In [1]:

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

%matplotlib inline
%pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

In [3]:

```
df = pd.read_csv('/users/bricepratt/desktop/python projects/kc_house_data.csv')
```

In [4]:

```
df.head()
```

Out[4]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft_above
0	7129300520	20141013T000000	221900.0	3	1.00	1180	5650	1.0	0	0	...	7	1180
1	6414100192	20141209T000000	538000.0	3	2.25	2570	7242	2.0	0	0	...	7	2170
2	5631500400	20150225T000000	180000.0	2	1.00	770	10000	1.0	0	0	...	6	770
3	2487200875	20141209T000000	604000.0	4	3.00	1960	5000	1.0	0	0	...	7	1050
4	1954400510	20150218T000000	510000.0	3	2.00	1680	8080	1.0	0	0	...	8	1680

5 rows × 21 columns

```
In [5]: df.isnull().any()
```

```
Out[5]: id                False
        date              False
        price             False
        bedrooms          False
        bathrooms         False
        sqft_living       False
        sqft_lot          False
        floors            False
        waterfront        False
        view              False
        condition         False
        grade             False
        sqft_above        False
        sqft_basement     False
        yr_built          False
        yr_renovated      False
        zipcode           False
        lat               False
        long              False
        sqft_living15     False
        sqft_lot15        False
        dtype: bool
```

In [6]:
```
df.dtypes
```

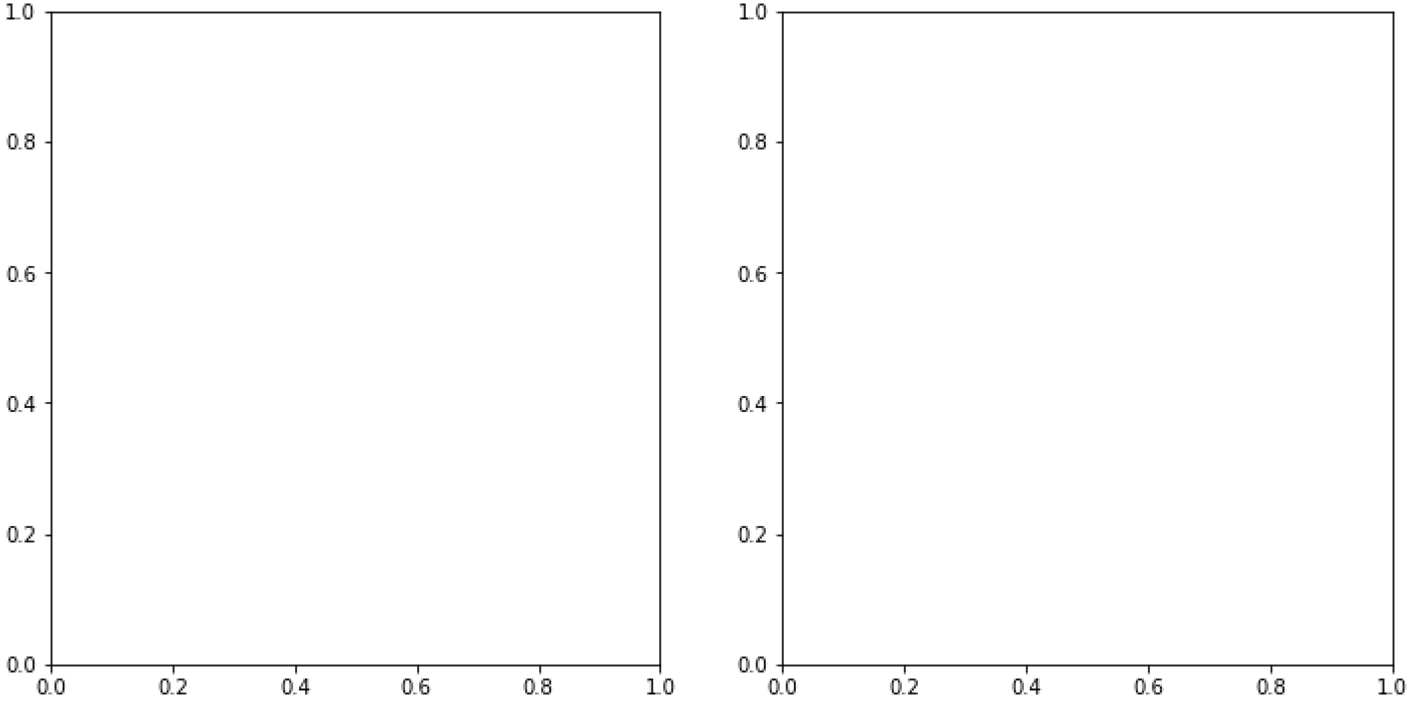
Out[6]:
```
id                 int64
date              object
price            float64
bedrooms           int64
bathrooms        float64
sqft_living        int64
sqft_lot           int64
floors           float64
waterfront         int64
view               int64
condition          int64
grade              int64
sqft_above         int64
sqft_basement      int64
yr_built           int64
yr_renovated       int64
zipcode            int64
lat              float64
long             float64
sqft_living15      int64
sqft_lot15         int64
dtype: object
```

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

Out[7]:

	id	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view
count	2.161300e+04	2.161300e+04	21613.000000	21613.000000	21613.000000	2.161300e+04	21613.000000	21613.000000	21613.000000
mean	4.580302e+09	5.400881e+05	3.370842	2.114757	2079.899736	1.510697e+04	1.494309	0.007542	0.234303
std	2.876566e+09	3.671272e+05	0.930062	0.770163	918.440897	4.142051e+04	0.539989	0.086517	0.766318
min	1.000102e+06	7.500000e+04	0.000000	0.000000	290.000000	5.200000e+02	1.000000	0.000000	0.000000
25%	2.123049e+09	3.219500e+05	3.000000	1.750000	1427.000000	5.040000e+03	1.000000	0.000000	0.000000
50%	3.904930e+09	4.500000e+05	3.000000	2.250000	1910.000000	7.618000e+03	1.500000	0.000000	0.000000
75%	7.308900e+09	6.450000e+05	4.000000	2.500000	2550.000000	1.068800e+04	2.000000	0.000000	0.000000
max	9.900000e+09	7.700000e+06	33.000000	8.000000	13540.000000	1.651359e+06	3.500000	1.000000	4.000000

```
In [8]: fig = plt.figure(figsize=(12,6))
        sqft = fig.add_subplot(121)
        cost = fig.add_subplot(122)
```



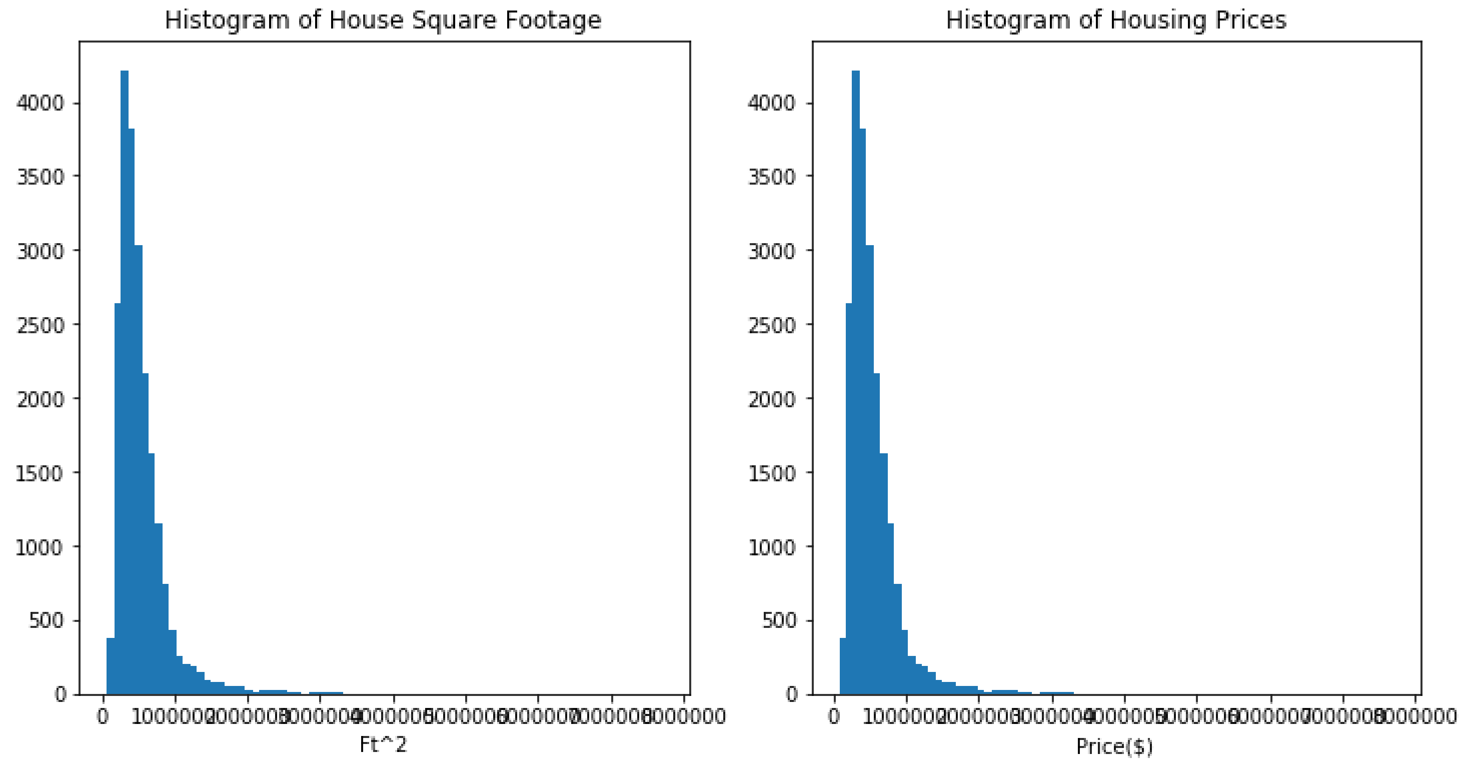
In [9]:

```
fig = plt.figure(figsize=(12,6))
sqft = fig.add_subplot(121)
cost = fig.add_subplot(122)

sqft.hist(df.price, bins=80)
sqft.set_xlabel('Ft^2')
sqft.set_title("Histogram of House Square Footage")

cost.hist(df.price, bins=80)
cost.set_xlabel('Price($)')
cost.set_title("Histogram of Housing Prices")

plt.show()
```



Histogram of House Square Footage

Histogram of Housing Prices

In [10]:

```
import statsmodels.api as sm
from statsmodels.formula.api import ols
```

In [11]:
```
m = ols('price ~ sqft_living', df).fit()
print(m.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.493
Model:                            OLS   Adj. R-squared:                  0.493
Method:                 Least Squares   F-statistic:                 2.100e+04
Date:                Sun, 03 May 2020   Prob (F-statistic):               0.00
Time:                        18:43:03   Log-Likelihood:            -3.0027e+05
No. Observations:               21613   AIC:                         6.005e+05
Df Residuals:                   21611   BIC:                         6.006e+05
Df Model:                           1
Covariance Type:            nonrobust
===============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
Intercept   -4.358e+04   4402.690     -9.899      0.000   -5.22e+04    -3.5e+04
sqft_living   280.6236      1.936    144.920      0.000     276.828     284.419
==============================================================================
Omnibus:                    14832.490   Durbin-Watson:                   1.983
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           546444.713
Skew:                           2.824   Prob(JB):                         0.00
Kurtosis:                      26.977   Cond. No.                     5.63e+03
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.63e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [12]:

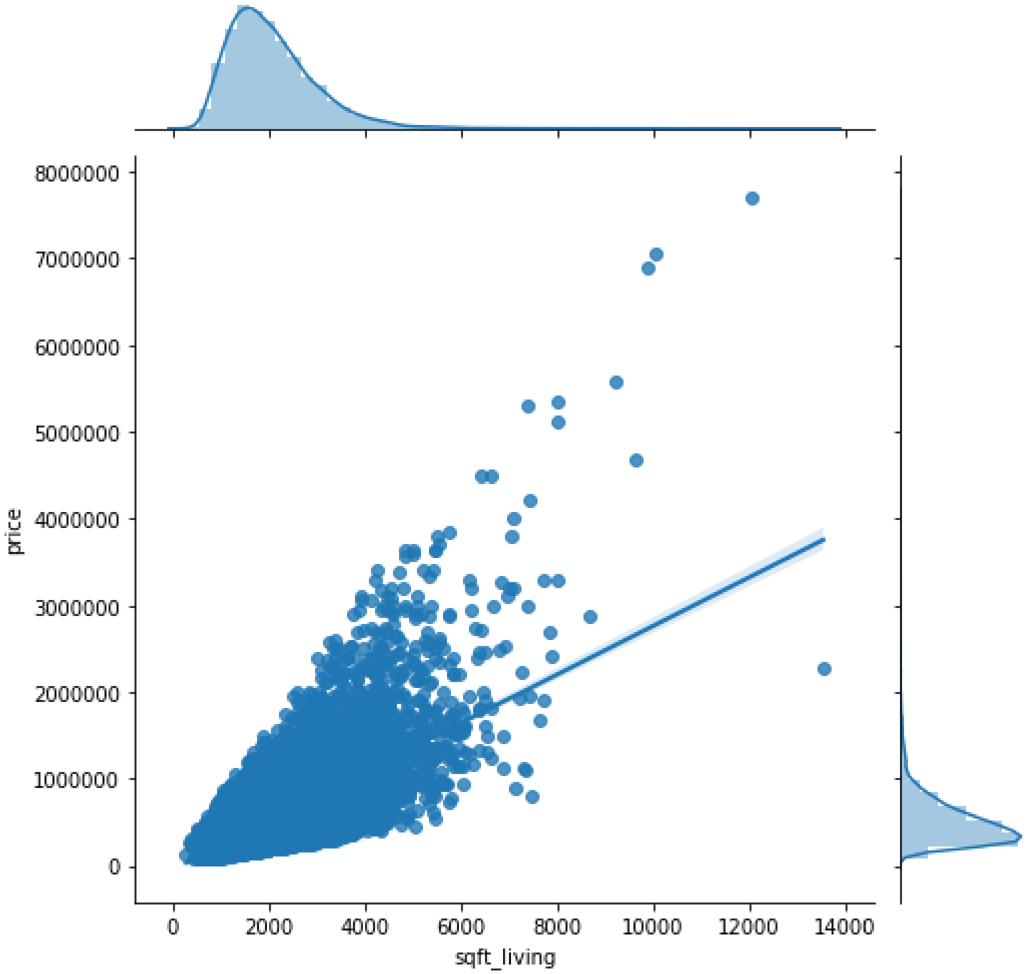
```
m = ols('price ~ sqft_living + bedrooms + grade + condition', df).fit()
print(m.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.555
Model:                            OLS   Adj. R-squared:                  0.555
Method:                 Least Squares   F-statistic:                     6749.
Date:                Sun, 03 May 2020   Prob (F-statistic):               0.00
Time:                        18:45:27   Log-Likelihood:            -2.9884e+05
No. Observations:               21613   AIC:                         5.977e+05
Df Residuals:                   21608   BIC:                         5.977e+05
Df Model:                           4
Covariance Type:            nonrobust
===============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
Intercept   -7.398e+05   1.81e+04    -40.855      0.000   -7.75e+05   -7.04e+05
sqft_living   212.3034      3.249     65.353      0.000     205.936     218.671
bedrooms    -4.568e+04   2222.205    -20.555      0.000      -5e+04   -4.13e+04
grade        1.001e+05   2241.553     44.673      0.000    9.57e+04    1.05e+05
condition    6.615e+04   2598.352     25.457      0.000    6.11e+04    7.12e+04
==============================================================================
Omnibus:                    16773.778   Durbin-Watson:                   1.988
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           973426.793
Skew:                           3.249   Prob(JB):                         0.00
Kurtosis:                      35.229   Cond. No.                     2.50e+04
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.5e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [14]:
```
sns.jointplot(x='sqft_living', y='price', data=df, kind = 'reg', fit_reg=True, size = 7)
plt.show()
```

```
/Users/bricepratt/opt/anaconda3/lib/python3.7/site-packages/seaborn/axisgrid.py:2272: UserWarning: T
he `size` parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)
```



In [ ]:
```
```