

# Package ‘HyMETT’

May 7, 2026

**Type** Package

**Title** Hydrologic Model Evaluation and Time-Series Tools

**Version** 1.1.3

**Date** 2024-08-26

**Description** Facilitates the analysis and evaluation of hydrologic model output and time-series data with functions focused on comparison of modeled (simulated) and observed data, period-of-record statistics, and trends.

**URL** <https://code.usgs.gov/hymett/hymett>

**BugReports** <https://code.usgs.gov/hymett/hymett/-/issues>

**Depends** R (>= 3.6.0)

**Imports** checkmate, dplyr, EnvStats, lmomco, lubridate, plyr, rlang, stats, tibble, zoo

**Suggests** knitr, rmarkdown, roxygen2, testthat

**License** CC0

**LazyLoad** yes

**LazyData** yes

**VignetteBuilder** knitr

**BuildVignettes** true

**Copyright** This software is in the public domain because it contains materials that originally came from the U.S. Geological Survey, an agency of the U.S. Department of Interior. For more information, see the official USGS copyright policy at [http://www.usgs.gov/visual-id/credit\\_usgs.html#copyright](http://www.usgs.gov/visual-id/credit_usgs.html#copyright)

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Colin Penn [aut, cre] (ORCID: <<https://orcid.org/0000-0002-5195-2744>>),  
Caelan Simeone [aut] (ORCID: <<https://orcid.org/0000-0003-3263-6452>>),  
Sara Levin [aut] (ORCID: <<https://orcid.org/0000-0002-2448-3129>>),

Samuel Saxe [aut] (ORCID: <<https://orcid.org/0000-0003-1151-8908>>),  
 Sydney Foks [aut] (ORCID: <<https://orcid.org/0000-0002-7668-9735>>),  
 Robert Dudley [dct] (ORCID: <<https://orcid.org/0000-0002-0934-0568>>),  
 Glenn Hodgkins [dct] (ORCID: <<https://orcid.org/0000-0002-4916-5565>>),  
 Timothy Hodson [aut] (ORCID: <<https://orcid.org/0000-0003-0962-5130>>),  
 Thomas Over [dct] (ORCID: <<https://orcid.org/0000-0001-8280-4368>>),  
 Amy Russell [dct] (ORCID: <<https://orcid.org/0000-0003-0582-0094>>)

**Maintainer** Colin Penn <cpenn@usgs.gov>

**Repository** CRAN

**Date/Publication** 2024-08-28 23:00:11 UTC

## Contents

|  |    |
|--|----|
| HyMETT-package . . . . .                 | 3  |
| benchmark_KGE_DOY . . . . .              | 3  |
| calc_annual_flow_stats . . . . .         | 4  |
| calc_annual_stat_trend . . . . .         | 7  |
| calc_logistic_regression . . . . .       | 9  |
| calc_qlpearsonIII . . . . .              | 10 |
| calc_qpearsonIII . . . . .               | 11 |
| sensor_values . . . . .                  | 12 |
| example_annual . . . . .                 | 13 |
| example_mod . . . . .                    | 16 |
| example_mod_zf . . . . .                 | 17 |
| example_obs . . . . .                    | 18 |
| example_obs_zf . . . . .                 | 19 |
| example_preproc . . . . .                | 20 |
| GOF_correlation_tests . . . . .          | 21 |
| GOF_kling_gupta_efficiency . . . . .     | 22 |
| GOF_mean_absolute_error . . . . .        | 23 |
| GOF_mean_error . . . . .                 | 23 |
| GOF_nash_sutcliffe_efficiency . . . . .  | 24 |
| GOF_percent_bias . . . . .               | 25 |
| GOF_rmse . . . . .                       | 26 |
| GOF_summary . . . . .                    | 27 |
| GOF_volumetric_efficiency . . . . .      | 29 |
| POR_apply_annual_hiflow_stats . . . . .  | 30 |
| POR_apply_annual_lowflow_stats . . . . . | 31 |
| POR_calc_amp_and_phase . . . . .         | 32 |
| POR_calc_AR1 . . . . .                   | 33 |
| POR_calc_lp3_quantile . . . . .          | 34 |
| POR_deseasonalize . . . . .              | 35 |
| POR_distribution_metrics . . . . .       | 36 |
| preproc_audit_data . . . . .             | 37 |
| preproc_fill_daily . . . . .             | 39 |
| preproc_main . . . . .                   | 40 |
| preproc_precondition_data . . . . .      | 42 |

|                                  |           |
|----------------------------------|-----------|
| <i>HyMETT-package</i>            | 3         |
| preproc_validate_daily . . . . . | 44        |
| <b>Index</b>                     | <b>45</b> |

---

|                |  |
|----------------|--|
| HyMETT-package | <i>Hydrologic Model Evaluation and Time-series Tools</i> |
|----------------|--|

---

### Description

Facilitates the analysis and evaluation of hydrologic model output and time-series data with functions focused on comparison of modeled (simulated) and observed data, period-of-record statistics, and trends.

### Details

Please see [doi:10.5066/P9FNXEVI](https://doi.org/10.5066/P9FNXEVI) for more details.

---

|                   |  |
|-------------------|--|
| benchmark_KGE_DOY | <i>Calculate benchmark Kling–Gupta efficiency (KGE) values from day-of-year (DOY) observations</i> |
|-------------------|--|

---

### Description

Calculate benchmark Kling–Gupta efficiency (KGE) values from daily observed time-series data

### Usage

```
benchmark_KGE_DOY(obs_preproc)
```

### Arguments

obs\_preproc     'data.frame' of daily observational data, preprocessed as output from [preproc\\_precondition\\_data](#) or [preproc\\_main](#) "daily".

### Details

This function calculates a "benchmark" KGE value (see Knoben and others, 2020) from a daily observed data time-series. First, the interannual mean and median is calculated for each day of the calendar year. Next, the interannual mean and median values are joined to each corresponding day in the observation time series. Finally, a KGE value ([GOF\\_kling\\_gupta\\_efficiency](#)) is calculated comparing the mean or median value repeated time series to the daily observational time series. These benchmark KGE values can be used as comparisons for modeled (simulated) calibration results.

### Value

A data.frame with columns "KGE\_DOY\_mean" and "KGE\_DOY\_median".

## References

Knoben, W.J.M, Freer, J.E., Peel, M.C., Fowler, K.J.A, Woods, R.A., 2020. A Brief Analysis of Conceptual Model Structure Uncertainty Using 36 Models and 559 Catchments: Water Resources Research, v. 56.

[Also available at <https://doi.org/10.1029/2019WR025975>.]

## Examples

```
benchmark_KGE_DOY(obs_preproc = example_preproc)
```

---

```
calc_annual_flow_stats
```

*Calculate annual flow statistics from daily data*

---

## Description

Calculate annual flow statistics from daily data

## Usage

```
calc_annual_flow_stats(
  data = NULL,
  Date,
  year_group,
  Q,
  Q3 = NA_real_,
  Q7 = NA_real_,
  Q30 = NA_real_,
  jd = NA_integer_,
  calc_high = FALSE,
  calc_low = FALSE,
  calc_percentiles = FALSE,
  calc_monthly = FALSE,
  calc_WSCVD = FALSE,
  longitude = NA,
  calc_ICVD = FALSE,
  zero_threshold = 33,
  quantile_type = 8,
  na.action = c("na.omit", "na.pass")
)
```

## Arguments

|      |   |
|------|---|
| data | 'data.frame'. Optional data.frame input, with columns containing Date, year_group, Q, and Q3, Q7, Q30, jd (if required). Column names are specified as strings in the corresponding parameter. Default is NULL. |
|------|---|

|                  |  |
|------------------|--|
| Date             | 'Date' or 'character' vector when data = NULL, or 'character' string identifying Date column name when data is specified. Date associated with each value in Q parameter.  |
| year_group       | 'numeric' vector when data = NULL, or 'character' string identifying grouping column name when data is specified. Year grouping for each daily value in Q parameter. Must be same length as Q parameter. Often year_group is water year or climate year.   |
| Q                | 'numeric' vector when data = NULL, or 'character' string identifying streamflow values column name when data is specified. Daily streamflow data. Must be same length as year_group.   |
| Q3               | 'numeric' vector when data = NULL, or 'character' string identifying Q3 column name when data is specified. 3-day moving average of daily streamflow data Q parameter, often returned from <a href="#">preproc_precondition_data</a> . Default is NA_real_, required if calc_high or calc_low = TRUE. If specified, must be same length as Q parameter.                          |
| Q7               | 'numeric' vector when data = NULL, or 'character' string identifying Q7 column name when data is specified. 7-day moving average of daily streamflow data Q parameter, often returned from <a href="#">preproc_precondition_data</a> . Default is NA_real_, required if calc_high or calc_low = TRUE. If specified, must be same length as Q parameter.                          |
| Q30              | 'numeric' vector when data = NULL, or 'character' string identifying Q30 column name when data is specified. 30-day average of daily streamflow data Q parameter, often returned from <a href="#">preproc_precondition_data</a> . Default is NA_real_, required if calc_high or calc_low = TRUE. If specified, must be same length as Q parameter.                               |
| jd               | 'numeric' vector when data = NULL, or 'character' string identifying jd column name when data is specified. Calendar Julian day of daily streamflow data Q parameter, often returned from <a href="#">preproc_precondition_data</a> . Default is NA_integer_, required if calc_high, calc_low, calc_WSCVD or calc_ICVD = TRUE. If specified, must be same length as Q parameter. |
| calc_high        | 'boolean' value. Calculate high flow statistics for years in year_group. Default is FALSE. See <b>Details</b> for more information.  |
| calc_low         | 'boolean' value. Calculate low flow statistics for years in year_group. Default is FALSE. See <b>Details</b> for more information.   |
| calc_percentiles | 'boolean' value. Calculate percentiles for years in year_group. Default is FALSE. See <b>Details</b> for more information.   |
| calc_monthly     | 'boolean' value. Calculate monthly statistics for years in year_group. Default is FALSE. See <b>Details</b> for more information.  |
| calc_WSCVD       | 'boolean' value. Calculate winter-spring center volume date for years in year_group. Default is FALSE. See <b>Details</b> for more information.  |
| longitude        | 'numeric' value. Site longitude in North American Datum of 1983 (NAD83), required in WSCVD calculation. Default is NA. See <b>Details</b> for more information.  |
| calc_ICVD        | 'boolean' value. Calculate inverse center volume date for years in year_group. Default is FALSE. See <b>Details</b> for more information.  |

|                |  |
|----------------|--|
| zero_threshold | 'numeric' value as percentage. The percentage of years of a statistic that need to be zero in order for it to be deemed a zero flow site for that statistic. For use in trend calculation. See <b>Details</b> on attributes. Default is 33 (33 percent) of the annual statistic values.          |
| quantile_type  | 'numeric' value. The distribution type used in the <code>stats::quantile</code> function. Default is 8 (median-unbiased regardless of distribution). Other types common in hydrology are 6 (Weibull) or 9 (unbiased for normal distributions).   |
| na.action      | 'character' string indicating na.action passed to <code>stats::aggregate</code> na.action parameter. Default is "na.omit", which removes NA values before aggregating statistics, or "na.pass", which will pass NA values and return NA in the grouped calculation if any NA values are present. |

## Details

year\_group is commonly water year, climate year, or calendar year.

Default annual statistics returned:

annual\_mean annual mean in year\_group

annual\_sd annual standard deviation in year\_group

annual\_sum annual sum in year\_group

If calc\_high/low are selected, annual statistics returned:

1-, 3-, 7-, and 30-day high/low and Julian date (jd) of n-day high/low.

high\_qn where  $n = 1, 3, 7, \text{ and } 30$

high\_qn\_jd where  $n = 1, 3, 7, \text{ and } 30$

low\_qn where  $n = 1, 3, 7, \text{ and } 30$

low\_qn\_jd where  $n = 1, 3, 7, \text{ and } 30$

If calc\_percentiles is selected, annual statistics returned:

1, 5, 10, 25, 50, 75, 90, 95, 99 percentile based on daily streamflow.

annual\_n\_percentile where  $n = 1, 5, 10, 25, 50, 75, 90, 95, \text{ and } 99$

If calc\_monthly is selected, annual statistics returned:

Monthly mean, standard deviation, max, min, percent of annual for each month in year\_group.

**month\_mean** monthly mean, where *month* = `month.abb`

**month\_sd** monthly standard deviation, where *month* = `month.abb`

**month\_max** monthly maximum, where *month* = `month.abb`

**month\_min** monthly minimum, where *month* = `month.abb`

**month\_percent\_annual** monthly percent of annual, where *month* = `month.abb`

If calc\_WSCVD is selected, Julian date of annual winter-spring center volume date is returned.

Longitude (in NAD83 datum) is used to determine the ending month of spring. July for longitudes West of  $-95$  degrees, May for longitudes east of  $-95$  degrees. See **References** Dudley and others, 2017. Commonly calculated when year\_group is water year.

WSCVD Julian date of winter-spring center volume

If calc\_ICVD is selected, Julian date of annual inverse center volume date is returned.  
Commonly calculated when year\_group is climate year.

ICVD Julian date of inverse center volume date

**Attribute:** zero\_flow\_years

A data.frame with each annual statistic calculated, the percentage of years where the statistic = 0, a flag indicating if the percentage is over the zero\_threshold parameter, and the number of years with a zero value. Columns in zero\_flow\_years:

annual\_stat annual statistic

percent\_zeros percentage of years with 0 statistic value

over\_threshold boolean if percentage is over threshold

number\_years number of years with 0 value statistic

The zero\_flow\_years attribute can be useful in trend calculation, where a trend may not be appropriate to calculate with many zero flow years.

### Value

A tibble (see [tibble::tibble](#)) with annual statistics depending on options selected. See **Details**.

### References

Dudley, R.W., Hodgkins, G.A, McHale, M.R., Kolian, M.J., Renard, B., 2017, Trends in snowmelt-related streamflow timing in the conterminous United States: Journal of Hydrology, v. 547, p. 208-221. [Also available at <https://doi.org/10.1016/j.jhydrol.2017.01.051>.]

### See Also

[preproc\\_precondition\\_data](#)

### Examples

```
calc_annual_flow_stats(data = example_preproc, Date = "Date", year_group = "WY", Q = "value")
```

---

calc\_annual\_stat\_trend

*Calculate trend in annual statistics*

---

### Description

Calculate trend in annual statistics

### Usage

```
calc_annual_stat_trend(data = NULL, year, value, ...)
```

**Arguments**

|       |   |
|-------|---|
| data  | 'data.frame'. Optional data.frame input, with columns containing year and value. Column names are specified as strings in the corresponding parameter. Default is NULL. |
| year  | 'numeric' vector when data = NULL, or 'character' string identifying year column name when data is specified. Year of each value in value parameter.                    |
| value | 'numeric' vector when data = NULL, or 'character' string identifying value column name when data is specified. Values to calculate trend on.                            |
| ...   | further arguments to be passed to or from <a href="#">EnvStats::kendallTrendTest</a> .  |

**Details**

This function is a wrapper for [EnvStats::kendallTrendTest](#) with the passed equation value ~ year. The returned values include Mann-Kendall test statistic and p-value, Theil-Sen slope and intercept values, and trend details (Millard, 2013; Helsel and others, 2020).

z\_stat Mann-Kendall test statistic, returned directly from [EnvStats::kendallTrendTest](#)  
 p\_value z\_stat p-value, returned directly from [EnvStats::kendallTrendTest](#)  
 sen\_slope Sen slope in units value per year, returned directly from [EnvStats::kendallTrendTest](#)  
 intercept Sen slope intercept, returned directly from [EnvStats::kendallTrendTest](#)  
 trend\_mag Trend magnitude over entire period, in units of value, calculated as  $\text{sen\_slope} * (\max(\text{year}) - \min(\text{year}))$   
 val\_beg/end Calculated value at beginning or end of period, calculated as  $\text{sen\_slope} * \text{year} + \text{intercept}$   
 val\_perc\_change Percentage change over period, calculated as  $(\text{val\_end} - \text{val\_beg}) / \text{val\_beg} * 100$

**Value**

A tibble (see [tibble::tibble](#)) with test statistic, p-value, trend coefficients, and trend calculations. See **Details**.

**References**

Millard, S.P., 2013, EnvStats: An R Package for Environmental Statistics: New York, New York, Springer, 291 p. [Also available at <https://doi.org/10.1007/978-1-4614-8456-1>.]  
 Helsel, D.R., Hirsch, R.M., Ryberg, K.R., Archfield, S.A., and Gilroy, E.J., 2020, Statistical methods in water resources: U.S. Geological Survey Techniques and Methods, book 4, chap. A3, 458 p. [Also available at <https://doi.org/10.3133/tm4a3>.]

**See Also**

[kendallTrendTest](#)

**Examples**

```
calc_annual_stat_trend(data = example_annual, year = "WY", value = "annual_mean")
```

---

 calc\_logistic\_regression

*Calculate logistic regression in annual statistics with zero values*


---

## Description

Calculate logistic regression (Everitt and Hothorn, 2009) in annual statistics with zero values. A model fit to compute the probability of a zero flow annual statistic.

## Usage

```
calc_logistic_regression(data = NULL, year, value, ...)
```

## Arguments

|       |   |
|-------|---|
| data  | 'data.frame'. Optional data.frame input, with columns containing year and value. Column names are specified as strings in the corresponding parameter. Default is NULL. |
| year  | 'numeric' vector when data = NULL, or 'character' string identifying year column name when data is specified. Year of each value in value parameter.                    |
| value | 'numeric' vector when data = NULL, or 'character' string identifying value column name when data is specified. Values to calculate logistic regression on.              |
| ...   | further arguments to be passed to or from <code>stats::glm</code> .   |

## Details

This function is a wrapper for `stats::glm(y ~ year, family = stats::binomial(link="logit"))` with  $y = 1$  when  $value = 0$  (for example a zero flow annual statistic) and  $y = 0$  otherwise. The returned values include

`p_value` Probability value of the explanatory (year) variable in the logistic model

`stdErr_slope` Standard error of the regression slope (log odds per year)

`odds_ratio` Exponential of the explanatory coefficient (year coefficient)

`prob_beg/end` Logistic regression predicted (fitted) values at the beginning and ending year.

`prob_change` Change in probability from beginning to end.

Example, an odds ratio of 1.05 represents the odds of a zero-flow year (versus non-zero) increase by a factor of 1.05 (or 5 percent).

## Value

A tibble (see `tibble::tibble`) with logistic regression p-value, standard error of slope, odds ratio, beginning and ending probability, and probability change. See **Details**.

**References**

Everitt, B. S. and Hothorn T., 2009, A Handbook of Statistical Analyses Using R, 2nd Ed. Boca Raton, Florida, Chapman and Hall/CRC, 376p.

**See Also**

[glm](#)

**Examples**

```
calc_logistic_regression(data = example_annual, year = "WY", value = "annual_mean")
```

---

calc\_qlpearsonIII      *Quantile of Pearson Type III distribution for log-transformed data*

---

**Description**

Quantile of Pearson Type III distribution for log-transformed data

**Usage**

```
calc_qlpearsonIII(p, meanlog = 0, sdlog = 1, skew = 0)
```

**Arguments**

|         |  |
|---------|--|
| p       | Vector of non-exceedance probabilities, between 0 and 1, to calculate quantiles. |
| meanlog | Vector of mean of the distribution of the log-transformed data.                  |
| sdlog   | Vector of standard deviation of the distribution of the log-transformed data.    |
| skew    | Vector of skewness of the distribution of the log-transformed data.              |

**Details**

[calc\\_qpearsonIII](#) and [calc\\_qlpearsonIII](#) are functions to fit a log-Pearson type III distribution from a given mean, standard deviation, and skew. This source code is replicated, unchanged, from the smwrBase package in order to reduce the dependency on that package.

**Value**

Quantiles for the described distribution

**References**

Asquith, W.H., Kiang, J.E., and Cohn, T.A., 2017, Application of at-site peak-streamflow frequency analyses for very low annual exceedance probabilities: U.S. Geological Survey Scientific Investigation Report 2017–5038, 93 p. [Also available at <https://doi.org/10.3133/sir20175038>.]

Lorenz, D.L., 2015, smwrBase—An R package for managing hydrologic data, version 1.1.1: U.S. Geological Survey Open-File Report 2015–1202, 7 p. [Also available at <https://doi.org/10.3133/ofr20151202>.]

**See Also**[calc\\_qpearsonIII](#)**Examples**

```
calc_qlpearsonIII(0.1)
```

---

|                  |  |
|------------------|--|
| calc_qpearsonIII | <i>Quantile of Pearson Type III distribution</i> |
|------------------|--|

---

**Description**

Quantile of Pearson Type III distribution

**Usage**

```
calc_qpearsonIII(p, mean = 0, sd = 1, skew = 0)
```

**Arguments**

|      |  |
|------|--|
| p    | Vector of non-exceedance probabilities, between 0 and 1, to calculate quantiles. |
| mean | Vector of means of the distribution of the data.                                 |
| sd   | Vector of standard deviation of the distribution of the data.                    |
| skew | Vector of skewness of the distribution of the data.                              |

**Details**

[calc\\_qpearsonIII](#) and [calc\\_qlpearsonIII](#) are functions to fit a log-Pearson type III distribution from a given mean, standard deviation, and skew. This source code is replicated, unchanged, from the `swmrBase` package in order to reduce the dependency on that package.

**Value**

Quantiles for the described distribution

**References**

Asquith, W.H., Kiang, J.E., and Cohn, T.A., 2017, Application of at-site peak-streamflow frequency analyses for very low annual exceedance probabilities: U.S. Geological Survey Scientific Investigation Report 2017–5038, 93 p. [Also available at <https://doi.org/10.3133/sir20175038>.]

Lorenz, D.L., 2015, `swmrBase`—An R package for managing hydrologic data, version 1.1.1: U.S. Geological Survey Open-File Report 2015–1202, 7 p. [Also available at <https://doi.org/10.3133/ofr20151202>.]

**Examples**

```
calc_qpearsonIII(0.1)
```

---

|               |   |
|---------------|---|
| censor_values | <i>Censor values above or below a threshold</i> |
|---------------|---|

---

### Description

Replaces values in a vector with NA when above or below a censor level.

Censoring is values `censor_symbol` `censor_threshold` are censored, for example with the defaults (values lte 0 set to NA) all values  $\leq 0$  are replaced with NA.

### Usage

```
censor_values(  
  value,  
  censor_threshold = 0,  
  censor_symbol = c("lte", "lt", "gt", "gte")  
)
```

### Arguments

`value` 'numeric' vector. Values to censor.

`censor_threshold` 'numeric' value. Threshold to censor values on. Default is 0.

`censor_symbol` 'character' string.  
Inequality symbol to censor values based on `censor_threshold`.  
Accepted values are "gt" (greater than),  
"gte" (greater than or equal to),  
"lt" (less than),  
or "lte" (less than or equal to).  
Default is "lte".

### Value

'numeric' vector with censored values replaced with NA

### Examples

```
censor_values(value = seq.int(1, 10, 1), censor_threshold = 5)
```

---

|                |                                    |
|----------------|------------------------------------|
| example_annual | <i>Example Annual Observations</i> |
|----------------|------------------------------------|

---

**Description**

An example dataset with daily observed streamflow processed to annual water year values.

**Usage**

example\_annual

**Format**

A data.frame with the following variables:

WY water year

annual\_mean annual mean

annual\_sd annual standard deviation

annual\_sum annual sum

high\_q1 annual maximum of daily mean

high\_q3 annual maximum of 3-day mean

high\_q7 annual maximum of 7-day mean

high\_q30 annual maximum of 30-day mean

high\_q1\_jd Julian day of annual maximum of daily mean

high\_q3\_jd Julian day of annual maximum of 3-day mean

high\_q7\_jd Julian day of annual maximum of 7-day mean

high\_q30\_jd Julian day of annual maximum of 30-day mean

low\_q7 annual minimum of 7-day mean

low\_q30 annual minimum of 30-day mean

low\_q3 annual minimum of 3-day mean

low\_q1 annual minimum of daily mean

low\_q7\_jd Julian day of annual minimum of 7-day mean

low\_q30\_jd Julian day of annual minimum of 30-day mean

low\_q3\_jd Julian day of annual minimum of 3-day mean

low\_q1\_jd Julian day of annual minimum of daily mean

annual\_1\_percentile annual first percentile

annual\_5\_percentile annual 5th percentile

annual\_10\_percentile annual 10th percentile

annual\_25\_percentile annual 25th percentile

annual\_50\_percentile annual 50th percentile

annual\_75\_percentile annual 75th percentile  
annual\_90\_percentile annual 90th percentile  
annual\_95\_percentile annual 95th percentile  
annual\_99\_percentile annual 99th percentile  
Jan\_mean annual January mean  
Jan\_sd annual January standard deviation  
Jan\_max annual January maximum  
Jan\_min annual January minimum  
Jan\_percent\_annual annual January percentage of annual sum  
Feb\_mean annual February mean  
Feb\_sd annual February standard deviation  
Feb\_max annual February maximum  
Feb\_min annual February minimum  
Feb\_percent\_annual annual February percentage of annual sum  
Mar\_mean annual March mean  
Mar\_sd annual March standard deviation  
Mar\_max annual March maximum  
Mar\_min annual March minimum  
Mar\_percent\_annual annual March percentage of annual sum  
Apr\_mean annual April mean  
Apr\_sd annual April standard deviation  
Apr\_max annual April maximum  
Apr\_min annual April minimum  
Apr\_percent\_annual annual April percentage of annual sum  
May\_mean annual May mean  
May\_sd annual May standard deviation  
May\_max annual May maximum  
May\_min annual May minimum  
May\_percent\_annual annual May percentage of annual sum  
Jun\_mean annual June mean  
Jun\_sd annual June standard deviation  
Jun\_max annual June maximum  
Jun\_min annual June minimum  
Jun\_percent\_annual annual June percentage of annual sum  
Jul\_mean annual July mean  
Jul\_sd annual July standard deviation  
Jul\_max annual July maximum

Jul\_min annual July minimum  
 Jul\_percent\_annual annual July percentage of annual sum  
 Aug\_mean annual August mean  
 Aug\_sd annual August standard deviation  
 Aug\_max annual August maximum  
 Aug\_min annual August minimum  
 Aug\_percent\_annual annual August percentage of annual sum  
 Sep\_mean annual September mean  
 Sep\_sd annual September standard deviation  
 Sep\_max annual September maximum  
 Sep\_min annual September minimum  
 Sep\_percent\_annual annual September percentage of annual sum  
 Oct\_mean annual October mean  
 Oct\_sd annual October standard deviation  
 Oct\_max annual October maximum  
 Oct\_min annual October minimum  
 Oct\_percent\_annual annual October percentage of annual sum  
 Nov\_mean annual November mean  
 Nov\_sd annual November standard deviation  
 Nov\_max annual November maximum  
 Nov\_min annual November minimum  
 Nov\_percent\_annual annual November percentage of annual sum  
 Dec\_mean annual December mean  
 Dec\_sd annual December standard deviation  
 Dec\_max annual December maximum  
 Dec\_min annual December minimum  
 Dec\_percent\_annual annual December percentage of annual sum  
 WSV winter-spring volume  
 wscvd Julian date of winter-spring center volume

### Details

Generated with [example\\_obs](#) from

```
HyMETT::preproc_main(data = example_obs,
                      Date = "Date", value = "streamflow_cfs", longitude = -68)$annual
```

### See Also

[example\\_obs](#), [preproc\\_main](#)

### Examples

```
str(example_annual)
```

---

`example_mod`*Example Model Output*

---

**Description**

An example dataset with daily modeled (simulated) streamflow.

**Usage**

```
example_mod
```

**Format**

A data.frame with the following variables:

`date` date as 'character' column class.

`streamflow_cfs` modeled streamflow in units of feet<sup>3</sup>/second.

`Date` date as 'Date' column class.

**Details**

Generated from example data available at `system.file("extdata", "01013500_MOD.csv", package = "HyMETT")`

**References**

Johnson, M., D. Blodgett, 2020, NOAA National Water Model Reanalysis Data at RENCI, HydroShare, accessed September 17, 2020 at <https://doi.org/10.4211/hs.89b0952512dd4b378dc5be8d2093310f>

Johnson, M., 2021, nwmHistoric: National Water Model Historic Data. R package version 0.0.0.9000, accessed September 17, 2020 at <https://github.com/mikejohnson51/nwmHistoric>

**Examples**

```
str(example_mod)
```

---

|                |   |
|----------------|---|
| example_mod_zf | <i>Example Model Output with zero flows</i> |
|----------------|---|

---

## Description

An example dataset with daily modeled (simulated) streamflow that includes zero flows.

## Usage

```
example_mod_zf
```

## Format

A data.frame with the following variables:

date date as 'character' column class.

streamflow\_cfs modeled streamflow in units of feet<sup>3</sup>/second.

Date date as 'Date' column class.

## Details

Generated from example data available at `system.file("extdata", "08202700_MOD.csv", package = "HyMETT")`

## References

Johnson, M., D. Blodgett, 2020, NOAA National Water Model Reanalysis Data at RENCI, HydroShare, accessed September 17, 2020 at <https://doi.org/10.4211/hs.89b0952512dd4b378dc5be8d2093310f>

Johnson, M., 2021, nwmHistoric: National Water Model Historic Data. R package version 0.0.0.9000, accessed September 17, 2020 at <https://github.com/mikejohnson51/nwmHistoric>

## Examples

```
str(example_mod_zf)
```

---

`example_obs`*Example Observations*

---

**Description**

An example dataset with daily observed streamflow.

**Usage**

```
example_obs
```

**Format**

A data.frame with the following variables:

`date` date as 'character' column class.

`streamflow_cfs` observed streamflow in units of feet<sup>3</sup>/second.

`quality_cd` qualifier for value in `streamflow_cfs` (U.S. Geological Survey, 2020b)

`Date` date as 'Date' column class.

**Details**

Generated from example data available at `system.file("extdata", "01013500_OBS.csv", package = "HyMETT")`

**References**

De Cicco, L.A., Hirsch, R.M., Lorenz, D., and Watkins, W.D., 2021, dataRetrieval: R packages for discovering and retrieving water data available from Federal hydrologic web services, accessed September 16, 2020 at <https://doi.org/10.5066/P9X4L3GE>.

U.S. Geological Survey, 2020a, USGS water data for the Nation: U.S. Geological Survey National Water Information System database, accessed September 16, 2020, at <https://doi.org/10.5066/F7P55KJN>.

U.S. Geological Survey, 2020b, Instantaneous and Daily Data-Value Qualification Codes, in USGS water data for the Nation: U.S. Geological Survey National Water Information System database, accessed September 16, 2020, at <https://doi.org/10.5066/F7P55KJN>. [information directly accessible at [https://help.waterdata.usgs.gov/codes-and-parameters/instantaneous-value-qualification-code-uv\\_rmk\\_cd.](https://help.waterdata.usgs.gov/codes-and-parameters/instantaneous-value-qualification-code-uv_rmk_cd.)]

**Examples**

```
str(example_obs)
```

---

`example_obs_zf`*Example Observations with zero flows*

---

**Description**

An example dataset with daily observed streamflow that includes zero flows.

**Usage**`example_obs_zf`**Format**

A data.frame with the following variables:

`date` date as 'character' column class.

`streamflow_cfs` observed streamflow in units of feet<sup>3</sup>/second.

`quality_cd` qualifier for value in `streamflow_cfs` (U.S. Geological Survey, 2020b)

`Date` date as 'Date' column class.

**Details**

Generated from example data available at `system.file("extdata", "08202700_OBS.csv", package = "HyMETT")`

**References**

De Cicco, L.A., Hirsch, R.M., Lorenz, D., and Watkins, W.D., 2021, dataRetrieval: R packages for discovering and retrieving water data available from Federal hydrologic web services, accessed September 16, 2020 at <https://doi.org/10.5066/P9X4L3GE>.

U.S. Geological Survey, 2020a, USGS water data for the Nation: U.S. Geological Survey National Water Information System database, accessed September 16, 2020, at <https://doi.org/10.5066/F7P55KJN>.

U.S. Geological Survey, 2020b, Instantaneous and Daily Data-Value Qualification Codes, in USGS water data for the Nation: U.S. Geological Survey National Water Information System database, accessed September 16, 2020, at <https://doi.org/10.5066/F7P55KJN>. [information directly accessible at [https://help.waterdata.usgs.gov/codes-and-parameters/instantaneous-value-qualification-code-uv\\_rmk\\_cd.](https://help.waterdata.usgs.gov/codes-and-parameters/instantaneous-value-qualification-code-uv_rmk_cd.)]

**Examples**`str(example_obs_zf)`

---

example\_preproc

*Example Observations preprocessed*

---

### Description

An example dataset with daily observed streamflow preprocessed to include additional timing and n-day moving averages.

### Usage

```
example_preproc
```

### Format

A data.frame with the following variables:

Date

value

year

month

day

decimal\_date

WY Water Year: October 1 - September 30

CY Climate Year: April 1 - March 30

Q3 3-Day Moving Average: computed at end of moving interval

Q7 7-Day Moving Average: computed at end of moving interval

Q30 30-Day Moving Average: computed at end of moving interval

jd Julian date

### Details

Generated with [example\\_obs](#) from

```
HyMETT::preproc_main(data = example_obs,  
                      Date = "Date", value = "streamflow_cfs", longitude = -68)$daily`
```

### See Also

[example\\_obs](#), [preproc\\_main](#)

### Examples

```
str(example_preproc)
```

---

GOF\_correlation\_tests *Calculates Kendall's Tau, Spearman's Rho, Pearson Correlation*

---

### Description

Calculates Kendall's Tau, Spearman's Rho, Pearson Correlation, and p-values as a wrapper to the `stats::cor.test` function. Output is tidy-style data.frame.

### Usage

```
GOF_correlation_tests(mod, obs, na.rm = TRUE, ...)
```

### Arguments

|                    |  |
|--------------------|--|
| <code>mod</code>   | 'numeric' vector. Modeled or simulated values. Must be same length as <code>obs</code> .   |
| <code>obs</code>   | 'numeric' vector. Observed or comparison values. Must be same length as <code>mod</code> .   |
| <code>na.rm</code> | 'boolean' TRUE or FALSE. Should NA values be removed before computing. If any NA values are present in <code>mod</code> or <code>obs</code> , the <i>i</i> th position from each will be removed before calculating. If NA values are present and <code>na.rm = FALSE</code> , then function will return NA. Default is TRUE |
| <code>...</code>   | Further arguments to be passed to or from <code>stats::cor.test</code> .   |

### Details

See `stats::cor.test` for more details and further arguments to be passed to or from methods. Defaults are used.

### Value

A tibble (`tibble::tibble`) with test statistic values and p-values.

### See Also

`cor.test`

### Examples

```
GOF_correlation_tests(mod = example_mod$streamflow_cfs, obs = example_obs$streamflow_cfs)
```

---

GOF\_kling\_gupta\_efficiency

*Calculate Kling–Gupta Efficiency (KGE)*


---

### Description

Calculate Kling–Gupta Efficiency (KGE) (or modified KGE ('KGE')) between modeled (simulated) and observed values.

### Usage

```
GOF_kling_gupta_efficiency(mod, obs, modified = FALSE, na.rm = TRUE)
```

### Arguments

|          |  |
|----------|--|
| mod      | 'numeric' vector. Modeled or simulated values. Must be same length as obs.   |
| obs      | 'numeric' vector. Observed or comparison values. Must be same length as mod.   |
| modified | 'boolean' TRUE or FALSE. Should the KGE calculation use the original variability ratio in the standard deviations (see Gupta and others, 2009) (modified = FALSE) or the modified variability ratio in the coefficient of variations (see Kling and others, 2012) (modified = TRUE). Default is FALSE. |
| na.rm    | 'boolean' TRUE or FALSE. Should NA values be removed before computing. If any NA values are present in mod or obs, the <i>i</i> th position from each will be removed before calculating. If NA values are present and na.rm = FALSE, then function will return NA. Default is TRUE.                   |

### Value

Value of computed KGE or 'KGE.

### References

Kling, H., Fuchs, M. and Paulin, M., 2012. Runoff conditions in the upper Danube basin under an ensemble of climate change scenarios: *Journal of Hydrology*, v. 424-425, p. 264-277.

[Also available at <https://doi.org/10.1016/j.jhydrol.2012.01.011>.]

Gupta, H.V., Kling, H., Yilmaz, K.K., and Martinez, G.G., 2009. Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modelling: *Journal of Hydrology*, v. 377, no.1-2, p. 80-91.

[Also available at <https://doi.org/10.1016/j.jhydrol.2009.08.003>.]

### Examples

```
GOF_kling_gupta_efficiency(
  mod = example_mod$streamflow_cfs, obs = example_obs$streamflow_cfs
)
```

---

`GOF_mean_absolute_error`*Calculates mean absolute error (MAE).*

---

**Description**

Calculates mean absolute error (MAE) between modeled (simulated) and observed values. Error is defined as modeled minus observed.

**Usage**

```
GOF_mean_absolute_error(mod, obs, na.rm = TRUE)
```

**Arguments**

|                    |   |
|--------------------|---|
| <code>mod</code>   | 'numeric' vector. Modeled or simulated values. Must be same length as <code>obs</code> .  |
| <code>obs</code>   | 'numeric' vector. Observed or comparison values. Must be same length as <code>mod</code> .  |
| <code>na.rm</code> | 'boolean' TRUE or FALSE. Should NA values be removed before computing. If any NA values are present in <code>mod</code> or <code>obs</code> , the <i>i</i> th position from each will be removed before calculating. If NA values are present and <code>na.rm = FALSE</code> , then function will return NA. Default is TRUE. |

**Details**

The absolute value of each modeled-observed pair error is calculated, then the mean of those values taken. Values returned are in units of input data.

**Value**

Value of calculated mean absolute error (MAE).

**Examples**

```
GOF_mean_absolute_error(mod = example_mod$streamflow_cfs, obs = example_obs$streamflow_cfs)
```

---

`GOF_mean_error`*Calculates mean error.*

---

**Description**

Calculates mean error between modeled (simulated) and observed values. Error is defined as modeled minus observed.

**Usage**

```
GOF_mean_error(mod, obs, na.rm = TRUE)
```

**Arguments**

|       |  |
|-------|--|
| mod   | 'numeric' vector. Modeled or simulated values. Must be same length as obs.   |
| obs   | 'numeric' vector. Observed or comparison values. Must be same length as mod.   |
| na.rm | 'boolean' TRUE or FALSE. Should NA values be removed before computing. If any NA values are present in mod or obs, the <i>i</i> th position from each will be removed before calculating. If NA values are present and na.rm = FALSE, then function will return NA. Default is TRUE. |

**Details**

Values returned are in units of input data.

**Value**

Value of calculated mean error.

**Examples**

```
GOF_mean_error(mod = example_mod$streamflow_cfs, obs = example_obs$streamflow_cfs)
```

---

GOF\_nash\_sutcliffe\_efficiency

*Calculate Nash–Sutcliffe Efficiency (NSE)*

---

**Description**

Calculate Nash–Sutcliffe Efficiency (NSE) (with options for modified NSE) between modeled (simulated) and observed values.

**Usage**

```
GOF_nash_sutcliffe_efficiency(mod, obs, j = 2, na.rm = TRUE)
```

**Arguments**

|       |  |
|-------|--|
| mod   | 'numeric' vector. Modeled or simulated values. Must be same length as obs.   |
| obs   | 'numeric' vector. Observed or comparison values. Must be same length as mod.   |
| j     | 'numeric' value. Exponent value for modified NSE (mNSE) equation. Default value is j = 2, which is traditional NSE equation.   |
| na.rm | 'boolean' TRUE or FALSE. Should NA values be removed before computing. If any NA values are present in mod or obs, the <i>i</i> th position from each will be removed before calculating. If NA values are present and na.rm = FALSE, then function will return NA. Default is TRUE. |

**Value**

Value of computed NSE or mNSE.

**References**

Krause, P., Boyle, D.P., and Base, F., 2005. Comparison of different efficiency criteria for hydrological model assessment: *Advances in Geosciences*, v. 5, p. 89-97. [Also available at <https://doi.org/10.5194/adgeo-5-89-2005>.]

Legates D.R and McCabe G.J., 1999, Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation: *Water Resources Research*. v. 35, no. 1, p. 233-241. [Also available at <https://doi.org/10.1029/1998WR900018>.]

Nash, J.E. and Sutcliffe, J.V., 1970, River flow forecasting through conceptual models part I: A discussion of principles: *Journal of Hydrology*, v. 10, no. 3, p. 282-290. [Also available at [https://doi.org/10.1016/0022-1694\(70\)90255-6](https://doi.org/10.1016/0022-1694(70)90255-6).]

**Examples**

```
GOF_nash_sutcliffe_efficiency(
  mod = example_mod$streamflow_cfs, obs = example_obs$streamflow_cfs
)
```

---

|                  |                                 |
|------------------|---------------------------------|
| GOF_percent_bias | <i>Calculates percent bias.</i> |
|------------------|---------------------------------|

---

**Description**

Calculates percent bias between modeled (simulated) and observed values.

**Usage**

```
GOF_percent_bias(mod, obs, na.rm = TRUE)
```

**Arguments**

|       |  |
|-------|--|
| mod   | 'numeric' vector. Modeled or simulated values. Must be same length as obs.   |
| obs   | 'numeric' vector. Observed or comparison values. Must be same length as mod.   |
| na.rm | 'boolean' TRUE or FALSE. Should NA values be removed before computing. If any NA values are present in mod or obs, the <i>i</i> th position from each will be removed before calculating. If NA values are present and na.rm = FALSE, then function will return NA. Default is TRUE. |

**Details**

Values returned are in percent.

**Value**

Value of calculated percent bias as percent.

**Examples**

```
GOF_percent_bias(mod = example_mod$streamflow_cfs, obs = example_obs$streamflow_cfs)
```

---

GOF\_rmse

*Calculate root-mean-square error with options to normalize*


---

**Description**

Calculate root-mean-square error (RMSE) between modeled (simulated) and observed values. Error is defined as modeled minus observed.

**Usage**

```
GOF_rmse(
  mod,
  obs,
  normalize = c("none", "mean", "range", "stdev", "iqr", "iqr-1", "iqr-2", "iqr-3",
    "iqr-4", "iqr-5", "iqr-6", "iqr-7", "iqr-8", "iqr-9", NULL),
  na.rm = TRUE
)
```

**Arguments**

|           |  |
|-----------|--|
| mod       | 'numeric' vector. Modeled or simulated values. Must be same length as obs.   |
| obs       | 'numeric' vector. Observed or comparison values. Must be same length as mod.   |
| normalize | 'character' value. Option to normalize the root-mean-square error (NRMSE) by several normalizing options. Default is 'none' (no normalizing). RMSE is returned.<br>'mean'. RMSE is normalized by the mean of obs.<br>'range'. RMSE is normalized by the range (max - min) of obs.<br>'stdev'. RMSE is normalized by the standard deviation of obs.<br>'iqr-#'. RMSE is normalized by the inter-quartile range of obs, with distribution type (see <a href="#">stats::quantile</a> function) indicated by integer number (for example "iqr-8"). If no type specified, default type is iqr-7, the quantile function default. |
| na.rm     | 'boolean' TRUE or FALSE. Should NA values be removed before computing. If any NA values are present in mod or obs, the <i>i</i> th position from each will be removed before calculating. If NA values are present and na.rm = FALSE, then function will return NA. Default is TRUE.   |

**Value**

'numeric' value of computed root-mean-square error (RMSE) or normalized root-mean-square error (NRMSE)

**Examples**

```
# RMSE
GOF_rmse(mod = example_mod$streamflow_cfs, obs = example_obs$streamflow_cfs)
# NRMSE
GOF_rmse(
  mod = example_mod$streamflow_cfs, obs = example_obs$streamflow_cfs, normalize = 'stdev'
)
```

GOF\_summary

*Calculate Goodness-of-fit metrics and output into table***Description**

Calculate Goodness-of-fit (GOF) metrics for correlation, Kling–Gupta efficiency, mean absolute error, mean error, Nash–Sutcliffe efficiency, percent bias, root-mean-square error, normalized root-mean-square error, and volumetric efficiency, and output into a table.

**Usage**

```
GOF_summary(
  mod,
  obs,
  metrics = c("cor", "kge", "mae", "me", "nse", "pb", "rmse", "nrmse", "ve"),
  censor_threshold = NULL,
  censor_symbol = NULL,
  na.rm = TRUE,
  kge_modified = FALSE,
  nse_j = 2,
  rmse_normalize = c("mean", "range", "stdev", "iqr", "iqr-1", "iqr-2", "iqr-3", "iqr-4",
    "iqr-5", "iqr-6", "iqr-7", "iqr-8", "iqr-9", NULL),
  ...
)
```

**Arguments**

**mod** 'numeric' vector. Modeled or simulated values. Must be same length as obs.

**obs** 'numeric' vector. Observed or comparison values. Must be same length as mod.

**metrics** 'character' vector. Which GOF metrics should be computed and output. Default is `c("cor", "kge", "mae", "me", "nse", "pb", "rmse", "nrmse", "ve")`.  
 "cor". Correlation tests computed from [GOF\\_correlation\\_tests](#).  
 "kge". Kling–Gupta efficiency computed from [GOF\\_kling\\_gupta\\_efficiency](#).

"mae". Mean absolute error computed from [GOF\\_mean\\_absolute\\_error](#).  
 "me". Mean error computed from [GOF\\_mean\\_error](#).  
 "nse". Nash–Sutcliffe efficiency computed from [GOF\\_nash\\_sutcliffe\\_efficiency](#) with option for modified NSE specified by parameter `nse_j`.  
 "pb". Percent bias computed from [GOF\\_percent\\_bias](#).  
 "rmse". Root-mean-square error computed from [GOF\\_rmse](#).  
 "nrmse". Normalized root-mean-square error computed from [GOF\\_rmse](#) and "normalize" option specified in parameter `rmse_normalize`.  
 "ve". Volumetric efficiency computed from [GOF\\_volumetric\\_efficiency](#).

|                                |  |
|--------------------------------|--|
| <code> censor_threshold</code> | 'numeric' value. Threshold to censor values on utilizing <a href="#">censor_values</a> function. Default is NULL, no censoring. If level specified, must also specify <code> censor_symbol</code> .  |
| <code> censor_symbol</code>    | 'character' string. Inequality symbol to censor values based on <code> censor_threshold</code> utilizing <a href="#">censor_values</a> function. Accepted values are "gt" (greater than), "gte" (greater than or equal to), "lt" (less than), or "lte" (less than or equal to). Default is NULL, no censoring. If symbol specified, must also specify <code> censor_value</code> .   |
| <code> na.rm</code>            | 'boolean' TRUE or FALSE. Should NA values be removed before computing. If any NA values are present in <code> mod</code> or <code> obs</code> , the <i>i</i> th position from each will be removed before calculating. If NA values are present and <code> na.rm = FALSE</code> , then function will return NA. Default is TRUE.   |
| <code> kge_modified</code>     | 'boolean' TRUE or FALSE. Should the KGE calculation use the original variability ratio in the standard deviations ( <code> kge_modified = FALSE</code> ) or the modified variability ratio in the coefficient of variations ( <code> kge_modified = TRUE</code> ). Default is FALSE.   |
| <code> nse_j</code>            | 'numeric' value. Exponent value for modified NSE (mNSE) equation, utilized if "nse" option is in parameter <code> metrics</code> . Default value is <code> nse_j = 2</code> , which is traditional NSE equation.   |
| <code> rmse_normalize</code>   | 'character' value. Normalize option for NRMSE, utilized if "nrmse" option is in parameter <code> metrics</code> . Default is "mean". Options are 'mean'. RMSE is normalized by the mean of <code> obs</code> . 'range'. RMSE is normalized by the range (max - min) of <code> obs</code> . 'stdev'. RMSE is normalized by the standard deviation of <code> obs</code> . 'iqr-#'. RMSE is normalized by the inter-quartile range of <code> obs</code> , with distribution type (see <a href="#">stats::quantile</a> function) indicated by integer number (for example "iqr-8"). If no type specified, default type is <code> iqr-7</code> , the quantile function default. |
| <code> ...</code>              | Further arguments to be passed to or from <a href="#">stats::cor.test</a> if "cor" is in <code> metrics</code> .   |

**Details**

See [GOF\\_correlation\\_tests](#), [GOF\\_kling\\_gupta\\_efficiency](#), [GOF\\_mean\\_absolute\\_error](#), [GOF\\_mean\\_error](#), [GOF\\_nash\\_sutcliffe\\_efficiency](#), [GOF\\_percent\\_bias](#), [GOF\\_rmse](#), and [GOF\\_volumetric\\_efficiency](#).

**Value**

A tibble (see [tibble::tibble](#)) with GOF metrics

**See Also**

[censor\\_values](#), [GOF\\_correlation\\_tests](#), [GOF\\_kling\\_gupta\\_efficiency](#), [GOF\\_mean\\_absolute\\_error](#), [GOF\\_mean\\_error](#), [GOF\\_nash\\_sutcliffe\\_efficiency](#), [GOF\\_percent\\_bias](#), [GOF\\_rmse](#), [GOF\\_volumetric\\_efficiency](#)

**Examples**

```
GOF_summary(mod = example_mod$streamflow_cfs, obs = example_obs$streamflow_cfs)
```

---

GOF\_volumetric\_efficiency

*Calculate Volumetric Efficiency*

---

**Description**

Calculate Volumetric efficiency (VE) between modeled (simulated) and observed values. VE is defined as the fraction of water delivered at the proper time (Criss and Winston, 2008).

**Usage**

```
GOF_volumetric_efficiency(mod, obs, na.rm = TRUE)
```

**Arguments**

|                    |   |
|--------------------|---|
| <code>mod</code>   | 'numeric' vector. Modeled or simulated values. Must be same length as <code>obs</code> .  |
| <code>obs</code>   | 'numeric' vector. Observed or comparison values. Must be same length as <code>mod</code> .  |
| <code>na.rm</code> | 'boolean' TRUE or FALSE. Should NA values be removed before computing. If any NA values are present in <code>mod</code> or <code>obs</code> , the <i>i</i> th position from each will be removed before calculating. If NA values are present and <code>na.rm = FALSE</code> , then function will return NA. Default is TRUE. |

**Details**

Volumetric efficiency was proposed in order to circumvent some problems associated to the Nash–Sutcliffe efficiency. It ranges from 0 to 1 and represents the fraction of water delivered at the proper time; its complement represents the fractional volumetric mismatch (Criss and Winston, 2008).

**Value**

Value of computed Volumetric efficiency.

**References**

Criss, R.E. and Winston, W.E., 2008, Do Nash values have value? Discussion and alternate proposals: *Hydrological Processes*, v. 22, p. 2723-2725.

[Also available at <https://doi.org/10.1002/hyp.7072>.]

Zambrano-Bigiarini, M., 2020, hydroGOF: Goodness-of-fit functions for comparison of simulated and observed hydrological time series R package version 0.4-0. accessed September 16, 2020, at <https://github.com/hzambran/hydroGOF>. [Also available at <https://doi.org/10.5281/zenodo.839854>.]

**Examples**

```
GOF_volumetric_efficiency(
  mod = example_mod$streamflow_cfs, obs = example_obs$streamflow_cfs
)
```

---

POR\_apply\_annual\_hiflow\_stats

*Calculate the 50th and 90th percentiles of a streamflow time series*

---

**Description**

This function computes the 50th and 90th percentiles of a streamflow time series from annual n-day high flow values and returns a data.frame in the format of other period-of-record (POR) metrics.

**Usage**

```
POR_apply_annual_hiflow_stats(annual_max, quantile_type = 8)
```

**Arguments**

|               |  |
|---------------|--|
| annual_max    | 'numeric' vector or data.frame. Vector or data.frame with columns of annual n-day maximum streamflows.   |
| quantile_type | 'numeric' value. The distribution type used in the <code>stats::quantile</code> function. Default is 8 (median-unbiased regardless of distribution). Other types common in hydrology are 6 (Weibull) or 9 (unbiased for normal distributions). |

**Details**

annual maximum of n-day moving averages can be computed during pre-processing step using [preproc\\_precondition\\_data](#) and [calc\\_annual\\_flow\\_stats](#), or [preproc\\_main](#) for both observed and modeled data.

**Value**

Data.frame of 0.5 and 0.9 non-exceedance probabilities (50th and 90th percentiles), with metric names if annual\_max is a data.frame with columns named by metric.

**See Also**

[quantile](#), [preproc\\_precondition\\_data](#), [calc\\_annual\\_flow\\_stats](#), [preproc\\_main](#)

**Examples**

```
POR_apply_annual_hiflow_stats(annual_max = example_annual[ , c("high_q1", "high_q30")])
```

---

```
POR_apply_annual_lowflow_stats
```

*Calculate 10-year and 2-year return periods of a streamflow time series*

---

**Description**

Calculates 10-year and 2-year return periods of a streamflow time series from annual n-day low streamflow values and returns a data.frame in the format of other period-of-record (POR) metrics.

**Usage**

```
POR_apply_annual_lowflow_stats(annual_min)
```

**Arguments**

`annual_min` 'numeric' vector or data.frame. Vector or data.frame with columns of annual n-day minimum streamflows.

**Details**

`POR_apply_POR_lowflow_metrics` is a helper function that applies the [POR\\_calc\\_lp3\\_quantile](#) function to the data.frame of n-day moving averages, which can be computed during pre-processing step using [preproc\\_precondition\\_data](#) and [calc\\_annual\\_flow\\_stats](#), or [preproc\\_main](#) for both observed and modeled data. This function returns a data.frame with the 10-year and 2-year return period streamflows for each n-day low streamflow in the input data.frame.

**Value**

data.frame with 10-year and 2-year return period of n-day streamflows.

**See Also**

[POR\\_calc\\_lp3\\_quantile](#), [preproc\\_precondition\\_data](#), [calc\\_annual\\_flow\\_stats](#),  
[preproc\\_main](#)

**Examples**

```
POR_apply_annual_lowflow_stats(annual_min = example_annual[ , c("low_q1", "low_q30")])
```

---

```
POR_calc_amp_and_phase
```

*Calculate the seasonal amplitude and phase of a daily time series*

---

**Description**

Calculates the seasonal amplitude and phase of a daily time series.

**Usage**

```
POR_calc_amp_and_phase(  
  data = NULL,  
  Date,  
  value,  
  time_step = c("daily", "monthly")  
)
```

**Arguments**

|           |   |
|-----------|---|
| data      | 'data.frame'. Optional data.frame input, with columns containing Date and value. Column names are specified as strings in the corresponding parameter. Default is NULL.     |
| Date      | 'numeric' vector of Dates corresponding to each value when data = NULL, or 'character' string identifying Date column name when data is specified.                          |
| value     | 'numeric' vector of values (often streamflow) when data = NULL, or 'character' string identifying value column name when data is specified. Assumed to be daily or monthly. |
| time_step | 'character' value. Either "daily" or "monthly", Default is "daily".   |

**Value**

A data.frame with calculated seasonal amplitude and phase

**References**

Farmer, W.H., Archfield, S.A., Over, T.M., Hay, L.E., LaFontaine, J.H., and Kiang, J.E., 2014, A comparison of methods to predict historical daily streamflow time series in the southeastern United States: U.S. Geological Survey Scientific Investigations Report 2014–5231, 34 p. [Also available at <https://doi.org/10.3133/sir20145231>.]

**Examples**

```
POR_calc_amp_and_phase(data = example_obs, Date = "Date", value = "streamflow_cfs")
```

---

|              |   |
|--------------|---|
| POR_calc_AR1 | <i>calculates lag-one autocorrelation (AR1) coefficient for a time series</i> |
|--------------|---|

---

**Description**

calculates lag-one autocorrelation (AR1) coefficient for a time series

**Usage**

```
POR_calc_AR1(data = NULL, Date, value, time_step = c("daily", "monthly"))
```

**Arguments**

|           |   |
|-----------|---|
| data      | 'data.frame'. Optional data.frame input, with columns containing Date and value. Column names are specified as strings in the corresponding parameter. Default is NULL.     |
| Date      | 'numeric' vector of Dates corresponding to each value when data = NULL, or 'character' string identifying Date column name when data is specified.                          |
| value     | 'numeric' vector of values (often streamflow) when data = NULL, or 'character' string identifying value column name when data is specified. Assumed to be daily or monthly. |
| time_step | 'character' value. Either "daily" or "monthly".   |

**Details**

The function calculates lag-one autocorrelation (AR1) coefficient for a time series using the [stats::ar](#) function. When applied to an observed or modeled time series of streamflow, the [POR\\_deseasonalize](#) function can be applied to the raw data prior to running the POR\_calc\_AR1 function.

**Value**

A data.frame with calculated seasonal amplitude and phase.

**References**

Farmer, W.H., Archfield, S.A., Over, T.M., Hay, L.E., LaFontaine, J.H., and Kiang, J.E., 2014, A comparison of methods to predict historical daily streamflow time series in the southeastern United States: U.S. Geological Survey Scientific Investigations Report 2014–5231, 34 p. [Also available at <https://doi.org/10.3133/sir20145231>.]

**See Also**

[POR\\_deseasonalize](#), [ar](#)

**Examples**

```
POR_calc_AR1(data = example_obs, Date = "Date", value = "streamflow_cfs")
```

---

POR\_calc\_lp3\_quantile *Calculate quantile from fitted log-Pearson type III distribution*

---

**Description**

Calculate the specified flow quantile from a fitted log-Pearson type III distribution from a time series of n-day low flows.

**Usage**

```
POR_calc_lp3_quantile(annual_min, p)
```

**Arguments**

|            |  |
|------------|--|
| annual_min | 'numeric' vector. Vector of minimum annual n-day mean flows.   |
| p          | 'numeric' value of exceedance probabilities. Quantile of fitted distribution that is returned ( $p=0.1$ for 10-year return period, $p=0.5$ for 2-year return period) |

**Details**

POR\_calc\_lp3\_quantile fits an log-Pearson type III distribution to a series of annual n-day flows and returns the quantile of a user-specified probability using [calc\\_qlpearsonIII](#). This represents a theoretical return period for than n-day flow.

**Value**

Specified quantile from the fitted log-Pearson type 3 distribution.

**References**

Asquith, W.H., Kiang, J.E., and Cohn, T.A., 2017, Application of at-site peak-streamflow frequency analyses for very low annual exceedance probabilities: U.S. Geological Survey Scientific Investigation Report 2017–5038, 93 p. [Also available at <https://doi.org/10.3133/sir20175038>.]

**See Also**

[calc\\_qlpearsonIII](#)

## Examples

```
POR_calc_lp3_quantile(annual_min = example_annual$low_q1, p = 0.1)
```

---

|                   |   |
|-------------------|---|
| POR_deseasonalize | <i>Removes seasonal trends from a daily or monthly time series.</i> |
|-------------------|---|

---

## Description

Removes seasonal trends from a daily or monthly time series. Daily data are deseasonalized by subtracting monthly mean values. Monthly data are deseasonalized by subtracting mean monthly values.

## Usage

```
POR_deseasonalize(data = NULL, Date, value, time_step = c("daily", "monthly"))
```

## Arguments

|           |   |
|-----------|---|
| data      | 'data.frame'. Optional data.frame input, with columns containing Date and value. Column names are specified as strings in the corresponding parameter. Default is NULL.       |
| Date      | 'numeric' vector of Dates corresponding to each value when data = NULL, or 'character' string identifying Date column name when data is specified.                            |
| value     | 'numeric' vector of values (often streamflow) when data = NULL, or 'character' string identifying value column name when data is specified. (assumed to be daily or monthly). |
| time_step | 'character' value. Either "daily" or "monthly".   |

## Details

The deseasonalize function removes seasonal trends from a daily or monthly time series and returns a deseasonalized time series, which can be used in the [POR\\_calc\\_AR1](#) function.

## Value

Deseasonalized values.

## See Also

[POR\\_calc\\_AR1](#)

## Examples

```
POR_deseasonalize(data = example_obs, Date = "Date", value = "streamflow_cfs")
```

---

POR\_distribution\_metrics

*Calculates various metrics that describe the distribution of a time series of streamflow*

---

## Description

Calculates various metrics that describe the distribution of a time series of streamflow, which can be of any time step.

## Usage

```
POR_distribution_metrics(value, quantile_type = 8, na.rm = TRUE)
```

## Arguments

|               |  |
|---------------|--|
| value         | 'numeric' vector of values (assumed to be streamflow) at any time step.  |
| quantile_type | 'numeric' value. The distribution type used in the <code>stats::quantile</code> function. Default is 8 (median-unbiased regardless of distribution). Other types common in hydrology are 6 (Weibull) or 9 (unbiased for normal distributions). |
| na.rm         | 'boolean' TRUE or FALSE. Should NA values be removed before computing. If NA values are present and <code>na.rm = FALSE</code> , then function will return NAs. Default is TRUE.   |

## Details

Metrics computed include:

`p_n` Flow-duration curve (FDC) percentile where  $n = 1, 5, 10, 25, 50, 75, 90, 95,$  and 99

`POR_mean` Period of record mean

`POR_sd` Period of record standard deviation

`POR_cv` Period of record coefficient of variation

`POR_min` Period of record minimum

`POR_max` Period of record maximum

`LCV` L-moment coefficient of variation

`Lskew` L-moment skewness

`Lkurtosis` L-moment kurtosis

## Value

A data.frame with FDC quantiles, and distribution metrics. See **Details**. This function calculates various metrics that describe the distribution of a time series of streamflow, which can be of any time step.

## References

Farmer, W.H., Archfield, S.A., Over, T.M., Hay, L.E., LaFontaine, J.H., and Kiang, J.E., 2014, A comparison of methods to predict historical daily streamflow time series in the southeastern United States: U.S. Geological Survey Scientific Investigations Report 2014–5231, 34 p. [Also available at <https://doi.org/10.3133/sir20145231>.]

Asquith, W.H., Kiang, J.E., and Cohn, T.A., 2017, Application of at-site peak-streamflow frequency analyses for very low annual exceedance probabilities: U.S. Geological Survey Scientific Investigation Report 2017–5038, 93 p. [Also available at <https://doi.org/10.3133/sir20175038>.]

Asquith, W.H., 2021, lmomco—L-moments, censored L-moments, trimmed L-moments, L-comoments, and many distributions. R package version 2.3.7, Texas Tech University, Lubbock, Texas.

## See Also

[lmoms](#), [quantile](#)

## Examples

```
POR_distribution_metrics(value = example_obs$streamflow_cfs)
```

---

|                    |  |
|--------------------|--|
| preproc_audit_data | <i>Audit daily data for total days in year</i> |
|--------------------|--|

---

## Description

Audit daily data for total days in year. An audit is performed to inventory and flag missing days in daily data and help determine if further analyses are appropriate.

## Usage

```
preproc_audit_data(  
  data = NULL,  
  Date,  
  value,  
  year_group,  
  use_specific_years = FALSE,  
  begin_year = NULL,  
  end_year = NULL,  
  days_cutoff = 360,  
  date_format = "%Y-%m-%d"  
)
```

**Arguments**

|                    |  |
|--------------------|--|
| data               | 'data.frame'. Optional data.frame input, with columns containing Date and value. Column names are specified as strings in the corresponding parameter. Default is NULL.                                |
| Date               | 'Date' or 'character' vector when data = NULL, or 'character' string identifying Date column name when data is specified. Dates associated with each value in value parameter.                         |
| value              | 'numeric' vector when data = NULL, or 'character' string identifying year column name when data is specified. Values to audit, must be daily data.   |
| year_group         | 'numeric' vector when data = NULL, or 'character' string identifying grouping column name when data is specified. Year grouping for each daily value in value parameter. Must be same length as value. |
| use_specific_years | 'boolean' value. Flag to clip data to a certain set of years in year_group. Default is FALSE.  |
| begin_year         | 'numeric' value. If use_specific_years = TRUE, beginning year to clip value. Default is NULL.  |
| end_year           | 'numeric' value. If use_specific_years = TRUE, ending year to clip value. Default is NULL.   |
| days_cutoff        | 'numeric' value. Designating the number of days required for a year to be counted as full. Default is 360.   |
| date_format        | 'character' string. Format of Date. Default is "%Y-%m-%d".   |

**Details**

Year grouping is commonly water year, climate year, or calendar year.

**Value**

A data.frame with year\_group, count (n, excluding NA values) of days in each year\_group, and a complete years 'boolean' flag.

**See Also**

[preproc\\_fill\\_daily](#), [preproc\\_precondition\\_data](#)

**Examples**

```
preproc_audit_data(
  data = example_preproc, Date = "Date", value = "value", year_group = "WY"
)
```

---

preproc\_fill\_daily      *Fills daily data with missing dates as NA values*

---

### Description

Fills daily data with missing dates as NA values. Days that are absent from the daily time series are inserted with a corresponding value of NA.

### Usage

```
preproc_fill_daily(  
  data = NULL,  
  Date,  
  value,  
  POR_start = NA,  
  POR_end = NA,  
  date_format = "%Y-%m-%d"  
)
```

### Arguments

|             |   |
|-------------|---|
| data        | 'data.frame'. Optional data.frame input, with columns containing Date and value. Column names are specified as strings in the corresponding parameter. Default is NULL.       |
| Date        | 'Date' or 'character' vector when data = NULL, or 'character' string identifying Date column name when data is specified. Date associated with each value in value parameter. |
| value       | 'numeric' vector when data = NULL, or 'character' string identifying values column name when data is specified.   |
| POR_start   | 'character' value. Optional period of record start. If not specified, defaults to min(Date).  |
| POR_end     | 'character' value. Optional period of record end. If not specified, defaults to max(Date).  |
| date_format | 'character' string. Format of Date. Default is "%Y-%m-%d".  |

### Details

Can be used prior to [preproc\\_precondition\\_data](#) to fill daily data before computation of n-day moving averages, or prior to [preproc\\_audit\\_data](#).

### Value

A data.frame with Date and value, sequenced from POR\_start to POR\_end by 1 day.

### See Also

[preproc\\_audit\\_data](#), [preproc\\_precondition\\_data](#)

**Examples**

```
Dates = c(seq.Date(as.Date("2020-01-01"), as.Date("2020-01-10"), by = "1 day"),
          seq.Date(as.Date("2020-01-20"), as.Date("2020-01-31"), by = "1 day"))
values = c(seq.int(1, 22, 1))
preproc_fill_daily(Date = Dates, value = values)
```

---

|              |   |
|--------------|---|
| preproc_main | <i>A wrapper function for preproc_precondition_data, preproc_audit_data, and calc_annual_flow_stats</i> |
|--------------|---|

---

**Description**

A wrapper function for [preproc\\_precondition\\_data](#), [preproc\\_audit\\_data](#), and [calc\\_annual\\_flow\\_stats](#)

**Usage**

```
preproc_main(
  data = NULL,
  Date,
  value,
  date_format = "%Y-%m-%d",
  year_group = c("WY", "CY", "year"),
  use_specific_years = FALSE,
  begin_year = NULL,
  end_year = NULL,
  days_cutoff = 360,
  calc_high = TRUE,
  calc_low = TRUE,
  calc_percentiles = TRUE,
  calc_monthly = TRUE,
  calc_WSCVD = TRUE,
  longitude = NA,
  calc_ICVD = FALSE,
  zero_threshold = 33,
  quantile_type = 8,
  na.action = c("na.omit", "na.pass")
)
```

**Arguments**

|      |   |
|------|---|
| data | 'data.frame'. Optional data.frame input, with columns containing Date and value. Column names are specified as strings in the corresponding parameter. Default is NULL. |
|------|---|

|                    |  |
|--------------------|--|
| Date               | 'Date' or 'character' vector when data = NULL, or 'character' string identifying Date column name when data is specified. Dates associated with each value in value parameter.   |
| value              | 'numeric' vector when data = NULL, or 'character' string identifying year column name when data is specified. Values to precondition and calculate n-day moving averages from. N-day moving averages only calculated for daily data.   |
| date_format        | 'character' string. Format of Date. Default is "%Y-%m-%d".   |
| year_group         | 'character' value. Specify either "year" for calendar year, "WY" for water year, or "CY" for climate year. Used to select data after preconditioning for audit and annual statistics. Default is "WY".   |
| use_specific_years | 'boolean' value. Flag to clip data to a certain set of years in year_group. Default is FALSE.  |
| begin_year         | 'numeric' value. If use_specific_years = TRUE, beginning year to clip value. Default is NULL.  |
| end_year           | 'numeric' value. If use_specific_years = TRUE, ending year to clip value. Default is NULL.   |
| days_cutoff        | 'numeric' value. Designating the number of days required for a year to be counted as full. Default is 360.   |
| calc_high          | 'boolean' value. Calculate high streamflow statistics for years in year_group. Default is TRUE. See <b>Details</b> for more information.   |
| calc_low           | 'boolean' value. Calculate low streamflow statistics for years in year_group. Default is TRUE. See <b>Details</b> for more information.  |
| calc_percentiles   | 'boolean' value. Calculate percentiles for years in year_group. Default is TRUE. See <b>Details</b> for more information.  |
| calc_monthly       | 'boolean' value. Calculate monthly statistics for years in year_group. Default is TRUE. See <b>Details</b> for more information.   |
| calc_WSCVD         | 'boolean' value. Calculate winter-spring center volume date for years in year_group. Default is TRUE. See <b>Details</b> for more information.   |
| longitude          | 'numeric' value. Site longitude in NAD83, required in WSCVD calculation. Default is NA. See <b>Details</b> for more information.   |
| calc_ICVD          | 'boolean' value. Calculate inverse center volume date for years in year_group. Default is FALSE. See <b>Details</b> for more information.  |
| zero_threshold     | 'numeric' value as percentage. The percentage of years of a statistic that need to be zero in order for it to be deemed a zero streamflow site for that statistic. For use in trend calculation. See <b>Details</b> on attributes. Default is 33 (33 percent) of the annual statistic values.    |
| quantile_type      | 'numeric' value. The distribution type used in the <code>stats::quantile</code> function. Default is 8 (median-unbiased regardless of distribution). Other types common in hydrology are 6 (Weibull) or 9 (unbiased for normal distributions).   |
| na.action          | 'character' string indicating na.action passed to <code>stats::aggregate</code> na.action parameter. Default is "na.omit", which removes NA values before aggregating statistics, or "na.pass", which will pass NA values and return NA in the grouped calculation if any NA values are present. |

**Details**

This is a wrapper function of [preproc\\_precondition\\_data](#), [preproc\\_audit\\_data](#), and [calc\\_annual\\_flow\\_stats](#). Data are first passed to the precondition function, then audited, then annual statistics are computed.

It also checks the timestep of the data to make sure that it is daily timestep. Other time steps are currently not supported and will return the data.frame without moving averages computed.

**Value**

A list of three data.frames: 1 of preconditioned data, 1 data audit, and 1 annual statistics.

**See Also**

[preproc\\_audit\\_data](#), [preproc\\_precondition\\_data](#), [calc\\_annual\\_flow\\_stats](#)

**Examples**

```
preproc_main(data = example_obs, Date = "Date", value = "streamflow_cfs", longitude = -68)
```

---

```
preproc_precondition_data
```

*Pre-conditions data with time information and n-day moving averages*

---

**Description**

Pre-conditions data with time information and n-day moving averages, with options to fill missing days with NA values.

**Usage**

```
preproc_precondition_data(  
  data = NULL,  
  Date,  
  value,  
  date_format = "%Y-%m-%d",  
  fill_daily = TRUE  
)
```

**Arguments**

|      |  |
|------|--|
| data | 'data.frame'. Optional data.frame input, with columns containing Date and value. Column names are specified as strings in the corresponding parameter. Default is NULL.        |
| Date | 'Date' or 'character' vector when data = NULL, or 'character' string identifying Date column name when data is specified. Dates associated with each value in value parameter. |

|             |  |
|-------------|--|
| value       | 'numeric' vector when data = NULL, or 'character' string identifying year column name when data is specified. Values to precondition and calculate n-day moving averages from. N-day moving averages only calculated for daily data. |
| date_format | 'character' string. Format of Date. Default is "%Y-%m-%d".   |
| fill_daily  | 'logical' value. Should gaps in Date and value be filled using <a href="#">preproc_fill_daily</a> . Default is TRUE.   |

## Details

These columns are added to the data:

year

month

day

decimal\_date

WY Water Year: October 1 to September 30

CY Climate Year: April 1 to March 30

Q3 3-Day Moving Average: computed at end of moving interval

Q7 7-Day Moving Average: computed at end of moving interval

Q30 30-Day Moving Average: computed at end of moving interval

jd Julian date

This function also checks the time step of the data to make sure that it is daily time step. Daily values with gaps are important to fill with NA to ensure proper calculation of n-day moving averages. Use `fill_daily = TRUE` or [preproc\\_fill\\_daily](#). Other time steps are currently not supported and will return the data.frame without moving averages computed.

## Value

A data.frame with Date, value, and additional columns with time and n-day moving average information.

## See Also

[preproc\\_fill\\_daily](#), [rollmean](#)

## Examples

```
preproc_precondition_data(data = example_obs, Date = "Date", value = "streamflow_cfs")
```

---

`preproc_validate_daily`*Validates that daily data do not contain gaps*

---

### Description

Validates that daily data do not contain gaps

### Usage

```
preproc_validate_daily(  
  data = NULL,  
  Date = "Date",  
  value = "value",  
  date_format = "%Y-%m-%d"  
)
```

### Arguments

|                          |   |
|--------------------------|---|
| <code>data</code>        | 'data.frame'. Optional data.frame input, with columns containing <code>Date</code> and <code>value</code> . Column names are specified as strings in the corresponding parameter. Default is <code>NULL</code> .  |
| <code>Date</code>        | 'Date' or 'character' vector when <code>data = NULL</code> , or 'character' string identifying <code>Date</code> column name when <code>data</code> is specified. Dates associated with each value in <code>value</code> parameter.                             |
| <code>value</code>       | 'numeric' vector when <code>data = NULL</code> , or 'character' string identifying year column name when <code>data</code> is specified. Values to precondition and calculate n-day moving averages from. N-day moving averages only calculated for daily data. |
| <code>date_format</code> | 'character' string. Format of <code>Date</code> . Default is "%Y-%m-%d".  |

### Details

Used to validate there are no gaps in the daily record before computing n-day moving averages in [preproc\\_precondition\\_data](#) or lag-1 autocorrelation in [POR\\_calc\\_AR1](#). If gaps are present, [preproc\\_fill\\_daily](#) can be used to fill them with NA values.

### Value

An error message with missing dates, otherwise nothing.

### Examples

```
preproc_validate_daily(data = example_obs, Date = "Date", value = "streamflow_cfs")
```

# Index

- \* **annual-statistics**
    - calc\_annual\_flow\_stats, 4
    - calc\_annual\_stat\_trend, 7
    - calc\_logistic\_regression, 9
    - POR\_apply\_annual\_hiflow\_stats, 30
    - POR\_apply\_annual\_lowflow\_stats, 31
    - POR\_calc\_lp3\_quantile, 34
    - preproc\_main, 40
  - \* **benchmark**
    - benchmark\_KGE\_DOY, 3
  - \* **datasets**
    - example\_annual, 13
    - example\_mod, 16
    - example\_mod\_zf, 17
    - example\_obs, 18
    - example\_obs\_zf, 19
    - example\_preproc, 20
  - \* **goodness-of-fit**
    - benchmark\_KGE\_DOY, 3
    - GOF\_correlation\_tests, 21
    - GOF\_kling\_gupta\_efficiency, 22
    - GOF\_mean\_absolute\_error, 23
    - GOF\_mean\_error, 23
    - GOF\_nash\_sutcliffe\_efficiency, 24
    - GOF\_percent\_bias, 25
    - GOF\_rmse, 26
    - GOF\_summary, 27
    - GOF\_volumetric\_efficiency, 29
  - \* **package**
    - HyMETT-package, 3
  - \* **period-of-record**
    - calc\_qlpearsonIII, 10
    - calc\_qpearsonIII, 11
    - POR\_apply\_annual\_hiflow\_stats, 30
    - POR\_apply\_annual\_lowflow\_stats, 31
    - POR\_calc\_amp\_and\_phase, 32
    - POR\_calc\_AR1, 33
    - POR\_calc\_lp3\_quantile, 34
    - POR\_deseasonalize, 35
    - POR\_distribution\_metrics, 36
  - \* **preprocessing-data**
    - sensor\_values, 12
    - POR\_deseasonalize, 35
    - preproc\_audit\_data, 37
    - preproc\_fill\_daily, 39
    - preproc\_main, 40
    - preproc\_precondition\_data, 42
    - preproc\_validate\_daily, 44
  - \* **trends**
    - calc\_annual\_stat\_trend, 7
    - calc\_logistic\_regression, 9
- \_PACKAGE (HyMETT-package), 3
- ar, 34
- benchmark\_KGE\_DOY, 3
- calc\_annual\_flow\_stats, 4, 31, 32, 40, 42
- calc\_annual\_stat\_trend, 7
- calc\_logistic\_regression, 9
- calc\_qlpearsonIII, 10, 10, 11, 34
- calc\_qpearsonIII, 10, 11, 11
- sensor\_values, 12, 28, 29
- cor.test, 21
- EnvStats::kendallTrendTest, 8
- example\_annual, 13
- example\_mod, 16
- example\_mod\_zf, 17
- example\_obs, 15, 18, 20
- example\_obs\_zf, 19
- example\_preproc, 20
- glm, 10
- GOF\_correlation\_tests, 21, 27, 29
- GOF\_kling\_gupta\_efficiency, 3, 22, 27, 29
- GOF\_mean\_absolute\_error, 23, 28, 29
- GOF\_mean\_error, 23, 28, 29
- GOF\_nash\_sutcliffe\_efficiency, 24, 28, 29

GOF\_percent\_bias, [25](#), [28](#), [29](#)  
GOF\_rmse, [26](#), [28](#), [29](#)  
GOF\_summary, [27](#)  
GOF\_volumetric\_efficiency, [28](#), [29](#), [29](#)

HyMETT (HyMETT-package), [3](#)  
HyMETT-package, [3](#)

kendallTrendTest, [8](#)

lmoms, [37](#)

month.abb, [6](#)

POR\_apply\_annual\_hiflow\_stats, [30](#)  
POR\_apply\_annual\_lowflow\_stats, [31](#)  
POR\_calc\_amp\_and\_phase, [32](#)  
POR\_calc\_AR1, [33](#), [35](#), [44](#)  
POR\_calc\_lp3\_quantile, [31](#), [32](#), [34](#)  
POR\_deseasonalize, [33](#), [34](#), [35](#)  
POR\_distribution\_metrics, [36](#)  
preproc\_audit\_data, [37](#), [39](#), [40](#), [42](#)  
preproc\_fill\_daily, [38](#), [39](#), [43](#), [44](#)  
preproc\_main, [3](#), [15](#), [20](#), [31](#), [32](#), [40](#)  
preproc\_precondition\_data, [3](#), [5](#), [7](#), [31](#), [32](#),  
[38–40](#), [42](#), [42](#), [44](#)  
preproc\_validate\_daily, [44](#)

quantile, [31](#), [37](#)

rollmean, [43](#)

stats::aggregate, [6](#), [41](#)  
stats::ar, [33](#)  
stats::cor.test, [21](#), [28](#)  
stats::glm, [9](#)  
stats::quantile, [6](#), [26](#), [28](#), [30](#), [36](#), [41](#)

tibble::tibble, [7–9](#), [21](#), [29](#)