Package ‘ffaframework’

July 17, 2025

Title Flood Frequency Analysis Framework

Version 0.2.1

Author Riley Wheadon [aut, cre]

Maintainer Riley Wheadon <rileywheadon@gmail.com>

Description Tools for exploratory data analysis and flood frequency analysis. Implements statistical tests and methods for distribution selection, parameter estimation, uncertainty quantification, and model assessment.

License CC BY 4.0

Encoding UTF-8

LazyData true

Depends R (>= 4.5.0), ggplot2, patchwork

Suggests knitr, testthat, vdiffr

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Contents

`ams_decomposition` *Decompose Annual Maximum Streamflow Data*

Description

Removes trends in the mean and/or variance of annual maximum series (AMS) data using Sen's trend estimator and/or a moving-window estimator of the variance. Four scenarios are supported:

1. No trend. The data is returned unmodified.
2. Trend in the mean only.
3. Trend in the variance only.
4. Trends in both the mean and the variance.

Usage

```
ams_decomposition(data, years, trend)
```

Arguments

`data` Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.

`years` Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.

`trend` Named list indicating which distribution parameters are modeled as nonstationary. Must contain:

- `location`: Logical scalar. If `TRUE`, the location parameter has a trend.
- `scale`: Logical scalar. If `TRUE`, the scale parameter has a trend.

Details

Internally, the function does the following:

1. Compute `covariate = (years - 1900) / 100`.
2. If there is a trend in the location, fit Sen's trend estimator to `data` and `covariate`. Then, remove the fitted linear trend.
3. If there is a trend in the scale, compute the moving-window variance using ams_mw_variability, fit Sen's trend estimator to those variances, and then rescales the series to remove trends in the scale.
4. If necessary, shift the data so that its minimum is at least 1.

Value

Numeric vector of decomposed AMS data with the same length as `data`.

See Also

ams_mw_variability, eda_sens_trend

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)
trend <- list(location = TRUE, scale = FALSE)
ams_decomposition(data, years, trend)
```

`ams_mw_variability` *Estimate Variance for Annual Maximum Streamflow Data*

Description

This function estimates the standard deviation of a vector of annual maximum series (AMS) data using a moving window algorithm, returning a list that pairs each window's mean year with its computed standard deviation. The parameters `size` and `step` parameters control the behaviour of the window.

Usage

```
ams_mw_variability(data, years, size = 10L, step = 5L)
```

Arguments

`data`	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.
`years`	Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.
`size`	Integer scalar. The number of years in each moving window. Must be a positive number less than or equal to `length(data)` (default is 10). If `length(data) < size`, an error is raised.
`step`	Integer scalar. The offset (in years) between successive moving windows. Must be a positive number (default is 5).

Value

A list with two entries:

- `years`: Numeric vector containing the mean year within each window.
- `std`: Numeric vector of standard deviations within each window.

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)
ams_mw_variability(data, years)
```

`eda_bbmk_test` *Block-Bootstrap Mann-Kendall Test for Trend Detection*

Description

Performs a bootstrapped version of the Mann-Kendall trend test to account for serial correlation in annual maximum series (AMS) data. The procedure uses Spearman's autocorrelation test to estimate the least insignificant lag, then applies a bootstrap procedure to obtain the empirical p-value and confidence bounds for the Mann-Kendall S-statistic.

Usage

```
eda_bbmk_test(data, alpha = 0.05, samples = 10000L, quiet = TRUE)
```

Arguments

`data`	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.
`alpha`	Numeric scalar in $[0.01, 0.1]$. The significance level for confidence intervals or hypothesis tests. Default is 0.05.
`samples`	Integer scalar. The number of bootstrap samples. Default is 10000.
`quiet`	Logical scalar. If `FALSE`, prints a summary of of the statistical test to the console. Default is `TRUE`.

Details

The block size for the bootstrap is selected as `least_lag + 1`, where `least_lag` is estimated using eda_spearman_test. Each bootstrap sample is generated by resampling blocks of the original data (without replacement) and computing the Mann-Kendall S-statistic. This procedure adjusts for autocorrelation in the data.

Value

List; the results of the test, including:

- `data`: The `data` argument.
- `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. If `TRUE`, the null hypothesis was rejected at `alpha`.
- `msg`: Character string summarizing the test result (printed if `quiet = FALSE`).

See Also

plot_bbmk_test, eda_mk_test, eda_spearman_test

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
eda_bbmk_test(data, samples = 1000L)
```

`eda_kpss_test` *Kwiatkowski–Phillips–Schmidt–Shin (KPSS) Unit Root Test*

Description

Performs the KPSS unit root test on annual maximum series (AMS) data. The null hypothesis is that the time series is trend-stationary with a linear trend and constant drift. The alternative hypothesis is that the time series has a unit root and is nonstationary.

Usage

```
eda_kpss_test(data, alpha = 0.05, quiet = TRUE)
```

Arguments

`data`	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.
`alpha`	Numeric scalar in $[0.01, 0.1]$. The significance level for confidence intervals or hypothesis tests. Default is 0.05.
`quiet`	Logical scalar. If `FALSE`, prints a summary of of the statistical test to the console. Default is `TRUE`.

Details

The implementation of the KPSS test is based on the **aTSA** package, which interpolates a significance table from Hobjin et al. (2004). Therefore, a result of $p = 0.01$ implies that $p \leq 0.01$ and a result of $p = 0.10$ implies that $p \geq 0.10$. This implementation uses the Type III KPSS test, which accounts for a linear trend in the data.

Value

A list containing the test results, including:

- `data`: The `data` argument.
- `statistic`: The KPSS test statistic.
- `p_value`: The interpolated p-value. See note regarding discrete thresholds.
- `reject`: Logical scalar. If, TRUE the null hypothesis is rejected at `alpha`.
- `msg`: Character string summarizing the test outcome, printed if `quiet` = `FALSE`.

References

Hobijn, B., Franses, P.H. and Ooms, M. (2004), Generalizations of the KPSS-test for stationarity. Statistica Neerlandica, 58: 483-502.

See Also

eda_pp_test

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
eda_kpss_test(data)
```

`eda_mks_test` *Mann–Kendall–Sneyers Test for Change Point Detection*

Description

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

Usage

```
eda_mks_test(data, years, alpha = 0.05, quiet = TRUE)
```

Arguments

`data`	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.
`years`	Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.
`alpha`	Numeric scalar in $[0.01, 0.1]$. The significance level for confidence intervals or hypothesis tests. Default is 0.05.
`quiet`	Logical scalar. If `FALSE`, prints a summary of of the statistical test to the console. Default is `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 quantiles of the normal distribution.

Value

A list containing the test results, including:

- `data`: The `data` argument.
- `years`: The `years` argument.
- `s_progressive`: Normalized progressive Mann–Kendall-Sneyers statistics.
- `s_regressive`: Normalized regressive Mann–Kendall-Sneyers statistics.
- `bound`: Critical confidence bound for significance based on `alpha`.
- `crossing_df`: Crossing points, including indices, years, statistics, and AMS.
- `change_df`: Subset of `crossing.df` with statistically significant crossings.
- `p_value`: Two-sided p-value derived from the maximum crossing statistic.
- `reject`: Logical. If `TRUE`, the 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

plot_mks_test

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)
eda_mks_test(data, years)
```

eda_mk_test *Mann–Kendall Trend Test*

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 nonparametric and accounts for tied observations in the data. The null hypothesis assumes there is no monotonic trend.

Usage

```
eda_mk_test(data, alpha = 0.05, quiet = TRUE)
```

Arguments

`data`	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.
`alpha`	Numeric scalar in $[0.01, 0.1]$. The significance level for confidence intervals or hypothesis tests. Default is 0.05.
`quiet`	Logical scalar. If `FALSE`, prints a summary of of the statistical test to the console. Default is `TRUE`.

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 list containing the test results, including:

- `data`: The `data` argument.
- `s_statistic`: The 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. If `TRUE`, the null hypothesis is rejected at `alpha`.
- `msg`: A character string summarizing the result, printed if `quiet` = `FALSE`.

See Also

eda_bbmk_test

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
eda_mk_test(data)
```

eda_pettitt_test *Pettitt Test for Abrupt Changes in the Mean of a Time Series*

Description

Performs the nonparametric Pettitt test to detect a single change point in the mean of a time series, used to identify abrupt shifts in hydrological data.

Usage

```
eda_pettitt_test(data, years, alpha = 0.05, quiet = TRUE)
```

Arguments

`data`	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.
`years`	Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.
`alpha`	Numeric scalar in $[0.01, 0.1]$. The significance level for confidence intervals or hypothesis tests. Default is 0.05.
`quiet`	Logical scalar. If `FALSE`, prints a summary of of the statistical test to the console. Default is `TRUE`.

Details

The Pettitt test is a rank-based nonparametric test that evaluates the hypothesis of a change point in the 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 list containing the test results, including:

- `data`: The `data` argument.
- `years`: The `years` argument.
- `u_t`: Numeric vector of absolute U-statistics for all time indices.
- `k_statistic`: Numeric scalar.Tthe maximum absolute U-statistic.
- `k_critical`: Numeric scalar. The critical K-statistic value for given `alpha`.
- `p_value`: Numeric scalar. Approximate p-value for the test.
- `change_index`: Integer scalar. Index of the detected change point (0 if none).
- `change_year`: Integer scalar. Year of the detected change point (0 if none).
- `reject`: Logical scalar. If `TRUE`, the null hypothesis was rejected.
- `msg`: Character scalar. A formatted summary message describing the test result.

See Also

plot_pettitt_test

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)
eda_pettitt_test(data, years)
```

`eda_pp_test` *Phillips–Perron Unit Root Test*

Description

Applies the Phillips–Perron (PP) test to check for a unit root in annual maximum series (AMS) data. The null hypothesis is that the series contains a unit root (and is thus nonstationary). This implementation of the PP test assumes the time series has both stationary drift and a linear trend.

Usage

```
eda_pp_test(data, alpha = 0.05, quiet = TRUE)
```

Arguments

`data`	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.
`alpha`	Numeric scalar in $[0.01, 0.1]$. The significance level for confidence intervals or hypothesis tests. Default is 0.05.
`quiet`	Logical scalar. If `FALSE`, prints a summary of of the statistical test to the console. Default is `TRUE`.

Details

The implementation of this test is based on the **aTSA** package, which interpolates p-values from a table of critical values presented in Fuller W. A. (1996). The critical values are only available for $\alpha \geq 0.01$. Therefore, a reported p-value of 0.01 indicates $p \leq 0.01$.

Value

List; the test results, consisting of:

- `data`: The `data` argument.
- `statistic`: The Z-statistic used to perform the test.
- `p.value`: Reported p-value from the test. See notes on interpolation thresholds.
- `reject`: Logical. If `TRUE`, the null hypothesis was rejected at `alpha`.
- `msg`: Character string summarizing the test result, printed if `quiet` = `FALSE`.

References

Fuller, W. A. (1996). Introduction to statistical time series, second ed., Wiley, New York.

See Also

eda_kpss_test

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
eda_pp_test(data)
```

`eda_runs_test` *Wald–Wolfowitz Runs Test for Randomness*

Description

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 significance level `alpha`, and a decision is returned along with a human-readable summary message.

Usage

```
eda_runs_test(results, alpha = 0.05, quiet = TRUE)
```

Arguments

`results`	A fitted linear model produced by eda_sens_trend.
`alpha`	Numeric scalar in $[0.01, 0.1]$. The significance level for confidence intervals or hypothesis tests. Default is 0.05.
`quiet`	Logical scalar. If `FALSE`, prints a summary of of the statistical test to the console. Default is `TRUE`.

Details

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

Value

A list containing the test results, including:

- `residuals`: Numeric vector of residual values from a fitted linear model.
- `n`: The length of the residuals vector after removing the median.
- `n_plus`, `n_minus`: The number of residuals above/below the median.
- `runs`: The number of runs in the transformed sequence of residuals.
- `statistic`: The runs test statistic, computed using `runs`.
- `p_value`: P-value from the Wald–Wolfowitz runs test applied to residuals.
- `reject`: Logical. If TRUE, the null hypothesis of random residuals is rejected.
- `msg`: Character string summarizing the test result, printed if `quiet` = `FALSE`.

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

plot_runs_test, eda_sens_trend

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)
sens <- eda_sens_trend(data, years)
eda_runs_test(sens)
```

---

eda_sens_trend *Sen's Trend Estimator*

Description

Computes Sen's trend estimator for a univariate time series.

Usage

```
eda_sens_trend(data, years, quiet = TRUE)
```

Arguments

`data`	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.
`years`	Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.
`quiet`	Logical scalar. If `FALSE`, prints a summary of of the statistical test to the console. Default is `TRUE`.

Details

Sen's slope estimator is a robust, nonparametric 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 list containing the estimated trend:

- `data`: The `data` argument.
- `years`: The `years` argument.
- `slope`: Median slope of all pairwise data-year combinations.
- `intercept`: Median intercept estimate of the fitted line.
- `residuals`: Vector of residuals between observed and fitted values.
- `msg`: Character string summarizing the results.

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

eda_runs_test, plot_sens_trend

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)
eda_sens_trend(data, years)
```

---

eda_spearman_test *Spearman Test for Autocorrelation*

Description

Performs the Spearman rank autocorrelation test on annual maximum series (AMS) data to check for autocorrelation at various lags. Reports the smallest lag where the autocorrelation is not statistically significant at the given significance level.

Usage

```
eda_spearman_test(data, alpha = 0.05, quiet = TRUE)
```

Arguments

`data`	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.
`alpha`	Numeric scalar in $[0.01, 0.1]$. The significance level for confidence intervals or hypothesis tests. Default is 0.05.
`quiet`	Logical scalar. If `FALSE`, prints a summary of of the statistical test to the console. Default is `TRUE`.

Details

For each lag from 1 to $n - 3$, the function computes the Spearman correlation coefficient between the AMS series and its lagged version. The first lag with an insignificant autocorrelation coefficient returned as `least.lag`.

Value

A list containing the test results, including:

- `data`: The `data` argument.
- `rho`: Numeric vector of autocorrelation estimates for lags 1 to $n - 3$.
- `sig`: Logical vector indicating which lags exhibit significant autocorrelation.
- `least.lag`: The smallest lag at which the autocorrelation is insignificant.
- `reject`: Logical. If `TRUE`, then `least.lag` > `0`.
- `msg`: Character string summarizing the test result, printed if `quiet` = `FALSE`.

See Also

cor.test, eda_bbmk_test, plot_spearman_test

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
eda_spearman_test(data)
```

---

eda_white_test *White Test for Heteroskedasticity*

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

```
eda_white_test(data, years, alpha = 0.05, quiet = TRUE)
```

Arguments

`data`	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.
`years`	Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.
`alpha`	Numeric scalar in $[0.01, 0.1]$. The significance level for confidence intervals or hypothesis tests. Default is 0.05.
`quiet`	Logical scalar. If `FALSE`, prints a summary of of the statistical test to the console. Default is `TRUE`.

Details

The White test regresses the squared residuals from a primary linear model `lm(data ~ years)` against both the original regressor and its square. The test statistic is calculated as nR^2, where R^2 is the coefficient of determination 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 heteroskedasticity in the residuals.

Value

A list containing the results of the White test:

- `data`: The `data` argument.
- `years`: The `years` argument.
- `r_squared`: Coefficient of determination from the auxiliary regression.
- `statistic`: White test statistic based on sample size and `r.squared`.
- `p_value`: The p-value derived from a Chi-squared distribution with `df = 2`.
- `reject`: Logical. If `TRUE`, 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

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)
eda_white_test(data, years)
```

`fit_lmom_fast` *Helper Function for L-moments Parameter Estimation*

Description

A helper function used by fit_lmom_methods. This function does not validate parameters and is intended for internal use.

Usage

```
fit_lmom_fast(data, model)
```

Arguments

`data`	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.
`model`	Character scalar. A three-character code indicating a distribution family. Must be one of: `"GUM"`, `"NOR"`, `"LNO"`, `"GEV"`, `"GLO"`, `"GNO"`, `"PE3"`, `"LP3"`, or `"WEI"`.

Value

A numeric vector of parameters:

- If `model` is `"GUM"`, `"NOR"`, or `"LNO"`, returns a vector of length 2.
- Otherwise, returns a vector of length 3.

References

Hosking, J.R.M. & Wallis, J.R., 1997. Regional frequency analysis: an approach based on L-Moments. Cambridge University Press, New York, USA.

See Also

lmom_sample, fit_lmom_kappa, fit_lmom_methods

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
fit_lmom_fast(data, "PE3")
```

fit_lmom_kappa *L-Moments Parameter Estimation for the Kappa Distribution*

Description

This functions estimates the parameters of the four-parameter Kappa distribution using the method of L-moments. Since there is no known closed form solution for the parameters in terms of the L-moments, the parameters are computed numerically using Newton-Raphson iteration.

Usage

```
fit_lmom_kappa(data)
```

Arguments

data	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.

Details

First, the sample L-moments of the data are computed using the lmom_sample method. Then, the optim function is used to determine the parameters by minimizing the `sumquad.tau3tau4` helper function. The implementation of this routine is based on the deprecated `homtest` package.

Value

A numeric vector of four parameters in the order location, scale, shape (1), shape (2).

References

Hosking, J.R.M. & Wallis, J.R., 1997. Regional frequency analysis: an approach based on L-Moments. Cambridge University Press, New York, USA.

See Also

lmom_sample, fit_lmom_fast, fit_lmom_methods

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
fit_lmom_kappa(data)
```

`fit_lmom_methods` *Parameter Estimation with L-Moments*

Description

Estimate the parameters of one of nine different distributions (`GUM`, `NOR`, `LNO`, `GEV`, `GLO`, `GNO`, `PE3`, `LP3`, and `WEI`) using the method of L-moments.

Usage

```
fit_lmom_gum(data)

fit_lmom_nor(data)

fit_lmom_lno(data)

fit_lmom_gev(data)

fit_lmom_glo(data)

fit_lmom_gno(data)

fit_lmom_pe3(data)

fit_lmom_lp3(data)

fit_lmom_wei(data)
```

Arguments

`data`	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.

Details

First, the sample L-moments of the data are computed using the lmom_sample method. Then formulas from Hosking (1997) are used to compute the parameters from the L-moments. Distributions `GNO`, `PE3`, and `LP3` use a rational approximation to compute the parameters.

Value

A numeric vector of parameters:

- If `model` is `"GUM"`, `"NOR"`, or `"LNO"`, returns a vector of length 2.
- Otherwise, returns a vector of length 3.

References

Hosking, J.R.M. & Wallis, J.R., 1997. Regional frequency analysis: an approach based on L-Moments. Cambridge University Press, New York, USA.

See Also

fit_lmom_fast, fit_lmom_kappa

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
fit_lmom_lp3(data)
```

fit_maximum_likelihood

Maximum Likelihood Parameter Estimation

Description

Estimates parameters of a probability distribution with an optional trend by maximizing the log-likelihood. Initial values are obtained through L-moment parameter estimation, and optimization is performed via nlminb with repeated perturbations if needed.

Usage

```
fit_maximum_likelihood(data, model, prior = NULL, years = NULL, trend = NULL)
```

Arguments

`data` Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.

`model` Character scalar. A three-character code indicating a distribution family. Must be one of: `"GUM"`, `"NOR"`, `"LNO"`, `"GEV"`, `"GLO"`, `"GNO"`, `"PE3"`, `"LP3"`, or `"WEI"`.

`prior` Numeric vector of length 2. Specifies the Beta prior shape parameters (p, q) for the shape parameter κ. Only used when `model = "GEV"`.

`years` Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.

`trend` Named list indicating which distribution parameters are modeled as nonstationary. Must contain:

- `location`: Logical scalar. If `TRUE`, the location parameter has a trend.
- `scale`: Logical scalar. If `TRUE`, the scale parameter has a trend.

Details

1. Calls fit_lmom_fast on `data` to obtain initial parameter estimates.
2. Initializes trend parameters to zero if necessary.
3. For `WEI` models, sets the location parameter to zero to ensure support.
4. Defines an objective function using the likelihood functions loglik_fast or general_loglik_fast.
5. Runs nlminb with box constraints. Attempts optimization up to 100 times if a maximum cannot be found.

Value

A list containing the results of parameter estimation:

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

Note

Although the more modern optim function is preferred over nlminb, we use nlminb because it supports infinite values of the likelihood function.

See Also

loglik_fast, general_loglik_fast, fit_lmom_fast, nlminb

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)
trend <- list(location = TRUE, scale = FALSE)
fit_maximum_likelihood(data, "GNO", NULL, years, trend)
```

---

general_loglik_fast

Generalized Log-Likelihood Helper Function

Description

A helper function used by general_loglik_gev. This function does not validate parameters and is intended for internal use.

Usage

```
general_loglik_fast(data, model, params, prior, years, trend)
```

Arguments

data — Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.

model — Character scalar. A three-character code indicating a distribution family. Must be one of: "GUM", "NOR", "LNO", "GEV", "GLO", "GNO", "PE3", "LP3", or "WEI".

params — Numeric vector of distribution parameters, in the order (location, scale, shape). The length must be between 2 and 5, depending on the specified model and trend.

prior — Numeric vector of length 2. Specifies the Beta prior shape parameters (p, q) for the shape parameter κ. Only used when model = "GEV".

years — Numeric vector of observation years corresponding to data. Must be the same length as data and strictly increasing.

`trend` Named list indicating which distribution parameters are modeled as nonstationary. Must contain:

- `location`: Logical scalar. If `TRUE`, the location parameter has a trend.
- `scale`: Logical scalar. If `TRUE`, the scale parameter has a trend.

Value

Numeric scalar. The generalized log-likelihood value.

See Also

general_loglik_gev.

Examples

```
# Initialize data, params, prior, years, and trend
data <- rnorm(n = 100, mean = 100, sd = 10)
params <- c(0, 1, 1, 0, 0)
prior <- c(5, 10)
years <- seq(from = 1901, to = 2000)
trend <- list(location = TRUE, scale = TRUE)

# Compute the generalized log-likelihood
general_loglik_fast(data, "GEV", params, prior, years, trend)
```

`general_loglik_gev` *Generalized Log-Likelihood Functions for GEV Models*

Description

Computes the generalized log-likelihood for stationary and nonstationary variants of the Generalized Extreme Value (GEV) distribution with a Beta prior on the shape parameter.

Usage

```
general_loglik_gev(data, params, prior, years = NULL, trend = NULL)
```

Arguments

`data` Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.

`params` Numeric vector of distribution parameters, in the order (location, scale, shape). The length must be between 2 and 5, depending on the specified `model` and `trend`.

`prior` Numeric vector of length 2. Specifies the Beta prior shape parameters (p, q) for the shape parameter κ. Only used when `model = "GEV"`.

`years` Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.

`trend` Named list indicating which distribution parameters are modeled as nonstationary. Must contain:

- `location`: Logical scalar. If `TRUE`, the location parameter has a trend.
- `scale`: Logical scalar. If `TRUE`, the scale parameter has a trend.

Details

The generalized log-likelihood is defined as sum of the log-likelihood of the specified model and the log-density of the Beta prior with parameters (p, q). The contribution of the prior is:

$$\log \pi(\kappa) = (p-1)\log(0.5-\kappa) + (q-1)\log(0.5+\kappa) - \log(\beta(p,q))$$

Value

Numeric scalar. The generalized log-likelihood value.

See Also

general_loglik_fast

Examples

```
# Initialize data, params, prior, years, and trend
data <- rnorm(n = 100, mean = 100, sd = 10)
params <- c(0, 1, 1, 0)
prior <- c(5, 10)
years <- seq(from = 1901, to = 2000)
trend <- list(location = TRUE, scale = FALSE)

# Compute the generalized log-likelihood
general_loglik_gev(data, params, prior, years, trend)
```

lmom_fast *Helper Function for L-moments Ratios*

Description

A helper function used by lmom_theoretical. This function does not validate parameters and is intended for internal use.

Usage

```
lmom_fast(model, params)
```

Arguments

`model`	Character scalar. A three-character code indicating a distribution family. Must be one of: `"GUM"`, `"NOR"`, `"LNO"`, `"GEV"`, `"GLO"`, `"GNO"`, `"PE3"`, `"LP3"`, or `"WEI"`.
`params`	Numeric vector of distribution parameters, in the order (location, scale, shape). The length must be between 2 and 5, depending on the specified `model` and `trend`.

Value

A numeric vector of length 4 containing:

- λ_1: L-mean
- λ_2: L-variance
- τ_3: L-skewness
- τ_4: L-kurtosis

Note

L-moment ratios for `NOR`/`LNO` and `PE3`/`LP3` are identical since it L-moments for the `LNO`/`LP3` distributions are compared with the sample L-moments of the logarithm of the data internally.

References

Hosking, J.R.M. & Wallis, J.R., 1997. Regional frequency analysis: an approach based on L-Moments. Cambridge University Press, New York, USA.

See Also

lmom_theoretical

Examples

```
lmom_fast("GLO", c(0, 1, 0))
```

`lmom_sample` *Sample L-moments*

Description

Computes the first four sample L-moments and L-moment ratios from a numeric vector of data. L-moments are linear combinations of order statistics that provide robust alternatives to conventional moments, with advantages in parameter estimation for heavy-tailed and skewed distributions.

Usage

```
lmom_sample(data)
```

Arguments

`data`	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.

Details

Given probability weighted moments $\beta_0, \beta_1, \beta_2, \beta_3$, the first four sample L-moments are:

- $l_1 = \beta_0$
- $l_2 = 2\beta_1 - \beta_0$
- $l_3 = 6\beta_2 - 6\beta_1 + \beta_0$
- $l_4 = 20\beta_3 - 30\beta_2 + 12\beta_1 - \beta_0$

Then, the sample L-skewness is $t_3 = l_3/l_2$ and the sample L-kurtosis is $t_4 = l_4/l_2$.

Value

A numeric vector containing the first four L-moments and L-moment ratios:

- `l1`: L-mean
- `l2`: L-variance
- `t3`: L-skewness
- `t4`: L-kurtosis

Note

This function has no parameter validation because uncertainty_bootstrap makes many calls to it. If you are getting errors, ensure that `data` is a positive, numeric vector.

References

Hosking, J. R. M. (1990). L-moments: Analysis and estimation of distributions using linear combinations of order statistics. *Journal of the Royal Statistical Society: Series B (Methodological)*, 52(1), 105–124.

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
lmom_sample(data)
```

`lmom_theoretical` *Theoretical L-moments of Probability Distributions*

Description

Computes the first four L-moments and L-moment ratios of seven different probability distributions (`GUM`, `NOR`, `GEV`, `GLO`, `GNO`, `PE3`, and `WEI`) given the parameters of the distribution.

Usage

```
lmom_theoretical_gum(params)

lmom_theoretical_nor(params)

lmom_theoretical_gev(params)

lmom_theoretical_glo(params)

lmom_theoretical_gno(params)

lmom_theoretical_pe3(params)

lmom_theoretical_wei(params)
```

Arguments

`params`	Numeric vector of distribution parameters, in the order (location, scale, shape). The length must be between 2 and 5, depending on the specified `model` and `trend`.

Details

The distributions `GUM`, `NOR`, `GEV`, `GLO`, and `WEI` have closed-form solutions for the L-moments and L-moment ratios in terms of the parameters. The distributions `GNO`, `PE3` use rational approximations of the L-moment ratios from Hosking (1997).

Value

A numeric vector of length 4 containing:

- λ_1: L-mean
- λ_2: L-variance
- τ_3: L-skewness
- τ_4: L-kurtosis

References

Hosking, J.R.M. & Wallis, J.R., 1997. Regional frequency analysis: an approach based on L-Moments. Cambridge University Press, New York, USA.

See Also

lmom_fast

Examples

```
lmom_theoretical_gev(c(0, 1, 0))
```

`loglik_fast` *Log-Likelihood Helper Function*

Description

A helper function used by loglik_methods. This function does not validate parameters and is intended for internal use.

Usage

```
loglik_fast(data, model, params, years, trend)
```

Arguments

`data` Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.

`model` Character scalar. A three-character code indicating a distribution family. Must be one of: `"GUM"`, `"NOR"`, `"LNO"`, `"GEV"`, `"GLO"`, `"GNO"`, `"PE3"`, `"LP3"`, or `"WEI"`.

`params` Numeric vector of distribution parameters, in the order (location, scale, shape). The length must be between 2 and 5, depending on the specified `model` and `trend`.

`years` Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.

`trend` Named list indicating which distribution parameters are modeled as nonstationary. Must contain:

- `location`: Logical scalar. If `TRUE`, the location parameter has a trend.
- `scale`: Logical scalar. If `TRUE`, the scale parameter has a trend.

Value

Numeric scalar. The log-likelihood value.

See Also

loglik_methods

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
params <- c(0, 1, 0)
years <- seq(from = 1901, to = 2000)
trend <- list(location = FALSE, scale = FALSE)
loglik_fast(data, "GEV", params, years, trend)
```

loglik_methods *Log-Likelihood Functions for Probability Models*

Description

Compute the log-likelihood value for stationary and non-stationary variants of nine different distributions: `"GUM"`, `"NOR"`, `"LNO"`, `"GEV"`, `"GLO"`, `"GNO"`, `"PE3"`, `"LP3"`, or `"WEI"`.

Usage

```
loglik_gum(data, params, years = NULL, trend = NULL)

loglik_nor(data, params, years = NULL, trend = NULL)

loglik_lno(data, params, years = NULL, trend = NULL)

loglik_gev(data, params, years = NULL, trend = NULL)

loglik_glo(data, params, years = NULL, trend = NULL)

loglik_gno(data, params, years = NULL, trend = NULL)

loglik_pe3(data, params, years = NULL, trend = NULL)

loglik_lp3(data, params, years = NULL, trend = NULL)

loglik_wei(data, params, years = NULL, trend = NULL)
```

Arguments

`data`	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.
`params`	Numeric vector of distribution parameters, in the order (location, scale, shape). The length must be between 2 and 5, depending on the specified `model` and `trend`.
`years`	Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.
`trend`	Named list indicating which distribution parameters are modeled as nonstationary. Must contain: • `location`: Logical scalar. If `TRUE`, the location parameter has a trend. • `scale`: Logical scalar. If `TRUE`, the scale parameter has a trend.

Value

Numeric scalar. The log-likelihood value.

See Also

loglik_fast

Examples

```
# Initialize data, params, years, and trend
data <- rnorm(n = 100, mean = 100, sd = 10)
params <- c(0, 1, 1, 0)
years <- seq(from = 1901, to = 2000)
trend <- list(location = TRUE, scale = FALSE)

# Compute the log-likelihood
loglik_gno(data, params, years, trend)
```

`model_diagnostics` *Evaluate Goodness-of-Fit for Fitted Flood Models*

Description

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

Usage

```
model_diagnostics(
  data,
  model,
  params,
  uncertainty,
  years = NULL,
  trend = NULL,
  alpha = 0.05,
  pp_formula = "Weibull"
)
```

Arguments

`data` Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.

`model` Character scalar. A three-character code indicating a distribution family. Must be one of: `"GUM"`, `"NOR"`, `"LNO"`, `"GEV"`, `"GLO"`, `"GNO"`, `"PE3"`, `"LP3"`, or `"WEI"`.

`params` Numeric vector of distribution parameters, in the order (location, scale, shape). The length must be between 2 and 5, depending on the specified `model` and `trend`.

`uncertainty` List; estimated reutrn levels and confidence intervals generated by uncertainty_bootstrap or uncertainty_rfpl.

`years` Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.

`trend` Named list indicating which distribution parameters are modeled as nonstationary. Must contain:

- `location`: Logical scalar. If `TRUE`, the location parameter has a trend.

- scale: Logical scalar. If TRUE, the scale parameter has a trend.

alpha — Numeric scalar in $[0.01, 0.1]$. The significance level for confidence intervals or hypothesis tests. Default is 0.05.

pp_formula — Character (1); string specifying the plotting position formula. Must be one of "Weibull", "Blom", "Cunnane", "Gringorten", or "Hazen"

Value

List containing model assessment metrics:

- data: The data argument.
- estimates: Quantile estimates for empirical return periods.
- R2: Coefficient of determination from linear regression of estimates vs. data.
- RMSE: Root mean squared error of quantile estimates.
- bias: Mean bias of quantile estimates.
- AIC: Akaike Information Criterion.
- BIC: Bayesian Information Criterion.
- AIC_MLL: Akaike Information Criterion, computed using MLE.
- BIC_MLL: Bayesian Information Criterion, computed using MLE.
- AW: Average width of the confidence interval(s).
- POC: Percent of observations covered by the confidence interval(s).
- CWI: Confidence width index, a metric that combines AW and POC.

See Also

uncertainty_bootstrap, uncertainty_rfpl, fit_maximum_likelihood

Examples

```
# Initialize example data and params
data <- rnorm(n = 100, mean = 100, sd = 10)
params <- c(100, 10)

# Perform uncertainty analysis
uncertainty <- uncertainty_bootstrap(data, "NOR", "L-moments")

# Evaluate model diagnostics
model_diagnostics(data, "NOR", params, uncertainty)
```

`plot_ams_data` *Plot Annual Maximum Series (AMS) Data*

Description

Generates a plot of annual maximum series data with an optional trend line.

Usage

```
plot_ams_data(data, years, show_line = TRUE, ...)
```

Arguments

`data`	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.
`years`	Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.
`show_line`	If `TRUE` (default), draw a line through the AMS data.
`...`	Optional named arguments: 'title', 'xlabel', and 'ylabel'.

Value

`ggplot`; A plot containing:

- Black dots at each (data, year) pair.
- An optional line thorugh the data points (if `show_line` == `TRUE`)

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)
plot_ams_data(data, years)
```

`plot_bbmk_test` *Plot Block-Bootstrap Mann–Kendall Test Results*

Description

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

Usage

```
plot_bbmk_test(results, ...)
```

Arguments

`results`	List of BB-MK test results generated by eda_bbmk_test.
`...`	Optional named arguments: 'title', 'xlabel', and 'ylabel'.

Value

`ggplot`; A plot containing:

- A gray histogram of the distribution of bootstrapped S-statistics.
- A red vertical line at the lower and upper confidence bounds.
- A black vertical line at the observed S-statistic.

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
results <- eda_bbmk_test(data, samples = 1000L)
plot_bbmk_test(results)
```

`plot_lmom_diagram` *Plot L-Moment Ratio Diagram*

Description

Generates a plot of L-moment ratios with the L-skewness on the x-axis and L-kurtosis on the y-axis. Plots the sample and log-sample L-moment ratios alongside the theoretical L-moment ratios for a set of candidate distributions. If the selection method is select_ldistance or select_lkurtosis, the plot will include a small inset around the L-moment ratios of the recommended distribution.

Usage

```
plot_lmom_diagram(results, ...)
```

Arguments

`results`	List of model selection results generated by select_ldistance, select_lkurtosis, or select_zstatistic.
`...`	Optional named arguments: ’title’, ’xlabel’, and ’ylabel’.

Value

`ggplot`; plot object containing the L-moment ratio diagram, with:

- L-moment ratio curves for each 3-parameter distribution.
- Points for the L-moment ratios of each 2-parameter distribution.
- Sample and log-sample L-moment ratio (t_3, t_4) points.

See Also

select_ldistance, select_lkurtosis, select_zstatistic

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
results <- select_ldistance(data)
plot_lmom_diagram(results)
```

plot_mks_test *Plot Mann–Kendall–Sneyers (MKS) Test Results*

Description

Constructs a two-panel visualization of the MKS test. The upper panel plots the normalized progressive and regressive Mann–Kendall S-statistics over time, with dashed confidence bounds and potential trend-change points. The lower panel contains the annual maximum series (AMS) data with the change points highlighted points along with an optional trend line.

Usage

```
plot_mks_test(results, show_line = TRUE, ...)
```

Arguments

`results`	A list generated by eda_mks_test.
`show_line`	If `TRUE` (default), draw a fitted line through the AMS data.
`...`	Optional named arguments: 'title', 'top_xlabel', 'top_ylabel', 'bottom_xlabel' and 'bottom_ylabel'.

Value

A `patchwork` object with two `ggplot2` panels stacked vertically.

See Also

eda_mks_test

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)
results <- eda_mks_test(data, years)
plot_mks_test(results)
```

plot_model_diagnostics *Plot Model Assessment Results*

Description

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

Usage

```
plot_model_diagnostics(results, ...)
```

Arguments

`results`	List; model assessment results generated by model_diagnostics.
`...`	Optional named arguments: 'title', 'xlabel', and 'ylabel'.

Value

`ggplot`; a plot containing:

- A black line representing a perfect model with no deviation from the data.
- Red points marking the estimated quantiles against the observed quantiles.

Examples

```
# Initialize example data and params
data <- rnorm(n = 100, mean = 100, sd = 10)
params <- c(100, 10)

# Perform uncertainty analysis
uncertainty <- uncertainty_bootstrap(data, "NOR", "L-moments")

# Evaluate model diagnostics
results <- model_diagnostics(data, "NOR", params, uncertainty)

# Generate a model assessment plot
plot_model_diagnostics(results)
```

`plot_nsffa` *Plot Return Levels and Confidence Intervals for NS-FFA*

Description

Constructs a time–series plot of estimated return levels and their confidence intervals for up to five slices. Each slice is displayed in a distinct color. Confidence bounds are shown as semi-transparent ribbons, and the point estimates are overlaid as solid lines. Return periods are shown on a logarithmic scale.

Usage

```
plot_nsffa(results, ...)
```

Arguments

`results`	A list of up to five result objects, each a list containing `periods`, `estimates`, `ci_lower`, `ci_upper`, and a `slice` year.
`...`	Optional named arguments: 'title', 'xlabel', and 'ylabel'.

Value

`ggplot`; a plot with one line and ribbon per slice.

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)

# Run the uncertainty bootstrap at slices 1920, 1960, 2000
results <- uncertainty_bootstrap(
    data,
   "GEV",
   "MLE",
   years = years,
   trend = list(location = TRUE, scale = FALSE),
   slices = c(1920, 1960, 2000),
   samples = 1000L
)

# Generate the plot
plot_nsffa(results)
```

`plot_pettitt_test` *Plot Results from the Pettitt Change-Point Test*

Description

Creates a two-panel visualization of the Mann–Whitney–Pettitt test. The upper panel plots the Pettitt U_t statistic over time along with the significance threshold and potential change point. The lower panel displays the annual maximum series (AMS) data with an optional trend line, estimates of the mean, and potential change point.

Usage

```
plot_pettitt_test(results, show_line = TRUE, ...)
```

Arguments

`results`	A list generated by eda_pettitt_test.
`show_line`	If `TRUE` (default), draw a fitted line through the AMS data.
`...`	Optional named arguments: 'title', 'top_xlabel', 'top_ylabel', 'bottom_xlabel' and 'bottom_ylabel'.

Value

A `patchwork` object with two `ggplot2` panels stacked vertically.

See Also

eda_pettitt_test

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)
results <- eda_pettitt_test(data, years)
plot_pettitt_test(results)
```

`plot_runs_test` *Plot Runs Test Results*

Description

Generates a residual plot of Sen's estimator applied to annual maximum series (AMS) data with a horizontal dashed line at zero and an annotation of the Runs test .

Usage

```
plot_runs_test(results, ...)
```

Arguments

`results`	A list of runs test results generated by eda_runs_test.
`...`	Optional named arguments: 'title', 'xlabel', and 'ylabel'.

Value

`ggplot`; a plot containing:

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

See Also

eda_runs_test

Examples

```
# Initialize data and years
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)

# Generate runs test plot
sens <- eda_sens_trend(data, years)
results <- eda_runs_test(sens)
plot_runs_test(results, "mean")
```

`plot_sens_trend` *Plot Sen's Trend Estimator*

Description

Produces a scatter plot of the annual maximum series (AMS) data or its variance against time, optionally overlaid with Sen's trend estimator of the mean and/or variability.

Usage

```
plot_sens_trend(
  data,
  years,
  mean_trend = NULL,
  variability_trend = NULL,
  show_line = TRUE,
  ...
)
```

Arguments

`data`	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.
`years`	Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.
`mean_trend`	A list of Sen's trend estimates generated by eda_sens_trend.
`variability_trend`	Trend in variability generated by eda_sens_trend.
`show_line`	If `TRUE` (default), draw a fitted line through the AMS data.
`...`	Optional named arguments: 'title', 'xlabel', and 'ylabel'.

Value

`ggplot`; a plot containing:

- Gray points for each year's AMS value.
- Optional gray line connecting the AMS data if `show_line = TRUE`.
- A solid black line representing a constant mean, if `mean_trend == NULL`.
- A solid blue line representing a trend in the mean, if `mean_trend != NULL`.
- A dashed black line representing constant variability, if `variability_trend == NULL`.
- A dashed blue line representing a trend in variability, if `variability_trend != NULL`.
- The equation for the trend in the mean, written in the form $mx + b$.

See Also

eda_sens_trend

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)
mean_trend <- eda_sens_trend(data, years)
plot_sens_trend(data, years, meant_trend = mean_trend)
```

plot_sffa *Plot Return Levels and Confidence Intervals for S-FFA*

Description

Constructs a time–series plot of estimated return levels along 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 return periods are displayed on a logarithmic scale.

Usage

```
plot_sffa(results, ...)
```

Arguments

`results`	A list of estimated return levels and confidence intervals generated by uncertainty_bootstrap or uncertainty_rfpl.
`...`	Optional named arguments: 'title', 'xlabel', and 'ylabel'.

Value

`ggplot`; a plot showing:

- A semi-transparent gray ribbon between `results$ci_lower` and `results$ci_upper`.
- A solid black line for the point estimates.

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)
results <- uncertainty_bootstrap(data, "WEI", "L-moments")
plot_sffa(results)
```

`plot_spearman_test` *Plot Spearman's Rho Autocorrelation*

Description

Visualizes Spearman's rho autocorrelation coefficients with shaded points indicating statistical significance.

Usage

```
plot_spearman_test(results, ...)
```

Arguments

`results`	A list generated by eda_spearman_test.
`...`	Optional named arguments: 'title', 'xlabel', and 'ylabel'.

Value

`ggplot`; a plot showing:

- Vertical segments from $y = 0$ up to each ρ value at its lag.
- Filled circles at each lag, filled black if serial correlation is detected.

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
results <- eda_spearman_test(data)
plot_spearman_test(results)
```

`quantile_fast` *Helper Function for Quantile Functions*

Description

A helper function used by quantile_methods.

Usage

```
quantile_fast(p, model, params, slice, trend)
```

Arguments

`p` Numeric vector of probabilities between 0 and 1 with no missing values.

`model` Character scalar. A three-character code indicating a distribution family. Must be one of: `"GUM"`, `"NOR"`, `"LNO"`, `"GEV"`, `"GLO"`, `"GNO"`, `"PE3"`, `"LP3"`, or `"WEI"`.

`params` Numeric vector of distribution parameters, in the order (location, scale, shape). The length must be between 2 and 5, depending on the specified `model` and `trend`.

`slice` Numeric scalar specifying the year at which to evaluate the quantiles of a nonstationary probability distribution. The slice does not have to be an element of the `years` argument. Note that if `trend$location` and `trend$scale` are both `FALSE`, this argument will have no effect the output of the function.

`trend` Named list indicating which distribution parameters are modeled as nonstationary. Must contain:

- `location`: Logical scalar. If `TRUE`, the location parameter has a trend.
- `scale`: Logical scalar. If `TRUE`, the scale parameter has a trend.

Value

A numeric vector of quantiles with the same length as `p`.

See Also

quantile_methods

Examples

```
# Initialize p, params, and trend
p <- runif(n = 10)
params <- c(0, 1, 1, 0)
trend <- list(location = FALSE, scale = TRUE)

# Compute the log-likelihood in the year 2000
quantile_fast(p, "GEV", params, 2000, trend)
```

`quantile_methods` *Quantile Functions for Probability Models*

Description

Compute the quantiles for stationary and nonstationary variants of nine different distributions: `"GUM"`, `"NOR"`, `"LNO"`, `"GEV"`, `"GLO"`, `"GNO"`, `"PE3"`, `"LP3"`, or `"WEI"`.

Usage

```
quantile_gum(p, params, slice = 1900, trend = NULL)

quantile_nor(p, params, slice = 1900, trend = NULL)

quantile_lno(p, params, slice = 1900, trend = NULL)

quantile_gev(p, params, slice = 1900, trend = NULL)

quantile_glo(p, params, slice = 1900, trend = NULL)

quantile_gno(p, params, slice = 1900, trend = NULL)

quantile_pe3(p, params, slice = 1900, trend = NULL)

quantile_lp3(p, params, slice = 1900, trend = NULL)

quantile_wei(p, params, slice = 1900, trend = NULL)

quantile_kap(p, params, slice = 1900, trend = NULL)
```

Arguments

p	Numeric vector of probabilities between 0 and 1 with no missing values.
params	Numeric vector of distribution parameters, in the order (location, scale, shape). The length must be between 2 and 5, depending on the specified `model` and `trend`.
slice	Numeric scalar specifying the year at which to evaluate the quantiles of a non-stationary probability distribution. The slice does not have to be an element of the `years` argument. Note that if `trend$location` and `trend$scale` are both `FALSE`, this argument will have no effect the output of the function.
trend	Named list indicating which distribution parameters are modeled as nonstationary. Must contain: • `location`: Logical scalar. If `TRUE`, the location parameter has a trend. • `scale`: Logical scalar. If `TRUE`, the scale parameter has a trend.

Value

A numeric vector of quantiles with the same length as `p`.

See Also

quantile_fast

Examples

```
# Initialize p and params
p <- runif(n = 10)
params <- c(0, 1, 0)

# Compute the quantiles
quantile_wei(p, params)
```

`select_ldistance` *L-Distance Method for Distribution Selection*

Description

Selects a distribution from a set of candidate distributions by minimizing the Euclidean distance between the theoretical L-moment ratios (τ_3, τ_4) and the sample L-moment ratios (t_3, t_4).

Usage

```
select_ldistance(data)
```

Arguments

`data` Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.

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 curve. For two-parameter distributions (Gumbel, Normal, Log-Normal), the theoretical L-moment ratios are compared directly with the sample L-moment ratios. The distribution with the minimum distance is selected. If a distribution is fit to log-transformed data (Log-Normal or Log-Pearson Type III), the L-moment ratios for the log-transformed sample are used instead.

Value

A list with the results of distribution selection:

- `method`: `"L-distance"`.
- `data`: The `data` argument.
- `metrics`: A list of L-distance metrics for each candidate distribution.
- `recommendation`: The name of the distribution with the smallest L-distance.

See Also

lmom_sample, select_lkurtosis, select_zstatistic, plot_lmom_diagram

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
select_ldistance(data)
```

select_lkurtosis *L-Kurtosis Method for Distribution Selection*

Description

Selects a probability distribution by minimizing the absolute distance between the theoretical L-kurtosis (τ_4) and the sample L-kurtosis (t_4). Only supports 3-parameter distributions. The sample L-skewness (t_3) is used to determine the shape parameter.

Usage

```
select_lkurtosis(data)
```

Arguments

`data`	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.

Details

This method computes the distance between the sample and theoretical L-kurtosis values at a fixed L-skewness. For three parameter distributions, the shape parameter that best replicates the sample L-skewness is derived using optim.

Value

A list with the results of distribution selection:

- `method`: `"L-kurtosis"`.
- `data`: The `data` argument.
- `metrics`: A list of L-kurtosis metrics for each distribution.
- `recommendation`: Name of the distribution with the smallest L-kurtosis metric

See Also

lmom_sample, select_ldistance, select_zstatistic, plot_lmom_diagram

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
select_lkurtosis(data)
```

select_zstatistic *Z-Statistic Method for Distribution Selection*

Description

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

Usage

```
select_zstatistic(data, samples = 10000L)
```

Arguments

`data`	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.
`samples`	Integer scalar. The number of bootstrap samples. Default is 10000.

Details

The method performs model selection using both raw and log-transformed data. The distributions which use the raw AMS data are GEV, GLO, PE3, GNO, and WEI. The LP3 distribution uses log-transformed data.

The Z-statistic is determined by fitting a four-parameter Kappa distribution to the raw and log-transformed data. Then, bootstrapped samples from this Kappa distribution The L-moments of these bootstrapped samples are used to estimate the Z-statistic for each distribution.

Value

A list with the results of distribution selection:

- `method`: `"Z-selection"`.
- `data`: The `data` argument.
- `metrics`: List of computed Z-statistics for each candidate distribution.
- `recommendation`: Name of the distribution with the smallest Z-statistic.
- `reg_params`: Kappa distribution parameters for the AMS data.
- `reg_bias_t4`: Bias of the L-kurtosis from the AMS bootstrap.
- `reg_std_t4`: Standard deviation of the L-kurtosis from the AMS bootstrap.
- `log_params`: Kappa distribution parameters for the log-AMS data.
- `log_bias_t4`: Bias of the L-kurtosis from the log-AMS bootstrap.
- `log_std_t4`: Standard deviation of the L-kurtosis from the log-AMS bootstrap.

See Also

select_ldistance, select_lkurtosis, fit_lmom_kappa, quantile_fast, plot_lmom_diagram

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
select_zstatistic(data)
```

---

uncertainty_bootstrap *Parametric Bootstrap Confidence Intervals for Flood Quantile Estimates*

Description

Computes estimates and confidence intervals for return levels at return periods 2, 5, 10, 20, 50, and 100 years using the parametric bootstrap method. This function supports a variety of probability models and parameter estimation methods.

Usage

```
uncertainty_bootstrap(
  data,
  model,
  method,
  prior = NULL,
  years = NULL,
  trend = NULL,
  slices = 1900,
  alpha = 0.05,
  samples = 10000L,
  periods = c(2, 5, 10, 20, 50, 100)
)
```

Arguments

`data` Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.

`model` Character scalar. A three-character code indicating a distribution family. Must be one of: `"GUM"`, `"NOR"`, `"LNO"`, `"GEV"`, `"GLO"`, `"GNO"`, `"PE3"`, `"LP3"`, or `"WEI"`.

`method` Character scalar specifying the estimation method. Must be `"L-moments"`, `"MLE"`, or `"GMLE"`.

`prior` Numeric vector of length 2. Specifies the Beta prior shape parameters (p, q) for the shape parameter κ. Only used when `model = "GEV"`.

`years` Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.

`trend` Named list indicating which distribution parameters are modeled as nonstationary. Must contain:

- `location`: Logical scalar. If `TRUE`, the location parameter has a trend.
- `scale`: Logical scalar. If `TRUE`, the scale parameter has a trend.

`slices` Numeric vector specifying the years at which to evaluate the return levels confidence intervals of a nonstationary probability distribution. The slices do not have to be elements of the `years` argument.

`alpha` Numeric scalar in $[0.01, 0.1]$. The significance level for confidence intervals or hypothesis tests. Default is 0.05.

`samples` Integer scalar. The number of bootstrap samples. Default is 10000.

`periods` Numeric vector used to set the return periods for FFA. All entries must be greater than or equal to 1.

Details

The bootstrap procedure samples from the fitted distribution via inverse transform sampling. For each bootstrapped sample, the parameters are re-estimated using the specified `method`. Then, the bootstrapped parameters are used to compute a new set of bootstrapped quantiles. Confidence intervals are obtained from the empirical nonexceedance probabilities of the bootstrapped quantiles.

Value

A list of lists containing the return levels and confidence intervals for each slice. Each sub-list contains:

- `method`: "Bootstrap"
- `estimates`: Estimated quantiles for each return period.
- `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; `c(2, 5, 10, 20, 50, 100)`.
- `slice`: The value of `slice` argument.
- `trend`: The value of `trend` argument.

Note

The parametric bootstrap is known to give unreasonably wide confidence intervals for small datasets. If this function detects a confidence interval that is 5+ times wider than the return levels themselves, it will return an error and recommend RFPL uncertainty quantification (with `uncertainty_rfpl`).

See Also

fit_lmom_fast, fit_maximum_likelihood, lmom_sample, quantile_fast, plot_sffa, plot_nsffa

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)
uncertainty_bootstrap(data, "WEI", "L-moments")
```

uncertainty_rfpl *Regula-Falsi Confidence Intervals for Flood Quantile Estimates*

Description

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

Usage

```
uncertainty_rfpl(
  data,
  model,
  prior = NULL,
  years = NULL,
  trend = NULL,
  slices = 1900,
  alpha = 0.05,
  eps = 0.01,
  periods = c(2, 5, 10, 20, 50, 100)
)
```

Arguments

`data`	Numeric vector of annual maximum series (AMS) values. Must be strictly positive, finite, and not missing.
`model`	Character scalar. A three-character code indicating a distribution family. Must be one of: `"GUM"`, `"NOR"`, `"LNO"`, `"GEV"`, `"GLO"`, `"GNO"`, `"PE3"`, `"LP3"`, or `"WEI"`.
`prior`	Numeric vector of length 2. Specifies the Beta prior shape parameters (p, q) for the shape parameter κ. Only used when `model = "GEV"`.
`years`	Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.
`trend`	Named list indicating which distribution parameters are modeled as nonstationary. Must contain: • `location`: Logical scalar. If `TRUE`, the location parameter has a trend. • `scale`: Logical scalar. If `TRUE`, the scale parameter has a trend.
`slices`	Numeric vector specifying the years at which to evaluate the return levels confidence intervals of a nonstationary probability distribution. The slices do not have to be elements of the `years` argument.
`alpha`	Numeric scalar in $[0.01, 0.1]$. The significance level for confidence intervals or hypothesis tests. Default is 0.05.
`eps`	Numeric scalar. The log-likelihood tolerance for the Regula-Falsi convergence (default is 0.01).
`periods`	Numeric vector used to set the return periods for FFA. All entries must be greater than or equal to 1.

Details

1. Fits the model using fit_maximum_likelihood to obtain parameter estimates and log-likelihood.
2. Defines an objective function $f(y_p, p)$ based on the chi-squared distribution.
3. Iteratively brackets the root by scaling initial guesses by 0.05 until $f(y_p, p)$ changes sign.
4. Uses the Regula Falsi method to solve $f(y_p, p) = 0$ for each return-period probability.
5. Returns lower and upper confidence bounds at significance level `alpha` given the return level estimates.

Value

A list of lists containing the return levels and confidence intervals for each slice. Each sub-list contains:

- `method`: Either "RFPL" or "RFGPL"
- `estimates`: Estimated quantiles for each return period.
- `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 `c(2, 5, 10, 20, 50, 100)` in years.
- `slice`: The value of `slice` argument.
- `trend`: The value of `trend` argument.

Note

Although the more modern optim function is preferred over nlminb, we use nlminb because it supports infinite values of the likelihood function.

RFPL uncertainty quantification can be numerically unstable for some datasets. If this function encounters an issue, it will return an error and recommend using the parametric bootstrap (`uncertainty_bootstrap`) instead.

See Also

quantile_fast, uncertainty_bootstrap, plot_sffa, plot_nsffa

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
uncertainty_rfpl(data, "GLO")
```

Index