scipy.stats.kendalltau

scipy.stats.**kendalltau**(*x*, *y*, *initial_lexsort=None*, *nan_policy='propagate'*, *method='auto'*)
(https://github.com/scipy/scipy/blob/v1.4.1/scipy/stats/stats.py#L3974-L4169) [source]

Calculate 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 1945 "tau-b" version of Kendall's tau [2], which can account for ties and which reduces to the 1938 "tau-a" version [1] in absence of 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*

Unused (deprecated).

nan_policy : *{'propagate', 'raise', 'omit'}, optional*

Defines how to handle when input contains nan. The following options are available (default is 'propagate'):

- 'propagate': returns nan
- 'raise': throws an error
- 'omit': performs the calculations ignoring nan values

method : *{'auto', 'asymptotic', 'exact'}, optional*

Defines which method is used to calculate the p-value [5]. The following options are available (default is 'auto'):

- 'auto': selects the appropriate method based on a trade-off between speed and accuracy
- 'asymptotic': uses a normal approximation valid for large samples
- 'exact': computes the exact p-value, but can only be used if no ties are present

Returns:

correlation : *float*

The tau statistic.

pvalue : *float*

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

See also:

spearmanr (scipy.stats.spearmanr.html#scipy.stats.spearmanr) Calculates a Spearman rank-order correlation coefficient.

theilslopes (scipy.stats.theilslopes.html#scipy.stats.theilslopes) Computes the Theil-Sen estimator for a set of points (x, y).

weightedtau (scipy.stats.weightedtau.html#scipy.stats.weightedtau) Computes a weighted version of Kendall's tau.

Notes

The definition of Kendall's tau that is used is [2]:

```
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

[1] Maurice G. Kendall, "A New Measure of Rank Correlation", Biometrika Vol. 30, No. 1/2, pp. 81-93, 1938.

2(1,2) Maurice G. Kendall, "The treatment of ties in ranking problems", Biometrika Vol. 33, No. 3, pp. 239-251. 1945.

3 Gottfried E. Noether, "Elements of Nonparametric Statistics", John Wiley & Sons, 1967.

4 Peter M. Fenwick, "A new data structure for cumulative frequency tables", Software: Practice and Experience, Vol. 24, No. 3, pp. 327-336, 1994.

[5] Maurice G. Kendall, "Rank Correlation Methods" (4th Edition), Charles Griffin & Co., 1970.

Examples

```
>>> from scipy import 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
0.2827454599327748
```


Previous topic

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

Next topic

scipy.stats.weightedtau (scipy.stats.weightedtau.html)

Quick search

search