scipy linregress function erroneous standard error return?

I have a weird situation with scipy.stats.linregress seems to be returning an incorrect standard error:

```
from scipy import stats
x = [5.05, 6.75, 3.21, 2.66]
y = [1.65, 26.5, -5.93, 7.96]
gradient, intercept, r_value, p_value, std_err = stats.linregress(x,y)
>>> gradient
5.3935773611970186
>>> intercept
-16.281127993087829
>>> r_value
0.72443514211849758
>>> r_value**2
0.52480627513624778
>>> std_err
3.6290901222878866
```

Whereas Excel returns the following:

```
slope: 5.394

intercept: -16.281

rsq: 0.525

steyX: 11.696
```

steyX is excel's standard error function, returning 11.696 versus scipy's 3.63. Anybody know what's going on here? Any alternative way of getting the standard error of a regression in python, *without going to Rpy*?

python scipy regression

edited Jul 16 '15 at 13:17 — Gabriel — 4,807 12 47 114

asked Jan 10 '10 at 21:19 — Thomas Browne — 4,170 13 43 65

2 Answers

You could try the statsmodels package:

```
In [37]: import statsmodels.api as sm

In [38]: x = [5.05, 6.75, 3.21, 2.66]

In [39]: y = [1.65, 26.5, -5.93, 7.96]

In [40]: X = sm.add_constant(x) # intercept

In [41]: model = sm.OLS(y, X)

In [42]: fit = model.fit()

In [43]: fit.params
Out[43]: array([  5.39357736, -16.28112799])

In [44]: fit.rsquared
Out[44]: 0.52480627513624789

In [45]: np.sqrt(fit.mse_resid)
Out[45]: 11.696414461570097
```

edited Jun 27 '12 at 23:19 — answered Jan 11 '10 at 5:36

bmu **12.5k** 2 50 79 ars **56.8k** 16 95 108

Brilliant. Thanks ars. Exactly what I needed. – Thomas Browne Jan 11 '10 at 20:07

Glad to help. :) – ars Jan 11 '10 at 22:36

I've just been informed by the SciPy user group that the std_err here represents the standard error of the gradient line, not the standard error of the predicted y's, as per Excel. Nevertheless users of this function should be careful, because this was not always the behaviour of this library - it used to output exactly as Excel, and the changeover appears to have occurred in the past few months.

Anyway still looking for an equivalent to STEYX in Python.

answered Jan 10 '10 at 22:15

Thomas Browne **4,170** 13 43 65