Package 'HyMETT'

November 22, 2023

Type Package

Title Hydrologic Model Evaluation and Time-Series Tools

Version 1.1.2

Date 2023-11-21

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

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

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

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

Maintainer Colin Penn <cpenn@usgs.gov>

Repository CRAN

Date/Publication 2023-11-22 19:20:02 UTC

R topics documented:

`HyMETT-package` *Hydrologic Model Evaluation and Time-series Tools*

Description

This package 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/P9FNXEWI for more details.

`benchmark_KGE_DOY` *Calculate benchmark Kling–Gupta efficiency (KGE) values from day-of-year (DOY) observations*

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 preproc_precondition_data. 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 preproc_precondition_data. 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 preproc_precondition_data. 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 preproc_precondition_data. 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 **Details** for more information.
calc_low	'boolean' value. Calculate low flow statistics for years in year_group. Default is FALSE. See **Details** for more information.
calc_percentiles	'boolean' value. Calculate percentiles for years in year_group. Default is FALSE. See **Details** for more information.
calc_monthly	'boolean' value. Calculate monthly statistics for years in year_group. Default is FALSE. See **Details** for more information.
calc_WSCVD	'boolean' value. Calculate winter-spring center volume date for years in year_group. Default is FALSE. See **Details** for more information.
longitude	'numeric' value. Site longitude in North American Datum of 1983 (NAD83), required in WSCVD calculation. Default is NA. See **Details** for more information.
calc_ICVD	'boolean' value. Calculate inverse center volume date for years in year_group. Default is FALSE. See **Details** 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 **Details** on attributes. Default is `33` (33 percent) of the annual statistic values.

`quantile_type` 'numeric' value. The distribution type used in the `stats::quantile` 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 `stats::aggregate` `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_q`***n*** where n = 1, 3, 7, and 30

`high_q`***n***`_jd` where n = 1, 3, 7, and 30

`low_q`***n*** where n = 1, 3, 7, and 30

`low_q`***n***`_jd` where n = 1, 3, 7, 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, 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 EnvStats::kendallTrendTest.

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 sen_slope * (max(year) − min(year))

val_beg/end Calculated value at beginning or end of period, calculated as sen_slope * year + intercept

val_perc_change Percentage change over period, calculated as (val_end - val_beg) / 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 `stats::glm`.

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 `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, 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 *Quantile of Pearson Type III distribution*

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, 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.]

Examples

```
calc_qpearsonIII(0.1)
```

censor_values *Censor values above or below a threshold*

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 <= 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 *Example Annual Observations*

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^3/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 *Example Model Output with zero flows*

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^3/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^3/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.]

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^3/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.]

Examples

```
str(example_obs_zf)
```

example_preproc *Example Observations prepocessed*

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

`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*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`
`...`	Further arguments to be passed to or from `stats::cor.test`.

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*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

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*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

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*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*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.]

Examples

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

GOF_percent_bias *Calculates percent bias.*

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*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. `'mean'`. RMSE is normalized by the mean of `obs`. `'range'`. RMSE is normalized by the range (`max - min`) of `obs`. `'stdev'`. RMSE is normalized by the standard deviation of `obs`. `'iqr-#'`. RMSE is normalized by the inter-quartile range of `obs`, with distribution type (see `stats::quantile` 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*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.

censor_threshold
: 'numeric' value. Threshold to censor values on utilizing censor_values function. Default is NULL, no censoring. If level specified, must also specify censor_symbol.

censor_symbol
: 'character' string. Inequality symbol to censor values based on censor_threshold utilizing censor_values 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 censor_value.

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*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.

kge_modified
: 'boolean' TRUE or FALSE. Should the KGE calculation use the original variability ratio in the standard deviations (kge_modified = FALSE) or the modified variability ratio in the coefficient of variations (kge_modified = TRUE). Default is FALSE.

nse_j
: 'numeric' value. Exponent value for modified NSE (mNSE) equation, utilized if "nse" option is in parameter metrics. Default value is nse_j = 2, which is traditional NSE equation.

rmse_normalize
: 'character' value. Normalize option for NRMSE, utilized if "nrmse" option is in paramter metrics. Default is "mean". Options are
'mean'. RMSE is normalized by the mean of obs.
'range'. RMSE is normalized by the range (max - min) of obs.
'stdev'. RMSE is normalized by the standard deviation of obs.
'iqr-#'. RMSE is normalized by the inter-quartile range of obs, with distribution type (see stats::quantile function) indicated by integer number (for example "iqr-8"). If no type specified, default type is iqr-7, the quantile function default.

...
: Further arguments to be passed to or from stats::cor.test if "cor" is in metrics.

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

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*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

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 compliment 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 `stats::quantile` 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` *calculates lag-one autocorrelation (AR1) coefficient for a time series*

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` *Removes seasonal trends from a daily or monthly time series.*

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 `stats::quantile` 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 `na.rm = FALSE`, then function will return `NA`s. 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` *Audit daily data for total days in year*

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` *A wrapper function for preproc_precondition_data, preproc_audit_data, and calc_annual_flow_stats*

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 **Details** for more information.

`calc_low` 'boolean' value. Calculate low streamflow statistics for years in `year_group`. Default is `TRUE`. See **Details** for more information.

`calc_percentiles` 'boolean' value. Calculate percentiles for years in `year_group`. Default is `TRUE`. See **Details** for more information.

`calc_monthly` 'boolean' value. Calculate monthly statistics for years in `year_group`. Default is `TRUE`. See **Details** for more information.

`calc_WSCVD` 'boolean' value. Calculate winter-spring center volume date for years in `year_group`. Default is `TRUE`. See **Details** for more information.

`longitude` 'numeric' value. Site longitude in NAD83, required in WSCVD calculation. Default is `NA`. See **Details** for more information.

`calc_ICVD` 'boolean' value. Calculate inverse center volume date for years in `year_group`. Default is `FALSE`. See **Details** 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 **Details** on attributes. Default is `33` (33 percent) of the annual statistic values.

`quantile_type` 'numeric' value. The distribution type used in the `stats::quantile` 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 `stats::aggregate` `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 preproc_fill_daily. 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

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".

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