Welcome back! If you found this question useful, don't forget to vote both the question and the answers up.

Using SciPy T.ppf to get p-value

Asked 4 years, 5 months ago Active 9 months ago Viewed 13k times



Trying to test code creating P-value manually against SciPy. The <u>Scipy Documentation</u> isn't the best, which makes it tought to know for sure what to do.



I am getting the correct t-stat and P-value with SciPy, but I'm not able to replicate the correct p-value manually - A friend steered me to <code>scipy.stats.t.ppf</code> - but I'm not getting a p-value from it.



What is the correct way to do scipy.stats.t.ppf()?

my version:

```
def t_test(sample, mu):
    mean = np.mean(sample)
    var = np.var(sample)
    sem = (var / len(sample)) ** .5
    t = abs(mu - mean)/sem
    df = len(sample) - 1
    p = scs.t.ppf(.95, df)
    return (t, p)
```

returns (0.081500599630942958, 1.7291328115213671)

scipy version:

```
scs.ttest_1samp(sample, 4.123)
returns (statistic=0.079436958358141435, pvalue=0.93751577779749051)
```

for testing, I'm using the following sample set and sample mu.

```
hypothesis-testing python scipy
```

Share Cite Edit Follow Flag

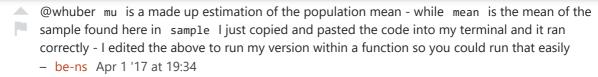
```
edited Dec 2 '20 at 13:28
reox
241 2 10
```

asked Apr 1 '17 at 18:10



Welcome back! If you found this question useful, don't forget to vote both the question and the answers up.

close this message



1 Answer

Active	Oldest	Votes
--------	--------	-------

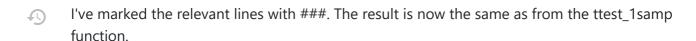


To get the same results, change two things:

- 2
- 1. Change the estimation of the variance such that the divisor is N-1



2. Calculate the p-value using the cdf, that is the probability of getting a more extreme value, here using that the t-distribution is symmetric around zero. Note that the function you're comparing with does a two-sided test, and therefore, so do I.



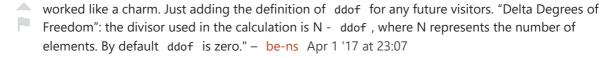
```
def t_test(sample, mu):
  mean = np.mean(sample)
  var = np.var(sample, ddof = 1) ###
  sem = (var / len(sample)) ** .5
  t = abs(mu - mean)/sem
  df = len(sample) - 1
  p = 2*(1-scs.t.cdf(t, df)) ###
  return (t, p)
```

Share Cite Edit Follow Flag

answered Apr 1 '17 at 20:03



12 19





@reox: That would be correct if 't' were the signed t-statistic, but note that in swmo's code, it is the absolute value. This ensures that we are always on the right side of the distribution.
 Ruben van Bergen Dec 2 '20 at 13:48

@RubenvanBergen ah yes you are right, I missed the abs. – reox Dec 3 '20 at 14:18