Package: ffaframework (via r-universe)

November 20, 2026

Title Flood Frequency Analysis Framework

Description Tools to support systematic and reproducible workflows for both stationary and nonstationary flood frequency analysis, with applications extending to other hydroclimate extremes, such as precipitation frequency analysis. This package implements the FFA framework proposed by Vidrio- Sahagún et al. (2024) <doi:10.1016/j.envsoft.2024.105940>, originally developed in 'MATLAB', now adapted for the 'R' environment. This work was funded by the Flood Hazard Identification and Mapping Program of Environment and Climate Change Canada, as well as the Canada Research Chair (Tier 1) awarded to Dr. Pietroniro.

Version 0.1.3

URL https://rileywheadon.github.io/ffa-framework/

BugReports https://github.com/rileywheadon/ffa-framework/issues

License AGPL (>= 3)

Encoding UTF-8

LazyData true

Depends R (>= 4.4.0)

Imports ggplot2, patchwork, httr, stats, glue, jsonlite, base64enc

Suggests knitr, rmarkdown, spelling, testthat, vdiffr

VignetteBuilder knitr

Config/Needs/website rmarkdown

Language en-US

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Riley Wheadon [aut, cre], Cuauhtémoc Vidrio-Sahagún [aut], Alain Pietroniro [aut, fnd], Jianxun He [aut], Environment and Climate Change Canada (ECCC) [fnd]

Maintainer Riley Wheadon <rileywheadon@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-11-20 18:30:02 UTC

RemoteUrl https://github.com/cran/ffaframework

RemoteRef HEAD

RemoteSha b8515804c15be46b21ebaf517376eb51f894f5ff

Contents

ffaframework-package . . . 3
CAN_05BB001 . . . 5
CAN_07BE001 . . . 6
CAN_08MH016 . . . 7
CAN_08NH021 . . . 7
CAN_08NM050 . . . 8
data_decomposition . . . 9
data_geomet . . . 10
data_local . . . 11
data_mw_variability . . . 12
data_screening . . . 13
eda_bbmk_test . . . 14
eda_kpss_test . . . 15
eda_mk_test . . . 16
eda_mks_test . . . 17
eda_pettitt_test . . . 19
eda_pp_test . . . 20
eda_runs_test . . . 21
eda_sens_trend . . . 23
eda_spearman_test . . . 24
eda_white_test . . . 25
fit_gmle . . . 26
fit_lmoments . . . 27
fit_lmoments_kappa . . . 28
fit_mle . . . 30
framework_eda . . . 31
framework_ffa . . . 33
framework_full . . . 35
model_assessment . . . 37
plot_ams_data . . . 39
plot_bbmk_test . . . 40
plot_lmom_diagram . . . 41
plot_mks_test . . . 42
plot_nsffa_assessment . . . 43
plot_nsffa_estimates . . . 44
plot_nsffa_fit . . . 45
plot_pettitt_test . . . 46
plot_runs_test . . . 47

plot_sffa_assessment 48
plot_sffa_estimates 48
plot_sffa_fit 49
plot_spearman_test 50
select_ldistance 51
select_lkurtosis 52
select_zstatistic 53
uncertainty_bootstrap 55
uncertainty_rfgpl 57
uncertainty_rfpl 59
utils_cdf 61
utils_generalized_likelihood 62
utils_log_likelihood 64
utils_quantiles 65
utils_sample_lmoments 66
utils_theoretical_lmoments 67

Index **69**

`ffaframework-package` *Flood Frequency Analysis Framework*

Description

This package provides tools for stationary (S-FFA) and nonstationary (NS-FFA) flood flood frequency analysis of annual maximum series data. High-level wrapper functions with the `framework_*` prefix orchestrate the EDA and/or FFA modules from Vidrio-Sahagún et al. (2024) and generate reports. Users who wish to develop customized workflows may use methods with the following prefixes:

1. `eda_*`: Explore annual maximum series data for evidence of nonstationarity to inform approach selection (S-FFA or NS-FFA):
   - Detect statistically significant change points.
   - Detect statistically significant temporal trends in the *mean* and *variability*.
2. `select_*`: Select a suitable probability distribution using the L-moments.
3. `fit_*`: Fit parameters given a distribution and approach (S-FFA or NS-FFA).
4. `uncertainty_*`: Quantify uncertainty by computing confidence intervals.
5. `model_assessment()` evaluates model performance using a variety of metrics.

Additional utility functions for visualization and computation are also available:

- `data_*` methods load, transform, and decompose annual maximum series data.
- `plot_*` methods produce diagnostic and summary plots.
- `utils_*` methods implement distribution-specific computations.

Datasets from five hydrometric stations in Canada are provided as representative use cases (other datasets in `/inst/extdata` are for testing purposes only):

- Athabasca River at Athabasca (CAN-07BE001): An unregulated station with no statistical evidence of trends or change points (S-FFA recommended).
- Kootenai River at Porthill (CAN-08NH21): A regulated station with evidence of an abrupt change in mean in 1972 (piecewise NS-FFA recommended).
- Bow River at Banff (CAN-05BB001). An unregulated station with statistical evidence of a trend in the mean (NS-FFA recommended).
- Chilliwack River at Chilliwack Lake (CAN-08MH016): An unregulated station with statistical evidence of a linear trend in variability (NS-FFA recommended).
- Okanagan River at Penticton (CAN-08NM050): A regulated station with statistical evidence of a linear trend in both the mean and variability (NS-FFA recommended).

This package assumes familiarity with statistical techniques used in FFA, including parameter estimation (e.g., L-moments and maximum likelihood), dataset decomposition, and uncertainty quantification (parametric bootstrap and profile likelihood). For an explanation of these methods, see the FFA Framework wiki. For examples, see the vignettes on exploratory data analysis and flood frequency analysis.

Author(s)

Maintainer: Riley Wheadon <rileywheadon@gmail.com>

Authors:

- Riley Wheadon <rileywheadon@gmail.com>
- Cuauhtémoc Vidrio-Sahagún <ct.vidrio-sahagun@usask.ca>
- Alain Pietroniro <alain.pietroniro@ucalgary.ca> [funder]
- Jianxun He <jianhe@ucalgary.ca>

Other contributors:

- Environment and Climate Change Canada (ECCC) [funder]

See Also

Useful links:

- https://rileywheadon.github.io/ffa-framework/
- Report bugs at https://github.com/rileywheadon/ffa-framework/issues

`CAN_05BB001` *CAN-05BB001*

Description

A dataframe of annual maximum series observations for station 05BB001, BOW RIVER AT BANFF in Alberta, Canada.

Usage

```
CAN_05BB001
```

Format

A dataframe with 110 rows and 2 columns spanning the period 1909-2018.

Details

Variables:

- `max`: Numeric; the observed annual maximum series, in m^3/s.
- `year`: Integer; the corresponding years.

Additional Information

This is an unregulated station in the RHBN. Whitfield & Pomeroy (2016) found that floods may be caused by rain, snow, or a combination of both. Therefore, practitioners should be careful when interpreting the results of FFA. Minimal human intervention in the basin means there is little justification for change points. EDA finds evidence of a decreasing trend in the mean.

Source

Meteorological Service of Canada (MSC) GeoMet Platform

References

Whitfield P. H., and Pomeroy J. W. (2016) Changes to flood peaks of a mountain river: implications for analysis of the 2013 flood in the Upper Bow River, Canada, Hydrological Processes, 30: 4657–4673. doi:10.1002/hyp.10957.

CAN_07BE001 *CAN-07BE001*

Description

A dataframe of annual maximum series observations for station 07BE001, ATHABASCA RIVER AT ATHABASCA in Alberta, Canada.

Usage

```
CAN_07BE001
```

Format

A dataframe with 108 rows and 2 columns spanning the period 1913-2020.

Details

Variables:

- `max`: Numeric; the observed annual maximum series, in m^3/s.
- `year`: Integer; the corresponding years.

Additional Information

This is an unregulated station that is not in the RHBN. Additionally,

- The MKS/Pettitt tests find no evidence of change points at the 0.05 significance level.
- Trend detection finds no evidence of trends in the mean or variability.

This dataset is an excellent introductory example to stationary FFA.

Source

Meteorological Service of Canada (MSC) GeoMet Platform

CAN_08MH016 *CAN-08MH016*

Description

A dataframe of annual maximum series observations for station 08MH016, CHILLIWACK RIVER AT CHILLIWACK LAKE in British Columbia, Canada.

Usage

```
CAN_08MH016
```

Format

A dataframe with 95 rows and 2 columns spanning the period 1922-2016.

Details

Variables:

- `max`: Numeric; the observed annual maximum series, in m^3/s.
- `year`: Integer; the corresponding years.

Additional Information

This is an unregulated station in the RHBN. Additionally,

- The MKS/Pettitt tests find no evidence of change points at the 0.05 significance level.
- Trend detection finds evidence of an increasing trend in the variability.

Source

Meteorological Service of Canada (MSC) GeoMet Platform

CAN_08NH021 *CAN-08NH021*

Description

A dataframe of annual maximum series observations for station 08NH021, KOOTENAI RIVER AT PORTHILL in British Columbia, Canada.

Usage

```
CAN_08NH021
```

Format

A dataframe with 91 rows and 2 columns spanning the period 1928-2018.

Details

Variables:

- `max`: Numeric; the observed annual maximum series, in m^3/s.
- `year`: Integer; the corresponding years.

Additional Information

This is a regulated station that is not in the RHBN. Additionally,

- The Libby dam was constructed upstream of this station in 1972.
- The Pettitt test finds evidence of a change point in 1972 at the 0.05 significance level.
- The MKS test finds evidence of change points in 1960 & 1985 at the 0.05 significance level.

This dataset is an excellent introduction to change point detection and piecewise NS-FFA.

Source

Meteorological Service of Canada (MSC) GeoMet Platform

`CAN_08NM050` *CAN-08NM050*

Description

A dataframe of annual maximum series observations for station 08NM050, OKANAGAN RIVER AT PENTICTON in British Columbia, Canada.

Usage

```
CAN_08NM050
```

Format

A dataframe with 97 rows and 2 columns spanning the period 1921-2017.

Details

Variables:

- `max`: Numeric; the observed annual maximum series, in m^3/s.
- `year`: Integer; the corresponding years.

Additional Information

This is a regulated station that is not part of the RHBN. The Okanagan River upstream of the station has been regulated since 1914 due to the construction of the first dam, followed by a second dam in 1920, and a regulation system in the early 1950s, consisting of four dams and 38 km of engineered channel. Rapid human settlement, development, and agricultural activity have also occurred in the watershed.

This dataset exhibits serial correlation and trends in both the mean and variability. Since the station is heavily influenced by reservoir operations, this dataset is intended for teaching purposes—not for practical flood estimation.

Source

Meteorological Service of Canada (MSC) GeoMet Platform

`data_decomposition` *Decompose Annual Maximum Series*

Description

Decomposes a nonstationary annual maxima series to derive its stationary stochastic component, which can be used to identify a best-fit distribution using conventional stationary methods, like those based on L-moments. The decomposition procedure follows that proposed by Vidrio-Sahagún and He (2022), which relies on the statistical representation of nonstationary stochastic processes.

Usage

```
data_decomposition(data, ns_years, ns_structure)
```

Arguments

`data`	Numeric vector of observed annual maximum series values. Must be strictly positive, finite, and not missing.
`ns_years`	For NS-FFA only: Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.
`ns_structure`	For NS-FFA only: Named list indicating which distribution parameters are modeled as nonstationary. Must contain two logical scalars: • `location`: If `TRUE`, the location parameter has a linear temporal trend. • `scale`: If `TRUE`, the scale parameter has a linear temporal trend.

Details

Internally, the function does the following:

1. If there is a trend in the location, fit Sen's trend estimator and subtract away the fitted trend.
2. If there is a trend in the scale, estimate the variability of the data with `data_mw_variability()`, fit Sen's trend estimator to the variability vector, and rescale the data to remove the trend.
3. If necessary, shift the data so that its minimum is at least 1.

Value

Numeric vector of decomposed data.

References

Vidrio-Sahagún, C. T., and He, J. (2022). The decomposition-based nonstationary flood frequency analysis. Journal of Hydrology, 612 (September 2022), 128186. doi:10.1016/j.jhydrol.2022.128186

See Also

`data_mw_variability()`, `eda_sens_trend()`

Examples

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

---

data_geomet *Fetch Data from MSC GeoMet API*

Description

Gets annual maximum series data for a hydrological monitoring station from the MSC GeoMet API.

Usage

```
data_geomet(station_id)
```

Arguments

`station_id`	A character scalar containing the ID of a hydrological monitoring station. You can search for station IDs by name, province, drainage basin, and location here.

Value

A dataframe with two columns:

- `max`: A float, the annual maximum series observation, in m^3/s.
- `year`: An integer, the corresponding year.

See Also

`data_local()`

Examples

```
# Get data for the BOW RIVER AT BANFF (05BB001)
df <- data_geomet("05BB001")
```

`data_local` *Fetch Local Package Data*

Description

Fetch annual maximum series data for a hydrological monitoring station from the package data directory.

Usage

```
data_local(csv_file)
```

Arguments

`csv_file`	A character scalar containing the file name of a local dataset in `/inst/extdata`. Must be one of: • `"CAN-05BA001.csv"`: BOW RIVER AT LAKE LOUISE • `"CAN-05BB001.csv"`: BOW RIVER AT BANFF • `"CAN-07BE001.csv"`: ATHABASCA RIVER AT ATHABASCA • `"CAN-08MH016.csv"`: CHILLIWACK RIVER AT CHILLIWACK LAKE • `"CAN-08NH021.csv"`: KOOTENAI RIVER AT PORTHILL • `"CAN-08NM050.csv"`: OKANAGAN RIVER AT PENTICTON • `"CAN-08NM116.csv"`: MISSION CREEK NEAR EAST KELOWNA

Value

A dataframe with two columns:

- `max`: A float, the annual maximum series observation, in m^3/s.
- `year`: An integer, the corresponding year.

See Also

`data_geomet()`

Examples

```
# Get data for the BOW RIVER AT BANFF (05BB001)
df <- data_local("CAN-05BB001.csv")
```

`data_mw_variability` *Estimate Variance for Annual Maximum Series Data*

Description

Generates a time series of standard deviations using a moving window algorithm, which can be used to explore potential evidence of nonstationarity in the variability of a dataset. It returns a list that pairs each window's mean year with its window standard deviation. The hyperparameters `size` and `step` control the behaviour of the moving window. Following the simulation findings from Vidrio-Sahagún and He (2022), the default window size and step are set to 10 and 5 years respectively. However, these can be changed by the user.

Usage

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

Arguments

`data`	Numeric vector of observed annual maximum series 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).
`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.

References

Vidrio-Sahagún, C. T., and He, J. (2022). The decomposition-based nonstationary flood frequency analysis. Journal of Hydrology, 612 (September 2022), 128186. doi:10.1016/j.jhydrol.2022.128186

Examples

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

`data_screening` *Perform Data Screening*

Description

Checks for missing entries and generates a list of summary statistics about a dataset.

Usage

```
data_screening(data, years)
```

Arguments

`data`	Numeric vector of observed annual maximum series 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.

Value

A list with seven entries:

- `years_min`: The minimum value in the 'years' argument.
- `years_max`: The maximum value in the 'years' argument.
- `data_min`: The minimum value in the 'data' argument.
- `data_med`: The median value in the 'data' argument.
- `data_max`: The maximum value in the 'data' argument.
- `missing_years`: An integer vector of years with no data.
- `missing_count`: The number of missing entries in the dataset.

Examples

```
data <- rnorm(n = 10, mean = 100, sd = 10)
years <- c(1900, 1902, 1903, 1904, 1905, 1907, 1909, 1911, 1912, 1914)
data_screening(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 adjust for serial correlation in annual maximum series data. The BBMK test uses Spearman's serial correlation test to identify the least insignificant lag, then applies a shuffling procedure to obtain the empirical p-value and confidence bounds for the Mann-Kendall test statistic.

Usage

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

Arguments

`data`	Numeric vector of observed annual maximum series 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.

Details

The block size for reshuffling is equal to the `least_lag` as estimated in `eda_spearman_test()`. Bootstrap samples are generated by resampling blocks of the original data *without replacement*. This procedure effectively removes serial correlation from the data.

Value

A list containing the test results, including:

- `data`: The `data` argument.
- `alpha`: The significance level as specified in the `alpha` argument.
- `null_hypothesis`: A string describing the null hypothesis.
- `alternative_hypothesis`: A string describing the alternative hypothesis.
- `statistic`: The Mann-Kendall S-statistic computed on the original series.
- `bootstrap`: Vector of bootstrapped Mann-Kendall test statistics.
- `p_value`: Empirical two-sided p-value derived from the bootstrap distribution.
- `bounds`: Empirical confidence interval bounds from the bootstrap distribution.
- `reject`: If `TRUE`, the null hypothesis was rejected at significance `alpha`.

References

Bayazit, M., 2015. Nonstationarity of hydrological records and recent trends in trend analysis: a state-of-the-art review. Environmental Processes 2 (3), 527–542. doi:10.1007/s4071001500817

Khaliq, M.N., Ouarda, T.B.M.J., Gachon, P., Sushama, L., St-Hilaire, A., 2009. Identification of hydrological trends in the presence of serial and cross correlations: a review of selected methods and their application to annual flow regimes of Canadian rivers. Journal Hydrolology 368 (1–4), 117–130. doi:10.1016/j.jhydrol.2009.01.035

Sonali, P., Nagesh Kumar, D., 2013. Review of trend detection methods and their application to detect temperature changes in India. Journal Hydrology 476, 212–227. doi:10.1016/j.jhydrol.2012.10.034

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 data. The null hypothesis is that the time series is trend-stationary with a linear deterministic trend and constant drift. The alternative hypothesis is that the time series has a unit root (also known as a stochastic trend).

Usage

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

Arguments

`data`	Numeric vector of observed annual maximum series 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.

Details

The implementation of the KPSS test is based on the 'aTSA' package, which interpolates a significance table from Hobijn 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 $p \geq 0.10$.

Value

A list containing the test results, including:

- `data`: The `data` argument.
- `alpha`: The significance level as specified in the `alpha` argument.
- `null_hypothesis`: A string describing the null hypothesis.
- `alternative_hypothesis`: A string describing the alternative hypothesis.
- `statistic`: The KPSS test statistic.
- `p_value`: The interpolated p-value. See the details for more information.
- `reject`: If `TRUE`, the null hypothesis was rejected at significance `alpha`.

References

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

Kwiatkowski, D.; Phillips, P. C. B.; Schmidt, P.; Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root. Journal of Econometrics, 54 (1-3): 159-178.

See Also

`eda_pp_test()`

Examples

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

`eda_mk_test` *Mann–Kendall Trend Test*

Description

Performs the Mann–Kendall trend test on a numeric vector to detect the presence of an increasing or decreasing monotonic trend 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)
```

Arguments

`data`	Numeric vector of observed annual maximum series 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.

Details

The test statistic S is 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.
- `alpha`: The significance level as specified in the `alpha` argument.
- `null_hypothesis`: A string describing the null hypothesis.
- `alternative_hypothesis`: A string describing the alternative hypothesis.
- `statistic`: The Mann–Kendall test statistic.
- `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`.

References

Kendall, M. (1975). Rank Correlation Methods. Griffin, London, 202 pp.

Mann, H. B. (1945). Nonparametric Tests Against Trend. Econometrica, 13(3): 245-25

See Also

`eda_bbmk_test()`

Examples

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

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

Description

Performs the Mann–Kendall–Sneyers (MKS) test to detect a trend change in annual maximum series data. The test computes normalized progressive and regressive Mann–Kendall statistics and identifies statistically significant crossing points, indicating potential change points. Under the null hypothesis, there are no trend changes.

Usage

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

Arguments

data	Numeric vector of observed annual maximum series 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.

Details

This function computes progressive and regressive Mann–Kendall-Sneyers statistics, normalized by their expected values and variances under the null hypothesis. The crossing points occur when the difference between the progressive and regressive statistics switches sign. The significance of detected crossing points is assessed using the quantiles of the normal distribution.

Value

A list containing the test results, including:

- data: The data argument.
- years: The years argument.
- alpha: The significance level as specified in the alpha argument.
- null_hypothesis: A string describing the null hypothesis.
- alternative_hypothesis: A string describing the alternative hypothesis.
- progressive_series: Normalized progressive Mann–Kendall-Sneyers statistics.
- regressive_series: Normalized regressive Mann–Kendall-Sneyers statistics.
- bound: Critical confidence bound for significance based on alpha.
- change_points: A dataframe of potential change points.
- p_value: Two-sided p-value of the most significant crossing point.
- reject: If TRUE, the null hypothesis was rejected at significance alpha.

change_points contains the years, test statistics, and p-values of each potential change point. If no change points were identified, change_points is empty.

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(), eda_pettitt_test()

Examples

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

`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 abrupt change in the central tendency of a time series. Under the null hypothesis, there is no change.

Usage

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

Arguments

`data`	Numeric vector of observed annual maximum series 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.

Value

A list containing the test results, including:

- `data`: The `data` argument.
- `years`: The `years` argument.
- `alpha`: The significance level as specified in the `alpha` argument.
- `null_hypothesis`: A string describing the null hypothesis.
- `alternative_hypothesis`: A string describing the alternative hypothesis.
- `u_series`: Numeric vector of absolute U-statistics for all years.
- `statistic`: The test statistic and maximum absolute U-statistic.
- `bound`: The critical value of the test statistic based on `alpha`.
- `change_points`: A dataframe containing the potential change point.
- `p_value`: An asymptotic approximation of the p-value for the test.
- `reject`: Logical scalar. If `TRUE`, the null hypothesis was rejected.

`change_points` contains the years, test statistics, and p-values of each potential change point. If no change points were identified, `change_points` is empty.

References

Pettitt, A.N., 1979. A Non-parametric Approach to the Change-point Problem. J. Royal Statistics Society 28 (2), 126–135. http://www.jstor.org/stable/2346729

See Also

plot_pettitt_test(), eda_mks_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 data. The null hypothesis assumes the time series contains a unit root (also known as a stochastic trend). The alternative hypothesis is that the time series is trend-stationary with a deterministic linear trend.

Usage

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

Arguments

data	Numeric vector of observed annual maximum series 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.

Details

The implementation of this test is based on the 'aTSA' package, which interpolates p-values from the 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 that $p \leq 0.01$.

Value

A list containing the test results, including:

- `data`: The `data` argument.
- `alpha`: The significance level as specified in the `alpha` argument.
- `null_hypothesis`: A string describing the null hypothesis.
- `alternative_hypothesis`: A string describing the alternative hypothesis.
- `statistic`: The PP test statistic.
- `p_value`: Reported p-value from the test. See the details for more information.
- `reject`: If `TRUE`, the null hypothesis was rejected at significance `alpha`.

References

Fuller, W. A. (1976). Introduction to Statistical Time Series. New York: John Wiley and Sons

Phillips, P. C. B.; Perron, P. (1988). Testing for a Unit Root in Time Series Regression. Biometrika, 75 (2): 335-346

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 in order to assess whether it behaves as a random sequence. The test statistic and p-value is computed using the number of runs (sequences of values above or below the median). Under the null hypothesis, the data is random. The runs test can be used to assess whether the residuals of a nonstationary model are random, indicating the suitability of the assumed nonstationary structure (e.g., linear).

Usage

```
eda_runs_test(values, years, alpha = 0.05)
```

Arguments

values	A numeric vector of values to check for randomness.
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.

Value

A list containing the test results, including:

- values: The values argument.
- years: The years argument.
- alpha: The significance level as specified in the alpha argument.
- null_hypothesis: A string describing the null hypothesis.
- alternative_hypothesis: A string describing the alternative hypothesis.
- n: The length of the input vector after removing the median.
- runs: The number of runs in the transformed sequence of residuals.
- statistic: The runs test statistic, computed using runs and n.
- p_value: The p-value derived from the normally distributed test statistic.
- reject: If TRUE, the null hypothesis was rejected at significance alpha.

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_trend <- eda_sens_trend(data, years)
eda_runs_test(sens_trend$residuals, years)
```

`eda_sens_trend` *Sen's Trend Estimator*

Description

Computes Sen's linear trend estimator for a univariate time series. The estimated slope and y-intercept are given in terms of the data and the *covariate* (time), which is derived from the years using the formula $(\text{Years} - 1900)/100$.

Usage

```
eda_sens_trend(data, years)
```

Arguments

`data`	Numeric vector of observed annual maximum series 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.

Details

Sen's slope estimator is a robust, nonparametric trend estimator based on the median of all pairwise slopes between data points. The corresponding intercept is the median of each $y_i - mx_i$ where m is the estimated slope.

Value

A list containing the estimated trend:

- `data`: The `data` argument.
- `years`: The `years` argument.
- `slope`: The estimated slope.
- `intercept`: The estimated y-intercept.
- `residuals`: Vector of differences between the predicted and observed values.

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_ams_data()`

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 serial correlation test on annual maximum series data to check for serial correlation at various lags. Reports the smallest lag where the serial correlation is not statistically significant at the given significance level (the *least insignificant lag*).

Usage

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

Arguments

`data`	Numeric vector of observed annual maximum series 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.

Value

A list containing the test results, including:

- `data`: The `data` argument.
- `alpha`: The significance level as specified in the `alpha` argument.
- `null_hypothesis`: A string describing the null hypothesis.
- `alternative_hypothesis`: A string describing the alternative hypothesis.
- `rho`: Numeric vector of serial correlation estimates for lags 1 to $n - 3$.
- `least_lag`: The smallest lag at which the serial correlation is insignificant.
- `significant`: Indicates whether the serial correlation is significant at each lag.
- `reject`: If `TRUE`, the null hypothesis was rejected at significance `alpha`.

See Also

`stats::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)
```

Arguments

`data`	Numeric vector of observed annual maximum series 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.

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 and n is the number of elements in the time series. Under the null hypothesis, the test statistic has the χ^2 distribution with 2 degrees of freedom.

Value

A list containing the results of the White test:

- `data`: The `data` argument.
- `years`: The `years` argument.
- `alpha`: The significance level as specified in the `alpha` argument.
- `null_hypothesis`: A string describing the null hypothesis.
- `alternative_hypothesis`: A string describing the alternative hypothesis.
- `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`: If `TRUE`, the null hypothesis was rejected at significance `alpha`.

References

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

See Also

stats::lm(), stats::pchisq()

Examples

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

`fit_gmle` *Generalized Maximum Likelihood Parameter Estimation*

Description

Estimates the parameters of the generalized extreme value (GEV) distribution by maximizing the generalized log-likelihood, which incorporates a Beta prior on the shape parameter. Initial parameter estimates are obtained using the method of L-moments and optimization is performed via stats::nlminb() with repeated perturbations if needed.

For NS-FFA: To estimate parameters for a nonstationary model, include the observation years (`ns_years`) and the nonstationary model structure (`ns_structure`).

Usage

```
fit_gmle(data, prior, ns_years = NULL, ns_structure = NULL)
```

Arguments

`data`	Numeric vector of observed annual maximum series values. Must be strictly positive, finite, and not missing.
`prior`	Numeric vector of length 2. Specifies the parameters of the Beta prior for the shape parameter κ.
`ns_years`	For NS-FFA only: Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.
`ns_structure`	For NS-FFA only: Named list indicating which distribution parameters are modeled as nonstationary. Must contain two logical scalars: • `location`: If `TRUE`, the location parameter has a linear temporal trend. • `scale`: If `TRUE`, the scale parameter has a linear temporal trend.

Details

1. Calls fit_lmoments() on the data to obtain initial parameter estimates.
2. Initializes trend parameters to zero if necessary.
3. Defines an objective function using utils_generalized_likelihood().
4. Runs stats::nlminb() with box constraints. Attempts minimization up to 100 times.

Value

A list containing the results of parameter estimation:

- `data`: The `data` argument.
- `prior`: The `prior` argument.
- `ns_years`: The `ns_years` argument, if given.
- `ns_structure`: The `ns_structure` argument, if given.
- `method`: `"GMLE"`.
- `params`: Numeric vector of estimated parameters.
- `mll`: The maximum value of the generalized log-likelihood.

References

El Adlouni, S., Ouarda, T.B.M.J., Zhang, X., Roy, R., Bobee, B., 2007. Generalized maximum likelihood estimators for the nonstationary generalized extreme value model. Water Resources Research 43 (3), 1–13. doi:10.1029/2005WR004545

Martins, E. S., and Stedinger, J. R. (2000). Generalized maximum-likelihood generalized extreme-value quantile estimators for hydrologic data. Water Resources Research, 36(3), 737–744. doi:10.1029/1999WR900330

See Also

`utils_generalized_likelihood()`, `fit_lmoments()`, `stats::nlminb()`

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
prior <- c(6, 9)
ns_years <- seq(from = 1901, to = 2000)
ns_structure <- list(location = TRUE, scale = FALSE)
fit_gmle(data, prior, ns_years, ns_structure)
```

`fit_lmoments` *L-Moments Parameter Estimation*

Description

For S-FFA only: Estimates the parameters of a *stationary* probability model using the L-moments.

Usage

```
fit_lmoments(data, distribution)
```

Arguments

`data`	Numeric vector of observed annual maximum series values. Must be strictly positive, finite, and not missing.
`distribution`	A three-character code indicating the distribution family. Must be `"GUM"`, `"NOR"`, `"LNO"`, `"GEV"`, `"GLO"`, `"GNO"`, `"PE3"`, `"LP3"`, or `"WEI"`.

Details

First, the sample L-moments of the data are computed using `utils_sample_lmoments()`. Then, formulas from Hosking (1997) are used to match the parameters to the sample L-moments. The distributions `"GNO"`, `"PE3"`, and `"LP3"` use a rational approximation of the parameters since no closed-form expression is known.

Value

A list containing the results of parameter estimation:

- `data`: The `data` argument.
- `distribution`: The `distribution` argument.
- `method`: `"L-moments"`.
- `params`: Numeric vector of estimated 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.

See Also

`fit_lmoments_kappa()`, `fit_mle()`, `fit_gmle()`

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
fit_lmoments(data, "GUM")
```

`fit_lmoments_kappa` *L-Moments Parameter Estimation for the Kappa Distribution*

Description

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

Usage

```
fit_lmoments_kappa(data)
```

Arguments

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

Details

First, the sample L-moments of the data are computed using utils_sample_lmoments(). Then, the stats::optim() function is used to determine the parameters by minimizing the euclidian distance between the sample and theoretical L-moment ratios. The implementation of this routine is based on the deprecated 'homtest' package, formerly available at https://CRAN.R-project.org/package=homtest.

Value

A list containing the results of parameter estimation:

- data: The data argument.
- distribution: "KAP".
- method: "L-moments".
- params: numeric vector of 4 parameters in the order location, scale, 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

utils_sample_lmoments(), fit_lmoments()

Examples

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

`fit_mle` *Maximum Likelihood Parameter Estimation*

Description

Estimates the parameters of a probability distribution by maximizing the log-likelihood. Initial parameter estimates are obtained using the method of L-moments and optimization is performed via `stats::nlminb()` with repeated perturbations if needed.

For NS-FFA: To estimate parameters for a nonstationary model, include the observation years (`ns_years`) and the nonstationary model structure (`ns_structure`).

Usage

```
fit_mle(data, distribution, ns_years = NULL, ns_structure = NULL)
```

Arguments

`data`	Numeric vector of observed annual maximum series values. Must be strictly positive, finite, and not missing.
`distribution`	A three-character code indicating the distribution family. Must be `"GUM"`, `"NOR"`, `"LNO"`, `"GEV"`, `"GLO"`, `"GNO"`, `"PE3"`, `"LP3"`, or `"WEI"`.
`ns_years`	For NS-FFA only: Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.
`ns_structure`	For NS-FFA only: Named list indicating which distribution parameters are modeled as nonstationary. Must contain two logical scalars: • `location`: If `TRUE`, the location parameter has a linear temporal trend. • `scale`: If `TRUE`, the scale parameter has a linear temporal trend.

Details

1. Calls `fit_lmoments()` 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 `utils_log_likelihood()`.
5. Runs `stats::nlminb()` with box constraints. Attempts minimization up to 100 times.

Value

A list containing the results of parameter estimation:

- `data`: The `data` argument.
- `distribution`: The `distribution` argument.
- `ns_years`: The `ns_years` argument, if given.
- `ns_structure`: The `ns_structure` argument, if given.

- method: "MLE".
- params: Numeric vector of estimated parameters.
- mll: The maximum value of the log-likelihood.

See Also

utils_log_likelihood(), fit_lmoments(), stats::nlminb()

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
ns_years <- seq(from = 1901, to = 2000)
ns_structure <- list(location = TRUE, scale = FALSE)
fit_mle(data, "GNO", ns_years, ns_structure)
```

---

framework_eda *Orchestrate Exploratory Data Analysis*

Description

First, this method identifies change points in the original annual maximum series data. Then, the user is given the option to split the dataset into two or more homogenous subperiods (trend-free or with monotonic trends). Finally, this method performs a collection of statistical tests for identifying monotonic nonstationarity in the mean and variability of each subperiod (if the dataset was split) or of the entire dataset (if it was not split). The results of EDA can help guide FFA approach selection (stationary or nonstationary) and FFA model determination.

Usage

```
framework_eda(
  data,
  years,
  ns_splits = NULL,
  generate_report = TRUE,
  report_path = NULL,
  report_formats = "html",
  ...
)
```

Arguments

data	Numeric vector of observed annual maximum series 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.

`ns_splits` An integer vector of years used to split the data into homogeneous subperiods. For S-FFA, set to `NULL` (default). For NS-FFA, specify an *integer* vector of years with physical justification for change points, or `NULL` if no such years exist. In R, integers have the suffix `L`, so `1950L` is a valid input to `ns_splits`, but `1950` is not (since it may be interpreted as a floating point number).

`generate_report` If `TRUE` (default), generate a report.

`report_path` A character scalar, the file path for the generated report. If `NULL` (default), the report will be saved to a new temporary directory.

`report_formats` A character vector specifying the output format for the report. Supported values are `"md"`, `"pdf"`, `"html"`, and `"json"`.

`...` Additional arguments. See the "Optional Arguments" section for a complete list.

Value

`eda_recommendations`: A list containing the recommended FFA approach, split point(s) and non-stationary structure(s) from EDA:

- `approach`: Either "S-FFA", "NS-FFA" (for a single homogeneous period), or "Piecewise NS-FFA" (for multiple homogeneous subperiods).
- `ns_splits`: The split point(s) identified by the change point detection test with the the lowest statistically significant p-value, or `NULL` if no such point exists.
- `ns_structures`: A list of structure objects for each homogeneous subperiod. Each structure is a list with boolean items `location` and `scale`, which represent a linear trend in the in the mean or variability of the data, respectively. If no trends were found in any homogeneous subperiod, `ns_structures` will be `NULL`.

`submodule_results`: A list of lists of statistical tests. Each list contains:

- `name`: Either "Change Point Detection" or "Trend Detection".
- `start`: The first year of the homogeneous subperiod.
- `end`: The last year of the homogeneous subperiod.
- Additional items from the statistical tests within the submodule.

Optional Arguments

- `alpha`: The numeric significance level for all statistical tests (default is 0.05).
- `bbmk_samples`: The number of samples used in the Block-Bootstrap Mann-Kendall (BBMK) test (default is 10000). Must be an integer.
- `window_size`: The size of the window used to compute the variability series.
- `window_step`: The number of years between successive moving windows. Used to compute the variability series.

See Also

`eda_pettitt_test()`, `eda_mks_test()`, `eda_mk_test()`, `eda_spearman_test()`, `eda_bbmk_test()`, `eda_pp_test()`, `eda_kpss_test()`, `eda_sens_trend()`, `eda_runs_test()`, `eda_white_test()`

Examples

```
# Get data for the BOW RIVER AT BANFF (05BB001)
df <- data_local("CAN-05BB001.csv")

# Run EDA (takes several minutes)
## Not run: framework_eda(df$max, df$year)
```

---

framework_ffa *Orchestrate Flood Frequency Analysis*

Description

Performs frequency analysis of annual maximum series data including distribution selection, parameter estimation, uncertainty quantification, and model assessment. Supports both stationary (S-FFA) or nonstationary (NS-FFA) flood frequency analysis.

Usage

```
framework_ffa(
  data,
  years,
  ns_splits = NULL,
  ns_structures = NULL,
  generate_report = TRUE,
  report_path = NULL,
  report_formats = "html",
  ...
)
```

Arguments

data	Numeric vector of observed annual maximum series 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.
ns_splits	An integer vector of years used to split the data into homogeneous subperiods. For S-FFA, set to NULL (default). For NS-FFA, specify an *integer* vector of years with physical justification for change points, or NULL if no such years exist. In R, integers have the suffix L, so 1950L is a valid input to ns_splits, but 1950 is not (since it may be interpreted as a floating point number).
ns_structures	For S-FFA, set to NULL (default) to use a stationary model for all homogeneous subperiods. For NS-FFA, provide a list of length(ns_splits) + 1 sublists specifying the nonstationary model structure for each homogeneous subperiod. Each sublist must contain logical elements location and scale, indicating monotonic trends in the mean and variability, respectively.

generate_report
: If TRUE (default), generate a report.

report_path
: A character scalar, the file path for the generated report. If NULL (default), the report will be saved to a new temporary directory.

report_formats
: A character vector specifying the output format for the report. Supported values are "md", "pdf", "html", and "json".

...
: Additional arguments. See the "Optional Arguments" section for a complete list.

Value

modelling_assumptions: A list describing the model(s) used for the analysis.

- approach: Either "S-FFA", "NS-FFA", or "Piecewise NS-FFA".
- ns_splits: The ns_splits argument, if given.
- ns_structures: The ns_structures argument, if given.

submodule_results: A list of lists of containing the results of frequency analysis. Each list contains:

- name: Either "Distribution Selection", "Parameter Estimation", "Uncertainty Quantification", or "Model Assessment".
- start: The first year of the homogeneous subperiod.
- end: The last year of the homogeneous subperiod.
- Additional items specific to the the submodule.

Optional Arguments

- selection: Distribution selection method (default is "L-distance"). Must be one of "L-distance", "L-kurtosis" or "Z-statistic". Alternatively, set selection to a three-letter distribution code (e.g., "GUM") to use a specific distribution.
- s_estimation: Parameter estimation method for S-FFA (default is "L-moments"). Must be "L-moments", "MLE", or "GMLE". Method "GMLE" requires selection = "GEV".
- ns_estimation: Parameter estimation method for NS-FFA (default is "MLE"). Must be "MLE" or "GMLE". Method "GMLE" requires selection = "GEV".
- s_uncertainty: Uncertainty quantification method for S-FFA (default is "Bootstrap"). Must be one of "Bootstrap", "RFPL", or RFGPL". Using method "RFPL" requires s_estimation = "MLE" and method "RFGPL" requires s_estimation = "GMLE".
- ns_uncertainty: Uncertainty quantification method for NS-FFA (default is "RFPL"). Must be one of "Bootstrap", "RFPL", or RFGPL". Using method "RFPL" requires ns_estimation = "MLE" and method "RFGPL" requires ns_estimation = "GMLE".
- z_samples: Integer number of bootstrap samples for selection method "Z-statistic" (default is 10000).
- gev_prior: Parameters for the prior distribution of the shape parameter of the GEV distribution (default is 6, 9). Used with estimation method "GMLE".
- return_periods: Integer list of return periods (in years) for estimating return levels. Uses the 2, 5, 10, 20, 50, and 100 year return periods by default.

- ns_slices: Integer vector of years at which to estimate the return levels for nonstationary models. Slices outside of the period will be ignored (default is 1925, 1975, 2025).
- bootstrap_samples: Integer number of samples for uncertainty quantification method '"Bootstrap" (default is 10000).
- rfpl_tolerance: Log-likelihood tolerance for uncertainty quantification method "RFPL"(default is 0.01).
- pp_formula: Plotting position formula for model assessment. Must be one of:
  - "Weibull" (default): $i/(n+1)$
  - "Blom": $(i-0.375)/(n+0.25)$
  - "Cunnane": $(i-0.4)/(n+0.2)$
  - "Gringorten": $(i-0.44)/(n+0.12)$
  - "Hazen": $(i-0.5)/n$

See Also

select_ldistance(), select_lkurtosis(), select_zstatistic(), fit_lmoments(), fit_mle(), fit_gmle(), uncertainty_bootstrap(), uncertainty_rfpl(), uncertainty_rfgpl(), model_assessment()

Examples

```
# Get data for the BOW RIVER AT BANFF (05BB001)
df <- data_local("CAN-05BB001.csv")

# Run the FFA module (takes several minutes)
## Not run: framework_ffa(df$max, df$year)
```

---

framework_full *Orchestrate the Full FFA Framework*

Description

Runs the entire flood frequency analysis framework using the results of exploratory data analysis (EDA) to guide approach selection (stationary or nonstationary) and perform flood frequency analysis. Returns a comprehensive and reproducible summary of the results.

Usage

```
framework_full(
  data,
  years,
  ns_splits = NULL,
  ns_structures = NULL,
  generate_report = TRUE,
  report_path = NULL,
  report_formats = "html",
  ...
)
```

Arguments

data	Numeric vector of observed annual maximum series 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.
ns_splits	An integer vector of years used to split the data into homogeneous subperiods. For S-FFA, set to NULL (default). For NS-FFA, specify an *integer* vector of years with physical justification for change points, or NULL if no such years exist. In R, integers have the suffix L, so 1950L is a valid input to ns_splits, but 1950 is not (since it may be interpreted as a floating point number).
ns_structures	For S-FFA, set to NULL (default) to use a stationary model for all homogeneous subperiods. For NS-FFA, provide a list of length(ns_splits) + 1 sublists specifying the nonstationary model structure for each homogeneous subperiod. Each sublist must contain logical elements location and scale, indicating monotonic trends in the mean and variability, respectively.
generate_report	If TRUE (default), generate a report.
report_path	A character scalar, the file path for the generated report. If NULL (default), the report will be saved to a new temporary directory.
report_formats	A character vector specifying the output format for the report. Supported values are "md", "pdf", "html", and "json".
...	Additional arguments to be passed to the statistical tests and frequency analysis functions. See the details of framework_eda() and framework_ffa() for a complete list.

Value

eda_recommendations: See framework_eda().

modelling_assumptions: See framework_ffa().

submodule_results: A list of lists of results. Each list contains:

- name: Either "Change Point Detection", "Trend Detection", "Distribution Selection", "Parameter Estimation", "Uncertainty Quantification", or "Model Assessment".
- start: The first year of the homogeneous subperiod.
- end: The last year of the homogeneous subperiod.
- Additional items specific to the the submodule.

See Also

framework_eda(), framework_ffa()

Examples

```
# Get data for the BOW RIVER AT BANFF (05BB001)
df <- data_local("CAN-05BB001.csv")

# Run the complete FFA framework (takes several minutes)
## Not run: framework_full(df$max, df$year)
```

`model_assessment` *Model Assessment*

Description

Computes various metrics for assessing the quality of a fitted flood frequency model. Metrics include accuracy (residual statistics), fitting efficiency (information criteria), and uncertainty (coverage based metrics using confidence intervals).

For NS-FFA: The metrics are computed from the normalized empirical/theoretical quantiles, which are determined based on the selected distribution family. Therefore, metrics from stationary and nonstationary models should not be compared.

Usage

```
model_assessment(
  data,
  distribution,
  method,
  prior = NULL,
  ns_years = NULL,
  ns_structure = NULL,
  alpha = 0.05,
  pp_formula = "Weibull",
  ci = NULL
)
```

Arguments

`data`	Numeric vector of observed annual maximum series values. Must be strictly positive, finite, and not missing.
`distribution`	A three-character code indicating the distribution family. Must be `"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 parameters of the Beta prior for the shape parameter κ.
`ns_years`	For NS-FFA only: Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.

`ns_structure` For NS-FFA only: Named list indicating which distribution parameters are modeled as nonstationary. Must contain two logical scalars:

- `location`: If `TRUE`, the location parameter has a linear temporal trend.
- `scale`: If `TRUE`, the scale parameter has a linear temporal 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 string specifying the plotting position formula. Must `"Weibull"` (default), `"Blom"`, `"Cunnane"`, `"Gringorten"`, or `"Hazen"`.

`ci` **For S-FFA only**. Dataframe containing return periods (in the column `periods`) and confidence intervals (in the columns `lower` and `upper`). Dataframes in this format can be generated with `uncertainty_bootstrap()`, `uncertainty_rfpl()`, or `uncertainty_rfgpl()`.

Details

These metrics are are computed for all models:

- `AIC_MLL`: Akaike Information Criterion, computed using the maximum log-likelihood.
- `BIC_MLL`: Bayesian Information Criterion, computed using the maximum log-likelihood.
- `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_RMSE`: Akaike Information Criterion, computed using the RMSE.
- `BIC_RMSE`: Bayesian Information Criterion, computed using the RMSE.

These metrics are only computed *if the* `ci` *argument is provided*:

- `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`.

Value

List containing the results of model assessment:

- `data`: The `data` argument.
- `q_theoretical`: The theoretical return level estimates based on the plotting positions. Normalized for nonstationary models.
- `q_empirical`: The empirical return levels. Normalized for nonstationary models.
- `metrics`: A list of model assessment metrics (see details).

See Also

`uncertainty_bootstrap()`, `uncertainty_rfpl()`, `uncertainty_rfgpl()`, `plot_sffa_assessment()`, `plot_nsffa_assessment()`

Examples

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

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

# Run model assessment
model_assessment(data, "NOR", "L-moments", ci = ci)
```

`plot_ams_data` *Plot Annual Maximum Series Data*

Description

Produces a scatterplot of annual maximum series data against time, optionally overlaid with the sample mean/variability or Sen's trend estimator of the mean/variability.

Usage

```
plot_ams_data(
  data,
  years,
  plot_mean = "None",
  plot_variability = "None",
  show_line = TRUE,
  ...
)
```

Arguments

`data`	Numeric vector of observed annual maximum series 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.
`plot_mean`	If "None" (default), the mean will not be plotted. If `"Constant"`, a black line is plotted at the sample mean. If `"Trend"`, the trend in the mean is estimated using `eda_sens_trend()` and plotted as a blue line.
`plot_variability`	If "None" (default), the variability will not be plotted. If `"Constant"`, dashed black lines are plotted at one standard deviation above/below the sample mean. If `"Trend"`, the trend in variability is estimated with `data_mw_variability()` and `eda_sens_trend()` and plotted as a dashed blue line.
`show_line`	If `TRUE` (default), a fitted line is drawn through the data.
`...`	Optional named arguments: 'title', 'xlabel', and 'ylabel'.

Value

ggplot; a plot containing:

- Gray points for each year's annual maximum series value.
- A gray line connecting the data if show_line = TRUE.
- A solid black line representing a constant mean, if plot_mean == "Constant".
- A solid blue line representing a trend in the mean, if plot_mean == "Trend".
- A dashed black line representing constant variability, if plot_variability == "Constant".
- A dashed blue line representing a trend in variability, if plot_variability == "Trend".

See Also

eda_sens_trend(), data_mw_variability()

Examples

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

---

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. Also includes a small inset around the L-moment ratios of the recommended distribution.

Usage

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

Arguments

results	List of distribution 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 data with the change points highlighted.

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 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_nsffa_assessment` *Plot Model Assessment for NS-FFA*

Description

Creates a normalized and detrended quantile–quantile plot (a worm plot) comparing empirical annual maximum series data to quantile estimates from a fitted, parametric, and nonstationary model. Confidence intervals are also provided. The worm plot can be interpreted using the following heuristics:

- For a satisfactory fit, the worm (red data points) should be within the 95% confidence interval (dashed black lines).
- If the worm (red points) is passes above/below the origin, the model mean is too small/large respectively.
- If the worm has a clear positive/negative slope, the model variance is too small/large respectively.
- If the worm has a U-shape or inverted U-shape, the model is too skew to the left/right respectively.

Usage

```
plot_nsffa_assessment(results, ...)
```

Arguments

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

Value

`ggplot`; a plot containing:

- A black line representing a model with no deviation from the empirical quantiles.
- Red points denoting the estimated quantiles against the empirical quantiles.
- A dashed black line showing the 95% confidence intervals of the residuals.

References

van Buuren, S., Fredriks, M. Worm plot: a simple diagnostic device for modelling growth reference curves. Statistics in Medicine 20 (8), 1259-1277. doi:10.1002/sim.746

Examples

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

# Evaluate model diagnostics
results <- model_assessment(data, "NOR", "MLE", NULL, years, structure)

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

---

plot_nsffa_estimates *Plot Estimated Return Levels for NS-FFA*

Description

Generates a plot with effective return periods on the x-axis and effective return levels (annual maxima magnitudes) on the y-axis. 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 have a logarithmic scale.

Usage

```
plot_nsffa_estimates(
  results,
  slices = c(1900, 1940, 1980, 2020),
  periods = c(2, 5, 10, 20, 50, 100),
  ...
)
```

Arguments

results	A fitted flood frequency model generated by fit_lmoments(), fit_mle() or fit_gmle() OR a fitted model with confidence intervals generated by uncertainty_bootstrap(), uncertainty_rfpl(), or uncertainty_rfgpl().
slices	Default time slices for plotting the return levels if confidence intervals are not provided.
periods	Numeric vector used to set the return periods for FFA. All entries must be greater than or equal to 1.
...	Optional named arguments: 'title', 'xlabel', and 'ylabel'.

Value

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

Examples

```
# Fit a nonstationary model
data <- rnorm(n = 100, mean = 100, sd = 10)
years <- seq(from = 1901, to = 2000)
ns_structure <- list(location = TRUE, scale = FALSE)

results <- fit_mle(
    data,
    "GEV",
    ns_years = years,
    ns_structure = ns_structure
)

# Generate the plot
plot_nsffa_estimates(results)
```

`plot_nsffa_fit` *Plot Fitted Probability Distributions for NS-FFA*

Description

Generates a plot showing probability densities of a nonstationary model for selected time slices (left panel) and the data (right panel).

Usage

```
plot_nsffa_fit(
  results,
  slices = c(1925, 1950, 1975, 2000),
  show_line = TRUE,
  ...
)
```

Arguments

`results`	A fitted flood frequency model generated by `fit_lmoments()`, `fit_mle()` or `fit_gmle()`.
`slices`	Years at which to plot the nonstationary probability model.
`show_line`	If `TRUE` (default), draw a fitted line through the data.
`...`	Optional named arguments: 'title', 'xlabel', and 'ylabel'.

Value

`ggplot`; a plot showing:

- The likelihood function of the distribution plotted vertically on the left panel.
- The data, connected with a line if `show_line == TRUE`, on the right panel.

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10) + seq(1, 100)
years <- seq(from = 1901, to = 2000)
ns_structure <- list(location = TRUE, scale = FALSE)

results <- fit_mle(
    data,
    "GEV",
    ns_years = years,
    ns_structure = ns_structure
)

plot_nsffa_fit(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 data with an optional trend line, the period mean(s), and potential change point(s).

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 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 data (or the variability of the data) with a horizontal dashed line at zero and an annotation indicating the p-value 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 p-value: 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 the runs test plot
sens_trend <- eda_sens_trend(data, years)
results <- eda_runs_test(sens_trend$residuals, years)
plot_runs_test(results)
```

plot_sffa_assessment *Plot Model Assessment for S-FFA*

Description

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

Usage

```
plot_sffa_assessment(results, ...)
```

Arguments

results	List; model assessment results generated by model_assessment().
...	Optional named arguments: 'title', 'xlabel', and 'ylabel'.

Value

ggplot; a plot containing:

- A black line representing a model with no deviation from the empirical quantiles.
- Red points denoting the estimated quantiles against the empirical quantiles.

Examples

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

# Evaluate model diagnostics
results <- model_assessment(data, "NOR", "L-moments")

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

plot_sffa_estimates *Plot Estimated Return Levels for S-FFA*

Description

Generates a plot with return periods on the x-axis and return levels (annual maxima magnitudes) on the y-axis for S-FFA. The confidence bound is shown as a semi-transparent ribbon, and the point estimates are overlaid as a solid line. Return periods are shown on a logarithmic scale.

Usage

```
plot_sffa_estimates(results, periods = c(2, 5, 10, 20, 50, 100), ...)
```

Arguments

results	A fitted flood frequency model generated by fit_lmoments(), fit_mle() or fit_gmle() OR a fitted model with confidence intervals generated by uncertainty_bootstrap(), uncertainty_rfpl(), or uncertainty_rfgpl().
periods	Numeric vector used to set the return periods for FFA. All entries must be greater than or equal to 1.
...	Optional named arguments: 'title', 'xlabel', and 'ylabel'.

Value

ggplot; a plot showing:

- A solid black line for the point estimates produced by the model.
- A semi-transparent gray ribbon indicating the confidence interval, if given.

Examples

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

---

plot_sffa_fit *Plot Fitted Probability Distribution for S-FFA*

Description

Generates a plot showing the probability density of a stationary model (left panel) and the data (right panel).

Usage

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

Arguments

results	A fitted flood frequency model generated by fit_lmoments(), fit_mle() or fit_gmle().
show_line	If TRUE (default), draw a fitted line through the data.
...	Optional named arguments: 'title', 'xlabel', and 'ylabel'.

Value

ggplot; a plot showing:

- The likelihood function of the distribution plotted vertically on the left panel.
- The data, connected with a line if show_line == TRUE, on the right panel.

Examples

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

---

plot_spearman_test *Plot Spearman's Rho Autocorrelation*

Description

Visualizes Spearman's rho serial correlation 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)
```

`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).

For NS-FFA: To select a distribution for a nonstationary model, include the observation years (`ns_years`) and the nonstationary model structure (`ns_structure`). Then, this method will detrend the original, nonstationary data internally using the `data_decomposition()` function prior to distribution selection.

Usage

```
select_ldistance(data, ns_years = NULL, ns_structure = NULL)
```

Arguments

`data`	Numeric vector of observed annual maximum series values. Must be strictly positive, finite, and not missing.
`ns_years`	For NS-FFA only: Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.
`ns_structure`	For NS-FFA only: Named list indicating which distribution parameters are modeled as nonstationary. Must contain two logical scalars: • `location`: If `TRUE`, the location parameter has a linear temporal trend. • `scale`: If `TRUE`, the scale parameter has a linear temporal trend.

Details

For each candidate distribution, this 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"`.
- `decomposed_data`: The detrended dataset used to compute the L-moments. For S-FFA, this is the `data` argument. For NS-FFA, it is output of `data_decomposition()`.
- `metrics`: A list of L-distance metrics for each candidate distribution.
- `recommendation`: The name of the distribution with the smallest L-distance.

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

utils_sample_lmoments(), 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.

For NS-FFA: To select a distribution for a nonstationary model, include the observation years (`ns_years`) and the nonstationary model structure (`ns_structure`). Then, this method will detrend the original, nonstationary data internally using the `data_decomposition()` function prior to distribution selection.

Usage

```
select_lkurtosis(data, ns_years = NULL, ns_structure = NULL)
```

Arguments

`data`	Numeric vector of observed annual maximum series values. Must be strictly positive, finite, and not missing.
`ns_years`	For NS-FFA only: Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.
`ns_structure`	For NS-FFA only: Named list indicating which distribution parameters are modeled as nonstationary. Must contain two logical scalars: • `location`: If `TRUE`, the location parameter has a linear temporal trend. • `scale`: If `TRUE`, the scale parameter has a linear temporal trend.

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 determined using `stats::optim()`.

Value

A list with the results of distribution selection:

- `method`: `"L-kurtosis"`.
- `decomposed_data`: The detrended dataset used to compute the L-moments. For S-FFA, this is the `data` argument. For NS-FFA, it is output of `data_decomposition()`.
- `metrics`: A list of L-kurtosis metrics for each distribution.
- `recommendation`: Name of the distribution with the smallest L-kurtosis metric.

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

`utils_sample_lmoments()`, `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 comparing 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.

For NS-FFA: To select a distribution for a nonstationary model, include the observation years (`ns_years`) and the nonstationary model structure (`ns_structure`). Then, this method will detrend the original, nonstationary data internally using the `data_decomposition()` function prior to distribution selection.

Usage

```
select_zstatistic(data, ns_years = NULL, ns_structure = NULL, samples = 10000L)
```

Arguments

`data`	Numeric vector of observed annual maximum series values. Must be strictly positive, finite, and not missing.
`ns_years`	For NS-FFA only: Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.
`ns_structure`	For NS-FFA only: Named list indicating which distribution parameters are modeled as nonstationary. Must contain two logical scalars: • `location`: If `TRUE`, the location parameter has a linear temporal trend. • `scale`: If `TRUE`, the scale parameter has a linear temporal trend.
`samples`	Integer scalar. The number of bootstrap samples. Default is 10000.

Details

First, this method fits a four-parameter Kappa distribution to both the original and log-transformed data. Then, bootstrapping is used to estimate the bias and variance of the L-kurtosis. These values, along with the difference between the sample and theoretical L-kurtosis, are used to compute the Z-statistic for each distribution.

Value

A list with the results of distribution selection:

- `method`: `"Z-selection"`.
- `decomposed_data`: The detrended dataset used to compute the L-moments. For S-FFA, this is the `data` argument. For NS-FFA, it is output of `data_decomposition()`.
- `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 data.
- `reg_bias_t4`: Bias of the L-kurtosis from the bootstrap.
- `reg_std_t4`: Standard deviation of the L-kurtosis from the bootstrap.
- `log_params`: Kappa distribution parameters for the log-transformed data.
- `log_bias_t4`: Bias of the L-kurtosis from the bootstrap using `log_params`.
- `log_std_t4`: Standard deviation of the L-kurtosis from the bootstrap using `log_params`.

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

`select_ldistance()`, `select_lkurtosis()`, `fit_lmoments_kappa()`, `utils_quantiles()`, `plot_lmom_diagram()`

Examples

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

---

uncertainty_bootstrap *Parametric Bootstrap Uncertainty Quantification*

Description

Computes return level estimates and confidence intervals at the specified return periods (defaults to 2, 5, 10, 20, 50, and 100 years) using the parametric bootstrap. This function supports many probability models and parameter estimation methods.

For NS-FFA: To perform uncertainty quantification for a nonstationary model, include the observation years (`ns_years`), the nonstationary model structure (`ns_structure`), and a list of years at which to compute the return level estimates and confidence intervals (`ns_slices`).

Usage

```
uncertainty_bootstrap(
  data,
  distribution,
  method,
  prior = NULL,
  ns_years = NULL,
  ns_structure = NULL,
  ns_slices = NULL,
  alpha = 0.05,
  samples = 10000L,
  periods = c(2, 5, 10, 20, 50, 100)
)
```

Arguments

`data`	Numeric vector of observed annual maximum series values. Must be strictly positive, finite, and not missing.
`distribution`	A three-character code indicating the distribution family. Must be `"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 parameters of the Beta prior for the shape parameter κ.
`ns_years`	For NS-FFA only: Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.

`ns_structure`	For NS-FFA only: Named list indicating which distribution parameters are modeled as nonstationary. Must contain two logical scalars: • `location`: If `TRUE`, the location parameter has a linear temporal trend. • `scale`: If `TRUE`, the scale parameter has a linear temporal trend.
`ns_slices`	For NS-FFA only: Numeric vector specifying the years at which to evaluate the return levels confidence intervals of a nonstationary probability distribution. `ns_slices` do not have to be elements of the `ns_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

Bootstrap samples are obtained from the fitted distribution via inverse transform sampling. For each bootstrapped sample, the parameters are re-estimated based on the `method` argument. 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 containing the following six items:

- `method`: "Bootstrap"
- `distribution`: The `distribution` argument.
- `params`: The fitted parameters.
- `ns_structure`: The `ns_structure` argument, if given.
- `ns_slices`: The `ns_slices` argument, if given.
- `ci`: A dataframe containing confidence intervals (S-FFA only)
- `ci_list`: A list of dataframes containing confidence intervals (NS-FFA only).

The dataframe(s) in `ci` and `ci_list` have four columns:

- `estimates`: Estimated quantiles for each return period.
- `lower`: Lower bound of the confidence interval for each return period.
- `upper`: Upper bound of the confidence interval for each return period.
- `periods`: The `periods` argument.

Note

The parametric bootstrap is known to give unreasonably wide confidence intervals for small datasets. If this method yields a confidence interval that is at least 5 times greater than the magnitude of the return levels, it will return an error and recommend `uncertainty_rfpl()` or `uncertainty_rfgpl()` as alternatives.

References

Vidrio-Sahagún, C.T., He, J. Enhanced profile likelihood method for the nonstationary hydrological frequency analysis, Advances in Water Resources 161, 10451 (2022). doi:10.1016/j.advwatres.2022.104151

See Also

`fit_lmoments()`, `fit_mle()`, `fit_gmle()`, `utils_sample_lmoments()` `utils_quantiles()`, `plot_sffa_estimates()`, `plot_nsffa_estimates()`

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
uncertainty_bootstrap(data, "WEI", "L-moments")
```

---

uncertainty_rfgpl *Regula-Falsi Generalized Profile Likelihood Uncertainty Quantification*

Description

Calculates return level estimates and confidence intervals at specified return periods (defaults to 2, 5, 10, 20, 50, and 100 years) using the regula-falsi generalized profile likelihood root-finding method for the GEV distribution.

For NS-FFA: To perform uncertainty quantification for a nonstationary model, include the observation years (`ns_years`), the nonstationary model structure (`ns_structure`), and a list of years at which to compute the return level estimates and confidence intervals (`ns_slices`).

Usage

```
uncertainty_rfgpl(
  data,
  prior,
  ns_years = NULL,
  ns_structure = NULL,
  ns_slices = NULL,
  alpha = 0.05,
  periods = c(2, 5, 10, 20, 50, 100),
  tolerance = 0.01
)
```

Arguments

`data`	Numeric vector of observed annual maximum series values. Must be strictly positive, finite, and not missing.
`prior`	Numeric vector of length 2. Specifies the parameters of the Beta prior for the shape parameter κ.

ns_years	For NS-FFA only: Numeric vector of observation years corresponding to data. Must be the same length as data and strictly increasing.
ns_structure	For NS-FFA only: Named list indicating which distribution parameters are modeled as nonstationary. Must contain two logical scalars: • location: If TRUE, the location parameter has a linear temporal trend. • scale: If TRUE, the scale parameter has a linear temporal trend.
ns_slices	For NS-FFA only: Numeric vector specifying the years at which to evaluate the return levels confidence intervals of a nonstationary probability distribution. ns_slices do not have to be elements of the ns_years argument.
alpha	Numeric scalar in $[0.01, 0.1]$. The significance level for confidence intervals or hypothesis tests. Default is 0.05.
periods	Numeric vector used to set the return periods for FFA. All entries must be greater than or equal to 1.
tolerance	The log-likelihood tolerance for Regula-Falsi convergence (default is 0.01).

Details

1. Uses fit_gmle() to obtain the maximum generalized log-likelihood.
2. Defines an objective function $f(y_p, p)$ by reparameterizing the generalized log-likelihood.
3. Iteratively brackets the root by rescaling 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.

Value

A list containing the following six items:

- method: "RFGPL"
- distribution: "GEV"
- params: The fitted parameters.
- ns_structure: The ns_structure argument, if given.
- ns_slices: The ns_slices argument, if given.
- ci: A dataframe containing confidence intervals (S-FFA only)
- ci_list: A list of dataframes containing confidence intervals (NS-FFA only).

The dataframe(s) in ci and ci_list have four columns:

- estimates: Estimated quantiles for each return period.
- lower: Lower bound of the confidence interval for each return period.
- upper: Upper bound of the confidence interval for each return period.
- periods: The periods argument.

Note

RFGPL uncertainty quantification can be numerically unstable for some datasets. If this function encounters an issue, it will return an error and recommend uncertainty_bootstrap() instead.

References

Vidrio-Sahagún, C.T., He, J. Enhanced profile likelihood method for the nonstationary hydrological frequency analysis, Advances in Water Resources 161, 10451 (2022). doi:10.1016/j.advwatres.2022.104151

Vidrio-Sahagún, C.T., He, J. & Pietroniro, A. Multi-distribution regula-falsi profile likelihood method for nonstationary hydrological frequency analysis. Stochastic Environmental Research and Risk Assessment 38, 843–867 (2024). doi:10.1007/s00477023026030

See Also

`utils_quantiles()`, `uncertainty_bootstrap()`, `uncertainty_rfpl()`, `plot_sffa_estimates()`, `plot_nsffa_estimates()`

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
uncertainty_rfgpl(data, c(6, 9))
```

`uncertainty_rfpl` *Regula-Falsi Profile Likelihood Uncertainty Quantification*

Description

Calculates return level estimates and confidence intervals at specified return periods (defaults to 2, 5, 10, 20, 50, and 100 years) using the regula-falsi profile likelihood root-finding method.

For NS-FFA: To perform uncertainty quantification for a nonstationary model, include the observation years (`ns_years`), the nonstationary model structure (`ns_structure`), and a list of years at which to compute the return level estimates and confidence intervals (`ns_slices`).

Usage

```
uncertainty_rfpl(
  data,
  distribution,
  ns_years = NULL,
  ns_structure = NULL,
  ns_slices = NULL,
  alpha = 0.05,
  periods = c(2, 5, 10, 20, 50, 100),
  tolerance = 0.01
)
```

Arguments

`data`	Numeric vector of observed annual maximum series values. Must be strictly positive, finite, and not missing.
`distribution`	A three-character code indicating the distribution family. Must be `"GUM"`, `"NOR"`, `"LNO"`, `"GEV"`, `"GLO"`, `"GNO"`, `"PE3"`, `"LP3"`, or `"WEI"`.
`ns_years`	For NS-FFA only: Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.
`ns_structure`	For NS-FFA only: Named list indicating which distribution parameters are modeled as nonstationary. Must contain two logical scalars: • `location`: If `TRUE`, the location parameter has a linear temporal trend. • `scale`: If `TRUE`, the scale parameter has a linear temporal trend.
`ns_slices`	For NS-FFA only: Numeric vector specifying the years at which to evaluate the return levels confidence intervals of a nonstationary probability distribution. `ns_slices` do not have to be elements of the `ns_years` argument.
`alpha`	Numeric scalar in $[0.01, 0.1]$. The significance level for confidence intervals or hypothesis tests. Default is 0.05.
`periods`	Numeric vector used to set the return periods for FFA. All entries must be greater than or equal to 1.
`tolerance`	The log-likelihood tolerance for Regula-Falsi convergence (default is 0.01).

Details

1. Uses `fit_mle()` to obtain the maximum log-likelihood.
2. Defines an objective function $f(y_p, p)$ by reparameterizing the log-likelihood.
3. Iteratively brackets the root by rescaling 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`.

Value

A list containing the following four items:

- `method`: "RFPL"
- `distribution`: The `distribution` argument.
- `params`: The fitted parameters.
- `ns_structure`: The `ns_structure` argument, if given.
- `ns_slices`: The `ns_slices` argument, if given.
- `ci`: A dataframe containing confidence intervals (S-FFA only)
- `ci_list`: A list of dataframes containing confidence intervals (NS-FFA only).

The dataframe(s) in `ci` and `ci_list` have four columns:

- `estimates`: Estimated quantiles for each return period.
- `lower`: Lower bound of the confidence interval for each return period.
- `upper`: Upper bound of the confidence interval for each return period.
- `periods`: The `periods` argument.

Note

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 method `uncertainty_bootstrap()` instead.

References

Vidrio-Sahagún, C.T., He, J. Enhanced profile likelihood method for the nonstationary hydrological frequency analysis, Advances in Water Resources 161, 10451 (2022). doi:10.1016/j.advwatres.2022.104151

Vidrio-Sahagún, C.T., He, J. & Pietroniro, A. Multi-distribution regula-falsi profile likelihood method for nonstationary hydrological frequency analysis. Stochastic Environmental Research and Risk Assessment 38, 843–867 (2024). doi:10.1007/s00477023026030

See Also

`utils_quantiles()`, `uncertainty_bootstrap()`, `uncertainty_rfgpl()`, `plot_sffa_estimates()`, `plot_nsffa_estimates()`

Examples

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

---

utils_cdf *Cumulative Distribution Functions for Probability Models*

Description

Compute probabilities from quantiles for both stationary and nonstationary models.

For NS-FFA: To compute the probabilities for a nonstationary model, specify a time slice (`ns_slice`) and the nonstationary model structure (`ns_structure`).

Usage

```
utils_cdf(q, distribution, params, ns_slice = 0, ns_structure = NULL)
```

Arguments

`q`	Numeric vector of quantiles with no missing values.
`distribution`	A three-character code indicating the distribution family. Must be `"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 `distribution` and `structure`.

`ns_slice` For NS-FFA only: Numeric scalar specifying the year at which to evaluate the quantiles of a nonstationary probability distribution. `ns_slice` does not have to be an element of the `ns_years` argument.

`ns_structure` For NS-FFA only: Named list indicating which distribution parameters are modeled as nonstationary. Must contain two logical scalars:

- `location`: If `TRUE`, the location parameter has a linear temporal trend.
- `scale`: If `TRUE`, the scale parameter has a linear temporal trend.

Value

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

Examples

```
q <- seq(1, 10)
params <- c(1, 1, 1)
utils_cdf(q, "GEV", params)
```

`utils_generalized_likelihood`
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 geophysical (Beta) prior distribution for the shape parameter (Martins and Stedinger, 2000).

For NS-FFA: To compute the generalized log-likelihood for a nonstationary probability model, include the observation years (`ns_years`) and the nonstationary model structure (`ns_structure`).

Usage

```
utils_generalized_likelihood(
  data,
  params,
  prior,
  ns_years = NULL,
  ns_structure = NULL
)
```

Arguments

`data` Numeric vector of observed annual maximum series 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 `distribution` and `structure`.
`prior`	Numeric vector of length 2. Specifies the parameters of the Beta prior for the shape parameter κ.
`ns_years`	For NS-FFA only: Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.
`ns_structure`	For NS-FFA only: Named list indicating which distribution parameters are modeled as nonstationary. Must contain two logical scalars: • `location`: If `TRUE`, the location parameter has a linear temporal trend. • `scale`: If `TRUE`, the scale parameter has a linear temporal trend.

Details

The generalized log-likelihood is defined as sum of (1) the log-likelihood and (2) 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.

References

El Adlouni, S., Ouarda, T.B.M.J., Zhang, X., Roy, R., Bobee, B., 2007. Generalized maximum likelihood estimators for the nonstationary generalized extreme value model. Water Resources Research 43 (3), 1–13. doi:10.1029/2005WR004545

Martins, E. S., and Stedinger, J. R. (2000). Generalized maximum-likelihood generalized extreme-value quantile estimators for hydrologic data. Water Resources Research, 36(3), 737–744. doi:10.1029/1999WR900330

See Also

`utils_log_likelihood()`

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
params <- c(100, 10, 0.1)
prior <- c(1, 1)

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

`utils_log_likelihood` *Log-Likelihood Functions for Probability Models*

Description

Compute the log-likelihood for stationary and nonstationary probability models.

For NS-FFA: To compute the log-likelihood for a nonstationary probability model, include the observation years (`ns_years`) and the nonstationary model structure (`ns_structure`).

Usage

```
utils_log_likelihood(
  data,
  distribution,
  params,
  ns_years = NULL,
  ns_structure = NULL
)
```

Arguments

`data`	Numeric vector of observed annual maximum series values. Must be strictly positive, finite, and not missing.
`distribution`	A three-character code indicating the distribution family. Must be `"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 `distribution` and `structure`.
`ns_years`	For NS-FFA only: Numeric vector of observation years corresponding to `data`. Must be the same length as `data` and strictly increasing.
`ns_structure`	For NS-FFA only: Named list indicating which distribution parameters are modeled as nonstationary. Must contain two logical scalars: • `location`: If `TRUE`, the location parameter has a linear temporal trend. • `scale`: If `TRUE`, the scale parameter has a linear temporal trend.

Value

Numeric scalar. The log-likelihood value.

See Also

`utils_generalized_likelihood()`

Examples

```
data <- rnorm(n = 100, mean = 100, sd = 10)
params <- c(100, 1, 10)
ns_years <- seq(from = 1901, to = 2000)
ns_structure <- list(location = TRUE, scale = FALSE)

# Compute the log-likelihood
utils_log_likelihood(data, "NOR", params, ns_years, ns_structure)
```

---

utils_quantiles *Quantile Functions for Probability Models*

Description

Compute the quantiles for stationary and nonstationary probability models.

For NS-FFA: To compute the quantiles for a nonstationary probability model, specify a time slice (`ns_slice`) and the nonstationary model structure (`ns_structure`).

Usage

```
utils_quantiles(p, distribution, params, ns_slice = 0, ns_structure = NULL)
```

Arguments

`p`	Numeric vector of probabilities between 0 and 1 with no missing values.
`distribution`	A three-character code indicating the distribution family. Must be `"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 `distribution` and `structure`.
`ns_slice`	For NS-FFA only: Numeric scalar specifying the year at which to evaluate the quantiles of a nonstationary probability distribution. `ns_slice` does not have to be an element of the `ns_years` argument.
`ns_structure`	For NS-FFA only: Named list indicating which distribution parameters are modeled as nonstationary. Must contain two logical scalars: • `location`: If `TRUE`, the location parameter has a linear temporal trend. • `scale`: If `TRUE`, the scale parameter has a linear temporal trend.

Value

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

Examples

```
p <- runif(n = 100)
params <- c(1, 1, 1)
utils_quantiles(p, "GEV", params)
```

`utils_sample_lmoments` *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

```
utils_sample_lmoments(data)
```

Arguments

`data`	Numeric vector of observed annual maximum series 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 sample L-moments and L-moment ratios:

- l_1: L-mean
- l_2: L-variance
- t_3: L-skewness
- t_4: 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)
utils_sample_lmoments(data)
```

---

utils_theoretical_lmoments

Theoretical L-moments of Probability Distributions

Description

Computes the first four L-moments and L-moment ratios for *stationary* probability models.

Usage

```
utils_theoretical_lmoments(distribution, params)
```

Arguments

`distribution`	A three-character code indicating the distribution family. Must be `"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 `distribution` and `structure`.

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). The L-moments ratios for the `"LNO"` and `"LP3"` distributions are should be compared to the log-transformed data and are thus identical to the `"NOR"` and `"PE3"` distributions respectively.

Value

A numeric vector of with four elements:

- 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

```
utils_theoretical_lmoments("GEV", c(1, 1, 1))
```

Index

∗ **datasets**
  CAN_05BB001, 5
  CAN_07BE001, 6
  CAN_08MH016, 7
  CAN_08NH021, 7
  CAN_08NM050, 8
∗ **package**
  ffaframework-package, 3

CAN_05BB001, 5
CAN_07BE001, 6
CAN_08MH016, 7
CAN_08NH021, 7
CAN_08NM050, 8

data_decomposition, 9
data_decomposition(), *51–54*
data_geomet, 10
data_geomet(), *11*
data_local, 11
data_local(), *10*
data_mw_variability, 12
data_mw_variability(), *9, 10*, *39, 40*
data_screening, 13

eda_bbmk_test, 14
eda_bbmk_test(), *17*, *24*, *32*, *40*
eda_kpss_test, 15
eda_kpss_test(), *21*, *32*
eda_mk_test, 16
eda_mk_test(), *15*, *32*
eda_mks_test, 17
eda_mks_test(), *20*, *32*, *42*
eda_pettitt_test, 19
eda_pettitt_test(), *18*, *32*, *46*
eda_pp_test, 20
eda_pp_test(), *16*, *32*
eda_runs_test, 21
eda_runs_test(), *23*, *32*, *47*
eda_sens_trend, 23
eda_sens_trend(), *10*, *22*, *32*, *39, 40*
eda_spearman_test, 24
eda_spearman_test(), *14, 15*, *32*, *50*
eda_white_test, 25
eda_white_test(), *32*

ffaframework *(*ffaframework-package*)*, 3
ffaframework-package, 3
fit_gmle, 26
fit_gmle(), *28*, *35*, *44, 45*, *49*, *57, 58*
fit_lmoments, 27
fit_lmoments(), *26, 27*, *29–31*, *35*, *44, 45*, *49*, *57*
fit_lmoments_kappa, 28
fit_lmoments_kappa(), *28*, *54*
fit_mle, 30
fit_mle(), *28*, *35*, *44, 45*, *49*, *57*, *60*
framework_eda, 31
framework_eda(), *36*
framework_ffa, 33
framework_ffa(), *36*
framework_full, 35

model_assessment, 37
model_assessment(), *3*, *35*, *43*, *48*

plot_ams_data, 39
plot_ams_data(), *23*
plot_bbmk_test, 40
plot_bbmk_test(), *15*
plot_lmom_diagram, 41
plot_lmom_diagram(), *52–54*
plot_mks_test, 42
plot_mks_test(), *18*
plot_nsffa_assessment, 43
plot_nsffa_assessment(), *38*
plot_nsffa_estimates, 44
plot_nsffa_estimates(), *57*, *59*, *61*
plot_nsffa_fit, 45
plot_pettitt_test, 46

plot_pettitt_test(), *20*
plot_runs_test, 47
plot_runs_test(), *22*
plot_sffa_assessment, 48
plot_sffa_assessment(), *38*
plot_sffa_estimates, 48
plot_sffa_estimates(), *57*, *59*, *61*
plot_sffa_fit, 49
plot_spearman_test, 50
plot_spearman_test(), *24*

select_ldistance, 51
select_ldistance(), *35*, *41*, *53, 54*
select_lkurtosis, 52
select_lkurtosis(), *35*, *41*, *52*, *54*
select_zstatistic, 53
select_zstatistic(), *35*, *41*, *52, 53*
stats::cor.test(), *24*
stats::lm(), *26*
stats::nlminb(), *26, 27*, *30, 31*
stats::optim(), *29*, *52*
stats::pchisq(), *26*

uncertainty_bootstrap, 55
uncertainty_bootstrap(), *35*, *38*, *44*, *49*, *58, 59*, *61*
uncertainty_rfgpl, 57
uncertainty_rfgpl(), *35*, *38*, *44*, *49*, *56*, *61*
uncertainty_rfpl, 59
uncertainty_rfpl(), *35*, *38*, *44*, *49*, *56*, *59*
utils_cdf, 61
utils_generalized_likelihood, 62
utils_generalized_likelihood(), *26, 27*, *64*
utils_log_likelihood, 64
utils_log_likelihood(), *30, 31*, *63*
utils_quantiles, 65
utils_quantiles(), *54*, *57*, *59*, *61*
utils_sample_lmoments, 66
utils_sample_lmoments(), *28, 29*, *52, 53*, *57*
utils_theoretical_lmoments, 67