HydroErr Documentation

Wade Roberts, Gustavious Williams

Apr 23, 2019

Contents:

CHAPTER 1

Installation

HydroErr is freely available on the Python Package index repository (PyPI). It can be installed with the following command using either virtualenv or Anaconda:

```
pip install HydroErr
```

The extension is also available through conda package management system. It can be installed with:

```
conda install -c conda-forge hydroerr
```

CHAPTER 2

List of Metrics of Hydrologic Skill

2.1 HydroErr.HydroErr Module

HydroErr contains a library of goodness of fit metrics that measure hydrologic skill. Each metric is contained in function, and every function has the parameters to treat missing values as well as remove zero and negative values from the timeseries data.

Each function contains two properties, name and abbr. These can be used in the Hydrostats package when creating tables and adding metrics to the plots. Link to the hydrostats package: https://github.com/BYU-Hydroinformatics/Hydrostats. An example of this functionality is shown below.

```
>>> import HydroErr as he
>>>
>>> he.acc.name
'Anomaly Correlation Coefficient'
>>> he.acc.abbr
'ACC'
```

2.1.1 Functions

me(simulated_array, observed_array[, ...])	Compute the mean error of the simulated and observed data.
mae(simulated_array, observed_array[, ...])	Compute the mean absolute error of the simulated and observed data.
mse(simulated_array, observed_array[, ...])	Compute the mean squared error of the simulated and observed data.
mle(simulated_array, observed_array[, ...])	Compute the mean log error of the simulated and observed data.
male(simulated_array, observed_array[, ...])	Compute the mean absolute log error of the simulated and observed data.

Continued on next page

Table 1 – continued from previous page

msle(simulated_array, observed_array[, ...])	Compute the mean squared log error of the simulated and observed data.
mde(simulated_array, observed_array[, ...])	Compute the median error (MdE) between the simulated and observed data.
mdae(simulated_array, observed_array[, ...])	Compute the median absolute error (MdAE) between the simulated and observed data.
mdse(simulated_array, observed_array[, ...])	Compute the median squared error (MdSE) between the simulated and observed data.
ed(simulated_array, observed_array[, ...])	Compute the Euclidean distance between predicted and observed values in vector space.
ned(simulated_array, observed_array[, ...])	Compute the normalized Euclidian distance between the simulated and observed data in vector space.
rmse(simulated_array, observed_array[, ...])	Compute the root mean square error between the simulated and observed data.
rmsle(simulated_array, observed_array[, ...])	Compute the root mean square log error between the simulated and observed data.
nrmse_range(simulated_array, observed_array)	Compute the range normalized root mean square error between the simulated and observed data.
nrmse_mean(simulated_array, observed_array)	Compute the mean normalized root mean square error between the simulated and observed data.
nrmse_iqr(simulated_array, observed_array[, ...])	Compute the IQR normalized root mean square error between the simulated and observed data.
irmse(simulated_array, observed_array[, ...])	Compute the inertial root mean square error (IRMSE) between the simulated and observed data.
mase(simulated_array, observed_array[, m, ...])	Compute the mean absolute scaled error between the simulated and observed data.
r_squared(simulated_array, observed_array[, ...])	Compute the the Coefficient of Determination (r2).
pearson_r(simulated_array, observed_array[, ...])	Compute the pearson correlation coefficient.
spearman_r(simulated_array, observed_array)	Compute the spearman rank correlation coefficient.
acc(simulated_array, observed_array[, ...])	Compute the the anomaly correlation coefficient (ACC).
mape(simulated_array, observed_array[, ...])	Compute the the mean absolute percentage error (MAPE).
mapd(simulated_array, observed_array[, ...])	Compute the the mean absolute percentage deviation (MAPD).
maape(simulated_array, observed_array[, ...])	Compute the the Mean Arctangent Absolute Percentage Error (MAAPE).
smape1(simulated_array, observed_array[, ...])	Compute the the Symmetric Mean Absolute Percentage Error (1) (SMAPE1).
smape2(simulated_array, observed_array[, ...])	Compute the the Symmetric Mean Absolute Percentage Error (2) (SMAPE2).
d(simulated_array, observed_array[, ...])	Compute the the index of agreement (d).
d1(simulated_array, observed_array[, ...])	Compute the the index of agreement (d1).
dmod(simulated_array, observed_array[, j, ...])	Compute the the modified index of agreement (dmod).
drel(simulated_array, observed_array[, ...])	Compute the the relative index of agreement (drel).
dr(simulated_array, observed_array[, ...])	Compute the the refined index of agreement (dr).
watt_m(simulated_array, observed_array[, ...])	Compute Watterson's M (M).
mb_r(simulated_array, observed_array[, ...])	Compute Mielke-Berry R value (MB R).
nse(simulated_array, observed_array[, ...])	Compute the Nash-Sutcliffe Efficiency.
nse_mod(simulated_array, observed_array[, ...])	Compute the modified Nash-Sutcliffe efficiency (NSE mod).

Continued on next page

Table 1 – continued from previous page

nse_rel(simulated_array, observed_array[, ...])	Compute the relative Nash-Sutcliffe efficiency (NSE rel).
kge_2009(simulated_array, observed_array[, ...])	Compute the Kling-Gupta efficiency (2009).
kge_2012(simulated_array, observed_array[, ...])	Compute the Kling-Gupta efficiency (2012).
lm_index(simulated_array, observed_array[, ...])	Compute the Legate-McCabe Efficiency Index.
d1_p(simulated_array, observed_array[, ...])	Compute the Legate-McCabe Index of Agreement.
ve(simulated_array, observed_array[, ...])	Compute the Volumetric Efficiency (VE).
sa(simulated_array, observed_array[, ...])	Compute the Spectral Angle (SA).
sc(simulated_array, observed_array[, ...])	Compute the Spectral Correlation (SC).
sid(simulated_array, observed_array[, ...])	Compute the Spectral Information Divergence (SID).
sga(simulated_array, observed_array[, ...])	Compute the Spectral Gradient Angle (SGA).
h1_mhe(simulated_array, observed_array[, ...])	Compute the H1 mean error.
h1_mahe(simulated_array, observed_array[, ...])	Compute the H1 absolute error.
h1_rmshe(simulated_array, observed_array[, ...])	Compute the H1 root mean square error.
h2_mhe(simulated_array, observed_array[, ...])	Compute the H2 mean error.
h2_mahe(simulated_array, observed_array[, ...])	Compute the H2 mean absolute error.
h2_rmshe(simulated_array, observed_array[, ...])	Compute the H2 root mean square error.
h3_mhe(simulated_array, observed_array[, ...])	Compute the H3 mean error.
h3_mahe(simulated_array, observed_array[, ...])	Compute the H3 mean absolute error.
h3_rmshe(simulated_array, observed_array[, ...])	Compute the H3 root mean square error.
h4_mhe(simulated_array, observed_array[, ...])	Compute the H4 mean error.
h4_mahe(simulated_array, observed_array[, ...])	Compute the H4 mean absolute error.
h4_rmshe(simulated_array, observed_array[, ...])	Compute the H4 mean error.
h5_mhe(simulated_array, observed_array[, ...])	Compute the H5 mean error.
h5_mahe(simulated_array, observed_array[, ...])	Compute the H5 mean absolute error.
h5_rmshe(simulated_array, observed_array[, ...])	Compute the H5 root mean square error.
h6_mhe(simulated_array, observed_array[, k, ...])	Compute the H6 mean error.
h6_mahe(simulated_array, observed_array[, ...])	Compute the H6 mean absolute error.
h6_rmshe(simulated_array, observed_array[, ...])	Compute the H6 root mean square error.
h7_mhe(simulated_array, observed_array[, ...])	Compute the H7 mean error.
h7_mahe(simulated_array, observed_array[, ...])	Compute the H7 mean absolute error.
h7_rmshe(simulated_array, observed_array[, ...])	Compute the H7 root mean square error.
h8_mhe(simulated_array, observed_array[, ...])	Compute the H8 mean error.
h8_mahe(simulated_array, observed_array[, ...])	Compute the H8 mean absolute error.
h8_rmshe(simulated_array, observed_array[, ...])	Compute the H8 root mean square error.
h10_mhe(simulated_array, observed_array[, ...])	Compute the H10 mean error.
h10_mahe(simulated_array, observed_array[, ...])	Compute the H10 mean absolute error.
h10_rmshe(simulated_array, observed_array[, ...])	Compute the H10 root mean square error.
g_mean_diff(simulated_array, observed_array)	Compute the geometric mean difference.
mean_var(simulated_array, observed_array[, ...])	Compute the mean variance.

me

HydroErr.HydroErr.**me**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the mean error of the simulated and observed data.

$$\mathrm{ME} = \frac{1}{n}\sum_{i=0}^{n}(S_i - O_i)$$

Range: -inf < MAE < inf, data units, closer to zero is better, indicates bias.

Notes: The mean error (ME) measures the difference between the simulated data and the observed data. For the mean error, a smaller number indicates a better fit to the original data. Note that if the error is in the form of random noise, the mean error will be very small, which can skew the accuracy of this metric. ME is cumulative and will be small even if there are large positive and negative errors that balance.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean error value.

Return type float

Examples

Note that in this example the random noise cancels, leaving a very small ME.

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> # Seed for reproducibility
>>> np.random.seed(54839)
```

```
>>> x = np.arange(100) / 20
>>> sim = np.sin(x) + 2
>>> obs = sim * (((np.random.rand(100) - 0.5) / 10) + 1)
>>> he.me(sim, obs)
-0.006832220968967168
```

References

- Fisher, R.A., 1920. A Mathematical Examination of the Methods of Determining the Accuracy of an Observation by the Mean Error, and by the Mean Square Error. Monthly Notices of the Royal Astronomical Society 80 758 - 770.

mae

HydroErr.HydroErr.**mae**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the mean absolute error of the simulated and observed data.

$$MAE = \frac{1}{n}\sum_{i=0}^{n}|S_i - O_i|$$

Range: 0 MAE < inf, data units, smaller is better.

Notes: The ME measures the absolute difference between the simulated data and the observed data. For the mean abolute error, a smaller number indicates a better fit to the original data. Also note that random errors do not cancel. Also referred to as an L1-norm.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean absolute error value.

Return type float

References

- Willmott, Cort J., and Kenji Matsuura. “Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in Assessing Average Model Performance.” Climate Research 30, no. 1 (2005): 79–82.
- Willmott, Cort J., and Kenji Matsuura. “On the Use of Dimensioned Measures of Error to Evaluate the Performance of Spatial Interpolators.” International Journal of Geographical Information Science 20, no. 1 (2006): 89–102.

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 6.8])
>>> he.mae(sim, obs)
0.5666666666666665
```

mse

`HydroErr.HydroErr.mse`(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the mean squared error of the simulated and observed data.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(S_i - O_i)^2$$

Range: 0 MSE < inf, data units squared, smaller is better.

Notes: Random errors do not cancel, highlights larger errors, also referred to as a squared L2-norm.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean squared error value.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 6.8])
>>> he.mse(sim, obs)
0.4333333333333333
```

References

- Wang, Zhou, and Alan C. Bovik. "Mean Squared Error: Love It or Leave It? A New Look at Signal Fidelity Measures." IEEE Signal Processing Magazine 26, no. 1 (2009): 98–117.

mle

`HydroErr.HydroErr.mle`(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the mean log error of the simulated and observed data.

$$MLE = \frac{1}{n}\sum_{i=0}^{n} ln(\frac{S_i}{O_i})$$

Range: -inf < MLE < inf, data units, closer to zero is better.

Notes Same as the mean erro (ME) only use log ratios as the error term. Limits the impact of outliers, more evenly weights high and low data values.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean log error value.

Return type float

Examples

Note that the value is very small because it is in log space.

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 6.8])
>>> he.mle(sim, obs)
0.002961767058151136
```

References

- Törnqvist, Leo, Pentti Vartia, and Yrjö O. Vartia. "How Should Relative Changes Be Measured?" The American Statistician 39, no. 1 (1985): 43–46.

male

HydroErr.HydroErr.**male**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the mean absolute log error of the simulated and observed data.

$$MALE = \frac{1}{n}\sum_{i=0}^{n}|ln(\frac{S_i}{O_i})|$$

Range: 0 MALE < inf, data units squared, smaller is better.

Notes Same as MAE only use log ratios as the error term. Limits the impact of outliers, more evenly weights high and low flows.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean absolute log error value.

Return type float

Examples

Note that the value is very small because it is in log space.

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 6.8])
>>> np.round(he.male(sim, obs), 6)
0.090417
```

References

- Törnqvist, Leo, Pentti Vartia, and Yrjö O. Vartia. "How Should Relative Changes Be Measured?" The American Statistician 39, no. 1 (1985): 43–46.

msle

HydroErr.HydroErr.**msle**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the mean squared log error of the simulated and observed data.

$$MALE = \frac{1}{n}\sum_{i=0}^{n}(ln(\frac{S_i}{O_i}))^2$$

Range: 0 MSLE < inf, data units squared, smaller is better.

Notes Same as the mean squared error (MSE) only use log ratios as the error term. Limits the impact of outliers, more evenly weights high and low values.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean squared log error value.

Return type float

Examples

Note that the value is very small because it is in log space.

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 6.8])
>>> np.round(he.msle(sim, obs), 6)
0.010426
```

References

- Törnqvist, Leo, Pentti Vartia, and Yrjö O. Vartia. "How Should Relative Changes Be Measured?" The American Statistician 39, no. 1 (1985): 43–46.

mde

HydroErr.HydroErr.**mde**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the median error (MdE) between the simulated and observed data.

$$MdE = median(S_i - O_i)$$

Range -inf < MdE < inf, closer to zero is better.

Notes This metric indicates bias. It is similar to the mean error (ME), only it takes the median rather than the mean. Median measures reduces the impact of outliers.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.

- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Examples

Note that the last outlier residual in the time series is negated using the median.

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 100])
>>> he.mde(sim, obs)
-0.10000000000000009
```

Returns The median error value.

Return type float

mdae

HydroErr.HydroErr.**mdae**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the median absolute error (MdAE) between the simulated and observed data.

$$MdE = median|S_i - O_i|$$

Range 0 MdAE < inf, closer to zero is better.

Notes Random errors (noise) do not cancel. It is the same as the mean absolute error (MAE), only it takes the median rather than the mean. Median measures reduces the impact of outliers.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.

- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Examples

Note that the last outlier residual in the time series is negated using the median.

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 100])
>>> he.mdae(sim, obs)
0.75
```

Returns The median absolute error value.

Return type float

mdse

HydroErr.HydroErr.**mdse**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the median squared error (MdSE) between the simulated and observed data.

$$MdE = median(S_i - O_i)^2$$

Range 0 MdSE < inf, closer to zero is better.

Notes Random errors (noise) do not cancel. It is the same as the mean squared error (MSE), only it takes the median rather than the mean. Median measures reduces the impact of outliers.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.

- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Examples

Note that the last outlier residual in the time series is negated using the median.

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 100])
>>> he.mdse(sim, obs)
0.625
```

Returns The median squared error value.

Return type float

ed

HydroErr.HydroErr.**ed**(*simulated_array, observed_array, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False*)

Compute the Euclidean distance between predicted and observed values in vector space.

$$ED = (\sum_{i=0}^{n} |S_i - O_i|^2)^{\frac{1}{2}}$$

Range 0 ED < inf, smaller is better. **Notes** Also sometimes referred to as the L2-norm.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.ed(sim, obs)
1.63707055437449
```

Returns The euclidean distance error value.

Return type float

References

- Kennard, M. J., Mackay, S. J., Pusey, B. J., Olden, J. D., & Marsh, N. (2010). Quantifying uncertainty in estimation of hydrologic metrics for ecohydrological studies. River Research and Applications, 26(2), 137-156.

ned

HydroErr.HydroErr.**ned**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the normalized Euclidian distance between the simulated and observed data in vector space.

$$NED = (\sum_{i=0}^{n} |\frac{S_i}{\bar{S}} - \frac{O_i}{\bar{O}}|^2)^{\frac{1}{2}}$$

Range 0 NED < inf, smaller is better.

Notes Also sometimes referred to as the squared L2-norm.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.

- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The normalized euclidean distance value.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.ned(sim, obs)
0.2872053604165771
```

References

- Kennard, M. J., Mackay, S. J., Pusey, B. J., Olden, J. D., & Marsh, N. (2010). Quantifying uncertainty in estimation of hydrologic metrics for ecohydrological studies. River Research and Applications, 26(2), 137-156.

rmse

`HydroErr.HydroErr.`**`rmse`**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the root mean square error between the simulated and observed data.

$$RMSE = (\frac{1}{n}\sum_{i=0}^{n}(S_i - O_i)^2)^{\frac{1}{2}}$$

Range 0 RMSE < inf, smaller is better.

Notes: The standard deviation of the residuals. A lower spread indicates that the points are better concentrated around the line of best fit (linear). Random errors do not cancel. This metric will highlights larger errors.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.

- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The root mean square error value.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.rmse(sim, obs)
0.668331255192114
```

References

- Willmott, C.J., Matsuura, K., 2005. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate Research 30(1) 79-82.
- Hyndman, R.J., Koehler, A.B., 2006. Another look at measures of forecast accuracy. International Journal of Forecasting 22(4) 679-688.

rmsle

HydroErr.HydroErr.**rmsle**(*simulated_array, observed_array, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False*)

Compute the root mean square log error between the simulated and observed data.

$$RMSLE = (\frac{1}{n}\sum_{i=0}^{n}(ln(\frac{S_i}{O_i}))^2)^{\frac{1}{2}}$$

Range: 0 RMSLE < inf. Smaller is better, and it does not indicate bias.

Notes: Random errors do not cancel while using this metric. This metric limits the impact of outliers by more evenly weighting high and low values. To calculate the log values, each value in the observed and simulated array is increased by one unit in order to avoid run-time errors and nan values (function np.log1p).

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.

- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The root mean square log error value.

Return type float

Examples

Notice that the value is very small because it is in log space.

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> np.round(he.rmsle(sim, obs), 6)
0.103161
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.
- Willmott, C.J., Matsuura, K., 2005. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate Research 30(1) 79-82.

nrmse_range

`HydroErr.HydroErr.`**nrmse_range**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the range normalized root mean square error between the simulated and observed data.

$$NRMSE_{Range} = \frac{RMSE}{O_{max} - O_{min}}$$

Range: 0 NRMSE < inf.

Notes: This metric is the RMSE normalized by the range of the observed time series (x). Normalizing allows comparison between data sets with different scales. The NRMSErange is the most sensitive to outliers of the three normalized rmse metrics.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The range normalized root mean square error value.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.nrmse_range(sim, obs)
0.0891108340256152
```

References

- Pontius, R.G., Thontteh, O., Chen, H., 2008. Components of information for multiple resolution comparison between maps that share a real variable. Environmental and Ecological Statistics 15(2) 111-142.

nrmse_mean

HydroErr.HydroErr.**nrmse_mean**(*simulated_array, observed_array, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False*)

Compute the mean normalized root mean square error between the simulated and observed data.

$$NRMSE_{Mean} = \frac{RMSE}{\bar{O}}$$

Range: 0 NRMSE < inf.

Notes: This metric is the RMSE normalized by the mean of the observed time series (x). Normalizing allows comparison between data sets with different scales.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean normalized root mean square error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.nrmse_mean(sim, obs)
0.11725109740212526
```

References

- Pontius, R.G., Thontteh, O., Chen, H., 2008. Components of information for multiple resolution comparison between maps that share a real variable. Environmental and Ecological Statistics 15(2) 111-142.

nrmse_iqr

HydroErr.HydroErr.**nrmse_iqr**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the IQR normalized root mean square error between the simulated and observed data.

$$NRMSE_{quartile} = \frac{RMSE}{Quartile_3 - Quartile_1}$$

Range: 0 NRMSE < inf.

Notes: This metric is the RMSE normalized by the interquartile range of the observed time series (x). Normalizing allows comparison between data sets with different scales. The NRMSEquartile is the least sensitive to outliers of the three normalized rmse metrics.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The IQR normalized root mean square error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.nrmse_iqr(sim, obs)
0.2595461185212093
```

References

- Pontius, R.G., Thontteh, O., Chen, H., 2008. Components of information for multiple resolution comparison between maps that share a real variable. Environmental and Ecological Statistics 15(2) 111-142.

irmse

`HydroErr.HydroErr.irmse(simulated_array, observed_array, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False)`

Compute the inertial root mean square error (IRMSE) between the simulated and observed data.

$$\Delta_o = (O_2 - O_1, O_3 - O_2, ..., O_n - O_{n-1})$$
$$\sigma_{\Delta_o} = \sqrt{\sum_{i=1}^{n} \frac{(\Delta_{o_i} - \overline{\Delta_o})^2}{n-1}} = \text{std}\left(\Delta_o\right)$$
$$\text{IRMSE} = \frac{RMSE}{\sigma_{\Delta_o}}$$

Range: 0 IRMSE < inf, lower is better.

Notes: This metric is the RMSE devided by by the standard deviation of the gradient of the observed timeseries data. This metric is meant to be help understand the ability of the model to predict changes in observation.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The inertial root mean square error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.irmse(sim, obs)
0.14572738134831856
```

References

- Daga, M., Deo, M.C., 2009. Alternative data-driven methods to estimate wind from waves by inverse modeling. Natural Hazards 49(2) 293-310.

mase

HydroErr.HydroErr.**mase**(*simulated_array, observed_array, m=1, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False*)

Compute the mean absolute scaled error between the simulated and observed data.

$$MASE = \frac{\sum_{i=1}^{n}|S_i - O_i|}{\frac{n}{n-1}\sum_{i=1}^{n}|O_i - O_{i-1}|}$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **m** (*int*) – If given, indicates the seasonal period m. If not given, the default is 1.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean absolute scaled error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.mase(sim, obs)
0.17341040462427745
```

References

- Hyndman, R.J., Koehler, A.B., 2006. Another look at measures of forecast accuracy. International Journal of Forecasting 22(4) 679-688.

r_squared

`HydroErr.HydroErr.r_squared`(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the the Coefficient of Determination (r2).

$$R^2 = \frac{(\sum_{i=1}^{n}(O_i - \overline{O})(S_i - \overline{S}))^2}{\sum_{i=1}^{n}(O_i - \overline{O})^2 \sum_{i=1}^{n}(S_i - \overline{S})^2}$$

Range: 0 r2 1. 1 indicates perfect correlation, 0 indicates complete randomness.

Notes: The Coefficient of Determination measures the linear relation between simulated and observed data. Because it is the pearson correlation coefficient squared, it is more heavily affected by outliers than the pearson correlation coefficient.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns

- *float* – The coefficient of determination (R^2).
- *>>> import HydroErr as he*
- *>>> import numpy as np*
- *>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])*
- *>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])*
- *>>> he.r_squared(sim, obs)*
- *0.9236735425294681*

References

pearson_r

`HydroErr.HydroErr.`**`pearson_r`**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the pearson correlation coefficient.

$$R_{Pearson} = \frac{\sum_{i=1}^{n}(O_i-\overline{O})(S_i-\overline{S})}{\sqrt{\sum_{i=1}^{n}(O_i-\overline{O})^2}\sqrt{\sum_{i=1}^{n}(S_i-\overline{S})^2}}$$

Range: -1 R (Pearson) 1. 1 indicates perfect postive correlation, 0 indicates complete randomness, -1 indicate perfect negative correlation.

Notes: The pearson r coefficient measures linear correlation. It is sensitive to outliers.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The Pearson correlation coefficient.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.pearson_r(sim, obs)
0.9610793632835262
```

References

- Pearson, K. (1895). Note on regression and inheritance in the case of two parents. Proceedings of the Royal Society of London, 58, 240-242.

spearman_r

`HydroErr.HydroErr.`**`spearman_r`**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the spearman rank correlation coefficient.

$$R_{Spearman} = \frac{\frac{1}{n}\sum_{i=1}^{n}(R(O_i)-\overline{R(O)})(R(S_i)-\overline{R(S)})}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(R(O_i)-\overline{R(O)})^2}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(R(S_i)-\overline{R(S)})^2}}$$

Range: -1 R (Pearson) 1. 1 indicates perfect postive correlation, 0 indicates complete randomness, -1 indicate perfect negative correlation.

Notes: The spearman r coefficient measures the monotonic relation between simulated and observed data. Because it uses a nonparametric measure of rank correlation, it is less sensitive to outliers compared to the Pearson correlation coefficient.

Parameters

- **`simulated_array`** (*`one dimensional ndarray`*) – An array of simulated data from the time series.
- **`observed_array`** (*`one dimensional ndarray`*) – An array of observed data from the time series.
- **`replace_nan`** (*`float, optional`*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **`replace_inf`** (*`float, optional`*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **`remove_neg`** (*`boolean, optional`*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **`remove_zero`** (*`boolean, optional`*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The Spearman rank correlation coefficient.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.spearman_r(sim, obs)
0.942857142857143
```

References

- Spearman C (1904). "The proof and measurement of association between two things". American Journal of Psychology. 15: 72–101. doi:10.2307/1412159

acc

HydroErr.HydroErr.**acc**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the the anomaly correlation coefficient (ACC).

$$ACC = \frac{1}{n}\frac{\sum_{i=1}^{n}(S_i-\overline{S})(O_i-\overline{O})}{\sigma_o \sigma_s}$$

Range: -1 ACC 1. -1 indicates perfect negative correlation of the variation pattern of the anomalies, 0 indicates complete randomness of the variation patterns of the anomalies, 1 indicates perfect correlation of the variation pattern of the anomalies.

Notes: Common measure in the verification of spatial fields. Measures the correlation between the variation pattern of the simulated data compared to the observed data.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The anomaly correlation coefficient.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.acc(sim, obs)
0.8008994694029383
```

References

- Langland, Rolf H., and Ryan N. Maue. "Recent Northern Hemisphere Mid-Latitude Medium-Range Deterministic Forecast Skill." Tellus A: Dynamic Meteorology and Oceanography 64, no. 1 (2012): 17531.
- Miyakoda, K., G. D. Hembree, R. F. Strickler, and I. Shulman. "Cumulative Results of Extended Forecast Experiments I. Model Performance for Winter Cases." Monthly Weather Review 100, no. 12(1972): 836–55.
- Murphy, Allan H., and Edward S. Epstein. "Skill Scores and Correlation Coefficients in Model Verification." Monthly Weather Review 117, no. 3 (1989): 572–82.

mape

HydroErr.HydroErr.**mape**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the the mean absolute percentage error (MAPE).

$$MAPE = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{S_i - O_i}{O_i} \right|$$

Range: 0% MAPE inf. 0% indicates perfect correlation, a larger error indicates a larger percent error in the data.

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.

- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean absolute percentage error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.mape(sim, obs)
11.639226612630866
```

References

mapd

`HydroErr.HydroErr.`**mapd**(*simulated_array, observed_array, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False*)

Compute the the mean absolute percentage deviation (MAPD).

$$MAPE = 100\% \frac{\sum_{i=1}^{n}|S_i - O_i|}{\sum_{i=1}^{n}|O_i|}$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the

observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.

- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean absolute percentage deviation.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.mapd(sim, obs)
0.10526315789473682
```

References

maape

`HydroErr.HydroErr.maape` (*simulated_array, observed_array, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False*)

Compute the the Mean Arctangent Absolute Percentage Error (MAAPE).

$$MAPE = \frac{1}{n}\sum_{i=1}^{n} arctan|\frac{S_i - O_i}{O_i}|$$

Range: 0 MAAPE < π/2, does not indicate bias, smaller is better.

Notes: Represents the mean absolute error as a percentage of the observed values. Handles 0s in the observed data. This metric is not as biased as MAPE by under-over predictions.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the

observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.

- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean arctangent absolute percentage error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.mape(sim, obs)
11.639226612630866
```

References

- Kim, S., Kim, H., 2016. A new metric of absolute percentage error for intermittent demand forecasts. International Journal of Forecasting 32(3) 669-679.

smape1

`HydroErr.HydroErr.smape1`(*simulated_array, observed_array, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False*)

Compute the the Symmetric Mean Absolute Percentage Error (1) (SMAPE1).

$$sMAPE1 = \frac{100\%}{n} \sum_{i=1}^{n} \frac{|S_i - O_i|}{|S_i| + |O_i|}$$

Range: 0 SMAPE1 < 100%, smaller is better, symmetrical.

Notes: This metric is an adjusted version of the MAPE.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.

- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The symmetric mean absolute percentage error (1).

Return type float

Examples

Note that if we switch the simulated and observed arrays the result is the same

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.smape1(sim, obs)
5.871915694397428
>>> he.smape1(obs, sim)
5.871915694397428
```

References

- Flores, B.E., 1986. A pragmatic view of accuracy measurement in forecasting. Omega 14(2) 93-98.
- Goodwin, P., Lawton, R., 1999. On the asymmetry of the symmetric MAPE. International Journal of Forecasting 15(4) 405-408.

smape2

HydroErr.HydroErr.**smape2**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the the Symmetric Mean Absolute Percentage Error (2) (SMAPE2).

$$sMAPE2 = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{S_i - O_i}{\frac{(S_i + O_i)}{2}} \right|$$

Range: 0 SMAPE1 < 200%, does not indicate bias, smaller is better, symmetrical.

Notes: This metric is an adjusted version of the MAPE with only positive metric values.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.

- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The symmetric mean absolute percentage error (2).

Return type float

Examples

Note that switching the simulated and observed arrays yields the same results

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.smape2(sim, obs)
11.743831388794856
>>> he.smape2(obs, sim)
11.743831388794856
```

References

- Flores, B.E., 1986. A pragmatic view of accuracy measurement in forecasting. Omega 14(2) 93-98.
- Goodwin, P., Lawton, R., 1999. On the asymmetry of the symmetric MAPE. International Journal of Forecasting 15(4) 405-408.

d

`HydroErr.HydroErr.d`(*simulated_array, observed_array, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False*)

Compute the the index of agreement (d).

$$d = 1 - \frac{\sum_{i=1}^{n}(S_i - O_i)^2}{\sum_{i=1}^{n}(|S_i - \overline{O}| + |O_i - \overline{O}|)^2}$$

Range: 0 d < 1, does not indicate bias, larger is better.

Notes: This metric is a modified approach to the Nash-Sutcliffe Efficiency metric.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The index of agreement (1).

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.d(sim, obs)
0.978477353035657
```

References

- Legates, D.R., McCabe Jr, G.J., 1999. Evaluating the use of "goodness-of-fit" Measures in hydrologic and hydroclimatic model validation. Water Resources Research 35(1) 233-241.
- Willmott, C.J., Robeson, S.M., Matsuura, K., 2012. A refined index of model performance. International Journal of Climatology 32(13) 2088-2094.

d1

HydroErr.HydroErr.**d1**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the the index of agreement (d1).

$$d_1 = 1 - \frac{\sum_{i=1}^{n} |S_i - O_i|}{\sum_{i=1}^{n} (|S_i - \overline{O}| + |O_i - \overline{O}|)}$$

Range: 0 d < 1, does not indicate bias, larger is better.

Notes: This metric is a modified approach to the Nash-Sutcliffe Efficiency metric. Compared to the other index of agreement (d) it has a reduced impact of outliers.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The index of agreement (d1).

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.d1(sim, obs)
0.8434782608695652
```

References

- Willmott, C.J., Robeson, S.M., Matsuura, K., 2012. A refined index of model performance. International Journal of Climatology 32(13) 2088-2094.

dmod

`HydroErr.HydroErr.dmod(simulated_array, observed_array, j=1, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False)`

Compute the the modified index of agreement (dmod).

$$d_{mod} = 1 - \frac{\sum_{i=1}^{n} |S_i - O_i|^j}{\sum_{i=1}^{n} (|S_i - \overline{O}| + |O_i - \overline{O}|)^j}$$

Range: 0 dmod < 1, does not indicate bias, larger is better.

Notes: When j=1, this metric is the same as d1. As j becomes larger, outliers have a larger impact on the value.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **j** (*int or float*) – Optional input indicating the j values desired. A higher j places more emphasis on outliers. j is 1 by default.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The modified index of agreement.

Return type float

Examples

Note that using the default is the same as calculating the d1 metric. Changing the value of j modification of the metric.

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.dmod(sim, obs)  # Same as d1
0.8434782608695652
```

(continues on next page)

(continued from previous page)

```
>>> he.dmod(sim, obs, j=1.5)
0.9413310986805733
```

References

- Krause, P., Boyle, D., Bäse, F., 2005. Comparison of different efficiency criteria for hydrological model assessment. Advances in geosciences 5 89-97.

drel

HydroErr.HydroErr.**drel**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the the relative index of agreement (drel).

$$d_{rel} = 1 - \frac{\sum_{i=1}^{n} (\frac{S_i - O_i}{O_i})^2}{\sum_{i=1}^{n} (\frac{|S_i - \bar{O}| + |O_i - \bar{O}|}{\bar{O}})^2}$$

Range: 0 drel < 1, does not indicate bias, larger is better.

Notes: Instead of absolute differences, this metric uses relative differences.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The relative index of agreement.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.drel(sim, obs)
0.9740868625579597
```

References

- Krause, P., Boyle, D., Bäse, F., 2005. Comparison of different efficiency criteria for hydrological model assessment. Advances in geosciences 5 89-97.

dr

`HydroErr.HydroErr.dr`(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the the refined index of agreement (dr).

$$\text{Note:} \quad ||S_i - O_i|| = \sum_{i=1}^{n} |S_i - O_i|$$
$$\text{When} \quad ||S_i - O_i|| \leq 2||O_i - \overline{O}||, \quad d_r = 1 - \frac{||S_i - O_i||}{2||O_i - \overline{O}||}$$
$$\text{When} \quad ||S_i - O_i|| > 2||O_i - \overline{O}||, \quad d_r = \frac{2||O_i - \overline{O}||}{||S_i - O_i||} - 1$$

Range: -1 dr < 1, does not indicate bias, larger is better.

Notes: Reformulation of Willmott's index of agreement. This metric was created to address issues in the index of agreement and the Nash-Sutcliffe efficiency metric. Meant to be a flexible metric for use in climatology.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The refined index of agreement.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.dr(sim, obs)
0.847457627118644
```

References

- Willmott, C.J., Robeson, S.M., Matsuura, K., 2012. A refined index of model performance. International Journal of Climatology 32(13) 2088-2094.

watt_m

HydroErr.HydroErr.**watt_m**(*simulated_array, observed_array, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False*)

Compute Watterson's M (M).

$$M = \left(\frac{2}{\pi}\right) sin^{-1}\left(1 - \frac{MSE}{\sigma_s^2+\sigma_o^2+(\overline{S}-\overline{O})^2}\right)$$

Range: -1 M < 1, does not indicate bias, larger is better.

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.

- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns Watterson's M value.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.watt_m(sim, obs)
0.8307913876595929
```

References

- Watterson, I.G., 1996. Non-dimensional measures of climate model performance. International Journal of Climatology 16(4) 379-391.

mb_r

HydroErr.HydroErr.**mb_r**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute Mielke-Berry R value (MB R).

$$\Re = 1 - \frac{MAE}{n^{-2}\sum_{j=1}^{n}\sum_{i=1}^{n}|S_j - O_i|}$$

Range: 0 MB R < 1, does not indicate bias, larger is better.

Notes: Compares prediction to probability it arose by chance.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the

observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.

- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The Mielke-Berry R value.

Return type float

Notes

If a more optimized version is desired, the numba package can be implemented for a much more optimized performance when computing this metric. An example is given below.

```
>>> from numba import njit, prange
```

```
>>> @njit(parallel=True, fastmath=True)
>>> def mb_par_fastmath(pred, obs):  # uses LLVM compiler
>>>     assert pred.size == obs.size
>>>     n = pred.size
>>>     tot = 0.0
>>>     mae = 0.0
>>>     for i in range(n):
>>>         for j in prange(n):
>>>             tot += abs(pred[i] - obs[j])
>>>         mae += abs(pred[i] - obs[i])
>>>     mae = mae / n
>>>     mb = 1 - ((n ** 2) * mae / tot)
>>>
>>> return mb
```

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.mb_r(sim, obs)
0.7726315789473684
```

References

- Berry, K.J., Mielke, P.W., 1988. A Generalization of Cohen's Kappa Agreement Measure to Interval Measurement and Multiple Raters. Educational and Psychological Measurement 48(4) 921-933.
- Mielke, P.W., Berry, K.J., 2007. Permutation methods: a distance function approach. Springer Science & Business Media.

nse

`HydroErr.HydroErr.nse`(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the Nash-Sutcliffe Efficiency.

$$NSE = 1 - \frac{\sum_{i=1}^{n}(S_i - O_i)^2}{\sum_{i=1}^{n}(O_i - \bar{O})^2}$$

Range: -inf < NSE < 1, does not indicate bias, larger is better.

Notes: The Nash-Sutcliffe efficiency metric compares prediction values to naive predictions (i.e. average value). One major flaw of this metric is that it punishes a higher variance in the observed values (denominator). This metric is analogous to the mean absolute error skill score (MAESS) using the mean flow as a benchmark.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The Nash-Sutcliffe Efficiency value.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.nse(sim, obs)
0.922093023255814
```

References

- Krause, P., Boyle, D., Bäse, F., 2005. Comparison of different efficiency criteria for hydrological model assessment. Advances in geosciences 5 89-97.
- McCuen, R.H., Knight, Z., Cutter, A.G., 2006. Evaluation of the Nash-Sutcliffe Efficiency Index. Journal of Hydraulic Engineering.
- Nash, J.E., Sutcliffe, J.V., 1970. River flow forecasting through conceptual models part I — A discussion of principles. Journal of Hydrology 282-290.
- Willmott, C.J., Robeson, S.M., Matsuura, K., 2012. A refined index of model performance. International Journal of Climatology 32(13) 2088-2094.

nse_mod

`HydroErr.HydroErr.`**`nse_mod`**(*simulated_array*, *observed_array*, *j=1*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the modified Nash-Sutcliffe efficiency (NSE mod).

$$NSE_{mod} = 1 - \frac{\sum_{i=1}^{n} |S_i - O_i|^j}{\sum_{i=1}^{n} |O_i - \overline{O}|^j}$$

Range: -inf < NSE (mod) < 1, does not indicate bias, larger is better.

Notes: The modified Nash-Sutcliffe efficiency metric gives less weight to outliers if j=1, or more weight to outliers if j is higher. Generally, j=1.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **j** (*int or float*) – If given, sets the value of j to the input. j is 1 by default. A higher j gives more emphasis to outliers
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The modified Nash-Sutcliffe efficiency value.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.nse_mod(sim, obs)
0.6949152542372882
```

References

- Krause, P., Boyle, D., Bäse, F., 2005. Comparison of different efficiency criteria for hydrological model assessment. Advances in geosciences 5 89-97.

nse_rel

`HydroErr.HydroErr.nse_rel`(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the relative Nash-Sutcliffe efficiency (NSE rel).

$$NSE_{rel} = 1 - \frac{\sum_{i=1}^{n}\left|\frac{S_i - O_i}{O_i}\right|^2}{\sum_{i=1}^{n}\left|\frac{O_i - \bar{O}}{\bar{O}}\right|^2}$$

Range: -inf < NSE (rel) < 1, does not indicate bias, larger is better.

Notes: The modified Nash-Sutcliffe efficiency metric gives less weight to outliers if j=1, or more weight to outliers if j is higher. Generally, j=1.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The relative Nash-Sutcliffe efficiency value.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.nse_rel(sim, obs)
0.9062004687708474
```

References

- Krause, P., Boyle, D., Bäse, F., 2005. Comparison of different efficiency criteria for hydrological model assessment. Advances in geosciences 5 89-97.

kge_2009

HydroErr.HydroErr.**kge_2009**(*simulated_array*, *observed_array*, *s=(1, 1, 1)*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*, *return_all=False*)

Compute the Kling-Gupta efficiency (2009).

$$KGE_{2009} = 1 - ED$$
$$ED = \sqrt{(s[1]*(r-1))^2 + (s[2]*(\alpha-1))^2 + (s[3]*(\beta-1))^2}$$
$$r = \text{Pearson Correlation Coefficient}$$
$$\beta = \mu_s/\mu_o$$
$$\alpha = \sigma_s/\sigma_o$$

Range: -inf < KGE (2009) < 1, larger is better.

Notes: Gupta et al. (2009) created this metric to demonstrate the relative importance of the three components of the NSE, which are correlation, bias and variability. This was done with hydrologic modeling as the context. This metric is meant to address issues with the NSE.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **s** (*tuple of length three*) – Represents the scaling factors to be used for re-scaling the Pearson product-moment correlation coefficient (r), Alpha, and Beta, respectively.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.

- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **return_all** (*bool*) – If True, returns all of the components of the KGE metric, which are r, alpha, and beta, respectively.

Returns The Kling-Gupta (2009) efficiency value, unless the return_all parameter is True.

Return type float (tuple of float)

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 6.8])
>>> he.kge_2009(sim, obs)
0.912223072345668
```

```
>>> he.kge_2009(sim, obs, return_all=True)  # Returns (r, alpha, beta, kge)
(0.9615951377405804, 0.927910707932087, 1.0058823529411764, 0.9181073779138655)
```

References

- Gupta, H. V., Kling, H., Yilmaz, K. K., & Martinez, G. F. (2009). Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling. Journal of Hydrology, 377(1-2), 80-91.

kge_2012

HydroErr.HydroErr.**kge_2012**(*simulated_array, observed_array, s=(1, 1, 1), replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False, return_all=False*)

Compute the Kling-Gupta efficiency (2012).

$$KGE_{2012} = 1 - ED$$
$$ED = \sqrt{(s[1]*(r-1))^2 + (s[2]*(\gamma-1))^2 + (s[3]*(\beta-1))^2}$$
$$r = \text{Pearson Correlation Coefficient}$$
$$\beta = \mu_s/\mu_o$$
$$\gamma = \frac{CV_s}{CV_o} = \frac{\sigma_s/\mu_s}{\sigma_o/\mu_o}$$

Range: -inf < KGE (2012) < 1, does not indicate bias, larger is better.

Notes: The modified version of the KGE (2009). Kling proposed this version to avoid cross-correlation between bias and variability ratios.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **s** (*tuple of length three*) – Represents the scaling factors to be used for re-scaling the Pearson product-moment correlation coefficient (r), gamma, and Beta, respectively.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **return_all** (*bool*) – If True, returns all of the components of the KGE metric, which are r, gamma, and beta, respectively.

Returns The Kling-Gupta (2012) efficiency value, unless the return_all parameter is True.

Return type float (tuple of float)

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 6.8])
>>> he.kge_2012(sim, obs)
0.9122230723456678
```

```
>>> he.kge_2012(sim, obs, return_all=True)  # Returns (r, alpha, beta, kge)
(0.9615951377405804, 0.9224843295231272, 1.0058823529411764, 0.9132923608280753)
```

References

- Kling, H., Fuchs, M., & Paulin, M. (2012). Runoff conditions in the upper Danube basin under an ensemble of climate change scenarios. Journal of Hydrology, 424, 264-277.

lm_index

HydroErr.HydroErr.**lm_index**(*simulated_array*, *observed_array*, *obs_bar_p=None*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the Legate-McCabe Efficiency Index.

$$E_1' = 1 - \frac{\sum_{i=1}^{n}|S_i - O_i|}{\sum_{i=1}^{n}\left|O_i - \overline{O_i'}\right|}$$

Range: 0 E1' < 1, does not indicate bias, larger is better.

Notes: The obs_bar_p argument represents a seasonal or other selected average.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **obs_bar_p** (*float*) – Seasonal or other selected average. If None, the mean of the observed array will be used.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The Legate-McCabe Efficiency index value.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.lm_index(sim, obs)
0.6949152542372882
```

References

- Legates, D.R., McCabe Jr, G.J., 1999. Evaluating the use of "goodness-of-fit" Measures in hydrologic and hydroclimatic model validation. Water Resources Research 35(1) 233-241. Lehmann, E.L., Casella, G., 1998. Springer Texts in Statistics. Springer-Verlag, New York.

d1_p

`HydroErr.HydroErr.d1_p`(*simulated_array*, *observed_array*, *obs_bar_p=None*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the Legate-McCabe Index of Agreement.

$$d_1' = 1 - \frac{\sum_{i=1}^{n}|S_i - O_i|}{\sum_{i=1}^{n}\left|S_i - \overline{O_i'}\right| + \left|O_i - \overline{O_i'}\right|}$$

Range: 0 d1' < 1, does not indicate bias, larger is better.

Notes: The obs_bar_p argument represents a seasonal or other selected average.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **obs_bar_p** (*float*) – Seasonal or other selected average. If None, the mean of the observed array will be used.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The Legate-McCabe Efficiency index of agreement.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.d1_p(sim, obs)
0.8434782608695652
```

References

- Legates, D.R., McCabe Jr, G.J., 1999. Evaluating the use of "goodness-of-fit" Measures in hydrologic and hydroclimatic model validation. Water Resources Research 35(1) 233-241. Lehmann, E.L., Casella, G., 1998. Springer Texts in Statistics. Springer-Verlag, New York.

ve

`HydroErr.HydroErr.`**`ve`**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the Volumetric Efficiency (VE).

$$VE = 1 - \frac{\sum_{i=1}^{n}|S_i - O_i|}{\sum_{i=1}^{n} O_i}$$

Range: 0 VE < 1 smaller is better, does not indicate bias.

Notes: Represents the error as a percentage of flow.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The Volumetric Efficiency value.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.ve(sim, obs)
0.8947368421052632
```

References

- Criss, R.E., Winston, W.E., 2008. Do Nash values have value? Discussion and alternate proposals. Hydrological Processes 22(14) 2723.

sa

`HydroErr.HydroErr.sa`(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the Spectral Angle (SA).

$$SA = arccos\left(\frac{\langle S,O \rangle}{||S||_2||O||_2}\right)$$

Range: -π/2 SA < π/2, closer to 0 is better.

Notes: The spectral angle metric measures the angle between the two vectors in hyperspace. It indicates how well the shape of the two series match – not magnitude.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The Spectral Angle value.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.sa(sim, obs)
0.10816831366492945
```

References

- Robila, S.A., Gershman, A., 2005. Spectral matching accuracy in processing hyperspectral data, Signals, Circuits and Systems, 2005. ISSCS 2005. International Symposium on. IEEE, pp. 163-166.

sc

`HydroErr.HydroErr.`**`sc`**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the Spectral Correlation (SC).

$$SC = arccos\left(\frac{\langle(S_i-\overline{S})(O_i-\overline{O})\rangle}{||S_i-\overline{S}||_2||O_i-\overline{O}||_2}\right)$$

Range: $-\pi/2$ SA $< \pi/2$, closer to 0 is better.

Notes: The spectral correlation metric measures the angle between the two vectors in hyperspace. It indicates how well the shape of the two series match – not magnitude.

Parameters

- **`simulated_array`** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **`observed_array`** (*one dimensional ndarray*) – An array of observed data from the time series.
- **`replace_nan`** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **`replace_inf`** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **`remove_neg`** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The Spectral Correlation value.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.sc(sim, obs)
0.27991341383646606
```

References

- Robila, S.A., Gershman, A., 2005. Spectral matching accuracy in processing hyperspectral data, Signals, Circuits and Systems, 2005. ISSCS 2005. International Symposium on. IEEE, pp. 163-166.

sid

HydroErr.HydroErr.**sid**(*simulated_array, observed_array, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False*)

Compute the Spectral Information Divergence (SID).

$$\left\langle \left(\frac{O_i}{\bar{O}} - \frac{S_i}{\bar{S}}\right), \left(log\left(\frac{O_i}{\bar{O}}\right) - log\left(\frac{S_i}{\bar{S}}\right)\right) \right\rangle$$

Range: -π/2 SID < π/2, closer to 0 is better.

Notes: The spectral information divergence measures the angle between the two vectors in hyperspace. It indicates how well the shape of the two series match – not magnitude.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.

- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The Spectral information divergence value.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.sid(sim, obs)
0.03517616895318012
```

References

- Robila, S.A., Gershman, A., 2005. Spectral matching accuracy in processing hyperspectral data, Signals, Circuits and Systems, 2005. ISSCS 2005. International Symposium on. IEEE, pp. 163-166.

sga

`HydroErr.HydroErr.`**sga**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the Spectral Gradient Angle (SGA).

$$\text{SG}_o = (O_2 - O_1, O_3 - O_2, ..., O_n - O_{n-1})$$
$$\text{SG}_s = (S_2 - S_1, S_3 - S_2, ..., S_n - S_{n-1})$$
$$\text{SGA} = SA(SG_o, SG_s)$$
$$\text{Note: SA=Spectral Angle Metric}$$

Range: -π/2 SID < π/2, closer to 0 is better.

Notes: The spectral gradient angle measures the angle between the two vectors in hyperspace. It indicates how well the shape of the two series match – not magnitude. SG is the gradient of the simulated or observed time series.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the

observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.

- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The Spectral Gradient Angle.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.sga(sim, obs)
0.26764286472739834
```

References

- Robila, S.A., Gershman, A., 2005. Spectral matching accuracy in processing hyperspectral data, Signals, Circuits and Systems, 2005. ISSCS 2005. International Symposium on. IEEE, pp. 163-166.

h1_mhe

HydroErr.HydroErr.**h1_mhe**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H1 mean error.

$$H_1 = \frac{S_i - O_i}{O_i}$$

$$\text{Mean H Error} = \frac{1}{n}\sum_{i=1}^{n} H$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.

- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean H1 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h1_mhe(sim, obs)
0.002106551840594386
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h1_mahe

HydroErr.HydroErr.**h1_mahe**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H1 absolute error.

$$H_1 = \frac{S_i - O_i}{O_i}$$

$$\text{Absolute H Error} = \frac{1}{n}\sum_{i=1}^{n}|H|$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The H1 absolute error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h1_mahe(sim, obs)
0.11639226612630865
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h1_rmshe

HydroErr.HydroErr.**h1_rmshe**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H1 root mean square error.

$$H_1 = \frac{S_i - O_i}{O_i}$$

$$\text{Root Mean Squared H Error} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} H^2}$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The root mean squared H1 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h1_rmshe(sim, obs)
0.12865571253672756
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h2_mhe

`HydroErr.HydroErr.`**`h2_mhe`**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H2 mean error.

$$H_2 = \frac{S_i - O_i}{S_i}$$

$$\text{Mean H Error} = \frac{1}{n}\sum_{i=1}^{n} H$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean H2 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h2_mhe(sim, obs)
-0.015319829424307036
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h2_mahe

`HydroErr.HydroErr.`**`h2_mahe`**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H2 mean absolute error.

$$H_2 = \frac{S_i - O_i}{S_i}$$

$$\text{Absolute H Error} = \frac{1}{n} \sum_{i=1}^{n} |H|$$

Range:

Notes:

Parameters

- **`simulated_array`** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **`observed_array`** (*one dimensional ndarray*) – An array of observed data from the time series.
- **`replace_nan`** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **`replace_inf`** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **`remove_neg`** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **`remove_zero`** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean absolute H2 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h2_mahe(sim, obs)
0.11997591408039167
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h2_rmshe

`HydroErr.HydroErr.h2_rmshe`(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H2 root mean square error.

$$H_1 = \frac{S_i - O_i}{O_i}$$

$$\text{Mean H Error} = \frac{1}{n} \sum_{i=1}^{n} H$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The root mean square H2 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h2_rmshe(sim, obs)
0.1373586680669673
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h3_mhe

`HydroErr.HydroErr.`**`h3_mhe`**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H3 mean error.

$$H_3 = \frac{S_i - O_i}{\frac{1}{2}(S_i + O_i)}$$

$$\text{Mean H Error} = \frac{1}{n}\sum_{i=1}^{n} H$$

Range:

Notes:

Parameters

- **`simulated_array`** (*`one dimensional ndarray`*) – An array of simulated data from the time series.
- **`observed_array`** (*`one dimensional ndarray`*) – An array of observed data from the time series.
- **`replace_nan`** (*`float, optional`*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **`replace_inf`** (*`float, optional`*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **`remove_neg`** (*`boolean, optional`*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **`remove_zero`** (*`boolean, optional`*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean H3 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h3_mhe(sim, obs)
-0.006322019630356533
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h3_mahe

`HydroErr.HydroErr.`**`h3_mahe`**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H3 mean absolute error.

$$H_3 = \frac{S_i - O_i}{\frac{1}{2}(S_i + O_i)}$$

$$\text{Absolute H Error} = \frac{1}{n} \sum_{i=1}^{n} |H|$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean absolute H3 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h3_mahe(sim, obs)
0.11743831388794855
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h3_rmshe

HydroErr.HydroErr.**h3_rmshe**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H3 root mean square error.

$$H_3 = \frac{S_i - O_i}{\frac{1}{2}(S_i + O_i)}$$

$$\text{Root Mean Squared H Error} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} H^2}$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the

observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.

- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The root mean square H3 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h3_rmshe(sim, obs)
0.13147667616722278
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h4_mhe

HydroErr.HydroErr.**h4_mhe**(*simulated_array, observed_array, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False*)

Compute the H4 mean error.

$$H_4 = \frac{S_i - O_i}{\sqrt{S_i O_i}}$$

$$\text{Mean H Error} = \frac{1}{n}\sum_{i=1}^{n} H$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.

- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean H4 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h4_mhe(sim, obs)
-0.0064637371129817
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h4_mahe

HydroErr.HydroErr.**h4_mahe**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H4 mean absolute error.

$$H_4 = \frac{S_i - O_i}{\sqrt{S_i O_i}}$$

$$\text{Absolute H Error} = \frac{1}{n}\sum_{i=1}^{n}|H|$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean absolute H4 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h4_mahe(sim, obs)
0.11781032209144082
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h4_rmshe

HydroErr.HydroErr.**h4_rmshe**(*simulated_array, observed_array, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False*)

Compute the H4 mean error.

$$H_4 = \frac{S_i - O_i}{\sqrt{S_i O_i}}$$

$$\text{Root Mean Squared H Error} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} H^2}$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The root mean square H4 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h4_rmshe(sim, obs)
0.13200901963465006
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h5_mhe

HydroErr.HydroErr.**h5_mhe**(*simulated_array, observed_array, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False*)

Compute the H5 mean error.

$$H_5 = \frac{S_i - O_i}{\left[\frac{1}{2}\left(O_i^{-1} + S_i^{-1}\right)\right]^{-1}}$$

$$\text{Mean H Error} = \frac{1}{n}\sum_{i=1}^{n} H$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean H5 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h5_mhe(sim, obs)
-0.006606638791856322
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h5_mahe

`HydroErr.HydroErr.h5_mahe`(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H5 mean absolute error.

$$H_5 = \frac{S_i - O_i}{\left[\frac{1}{2}\left(O_i^{-1} + S_i^{-1}\right)\right]^{-1}}$$

$$\text{Absolute H Error} = \frac{1}{n}\sum_{i=1}^{n}|H|$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean absolute H5 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h5_mahe(sim, obs)
0.11818409010335018
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h5_rmshe

`HydroErr.HydroErr.h5_rmshe`(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H5 root mean square error.

$$H_5 = \frac{S_i - O_i}{\left[\frac{1}{2}\left(O_i^{-1} + S_i^{-1}\right)\right]^{-1}}$$

$$\text{Root Mean Squared H Error} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} H^2}$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The root mean square H5 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h5_rmshe(sim, obs)
0.13254476469410933
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h6_mhe

HydroErr.HydroErr.**h6_mhe**(*simulated_array*, *observed_array*, *k=1*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H6 mean error.

$$H_6 = \frac{S_i - O_i}{\left[\frac{1}{2}\left(O_i^k + S_i^k\right)\right]^{1/k}}$$

$$\text{Mean H Error} = \frac{1}{n}\sum_{i=1}^{n} H$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **k** (*int or float*) – If given, sets the value of k. If None, k=1.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean H6 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h6_mhe(sim, obs)
-0.006322019630356514
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h6_mahe

HydroErr.HydroErr.**h6_mahe**(*simulated_array*, *observed_array*, *k=1*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H6 mean absolute error.

$$H_6 = \frac{S_i - O_i}{\left[\frac{1}{2}\left(O_i^k + S_i^k\right)\right]^{1/k}}$$

$$\text{Absolute H Error} = \frac{1}{n}\sum_{i=1}^{n}|H|$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **k** (*int or float*) – If given, sets the value of k. If None, k=1.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean absolute H6 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h6_mahe(sim, obs)
0.11743831388794852
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h6_rmshe

HydroErr.HydroErr.**h6_rmshe**(*simulated_array, observed_array, k=1, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False*)

Compute the H6 root mean square error.

$$H_6 = \frac{S_i - O_i}{\left[\frac{1}{2}\left(O_i^k + S_i^k\right)\right]^{1/k}}$$

$$\text{Root Mean Squared H Error} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} H^2}$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **k** (*int or float*) – If given, sets the value of k. If None, k=1.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.

- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The root mean square H6 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h6_rmshe(sim, obs)
0.13147667616722278
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h7_mhe

HydroErr.HydroErr.**h7_mhe**(*simulated_array, observed_array, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False*)

Compute the H7 mean error.

$$H_7 = \frac{S_i - O_i}{min(O_i, S_i)}$$

$$\text{Mean H Error} = \frac{1}{n}\sum_{i=1}^{n} H$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.

- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean H7 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h7_mhe(sim, obs)
0.0026331898007430263
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h7_mahe

HydroErr.HydroErr.**h7_mahe**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H7 mean absolute error.

$$H_7 = \frac{S_i - O_i}{min(O_i, S_i)}$$

$$\text{Absolute H Error} = \frac{1}{n}\sum_{i=1}^{n}|H|$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean absolute H7 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h7_mahe(sim, obs)
0.14549033265788583
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h7_rmshe

HydroErr.HydroErr.**h7_rmshe**(*simulated_array, observed_array, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False*)

Compute the H7 root mean square error.

$$H_7 = \frac{S_i - O_i}{min(O_i, S_i)}$$

$$\text{Root Mean Squared H Error} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} H^2}$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The root mean square H7 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h7_rmshe(sim, obs)
0.16081964067090945
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h8_mhe

HydroErr.HydroErr.**h8_mhe**(*simulated_array, observed_array, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False*)

Compute the H8 mean error.

$H_8 = \frac{S_i - O_i}{max(O_i, S_i)}$

$\text{Mean H Error} = \frac{1}{n}\sum_{i=1}^{n} H$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean H8 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h8_mhe(sim, obs)
0.0018056158633666466
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h8_mahe

HydroErr.HydroErr.**h8_mahe**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H8 mean absolute error.

$$H_8 = \frac{S_i - O_i}{max(O_i, S_i)}$$

$$\text{Absolute H Error} = \frac{1}{n}\sum_{i=1}^{n}|H|$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean absolute H8 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h8_mahe(sim, obs)
0.099764799536836
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h8_rmshe

`HydroErr.HydroErr.`**`h8_rmshe`**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H8 root mean square error.

$$H_8 = \frac{S_i - O_i}{max(O_i, S_i)}$$

$$\text{Root Mean Squared H Error} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} H^2}$$

Range:

Notes:

Parameters

- **`simulated_array`** (*`one dimensional ndarray`*) – An array of simulated data from the time series.
- **`observed_array`** (*`one dimensional ndarray`*) – An array of observed data from the time series.
- **`replace_nan`** (*`float, optional`*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **`replace_inf`** (*`float, optional`*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **`remove_neg`** (*`boolean, optional`*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **`remove_zero`** (*`boolean, optional`*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The root mean square H8 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h8_rmshe(sim, obs)
0.11027632503148076
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h10_mhe

HydroErr.HydroErr.**h10_mhe**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H10 mean error.

$$H_{10} = \ln \frac{S_i}{O_i}$$

$$\text{Mean H Error} = \frac{1}{n} \sum_{i=1}^{n} H$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean H10 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.h10_mhe(sim, obs)
-0.0012578676058971154
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h10_mahe

`HydroErr.HydroErr.`**`h10_mahe`**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H10 mean absolute error.

$$H_{10} = \ln \frac{S_i}{O_i}$$

$$\text{Absolute H Error} = \frac{1}{n} \sum_{i=1}^{n} |H|$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean absolute H10 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> np.round(he.h10_mahe(sim, obs), 6)
0.094636
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

h10_rmshe

HydroErr.HydroErr.**h10_rmshe**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the H10 root mean square error.

$$H_{10} = \ln \frac{S_i}{O_i}$$

$$\text{Root Mean Squared H Error} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} H^2}$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.

- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The root mean square H10 error.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> np.round(he.h10_rmshe(sim, obs), 6)
0.103161
```

References

- Tornquist, L., Vartia, P., Vartia, Y.O., 1985. How Should Relative Changes be Measured? The American Statistician 43-46.

g_mean_diff

HydroErr.HydroErr.**g_mean_diff**(*simulated_array*, *observed_array*, *replace_nan=None*, *replace_inf=None*, *remove_neg=False*, *remove_zero=False*)

Compute the geometric mean difference.

$$\text{GM} = e^{(S' - O')}$$
$$S' = \left(\prod_{i=1}^{n} \ln\left(S_i\right)\right)^{\frac{1}{n}}$$
$$O' = \left(\prod_{i=1}^{n} \ln\left(O_i\right)\right)^{\frac{1}{n}}$$

Range:

Notes: For the difference of geometric means, the geometric mean is computed for each of two samples then their difference is taken.

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.

- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The geometric mean difference value.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> he.g_mean_diff(sim, obs)
0.988855412098022
```

References

mean_var

HydroErr.HydroErr.**mean_var**(*simulated_array, observed_array, replace_nan=None, replace_inf=None, remove_neg=False, remove_zero=False*)

Compute the mean variance.

$$\mathrm{MV} = \mathrm{var}(\ln(O_1), \ln(O_2), ..., \ln(O_n)) - \mathrm{var}(\ln(S_1), \ln(S_2), ..., \ln(S_n))$$

Range:

Notes:

Parameters

- **simulated_array** (*one dimensional ndarray*) – An array of simulated data from the time series.
- **observed_array** (*one dimensional ndarray*) – An array of observed data from the time series.
- **replace_nan** (*float, optional*) – If given, indicates which value to replace NaN values with in the two arrays. If None, when a NaN value is found at the i-th position in the observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.
- **replace_inf** (*float, optional*) – If given, indicates which value to replace Inf values with in the two arrays. If None, when an inf value is found at the i-th position in the

observed OR simulated array, the i-th value of the observed and simulated array are removed before the computation.

- **remove_neg** (*boolean, optional*) – If True, when a negative value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.
- **remove_zero** (*boolean, optional*) – If true, when a zero value is found at the i-th position in the observed OR simulated array, the i-th value of the observed AND simulated array are removed before the computation.

Returns The mean variance.

Return type float

Examples

```
>>> import HydroErr as he
>>> import numpy as np
```

```
>>> sim = np.array([5, 7, 9, 2, 4.5, 6.7])
>>> obs = np.array([4.7, 6, 10, 2.5, 4, 7])
>>> np.round(he.mean_var(sim, obs), 6)
0.010641
```

References

CHAPTER 3

Contributing to HydroErr

3.1 Steps

The steps to contributing are simple:

1. Download git.
2. Clone the HydroErr repository.
3. Develop features that you would like implemented, including a docstring in functions of classes in the numpydoc format.
4. Add a test case to cover the code that you created (HydroErr/tests directory, in tests.py).
5. Create a new pull request with your changes.

A note to members of the BYU-Hydroinformatics group. Please make pull requests before merging changes to the master branch, as this allows continuous integration testing to take place.

CHAPTER 4

Release Notes

This is the list of changes to HydroErr between each release. For full details, see the commit logs at https://github.com/BYU-Hydroinformatics/HydroErr.

4.1 Version 1.24

- Added .name and .abbr properties to each metric to give a little bit more info to the users, and also for ease of use in the Hydrostats package.
- Allowed users to import all metrics by simply using `import HydroErr` instead of `import HydroErr.HydroErr`.

CHAPTER 5

Indices and tables

- genindex
- modindex
- search

Python Module Index

h

Index

P

R

S

V

W