Package ‘ffaframework’

June 12, 2025

Title Flood Frequency Analysis Framework

Version 0.1.0

Author Riley Wheadon [aut, cre]

Maintainer Riley Wheadon `<rileywheadon@gmail.com>`

Description Tools for exploratory data analysis and flood frequency analysis using L-moments, statistical tests, and model assessment. Implements several methods for distribution selection, trend detection, and uncertainty quantification.

License CC BY 4.0

Encoding UTF-8

LazyData true

Imports stats,
randtests,
aTSA,
patchwork,
lmom,
ggplot2,
scales,
parallel,
grid

Suggests testthat,
vdiffr,
covr

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Contents

`ams.decomposition` *Decompose Annual Maximum Streamflow*

Description

Removes trends in the means and/or variances of an annual maximum streamflow (AMS) data using Sen's slope estimator and a moving-window variance estimator. Three scenarios are supported:

1. Trend in means only
2. Trend in variance only
3. Trends in both means and variance

Usage

```
ams.decomposition(df, scenario)
```

Arguments

`df`	A data frame containing: `max`: numeric vector of annual maximum streamflow values. `year`: any values (will be replaced internally by a scaled covariate).
`scenario`	Integer (1, 2, or 3) indicating which trend component(s) to remove: 1: Remove linear trend in the mean. 2: Remove trend in variance. 3: Remove both mean and variance trends sequentially.

Details

Internally, the function:

1. Constructs a covariate linearly spaced on `[0,1]` over the record length.
2. For scenario 1, fits Sen's slope (`sens.trend`) to (`max` vs. covariate) and removes the fitted linear mean trend.
3. For scenario 2, computes moving-window standard deviations (`mw.variance`), fits Sen's slope to those deviations, and rescales the series to remove variance trends around the overall mean.
4. For scenario 3, applies scenario 1 then scenario 2 sequentially.
5. Ensures all returned values are great than 1 by shifting the data if any negatives occur.

Value

A numeric vector of the same length as `df`, containing the "decomposed" AMS values with the specified trend(s) removed and adjusted to be strictly positive.

`assessment.plot` *Plot Model vs. Observed Quantiles for Assessment Results*

Description

Creates a quantile–quantile plot comparing observed annual maximum series (AMS) to model-derived quantile estimates. The 1:1 line is drawn in black, and the model estimates are overplotted as semi-transparent red points.

Usage

```
assessment.plot(ams, assessment)
```

Arguments

`ams`	Numeric vector of observed annual maximum series.
`assessment`	List containing assessment results. See `model.assessment`.

Details

The function orders `ams` in decreasing order to define the observed quantiles, then binds the corresponding `assessment$estimates` into a data frame. Axis labels are rendered with expression notation (cubic meters per second).

Value

A `ggplot` object showing:

- A black line for the theoretical 1:1 relationship between observed and model quantiles.
- Red points marking the model's estimated quantiles against the observed quantiles.

See Also

`geom_line`, `geom_point`

`bbmk.plot` *Plot Block-Bootstrap Mann–Kendall Test Results*

Description

Generates a histogram of block-bootstrap Mann–Kendall S-statistics with vertical lines indicating the empirical S-statistic and confidence bounds.

Usage

```
bbmk.plot(results)
```

Arguments

`results`	A list containing the BB-MK test output. See `bbmk.test`.

Details

The histogram uses 30 bins by default, with bootstrapped values in light gray. The legend is custom-mapped to distinguish “Bootstrapped Statistics” (gray), “S-Statistic” (black), and “Confidence Bounds” (red). The plot title and axis labels are set for clarity.

Value

A `ggplot` object showing:

- A gray histogram of the distribution of bootstrapped S-statistics.
- A red vertical line at each confidence bound.
- A black vertical line at the observed S-statistic.

See Also

`geom_histogram`, `geom_vline`, `scale_color_manual`

`bbmk.test` *Block-Bootstrap Mann-Kendall Test for Trend Detection*

Description

Performs a block-bootstrap version of the Mann-Kendall trend test to account for serial correlation in annual maximum streamflow (AMS) data. The procedure uses Spearman’s test to estimate the effective block size, then applies a parallelized bootstrap procedure to obtain the empirical p-value and confidence bounds for the Mann-Kendall statistic.

Usage

```
bbmk.test(ams, alpha = 0.05, n_sim = 10000, parallel = FALSE, quiet = TRUE)
```

Arguments

`ams`	A numeric vector of annual maximum streamflow data without NA values.
`alpha`	A numeric value indicating the significance level (default is 0.05).
`n_sim`	An integer number of bootstrap simulations (default is 10000).
`parallel`	Logical. If TRUE, runs the bootstrap in parallel (default is FALSE).
`quiet`	Logical. If FALSE, prints a summary of results (default is TRUE).

Details

The block size for the bootstrap is selected as `least_lag + 1`, where `least_lag` is estimated using Spearman's rank autocorrelation test. This addresses potential autocorrelation in the AMS data. Each bootstrap sample is generated by resampling blocks of the original data (without replacement) and computing the Mann-Kendall statistic.

Using `parallel = TRUE` can reduce computation time by approximately 50%. However, it is not possible to use `set.seed()` if the test is running in parallel, so your results may not be reproducible.

Value

A named list containing:

s.bootstrap Vector of bootstrapped test statistics used for plotting.

s.statistic The Mann-Kendall test statistic computed on the original series.

p.value Empirical two-sided p-value computed from the bootstrap distribution.

bounds Confidence interval bounds for the null distribution of the statistic.

reject Logical. TRUE if the null hypothesis was rejected at significance level alpha.

msg Character string summarizing the test result (printed if `quiet = FALSE`).

See Also

`mk.test`, `spearman.test`

`fixed.likelihood` *Compute Profile Log-Likelihood with Fixed Quantile Parameter*

Description

Maximizes the log-likelihood over nuisance parameters while holding a quantile parameter (yp) and exceedance probability (pe) fixed. Uses nlminb with repeated perturbations to ensure convergence.

Usage

```
fixed.likelihood(data, model, yp, pe, params)
```

Arguments

`data`	Numeric vector of observations. NaN values are handled internally.
`model`	Character string specifying the distribution code (e.g. 'GEV', 'GLO', 'GNO', 'PE3', 'LP3', 'WEI') with optional signature '10'/'100' or '11'/'110' for covariate trends in location and/or scale.
`yp`	Numeric value of a quantile at the specified exceedance probability.
`pe`	Numeric exceedance probability associated with yp.
`params`	Numeric vector of initial values for the nuisance.

Details

1. Constructs box constraints for nuisance parameters: scale > 0, optional shape unbounded.
2. Defines an objective function as the negative of reparameterized.likelihood().
3. Runs nlminb() with the given initial params, repeating up to 100 times with random perturbations if an error occurs.
4. Returns the profile log-likelihood (the negative of the final objective).

Value

A single numeric value: the maximized profile log-likelihood for the specified model, holding yp and pe fixed.

`generalized.likelihood` *Compute Generalized Log-Likelihood for GEV Models with Beta Prior*

Description

Calculates the log-likelihood of a sample under a Generalized Extreme Value (GEV) distribution (stationary or with covariate trends) combined with a Beta(p, q) prior on the shape parameter. Supports three model forms:

- `'GEV'` : stationary location, scale, and shape
- `'GEV100'`: linear trend in location only
- `'GEV110'`: linear trends in location and scale

Usage

```
generalized.likelihood(data, model, theta, prior)
```

Arguments

`data`	Numeric vector of observations (e.g., annual maxima). Any `NaN` values are removed prior to likelihood computation.
`model`	Character string specifying the model form. One of `'GEV'`, `'GEV100'`, or `'GEV110'`.
`theta`	Numeric vector of GEV parameters:

- For `'GEV'`: `theta = (mu, sigma, kappa)`
- For `'GEV100'`: `theta = (mu1, mu2, sigma, kappa)`
- For `'GEV110'`: `theta = (mu1, mu2, sigma1, sigma2, kappa)`
  Trends are applied as `mu` = `mu1 + (covariate * mu2)` and/or `sigma` = `sigma1 + (covariate * sigma2)`, where `covariate` is scaled linearly on `[0,1]` over the sample.

`prior`	Numeric vector of length 2 giving the Beta prior parameters `(p, q)` on the shape parameter kappa.

Details

1. Removes `NaN` values from `data` and constructs a covariate vector of the same length, linearly spaced on `[0,1]`.
2. Unpacks `theta` into `(mu, sigma, kappa)` or trend variants depending on `model`.
3. Computes the GEV log-density for each data point.
4. Computes the log-density of the Beta(p, q) prior on kappa.
5. Returns the total log-likelihood (the sum of steps 3 and 4).

Value

A single numeric value: the sum of the pointwise log-likelihoods of the data under the specified GEV model plus the log-density of the Beta prior on kappa. Returns `-Inf` if likelihood is zero.

`get.distributions` *Retrieve Candidate Distribution Specifications*

Description

Returns a named list of distribution specifications used for flood-frequency analysis. Each element is itself a list describing one distribution's:

- `name`: three-letter acronym.
- `quantile`: takes (periods, params) and returns quantiles.
- `estimate`: takes (ams) and returns parameter estimates.
- `n_params`: integer number of distribution parameters.
- `log`: logical; `TRUE` if fit on `log(ams)`, `FALSE` otherwise.

For two-parameter distributions, the element also contains:

- `t3_t4`: numeric $(L\ skewness,\ L\ kurtosis)$ values.

For three-parameter distributions, the element also contains:

- `lmr_function`: function(params) → vector of the first four L-moment ratios.
- `kappa_lower`, `kappa_upper`: numeric shape-parameter bounds.

Usage

```
get.distributions()
```

Value

A named `list` of distribution specifications. Valid names: `GEV`, `GUM`, `NOR`, `LNO`, `GLO`, `PE3`, `LP3`, `GNO`, `WEI`. Each element is a list as described above.

`gmle.estimation` *Perform Generalized MLE for GEV Models with Trend and Prior*

Description

Fits a Generalized Extreme Value (GEV) model—stationary or with linear trends in location and/or scale—by maximizing the posterior log-likelihood (MLE with Beta prior on the shape). Uses L-moment estimates for initialization and nlminb for constrained optimization.

Usage

```
gmle.estimation(data, model, prior)
```

Arguments

`data`	Numeric vector of observations (e.g. annual maxima). Any 'NaN' values are removed before fitting.
`model`	Character string specifying the GEV model form: 'GEV' = stationary, 'GEV100' = trend in location only, 'GEV110' = trends in location and scale.
`prior`	Numeric vector of length 2 giving the Beta prior parameters (p, q) on the shape parameter kappa.

Details

1. Calls lmom.estimation() on the cleaned data (removes NaN) to obtain initial estimates of (mu, sigma, kappa).
2. Sets initial trend parameters to 0 for non-stationary models.
3. Defines lower and upper bounds to enforce sigma > 0 and |kappa| < 0.5.
4. Constructs an objective function that returns the negative of generalized.likelihood().
5. Runs nlminb() with box constraints to find the posterior mode.

Value

A list with components:

- params: Numeric vector of the estimated GEV parameters in the same order as used by generalized.likelihood().
- mll : The maximized log-posterior (i.e. log-likelihood plus log-prior).

`kpss.test` *Kwiatkowski–Phillips–Schmidt–Shin (KPSS) Unit Root Test*

Description

Performs the KPSS test for stationarity in annual maximum streamflow (AMS) data using the **aTSA** package. The null hypothesis is that the time series is stationary.

Usage

```
kpss.test(ams, alpha = 0.05, quiet = TRUE)
```

Arguments

`ams`	A numeric vector of annual maximum streamflow data. Must not contain NA values.
`alpha`	A numeric value indicating the significance level. Must be 0.01, 0.05, or 0.10.
`quiet`	Logical. If FALSE, prints a summary message to the console (default is TRUE).

Details

The KPSS test implementation in **aTSA** uses interpolation from the Hobjin et al. (2004) significance table, which only includes thresholds for 0.01, 0.05, and 0.10. As such, the returned p-values are discretized. Specifically, $p = 0.01$ implies $p \leq 0.01$, and $p = 0.10$ implies $p \geq 0.10$. We use the Type III KPSS test, which accounts for drift and a nonstationary trend in the data by fitting an auxillary model.

Value

A named list with the following components:

p.value The reported p-value from the test. See notes regarding discrete thresholds.

reject Logical. TRUE if the null hypothesis of stationarity is rejected at `alpha`.

msg Character string summarizing the test outcome, printed if `quiet` = `FALSE`.

See Also

`kpss.test`, `pp.test`

`ld.selection` *L-Distance Method for Distribution Selection Using L-Moment Ratios*

Description

Selects the best-fit distribution from a candidate set by minimizing the Euclidean distance between theoretical and sample L-moment ratios (τ_3, τ_4). This method quantifies goodness-of-fit in the L-moment ratio space and returns the closest matching distribution.

Usage

```
ld.selection(ams)
```

Arguments

`ams`	A numeric vector containing the AMS data without NaN values

Details

For each candidate distribution, the method computes the Euclidean distance between sample L-moment ratios (τ_3, τ_4) and the closest point on the theoretical distribution's L-moment surface. The distribution with the minimum distance is selected.

If a distribution is flagged as requiring log-transformed data, the `log_lm` component is used for matching.

Value

A named list containing:

distance A list of fitted moment points for each candidate distribution with associated L-distance metrics.

recommendation The name of the distribution with the smallest L-distance.

See Also

`z.selection`, `lk.selection`

`likelihood` *Compute Log-Likelihood for Multiple Extreme-Value Models*

Description

Calculates the total log-likelihood of a data vector under various extreme-value distributions: GUM, NOR, LNO, GEV, GLO, GNO PE3, LP3, and WEI. Allows for optional linear covariate trends in location and/or scale. Returns -Inf whenever parameters or data violate distribution support.

Usage

```
likelihood(data, model, theta)
```

Arguments

`data`	Numeric vector of observations. NaN values are removed before computation.
`model`	Character string giving the model code. The first three letters specify the distribution: 'GUM', 'NOR', 'LNO', 'GEV', 'GLO', 'GNO', 'PE3', 'LP3', or 'WEI'. A five or six-character code adds a covariate signature: '10'/`100` = trend in location, '11'/`110` = trends in location and scale.
`theta`	Numeric parameter vector for the specified model and signature. From the parameter vector, we determine: • u (location). • s (scale). • k (shape) for three-parameter distributions. • Additional trend coefficients if there is non-stationarity.

Details

1. Removes NaN values from data and builds a covariate scaled on `[0,1]`.
2. Parses the stationary or non-stationary signature to compute u and s.
3. Adds k for three-parameter families and reparameterizes PE3/LP3 when needed.
4. Computes the log-density for each supported distribution, replacing invalid or NaN terms with -Inf.
5. Sums the pointwise log-likelihoods for output.

Value

A single numeric value: the sum of pointwise log-densities. Returns -Inf if any transformation term is non-positive or parameters are NaN.

`lk.selection` *L-Kurtosis Method for Distribution Selection Using L-Moment Ratios*

Description

Selects a best-fit probability distribution by minimizing the absolute vertical distance (in τ_4) between the sample L-moment ratios and the theoretical L-moment curves. For 3-parameter distributions, we use the shape parameter that best replicates the L-skewness of the data.

Usage

```
lk.selection(ams)
```

Arguments

`ams`	A numeric vector containing the AMS data without NaN values

Details

This method computes the vertical distance in τ_4 (L-kurtosis) between the sample and theoretical L-moment ratio diagrams at fixed τ_3 (L-skewness). The interpolated τ_4 and κ values are derived using `approx`.

Only 3-parameter distributions are considered in this method. Specifically, it evaluates GEV, GLO, PE3, LP3, GNO, and WEI. For more information, see the FFA framework website.

Value

A named list containing:

distance A list of interpolated L-moment matches and kurtosis-based metrics for each distribution.

recommendation Name of the distribution with the smallest L-kurtosis deviation.

See Also

`ld.selection`, `z.selection`

`lmom.estimation` *Estimate Distribution Parameters via L-Moments*

Description

Computes parameter estimates for a specified distribution using the method of L-moments. The function looks up the distribution's estimation function from `get.distributions()` and applies it to the observed annual maximum series (AMS).

Usage

```
lmom.estimation(ams, distribution)
```

Arguments

`ams`	Numeric vector of observed annual maximum streamflow values.
`distribution`	String giving the three-letter acronym of the target distribution. Must match one of the names returned by `names(get.distributions())`.

Details

Internally, `lmom.estimation` calls:

```
dist_list <- get.distributions()
dist_list[[distribution]]$estimate(ams)
```

If `distribution` is not a valid key, this will raise an error. Users should ensure that `ams` contains no missing values and that `length(ams)` is sufficient for L-moment estimation.

Value

A numeric vector of estimated distribution parameters, in the order expected by the corresponding quantile function. The number of components equals the distribution's `n_params`.

See Also

`get.distributions`

`lmom.plot` *Plot L-Moment Ratio Diagram with Sample and Candidate Distributions*

Description

Constructs an L-moment ratio plot (t4 vs. t3) showing the sample and log-sample L-moment ratios alongside the theoretical locations for a set of candidate distributions. Optionally highlights a small inset around the recommended distribution based on either the L-distance or L-kurtosis metric.

Usage

```
lmom.plot(ams, metric, results)
```

Arguments

`ams`	Numeric vector of observed annual maximum series (AMS) values.
`metric`	Character string specifying the selection metric. Must be one of: • `"L-distance"`: selects the distribution whose (t3, t4) location is closest in Euclidean distance to the sample point. See `ld.selection`. • `"L-kurtosis"`: selects the distribution point with the same t3 as the sample, minimizing |t4 difference|. See `lk.selection`. • `"Z-statistic"`: selects the distribution by using a Kappa distribution and a parametric bootstrap. See `z.selection`.
`results`	List of results from the model-selection procedure. Must contain: `recommendation` Three-letter acronym of the selected distribution, e.g. `"GEV"`, `"LP3"`. Generated by `ld.selection`, `lk.selection`, `z.selection`

Value

A `ggplot` object of the L-moment ratio diagram, with:

- Lines for each 3-parameter distribution locus.
- Points for each 2-parameter distribution reference point.
- Sample and log-sample t3, t4 points.
- A legend mapping colors and shapes.
- If `metric` is `"L-distance"` or `"L-kurtosis"`, an inset magnification of the neighborhood around the recommended point.

`mk.test` *Mann–Kendall Test for Monotonic Trends*

Description

Performs the Mann–Kendall trend test on a numeric vector to detect the presence of a monotonic trend (increasing or decreasing) over time. The test is non-parametric and accounts for tied observations in the data.

Usage

```
mk.test(data, alpha = 0.05, quiet = TRUE)
```

Arguments

`data`	A numeric vector of AMS values or their variances. Must not contain NA values.
`alpha`	A numeric value specifying the significance level (default is 0.05).
`quiet`	Logical. If FALSE, prints a summary of the test result to the console.

Details

The statistic \(S\) is computed as the sum over all pairs \(i < j\) of the sign of the difference \(x_j - x_i\). Ties are explicitly accounted for when calculating the variance of \(S\), using grouped frequencies of tied observations.

The test statistic \(Z\) is then computed based on the sign and magnitude of \(S\), and the p-value is derived from the standard normal distribution.

Value

A named list with the following components:

s.statistic The raw Mann–Kendall test statistic \(S\).

s.variance The variance of the test statistic under the null hypothesis.

p.value The p-value associated with the two-sided hypothesis test.

reject Logical. TRUE if the null hypothesis of no trend is rejected at `alpha`.

msg A character string summarizing the result (printed if `quiet` = `FALSE`).

See Also

`bbmk.test` for a bootstrap-based variant of this test.

`mks.plot` *Plot Results from the Mann–Kendall–Sneyers (MKS) Test*

Description

Constructs a two-panel visualization of the MKS test:

1. Upper panel: normalized progressive and regressive Mann–Kendall S-statistics over time, with dashed confidence bounds and marked potential trend-change points.
2. Lower panel: the original annual maximum series (AMS) with optional trend line and highlighted change points.

Usage

```
mks.plot(df, result, show_trend = TRUE)
```

Arguments

`df`	A `data.frame` containing at least the following columns: `year` Numeric or integer vector of years. `max` Numeric vector of observed annual maxima. After loading `result` into the environment, the function adds: • `s.progressive`: normalized progressive S-statistic (numeric). • `s.regressive`: normalized regressive S-statistic (numeric).
`result`	A `list` containing the MKS test outputs, generated by `mks.test`.
`show_trend`	Logical; if `TRUE` (default), draw a fitted line through the AMS series.

Details

- The upper plot uses `geom_line()` for S-statistics, `geom_hline()` for +/-`bound`, and `geom_point()` for `change.df$statistic`.
- The lower plot marks all AMS points with `geom_point()`, overlays a trend line if `show_trend = TRUE`, and highlights change points from `change.df`.
- Both panels share a common color legend, with custom labels.

Value

A `patchwork` object stacking two `ggplot2` panels vertically: the test–statistic time series with bounds and change points, and the AMS series with optional trend.

`mks.test` *Mann–Kendall–Sneyers Test for Detecting Change Points in Trends*

Description

Performs the Mann–Kendall–Sneyers (MKS) test to detect the approximate time of onset of a monotonic trend in annual maximum streamflow (AMS) data. The test computes progressive and regressive Mann–Kendall statistics and identifies statistically significant crossing points, indicating potential change points in the trend.

Usage

```
mks.test(ams, year, alpha = 0.05, quiet = TRUE)
```

Arguments

`ams`	Numeric vector of annual maximum streamflow values with no missing values.
`year`	Numeric vector of years corresponding to `ams`, with no missing values.
`alpha`	Numeric significance level for the test (default 0.05).
`quiet`	Logical flag to suppress or print a summary message (default TRUE).

Details

The function computes progressive and regressive Mann–Kendall statistics \(S_t\), normalized by their expected values and variances under the null hypothesis. The crossing points where the difference between these normalized statistics changes sign are identified using linear interpolation. The significance of detected crossings is assessed using normal quantiles and the maximum absolute crossing statistic.

Value

A named list containing:

s.progressive Normalized progressive Mann–Kendall statistics over time.

s.regressive Normalized regressive Mann–Kendall statistics over time.

bound Critical confidence bound for significance based on `alpha`.

crossing.df Data frame of crossing points with indices, years, statistics, and AMS values.

change.df Subset of `crossing_df` where crossing statistics exceed confidence bounds.

p.value Two-sided p-value assessing the significance of maximum crossing statistic.

reject Logical indicating whether null hypothesis of no change point is rejected.

msg Character string summarizing the test result (printed if `quiet` = `FALSE`).

References

Sneyers, R. (1990). On the statistical analysis of series of observations. Technical note No. 143, World Meteorological Organization, Geneva.

See Also

`mk.test` for the classical Mann–Kendall test.

`mle.estimation` *Fit Extreme-Value Model using Maximum Likelihood*

Description

Estimates parameters of an extreme-value distribution (GUM, NOR, LNO, GEV, GLO, GNO, PE3, LP3, or WEI) with optional linear covariate trends in location and/or scale by maximizing the log-likelihood. Initial values are obtained from L-moment estimation, and optimization is performed via nlminb with repeated perturbations if needed.

Usage

```
mle.estimation(data, model)
```

Arguments

`data`	Numeric vector of observations. Any NaN values are removed prior to fitting.
`model`	Character string specifying the distribution code. The first three letters denote the family: 'GUM', 'NOR', 'LNO', 'GEV', 'GLO', 'GNO', 'PE3', 'LP3', or 'WEI'. A trailing signature of '10' or '100' indicates a linear trend in location; '11' or '110' indicates linear trends in both location and scale.

Details

1. Removes NaN values from data.
2. Calls lmom.estimation() on cleaned data to obtain initial estimates of location, scale, and shape (if applicable).
3. Initializes trend parameters to zero if there is a trailing signature.
4. For Weibull models, forces the location estimate to zero to ensure support.
5. Defines an objective function as the negative of likelihood().
6. Runs nlminb() with box constraints on scale (>0) and repeats up to 100 times with random perturbations if convergence returns Inf.

Value

A list with components:

- params: Numeric vector of estimated parameters.
- mll : Maximized log-likelihood value.

`model.assessment` *Evaluate Goodness-of-Fit and Uncertainty Coverage for Fitted Flood Models*

Description

Computes multiple performance metrics and diagnostic indicators to assess the quality of a fitted flood frequency distribution. This includes residual statistics, information criteria, and coverage-based metrics using bootstrapped confidence intervals.

Usage

```
model.assessment(
  ams,
  distribution,
  params,
  uncertainty,
  plotting_position = "Weibull",
  alpha = 0.05
)
```

Arguments

`ams`	Numeric vector of annual maximum streamflow values (no missing values).
`distribution`	The three letter code for a probability distribution.
`params`	Numeric vector of fitted distribution parameters.
`uncertainty`	A list containing bootstrap confidence interval estimates, typically the output from `sb.uncertainty`. Must include `t`, `ci_lower`, and `ci_upper`.
`plotting_position`	Character string specifying the plotting position formula. One of `"Weibull"`, `"Blom"`, `"Cunnane"`, `"Gringorten"`, or `"Hazen"`.
`alpha`	Significance level for interval-based metrics (default is 0.05).

Details

Empirical return periods are calculated using the specified plotting position formula. Quantile estimates are generated from the fitted model. Standard residual metrics (R^2, RMSE, Bias) are computed. The confidence interval width (AW), empirical coverage (POC), and CWI are used to assess uncertainty calibration.

Value

A named list of model assessment metrics:

estimates Quantile estimates for empirical return periods.

R2 Coefficient of determination comparing estimates vs observed AMS.

RMSE Root mean squared error of quantile estimates.

Bias Mean bias of quantile estimates.

AIC Akaike Information Criterion.

BIC Bayesian Information Criterion.

AW Average width of bootstrap confidence intervals.

POC Percent of observations covered by the confidence intervals.

CWI Confidence Width Index, penalizing over/under-coverage.

Note

The confidence intervals are interpolated on the log-return scale. Coverage statistics exclude missing widths due to extrapolation or poor overlap.

See Also

`sb.uncertainty`, `lm`, `approx`

`mw.variance` *Compute moving-window standard deviations for an AMS data frame*

Description

This function calculates the standard deviation of the annual maximum series ("max") in non-overlapping (or partially overlapping) moving windows, returning a data frame that pairs each window's mean "year" with its computed standard deviation. The window parameters control the length and step size.

Usage

```
mw.variance(df, window_length = 10, window_step = 5)
```

Arguments

`df`	A data.frame (or tibble) containing at least two numeric columns: **year** Numeric or integer vector of years. **max** Numeric vector of annual maximum values. If either column is missing or not numeric, the function will throw an error.
`window_length`	Integer(1). The number of consecutive rows in each moving window. Must be a positive integer less than or equal to `nrow(df)`. Defaults to `10`. If `nrow(df) < window_length`, an error is raised.
`window_step`	Integer(1). The offset (in rows) between the start of successive windows. Must be a positive integer. Defaults to `5`.

Value

A data.frame with two columns:

year Numeric. The mean of `df$year` within each window.

std Numeric. The standard deviation of `df$max` within each window, computed with `na.rm = TRUE`.

Each row corresponds to one window. The number of rows equals `floor((nrow(df) - window_length) / window_step) + 1`.

`pettitt.plot` *Plot Results from the Pettitt Change-Point Test*

Description

Creates a two-panel visualization of the Mann–Whitney–Pettitt test:

1. Upper panel: Pettitt Ut statistic over time with its critical threshold and highlighted potential change point(s).
2. Lower panel: the original annual maximum series (AMS) with optional trend line and segment-wise mean estimates, marking the change point.

Usage

```
pettitt.plot(df, results, show_trend = TRUE)
```

Arguments

`df`	A data frame containing at least the following columns: `year` Numeric or integer vector of observation years. `max` Numeric vector of observed annual maxima (AMS). The function will add: • `ut`: Pettitt Ut statistic for each year.
`results`	A list of Pettitt test outputs. Generated by `pettitt.test`.
`show_trend`	Logical; if `TRUE` (default), overlays a fitted line through the AMS series.

Details

1. The Ut statistic is plotted as a solid line against year; the critical value is shown with a dashed red line.
2. If `change.index` > 0, that year's point is highlighted in blue.
3. The lower panel divides the series into one or two segments (pre- and post-change) and draws their mean flow levels as green horizontal segments.

Value

A **patchwork** object consisting of two vertically stacked **ggplot2** panels:

- The time series of Ut with dashed horizontal line at +/-`k.critical` and points marking the potential change point.
- The AMS series with optional trend line, segment means drawn as horizontal segments, and change-point markers.

See Also

`geom_line`, `geom_hline`, `geom_segment`

`pettitt.test` *Pettitt Test for Abrupt Changes in the Mean of a Time Series*

Description

Performs the non-parametric Pettitt test to detect a single change point in the mean of a time series, often used for abrupt shifts in hydrological data.

Usage

```
pettitt.test(ams, year, alpha = 0.05, quiet = TRUE)
```

Arguments

`ams`	Numeric vector of annual maximum streamflow values, no missing data.
`year`	Numeric vector of years corresponding to `ams`, no missing data.
`alpha`	Numeric significance level for hypothesis testing (default 0.05).
`quiet`	Logical; if FALSE, print test summary messages (default TRUE).

Details

The Pettitt test is a rank-based non-parametric test that evaluates the hypothesis of a change point in the median/mean of a time series. It computes the maximum of the absolute value of the U-statistic over all possible split points. The p-value is approximated using an asymptotic formula.

Value

A named list containing:

ut Vector of absolute U-statistics for all time indices.

k.statistic Maximum absolute U-statistic (test statistic).

k.critical Critical K-statistic value for given `alpha`.

p.value Approximate p-value for the test.

change.index Index of the detected change point (0 if none).

change.year Year of the detected change point (0 if none).

reject Logical indicating if null hypothesis was rejected.

msg Formatted summary message describing the test result.

References

Pettitt, A.N. (1979) A non-parametric approach to the change-point problem, *Applied Statistics*, 28(2), 126-135.

`pp.test` *Phillips–Perron Unit Root Test*

Description

Applies the Phillips–Perron (PP) test to assess the presence of a unit root in annual maximum streamflow (AMS) data. The null hypothesis is that the series contains a unit root (i.e., is non-stationary).

Usage

```
pp.test(ams, alpha = 0.05, quiet = TRUE)
```

Arguments

`ams`	Numeric vector of annual maximum streamflow data with no missing values.
`alpha`	Numeric significance level. Must be one of 0.01, 0.05, or 0.10.
`quiet`	Logical. If FALSE, prints a summary message to the console (default is TRUE).

Details

The test is implemented using the **aTSA** package, which interpolates p-values from the critical values in Banerjee et al. (1993). The critical values are only available for `alpha` = `0.01`, `0.05`, and `0.10`. A reported p-value of 0.01 indicates $p \leq 0.01$, and 0.10 indicates $p \geq 0.10$.

The null hypothesis is that the time series contains a unit root (non-stationary). Rejection of the null suggests that the series is stationary.

Value

A named list containing:

p.value Reported p-value from the test. See notes on interpolation thresholds.

reject Logical. TRUE if the null hypothesis of a unit root is rejected at `alpha`.

msg Character string summarizing the test result (printed if `quiet` = `FALSE`).

References

Banerjee, A., Dolado, J., Galbraith, J.W., & Hendry, D.F. (1993). *Cointegration, Error Correction, and the Econometric Analysis of Non-Stationary Data*. Oxford University Press.

See Also

`pp.test`, `kpss.test`

`reparameterized.likelihood`
Compute Reparameterized Log-Likelihood for Extreme-Value Models

---

Description

Calculates the log-likelihood of data under various extreme-value distributions with a fixed reference quantile yp at exceedance probability pe. Supports optional linear covariate trends in location and/or scale via model signatures.

Usage

```
reparameterized.likelihood(data, model, yp, pe, theta)
```

Arguments

`data`	Numeric vector of observations. NaN values are removed internally.
`model`	Character string specifying the distribution code. The first three letters denote the family: 'GUM', 'NOR', 'LNO', 'GEV', 'GLO', 'GNO', 'PE3', 'LP3', or 'WEI'. A trailing signature of '10'/'100' indicates a trend in location. '11'/'110' indicates trends in both location and scale.
`yp`	Numeric value of the fixed reference quantile at exceedance probability pe.
`pe`	Numeric exceedance probability corresponding to yp.
`theta`	Numeric vector of nuisance parameters to be optimized: • s (scale). • optional trend coefficients for nonstationary models. • k (shape) for three-parameter families.

Details

1. Removes NaN values from data and constructs a covariate vector on `[0,1]`.
2. Computes the model quantile qpe at probability pe using get.distributions().
3. Determines location u and scale s from yp, qpe, and theta, applying covariate trends.
4. Includes shape parameter k for three-parameter distributions.
5. Computes the log-density for each supported family, replacing invalid values with –Inf.
6. Returns the total log-likelihood.

Value

A single numeric value: the sum of pointwise log-likelihoods under the specified model with the given reparameterization. Invalid or out-of-support cases return –Inf.

`rfpl.uncertainty` *Compute Uncertainty via Regula Falsi Profile Likelihood*

Description

Calculates point estimates and confidence intervals for return levels at standard return periods (2, 5, 10, 20, 50, 100 years) using profile likelihood and the Regula Falsi root-finding method.

Usage

```
rfpl.uncertainty(data, model, alpha = 0.05, eps = 0.01)
```

Arguments

`data`	Numeric vector of observations. NaN values are removed internally.
`model`	Character string specifying the distribution code. The first three letters denote the family: 'GUM', 'NOR', 'LNO', 'GEV', 'GLO', 'GNO', 'PE3', 'LP3', or 'WEI'. A trailing signature of '10'/'100' indicates a trend in location. '11'/'110' indicates trends in both location and scale.
`alpha`	Numeric significance level for confidence intervals (default 0.05).
`eps`	Numeric tolerance for the Regula Falsi convergence criterion (default 1e-2).

Details

1. Retrieves the quantile function for the specified model via get.distributions().
2. Fits the model by ordinary MLE to obtain parameter estimates and log-likelihood.
3. Defines an objective function f(yp, pe) based on the chi-squared distribution.
4. Iteratively brackets the root by scaling initial guesses by 0.05 until f changes sign.
5. Uses the Regula Falsi method to solve f(yp, pe) = 0 for each return-period probability.
6. Returns lower and upper confidence bounds at level alpha and the point estimates.

Value

A list with components:

- ci_lower : Numeric vector of lower confidence bounds for each return period.
- estimates: Numeric vector of profile-likelihood point estimates (return levels).
- ci_upper : Numeric vector of upper confidence bounds for each return period.

`runs.plot` *Plot Residuals and Runs Test p-Value for Sen's Estimator*

Description

Generates a residual plot of Sen's estimator applied to annual maximum series (AMS), with a horizontal dashed line at zero and an annotation of the Runs test p-value. The function can display either the residuals of the variance or the mean estimator, depending on the `name` argument.

Usage

```
runs.plot(df, results, name, show_trend = TRUE)
```

Arguments

`df`	A `data.frame` containing at least: `year` Numeric or integer vector of years. `std` Numeric vector of standard deviations (if `name = "sens-variance"`). `max` Numeric vector of observed AMS values (if `name = "sens-mean"`).
`results`	A `list` generated by `runs.test`.
`name`	Character; either `"sens-variance"` or `"sens-mean"`.
`show_trend`	Logical; if `TRUE` (default), overlay a trend line through the AMS series.

Value

A `ggplot` object showing:

- Black points for each residual versus year.
- A red dashed horizontal line at $y = 0$.
- A text annotation of "Runs p-value: X.XXX" in the plot area.

See Also

`geom_point`, `geom_hline`

`runs.test` *Perform a Wald–Wolfowitz runs test on model residuals*

Description

`runs.test` applies the Wald–Wolfowitz runs test to a numeric vector of residuals in order to assess whether they behave as a random sequence. The test statistic's p-value is compared to the specified significance level `alpha`, and a pass/fail decision is returned along with a human-readable summary message.

Usage

```
runs.test(residuals, alpha = 0.05, quiet = TRUE)
```

Arguments

`residuals`	Numeric vector of residual values from a fitted linear model.
`alpha`	Numeric significance level for the runs test (default is 0.05).
`quiet`	Logical. If FALSE, prints a summary message describing results (default is TRUE).

Details

The Wald–Wolfowitz runs test examines the sequence of positive and negative residuals to test for randomness around the median. A small p-value suggests non-random clustering, which may indicate that a linear model is inappropriate for the data.

Internally, this function calls `runs.test`.

Value

A named list with components:

p.value P-value from the Wald–Wolfowitz runs test applied to residuals.

residuals Numeric vector of residual values from a fitted linear model.

reject Logical. TRUE if null hypothesis of random residuals is rejected.

msg Character string summarizing the test result.

References

Wald, A. and Wolfowitz, J. (1940). On a test whether two samples are from the same population. Annals of Mathematical Statistics, 11(2), 147–162.

See Also

`runs.test`, `sens.trend`

`sb.uncertainty` *Sample Bootstrap Confidence Intervals for Flood Quantile Estimates*

Description

Computes confidence intervals for flood quantile estimates using the nonparametric sample bootstrap method, based on L-moment parameter estimation. This function supports uncertainty quantification for return period estimates derived from a fitted distribution.

Usage

```
sb.uncertainty(
  ams,
  distribution,
  method,
  n_sim = 10000,
  alpha = 0.05,
  parallel = FALSE
)
```

Arguments

`ams`	Numeric vector of annual maximum streamflow values (no missing values).
`distribution`	Name of the selected distribution
`method`	Character string specifying the estimation method. Currently supports `"L-moments"`, `"MLE"`, and `"GMLE"`.
`n_sim`	Integer number of bootstrap simulations (default is 100000).
`alpha`	Numeric significance level for the confidence intervals (default is 0.05).
`parallel`	Logical. If TRUE, runs the bootstrap in parallel (default is FALSE).

Details

The bootstrap procedure simulates resamples from the fitted distribution via inverse transform sampling using the estimated parameters. For each resample, L-moment parameters are re-estimated and used to compute quantiles. Confidence intervals are obtained by applying empirical quantiles to the resulting distribution of estimates.

Using `parallel = TRUE` can reduce computation time by approximately 50%. However, using this option will nullify any calls to `set.seed()`, so your results may not be reproducible.

Value

A named list containing:

estimates Vector of estimated quantiles for return periods 2, 5, 10, 20, 50, and 100.

ci_lower Lower bound of the confidence interval for each return period.

ci_upper Upper bound of the confidence interval for each return period.

t Vector of return periods (2, 5, 10, 20, 50, and 100).

See Also

`samlmu`, `quantile`

`sens.plot` *Plot Results from Sen's Trend Estimator*

Description

Produces a scatter plot of the annual maximum series (AMS) or its variance against time, overlaid with Sen's estimated trend line and a text annotation of the fitted equation.

Usage

```
sens.plot(df, results, name, show_trend = TRUE)
```

Arguments

`df`	A `data.frame` containing at least: `year` Numeric or integer vector of observation years. `std` Numeric vector of estimated AMS variances (if `name = "sens-variance"`). `max` Numeric vector of observed AMS values (if `name = "sens-mean"`).
`results`	A `list` generated by `sens.trend`.
`name`	Character; if `"sens-variance"`, plots residual variance (`df$std`) and titles the plot accordingly. If `"sens-mean"`, plots AMS mean (`df$max`).
`show_trend`	Logical; if `TRUE` (default), draws the AMS data trend line.

Value

A `ggplot` object showing:

- Black points for each year's AMS (or variance) value.
- Optional black line connecting the raw data if `show_trend` = `TRUE`.
- Blue line representing Sen's trend estimate.
- A text annotation displaying the fitted equation $y = mx + b$.

`sens.trend` *Sen's Slope Estimator*

Description

Computes Sen's slope estimator and intercept for a univariate time series

Usage

```
sens.trend(data, year, quiet = TRUE)
```

Arguments

`data`	Numeric vector of AMS values or variances with no missing values.
`year`	Numeric vector of years corresponding to `data`, with no missing values.
`quiet`	Logical. If FALSE, prints a summary message describing results (default is TRUE).

Details

Sen's slope estimator is a robust, non-parametric trend estimator computed from the median of all pairwise slopes between data points. The corresponding intercept is taken as the median of residual-corrected values.

Value

A named list containing:

sens.slope Median slope of all pairwise data-year combinations (Sen's slope).

sens.intercept Median intercept estimate of the fitted line.

residuals Vector of residuals between observed and fitted values.

msg Character string summarizing the estimator.

References

Sen, P.K. (1968). Estimates of the regression coefficient based on Kendall's tau. *Journal of the American Statistical Association*, 63(324), 1379–1389.

See Also

`runs.test`, `mk.test`

`spearman.plot` *Plot Spearman's Rho Autocorrelation Results*

Description

Visualizes the lagged Spearman's rho autocorrelation coefficients with stems from zero and points indicating statistical significance.

Usage

```
spearman.plot(results)
```

Arguments

`results`	A `list` generated by `spearman.test`.

Value

A `ggplot` object showing:

- Vertical segments from $y = 0$ up to each ρ value at its lag.
- Filled circles at each lag, filled black if `sig` == `TRUE` (no serial correlation) or white if `sig` == `FALSE` (serial correlation detected).

`spearman.test` *Spearman Test for Serial Correlation in Time Series*

Description

Performs the Spearman rank autocorrelation test on annual maximum streamflow (AMS) data to assess the presence of serial correlation at various lags. Reports the first lag where autocorrelation is no longer statistically significant at the specified level.

Usage

```
spearman.test(ams, alpha = 0.05, quiet = TRUE)
```

Arguments

`ams`	Numeric vector of annual maximum streamflow data with no missing values.
`alpha`	Numeric significance level for the test (default is 0.05).
`quiet`	Logical. If FALSE, prints a summary message describing the result (default is TRUE).

Details

For each lag from 1 to `n - 3`, the function computes the Spearman rank correlation between the AMS series and its lagged version. The first lag with a non-significant autocorrelation (p-value > `alpha`) is returned as `least.lag`.

This test is useful for identifying the minimum temporal separation required to ensure approximate independence, especially when constructing block-bootstrap resampling schemes.

Value

A named list containing:

rho Vector of Spearman autocorrelation estimates for lags `1` to `n - 3`.

sig Logical vector indicating which lags exhibit significant autocorrelation.

least.lag The smallest lag at which autocorrelation is not statistically significant.

reject Logical. TRUE if we reject the null hypothesis of no serial correlation.

msg Character string summarizing the test result (printed if `quiet = FALSE`).

See Also

`cor.test`, `bbmk.test`

`uncertainty.plot` *Plot Uncertainty Bounds and Estimates Over Time*

Description

Constructs a time–series plot of parameter estimates with their associated confidence intervals. The confidence bounds are shown as a shaded ribbon, and the point estimates are overlaid as a solid line. The x-axis is displayed on a log10 scale.

Usage

```
uncertainty.plot(results)
```

Arguments

`results`	A `list` containing the uncertainty analysis output. Generated by `sb.uncertainty`.

Value

A `ggplot` object showing:

- A dashed black line for both the lower and upper confidence bounds.
- A gray ribbon between `ci_lower` and `ci_upper`, semi-transparent.
- A solid blue line for the point estimates.
- A log10-scaled x-axis with ticks at each `t`.

`white.test` *White Test for Heteroskedasticity in Annual Maximum Streamflow*

Description

Performs the White test for heteroskedasticity by regressing the squared residuals of a linear model on the original regressors and their squared terms. The null hypothesis is homoskedasticity.

Usage

```
white.test(ams, year, alpha = 0.05, quiet = TRUE)
```

Arguments

`ams`	Numeric vector of annual maximum streamflow values with no missing values.
`year`	Numeric vector of years corresponding to `ams`, with no missing values.
`alpha`	Numeric significance level for the test (default is 0.05).
`quiet`	Logical. If FALSE, prints a summary message to the console (default is TRUE).

Details

The White test regresses the squared residuals from a primary linear model `lm(ams ~ year)` against both the original regressor and its square. The test statistic is calculated as `n * R^2`, where `R^2` is from the auxiliary regression. Under the null hypothesis, this statistic follows a χ^2 distribution with 2 degrees of freedom.

Rejection of the null hypothesis suggests the presence of heteroskedasticity in the residuals.

Value

A named list containing:

r.squared Coefficient of determination from the auxiliary regression.

statistic White test statistic based on sample size and auxiliary `R^2`.

p.value P-value computed from the Chi-squared distribution with 2 degrees of freedom.

reject Logical. TRUE if the null hypothesis is rejected at `alpha`.

msg Character string summarizing the test result (printed if `quiet = FALSE`).

References

White, H. (1980). A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, 48(4), 817–838.

See Also

`lm`, `pchisq`

`z.selection` *Z-Statistic Method for Distribution Selection*

Description

Selects the best-fit distribution by computing a bias-corrected Z-statistic for the sample τ_4 (L-kurtosis) against theoretical L-moment surfaces for a set of candidate distributions. The distribution with the smallest absolute Z-score is selected.

Usage

```
z.selection(ams, n_sim = 1e+05, parallel = FALSE)
```

Arguments

`ams`	Numeric vector of annual maximum streamflow values (no missing values).
`n_sim`	Number of bootstrap samples to generate (default = 100000).
`parallel`	Logical. If TRUE, runs the bootstrap in parallel (default is FALSE).

Details

The method evaluates both raw and log-transformed data. Raw-data distributions include GEV, GLO, PE3, GNO, and WEI. Log-data distributions include LP3. A Kappa distribution is fitted to each and used to simulate bootstrapped L-moments. The observed τ_4 is then compared to each theoretical distribution using the Z-statistic framework.

Using `parallel = TRUE` can reduce computation time by approximately 50%. However, using this option will nullify any calls to `set.seed()`, so your results may not be reproducible.

Value

A list containing:

params Kappa parameters fitted to the raw AMS data.

log_params Kappa parameters fitted to the log-transformed AMS data.

bootstrap List of bootstrap estimates of bias and standard deviation for τ_4.

distance List of computed Z-statistics for each candidate distribution.

recommendation Name of the best-fit distribution based on the smallest Z-statistic.

See Also

`ld.selection`, `lk.selection`, `pelkap`, `quakap`

Index