Package 'RecordTest'

August 8, 2021

Type Package

Title Inference Tools in Time Series Based on Record Statistics

Version 2.1.0

Date 2021-08-05

Author Jorge Castillo-Mateo [aut, cre, cph]
(<https://orcid.org/0000-0003-3859-0248>),
Ana C. Cebrián [ths] (<https://orcid.org/0000-0002-9052-9674>),
Jesús Asín [ths] (<https://orcid.org/0000-0002-0174-789X>)

Maintainer Jorge Castillo-Mateo <jorgecastillomateo@gmail.com>

Depends R (>= 3.5.0)

Imports ggplot2, stats

Suggests ggpubr, knitr, rmarkdown

Description Statistical tools based on the probabilistic properties of the record occurrence in a sequence of independent and identically distributed continuous random variables. In particular, tools to prepare a time series as well as distribution-free trend and change-point tests and graphical tools to study the record occurrence.

License GPL-3

URL https://github.com/JorgeCastilloMateo/RecordTest

BugReports https://github.com/JorgeCastilloMateo/RecordTest/issues

VignetteBuilder knitr

Encoding UTF-8

LazyData true

NeedsCompilation no

RoxygenNote 7.1.1

Repository CRAN

Date/Publication 2021-08-08 15:50:01 UTC

R topics documented:

`RecordTest-package` **RecordTest**: *A Package for Testing the Classical Record Model*

Description

RecordTest provides data preparation, exploratory data analysis and inference tools based on theory of records to describe the record occurrence and detect trends, change-points or non-stationarities in the tails of the time series.

Details

The Classical Record Model:

Record statistics are used primarily to quantify the stochastic behaviour of a process at never-seen-before values, either upper or lower. The setup of independent and identically distibuted (IID) continuous random variables (RVs), often called the classical record model, is particularly interesting because the common continuous distribution underlying the IID continuous RVs will not affect the distribution of the variables relative to the record occurrence. Many fields have begun to use the theory of records to study these remarkable events. Particularly productive is the study of record-breaking temperatures and their connection with climate change, but also records in other environmental fields (precipitations, floods, earthquakes, etc.), economy, biology, physics or even sports have been analysed. See Arnold, Balakrishnan and Nagaraja (1998) for an extensive theoretical introduction to the theory of records and in particular the classical record model. See Foster and Stuart (1954), Diersen and Trenkler (1996, 2001) and Cebrián, Castillo-Mateo and Asín (2021) for the distribution-free trend detection tests and Castillo-Mateo (2021) for the distribution-free change-point detection tests based on the classical record model. For an easy introduction to **RecordTest** use `vignette("RecordTest")`.

This package provides tests to study the hypothesis of the classical record model, that is that the record occurrence from a series of values observed at regular time units come from an IID series of continuous RVs. If we have sequences of independent variables with no seasonal component, the hypothesis of IID variables is equivalent to test the hypothesis of homogeneity and stationarity.

The functions in the data preparation step:

The functions admit a vector `X` corresponding to a single series as an argument. However, some situations could take advantage of having M uncorrelated vectors to infer from the sample. Then, the input of the functions to perform the statistical tools can be a matrix `X` where each column corresponds to a vector formed by the values of a series X_t, for $t = 1, \ldots, T$, so that each row of the matrix correspond to a time t.

In many real problems, such as those related to environmental phenomena, the series of variables to analyse show a seasonal behaviour, and only one realisation is available. In order to be able to apply the suggested tools to detect the existence of a trend, the seasonal component has to be removed and a sample of M uncorrelated series should be obtained. Those problems can be solved by preparing the data adequately. A wide set of tools to carry out a preliminary analysis and to prepare data with a seasonal pattern are implemented in the following functions.

`series_record`: If only the record times are available.

`series_split`, `series_double`: To split the series in several subseries and remove the seasonal component and autocorrelation.

`series_uncor`: To extract a subset of uncorrelated subseries

`series_ties`, `series_untie`: To deal with record ties.

`series_rev`: To study the series backwards.

The functions to compute the record statistics are:

`I.record`: Computes the observed record indicators. `NA` values are taken as no records unless they appear at $t = 1$.

`N.record`, `Nmean.record`: Compute the observed number of records up to time t.

`S.record`: Computes the observed number of records at every time t, using M series.

p.record: Computes the estimated record probability at every time t, using M series.

L.record: Computes the observed record times.

R.record: Computes the observed record values.

The functions to compute the tests:

All the tests performed are distribution-free/non-parametric tests in time series for trend, change-point and non-stationarity in the extremes of the distribution based on the null hypothesis that the record indicators are independent and the probabilities of record at time t are $p_t = 1/t$.

change.point: Implements Castillo-Mateo change-point tests.

foster.test: Implements Foster-Stuart and Diersen-Trenkler trend tests.

N.test: Implements tests based on the (weighted) number of records.

brown.method: Brown's method to combine dependent p-values from N.test.

fisher.method: General function to apply Fisher's method to independent p-values.

p.regression.test: Implements a regression test based on the record probabilities.

p.chisq.test: Implements a χ^2-test based on the record probabilities.

lr.test: Implements likelihood ratio tests based on the record indicators.

score.test: Implements score or Lagrange multiplier tests based on the record indicators.

The functions to compute the graphical tools:

records: Shows the series remarking its records.

L.plot: Shows record times in several series.

foster.plot: Shows plots based on Foster-Stuart and Diersen-Trenkler statistics.

N.plot: Shows the (weighted) number of records.

p.plot: Shows the record probabilities in different plots.

All the tests and graphical tools can be applied to both upper and lower records in the forward and backward directions.

Other functions:

rcrm: Random generation for the classical record model.

dpoisbinom, ppoisbinom, qpoisbinom, rpoisbinom: Density, distribution function, quantile function and random generation for the Poisson binomial distribution. Related to the probability distribution function of the number of records under the null hypothesis.

Example datasets:

There are two example datasets included with this package. It is possible to load these datasets into R using the `data` function. The datasets have their own help file, which can be accessed by `help([dataset_name])`. Data included with **RecordTest** are:

TX_Zaragoza - Daily maximum temperatures at Zaragoza (Spain).

ZaragozaSeries - Splitted and uncorrelated subseries TX_Zaragoza$TX.

Olympic_records_200m - 200-meter Olympic records from 1900 to 2020.

To see how to cite **RecordTest** in publications or elsewhere, use `citation("RecordTest")`.

Author(s)

Jorge Castillo-Mateo, AC Cebrián, J Asín

References

Arnold BC, Balakrishnan N, Nagaraja HN (1998). *Records*. Wiley Series in Probability and Statistics. Wiley, New York.

Castillo-Mateo J (2021). "Distribution-Free Changepoint Detection Tests Based on the Breaking of Records." Available at arXiv:2105.08186

Cebrián A, Castillo-Mateo J, Asín J (2021). "Record Tests to Detect Non Stationarity in the Tails with an Application to Climate Change." Available at Research Square doi: 10.21203/rs.3.rs214787/v1

Diersen J, Trenkler G (1996). "Records Tests for Trend in Location." *Statistics*, **28**(1), 1-12. doi: 10.1080/02331889708802543

Diersen J, Trenkler G (2001). "Weighted Records Tests for Splitted Series of Observations." In J Kunert, G Trenkler (eds.), *Mathematical Statistics with Applications in Biometry: Festschrift in Honour of Prof. Dr. Siegfried Schach*, pp. 163–178. Lohmar: Josef Eul Verlag.

Foster FG, Stuart A (1954). "Distribution-Free Tests in Time-Series Based on the Breaking of Records." *Journal of the Royal Statistical Society. Series B (Methodological)*, **16**(1), 1-22.

`brown.method` *Brown's Method on the Number of Records*

Description

Performs Brown's method on the p-values of `N.test` as proposed by Cebrián, Castillo-Mateo and Asín (2021). The null hypothesis of the classical record model (i.e., of IID continuous RVs) is tested against the alternative hypothesis.

Usage

```
brown.method(
  X,
  weights = function(t) 1,
  record = c(FU = 1, FL = 1, BU = 1, BL = 1),
  alternative = c(FU = "greater", FL = "less", BU = "less", BL = "greater"),
  correct = TRUE
)
```

Arguments

`X`	A numeric vector, matrix (or data frame).
`weights`	A function indicating the weight given to the different records according to their position in the series, e.g., if `function(t) t-1` then $\omega_t = t - 1$.

`record`	Vector of length four. Each element is a logical indicating if the p-value of the test for forward upper, forward lower, backward upper and backward lower are going to be used, respectively. Logical values or 0,1 values are accepted.
`alternative`	Vector of length four. Each element is one of `"greater"` or `"less"` indicating the alternative hypothesis in every test (for forward upper, forward lower, backward upper and backward lower records, respectively). Under the alternative hypothesis of linear trend the FU and BL records will be greater and the FL and BU records will be less than under the null, but other combinations (e.g., for trend in variation) could be considered.
`correct`	Logical. Indicates, whether a continuity correction should be applied in `N.test`; defaults to `TRUE`.

Details

The test is implemented as given by Cebrián, Castillo-Mateo and Asín (2021), where the p-values $p^{(FU)}$, $p^{(FL)}$, $p^{(BU)}$, and $p^{(BL)}$ of the test `N.test` for the four types of record are used for the statistic:

$$-2\left(\log(p^{(FU)}) + \log(p^{(FL)}) + \log(p^{(BU)}) + \log(p^{(BL)})\right).$$

Any other combination of p-values for the test is also allowed (see argument `record`).

According to Brown's method (Brown, 1975) for the union of dependent p-values, the statistic follows a $c\chi^2_{df}$ distribution, with a scale parameter c and df degrees of freedom that depend on the covariance of the p-values. This covariances are approximated according to Kost and McDermott (2002):

$$\text{COV}\left(-2\log(p^{(i)}), -2\log(p^{(j)})\right) \approx 3.263\rho_{ij} + 0.710\rho_{ij}^2 + 0.027\rho_{ij}^3,$$

where ρ_{ij} is the correlation between their respective statistics.

Power studies indicate that this and `foster.test` using all four types of record and linear weights are the two most powerful records tests for trend detection against a linear drift model. In particular, this test is more powerful than Mann-Kendall test against alternatives with a linear drift in location in series of generalised Pareto variables and some cases of the generalised extreme value variables (see Cebrián, Castillo-Mateo and Asín, 2021).

Value

A `"htest"` object with elements:

`statistic`	Value of the chi-square statistic (not scaled).
`parameter`	Degrees of freedom df and scale parameter c.
`p.value`	P-value.
`method`	A character string indicating the type of test performed.
`data.name`	A character string giving the name of the data.

Author(s)

Jorge Castillo-Mateo

References

Brown M (1975). "A Method for Combining Non-Independent, One-Sided Tests of Significance." *Biometrics*. **31**(4), 987–992.

Cebrián A, Castillo-Mateo J, Asín J (2021). "Record Tests to Detect Non Stationarity in the Tails with an Application to Climate Change." Available at Research Square doi: 10.21203/rs.3.rs214787/v1

Kost JT, McDermott MP (2002). "Combining Dependent P-Values." *Statistics & Probability Letters*, **60**(2), 183-190.

See Also

`fisher.method`, `foster.test`, `N.test`

Examples

```
brown.method(ZaragozaSeries)
brown.method(ZaragozaSeries, weights = function(t) t-1)
brown.method(ZaragozaSeries, weights = function(t) t-1, correct = FALSE)

# Join p-values of upper records
brown.method(ZaragozaSeries, weights = function(t) t-1, record = c(1,0,1,0))
# Join p-values of lower records
brown.method(ZaragozaSeries, weights = function(t) t-1, record = c(0,1,0,1))
```

`change.point` *Change-point Detection Tests Based on Records*

Description

Performs change-point detection tests based on the record occurrence. The hypothesis of the classical record model (i.e., of IID continuous RVs) is tested against the alternative hypothesis that after a certain time the series stops being stationary.

Usage

```
change.point(
  X,
  weights = function(t) 1,
  record = c("upper", "lower", "d", "s"),
  correct = c("none", "fisher", "vrbik"),
  simulate.p.value = FALSE,
  B = 1000
)
```

Arguments

`X` — A numeric vector, matrix (or data frame).

`weights` — A function indicating the weight given to the different records according to their position in the series. Castillo-Mateo (2021) showed that the weights that get more power for this test are $\omega_t = \sqrt{t^2/(t-1)}$, i.e., `weights = function(t) ifelse(t == 1, 0, sqrt(t^2 / (t-1)))` if `record = "upper"` or `= "lower"`. $\omega_t = \sqrt{t}$, i.e., `weights = function(t) sqrt(t)` if `record = "d"` and $\omega_t = \sqrt{t^2/(t-2)}$, i.e., `weights = function(t) ifelse(t %in% 1:2, 0, sqrt(t^2 / (t-2)))` if `record = "s"`.

`record` — A character string that indicates the type of statistic used. If `record = "upper"` or `= "lower"`, the I_{tm}'s are the upper or lower record indicators, respectively. If `record = "d"` or `= "s"`, I_{tm} is substituted in the expressions above by $I_{tm}^{(FU)} - I_{tm}^{(FL)}$ or $I_{tm}^{(FU)} + I_{tm}^{(FL)}$, respectively.

`correct` — A character string that indicates the continuity correction in the Kolmogorov distribution made to the statistic. "fisher" (Fisher and Robbins 2019), "vrbik" for the proposal by Vrbik (2020) or "none" (the default) if no correction is made. The former shows better size and power, but if the value of the statistic is too large it becomes `NaN` and p-value `NA`.

`simulate.p.value` — Logical. Indicates whether to compute p-values by Monte Carlo simulation.

`B` — If `simulate.p.value = TRUE`, an integer specifying the number of replicates used in the Monte Carlo estimation.

Details

The test is implemented as given by Castillo-Mateo (2021). The null hypothesis is that

$$H_0 : p_t = 1/t, \qquad t = 1, \ldots, T,$$

where p_t is the probability of (upper and/or lower) record at time t and T is the length of the series, which corresponds to the setup where the data come from a population with continuous independent and identically distributed realisations. The two-sided alternative hypothesis is that

$$H_1 : p_t = 1/t, \quad t = 1, \ldots, t_0, \qquad p_t \neq 1/t, \quad t = t_0 + 1, \ldots, T,$$

for a change-point t_0.

The variables used for the statistic are

$$\xi_{Tt} = \omega_t \frac{\left(\frac{1}{M} \sum_{m=1}^{M} I_{tm} - E(I_{tm}) \right)}{\sigma_T},$$

where $\sigma_t^2 = \sum_{k=1}^{t} \omega_k^2 Var(I_k)/M$ and ω_t are weights given to the different records according to their position in the mth series of length T. If `record = "upper"` or `= "lower"`, the I_{tm}'s are the upper or lower record indicators, respectively (see `I.record`). If `record = "d"` or `= "s"`, I_{tm} is substituted in the expressions above by $I_{tm}^{(FU)} - I_{tm}^{(FL)}$ or $I_{tm}^{(FU)} + I_{tm}^{(FL)}$, respectively. Let $S_{Tt} = \sum_{k=1}^{t} \xi_{Tk}$, $s_{Tt} = \sum_{k=1}^{t} Var(\xi_{Tk})$ and define the statistic

$$W_T(s) = S_{Tt} + \xi_{T,t+1} \frac{s - s_{Tt}}{s_{T,t+1} - s_{Tt}},$$

where $s \in [s_{Tt}, s_{T,t+1}]$.

The test statistic is the maximum of the absolute value of the vector

$$K_T = \max_{1 \le t \le T} |B_T(s_{Tt})|,$$

where $B_T(s) = W_T(s) - sW_T(1)$, $s \in [0, 1]$.

The probable change-point $\hat{t}_0$ is located where $|B_T(s_{Tt})|$ has its maximum. The p-value is calculated by means of the asymptotic Kolmogorov distribution. When $\omega_t \neq 1$, the asymptotic result is not fulfilled. In that case, the p-value can be simulated using Monte Carlo techniques with the option `simulate.p.value = TRUE`.

As this is an asymptotic result, it has been seen that the size and power may be a little below than expected, to correct this, any of the continuity corrections can be used:

If `correct = "fisher"`,

$$K_T = -\sqrt{T} \log\left(1 - \frac{K_T}{\sqrt{T}}\right).$$

If `correct = "vrbik"`,

$$K_T = K_T + \frac{1}{6\sqrt{T}} + \frac{K_T - 1}{4T}.$$

Value

A `"htest"` object with elements:

`statistic`	Value of the test statistic.
`p.value`	P-value.
`alternative`	The alternative hypothesis.
`estimate`	The estimated change-point time.
`method`	A character string indicating the type of test performed.
`data.name`	A character string giving the name of the data.

Author(s)

Jorge Castillo-Mateo

References

Castillo-Mateo J (2021). "Distribution-Free Changepoint Detection Tests Based on the Breaking of Records." Available at arXiv:2105.08186

Fisher TJ, Robbins MW (2019). "A Cheap Trick to Improve the Power of a Conservative Hypothesis Test." *The American Statistician*, **73**(3), 232-242. doi: 10.1080/00031305.2017.1395364

Vrbik J (2020). "Deriving CDF of Kolmogorov-Smirnov Test Statistic." *Applied Mathematics*, **11**(3), 227-246. doi: 10.4236/am.2020.113018

See Also

`records`, `foster.test`

Examples

```
change.point(ZaragozaSeries)

change.point(ZaragozaSeries, record = "d",
  weights = function(t) sqrt(t), simulate.p.value = TRUE)

test.result <- change.point(rowMeans(ZaragozaSeries))
test.result

## Not run: Load package ggplot2 to plot the changepoint
#library("ggplot2")
#records(rowMeans(ZaragozaSeries)) +
#  ggplot2::geom_vline(xintercept = test.result$estimate, colour = "red")
```

`fisher.method` *Fisher's Method*

Description

Performs Fisher's method, which is used to combine the p-values from several independent tests bearing upon the same overall null hypothesis.

Usage

```
fisher.method(p.values)
```

Arguments

`p.values` A vector containing the p-values from the single tests.

Details

Fisher's method (Fisher, 1925) combines the p-values from k independent tests, into one test statistic using the following formula:

$$-2 \sum_{i=1}^{k} \log(p_i) \sim \chi^2_{2k},$$

where p_i is the p-value for the ith hypothesis test.

For example, Foster and Stuart (1954) proposed this method to combine the information of the p-values from the D and S-statistics (see Examples), since they are independent.

Value

A "htest" object with elements:

`statistic` Value of the test statistic.

`parameter` Degrees of freedom.

p.value	P-value.
method	A character string indicating the type of test performed.
data.name	A character string giving the name of the data.

Author(s)

Jorge Castillo-Mateo

References

Fisher RA (1925). *Statistical Methods for Research Workers*. Oliver and Boyd, Edinburgh.

Foster FG, Stuart A (1954). "Distribution-Free Tests in Time-Series Based on the Breaking of Records." *Journal of the Royal Statistical Society. Series B (Methodological)*, **16**(1), 1-22.

See Also

brown.method, foster.test

Examples

```
# Join the independent p-values of the D and S-statistics
fisher.method(c(foster.test(ZaragozaSeries, statistic = "D")$p.value,
  foster.test(ZaragozaSeries, statistic = "S")$p.value))
# Adding weights
fisher.method(c(foster.test(ZaragozaSeries, weights = function(t) t-1, statistic = "D")$p.value,
  foster.test(ZaragozaSeries, weights = function(t) t-1, statistic = "S")$p.value))
```

---

foster.plot — *Plots Based on Foster-Stuart and Diersen-Trenkler Statistics*

Description

This function builds a ggplot object to display two-sided confidence intervals based on Foster-Stuart and Diersen-Trenkler statistics to study the hypothesis of the classical record model (i.e., of IID continuous RVs).

Usage

```
foster.plot(
  X,
  weights = function(t) 1,
  statistic = c("D", "d", "S", "s", "U", "L", "W"),
  point.col = "black",
  point.shape = 19,
  conf.int = TRUE,
  conf.level = 0.9,
```

```
  conf.aes = c("ribbon", "errorbar"),
  conf.col = "grey69"
)
```

Arguments

`X`	A numeric vector, matrix (or data frame).
`weights`	A function indicating the weight given to the different records according to their position in the series, e.g., if `function(t) t-1` then $\omega_t = t - 1$.
`statistic`	A character string indicating the type of statistic to be calculated, i.e., one of `"D"`, `"d"`, `"S"`, `"s"`, `"U"`, `"L"` or `"W"` (see `foster.test`).
`point.col`, `point.shape`	Value with the colour and shape of the points.
`conf.int`	Logical. Indicates if the CIs are also shown.
`conf.level`	(If `conf.int == TRUE`) Confidence level of the CIs.
`conf.aes`	(If `conf.int == TRUE`) A character string indicating the aesthetic to display for the CIs, `"ribbon"` (grey area) or `"errorbar"` (vertical lines).
`conf.col`	Colour used to plot the expected value and (if `conf.int == TRUE`) CIs.

Details

The function plots the observed values of the statistic selected with `statistic`, obtained with the series up to time t for $t = 1, \ldots, T$. The plot also includes the expected values and confidence intervals (CIs) based on the asymptotic normal distribution of the statistics under the null hypothesis.

This function implements the same ideas that `N.plot`, but with the statistics computed in `foster.test`.

These plots are useful to see the evolution in the record occurrence and to follow the evolution of the trend. The plot was proposed by Cebrián, Castillo-Mateo, Asín (2021) where its application is shown.

Value

A ggplot graph object.

Author(s)

Jorge Castillo-Mateo

References

Cebrián A, Castillo-Mateo J, Asín J (2021). “Record Tests to Detect Non Stationarity in the Tails with an Application to Climate Change.” Available at Research Square doi: 10.21203/rs.3.rs214787/v1

Diersen J, Trenkler G (1996). “Records Tests for Trend in Location.” *Statistics*, **28**(1), 1-12. doi: 10.1080/02331889708802543

Diersen J, Trenkler G (2001). “Weighted Records Tests for Splitted Series of Observations.” In J Kunert, G Trenkler (eds.), *Mathematical Statistics with Applications in Biometry: Festschrift in Honour of Prof. Dr. Siegfried Schach*, pp. 163–178. Lohmar: Josef Eul Verlag.

Foster FG, Stuart A (1954). "Distribution-Free Tests in Time-Series Based on the Breaking of Records." *Journal of the Royal Statistical Society. Series B (Methodological)*, **16**(1), 1-22.

See Also

foster.test, N.plot, N.test

Examples

```
# D-statistic
foster.plot(ZaragozaSeries)
# D-statistic with linear weights
foster.plot(ZaragozaSeries, weights = function(t) t-1)
# S-statistic with linear weights
foster.plot(ZaragozaSeries, statistic = "S", weights = function(t) t-1)
# U-statistic with weights (upper tail)
foster.plot(ZaragozaSeries, statistic = "U", weights = function(t) t-1)
# L-statistic with weights (lower tail)
foster.plot(ZaragozaSeries, statistic = "L", weights = function(t) t-1)
```

---

foster.test *Foster-Stuart and Diersen-Trenkler Tests*

Description

Performs Foster-Stuart, Diersen-Trenkler and Cebrián-Castillo-Asín records tests for trend in location, variation or the tails. The hypothesis of the classical record model (i.e., of IID continuous RVs) is tested against the alternative hypothesis.

Usage

```
foster.test(
  X,
  weights = function(t) 1,
  statistic = c("D", "d", "S", "s", "U", "L", "W"),
  distribution = c("normal", "t"),
  alternative = c("greater", "less"),
  correct = FALSE,
  simulate.p.value = FALSE,
  B = 1000
)
```

Arguments

X	A numeric vector, matrix (or data frame).
weights	A function indicating the weight given to the different records according to their position in the series, e.g., if `function(t) t-1` then $\omega_t = t - 1$.

`statistic` A character string indicating the type of statistic to be calculated, i.e., one of `"D"`, `"d"`, `"S"`, `"s"`, `"U"`, `"L"` or `"W"` (see Details).

`distribution` A character string indicating the asymptotic distribution of the statistic, `"normal"` or Student's `"t"` distribution.

`alternative` A character string indicating the type of alternative hypothesis, `"greater"` number of records or `"less"` number of records.

`correct` Logical. Indicates, whether a continuity correction should be done; defaults to `FALSE`. No correction is done if `simulate.p.value = TRUE`.

`simulate.p.value`
Logical. Indicates whether to compute p-values by Monte Carlo simulation.

`B` If `simulate.p.value = TRUE`, an integer specifying the number of replicates used in the Monte Carlo estimation.

Details

In this function, the tests are implemented as given by Foster and Stuart (1954), Diersen and Trenkler (1996, 2001) and some modifications in the standardisation of the previous statistics given by Cebrián, Castillo-Mateo and Asín (2021). The null hypothesis is that the data come from a population with independent and identically distributed realisations. The one-sided alternative hypothesis is that the chosen statistic is greater (or less) than under the null hypothesis. The different statistics are calculated according to:

If `statistic == "d"`,

$$\sum_{m=1}^{M} \sum_{t=1}^{T} \omega_t \left(I_{tm}^{(FU)} - I_{tm}^{(FL)} \right).$$

If `statistic == "D"`,

$$\sum_{m=1}^{M} \sum_{t=1}^{T} \omega_t \left(I_{tm}^{(FU)} - I_{tm}^{(FL)} - I_{tm}^{(BU)} + I_{tm}^{(BL)} \right).$$

If `statistic == "s"`,

$$\sum_{m=1}^{M} \sum_{t=1}^{T} \omega_t \left(I_{tm}^{(FU)} + I_{tm}^{(FL)} \right).$$

If `statistic == "S"`,

$$\sum_{m=1}^{M} \sum_{t=1}^{T} \omega_t \left(I_{tm}^{(FU)} + I_{tm}^{(FL)} + I_{tm}^{(BU)} + I_{tm}^{(BL)} \right).$$

If `statistic == "U"`,

$$\sum_{m=1}^{M} \sum_{t=1}^{T} \omega_t \left(I_{tm}^{(FU)} - I_{tm}^{(BU)} \right).$$

If `statistic == "L"`,

$$\sum_{m=1}^{M} \sum_{t=1}^{T} \omega_t \left(I_{tm}^{(BL)} - I_{tm}^{(FL)} \right).$$

If `statistic == "W"`,

$$\sum_{m=1}^{M}\sum_{t=1}^{T}\omega_t\left(I_{tm}^{(FU)}+I_{tm}^{(BL)}\right).$$

Where ω_t are weights given to the different records according to their position in the series, I_{tm} are the record indicators (see `I.record`), and (FU), (FL), (BU), and (BL) represent forward upper, forward lower, backward upper and backward lower records, respectively. The statistics d, D and W may be used for trend in location; s and S may be used for trend in variation; and U and L may be used for trend in the upper and lower tails of the distribution respectively.

The statistics, say X, are approximately normally distributed, with

$$Z=\frac{X-\mu}{\sigma},$$

while the mean μ of the particular statistic considered is simple to calculate, its variance σ^2 become a cumbersome expression and some are given by Diersen and Trenkler (2001) and all of them can be easily computed out of the expression of the covariances given by Cebrián, Castillo-Mateo and Asín (2021).

If `correct = TRUE`, then a continuity correction will be employed:

$$Z=\frac{X\pm 0.5-\mu}{\sigma},$$

with "$-$" if the alternative is greater and "$+$" if the alternative is less. Not recommended for the statistics with $\mu=0$.

When $M>1$, the expression of the variance under the null hypothesis can be substituted by the sample variance in the M series, $\hat{\sigma}^2$. In this case, the statistics are asymptotically t distributed, which is a more robust alternative against serial correlation.

If `simulate.p.value = TRUE`, the p-value is estimated by Monte Carlo simulations.

Value

A `"htest"` object with elements:

`statistic`	Value of the test statistic.
`parameter`	(If `distribution = "t"`) Degrees of freedom of the t statistic (equal to $M-1$).
`p.value`	P-value.
`alternative`	The alternative hypothesis.
`estimate`	(If `distribution = "normal"`) A vector with the value of the statistic, μ and σ^2.
`method`	A character string indicating the type of test performed.
`data.name`	A character string giving the name of the data.

Author(s)

Jorge Castillo-Mateo

References

Cebrián A, Castillo-Mateo J, Asín J (2021). "Record Tests to Detect Non Stationarity in the Tails with an Application to Climate Change." Available at Research Square doi: 10.21203/rs.3.rs214787/v1

Diersen J, Trenkler G (1996). "Records Tests for Trend in Location." *Statistics*, **28**(1), 1-12. doi: 10.1080/02331889708802543

Diersen J, Trenkler G (2001). "Weighted Records Tests for Splitted Series of Observations." In J Kunert, G Trenkler (eds.), *Mathematical Statistics with Applications in Biometry: Festschrift in Honour of Prof. Dr. Siegfried Schach*, pp. 163–178. Lohmar: Josef Eul Verlag.

Foster FG, Stuart A (1954). "Distribution-Free Tests in Time-Series Based on the Breaking of Records." *Journal of the Royal Statistical Society. Series B (Methodological)*, **16**(1), 1-22.

See Also

`foster.plot`, `N.plot`, `N.test`

Examples

```
# D-statistic
foster.test(ZaragozaSeries)
# D-statistic with linear weights
foster.test(ZaragozaSeries, weights = function(t) t-1)
# S-statistic with linear weights
foster.test(ZaragozaSeries, statistic = "S", weights = function(t) t-1)
# D-statistic with weights and t approach
foster.test(ZaragozaSeries, distribution = "t", weights = function(t) t-1)
# U-statistic with weights (upper tail)
foster.test(ZaragozaSeries, statistic = "U", weights = function(t) t-1)
# L-statistic with weights (lower tail)
foster.test(ZaragozaSeries, statistic = "L", weights = function(t) t-1)
```

`global.test` *Global Statistic for Two-Sided Tests*

Description

Performs a more powerful generalisation of the two-sided tests in this package by means of the sum of the statistics of upper and lower records in the forward and backward directions to study the hypothesis of the classical record model (i.e., of IID continuous RVs). The tests considered are the chi-square goodness-of-fit test `p.chisq.test`, the regression test `p.regression.test`, the likelihood-ratio test `lr.test`, and the score test `score.test`.

Usage

```
global.test(X, FUN, record = c(FU = 1, FL = 1, BU = 1, BL = 1), B = 1000, ...)
```

Arguments

X	A numeric vector, matrix (or data frame).
FUN	One of the functions whose statistic is going to be used. One of p.chisq.test, p.regression.test, lr.test or score.test.
record	Logical vector. Vector with four elements indicating if forward upper, forward lower, backward upper and backward lower are going to be shown, respectively. Logical values or 0,1 values are accepted.
B	An integer specifying the number of replicates used in the Monte Carlo approach.
...	Further arguments in the FUN function.

Details

The statistics, say X, of the tests p.chisq.test, p.regression.test, lr.test or score.test applied to the series of the forward upper, forward lower, backward upper and backward lower records are summed to develop a more powerful statistic:

$$X^{(FU)} + X^{(FL)} + X^{(BU)} + X^{(BL)}.$$

Other sums of statistics are allowed.

The distribution of this global statistics is unknown, but the p-value can be estimated with Monte Carlo simulations

Value

A list of class "htest" with the following elements:

statistic	Value of the statistic.
p.value	Simulated p-value.
method	A character string indicating the type of test.
data.name	A character string giving the name of the data.

Author(s)

Jorge Castillo-Mateo

See Also

p.chisq.test, p.regression.test, lr.test, score.test

Examples

```
# not run because the simulations take a while if B > 1000
## global statistic with 4 types of record for p.chisq.test
#global.test(ZaragozaSeries, FUN = p.chisq.test)
## global statistic with 4 types of record for p.regression.test
#global.test(ZaragozaSeries, FUN = p.regression.test)
## global statistic with 4 types of record for score.test with restricted alternative
```

```
#global.test(ZaragozaSeries, FUN = score.test, probabilities = "equal")
## global statistic with 4 types of record for lr.test with restricted alternative
#global.test(ZaragozaSeries, FUN = lr.test, probabilities = "equal")
## global statistic with 2 types of 'almost' independent records for lr.test
#global.test(ZaragozaSeries, FUN = lr.test, record = c(1,0,0,1), probabilities = "different")
```

`I.record` *Record Indicators*

Description

Returns the record indicators of the values in a vector. The record indicator for each value in a vector is a binary variable which takes the value 1 if the corresponding value in the vector is a record and 0 otherwise.

If the argument `X` is a matrix, then each column is treated as a different vector.

Usage

```
I.record(X, record = c("upper", "lower"), weak = FALSE)

## Default S3 method:
I.record(X, record = c("upper", "lower"), weak = FALSE)

## S3 method for class 'numeric'
I.record(X, record = c("upper", "lower"), weak = FALSE)

## S3 method for class 'matrix'
I.record(X, record = c("upper", "lower"), weak = FALSE)
```

Arguments

`X`	A numeric vector, matrix (or data frame).
`record`	A character string indicating the type of record to be calculated, "upper" or "lower".
`weak`	Logical. If `TRUE`, weak records are also counted. Default to `FALSE`.

Details

Let $\{X_1, \dots, X_T\}$ be a vector of random variables of size T. An observation X_t will be called an upper record value if its value exceeds that of all previous observations. An analogous definition deals with lower record values. Here, X_1 is referred to as the reference value or the trivial record. Then, the sequence of record indicator random variables $\{I_1, \dots, I_T\}$ is given by

$$I_t = \begin{cases} 1 & \text{if } X_t \text{ is a record,} \\ 0 & \text{if } X_t \text{ is not a record.} \end{cases}$$

The method `I.record` calculates the sample sequence above if the argument `X` is a numeric vector. If the argument `X` is a matrix (or data frame) with M columns, the method `I.record` calculates the sample sequence above for each column of the object as if all columns were different sequences.

In summary:

$$\texttt{I.record} : \texttt{X} = \begin{pmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,M} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,M} \\ \vdots & \vdots & & \vdots \\ X_{T,1} & X_{T,2} & \cdots & X_{T,M} \end{pmatrix} \longrightarrow \begin{pmatrix} I_{1,1} & I_{1,2} & \cdots & I_{1,M} \\ I_{2,1} & I_{2,2} & \cdots & I_{2,M} \\ \vdots & \vdots & & \vdots \\ I_{T,1} & I_{T,2} & \cdots & I_{T,M} \end{pmatrix}.$$

Indicators of record occurrence can be calculated for both upper and lower records.

All the procedure above can be extended to weak records, which also count the ties as a new (weak) record. Ties are possible in discrete variables or if a continuous variable has been rounded. Weak records can be computed if `weak = TRUE`.

`NA` values in `X` are assigned `-Inf` for upper records and `Inf` for lower records, so they are records only if they are placed at $t = 1$.

Value

A binary matrix of the same length or dimension as `X`, indicating the record occurrence.

Author(s)

Jorge Castillo-Mateo

References

Arnold BC, Balakrishnan N, Nagaraja HN (1998). *Records*. Wiley Series in Probability and Statistics. Wiley, New York.

See Also

`L.record`, `N.record`, `Nmean.record`, `p.record`, `R.record`, `records`, `S.record`

Examples

```
X <- c(1, 5, 3, 6, 6, 9, 2, 11, 17, 8)
I.record(X)
I.record(X, weak = TRUE)

I.record(ZaragozaSeries)
# record argument can be shortened
I.record(ZaragozaSeries, record = "l")
```

`L.plot` *Times of Record Plot*

Description

This function builds a ggplot object to display the upper and lower record times for both forward and backward directions.

Usage

```
L.plot(
  X,
  all = TRUE,
  record = c("upper", "lower"),
  point.col = "grey23",
  point.alpha = 0.8,
  line.col = "grey95"
)
```

Arguments

`X`	A numeric vector, matrix (or data frame).
`all`	Logical. If `TRUE` (the default) the four types of record are displayed.
`record`	If `all` = `FALSE`, a character string indicating the type of record to be calculated, "upper" or "lower".
`point.col, point.alpha`	Colour and transparency of the points.
`line.col`	Colour to plot lines.

Details

The function can be applied to plot the record times in a vector (if argument `X` is a vector) or to plot and compare the record times in a set of vectors (if argument `X` is a matrix). In the latter case, the approach to obtain the record times is applied to each column of the matrix.

If `all` = `TRUE`, a matrix of four panels is displayed for upper and lower records, and for the forward and backward (`series_rev`) directions. Otherwise, only one type of forward record is displayed.

An example of use of a plot with similar ideas is shown in Benestad (2004, Figures 3 and 8).

Value

A ggplot object.

Author(s)

Jorge Castillo-Mateo

References

Benestad RE (2004). “Record-Values, Nonstationarity Tests and Extreme Value Distributions.” *Global and Planetary Change*, **44**(1-4), 11-26.

See Also

`L.record`

Examples

```
Y <- c(1, 5, 3, 6, 6, 9, 2, 11, 17, 8)
L.plot(Y, all = FALSE)

L.plot(ZaragozaSeries, point.col = 1)
```

`L.record` *Record Times*

Description

Returns the record times of the values in a vector. The record times are the positions in a vector where a record occurs.

If the argument `X` is a matrix, then each column is treated as a different vector.

Usage

```
L.record(X, record = c("upper", "lower"), weak = FALSE)
```

Arguments

`X`	A numeric vector, matrix (or data frame).
`record`	A character string indicating the type of record to be calculated, "upper" or "lower".
`weak`	Logical. If `TRUE`, weak records are also counted. Default to `FALSE`.

Details

The sequence of record times $\{L_1, \ldots, L_I\}$ can be expressed in terms of the record indicator random variables `I.record` by

$$L_i = \min\{t \mid I_1 + I_2 + \ldots + I_t = i\}.$$

Record times can be calculated for both upper and lower records.

Value

If X is a vector, the function returns a column matrix containing the record times. If X is a matrix, the function returns a list where each element is a vector indicating the record times of the corresponding X column.

Author(s)

Jorge Castillo-Mateo

References

Arnold BC, Balakrishnan N, Nagaraja HN (1998). *Records*. Wiley Series in Probability and Statistics. Wiley, New York.

See Also

I.record, N.record, Nmean.record, p.record, R.record, records, S.record

Examples

```
Y1 <- c( 1,  5,  3,  6,  6,  9,  2)
Y2 <- c(10,  5,  3,  6,  6,  9,  2)
Y3 <- c( 5,  7,  3,  6, 19,  2, 20)
Y  <- cbind(Y1, Y2, Y3)

L.record(Y1)
L.record(Y)
```

---

lr.test — *Likelihood-Ratio Test for the Likelihood of the Record Indicators*

Description

This function performs likelihood-ratio tests for the likelihood of the record indicators I_t to study the hypothesis of the classical record model (i.e., of IID continuous RVs).

Usage

```
lr.test(
  X,
  record = c("upper", "lower"),
  alternative = c("two.sided", "greater", "less"),
  probabilities = c("different", "equal"),
  simulate.p.value = FALSE,
  B = 1000
)
```

Arguments

`X`	A numeric vector, matrix (or data frame).
`record`	A character string indicating the type of record, "upper" or "lower".
`alternative`	A character indicating the alternative hypothesis (`"two.sided"`, `"greater"` or `"less"`). Different statistics are used in the one-sided and two-sided alternatives (see Details).
`probabilities`	A character indicating if the alternative hypothesis assume all series with `"equal"` or `"different"` probabilities of record.
`simulate.p.value`	Logical. Indicates whether to compute p-values by Monte Carlo simulation.
`B`	An integer specifying the number of replicates used in the Monte Carlo estimation.

Details

The null hypothesis of the likelihood-ratio tests is that in every vector (columns of the matrix `X`), the probability of record at time t is $1/t$ as in the classical record model, and the alternative depends on the `alternative` and `probabilities` arguments. The probability at time t is any value, but equal in the M series if `probabilities = "equal"` or different in the M series if `probabilities = "different"`. The alternative hypothesis is more specific in the first case than in the second one. Furthermore, the `"two.sided"` `alternative` is tested with the usual likelihood ratio statistic, while the one-sided `alternatives` use specific statistics based on likelihoods (see Cebrián, Castillo-Mateo and Asín, 2021, for the details).

If `alternative = "two.sided"` & `probabilities = "equal"`, under the null, the likelihood ratio statistic has an asymptotic χ^2 distribution with $T-1$ degrees of freedom. It has been seen that for the approximation to be adequate M must be between 4 and 5 times greater than T. Otherwise, a `simulate.p.value` is recommended.

If `alternative = "two.sided"` & `probabilities = "different"`, the asymptotic behaviour is not fulfilled, but the Monte Carlo approach to simulate the p-value is applied. This statistic is the same as ℓ below multiplied by a factor of 2, so the p-value is the same.

If `alternative` is one-sided and `probabilities = "equal"`, the statistic of the test is

$$-2\sum_{t=2}^{T}\left\{-S_t\log\left(\frac{tS_t}{M}\right)+(M-S_t)\left(\log\left(1-\frac{1}{t}\right)-\log\left(1-\frac{S_t}{M}\right)I_{\{S_t<M\}}\right)\right\}I_{\{S_t>M/t\}}.$$

The p-value of this test is estimated with Monte Carlo simulations, because the computation of its exact distribution is very expensive.

If `alternative` is one-sided and `probabilities = "different"`, the statistic of the test is

$$\ell=\sum_{t=2}^{T}S_t\log(t-1)-M\log\left(1-\frac{1}{t}\right).$$

The p-value of this test is estimated with Monte Carlo simulations. However, it is equivalent to the statistic of the weighted number of records `N.test` with weights $\omega_t=\log(t-1)$ $(t=2,\ldots,T)$.

Value

A list of class "htest" with the following elements:

statistic	Value of the statistic.
parameter	Degrees of freedom of the approximate χ^2 distribution.
p.value	(Estimated) P-value.
method	A character string indicating the type of test.
data.name	A character string giving the name of the data.
alternative	A character string indicating the alternative hypothesis.

Author(s)

Jorge Castillo-Mateo

References

Cebrián A, Castillo-Mateo J, Asín J (2021). "Record Tests to Detect Non Stationarity in the Tails with an Application to Climate Change." Available at Research Square doi: 10.21203/rs.3.rs214787/v1

See Also

global.test, score.test

Examples

```
set.seed(23)
# two-sided and different probabilities of record, always simulated the p-value
lr.test(ZaragozaSeries, probabilities = "different")
# equal probabilities
lr.test(ZaragozaSeries, probabilities = "equal")
# equal probabilities with simulated p-value
lr.test(ZaragozaSeries, probabilities = "equal", simulate.p.value = TRUE)

# one-sided and different probabilities of record
lr.test(ZaragozaSeries, alternative = "greater", probabilities = "different")
# different probabilities with simulated p-value
lr.test(ZaragozaSeries, alternative = "greater", probabilities = "different",
  simulate.p.value = TRUE)
# equal probabilities, always simulated the p-value
lr.test(ZaragozaSeries, alternative = "greater", probabilities = "equal")
```

N.plot *Number of Records Plot*

Description

This function builds a ggplot object to compare the sample means of the (weighted) number of records in a vector up to time t, $\bar{N}^{\omega}_{t\cdot}$, and the expected values $\mathrm{E}(N^{\omega}_t)$ under the classical record model (i.e., of IID continuous RVs).

Usage

```
N.plot(
  X,
  weights = function(t) 1,
  record = c(FU = 1, FL = 1, BU = 1, BL = 1),
  backward = c("T", "t"),
  point.col = c(FU = "red", FL = "blue", BU = "red", BL = "blue"),
  point.shape = c(FU = 19, FL = 19, BU = 4, BL = 4),
  conf.int = TRUE,
  conf.level = 0.9,
  conf.aes = c("ribbon", "errorbar"),
  conf.col = "grey69"
)
```

Arguments

`X`	A numeric vector, matrix (or data frame).
`weights`	A function indicating the weight given to the different records according to their position in the series, e.g., if `function(t) t-1` then $\omega_t = t - 1$.
`record`	Logical vector. Vector with four elements indicating if forward upper, forward lower, backward upper and backward lower are going to be shown, respectively. Logical values or 0,1 values are accepted.
`backward`	A character string `"T"` or `"t"` indicating if the backward number of records shown are calculated up to time t in the backward series $\{X_T, \ldots, X_1\}$ or in the series $\{X_t, \ldots, X_1\}$. While the first option considers the evolution of a series of records observed up to time T, the second considers that until each time t the series has only been observed up to t.
`point.col, point.shape`	Vector with four elements indicating the colour and shape of the points. Every one of the four elements represents forward upper, forward lower, backward upper and backward lower, respectively.
`conf.int`	Logical. Indicates if the CIs are also shown.
`conf.level`	(If `conf.int == TRUE`) Confidence level of the CIs.
`conf.aes`	(If `conf.int == TRUE`) A character string indicating the aesthetic to display for the CIs, `"ribbon"` (grey area) or `"errorbar"` (vertical lines).
`conf.col`	Colour used to plot the expected value and (if `conf.int == TRUE`) CIs.

Details

This plot is associated to the test N.test. It calculates the sample means of the number of records in a set of vectors up to every time t (see Nmean.record). These sample means $\bar{N}^{\omega}_{t\cdot}$ are calculated from the sample of M values obtained from M vectors, the columns of matrix X. Then, these values are plotted and compared with the expected values $\mathrm{E}(N^{\omega}_{t})$ and their confidence intervals (CIs), under the hypothesis of the classical record model. The CIs of $\bar{N}^{\omega}_{t\cdot}$ uses the fact that, under the classical record model, the statistic is asymptotically Normal.

The plot can show the four types of record at the same time (i.e., forward upper, forward lower, backward upper and backward lower). In their interpretations one must be careful, for forward records each time t corresponds to the same year of observation, but for the backward series, time t corresponds to the year of observation $T - t + 1$ where T is the total number of observations in every series. Two types of backward records can be considered (see argument backward).

More details of this plot are shown in Cebrián, Castillo-Mateo, Asín (2021).

Value

A ggplot object.

Author(s)

Jorge Castillo-Mateo

References

Cebrián A, Castillo-Mateo J, Asín J (2021). “Record Tests to Detect Non Stationarity in the Tails with an Application to Climate Change.” Available at Research Square doi: 10.21203/rs.3.rs214787/v1

See Also

N.record, N.test, foster.test, foster.plot

Examples

```
# Plot at Zaragoza, with linear weights and error bar as CIs aesthetic
N.plot(ZaragozaSeries, weights = function(t) t-1, conf.aes = "errorbar")

# Plot only upper records
N.plot(ZaragozaSeries, record = c(1, 0, 1, 0))

# Change point colour and shape
Zplot <- N.plot(ZaragozaSeries,
  point.col = c("red", "red", "blue", "blue"),
  point.shape = c(19, 4, 19, 4))

## Not run: Load package ggplot2 to change the plot
#library("ggplot2")
## Remove legend
#Zplot + ggplot2::theme(legend.position = "none")
## Fancy axis
```

```
# Zplot +
#   ggplot2::scale_x_continuous(name = "Year (forward)",
#     breaks = c(10, 30, 50, 70),
#     labels=c("1960", "1980", "2000", "2020"),
#    sec.axis = ggplot2::sec_axis(~ nrow(ZaragozaSeries) - . + 1951, name = "Year (backward)")) +
#   ggplot2::theme(axis.title.x = ggplot2::element_text(colour = "red"),
#     axis.text.x = ggplot2::element_text(colour = "red"),
#     axis.title.x.top = ggplot2::element_text(colour = "blue"),
#     axis.text.x.top = ggplot2::element_text(colour = "blue"))
```

`N.record` *Number of Records*

Description

Returns the number of records up to time t of the values in a vector.

If the argument `X` is a matrix, then each column is treated as a different vector.

Usage

```
N.record(X, record = c("upper", "lower"), weak = FALSE)

Nmean.record(X, record = c("upper", "lower"), weak = FALSE)
```

Arguments

`X`	A numeric vector, matrix (or data frame).
`record`	A character string indicating the type of record to be calculated, "upper" or "lower".
`weak`	Logical. If `TRUE`, weak records are also counted. Default to `FALSE`.

Details

The record counting process $\{N_1, \ldots, N_T\}$ is defined by the number of records up to time t, and can be expressed in terms of the record indicator random variables `I.record` by

$$N_t = I_1 + I_2 + \ldots + I_t.$$

If `X` is a matrix with $M > 1$ columns, each column is treated as a vector and `Nmean.record` calculates for each t,

$$\bar{N}_t = \frac{N_{t1} + \ldots + N_{tM}}{M}.$$

In summary:

$$\texttt{N.record} : \texttt{X} = \left(\begin{array}{cccc} X_{1,1} & X_{1,2} & \cdots & X_{1,M} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,M} \\ \vdots & \vdots & & \vdots \\ X_{T,1} & X_{T,2} & \cdots & X_{T,M} \end{array} \right) \longrightarrow \left(\begin{array}{cccc} N_{1,1} & N_{1,2} & \cdots & N_{1,M} \\ N_{2,1} & N_{2,2} & \cdots & N_{2,M} \\ \vdots & \vdots & & \vdots \\ N_{T,1} & N_{T,2} & \cdots & N_{T,M} \end{array} \right)$$

and

$$\texttt{Nmean.record} : \texttt{X} \longrightarrow (\bar{N}_1, \bar{N}_2, \cdots, \bar{N}_T).$$

Number and mean number of records for both upper and lower records can be calculated.

Value

`N.record` returns a numeric matrix with the number of records up to each time (row) t for a vector or each column in `X`. `Nmean.record` returns a numeric vector with the mean number of records in M series (columns) up to each time (row) t.

Note

If `X` is a vector both functions return the same values, `N.record` as a matrix and `Nmean.record` as a vector.

Author(s)

Jorge Castillo-Mateo

References

Arnold BC, Balakrishnan N, Nagaraja HN (1998). *Records*. Wiley Series in Probability and Statistics. Wiley, New York.

See Also

`I.record`, `L.record`, `p.record`, `R.record`, `records`, `S.record`

Examples

```
Y1 <- c( 1,  5,  3,  6,  6,  9,  2)
Y2 <- c(10,  5,  3,  6,  6,  9,  2)
Y3 <- c( 5,  7,  3,  6, 19,  2, 20)
Y  <- cbind(Y1, Y2, Y3)

N.record(Y)
Nmean.record(Y)

N.record(ZaragozaSeries)
Nmean.record(ZaragozaSeries, record = 'l')
```

`N.test` *Number of Records Test*

Description

Performs tests based on the (weighted) number of records, N^ω. The hypothesis of the classical record model (i.e., of IID continuous RVs) is tested against the alternative hypothesis.

Usage

```
N.test(
  X,
  weights = function(t) 1,
  record = c("upper", "lower"),
  distribution = c("normal", "t", "poisson-binomial"),
  alternative = c("greater", "less"),
  correct = TRUE,
  method = c("mixed", "dft", "butler"),
  simulate.p.value = FALSE,
  B = 1000
)
```

Arguments

- `X` — A numeric vector, matrix (or data frame).
- `weights` — A function indicating the weight given to the different records according to their position in the series, e.g., if `function(t) t-1` then $\omega_t = t - 1$.
- `record` — A character string indicating the type of record to be calculated, `"upper"` or `"lower"`.
- `distribution` — A character string indicating the asymptotic distribution of the statistic, `"normal"` distribution, Student's `"t"`-distribution or exact `"poisson-binomial"` distribution.
- `alternative` — A character string indicating the type of alternative hypothesis, `"greater"` number of records or `"less"` number of records.
- `correct` — Logical. Indicates, whether a continuity correction should be done; defaults to `TRUE`. No correction is done if `simulate.p.value = TRUE` or `distribution = "poisson-binomial"`.
- `method` — (If `distribution = "poisson-binomial"`.) A character string that indicates the method by which the cdf of the Poisson binomial distribution is calculated and therefore the p-value. `"mixed"` is the preferred (and default) method, it is a more efficient combination of the later algorithms. `"dft"` uses the discrete Fourier transform which algorithm is given in Hong (2013). `"butler"` use the algorithm given by Butler and Stephens (2016).
- `simulate.p.value` — Logical. Indicates whether to compute p-values by Monte Carlo simulation. No simulation is done if `distribution = "poisson-binomial"`.

B If `simulate.p.value = TRUE`, an integer specifying the number of replicates used in the Monte Carlo estimation.

Details

The null hypothesis is that the data come from a population with independent and identically distributed continuous realisations. The one-sided alternative hypothesis is that the (weighted) number of records is greater (or less) than under the null hypothesis. The (weighted)-number-of-records statistic is calculated according to:

$$N^{\omega}_{\cdot\cdot} = \sum_{m=1}^{M} \sum_{t=1}^{T} \omega_t I_{tm},$$

where ω_t are weights given to the different records according to their position in the series and I_{tm} are the record indicators (see `I.record`).

The statistic $N^{\omega}_{\cdot\cdot}$ is exact Poisson binomial distributed when the ω_t's only take values in $\{0, 1\}$. In any case, it is also approximately normally distributed, with

$$Z = \frac{N^{\omega}_{\cdot\cdot} - \mu}{\sigma},$$

where its mean and variance are

$$\mu = M \sum_{t=1}^{T} \omega_t \frac{1}{t},$$

$$\sigma^2 = M \sum_{t=2}^{T} \omega_t^2 \frac{1}{t} \left(1 - \frac{1}{t}\right).$$

If `correct = TRUE`, then a continuity correction will be employed:

$$Z = \frac{N^{\omega}_{\cdot\cdot} \pm 0.5 - \mu}{\sigma},$$

with "$-$" if the alternative is greater and "$+$" if the alternative is less.

When $M > 1$, the expression of the variance under the null hypothesis can be substituted by the sample variance in the M series, $\hat{\sigma}^2$. In this case, the statistic $N^{\omega}_{S,\cdot\cdot}$ is asymptotically t distributed, which is a more robust alternative against serial correlation.

If `simulate.p.value = TRUE`, the p-value is estimated by Monte Carlo simulations.

The size of the tests is adequate for any values of T and M. Some comments and a power study are given by Cebrián, Castillo-Mateo and Asín (2021).

Value

A `"htest"` object with elements:

`statistic` Value of the test statistic.

`parameter` (If `distribution = "t"`.) Degrees of freedom of the t statistic (equal to $M-1$).

`p.value` P-value.

`alternative`	The alternative hypothesis.
`estimate`	(If `distribution = "normal"`) A vector with the value of $N^{\omega}_{\cdot\cdot}$, μ and σ^2.
`method`	A character string indicating the type of test performed.
`data.name`	A character string giving the name of the data.

Author(s)

Jorge Castillo-Mateo

References

Butler K, Stephens MA (2017). "The Distribution of a Sum of Independent Binomial Random Variables." *Methodology and Computing in Applied Probability*, **19**(2), 557-571. doi: 10.1007/s1100901695334

Cebrián A, Castillo-Mateo J and Asín J (2021). "Record Tests to Detect Non Stationarity in the Tails with an Application to Climate Change." Available at Research Square doi: 10.21203/rs.3.rs-214787/v1

Hong Y (2013). "On Computing the Distribution Function for the Poisson Binomial Distribution." *Computational Statistics & Data Analysis*, **59**(1), 41-51. doi: 10.1016/j.csda.2012.10.006

See Also

`N.record`, `N.plot`, `foster.test`, `foster.plot`, `brown.method`

Examples

```
# Forward Upper records
N.test(ZaragozaSeries)
# Forward Lower records
N.test(ZaragozaSeries, record = "lower", alternative = "less")
# Forward Upper records
N.test(series_rev(ZaragozaSeries), alternative = "less")
# Forward Upper records
N.test(series_rev(ZaragozaSeries), record = "lower")

# Exact test
N.test(ZaragozaSeries, distribution = "poisson-binom")
# Exact test for records in the last decade
N.test(ZaragozaSeries, weights = function(t) ifelse(t < 61, 0, 1), distribution = "poisson-binom")
# Linear weights for a more powerful test (without continuity correction)
N.test(ZaragozaSeries, weights = function(t) t-1, correct = FALSE)
```

Olympic_records_200m *200-Meter Olympic Records from 1900 to 2020*

Description

A data set containing the record times and record values of the 200-meter competition at the Olympic games, from 1900 to 2020. The variables are the following:

- year : Year of the record time
- time : Record time
- value : Record value in seconds

Usage

```
data(Olympic_records_200m)
```

Format

A data frame with 12 rows and 3 variables.

Note

In this data set, the interest lies in the lower records. Although the Olympic Games are held every 4 years, not all of these occasions have been held, so only the games that have taken place are considered in the definition of time.

Source

https://olympics.com/en/

See Also

series_record

p.chisq.test *Pearson's Chi-Square Test for Probabilities of Record*

Description

This function performs a chi-square goodness-of-fit test based on the record probabiliteis p_t to study the hypothesis of the classical record model (i.e., of IID continuous RVs).

Usage

```
p.chisq.test(
  X,
  record = c("upper", "lower"),
  simulate.p.value = FALSE,
  B = 1000
)
```

Arguments

`X` A numeric vector, matrix (or data frame).

`record` A character string indicating the type of record to be calculated, "upper" or "lower".

`simulate.p.value` Logical. Indicates whether to compute p-values by Monte Carlo simulation. It is recommended if the function returns a warning (see Details).

`B` If `simulate.p.value = TRUE`, an integer specifying the number of replicates used in the Monte Carlo estimation.

Details

The null hypothesis of this chi-square test is that in every vector (columns of the matrix `X`), the probability of record at time t is $1/t$ as in the classical record model, and the alternative that the probabilities are not equal to those values. First, the chi-square goodness-of-fit statistics to study the null hypothesis $H_0 : p_t = 1/t$ are calculated for each time $t = 2, \ldots, T$, where the observed value is the number of records at time t in the M vectors and the expected value under the null is M/t. The test statistic is the sum of the previous $T - 1$ statistics and its distribution under the null is approximately χ^2_{T-1}.

The chi-square approximation may not be valid with low M, since it requires expected values > 5 or up to 20% of the expected values are between 1 and 5. If this condition is not satisfied, a warning is displayed. In order to avoid this problem, a `simulate.p.value` can be made by means of Monte Carlo simulations.

Value

A `"htest"` object with elements:

`statistic` Value of the chi-squared statistic.

`df` Degrees of freedom.

`p.value` P-value.

`method` A character string indicating the type of test performed.

`data.name` A character string giving the name of the data.

Author(s)

Jorge Castillo-Mateo

References

Benestad RE (2003). "How Often Can We Expect a Record Event?" *Climate Research*, **25**(1), 3–13.

Benestad RE (2004). "Record-Values, Nonstationarity Tests and Extreme Value Distributions." *Global and Planetary Change*, **44**(1–4), 11–26.

See Also

`global.test`, `score.test`, `p.record`, `p.regression.test`, `lr.test`

Examples

```
# Warning, M = 76 small for the value of T = 70
p.chisq.test(ZaragozaSeries)
# Simulate p-value
p.chisq.test(ZaragozaSeries, simulate.p.value = TRUE, B = 10000)
```

`p.plot` *Probabilities of Record Plots*

Description

This function builds a ggplot object to display different functions of the record probabilities at time t, p_t. A graphical tool to study the hypothesis of the classical record model (i.e., of IID continuous RVs).

Usage

```
p.plot(
  X,
  plot = c("1", "2", "3"),
  record = c(FU = 1, FL = 1, BU = 1, BL = 1),
  point.col = c(FU = "red", FL = "blue", BU = "red", BL = "blue"),
  point.shape = c(FU = 19, FL = 19, BU = 4, BL = 4),
  conf.int = TRUE,
  conf.level = 0.9,
  conf.aes = c("ribbon", "errorbar"),
  conf.col = "grey69",
  smooth = TRUE,
  smooth.formula = y ~ x,
  smooth.method = stats::lm,
  smooth.weight = TRUE,
  smooth.linetype = c(FU = 1, FL = 1, BU = 2, BL = 2),
  ...
)
```

Arguments

`X`	A numeric vector, matrix (or data frame).
`plot`	One of the values "1", "2" or "3" (character or numeric class are both allowed). It determines the type of plot to be displayed (see Details).
`record`	Logical vector. Vector with four elements indicating if forward upper, forward lower, backward upper and backward lower are going to be shown, respectively. Logical values or 0,1 values are accepted.
`point.col, point.shape`	Vector with four elements indicating the colour and shape of the points. Every one of the four elements represents forward upper, forward lower, backward upper and backward lower, respectively.
`conf.int`	Logical. Indicates if the CIs are also shown.
`conf.level`	(If `conf.int == TRUE`) Confidence level of the CIs.
`conf.aes`	(If `conf.int == TRUE`) A character string indicating the aesthetic to display for the CIs, `"ribbon"` (grey area) or `"errorbar"` (vertical lines).
`conf.col`	Colour used to plot the expected value and (if `conf.int == TRUE`) CIs.
`smooth`	(If `plot = 1` or `3`) Logical. If `TRUE`, a smoothing in the probabilities is also plotted.
`smooth.formula`	(`smooth = TRUE`) `formula` to use in the smooth function, e.g., `y ~ x`, `y ~ poly(x,2,raw = TRUE)`, `y ~ log(x)`.
`smooth.method`	(If `smooth = TRUE`) Smoothing method (function) to use, e.g., `lm` or `loess`.
`smooth.weight`	(If `smooth = TRUE`) Logical. If `TRUE` (the default) the smoothing is estimated with weights.
`smooth.linetype`	(If `smooth = TRUE`) Vector with four elements indicating the line type of the smoothing. Every one of the four elements represents forward upper, forward lower, backward upper and backward lower, respectively.
`...`	Further arguments to pass through the smooth (see `ggplot2::geom_smooth`).

Details

Three different types of plots which aim to analyse the hypothesis of the classical record model using the record probabilities are implemented. Estimations of the record probabilities $\hat{p}_t$ used in the plots are obtained as the proportion of records at time t in M vectors (columns of matrix `X`) (see `p.record`).

Type 1 is the plot of the observed values $t\hat{p}_t$ versus time t (see `p.regression.test` for its associated test and details). The expected values under the classical record model are 1 for any value t, so that a cloud of points around 1 and with no trend should be expected. The estimated values are plotted, together with binomial confidence intervals (CIs). In addition, a smoothing function can be fitted to the cloud of points.

Type 2 is the plot of the estimated record probabilities p_t versus time t. The expected probabilities under the classical record model, $p_t = 1/t$, are also plotted, together with binomial CIs.

Type 3 is the same plot but on a logarithmic scale, so that the expected value is $-\log(t)$. In this case, another smoothing function can be fitted to the cloud of points.

Type 1 plot was proposed by Cebrián, Castillo-Mateo, Asín (2021), while type 2 and 3 appear in Benestad (2003, Figures 8 and 9, 2004, Figure 4).

Value

A ggplot object.

Author(s)

Jorge Castillo-Mateo

References

Benestad RE (2003). "How Often Can We Expect a Record Event?" *Climate Research*, **25**(1), 3–13.

Benestad RE (2004). "Record-Values, Nonstationarity Tests and Extreme Value Distributions." *Global and Planetary Change*, **44**(1–4), 11–26.

Cebrián A, Castillo-Mateo J and Asín J (2021). "Record Tests to Detect Non Stationarity in the Tails with an Application to Climate Change." Available at Research Square doi: 10.21203/rs.3.rs-214787/v1

See Also

`p.regression.test`

Examples

```
# three plots available
p.plot(ZaragozaSeries, plot = 1)
p.plot(ZaragozaSeries, plot = 2)
p.plot(ZaragozaSeries, plot = 3)

# Posible fits (plot 1):
#fit a line
p.plot(ZaragozaSeries, record = c(1,0,0,0))
# fit a second order polynomial
p.plot(ZaragozaSeries, record = c(1,0,0,0),
  smooth.formula = y ~ poly(x, degree = 2))
# force the line to pass by E(t*p_t) = 1 when t = 1, i.e., E(t*p_t) = 1 + beta_1 * (t-1)
p.plot(ZaragozaSeries, record = c(1,0,0,0),
  smooth.formula = y ~ I(x-1) - 1 + offset(rep(1, length(x))))
# force the second order polynomial pass by E(t*p_t) = 1 when t = 1
p.plot(ZaragozaSeries, record = c(1,0,0,0),
  smooth.formula = y ~ I(x-1) + I(x^2-1) - 1 + offset(rep(1, length(x))))
# fit a loess
p.plot(ZaragozaSeries, record = c(1,0,0,0),
  smooth.method = stats::loess, span = 0.25)
```

p.record *Probabilities of Record*

Description

`S.record` and `p.record` return the sample number of records and mean number of records at each time t in a set of M vectors (columns of `X`), respectively. In particular, `p.record` is the estimated record probability at each time t.

(For the introduccion to records see Details in `I.record`.)

Usage

```
p.record(X, record = c("upper", "lower"), weak = FALSE)

S.record(X, record = c("upper", "lower"), weak = FALSE)
```

Arguments

`X`	A numeric vector, matrix (or data frame).
`record`	A character string indicating the type of record to be calculated, "upper" or "lower".
`weak`	Logical. If `TRUE`, weak records are also counted. Default to `FALSE`.

Details

Given a matrix formed by M vectors (columns), measured at T times (rows), `M.record` calculates the number of records in the M vectors at each observed time t, S_t.

The function `p.record` is equivalent, but calculates the proportion of records at each time t, that is the ratio:

$$\hat{p}_t = \frac{S_t}{M} = \frac{I_{t,1} + \ldots + I_{t,M}}{M},$$

this proportion is an estimation of the probability of record at that time.

Following the notation in `I.record`, in summary:

$$\texttt{X} = \begin{pmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,M} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,M} \\ \vdots & \vdots & & \vdots \\ X_{T,1} & X_{T,2} & \cdots & X_{T,M} \end{pmatrix} \begin{matrix} \overset{\texttt{S.record}}{\longrightarrow} & \left(S_1, S_2, \cdots, S_T\right) \\ \\ \overset{\texttt{p.record}}{\longrightarrow} & \left(\hat{p}_1, \hat{p}_2, \cdots, \hat{p}_T\right) \end{matrix}$$

Summaries for both upper and lower records can be calculated.

Value

A vector with the number (or proportion in the case of `p.record`) of records at each time t (row).

Author(s)

Jorge Castillo-Mateo

References

Cebrián A, Castillo-Mateo J, Asín J (2021). "Record Tests to Detect Non Stationarity in the Tails with an Application to Climate Change." Available at Research Square doi: 10.21203/rs.3.rs214787/v1

See Also

`I.record`, `L.record`, `N.record`, `Nmean.record`, `R.record`, `records`

Examples

```
Y1 <- c( 1,  5,  3,  6,  6,  9,  2)
Y2 <- c(10,  5,  3,  6,  6,  9,  2)
Y3 <- c( 5,  7,  3,  6, 19,  2, 20)
Y  <- cbind(Y1, Y2, Y3)

S.record(Y)
p.record(Y)

S.record(ZaragozaSeries)
p.record(ZaragozaSeries, record = "l")
```

`p.regression.test` *Probabilities of Record Regression Test*

Description

This function performs a linear hypothesis test based on a regression for the record probabilities p_t to study the hypothesis of the classical record model (i.e., of IID continuous RVs).

Usage

```
p.regression.test(
  X,
  record = c("upper", "lower"),
  formula = y ~ x,
  simulate.p.value = FALSE,
  B = 1000
)
```

Arguments

X
: A numeric vector, matrix (or data frame).

record
: A character string indicating the type of records to be calculated, "upper" or "lower".

formula
: "formula" to use in lm function, e.g., `y ~ x`, `y ~ poly(x,2,raw = TRUE)`, `y ~ log(x)`. By default `formula = y ~ x`. See Note for a caveat.

simulate.p.value
: Logical. Indicates whether to compute p-values by Monte Carlo simulation. It is recommended if the number of columns of `X` (i.e., the number of series) is lower than 12, since for lower values the size of the test is not fulfilled.

B
: If `simulate.p.value = TRUE`, an integer specifying the number of replicates used in the Monte Carlo estimation.

Details

The null hypothesis is that the data come from a population with independent and identically distributed realisations. This implies that in all the vectors (columns in matrix `X`), the sample probability of record at time t (`p.record`) is $1/t$, so that

$$t\,\mathrm{E}(\hat{p}_t) = 1.$$

Then,

$$H_0 : \, p_t = 1/t, \, t = 2, ..., T \iff H_0 : \, \beta_0 = 1, \, \beta_1 = 0,$$

where β_0 and β_1 are the coefficients of the regression model

$$t\,\mathrm{E}(\hat{p}_t) = \beta_0 + \beta_1 t.$$

The model has to be estimated by weighted least squares since the response is heteroskedastic.

Other models can be considered with the `formula` argument. However, for the test to be correct, the model must leave the intercept free or fix it to 1 (see Examples for possible models).

The F statistic is computed for carrying out a comparison between the restricted model under the null hypothesis and the more general model (e.g., the alterantive hypothesis where $t\,\mathrm{E}(\hat{p}_t)$ is a linear function of time t). This alternative hypothesis may be reasonable in many real examples, but not always.

If the sample size (i.e., the number of series or columns of `X`) is lower than 8 or 12 the `simulate.p.value` option is recommended.

Value

A `"htest"` object with elements:

null.value
: Value of the coefficients under the null hypothesis when more than one coefficient is fitted.

alternative
: Character string indicating the type of alternative hypothesis.

method
: A character string indicating the type of test performed.

estimate
: Value of the fitted coefficients.

`data.name`	A character string giving the name of the data.
`statistic`	Value of the F statistic.
`parameters`	Degrees of freedom of the F statistic.
`p.value`	P-value.

Note

IMPORTANT: In `formula` the intercept has to be free or fixed to 1 so that the test is correct.

Author(s)

Jorge Castillo-Mateo

See Also

`p.chisq.test`, `p.plot`

Examples

```
# Simple test for upper records (p-value = 0.01202)
p.regression.test(ZaragozaSeries)
# Simple test for lower records (p-value = 0.006175)
p.regression.test(ZaragozaSeries, record = "lower")

# Fit a 2nd term polynomial for upper records (p-value = 0.0003933)
p.regression.test(ZaragozaSeries, formula = y ~ I(x^2))
# Fit a 2nd term polynomial for lower records (p-value = 0.005108)
p.regression.test(ZaragozaSeries, record = "lower", formula = y ~ I(x^2))

# Fix the intercept to 1 for upper records (p-value = 0.01416)
p.regression.test(ZaragozaSeries, formula = y ~ I(x-1) - 1 + offset(rep(1, length(x))))
# Fix the intercept to 1 for lower records (p-value = 0.00138)
p.regression.test(ZaragozaSeries, record = "lower",
  formula = y ~ I(x-1) - 1 + offset(rep(1, length(x))))

# Simulate p-value when the number of series is small
TxZ <- apply(series_split(TX_Zaragoza$TX), 1, max, na.rm = TRUE)
p.regression.test(TxZ, simulate.p.value = TRUE)
```

`Poisson-Binomial` *The Poisson Binomial Distribution*

Description

Density, distribution function, quantile function and random generation for the Poisson binomial distribution with parameters `size` and `prob`.

This is conventionally interpreted as the number of successes in `size * length(prob)` trials with success probabilities `prob`.

Usage

```
dpoisbinom(x, size = 1, prob, log = FALSE)

ppoisbinom(q, size = 1, prob, lower.tail = TRUE, log.p = FALSE)

qpoisbinom(p, size = 1, prob, lower.tail = TRUE, log.p = FALSE)

rpoisbinom(n, size = 1, prob)
```

Arguments

`x, q`	Vector of quantiles.
`size`	The Poisson binomial distribution has `size` times the vector of probabilities `prob`.
`prob`	Vector with the probabilities of success on each trial.
`log, log.p`	Logical. If `TRUE`, probabilities p are given as $\log(p)$.
`lower.tail`	Logical. If `TRUE` (default), probabilities are $P(X \leq x)$, otherwise, $P(X > x)$.
`p`	Vector of probabilities.
`n`	Number of observations.

Details

The Poisson binomial distribution with `size = 1` and `prob` $= (p_1, p_2, \ldots, p_n)$ has density

$$p(x) = \sum_{A \in F_x} \prod_{i \in A} p_i \prod_{j \in A^c} (1 - p_j)$$

for $x = 0, 1, \ldots, n$; where F_x is the set of all subsets of x integers that can be selected from $\{1, 2, \ldots, n\}$.

$p(x)$ is computed using Hong (2013) algorithm, see the reference below.

The quantile is defined as the smallest value x such that $F(x) \geq p$, where F is the cumulative distribution function.

Value

`dpoisbinom` gives the density, `ppoisbinom` gives the distribution function, `qpoisbinom` gives the quantile function and `rpoisbinom` generates random deviates.

The length of the result is determined by `x`, `q`, `p` or `n`.

Author(s)

Jorge Castillo-Mateo

References

Hong Y (2013). "On Computing the Distribution Function for the Poisson Binomial Distribution." *Computational Statistics & Data Analysis*, **59**(1), 41-51.

`R.record` *Record Values*

Description

Returns the record values of the values in a vector. A record value is the magnitude of a record observation.

If the argument `X` is a matrix, then each column is treated as a different vector.

Usage

```
R.record(X, record = c("upper", "lower"), weak = FALSE)
```

Arguments

`X`	A numeric vector, matrix (or data frame).
`record`	A character string indicating the type of record to be calculated, "upper" or "lower".
`weak`	Logical. If `TRUE`, weak records are also counted. Default to `FALSE`.

Details

The sequence of record values $\{R_1, \ldots, R_I\}$ can be expressed in terms of the record times `L.record` by

$$R_i = X_{L_i}.$$

Record values can be calculated for both upper and lower records.

Value

If `X` is a vector, the function returns a column matrix containing the record values. If `X` is a matrix, the function returns a list where each element is a vector indicating the record values of the corresponding `X` column.

Author(s)

Jorge Castillo-Mateo

References

Arnold BC, Balakrishnan N, Nagaraja HN (1998). *Records*. Wiley Series in Probability and Statistics. Wiley, New York.

See Also

`I.record`, `L.record`, `N.record`, `Nmean.record`, `p.record`, `R.record`, `records`, `S.record`

Examples

```
Y1 <- c( 1,  5,  3,  6,  6,  9,  2)
Y2 <- c(10,  5,  3,  6,  6,  9,  2)
Y3 <- c( 5,  7,  3,  6, 19,  2, 20)
Y  <- cbind(Y1, Y2, Y3)

R.record(Y1)
R.record(Y)
```

---

rcrm *The Classical Record Model*

Description

Random generation for the classical record model, i.e., sequences of independent and identically distributed (IID) continuous random variables (RVs).

Usage

```
rcrm(Trows = 50, Mcols = 100, rdist = stats::rnorm, ...)
```

Arguments

`Trows, Mcols`	Integers indicating the number of rows and columns of the returned matrix, i.e., the length and number of series for the record analysis.
`rdist`	A function that simulates continuous random variables, e.g., `runif` (fastest in `stats` package), `rnorm` or `rexp`.
`...`	Further arguments to introduce in the `rdist` function.

Value

A matrix of draws of IID continuous RVs with common distribution `rdist`.

Author(s)

Jorge Castillo-Mateo

References

Arnold BC, Balakrishnan N, Nagaraja HN (1998). *Records*. Wiley Series in Probability and Statistics. Wiley, New York.

See Also

`L.record`, `S.record`, `N.record`, `Nmean.record`, `p.record`, `records`

Examples

```
# By default, draw a sample of 100 series of length 50
# with observations coming from a standard normal distribution
X <- rcrm()
# Compute its record indicators
I <- I.record(X)
# Implement some tests
N.test(X, distribution = "poisson-binomial")
foster.test(X, weights = function(t) t-1, statistic = "D")
```

---

records *Record Values and Record Times*

Description

This function identifies (and plots if argument `plot = TRUE`) the record values (R_i), and the record times (L_i) in a vector, for all upper and lower records in forward and backward directions.

Usage

```
records(
  X,
  plot = TRUE,
  direction = c("forward", "backward", "both"),
  variable,
  type = c("lines", "points"),
  col = c(T = "black", U = "salmon", L = "skyblue", O = "black"),
  alpha = c(T = 1, U = 1, L = 1, O = 1),
  shape = c(F = 19, B = 4, O = 19),
  linetype = c(F = 1, B = 2)
)
```

Arguments

`X`	A numeric vector.
`plot`	Logical. If `TRUE` (the default) the records are plotted.
`direction`	A character string indicating the type of record to show in the plot if `plot == TRUE`: `"forward"`, `"backward"` or `"both"` (see Details).
`variable`	Optional. A vector, containing other variable related to `X` and measured at the same times. Only used if `plot = FALSE`.
`type`	Character string indicating if `X` is shown with `"lines"` or `"points"`.
`col, alpha`	Character and numeric vectors of length four, respectively. These arguments represent respectively the colour and transparency of the points or lines: trivial record, upper records, lower records and observations respectively. Vector names in the default are only indicative.

shape — If `type == "points"`. Integer vector of length 3 indicating the shape of the points for forward records, backward records and observations. Vector names in the default are only indicative.

linetype — Integer vector of length 2 indicating the line type of the step functions in the forward and backward records, respectively. Vector names in the default are only indicative.

Details

Customarily, the records in a time series (X_t) observed in T instances $t = 1, 2, \ldots, T$ can be obtained using chronological order. Besides, we could also compute the records in similar sequences of random variables if we consider reversed chronological order starting from the last observation, i.e., $t' = T, \ldots, 2, 1$. The analysis of series with reversed order is customarily referred to as backward, as opposed to a forward analysis.

Value

If `plot = TRUE` a ggplot object, otherwise a list with four data frames where the first column are the record times, the second the record values and, if `variable` is not null, the third column are their values at the record times, respectively for upper and lower records in forward and backward series.

Author(s)

Jorge Castillo-Mateo

See Also

I.record, series_double, series_rev, series_split, series_uncor, series_untie

Examples

```
Y <- c(5, 7, 3, 6, 19, 2, 20)
records(Y, plot = FALSE, variable = seq_along(Y))

# Show the whole series and its upper and lower records
records(TX_Zaragoza$TX)
# Compute tables for the whole series
TxZ.record <- records(TX_Zaragoza$TX, plot = FALSE, variable = TX_Zaragoza$DATE)
TxZ.record
names(TxZ.record)
# To show the Forward Upper records
TxZ.record[[1]]
plot(TxZ.record[[1]]$Times, TxZ.record[[1]]$Values)

# Annual maximum daily maximum temperatures
TxZ <- apply(series_split(TX_Zaragoza$TX), 1, max)
# Plot for the records in forward and backward directions
records(TxZ, direction = "both")
# Compute tables for the annual maximum
records(TxZ, plot = FALSE, variable = 1951:2020)
```

`score.test` *Score Test for the Likelihood of the Record Indicators*

Description

This function performs score (or Lagrange multiplier) tests for the likelihood of the record indicators I_t to study the hypothesis of the classical record model (i.e., of IID continuous RVs).

Usage

```
score.test(
  X,
  record = c("upper", "lower"),
  alternative = c("two.sided", "greater", "less"),
  probabilities = c("different", "equal"),
  simulate.p.value = FALSE,
  B = 1000
)
```

Arguments

`X`	A numeric vector, matrix (or data frame).
`record`	A character string indicating the type of record, "upper" or "lower".
`alternative`	A character indicating the alternative hypothesis (`"two.sided"`, `"greater"` or `"less"`). Different statistics are used in the one-sided and two-sided alternatives (see Details).
`probabilities`	A character indicating if the alternative hypothesis assume all series with `"equal"` or `"different"` probabilities of record.
`simulate.p.value`	Logical. Indicates whether to compute p-values by Monte Carlo simulation.
`B`	An integer specifying the number of replicates used in the Monte Carlo estimation.

Details

The null hypothesis of the score tests is that in every vector (columns of the matrix `X`), the probability of record at time t is $1/t$ as in the classical record model, and the alternative depends on the `alternative` and `probabilities` arguments. The probability at time t is any value, but equal in the M series if `probabilities = "equal"` or different in the M series if `probabilities = "different"`. The alternative hypothesis is more specific in the first case than in the second one. Furthermore, the `"two.sided"` `alternative` is tested with the usual Lagrange multiplier statistic, while the one-sided `alternatives` use specific statistics based on scores. (See Cebrián, Castillo-Mateo and Asín (2021) for details on these tests.)

If `alternative = "two.sided" & probabilities = "equal"`, under the null, the Lagrange multiplier statistic has an asymptotic χ^2 distribution with $T-1$ degrees of freedom. It has been seen that for the approximation to be adequate M should be greater than T. Otherwise, a `simulate.p.value` can be computed.

If `alternative = "two.sided"` & `probabilities = "different"`, the asymptotic behaviour of the Lagrange multiplier statistic is not fulfilled, but the Monte Carlo approach to simulate the p-value is applied.

If `alternative` is one-sided and `probabilities = "equal"`, the statistic of the test is

$$\mathcal{T} = \sum_{t=2}^{T} \frac{(tS_t - M)^2}{M(t-1)} I_{\{S_t > M/t\}}.$$

The p-value of this test is estimated with Monte Carlo simulations, since the compute the exact distribution of $\mathcal{T}$ is very expensive.

If `alternative` is one-sided and `probabilities = "different"`, the statistic of the test is

$$\mathcal{S} = \frac{\sum_{t=2}^{T} t(tS_t - M)/(t-1)}{\sqrt{M \sum_{t=2}^{T} t^2/(t-1)}},$$

which is asymptotically standard normal distributed in M. It is equivalent to the statistic of the weighted number of records `N.test` with weights $\omega_t = t^2/(t-1)$ $(t = 2, \ldots, T)$.

Value

A list of class `"htest"` with the following elements:

`statistic`	Value of the statistic.
`parameter`	Degrees of freedom of the approximate χ^2 distribution.
`p.value`	P-value.
`method`	A character string indicating the type of test.
`data.name`	A character string giving the name of the data.
`alternative`	A character string indicating the alternative hypothesis.

Author(s)

Jorge Castillo-Mateo

References

Cebrián A, Castillo-Mateo J, Asín J (2021). “Record Tests to Detect Non Stationarity in the Tails with an Application to Climate Change.” Available at Research Square doi: 10.21203/rs.3.rs214787/v1

See Also

`lr.test`, `global.test`

Examples

```
set.seed(23)
# two-sided and different probabilities of record, always simulated the p-value
score.test(ZaragozaSeries, probabilities = "different")
# equal probabilities
score.test(ZaragozaSeries, probabilities = "equal")
# equal probabilities with simulated p-value
score.test(ZaragozaSeries, probabilities = "equal", simulate.p.value = TRUE)

# one-sided and different probabilities of record
score.test(ZaragozaSeries, alternative = "greater", probabilities = "different")
# different probabilities with simulated p-value
score.test(ZaragozaSeries, alternative = "greater", probabilities = "different",
  simulate.p.value = TRUE)
# equal probabilities, always simulated the p-value
score.test(ZaragozaSeries, alternative = "greater", probabilities = "equal")
```

`series_double` *Double the Number of Series*

Description

This function changes the format of a matrix transforming a $T \times M$ matrix in a $\lfloor T/\mathsf{k} \rfloor \times \mathsf{k}\, M$ matrix in the following way.

First, the matrix is divided into `k` matrices $\lfloor T/\mathsf{k} \rfloor \times M$, containing the rows whose remainder of the division of the row number by `k` is $1, 2, \ldots, \mathsf{k}-1, 0$, respectively; and secondly those matrices are `cbind`ed.

Usage

```
series_double(X, k = 2)
```

Arguments

`X`	A numeric vector, matrix (or data frame).
`k`	Integer > 1, times to increase the number of columns.

Details

This function is used in the data preparation (or pre-processing) often required to apply the exploratory and inference tools based on theory of records within this package.

Most of the record inference tools require a high number of independent series M (number of columns) to be applied. If M is low and the time period of observation, T, is high enough, the following procedure can be applied in order to multiply by `k` the value M. The approach consists of considering that the observations at two (or more) consecutive times, t and $t+1$ (or $t+\mathsf{k}-1$), are independent observations measured at the same time unit. That means that we are doubling (or multiplying by k) the original time unit of the records, so that the length of the observation period will be $\lfloor T/\text{k} \rfloor$. This function rearranges the original data matrix into the new format.

If the number of rows of the original matrix is not divisible by k, the first nrow(X) %% k rows are deleted.

Value

A $\lfloor T/\text{k} \rfloor \times \text{k}\, M$ matrix.

Author(s)

Jorge Castillo-Mateo

See Also

series_record, series_rev, series_split, series_ties series_uncor, series_untie

Examples

```
series_double(matrix(1:100, 10, 10))

series_double(ZaragozaSeries, k = 4)
```

---

series_record *From Record Times to Time Series*

Description

If only the record times are available (upper or lower, or both) and not the complete series, series_record builds a complete series with the same record occurrence as specified in the arguments. This function is useful to apply the plots and tests within RecordTest-package to a vector of record times.

Usage

```
series_record(L_upper, R_upper, L_lower, R_lower, Trows = NA)
```

Arguments

L_upper, L_lower
: A vector of (increasing) integers denoting the upper or/and lower record times.

R_upper, R_lower
: (Optional) A vector of (increasing/decreasing) values denoting the upper or/and lower record values.

Trows
: Integer indicating the actual length of the series. If it is not specified, then the length of the series is assumed equal to the last record occurrence.

Value

A vector of length `Trows` with `L_upper` upper or/and `L_lower` lower record times and `R_upper` upper or/and `R_lower` lower record values.

Note

Remember that the first observation in a series is always a record time.

Author(s)

Jorge Castillo-Mateo

See Also

`series_double`, `series_rev`, `series_split`, `series_ties`, `series_uncor`, `series_untie`

Examples

```
# upper record times observed in a 100 length time series
L <- c(1, 4, 14, 40, 45, 90)
X <- series_record(L_upper = L, Trows = 100)

# now you can apply plots and tests for upper records to the X series
#N.plot(X)
#N_normal.test(X)

# if you also have lower record times
L_lower <- c(1, 2, 12, 56, 57, 78, 91)
X <- series_record(L_upper = L, L_lower = L_lower, Trows = 100)

# now you can apply plots and tests to the X series with both types of record times
#foster.plot(X, statistic = 'd')
#foster.test(X, statistic = 'd')

# apply to the 200-meter Olympic records from 1900 to 2020
or200m <- series_record(L_lower = Olympic_records_200m$t,
                        R_lower = Olympic_records_200m$value,
                        Trows = 27)
# some plots and tests
N.plot(or200m, record = c(0,1,0,0))
N.test(or200m, record = "lower", distribution = "poisson-binomial")
```

---

series_rev *Reverse Elements by Columns*

Description

Result of applying `rev` function by columns to the matrix. This allows the study of the series backwards and not only forward.

Usage

```
series_rev(X)
```

Arguments

X	A numeric vector, matrix (or data frame).

Author(s)

Jorge Castillo-Mateo

See Also

series_double, series_record, series_split, series_ties, series_uncor, series_untie

Examples

```
series_rev(matrix(1:100, 10, 10))

series_rev(ZaragozaSeries)
```

---

series_split *Splitted Series*

Description

The vector X of length T is broken into Mcols blocks, each part containing $T/$Mcols elements.

If the vector X represents consecutive daily values in a year, then Mcols = 365 is preferred. This function rearranges X into a matrix format, where each column is the vector of values at the same day of the year. For monthly data in a year, Mcols = 12 should be used.

Usage

```
series_split(X, Mcols = 365)
```

Arguments

X	A numeric vector.
Mcols	An integer number, giving the number of columns in the final matrix.

Details

This function is used in the data preparation (or pre-processing) often required to apply the exploratory and inference tools based on theory of records within this package when the time series presents seasonality.

This function transforms a vector into a matrix, applying the following procedure: the first row of the matrix is built of the first `Mcols` elements of the vector, the second row by the `Mcols` following elements, and so on. The length of the vector must be a multiple of `Mcols` (see Note otherwise).

In the case of a vector of daily values, `Mcols` is usually 365, so that the first column corresponds to all the values observed at the 1st of January, the second to the 2nd of January, etc.

If $X_{t,m}$ represents the value in day m of year t, then if

$$\texttt{X} = (X_{1,1}, X_{1,2}, \ldots, X_{1,365}, X_{2,1}, X_{2,2}, \ldots, X_{T,365}),$$

applying `series_split` to `X` returns the following matrix:

$$\begin{pmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,365} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,365} \\ \vdots & \vdots & & \vdots \\ X_{T,1} & X_{T,2} & \cdots & X_{T,365} \end{pmatrix}_{T \times 365}.$$

Value

A matrix with `Mcols` columns.

Note

`series_double` can be implemented for the same purpose as this function but without requiring that the length of `X` be divisible by `Mcols`. It removes the first elements of `X` until its length is divisible by `Mcols`.

Author(s)

Jorge Castillo-Mateo

References

Cebrián A, Castillo-Mateo J, Asín J (2021). “Record Tests to Detect Non Stationarity in the Tails with an Application to Climate Change.” Available at Research Square doi: 10.21203/rs.3.rs214787/v1

See Also

`series_double`, `series_record`, `series_rev`, `series_ties`, `series_uncor`, `series_untie`

Examples

```
series_split(1:100, Mcols = 10)

TxZ <- series_split(TX_Zaragoza$TX)
dim(TxZ)
```

`series_ties` *Summary of Record Ties*

Description

This function compares the number of strog and weak records to quantify whether rounding effects could greatly skew the conclusions.

Usage

```
series_ties(X, record = c("upper", "lower"))
```

Arguments

`X`	A numeric vector, matrix (or data frame).
`record`	A character string indicating the type of record to be assessed, "upper" or "lower".

Details

This function is used in the data preparation (or pre-processing) often required to apply the exploratory and inference tools based on theory of records within this package.

The theory of records on which the hypothesis tests are based assumes that the random variables are continuous, proving that the probability that two observations take the same value is zero. Most of the data collected is rounded, giving a certain probability to the tie between records, thereby reducing the number of new records(see, e.g., Wergen et al. 2012).

This function summarises the difference between the number of observed strong records and the weak records.

Value

A `list` object with elements:

`number`	Number of records: A vector containing the observed total, strong and weak number of records and the expected under IID.
`percentage`	% of weak records: Percentage of weak records within the total.
`percentage.position`	% of weak records by position: A vector with the percentage of weak records with `names` corresponding to its observed instant.

Author(s)

Jorge Castillo-Mateo

References

Wergen G, Volovik D, Redner S, Krug J (2012). "Rounding Effects in Record Statistics." *Physical Review Letters*, **109**(16), 164102.

See Also

series_double, series_record, series_rev, series_split, series_uncor, series_untie

Examples

```
series_ties(ZaragozaSeries)
```

---

series_uncor *Subset of Uncorrelated Series*

Description

Given a matrix this function extracts a subset of uncorrelated columns (see Details).

Usage

```
series_uncor(
  X,
  return.value = c("series", "indexes"),
  type = c("adjacent", "all"),
  first.last = TRUE,
  m = 1,
  alpha = 0.05,
  ...
)
```

Arguments

X	A numeric matrix (or data frame) where the uncorrelated vectors are extracted from.
return.value	A character string indicating the return of the function, "series" for a matrix with uncorrelated columns or "indexes" for a vector with the position of the uncorrelated columns in X.
type	A character string indicating the type of uncorrelation wanted between the extracted series (or columns), "adjacent" or "all" (see Details).
first.last	Logical. Indicates if the first and last columns have also to be uncorrelated (when type = "adjacent").

m	Integer value giving the starting column.
alpha	Numeric value in $(0, 1)$. It gives the significance level of the correlation test where alternative hypothesis is that the true correlation is not equal to 0.
...	Further arguments to be passed to `cor.test` function (see `cor.test` for possible arguments).

Details

This function is used in the data preparation (or pre-processing) often required to apply the exploratory and inference tools based on theory of records within this package.

Given a matrix `X` considered as a set of M^* vectors, which are the columns of `X`, this function extracts the biggest subset of uncorrelated vectors (columns), using the following procedure: starting from column `m`, the test `cor.test` is applied to study the correlation between columns depending on argument `type`.

If `type = "adjacent"`, the test is computed between `m` and $\mathtt{m} + 1, \mathtt{m} + 2, \ldots$ and so on up to find a column $\mathtt{m} + k$ which is not significantly correlated with column `m`. Then, the process is repeated starting at column $\mathtt{m} + k$. All columns are checked.

When the first and last columns may not have a significant correlation, where `m` is the first column, the parameter `first.last` should be `FALSE`. When the first and last columns could be correlated, the function requires `first.last = TRUE`.

If `type = "all"`, the procedure is similar as above but the new kept column cannot be significant correlated with any other column already kept, not only the previous one. So this option results in a fewer number of columns.

Value

A matrix or a vector as specified by `return.value`.

Author(s)

Jorge Castillo-Mateo

References

Cebrián A, Castillo-Mateo J, Asín J (2021). "Record Tests to Detect Non Stationarity in the Tails with an Application to Climate Change." Available at Research Square doi: 10.21203/rs.3.rs214787/v1

See Also

`series_double`, `series_record`, `series_rev`, `series_split`, `series_ties`, `series_untie`

Examples

```
# Split Zaragoza series
TxZ <- series_split(TX_Zaragoza$TX)

# Index of uncorrelated columns depending on the criteria
```

```
series_uncor(TxZ, return.value = "indexes", type = "adjacent")
series_uncor(TxZ, return.value = "indexes", type = "all")

# Return the set of uncorrelated vectors
ZaragozaSeries <- series_uncor(TxZ)
```

---

series_untie *Breaking Record Ties*

Description

Breaks record ties when observations have been rounded.

Usage

```
series_untie(X)
```

Arguments

X	A numeric vector, matrix (or data frame).

Details

This function is used in the data preparation (or pre-processing) often required to apply the exploratory and inference tools based on theory of records within this package.

The theory of records on which the hypothesis tests are based assumes that the random variables are continuous, so that the probability that two observations take the same value is zero. Most of the data collected is rounded, giving a certain probability to the tie between records, thereby reducing the number of new records (see, e.g., Wergen et al. 2012).

This function breaks ties by adding a uniform random variable to each element of X. The function computes the maximum number of decimal digits (let it be n) for any element in X. Then the uniform variable is sampled in the interval $(-5 \times 10^{-(n+1)}, 5 \times 10^{-(n+1)})$, so the records in the original (rounded) series are also records in the new series.

Value

A matrix equal to X whose elements have been added a sample from a uniform variable, different for each element.

Author(s)

Jorge Castillo-Mateo

References

Wergen G, Volovik D, Redner S, Krug J (2012). “Rounding Effects in Record Statistics.” *Physical Review Letters*, **109**(16), 164102.

See Also

series_double, series_record, series_rev, series_split, series_ties, series_uncor

Examples

```
set.seed(23)
X <- matrix(round(stats::rnorm(100), digits = 1), nrow = 10, ncol = 10)
series_untie(X)

series_untie(ZaragozaSeries)
```

TX_Zaragoza — *Time Series of Daily Maximum Temperature at Zaragoza (Spain)*

Description

A dataset containing the series of daily maximum temperature at Zaragoza aeropuerto (Spain), from 01/01/1951 to 31/12/2020. Zaragoza is located at the north-east (+41:39:42 N, -001:00:29 W) of Iberian Peninsula at 247 m above mean sea level. This series is obtained from the ECA&D series but it has been transformed, by removing days February 29th. The series has three NA missing observations corresponding to 31/03/1951, 04/01/1965, and 05/10/1965. The variables are the following:

- STAID : Station identifier
- SOUID : Source identifier
- DATE : Date YYYYMMDD
- TX : Maximum temperature in 0.1ºC
- Q_TX : quality code for TX (0='valid'; 1='suspect'; 9='missing')

Usage

```
data(TX_Zaragoza)
```

Format

A data frame with 25550 rows and 5 variables.

Source

EUROPEAN CLIMATE ASSESSMENT & DATASET (ECA&D)

References

Klein Tank AMG and Coauthors (2002). Daily Dataset of 20th-Century Surface Air Temperature and Precipitation Series for the European Climate Assessment. *International Journal of Climatology*, **22**(12), 1441-1453.

See Also

ZaragozaSeries

ZaragozaSeries *Splitted and Uncorrelated Time Series* TX_Zaragoza

---

Description

The matrix resulting from the data preparation (or pre-processing) of TX_Zaragoza$TX.

Usage

```
data(ZaragozaSeries)
```

Format

A matrix with 70 rows and 76 columns.

Details

The matrix is the result from applying: `series_uncor(series_split(TX_Zaragoza$TX))`.

The data matrix corresponds to the 70 years with observations in TX_Zaragoza$TX and to the 76 days in the year where adjacent daily maximum temperature subseries are uncorrelated. By coincidence, none of the subseries 4, 90 or 278 with missing observations is kept within the 76 uncorrelated days.

See Also

TX_Zaragoza

Index