EDA Project
house sells data

Diving into data :)

Content

- Quick overview
  - Pairplot
  - Histplot
  - Corrplot
- Questioning
- Linear Regression

King County Data Set

* **id** - unique identified for a house
* **dateDate** - house was sold
* **pricePrice** - is prediction target
* **bedroomsNumber** - # of bedrooms
* **bathroomsNumber** - # of bathrooms
* **sqft_livingsquare** - footage of the home
* **sqft_lotsquare** - footage of the lot
* **floorsTotal** - floors (levels) in house
* **waterfront** - House which has a view to a waterfront
* **view** - Has been viewed
* **condition** - How good the condition is (Overall)

* **grade** - overall grade given to the housing unit, based on King County grading system

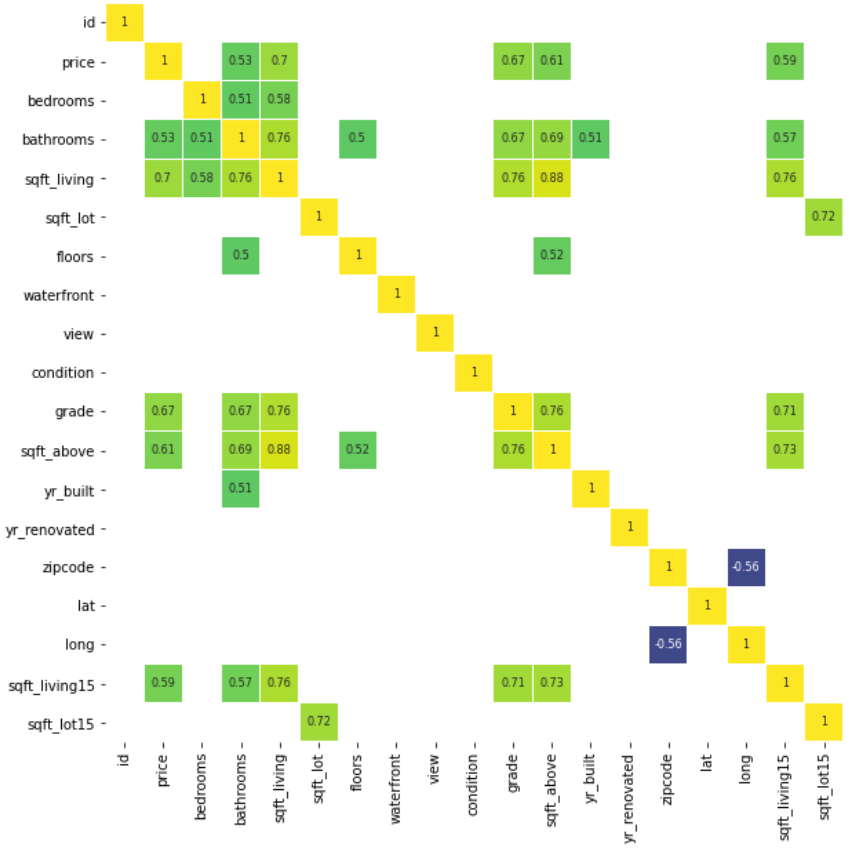
King County Data Set

* **sqft_above** - square footage of house apart from basement
* **sqft_basement** - square footage of the basement
* **yr_built** - Built Year
* **yr_renovated** - Year when house was renovated
* **zipcode** - zip
* **lat** - Latitude coordinate
* **long** - Longitude coordinate

* **sqft_living15** - The square footage of interior housing living space for the nearest 15 neighbors

* **sqft_lot15** - The square footage of the land lots of the nearest 15 neighbors

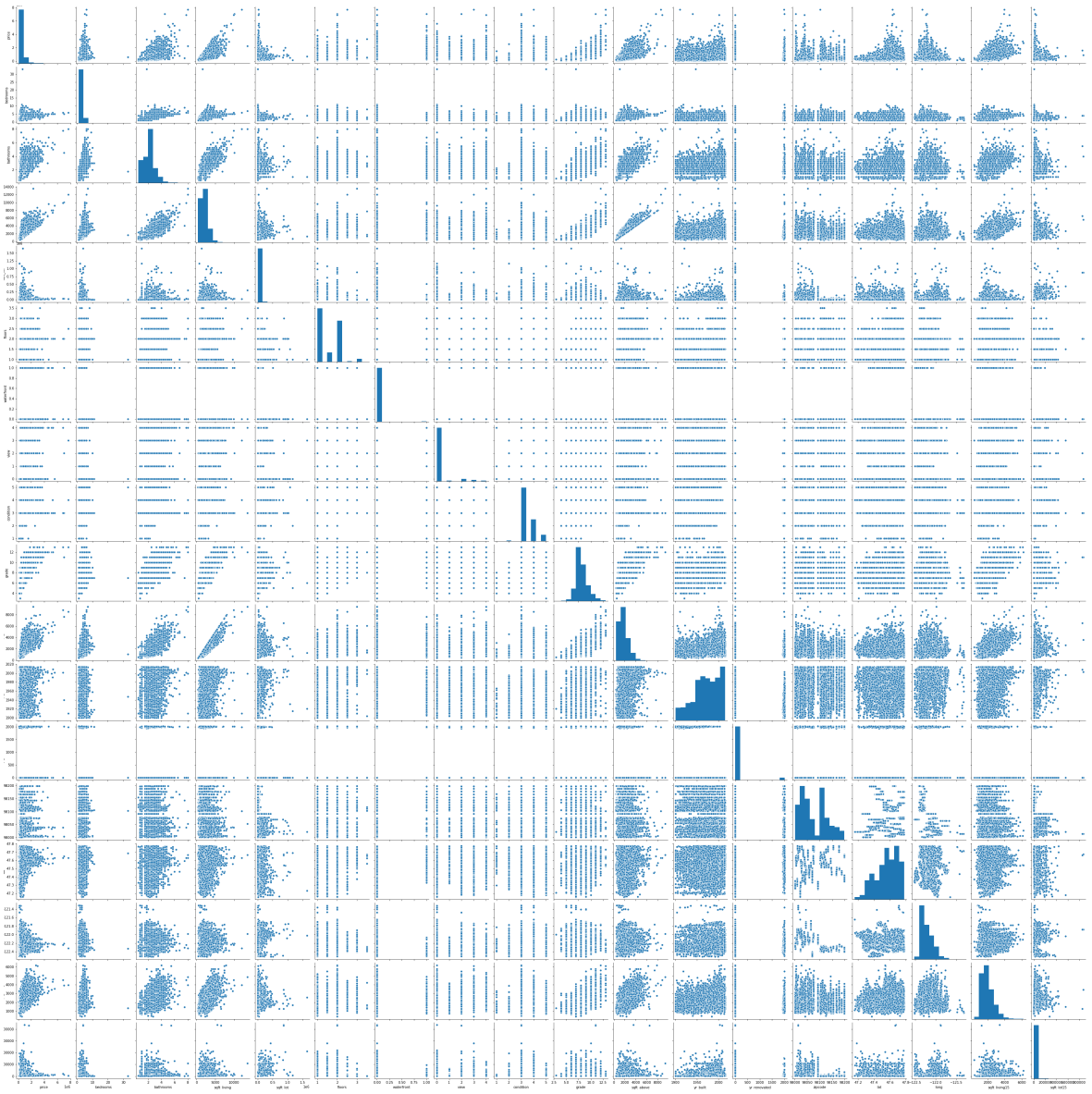
Corrplot of the numerical data



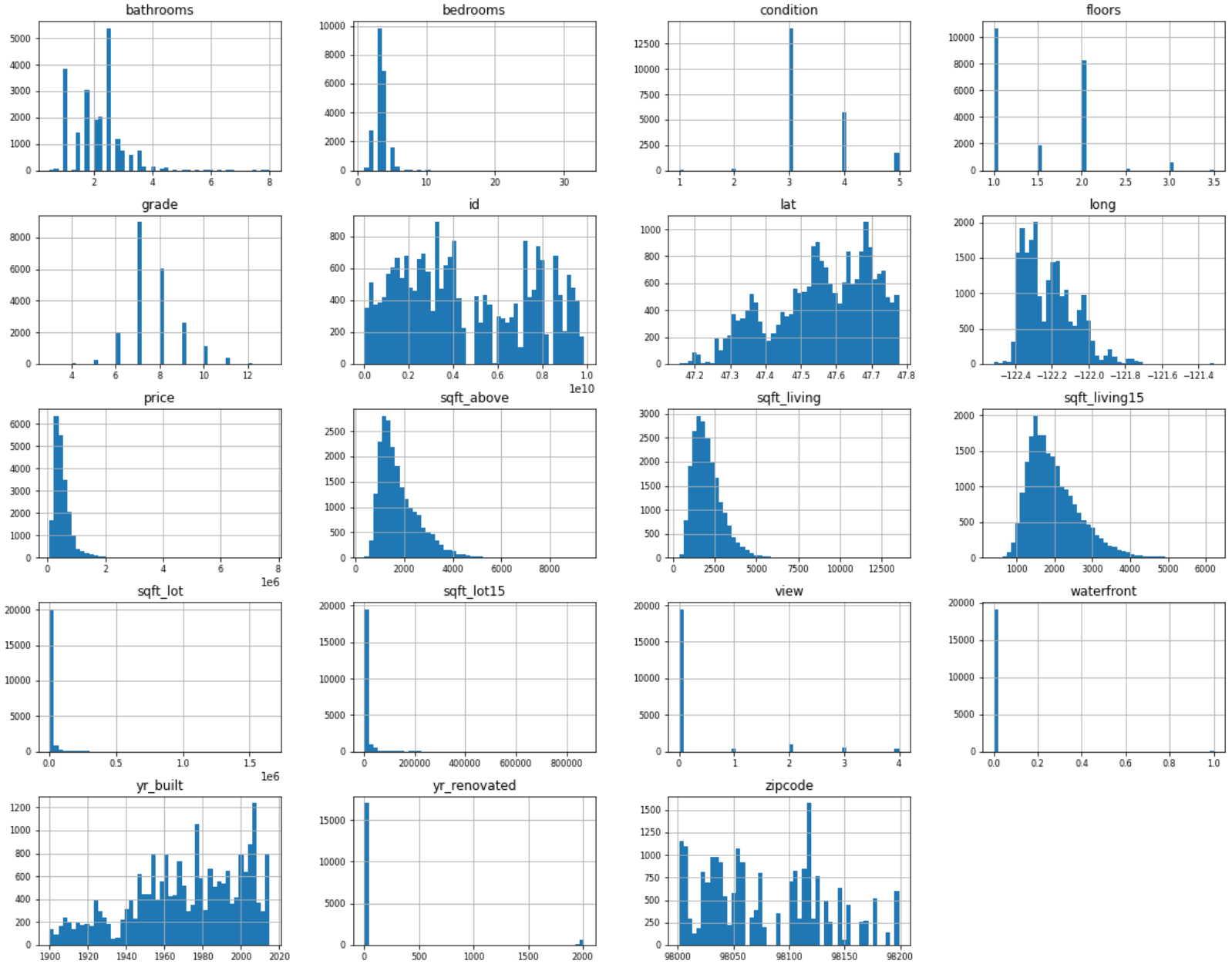
King County Data Set

#	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	datetime	21597 non-null	datetime64[ns]

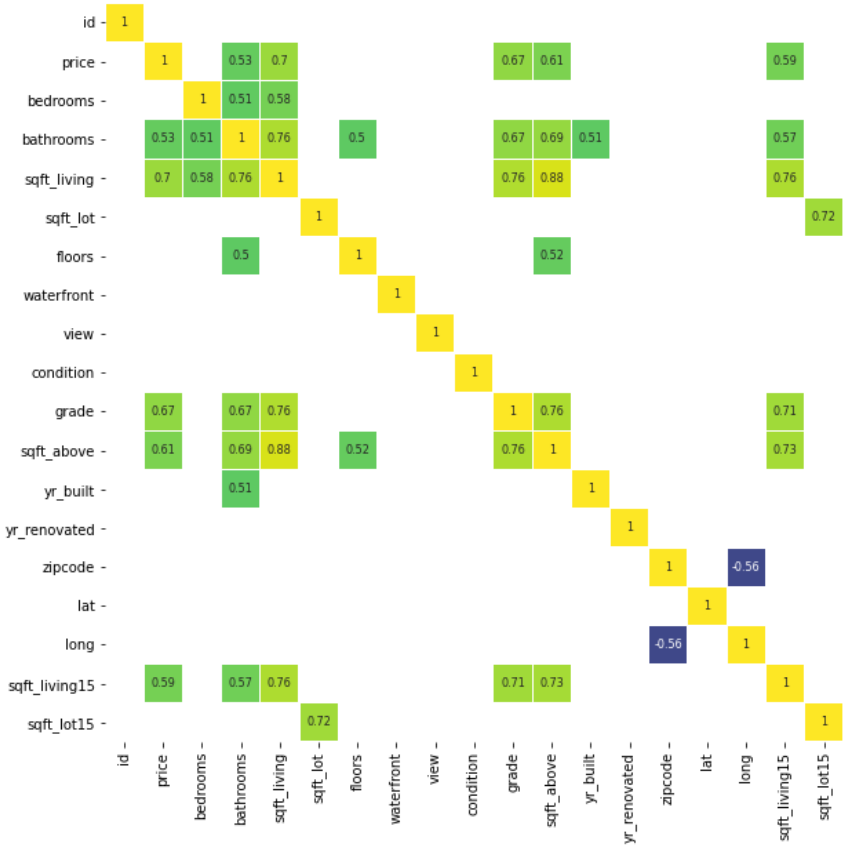
Pairplot of the numerical data



Histplot of the numerical data



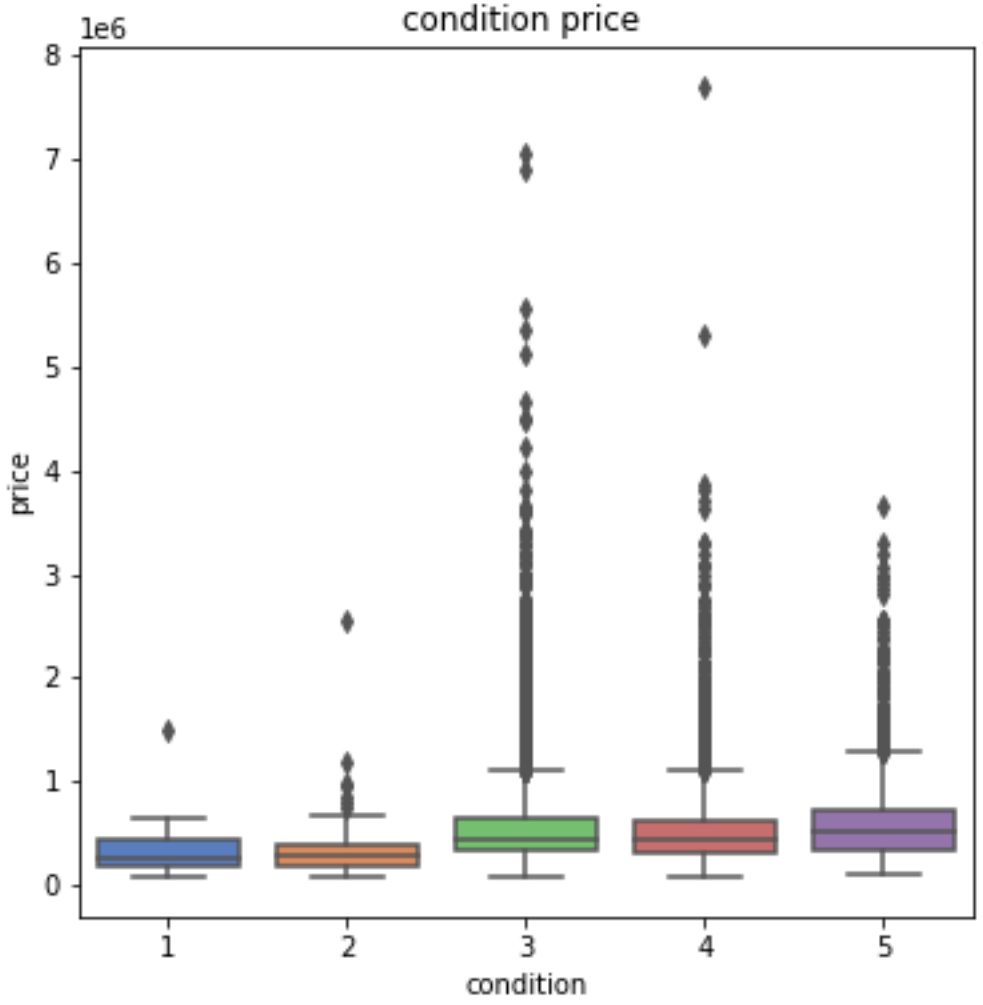
Corrplot of the numerical data

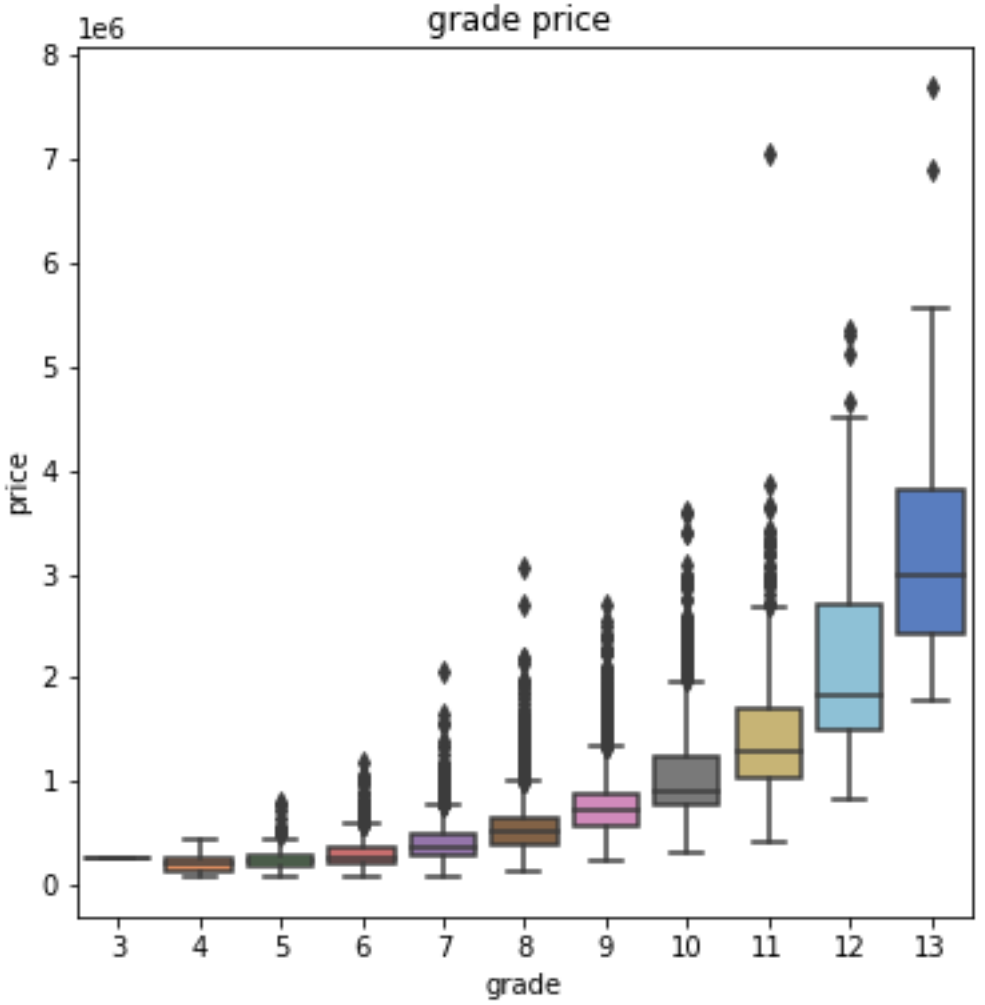


Questions

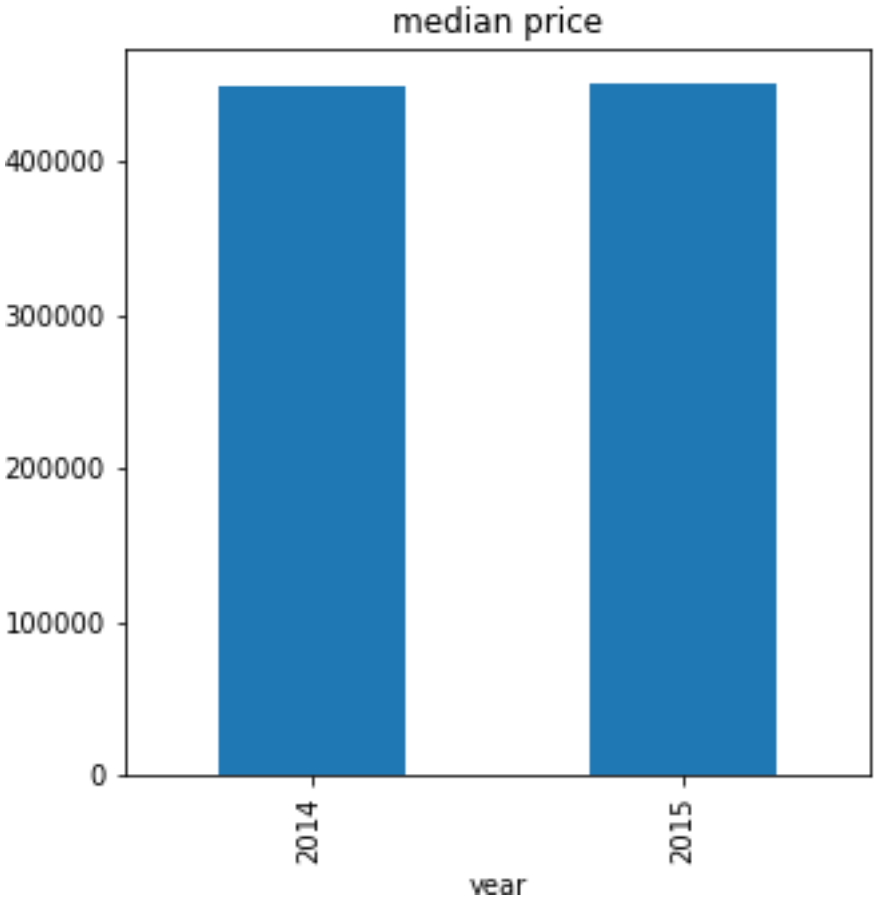
- Is condition or grade better to estimate the price?
- What is the best time to buy or sell a house?
- Is a house with a waterfront more expensive than others?
- Does the house size effect the price?
- Do the neighbours effect the house price?

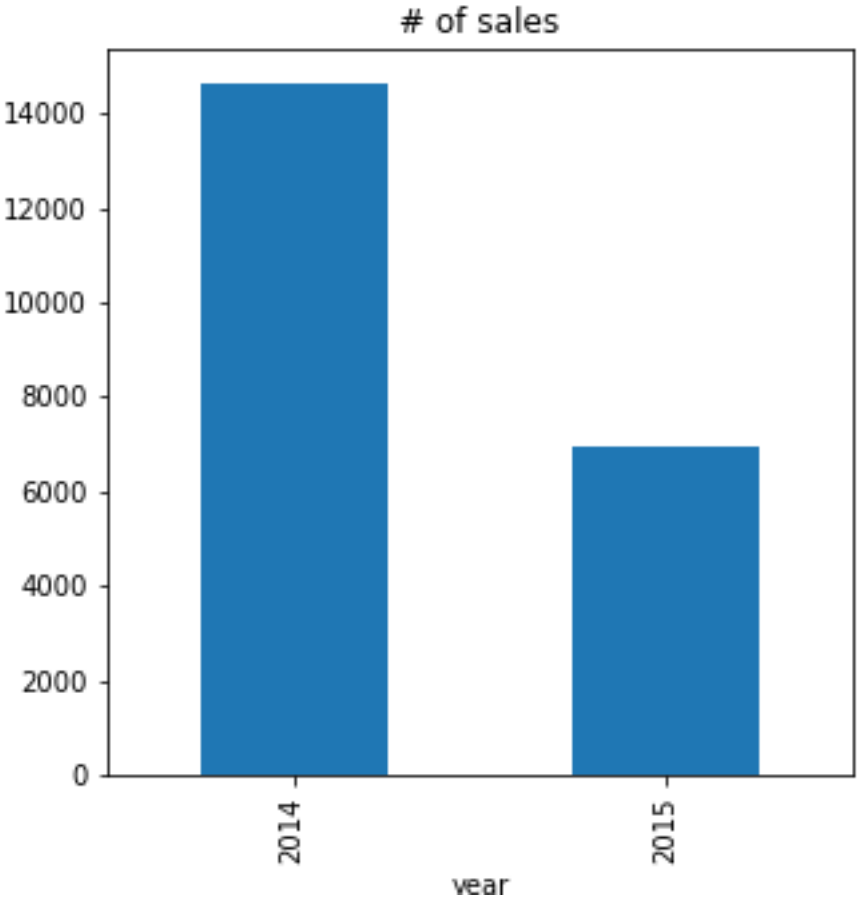
Is condition or grade better to estimate the price?



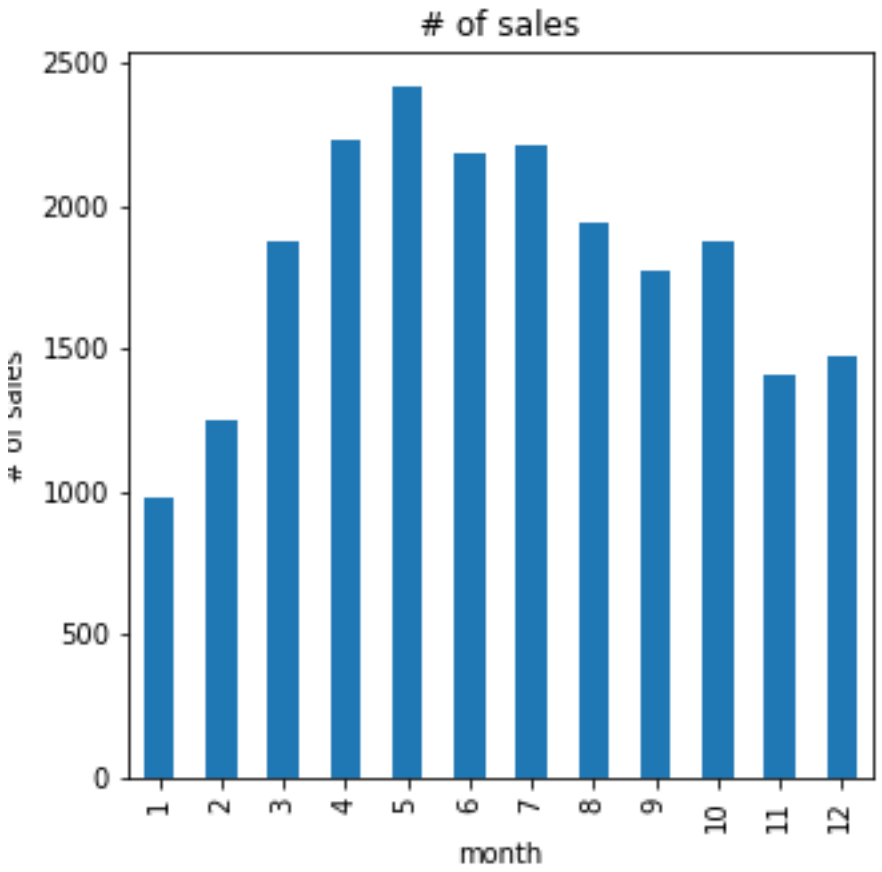


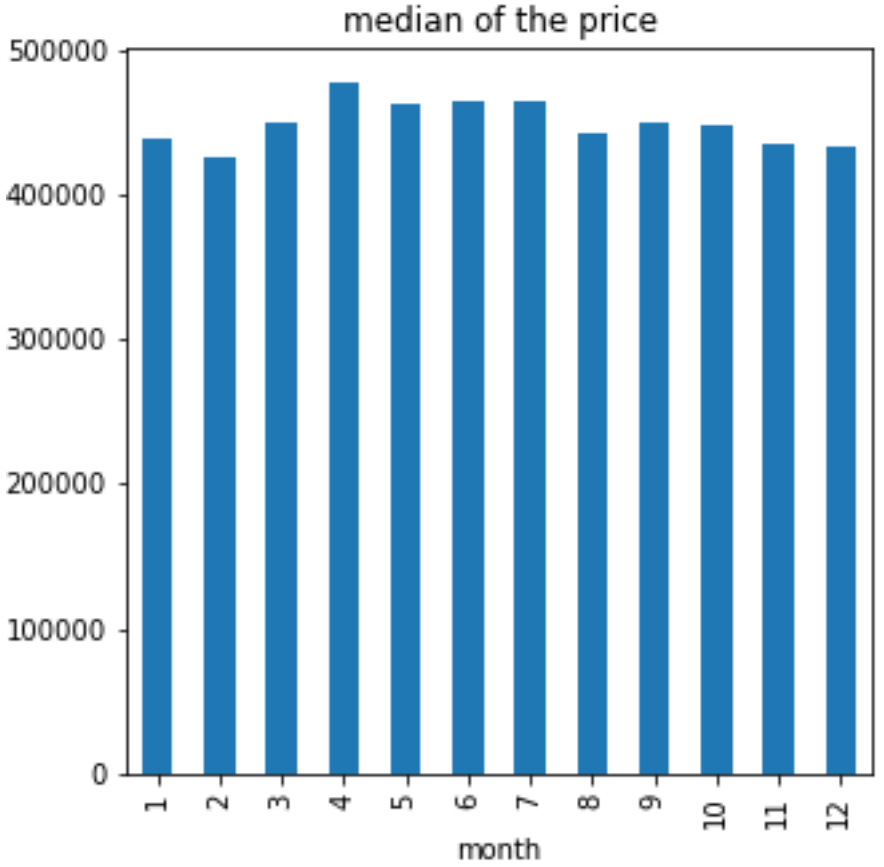
What is the best time to buy or sell a house?



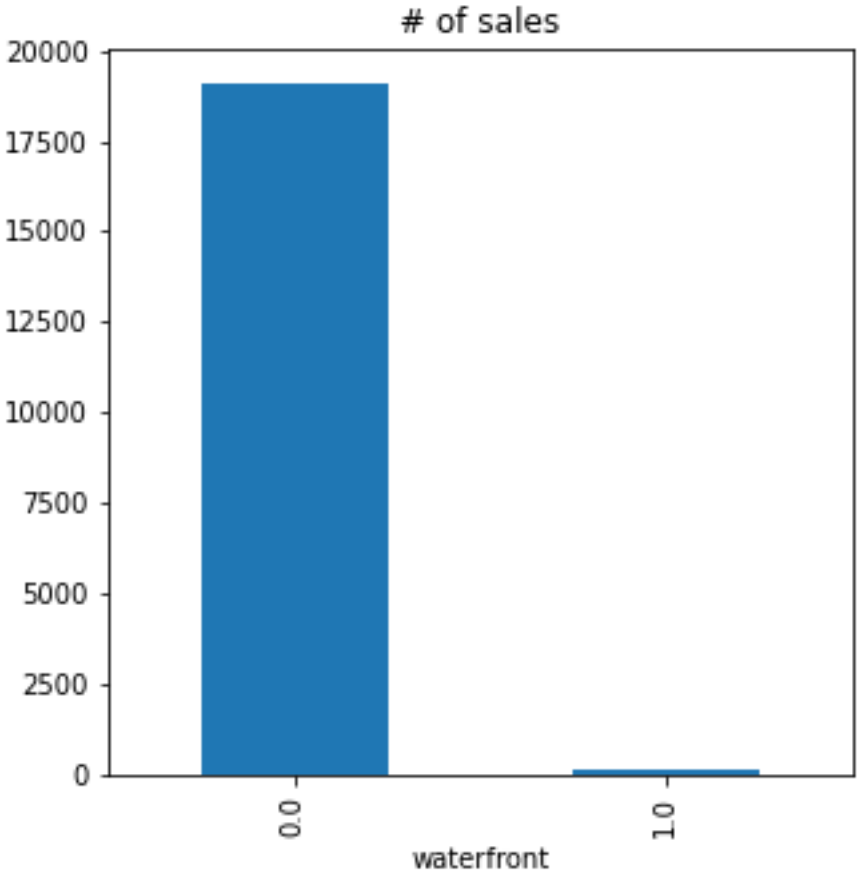


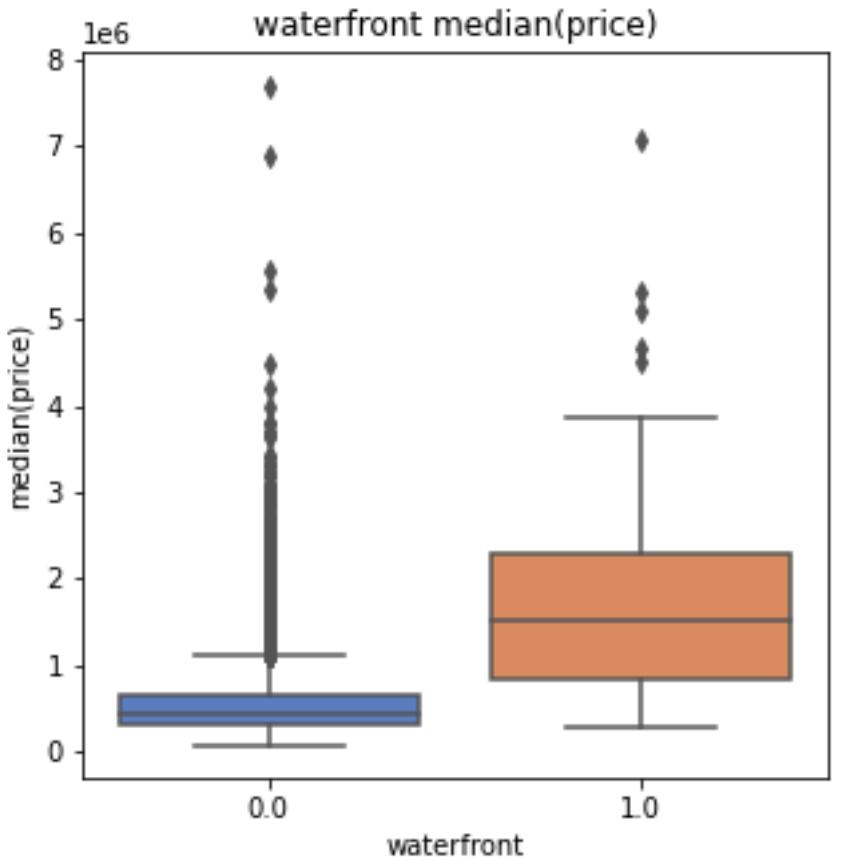
What is the best time to buy or sell a house?



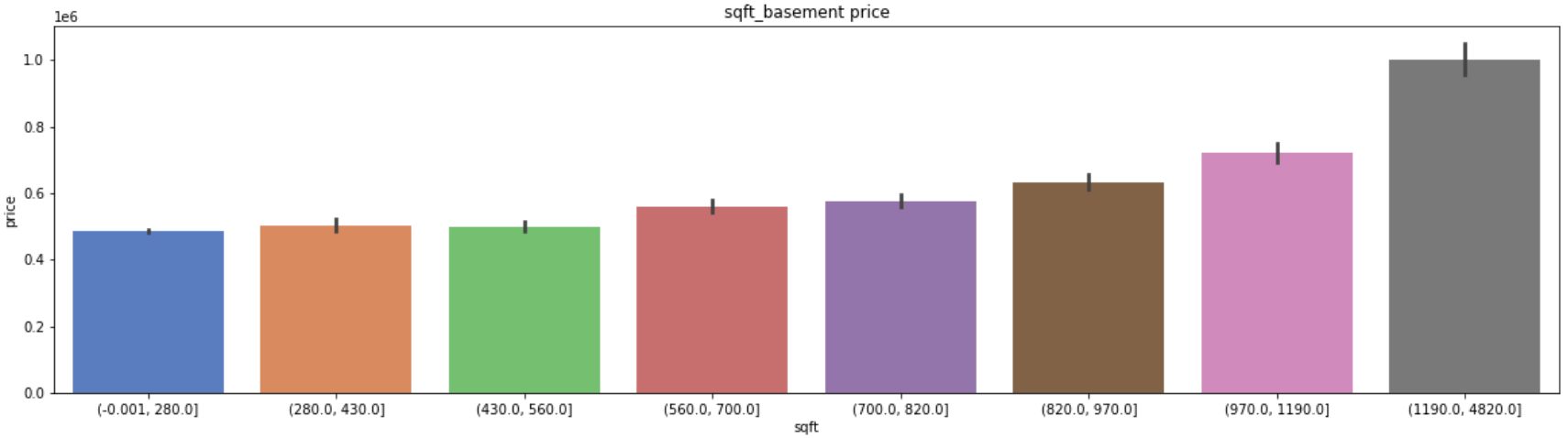


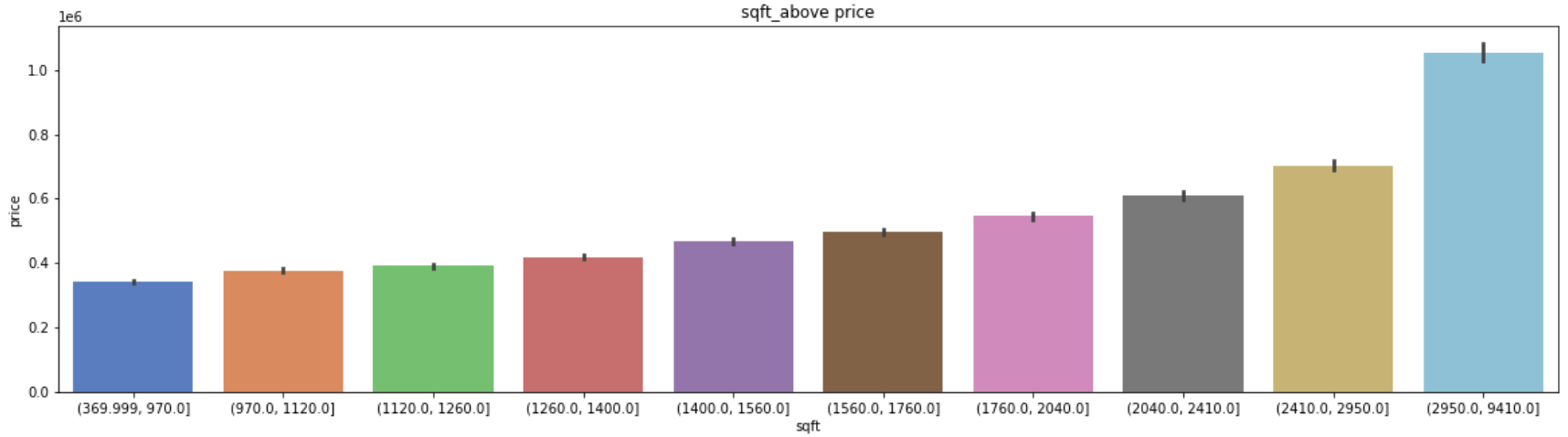
Is a house with a waterfront more expensive than others?



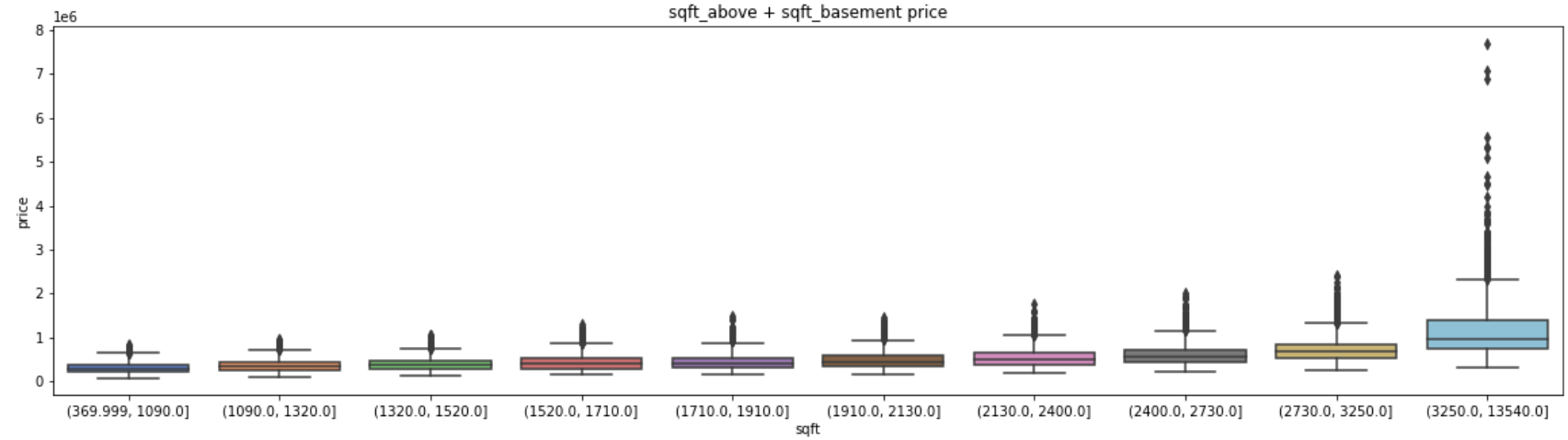


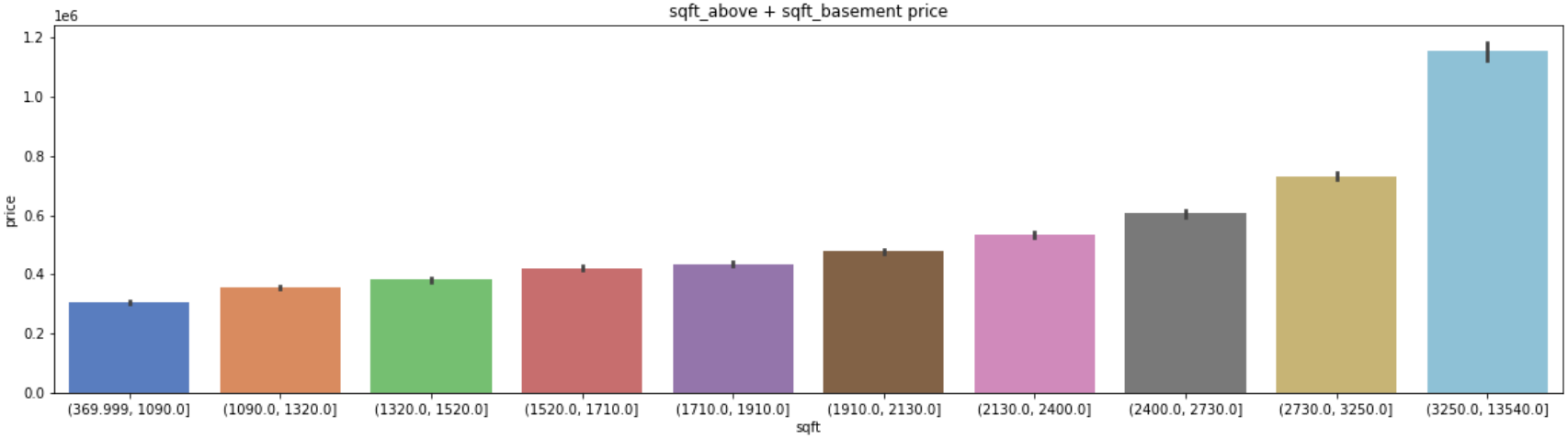
Does the house size effect the price?



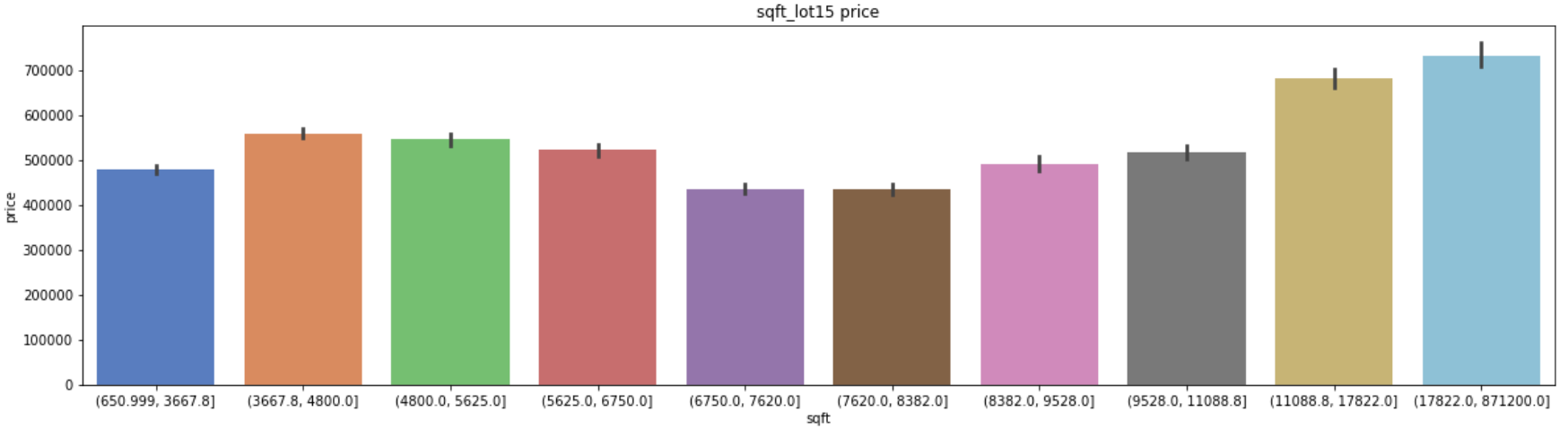


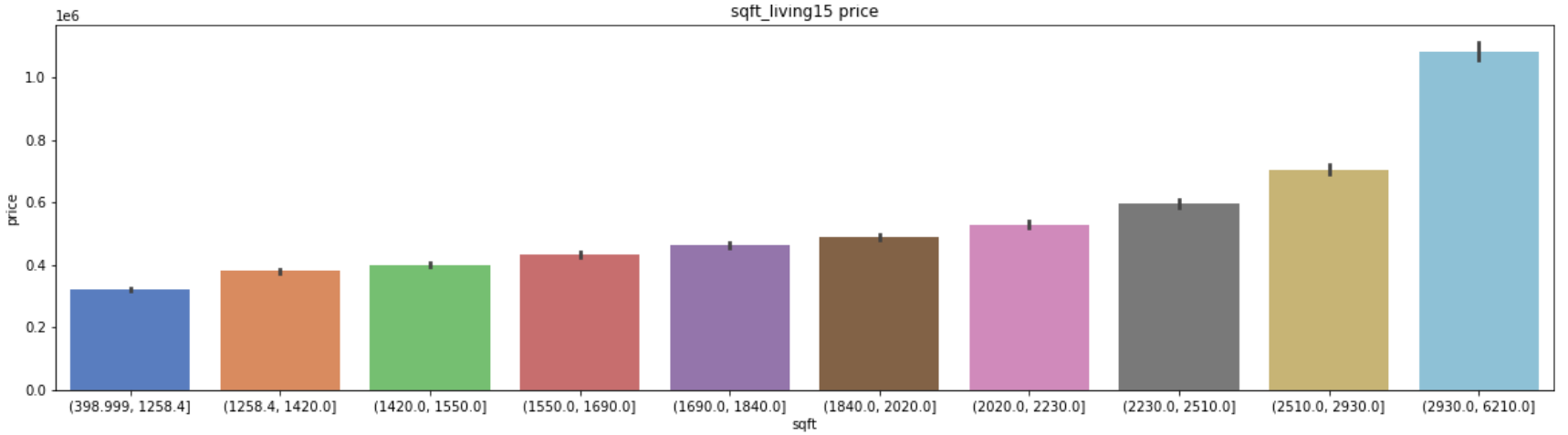
Does the house size effect the price?



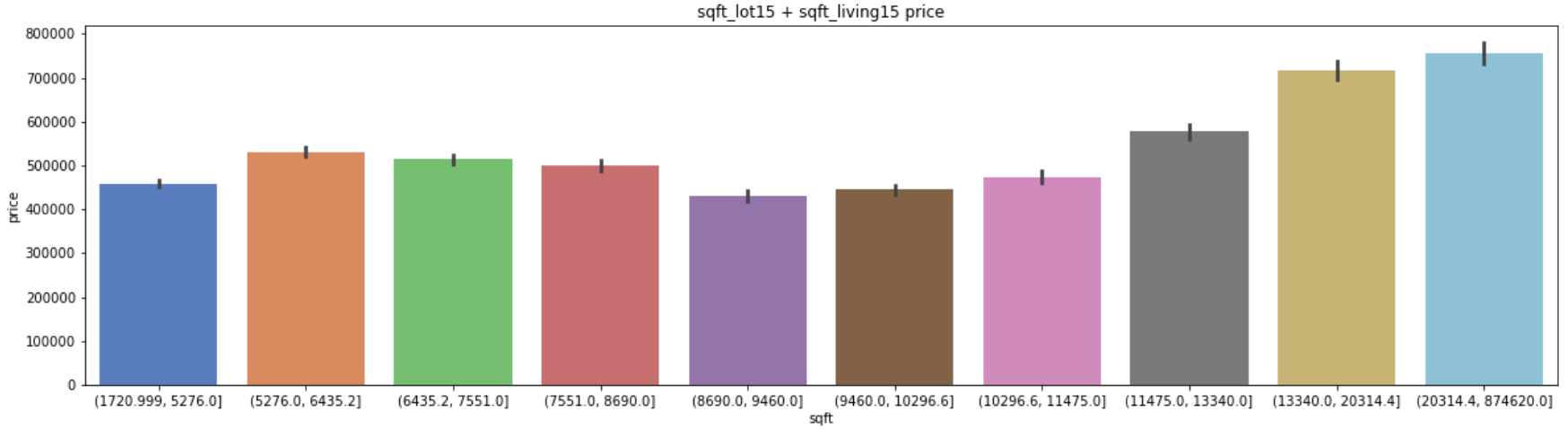


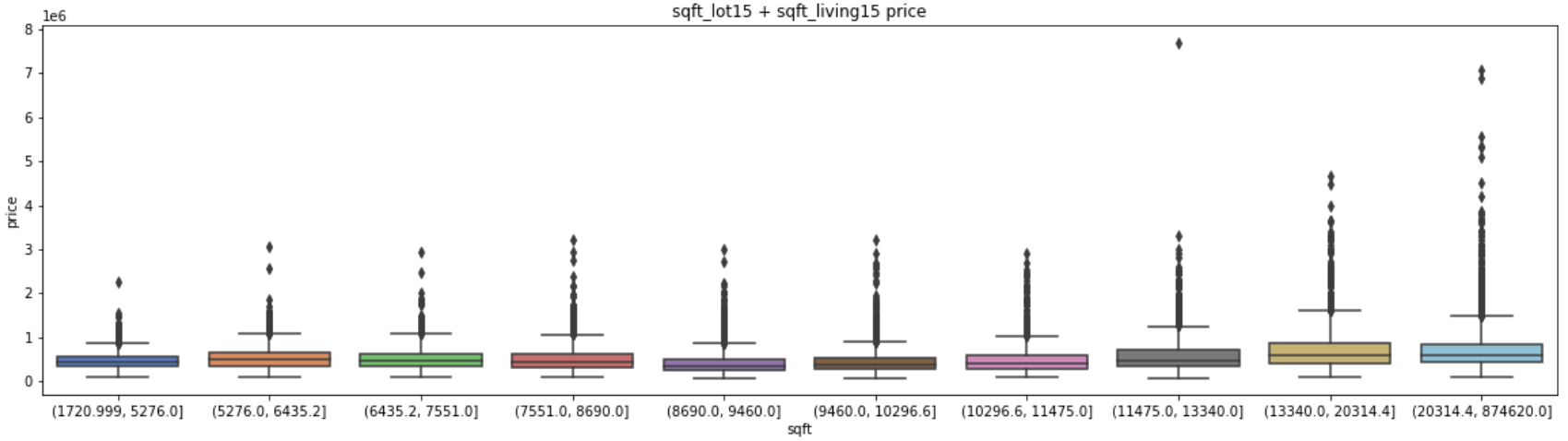
Do the neighbours effect the house price?

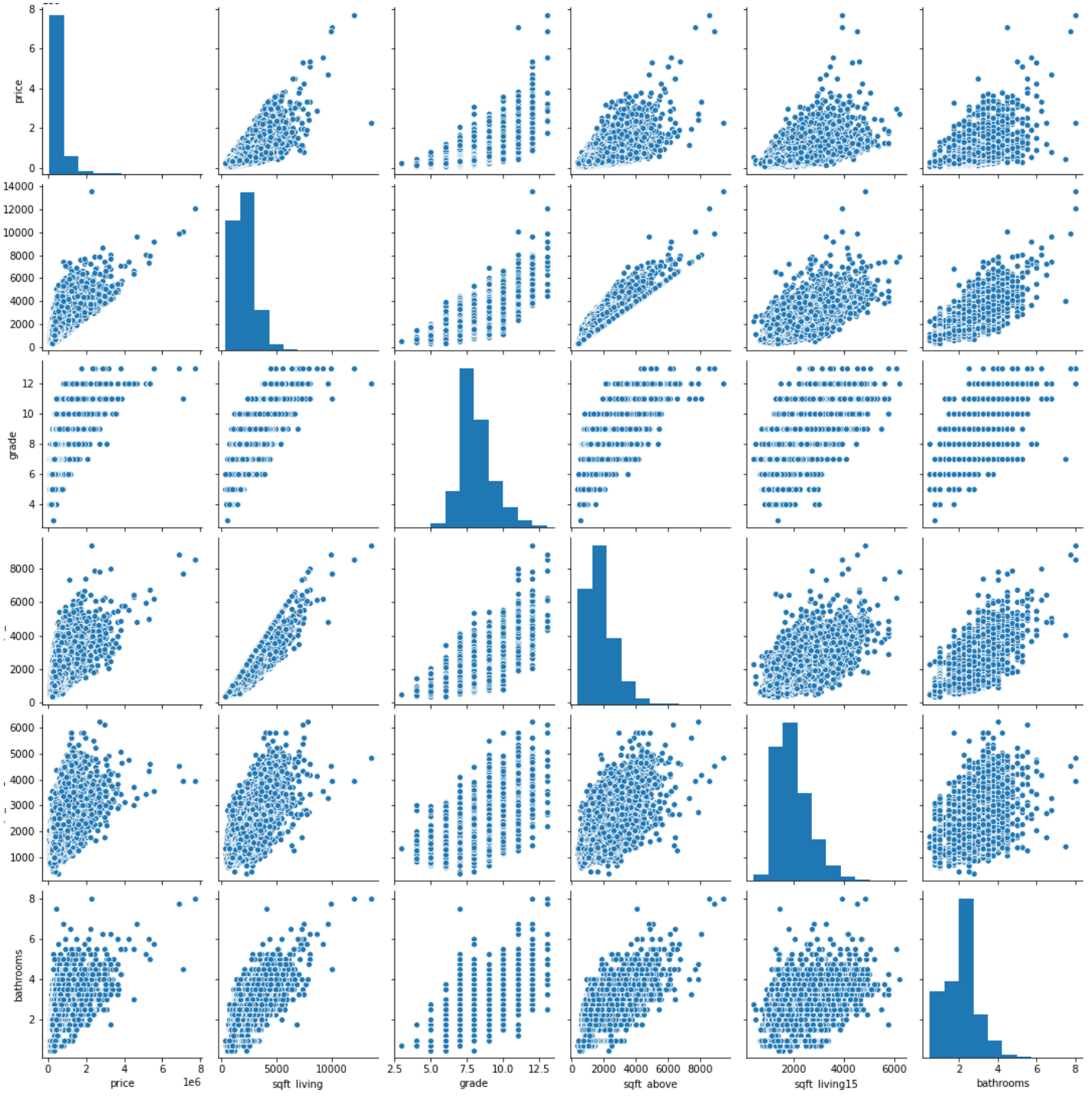




Do the neighbours effect the house price?

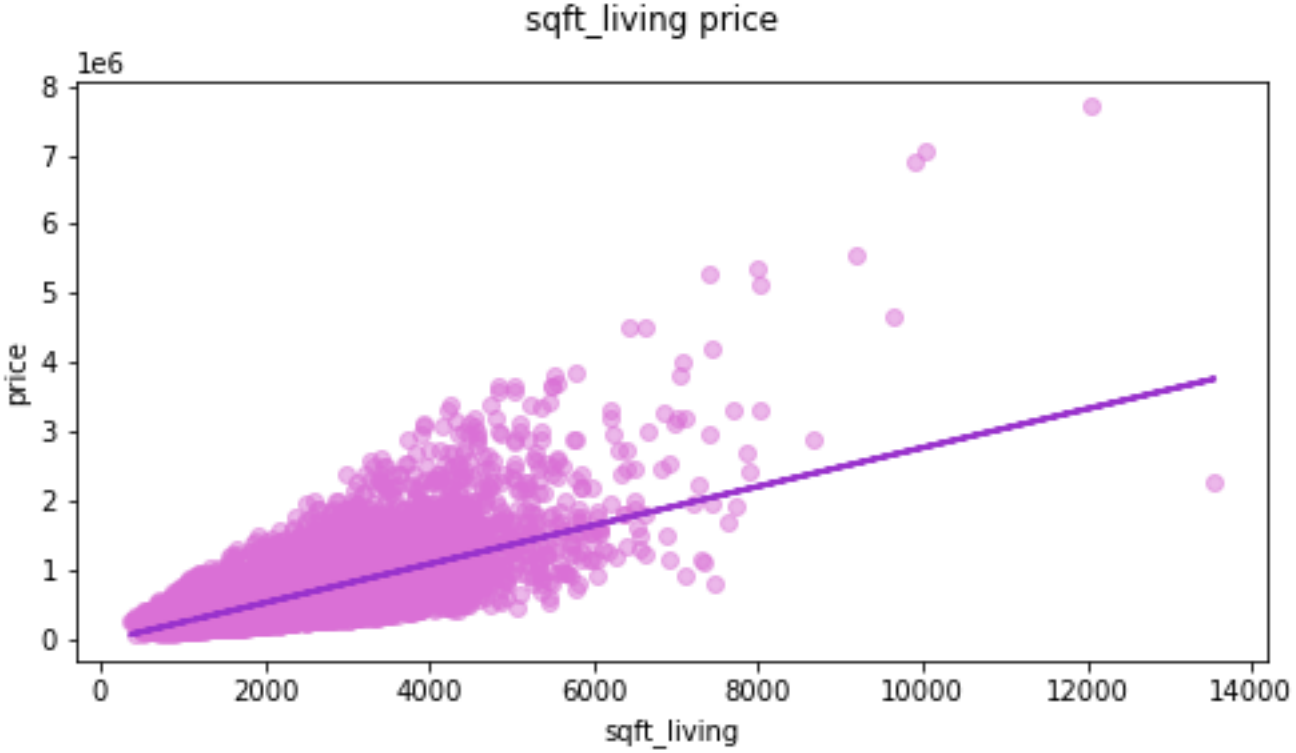





price
grade
bathrooms
1e6
price
sqft_living
grade
sqft_above
sqft_living15
bathrooms

Linear Regression with statsmodels.formula.api

- R-squared: 0.493
- price ~ sqft_living



Multi Linear Regression with statsmodels.formula.api

- R-squared: 0.854
- Adj. R-squared: 0.847

Code:

```
model = smf.ols('price ~ datetime + C(bedrooms) + C(bathrooms) +\
                      sqft_living +  sqft_lot +\
                      C(floors) +   sqft_living15\
                      C(waterfront)     + C(view) + \
                      C(condition)      + C(grade) +\
                      sqft_above        + sqft_basement_num +\
                      C(yr_built)       + C(yr_renovated) +\
                      C(zipcode)        + lat + \
                      long              +  sqft_lot15', data=data)
```

Multi Linear Regression with sklearn

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from scipy import stats
import seaborn as sns

import statsmodels.formula.api as smf
import pathlib

from sklearn.metrics import mean_squared_error, r2_score
from sklearn.metrics import accuracy_score
from sklearn import linear_model
from sklearn.model_selection import train_test_split

import pickle
```

Multi Linear Regression with sklearn

```
data = pd.read_csv('kc_house_prices/King_County_House_prices_dataset.csv')
data['sqft_basement_num'] = pd.to_numeric(data[data['sqft_basement']!
='?'].sqft_basement)
data.head()


data = data.replace([np.inf, -np.inf], np.nan)
data.dropna(inplace=True)


Y = data['price']

X = data[['price', 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot',
       'floors', 'waterfront', 'view', 'condition', 'grade', 'sqft_above',
       'yr_built', 'yr_renovated', 'zipcode', 'lat', 'long',
       'sqft_living15', 'sqft_lot15', 'sqft_basement_num']]

X = pd.get_dummies(data=X, drop_first=True)
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = .20,
random_state = 42)
```

Multi Linear Regression with sklearn

```
my_model = linear_model.LinearRegression() # Do not use fit_intercept = False if you
have removed 1 column after dummy encoding
my_model.fit(X_train, Y_train)
y_pred = my_model.predict(X_test)

#print('Coefficients: \n', regr.coef_)
# The mean squared error
print('Mean squared error: %.2f'
      % mean_squared_error(Y_test, y_pred))

# The coefficient of determination: 1 is perfect prediction
print('Coefficient of determination: %.2f'
      % r2_score(Y_test, y_pred))

_> output
Mean squared error: 0.00
Coefficient of determination: 1.00
```

Multi Linear Regression with sklearn

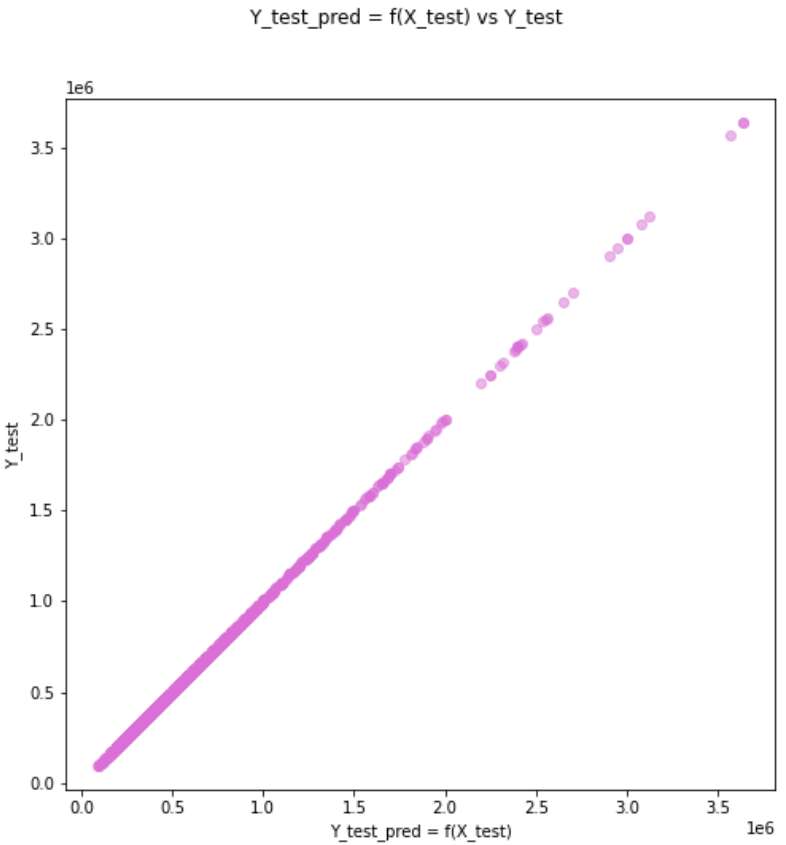
```
# save the model to disk
filename = 'finalized_model.sav'
pickle.dump(my_model, open(filename, 'wb'))

print('\n some time later...')

# load the model from disk
loaded_model = pickle.load(open(filename, 'rb'))
result = loaded_model.score(X_test, Y_test)
print(result)
```

Multi Linear Regression with sklearn

```
fig, ax = plt.subplots(figsize=(8, 8))
ax.scatter(y_pred, Y_test, \
         alpha=0.5, color='orchid')
fig.suptitle(\
    'Y_test_pred = f(X_test) vs Y_test ')
ax.axis('equal')
ax.set_ylabel("Y_test");
ax.set_xlabel("Y_test_pred = f(X_test)");
plt.show()
```



To improve

- Renovation via price
- Model
  - Refine the linear model
  - Cut outliners
  - Normalize the data set
  - Compare linear model to other models
- Map Gps points
- Convert to „normale“ units Sqrf to sqrm :)
- Create use case for customers
  - I like to have/sell that house with these properties and I want to have ...