Package ‘ffaframework’

June 26, 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. 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 testthat, vdiffr

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Contents

`ams.decomposition` *Decompose Annual Maximum Streamflow Data*

Description

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

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

Usage

```
ams.decomposition(data, years, signature)
```

Arguments

`data` Numeric; a vector of annual maximum streamflow data.

`years` Numeric; a vector of years with the same length as `data`.

`signature` Character (1); Indicates trend(s) to remove. Must be one of:

- `NULL`: Do not remove any trends.
- `"10"`: Remove linear trend in the mean.
- `"01"`: Remove linear trend in the variance.
- `"11"`: Remove trends in both the mean and variance.

Details

Internally, the function does the following:

1. If `signature == NULL`, returns `data` without modification.
2. Constructs a `covariate` based on `years` using the formula `(years - 1900) / 100`.
3. If `signature == "10"`, fits Sen's trend estimator to `data` and `covariate` and removes the fitted linear mean trend.
4. If `signature == "01"`, computes moving-window variances using mw.variance, fits Sen's trend estimator to those variances, and rescales the series to remove trends in the variance.
5. If `signature == "11"`, applies (3) then (4) sequentially.
6. Ensures all returned values are greater than 1 by shifting the decomposed data.

Value

Numeric; a vector with the same length as `data` and `years` containing the decomposed AMS data with the specified trend(s) removed.

See Also

mw.variance, sens.trend

Examples

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

`assessment.plot` *Plot Model Assessment Results*

Description

Creates a quantile–quantile plot comparing observed annual maximum streamflow (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

```
assessment.plot(data, assessment)
```

Arguments

`data`	Numeric; a vector of annual maximum streamflow data.
`assessment`	List; model assessment results generated by model.assessment.

Value

`ggplot`; a plot containing:

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

Examples

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

# Perform uncertainty analysis
uncertainty <- sb.uncertainty(data, years, "NOR", "L-moments")

# Perform model assessment
assessment <- model.assessment(data, years, "NOR", params, uncertainty)

# Generate a model assessment plot
assessment.plot(data, assessment)
```

`bbmk.plot` *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

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

Arguments

`results`	List; BB-MK test results generated by bbmk.test.

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 <- bbmk.test(data, n_sim = 1000)
bbmk.plot(results)
```

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 streamflow (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

```
bbmk.test(data, alpha = 0.05, n_sim = 10000, quiet = TRUE)
```

Arguments

`data`	Numeric; a vector of annual maximum streamflow data.
`alpha`	Numeric (1); the significance level (default is 0.05).
`n_sim`	Integer (1); the number of bootstrap simulations (default is 10000).
`quiet`	Logical (1); 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.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:

- `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

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
bbmk.test(data, n_sim = 1000)
```

`gll-functions` *Generalized Log-Likelihood Functions for GEV Models*

Description

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

Usage

```
gllgev(data, params, prior, years = NULL)

gllgev100(data, params, prior, years)

gllgev110(data, params, prior, years)
```

Arguments

`data`	Numeric; a vector of annual maximum streamflow data.
`params`	Numeric; a vector of parameters. Must have the correct length for the model.
`prior`	Numeric (2); a vector of parameters(p, q) of the Beta prior on κ.
`years`	Numeric; a vector of years with the same length as `data`. Only required for `gllgev100()` and `gllgev110()`.

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:

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

Each function corresponds to a different parameterization of the GEV model:

- `gllgev()`: Stationary location and scale, 3 parameters.
- `gllgev100()`: Time-varying location, stationary scale, 4 parameters.
- `gllgev110()`: Time-varying location and scale, 5 parameters.

Value

Numeric (1); the generalized log-likelihood value.

Note

The `gllgev`, `gllgev100`, and `gllgev110` functions perform extensive parameter validation, which can be slow. If you plan to call these methods often, it is recommended to use the `gllxxx` helper function instead.

See Also

gllxxx

Examples

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

# Compute the generalized log-likelihood
gllgev100(data, params, prior, years)
```

`gllxxx` *Generalized Log-Likelihood Helper Function*

Description

A helper function used by gllgev, gllgev100, and gllgev110.

Usage

```
gllxxx(name, signature, data, params, prior, covariate = NULL)
```

Arguments

`name`	Character (1); the name of the probability distribution.
`signature`	Character (1); the non-stationary signature (`NULL`, `"10"`, or `"11"`).
`data`	Numeric; a vector of annual maximum streamflow data.
`params`	Numeric; a vector of parameters. Must have the correct length for the model.
`prior`	Numeric (2); a vector of parameters (p, q) of the Beta prior on κ.
`covariate`	Numeric; a vector with the same length as `data`. Required if `signature` is `"10"` or `"11"`.

Value

Numeric (1); the generalized log-likelihood value.

Note

This function does not perform parameter validation, which improves performance but may cause unpredictable behaviour. Use at your own risk.

See Also

gll-functions

Examples

```
# Initialize data, params, params, prior, and covariate
data <- rnorm(n = 100, mean = 100, sd = 10)
params <- c(0, 1, 1, 0, 0)
prior <- c(5, 10)
covariate <- seq(0, 1, length.out = 100)

# Compute the generalized log-likelihood
gllxxx("GEV", "11", data, params, prior, covariate)
```

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

Description

Performs the KPSS unit root test on annual maximum streamflow (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 non-stationary.

Usage

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

Arguments

`data`	Numeric; a vector of annual maximum streamflow data.
`alpha`	Numeric (1); the significance level (default is 0.05).
`quiet`	Logical (1); if FALSE, prints a summary of results (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

List; the test results, consisting of:

- `statistic`: The KPSS test statistic.
- `p.value`: The reported p-value from the test. See notes regarding discrete thresholds.
- `reject`: Logical. TRUE if 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

pp.test

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
kpss.test(data)
```

ld.selection *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

```
ld.selection(data)
```

Arguments

`data`	Numeric; a vector of annual maximum streamflow data.

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

List; results of distribution selection:

- `method`: `"L-distance"`
- `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, lk.selection, z.selection, optim

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
ld.selection(data)
```

`lk.selection` *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). For 3-parameter distributions, we use the shape parameter that best replicates the sample L-skewness (t_3) of the data.

Usage

```
lk.selection(data)
```

Arguments

`data`	Numeric; a vector of annual maximum streamflow data.

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

List; results of distribution selection:

- `method`: `"L-kurtosis"`
- `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, ld.selection, z.selection, optim

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
lk.selection(data)
```

`llv-functions` *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`, and `WEI`).

Usage

```
llvgum(data, params, years = NULL)

llvgum10(data, params, years)

llvgum11(data, params, years)

llvnor(data, params, years = NULL)

llvnor10(data, params, years)

llvnor11(data, params, years)

llvlno(data, params, years = NULL)

llvlno10(data, params, years)

llvlno11(data, params, years)

llvgev(data, params, years = NULL)

llvgev100(data, params, years)

llvgev110(data, params, years)

llvglo(data, params, years = NULL)

llvglo100(data, params, years)

llvglo110(data, params, years)

llvgno(data, params, years = NULL)

llvgno100(data, params, years)

llvgno110(data, params, years)

llvpe3(data, params, years = NULL)

llvpe3100(data, params, years)

llvpe3110(data, params, years)
```

```
llvlp3(data, params, years = NULL)

llvlp3100(data, params, years)

llvlp3110(data, params, years)

llvwei(data, params, years = NULL)

llvwei100(data, params, years)

llvwei110(data, params, years)
```

Arguments

`data`	Numeric; a vector of annual maximum streamflow data.
`params`	Numeric; a vector of parameters. Must have the correct length for the model.
`years`	Numeric; a vector of years with the same length as `data`. Required for non-stationary models, which end in `10`, `11`, `100`, or `110`.

Details

The log-likelihood is the logarithm of the probability density function. For two-parameter distributions (`GUM`, `NOR`, `LNO`), there are three different `llv` functions:

- `llv...()`: Stationary location and scale, 2 parameters.
- `llv...10()`: Time-varying location, stationary scale, 3 parameters.
- `llv...11()`: Time-varying location and scale, 4 parameters.

For three-parameter distributions (`GEV`, `GLO`, `GNO`, `PE3`, `LP3`, `WEI`), there are also three different `llv` functions:

- `llv...()`: Stationary location and scale, 3 parameters.
- `llv...100()`: Time-varying location, stationary scale, 4 parameters.
- `llv...110()`: Time-varying location and scale, 5 parameters.

Value

Numeric (1); the log-likelihood value.

Note

The `llv...` functions perform extensive parameter validation, which can be slow. If you plan to make calls these methods often, it is recommended to use the llvxxx helper function instead.

See Also

llvxxx

Examples

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

# Compute the log-likelihood
llvgno100(data, params, years)
```

---

llvxxx *Log-Likelihood Helper Function*

Description

A helper function used by llv-functions.

Usage

```
llvxxx(name, signature, data, params, covariate = NULL)
```

Arguments

`name`	Character (1); the name of the probability distribution.
`signature`	Character (1); the non-stationary signature (`NULL`, `"10"`, or `"11"`).
`data`	Numeric; a vector of annual maximum streamflow data.
`params`	Numeric; a vector of parameters. Must have the correct length for the model.
`covariate`	Numeric; a vector with the same length as `data`. Required if `signature` is `"10"` or `"11"`.

Value

Numeric (1); the log-likelihood value.

Note

This function does not perform parameter validation, which improves performance at the cost of potentially unpredictable behaviour. Use at your own risk.

See Also

llv-functions

Examples

```
# Initialize data and params
data <- rnorm(n = 100, mean = 100, sd = 10)
params <- c(0, 1, 0)

# Compute the log-likelihood
llvxxx("GEV", NULL, data, params)
```

`lmom.plot` *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 ld.selection or lk.selection, the plot will include a small inset around the L-moment ratios of the recommended distribution.

Usage

```
lmom.plot(data, results)
```

Arguments

`data`	Numeric; a vector of annual maximum streamflow data.
`results`	List; output of ld.selection, lk.selection, or z.selection.

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

ld.selection, lk.selection, z.selection

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
results <- ld.selection(data)
lmom.plot(data, results)
```

`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; a vector of annual maximum streamflow data.

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

Numeric (4); a vector containing the first four L-moments and L-moment ratios:

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

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)
```

`lmr-functions` *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

```
lmrgum(params)

lmrnor(params)

lmrgev(params)

lmrglo(params)

lmrgno(params)

lmrpe3(params)

lmrwei(params)
```

Arguments

`params`	Numeric; a vector of parameters. Must have the correct length for the model.

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` and `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.

Examples

```
lmrgev(c(0, 1, 0))
```

`lmrxxx` *Helper Function for L-moments Ratios*

Description

A helper function used by lmr-functions.

Usage

```
lmrxxx(model, params)
```

Arguments

`model`	Character; `GUM`, `NOR`, `LNO`, `GEV`, `GLO`, `GNO`, `PE3`, `LP3`, or `WEI`.
`params`	Numeric; a vector of parameters. • Numeric (2) if `model` is `GUM`, `NOR`, or `LNO`. • Numeric (3) if `model` is `GEV`, `GLO`, `GNO`, `PE3`, `LP3`, or `WEI`.

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 is assumed that the `LNO`/`LP3` L-moments will be compared to the sample L-moments of the logarithm of the data.

References

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

Examples

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

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 non-parametric and accounts for tied observations in the data.

Usage

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

Arguments

`data`	Numeric; a vector of annual maximum streamflow data.
`alpha`	Numeric (1); the significance level (default is 0.05).
`quiet`	Logical (1); if FALSE, prints a summary of results (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

List; the test results:

- `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

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
mk.test(data)
```

mks.plot *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 streamflow (AMS) data with the change points highlighted points along with an optional trend line.

Usage

```
mks.plot(data, years, results, show_trend = TRUE)
```

Arguments

`data`	Numeric; a vector of annual maximum streamflow data.
`years`	Numeric; a vector of years with the same length as `data`.
`results`	List; results of the MKS test, generated by mks.test.
`show_trend`	Logical; if TRUE (default), draw a fitted line through the AMS data.

Value

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

See Also

mks.test

Examples

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

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 streamflow (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

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

Arguments

`data` Numeric; a vector of annual maximum streamflow data.

`years` Numeric; a vector of years with the same length as `data`.

`alpha` Numeric (1); the significance level (default is 0.05).

`quiet` Logical (1); if FALSE, prints a summary of results (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 normal quantiles and the maximum absolute crossing statistic.

Value

List; results of the MKS test:

- `s.progressive`: Normalized progressive Mann–Kendall statistics.
- `s.regressive`: Normalized regressive Mann–Kendall statistics.
- `bound`: Critical confidence bound for significance based on `alpha`.
- `crossing.df`: Data frame of crossing points with indices, years, statistics, and AMS.
- `change.df`: Subset of `crossing.df` with statistically significant crossings.
- `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

mks.plot

Examples

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

`mle.estimation` *Maximum Likelihood Parameter Estimation*

Description

Estimates parameters of a probability distribution (`GUM`, `NOR`, `LNO`, `GEV`, `GLO`, `GNO`, `PE3`, `LP3`, or `WEI`) or non-stationary variant 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

```
mle.estimation(data, years, model, prior = NULL)
```

Arguments

`data` Numeric; a vector of annual maximum streamflow data.

`years` Numeric; a vector of years with the same length as `data`.

`model` Character (1); string specifying the probability model. 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.

`prior` Numeric (2); optional vector of parameters (p, q) that specifies the parameters of a Beta prior on κ. Only works with models `GEV`, `GEV100`, and `GEV110`.

Details

1. Calls pelxxx on `data` to obtain initial parameter estimates.
2. Initializes trend parameters to zero if there is a trailing signature.
3. For `WEI` models, sets the location parameter to zero to ensure support.
4. Defines an objective function as the negative of the llvxxx function.
5. Runs nlminb with box constraints. Attempts optimization up to 100 times.

Value

List; results of parameter estimation:

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

See Also

llvxxx, gllxxx, pelxxx, nlminb

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)
mle.estimation(data, years, "GNO100")
```

`model.assessment` *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 bootstrapped confidence intervals.

Usage

```
model.assessment(
  data,
  years,
  model,
  params,
  uncertainty,
  pp.formula = "Weibull",
  alpha = 0.05
)
```

Arguments

`data` Numeric; a vector of annual maximum streamflow data.

`years` Numeric; a vector of years with the same length as `data`.

`model` Character; string specifying the probability model. 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.

`params` Numeric; a vector of fitted model parameters generated by pelxxx or mle.estimation.

`uncertainty` List; quantiles and confidence intervals generated by sb.uncertainty or rfpl.uncertainty.

`pp.formula` Character (1); string specifying the plotting position formula. One of `"Weibull"`, `"Blom"`, `"Cunnane"`, `"Gringorten"`, or `"Hazen"` (default is "Weibull").

`alpha` Numeric (1); the significance level (default is 0.05).

Value

List; model assessment metrics:

- `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 the maximum log-likelihood.
- `BIC_MLL`: Bayesian Information Criterion, computed using the maximum log-likelihood.
- `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

sb.uncertainty, rfpl.uncertainty, lm, approx

Examples

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

# Perform uncertainty analysis
uncertainty <- sb.uncertainty(data, years, "NOR", "L-moments")

# Perform model assessment
model.assessment(data, years, "NOR", params, uncertainty)
```

`mw.variance` *Estimate Variance for Annual Maximum Streamflow Data*

Description

This function estimates the standard deviation of a vector of annual maximum streamflow (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

```
mw.variance(data, years, size = 10, step = 5)
```

Arguments

`data` Numeric; a vector of annual maximum streamflow data.

`years` Numeric; a vector of years with the same length as `data`.

`size` Integer (1); the number of consecutive indices in each moving window. Must be a positive integer less than or equal to `length(data)` (default is 10). If `length(data) < size`, an error is raised.

`step` Integer (1); the offset (in indices) between successive moving windows. Must be a positive integer (default is 5).

Value

A list with two entries:

- `years`: Numeric; the mean year within each window.
- `std`: Numeric; the standard deviation of the data within each window.

Examples

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

`pel-functions` *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

```
pelgum(data)

pelnor(data)

pellno(data)

pelgev(data)

pelglo(data)

pelgno(data)

pelpe3(data)

pellp3(data)

pelwei(data)
```

Arguments

`data`	Numeric; a vector of annual maximum streamflow data.

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

Numeric; a vector of parameters.

References

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

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
pellp3(data)
```

`pelkap` *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

```
pelkap(data)
```

Arguments

`data`	Numeric; a vector of annual maximum streamflow data.

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

Numeric (4); a vector of parameters.

References

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

Examples

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

pelxxx *Helper Function for L-moments Parameter Estimation*

Description

A helper function used by pel-functions.

Usage

```
pelxxx(model, data)
```

Arguments

`model`	Character; `GUM`, `NOR`, `LNO`, `GEV`, `GLO`, `GNO`, `PE3`, `LP3`, or `WEI`.
`data`	Numeric; a vector of annual maximum streamflow data.

Value

Numeric; a vector of parameters.

- Numeric (2) if `model` is `GUM`, `NOR`, or `LNO`.
- Numeric (3) if `model` is `GEV`, `GLO`, `GNO`, `PE3`, `LP3`, or `WEI`.

References

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

Examples

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

pettitt.plot *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 streamflow (AMS) data with an optional trend line, estimates of the mean, and potential change point.

Usage

```
pettitt.plot(data, years, results, show_trend = TRUE)
```

Arguments

`data`	Numeric; a vector of annual maximum streamflow data.
`years`	Numeric; a vector of years with the same length as `data`.
`results`	List; results of the MKS test, generated by pettitt.test.
`show_trend`	Logical; if TRUE (default), draw a fitted line through the AMS data.

Value

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

See Also

pettitt.test

Examples

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

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

Arguments

`data`	Numeric; a vector of annual maximum streamflow data.
`years`	Numeric; a vector of years with the same length as `data`.
`alpha`	Numeric (1); the significance level (default is 0.05).
`quiet`	Logical (1); if FALSE, prints a summary of results (default is 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. TRUE if the null hypothesis was rejected.
- msg: Formatted summary message describing the test result.

See Also

pettitt.plot

Examples

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

pp.test *Phillips–Perron Unit Root Test*

Description

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

Usage

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

Arguments

data	Numeric; a vector of annual maximum streamflow data.
alpha	Numeric (1); the significance level (default is 0.05).
quiet	Logical (1); if FALSE, prints a summary of results (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:

- `statistic`: The Z-statistic used to perform the test.
- `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

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

See Also

kpss.test

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
pp.test(data)
```

`qnt-functions` *Quantile Functions for Probability Models*

Description

Compute the quantiles for stationary and non-stationary variants of nine different distributions (`GUM`, `NOR`, `LNO`, `GEV`, `GLO`, `GNO`, `PE3`, `LP3`, and `WEI`).

Usage

```
qntgum(p, params, years = NULL)

qntgum10(p, params, years)

qntgum11(p, params, years)

qntnor(p, params, years = NULL)

qntnor10(p, params, years)

qntnor11(p, params, years)

qntlno(p, params, years = NULL)

qntlno10(p, params, years)

qntlno11(p, params, years)

qntgev(p, params, years = NULL)
```

```
qntgev100(p, params, years)

qntgev110(p, params, years)

qntglo(p, params, years = NULL)

qntglo100(p, params, years)

qntglo110(p, params, years)

qntgno(p, params, years = NULL)

qntgno100(p, params, years)

qntgno110(p, params, years)

qntpe3(p, params, years = NULL)

qntpe3100(p, params, years)

qntpe3110(p, params, years)

qntlp3(p, params, years = NULL)

qntlp3100(p, params, years)

qntlp3110(p, params, years)

qntwei(p, params, years = NULL)

qntwei100(p, params, years)

qntwei110(p, params, years)

qntkap(p, params, years = NULL)
```

Arguments

`p` Numeric; a vector of probabilities between 0 and 1.

`params` Numeric; a vector of parameters. Must have the correct length for the model.

`years` Numeric; a vector of years with the same length as `data`. Required for non-stationary models, which end in `10`, `11`, `100`, or `110`.

Details

The quantile function is the inverse of the cumulative distribution function. For the two-parameter distributions (`GUM`, `NOR`, `LNO`), there are three different `qnt` functions:

- `qnt...()`: Stationary location and scale, 2 parameters.
- `qnt...10()`: Time-varying location, stationary scale, 3 parameters.
- `qnt...11()`: Time-varying location and scale, 4 parameters.

For three-parameter distributions (GEV, GLO, GNO, PE3, LP3, WEI), there are also three qnt functions:

- qnt...(): Stationary location and scale, 3 parameters.
- qnt...100(): Time-varying location, stationary scale, 4 parameters.
- qnt...110(): Time-varying location and scale, 5 parameters.

Value

If p or years is a scalar, returns a numeric vector. Otherwise, returns a matrix.

Note

The qnt..., functions perform extensive parameter validation, which can be slow. If you plan to make many calls to these methods, it is recommended to use the qntxxx helper function instead.

See Also

qntxxx

Examples

```
# Initialize p, years, and params
p <- runif(n = 10)
years <- seq(from = 1901, to = 2000)
params <- c(0, 1, 1, 1, 0)

# Compute the quantiles
qntwei110(p, params, years)
```

---

qntxxx *Helper Function for Quantile Functions*

Description

A helper function used by qnt-functions.

Usage

```
qntxxx(name, signature, p, params, covariate = 0)
```

Arguments

name	Character (1); the name of the probability distribution.
signature	Character (1); the non-stationary signature (NULL, "10", or "11").
p	Numeric; a vector of probabilities. Must be between 0 and 1.
params	Numeric; a vector of parameters. Must have the correct length for the model.
covariate	Numeric; a vector with the same length as data. Required if signature is "10" or "11".

Value

If p or years is a scalar, returns a numeric vector. Otherwise, returns a matrix.

See Also

qnt-functions

Examples

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

# Compute the log-likelihood
qntxxx("GEV", NULL, p, params)
```

---

rfpl.uncertainty *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

```
rfpl.uncertainty(
  data,
  years,
  model,
  slices = "last",
  alpha = 0.05,
  eps = 0.01,
  prior = NULL
)
```

Arguments

data Numeric; a vector of annual maximum streamflow data.

years Numeric; a vector of years with the same length as data.

model Character (1); string specifying the probability model. 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.

slices Character (1) or Numeric; specifies the years at which to compute the estimates and confidence intervals (default is "last").

- "all": returns estimates for all values in years.
- "first": returns estimates for first year in the dataset.
- "last": returns estimates for last year in the dataset.

- Passing a numeric vector to `slices` allows for custom values.

If the chosen model is stationary, the results will be the same for all slices.

`alpha` Numeric (1); the significance level (default is 0.05).

`eps` Numeric (1); tolerance for the Regula-Falsi convergence (default is 0.01).

`prior` Numeric (2); optional vector of parameters (p, q) that specifies the parameters of a Beta prior on κ. Only works with models `GEV`, `GEV100`, and `GEV110`.

Details

1. Fits the model using mle.estimation 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 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 level alpha and the quantile estimates.

Value

List; quantiles and confidence intervals. Each year maps to a sub-list:

- `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).`

See Also

qntxxx, sb.uncertainty, nlminb

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)
rfpl.uncertainty(data, years, "GLO110")
```

`runs.plot` *Plot Runs Test Results*

Description

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

Usage

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

Arguments

`years`	Numeric; a vector of years with the same length as `data`.
`residuals`	Numeric; A vector of residuals produced by sens.trend.
`results`	List; runs test results generated by runs.test.
`name`	Character; either `"sens-variance"` or `"sens-mean"`.
`show_trend`	Logical; if TRUE (default), draw a fitted line through the AMS data.

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 p-value: X.XXX" in the plot area.

See Also

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
residuals <- sens.trend(data, years)$residuals
results <- runs.test(residuals)
runs.plot(years, residuals, results, "sens-mean")
```

`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

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

Arguments

`residuals`	Numeric; A vector of residuals produced by sens.trend.
`alpha`	Numeric (1); the significance level (default is 0.05).
`quiet`	Logical (1); if FALSE, prints a summary of results (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 non-random clustering, which may indicate that a linear model is inappropriate for the data.

Value

List; test results, including:

- `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 the 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.plot, sens.trend

Examples

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

# Perform the runs test
residuals <- sens.trend(data, years)$residuals
runs.test(residuals)
```

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

Description

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

Usage

```
sb.uncertainty(
  data,
  years,
  model,
  method,
  slices = "last",
  n_sim = 10000,
  alpha = 0.05,
  prior = NULL
)
```

Arguments

`data` Numeric; a vector of annual maximum streamflow data.

`years` Numeric; a vector of years with the same length as `data`.

`model` Character (1); string specifying the probability model. 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.

`method` Character (1); string specifying the estimation method. Must be `"L-moments"`, `"MLE"`, or `"GMLE"`.

`slices` Character (1) or Numeric; specifies the years at which to compute the estimates and confidence intervals (default is "last").

- `"all"`: returns estimates for all values in `years`.
- `"first"`: returns estimates for first year in the dataset.
- `"last"`: returns estimates for last year in the dataset.
- Passing a numeric vector to `slices` allows for custom values.

If the chosen model is stationary, the results will be the same for all slices.

`n_sim` Integer (1); the number of bootstrap simulations (default is 10000).

`alpha` Numeric (1); the significance level (default is 0.05).

`prior` Numeric (2); optional vector of parameters (p, q) that specifies the parameters of a Beta prior on κ. Only works with models `GEV`, `GEV100`, and `GEV110`.

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 non-exceedance probabilities of the bootstrapped quantiles.

Value

List; quantiles and confidence intervals. Each year maps to a sub-list:

- `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)`.

See Also

pelxxx, mle.estimation, lmom.sample, quantile

Examples

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

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

Description

Produces a scatter plot of the annual maximum streamflow (AMS) data or its variance against time, overlaid with Sen's trend estimator and an annotation of the fitted equation.

Usage

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

Arguments

`data`	Numeric; a vector of annual maximum streamflow data.
`years`	Numeric; a vector of years with the same length as `data`.
`results`	List; runs test results generated by sens.trend.
`name`	Character; either `"sens-variance"` or `"sens-mean"`.
`show_trend`	Logical; if TRUE (default), draw a fitted line through the AMS data.

Value

`ggplot`; a plot containing:

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

See Also

sens.trend

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)
results <- sens.trend(data, years)
sens.plot(data, years, results, "sens-mean")
```

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

Description

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

Usage

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

Arguments

`data` Numeric; a vector of annual maximum streamflow data.

`years` Numeric; a vector of years with the same length as `data`.

`quiet` Logical (1); if FALSE, prints a summary of 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

List; the estimated trend:

- `sens.slope`: Median slope of all pairwise data-year combinations.
- `sens.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

runs.test

Examples

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

spearman.plot *Plot Spearman's Rho Autocorrelation*

Description

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

Usage

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

Arguments

`results`	List; autocorrelation coefficients generated by spearman.test.

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 <- spearman.test(data)
spearman.plot(results)
```

spearman.test *Spearman Test for Autocorrelation*

Description

Performs the Spearman rank autocorrelation test on annual maximum streamflow (AMS) data to check for autocorrelation at various lags. Reports the first lag where the autocorrelation is no longer statistically significant at the given significance level.

Usage

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

Arguments

`data`	Numeric; a vector of annual maximum streamflow data.
`alpha`	Numeric (1); the significance level (default is 0.05).
`quiet`	Logical (1); if FALSE, prints a summary of results (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

List; test results, including:

- `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 the autocorrelation is not statistically significant.
- `reject`: Logical. TRUE if `least.lag` > `0`.
- `msg`: Character string summarizing the test result (printed if `quiet` = `FALSE`).

See Also

cor.test, bbmk.test

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
spearman.test(data)
```

`uncertainty.plot` *Plot Return Levels and Confidence Intervals*

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.

If the specified model is non-stationary, the `slice` argument is required and the return levels are reported as "Effective Return Periods".

Usage

```
uncertainty.plot(model, results, slice = 1)
```

Arguments

`model`	Character (1); string specifying the probability model. 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.
`results`	List; generated by sb.uncertainty or rfpl.uncertainty.
`slice`	Character (1); the anchor year for non-stationary return level estimates. Required for models ending in `10`, `100`, `11`, or `110`.

Value

ggplot; a plot showing:

- A dashed black line for both the lower and upper confidence bounds.
- A semi-transparent gray ribbon between results$ci_lower and results$ci_upper.
- A solid blue line for the point estimates.

Examples

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

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

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

Arguments

data	Numeric; a vector of annual maximum streamflow data.
years	Numeric; a vector of years with the same length as data.
alpha	Numeric (1); the significance level (default is 0.05).
quiet	Logical (1); if FALSE, prints a summary of results (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

List; results of the White test:

- r.squared: Coefficient of determination from the auxiliary regression.
- statistic: White test statistic based on sample size and r.squared.
- p.value: P-value derived from a Chi-squared distribution with 2 degrees of freedom.
- reject: Logical. TRUE if the null hypothesis is rejected at significance 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)
white.test(data, years)
```

`z.selection` *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-score is selected.

Usage

```
z.selection(data, n_sim = 20000)
```

Arguments

`data`	Numeric; a vector of annual maximum streamflow data.
`n_sim`	Integer (1); the number of bootstrap simulations (default is 20000).

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

List; results of distribution selection:

- `method`: `"Z-selection"`
- `params`: Kappa distribution parameters for the raw AMS data.
- `log_params`: Kappa distribution parameters for the log-transformed AMS data.
- `bootstrap`: Bias and standard deviation of the estimated L-kurtosis.
- `distance`: List of computed Z-statistics for each candidate distribution.
- `recommendation`: Name of the distribution with the smallest Z-statistic.

See Also

ld.selection, lk.selection, pelkap, qntxxx

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
z.selection(data)
```

Index