2) Linear Regression

Vitor Kamada

July 2018

Reference

Tables, Graphics, and Figures from

https://www.quantopian.com/lectures

Lecture 12 Linear Regression

Linear Regression

$$Y_{TSLS} = \alpha + \beta X_{SPY} + \epsilon$$

```
import numpy as np
from statsmodels import regression
import statsmodels.api as sm
import matplotlib.pyplot as plt
import math
```

Create a Function for Linear Regression

```
def linreg(X,Y):
    # Running the linear regression
    X = sm.add_constant(X)
    model = regression.linear_model.OLS(Y, X).fit()
    a = model.params[0]
    b = model.params[1]
    X = X[:, 1]

    # Return summary of the regression and plot results
    X2 = np.linspace(X.min(), X.max(), 100)
    Y_hat = X2 * b + a
    plt.scatter(X, Y, alpha=0.3) # Plot the raw data
    plt.plot(X2, Y_hat, 'r', alpha=0.9);  # Add the regression l
    plt.xlabel('X Value')
    plt.ylabel('Y Value')
    return model.summary()
```

Get Data

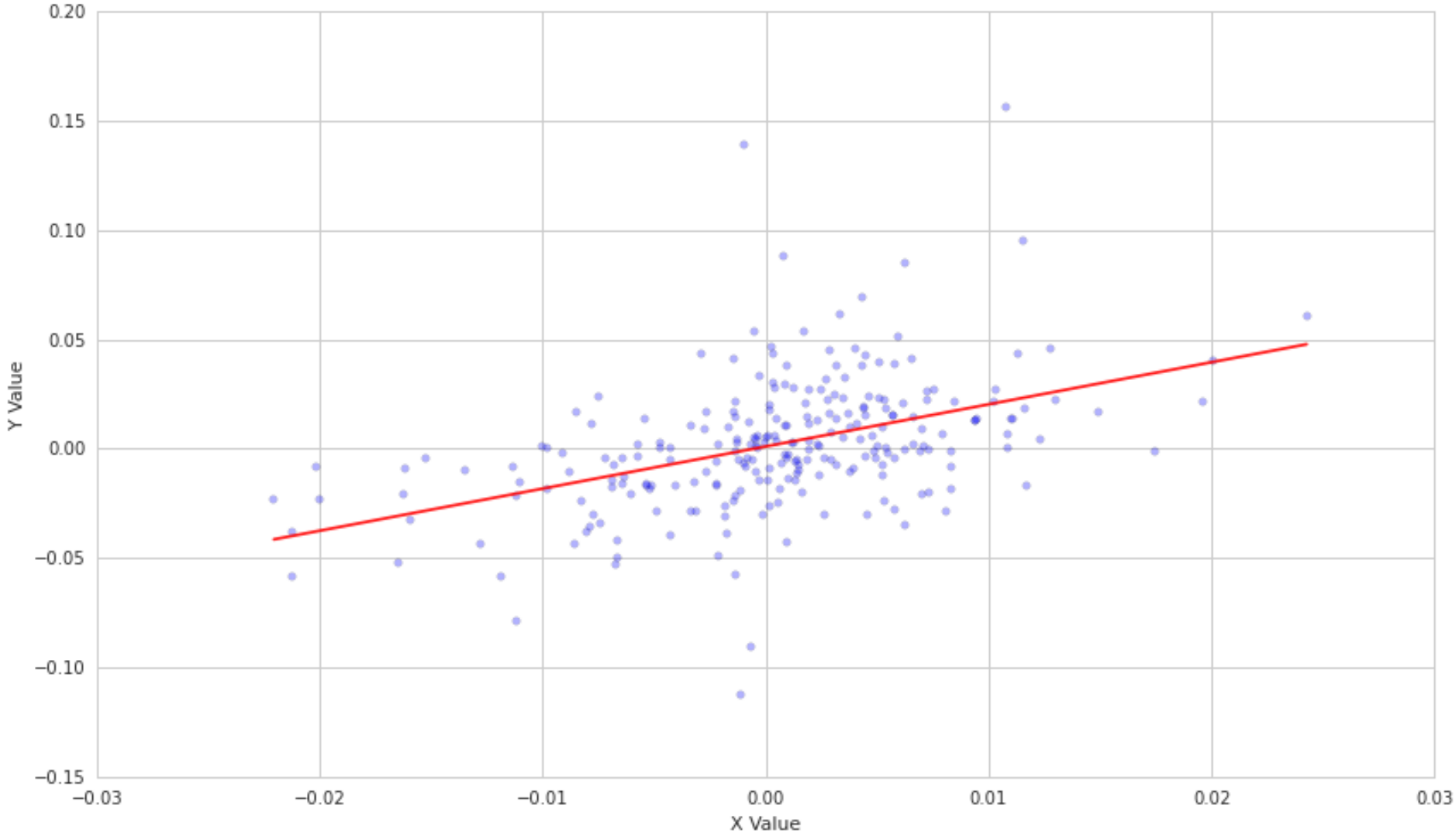
```
start = '2014-01-01'
end = '2015-01-01'
asset = get_pricing('TSLA', fields='price',
                    start_date=start, end_date=end)
benchmark = get_pricing('SPY', fields='price',
                        start_date=start, end_date=end)

# We have to take the percent changes to get to returns
# Get rid of the first (0th) element because it is NAN
r_a = asset.pct_change()[1:]
r_b = benchmark.pct_change()[1:]

linreg(r_b.values, r_a.values)
```

$Y_{TSLS} = \alpha + \beta X_{SPY} + \epsilon$

Dep. Variable:	y	**R-squared:**	0.202
Model:	OLS	**Adj. R-squared:**	0.199
Method:	Least Squares	**F-statistic:**	63.14
Date:	Thu, 17 Sep 2015	**Prob (F-statistic):**	6.66e-14
Time:	20:55:12	**Log-Likelihood:**	548.81
No. Observations:	251	**AIC:**	-1094.
Df Residuals:	249	**BIC:**	-1087.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[95.0% Conf. Int.]**
const	0.0011	0.002	0.626	0.532	-0.002 0.004
x1	1.9271	0.243	7.946	0.000	1.449 2.405

$\hat{Y}_{TSLS} = 0.0011 + 1.92 X_{SPY}$



Ordinary Least Squares (OLS)

$$Y = \alpha + \beta X + \epsilon$$

$$\sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (Y_i - \hat{a} - \hat{\beta} X_i)^2$$

Residual Sum of Squares (RSS)

$$\sum_{i=1}^{n} e_i^2 = e_1^2 + e_2^2 + ... + e_n^2$$

OLS Derivation

$$RSS = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - \hat{\alpha} - \hat{\beta} x_i)^2$$

$$\frac{\partial \sum_{i=1}^{n} e_i^2}{\partial \hat{\alpha}} = -2 \sum_{i=1}^{n} (y_i - \hat{\alpha} - \hat{\beta} x_i) = 0 = \sum_{i=1}^{n} e_i$$

$$\frac{\partial \sum_{i=1}^{n} e_i^2}{\partial \hat{\beta}} = -2 \sum_{i=1}^{n} [x_i (y_i - \hat{\alpha} - \hat{\beta} x_i)] = 0$$

$$\sum_{i=1}^{n} x_i e_i = 0$$

Estimating the Coefficients

$$\hat{y} = \hat{\alpha} + \hat{\beta}x$$

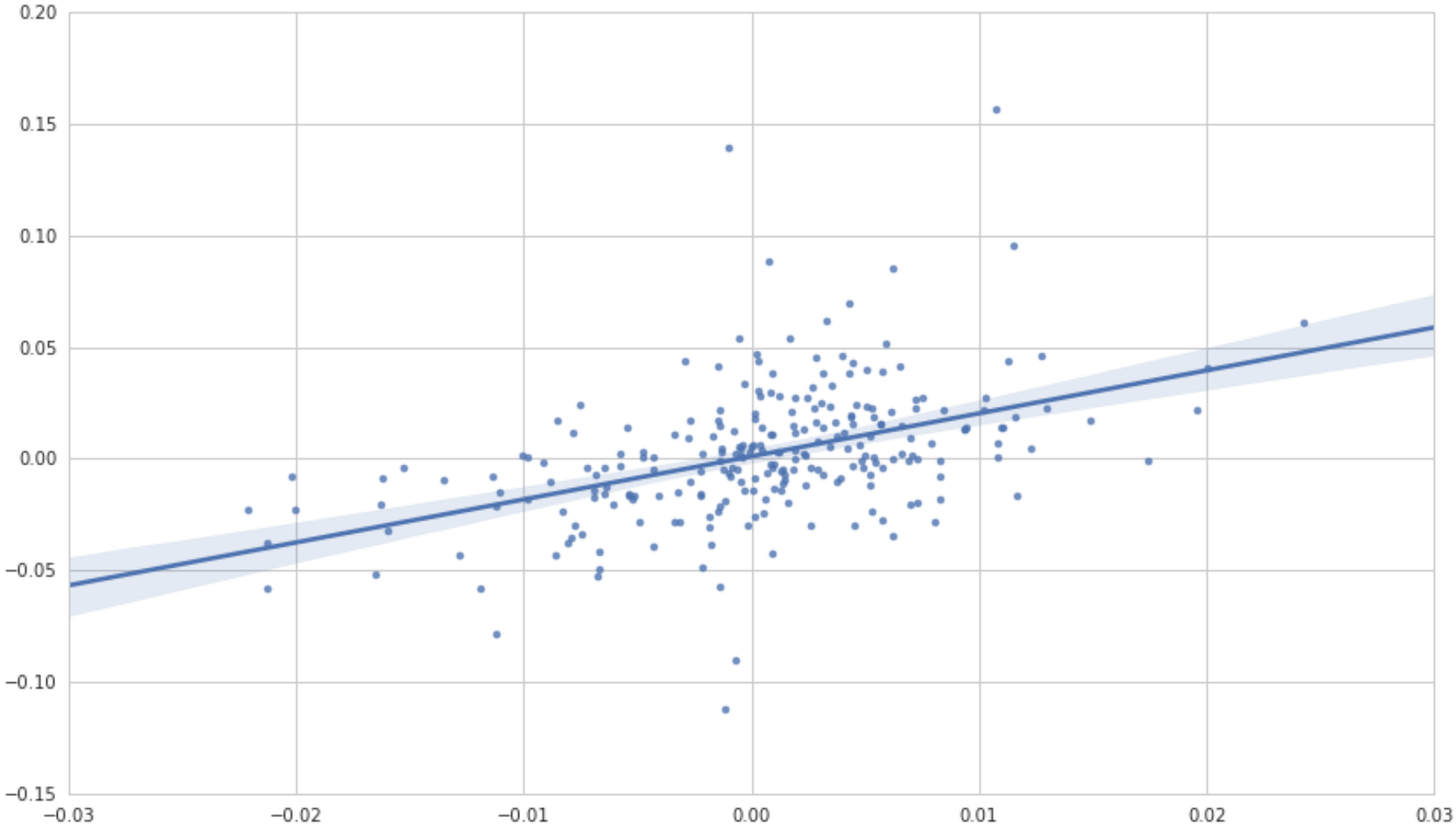
$$\hat{\beta} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$$

Seaborn Library

```
import seaborn

start = '2014-01-01'
end = '2015-01-01'
asset = get_pricing('TSLA', fields='price',
                    start_date=start, end_date=end)
benchmark = get_pricing('SPY', fields='price',
                        start_date=start, end_date=end)

# We have to take the percent changes to get to returns
# Get rid of the first (0th) element because it is NAN
r_a = asset.pct_change()[1:]
r_b = benchmark.pct_change()[1:]

seaborn.regplot(r_b.values, r_a.values);
```

95% Confidence Intervals



Standard Errors and Prediction Interval

$$s = \sqrt{\frac{\sum_{i=1}^{n} \epsilon_i^2}{n-2}}$$

$$s_f^2 = s^2\left(1 + \frac{1}{n} + \frac{(X-\mu_x)^2}{(n-1)\sigma_x^2}\right)$$

$$\hat{Y} \pm t_c s_f$$