14) Residual Analysis: Autocorrelation

Vitor Kamada

July 2018

Reference

Tables, Graphics, and Figures from

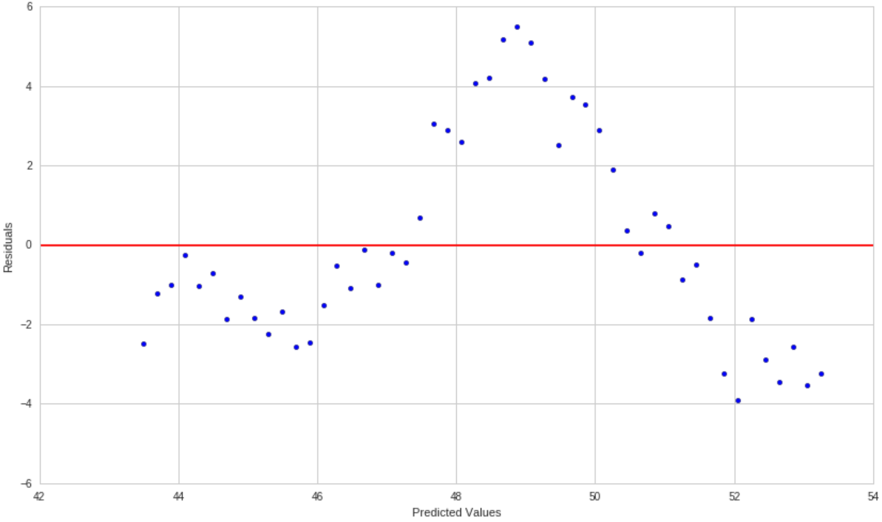
https://www.quantopian.com/lectures

Lecture 18 Residual Analysis

$Y_i = Y_{i-1} + \epsilon$

```
n = 50
X = np.linspace(0, n, n)
Y_autocorrelated = np.zeros(n)
Y_autocorrelated[0] = 50
for t in range(1, n):
    Y_autocorrelated[t] = Y_autocorrelated[t-1]\
                        + np.random.normal(0, 1)

# Regressing X and Y_autocorrelated
model = sm.OLS(Y_autocorrelated,
                   sm.add_constant(X)).fit()
B0, B1 = model.params
residuals = model.resid

plt.scatter(model.predict(), residuals);
plt.axhline(0, color='red')
plt.xlabel('Predicted Values');
plt.ylabel('Residuals');
```

Autocorrelation in the Residuals



Sample Autocorrelations

$$\rho(\tau) = \frac{E[(y_t - \mu)(y_{t-\tau} - \mu)]}{E[(y_t - \mu)^2]}$$

$$\hat{\rho}(\tau) = \frac{\sum_{t=\tau+1}^{T} [(y_t - \bar{y})(y_{t-\tau} - \bar{y})]}{\sum_{t=1}^{T} (y_t - \bar{y})^2}$$

$$\hat{\rho}(\tau) \sim N(0, \tfrac{1}{T})$$

$$\sqrt{T}\hat{\rho}(\tau) \sim N(0, 1)$$

$$T\hat{\rho}^2(\tau) \sim \chi_1^2$$

Box-Pierce Q-statistic

$$Q_{BP} = T \sum_{\tau=1}^{m} \hat{\rho}^2(\tau)$$

$$Q_{LB} = T(T+2) \sum_{\tau=1}^{m} \left(\frac{1}{T-\tau}\right)\hat{\rho}^2(\tau)$$

H_0 : The data are independently distributed

H_a : The data exhibit serial correlation

Ljung-Box Test

```
ljung_box = smd.acorr_ljungbox(residuals, lags = 10)
print "Lagrange Multiplier Statistics:", ljung_box[0]
print "\nP-values:", ljung_box[1], "\n"

if any(ljung_box[1] < 0.05):
    print "The residuals are autocorrelated."
else:
    print "The residuals are not autocorrelated."
```

```
Lagrange Multiplier Statistics: [  43.65325348   80.80728237  112.66873613  138.1
157.50322113
  171.78472133  179.18420508  181.49990291  181.72987791  181.8555585 ]

P-values: [  3.92024856e-11   2.83740666e-18   2.92375611e-24   7.05507263e-29
   3.36983459e-32   1.88110240e-34   2.89663288e-35   4.98447013e-35
   2.20529138e-34   9.64841145e-34]

The residuals are autocorrelated.
```

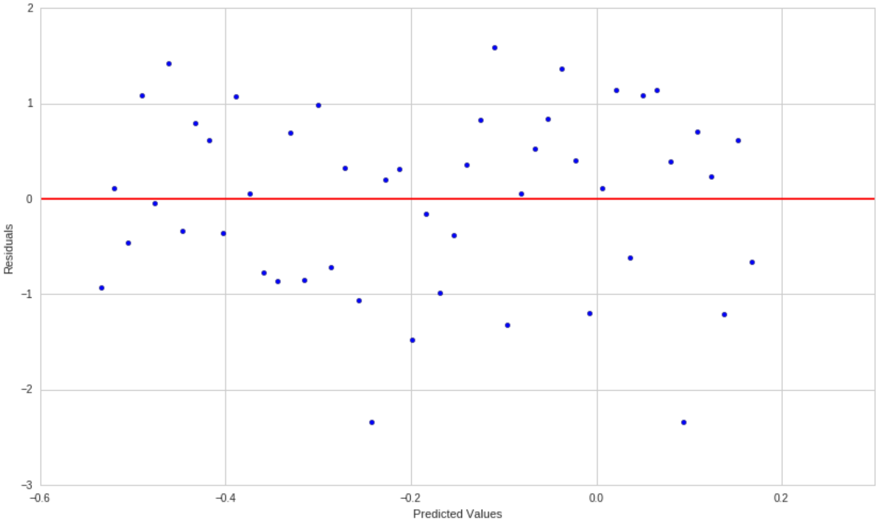
First-Order Differences

```
Y_autocorrelated_diff = np.diff(Y_autocorrelated)
```

```
model = sm.OLS(Y_autocorrelated_diff, sm.add_constant(X[1:])).fit()
B0, B1 = model.params
residuals = model.resid

plt.scatter(model.predict(), residuals);
plt.axhline(0, color='red')
plt.xlabel('Predicted Values');
plt.ylabel('Residuals');

# Running and interpreting a Ljung-Box test
ljung_box = smd.acorr_ljungbox(residuals, lags = 10)
print "P-values:", ljung_box[1], "\n"

if any(ljung_box[1] < 0.05):
    print "The residuals are autocorrelated."
else:
    print "The residuals are not autocorrelated."
```

```
P-values: [ 0.46043772  0.74908377  0.82067765  0.92091356  0.96659539  0.91694346
  0.8644615   0.78746052  0.63628018  0.62962119]

The residuals are not autocorrelated.
```

Residuals Are not Autocorrelated



Get Data

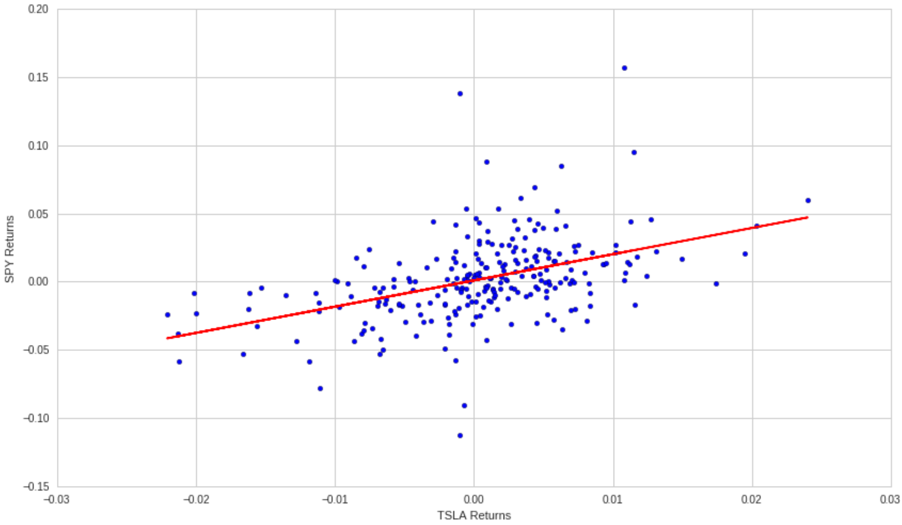
```
start = '2014-01-01'
end = '2015-01-01'
asset = get_pricing('TSLA', fields='price',
                    start_date=start, end_date=end)
benchmark = get_pricing('SPY', fields='price',
                    start_date=start, end_date=end)

# 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:].values
r_b = benchmark.pct_change()[1:].values

# Regressing the benchmark b and asset a
r_b = sm.add_constant(r_b)
model = sm.OLS(r_a, r_b).fit()
r_b = r_b[:, 1]
B0, B1 = model.params
```

Market Beta Calculation

```
A_hat = (B1*r_b + B0)
plt.scatter(r_b, r_a, alpha=1) # Plot the raw data
# Add the regression line, colored in red
plt.plot(r_b, A_hat, 'r', alpha=1);
plt.xlabel('TSLA Returns')
plt.ylabel('SPY Returns')

# Print our result
print "Estimated TSLA Beta:", B1

# Calculating the residuals
residuals = model.resid
```

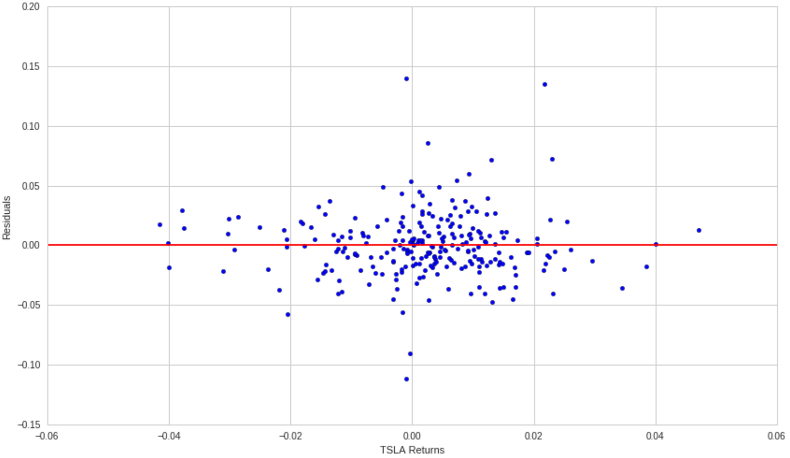
```
Estimated TSLA Beta: 1.92533467685
```

$\hat{Y} = \hat{\alpha} + 1.92X$



Residual Analysis

```
plt.scatter(model.predict(), residuals);
plt.axhline(0, color='red')
plt.xlabel('TSLA Returns');
plt.ylabel('Residuals');
```



Breusch-Pagan Heteroscedasticity Test

```
bp_test = smd.het_breushpagan(residuals, model.model.exog)

print "Lagrange Multiplier Statistic:", bp_test[0]
print "P-value:", bp_test[1]
print "f-value:", bp_test[2]
print "f_p-value:", bp_test[3], "\n"
if bp_test[1] > 0.05:
    print "The relationship is not heteroscedastic."
if bp_test[1] < 0.05:
    print "The relationship is heteroscedastic."
```

```
Lagrange Multiplier Statistic: 0.669337376498
P-value: 0.413282723143
f-value: 0.665779433495
f_p-value: 0.415306831916

The relationship is not heteroscedastic.
```

Ljung-Box Autocorrelation Test

```
ljung_box = smd.acorr_ljungbox(r_a)
print "P-Values:", ljung_box[1], "\n"
if any(ljung_box[1] < 0.05):
    print "The residuals are autocorrelated."
else:
    print "The residuals are not autocorrelated."
```

```
P-Values: [ 0.8846583   0.88950844  0.96229443  0.96341497  0.91599599  0.9303213
  0.9643816   0.97852899  0.98390172  0.98786945  0.99167638  0.97134708
  0.91203802  0.9216252   0.94242703  0.87812148  0.90007513  0.92664875
  0.94471082  0.88594594  0.88744682  0.91583141  0.8960177   0.92045423
  0.87780239  0.89866989  0.91536025  0.93228388  0.93825939  0.95373621
  0.91155827  0.9313345   0.94201011  0.94355971  0.94414366  0.95678029
  0.96694651  0.96684993  0.95284072  0.96254994]

The residuals are not autocorrelated.
```