Pearson correlation coefficient 2-tailed p-value meaning [closed]

from the sciPy library I used: scipy.stats.stats import pearsonr to calculate the correlation coefficient for two arrays and I got a value of: (0.80751532276005755, 0.19248467723994242).

I thought that I would have just got one value within the range -1 to +1, so I'm unsure how to interpret these two results. These are my two arrays:

```
x = [50,500,1500,2500]
y = [17, 6, 6, 194]
```

and I did:

```
pearsonr(x,y)
```

Thanks

python math numpy statistics scipy

asked Nov 30 '12 at 22:13

user94628
768 1 15 35

closed as off topic by talonmies, Rohan, 一二三, hims056, tibtof Dec 1 '12 at 11:37

Questions on Stack Overflow are expected to relate to programming within the scope defined by the community. Consider editing the question or leaving comments for improvement if you believe the question can be reworded to fit within the scope. Read more about reopening questions here.

If this question can be reworded to fit the rules in the help center, please edit the question.

Did you try reading the documentation for that function? – Femaref Nov 30 '12 at 22:14

Yes but I'm not entirely sure about the meaning of a 2-tailed p-value. So the first value is the correlation and the second value the probability of an uncorrealted set producing the same result. – user94628 Nov 30 '12 at 22:18

1 This should, now , be moved to the stats forum. It is a basic stats question. – AGS Nov 30 '12 at 22:58

1 Answer

`pearsonr()` returns a two-tuple consisting of the correlation coefficient and the corresponding p-value:

- The correlation coefficient can range from -1 to +1.
- The null hypothesis is that the two variables are uncorrelated. The p-value is a number between zero and one that represents the probability that your data would have arisen if the null hypothesis were true.

For a further discussion, see http://www.eecs.qmul.ac.uk/~norman/blog_articles/p_values.pdf

> I thought that I would have just got one value within the range -1 to +1

If you just need to the correlation coefficient, simply ignore the second element of the tuple (the p-value):

```
corrxy = pearsonr(x,y)[0]
```

It might be worth mentioning that there's also `numpy.corrcoef()`, which computes the correlation matrix (without p-values).

edited Dec 1 '12 at 12:28

answered Nov 30 '12 at 22:57

NPE

229k 31 484 675

I'll also take a look at the numpy method. Thanks for the explanation. – user94628 Dec 1 '12 at 0:05