11) Multiple Linear Regression

Vitor Kamada

December 2018

Reference

Tables, Graphics, and Figures from

https://www.quantopian.com/lectures

Lecture 15 Multiple Linear Regression

Multiple Linear Regression

$$Y_i = \alpha + \beta_1 X_{1i} + ... + \beta_k X_{ki} + \epsilon_i$$

```
import numpy as np
import pandas as pd
import statsmodels.api as sm

from statsmodels import regression
import matplotlib.pyplot as plt
```

$\sum_{i=1}^{n} e_i^2$

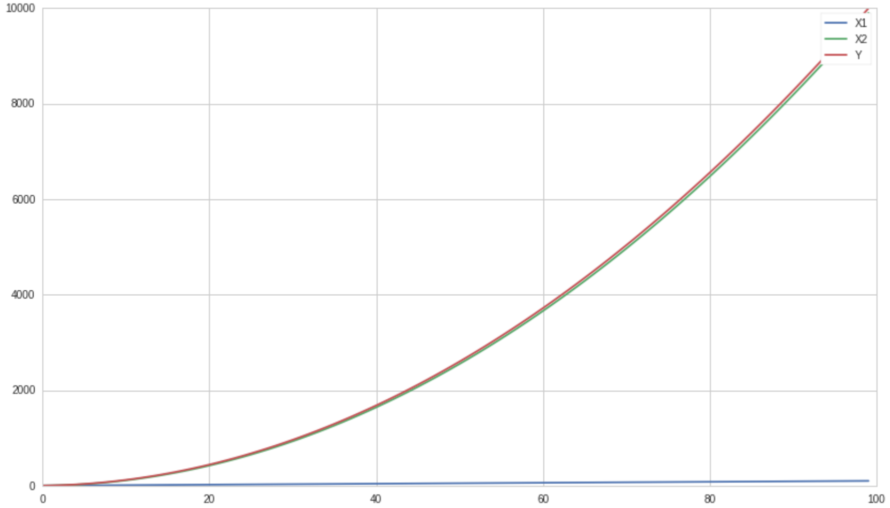
```python
Y = np.array([1, 3.5, 4, 8, 12])
Y_hat = np.array([1, 3, 5, 7, 9])

print 'Error ' + str(Y_hat - Y)

# Compute squared error
SE = (Y_hat - Y) ** 2

print 'Squared Error ' + str(SE)
print 'Sum Squared Error ' + str(np.sum(SE))
```

```
Error [ 0.  -0.5  1.  -1.  -3. ]
Squared Error [ 0.    0.25  1.    1.    9.  ]
Sum Squared Error 11.25
```

Toy Example

```
# Construct a simple linear curve of 1, 2, 3, ...
X1 = np.arange(100)

# Make a parabola and add X1 to it, this is X2
X2 = np.array([i ** 2 for i in range(100)]) + X1

# This is our real Y, constructed using a
# linear combination of X1 and X2
Y = X1 + X2

plt.plot(X1, label='X1')
plt.plot(X2, label='X2')
plt.plot(Y, label='Y')
plt.legend();
```

$Y = X_1 + X_2$



$\hat{Y} = 0 + \hat{X}_1 + \hat{X}_2$

```
X = sm.add_constant( np.column_stack( (X1, X2) ) )

# Run the model
results = regression.linear_model.OLS(Y, X).fit()

print 'Beta_0:', results.params[0]
print 'Beta_1:', results.params[1]
print 'Beta_2:', results.params[2]
```

```
Beta_0: 1.36424205266e-12
Beta_1: 1.0
Beta_2: 1.0
```

$$Y = X_1 + X_2$$

$$Y = X_1 + X^2 + X_1$$

$$Y = 2X_1 + X^2$$

$MSFT = \alpha + \beta_1 INTC + \epsilon$

```
start = '2014-01-01'
end = '2015-01-01'
asset1 = get_pricing('MSFT', fields='price',
                     start_date=start, end_date=end)
asset2 = get_pricing('INTC', fields='price',
                     start_date=start, end_date=end)
benchmark = get_pricing('SPY', fields='price',
                        start_date=start, end_date=end)

# First, run a linear regression on the two assets
slr = regression.linear_model.OLS(asset1,
                        sm.add_constant(asset2)).fit()
print 'SLR beta of asset2:', slr.params[1]
```

```
SLR beta of asset2: 0.856395904276
```

$\hat{MSFT} = \hat{\alpha}_m + 0.48INTC + 0.21SPY$

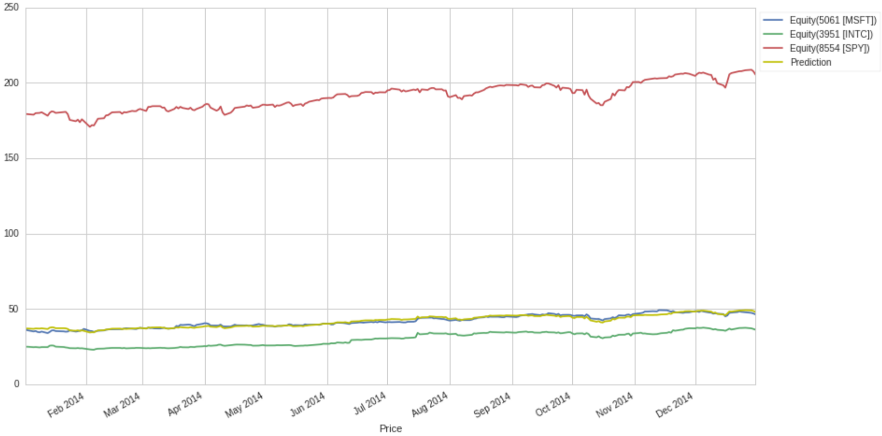
```
mlr = regression.linear_model.OLS(asset1,
     sm.add_constant(np.column_stack((asset2,
                              benchmark)))).fit()

prediction=mlr.params[0] + mlr.params[1]*asset2 \
             + mlr.params[2]*benchmark
prediction.name = 'Prediction'

print 'MLR beta of asset2:', mlr.params[1], \
  '\nMLR beta of S&P 500:', mlr.params[2]
```

```
MLR beta of asset2: 0.480204841524
MLR beta of S&P 500: 0.210495422398
```

$$\hat{MSFT} = \hat{\alpha}_s + 0.85INTC$$

3 Variables and MLR Prediction

```
asset1.plot()
asset2.plot()
benchmark.plot()
prediction.plot(color='y')
plt.xlabel('Price')
plt.legend(bbox_to_anchor=(1,1), loc=2);
```



Dependent Variable and Prediction

```
asset1.plot()
prediction.plot(color='y')
plt.xlabel('Price')
plt.legend();
```

mlr.summary()

Dep. Variable:	Equity(5061 [MSFT])	**R-squared:**	0.923
Model:	OLS	**Adj. R-squared:**	0.922
Method:	Least Squares	**F-statistic:**	1484.
Date:	Tue, 21 Aug 2018	**Prob (F-statistic):**	4.30e-139
Time:	17:15:21	**Log-Likelihood:**	-394.38
No. Observations:	252	**AIC:**	794.8
Df Residuals:	249	**BIC:**	805.4
Df Model:	2		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[95.0% Conf. Int.]**
const	-12.6540	3.140	-4.030	0.000	-18.838 -6.470
x1	0.4802	0.043	11.068	0.000	0.395 0.566
x2	0.2105	0.023	9.324	0.000	0.166 0.255

R^2 and *Adjusted* R^2

$$R^2 = \frac{TSS-RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

$$RSS = \sum_{i=1}^{n} e_i^2$$

$$Adjusted\,R^2 = 1 - (1 - R^2)\frac{n-1}{n-k-1}$$

Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and *Adjusted* R^2

$$AIC = \frac{1}{n\hat{\sigma}^2}(RSS + 2k\hat{\sigma}^2)$$

$$BIC = \frac{1}{n\hat{\sigma}^2}(RSS + log(n)k\hat{\sigma}^2)$$

$$\text{Adjusted } R^2 = 1 - \frac{RSS/(n-k-1)}{TSS/(n-1)}$$