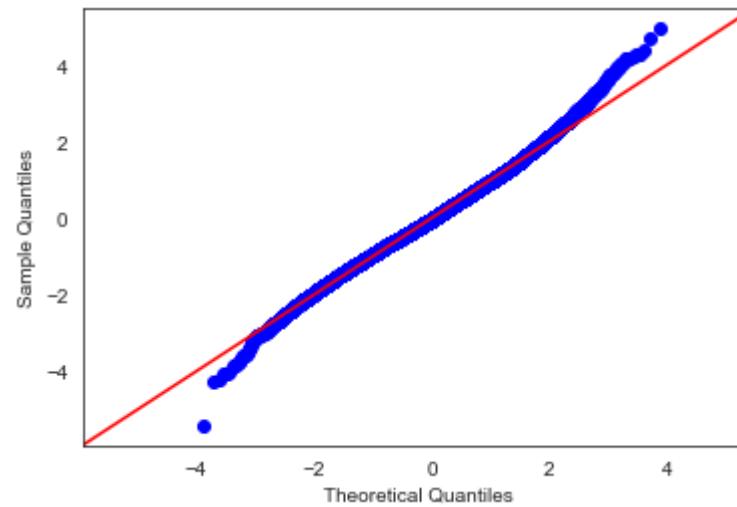


## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [198]:

```
1 sm.qqplot(model_4.resid, dist = stats.norm, line = "45", fi
```



## Contents ⚙️

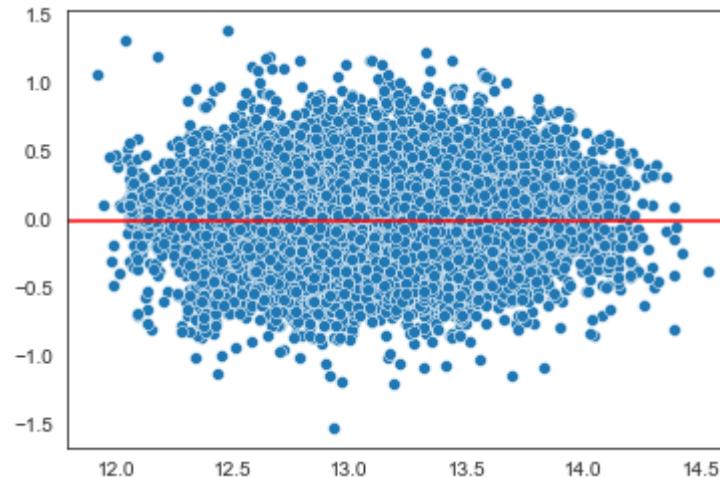
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

In [199]:

```
1 sns.scatterplot(model_4.predict(sm.add_constant(X)), model_
2 plt.axhline(0, color = 'red')
```

/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/sklearn/utils/\_param\_validation.py:6: FutureWarning: Pass the following variables as keyword args: x, y. From version 21.0, valid positional argument will be `data`, and passing other arguments with `x` or `y` will result in an error or misinterpretation.  
warnings.warn(  
 "Pass the following variables as keyword args: x, y. From version 21.0, valid positional argument will be `data`, and passing other arguments with `x` or `y` will result in an error or misinterpretation.",  
 FutureWarning)

&lt;matplotlib.lines.Line2D at 0x7ff365951f10&gt;



Observations: now there is a slight cone shape on the right side. I will not do anything to fix it for the next model.

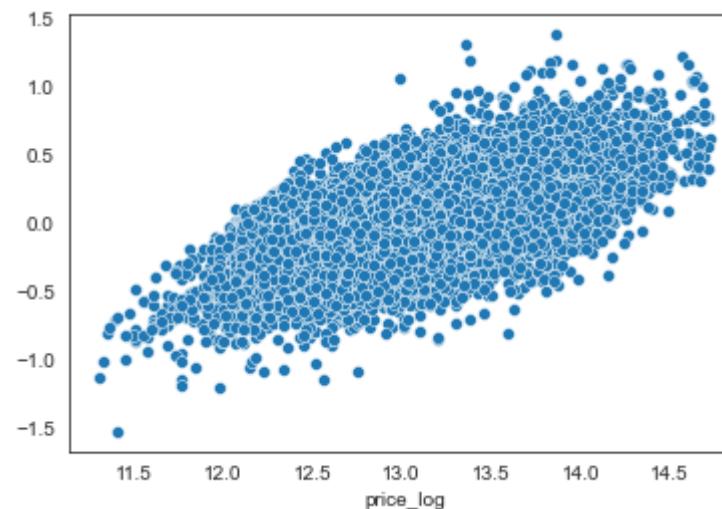
## Contents ⚙️

- 6.1 Comparing Renovated Homes with Unrenovated
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

In [200]:

```
1 sns.scatterplot(y, model_4.resid);
```

```
/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/
6: FutureWarning: Pass the following variables as keyword args: x, y. From
valid positional argument will be `data`, and passing other arguments with
will result in an error or misinterpretation.
warnings.warn(
```



Observations:



**12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors, Longitude and Age of Home**

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

```
In [201]: 1
           2 next_features(order, features_to_remove)

           ['sqft_lot15', 'long', 'age_of_home']

In [202]: 1 features_5 = ['sqft_lot15', 'long', 'age_of_home']

In [203]: 1 X = pd.concat([X, zipcode_data[features_5]], axis = 1)
           2 X.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 19414 entries, 0 to 21596
Data columns (total 12 columns):
 #   Column           Non-Null Count Dtype  
--- 
 0   grade            19414 non-null  int64   
 1   bathrooms        19414 non-null  float64 
 2   lat              19414 non-null  float64 
 3   bedrooms         19414 non-null  int64   
 4   floors            19414 non-null  float64 
 5   closest_school_distance  19414 non-null  float64 
 6   sqft_basement     19414 non-null  float64 
 7   closest_hospital_distance  19414 non-null  float64 
 8   yr_renovated      19414 non-null  float64 
 9   sqft_lot15         19414 non-null  int64   
 10  long              19414 non-null  float64 
 11  age_of_home        19414 non-null  int64   

dtypes: float64(8), int64(4)
memory usage: 1.9 MB
```

## Contents ⚙️

6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Now
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest Hospital
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

```
In [204]: 1 model_5 = sm.OLS(y, sm.add_constant(X)).fit()
2 model_5.summary()
```

### OLS Regression Results

Dep. Variable:	price_log	R-squared:	0.706			
Model:	OLS	Adj. R-squared:	0.706			
Method:	Least Squares	F-statistic:	3878.			
Date:	Tue, 12 Apr 2022	Prob (F-statistic):	0.00			
Time:	17:54:51	Log-Likelihood:	-1542.8			
No. Observations:	19414	AIC:	3112.			
Df Residuals:	19401	BIC:	3214.			
Df Model:	12					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	46.0883	3.775	12.208	0.000	38.689	53.488
grade	0.2543	0.002	103.909	0.000	0.250	0.259
bathrooms	0.1398	0.004	32.812	0.000	0.131	0.148
lat	1.0569	0.019	54.354	0.000	1.019	1.095
bedrooms	0.0291	0.003	11.393	0.000	0.024	0.034
floors	0.0551	0.005	11.450	0.000	0.046	0.065
closest_school_distance	-0.0140	0.001	-18.848	0.000	-0.015	-0.013
sqft_basement	0.0625	0.005	13.702	0.000	0.054	0.071
closest_hospital_distance	-0.0150	0.001	-13.945	0.000	-0.017	-0.013
yr_renovated	2.387e-05	5.69e-06	4.197	0.000	1.27e-05	3.5e-05
sqft_lot15	1.068e-06	1.66e-07	6.441	0.000	7.43e-07	1.39e-06
long	0.6962	0.026	26.315	0.000	0.644	0.748
age_of_home	0.0045	9.51e-05	47.781	0.000	0.004	0.005

## Contents ⚙️

- 6.2 Visualizing the Comparison
- ▼ 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- ▼ 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - ▼ 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- ▼ 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Model
  - 9 Zipcode
  - 10 Seeing the Correlation
  - ▼ 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, and Age of Home
    - 11.2 Second Model: Transforming the Price with Log
  - ▼ 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School, and Zipcode
    - 12.2 Fourth Model: Square Foot Basement, Closest Hospital, and Zipcode
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - ▼ 14 Recommendations:
    - 14.0.1 Visualizing Schools

Omnibus:	363.810	Durbin-Watson:	1.991
Prob(Omnibus):	0.000	Jarque-Bera (JB):	726.252
Skew:	0.083	Prob(JB):	1.98e-158
Kurtosis:	3.933	Cond. No.	3.11e+07

### Notes:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 3.11e+07. This might indicate that there is strong multicollinearity or other numerical problems.

In [205]:

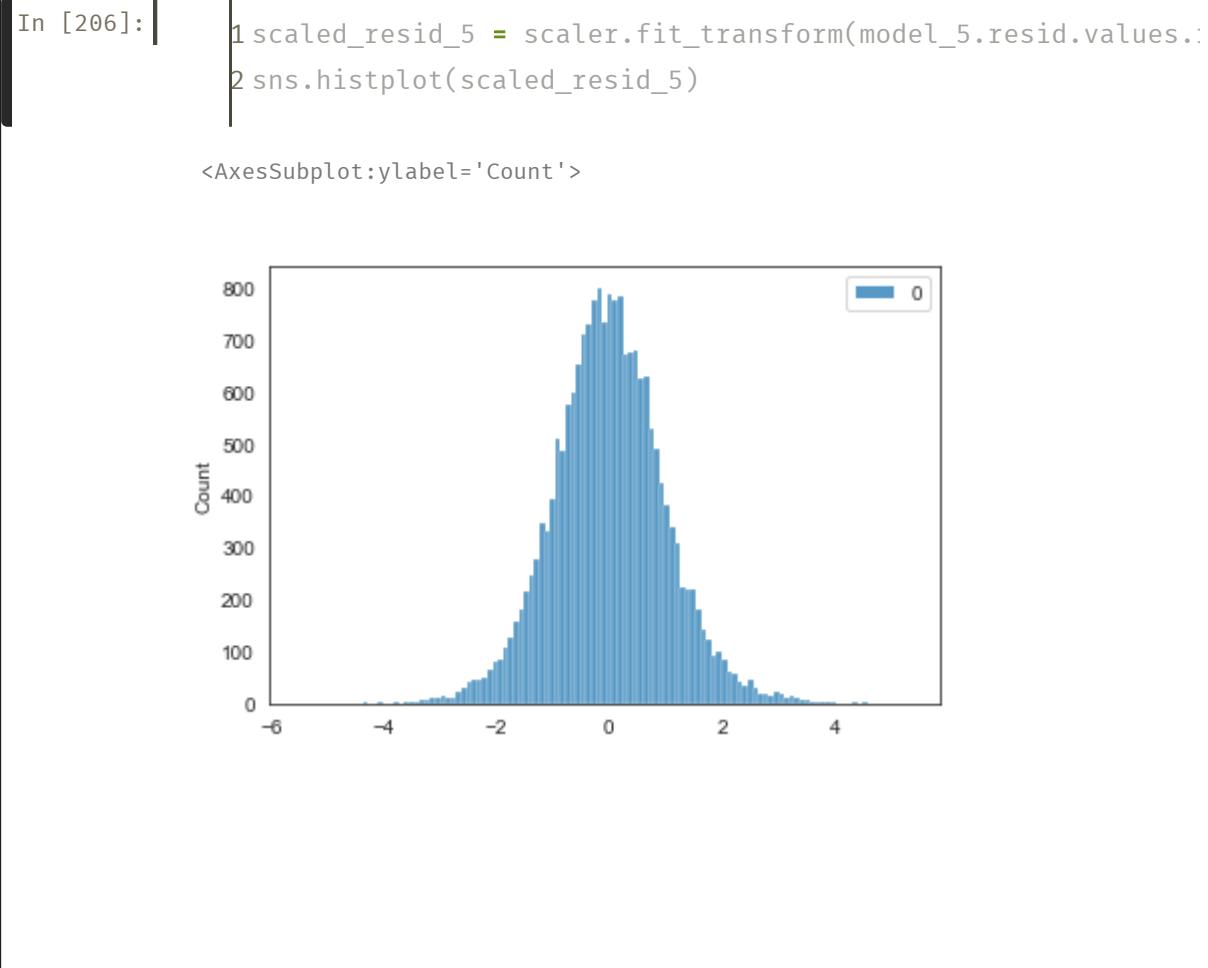
```
1 high_p_vals(model_5)
```

```
const has an acceptable p_value
grade has an acceptable p_value
bathrooms has an acceptable p_value
lat has an acceptable p_value
bedrooms has an acceptable p_value
floors has an acceptable p_value
closest_school_distance has an acceptable p_value
sqft_basement has an acceptable p_value
closest_hospital_distance has an acceptable p_value
yr_renovated has an acceptable p_value
sqft_lot15 has an acceptable p_value
long has an acceptable p_value
age_of_home has an acceptable p_value
```

'drop [] for the next model due to high p\_value(s)'

## Contents ⚙

- 6.2 Visualizing the Comparison
- ▼ 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- ▼ 7 Distance Data
  - ▼ 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - ▼ 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- ▼ 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- ▼ 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- ▼ 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- ▼ 14 Recommendations:
  - 14.0.1 Visualizing Schools

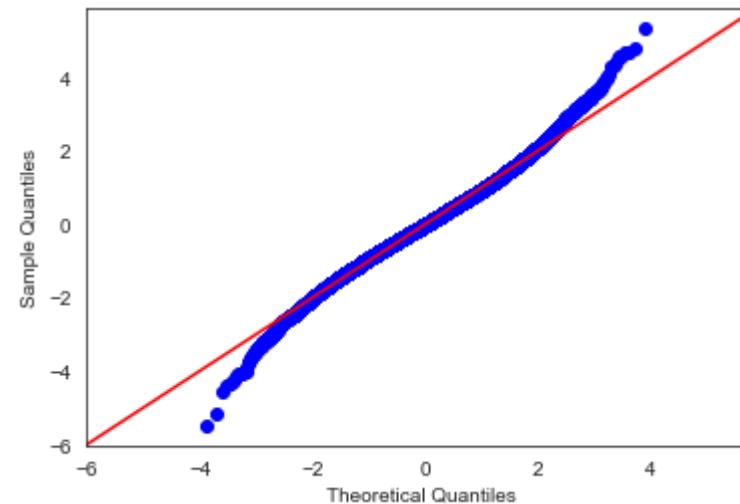


## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

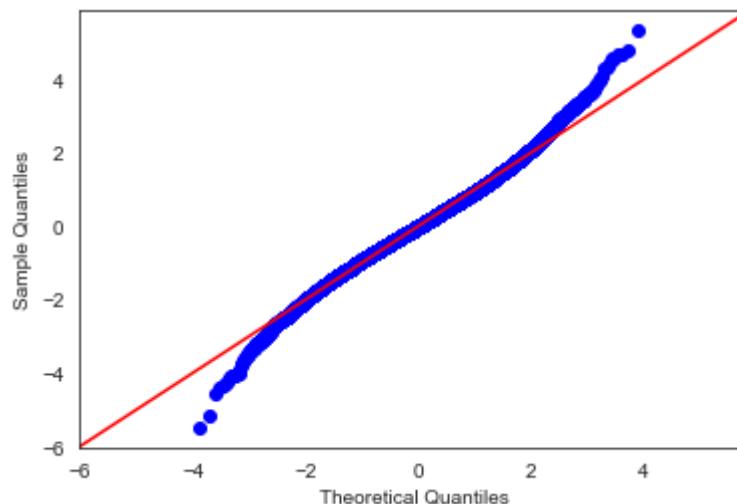
In [207]:

```
1 sm.qqplot(model_5.resid, dist = stats.norm, line = "45", fi
```



## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Zipcode
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools



## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

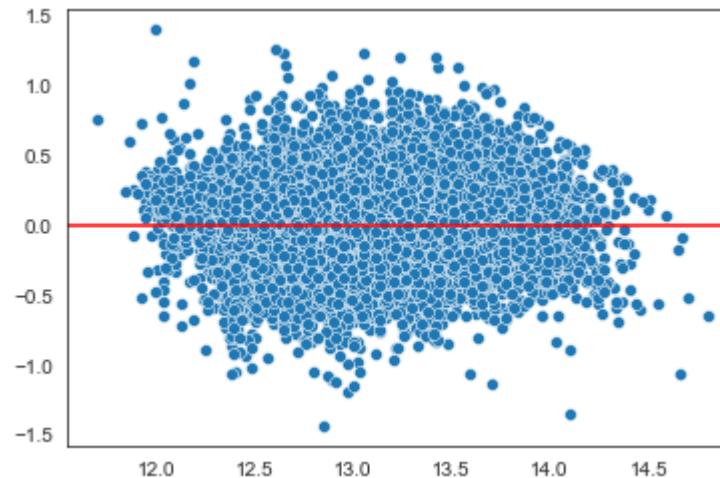
In [208]:

```
1 sns.scatterplot(model_5.predict(sm.add_constant(X)), model_5.y)
```

```
2 plt.axhline(0, color = 'red')

/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/
6: FutureWarning: Pass the following variables as keyword args: x, y. From
valid positional argument will be `data`, and passing other arguments with
will result in an error or misinterpretation.
warnings.warn(
```

```
<matplotlib.lines.Line2D at 0x7ff364b3c6a0>
```



## Contents ⚙️

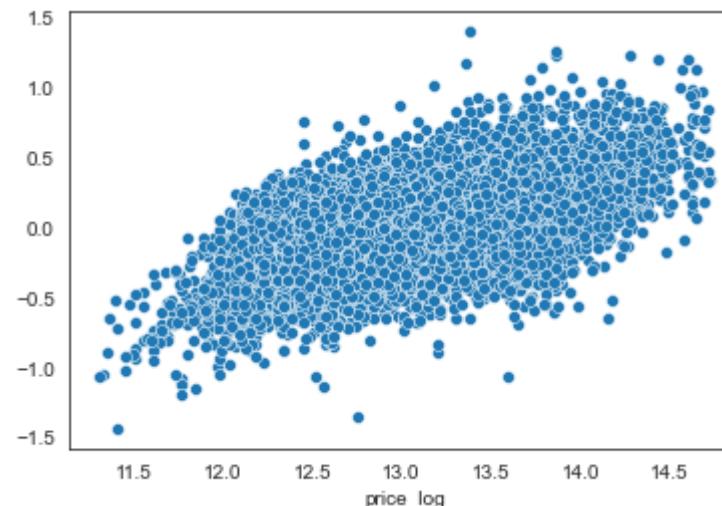
- 6.1 Comparing Renovated Homes with Unrenovated
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Logarithms
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [209]:

```
1 sns.scatterplot(y, model_5.resid)
```

```
/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/
6: FutureWarning: Pass the following variables as keyword args: x, y. From
valid positional argument will be `data`, and passing other arguments with
will result in an error or misinterpretation.
warnings.warn(
```

```
<AxesSubplot:xlabel='price_log'>
```



## 12.4 Sixth Model: Condition, Waterfront View

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, and Longitude
    - 11.2 Second Model: Transforming the Price with Logarithms
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

```
In [210]: 1 next_features(order, features_to_remove)

['condition', 'waterfront', 'view']

In [211]: 1 features_6 = ['condition', 'waterfront', 'view']

In [212]: 1 X = pd.concat([X, zipcode_data[features_6]], axis = 1)
2 X.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 19414 entries, 0 to 21596
Data columns (total 15 columns):
 #   Column           Non-Null Count Dtype  
--- 
 0   grade            19414 non-null  int64   
 1   bathrooms        19414 non-null  float64 
 2   lat              19414 non-null  float64 
 3   bedrooms         19414 non-null  int64   
 4   floors            19414 non-null  float64 
 5   closest_school_distance  19414 non-null  float64 
 6   sqft_basement     19414 non-null  float64 
 7   closest_hospital_distance  19414 non-null  float64 
 8   yr_renovated      19414 non-null  float64 
 9   sqft_lot15        19414 non-null  int64   
 10  long             19414 non-null  float64 
 11  age_of_home       19414 non-null  int64   
 12  condition         19414 non-null  int64   
 13  waterfront        19414 non-null  int64   
 14  view              19414 non-null  float64 

dtypes: float64(9), int64(6)
memory usage: 2.4 MB
```

## Contents ⚙️

6.2	Visualizing the Comparison
6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Now
9	Zipcode
10	Seeing the Correlation
11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest Hospital
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
14	Recommendations:
14.0.1	Visualizing Schools

In [213]:

```
1 model_6 = sm.OLS(y, sm.add_constant(X)).fit()
2 model_6.summary()
```

```
/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/
linear_model.py:1860: RuntimeWarning: divide by zero encountered in double_
return np.sqrt(eigvals[0]/eigvals[-1])
```

### OLS Regression Results

Dep. Variable:	price_log	R-squared:	0.711
Model:	OLS	Adj. R-squared:	0.711
Method:	Least Squares	F-statistic:	3675.
Date:	Tue, 12 Apr 2022	Prob (F-statistic):	0.00
Time:	17:54:51	Log-Likelihood:	-1362.2
No. Observations:	19414	AIC:	2752.
Df Residuals:	19400	BIC:	2863.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	P> t	[0.025	0.975]
const	43.4383	3.743	11.606	0.000	36.102	50.775
grade	0.2550	0.002	105.124	0.000	0.250	0.260
bathrooms	0.1329	0.004	31.358	0.000	0.125	0.141
lat	1.0735	0.019	55.665	0.000	1.036	1.111
bedrooms	0.0275	0.003	10.853	0.000	0.023	0.032
floors	0.0643	0.005	13.422	0.000	0.055	0.074
closest_school_distance	-0.0139	0.001	-18.936	0.000	-0.015	-0.012
sqft_basement	0.0605	0.005	13.389	0.000	0.052	0.069
closest_hospital_distance	-0.0151	0.001	-14.177	0.000	-0.017	-0.013
yr_renovated	4.025e-05	5.7e-06	7.062	0.000	2.91e-05	5.14e-05

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

sqft_lot15	9.948e-07	1.64e-07	6.057	0.000	6.73e-07	1.32e-06
long	0.6819	0.026	26.007	0.000	0.631	0.733
age_of_home	0.0040	9.89e-05	40.225	0.000	0.004	0.004
condition	0.0613	0.003	19.090	0.000	0.055	0.068
waterfront	0	0	nan	nan	0	0
view	0	0	nan	nan	0	0
Omnibus:	365.540	Durbin-Watson:	1.992			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	681.974			
Skew:	0.124	Prob(JB):	8.15e-149			
Kurtosis:	3.884	Cond. No.	inf			

Notes:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The smallest eigenvalue is 0. This might indicate that there are strong multicollinearity problems or that the design matrix is singular.

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Model
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, and Longitude
  - 11.2 Second Model: Transforming the Price with Logarithm
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

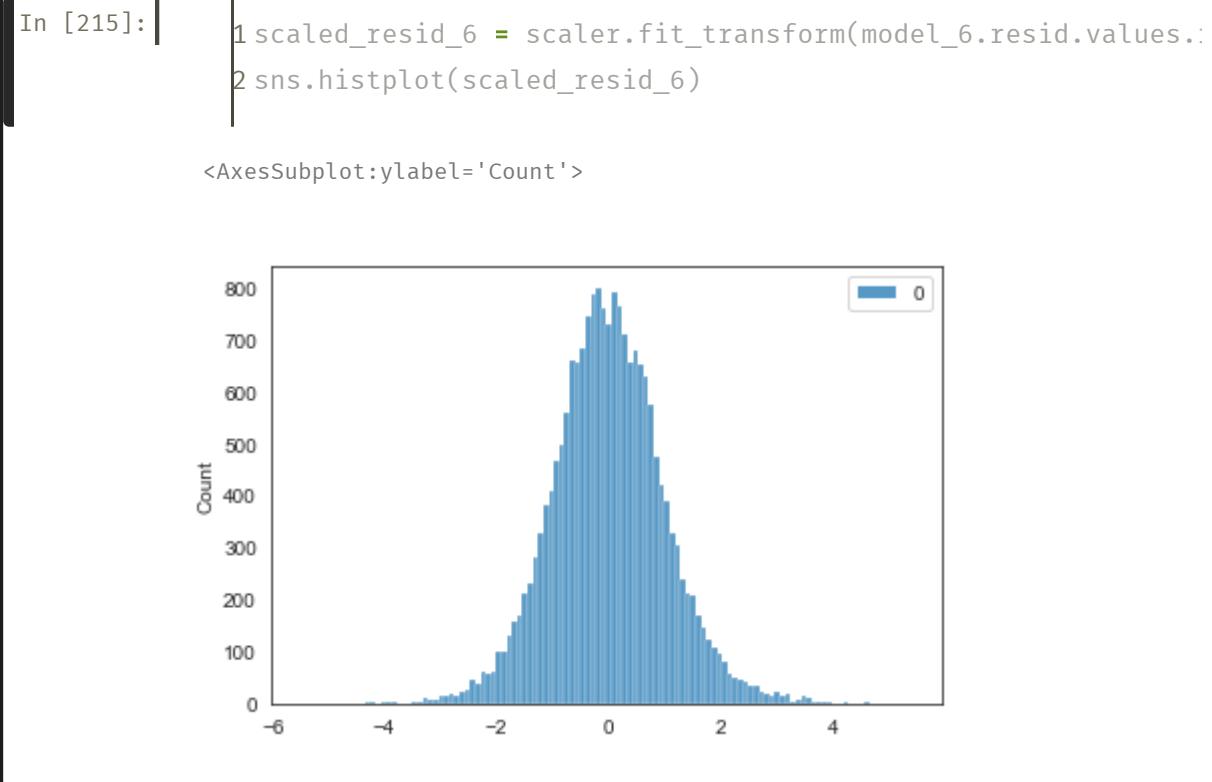
```
In [214]: 1 high_p_vals(model_6)
```

```
const has an acceptable p_value
grade has an acceptable p_value
bathrooms has an acceptable p_value
lat has an acceptable p_value
bedrooms has an acceptable p_value
floors has an acceptable p_value
closest_school_distance has an acceptable p_value
sqft_basement has an acceptable p_value
closest_hospital_distance has an acceptable p_value
yr_renovated has an acceptable p_value
sqft_lot15 has an acceptable p_value
long has an acceptable p_value
age_of_home has an acceptable p_value
condition has an acceptable p_value
waterfront has an acceptable p_value
view has an acceptable p_value
```

```
'drop [] for the next model due to high p_value(s)'
```

## Contents ⚙

- 6.2 Visualizing the Comparison
- ▼ 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- ▼ 7 Distance Data
  - ▼ 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - ▼ 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- ▼ 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- ▼ 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- ▼ 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- ▼ 14 Recommendations:
  - 14.0.1 Visualizing Schools

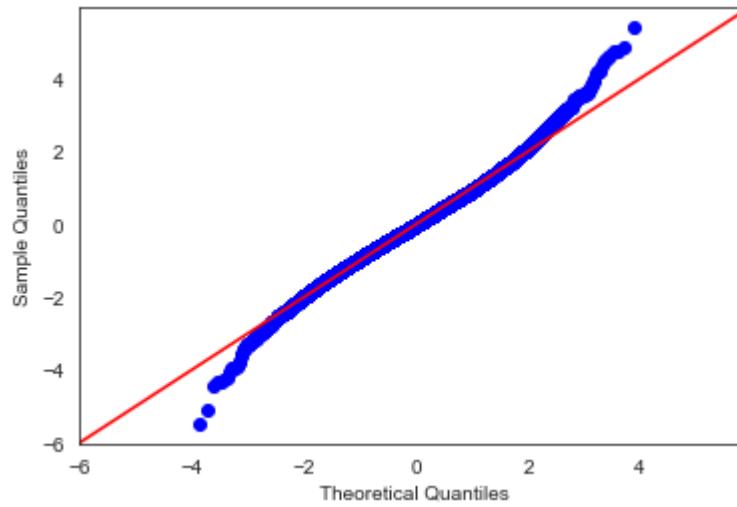
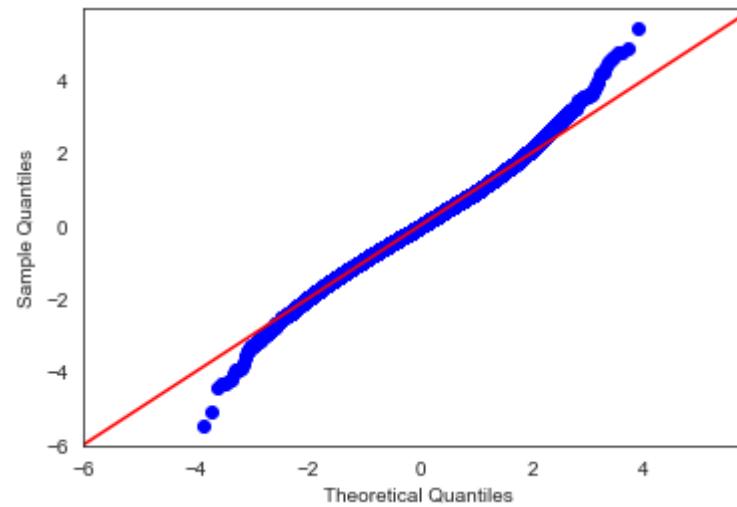


## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [216]:

```
1 sm.qqplot(model_6.resid, dist = stats.norm, line = "45", fi
```



## Contents ⚙

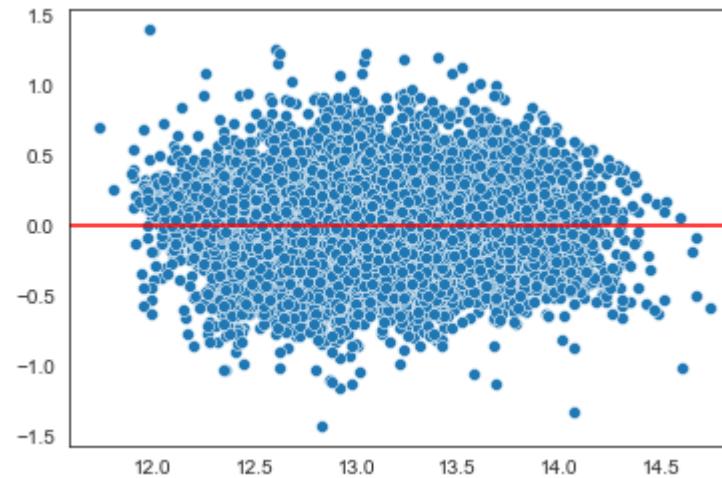
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [217]:

```
1 sns.scatterplot(model_6.predict(sm.add_constant(X)), model_6.predict(sm.add_constant(X)))
2 plt.axhline(0, color = 'red')
```

/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/sklearn/utils/\_param\_validation.py:6: FutureWarning: Pass the following variables as keyword args: x, y. From version 21.0, valid positional argument will be `data`, and passing other arguments with `x` and `y` will result in an error or misinterpretation.  
warnings.warn(  
 "Pass the following variables as keyword args: x, y. From version 21.0, valid positional argument will be `data`, and passing other arguments with `x` and `y` will result in an error or misinterpretation.",  
 FutureWarning)

&lt;matplotlib.lines.Line2D at 0x7ff34422c0a0&gt;



## Contents ⚙️

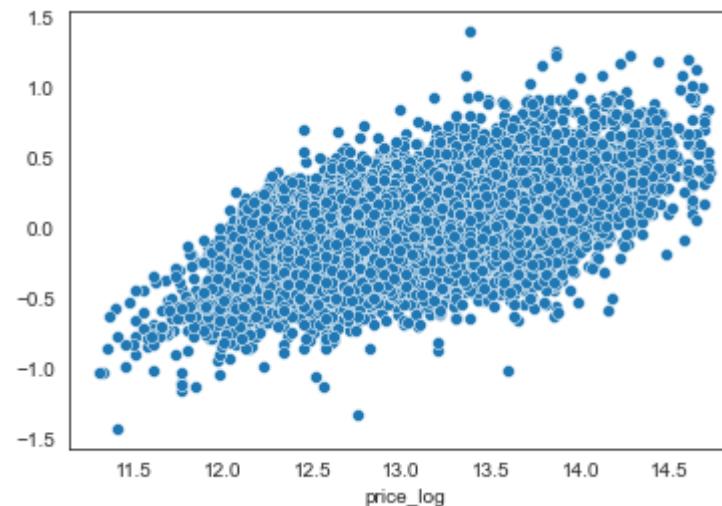
- 6.1 Computing Renovated Homes with One-hot Encoding
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

In [218]:

```
1 sns.scatterplot(y, model_6.resid)
```

```
/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/
6: FutureWarning: Pass the following variables as keyword args: x, y. From
valid positional argument will be `data`, and passing other arguments with
will result in an error or misinterpretation.
warnings.warn(
```

```
<AxesSubplot:xlabel='price_log'>
```



## 12.5 Seventh Model: Adding Zipcodes

## Contents ⚙️

- .. Comparing Renovated Homes with Unrenovated
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

```
In [219]: 1 zipcode_data.columns
```

```
Index(['id', 'date', 'price', 'bedrooms', 'bathrooms', 'floors', 'waterfront',  
       'view', 'condition', 'grade', 'sqft_basement', 'yr_renovated', 'lat',  
       'long', 'sqft_lot15', 'age_of_home', 'closest_hospital_distance',  
       'closest_school_distance', 'zipcode_98001', 'zipcode_98002',  
       'zipcode_98003', 'zipcode_98004', 'zipcode_98005', 'zipcode_98006',  
       'zipcode_98007', 'zipcode_98008', 'zipcode_98010', 'zipcode_98011',  
       'zipcode_98014', 'zipcode_98019', 'zipcode_98023', 'zipcode_98024',  
       'zipcode_98027', 'zipcode_98028', 'zipcode_98029', 'zipcode_98030',  
       'zipcode_98031', 'zipcode_98032', 'zipcode_98033', 'zipcode_98034',  
       'zipcode_98038', 'zipcode_98040', 'zipcode_98042', 'zipcode_98052',  
       'zipcode_98053', 'zipcode_98055', 'zipcode_98056', 'zipcode_98058',  
       'zipcode_98059', 'zipcode_98065', 'zipcode_98070', 'zipcode_98072',  
       'zipcode_98074', 'zipcode_98075', 'zipcode_98077', 'zipcode_98092',  
       'zipcode_98102', 'zipcode_98103', 'zipcode_98105', 'zipcode_98106',  
       'zipcode_98107', 'zipcode_98108', 'zipcode_98109', 'zipcode_98112',  
       'zipcode_98115', 'zipcode_98116', 'zipcode_98117', 'zipcode_98118',  
       'zipcode_98119', 'zipcode_98122', 'zipcode_98125', 'zipcode_98126',  
       'zipcode_98133', 'zipcode_98136', 'zipcode_98144', 'zipcode_98146',  
       'zipcode_98148', 'zipcode_98155', 'zipcode_98166', 'zipcode_98168',  
       'zipcode_98177', 'zipcode_98178', 'zipcode_98188', 'zipcode_98198',  
       'zipcode_98199', 'price_log'],  
      dtype='object')
```

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

```
In [220]: 1 features_7 = ['zipcode_98001',  
2     'zipcode_98002', 'zipcode_98003', 'zipcode_98004', '  
3     'zipcode_98006', 'zipcode_98007', 'zipcode_98008', '  
4     'zipcode_98011', 'zipcode_98014', 'zipcode_98019', '  
5     'zipcode_98024', 'zipcode_98027', 'zipcode_98028', '  
6     'zipcode_98030', 'zipcode_98031', 'zipcode_98032', '  
7     'zipcode_98034', 'zipcode_98038', 'zipcode_98040', '  
8     'zipcode_98052', 'zipcode_98053', 'zipcode_98055', '  
9     'zipcode_98058', 'zipcode_98059', 'zipcode_98065', '  
10    'zipcode_98072', 'zipcode_98074', 'zipcode_98075', '  
11    'zipcode_98092', 'zipcode_98102', 'zipcode_98103', '  
12    'zipcode_98106', 'zipcode_98107', 'zipcode_98108', '  
13    'zipcode_98112', 'zipcode_98115', 'zipcode_98116', '  
14    'zipcode_98118', 'zipcode_98119', 'zipcode_98122', '  
15    'zipcode_98126', 'zipcode_98133', 'zipcode_98136', '  
16    'zipcode_98146', 'zipcode_98148', 'zipcode_98155', '  
17    'zipcode_98168', 'zipcode_98177', 'zipcode_98178', '  
18    'zipcode_98198', 'zipcode_98199']
```

## Contents ⚙️

6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Modeling
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, and Longitude
11.2	Second Model: Transforming the Price with Logarithms
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest Hospital
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

In [221]:

```
1 X = pd.concat([X, zipcode_data[features_7]], axis = 1)
2 X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 19414 entries, 0 to 21596
Data columns (total 82 columns):
 #   Column           Non-Null Count Dtype  
 --- 
 0   grade            19414 non-null  int64   
 1   bathrooms        19414 non-null  float64 
 2   lat              19414 non-null  float64 
 3   bedrooms         19414 non-null  int64   
 4   floors            19414 non-null  float64 
 5   closest_school_distance  19414 non-null  float64 
 6   sqft_basement     19414 non-null  float64 
 7   closest_hospital_distance  19414 non-null  float64 
 8   yr_renovated      19414 non-null  float64 
 9   sqft_lot15         19414 non-null  int64   
 10  long             19414 non-null  float64 
 11  age_of_home       19414 non-null  int64   
 12  condition         19414 non-null  int64   
 13  waterfront        19414 non-null  int64   
 14  view              19414 non-null  float64 
 15  zipcode_98001      19414 non-null  uint8  
 16  zipcode_98002      19414 non-null  uint8  
 17  zipcode_98003      19414 non-null  uint8  
 18  zipcode_98004      19414 non-null  uint8  
 19  zipcode_98005      19414 non-null  uint8  
 20  zipcode_98006      19414 non-null  uint8  
 21  zipcode_98007      19414 non-null  uint8  
 22  zipcode_98008      19414 non-null  uint8  
 23  zipcode_98010      19414 non-null  uint8  
 24  zipcode_98011      19414 non-null  uint8  
 25  zipcode_98014      19414 non-null  uint8  
 26  zipcode_98019      19414 non-null  uint8  
 27  zipcode_98023      19414 non-null  uint8  
 28  zipcode_98024      19414 non-null  uint8  
 29  zipcode_98027      19414 non-null  uint8  
 30  zipcode_98028      19414 non-null  uint8  
 31  zipcode_98029      19414 non-null  uint8  
 32  zipcode_98030      19414 non-null  uint8 
```

## Contents ⚙️

6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
▼ 7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
▼ 7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Modeling
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest School
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

33	zipcode_98031	19414	non-null	uint8
34	zipcode_98032	19414	non-null	uint8
35	zipcode_98033	19414	non-null	uint8
36	zipcode_98034	19414	non-null	uint8
37	zipcode_98038	19414	non-null	uint8
38	zipcode_98040	19414	non-null	uint8
39	zipcode_98042	19414	non-null	uint8
40	zipcode_98052	19414	non-null	uint8
41	zipcode_98053	19414	non-null	uint8
42	zipcode_98055	19414	non-null	uint8
43	zipcode_98056	19414	non-null	uint8
44	zipcode_98058	19414	non-null	uint8
45	zipcode_98059	19414	non-null	uint8
46	zipcode_98065	19414	non-null	uint8
47	zipcode_98070	19414	non-null	uint8
48	zipcode_98072	19414	non-null	uint8
49	zipcode_98074	19414	non-null	uint8
50	zipcode_98075	19414	non-null	uint8
51	zipcode_98077	19414	non-null	uint8
52	zipcode_98092	19414	non-null	uint8
53	zipcode_98102	19414	non-null	uint8
54	zipcode_98103	19414	non-null	uint8
55	zipcode_98105	19414	non-null	uint8
56	zipcode_98106	19414	non-null	uint8
57	zipcode_98107	19414	non-null	uint8
58	zipcode_98108	19414	non-null	uint8
59	zipcode_98109	19414	non-null	uint8
60	zipcode_98112	19414	non-null	uint8
61	zipcode_98115	19414	non-null	uint8
62	zipcode_98116	19414	non-null	uint8
63	zipcode_98117	19414	non-null	uint8
64	zipcode_98118	19414	non-null	uint8
65	zipcode_98119	19414	non-null	uint8
66	zipcode_98122	19414	non-null	uint8
67	zipcode_98125	19414	non-null	uint8
68	zipcode_98126	19414	non-null	uint8
69	zipcode_98133	19414	non-null	uint8
70	zipcode_98136	19414	non-null	uint8
71	zipcode_98144	19414	non-null	uint8
72	zipcode_98146	19414	non-null	uint8
73	zipcode_98148	19414	non-null	uint8
74	zipcode_98155	19414	non-null	uint8
75	zipcode_98166	19414	non-null	uint8

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
- 7.3 Exploring Distance Data
  - 7.3.1 Downtown
  - 7.3.2 Hospitals
  - 7.3.3 Police Stations
  - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

## Seattle\_Section\_8\_Project - Jupyter Notebook

```
76    zipcode_98168           19414 non-null  uint8
77    zipcode_98177           19414 non-null  uint8
78    zipcode_98178           19414 non-null  uint8
79    zipcode_98188           19414 non-null  uint8
80    zipcode_98198           19414 non-null  uint8
81    zipcode_98199           19414 non-null  uint8
dtypes: float64(9), int64(6), uint8(67)
memory usage: 3.6 MB
```

In [222]:

```
1 model_7 = sm.OLS(y, sm.add_constant(X)).fit()
2 model_7.summary()
```

### OLS Regression Results

Dep. Variable:	price_log	R-squared:	0.824			
Model:	OLS	Adj. R-squared:	0.824			
Method:	Least Squares	F-statistic:	1135.			
Date:	Tue, 12 Apr 2022	Prob (F-statistic):	0.00			
Time:	17:54:52	Log-Likelihood:	3469.4			
No. Observations:	19414	AIC:	-6777.			
Df Residuals:	19333	BIC:	-6139.			
Df Model:	80					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	42.7426	14.032	3.046	0.002	15.238	70.247
...	...	...	...	...	...	...

## Contents ⚙️

6.2 Visualizing the Comparison
▼ 6.3 Home History
6.3.1 Age of Home
6.3.2 Year Modified
▼ 7 Distance Data
▼ 7.1 Getting Infrastructure Data
7.1.1 Downtown
7.1.2 Hospitals
7.1.3 Police Stations
7.1.4 Schools
7.2 Finding Distance for Infrastructure
▼ 7.3 Exploring Distance Data
7.3.1 Downtown
7.3.2 Hospitals
7.3.3 Police Stations
7.3.4 Schools
▼ 8 Preparing for Modeling
8.0.1 Dropping View and Waterfront Properties for Model
9 Zipcode
10 Seeing the Correlation
▼ 11 Building a model
11.1 Baseline Model: Grade, Bathrooms, Lat, Long
11.2 Second Model: Transforming the Price with Log
▼ 12 Improving the Model with Adding Features
12.1 Third Model: Bedrooms, Floors, Closest School
12.2 Fourth Model: Square Foot Basement, Closest School
12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
12.4 Sixth Model: Condition, Waterfront and View
12.5 Seventh Model: Adding Zipcodes
12.6 Eighth Model: Latitude and Closest Hospital Distance
13 Conclusion
▼ 14 Recommendations:
14.0.1 Visualizing Schools

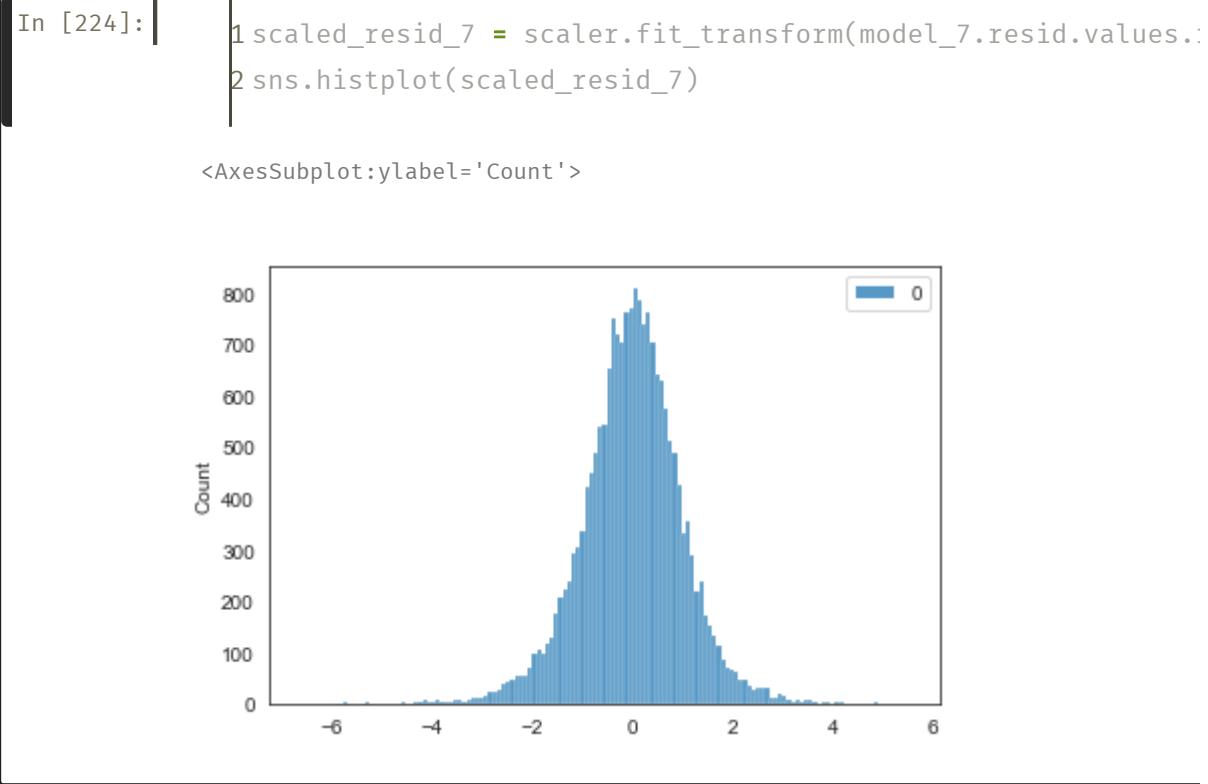
In [223]: 1 high\_p\_vals(model\_7)

```
const has an acceptable p_value
grade has an acceptable p_value
bathrooms has an acceptable p_value
lat has an acceptable p_value
bedrooms has an acceptable p_value
floors has an acceptable p_value
closest_school_distance has an acceptable p_value
sqft_basement has an acceptable p_value
yr_renovated has an acceptable p_value
sqft_lot15 has an acceptable p_value
long has an acceptable p_value
age_of_home has an acceptable p_value
condition has an acceptable p_value
waterfront has an acceptable p_value
view has an acceptable p_value
zipcode_98001 has an acceptable p_value
zipcode_98002 has an acceptable p_value
zipcode_98003 has an acceptable p_value
zipcode_98004 has an acceptable p_value
zipcode_98005 has an acceptable p value
```

Adding in zipcodes has made the latitude and the closest hospital predictors and I'll be removing those features for the next iteration.

## Contents ⚙

- ... Comparing New Model with Old Model
- 6.2 Visualizing the Comparison
- ▼ 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- ▼ 7 Distance Data
  - ▼ 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - ▼ 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- ▼ 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- ▼ 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- ▼ 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- ▼ 14 Recommendations:
  - 14.0.1 Visualizing Schools

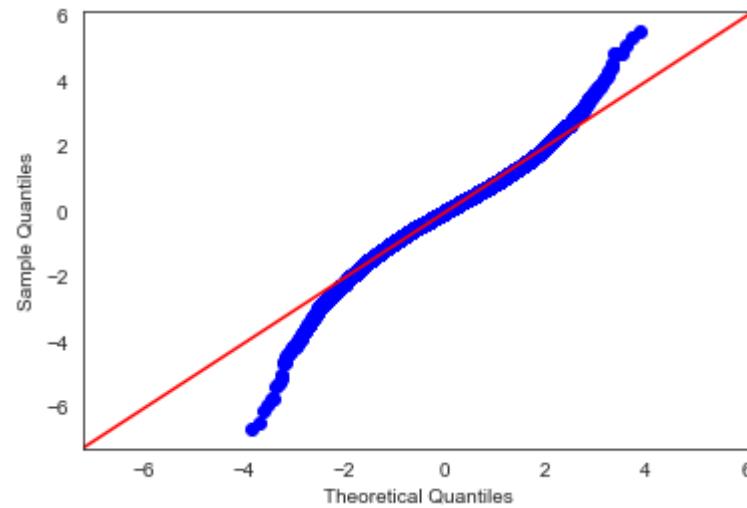


## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

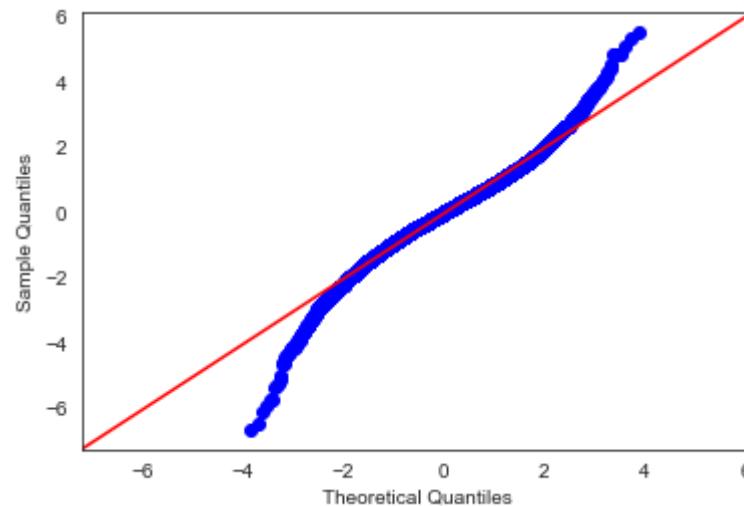
In [225]:

```
1 sm.qqplot(model_7.resid, dist = stats.norm, line = "45", fi
```



## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
- 7.3 Exploring Distance Data
  - 7.3.1 Downtown
  - 7.3.2 Hospitals
  - 7.3.3 Police Stations
  - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools



Observations: zipcode data has moved the model into performing worse. The residuals are now more residuals in the -4 through -2 and 2 through 4 distance range. Basically, the model is predicting less correctly where the data points fall outside the red line.

## Contents ⚙

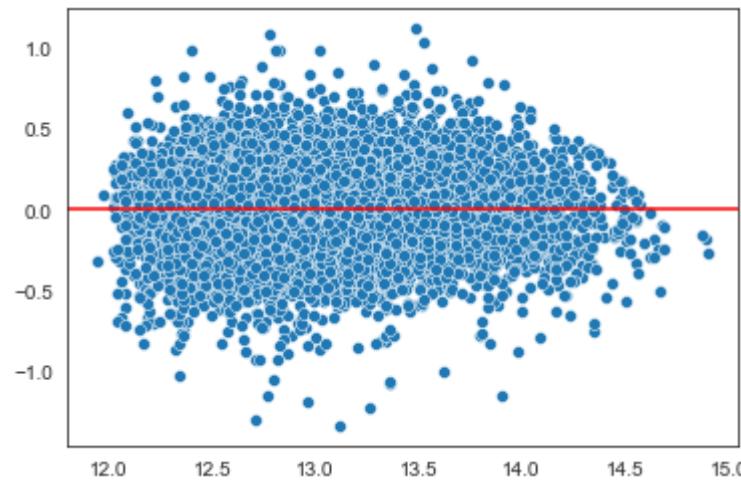
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [226]:

```
1 sns.scatterplot(model_7.predict(sm.add_constant(X)), model_7.y)
```

```
/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/
6: FutureWarning: Pass the following variables as keyword args: x, y. From
valid positional argument will be `data`, and passing other arguments with
will result in an error or misinterpretation.
warnings.warn(
```

```
<matplotlib.lines.Line2D at 0x7ff343f8cfa0>
```



## Contents ⚙️

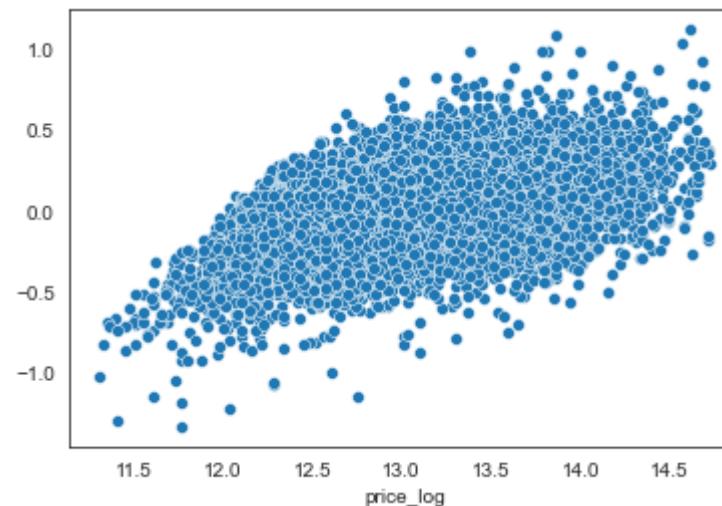
- .. Comparing Renovated Homes with Unrenovated
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [227]:

```
1 sns.scatterplot(y, model_7.resid)
```

```
/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/
6: FutureWarning: Pass the following variables as keyword args: x, y. From
valid positional argument will be `data`, and passing other arguments with
will result in an error or misinterpretation.
warnings.warn(
```

```
<AxesSubplot:xlabel='price_log'>
```



## 12.6 Eighth Model: Latitude and Closest Hospital Distance

## Contents ⚙️

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Modeling
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, and Longitude
    - 11.2 Second Model: Transforming the Price with Logarithms
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

```
In [228]: 1 X = X.drop(columns = ['lat', 'closest_hospital_distance'])

In [229]: 1 features_8 = [col for col in X.columns]

In [230]: 1 X = zipcode_data.loc[:, features_8]

In [231]: 1 X.info()

63    zipcode_98119      19414 non-null  uint8
64    zipcode_98122      19414 non-null  uint8
65    zipcode_98125      19414 non-null  uint8
66    zipcode_98126      19414 non-null  uint8
67    zipcode_98133      19414 non-null  uint8
68    zipcode_98136      19414 non-null  uint8
69    zipcode_98144      19414 non-null  uint8
70    zipcode_98146      19414 non-null  uint8
71    zipcode_98148      19414 non-null  uint8
72    zipcode_98155      19414 non-null  uint8
73    zipcode_98166      19414 non-null  uint8
74    zipcode_98168      19414 non-null  uint8
75    zipcode_98177      19414 non-null  uint8
76    zipcode_98178      19414 non-null  uint8
77    zipcode_98188      19414 non-null  uint8
78    zipcode_98198      19414 non-null  uint8
79    zipcode_98199      19414 non-null  uint8
dtypes: float64(7), int64(6), uint8(67)
memory usage: 3.3 MB
```

## Contents ⚙️

6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Now
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Log
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest School
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

```
In [232]: 1 model_8 = sm.OLS(y, sm.add_constant(X)).fit()
2 model_8.summary()
```

### OLS Regression Results

Dep. Variable:	price_log	R-squared:	0.824			
Model:	OLS	Adj. R-squared:	0.824			
Method:	Least Squares	F-statistic:	1163.			
Date:	Tue, 12 Apr 2022	Prob (F-statistic):	0.00			
Time:	17:54:53	Log-Likelihood:	3465.5			
No. Observations:	19414	AIC:	-6773.			
Df Residuals:	19335	BIC:	-6151.			
Df Model:	78					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
const	57.6242	12.627	4.564	0.000	32.875	82.373
grade	0.1967	0.002	95.575	0.000	0.192	0.200

## Contents ⚙️

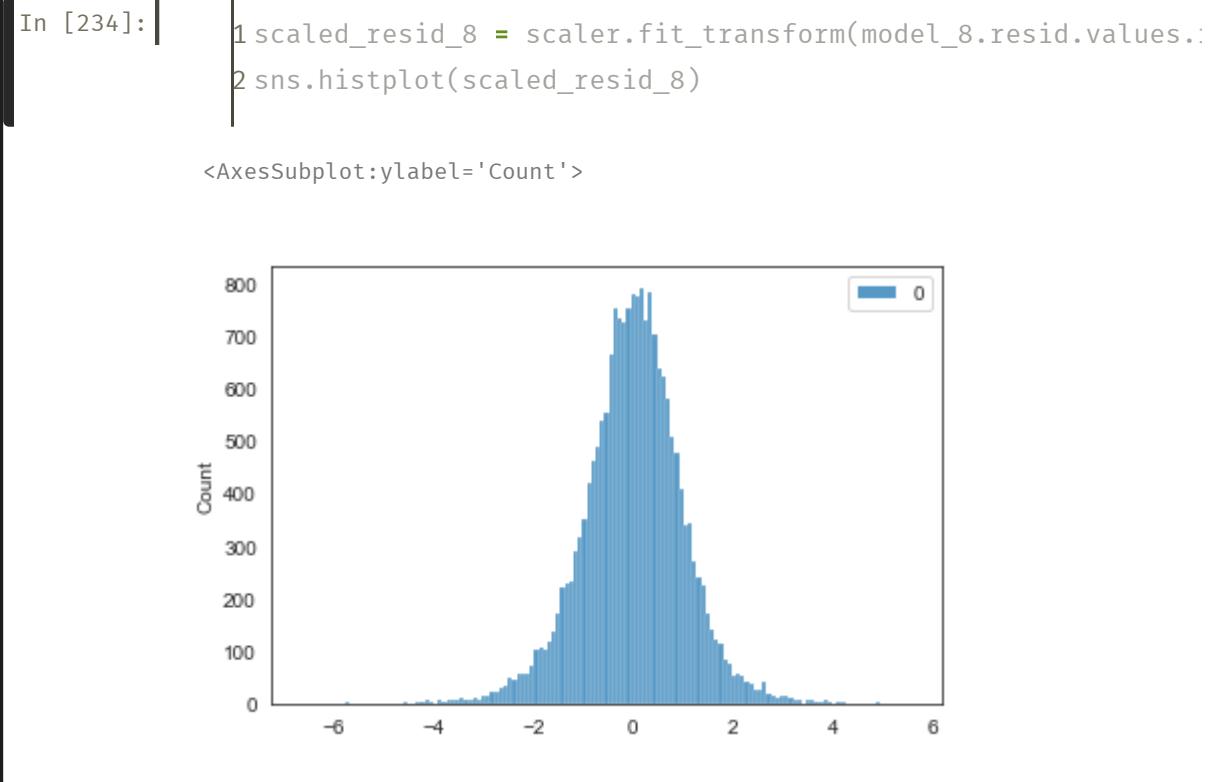
6.2 Visualizing the Comparison
▼ 6.3 Home History
6.3.1 Age of Home
6.3.2 Year Modified
▼ 7 Distance Data
▼ 7.1 Getting Infrastructure Data
7.1.1 Downtown
7.1.2 Hospitals
7.1.3 Police Stations
7.1.4 Schools
7.2 Finding Distance for Infrastructure
▼ 7.3 Exploring Distance Data
7.3.1 Downtown
7.3.2 Hospitals
7.3.3 Police Stations
7.3.4 Schools
▼ 8 Preparing for Modeling
8.0.1 Dropping View and Waterfront Properties for Modeling
9 Zipcode
10 Seeing the Correlation
▼ 11 Building a model
11.1 Baseline Model: Grade, Bathrooms, Latitude, and Longitude
11.2 Second Model: Transforming the Price with Logarithms
▼ 12 Improving the Model with Adding Features
12.1 Third Model: Bedrooms, Floors, Closest School, and Zipcode
12.2 Fourth Model: Square Foot Basement, Closest School, and Zipcode
12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4 Sixth Model: Condition, Waterfront and View
12.5 Seventh Model: Adding Zipcodes
12.6 Eighth Model: Latitude and Closest Hospital Distance
13 Conclusion
▼ 14 Recommendations:
14.0.1 Visualizing Schools

In [233]: 1 high\_p\_vals(model\_8)

```
const has an acceptable p_value
grade has an acceptable p_value
bathrooms has an acceptable p_value
bedrooms has an acceptable p_value
floors has an acceptable p_value
closest_school_distance has an acceptable p_value
sqft_basement has an acceptable p_value
yr_renovated has an acceptable p_value
sqft_lot15 has an acceptable p_value
long has an acceptable p_value
age_of_home has an acceptable p_value
condition has an acceptable p_value
waterfront has an acceptable p_value
view has an acceptable p_value
zipcode_98001 has an acceptable p_value
zipcode_98002 has an acceptable p_value
zipcode_98003 has an acceptable p_value
zipcode_98004 has an acceptable p_value
zipcode_98005 has an acceptable p_value
zipcode_98006 has an acceptable p value
```

## Contents ⚙

- 6.2 Visualizing the Comparison
- ▼ 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- ▼ 7 Distance Data
  - ▼ 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - ▼ 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- ▼ 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- ▼ 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- ▼ 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest School
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- ▼ 14 Recommendations:
  - 14.0.1 Visualizing Schools

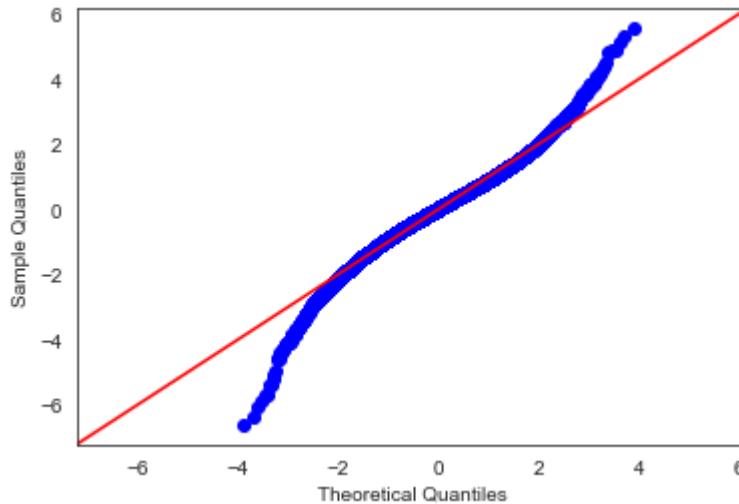
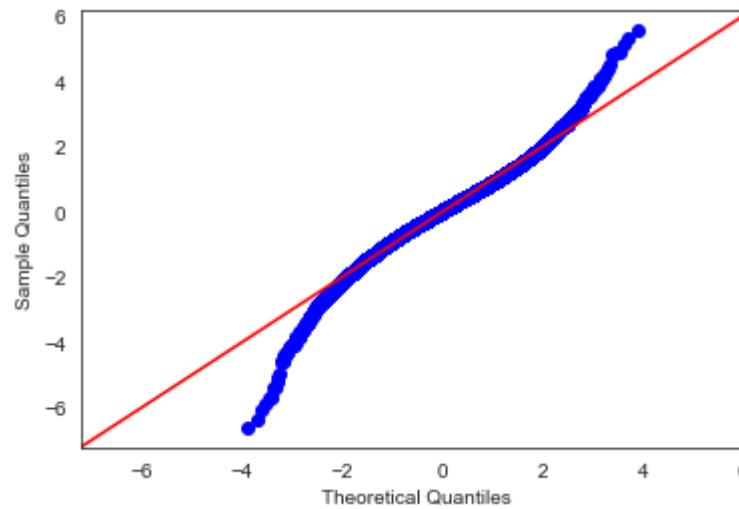


## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

In [235]:

```
1 sm.qqplot(model_8.resid, dist = stats.norm, line = "45", fi
```



## Contents ⚙

- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
- 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- 14 Recommendations:
  - 14.0.1 Visualizing Schools

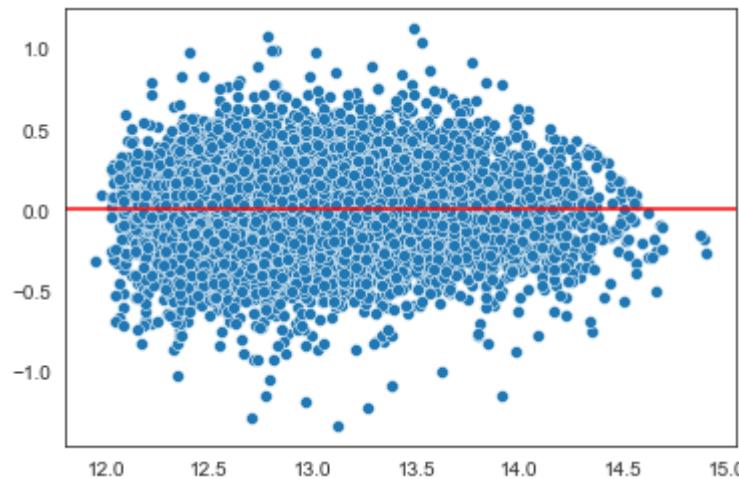
In [236]:

```
1 sns.scatterplot(model_8.predict(sm.add_constant(X)), model_8.y)
```

```
2 plt.axhline(0, color = 'red')

/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/
6: FutureWarning: Pass the following variables as keyword args: x, y. From
valid positional argument will be `data`, and passing other arguments with
will result in an error or misinterpretation.
warnings.warn(
```

```
<matplotlib.lines.Line2D at 0x7ff344b0aa0>
```



## Contents ⚙️

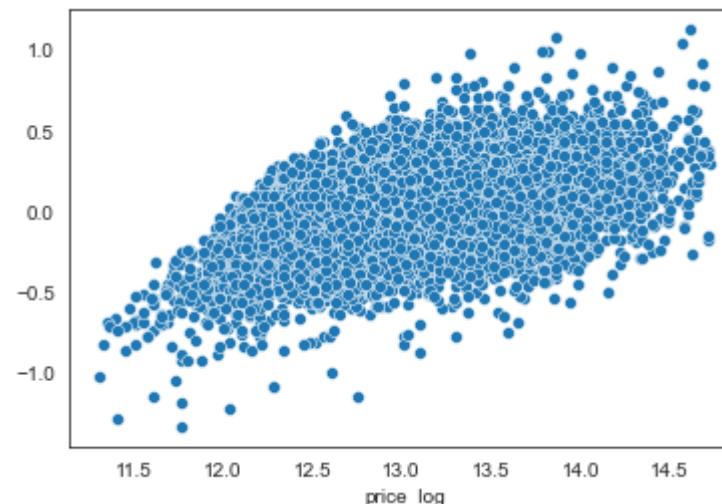
- 6.1 Comparing Renovated Homes with Unrenovated
- 6.2 Visualizing the Comparison
- 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
  - 9 Zipcode
  - 10 Seeing the Correlation
  - 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Logarithms
  - 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - 14 Recommendations:
    - 14.0.1 Visualizing Schools

In [237]:

```
1 sns.scatterplot(y, model_8.resid)
```

```
/Users/b0ihazard/opt/anaconda3/envs/learn-env/lib/python3.8/site-packages/
6: FutureWarning: Pass the following variables as keyword args: x, y. From
valid positional argument will be `data`, and passing other arguments with
will result in an error or misinterpretation.
warnings.warn(
```

```
<AxesSubplot:xlabel='price_log'>
```



## 13 Conclusion

I'm going to generate a report of the findings of the model's coefficients.

## Contents ⚙

6.2 Visualizing the Comparison
▼ 6.3 Home History
6.3.1 Age of Home
6.3.2 Year Modified
▼ 7 Distance Data
7.1 Getting Infrastructure Data
7.1.1 Downtown
7.1.2 Hospitals
7.1.3 Police Stations
7.1.4 Schools
7.2 Finding Distance for Infrastructure
7.3 Exploring Distance Data
7.3.1 Downtown
7.3.2 Hospitals
7.3.3 Police Stations
7.3.4 Schools
▼ 8 Preparing for Modeling
8.0.1 Dropping View and Waterfront Properties for Now
9 Zipcode
10 Seeing the Correlation
▼ 11 Building a model
11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2 Second Model: Transforming the Price with Log
▼ 12 Improving the Model with Adding Features
12.1 Third Model: Bedrooms, Floors, Closest School
12.2 Fourth Model: Square Foot Basement, Closest School
12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
12.4 Sixth Model: Condition, Waterfront and View
12.5 Seventh Model: Adding Zipcodes
12.6 Eighth Model: Latitude and Closest Hospital Distance
13 Conclusion
▼ 14 Recommendations:
14.0.1 Visualizing Schools

```
In [238]: 1 pd.set_option('display.max_rows', None)
2
3 coeff_s = model_8.params
4 coeff_s = coeff_s.drop('const')
5
6 labels = coeff_s.index
7 coefficients = dict(zip(labels, coeff_s))
8
9 # grabbing the coefficients from the stats models reports
```

By generating a report of the findings, should this model be used in the future, understanding the model will become much more approachable and friendly. This is good news for the civic workers that will be using it.

Contents  

- 6.2 Visualizing the Comparison
  - ▼ 6.3 Home History
    - 6.3.1 Age of Home
    - 6.3.2 Year Modified
  - ▼ 7 Distance Data
    - 7.1 Getting Infrastructure Data
      - 7.1.1 Downtown
      - 7.1.2 Hospitals
      - 7.1.3 Police Stations
      - 7.1.4 Schools
    - 7.2 Finding Distance for Infrastructure
    - 7.3 Exploring Distance Data
      - 7.3.1 Downtown
      - 7.3.2 Hospitals
      - 7.3.3 Police Stations
      - 7.3.4 Schools
  - ▼ 8 Preparing for Modeling
    - 8.0.1 Dropping View and Waterfront Properties for Model
  - 9 Zipcode
  - 10 Seeing the Correlation
  - ▼ 11 Building a model
    - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
    - 11.2 Second Model: Transforming the Price with Log
  - ▼ 12 Improving the Model with Adding Features
    - 12.1 Third Model: Bedrooms, Floors, Closest School
    - 12.2 Fourth Model: Square Foot Basement, Closest School
    - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighbors
    - 12.4 Sixth Model: Condition, Waterfront and View
    - 12.5 Seventh Model: Adding Zipcodes
    - 12.6 Eighth Model: Latitude and Closest Hospital Distance
  - 13 Conclusion
  - ▼ 14 Recommendations:
    - 14.0.1 Visualizing Schools

```
In [239]: 1 for x, y in coefficients.items():
2     y = y * 100
3
4     if x.startswith('view') or x.startswith('waterfront'):
5         print(f' If {x} was a feature of a property, the price would increase by {y} percent.')
6
7     elif not x.startswith('zipcode') and not x.startswith('const'):
8         print(f' As {x} increases by one unit, the price changes by {y} percent.')
9
10
11 else:
12     if x.startswith('zipcode') and not x.startswith('const'):
13         print(f' If a property was in {x} zipcode instead of 98039, the price would change by {y} percent.')
14
15     if x == 'zipcode_98146':
16         print(f' If a property was in zipcode_98146 zipcode instead of 98039, the price would increase by {y} percent.')
17
18     if x == 'zipcode_98148':
19         print(f' If a property was in zipcode_98148 zipcode instead of 98039, the price would increase by {y} percent.')
20
21     if x == 'zipcode_98155':
22         print(f' If a property was in zipcode_98155 zipcode instead of 98039, the price would increase by {y} percent.')
23
24     if x == 'zipcode_98166':
25         print(f' If a property was in zipcode_98166 zipcode instead of 98039, the price would increase by {y} percent.')
26
27     if x == 'zipcode_98168':
28         print(f' If a property was in zipcode_98168 zipcode instead of 98039, the price would increase by {y} percent.')
29
30     if x == 'zipcode_98177':
31         print(f' If a property was in zipcode_98177 zipcode instead of 98039, the price would increase by {y} percent.')
32
33     if x == 'zipcode_98178':
34         print(f' If a property was in zipcode_98178 zipcode instead of 98039, the price would increase by {y} percent.')
35
36     if x == 'zipcode_98188':
37         print(f' If a property was in zipcode_98188 zipcode instead of 98039, the price would increase by {y} percent.')
38
39     if x == 'zipcode_98198':
40         print(f' If a property was in zipcode_98198 zipcode instead of 98039, the price would increase by {y} percent.')
41
42     if x == 'zipcode_98199':
43         print(f' If a property was in zipcode_98199 zipcode instead of 98039, the price would increase by {y} percent.'
```

## 14 Recommendations:

- 1) As we can read in the printed report above, improving the grac (as is our aim with renovations) would make the value of the prop roughly 1/4th. *Therefore I'm recommending that the city of Seattle*

## Contents

<a href="#">6.1 Comparing Renovated Homes with Unrenovated</a>
6.2 Visualizing the Comparison
▼ 6.3 Home History
6.3.1 Age of Home
6.3.2 Year Modified
▼ 7 Distance Data
▼ 7.1 Getting Infrastructure Data
7.1.1 Downtown
7.1.2 Hospitals
7.1.3 Police Stations
7.1.4 Schools
7.2 Finding Distance for Infrastructure
▼ 7.3 Exploring Distance Data
7.3.1 Downtown
7.3.2 Hospitals
7.3.3 Police Stations
7.3.4 Schools
▼ 8 Preparing for Modeling
8.0.1 Dropping View and Waterfront Properties for
9 Zipcode
10 Seeing the Correlation
▼ 11 Building a model
11.1 Baseline Model: Grade, Bathrooms, Lat
11.2 Second Model: Transforming the Price with Log
▼ 12 Improving the Model with Adding Features
12.1 Third Model: Bedrooms, Floors, Closest School
12.2 Fourth Model: Square Foot Basement, Closest
12.3 Fifth Model: Square Foot Lot of Closest 15 Nei
12.4 Sixth Model: Condition, Waterfront and View
12.5 Seventh Model: Adding Zipcodes
12.6 Eighth Model: Latitude and Closest Hospital D
13 Conclusion
▼ 14 Recommendations:
14.0.1 Visualizing Schools

both grade and condition of properties during the renovation. In the year\_renovated by itself had little affect on the price of the home. This is that it's not enough to simply renovate a home, but we must improve and/or grade with that renovation.

Choosing which properties to renovate: the properties where the changes will increase the grade, and/or condition of the property will have a high return on investment.

2) There are certain zipcodes where the class disparity is quite large. In the city of Seattle spend 1/3 of this current projects budget just for renovating properties in these zipcodes with the lowest grade/conditions, and these are mostly public housing. These zipcodes are as follows: 98168, 98188, 98146, 98108, 98092, 98058, 98055, 98042, 98032, 98031, 98030, 98002, 98001. The data shows that properties in these zipcodes would increase value by over 100% if they were in the 98039 zipcode instead.

3) While one and a half percent may not seem like a lot--there is some evidence that putting new schools specifically where the closest school is will improve the property values of all the homes closest to that school. By increasing distance away from homes, the property value of the homes would increase by 1.5%. While building schools is outside the range of the scope of this project, it is a finding to take back to the city counsel and state se to recommend that when selecting properties for this project, flagging those within 3 miles of a public school for selection across zipcodes. People deserve functional infrastructure.

### 14.0.1 Visualizing Schools

```
In [240]: 1 school_coords = []
            2 for x, y in coordinates.items():
            3     if x in school_addresses:
            4         school_coords.append(y)
```

## Contents ⚙️

...	Comparing Neighborhood Names with Zipcodes
6.2	Visualizing the Comparison
▼ 6.3	Home History
6.3.1	Age of Home
6.3.2	Year Modified
▼ 7	Distance Data
7.1	Getting Infrastructure Data
7.1.1	Downtown
7.1.2	Hospitals
7.1.3	Police Stations
7.1.4	Schools
7.2	Finding Distance for Infrastructure
7.3	Exploring Distance Data
7.3.1	Downtown
7.3.2	Hospitals
7.3.3	Police Stations
7.3.4	Schools
▼ 8	Preparing for Modeling
8.0.1	Dropping View and Waterfront Properties for Modeling
9	Zipcode
10	Seeing the Correlation
▼ 11	Building a model
11.1	Baseline Model: Grade, Bathrooms, Latitude, Longitude
11.2	Second Model: Transforming the Price with Logarithmic Scale
▼ 12	Improving the Model with Adding Features
12.1	Third Model: Bedrooms, Floors, Closest School
12.2	Fourth Model: Square Foot Basement, Closest School
12.3	Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
12.4	Sixth Model: Condition, Waterfront and View
12.5	Seventh Model: Adding Zipcodes
12.6	Eighth Model: Latitude and Closest Hospital Distance
13	Conclusion
▼ 14	Recommendations:
14.0.1	Visualizing Schools

In [241]: 1 school\_labels = dict(zip(school\_names, school\_coords))

In [ ]: 1 base\_map = folium.Map(downtown, zoom\_start=11)  
 2  
 3 for school, location in school\_labels.items():  
 4 lat = location[0]  
 5 long = location[1]  
 6 popup\_text = school  
 7 popup = folium.Popup(popup\_text, parse\_html=True)  
 8 marker = folium.CircleMarker(location=[lat, long], popups=[popup],  
 9 fill = True, radius = 20)  
 10 marker.add\_to(base\_map)  
 11  
 12 base\_map

While we can visually inspect the map above and look for places that could go, we can also combine our knowledge of the zipcode data by building a new school in the zipcodes that have the highest class such as the wealthiest zipcode such as zipcode 98032 (the zipcode with the most properties in this dataset.)

## FURTHER WORK:

With more time and resources, I'd like to exactly calculate the area of the circles that are isolated from public schools, but the above map shows a rough idea of the areas of the map not shaded blue by a circle. In the future, I'd like to calculate the distance of these circles in human measurements, not in pixels.

## Contents ⚙️

- 6.2 Visualizing the Comparison
- ▼ 6.3 Home History
  - 6.3.1 Age of Home
  - 6.3.2 Year Modified
- ▼ 7 Distance Data
  - 7.1 Getting Infrastructure Data
    - 7.1.1 Downtown
    - 7.1.2 Hospitals
    - 7.1.3 Police Stations
    - 7.1.4 Schools
  - 7.2 Finding Distance for Infrastructure
  - ▼ 7.3 Exploring Distance Data
    - 7.3.1 Downtown
    - 7.3.2 Hospitals
    - 7.3.3 Police Stations
    - 7.3.4 Schools
- ▼ 8 Preparing for Modeling
  - 8.0.1 Dropping View and Waterfront Properties for Now
- 9 Zipcode
- 10 Seeing the Correlation
- ▼ 11 Building a model
  - 11.1 Baseline Model: Grade, Bathrooms, Latitude, Longitude
  - 11.2 Second Model: Transforming the Price with Log
- ▼ 12 Improving the Model with Adding Features
  - 12.1 Third Model: Bedrooms, Floors, Closest School
  - 12.2 Fourth Model: Square Foot Basement, Closest Hospital
  - 12.3 Fifth Model: Square Foot Lot of Closest 15 Neighborhoods
  - 12.4 Sixth Model: Condition, Waterfront and View
  - 12.5 Seventh Model: Adding Zipcodes
  - 12.6 Eighth Model: Latitude and Closest Hospital Distance
- 13 Conclusion
- ▼ 14 Recommendations:
  - 14.0.1 Visualizing Schools