`scipy.stats.linregress(x, y=None)`

Calculate a regression line

This computes a least-squares regression for two sets of measurements.

Parameters: x, y : array_like two sets of measurements. Both arrays should have the same length. If only x is given (and y=None), then it must be a two-dimensional array where one dimension has length 2. The two sets of measurements are then found by splitting the array along the length-2 dimension.

Returns:

- slope : float slope of the regression line
- intercept : float intercept of the regression line
- r-value : float correlation coefficient
- `p-value` : float two-sided p-value for a hypothesis test whose null hypothesis is that the slope is zero.
- `stderr` : float Standard error of the estimate

Examples

```
>>> from scipy import stats
>>> import numpy as np
>>> x = np.random.random(10)
>>> y = np.random.random(10)
>>> slope, intercept, r_value, p_value, std_err = stats.linregress(x,y)
```

To get coefficient of determination (`r_squared`)

```
>>>
>>> print "r-squared:", r_value**2
$ r-squared: 0.15286643777 sis test whose null hypothesis is that the slope is zero.
\item stderr : float
Standard error of the estimate

\subsubsection*{Examples}

\begin{framed}
\begin{verbatim}
>>>
```

```
>>> from scipy import stats
>>> import numpy as np
>>> x = np.random.random(10)
>>> y = np.random.random(10)
>>> slope, intercept, r_value, p_value, std_err = stats.linregress(x,y)
# To get coefficient of determination (r_squared)

>>> $
>>> print "r-squared:", r_value**2
r-squared: 0.15286643777
```