SciPy.org Sponsored By ENTHOUGHT (../index.html)

Scipy.org (http://scipy.org/) Docs (http://docs.scipy.org/)

SciPy v0.15.1 Reference Guide (../index.html) Statistical functions (**`scipy.stats`**) (../stats.html)

index (../genindex.html) modules (../py-modindex.html) next (scipy.stats.linregress.html)

previous (scipy.stats.pointbiserialr.html)

scipy.stats.kendalltau

scipy.stats.kendalltau(*x*, *y*, *initial_lexsort=True***)** **[source] (http://github.com/scipy/scipy/blob/v0.15.1/scipy/stats/stats.py#L2827)**

Calculates Kendall’s tau, a correlation measure for ordinal data.

Kendall’s tau is a measure of the correspondence between two rankings. Values close to 1 indicate strong agreement, values close to -1 indicate strong disagreement. This is the tau-b version of Kendall’s tau which accounts for ties.

Parameters:

x, y : *array_like*

Arrays of rankings, of the same shape. If arrays are not 1-D, they will be flattened to 1-D.

initial_lexsort : *bool, optional*

Whether to use lexsort or quicksort as the sorting method for the initial sort of the inputs. Default is lexsort (True), for which kendalltau is of complexity O(n log(n)). If False, the complexity is O(n^2), but with a smaller pre-factor (so quicksort may be faster for small arrays).

Returns:

Kendall’s tau : *float*

The tau statistic.

p-value : *float*

The two-sided p-value for a hypothesis test whose null hypothesis is an absence of association, tau = 0.

Notes

The definition of Kendall’s tau that is used is:

```
tau = (P - Q) / sqrt((P + Q + T) * (P + Q + U))
```

where P is the number of concordant pairs, Q the number of discordant pairs, T the number of ties only in *x*, and U the number of ties only in *y*. If a tie occurs for the same pair in both *x* and *y*, it is not added to either T or U.

References

W.R. Knight, “A Computer Method for Calculating Kendall’s Tau with Ungrouped Data”, Journal of the American Statistical Association, Vol. 61, No. 314, Part 1, pp. 436-439, 1966.

Examples

>>>

```
>>> import scipy.stats as stats
>>> x1 = [12, 2, 1, 12, 2]
>>> x2 = [1, 4, 7, 1, 0]
>>> tau, p_value = stats.kendalltau(x1, x2)
>>> tau
-0.47140452079103173
>>> p_value
0.24821309157521476
```

Previous topic

scipy.stats.pointbiserialr (scipy.stats.pointbiserialr.html)

Next topic

scipy.stats.linregress (scipy.stats.linregress.html)