Linear Regression

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
df = pd.read_csv("Housing dataset.csv")
df
```

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	..
0	7129300520	20141013T000000	221900.0	3	1.00	1180	5650	1.0	0	0	
1	6414100192	20141209T000000	538000.0	3	2.25	2570	7242	2.0	0	0	
2	5631500400	20150225T000000	180000.0	2	1.00	770	10000	1.0	0	0	
3	2487200875	20141209T000000	604000.0	4	3.00	1960	5000	1.0	0	0	
4	1954400510	20150218T000000	510000.0	3	2.00	1680	8080	1.0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...
21608	263000018	20140521T000000	360000.0	3	2.50	1530	1131	3.0	0	0	
21609	6600060120	20150223T000000	400000.0	4	2.50	2310	5813	2.0	0	0	
21610	1523300141	20140623T000000	402101.0	2	0.75	1020	1350	2.0	0	0	
21611	291310100	20150116T000000	400000.0	3	2.50	1600	2388	2.0	0	0	
21612	1523300157	20141015T000000	325000.0	2	0.75	1020	1076	2.0	0	0	

21613 rows × 21 columns

```
Y = df[['price']]
X = df.drop(['price', 'id', 'date'],  axis=1)
X.head()
```

	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	sqft_basement
0	3	1.00	1180	5650	1.0	0	0	3	7	1180	0
1	3	2.25	2570	7242	2.0	0	0	3	7	2170	400
2	2	1.00	770	10000	1.0	0	0	3	6	770	0
3	4	3.00	1960	5000	1.0	0	0	5	7	1050	910
4	3	2.00	1680	8080	1.0	0	0	3	8	1680	0

```
X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 18 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   bedrooms       21613 non-null  int64
 1   bathrooms      21613 non-null  float64
 2   sqft_living    21613 non-null  int64
```

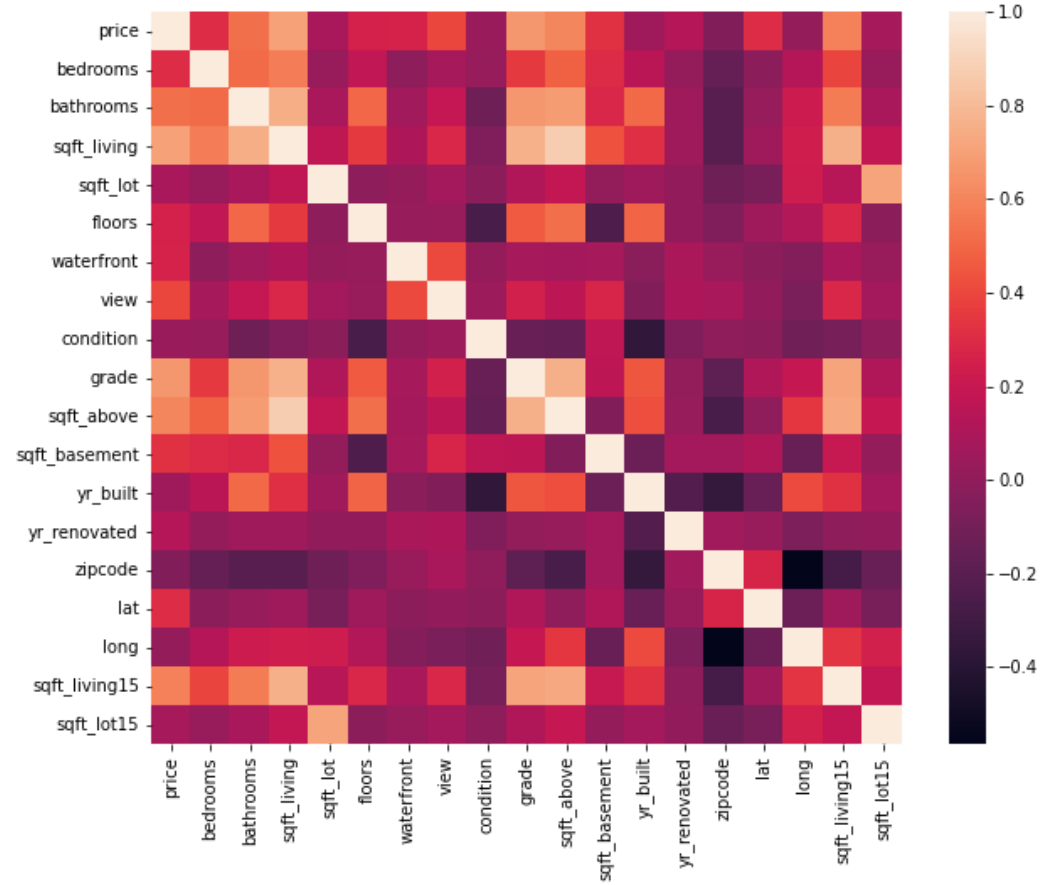
```
 3   sqft_lot        21613 non-null  int64
 4   floors          21613 non-null  float64
 5   waterfront      21613 non-null  int64
 6   view            21613 non-null  int64
 7   condition       21613 non-null  int64
 8   grade           21613 non-null  int64
 9   sqft_above      21613 non-null  int64
 10  sqft_basement   21613 non-null  int64
 11  yr_built        21613 non-null  int64
 12  yr_renovated    21613 non-null  int64
 13  zipcode         21613 non-null  int64
 14  lat             21613 non-null  float64
 15  long            21613 non-null  float64
 16  sqft_living15   21613 non-null  int64
 17  sqft_lot15      21613 non-null  int64
dtypes: float64(4), int64(14)
memory usage: 3.0 MB
```

```
df = df.drop(['id', 'date'], axis=1)
df.head()
```

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	sqft_
0	221900.0	3	1.00	1180	5650	1.0	0	0	3	7	1180	
1	538000.0	3	2.25	2570	7242	2.0	0	0	3	7	2170	
2	180000.0	2	1.00	770	10000	1.0	0	0	3	6	770	
3	604000.0	4	3.00	1960	5000	1.0	0	0	5	7	1050	
4	510000.0	3	2.00	1680	8080	1.0	0	0	3	8	1680	

```
plt.subplots(figsize=(10,8))
sns.heatmap(df.corr())
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd36d631990>
```



Linear Regression Implementation using seaborn.regplot() and scipy.stats

```
from scipy import stats
sns.set(color_codes=True)

slope, intercept, r_value, p_value, std_err = stats.linregress(df['sqft_living'],df['price'])

f = plt.figure(figsize=(10,6))
data = df[['price','sqft_living']]
ax = sns.regplot(x='sqft_living', y='price', data=data,
                 scatter_kws={"color": "g"},
                line_kws={'color': 'r', 'label':"y={0:.1f}x+{1:.1f}".format(slope,intercept)})
ax.legend()
```

```
<matplotlib.legend.Legend at 0x7fd36a817850>
```

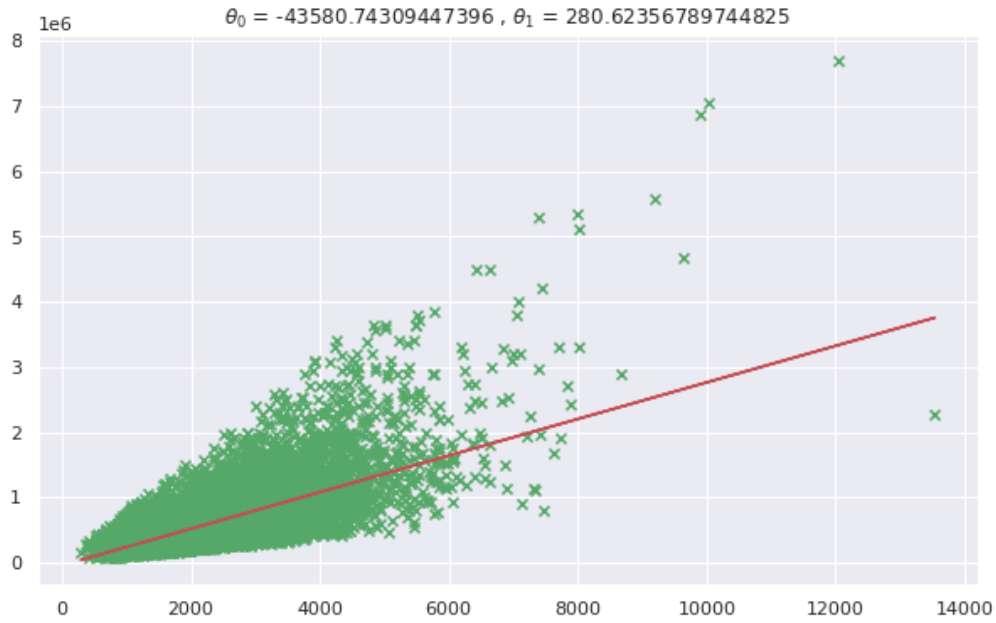
Linear Regression Implementation using Scikit-Learn.

```
x = X[['sqft_living']]
y = Y


xsl = x.values.reshape(-1,1)
ysl = y.values.reshape(-1,1)
xsl = np.concatenate((np.ones(len(xsl)).reshape(-1,1), xsl), axis=1)

from sklearn.linear_model import LinearRegression

slr = LinearRegression()
slr.fit(xsl[:,1].reshape(-1,1), ysl.reshape(-1,1))
y_hat = slr.predict(xsl[:,1].reshape(-1,1))

print('theta[0] = ', slr.intercept_)
print('theta[1] = ', slr.coef_)

thetas = np.array((slr.intercept_, slr.coef_)).squeeze()
```

```
theta[0] =  [-43580.74309447]
theta[1] =  [[280.6235679]]
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:14: VisibleDeprecationWarning: Creating an ndarray from r
```

```
plt.figure(figsize=(10,6))
plt.title('$\\theta_0$ = {} , $\\theta_1$ = {}'.format(thetas[0], thetas[1]))
plt.scatter(xsl[:,1],y, marker='x', color='g')
plt.plot(xsl[:,1], np.dot(xsl, thetas), 'r')
```

```
[<matplotlib.lines.Line2D at 0x7fd367aa9410>]
```




```
print("Slope = ", slope, "\nIntercept = ",intercept)
```

```
Slope =  280.6235678974483
Intercept =  -43580.74309447408
```

```
print("Standard Error = ",std_err)
```

```
Standard Error =  1.9363985519989133
```