# Linear Regression Model for HGTV Show

## Buiness Problem

---

One of HGTV Network's couples is moving to KingS Country and HGTV does not want to lose them. Instead of letting them go HGTV will be using the King county's data set to figure out what type of show would fit with the couple's new areas. The most important indicator is cost since that will be use to determine the budget.

## Data Mining - Uploading and Reviewing Data

---

### Importing

In [146…

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.formula.api import ols
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import statsmodels.api as sm
import scipy.stats as stats

%matplotlib inline
```

In [147…

```
df = pd.read_csv('./Cloned-folder/dsc-phase-2-project/data/kc_house_data.csv')
df.head()
```

Out[147…

| | id | date | price | bedrooms | bathrooms | sqft_living | sqft_lot | floors | waterfront | vi |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 7129300520 | 10/13/2014 | 221900.0 | 3 | 1.00 | 1180 | 5650 | 1.0 | NaN | |
| **1** | 6414100192 | 12/9/2014 | 538000.0 | 3 | 2.25 | 2570 | 7242 | 2.0 | 0.0 | |
| **2** | 5631500400 | 2/25/2015 | 180000.0 | 2 | 1.00 | 770 | 10000 | 1.0 | 0.0 | |
| **3** | 2487200875 | 12/9/2014 | 604000.0 | 4 | 3.00 | 1960 | 5000 | 1.0 | 0.0 | |
| **4** | 1954400510 | 2/18/2015 | 510000.0 | 3 | 2.00 | 1680 | 8080 | 1.0 | 0.0 | |

5 rows × 21 columns

# Data Cleaning/ Exploration

## Functions Used

In [148…
```
#Convert datatype to INT
def convert_int(df, col):
    df[col] = df[col].astype(str).astype(int)
    return df.info()
```

In [149…
```
#Convert datatype to FLOAT
def convert_float(df, col):
    df[col] = df[col].astype(str).astype(float)
```

In [150…
```
#Convert datatype to DATE
def convert_date(df, col):
    df[col] = pd.to_datetime(df[col])
    return df.info()
```

In [151…
```
#converting Null falues to 0
def convert_nan_to_0(col):
    #df['view'].fillna(0.0)

    df[col].fillna(0.0, inplace=True)
    return df.info()
```

## Column Names and Descriptions for Kings County Data Set

In [152…
```
df.info()
```

```
<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  float64
 9   view           21534 non-null  float64
 10  condition      21597 non-null  int64  
 11  grade          21597 non-null  int64  
 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(8), int64(11), object(2)
memory usage: 3.5+ MB
```

Removing date from day from date during the converting because filming for any show would be continous and trying to fit a show around a few days in a single month would not be fesiable with productivity. As shown below it is having the day does not contribute addition infomation, the dip in houses sold would be just as evident if only month and year was used. Putting the data in to a histogram is an option it would also not beheltful becuase the information would not be seprated by month(which has unequal days moth to month.

In [153…

```
convert_date(df, 'date')
```

```
<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  datetime64[ns]
 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  float64
 9   view           21534 non-null  float64
 10  condition      21597 non-null  int64
 11  grade          21597 non-null  int64
 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: datetime64[ns](1), float64(8), int64(11), object(1)
memory usage: 3.5+ MB
```

In [154…

```
df['sqft_basement']
```

Out[154…

```
0          0.0
1        400.0
2          0.0
3        910.0
4          0.0
         ...
21592      0.0
21593      0.0
21594      0.0
21595      0.0
21596      0.0
Name: sqft_basement, Length: 21597, dtype: object
```

In [155…

```
df_date = df.sort_values(['date'])
```

In [156…

```
fig, ax = plt.subplots(figsize=(100,10))
sns.countplot(x ='date', ax = ax, data = df_date)
```

```
plt.xticks(rotation=65, horizontalalignment='right')
```

Out[156…

```
(array([  0,   1,   2,   3,   4,   5,   6,   7,   8,   9,  10,  11,  12,
         13,  14,  15,  16,  17,  18,  19,  20,  21,  22,  23,  24,  25,
         26,  27,  28,  29,  30,  31,  32,  33,  34,  35,  36,  37,  38,
         39,  40,  41,  42,  43,  44,  45,  46,  47,  48,  49,  50,  51,
         52,  53,  54,  55,  56,  57,  58,  59,  60,  61,  62,  63,  64,
         65,  66,  67,  68,  69,  70,  71,  72,  73,  74,  75,  76,  77,
         78,  79,  80,  81,  82,  83,  84,  85,  86,  87,  88,  89,  90,
         91,  92,  93,  94,  95,  96,  97,  98,  99, 100, 101, 102, 103,
        104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116,
        117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129,
        130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142,
        143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155,
        156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168,
        169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181,
        182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194,
        195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207,
        208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220,
        221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233,
        234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246,
        247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259,
        260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272,
        273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285,
        286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298,
        299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311,
        312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324,
        325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337,
        338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350,
        351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363,
        364, 365, 366, 367, 368, 369, 370, 371]),
 [Text(0, 0, '2014-05-02T00:00:00.000000000'),
  Text(1, 0, '2014-05-03T00:00:00.000000000'),
  Text(2, 0, '2014-05-04T00:00:00.000000000'),
  Text(3, 0, '2014-05-05T00:00:00.000000000'),
  Text(4, 0, '2014-05-06T00:00:00.000000000'),
  Text(5, 0, '2014-05-07T00:00:00.000000000'),
  Text(6, 0, '2014-05-08T00:00:00.000000000'),
  Text(7, 0, '2014-05-09T00:00:00.000000000'),
  Text(8, 0, '2014-05-10T00:00:00.000000000'),
  Text(9, 0, '2014-05-11T00:00:00.000000000'),
  Text(10, 0, '2014-05-12T00:00:00.000000000'),
  Text(11, 0, '2014-05-13T00:00:00.000000000'),
  Text(12, 0, '2014-05-14T00:00:00.000000000'),
  Text(13, 0, '2014-05-15T00:00:00.000000000'),
  Text(14, 0, '2014-05-16T00:00:00.000000000'),
  Text(15, 0, '2014-05-17T00:00:00.000000000'),
  Text(16, 0, '2014-05-18T00:00:00.000000000'),
  Text(17, 0, '2014-05-19T00:00:00.000000000'),
  Text(18, 0, '2014-05-20T00:00:00.000000000'),
  Text(19, 0, '2014-05-21T00:00:00.000000000'),
  Text(20, 0, '2014-05-22T00:00:00.000000000'),
  Text(21, 0, '2014-05-23T00:00:00.000000000'),
  Text(22, 0, '2014-05-24T00:00:00.000000000'),
  Text(23, 0, '2014-05-25T00:00:00.000000000'),
  Text(24, 0, '2014-05-26T00:00:00.000000000'),
  Text(25, 0, '2014-05-27T00:00:00.000000000'),
  Text(26, 0, '2014-05-28T00:00:00.000000000'),
  Text(27, 0, '2014-05-29T00:00:00.000000000'),
  Text(28, 0, '2014-05-30T00:00:00.000000000'),
  Text(29, 0, '2014-05-31T00:00:00.000000000'),
  Text(30, 0, '2014-06-01T00:00:00.000000000'),
  Text(31, 0, '2014-06-02T00:00:00.000000000'),
  Text(32, 0, '2014-06-03T00:00:00.000000000'),
  Text(33, 0, '2014-06-04T00:00:00.000000000'),
```

```
Text(34, 0, '2014-06-05T00:00:00.000000000'),
Text(35, 0, '2014-06-06T00:00:00.000000000'),
Text(36, 0, '2014-06-07T00:00:00.000000000'),
Text(37, 0, '2014-06-08T00:00:00.000000000'),
Text(38, 0, '2014-06-09T00:00:00.000000000'),
Text(39, 0, '2014-06-10T00:00:00.000000000'),
Text(40, 0, '2014-06-11T00:00:00.000000000'),
Text(41, 0, '2014-06-12T00:00:00.000000000'),
Text(42, 0, '2014-06-13T00:00:00.000000000'),
Text(43, 0, '2014-06-14T00:00:00.000000000'),
Text(44, 0, '2014-06-15T00:00:00.000000000'),
Text(45, 0, '2014-06-16T00:00:00.000000000'),
Text(46, 0, '2014-06-17T00:00:00.000000000'),
Text(47, 0, '2014-06-18T00:00:00.000000000'),
Text(48, 0, '2014-06-19T00:00:00.000000000'),
Text(49, 0, '2014-06-20T00:00:00.000000000'),
Text(50, 0, '2014-06-21T00:00:00.000000000'),
Text(51, 0, '2014-06-22T00:00:00.000000000'),
Text(52, 0, '2014-06-23T00:00:00.000000000'),
Text(53, 0, '2014-06-24T00:00:00.000000000'),
Text(54, 0, '2014-06-25T00:00:00.000000000'),
Text(55, 0, '2014-06-26T00:00:00.000000000'),
Text(56, 0, '2014-06-27T00:00:00.000000000'),
Text(57, 0, '2014-06-28T00:00:00.000000000'),
Text(58, 0, '2014-06-29T00:00:00.000000000'),
Text(59, 0, '2014-06-30T00:00:00.000000000'),
Text(60, 0, '2014-07-01T00:00:00.000000000'),
Text(61, 0, '2014-07-02T00:00:00.000000000'),
Text(62, 0, '2014-07-03T00:00:00.000000000'),
Text(63, 0, '2014-07-04T00:00:00.000000000'),
Text(64, 0, '2014-07-05T00:00:00.000000000'),
Text(65, 0, '2014-07-06T00:00:00.000000000'),
Text(66, 0, '2014-07-07T00:00:00.000000000'),
Text(67, 0, '2014-07-08T00:00:00.000000000'),
Text(68, 0, '2014-07-09T00:00:00.000000000'),
Text(69, 0, '2014-07-10T00:00:00.000000000'),
Text(70, 0, '2014-07-11T00:00:00.000000000'),
Text(71, 0, '2014-07-12T00:00:00.000000000'),
Text(72, 0, '2014-07-13T00:00:00.000000000'),
Text(73, 0, '2014-07-14T00:00:00.000000000'),
Text(74, 0, '2014-07-15T00:00:00.000000000'),
Text(75, 0, '2014-07-16T00:00:00.000000000'),
Text(76, 0, '2014-07-17T00:00:00.000000000'),
Text(77, 0, '2014-07-18T00:00:00.000000000'),
Text(78, 0, '2014-07-19T00:00:00.000000000'),
Text(79, 0, '2014-07-20T00:00:00.000000000'),
Text(80, 0, '2014-07-21T00:00:00.000000000'),
Text(81, 0, '2014-07-22T00:00:00.000000000'),
Text(82, 0, '2014-07-23T00:00:00.000000000'),
Text(83, 0, '2014-07-24T00:00:00.000000000'),
Text(84, 0, '2014-07-25T00:00:00.000000000'),
Text(85, 0, '2014-07-26T00:00:00.000000000'),
Text(86, 0, '2014-07-27T00:00:00.000000000'),
Text(87, 0, '2014-07-28T00:00:00.000000000'),
Text(88, 0, '2014-07-29T00:00:00.000000000'),
Text(89, 0, '2014-07-30T00:00:00.000000000'),
Text(90, 0, '2014-07-31T00:00:00.000000000'),
Text(91, 0, '2014-08-01T00:00:00.000000000'),
Text(92, 0, '2014-08-02T00:00:00.000000000'),
Text(93, 0, '2014-08-03T00:00:00.000000000'),
Text(94, 0, '2014-08-04T00:00:00.000000000'),
Text(95, 0, '2014-08-05T00:00:00.000000000'),
Text(96, 0, '2014-08-06T00:00:00.000000000'),
Text(97, 0, '2014-08-07T00:00:00.000000000'),
Text(98, 0, '2014-08-08T00:00:00.000000000'),
```

```
Text(99, 0, '2014-08-09T00:00:00.000000000'),
Text(100, 0, '2014-08-10T00:00:00.000000000'),
Text(101, 0, '2014-08-11T00:00:00.000000000'),
Text(102, 0, '2014-08-12T00:00:00.000000000'),
Text(103, 0, '2014-08-13T00:00:00.000000000'),
Text(104, 0, '2014-08-14T00:00:00.000000000'),
Text(105, 0, '2014-08-15T00:00:00.000000000'),
Text(106, 0, '2014-08-16T00:00:00.000000000'),
Text(107, 0, '2014-08-17T00:00:00.000000000'),
Text(108, 0, '2014-08-18T00:00:00.000000000'),
Text(109, 0, '2014-08-19T00:00:00.000000000'),
Text(110, 0, '2014-08-20T00:00:00.000000000'),
Text(111, 0, '2014-08-21T00:00:00.000000000'),
Text(112, 0, '2014-08-22T00:00:00.000000000'),
Text(113, 0, '2014-08-23T00:00:00.000000000'),
Text(114, 0, '2014-08-24T00:00:00.000000000'),
Text(115, 0, '2014-08-25T00:00:00.000000000'),
Text(116, 0, '2014-08-26T00:00:00.000000000'),
Text(117, 0, '2014-08-27T00:00:00.000000000'),
Text(118, 0, '2014-08-28T00:00:00.000000000'),
Text(119, 0, '2014-08-29T00:00:00.000000000'),
Text(120, 0, '2014-08-30T00:00:00.000000000'),
Text(121, 0, '2014-08-31T00:00:00.000000000'),
Text(122, 0, '2014-09-01T00:00:00.000000000'),
Text(123, 0, '2014-09-02T00:00:00.000000000'),
Text(124, 0, '2014-09-03T00:00:00.000000000'),
Text(125, 0, '2014-09-04T00:00:00.000000000'),
Text(126, 0, '2014-09-05T00:00:00.000000000'),
Text(127, 0, '2014-09-06T00:00:00.000000000'),
Text(128, 0, '2014-09-07T00:00:00.000000000'),
Text(129, 0, '2014-09-08T00:00:00.000000000'),
Text(130, 0, '2014-09-09T00:00:00.000000000'),
Text(131, 0, '2014-09-10T00:00:00.000000000'),
Text(132, 0, '2014-09-11T00:00:00.000000000'),
Text(133, 0, '2014-09-12T00:00:00.000000000'),
Text(134, 0, '2014-09-13T00:00:00.000000000'),
Text(135, 0, '2014-09-14T00:00:00.000000000'),
Text(136, 0, '2014-09-15T00:00:00.000000000'),
Text(137, 0, '2014-09-16T00:00:00.000000000'),
Text(138, 0, '2014-09-17T00:00:00.000000000'),
Text(139, 0, '2014-09-18T00:00:00.000000000'),
Text(140, 0, '2014-09-19T00:00:00.000000000'),
Text(141, 0, '2014-09-20T00:00:00.000000000'),
Text(142, 0, '2014-09-21T00:00:00.000000000'),
Text(143, 0, '2014-09-22T00:00:00.000000000'),
Text(144, 0, '2014-09-23T00:00:00.000000000'),
Text(145, 0, '2014-09-24T00:00:00.000000000'),
Text(146, 0, '2014-09-25T00:00:00.000000000'),
Text(147, 0, '2014-09-26T00:00:00.000000000'),
Text(148, 0, '2014-09-27T00:00:00.000000000'),
Text(149, 0, '2014-09-28T00:00:00.000000000'),
Text(150, 0, '2014-09-29T00:00:00.000000000'),
Text(151, 0, '2014-09-30T00:00:00.000000000'),
Text(152, 0, '2014-10-01T00:00:00.000000000'),
Text(153, 0, '2014-10-02T00:00:00.000000000'),
Text(154, 0, '2014-10-03T00:00:00.000000000'),
Text(155, 0, '2014-10-04T00:00:00.000000000'),
Text(156, 0, '2014-10-05T00:00:00.000000000'),
Text(157, 0, '2014-10-06T00:00:00.000000000'),
Text(158, 0, '2014-10-07T00:00:00.000000000'),
Text(159, 0, '2014-10-08T00:00:00.000000000'),
Text(160, 0, '2014-10-09T00:00:00.000000000'),
Text(161, 0, '2014-10-10T00:00:00.000000000'),
Text(162, 0, '2014-10-11T00:00:00.000000000'),
Text(163, 0, '2014-10-12T00:00:00.000000000'),
```

```
Text(164, 0, '2014-10-13T00:00:00.000000000'),
Text(165, 0, '2014-10-14T00:00:00.000000000'),
Text(166, 0, '2014-10-15T00:00:00.000000000'),
Text(167, 0, '2014-10-16T00:00:00.000000000'),
Text(168, 0, '2014-10-17T00:00:00.000000000'),
Text(169, 0, '2014-10-18T00:00:00.000000000'),
Text(170, 0, '2014-10-19T00:00:00.000000000'),
Text(171, 0, '2014-10-20T00:00:00.000000000'),
Text(172, 0, '2014-10-21T00:00:00.000000000'),
Text(173, 0, '2014-10-22T00:00:00.000000000'),
Text(174, 0, '2014-10-23T00:00:00.000000000'),
Text(175, 0, '2014-10-24T00:00:00.000000000'),
Text(176, 0, '2014-10-25T00:00:00.000000000'),
Text(177, 0, '2014-10-26T00:00:00.000000000'),
Text(178, 0, '2014-10-27T00:00:00.000000000'),
Text(179, 0, '2014-10-28T00:00:00.000000000'),
Text(180, 0, '2014-10-29T00:00:00.000000000'),
Text(181, 0, '2014-10-30T00:00:00.000000000'),
Text(182, 0, '2014-10-31T00:00:00.000000000'),
Text(183, 0, '2014-11-01T00:00:00.000000000'),
Text(184, 0, '2014-11-02T00:00:00.000000000'),
Text(185, 0, '2014-11-03T00:00:00.000000000'),
Text(186, 0, '2014-11-04T00:00:00.000000000'),
Text(187, 0, '2014-11-05T00:00:00.000000000'),
Text(188, 0, '2014-11-06T00:00:00.000000000'),
Text(189, 0, '2014-11-07T00:00:00.000000000'),
Text(190, 0, '2014-11-08T00:00:00.000000000'),
Text(191, 0, '2014-11-09T00:00:00.000000000'),
Text(192, 0, '2014-11-10T00:00:00.000000000'),
Text(193, 0, '2014-11-11T00:00:00.000000000'),
Text(194, 0, '2014-11-12T00:00:00.000000000'),
Text(195, 0, '2014-11-13T00:00:00.000000000'),
Text(196, 0, '2014-11-14T00:00:00.000000000'),
Text(197, 0, '2014-11-15T00:00:00.000000000'),
Text(198, 0, '2014-11-16T00:00:00.000000000'),
Text(199, 0, '2014-11-17T00:00:00.000000000'),
Text(200, 0, '2014-11-18T00:00:00.000000000'),
Text(201, 0, '2014-11-19T00:00:00.000000000'),
Text(202, 0, '2014-11-20T00:00:00.000000000'),
Text(203, 0, '2014-11-21T00:00:00.000000000'),
Text(204, 0, '2014-11-22T00:00:00.000000000'),
Text(205, 0, '2014-11-23T00:00:00.000000000'),
Text(206, 0, '2014-11-24T00:00:00.000000000'),
Text(207, 0, '2014-11-25T00:00:00.000000000'),
Text(208, 0, '2014-11-26T00:00:00.000000000'),
Text(209, 0, '2014-11-28T00:00:00.000000000'),
Text(210, 0, '2014-11-29T00:00:00.000000000'),
Text(211, 0, '2014-11-30T00:00:00.000000000'),
Text(212, 0, '2014-12-01T00:00:00.000000000'),
Text(213, 0, '2014-12-02T00:00:00.000000000'),
Text(214, 0, '2014-12-03T00:00:00.000000000'),
Text(215, 0, '2014-12-04T00:00:00.000000000'),
Text(216, 0, '2014-12-05T00:00:00.000000000'),
Text(217, 0, '2014-12-06T00:00:00.000000000'),
Text(218, 0, '2014-12-07T00:00:00.000000000'),
Text(219, 0, '2014-12-08T00:00:00.000000000'),
Text(220, 0, '2014-12-09T00:00:00.000000000'),
Text(221, 0, '2014-12-10T00:00:00.000000000'),
Text(222, 0, '2014-12-11T00:00:00.000000000'),
Text(223, 0, '2014-12-12T00:00:00.000000000'),
Text(224, 0, '2014-12-13T00:00:00.000000000'),
Text(225, 0, '2014-12-14T00:00:00.000000000'),
Text(226, 0, '2014-12-15T00:00:00.000000000'),
Text(227, 0, '2014-12-16T00:00:00.000000000'),
Text(228, 0, '2014-12-17T00:00:00.000000000'),
```

```
Text(229, 0, '2014-12-18T00:00:00.000000000'),
Text(230, 0, '2014-12-19T00:00:00.000000000'),
Text(231, 0, '2014-12-20T00:00:00.000000000'),
Text(232, 0, '2014-12-21T00:00:00.000000000'),
Text(233, 0, '2014-12-22T00:00:00.000000000'),
Text(234, 0, '2014-12-23T00:00:00.000000000'),
Text(235, 0, '2014-12-24T00:00:00.000000000'),
Text(236, 0, '2014-12-26T00:00:00.000000000'),
Text(237, 0, '2014-12-27T00:00:00.000000000'),
Text(238, 0, '2014-12-29T00:00:00.000000000'),
Text(239, 0, '2014-12-30T00:00:00.000000000'),
Text(240, 0, '2014-12-31T00:00:00.000000000'),
Text(241, 0, '2015-01-02T00:00:00.000000000'),
Text(242, 0, '2015-01-05T00:00:00.000000000'),
Text(243, 0, '2015-01-06T00:00:00.000000000'),
Text(244, 0, '2015-01-07T00:00:00.000000000'),
Text(245, 0, '2015-01-08T00:00:00.000000000'),
Text(246, 0, '2015-01-09T00:00:00.000000000'),
Text(247, 0, '2015-01-10T00:00:00.000000000'),
Text(248, 0, '2015-01-12T00:00:00.000000000'),
Text(249, 0, '2015-01-13T00:00:00.000000000'),
Text(250, 0, '2015-01-14T00:00:00.000000000'),
Text(251, 0, '2015-01-15T00:00:00.000000000'),
Text(252, 0, '2015-01-16T00:00:00.000000000'),
Text(253, 0, '2015-01-17T00:00:00.000000000'),
Text(254, 0, '2015-01-19T00:00:00.000000000'),
Text(255, 0, '2015-01-20T00:00:00.000000000'),
Text(256, 0, '2015-01-21T00:00:00.000000000'),
Text(257, 0, '2015-01-22T00:00:00.000000000'),
Text(258, 0, '2015-01-23T00:00:00.000000000'),
Text(259, 0, '2015-01-24T00:00:00.000000000'),
Text(260, 0, '2015-01-25T00:00:00.000000000'),
Text(261, 0, '2015-01-26T00:00:00.000000000'),
Text(262, 0, '2015-01-27T00:00:00.000000000'),
Text(263, 0, '2015-01-28T00:00:00.000000000'),
Text(264, 0, '2015-01-29T00:00:00.000000000'),
Text(265, 0, '2015-01-30T00:00:00.000000000'),
Text(266, 0, '2015-01-31T00:00:00.000000000'),
Text(267, 0, '2015-02-01T00:00:00.000000000'),
Text(268, 0, '2015-02-02T00:00:00.000000000'),
Text(269, 0, '2015-02-03T00:00:00.000000000'),
Text(270, 0, '2015-02-04T00:00:00.000000000'),
Text(271, 0, '2015-02-05T00:00:00.000000000'),
Text(272, 0, '2015-02-06T00:00:00.000000000'),
Text(273, 0, '2015-02-07T00:00:00.000000000'),
Text(274, 0, '2015-02-09T00:00:00.000000000'),
Text(275, 0, '2015-02-10T00:00:00.000000000'),
Text(276, 0, '2015-02-11T00:00:00.000000000'),
Text(277, 0, '2015-02-12T00:00:00.000000000'),
Text(278, 0, '2015-02-13T00:00:00.000000000'),
Text(279, 0, '2015-02-14T00:00:00.000000000'),
Text(280, 0, '2015-02-15T00:00:00.000000000'),
Text(281, 0, '2015-02-16T00:00:00.000000000'),
Text(282, 0, '2015-02-17T00:00:00.000000000'),
Text(283, 0, '2015-02-18T00:00:00.000000000'),
Text(284, 0, '2015-02-19T00:00:00.000000000'),
Text(285, 0, '2015-02-20T00:00:00.000000000'),
Text(286, 0, '2015-02-21T00:00:00.000000000'),
Text(287, 0, '2015-02-22T00:00:00.000000000'),
Text(288, 0, '2015-02-23T00:00:00.000000000'),
Text(289, 0, '2015-02-24T00:00:00.000000000'),
Text(290, 0, '2015-02-25T00:00:00.000000000'),
Text(291, 0, '2015-02-26T00:00:00.000000000'),
Text(292, 0, '2015-02-27T00:00:00.000000000'),
Text(293, 0, '2015-02-28T00:00:00.000000000'),
```

```
Text(294, 0, '2015-03-01T00:00:00.000000000'),
Text(295, 0, '2015-03-02T00:00:00.000000000'),
Text(296, 0, '2015-03-03T00:00:00.000000000'),
Text(297, 0, '2015-03-04T00:00:00.000000000'),
Text(298, 0, '2015-03-05T00:00:00.000000000'),
Text(299, 0, '2015-03-06T00:00:00.000000000'),
Text(300, 0, '2015-03-07T00:00:00.000000000'),
Text(301, 0, '2015-03-08T00:00:00.000000000'),
Text(302, 0, '2015-03-09T00:00:00.000000000'),
Text(303, 0, '2015-03-10T00:00:00.000000000'),
Text(304, 0, '2015-03-11T00:00:00.000000000'),
Text(305, 0, '2015-03-12T00:00:00.000000000'),
Text(306, 0, '2015-03-13T00:00:00.000000000'),
Text(307, 0, '2015-03-14T00:00:00.000000000'),
Text(308, 0, '2015-03-15T00:00:00.000000000'),
Text(309, 0, '2015-03-16T00:00:00.000000000'),
Text(310, 0, '2015-03-17T00:00:00.000000000'),
Text(311, 0, '2015-03-18T00:00:00.000000000'),
Text(312, 0, '2015-03-19T00:00:00.000000000'),
Text(313, 0, '2015-03-20T00:00:00.000000000'),
Text(314, 0, '2015-03-21T00:00:00.000000000'),
Text(315, 0, '2015-03-22T00:00:00.000000000'),
Text(316, 0, '2015-03-23T00:00:00.000000000'),
Text(317, 0, '2015-03-24T00:00:00.000000000'),
Text(318, 0, '2015-03-25T00:00:00.000000000'),
Text(319, 0, '2015-03-26T00:00:00.000000000'),
Text(320, 0, '2015-03-27T00:00:00.000000000'),
Text(321, 0, '2015-03-28T00:00:00.000000000'),
Text(322, 0, '2015-03-29T00:00:00.000000000'),
Text(323, 0, '2015-03-30T00:00:00.000000000'),
Text(324, 0, '2015-03-31T00:00:00.000000000'),
Text(325, 0, '2015-04-01T00:00:00.000000000'),
Text(326, 0, '2015-04-02T00:00:00.000000000'),
Text(327, 0, '2015-04-03T00:00:00.000000000'),
Text(328, 0, '2015-04-04T00:00:00.000000000'),
Text(329, 0, '2015-04-05T00:00:00.000000000'),
Text(330, 0, '2015-04-06T00:00:00.000000000'),
Text(331, 0, '2015-04-07T00:00:00.000000000'),
Text(332, 0, '2015-04-08T00:00:00.000000000'),
Text(333, 0, '2015-04-09T00:00:00.000000000'),
Text(334, 0, '2015-04-10T00:00:00.000000000'),
Text(335, 0, '2015-04-11T00:00:00.000000000'),
Text(336, 0, '2015-04-12T00:00:00.000000000'),
Text(337, 0, '2015-04-13T00:00:00.000000000'),
Text(338, 0, '2015-04-14T00:00:00.000000000'),
Text(339, 0, '2015-04-15T00:00:00.000000000'),
Text(340, 0, '2015-04-16T00:00:00.000000000'),
Text(341, 0, '2015-04-17T00:00:00.000000000'),
Text(342, 0, '2015-04-18T00:00:00.000000000'),
Text(343, 0, '2015-04-19T00:00:00.000000000'),
Text(344, 0, '2015-04-20T00:00:00.000000000'),
Text(345, 0, '2015-04-21T00:00:00.000000000'),
Text(346, 0, '2015-04-22T00:00:00.000000000'),
Text(347, 0, '2015-04-23T00:00:00.000000000'),
Text(348, 0, '2015-04-24T00:00:00.000000000'),
Text(349, 0, '2015-04-25T00:00:00.000000000'),
Text(350, 0, '2015-04-26T00:00:00.000000000'),
Text(351, 0, '2015-04-27T00:00:00.000000000'),
Text(352, 0, '2015-04-28T00:00:00.000000000'),
Text(353, 0, '2015-04-29T00:00:00.000000000'),
Text(354, 0, '2015-04-30T00:00:00.000000000'),
Text(355, 0, '2015-05-01T00:00:00.000000000'),
Text(356, 0, '2015-05-02T00:00:00.000000000'),
Text(357, 0, '2015-05-03T00:00:00.000000000'),
Text(358, 0, '2015-05-04T00:00:00.000000000'),
```

```
 Text(359, 0, '2015-05-05T00:00:00.000000000'),
 Text(360, 0, '2015-05-06T00:00:00.000000000'),
 Text(361, 0, '2015-05-07T00:00:00.000000000'),
 Text(362, 0, '2015-05-08T00:00:00.000000000'),
 Text(363, 0, '2015-05-09T00:00:00.000000000'),
 Text(364, 0, '2015-05-10T00:00:00.000000000'),
 Text(365, 0, '2015-05-11T00:00:00.000000000'),
 Text(366, 0, '2015-05-12T00:00:00.000000000'),
 Text(367, 0, '2015-05-13T00:00:00.000000000'),
 Text(368, 0, '2015-05-14T00:00:00.000000000'),
 Text(369, 0, '2015-05-15T00:00:00.000000000'),
 Text(370, 0, '2015-05-24T00:00:00.000000000'),
 Text(371, 0, '2015-05-27T00:00:00.000000000')])
```

Interesting findings

- Very few houses get closed on the weekend, at the same time closing date is arbitary since there is so much that goes on in order to close a house.
- There is a fluxuation of number of homes sold theought out these 2 years. shouls group by month to see if there is any futher trends
- The range og the data is from 05-2014 to 05-2015

In [157…

```
df['month_year'] = pd.to_datetime(df_date['date']).dt.strftime('%Y%m')
df_date['month'] = pd.to_datetime(df_date['date']).dt.strftime('%Y-%m')
df['date'] = pd.to_datetime(df['date']).dt.strftime('%Y-%m')
```

In [158…

```
fig, ax = plt.subplots(figsize=(10,10))
sns.countplot(x ='month', ax = ax, data = df_date)

plt.xticks(rotation=65, horizontalalignment='right')
```

Out[158…

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12]),
 [Text(0, 0, '2014-05'),
  Text(1, 0, '2014-06'),
  Text(2, 0, '2014-07'),
  Text(3, 0, '2014-08'),
  Text(4, 0, '2014-09'),
  Text(5, 0, '2014-10'),
  Text(6, 0, '2014-11'),
  Text(7, 0, '2014-12'),
  Text(8, 0, '2015-01'),
  Text(9, 0, '2015-02'),
  Text(10, 0, '2015-03'),
  Text(11, 0, '2015-04'),
  Text(12, 0, '2015-05')])
```

- It look like as the months move closer to the summer time there are more homes being sold.
- It looke like the lost month had a surprising low amount of houses sold for it to be the summer but that could be the result of incomplete data, since the time between when people close on there home and when that information is public has a time gap.
- should in investigate next if time of year had an effect on price.

In [159…

```
dff = df[['price', 'date']].sort_values('date')
```

In [160…

```
figure, ax= plt.subplots(figsize = (100,20))
sns.barplot(x = 'date', y = 'price', data = dff)
plt.xticks(rotation=65, horizontalalignment='right')
```

Out[160…

```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12]),
 [Text(0, 0, '2014-05'),
  Text(1, 0, '2014-06'),
  Text(2, 0, '2014-07'),
  Text(3, 0, '2014-08'),
  Text(4, 0, '2014-09'),
```

```
 Text(5, 0, '2014-10'),
 Text(6, 0, '2014-11'),
 Text(7, 0, '2014-12'),
 Text(8, 0, '2015-01'),
 Text(9, 0, '2015-02'),
 Text(10, 0, '2015-03'),
 Text(11, 0, '2015-04'),
 Text(12, 0, '2015-05')])
```

In [161…

```
figure, ax= plt.subplots(figsize = (10,1))
sns.barplot(x = 'month', y = 'price', data = df_date)
```

Out[161…

```
<AxesSubplot:xlabel='month', ylabel='price'>
```

- The above graph shows that homes sell for slightly more in the summer months. Note there was a peek in October which is a result of the home(s) sold on October 11, 2014 which totaled about 2 million dollars.
- Since the month_year column is a good representiation of homes sold over time the date column is redundant and will be droped.

In [162…

```
df = df.drop('date', axis = 1)
```

In [163…

```
convert_int(df, 'month_year')
```

```
<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   price          21597 non-null  float64
 2   bedrooms       21597 non-null  int64
 3   bathrooms      21597 non-null  float64
 4   sqft_living    21597 non-null  int64
 5   sqft_lot       21597 non-null  int64
 6   floors         21597 non-null  float64
 7   waterfront     19221 non-null  float64
 8   view           21534 non-null  float64
 9   condition      21597 non-null  int64
 10  grade          21597 non-null  int64
 11  sqft_above     21597 non-null  int64
 12  sqft_basement  21597 non-null  object
 13  yr_built       21597 non-null  int64
 14  yr_renovated   17755 non-null  float64
 15  zipcode        21597 non-null  int64
```

```
 16  lat            21597 non-null  float64
 17  long           21597 non-null  float64
 18  sqft_living15  21597 non-null  int64
 19  sqft_lot15     21597 non-null  int64
 20  month_year     21597 non-null  int32
dtypes: float64(8), int32(1), int64(11), object(1)
memory usage: 3.4+ MB
```

In [164…

```
df['sqft_basement'].replace(to_replace =["?"], value ="0", inplace=True)
#df[['sqft_basement']].value_counts(ascending=False)
```

In [165…

```
df[['sqft_basement']].value_counts(ascending=False)
```

Out[165…

```
sqft_basement
0.0              12826
0                  454
600.0              217
500.0              209
700.0              208
                 ...
225.0                1
2240.0               1
2196.0               1
2190.0               1
2180.0               1
Length: 304, dtype: int64
```

In [166…

```
convert_float(df, 'sqft_basement')
df.info()
```

```
<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   price          21597 non-null  float64
 2   bedrooms       21597 non-null  int64
 3   bathrooms      21597 non-null  float64
 4   sqft_living    21597 non-null  int64
 5   sqft_lot       21597 non-null  int64
 6   floors         21597 non-null  float64
 7   waterfront     19221 non-null  float64
 8   view           21534 non-null  float64
 9   condition      21597 non-null  int64
 10  grade          21597 non-null  int64
 11  sqft_above     21597 non-null  int64
 12  sqft_basement  21597 non-null  float64
 13  yr_built       21597 non-null  int64
 14  yr_renovated   17755 non-null  float64
 15  zipcode        21597 non-null  int64
 16  lat            21597 non-null  float64
 17  long           21597 non-null  float64
 18  sqft_living15  21597 non-null  int64
 19  sqft_lot15     21597 non-null  int64
 20  month_year     21597 non-null  int32
dtypes: float64(9), int32(1), int64(11)
memory usage: 3.4 MB
```

In [167…

```
df_sqft = df[['sqft_living', 'sqft_lot', 'sqft_above', 'sqft_basement', 'sqft_living15'
df_sqft.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21597 entries, 0 to 21596
```

```
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   sqft_living    21597 non-null  int64
 1   sqft_lot       21597 non-null  int64
 2   sqft_above     21597 non-null  int64
 3   sqft_basement  21597 non-null  float64
 4   sqft_living15  21597 non-null  int64
 5   sqft_lot15     21597 non-null  int64
dtypes: float64(1), int64(5)
memory usage: 1012.5 KB
```

In [168…

```
df_sqft['base_above'] = df_sqft['sqft_above'] + df_sqft['sqft_basement']
df_sqft[['sqft_living', 'base_above']]
```

```
<ipython-input-168-701d7f16f4a9>:1: 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-a-view-versus-a-copy
  df_sqft['base_above'] = df_sqft['sqft_above'] + df_sqft['sqft_basement']
```

Out[168…

| | sqft_living | base_above |
|---|---|---|
| **0** | 1180 | 1180.0 |
| **1** | 2570 | 2570.0 |
| **2** | 770 | 770.0 |
| **3** | 1960 | 1960.0 |
| **4** | 1680 | 1680.0 |
| **...** | ... | ... |
| **21592** | 1530 | 1530.0 |
| **21593** | 2310 | 2310.0 |
| **21594** | 1020 | 1020.0 |
| **21595** | 1600 | 1600.0 |
| **21596** | 1020 | 1020.0 |

21597 rows × 2 columns

In [169…

```
figure, ax= plt.subplots(figsize = (10,10))
sns.scatterplot( x = 'sqft_living', y = 'base_above' , data = df_sqft)
```

Out[169…

```
<AxesSubplot:xlabel='sqft_living', ylabel='base_above'>
```

from the chart and from the graph it shows that sqft_above and sqft_basement add together to equal sqft living. The ones that are below the line are the homes that didnt have sqft_basment listed. From here sqft_basement can be recalculated to the correct values by subtracting sqft_above from sdft_living.

In [170…

```
df['new_sqft_basement'] = df['sqft_living'] - df['sqft_above']
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21597 entries, 0 to 21596
Data columns (total 22 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 21597 non-null  int64
 1   price              21597 non-null  float64
 2   bedrooms           21597 non-null  int64
 3   bathrooms          21597 non-null  float64
 4   sqft_living        21597 non-null  int64
 5   sqft_lot           21597 non-null  int64
 6   floors             21597 non-null  float64
 7   waterfront         19221 non-null  float64
 8   view               21534 non-null  float64
 9   condition          21597 non-null  int64
```

```
 10  grade              21597 non-null  int64
 11  sqft_above         21597 non-null  int64
 12  sqft_basement      21597 non-null  float64
 13  yr_built           21597 non-null  int64
 14  yr_renovated       17755 non-null  float64
 15  zipcode            21597 non-null  int64
 16  lat                21597 non-null  float64
 17  long               21597 non-null  float64
 18  sqft_living15      21597 non-null  int64
 19  sqft_lot15         21597 non-null  int64
 20  month_year         21597 non-null  int32
 21  new_sqft_basement  21597 non-null  int64
dtypes: float64(9), int32(1), int64(12)
memory usage: 3.5 MB
```

In [171…

```
df_sqft = df[['sqft_living', 'sqft_lot', 'sqft_above', 'sqft_basement', 'sqft_living15'
```

In [172…

```
df_sqft['new_base_above'] = df_sqft['sqft_above'] + df_sqft['new_sqft_basement']
```

```
<ipython-input-172-8178a0893752>:1: 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-a-view-versus-a-copy
  df_sqft['new_base_above'] = df_sqft['sqft_above'] + df_sqft['new_sqft_basement']
```

In [173…

```
figure, ax= plt.subplots(figsize = (10,10))
sns.scatterplot( x = 'sqft_living' , y ='new_base_above' , data = df_sqft)
```

Out[173…

```
<AxesSubplot:xlabel='sqft_living', ylabel='new_base_above'>
```

In [174…
```
abs = df[['id', 'price']]
```

In [175…
```
figure, ax= plt.subplots(figsize = (10,10))
sns.histplot(data = df, x = df['price'])
```

Out[175…
```
<AxesSubplot:xlabel='price', ylabel='Count'>
```

sqft living can now be droped since is already represent with sqft_above and sqft_basemant.

In [176…
```
df = df.drop('sqft_living', axis = 1)
```

In [177…
```
df = df.drop('sqft_basement', axis = 1)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21597 entries, 0 to 21596
Data columns (total 20 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 21597 non-null  int64
 1   price              21597 non-null  float64
 2   bedrooms           21597 non-null  int64
 3   bathrooms          21597 non-null  float64
 4   sqft_lot           21597 non-null  int64
 5   floors             21597 non-null  float64
 6   waterfront         19221 non-null  float64
 7   view               21534 non-null  float64
 8   condition          21597 non-null  int64
 9   grade              21597 non-null  int64
 10  sqft_above         21597 non-null  int64
 11  yr_built           21597 non-null  int64
```

```
 12  yr_renovated       17755 non-null  float64
 13  zipcode            21597 non-null  int64
 14  lat                21597 non-null  float64
 15  long               21597 non-null  float64
 16  sqft_living15      21597 non-null  int64
 17  sqft_lot15         21597 non-null  int64
 18  month_year         21597 non-null  int32
 19  new_sqft_basement  21597 non-null  int64
dtypes: float64(8), int32(1), int64(11)
memory usage: 3.2 MB
```

In [178…

```
figure, ax= plt.subplots(figsize = (100,10))
sns.barplot( x = df['yr_built'] , y ='price' , data = df )
```

Out[178…

```
<AxesSubplot:xlabel='yr_built', ylabel='price'>
```

In [179…

```
df.describe()
```

Out[179…

| | id | price | bedrooms | bathrooms | sqft_lot | floors | waterfro |
|---|---|---|---|---|---|---|---|
| **count** | 2.159700e+04 | 2.159700e+04 | 21597.000000 | 21597.000000 | 2.159700e+04 | 21597.000000 | 19221.0000 |
| **mean** | 4.580474e+09 | 5.402966e+05 | 3.373200 | 2.115826 | 1.509941e+04 | 1.494096 | 0.0075 |
| **std** | 2.876736e+09 | 3.673681e+05 | 0.926299 | 0.768984 | 4.141264e+04 | 0.539683 | 0.0868 |
| **min** | 1.000102e+06 | 7.800000e+04 | 1.000000 | 0.500000 | 5.200000e+02 | 1.000000 | 0.0000 |
| **25%** | 2.123049e+09 | 3.220000e+05 | 3.000000 | 1.750000 | 5.040000e+03 | 1.000000 | 0.0000 |
| **50%** | 3.904930e+09 | 4.500000e+05 | 3.000000 | 2.250000 | 7.618000e+03 | 1.500000 | 0.0000 |
| **75%** | 7.308900e+09 | 6.450000e+05 | 4.000000 | 2.500000 | 1.068500e+04 | 2.000000 | 0.0000 |
| **max** | 9.900000e+09 | 7.700000e+06 | 33.000000 | 8.000000 | 1.651359e+06 | 3.500000 | 1.0000 |

In [180…

```
257/21597
```

Out[180…

```
0.011899800898272908
```

In [181…

```
df_count = df.loc[df['yr_renovated'] > 2004]
df_count.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 265 entries, 103 to 20946
Data columns (total 20 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 265 non-null    int64
 1   price              265 non-null    float64
 2   bedrooms           265 non-null    int64
 3   bathrooms          265 non-null    float64
 4   sqft_lot           265 non-null    int64
 5   floors             265 non-null    float64
 6   waterfront         237 non-null    float64
```

```
 7   view               265 non-null    float64
 8   condition          265 non-null    int64
 9   grade              265 non-null    int64
 10  sqft_above         265 non-null    int64
 11  yr_built           265 non-null    int64
 12  yr_renovated       265 non-null    float64
 13  zipcode            265 non-null    int64
 14  lat                265 non-null    float64
 15  long               265 non-null    float64
 16  sqft_living15      265 non-null    int64
 17  sqft_lot15         265 non-null    int64
 18  month_year         265 non-null    int32
 19  new_sqft_basement  265 non-null    int64
dtypes: float64(8), int32(1), int64(11)
memory usage: 42.4 KB
```

In [ ]:

In [182…

```
df1 = df.loc[df['yr_renovated'] > 1]
figure, ax= plt.subplots(figsize = (20,10))
sns.lineplot( x = 'yr_renovated', y ='price' , hue='grade', legend=True,
                palette="tab10",  data = df1 )
```

Out[182…

```
<AxesSubplot:xlabel='yr_renovated', ylabel='price'>
```

In [183…

```
df1 = df.loc[df['yr_renovated'] > 1]
df_grade_under_nine = df1.loc[df1['grade'] < 9]
figure, ax= plt.subplots(figsize = (10, 5))
sns.lineplot( x = 'yr_renovated', y ='price' , hue='grade', legend=True,
                palette="tab10",  data = df_grade_under_nine )
df1_2 = df.loc[df['yr_renovated'] > 1]
df_grade_under_nine_2 = df1_2.loc[df1_2['grade'] > 8]
figure, ax= plt.subplots(figsize = (10,5))
sns.lineplot( x = 'yr_renovated', y ='price' , hue='grade', legend=True,
                palette="tab10",  data = df_grade_under_nine_2 )
```

Out[183…

```
<AxesSubplot:xlabel='yr_renovated', ylabel='price'>
```

In [184…

```
ddd = df.loc[df['grade'] == 9]
print(ddd)
```

```
               id      price  bedrooms  bathrooms  sqft_lot  floors  \
15     9297300055   650000.0         4       3.00      5000     2.0
21     2524049179  2000000.0         3       2.75     44867     1.0
40     5547700270   625000.0         4       2.50      5520     2.0
42     7203220400   861990.0         5       2.75      5639     2.0
47     4178300310   785000.0         4       2.50     13416     2.0
...           ...        ...       ...        ...       ...     ...
21580  7502800100   679950.0         5       2.75      9437     2.0
21582  8956200760   541800.0         4       2.50      7866     2.0
21583  7202300110   810000.0         4       3.00      7838     2.0
21589  3448900210   610685.0         4       2.50      6023     2.0
21590  7936000429  1010000.0         4       3.50      7200     2.0

       waterfront  view  condition  grade  sqft_above  yr_built  yr_renovated  \
15             0.0   3.0          3      9        1980      1979           0.0
21             0.0   4.0          3      9        2330      1968           0.0
```

```
40            NaN   0.0          3      9        2570      2000           NaN
42            0.0   0.0          3      9        3595      2014           0.0
47            0.0   0.0          4      9        2290      1981           0.0
...           ...   ...        ...    ...         ...       ...           ...
21580         0.0   0.0          3      9        3600      2014           0.0
21582         NaN   2.0          3      9        3118      2014           0.0
21583         0.0   0.0          3      9        3990      2003           NaN
21589         0.0   NaN          3      9        2520      2014           0.0
21590         0.0   0.0          3      9        2600      2009           0.0

       zipcode      lat     long  sqft_living15  sqft_lot15  month_year  \
15       98126  47.5714 -122.375           2140        4000      201501
21       98040  47.5316 -122.233           4110       20336      201408
40       98074  47.6145 -122.027           2470        5669      201407
42       98053  47.6848 -122.016           3625        5639      201407
47       98007  47.6194 -122.151           2680       13685      201407
...        ...      ...      ...            ...         ...         ...
21580    98059  47.4822 -122.131           3550        9421      201408
21582    98001  47.2931 -122.264           2673        6500      201410
21583    98053  47.6857 -122.046           3370        6814      201409
21589    98056  47.5137 -122.167           2520        6023      201410
21590    98136  47.5537 -122.398           2050        6200      201503

       new_sqft_basement
15                   970
21                   720
40                     0
42                     0
47                     0
...                  ...
21580                  0
21582                  0
21583                  0
21589                  0
21590                910

[2615 rows x 20 columns]
```

In [185…

```
df1 = df.loc[df['yr_renovated'] > 1]
df_grade_under_nine = df1.loc[df1['grade'] < 14]
figure, ax= plt.subplots(figsize = (10, 5))
sns.boxplot( x = 'grade', y ='price' , hue='grade',
                palette="tab10",  data = df_grade_under_nine )
```

Out[185…

```
<AxesSubplot:xlabel='grade', ylabel='price'>
```

In [186…

```
df_columns = ddd.columns
for i in df_columns:
    print('\033[1m' + i.upper() + '\033[0m')
    print(ddd[[i]].value_counts(ascending=False))
    print('_____________________________________________________')
```

```
ID
id
7504021310    2
9809000020    2
2619920170    2
7853420110    2
7853400250    2
              ..
6635000110    1
6632300567    1
6613000750    1
6613000715    1
5200087       1
Length: 2606, dtype: int64
_____________________________________________________
PRICE
price
650000.0    25
850000.0    25
775000.0    22
600000.0    22
700000.0    22
            ..
732500.0     1
732350.0     1
731781.0     1
731688.0     1
230000.0     1
Length: 990, dtype: int64
_____________________________________________________
BEDROOMS
bedrooms
4           1351
3            832
5            313
2             78
```

```
6             33
7              4
1              2
10             1
8              1
dtype: int64
```

---

**BATHROOMS**

```
bathrooms
2.50         1226
2.75          286
2.25          237
3.50          223
3.25          182
3.00          161
2.00           93
1.75           92
3.75           34
1.50           23
4.00           22
4.50           15
4.25            9
1.00            7
1.25            2
5.25            2
4.75            1
dtype: int64
```

---

**SQFT_LOT**

```
sqft_lot
5000        44
4000        26
6000        26
3600        16
5500        15
            ..
12573        1
12591        1
12593        1
12600        1
635          1
Length: 2103, dtype: int64
```

---

**FLOORS**

```
floors
2.0       1935
1.0        447
1.5        105
3.0         82
2.5         46
dtype: int64
```

---

**WATERFRONT**

```
waterfront
0.0           2298
1.0             22
dtype: int64
```

---

**VIEW**

```
view
0.0     2152
2.0      188
3.0      129
4.0       73
1.0       62
```

```
dtype: int64
```

---

**CONDITION**

```
condition
3            2041
4             446
5             126
2               2
dtype: int64
```

---

**GRADE**

```
grade
9        2615
dtype: int64
```

---

**SQFT_ABOVE**

```
sqft_above
2550          29
2700          23
2620          22
2820          22
2500          22
              ..
2044           1
2031           1
3216           1
2029           1
5450           1
Length: 417, dtype: int64
```

---

**YR_BUILT**

```
yr_built
2014        214
2006        108
2005        106
2004        101
2007         93
           ...
1919          1
1932          1
1944          1
1902          1
1901          1
Length: 112, dtype: int64
```

---

**YR_RENOVATED**

```
yr_renovated
0.0             2065
2007.0            10
2014.0             9
2005.0             9
2000.0             7
2003.0             6
1991.0             5
2004.0             5
2001.0             5
2013.0             4
2008.0             4
2006.0             4
2002.0             4
1989.0             3
1990.0             3
1994.0             3
2015.0             3
1999.0             3
```

```
2010.0             3
2011.0             2
1995.0             2
1993.0             2
1992.0             2
2009.0             2
1983.0             2
1980.0             1
1956.0             1
1968.0             1
1970.0             1
1998.0             1
1982.0             1
1985.0             1
1996.0             1
1997.0             1
dtype: int64
```

---

**ZIPCODE**

```
zipcode
98074      142
98052      133
98006      115
98029       88
98075       84
          ...
98133        7
98188        5
98178        4
98032        2
98168        1
Length: 68, dtype: int64
```

---

**LAT**

```
lat
47.6272    8
47.5489    6
47.6285    6
47.6067    6
47.6265    6
          ..
47.6208    1
47.6219    1
47.6222    1
47.6224    1
47.1593    1
Length: 1796, dtype: int64
```

---

**LONG**

```
long
-122.290    22
-122.030    17
-122.208    16
-122.023    15
-122.146    15
            ..
-122.008     1
-122.039     1
-122.043     1
-122.075     1
-122.515     1
Length: 530, dtype: int64
```

---

**SQFT_LIVING15**

```
sqft_living15
```

```
2650             28
2590             28
3010             27
2570             25
2580             25
                 ..
2612              1
4950              1
2574              1
2547              1
720               1
Length: 382, dtype: int64
```

---

**SQFT_LOT15**

```
sqft_lot15
5000          52
4000          33
6000          32
5500          17
4500          17
              ..
11483          1
11491          1
11499          1
11503          1
651            1
Length: 1946, dtype: int64
```

---

**MONTH_YEAR**

```
month_year
201407        301
201406        276
201504        256
201405        245
201503        235
201409        234
201410        219
201408        212
201412        174
201411        167
201502        124
201501        116
201505         56
dtype: int64
```

---

**NEW_SQFT_BASEMENT**

```
new_sqft_basement
0                    1710
800                    23
900                    21
1000                   20
700                    19
                     ...
2250                    1
1080                    1
1310                    1
1520                    1
2720                    1
Length: 203, dtype: int64
```

---

In [187…

```
sns.lineplot( x = 'grade', y ='price' , legend=True,
                palette="deep",  data = df )
plt.title('Grade verses Cost')
```

```
plt.xlabel('Grade')
plt.ylabel('Cost (in millions)')
```

Out[187… Text(0, 0.5, 'Cost (in millions)')

from the above line graph it looks like rennovating homes to a certain level will have result in a higher sell price. you can also see the graph's price takes off at around the grade 8 mark

In [188…

```
grade1 = df.loc[(df["grade"] <= 8)]
grade1[['price']].describe()
```

Out[188…

| | price |
|---|---|
| **count** | 1.734700e+04 |
| **mean** | 4.373974e+05 |
| **std** | 1.958205e+05 |
| **min** | 7.800000e+04 |
| **25%** | 2.990000e+05 |
| **50%** | 4.015000e+05 |
| **75%** | 5.390000e+05 |
| **max** | 3.070000e+06 |

In [189…

```
sns.lineplot( x = 'grade', y ='price' , legend=True,
                palette="deep",  data = grade1 )
```

Out[189… <AxesSubplot:xlabel='grade', ylabel='price'>

In [190…

```
grade2 = df.loc[(df["grade"] > 8)]
grade2[['price']].describe()
```

Out[190…

| | price |
|---|---|
| **count** | 4.250000e+03 |
| **mean** | 9.602945e+05 |
| **std** | 5.565400e+05 |
| **min** | 2.300000e+05 |
| **25%** | 6.423508e+05 |
| **50%** | 8.100000e+05 |
| **75%** | 1.100000e+06 |
| **max** | 7.700000e+06 |

In [191…

```
sns.lineplot( x = 'grade', y ='price' , legend=True,
                palette="deep",  data = grade2 )
```

Out[191… <AxesSubplot:xlabel='grade', ylabel='price'>

In [ ]:

In [192…
```
# df1 = df.loc[df['yr_renovated'] > 1 ]
# figure, ax= plt.subplots(figsize = (20,10))
# sns.scatterplot( x = 'bedrooms', y ='bathrooms' , hue='grade', palette= "deep", data
#                      size="grade", sizes=(20, 1000))
```

In [193…
```
#
```

In [194…
```
df['bedrooms'].describe()
```

Out[194…
```
count    21597.000000
mean         3.373200
std          0.926299
min          1.000000
25%          3.000000
50%          3.000000
75%          4.000000
max         33.000000
Name: bedrooms, dtype: float64
```

The mean of bed rooms is between 3 and 4 room with a STD of less than one. Also a home have 33 roomes seems very unlikely

In [195…
```
df.loc[df['bedrooms'] == 33]
```

Out[195…

| | id | price | bedrooms | bathrooms | sqft_lot | floors | waterfront | view | condition | grac |
|---|---|---|---|---|---|---|---|---|---|---|
| **15856** | 2402100895 | 640000.0 | 33 | 1.75 | 6000 | 1.0 | 0.0 | 0.0 | 5 | |

For a home to have 33 rooms with 1620 sqsf of living space gives about 49 sqsf of space per room and 1 shower does not make since as a home wo that value will bedroped

In [ ]:

In [196…
```
df = df.loc[(df["bedrooms"] < 33)]
#df = df.drop(['bedrooms'] == 33, axis = 0)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 21596 entries, 0 to 21596
Data columns (total 20 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 21596 non-null  int64
 1   price              21596 non-null  float64
 2   bedrooms           21596 non-null  int64
 3   bathrooms          21596 non-null  float64
 4   sqft_lot           21596 non-null  int64
 5   floors             21596 non-null  float64
 6   waterfront         19220 non-null  float64
 7   view               21533 non-null  float64
 8   condition          21596 non-null  int64
 9   grade              21596 non-null  int64
 10  sqft_above         21596 non-null  int64
 11  yr_built           21596 non-null  int64
 12  yr_renovated       17754 non-null  float64
```

```
 13  zipcode            21596 non-null  int64
 14  lat                21596 non-null  float64
 15  long               21596 non-null  float64
 16  sqft_living15      21596 non-null  int64
 17  sqft_lot15         21596 non-null  int64
 18  month_year         21596 non-null  int32
 19  new_sqft_basement  21596 non-null  int64
dtypes: float64(8), int32(1), int64(11)
memory usage: 3.4 MB
```

we are not looking for mansion style homes how homes with 6 or more homes will be droped

In [197…

```
df = df.loc[(df["bedrooms"] < 6)]
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 21263 entries, 0 to 21596
Data columns (total 20 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 21263 non-null  int64
 1   price              21263 non-null  float64
 2   bedrooms           21263 non-null  int64
 3   bathrooms          21263 non-null  float64
 4   sqft_lot           21263 non-null  int64
 5   floors             21263 non-null  float64
 6   waterfront         18937 non-null  float64
 7   view               21202 non-null  float64
 8   condition          21263 non-null  int64
 9   grade              21263 non-null  int64
 10  sqft_above         21263 non-null  int64
 11  yr_built           21263 non-null  int64
 12  yr_renovated       17473 non-null  float64
 13  zipcode            21263 non-null  int64
 14  lat                21263 non-null  float64
 15  long               21263 non-null  float64
 16  sqft_living15      21263 non-null  int64
 17  sqft_lot15         21263 non-null  int64
 18  month_year         21263 non-null  int32
 19  new_sqft_basement  21263 non-null  int64
dtypes: float64(8), int32(1), int64(11)
memory usage: 3.3 MB
```

In [198…

```
figure, ax= plt.subplots(figsize = (100,10))
sns.lineplot( x = df['yr_built'] , y ='price' , data = df )
```

Out[198…

```
<AxesSubplot:xlabel='yr_built', ylabel='price'>
```

In [199…

```
df.loc[(df["price"] >3000000)]
```

Out[199…

| | id | price | bedrooms | bathrooms | sqft_lot | floors | waterfront | view | condition | gra |
|---|---|---|---|---|---|---|---|---|---|---|
| **300** | 3225069065 | 3080000.0 | 4 | 5.00 | 18641 | 1.0 | 1.0 | 4.0 | 3 | |
| **656** | 3760500116 | 3070000.0 | 3 | 2.50 | 55867 | 1.0 | 1.0 | 4.0 | 4 | |
| **1162** | 1247600105 | 5110000.0 | 5 | 5.25 | 45517 | 2.0 | 1.0 | 4.0 | 3 | |
| **1446** | 8907500070 | 5350000.0 | 5 | 5.00 | 23985 | 2.0 | 0.0 | 4.0 | 3 | |

| | id | price | bedrooms | bathrooms | sqft_lot | floors | waterfront | view | condition | gra |
|---|---|---|---|---|---|---|---|---|---|---|
| **2083** | 8106100105 | 3850000.0 | 4 | 4.25 | 21300 | 2.0 | 1.0 | 4.0 | 4 | |
| **2442** | 7524900003 | 3280000.0 | 2 | 1.75 | 10000 | 2.5 | 1.0 | 4.0 | 3 | |
| **2624** | 7738500731 | 4500000.0 | 5 | 5.50 | 40014 | 2.0 | 1.0 | 4.0 | 3 | |
| **2862** | 4114601570 | 3600000.0 | 3 | 3.25 | 12431 | 2.0 | 1.0 | 4.0 | 3 | |
| **3910** | 9808700762 | 7060000.0 | 5 | 4.50 | 37325 | 2.0 | 1.0 | 2.0 | 3 | |
| **4145** | 6447300265 | 4000000.0 | 4 | 5.50 | 16573 | 2.0 | 0.0 | 0.0 | 3 | |
| **4407** | 2470100110 | 5570000.0 | 5 | 5.75 | 35069 | 2.0 | 0.0 | 0.0 | 3 | |
| **5874** | 2525049148 | 3420000.0 | 5 | 5.00 | 20412 | 2.0 | 0.0 | 0.0 | 3 | |
| **7028** | 853200010 | 3800000.0 | 5 | 5.50 | 42840 | 1.0 | 0.0 | 2.0 | 4 | |
| **7306** | 4131900066 | 3100000.0 | 3 | 3.00 | 13085 | 2.0 | 1.0 | 4.0 | 4 | |
| **7982** | 9362000040 | 3400000.0 | 3 | 4.50 | 17826 | 2.0 | 1.0 | 4.0 | 3 | |
| **8085** | 1924059029 | 4670000.0 | 5 | 6.75 | 13068 | 1.0 | 1.0 | 4.0 | 3 | |
| **8629** | 3835500195 | 4490000.0 | 4 | 3.00 | 27517 | 2.0 | 0.0 | 0.0 | 3 | |
| **10435** | 1118000320 | 3400000.0 | 4 | 4.00 | 11765 | 2.0 | 0.0 | 0.0 | 3 | |
| **10454** | 333100295 | 3120000.0 | 3 | 3.50 | 56609 | 2.0 | 1.0 | 4.0 | 3 | |
| **11523** | 8964800890 | 3200000.0 | 3 | 3.25 | 13363 | 1.0 | NaN | 4.0 | 3 | |
| **12358** | 6065300370 | 4210000.0 | 5 | 6.00 | 21540 | 2.0 | 0.0 | 0.0 | 3 | |
| **13515** | 3025059124 | 3170000.0 | 5 | 3.50 | 11979 | 1.0 | 0.0 | 4.0 | 3 | |
| **13954** | 3126059023 | 3400000.0 | 4 | 3.50 | 47870 | 1.0 | 1.0 | 4.0 | 3 | |
| **14070** | 3625059043 | 3300000.0 | 5 | 4.75 | 13873 | 2.0 | 1.0 | 4.0 | 4 | |
| **15241** | 2425049063 | 3640000.0 | 4 | 3.25 | 22257 | 2.0 | 1.0 | 4.0 | 4 | |
| **15244** | 1732800780 | 3070000.0 | 5 | 3.00 | 7500 | 2.5 | 0.0 | 4.0 | 5 | |
| **15468** | 624069108 | 3200000.0 | 4 | 3.25 | 28206 | 1.0 | 1.0 | 4.0 | 4 | |
| **16288** | 7397300170 | 3710000.0 | 4 | 3.50 | 28078 | 2.0 | 0.0 | 2.0 | 4 | |
| **16955** | 3025059093 | 3100000.0 | 5 | 5.25 | 23669 | 2.0 | 0.0 | 0.0 | 3 | |
| **18185** | 3625059152 | 3300000.0 | 3 | 3.25 | 41300 | 1.0 | 1.0 | 4.0 | 4 | |
| **18288** | 6072800246 | 3300000.0 | 5 | 6.25 | 21738 | 2.0 | 0.0 | 0.0 | 3 | |
| **18467** | 4389201095 | 3650000.0 | 5 | 3.75 | 8694 | 2.0 | 0.0 | 1.0 | 3 | |
| **19002** | 2303900100 | 3800000.0 | 3 | 4.25 | 35000 | 2.0 | 0.0 | 4.0 | 3 | |
| **19133** | 3625049042 | 3640000.0 | 5 | 6.00 | 19897 | 2.0 | 0.0 | 0.0 | 3 | |
| **20279** | 251500080 | 3200000.0 | 4 | 4.00 | 18851 | 2.0 | 0.0 | 3.0 | 3 | |
| **20444** | 9808100150 | 3350000.0 | 5 | 3.75 | 15360 | 1.0 | 0.0 | 1.0 | 3 | |
| **21560** | 9253900271 | 3570000.0 | 5 | 4.50 | 10584 | 2.0 | 1.0 | 4.0 | 3 | |

In [ ]:

In [200…
```
df.columns
```

Out[200…
```
Index(['id', 'price', 'bedrooms', 'bathrooms', 'sqft_lot', 'floors',
       'waterfront', 'view', 'condition', 'grade', 'sqft_above', 'yr_built',
       'yr_renovated', 'zipcode', 'lat', 'long', 'sqft_living15', 'sqft_lot15',
       'month_year', 'new_sqft_basement'],
      dtype='object')
```

In [201…
```
df_columns = df.columns
for i in df_columns:
    print('\033[1m' + i.upper() + '\033[0m')
    print(df[[i]].value_counts(ascending=False))
    print('____________________________________________________')
```

```
ID
id
795000620     3
726049190     2
1901600090    2
9809000020    2
3598600049    2
             ..
6306810110    1
6308000020    1
6321000045    1
6324000090    1
1200019       1
Length: 21093, dtype: int64
____________________________________________________
PRICE
price
350000.0    170
450000.0    168
550000.0    157
500000.0    152
425000.0    150
           ...
529888.0      1
529219.0      1
529100.0      1
527950.0      1
78000.0       1
Length: 3594, dtype: int64
____________________________________________________
BEDROOMS
bedrooms
3           9824
4           6882
2           2760
5           1601
1            196
dtype: int64
____________________________________________________
BATHROOMS
bathrooms
2.50         5345
1.00         3844
1.75         3031
2.25         2029
2.00         1905
```

```
1.50         1437
2.75         1148
3.50          708
3.00          699
3.25          575
3.75          139
4.00          117
4.50           72
0.75           71
4.25           69
4.75           18
5.00           14
5.25            9
5.50            9
1.25            9
0.50            4
6.00            4
5.75            3
6.25            2
6.50            1
6.75            1
dtype: int64
```

---

**SQFT_LOT**

```
sqft_lot
5000        346
6000        287
4000        248
7200        215
4800        117
           ...
12286         1
12289         1
12292         1
12293         1
520           1
Length: 9673, dtype: int64
```

---

**FLOORS**

```
floors
1.0       10553
2.0        8084
1.5        1872
3.0         605
2.5         143
3.5           6
dtype: int64
```

---

**WATERFRONT**

```
waterfront
0.0           18795
1.0             142
dtype: int64
```

---

**VIEW**

```
view
0.0     19154
2.0       930
3.0       492
1.0       324
4.0       302
dtype: int64
```

---

**CONDITION**

```
condition
```

```
3            13821
4             5576
5             1671
2              167
1               28
dtype: int64
```

---

**GRADE**

```
grade
7        8851
8        5958
9        2576
6        2030
10       1106
11        381
5         239
12         84
4          27
13         10
3           1
dtype: int64
```

---

**SQFT_ABOVE**

```
sqft_above
1300          212
1010          209
1200          202
1220          191
1140          184
             ...
2497            1
2496            1
2481            1
2478            1
370             1
Length: 926, dtype: int64
```

---

**YR_BUILT**

```
yr_built
2014        555
2006        452
2005        448
2004        429
2003        417
           ...
1933         29
1901         27
1902         26
1935         24
1934         21
Length: 116, dtype: int64
```

---

**YR_RENOVATED**

```
yr_renovated
0.0             16756
2014.0             73
2007.0             30
2013.0             30
2000.0             29
                ...
1976.0              1
1944.0              1
1971.0              1
1934.0              1
1951.0              1
```

```
Length: 70, dtype: int64
```

---

**ZIPCODE**

```
zipcode
98103      589
98038      589
98115      577
98052      565
98042      544
          ...
98010      100
98102       99
98024       79
98148       57
98039       48
Length: 70, dtype: int64
```

---

**LAT**

```
lat
47.6624    17
47.5322    17
47.5491    17
47.6846    17
47.6711    16
           ..
47.4236     1
47.3228     1
47.4237     1
47.3226     1
47.1559     1
Length: 5022, dtype: int64
```

---

**LONG**

```
long
-122.290    114
-122.300    108
-122.362    100
-122.291     97
-122.357     96
           ...
-121.788      1
-121.792      1
-121.797      1
-121.803      1
-122.519      1
Length: 751, dtype: int64
```

---

**SQFT_LIVING15**

```
sqft_living15
1540             194
1440             192
1560             187
1500             178
1580             167
                ...
2304               1
2316               1
2323               1
2326               1
399                1
Length: 770, dtype: int64
```

---

**SQFT_LOT15**

```
sqft_lot15
5000          416
```

```
4000          352
6000          282
7200          208
4800          143
             ...
11377           1
11381           1
11393           1
11396           1
651             1
Length: 8608, dtype: int64
```

---

**MONTH_YEAR**

```
month_year
201504        2202
201407        2178
201406        2142
201408        1912
201410        1853
201503        1842
201409        1744
201405        1737
201412        1441
201411        1390
201502        1227
201501         956
201505         639
dtype: int64
```

---

**NEW_SQFT_BASEMENT**

```
new_sqft_basement
0                    13006
600                    214
500                    212
700                    212
800                    199
                     ...
1281                     1
1284                     1
1481                     1
1525                     1
1024                     1
Length: 296, dtype: int64
```

---

Below is a table detailing each column in the above data table represents. The "X" in the categorical column represents weather the data is catorgorical. If there is no X the data is continious/ numerical.

| Ref # | Column | Description | Categorical |
|---|---|---|---|
| 1 | id | unique identified for a house | |
| 2 | dateDate | house was sold | |
| 3 | pricePrice | is prediction target | |
| 4 | bedroomsNumber | of Bedrooms/House | |
| 5 | bathroomsNumber | of bathrooms/bedrooms | |
| 6 | sqft_livingsquare | footage of the home | |
| 7 | sqft_lotsquare | footage of the lot | |
| 8 | floorsTotal | floors (levels) in house | |

| | | | |
|---|---|---|---|
| 9 | waterfront | House which has a view to a waterfront | X |
| 10 | view | Has a view | X |
| 11 | condition | How good the condition is ( Overall ) | X |
| 12 | grade | index from 1 to 13, where 1-3 falls short of building construction and design, 7 has an average level of construction and design, and 11-13 have a high quality level of construction and design. | X |
| 13 | sqft_above | square footage of house apart from basement | |
| 14 | sqft_basement | square footage of the basement | |
| 15 | yr_built | Built Year | |
| 16 | yr_renovated | Year when house was renovated | X |
| 17 | zipcode | zip | X |
| 18 | lat | Latitude coordinate | |
| 19 | long | Longitude coordinate | |
| 20 | sqft_living15 | The square footage of interior housing living space for the nearest 15 neighbors | |
| 21 | sqft_lot15 | The square footage of the land lots of the nearest 15 neighbors | |

In [ ]:

# Investigating Missing Data

In [202…

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 21263 entries, 0 to 21596
Data columns (total 20 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id                 21263 non-null  int64  
 1   price              21263 non-null  float64
 2   bedrooms           21263 non-null  int64  
 3   bathrooms          21263 non-null  float64
 4   sqft_lot           21263 non-null  int64  
 5   floors             21263 non-null  float64
 6   waterfront         18937 non-null  float64
 7   view               21202 non-null  float64
 8   condition          21263 non-null  int64  
 9   grade              21263 non-null  int64  
 10  sqft_above         21263 non-null  int64  
 11  yr_built           21263 non-null  int64  
 12  yr_renovated       17473 non-null  float64
 13  zipcode            21263 non-null  int64  
 14  lat                21263 non-null  float64
 15  long               21263 non-null  float64
 16  sqft_living15      21263 non-null  int64  
 17  sqft_lot15         21263 non-null  int64  
 18  month_year         21263 non-null  int32  
 19  new_sqft_basement  21263 non-null  int64  
dtypes: float64(8), int32(1), int64(11)
memory usage: 3.3 MB
```

# Missing Data

In [203…
```
print(str(round(((((df['waterfront'].isna().sum()))/len(df))*100),2)) + '% Null in wate
print(str(round(((((df['view'].isna().sum()))/len(df))*100),2)) + '% Null in view')
print(str(round(((((df['yr_renovated'].isna().sum()))/len(df))*100),2)) + '% Null in yr
```

```
10.94% Null in waterfront
0.29% Null in view
17.82% Null in yr_renovated
```

In [204…
```
df[['waterfront','view', 'yr_renovated']].describe()
```

Out[204…

| | waterfront | view | yr_renovated |
|---|---|---|---|
| **count** | 18937.000000 | 21202.000000 | 17473.000000 |
| **mean** | 0.007499 | 0.229601 | 81.902707 |
| **std** | 0.086271 | 0.758155 | 395.959042 |
| **min** | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 0.000000 | 0.000000 | 0.000000 |
| **50%** | 0.000000 | 0.000000 | 0.000000 |
| **75%** | 0.000000 | 0.000000 | 0.000000 |
| **max** | 1.000000 | 4.000000 | 2015.000000 |

In [205…
```
df[['waterfront']].value_counts(ascending=False)
```

Out[205…
```
waterfront
0.0            18795
1.0              142
dtype: int64
```

In [206…
```
df[['view']].value_counts(ascending=False)
```

Out[206…
```
view
0.0     19154
2.0       930
3.0       492
1.0       324
4.0       302
dtype: int64
```

In [207…
```
df[['yr_renovated']].value_counts(ascending=False)
```

Out[207…
```
yr_renovated
0.0             16756
2014.0             73
2007.0             30
2013.0             30
2000.0             29
                ...
1976.0              1
1944.0              1
1971.0              1
1934.0              1
1951.0              1
Length: 70, dtype: int64
```

The water front column, view column, yr_renovated have missing data.

- Having a waterfront home is a uncommonand very desireable feature within the housing market, more than 75% of the data that is not waterfront based on the above .describe(). It is hard to imagin some one missing that informations which will result in a much higher competition and sells price. Based on that the assumtion will be that NULL values for waterfront are 0.0.
- Having a home with a view is more common than a water front home since a veiw can included but is not limited to the city, a park, water, art, ect. similar to the water front more than 75% of the data that is not waterfront based on the above .describe(). A home with a veiw is also a desireable feature within the housing market and creates higher competition which will result in a higher sell price. Based on that the assumtion will be that NULL values for veiw are 0.0.
- yr_renovated similar to both the waterfront column and the view column it is a very desireable feature that will drive price and creat competition. with more than 75% of the dats being 0, ie, not renovated it is assumed that the NULL values will be 0.0.

In [208…

```
convert_nan_to_0('waterfront')
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 21263 entries, 0 to 21596
Data columns (total 20 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id                 21263 non-null  int64  
 1   price              21263 non-null  float64
 2   bedrooms           21263 non-null  int64  
 3   bathrooms          21263 non-null  float64
 4   sqft_lot           21263 non-null  int64  
 5   floors             21263 non-null  float64
 6   waterfront         21263 non-null  float64
 7   view               21202 non-null  float64
 8   condition          21263 non-null  int64  
 9   grade              21263 non-null  int64  
 10  sqft_above         21263 non-null  int64  
 11  yr_built           21263 non-null  int64  
 12  yr_renovated       17473 non-null  float64
 13  zipcode            21263 non-null  int64  
 14  lat                21263 non-null  float64
 15  long               21263 non-null  float64
 16  sqft_living15      21263 non-null  int64  
 17  sqft_lot15         21263 non-null  int64  
 18  month_year         21263 non-null  int32  
 19  new_sqft_basement  21263 non-null  int64  
dtypes: float64(8), int32(1), int64(11)
memory usage: 3.3 MB
```

In [209…

```
convert_nan_to_0('view')
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 21263 entries, 0 to 21596
Data columns (total 20 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id                 21263 non-null  int64  
 1   price              21263 non-null  float64
 2   bedrooms           21263 non-null  int64  
 3   bathrooms          21263 non-null  float64
 4   sqft_lot           21263 non-null  int64  
```

```
 5   floors             21263 non-null  float64
 6   waterfront         21263 non-null  float64
 7   view               21263 non-null  float64
 8   condition          21263 non-null  int64
 9   grade              21263 non-null  int64
 10  sqft_above         21263 non-null  int64
 11  yr_built           21263 non-null  int64
 12  yr_renovated       17473 non-null  float64
 13  zipcode            21263 non-null  int64
 14  lat                21263 non-null  float64
 15  long               21263 non-null  float64
 16  sqft_living15      21263 non-null  int64
 17  sqft_lot15         21263 non-null  int64
 18  month_year         21263 non-null  int32
 19  new_sqft_basement  21263 non-null  int64
dtypes: float64(8), int32(1), int64(11)
memory usage: 3.3 MB
```

In [210…

```
convert_nan_to_0('yr_renovated')
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 21263 entries, 0 to 21596
Data columns (total 20 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 21263 non-null  int64
 1   price              21263 non-null  float64
 2   bedrooms           21263 non-null  int64
 3   bathrooms          21263 non-null  float64
 4   sqft_lot           21263 non-null  int64
 5   floors             21263 non-null  float64
 6   waterfront         21263 non-null  float64
 7   view               21263 non-null  float64
 8   condition          21263 non-null  int64
 9   grade              21263 non-null  int64
 10  sqft_above         21263 non-null  int64
 11  yr_built           21263 non-null  int64
 12  yr_renovated       21263 non-null  float64
 13  zipcode            21263 non-null  int64
 14  lat                21263 non-null  float64
 15  long               21263 non-null  float64
 16  sqft_living15      21263 non-null  int64
 17  sqft_lot15         21263 non-null  int64
 18  month_year         21263 non-null  int32
 19  new_sqft_basement  21263 non-null  int64
dtypes: float64(8), int32(1), int64(11)
memory usage: 3.3 MB
```

In [211…

```
print(str(round(((((df['waterfront'].isna().sum()))/len(df))*100),2)) + '% Null in wate
print(str(round(((((df['view'].isna().sum()))/len(df))*100),2)) + '% Null in view')
print(str(round(((((df['yr_renovated'].isna().sum()))/len(df))*100),2)) + '% Null in yr
```

```
0.0% Null in waterfront
0.0% Null in view
0.0% Null in yr_renovated
```

In [212…

```
# convert_int(df, 'yr_renovated')
```

In [213…

```
# df['yr_renovated'] = df['yr_renovated'].apply(lambda x: [y if y <= 9 else 11 for y in

# df['yr_renovated']
```

# Feature Engineering

From the data there are quite a few of homes with the flowwing trates that can be turn in to a rennovation tv show.

- With 5 bedroom or less
- Less than a million
- Grade lower than 8

In [214…

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 21263 entries, 0 to 21596
Data columns (total 20 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   id                 21263 non-null  int64  
 1   price              21263 non-null  float64
 2   bedrooms           21263 non-null  int64  
 3   bathrooms          21263 non-null  float64
 4   sqft_lot           21263 non-null  int64  
 5   floors             21263 non-null  float64
 6   waterfront         21263 non-null  float64
 7   view               21263 non-null  float64
 8   condition          21263 non-null  int64  
 9   grade              21263 non-null  int64  
 10  sqft_above         21263 non-null  int64  
 11  yr_built           21263 non-null  int64  
 12  yr_renovated       21263 non-null  float64
 13  zipcode            21263 non-null  int64  
 14  lat                21263 non-null  float64
 15  long               21263 non-null  float64
 16  sqft_living15      21263 non-null  int64  
 17  sqft_lot15         21263 non-null  int64  
 18  month_year         21263 non-null  int32  
 19  new_sqft_basement  21263 non-null  int64  
dtypes: float64(8), int32(1), int64(11)
memory usage: 3.3 MB
```

In [215…

```
df.corr()
```

Out[215…

| | id | price | bedrooms | bathrooms | sqft_lot | floors | waterfront | |
|---|---|---|---|---|---|---|---|---|
| **id** | 1.000000 | -0.017381 | 0.006358 | 0.008093 | -0.131518 | 0.020885 | -0.004709 | 0.01 |
| **price** | -0.017381 | 1.000000 | 0.312185 | 0.520621 | 0.090662 | 0.253822 | 0.265070 | 0.39 |
| **bedrooms** | 0.006358 | 0.312185 | 1.000000 | 0.516771 | 0.033895 | 0.187446 | -0.005088 | 0.07 |
| **bathrooms** | 0.008093 | 0.520621 | 0.516771 | 1.000000 | 0.086682 | 0.507936 | 0.061548 | 0.17 |
| **sqft_lot** | -0.131518 | 0.090662 | 0.033895 | 0.086682 | 1.000000 | -0.006754 | 0.021281 | 0.07 |
| **floors** | 0.020885 | 0.253822 | 0.187446 | 0.507936 | -0.006754 | 1.000000 | 0.019964 | 0.02 |
| **waterfront** | -0.004709 | 0.265070 | -0.005088 | 0.061548 | 0.021281 | 0.019964 | 1.000000 | 0.38 |
| **view** | 0.011758 | 0.391610 | 0.071583 | 0.178344 | 0.074404 | 0.024400 | 0.384743 | 1.00 |
| **condition** | -0.024033 | 0.032663 | 0.022055 | -0.129502 | -0.009339 | -0.266767 | 0.015931 | 0.04 |
| **grade** | 0.009678 | 0.670336 | 0.386365 | 0.672854 | 0.113313 | 0.458708 | 0.081855 | 0.24 |

| | id | price | bedrooms | bathrooms | sqft_lot | floors | waterfront | |
|---|---|---|---|---|---|---|---|---|
| **sqft_above** | -0.007324 | 0.598889 | 0.498238 | 0.680879 | 0.181091 | 0.522309 | 0.070722 | 0.15 |
| **yr_built** | 0.023793 | 0.062405 | 0.197471 | 0.526513 | 0.052362 | 0.498932 | -0.024246 | -0.05 |
| **yr_renovated** | -0.011600 | 0.114621 | 0.005094 | 0.039858 | 0.004464 | -0.000971 | 0.073933 | 0.09 |
| **zipcode** | -0.009853 | -0.056615 | -0.179090 | -0.213739 | -0.128692 | -0.061938 | 0.028293 | 0.08 |
| **lat** | -0.001034 | 0.313484 | -0.025082 | 0.020034 | -0.085651 | 0.048952 | -0.012408 | 0.00 |
| **long** | 0.021608 | 0.027181 | 0.163422 | 0.236493 | 0.228574 | 0.128728 | -0.037404 | -0.07 |
| **sqft_living15** | -0.001067 | 0.589025 | 0.427325 | 0.576086 | 0.142869 | 0.278950 | 0.081228 | 0.27 |
| **sqft_lot15** | -0.138964 | 0.083584 | 0.033155 | 0.087040 | 0.716199 | -0.013179 | 0.030903 | 0.07 |
| **month_year** | 0.010187 | 0.004967 | -0.009779 | -0.026832 | 0.007462 | -0.022854 | -0.005676 | 0.00 |
| **new_sqft_basement** | -0.007982 | 0.313396 | 0.281930 | 0.265911 | 0.012067 | -0.255169 | 0.084185 | 0.27 |

In [216…

```
plt.figure(figsize = (15, 15))
sns.heatmap(data = df.corr(), center=0, annot = True)
```

Out[216…

```
<AxesSubplot:>
```

# Predictive Modeling - Building Model

In [217…

```
#df = df.drop('sqft_basement', axis = 1)
#df = df.drop('month', axis = 1)
#df = df.drop('date', axis = 1)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 21263 entries, 0 to 21596
Data columns (total 20 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 21263 non-null  int64
 1   price              21263 non-null  float64
 2   bedrooms           21263 non-null  int64
 3   bathrooms          21263 non-null  float64
 4   sqft_lot           21263 non-null  int64
 5   floors             21263 non-null  float64
```

```
 6   waterfront         21263 non-null  float64
 7   view               21263 non-null  float64
 8   condition          21263 non-null  int64
 9   grade              21263 non-null  int64
 10  sqft_above         21263 non-null  int64
 11  yr_built           21263 non-null  int64
 12  yr_renovated       21263 non-null  float64
 13  zipcode            21263 non-null  int64
 14  lat                21263 non-null  float64
 15  long               21263 non-null  float64
 16  sqft_living15      21263 non-null  int64
 17  sqft_lot15         21263 non-null  int64
 18  month_year         21263 non-null  int32
 19  new_sqft_basement  21263 non-null  int64
dtypes: float64(8), int32(1), int64(11)
memory usage: 3.3 MB
```

# Baseline Model

In [218…

```
#Dependent variable
outcome = 'price'
#Independent variables
predictors = df.drop('price', axis = 1)
predictor_variables = '+'.join(predictors.columns)
formula = outcome + '~' + predictor_variables
```

In [219…

```
formula
```

Out[219…

```
'price~id+bedrooms+bathrooms+sqft_lot+floors+waterfront+view+condition+grade+sqft_above+
yr_built+yr_renovated+zipcode+lat+long+sqft_living15+sqft_lot15+month_year+new_sqft_base
ment'
```

In [220…

```
baseline_model = ols(formula = formula, data = df).fit()
baseline_model.summary()
```

Out[220…

OLS Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | price | **R-squared:** | 0.704 |
| **Model:** | OLS | **Adj. R-squared:** | 0.704 |
| **Method:** | Least Squares | **F-statistic:** | 2660. |
| **Date:** | Mon, 07 Jun 2021 | **Prob (F-statistic):** | 0.00 |
| **Time:** | 23:47:55 | **Log-Likelihood:** | -2.8895e+05 |
| **No. Observations:** | 21263 | **AIC:** | 5.779e+05 |
| **Df Residuals:** | 21243 | **BIC:** | 5.781e+05 |
| **Df Model:** | 19 | | |
| **Covariance Type:** | nonrobust | | |

| | **coef** | **std err** | **t** | **P>\|t\|** | **[0.025** | **0.975]** |
|---|---|---|---|---|---|---|
| **Intercept** | -5.972e+07 | 6.67e+06 | -8.953 | 0.000 | -7.28e+07 | -4.66e+07 |
| **id** | -1.467e-06 | 4.67e-07 | -3.145 | 0.002 | -2.38e-06 | -5.53e-07 |
| **bedrooms** | -3.825e+04 | 2069.073 | -18.486 | 0.000 | -4.23e+04 | -3.42e+04 |
| **bathrooms** | 4.111e+04 | 3224.274 | 12.750 | 0.000 | 3.48e+04 | 4.74e+04 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **sqft_lot** | 0.1423 | 0.046 | 3.079 | 0.002 | 0.052 | 0.233 |
| **floors** | 8414.8521 | 3498.399 | 2.405 | 0.016 | 1557.725 | 1.53e+04 |
| **waterfront** | 6.008e+05 | 1.77e+04 | 33.978 | 0.000 | 5.66e+05 | 6.35e+05 |
| **view** | 5.276e+04 | 2073.717 | 25.442 | 0.000 | 4.87e+04 | 5.68e+04 |
| **condition** | 2.733e+04 | 2278.019 | 11.999 | 0.000 | 2.29e+04 | 3.18e+04 |
| **grade** | 9.659e+04 | 2104.651 | 45.895 | 0.000 | 9.25e+04 | 1.01e+05 |
| **sqft_above** | 172.6441 | 3.634 | 47.514 | 0.000 | 165.522 | 179.766 |
| **yr_built** | -2552.5411 | 70.131 | -36.397 | 0.000 | -2690.002 | -2415.080 |
| **yr_renovated** | 23.7810 | 3.884 | 6.123 | 0.000 | 16.168 | 31.394 |
| **zipcode** | -575.2778 | 31.879 | -18.045 | 0.000 | -637.764 | -512.792 |
| **lat** | 6.004e+05 | 1.04e+04 | 57.867 | 0.000 | 5.8e+05 | 6.21e+05 |
| **long** | -2.065e+05 | 1.27e+04 | -16.266 | 0.000 | -2.31e+05 | -1.82e+05 |
| **sqft_living15** | 25.9311 | 3.371 | 7.693 | 0.000 | 19.325 | 32.538 |
| **sqft_lot15** | -0.3925 | 0.071 | -5.516 | 0.000 | -0.532 | -0.253 |
| **month_year** | 331.1382 | 29.934 | 11.062 | 0.000 | 272.465 | 389.812 |
| **new_sqft_basement** | 146.0804 | 4.348 | 33.601 | 0.000 | 137.559 | 154.602 |

| | | | |
|---|---|---|---|
| **Omnibus:** | 15933.177 | **Durbin-Watson:** | 1.990 |
| **Prob(Omnibus):** | 0.000 | **Jarque-Bera (JB):** | 944645.423 |
| **Skew:** | 3.062 | **Prob(JB):** | 0.00 |
| **Kurtosis:** | 35.074 | **Cond. No.** | 2.73e+13 |

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.73e+13. This might indicate that there are
strong multicollinearity or other numerical problems.

Make Observatiosn!!! about baseline model

In [221…
```
X = df.drop('price', axis =1)
y = df['price']
```

In [222…
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_stat
```

In [223…
```
len(X_train)
```

Out[223… 15947

In [224…
```
len(X_test)
```

5316

Out[224…

In [225…
```
baseline_linreg = LinearRegression()
```

In [226…
```
baseline_linreg.fit(X_train, y_train)
```

Out[226…
```
LinearRegression()
```

In [227…
```
y_pred = baseline_linreg.predict(X_test)
```

In [228…
```
residuals = y_pred - y_test
```

In [229…
```
X.columns
```

Out[229…
```
Index(['id', 'bedrooms', 'bathrooms', 'sqft_lot', 'floors', 'waterfront',
       'view', 'condition', 'grade', 'sqft_above', 'yr_built', 'yr_renovated',
       'zipcode', 'lat', 'long', 'sqft_living15', 'sqft_lot15', 'month_year',
       'new_sqft_basement'],
      dtype='object')
```

In [230…
```
baseline_linreg.coef_
```

Out[230…
```
array([-1.44243989e-06, -3.88005202e+04,  4.09480394e+04,  1.12156493e-01,
        7.83681434e+03,  6.32722371e+05,  5.01154981e+04,  2.70576753e+04,
        9.96543842e+04,  1.72189225e+02, -2.54455635e+03,  2.83918588e+01,
       -5.71048370e+02,  5.93557035e+05, -2.05885418e+05,  2.47798559e+01,
       -3.91153104e-01,  3.46622757e+02,  1.45948461e+02])
```

In [231…
```
mse_train = mean_squared_error (y_train, baseline_linreg.predict(X_train))
mse_test = mean_squared_error (y_test, y_pred)

print('Train RMSE:', np.sqrt(mse_train))
print('Train RMSE:', np.sqrt(mse_test))
```

```
Train RMSE: 193851.71413590503
Train RMSE: 190589.97613898848
```

In [232…
```
fig = sm.graphics.qqplot(baseline_model.resid, dist = stats.norm, line = '45', fit = Tr
```

In [233…
```
df.price.mean()
```

Out[233…
```
535420.5864177209
```

# Model 2 - Dealing with outliers

dropping condition because it is similar information to grade.

In [234…
```
df = df.drop('condition', axis =1)
```

In [235…
```
df.describe()
```

Out[235…

| | id | price | bedrooms | bathrooms | sqft_lot | floors | waterfro |
|---|---|---|---|---|---|---|---|
| **count** | 2.126300e+04 | 2.126300e+04 | 21263.000000 | 21263.000000 | 2.126300e+04 | 21263.000000 | 21263.0000 |
| **mean** | 4.583740e+09 | 5.354206e+05 | 3.326012 | 2.099163 | 1.505278e+04 | 1.491911 | 0.0066 |
| **std** | 2.877828e+09 | 3.547799e+05 | 0.827807 | 0.749797 | 4.146123e+04 | 0.539395 | 0.0814 |
| **min** | 1.200019e+06 | 7.800000e+04 | 1.000000 | 0.500000 | 5.200000e+02 | 1.000000 | 0.0000 |
| **25%** | 2.123700e+09 | 3.200000e+05 | 3.000000 | 1.500000 | 5.040000e+03 | 1.000000 | 0.0000 |
| **50%** | 3.904940e+09 | 4.500000e+05 | 3.000000 | 2.250000 | 7.600000e+03 | 1.500000 | 0.0000 |
| **75%** | 7.325900e+09 | 6.399665e+05 | 4.000000 | 2.500000 | 1.062300e+04 | 2.000000 | 0.0000 |
| **max** | 9.900000e+09 | 7.060000e+06 | 5.000000 | 6.750000 | 1.651359e+06 | 3.500000 | 1.0000 |

In [236…
```
df_no_outliers = df[df.bathrooms >= 1 ]
```

In [237…
```
df_no_outliers = df_no_outliers[df_no_outliers.price <= 1500000 ]
```

In [ ]:
```

```

In [238…
```
df_no_outliers.describe()
```

Out[238…

| | id | price | bedrooms | bathrooms | sqft_lot | floors | waterfro |
|---|---|---|---|---|---|---|---|
| **count** | 2.070400e+04 | 2.070400e+04 | 20704.000000 | 20704.000000 | 2.070400e+04 | 20704.000000 | 20704.0000 |
| **mean** | 4.596222e+09 | 4.998898e+05 | 3.313659 | 2.072317 | 1.470004e+04 | 1.485148 | 0.0032 |
| **std** | 2.880678e+09 | 2.469503e+05 | 0.817711 | 0.713538 | 4.007154e+04 | 0.538306 | 0.0567 |
| **min** | 1.200019e+06 | 7.800000e+04 | 1.000000 | 1.000000 | 5.200000e+02 | 1.000000 | 0.0000 |
| **25%** | 2.125059e+09 | 3.199375e+05 | 3.000000 | 1.500000 | 5.000000e+03 | 1.000000 | 0.0000 |
| **50%** | 3.905046e+09 | 4.450000e+05 | 3.000000 | 2.250000 | 7.536000e+03 | 1.000000 | 0.0000 |
| **75%** | 7.338200e+09 | 6.225000e+05 | 4.000000 | 2.500000 | 1.041450e+04 | 2.000000 | 0.0000 |
| **max** | 9.900000e+09 | 1.500000e+06 | 5.000000 | 6.000000 | 1.651359e+06 | 3.500000 | 1.0000 |

In [266…
```
#Dependent variable
outcome = 'price'
```

```
#Independent variables
predictors = df_no_outliers.drop(['price', 'id'], axis = 1)
predictor_variables = '+'.join(predictors.columns)
formula = outcome + '~' + predictor_variables
```

In [267…

```
formula
```

Out[267…

```
'price~bedrooms+bathrooms+sqft_lot+floors+waterfront+view+grade+sqft_above+yr_built+yr_renovated+zipcode+lat+long+sqft_living15+sqft_lot15+month_year+new_sqft_basement'
```

In [240…

```
model_2 = ols(formula = formula, data = df_no_outliers).fit()
model_2.summary()
```

Out[240…

OLS Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | price | **R-squared:** | 0.708 |
| **Model:** | OLS | **Adj. R-squared:** | 0.708 |
| **Method:** | Least Squares | **F-statistic:** | 2955. |
| **Date:** | Mon, 07 Jun 2021 | **Prob (F-statistic):** | 0.00 |
| **Time:** | 23:47:57 | **Log-Likelihood:** | -2.7370e+05 |
| **No. Observations:** | 20704 | **AIC:** | 5.474e+05 |
| **Df Residuals:** | 20686 | **BIC:** | 5.476e+05 |
| **Df Model:** | 17 | | |
| **Covariance Type:** | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **Intercept** | -4.585e+07 | 4.65e+06 | -9.867 | 0.000 | -5.5e+07 | -3.67e+07 |
| **bedrooms** | -1.305e+04 | 1468.742 | -8.887 | 0.000 | -1.59e+04 | -1.02e+04 |
| **bathrooms** | 3.209e+04 | 2292.533 | 13.996 | 0.000 | 2.76e+04 | 3.66e+04 |
| **sqft_lot** | 0.1792 | 0.033 | 5.407 | 0.000 | 0.114 | 0.244 |
| **floors** | 2.883e+04 | 2471.906 | 11.663 | 0.000 | 2.4e+04 | 3.37e+04 |
| **waterfront** | 1.999e+05 | 1.71e+04 | 11.661 | 0.000 | 1.66e+05 | 2.33e+05 |
| **view** | 4.011e+04 | 1516.004 | 26.458 | 0.000 | 3.71e+04 | 4.31e+04 |
| **grade** | 8.623e+04 | 1498.485 | 57.546 | 0.000 | 8.33e+04 | 8.92e+04 |
| **sqft_above** | 87.6780 | 2.715 | 32.297 | 0.000 | 82.357 | 92.999 |
| **yr_built** | -2361.8901 | 47.208 | -50.032 | 0.000 | -2454.421 | -2269.359 |
| **yr_renovated** | 12.1787 | 2.767 | 4.402 | 0.000 | 6.756 | 17.602 |
| **zipcode** | -352.5432 | 22.267 | -15.833 | 0.000 | -396.188 | -308.898 |
| **lat** | 5.579e+05 | 7217.272 | 77.305 | 0.000 | 5.44e+05 | 5.72e+05 |
| **long** | -9.595e+04 | 8906.140 | -10.773 | 0.000 | -1.13e+05 | -7.85e+04 |
| **sqft_living15** | 44.6738 | 2.476 | 18.044 | 0.000 | 39.821 | 49.527 |
| **sqft_lot15** | -0.1569 | 0.051 | -3.092 | 0.002 | -0.256 | -0.057 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **month_year** | 229.9665 | 20.953 | 10.975 | 0.000 | 188.896 | 271.037 |
| **new_sqft_basement** | 90.3913 | 3.163 | 28.574 | 0.000 | 84.191 | 96.592 |

| | | | |
|---|---|---|---|
| **Omnibus:** | 3748.694 | **Durbin-Watson:** | 1.978 |
| **Prob(Omnibus):** | 0.000 | **Jarque-Bera (JB):** | 11965.503 |
| **Skew:** | 0.926 | **Prob(JB):** | 0.00 |
| **Kurtosis:** | 6.231 | **Cond. No.** | 1.13e+09 |

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.13e+09. This might indicate that there are
strong multicollinearity or other numerical problems.

In [241…
```
mse_train = mean_squared_error (y_train, model_2.predict(X_train))
mse_test = mean_squared_error (y_test, y_pred)

print('Train RMSE:', np.sqrt(mse_train))
print('Test RMSE:', np.sqrt(mse_test))
```

```
Train RMSE: 211039.83584618906
Test RMSE: 190589.97613898848
```

In [242…
```
fig = sm.graphics.qqplot(model_2.resid, dist = stats.norm, line = '45', fit = True)
```

# Model 3 Dealing with dummies

In [243…
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 21263 entries, 0 to 21596
Data columns (total 19 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 21263 non-null  int64
 1   price              21263 non-null  float64
 2   bedrooms           21263 non-null  int64
```

```
 3   bathrooms          21263 non-null  float64
 4   sqft_lot           21263 non-null  int64
 5   floors             21263 non-null  float64
 6   waterfront         21263 non-null  float64
 7   view               21263 non-null  float64
 8   grade              21263 non-null  int64
 9   sqft_above         21263 non-null  int64
 10  yr_built           21263 non-null  int64
 11  yr_renovated       21263 non-null  float64
 12  zipcode            21263 non-null  int64
 13  lat                21263 non-null  float64
 14  long               21263 non-null  float64
 15  sqft_living15      21263 non-null  int64
 16  sqft_lot15         21263 non-null  int64
 17  month_year         21263 non-null  int32
 18  new_sqft_basement  21263 non-null  int64
dtypes: float64(8), int32(1), int64(10)
memory usage: 3.2 MB
```

In [244…
```
waterfront_dummies = pd.get_dummies(df_no_outliers['waterfront'], prefix = 'waterfront'
```

In [245…
```
view_dummies = pd.get_dummies(df_no_outliers['view'], prefix = 'view', drop_first = Tru
```

In [246…
```
grade_dummies = pd.get_dummies(df_no_outliers['grade'],prefix = 'grade', drop_first = T
```

In [247…
```
zip_dummies = pd.get_dummies(df_no_outliers['zipcode'], prefix = 'zip', drop_first = Tr
```

In [248…
```
df_cat = pd.concat([df_no_outliers, wf_dummies, view_dummies, grade_dummies, zip_dummie
```

In [249…
```
df_cat.head()
```

Out[249…

| | id | price | bedrooms | bathrooms | sqft_lot | floors | waterfront | view | grade | sqft_above |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 7129300520 | 221900.0 | 3 | 1.00 | 5650 | 1.0 | 0.0 | 0.0 | 7 | 1180 |
| **1** | 6414100192 | 538000.0 | 3 | 2.25 | 7242 | 2.0 | 0.0 | 0.0 | 7 | 2170 |
| **2** | 5631500400 | 180000.0 | 2 | 1.00 | 10000 | 1.0 | 0.0 | 0.0 | 6 | 770 |
| **3** | 2487200875 | 604000.0 | 4 | 3.00 | 5000 | 1.0 | 0.0 | 0.0 | 7 | 1050 |
| **4** | 1954400510 | 510000.0 | 3 | 2.00 | 8080 | 1.0 | 0.0 | 0.0 | 8 | 1680 |

5 rows × 101 columns

In [250…
```
df_cat.drop(columns = ['id', 'waterfront', 'view', 'zipcode', 'grade'], inplace = True)
```

In [251…
```
df_cat.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 20704 entries, 0 to 21595
Data columns (total 96 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   price              20704 non-null  float64
 1   bedrooms           20704 non-null  int64
 2   bathrooms          20704 non-null  float64
 3   sqft_lot           20704 non-null  int64
```

```
 4   floors             20704 non-null  float64
 5   sqft_above         20704 non-null  int64
 6   yr_built           20704 non-null  int64
 7   yr_renovated       20704 non-null  float64
 8   lat                20704 non-null  float64
 9   long               20704 non-null  float64
 10  sqft_living15      20704 non-null  int64
 11  sqft_lot15         20704 non-null  int64
 12  month_year         20704 non-null  int32
 13  new_sqft_basement  20704 non-null  int64
 14  wf_1.0             20704 non-null  uint8
 15  view_1.0           20704 non-null  uint8
 16  view_2.0           20704 non-null  uint8
 17  view_3.0           20704 non-null  uint8
 18  view_4.0           20704 non-null  uint8
 19  grade_5            20704 non-null  uint8
 20  grade_6            20704 non-null  uint8
 21  grade_7            20704 non-null  uint8
 22  grade_8            20704 non-null  uint8
 23  grade_9            20704 non-null  uint8
 24  grade_10           20704 non-null  uint8
 25  grade_11           20704 non-null  uint8
 26  grade_12           20704 non-null  uint8
 27  zip_98002          20704 non-null  uint8
 28  zip_98003          20704 non-null  uint8
 29  zip_98004          20704 non-null  uint8
 30  zip_98005          20704 non-null  uint8
 31  zip_98006          20704 non-null  uint8
 32  zip_98007          20704 non-null  uint8
 33  zip_98008          20704 non-null  uint8
 34  zip_98010          20704 non-null  uint8
 35  zip_98011          20704 non-null  uint8
 36  zip_98014          20704 non-null  uint8
 37  zip_98019          20704 non-null  uint8
 38  zip_98022          20704 non-null  uint8
 39  zip_98023          20704 non-null  uint8
 40  zip_98024          20704 non-null  uint8
 41  zip_98027          20704 non-null  uint8
 42  zip_98028          20704 non-null  uint8
 43  zip_98029          20704 non-null  uint8
 44  zip_98030          20704 non-null  uint8
 45  zip_98031          20704 non-null  uint8
 46  zip_98032          20704 non-null  uint8
 47  zip_98033          20704 non-null  uint8
 48  zip_98034          20704 non-null  uint8
 49  zip_98038          20704 non-null  uint8
 50  zip_98039          20704 non-null  uint8
 51  zip_98040          20704 non-null  uint8
 52  zip_98042          20704 non-null  uint8
 53  zip_98045          20704 non-null  uint8
 54  zip_98052          20704 non-null  uint8
 55  zip_98053          20704 non-null  uint8
 56  zip_98055          20704 non-null  uint8
 57  zip_98056          20704 non-null  uint8
 58  zip_98058          20704 non-null  uint8
 59  zip_98059          20704 non-null  uint8
 60  zip_98065          20704 non-null  uint8
 61  zip_98070          20704 non-null  uint8
 62  zip_98072          20704 non-null  uint8
 63  zip_98074          20704 non-null  uint8
 64  zip_98075          20704 non-null  uint8
 65  zip_98077          20704 non-null  uint8
 66  zip_98092          20704 non-null  uint8
 67  zip_98102          20704 non-null  uint8
 68  zip_98103          20704 non-null  uint8
```

```
 69  zip_98105          20704 non-null  uint8
 70  zip_98106          20704 non-null  uint8
 71  zip_98107          20704 non-null  uint8
 72  zip_98108          20704 non-null  uint8
 73  zip_98109          20704 non-null  uint8
 74  zip_98112          20704 non-null  uint8
 75  zip_98115          20704 non-null  uint8
 76  zip_98116          20704 non-null  uint8
 77  zip_98117          20704 non-null  uint8
 78  zip_98118          20704 non-null  uint8
 79  zip_98119          20704 non-null  uint8
 80  zip_98122          20704 non-null  uint8
 81  zip_98125          20704 non-null  uint8
 82  zip_98126          20704 non-null  uint8
 83  zip_98133          20704 non-null  uint8
 84  zip_98136          20704 non-null  uint8
 85  zip_98144          20704 non-null  uint8
 86  zip_98146          20704 non-null  uint8
 87  zip_98148          20704 non-null  uint8
 88  zip_98155          20704 non-null  uint8
 89  zip_98166          20704 non-null  uint8
 90  zip_98168          20704 non-null  uint8
 91  zip_98177          20704 non-null  uint8
 92  zip_98178          20704 non-null  uint8
 93  zip_98188          20704 non-null  uint8
 94  zip_98198          20704 non-null  uint8
 95  zip_98199          20704 non-null  uint8
dtypes: float64(6), int32(1), int64(7), uint8(82)
memory usage: 3.9 MB
```

In [252…

```
df_cat.columns = ['price', 'bedrooms', 'bathrooms', 'sqft_lot', 'floors', 'sqft_above',
       'yr_built', 'yr_renovated', 'lat', 'long', 'sqft_living15',
       'sqft_lot15', 'month_year', 'new_sqft_basement', 'waterfront_1', 'view_1',
       'view_2', 'view_3', 'view_4', 'grade_5', 'grade_6', 'grade_7',
       'grade_8', 'grade_9', 'grade_10', 'grade_11', 'grade_12', 'zip_98002',
       'zip_98003', 'zip_98004', 'zip_98005', 'zip_98006', 'zip_98007',
       'zip_98008', 'zip_98010', 'zip_98011', 'zip_98014', 'zip_98019',
       'zip_98022', 'zip_98023', 'zip_98024', 'zip_98027', 'zip_98028',
       'zip_98029', 'zip_98030', 'zip_98031', 'zip_98032', 'zip_98033',
       'zip_98034', 'zip_98038', 'zip_98039', 'zip_98040', 'zip_98042',
       'zip_98045', 'zip_98052', 'zip_98053', 'zip_98055', 'zip_98056',
       'zip_98058', 'zip_98059', 'zip_98065', 'zip_98070', 'zip_98072',
       'zip_98074', 'zip_98075', 'zip_98077', 'zip_98092', 'zip_98102',
       'zip_98103', 'zip_98105', 'zip_98106', 'zip_98107', 'zip_98108',
       'zip_98109', 'zip_98112', 'zip_98115', 'zip_98116', 'zip_98117',
       'zip_98118', 'zip_98119', 'zip_98122', 'zip_98125', 'zip_98126',
       'zip_98133', 'zip_98136', 'zip_98144', 'zip_98146', 'zip_98148',
       'zip_98155', 'zip_98166', 'zip_98168', 'zip_98177', 'zip_98178',
       'zip_98188', 'zip_98198', 'zip_98199']
```

In [253…

```
df_cat.columns
```

Out[253…

```
Index(['price', 'bedrooms', 'bathrooms', 'sqft_lot', 'floors', 'sqft_above',
       'yr_built', 'yr_renovated', 'lat', 'long', 'sqft_living15',
       'sqft_lot15', 'month_year', 'new_sqft_basement', 'waterfront_1',
       'view_1', 'view_2', 'view_3', 'view_4', 'grade_5', 'grade_6', 'grade_7',
       'grade_8', 'grade_9', 'grade_10', 'grade_11', 'grade_12', 'zip_98002',
       'zip_98003', 'zip_98004', 'zip_98005', 'zip_98006', 'zip_98007',
       'zip_98008', 'zip_98010', 'zip_98011', 'zip_98014', 'zip_98019',
       'zip_98022', 'zip_98023', 'zip_98024', 'zip_98027', 'zip_98028',
       'zip_98029', 'zip_98030', 'zip_98031', 'zip_98032', 'zip_98033',
       'zip_98034', 'zip_98038', 'zip_98039', 'zip_98040', 'zip_98042',
       'zip_98045', 'zip_98052', 'zip_98053', 'zip_98055', 'zip_98056',
```

```
       'zip_98058', 'zip_98059', 'zip_98065', 'zip_98070', 'zip_98072',
       'zip_98074', 'zip_98075', 'zip_98077', 'zip_98092', 'zip_98102',
       'zip_98103', 'zip_98105', 'zip_98106', 'zip_98107', 'zip_98108',
       'zip_98109', 'zip_98112', 'zip_98115', 'zip_98116', 'zip_98117',
       'zip_98118', 'zip_98119', 'zip_98122', 'zip_98125', 'zip_98126',
       'zip_98133', 'zip_98136', 'zip_98144', 'zip_98146', 'zip_98148',
       'zip_98155', 'zip_98166', 'zip_98168', 'zip_98177', 'zip_98178',
       'zip_98188', 'zip_98198', 'zip_98199'],
      dtype='object')
```

In [269…

```
#Dependent variable
outcome = 'price'
#Independent variables
predictors = df_cat.drop(['price'], axis = 1)
predictor_variables = '+'.join(predictors.columns)
formula = outcome + '~' + predictor_variables
```

In [270…

```
formula
```

Out[270…

```
'price~bedrooms+bathrooms+sqft_lot+floors+sqft_above+yr_built+yr_renovated+lat+long+sqft
_living15+sqft_lot15+month_year+new_sqft_basement+waterfront_1+view_1+view_2+view_3+view
_4+grade_5+grade_6+grade_7+grade_8+grade_9+grade_10+grade_11+grade_12+zip_98002+zip_9800
3+zip_98004+zip_98005+zip_98006+zip_98007+zip_98008+zip_98010+zip_98011+zip_98014+zip_98
019+zip_98022+zip_98023+zip_98024+zip_98027+zip_98028+zip_98029+zip_98030+zip_98031+zip_
98032+zip_98033+zip_98034+zip_98038+zip_98039+zip_98040+zip_98042+zip_98045+zip_98052+zi
p_98053+zip_98055+zip_98056+zip_98058+zip_98059+zip_98065+zip_98070+zip_98072+zip_98074+
zip_98075+zip_98077+zip_98092+zip_98102+zip_98103+zip_98105+zip_98106+zip_98107+zip_9810
8+zip_98109+zip_98112+zip_98115+zip_98116+zip_98117+zip_98118+zip_98119+zip_98122+zip_98
125+zip_98126+zip_98133+zip_98136+zip_98144+zip_98146+zip_98148+zip_98155+zip_98166+zip_
98168+zip_98177+zip_98178+zip_98188+zip_98198+zip_98199'
```

In [262…

```
df_cat.columns
```

Out[262…

```
Index(['price', 'bedrooms', 'bathrooms', 'sqft_lot', 'floors', 'sqft_above',
       'yr_built', 'yr_renovated', 'lat', 'long', 'sqft_living15',
       'sqft_lot15', 'month_year', 'new_sqft_basement', 'waterfront_1',
       'view_1', 'view_2', 'view_3', 'view_4', 'grade_5', 'grade_6', 'grade_7',
       'grade_8', 'grade_9', 'grade_10', 'grade_11', 'grade_12', 'zip_98002',
       'zip_98003', 'zip_98004', 'zip_98005', 'zip_98006', 'zip_98007',
       'zip_98008', 'zip_98010', 'zip_98011', 'zip_98014', 'zip_98019',
       'zip_98022', 'zip_98023', 'zip_98024', 'zip_98027', 'zip_98028',
       'zip_98029', 'zip_98030', 'zip_98031', 'zip_98032', 'zip_98033',
       'zip_98034', 'zip_98038', 'zip_98039', 'zip_98040', 'zip_98042',
       'zip_98045', 'zip_98052', 'zip_98053', 'zip_98055', 'zip_98056',
       'zip_98058', 'zip_98059', 'zip_98065', 'zip_98070', 'zip_98072',
       'zip_98074', 'zip_98075', 'zip_98077', 'zip_98092', 'zip_98102',
       'zip_98103', 'zip_98105', 'zip_98106', 'zip_98107', 'zip_98108',
       'zip_98109', 'zip_98112', 'zip_98115', 'zip_98116', 'zip_98117',
       'zip_98118', 'zip_98119', 'zip_98122', 'zip_98125', 'zip_98126',
       'zip_98133', 'zip_98136', 'zip_98144', 'zip_98146', 'zip_98148',
       'zip_98155', 'zip_98166', 'zip_98168', 'zip_98177', 'zip_98178',
       'zip_98188', 'zip_98198', 'zip_98199'],
      dtype='object')
```

In [263…

```
model_3 = ols(formula = formula, data = df_cat).fit()
model_3.summary()
```

Out[263…

OLS Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | price | **R-squared:** | 0.840 |
| **Model:** | OLS | **Adj. R-squared:** | 0.840 |
| **Method:** | Least Squares | **F-statistic:** | 1142. |

| | | | |
|---|---|---|---|
| **Date:** | Mon, 07 Jun 2021 | **Prob (F-statistic):** | 0.00 |
| **Time:** | 23:50:00 | **Log-Likelihood:** | -2.6746e+05 |
| **No. Observations:** | 20704 | **AIC:** | 5.351e+05 |
| **Df Residuals:** | 20608 | **BIC:** | 5.359e+05 |
| **Df Model:** | 95 | | |
| **Covariance Type:** | nonrobust | | |

| | coef | std err | t | P>\|t\| | [0.025 | 0.975] |
|---|---|---|---|---|---|---|
| **Intercept** | -6.463e+07 | 5e+06 | -12.917 | 0.000 | -7.44e+07 | -5.48e+07 |
| **bedrooms** | -1394.3760 | 1124.664 | -1.240 | 0.215 | -3598.806 | 810.054 |
| **bathrooms** | 2.364e+04 | 1720.641 | 13.736 | 0.000 | 2.03e+04 | 2.7e+04 |
| **sqft_lot** | 0.2423 | 0.025 | 9.812 | 0.000 | 0.194 | 0.291 |
| **floors** | -2.175e+04 | 2041.188 | -10.655 | 0.000 | -2.57e+04 | -1.77e+04 |
| **sqft_above** | 119.0222 | 2.117 | 56.231 | 0.000 | 114.873 | 123.171 |
| **yr_built** | -702.9246 | 39.458 | -17.815 | 0.000 | -780.265 | -625.584 |
| **yr_renovated** | 15.3771 | 2.064 | 7.450 | 0.000 | 11.331 | 19.423 |
| **lat** | 1.024e+05 | 3.98e+04 | 2.572 | 0.010 | 2.44e+04 | 1.8e+05 |
| **long** | -9.631e+04 | 2.92e+04 | -3.303 | 0.001 | -1.53e+05 | -3.92e+04 |
| **sqft_living15** | 25.0284 | 1.936 | 12.926 | 0.000 | 21.233 | 28.824 |
| **sqft_lot15** | 0.0176 | 0.039 | 0.454 | 0.650 | -0.059 | 0.094 |
| **month_year** | 244.9422 | 15.568 | 15.734 | 0.000 | 214.428 | 275.456 |
| **new_sqft_basement** | 79.4837 | 2.375 | 33.473 | 0.000 | 74.829 | 84.138 |
| **waterfront_1** | 2.34e+05 | 1.4e+04 | 16.659 | 0.000 | 2.06e+05 | 2.61e+05 |
| **view_1** | 6.737e+04 | 5953.164 | 11.316 | 0.000 | 5.57e+04 | 7.9e+04 |
| **view_2** | 6.733e+04 | 3601.982 | 18.693 | 0.000 | 6.03e+04 | 7.44e+04 |
| **view_3** | 1.212e+05 | 5083.455 | 23.842 | 0.000 | 1.11e+05 | 1.31e+05 |
| **view_4** | 1.939e+05 | 8378.098 | 23.148 | 0.000 | 1.78e+05 | 2.1e+05 |
| **grade_5** | 1.246e+04 | 2.73e+04 | 0.456 | 0.648 | -4.11e+04 | 6.6e+04 |
| **grade_6** | 2.179e+04 | 2.66e+04 | 0.818 | 0.413 | -3.04e+04 | 7.4e+04 |
| **grade_7** | 4.494e+04 | 2.66e+04 | 1.689 | 0.091 | -7213.433 | 9.71e+04 |
| **grade_8** | 8.591e+04 | 2.67e+04 | 3.219 | 0.001 | 3.36e+04 | 1.38e+05 |
| **grade_9** | 1.678e+05 | 2.69e+04 | 6.248 | 0.000 | 1.15e+05 | 2.2e+05 |
| **grade_10** | 2.386e+05 | 2.71e+04 | 8.808 | 0.000 | 1.85e+05 | 2.92e+05 |
| **grade_11** | 3.343e+05 | 2.78e+04 | 12.022 | 0.000 | 2.8e+05 | 3.89e+05 |
| **grade_12** | 4.009e+05 | 3.43e+04 | 11.680 | 0.000 | 3.34e+05 | 4.68e+05 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **zip_98002** | 2.199e+04 | 9056.241 | 2.428 | 0.015 | 4235.456 | 3.97e+04 |
| **zip_98003** | -8935.8967 | 8013.178 | -1.115 | 0.265 | -2.46e+04 | 6770.566 |
| **zip_98004** | 5.664e+05 | 1.51e+04 | 37.447 | 0.000 | 5.37e+05 | 5.96e+05 |
| **zip_98005** | 3.041e+05 | 1.58e+04 | 19.299 | 0.000 | 2.73e+05 | 3.35e+05 |
| **zip_98006** | 2.62e+05 | 1.29e+04 | 20.257 | 0.000 | 2.37e+05 | 2.87e+05 |
| **zip_98007** | 2.398e+05 | 1.63e+04 | 14.687 | 0.000 | 2.08e+05 | 2.72e+05 |
| **zip_98008** | 2.323e+05 | 1.55e+04 | 14.991 | 0.000 | 2.02e+05 | 2.63e+05 |
| **zip_98010** | 9.798e+04 | 1.38e+04 | 7.107 | 0.000 | 7.1e+04 | 1.25e+05 |
| **zip_98011** | 9.855e+04 | 2.01e+04 | 4.915 | 0.000 | 5.92e+04 | 1.38e+05 |
| **zip_98014** | 9.745e+04 | 2.23e+04 | 4.378 | 0.000 | 5.38e+04 | 1.41e+05 |
| **zip_98019** | 8.22e+04 | 2.18e+04 | 3.772 | 0.000 | 3.95e+04 | 1.25e+05 |
| **zip_98022** | 3.716e+04 | 1.2e+04 | 3.085 | 0.002 | 1.35e+04 | 6.08e+04 |
| **zip_98023** | -3.35e+04 | 7384.055 | -4.536 | 0.000 | -4.8e+04 | -1.9e+04 |
| **zip_98024** | 1.525e+05 | 1.97e+04 | 7.744 | 0.000 | 1.14e+05 | 1.91e+05 |
| **zip_98027** | 1.865e+05 | 1.33e+04 | 14.063 | 0.000 | 1.61e+05 | 2.12e+05 |
| **zip_98028** | 8.747e+04 | 1.95e+04 | 4.491 | 0.000 | 4.93e+04 | 1.26e+05 |
| **zip_98029** | 2.241e+05 | 1.52e+04 | 14.787 | 0.000 | 1.94e+05 | 2.54e+05 |
| **zip_98030** | 5715.0146 | 8838.895 | 0.647 | 0.518 | -1.16e+04 | 2.3e+04 |
| **zip_98031** | 1.696e+04 | 9205.413 | 1.843 | 0.065 | -1082.306 | 3.5e+04 |
| **zip_98032** | -7064.1501 | 1.07e+04 | -0.659 | 0.510 | -2.81e+04 | 1.39e+04 |
| **zip_98033** | 3.158e+05 | 1.68e+04 | 18.829 | 0.000 | 2.83e+05 | 3.49e+05 |
| **zip_98034** | 1.555e+05 | 1.8e+04 | 8.662 | 0.000 | 1.2e+05 | 1.91e+05 |
| **zip_98038** | 5.584e+04 | 1e+04 | 5.561 | 0.000 | 3.62e+04 | 7.55e+04 |
| **zip_98039** | 7.158e+05 | 2.91e+04 | 24.626 | 0.000 | 6.59e+05 | 7.73e+05 |
| **zip_98040** | 4.404e+05 | 1.33e+04 | 33.089 | 0.000 | 4.14e+05 | 4.66e+05 |
| **zip_98042** | 2.199e+04 | 8494.315 | 2.589 | 0.010 | 5341.157 | 3.86e+04 |
| **zip_98045** | 1.347e+05 | 1.87e+04 | 7.202 | 0.000 | 9.8e+04 | 1.71e+05 |
| **zip_98052** | 2.281e+05 | 1.71e+04 | 13.340 | 0.000 | 1.95e+05 | 2.62e+05 |
| **zip_98053** | 2.168e+05 | 1.84e+04 | 11.809 | 0.000 | 1.81e+05 | 2.53e+05 |
| **zip_98055** | 2.908e+04 | 1.03e+04 | 2.828 | 0.005 | 8924.043 | 4.92e+04 |
| **zip_98056** | 8.986e+04 | 1.12e+04 | 8.036 | 0.000 | 6.79e+04 | 1.12e+05 |
| **zip_98058** | 3.489e+04 | 9733.170 | 3.585 | 0.000 | 1.58e+04 | 5.4e+04 |
| **zip_98059** | 9.876e+04 | 1.1e+04 | 8.985 | 0.000 | 7.72e+04 | 1.2e+05 |
| **zip_98065** | 1.261e+05 | 1.72e+04 | 7.338 | 0.000 | 9.24e+04 | 1.6e+05 |
| **zip_98070** | 6.751e+04 | 1.31e+04 | 5.145 | 0.000 | 4.18e+04 | 9.32e+04 |

| | | | | | | |
|---|---|---|---|---|---|---|
| **zip_98072** | 1.407e+05 | 2e+04 | 7.050 | 0.000 | 1.02e+05 | 1.8e+05 |
| **zip_98074** | 1.849e+05 | 1.62e+04 | 11.400 | 0.000 | 1.53e+05 | 2.17e+05 |
| **zip_98075** | 2.065e+05 | 1.56e+04 | 13.205 | 0.000 | 1.76e+05 | 2.37e+05 |
| **zip_98077** | 1.319e+05 | 2.08e+04 | 6.337 | 0.000 | 9.11e+04 | 1.73e+05 |
| **zip_98092** | -1.128e+04 | 8039.677 | -1.403 | 0.161 | -2.7e+04 | 4479.262 |
| **zip_98102** | 3.97e+05 | 1.74e+04 | 22.765 | 0.000 | 3.63e+05 | 4.31e+05 |
| **zip_98103** | 2.859e+05 | 1.62e+04 | 17.688 | 0.000 | 2.54e+05 | 3.18e+05 |
| **zip_98105** | 3.855e+05 | 1.68e+04 | 23.013 | 0.000 | 3.53e+05 | 4.18e+05 |
| **zip_98106** | 8.467e+04 | 1.2e+04 | 7.078 | 0.000 | 6.12e+04 | 1.08e+05 |
| **zip_98107** | 2.769e+05 | 1.67e+04 | 16.614 | 0.000 | 2.44e+05 | 3.1e+05 |
| **zip_98108** | 8.583e+04 | 1.32e+04 | 6.495 | 0.000 | 5.99e+04 | 1.12e+05 |
| **zip_98109** | 4.081e+05 | 1.73e+04 | 23.529 | 0.000 | 3.74e+05 | 4.42e+05 |
| **zip_98112** | 4.566e+05 | 1.55e+04 | 29.491 | 0.000 | 4.26e+05 | 4.87e+05 |
| **zip_98115** | 2.892e+05 | 1.64e+04 | 17.601 | 0.000 | 2.57e+05 | 3.21e+05 |
| **zip_98116** | 2.56e+05 | 1.33e+04 | 19.182 | 0.000 | 2.3e+05 | 2.82e+05 |
| **zip_98117** | 2.651e+05 | 1.66e+04 | 15.930 | 0.000 | 2.32e+05 | 2.98e+05 |
| **zip_98118** | 1.332e+05 | 1.17e+04 | 11.406 | 0.000 | 1.1e+05 | 1.56e+05 |
| **zip_98119** | 3.916e+05 | 1.63e+04 | 23.973 | 0.000 | 3.6e+05 | 4.24e+05 |
| **zip_98122** | 2.847e+05 | 1.45e+04 | 19.660 | 0.000 | 2.56e+05 | 3.13e+05 |
| **zip_98125** | 1.535e+05 | 1.78e+04 | 8.644 | 0.000 | 1.19e+05 | 1.88e+05 |
| **zip_98126** | 1.532e+05 | 1.22e+04 | 12.505 | 0.000 | 1.29e+05 | 1.77e+05 |
| **zip_98133** | 1.079e+05 | 1.83e+04 | 5.889 | 0.000 | 7.2e+04 | 1.44e+05 |
| **zip_98136** | 2.13e+05 | 1.26e+04 | 16.964 | 0.000 | 1.88e+05 | 2.38e+05 |
| **zip_98144** | 2.229e+05 | 1.35e+04 | 16.547 | 0.000 | 1.96e+05 | 2.49e+05 |
| **zip_98146** | 7.541e+04 | 1.12e+04 | 6.707 | 0.000 | 5.34e+04 | 9.74e+04 |
| **zip_98148** | 3.368e+04 | 1.51e+04 | 2.237 | 0.025 | 4163.210 | 6.32e+04 |
| **zip_98155** | 9.169e+04 | 1.91e+04 | 4.811 | 0.000 | 5.43e+04 | 1.29e+05 |
| **zip_98166** | 7.032e+04 | 1.03e+04 | 6.829 | 0.000 | 5.01e+04 | 9.05e+04 |
| **zip_98168** | 1.871e+04 | 1.08e+04 | 1.728 | 0.084 | -2512.115 | 3.99e+04 |
| **zip_98177** | 1.676e+05 | 1.92e+04 | 8.749 | 0.000 | 1.3e+05 | 2.05e+05 |
| **zip_98178** | 2.26e+04 | 1.12e+04 | 2.019 | 0.044 | 659.547 | 4.45e+04 |
| **zip_98188** | 1.365e+04 | 1.15e+04 | 1.187 | 0.235 | -8887.001 | 3.62e+04 |
| **zip_98198** | 3809.0332 | 8644.722 | 0.441 | 0.659 | -1.31e+04 | 2.08e+04 |
| **zip_98199** | 3.29e+05 | 1.58e+04 | 20.787 | 0.000 | 2.98e+05 | 3.6e+05 |

**Omnibus:** 3381.785 **Durbin-Watson:** 1.982

| | | | |
|---|---|---|---|
| **Prob(Omnibus):** | 0.000 | **Jarque-Bera (JB):** | 20193.151 |
| **Skew:** | 0.652 | **Prob(JB):** | 0.00 |
| **Kurtosis:** | 7.659 | **Cond. No.** | 1.47e+09 |

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.47e+09. This might indicate that there are
strong multicollinearity or other numerical problems.

In [264…

```
ff = df.loc[df['new_sqft_basement'] > 1]
f = ff['new_sqft_basement'].mean()
print('Mean of basement AREA is ' + str(f))

print('cost add for basement ' + str(int(f) * 79.4837))
```

```
Mean of basement AREA is 730.2660772677727
cost add for basement 58023.101
```

In [265…

```
mse_train = mean_squared_error (y_train, model_3.predict(X_train))
mse_test = mean_squared_error (y_test, y_pred)

print('Train RMSE:', np.sqrt(mse_train))
print('Train RMSE:', np.sqrt(mse_test))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\anaconda3\lib\site-packages\patsy\compat.py in call_and_wrap_exc(msg, origin, f, *arg
s, **kwargs)
     35     try:
---> 36         return f(*args, **kwargs)
     37     except Exception as e:

~\anaconda3\lib\site-packages\patsy\eval.py in eval(self, expr, source_name, inner_names
pace)
    164         code = compile(expr, source_name, "eval", self.flags, False)
--> 165         return eval(code, {}, VarLookupDict([inner_namespace]
    166                                             + self._namespaces))

<string> in <module>

NameError: name 'waterfront_1' is not defined

The above exception was the direct cause of the following exception:

PatsyError                                Traceback (most recent call last)
~\anaconda3\lib\site-packages\statsmodels\base\model.py in predict(self, exog, transfor
m, *args, **kwargs)
   1076             try:
-> 1077                 exog = dmatrix(design_info, exog, return_type="dataframe")
   1078             except Exception as exc:

~\anaconda3\lib\site-packages\patsy\highlevel.py in dmatrix(formula_like, data, eval_en
v, NA_action, return_type)
    289     eval_env = EvalEnvironment.capture(eval_env, reference=1)
--> 290     (lhs, rhs) = _do_highlevel_design(formula_like, data, eval_env,
    291                                       NA_action, return_type)
```

```
~\anaconda3\lib\site-packages\patsy\highlevel.py in _do_highlevel_design(formula_like, d
ata, eval_env, NA_action, return_type)
    166     if design_infos is not None:
--> 167         return build_design_matrices(design_infos, data,
    168                                     NA_action=NA_action,

~\anaconda3\lib\site-packages\patsy\build.py in build_design_matrices(design_infos, dat
a, NA_action, return_type, dtype)
    887             if factor_info not in factor_info_to_values:
--> 888                 value, is_NA = _eval_factor(factor_info, data, NA_action)
    889                 factor_info_to_isNAs[factor_info] = is_NA

~\anaconda3\lib\site-packages\patsy\build.py in _eval_factor(factor_info, data, NA_actio
n)
     62     factor = factor_info.factor
---> 63     result = factor.eval(factor_info.state, data)
     64     # Returns either a 2d ndarray, or a DataFrame, plus is_NA mask

~\anaconda3\lib\site-packages\patsy\eval.py in eval(self, memorize_state, data)
    563     def eval(self, memorize_state, data):
--> 564         return self._eval(memorize_state["eval_code"],
    565                           memorize_state,

~\anaconda3\lib\site-packages\patsy\eval.py in _eval(self, code, memorize_state, data)
    546         inner_namespace = VarLookupDict([data, memorize_state["transforms"]])
--> 547         return call_and_wrap_exc("Error evaluating factor",
    548                                  self,

~\anaconda3\lib\site-packages\patsy\compat.py in call_and_wrap_exc(msg, origin, f, *arg
s, **kwargs)
     42             # Use 'exec' to hide this syntax from the Python 2 parser:
---> 43             exec("raise new_exc from e")
     44         else:

~\anaconda3\lib\site-packages\patsy\compat.py in <module>

PatsyError: Error evaluating factor: NameError: name 'waterfront_1' is not defined
    price~bedrooms+bathrooms+sqft_lot+floors+sqft_above+yr_built+yr_renovated+lat+long+s
qft_living15+sqft_lot15+month_year+new_sqft_basement+waterfront_1+view_1+view_2+view_3+v
iew_4+grade_5+grade_6+grade_7+grade_8+grade_9+grade_10+grade_11+grade_12+zip_98002+zip_9
8003+zip_98004+zip_98005+zip_98006+zip_98007+zip_98008+zip_98010+zip_98011+zip_98014+zip
_98019+zip_98022+zip_98023+zip_98024+zip_98027+zip_98028+zip_98029+zip_98030+zip_98031+z
ip_98032+zip_98033+zip_98034+zip_98038+zip_98039+zip_98040+zip_98042+zip_98045+zip_98052
+zip_98053+zip_98055+zip_98056+zip_98058+zip_98059+zip_98065+zip_98070+zip_98072+zip_980
74+zip_98075+zip_98077+zip_98092+zip_98102+zip_98103+zip_98105+zip_98106+zip_98107+zip_9
8108+zip_98109+zip_98112+zip_98115+zip_98116+zip_98117+zip_98118+zip_98119+zip_98122+zip
_98125+zip_98126+zip_98133+zip_98136+zip_98144+zip_98146+zip_98148+zip_98155+zip_98166+z
ip_98168+zip_98177+zip_98178+zip_98188+zip_98198+zip_98199

^^^^^^^^^^^^

During handling of the above exception, another exception occurred:

PatsyError                                Traceback (most recent call last)
<ipython-input-265-4605fd16848a> in <module>
----> 1 mse_train = mean_squared_error (y_train, model_3.predict(X_train))
      2 mse_test = mean_squared_error (y_test, y_pred)
      3
      4 print('Train RMSE:', np.sqrt(mse_train))
      5 print('Train RMSE:', np.sqrt(mse_test))

~\anaconda3\lib\site-packages\statsmodels\base\model.py in predict(self, exog, transfor
m, *args, **kwargs)
   1082                         '\n\nThe original error message returned by patsy is:\n'
   1083                         '{0}'.format(str(str(exc))))
```

```
-> 1084                 raise exc.__class__(msg)
   1085             if orig_exog_len > len(exog) and not is_dict:
   1086                 if exog_index is None:

PatsyError: predict requires that you use a DataFrame when predicting from a model
that was created using the formula api.

The original error message returned by patsy is:
Error evaluating factor: NameError: name 'waterfront_1' is not defined
    price~bedrooms+bathrooms+sqft_lot+floors+sqft_above+yr_built+yr_renovated+lat+long+s
qft_living15+sqft_lot15+month_year+new_sqft_basement+waterfront_1+view_1+view_2+view_3+v
iew_4+grade_5+grade_6+grade_7+grade_8+grade_9+grade_10+grade_11+grade_12+zip_98002+zip_9
8003+zip_98004+zip_98005+zip_98006+zip_98007+zip_98008+zip_98010+zip_98011+zip_98014+zip
_98019+zip_98022+zip_98023+zip_98024+zip_98027+zip_98028+zip_98029+zip_98030+zip_98031+z
ip_98032+zip_98033+zip_98034+zip_98038+zip_98039+zip_98040+zip_98042+zip_98045+zip_98052
+zip_98053+zip_98055+zip_98056+zip_98058+zip_98059+zip_98065+zip_98070+zip_98072+zip_980
74+zip_98075+zip_98077+zip_98092+zip_98102+zip_98103+zip_98105+zip_98106+zip_98107+zip_9
8108+zip_98109+zip_98112+zip_98115+zip_98116+zip_98117+zip_98118+zip_98119+zip_98122+zip
_98125+zip_98126+zip_98133+zip_98136+zip_98144+zip_98146+zip_98148+zip_98155+zip_98166+z
ip_98168+zip_98177+zip_98178+zip_98188+zip_98198+zip_98199

^^^^^^^^^^^^
```

In [144…

```
fig = sm.graphics.qqplot(test.resid, dist = stats.norm, line = '45', fit = True)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-144-68cfacae8e6e> in <module>
----> 1 fig = sm.graphics.qqplot(test.resid, dist = stats.norm, line = '45', fit = True
)

AttributeError: 'LinearRegression' object has no attribute 'resid'
```

# Model 5 - Log transformation

In [271…

```
df_columns = df.columns
df_columns = ['price', 'bedrooms', 'bathrooms', 'sqft_lot', 'floors',
       'waterfront', 'view', 'grade', 'sqft_above', 'yr_built', 'yr_renovated',
       'zipcode', 'lat', 'long', 'sqft_living15', 'sqft_lot15', 'month_year',
       'new_sqft_basement']
```

In [272…

```
for i in df_columns:
    figure, ax= plt.subplots(figsize = (10,10))
    sns.distplot(df, x = df[i])
    plt.title(i)
```

```
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

```
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

bedrooms
Density
0
1
2
3
4
5
6
1
2
3
4
5

bathrooms
Density
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
1
2
3
4
5
6
7

sqft_lot
1e−5
5
4
3
2
1
0
Density
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
1e6

floors
Density
0
1
2
3
4
5
6
7
1.0
1.5
2.0
2.5
3.0
3.5

waterfront
Density
50
40
30
20
10
0
0.0
0.2
0.4
0.6
0.8
1.0

view
Density
0
2
4
6
8
10
0
1
2
3
4

grade
Density
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
2
4
6
8
10
12
14

sqft_above
Density
0.0007
0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000
0
2000
4000
6000
8000

yr_built
Density
0.020
0.015
0.010
0.005
0.000
1900
1920
1940
1960
1980
2000
2020

yr_renovated
Density
0.025
0.020
0.015
0.010
0.005
0.000
0
500
1000
1500
2000

zipcode
Density
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
98000
98050
98100
98150
98200

lat
Density
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
47.1
47.2
47.3
47.4
47.5
47.6
47.7
47.8

long
Density
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−122.6
−122.4
−122.2
−122.0
−121.8
−121.6
−121.4
−121.2

sqft_living15
Density
0.0008
0.0007
0.0006
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000
0
1000
2000
3000
4000
5000
6000

sqft_lot15
1e−5
Density
0
1
2
3
4
5
6
7
0
200000
400000
600000
800000

month_year
Density
0.08
0.06
0.04
0.02
0.00
201400
201420
201440
201460
201480
201500
201520

from the above graphs the following columns will have log transformations: sqft_lot, sqft_above, sqft_living15, sqft_lot15

In [273…

```
df_no_outliers['sqft_lot_log'] = np.log(df_no_outliers['sqft_lot'])
df_no_outliers['sqft_above_log'] = np.log(df_no_outliers['sqft_above'])
df_no_outliers['sqft_living15_log'] = np.log(df_no_outliers['sqft_living15'])
df_no_outliers['sqft_lot15_log'] = np.log(df_no_outliers['sqft_lot15'])
```

In [274…

```
log_columns = ['sqft_lot_log', 'sqft_above_log', 'sqft_living15_log', 'sqft_lot15_log']
for i in log_columns:
    figure, ax= plt.subplots(figsize = (10,10))
    sns.distplot(df, x = df_no_outliers[i])
    plt.title(i)
```

```
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
```

```
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
C:\Users\laure\anaconda3\lib\site-packages\seaborn\distributions.py:2551: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

sqft_above_log
Density
1.0
0.8
0.6
0.4
0.2
0.0
6.0
6.5
7.0
7.5
8.0
8.5
9.0

sqft_living15_log
Density
0.0
0.2
0.4
0.6
0.8
1.0
1.2
6.0
6.5
7.0
7.5
8.0
8.5

In [ ]:

In [276…
```
#Dependent variable
outcome = 'price'
#Independent variables
predictors = df_no_outliers.drop(['price', 'id'], axis = 1)
predictor_variables = '+'.join(predictors.columns)
formula = outcome + '~' + predictor_variables
```

In [277…
```
formula
```

Out[277…
```
'price~bedrooms+bathrooms+sqft_lot+floors+waterfront+view+grade+sqft_above+yr_built+yr_renovated+zipcode+lat+long+sqft_living15+sqft_lot15+month_year+new_sqft_basement+sqft_lot_log+sqft_above_log+sqft_living15_log+sqft_lot15_log'
```

In [278…
```
model_5 = ols(formula = formula, data = df_no_outliers).fit()
model_5.summary()
```

Out[278…

OLS Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | price | **R-squared:** | 0.713 |
| **Model:** | OLS | **Adj. R-squared:** | 0.713 |

| | | | |
|---|---|---|---|
| **Method:** | Least Squares | **F-statistic:** | 2444. |
| **Date:** | Mon, 07 Jun 2021 | **Prob (F-statistic):** | 0.00 |
| **Time:** | 23:57:46 | **Log-Likelihood:** | -2.7354e+05 |
| **No. Observations:** | 20704 | **AIC:** | 5.471e+05 |
| **Df Residuals:** | 20682 | **BIC:** | 5.473e+05 |
| **Df Model:** | 21 | | |
| **Covariance Type:** | nonrobust | | |

| | **coef** | **std err** | **t** | **P>\|t\|** | **[0.025** | **0.975]** |
|---|---|---|---|---|---|---|
| **Intercept** | -3.71e+07 | 4.65e+06 | -7.984 | 0.000 | -4.62e+07 | -2.8e+07 |
| **bedrooms** | -1.002e+04 | 1501.000 | -6.673 | 0.000 | -1.3e+04 | -7073.604 |
| **bathrooms** | 2.966e+04 | 2301.600 | 12.885 | 0.000 | 2.51e+04 | 3.42e+04 |
| **sqft_lot** | 0.2937 | 0.039 | 7.443 | 0.000 | 0.216 | 0.371 |
| **floors** | 8222.3394 | 2807.416 | 2.929 | 0.003 | 2719.583 | 1.37e+04 |
| **waterfront** | 2.256e+05 | 1.71e+04 | 13.215 | 0.000 | 1.92e+05 | 2.59e+05 |
| **view** | 4.014e+04 | 1509.004 | 26.602 | 0.000 | 3.72e+04 | 4.31e+04 |
| **grade** | 8.521e+04 | 1493.737 | 57.046 | 0.000 | 8.23e+04 | 8.81e+04 |
| **sqft_above** | 99.3024 | 6.129 | 16.203 | 0.000 | 87.290 | 111.315 |
| **yr_built** | -2411.7094 | 47.123 | -51.179 | 0.000 | -2504.075 | -2319.344 |
| **yr_renovated** | 13.2034 | 2.749 | 4.804 | 0.000 | 7.816 | 18.591 |
| **zipcode** | -384.4474 | 22.271 | -17.262 | 0.000 | -428.100 | -340.795 |
| **lat** | 5.471e+05 | 7200.628 | 75.974 | 0.000 | 5.33e+05 | 5.61e+05 |
| **long** | -7.01e+04 | 8994.424 | -7.794 | 0.000 | -8.77e+04 | -5.25e+04 |
| **sqft_living15** | 87.6733 | 8.410 | 10.425 | 0.000 | 71.189 | 104.158 |
| **sqft_lot15** | 0.3625 | 0.065 | 5.546 | 0.000 | 0.234 | 0.491 |
| **month_year** | 224.9249 | 20.799 | 10.814 | 0.000 | 184.157 | 265.693 |
| **new_sqft_basement** | 90.5068 | 3.147 | 28.759 | 0.000 | 84.338 | 96.675 |
| **sqft_lot_log** | -1.021e+04 | 3274.865 | -3.118 | 0.002 | -1.66e+04 | -3791.837 |
| **sqft_above_log** | -3633.0193 | 1.12e+04 | -0.325 | 0.745 | -2.55e+04 | 1.83e+04 |
| **sqft_living15_log** | -7.462e+04 | 1.63e+04 | -4.566 | 0.000 | -1.07e+05 | -4.26e+04 |
| **sqft_lot15_log** | -2.571e+04 | 3663.653 | -7.017 | 0.000 | -3.29e+04 | -1.85e+04 |

| | | | |
|---|---|---|---|
| **Omnibus:** | 4006.615 | **Durbin-Watson:** | 1.980 |
| **Prob(Omnibus):** | 0.000 | **Jarque-Bera (JB):** | 13551.892 |
| **Skew:** | 0.969 | **Prob(JB):** | 0.00 |
| **Kurtosis:** | 6.457 | **Cond. No.** | 1.14e+09 |

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.14e+09. This might indicate that there are
strong multicollinearity or other numerical problems.

In [280…

```
mse_train = mean_squared_error (y_train, model_5.predict(X_train))
mse_test = mean_squared_error (y_test, y_pred)

print('Train RMSE:', np.sqrt(mse_train))
print('Test RMSE:', np.sqrt(mse_test))
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\anaconda3\lib\site-packages\patsy\compat.py in call_and_wrap_exc(msg, origin, f, *arg
s, **kwargs)
     35     try:
---> 36         return f(*args, **kwargs)
     37     except Exception as e:

~\anaconda3\lib\site-packages\patsy\eval.py in eval(self, expr, source_name, inner_names
pace)
    164         code = compile(expr, source_name, "eval", self.flags, False)
--> 165         return eval(code, {}, VarLookupDict([inner_namespace]
    166                                             + self._namespaces))

<string> in <module>

NameError: name 'sqft_lot_log' is not defined

The above exception was the direct cause of the following exception:

PatsyError                                Traceback (most recent call last)
~\anaconda3\lib\site-packages\statsmodels\base\model.py in predict(self, exog, transfor
m, *args, **kwargs)
   1076             try:
-> 1077                 exog = dmatrix(design_info, exog, return_type="dataframe")
   1078             except Exception as exc:

~\anaconda3\lib\site-packages\patsy\highlevel.py in dmatrix(formula_like, data, eval_en
v, NA_action, return_type)
    289     eval_env = EvalEnvironment.capture(eval_env, reference=1)
--> 290     (lhs, rhs) = _do_highlevel_design(formula_like, data, eval_env,
    291                                       NA_action, return_type)

~\anaconda3\lib\site-packages\patsy\highlevel.py in _do_highlevel_design(formula_like, d
ata, eval_env, NA_action, return_type)
    166     if design_infos is not None:
--> 167         return build_design_matrices(design_infos, data,
    168                                      NA_action=NA_action,

~\anaconda3\lib\site-packages\patsy\build.py in build_design_matrices(design_infos, dat
a, NA_action, return_type, dtype)
    887             if factor_info not in factor_info_to_values:
--> 888                 value, is_NA = _eval_factor(factor_info, data, NA_action)
    889                 factor_info_to_isNAs[factor_info] = is_NA

~\anaconda3\lib\site-packages\patsy\build.py in _eval_factor(factor_info, data, NA_actio
n)
     62     factor = factor_info.factor
---> 63     result = factor.eval(factor_info.state, data)
```

```
     64     # Returns either a 2d ndarray, or a DataFrame, plus is_NA mask

~\anaconda3\lib\site-packages\patsy\eval.py in eval(self, memorize_state, data)
    563     def eval(self, memorize_state, data):
--> 564         return self._eval(memorize_state["eval_code"],
    565                           memorize_state,

~\anaconda3\lib\site-packages\patsy\eval.py in _eval(self, code, memorize_state, data)
    546         inner_namespace = VarLookupDict([data, memorize_state["transforms"]])
--> 547         return call_and_wrap_exc("Error evaluating factor",
    548                                  self,

~\anaconda3\lib\site-packages\patsy\compat.py in call_and_wrap_exc(msg, origin, f, *arg
s, **kwargs)
     42             # Use 'exec' to hide this syntax from the Python 2 parser:
---> 43             exec("raise new_exc from e")
     44         else:

~\anaconda3\lib\site-packages\patsy\compat.py in <module>

PatsyError: Error evaluating factor: NameError: name 'sqft_lot_log' is not defined
    price~bedrooms+bathrooms+sqft_lot+floors+waterfront+view+grade+sqft_above+yr_built+y
r_renovated+zipcode+lat+long+sqft_living15+sqft_lot15+month_year+new_sqft_basement+sqft_
lot_log+sqft_above_log+sqft_living15_log+sqft_lot15_log

^^^^^^^^^^^^

During handling of the above exception, another exception occurred:

PatsyError                                Traceback (most recent call last)
<ipython-input-280-98e55314d7e6> in <module>
----> 1 mse_train = mean_squared_error (y_train, model_5.predict(X_train))
      2 mse_test = mean_squared_error (y_test, y_pred)
      3
      4 print('Train RMSE:', np.sqrt(mse_train))
      5 print('Test RMSE:', np.sqrt(mse_test))

~\anaconda3\lib\site-packages\statsmodels\base\model.py in predict(self, exog, transfor
m, *args, **kwargs)
   1082                        '\n\nThe original error message returned by patsy is:\n'
   1083                        '{0}'.format(str(str(exc))))
-> 1084                 raise exc.__class__(msg)
   1085             if orig_exog_len > len(exog) and not is_dict:
   1086                 if exog_index is None:

PatsyError: predict requires that you use a DataFrame when predicting from a model
that was created using the formula api.

The original error message returned by patsy is:
Error evaluating factor: NameError: name 'sqft_lot_log' is not defined
    price~bedrooms+bathrooms+sqft_lot+floors+waterfront+view+grade+sqft_above+yr_built+y
r_renovated+zipcode+lat+long+sqft_living15+sqft_lot15+month_year+new_sqft_basement+sqft_
lot_log+sqft_above_log+sqft_living15_log+sqft_lot15_log

^^^^^^^^^^^^
```

In [242…

```
fig = sm.graphics.qqplot(model_2.resid, dist = stats.norm, line = '45', fit = True)
```

Sample Quantiles
Theoretical Quantiles
6
4
2
0
−2
−4
−4
−2
0
2
4
6