# Package 'RecordTest'

October 21, 2019

**Type** Package

**Title** Inference Tools Based on Record Statistics

**Version** 0.1.0

**Date** 2019-09-23

**Author** Jorge Castillo-Mateo [aut, cre, cph]
(<https://orcid.org/0000-0003-3859-0248>),
Ana C. Cebrián [aut],
Jesús Asín [dtc]

**Maintainer** Jorge Castillo-Mateo <720193@unizar.es>

**Depends** R (>= 2.10)

**Imports** ggplot2, car, Smisc

**Description** Statistical tools based on probabilistic properties of the occurrence of records in a sequence X_t of independent and identically distributed random variables. More details on the objective of the library can be found with help(RecordTest-package).

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-10-21 13:20:02 UTC

## R topics documented:

---

`RecordTest-package` *Record test: A package for testing the classical record model*

---

## Description

Record test provides three families of tests to study the hypothesis of the classical record model. It also provides some preprocessing tools often needed to adequately prepare a dataset to apply those tests.

## Details

This package provides three families of tests to study the hypothesis of the classical record model, that is that the records from a series of values observed at regular time units (a vector) come from an i.i.d. series of continous random variables. A particular problem, common for example in climate problems, where these tools can be useful, is to detect the existence of a trend in a series of variables. If we have sequences of uncorrelated variables with no seasonal component, the hypothesis of i.i.d. variables is equivalent to test the hypothesis of no trend.

A sample of $M$ vectors uncorrelated between them is needed to apply the implemented tests. Then, the input of the functions to perform the statistical tools is a matrix `XM_T` where each column corresponds to a vector formed by the values of a series $X_t$, from $t = 1, ..., T$, 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 analyze show a seasonal behaviour, and only one realization 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 has to be obtained, with a high enough value $M$.

Those problems can be solved, preprocessing the data adequately. A wide set of tools to carry out a preliminary analysis and to preprocess daily data with a yearly seasonal pattern are implemented: `std.fun`, `dailymean.fun`, `dailysd.fun`,`DaySeries.fun`, `mintime.cor.test` and `double.fun`.

There is also available a set of functions to characterize the occurrence time, the value and other statistics related to the occurrence of records: `I.rec`, `N.rec`, `Nmean.rec`, `M.rec`, `L.rec`, `value.rec`.

All the tests are based on the occurrence of records, but three families are distinguished: the first is based on the indicators variables of record $I_t$, (`P_exactPB.test`, `P_chisq.test`,`P_regression.test`), and the second on the times of records $L_i$ (`L_MC.test`, `L_lr.test`). In both cases the null hypothesis is that the probabilies of record at time t are $p_t = 1/t$. The third family is based on the number of records up to time t, $N_t$ (`N_normal.test`, `N_joint.test`) and the null hypotehsis is related to the values of $E(N_t)$ and $Var(N_t)$ under the classical record model.

Finally, there is another set of functions aiming to plot different features related to records: `L.plot`, `N.plot`, `N_joint.plot`, and `P_regression.plot`.

All the tests and preprocessing tools can be applied to both upper and lower records, using the corresponding value in the argument `record`.

---

`dailymean.fun` *Seasonal pattern of the mean values*

---

## Description

This function estimates the seasonal pattern of the mean in a daily sequence of variables, that is the mean value of the variable at each day of the year.

## Usage

```
dailymean.fun(X_T, method = "moving", Nyear, day.year = NULL,
  harmonics = 3, window = 3)
```

## Arguments

| | |
|---|---|
| `X_T` | A numeric vector of daily observations. Its length has to be higher than 365. |
| `method` | A character string indicating the estimation procedure: `"harmonic"` or `"moving"`. |
| `Nyear` | Integer. Number of observed years. First and last year may have incomplete records. |
| `day.year` | Vector of integer values in (1:365) giving the day of the year where the observation was recorded. If it is null, records of complete years are assumed. |
| `harmonics` | Integer number of harmonics used in the estimation. Only used if `method=='harmonic'`. |
| `window` | Integer width of the moving window, which must be an odd number. Only used if `method=='moving'`. |

## Details

This function can be used in a preliminary analysis of data, in order to determine if the data show a seasonal behaviour, and to estimate the seasonal component.

Two approaches are implemented to calculate the seasonal pattern of the means, one based on linear combination of harmonics and the other on centered moving averages. In the last case, if `window=2p+1`, values obsered at days in the interval $(t-p, t+p)$ over all the years are used to calculate the mean at $t$. If `window=1`, only values obsered at day $t$ are used to calculate the mean

## Value

A numeric vector of length 365. Each element is the estimated mean at a day of the year.

## See Also

`std.fun`, `dailysd.fun`

## Examples

```
dailymean.fun(TX_Zaragoza$TX, method = 'moving', window = 1, Nyear = 66)
```

---

`dailysd.fun` *Seasonal pattern of the standar deviation*

---

## Description

This function estimates the seasonal pattern of the standard deviation in a daily sequence of variables, that is the standard deviation of the variable at each day of the year.

## Usage

```
dailysd.fun(X_T, method = "moving", Nyear, day.year = NULL,
  harmonics = 3, window = 3)
```

## Arguments

| | |
|---|---|
| `X_T` | A numeric vector of daily observations. Its length has to be higher than 365. |
| `method` | A character string indicating the estimation procedure: `"harmonic"` or `"moving"`. |
| `Nyear` | An integer. Number of observed years. First and last year may have incomplete records. |
| `day.year` | A vector of integer values in (1: 365) giving the day of the year where the observation was recorded. If it is null, records of complete years are assumed. |
| `harmonics` | Integer number of harmonics used in the estimation. Only used if `method=='harmonic'`. |
| `window` | Integer width of the moving window, which must be an odd number. Only used if `method=='moving'`. |

## Details

This function can be used in a preliminary analysis of data, in order to determine if the data show a seasonal behaviour, and to estimate the seasonal component.

Two approaches are implemented to calculate the seasonal pattern of the standard deviation, one based on linear combination of harmonics and the other on centered moving standard deviations. In the last case, if `window=2p+1`, values obsered at days in the interval$(t-p, t+p)$ over all the years are used to calculate the standard deviation at $t$. If `window=1`, only values obsered at day $t$ are used to calculate the standard deviation

## Value

A numeric vector of length 365. Each element is the estimated standard deviation at a day of the year.

## See Also

`std.fun`, `dailymean.fun`

## Examples

```
dailysd.fun(TX_Zaragoza$TX, method = 'moving', window = 1, Nyear = 66)
```

---

`DaySeries.fun` *Transforms a daily values vector into a matrix*

---

## Description

This function rearranges a vector of consecutive daily values into a matrix format, where each column is the vector of values at the same day of the year.

## Usage

```
DaySeries.fun(vect, ncols = 365)
```

## Arguments

| | |
|---|---|
| `vect` | A numeric vector. |
| `ncols` | 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 record inference tools in this package.

This function transforms a daily values vector into a matrix, applying the following procedure: the first row of the matrix is made up of the first `ncols` elements of the vector, the second row by the `ncols` following elements, and so on. The length of the vector must be a multiple of `ncols`.

In the case of a vector of daily values, `ncols` is usually 365, so that the first column corresponds to all the values observed at the first of January, the second to the second of January, etc,

## Value

A matrix.

## See Also

`double.fun`, `mintime.cor.test`

## Examples

```
DaySeries.fun(1:100, ncols = 10)
```

---

`double.fun` *Transforms a* $T \times M$ *in a* $[T/2] \times 2M$ *matrix*

---

## Description

It changes the format of a matrix, in the following way. First, the matrix is divided into two matrices $[T/2] \times M$ , containing the odd and the even rows of the original matrix, respectively, and secondly those matrices are cbinded.

## Usage

```
double.fun(mm)
```

## Arguments

| | |
|---|---|
| `mm` | A numeric matrix. |

## Details

This function is used in the data preparation (or pre-processing) often required to apply the record inference tools in 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 double the value $M$. The approach consists of considering that the observations at two consesutive times, $t$ and $t+1$, are independent observations measured at the same time unit. That means that we are doubling the original time unit of the records, so that the length of the observation period will be $[T/2]$. This function rearranges the original data matrix into the new format.

If the number of rows of the original matrix is even, the first row is deleted.

## Value

A $[T/2] \times 2M$ matrix.

## See Also

DaySeries.fun, mintime.cor.test

## Examples

```
double.fun(matrix(1:100,10,10))
```

---

# I.rec — *Indicator of record occurrence*

---

## Description

Given a vector, this function calculates a binay variable which takes the value 1 if the corresponding value in the vector is a record and 0 otherwise.

## Usage

```
I.rec(X_T, record = "upper")

I.rec.matrix(XM_T, record = "upper")
```

## Arguments

| | |
|---|---|
| X_T | A numeric vector. |
| record | A character string indicating the type of record to be calculated, "upper" or "lower". |
| XM_T | A numeric matrix. |

## Details

Indicators of both upper and lower records can be calculated.

The function `I.rec.matrix` is equivalent to `I.rec` but it is applied to a matrix. The previous procedure to obtain a binary variable is applied to each column of the matrix. In this function, argument `X_T` is a matrix called `XM_T`, and the return value is a matrix formed by the corresponding binary variables.

## Value

A binary vector, indicating the record occurrence.

## See Also

`M.rec`, `N.rec`, `I.rec`, `value.rec`

## Examples

```
Y1<-c(5,7,3,6,19,2,20)
I.rec(Y1)
I.rec.matrix(ZaragozaSeries)
```

---

`L.plot` *Plot of the times of record*

---

## Description

This function constructs a ggplot object to display the times of record.

## Usage

```
L.plot(XM_T, record = "upper", backward = FALSE,
  colour_point = "salmon", colour_line = "grey95")
```

## Arguments

| | |
|---|---|
| `XM_T` | A numeric vector or matrix. |
| `record` | A character string indicating the type of record to be calculated, "upper" or "lower". |
| `backward` | Logical flag. If `TRUE`, the input vector is reversed before calculating the times of records. |
| `colour_point` | Colour to plot points. |
| `colour_line` | Colour to plot lines. |

## Details

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

## Value

A ggplot object.

### See Also

L.rec, L_lr.test

### Examples

```
Y1<-c(1,5,3,6,6,9,2, 11, 17, 8)
L.plot(Y1)
L.plot(ZaragozaSeries)
```

## L.rec    *Times of records*

### Description

This function calculates the time (position in the vector) where records occur.

### Usage

```
L.rec(X_T, record = "upper")

L.rec.matrix(XM_T, record = "upper")
```

### Arguments

| | |
|---|---|
| X_T | A numeric vector. |
| record | A character string indicating the type of record to be calculated, "upper" or "lower". |
| XM_T | A numeric matrix. |

### Details

The function L.rec.matrix is equivalent to L.rec but it is applied to a matrix. The approach to obtain times of records is applied to each column of the matrix. In this function, argument X_T is a matrix called XM_T, and the return value is a list formed by the resulting vectors.

### Value

A vector, containing the times of records.

### See Also

I.rec,M.rec, N.rec, value.rec

### 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)
L.rec(Y1)
L.rec.matrix(cbind(Y1,Y2,Y3))
```

---

## L_lr.test    *Asymptotic likelihood ratio test on record times*

---

### Description

This function performs an asymptotic likelihood ratio test based on the record times $L_i$ to study the hypothesis of the classical record model.

### Usage

```
L_lr.test(XM_T, record = "upper")
```

### Arguments

| | |
|---|---|
| XM_T | A numeric matrix. |
| record | A character string indicating the type of record, "upper" or "lower". |

### Details

The null hypothesis of this likelihood ratio test is that in all the series, $m = 1, ...M$, the probability of record at time $t$ is $1/t$, and the alternative that the probability at time $t$ is any value, but equal in the $M$ series. The alternative hypothesis is more specific than the one in L_MC.test.

Under the null, the likelihood ratio statistic has an asymptotic $\chi^2$ distribution with $T - 1$ degrees of freedom.

### Value

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

| | |
|---|---|
| statistic | Value of the statistic. |
| parameter | Degrees of freedom of the approximate $\chi^2$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. |

### See Also

L.rec, L.plot, L_MC.test

## Examples

```
L_lr.test(ZaragozaSeries)
```

---

`L_MC.test` *Monte Carlo likelihood ratio test on record times*

---

## Description

This function performs a likelihood ratio test based on the record times $L_i$ to study the hypothesis of the classical record model using a Monte Carlo approach.

## Usage

```
L_MC.test(XM_T, record = "upper", samples = 1000)
```

## Arguments

| | |
|---|---|
| `XM_T` | A matrix. |
| `record` | A character string indicating the type of record to be calculated, "upper" or "lower". |
| `samples` | An integer specifying the number of replicates used in the Monte Carlo approach. |

## Details

The null hypothesis of this likelihood ratio test is that in all the series, $m = 1, ..., M$, the probability of record at time $t$ is $1/t$, and the alternative that the probability at time $t$ is any value, in any of the $M$ series. The alternative hypothesis is more general than the one in `L_lr.test`.

The statistic is the likelihood ratio statistic, and the p-value is obtained using a Monte Carlo approach.

## Value

A `"htest"` object with elements:

| | |
|---|---|
| `statistic` | Value of the likelihood ratio statistic. |
| `p.value` | P-value |
| `method` | A character string indicating the type of test. |
| `data.name` | A character string giving the name of the data. |

## See Also

`L.rec`, `L.plot`, `L_lr.test`

## Examples

```
L_MC.test(ZaragozaSeries, samples = 200)
```

M.rec *Number of records in M vectors at each time*

### Description

This function calculates the number of records at each time $t$ in a set of $M$ vectors.

### Usage

```
M.rec(XM_T, record = "upper")

P.rec(XM_T, record = "upper")
```

### Arguments

| | |
|---|---|
| XM_T | A numeric matrix. |
| record | A character string indicating the type of record to be calulated, "upper" or "lower". |

### Details

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

Summaries for both upper and lower records can be calculated.

The function P.rec is equivalent, but calculates the proportion of records at each time, that is the ratio: $(numberofrecords)/M$. This proportion is an estimation of the probability of record at that time.

### Value

A vector with the number (proportion in the case of P.rec) of records at each time (row).

### See Also

I.rec, N.rec, value.rec

### 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)
M.rec(cbind(Y1,Y2,Y3))
M.rec(ZaragozaSeries)
P.rec(ZaragozaSeries, record = 'lower')
```

mintime.cor.test *Extracts a subset of uncorrelated vectors*

## Description

Given a a set of $M$ vectors, this function extracts a subset of them which are uncorrelated.

## Usage

```
mintime.cor.test(XM_T, m = 1, alpha = 0.05)
```

## Arguments

| | |
|---|---|
| XM_T | A numeric matrix where the uncorrelated vectors are extracted from. |
| m | Integer valu giving the starting column. |
| alpha | Numeric value in $(0, 1)$. It gives the sgnificance level of the correlation test. |

## Details

This function is used in the data preparation (or pre-processing) often required to apply the record inference tools in this package.

Given a a set of $M$ vectors, which are the columns of matrix XM_T, 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 m and m + 1, m + 2, ... an so on up to find a column m + k which is not significantly correlated with column m. Then, the process is repeated starting at column m + k.

## Value

A vector with the index of the uncorrelated columns in the matrix.

## See Also

DaySeries.fun, double.fun

## Examples

```
ZM_T <- DaySeries.fun(TX_Zaragoza$TX)
mintime.cor.test(ZM_T)
```

N.plot *Plot of the mean number of records up to time t*

## Description

This function constructs a ggplot object to compare the sample means of the number of records in a vector up to time $t$, $\bar{N}_t$, and the expected values $E(N_t)$ under the classical record model.

## Usage

```
N.plot(XM_T, record = "both", interval = "ribbon", conf = 0.95,
  bootstrap = FALSE, samples = 1000, colour = "salmon")
```

## Arguments

| | |
|---|---|
| `XM_T` | A matrix. |
| `record` | A character string indicating the type of record, `"upper"`, `"lower"` or `"both"`. |
| `interval` | A character string indicating the type of display of the confidence intervals, `"ribbon"` (grey area) or `"errorbar"` (vertical lines). |
| `conf` | Confidence level of the confidence intervals. |
| `bootstrap` | A logical flag. If `FALSE` calculates an asymptotically normal CI, if `TRUE` calculates a Monte Carlo CI. |
| `samples` | An integer giving the number of replicates used to calculate Monte Carlo CI. Only used if `bootstrap = TRUE`. |
| `colour` | Colour used to plot the expected values of $N_t$, and the CI. See `ggplot` for valid values. |

## Details

First, this function calculates the sample means of the number of records in a vector up to time $t$. These sample means $\bar{N}_t$ are calculated from the sample of $M$ values obtained from $M$ vectors, the columns of matrix `XM_T`. Then, these values are plotted and compared with the expected values $E(N_t)$ and their confidence intervals (CI), under the hypothesis of the classical record model.

Two types of CI of $E(N_t)$ can be built. The first uses the fact that, under the classical record model, the statistic $\bar{N}_t$ is asymptotically Normal. The second is obtained using a bootstrap approach,

## Value

A ggplot object.

## See Also

`N_joint.test`, `N_normal.test`

## Examples

```
Zplot<-N.plot(ZaragozaSeries,  interval='errorbar', bootstrap=TRUE, samples=200)
Zplot
library(ggplot2)
Zplot +
  theme(axis.text = element_text(size = 18), legend.position = c(0.8,0.2),
    legend.text = element_text(size = 25), axis.title = element_text(size = 30)) +
  labs(title = expression(paste("Normal CI of ", N[t])),
    caption = "Zaragoza Data",
    x = expression(paste("t")),
    y = expression(paste("Mean number of records ", bar(N)[t]))) +
  scale_y_continuous(limits = c(1, 4.5)) +
  guides(colour = guide_legend(order = 1,reverse=TRUE), shape = guide_legend(order = 2))
```

---

N.rec *Number of records up to time t*

---

## Description

This function calculates $N_t$, the number of records up to time $t$ in a vector.

## Usage

```
N.rec(X_T, record = "upper")

N.rec.matrix(XM_T, record = "upper")
```

## Arguments

| | |
|---|---|
| X_T | A numeric vector. |
| record | A character string indicating the type of record to be calculated, "upper" or "lower". |
| XM_T | A numeric matrix. |

## Details

The function `N.rec.matrix` is equivalent to `N.rec` but the input is a matrix, and the previous procedure to obtain the number of records is applied to each column of the matrix. In this function, argument `X_T` is a matrix called `XM_T`, and the return value is a matrix formed by the output vectors obtained from each column.

## Value

A numeric vector with the number of records up to each time (row).

## See Also

Nmean.rec, L.rec, I.rec

## 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)
N.rec(Y1)
N.rec.matrix(cbind(Y1,Y2,Y3))
```

---

# Nmean.rec *Mean number of records*

---

## Description

This function calculates the mean number of records up to time $t$ in a vector.

## Usage

```
Nmean.rec(XM_T, record = "upper")
```

## Arguments

| | |
|---|---|
| XM_T | A numeric vector or matrix. |
| record | A character string indicating the type of record to be calculated, "upper" or "lower". |

## Details

If the argument XM_T is a matrix, the approach to obtain the mean number of records is applied to each column of the matrix, and the return value is a matrix formed by the corresponding vectors.

## Value

A numeric vector or matrix (depending on the input XM_T) with the mean number of records up to each time (row).

## See Also

N.rec, L.rec, I.rec

## 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)
Nmean.rec(Y1)
Nmean.rec(cbind(Y1,Y2,Y3))
```

| N_joint.plot | *Plot of the difference (or ratio) between the number of upper and lower records* |
|---|---|

## Description

This function constructs a ggplot object to display the difference (or ratio) between the mean number of upper records and the mean number of lower records. The expected values and confidence intervals of $E(N_t) - E(N_t^{low})$ (or $E(N_t)/E(N_t^{low})$), under the hypothesis of the classical record model, are also plotted.

## Usage

```
N_joint.plot(XM_T, type = "ratio", interval = "ribbon", conf = 0.95,
  samples = 1000, colour = "salmon")
```

## Arguments

| | |
|---|---|
| `XM_T` | A matrix. |
| `type` | A character string indicating the type of statistic to be plotted, "ratio" or "difference". |
| `interval` | A character string indicating the type of display of the confidence intervals, `"ribbon"` (grey area) or `"errorbar"` (vertical lines). |
| `conf` | Numeriv value in $(0, 1)$. Confidence level of the confidence intervals. |
| `samples` | An integer giving the number of replicates used to calculate Monte Carlo confidence intervals. |
| `colour` | Colour used to plot the expected values and the CI. See `ggplot` for valid values. |

## Details

First, this function calculates the difference (or ratio) between the mean number of upper records and the mean number of lower records up to time t, that is $\bar{N}_t - \bar{N}_t^{low}$ (or $\bar{N}_t/\bar{N}_t^{low}$), The sample means $\bar{N}_t$ and $\bar{N}_t^{low}$ are calculated from the sample of values $N_t$ and $N_t^{low}$, obtained from $M$ vectors (columns of matrix `XM_T`). Then, these values are plotted and compared with the functions of the expected values $E(N_t) - E(N_t^{low})$ (or $E(N_t)/E(N_t^{low})$) and their confidence intervals under the hypothesis of the classical record model,which are obtained using a bootstrap approach.

## Value

A ggplot graph object.

## See Also

`N_joint.test`, `N.plot`, `N_normal.test`

## Examples

```
N_joint.plot(ZaragozaSeries, samples=200)
```

## N_joint.test — *Monte Carlo test on the number of upper and lower records*

### Description

This function performs a Monte Carlo test based on the number of upper and lower records, $N_t$ and $N_t^{low}$, to study the hypothesis of the classical record model.

### Usage

```
N_joint.test(XM_T, stand = FALSE, samples = 1000,
  trend = "increasing")
```

### Arguments

| | |
|---|---|
| `XM_T` | A matrix. |
| `stand` | Logical flag. If `TRUE` uses a standarized version of the test statistic. |
| `samples` | An integer specifying the number of replicates used in the Monte Carlo approach. |
| `trend` | A character string indicating the type of trend ("increasing" or "decreasing") of the alternative hypothesis. |

### Details

The null hypothesis of this test is that the expected value of the number of upper and lower records in the observation period is the same $E(N_T) = E(N_T^{low})$. Two statistics can be used, $\bar{N}_t - \bar{N}_t^{low}$, or its standardized version; $\bar{N}_t$ and $\bar{N}_t^{low}$ are the mean of the number of records up to time $t$, calculated from a samples of $M$ vectors (columns in `XM_T`).

These statistics are useful when the alternative hypotehsis is that the sequences of variables in the vectors have a monotonou trend. If the trend is increasing the statistic will take high values while if it is decreasing, it will take low values. The type of trend has to be specified at argument `trend` to calculate the adequate p-value.

The distribution of the statistic under the hypothesis of the classical record model is obtained using a bootstrap approach.

### Value

A list with class `"htest"` containing the following components:

| | |
|---|---|
| `statistic` | Value of the likelihood ratio statistic. |
| `p.value` | P-value. |
| `samples` | Number of samples used in the Monte Carlo approach. |
| `method` | A character string indicating if the statistic is standardized or not. |
| `data.name` | A character string giving the name of the data. |

### See Also

N_joint.plot, N.plot, N_normal.test

### Examples

```
N_joint.test(ZaragozaSeries, stan=TRUE, samples=200)
```

---

## N_normal.test *Normal test on the number of records*

---

### Description

This function performs a test based on $N_T$, the number of records in the observation period, to study the hypothesis of the classical record model.

### Usage

```
N_normal.test(XM_T, record = "upper", type = "uni")
```

### Arguments

| | |
|---|---|
| XM_T | A numeric matrix. |
| record | A character string indicating the type of record to be calculated, "upper" or "lower". |
| type | A character string indicating the type of alternative hypothesis, unilateral "uni" or bilateral "bi". |

### Details

In this test the null hypothesis is that the expected value of $N_T$, the number of records in the observation period (0,T) is $\mu_T = \sum_{i=1}^{T} 1/i$ and the variance $\sigma_T^2 = \sum_{i=1}^{T}(1/i - 1/i^2)$; these are the values obtained under the classical record model (see Arnold et al. (2008)). The test statistic is based on $\bar{N}_t$, the mean of the number of records up to time t, calculated from a sample of M vectors (columns in XM_T). The distribution of the statistic under the null is asymptotically Normal.

If the sequences of variables in the vectors are not i.i.d, but they have a monotonous increasing trend, an unilateral alternative hypothesis must be stated, which in the case of upper records is $\mu_T > \sum_{i=1}^{T} 1/i$ and $\sigma_T^2 > \sum_{i=1}^{T}(1/i - 1/i^2)$, and in the case of lower records is $\mu_T < \sum_{i=1}^{T} 1/i$ and $\sigma_T^2 < \sum_{i=1}^{T}(1/i - 1/i^2)$.

### Value

A "htest" object with elements:

| | |
|---|---|
| statistic | Value of the test statistic. |
| parameter | Length of the observation period $T$. |
| p.value | P-value. |
| type | A character string indicating the type of test performed. |
| data.name | A character string giving the name of the data. |

### References

Arnold, B.C., Balakrishnan, N. and Nagaraja, H.N. (2008). Record Values. En R.E. O'Malley (Ed.) A First Course in Order Statistics. SIAM.

### See Also

N.plot

### Examples

```
N_normal.test(ZaragozaSeries)
```

---

## P_chisq.test — *Chi-Square test on record probabilities*

---

### Description

This function performs a chi-square test based on the record probabiliteis $p_t$ to study the hypothesis of the classical record model.

### Usage

```
P_chisq.test(XM_T, record = "upper")
```

### Arguments

| | |
|---|---|
| XM_T | A matrix. |
| record | A character string indicating the type of record to be calculated, "upper" or "lower". |

### Details

The null hypothesis of this likelihood ratio test is that in all the vectors (columns of matrix XM_T), the probability of record at time $t$ is $1/t$, and the alternative that the probabilities are not equal to those values. First, the chi-square goodness of fit statistics to study the null hypotehsis $H_0 : p_t = 1/t$ are calculated for each time $t = 2, ..., T$, 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 staitstic is the sum of the previous $T - 1$ statistics and its distribution under the null is approximately $\chi^2_{T-1}$.

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 is between 1 and 5. If this condition is not satisfied, a warning is displayed.

## Value

A "htest" object with elements:

| | |
|---|---|
| statistic | Value of the likelihood ratio statistic. |
| df | Degrees of freedom of the approximate chi-squared. |
| 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. |

## See Also

P_exactPB.test, P_regression.test

## Examples

```
P_chisq.test(ZaragozaSeries)
```

---

P_exactPB.test *Exact Poisson binomial test on record probabilities*

---

## Description

This function performs an exact test based on the record probabiliteis $p_t$ to study the hypothesis of the classical record model.

## Usage

```
P_exactPB.test(XM_T, record = "upper", method = "butler")

N_exactPB.test(XM_T, record = "upper", method = "butler")
```

## Arguments

| | |
|---|---|
| XM_T | A matrix. |
| record | A character string indicating the type of record to be calculated, "upper" or "lower". |
| method | A character string indicating the method to calculate the Poisson binomial distribution, "butler", "naive" or "fft". See argument method in pkbinom. |

### Details

The null hypothesis of this likelihood ratio test is that in all the vectors (columns of matrix XM_T), the probability of record at time $t$ is $1/t$. The test statistic is the total number of records at times $t = 2, ..., T$, in the $M$ vectors. Under the null, this is the sum of $M(T-1)$ independent Bernoulli variables, with probabilities $p_2, ..., p_2, ..., p_T, ...p_T$ with $p_t = 1/t$, so that its distribution is a Poisson-Binomial.

Only unilateral alternative hypotehesis $p_t > 1/t, t = 2, ..., T$ or $p_t < 1/t, t = 2, ..., T$ are valid, since otherwise the statistic is not able to detect deviations from the null hypothesis.

N_exactPB.test is the same test, but applied to only one vector, instead of M. Note in this case this test considers the probability at time $t = 1$ (by definition of $N_t$), but the p-value is the same.

### Value

A "htest" object with elements:

| | |
|---|---|
| statistic | Value of the likelihood ratio statistic. |
| parameter | Number of Bernoulli independent variables summed in the statistic. |
| 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. |

### See Also

P_chisq.test,P_regression.test

### Examples

```
P_exactPB.test(ZaragozaSeries)
N_exactPB.test(ZaragozaSeries[,23])
```

---

## P_regression.plot *Plot of record probabilities*

---

### Description

This function constructs a ggplot object to display different functions of the record probabilities at time t, $p_t$.

### Usage

```
P_regression.plot(XM_T, record = "upper", plot = 2,
  interval = "ribbon", conf = 0.95, samples = 5000,
  colour_point = "black", colour_CI = "salmon",
  color_lm = "royalblue4")
```

## Arguments

| | |
|---|---|
| XM_T | A matrix. |
| record | A character string indicating the type of record to be calculated, "upper" or "lower". |
| plot | One of the values 1, 2 or 3. It determines the type of plot to be displayed. See Details. |
| interval | A character string indicating the type of display of the confidence intervals, "ribbon" (grey area) or "errorbar" (vertical lines). |
| conf | Numeric value in (0,1). Confidence level of the confidence intervals. |
| samples | An integer giving the number of replicates used to calculate Monte Carlo CI. |
| colour_point | Colour used to plot the points. See ggplot for valid values. |
| colour_CI | Colour used to plot the expected values and the CI. |
| color_lm | Colour used to plot the regression line. Only used if plot==2. |

## Details

Three different types of plots which aim to analyse the hypothesis of the record classic 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 XM_T).

Type 1 is the plot of the estimated record probabilities $p_t$ versus time. The expected probabilities under the record classic model, $p_t = 1/t$, are also plotted, together with bootstrap confidence intervals. Type 3 is the same plot but on a logarithmic scale, so that the expected value is $-log(t)$.

Type 2 is the plot of the observed values $t\hat{p}_t$ versus time. 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 bootstrap confidence intervals. In addition, a regression line is fitted to the cloud of points and plotted together with confidence intervals of the response. If the classical record model is true, the confidence band (in grey) should contain the horizontal line equal to 1. Plots of type 2 are easier to interpret than types 1 and 3.

## Value

A ggplot object.

## See Also

P_regression.test

## Examples

```
P_regression.plot(ZaragozaSeries,  plot=2, interval='errorbar', samples=200)
```

## P_regression.test *Regression test on record probabilities*

### Description

This function performs a test based on a regression on the record probabilities $p_t$ to study the hypothesis of the classical record model.

### Usage

```
P_regression.test(XM_T, record = "upper")
```

### Arguments

| | |
|---|---|
| XM_T | A matrix |
| record | A character string indicating the type of records to be calculated, "upper" or "lower". |

### Details

The null hypothesis of this likelihood ratio test is that in all the vectors (columns in matrix XM_T), the probability of record at time $t$ is $1/t$, so that $tp_t = 1$. Then, hypothesis $H_0 : p_t = 1/t,\ t = 2, ..., T$ is equivalent to $H_0 : \beta_0 = 1, \beta_1 = 0$ where $\beta_0$ and $\beta_1$ are the coefficients of the regression model $tp_t = \beta_0 + \beta_1 t$. The model has to be estimated by weighted least squares since the response is heteroskedastic.

The F statistic is used to compare the regression model under the null and a linear regression model with no restriction (the alterantive hypotehsis is then that $tp_t$ is a linear function of time). This alternative hypotehsis may be reasonable in many real examples, but not always.

### Value

A "htest" object with elements:

| | |
|---|---|
| statistic | Value of the likelihood ratio statistic. |
| intercept | Estimated intercept of the regression line $\hat{\beta}_0$. |
| slope | Estimated slope of the regression $\hat{\beta}_1$. |
| 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. |

### See Also

P_exactPB.test, P_chisq.test,P_regression.plot

### Examples

```
P_regression.test(ZaragozaSeries)
```

std.fun *Removing seasonal components of a vector*

## Description

This function removes a yearly seasonal behaviour of a vector of daily data by subtracting the means and dividing by the standard deviations, which define a seasonal pattern, that is the means and standard deviations calculated for each day of the year.

## Usage

```
std.fun(X_T, method = "moving", Nyear, day.year = NULL,
  harmonics = 3, window = 3)
```

## Arguments

| | |
|---|---|
| `X_T` | A numeric vector. |
| `method` | A character string indicating the estimation procedure, `"harmonic"` or `"moving"`. |
| `Nyear` | Integer number of observed years. First and last year may be incomplete. |
| `day.year` | Vector of integer values in (1:365) giving the day of the year where the observation was measured. If it is null, 365 observations per year are assumed. |
| `harmonics` | Integer number of harmonics used in the estimation. Only used if `method=='harmonic'`. |
| `window` | Integer width of the moving window, which must be an odd number. Only used if `method=='moving'`. |

## Details

This function can be used in the data preparation (or pre-processing) often required to apply the record inference tools in this package.

Two approaches are implemented to calculate the mean and standard deviations defining the seasonal pattern, one based on centered moving averages and the other on a linear combination of harmonics.

## Value

Input vector, standardized by the seasonal components.

## See Also

`dailymean.fun`, `dailysd.fun`

## Examples

```
std.fun(TX_Zaragoza$TX, method = 'harmonic', harmonics = 2, Nyear=66)
```

`TX_Zaragoza` *TX_Zaragoza*

## Description

A dataset containing the series of daily maximum temperature at Zaragoza (Spain), from 1/1/1953 to 31/12/2018. This series is obtaned from the ECA series but it has been transformed, by removing days February 29th and filling the missing values. The variables are the following:

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

## Usage

```
data(TX_Zaragoza)
```

## Format

A data frame with 28670 rows and 5 variables.

## Source

EUROPEAN CLIMATE ASSESSMENT & DATASET (ECA&D)

## References

Klein Tank, A.M.G. and Coauthors, 2002. Daily dataset of 20th-century surface air temperature and precipitation series for the European Climate Assessment. Int. J. of Climatol., 22, 1441-1453.

`value.rec` *Record values*

## Description

This function identifies the record values ($R_i$), and the record times ($L_i$), in a vector.

## Usage

```
value.rec(X_T, record = "upper", variables = NULL)
```

## Arguments

| | |
|---|---|
| X_T | A numeric vector. |
| record | A character string indicating the type of record be calculated, "upper" or "lower". |
| variables | Optional. A matrix, containing other variables related to X_T and measured at the same times. |

## Value

A data frame where the first column are the record times, the second the record values and, if variables is not null, the third column are their values at the record times.

## Examples

```
 Y1<-c(5,7,3,6,19,2,20)
value.rec(Y1)
value.rec(TX_Zaragoza$TX,  variable= TX_Zaragoza$DATE)
```

---

ZaragozaSeries *ZaragozaSeries*

---

## Description

The matrix resulting from the preprocessing of TX_Zaragoza$TX. The matrix is the result from applying consecutively: DaySeries.fun, mintime.cor.test and double.fun.

## Usage

```
data(ZaragozaSeries)
```

## Format

A matrix with 33 rows and 150 columns.

# Index