Package ‘hydroGOF’

August 8, 2017

Type Package
Title Goodness-of-Fit Functions for Comparison of Simulated and Observed Hydrological Time Series
Version 0.3-10
Date 2017-08-08
Maintainer Mauricio Zambrano-Bigiarini <mzb.devel@gmail.com>
Description S3 functions implementing both statistical and graphical goodness-of-fit measures between observed and simulated values, mainly oriented to be used during the calibration, validation, and application of hydrological models. Missing values in observed and/or simulated values can be removed before computations. Comments/questions/collaboration of any kind are very welcomed.
License GPL (>= 2)
Depends R (>= 2.10.0), zoo (>= 1.7-2)
Imports hydroTSM (>= 0.5-0), xts (>= 0.8-2), methods
URL http://hzmbran.github.io/hydroGOF/
BugReports https://github.com/hzmbran/hydroGOF/issues
LazyLoad yes
NeedsCompilation no
Repository CRAN
Author Mauricio Zambrano-Bigiarini [aut, cre, cph]
Date/Publication 2017-08-08 15:11:19 UTC

R topics documented:

hydroGOF-package .................................................... 2
br2 ................................................................. 5
cp ................................................................. 7
d ................................................................. 10
EgaEnEstellaQts ..................................................... 12
ggof ............................................................... 13
hydroGOF-package

Goodness-of-fit (GoF) functions for numerical and graphical comparison of simulated and observed time series, mainly focused on hydrological modelling.

Description

S3 functions implementing both statistical and graphical goodness-of-fit measures between observed and simulated values, to be used during the calibration, validation, and application of hydrological models.

Missing values in observed and/or simulated values can be removed before computations.

Quantitative statistics included are: Mean Error (me), Mean Absolute Error (mae), Root Mean Square Error (rms), Normalized Root Mean Square Error (nrmse), Pearson product-moment correlation coefficient (r), Spearman Correlation coefficient (r.Spearman), Coefficient of Determination (R2), Ratio of Standard Deviations (rSD), Nash-Sutcliffe efficiency (NSE), Modified Nash-Sutcliffe efficiency (mNSE), Relative Nash-Sutcliffe efficiency (rNSE), Index of Agreement (d), Modified Index of Agreement (md), Relative Index of Agreement (rd), Coefficient of Persistence.
(cp), Percent Bias (pbias), Kling-Gupta efficiency (KGE), the coef. of determination multiplied by the slope of the linear regression between 'sim' and 'obs' (bR2), and volumetric efficiency (VE).

Details

Package: hydroGOF
Type: Package
Version: 0.3-10
Date: 2017-08-08
License: GPL >= 2
LazyLoad: yes
Packaged: Tue Aug 8 09:11:25 CLT 2017; MZB
BuiltUnder: R version 3.4.1 (2017-06-30) – "Single Candle"; x86_64-pc-linux-gnu (64-bit)

Author(s)

Mauricio Zambrano Bigiarini <mauricio.zambrano@ing.unitn.it>
Maintainer: Mauricio Zambrano Bigiarini <mauricio.zambrano@ing.unitn.it>

References


See Also

https://CRAN.R-project.org/package=hydroPSO
https://CRAN.R-project.org/package=hydroTSM

Examples

obs <- 1:100
sim <- obs

# Numerical goodness of fit
gof(sim,obs)

# Reverting the order of simulated values
sim <- 100:1
gof(sim,obs)

## Not run:
ggof(sim, obs)

## End(Not run)

############################
Loading daily streamflows of the Ega River (Spain), from 1961 to 1970:
```r
require(zoo)
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts
```

Generating a simulated daily time series, initially equal to observations:
```r
sim <- obs
```

Getting the numeric goodness-of-fit measures for the "best" (unattainable) case:
```r
gof(sim=sim, obs=obs)
```

Randomly changing the first 2000 elements of 'sim', by using a normal distribution with mean 10 and standard deviation equal to 1 (default of 'rnorm').
```r
```

Getting the new numeric goodness of fit:
```r
gof(sim=sim, obs=obs)
```

Graphical representation of 'obs' vs 'sim', along with the numeric goodness-of-fit measures:
```r
ggof(sim=sim, obs=obs)
```

Description

Coefficient of determination (r²) multiplied by the slope of the regression line between sim and obs, with treatment of missing values.

Usage

```r
br2(sim, obs, ...)
```

Default S3 method:
```r
br2(sim, obs, na.rm=TRUE, ...)
```

S3 method for class 'data.frame':
```r
br2(sim, obs, na.rm=TRUE, ...)
```

S3 method for class 'matrix':
```r
br2(sim, obs, na.rm=TRUE, ...)
```

S3 method for class 'zoo':
```r
br2(sim, obs, na.rm=TRUE, ...)
```
Arguments

- **sim**: numeric, zoo, matrix or data.frame with simulated values
- **obs**: numeric, zoo, matrix or data.frame with observed values
- **na.rm**: a logical value indicating whether 'NA' should be stripped before the computation proceeds.

... further arguments passed to or from other methods.

Details

\[
br^2 = |b| R^2, |b| \leq 1; br^2 = \frac{R^2}{|b|} b > 1
\]

A model that systematically over or under-predicts all the time will still result in "good" $r^2$ (close to 1), even if all predictions were wrong (Krause et al., 2005). The $br^2$ coefficient allows accounting for the discrepancy in the magnitude of two signals (depicted by 'b') as well as their dynamics (depicted by $r^2$).

Value

$br^2$ between sim and obs.

If sim and obs are matrixes, the returned value is a vector, with the $br^2$ between each column of sim and obs.

Note

obs and sim has to have the same length/dimension

The missing values in obs and sim are removed before the computation proceeds, and only those positions with non-missing values in obs and sim are considered in the computation

The slope $b$ is computed as the coefficient of the linear regression between sim and obs, forcing the intercept be equal to zero.

Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

References

see the difference between \( r_R \) and \( b_R \) for a case with systematic over-prediction of observed values

\[
\text{obs} < Q: \text{QP} \quad \text{simQ} < Q: R^* \text{obs} \quad K \ U \\
\text{simR} < Q: R^* \text{obs} \quad K \ RU
\]

The coefficient of determination is equal to 1 even if there is no one single simulated value equal to its corresponding observed counterpart

\[
\text{rR} < Q: (\text{cor}\text{(simL, obs, method="pearson"))}^2 \quad # \text{rR=1}
\]

'\( b_R \)' effectively penalises the systematic over-estimation

\[
\text{br2(\text{simL, obs})} \quad # \text{br2} = 0.3684211 \\
\text{br2(\text{sim2, obs})} \quad # \text{br2} = 0.1794072
\]

\[
\text{ggof(\text{simL, obs})} \\
\text{ggof(\text{sim2, obs})}
\]

Computing '\( b_R \)' without forcing the intercept be equal to zero

\[
\text{br2.2} <- r2/2 \quad # \text{br2 = 0.5}
\]

-------------------------------

Loading daily streamflows of the Ega River (Spain), from 1961 to 1970

data(EgaEnEstellaQts)

\[
\text{obs} <- \text{EgaEnEstellaQts}
\]

Generating a simulated daily time series, initially equal to the observed series

\[
\text{sim} <- \text{obs}
\]

Computing '\( b_R \)' for the "best" (unattainable) case

\[
\text{br2(\text{sim=\text{sim}, obs=\text{obs})}
\]

Randomly changing the first 2000 elements of '\text{sim}', by using a normal distribution with mean 10 and standard deviation equal to 1 (default of '\text{rnorm}').

\[
\]

Computing the new '\( b_R \)'

\[
\text{br2(\text{sim=\text{sim}, obs=\text{obs})}
\]

---

### cp

**Coefficient of persistence**

#### Description

Coefficient of persistence between sim and obs, with treatment of missing values.
Usage

\[ cp(\text{sim}, \text{obs}, \ldots) \]

## Default S3 method:
\[ cp(\text{sim}, \text{obs}, \text{na.rm=TRUE}, \ldots) \]

## S3 method for class 'data.frame'
\[ cp(\text{sim}, \text{obs}, \text{na.rm=TRUE}, \ldots) \]

## S3 method for class 'matrix'
\[ cp(\text{sim}, \text{obs}, \text{na.rm=TRUE}, \ldots) \]

## S3 method for class 'zoo'
\[ cp(\text{sim}, \text{obs}, \text{na.rm=TRUE}, \ldots) \]

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sim</td>
<td>numeric, zoo, matrix or data.frame with simulated values</td>
</tr>
<tr>
<td>obs</td>
<td>numeric, zoo, matrix or data.frame with observed values</td>
</tr>
<tr>
<td>na.rm</td>
<td>a logical value indicating whether 'NA' should be stripped before the computation proceeds. When an 'NA' value is found at the i-th position in obs OR sim, the i-th value of obs AND sim are removed before the computation.</td>
</tr>
<tr>
<td>...</td>
<td>further arguments passed to or from other methods.</td>
</tr>
</tbody>
</table>

Details

\[
cp = 1 - \frac{\sum_{i=2}^{N} (S_i - O_i)^2}{\sum_{i=1}^{N-1} (O_{i+1} - O_i)^2}
\]

Coefficient of persistence (Kitadinis and Bras, 1980; Corradini et al., 1986) is used to compare the model performance against a simple model using the observed value of the previous day as the prediction for the current day.

The coefficient of persistence compare the predictions of the model with the predictions obtained by assuming that the process is a Wiener process (variance increasing linearly with time), in which case, the best estimate for the future is given by the latest measurement (Kitadinis and Bras, 1980).

Persistence model efficiency is a normalized model evaluation statistic that quantifies the relative magnitude of the residual variance (noise) to the variance of the errors obtained by the use of a simple persistence model (Moriasi et al., 2007).

CP ranges from 0 to 1, with CP = 1 being the optimal value and it should be larger than 0.0 to indicate a minimally acceptable model performance.
Value

Coefficient of persistence between `sim` and `obs`.

If `sim` and `obs` are matrixes, the returned value is a vector, with the coefficient of persistence between each column of `sim` and `obs`.

Note

`obs` and `sim` has to have the same length/dimension.

The missing values in `obs` and `sim` are removed before the computation proceeds, and only those positions with non-missing values in `obs` and `sim` are considered in the computation.

Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

References


See Also

gof

Examples

```r
obs <- 1:10
sim <- 1:10
cp(sim, obs)

obs <- 1:10
sim[2:10] <- obs[1:9]
cp(sim, obs)
```

`###########`

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Generating a simulated daily time series, initially equal to the observed series
sim <- obs

# Computing 'cp' for the "best" (unattainable) case
cp(sim=sim, obs=obs)
```
# Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').

# Computing the new 'cp'
cp(sim=sim, obs=obs)

---

### Index of Agreement

**Description**

This function computes the Index of Agreement between sim and obs, with treatment of missing values.

If x is a matrix or a data frame, a vector of the Index of Agreement of the columns is returned.

**Usage**

```r
 d(sim, obs, ...)  
```

```r
## Default S3 method:
d(sim, obs, na.rm=TRUE, ...)
```

```r
## S3 method for class 'data.frame'
d(sim, obs, na.rm=TRUE, ...)
```

```r
## S3 method for class 'matrix'
d(sim, obs, na.rm=TRUE, ...)
```

```r
## S3 method for class 'zoo'
d(sim, obs, na.rm=TRUE, ...)
```

**Arguments**

- `sim` numeric, zoo, matrix or data.frame with simulated values
- `obs` numeric, zoo, matrix or data.frame with observed values
- `na.rm` a logical value indicating whether 'NA' should be stripped before the computation proceeds.
  When an 'NA' value is found at the i-th position in obs OR sim, the i-th value of obs AND sim are removed before the computation.
- `...` further arguments passed to or from other methods.
\[ d = 1 - \frac{\sum_{i=1}^{N} (O_i - S_i)^2}{\sum_{i=1}^{N} (|S_i - O| + |O_i - \bar{O}|)^2} \]

The Index of Agreement (d) developed by Willmott (1981) as a standardized measure of the degree of model prediction error and varies between 0 and 1. A value of 1 indicates a perfect match, and 0 indicates no agreement at all (Willmott, 1981).

The index of agreement can detect additive and proportional differences in the observed and simulated means and variances; however, it is overly sensitive to extreme values due to the squared differences (Legates and McCabe, 1999).

**Value**

Index of agreement between \texttt{sim} and \texttt{obs}.

If \texttt{sim} and \texttt{obs} are matrixes, the returned value is a vector, with the index of agreement between each column of \texttt{sim} and \texttt{obs}.

**Note**

\texttt{obs} and \texttt{sim} has to have the same length/dimension.

The missing values in \texttt{obs} and \texttt{sim} are removed before the computation proceeds, and only those positions with non-missing values in \texttt{obs} and \texttt{sim} are considered in the computation.

**Author(s)**

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

**References**


**See Also**

\texttt{md}, \texttt{rd}, \texttt{gof}, \texttt{ggof}
Examples

obs <- 1:10
sim <- 1:10
d(sim, obs)

obs <- 1:10
sim <- 2:11
d(sim, obs)

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Generating a simulated daily time series, initially equal to the observed series
sim <- obs

# Computing the index of agreement for the "best" (unattainable) case
d(sim, sim, obs=obs)

# Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').

# Computing the new index of agreement
d(sim, sim, obs=obs)

EgaEnEstellaQts Ega in "Estella" (Q071), ts with daily streamflows.

Description

Time series with daily streamflows of the Ega River (subcatchment of the Ebro River basin, Spain) measured at the gauging station "Estella" (Q071), for the period 01/Jan/1961 to 31/Dec/1970

Usage

data(EgaEnEstellaQts)

Format

zoo object.

Source

Downloaded from: http://www.chebro.es. Last accessed [March 2010].
These data are intended to be used for research purposes only, being distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY.
**ggof**  

**Graphical Goodness of Fit**

**Description**

Graphical comparison between two vectors (numeric, ts or zoo), with several numerical goodness of fit printed as a legend.

Missing values in observed and/or simulated values can removed before the computations.

**Usage**

```r
ggof(sim, obs, na.rm = TRUE, dates, date.fmt = "%Y-%m-%d", 
pt.style = "ts", ftype = "o", FUN, 
stype="default", season.names=c("Winter", "Spring", "Summer", "Autumn"), 
gof.leg = TRUE, digits=2, 
gofs=c("ME", "MAE", "RMSE", "NRMSE", "PBIAS", "RSR", "rSD", "NSE", "mNSE", 
"rNSE", "d", "md", "rd", "r", "R2", "bR2", "KGE", "VE"), 
legend, leg.cex=1, 
tick.tstep = "auto", lab.tstep = "auto", lab.fmt=NULL, 
cal.ini=NA, val.ini=NA, 
main, xlab = "Time", ylab=c("Q, [m3/s]"), 
col = c("blue", "black"), 
cex = c(0.5, 0.5), cex.axis=1.2, cex.lab=1.2, 
lwd = c(1, 1), lty = c(1, 3), pch = c(1, 9), ...)
```

**Arguments**

- **sim**: numeric or zoo object with with simulated values
- **obs**: numeric or zoo object with observed values
- **na.rm**: a logical value indicating whether 'NA' should be stripped before the computation proceeds. When an 'NA' value is found at the i-th position in obs OR sim, the i-th value of obs AND sim are removed before the computation.
- **dates**: character, factor, Date or POSIXct object indicating how to obtain the dates for the corresponding values in the sim and obs time series. If dates is a character or factor, it is converted into Date/POSIXct class, using the date format specified by date.fmt
- **date.fmt**: OPTIONAL, character indicating the format in which the dates are stored in dates, cal.ini and val.ini. See format in as.Date. Default value is %Y-%m-%d. ONLY required when class(dates)=="character" or class(dates)=="factor" or when cal.ini and/or val.ini is provided.
- **pt.style**: Character indicating if the 2 ts have to be plotted as lines or bars. When ftype is NOT o, it only applies to the annual values. Valid values are:
  - ts: (default) each ts is plotted as a lines along the 'x' axis
  - bar: both series are plotted as barplots.
ftype  Character indicating how many plots are desired by the user. Valid values are:
        - o: only the original sim and obs time series are plotted
        - dm: it assumes that sim and obs are daily time series and Daily and Monthly values are plotted
        - ma: it assumes that sim and obs are daily or monthly time series and Monthly and Annual values are plotted
        - dma: it assumes that sim and obs are daily time series and Daily, Monthly and Annual values are plotted
        - seasonal: seasonal values are plotted. See stype and season.names

FUN  OPTIONAL, ONLY required when ftype is in c('dm', 'ma', 'dma', 'seasonal').
     Function that have to be applied for transforming the original ts into monthly,
     annual or seasonal time step (e.g., for precipitation FUN MUST be sum, for
     temperature and flow time series, FUN MUST be mean)

stype  OPTIONAL, only used when ftype=seasonal,
       character, indicating what weather seasons will be used for computing the output.
       Possible values are:
       - default => "winter"= DJF = Dec, Jan, Feb; "spring"= MAM = Mar, Apr, May; "summer"= JJA = Jun, Jul, Aug; "autumn"= SON = Sep, Oct, Nov
       - FrenchPolynesia => "winter"= DJFM = Dec, Jan, Feb, Mar; "spring"= AM = Apr, May; "summer"= JJAS = Jun, Jul, Aug, Sep; "autumn"= ON = Oct, Nov

season.names  OPTIONAL, only used when ftype=seasonal.
                character of length 4 indicating the names of each one of the weather seasons
                defined by stype. These names are only used for plotting purposes

gof.leg  logical, indicating if several numerical goodness of fit have to be computed
        between sim and obs, and plotted as a legend on the graph. If leg.gof=TRUE, then
        x is considered as observed and y as simulated values (for some gof functions
        this is important).

digits  OPTIONAL, only used when leg.gof=TRUE. Numeric, representing the decimal
       places used for rounding the goodness-of-fit indexes.

gofs  character, with one or more strings indicating the goodness-of-fit measures to be
       shown in the legend when gof.leg=TRUE.
       Possible values when ftype!='seasonal' are in c("ME", "MAE", "MSE", "RMSE", "NRMSE", "PBIAS"
       Possible values when ftype='seasonal' are in c("ME", "RMSE", "PBIAS", "RSR", "NSE", "d", "R2", "KGE", "VE")

legend  character of length 2 to appear in the legend.

leg.cex  OPTIONAL. ONLY used when leg.gof=TRUE. Character expansion factor for
drawing the legend, *relative* to current 'par("cex")'. Used for text, and pro-
vides the default for 'pt.cex' and 'title.cex'. Default value = 1

tick.tstep  character, indicating the time step that have to be used for putting the ticks on
        the time axis. Valid values are: auto, years, months, weeks, days, hours, minutes,
        seconds.

lab.tstep  character, indicating the time step that have to be used for putting the labels
        on the time axis. Valid values are: auto, years, months, weeks, days, hours,
        minutes, seconds.

lab.fmt  Character indicating the format to be used for the label of the axis. See lab.fmt in
        drawTimeAxis.
cal.ini  OPTIONAL. Character, indicating the date in which the calibration period started. When cal.ini is provided, all the values in obs and sim with dates previous to cal.ini are SKIPPED from the computation of the goodness-of-fit measures (when gof.leg=TRUE), but their values are still plotted, in order to examine if the warming up period was too short, acceptable or too long for the chosen calibration period. In addition, a vertical red line in drawn at this date.

val.ini  OPTIONAL. Character, the date in which the validation period started. ONLY used for drawing a vertical red line at this date.

main  character representing the main title of the plot.

xlab  label for the 'x' axis.

ylab  label for the 'y' axis.

col  character, representing the colors of sim and obs

cex  numeric, representing the values controlling the size of text and symbols of 'x' and 'y' with respect to the default

cex.axis  numeric, representing the magnification to be used for the axis annotation relative to 'cex'. See par.

cex.lab  numeric, representing the magnification to be used for x and y labels relative to the current setting of 'cex'. See par.

lwd  vector with the line width of sim and obs

lty  numeric with the line type of sim and obs

pch  numeric with the type of symbol for x and y. (e.g., 1: white circle; 9: white rhombus with a cross inside)

...  further arguments passed to or from other methods.

Details
Plots observed and simulated values in the same graph.
If gof.leg=TRUE, it computes the numerical values of:
'me', 'mae', 'rmse', 'nrmse', 'PBIAS', 'RSR', 'rSD', 'NSE', 'mNSE', 'rNSE', 'd', 'md', 'rd', 'cp', 'r', 'r.Spearman', 'R2', 'bR2', 'KGE', 'VE'

Value
me  Mean Error
mae  Mean Absolute Error
rmse  Root Mean Square Error
nrmse  Normalized Root Mean Square Error
PBIAS  Percent Bias
pbiasdfc  PBIAS in the slope of the midsegment of the Flow Duration Curve
RSR  Ratio of RMSE to the Standard Deviation of the Observations, RSR = rms / sd(obs). ( 0 <= RSR <= +Inf )

rSD  Ratio of Standard Deviations, rSD = sd(sim) / sd(obs)
NSE  Nash-Sutcliffe Efficiency \((-\infty \leq \text{NSE} \leq 1)\)

mNSE  Modified Nash-Sutcliffe Efficiency

rNSE  Relative Nash-Sutcliffe Efficiency

d  Index of Agreement \((0 \leq d \leq 1)\)

md  Modified Index of Agreement

rd  Relative Index of Agreement

cp  Persistence Index \((0 \leq PI \leq 1)\)

r  Pearson product-moment correlation coefficient \((-1 \leq r \leq 1)\)

r.Spearman  Spearman Correlation coefficient \((-1 \leq r.Spearman \leq 1)\)

R2  Coefficient of Determination \((0 \leq R^2 \leq 1)\).
Gives the proportion of the variance of one variable that is predictable from the other variable

bR2  R2 multiplied by the coefficient of the regression line between sim and obs
\((0 \leq bR^2 \leq 1)\)

KGE  Kling-Gupta efficiency between sim and obs
\((0 \leq KGE \leq 1)\)

VE  Volumetric efficiency between sim and obs
\((-\infty \leq VE \leq 1)\)

**Author(s)**

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

**References**


See Also
gof, plot2, me, mae, rmse, nrmse, pbias, pbiasfdc, rSD, NSE, mNSE, rNSE, d, md, rd, cp, br2, KGE, VE

Examples

```r
obs <- 1:10
sim <- 2:11

## Not run:
gof(sim, obs)

## End(Not run)

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
ox <- EgaEnEstellaQts

# Generating a simulated daily time series, initially equal to the observed series
sim <- obs

# Getting the numeric goodness of fit for the "best" (unattainable) case
gof(sim=sim, obs=obs)

# Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').

# Getting the new numeric goodness-of-fit measures
gof(sim=sim, obs=obs)
```
# Getting the graphical representation of 'obs' and 'sim' along with the numeric
goodness-of-fit measures for the daily and monthly time series
## Not run:
ggof(sim=sim, obs=obs, ftype="dm", FUN=mean)

## End(Not run)

## Getting the graphical representation of 'obs' and 'sim' along with some numeric
goodness-of-fit measures for the seasonal time series
## Not run:
ggof(sim=sim, obs=obs, ftype="seasonal", FUN=mean)

## End(Not run)

## Computing the daily residuals
## even if this is a dummy example, it is enough for illustrating the capability
r <- sim-obs

## Summarizing and plotting the residuals
## Not run:
library(hydroTSM)

## summary
smry(r)

## daily, monthly and annual plots, boxplots and histograms
hydroplot(r, FUN=mean)

## seasonal plots and boxplots
hydroplot(r, FUN=mean, pfreq="seasonal")

## End(Not run)

---

### gof  
**Numerical Goodness-of-fit measures**

#### Description

Numerical goodness-of-fit measures between sim and obs, with treatment of missing values. Several performance indices for comparing two vectors, matrices or data.frames

#### Usage

```r
gof(sim, obs, ...)
```

## Default S3 method:
```r
gof(sim, obs, na.rm=TRUE, do.spearman=FALSE, do.pbfdc=FALSE,
   j=1, norm="sd", s=c(1,1,1), method=c("2009", "2012"), lq.thr=0.7,
```
Arguments

- **sim**: numeric, zoo, matrix or data.frame with simulated values
- **obs**: numeric, zoo, matrix or data.frame with observed values
- **na.rm**: a logical value indicating whether 'NA' should be stripped before the computation proceeds. When an 'NA' value is found at the i-th position in **obs** OR **sim**, the i-th value of **obs** AND **sim** are removed before the computation.
- **do.spearman**: logical. Indicates if the Spearman correlation has to be computed. The default is FALSE.
- **do.pbfdec**: logical. Indicates if the Percent Bias in the Slope of the midsegment of the Flow Duration Curve (**pbiasfdc**) has to be computed. The default is FALSE.
- **j**: argument passed to the **mnse** function
- **norm**: argument passed to the **nrmse** function
- **s**: argument passed to the **KGE** function
- **method**: argument passed to the **KGE** function
- **1Q.thr**: argument passed to the (optional) **pbiasfdc** function
- **hQ.thr**: argument passed to the (optional) **pbiasfdc** function
- **digits**: decimal places used for rounding the goodness-of-fit indexes.
- **...**: further arguments passed to or from other methods.

Value

The output of the **gof** function is a matrix with one column only, and the following rows:

- **me**: Mean Error
- **mae**: Mean Absolute Error
- **mse**: Mean Squared Error
- **rmse**: Root Mean Square Error
- **nrmse**: Normalized Root Mean Square Error ( -100% <= nrm <= 100% )
PBIAS  Percent Bias
pbiasfdc  PBIAS in the slope of the midsegment of the Flow Duration Curve
RSR  Ratio of RMSE to the Standard Deviation of the Observations, $\text{RSR} = \frac{\text{rms}}{\text{sd(obs)}}$. ($0 \leq \text{RSR} \leq +\text{Inf}$)
$rSD$  Ratio of Standard Deviations, $\text{rSD} = \frac{\text{sd(sim)}}{\text{sd(obs)}}$
NSE  Nash-Sutcliffe Efficiency ($-\text{Inf} \leq \text{NSE} \leq 1$)
mNSE  Modified Nash-Sutcliffe Efficiency
$rNSE$  Relative Nash-Sutcliffe Efficiency
d  Index of Agreement ($0 \leq d \leq 1$)
d1  Modified Index of Agreement
rd  Relative Index of Agreement
cp  Persistence Index ($0 \leq \text{PI} \leq 1$)
r  Pearson Correlation coefficient ($-1 \leq r \leq 1$)
r.Spearman  Spearman Correlation coefficient ($-1 \leq \text{r.Spearman} \leq 1$)
$R2$  Coefficient of Determination ($0 \leq R2 \leq 1$). Gives the proportion of the variance of one variable that is predictable from the other variable
brR2  R2 multiplied by the coefficient of the regression line between sim and obs ($0 \leq \text{bR2} \leq 1$)
KGE  Kling-Gupta efficiency between sim and obs ($0 \leq \text{KGE} \leq 1$)
VE  Volumetric efficiency between sim and obs ($-\text{Inf} \leq \text{VE} \leq 1$)

Note

obs and sim has to have the same length/dimension.

Missing values in obs and/or sim can be removed before the computations, depending on the value of na.rm.

Although $r$ and $R2$ have been widely used for model evaluation, these statistics are over-sensitive to outliers and insensitive to additive and proportional differences between model predictions and measured data (Legates and McCabe, 1999)

Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>
References


Krause P., Boyle D.P., and B"ase F., Comparison of different efficiency criteria for hydrological model assessment, Advances in Geosciences 5 (2005), pp. 89–97


See Also

me, mae, rmse, nrmse, pbias, pbiasfsc, rSD, NSE, mnNSE, rNSE, d, md, rd, cp, br2, KGE, VE

Examples

```r
sim <- 1:10
obs <- 1:10
gof(sim, obs)
```
Kling-Gupta Efficiency

Description

Kling-Gupta efficiency between sim and obs, with treatment of missing values.

This goodness-of-fit measure was developed by Gupta et al. (2009) to provide a diagnostically interesting decomposition of the Nash-Sutcliffe efficiency (and hence MSE), which facilitates the analysis of the relative importance of its different components (correlation, bias and variability) in the context of hydrological modelling.
Kling et al. (2012), proposed a revised version of this index, to ensure that the bias and variability ratios are not cross-correlated.

In the computation of this index, there are three main components involved:
1) \( r \) : the Pearson product-moment correlation coefficient. Ideal value is \( r=1 \)
2) \( \beta \) : the ratio between the mean of the simulated values and the mean of the observed ones. Ideal value is \( \beta=1 \)
3) \( vr \) : variability ratio, which could be computed using the standard deviation (\( \alpha \)) or the coefficient of variation (\( \gamma \)) of \( \text{sim} \) and \( \text{obs} \), depending on the value of \text{method}.

3.1) \( \alpha \) : the ratio between the standard deviation of the simulated values and the standard deviation of the observed ones. Ideal value is \( \alpha=1 \).
3.2) \( \gamma \) : the ratio between the coefficient of variation (\( \text{CV} \)) of the simulated values to the coefficient of variation of the observed ones. Ideal value is \( \gamma=1 \).

For a full discussion of the Kling-Gupta index, and its advantages over the Nash-Sutcliffe efficiency (\text{NSE}) see Gupta et al. (2009).

Usage

```R
KGE(sim, obs, ...) # Default S3 method:
KGE(sim, obs, s=c(1,1,1), na.rm=TRUE, method=c("2009", "2012"),
out.type=c("single", "full"), ...)
```

```R
KGE(sim, obs, s=c(1,1,1), na.rm=TRUE, method=c("2009", "2012"),
out.type=c("single", "full"), ...)
```

```R
KGE(sim, obs, s=c(1,1,1), na.rm=TRUE, method=c("2009", "2012"),
out.type=c("single", "full"), ...)
```

```R
KGE(sim, obs, s=c(1,1,1), na.rm=TRUE, method=c("2009", "2012"),
out.type=c("single", "full"), ...)
```

Arguments

- \( \text{sim} \) : numeric, zoo, matrix or data.frame with simulated values
- \( \text{obs} \) : numeric, zoo, matrix or data.frame with observed values
- \( s \) : numeric of length 3, representing the scaling factors to be used for re-scaling the criteria space before computing the Euclidean distance from the ideal point \( c(1,1,1) \), i.e., \( s \) elements are used for adjusting the emphasis on different components. The first elements is used for rescaling the Pearson product-moment correlation coefficient \( r \), the second element is used for rescaling \( \alpha \) and the third element is used for rescaling \( \beta \).
na.rm  a logical value indicating whether ‘NA’ should be stripped before the computation proceeds.

When an ‘NA’ value is found at the i-th position in obs OR sim, the i-th value of obs AND sim are removed before the computation.

method  character, indicating the formula used to compute the variability ratio in the Kling-Gupta efficiency. Valid values are:

- 2009: the variability is defined as ‘Alpha’, the ratio of the standard deviation of sim values to the standard deviation of obs. This is the default option. See Gupta et al. 2009

- 2012: the variability is defined as ‘Gamma’, the ratio of the coefficient of variation of sim values to the coefficient of variation of obs. See Kling et al. 2012.

out.type  character, indicating if the output of the function has to include or not each one of the three terms used in the computation of the Kling-Gupta efficiency. Valid values are:

- single: the output is a numeric with the Kling-Gupta efficiency only
- full: the output is a list of two elements: the first one with the Kling-Gupta efficiency, and the second is a numeric with 3 elements: the Pearson product-moment correlation coefficient (‘r’), the ratio between the mean of the simulated values to the mean of observations (‘Beta’), and the variability measure (‘Gamma’ or ‘Alpha’, depending on the value of method)

Details

\[ KGE = 1 - ED \]

\[ ED = \sqrt{(s[1] \times (r - 1))^2 + (s[2] \times (vr - 1))^2 + (s[3] \times (\beta - 1))^2} \]

\[ r = \text{Pearson product-moment correlation coefficient} \]

\[ \beta = \mu_s/\mu_o \]

\[ vr = \begin{cases} 
\alpha, & \text{method="2009"} \\
\gamma, & \text{method="2012"} 
\end{cases} \]

\[ \alpha = \sigma_s/\sigma_o \]

\[ \gamma = CV_s/CV_o = \sigma_s/\mu_s \]

Kling-Gupta efficiencies range from -Inf to 1. Essentially, the closer to 1, the more accurate the model is.
KGE

Value

If out.type=single: numeric with the Kling-Gupta efficiency between sim and obs. If sim and obs are matrices, the output value is a vector, with the Kling-Gupta efficiency between each column of sim and obs.

If out.type=full: a list of two elements:

- KGE.value numeric with the Kling-Gupta efficiency. If sim and obs are matrices, the output value is a vector, with the Kling-Gupta efficiency between each column of sim and obs.

- KGE.elements numeric with 3 elements: the Pearson product-moment correlation coefficient (‘r’), the ratio between the mean of the simulated values to the mean of observations (‘Beta’), and the variability measure (‘Gamma’ or ‘Alpha’, depending on the value of method). If sim and obs are matrices, the output value is a matrix, with the previous three elements computed for each column of sim and obs.

Note

obs and sim has to have the same length/dimension

The missing values in obs and sim are removed before the computation proceeds, and only those positions with non-missing values in obs and sim are considered in the computation.

Author(s)

Mauricio Zambrano-Bigiarini <mzb.devel@gmail.com>

References


See Also

NSE, gof, ggof

Examples

# Example1: basic ideal case
obs <- 1:10
sim <- 1:10
KGE(sim, obs)

obs <- 1:10
```r
sim <- 2:11
KGE(sim, obs)

# Example2: Looking at the difference between 'method=2009' and 'method=2012'
# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Simulated daily time series, initially equal to twice the observed values
sim <- 2*obs

# KGE 2009
KGE(sim=sim, obs=obs, method="2009", out.type="full")

# KGE 2012
KGE(sim=sim, obs=obs, method="2012", out.type="full")

# Example3: KGE for simulated values equal to observations plus random noise
# on the first half of the observed values
# Randomly changing the first 1826 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').
sim <- obs

# Computing the new 'KGE'
KGE(sim=sim, obs=obs)

# Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').

# Computing the new 'KGE'
KGE(sim=sim, obs=obs)
```

### mae

**Mean Absolute Error**

**Description**

Mean absolute error between sim and obs, in the same units of them, with treatment of missing values.

**Usage**

```r
mae(sim, obs, ...)
```

## Default S3 method:

```r
mae(sim, obs, na.rm=TRUE, ...)
```
## Arguments

- `sim` numeric, zoo, matrix or data.frame with simulated values
- `obs` numeric, zoo, matrix or data.frame with observed values
- `na.rm` a logical value indicating whether 'NA' should be stripped before the computation proceeds.
  When an 'NA' value is found at the i-th position in `obs` or `sim`, the i-th value of `obs` and `sim` are removed before the computation.
- `...` further arguments passed to or from other methods.

## Details

\[
\text{mae} = \frac{1}{N} \sum_{i=1}^{N} |S_i - O_i| 
\]

## Value

Mean absolute error between `sim` and `obs`.

If `sim` and `obs` are matrixes, the returned value is a vector, with the mean absolute error between each column of `sim` and `obs`.

## Note

`obs` and `sim` have to have the same length/dimension

The missing values in `obs` and `sim` are removed before the computation proceeds, and only those positions with non-missing values in `obs` and `sim` are considered in the computation.

## Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

## References

[http://en.wikipedia.org/wiki/Mean_absolute_error](http://en.wikipedia.org/wiki/Mean_absolute_error)
See Also
  me, gof, ggof

Examples

```r
obs <- 1:10
sim <- 1:10
mae(sim, obs)

obs <- 1:10
sim <- 2:11
mae(sim, obs)

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Generating a simulated daily time series, initially equal to the observed series
sim <- obs

# Computing the mean absolute error for the "best" case
mae(sim=sim, obs=obs)

# Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').

# Computing the new mean absolute error
mae(sim=sim, obs=obs)
```

### md

**Modified index of agreement**

**Description**

This function computes the modified Index of Agreement between `sim` and `obs`, with treatment of missing values.

If 'x' is a matrix or a data frame, a vector of the modified index of agreement among the columns is returned.

**Usage**

```r
md(sim, obs, ...)
```

## Default S3 method:

```r
md(sim, obs, j=1, na.rm=TRUE, ...)
```

## S3 method for class 'data.frame'

```r
md(sim, obs, j=1, na.rm=TRUE, ...)
```
md(sim, obs, j=1, na.rm=TRUE, ...)

## S3 method for class 'matrix'
md(sim, obs, j=1, na.rm=TRUE, ...)

## S3 method for class 'zoo'
md(sim, obs, j=1, na.rm=TRUE, ...)

### Arguments

- **sim**: numeric, zoo, matrix or data.frame with simulated values
- **obs**: numeric, zoo, matrix or data.frame with observed values
- **j**: numeric, with the exponent to be used in the computation of the modified index of agreement. The default value is j=1.
- **na.rm**: a logical value indicating whether 'NA' should be stripped before the computation proceeds. When an 'NA' value is found at the i-th position in obs OR sim, the i-th value of obs AND sim are removed before the computation.
- **...**: further arguments passed to or from other methods.

### Details

\[
md = 1 - \frac{\sum_{i=1}^{N} |O_i - S_i|^j}{\sum_{i=1}^{N} |S_i - \bar{O}| + |O_i - \bar{O}|^j}
\]

The Index of Agreement (d) developed by Willmott (1981) as a standardized measure of the degree of model prediction error and varies between 0 and 1. A value of 1 indicates a perfect match, and 0 indicates no agreement at all (Willmott, 1981).

The index of agreement can detect additive and proportional differences in the observed and simulated means and variances; however, it is overly sensitive to extreme values due to the squared differences (Legates and McCabe, 1999).

### Value

Modified index of agreement between sim and obs.

If sim and obs are matrixes, the returned value is a vector, with the modified index of agreement between each column of sim and obs.

### Note

obs and sim has to have the same length/dimension

The missing values in obs and sim are removed before the computation proceeds, and only those positions with non-missing values in obs and sim are considered in the computation.
Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

References


See Also
d, rd, gof, ggof

Examples

```r
obs <- 1:10
sim <- 1:10
md(sim, obs)

obs <- 1:10
sim <- 2:11
md(sim, obs)
```

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Generating a simulated daily time series, initially equal to the observed series
sim <- obs

# Computing the modified index of agreement for the "best" (unattainable) case
md(sim=sim, obs=obs)

# Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').
```
# Computing the new 'dl'
md(sim=sim, obs=obs)

---

```r
me Mean Error
```

### Description
Mean error between `sim` and `obs`, in the same units of them, with treatment of missing values.

### Usage
```
me(sim, obs, ...)
```

```r
## Default S3 method:
me(sim, obs, na.rm=TRUE, ...)
```

```r
## S3 method for class 'data.frame'
me(sim, obs, na.rm=TRUE, ...)
```

```r
## S3 method for class 'matrix'
me(sim, obs, na.rm=TRUE, ...)
```

```r
## S3 method for class 'zoo'
me(sim, obs, na.rm=TRUE, ...)
```

### Arguments
- `sim`: numeric, zoo, matrix or data.frame with simulated values
- `obs`: numeric, zoo, matrix or data.frame with observed values
- `na.rm`: a logical value indicating whether 'NA' should be stripped before the computation proceeds.
  When an 'NA' value is found at the i-th position in `obs` OR `sim`, the i-th value of `obs` AND `sim` are removed before the computation.
- `...`: further arguments passed to or from other methods.

### Details

\[
me = \frac{1}{N} \sum_{i=1}^{N} (S_i - O_i)
\]

### Value
Mean error between `sim` and `obs.

If `sim` and `obs` are matrixes, the returned value is a vector, with the mean error between each column of `sim` and `obs`. 

Note

obs and sim has to have the same length/dimension

The missing values in obs and sim are removed before the computation proceeds, and only those positions with non-missing values in obs and sim are considered in the computation.

Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

See Also

mae, gof, ggof

Examples

```r
obs <- 1:10
sim <- 1:10
me(sim, obs)

obs <- 1:10
sim <- 2:11
me(sim, obs)
```

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
```r
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts
```

# Generating a simulated daily time series, initially equal to the observed series
```r
sim <- obs
```

# Computing the mean error for the "best" case
```r
me(sim=sim, obs=obs)
```

# Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').
```r
```

# Computing the new mean error
```r
me(sim=sim, obs=obs)
```

---

**mNSE**

*Modified Nash-Sutcliffe efficiency*

Description

Modified Nash-Sutcliffe efficiency between sim and obs, with treatment of missing values.
**mNSE**

**Usage**

\[
mNSE(sim, obs, ...)\]

### Default S3 method:

\[
mNSE(sim, obs, j=1, na.rm=TRUE, ...)\]

### S3 method for class 'data.frame'

\[
mNSE(sim, obs, j=1, na.rm=TRUE, ...)\]

### S3 method for class 'matrix'

\[
mNSE(sim, obs, j=1, na.rm=TRUE, ...)\]

### S3 method for class 'zoo'

\[
mNSE(sim, obs, j=1, na.rm=TRUE, ...)\]

**Arguments**

- `sim`: numeric, zoo, matrix or data.frame with simulated values
- `obs`: numeric, zoo, matrix or data.frame with observed values
- `j`: numeric, with the exponent to be used in the computation of the modified Nash-Sutcliffe efficiency. The default value is \( j=1 \).
- `na.rm`: a logical value indicating whether 'NA' should be stripped before the computation proceeds.
  
  When an 'NA' value is found at the i-th position in `obs` OR `sim`, the i-th value of `obs` AND `sim` are removed before the computation.
  
  ... further arguments passed to or from other methods.

**Details**

\[
mNSE = 1 - \frac{\sum_{i=1}^{N} |S_i - O_i|^j}{\sum_{i=1}^{N} |O_i - O|^j}\]

When \( j=1 \), the modified NSeff is not inflated by the squared values of the differences, because the squares are replaced by absolute values.

**Value**

Modified Nash-Sutcliffe efficiency between `sim` and `obs`.

If `sim` and `obs` are matrixes, the returned value is a vector, with the modified Nash-Sutcliffe efficiency between each column of `sim` and `obs`.

**Note**

`obs` and `sim` has to have the same length/dimension
The missing values in obs and sim are removed before the computation proceeds, and only those positions with non-missing values in obs and sim are considered in the computation.

Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

References


See Also

NSE, rNSE, gof, ggof

Examples

```r
sim <- 1:10
obs <- 1:10
mNSE(sim, obs)

sim <- 2:11
obs <- 1:10
mNSE(sim, obs)

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Generating a simulated daily time series, initially equal to the observed series
sim <- obs

# Computing the 'mNSE' for the "best" (unattainable) case
mNSE(sim(sim), obs=obs)

# Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').

# Computing the new 'mNSE'
mNSE(sim, obs=obs)
```
### Description

Mean squared error between sim and obs, in the squared units of sim and obs, with treatment of missing values.

### Usage

```r
mse(sim, obs, ...)  
```

#### # Default S3 method:
```r
mse(sim, obs, na.rm=TRUE, ...)  
```

#### # S3 method for class 'data.frame'
```r
mse(sim, obs, na.rm=TRUE, ...)  
```

#### # S3 method for class 'matrix'
```r
mse(sim, obs, na.rm=TRUE, ...)  
```

#### # S3 method for class 'zoo'
```r
mse(sim, obs, na.rm=TRUE, ...)  
```

### Arguments

- **sim**: numeric, zoo, matrix or data.frame with simulated values
- **obs**: numeric, zoo, matrix or data.frame with observed values
- **na.rm**: a logical value indicating whether 'NA' should be stripped before the computation proceeds. When an 'NA' value is found at the i-th position in obs OR sim, the i-th value of obs AND sim are removed before the computation.
- **...**: further arguments passed to or from other methods.

### Details

\[
ms{e} = \frac{1}{N} \sum_{i=1}^{N} (S_i - O_i)^2
\]

### Value

Mean squared error between sim and obs.

If sim and obs are matrices, the returned value is a vector, with the mean squared error between each column of sim and obs.
Note

obs and sim has to have the same length/dimension

The missing values in obs and sim are removed before the computation proceeds, and only those positions with non-missing values in obs and sim are considered in the computation.

Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

References


See Also

mae, me, gof

Examples

```r
obs <- 1:10
sim <- 1:10
mse(sim, obs)

obs <- 1:10
sim <- 2:11
mse(sim, obs)

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Generating a simulated daily time series, initially equal to the observed series
sim <- obs

# Computing the mean squared error for the "best" case
mse(sim=sim, obs=obs)

# Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').

# Computing the new mean squared error
mse(sim=sim, obs=obs)
```
Normalized Root Mean Square Error

Description

Normalized root mean square error (NRMSE) between sim and obs, with treatment of missing values.

Usage

\[
nrmse(sim, obs, ...) \]

## Default S3 method:
nrmse(sim, obs, na.rm=TRUE, norm="sd", ...)

## S3 method for class 'data.frame'
nrmse(sim, obs, na.rm=TRUE, norm="sd", ...)

## S3 method for class 'matrix'
nrmse(sim, obs, na.rm=TRUE, norm="sd", ...)

## S3 method for class 'zoo'
nrmse(sim, obs, na.rm=TRUE, norm="sd", ...)

Arguments

sim numeric, zoo, matrix or data.frame with simulated values
obs numeric, zoo, matrix or data.frame with observed values
na.rm a logical value indicating whether 'NA' should be stripped before the computation proceeds.
When an 'NA' value is found at the i-th position in obs OR sim, the i-th value of obs AND sim are removed before the computation.
norm character, indicating the value to be used for normalising the root mean square error (RMSE). Valid values are:
- \(sd\): standard deviation of observations (default).
- \(maxmin\): difference between the maximum and minimum observed values
... further arguments passed to or from other methods.

Details

\[
nrmse = 100 \sqrt{\frac{1}{N} \sum_{i=1}^{N} (S_i - O_i)^2 } / nval
\]

\[
nval = \begin{cases} 
  sd(O_i) & \text{norm}="sd" \\
  O_{max} - O_{min} & \text{norm}="maxmin"
\end{cases}
\]
Value

Normalized root mean square error (nrmse) between sim and obs. The result is given in percentage (%).

If sim and obs are matrices, the returned value is a vector, with the normalized root mean square error between each column of sim and obs.

Note

obs and sim have to have the same length/dimension.

Missing values in obs and sim are removed before the computation proceeds, and only those positions with non-missing values in obs and sim are considered in the computation.

Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

See Also

rmse, ssq, gof, ggof

Examples

```r
obs <- 1:10
sim <- 1:10
nrmse(sim, obs)

obs <- 1:10
sim <- 2:11
nrmse(sim, obs)
```

Loading daily streamflows of the Ega River (Spain), from 1961 to 1970

data(EgaEnEstellaQts)

obs <- EgaEnEstellaQts

# Generating a simulated daily time series, initially equal to the observed series
sim <- obs

# Computing the normalized root mean squared error for the "best" (unattainable) case
nrmse(sim=sim, obs=obs)

# Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').

# Computing the new normalized root mean squared error
nrmse(sim=sim, obs=obs)
```
Description

Nash-Sutcliffe efficiency between \( \text{sim} \) and \( \text{obs} \), with treatment of missing values.

Usage

\[
\text{NSE}(\text{sim}, \text{obs}, \ldots)
\]

### Default S3 method:
\[
\text{NSE}(\text{sim}, \text{obs}, \text{na.rm=} \text{TRUE}, \text{FUN=} \text{NULL},
\quad \text{epsilon=} \text{c}(0, "\text{Pushpalatha2012}\), "\text{other}\), \text{epsilon.value=} \text{NA}, \ldots)
\]

### S3 method for class 'data.frame'
\[
\text{NSE}(\text{sim}, \text{obs}, \text{na.rm=} \text{TRUE}, \text{FUN=} \text{NULL},
\quad \text{epsilon=} \text{c}(0, "\text{Pushpalatha2012}\), "\text{other}\), \text{epsilon.value=} \text{NA}, \ldots)
\]

### S3 method for class 'matrix'
\[
\text{NSE}(\text{sim}, \text{obs}, \text{na.rm=} \text{TRUE}, \text{FUN=} \text{NULL},
\quad \text{epsilon=} \text{c}(0, "\text{Pushpalatha2012}\), "\text{other}\), \text{epsilon.value=} \text{NA}, \ldots)
\]

### S3 method for class 'zoo'
\[
\text{NSE}(\text{sim}, \text{obs}, \text{na.rm=} \text{TRUE}, \text{FUN=} \text{NULL},
\quad \text{epsilon=} \text{c}(0, "\text{Pushpalatha2012}\), "\text{other}\), \text{epsilon.value=} \text{NA}, \ldots)
\]

Arguments

- **sim** numeric, zoo, matrix or data.frame with simulated values
- **obs** numeric, zoo, matrix or data.frame with observed values
- **na.rm** a logical value indicating whether 'NA' should be stripped before the computation proceeds.
  When an 'NA' value is found at the \( i \)-th position in \( \text{obs} \) OR \( \text{sim} \), the \( i \)-th value of \( \text{obs} \) AND \( \text{sim} \) are removed before the computation.
- **FUN** function to be applied to \( \text{sim} \) and \( \text{obs} \) in order to obtain transformed values thereof before computing the Nash-Sutcliffe efficiency.
- **epsilon** argument used to define a numeric value to be added to both \( \text{sim} \) and \( \text{obs} \) before applying \( \text{FUN} \). It is was designed to allow the use of logarithm and other similar functions that do not work with zero values.
  Valid values are:
  1) \( 0 \): zero is added to both \( \text{sim} \) and \( \text{obs} \).
  2) "\text{Pushpalatha2012}" : one-hundredth of the mean observed values is added to both \( \text{sim} \) and \( \text{obs} \), as described in Pushpalatha et al., (2012).
  3) "\text{other}" : the numeric value defined in the \( \text{epsilon.value} \) argument is added to both \( \text{sim} \) and \( \text{obs} \).
Nash-Sutcliffe efficiency (NSE) is a normalized statistic that determines the relative magnitude of the residual variance ("noise") compared to the measured data variance ("information") (Nash and Sutcliffe, 1970).

\[
NSE = 1 - \frac{\sum_{i=1}^{N} (S_i - O_i)^2}{\sum_{i=1}^{N} (O_i - \bar{O})^2}
\]

The Nash-Sutcliffe efficiency (NSE) is a normalized statistic that determines the relative magnitude of the residual variance ("noise") compared to the measured data variance ("information") (Nash and Sutcliffe, 1970).

Nash-Sutcliffe efficiencies range from -Inf to 1. Essentially, the closer to 1, the more accurate the model is.
- NSE = 1, corresponds to a perfect match of modelled to the observed data.
- NSE = 0, indicates that the model predictions are as accurate as the mean of the observed data,
- -Inf < NSE < 0, indicates that the observed mean is better predictor than the model.

Value

Nash-Sutcliffe efficiency between sim and obs.

If sim and obs are matrixes, the returned value is a vector, with the Nash-Sutcliffe efficiency between each column of sim and obs.

Note

obs and sim has to have the same length/dimension

The missing values in obs and sim are removed before the computation proceeds, and only those positions with non-missing values in obs and sim are considered in the computation.

Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

References


http://en.wikipedia.org/wiki/Nash%E2%80%93Sutcliffe_model_efficiency_coefficient

Percent Bias between `sim` and `obs`, with treatment of missing values.

Usage

\`pbias(sim, obs, ...)\`

## Default S3 method:
pbias(sim, obs, na.rm=TRUE, ...)

## S3 method for class 'data.frame'
pbias(sim, obs, na.rm=TRUE, ...)

## S3 method for class 'matrix'
pbias(sim, obs, na.rm=TRUE, ...)

## S3 method for class 'zoo'
pbias(sim, obs, na.rm=TRUE, ...)

### Arguments

<table>
<thead>
<tr>
<th>sim</th>
<th>numeric, zoo, matrix or data.frame with simulated values</th>
</tr>
</thead>
<tbody>
<tr>
<td>obs</td>
<td>numeric, zoo, matrix or data.frame with observed values</td>
</tr>
<tr>
<td>na.rm</td>
<td>a logical value indicating whether 'NA' should be stripped before the computation proceeds. When an 'NA' value is found at the i-th position in obs OR sim, the i-th value of obs AND sim are removed before the computation.</td>
</tr>
</tbody>
</table>

... further arguments passed to or from other methods.

### Details

\[
PBIAS = 100 \frac{\sum_{i=1}^{N} (S_i - O_i)}{\sum_{i=1}^{N} O_i}
\]

Percent bias (PBIAS) measures the average tendency of the simulated values to be larger or smaller than their observed ones.

The optimal value of PBIAS is 0.0, with low-magnitude values indicating accurate model simulation. Positive values indicate overestimation bias, whereas negative values indicate model underestimation bias.

### Value

Percent bias between sim and obs. The result is given in percentage (%)

If sim and obs are matrixes, the returned value is a vector, with the percent bias between each column of sim and obs.

### Note

obs and sim has to have the same length/dimension

The missing values in obs and sim are removed before the computation proceeds, and only those positions with non-missing values in obs and sim are considered in the computation.
pbiasfdc

Author(s)
Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

References


See Also
gof, ggof

Examples
obs <- 1:10
sim <- 1:10
pbias(sim, obs)

obs <- 1:10
sim <- 2:11
pbias(sim, obs)

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Generating a simulated daily time series, initially equal to the observed series
sim <- obs

# Computing the 'pbias' for the "best" case
pbias(sim, obs)

# Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').

# Computing the new 'pbias'
pbias(sim, obs)

pbiasfdc  Percent Bias in the Slope of the Midsegment of the Flow Duration Curve
Description

Percent Bias in the slope of the midsegment of the flow duration curve (FDC) [%]. It is related to the vertical soil moisture redistribution.

Usage

pbiasfdc(sim, obs, ...)

## Default S3 method:
pbiasfdc(sim, obs, lQ.thr=0.7, hQ.thr=0.2, na.rm=TRUE, 
plot=TRUE, verbose=FALSE, ...)

## S3 method for class 'data.frame'
pbiasfdc(sim, obs, lQ.thr=0.7, hQ.thr=0.2, na.rm=TRUE, 
plot=TRUE, verbose=FALSE, ...)

## S3 method for class 'matrix'
pbiasfdc(sim, obs, lQ.thr=0.7, hQ.thr=0.2, na.rm=TRUE, 
plot=TRUE, verbose=FALSE, ...)

## S3 method for class 'zoo'
pbiasfdc(sim, obs, lQ.thr=0.7, hQ.thr=0.2, na.rm=TRUE, 
plot=TRUE, verbose=FALSE, ...)

Arguments

sim numeric, zoo, matrix or data.frame with simulated values
obs numeric, zoo, matrix or data.frame with observed values
lQ.thr numeric, used to classify low flows. All the streamflows with a probability of exceedence larger or equal to lQ.thr are classified as low flows
hQ.thr numeric, used to classify high flows. All the streamflows with a probability of exceedence larger or equal to hQ.thr are classified as high flows
na.rm a logical value indicating whether 'NA' values should be stripped before the computation proceeds.
plot a logical value indicating if the flow duration curves corresponding to obs and sim have to be plotted or not.
verbose logical; if TRUE, progress messages are printed
... further arguments passed to the fdc function of the hydroTSM package or from other methods.

Value

Percent Bias in the slope of the midsegment of the flow duration curve, between sim and obs.

If sim and obs are matrixes, the returned value is a vector, with the Percent Bias in the slope of the midsegment of the flow duration curve, between each column of sim and obs.
Note

The result is given in percentage (%).

It requires the **hydroTSM** package.

Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

References


See Also

**fdc, gof**

Examples

```r
## Not run:
sim <- 1:10
obs <- 1:10
pbiasfdc(sim, obs)

sim <- 2:11
obs <- 1:10
pbiasfdc(sim, obs)

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Generating a simulated daily time series, initially equal to the observed series
sim <- obs

# Computing the relative index of agreement for the "best" (unattainable) case
pbiasfdc(sim=sim, obs=obs)

# Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm')

# Computing the new relative index of agreement
pbiasfdc(sim=sim, obs=obs, col=c("black", "blue"))
```
**pfactor**

### Description

*A* factor is the percent of observations that are within the given uncertainty bounds.

Ideally, i.e., with a combination of model structure and parameter values that perfectly represents the catchment under study, and in absence of measurement errors and other additional sources of uncertainty, all the simulated values should be in a perfect match with the observations, leading to a *P*-factor equal to 1, and an *R*-factor equal to zero. However, in real-world applications we aim at encompassing as much observations as possible within the given uncertainty bounds (*P*-factor close to 1) while keeping the width of the uncertainty bounds as small as possible (*R*-factor close to 0), in order to avoid obtaining a good bracketing of observations at expense of uncertainty bounds too wide to be informative for the decision-making process.

### Usage

```r
pfactor(x, ...)
```

#### Default S3 method:

```r
pfactor(x, lband, uband, na.rm=TRUE, ...)
```

#### S3 method for class 'data.frame'

```r
pfactor(x, lband, uband, na.rm=TRUE, ...)
```

#### S3 method for class 'matrix'

```r
pfactor(x, lband, uband, na.rm=TRUE, ...)
```

### Arguments

- **x**: ts or zoo object with the observed values.
- **lband**: numeric, ts or zoo object with the values of the lower uncertainty bound
- **uband**: numeric, ts or zoo object with the values of the upper uncertainty bound
- **na.rm**: a logical value indicating whether 'NA' values should be stripped before the computation proceeds.
- **...**: further arguments passed to or from other methods.

### Value

Percent of the *x* observations that are within the given uncertainty bounds given by *lband* and *uband*.

If *sim* and *obs* are matrixes, the returned value is a vector, with the *P*-factor between each column of *sim* and *obs*. 
Note

So far, the argument na.rm is not being taken into account.

Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

References


See Also

rfactor, plotbands

Examples

```r
x <- 1:10
lband <- x - 0.1
uband <- x + 0.1
pfactor(x, lband, uband)

lband <- x - rnorm(10)
uband <- x + rnorm(10)
pfactor(x, lband, uband)
```

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Selecting only the daily values belonging to the year 1961
obs <- window(obs, end=as.Date("1961-12-31"))

# Generating the lower and upper uncertainty bounds, centred at the observations
```r
lband <- obs - 5
uband <- obs + 5

pfactor(obs, lband, uband)

# Randomly generating the lower and upper uncertainty bounds
uband <- obs + rnorm(length(obs))
lband <- obs - rnorm(length(obs))

pfactor(obs, lband, uband)
```

---

**plot2**

*Plotting 2 Time Series*

**Description**

Plotting of 2 time series, in two different vertical windows or overlapped in the same window. It requires the *hydroTSM* package.

**Usage**

```r
plot2(x, y, plot.type = "multiple",
      tick.tstep = "auto", lab.tstep = "auto", lab.fmt=NULL,
      main, xlab = "Time", ylab,
      cal.ini=NA, val.ini=NA, date.fmt="%Y-%m-%d",
      gof.leg = FALSE, gof.digits=2,
      goes=c("ME", "MAE", "RMSE", "NRMSE", "PBIAS", "RSR", "r SD", "NSE", "mNSE",
             "rNSE", "d", "md", "rd", "r", "R2", "bR2", "KGE", "VE"),
      legend, leg.cex = 1,
      col = c("black", "blue"),
      cex = c(0.5, 0.5), cex.axis=1.2, cex.lab=1.2,
      lwd= c(1,1), lty=c(1,3), pch = c(1, 9),
      pt.style = "ts", add = FALSE,
      ...)
```

**Arguments**

- **x**
  - Time series that will be plotted. class(x) must be ts or zoo. If leg.gof=TRUE, then x is considered as **simulated** (for some goodness-of-fit functions this is important)

- **y**
  - Time series that will be plotted. class(x) must be ts or zoo. If leg.gof=TRUE, then y is considered as **observed** values (for some goodness-of-fit functions this is important)

- **plot.type**
  - Character, indicating if the 2 ts have to be plotted in the same window or in two different vertical ones. Valid values are:
    - single: (default) superimposes the 2 ts on a single plot
    - multiple: plots the 2 series on 2 multiple vertical plots
tick.tstep character, indicating the time step that have to be used for putting the ticks on
the time axis. Valid values are: auto, years, months, weeks, days, hours, minutes, seconds.

lab.tstep character, indicating the time step that have to be used for putting the labels
on the time axis. Valid values are: auto, years, months, weeks, days, hours, minutes, seconds.

lab.fmt Character indicating the format to be used for the label of the axis. See lab.fmt in
drawTimeAxis.

main an overall title for the plot: see title

xlab label for the 'x' axis

ylab label for the 'y' axis

cal.ini OPTIONAL. Character, indicating the date in which the calibration period started.
When cal.ini is provided, all the values in obs and sim with dates previous
to cal.ini are SKIPPED from the computation of the goodness-of-fit measures
(when gof.leg=TRUE), but their values are still plotted, in order to examine if
the warming up period was too short, acceptable or too long for the chosen cal-
ibration period. In addition, a vertical red line in drawn at this date.

date.fmt OPTIONAL. Character indicating the format in which the dates entered are
stored in cal.ini and val.ini. Default value is %Y-%m-%d. ONLY required
when cal.ini or val.ini is provided.

gof.leg logical, indicating if several numerical goodness-of-fit values have to be com-
puted between sim and obs, and plotted as a legend on the graph. If gof.leg=TRUE
(default value), then x is considered as observed and y as simulated values
(for some gof functions this is important). This legend is ONLY plotted when
plot.type="single"

gof.digits OPTIONAL, only used when gof.leg=TRUE. Decimal places used for rounding
the goodness-of-fit indexes.

gofs character, with one or more strings indicating the goodness-of-fit measures to be
shown in the legend of the plot when gof.leg=TRUE.

Possible values are in c("ME", "MAE", "MSE", "RMSE", "NRMSE", "PBIAS", "RSR", "rSD", "NSE", "KGE", "VEI"

legend vector of length 2 to appear in the legend.

leg.cex numeric, indicating the character expansion factor *relative* to current 'par("cex")'.

Used for text, and provides the default for 'pt.cex' and 'title.cex'. Default value = 1

So far, it controls the expansion factor of the 'GoF' legend and the legend refer-
ing to x and y

col character, with the colors of x and y

cex numeric, with the values controlling the size of text and symbols of x and y with
respect to the default

cex.axis numeric, with the magnification of axis annotation relative to 'cex'. See par.
cex.lab numeric, with the magnification to be used for x and y labels relative to the
current setting of 'cex'. See par.
lwd vector with the line width of x and y
lty vector with the line type of x and y
pch vector with the type of symbol for x and y. (e.g.: 1: white circle; 9: white rhombus with a cross inside)
pt.style Character, indicating if the 2 ts have to be plotted as lines or bars. Valid values are:
-> ts: (default) each ts is plotted as a line along the x axis
-> bar: the 2 series are plotted as a barplot.
add logical indicating if other plots will be added in further calls to this function.
-> FALSE => the plot and the legend are plotted on the same graph
-> TRUE => the legend is plotted in a new graph, usually when called from another function (e.g.: ggoF)
...
... further arguments passed to plot.Zoo function for plotting x, or from other methods

Note

It requires the package hydroTSM.

Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

See Also

ggoF

Examples

sim <- 2:11
obs <- 1:30
## Not run:
plot2(sim, obs)

## End(Not run)

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Generating a simulated daily time series, initially equal to the observed series
sim <- obs

# Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').

# Plotting 'sim' and 'obs' in 2 separate panels
plot2(x=obs, y=sim)
# Plotting 'sim' and 'obs' in the same window
plot2(x=obs, y=sim, plot.type="single")

plotbands  

Plot a ts with observed values and two confidence bounds

Description

It plots a ts with observed values and two confidence bounds. Optionally can also add a simulated time series, in order to be compared with 'x'.

Usage

plotbands(x, lband, uband, sim,  
dates, date.fmt="%Y-%m-%d",  
gof.leg= TRUE, gof.digits=2,  
legend=c("Obs", "Sim", "95PPU"), leg.cex=1,  
bands.col="lightblue", border= NA,  
tick.tstep= "auto", lab.tstep= "auto", lab.fmt=NULL,  
cal.ini=NA, val.ini=NA,  
main="Confidence Bounds for 'x'",  
xlab="Time", ylab="Q, [m3/s]", ylim,  
col=c("black", "blue"), type= c("lines", "lines"),  
cex= c(0.5, 0.5), cex.axis=1.2, cex.lab=1.2,  
lwd=c(0.6, 1), lty=c(3, 4), pch=c(1,9), ...)

Arguments

x        zoo or xts object with the observed values.

lband    zoo or xts object with the values of the lower band.

uband    zoo or xts object with the values of the upper band.

sim      OPTIONAL. zoo or xts object with the simulated values.

dates    OPTIONAL. Date, factor, or character object indicating the dates that will be assigned to x, lband, uband, and sim (when provided). If dates is a factor or character vector, its values are converted to dates using the date format specified by date.fmt.
When x, lband, uband, and sim are already of zoo class, the values provided by dates over-write the original dates of the objects.

date.fmt OPTIONAL. Character indicating the format in which the dates entered are stored in cal.ini and val.ini. See format in as.Date.
Default value is %Y-%m-%d
ONLY required when cal.ini, val.ini or dates is provided.

gof.leg  logical indicating if the p-factor and r-factor have to be computed and plotted as legends on the graph.
plotbands

- gof.digits: OPTIONAL, numeric. Only used when gof.leg=TRUE. Decimal places used for rounding the goodness-of-fit indexes.

- legend: OPTIONAL. logical or character vector of length 3 with the strings that will be used for the legend of the plot.
  - When legend is a character vector, the first element is used for labelling the observed series, the second for labelling the simulated series and the third one for the predictive uncertainty bounds. Default value is c("obs", "sim", "95PPU").
  - When legend=FALSE, the legend is not drawn.

- leg.cex: OPTIONAL. numeric. Used for the GoF legend. Character expansion factor *relative* to current 'par("cex")'. Used for text, and provides the default for 'pt.cex' and 'title.cex'. Default value is 1.

- bands.col: See polygon. Color to be used for filling the area between the lower and upper uncertainty bound.

- border: See polygon. The color to draw the border. The default, 'NULL', means to use 'par("fg")'. Use 'border = NA' to omit borders.

- tick.tstep: character, indicating the time step that have to be used for putting the ticks on the time axis. Valid values are: auto, years, months, weeks, days, hours, minutes, seconds.

- lab.tstep: character, indicating the time step that have to be used for putting the labels on the time axis. Valid values are: auto, years, months, weeks, days, hours, minutes, seconds.

- lab.fmt: Character indicating the format to be used for the label of the axis. See lab.fmt in drawTimeAxis.

- cal.ini: OPTIONAL. Character with the date in which the calibration period started. ONLY used for drawing a vertical red line at this date.

- val.ini: OPTIONAL. Character with the date in which the validation period started. ONLY used for drawing a vertical red line at this date.

- main: an overall title for the plot: see 'title'

- xlab: a title for the x axis: see 'title'

- ylab: a title for the y axis: see 'title'

- ylim: the y limits of the plot. See plot.default.

- col: colors to be used for plotting the x and y series.

- type: character. Indicates if the observed and simulated series have to be plotted as lines or points. Possible values are:
  - lines: the observed/simulated series are plotted as lines
  - points: the observed/simulated series are plotted as points

- cex: See code plot.default. A numerical vector giving the amount by which plotting characters and symbols should be scaled relative to the default. This works as a multiple of 'par("cex")'. 'NULL' and 'NA' are equivalent to '1.0'. Note that this does not affect annotation.

- cex.axis: magnification of axis annotation relative to 'cex'.

- cex.lab: Magnification to be used for x and y labels relative to the current setting of 'cex'. See '?par'.
plotbands

lwd
See plot.default. The line width, see 'par'.
lty
See plot.default. The line type, see 'par'.
pch
numeric, with the type of symbol for x and y. (e.g.: 1: white circle; 9: white rhombus with a cross inside)
...

Note
It requires the hydroTSM package

Author(s)
Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

See Also
pfactor, rfactor

Examples

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Selecting only the daily values belonging to the year 1961
obs <- window(obs, end=as.Date("1961-12-31"))

# Generating the lower and upper uncertainty bounds
lband <- obs - 5
uband <- obs + 5

## Not run:
plotbands(obs, lband, uband)

## End(Not run)

# Randomly generating a simulated time series
sim <- obs + rnorm(length(obs), mean=3)

## Not run:
plotbands(obs, lband, uband, sim)

## End(Not run)
plotbandsonly

Adds uncertainty bounds to an existing plot.

Description

Adds a polygon representing uncertainty bounds to an existing plot.

Usage

plotbandsonly(lband, uband, dates, date.fmt="%Y-%m-%d",
    bands.col="lightblue", border = NA, ...)

Arguments

lband
  zoo or xts object with the values of the lower band.
uband
  zoo or xts object with the values of the upper band.
dates
  OPTIONAL. Date, factor, or character object indicating the dates that will be
  assigned to lband and uband.
  If dates is a factor or character vector, its values are converted to dates using
  the date format specified by date.fmt.
  When lband and uband are already of zoo class, the values provided by dates
  over-write the original dates of the objects.
date.fmt
  OPTIONAL. Character indicating the format of dates. See format in as.Date.
bands.col
  See polygon. Color to be used for filling the area between the lower and upper
  uncertainty bound.
border
  See polygon. The color to draw the border. The default, 'NULL', means to use
  'par("fg")'. Use 'border = NA' to omit borders.
...
  further arguments passed to the codepolygon function for plotting the bands, or
  from other methods

Note

It requires the hydroTSM package

Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

See Also

pfactor, rfactor
Examples

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Selecting only the daily values belonging to the year 1961
obs <- window(obs, end=as.Date("1961-12-31"))

# Generating the lower and upper uncertainty bounds
lband <- obs - 5
uband <- obs + 5

## Not run:
plot(obs, type="n")
plotbandsonly(lband, uband)
points(obs, col="blue", cex=0.6, type="o")

## End(Not run)

---

rd

Relative Index of Agreement

Description

This function computes the Relative Index of Agreement (d) between sim and obs, with treatment of missing values. If x is a matrix or a data frame, a vector of the relative index of agreement among the columns is returned.

Usage

rd(sim, obs, ...)

## Default S3 method:
rd(sim, obs, na.rm=TRUE, ...)

## S3 method for class 'data.frame'
rd(sim, obs, na.rm=TRUE, ...)

## S3 method for class 'matrix'
rd(sim, obs, na.rm=TRUE, ...)

## S3 method for class 'zoo'
rd(sim, obs, na.rm=TRUE, ...)
Arguments

sim numeric, zoo, matrix or data.frame with simulated values
obs numeric, zoo, matrix or data.frame with observed values
na.rm a logical value indicating whether 'NA' should be stripped before the computation proceeds.

When an 'NA' value is found at the i-th position in obs OR sim, the i-th value of obs AND sim are removed before the computation.

... further arguments passed to or from other methods.

Details

\[
rd = 1 - \frac{\sum_{i=1}^{N} \left( \frac{O_i - S_i}{O_i} \right)^2}{\sum_{i=1}^{N} \left( \frac{|S_i - \bar{O}| + |O_i - \bar{O}|}{\bar{O}} \right)^2}
\]

It varies between 0 and 1. A value of 1 indicates a perfect match, and 0 indicates no agreement at all.

Value

Relative index of agreement between sim and obs.

If sim and obs are matrixes, the returned value is a vector, with the relative index of agreement between each column of sim and obs.

Note

obs and sim has to have the same length/dimension

The missing values in obs and sim are removed before the computation proceeds, and only those positions with non-missing values in obs and sim are considered in the computation.

If some of the observed values are equal to zero (at least one of them), this index can not be computed.

Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

References


See Also
d, md, gof, ggof

Examples

```r
obs <- 1:10
sim <- 1:10
rd(sim, obs)

obs <- 1:10
sim <- 2:11
rd(sim, obs)

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Generating a simulated daily time series, initially equal to the observed series
sim <- obs

# Computing the relative index of agreement for the "best" (unattainable) case
rd(sim=sim, obs=obs)

# Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').

# Computing the new relative index of agreement
rd(sim=sim, obs=obs)
```
R-factor represents the average width of the given uncertainty bounds divided by the standard deviation of the observations.

Ideally, i.e., with a combination of model structure and parameter values that perfectly represents the catchment under study, and in absence of measurement errors and other additional sources of uncertainty, all the simulated values should be in a perfect match with the observations, leading to a P-factor equal to 1, and an R-factor equal to zero. However, in real-world applications we aim at encompassing as much observations as possible within the given uncertainty bounds (P-factor close to 1) while keeping the width of the uncertainty bounds as small as possible (R-factor close to 0), in order to avoid obtaining a good bracketing of observations at expense of uncertainty bounds too wide to be informative for the decision-making process.

Usage

rfactor(x, ...)

## Default S3 method:
rfactor(x, lband, uband, na.rm=TRUE, ...)

## S3 method for class 'data.frame'
rfactor(x, lband, uband, na.rm=TRUE, ...)

## S3 method for class 'matrix'
rfactor(x, lband, uband, na.rm=TRUE, ...)

Arguments

x ts or zoo object with the observed values.

lband numeric, ts or zoo object with the values of the lower uncertainty bound

uband numeric, ts or zoo object with the values of the upper uncertainty bound

na.rm logical value indicating whether 'NA' values should be stripped before the computation proceeds.

... further arguments passed to or from other methods.

Value

Average width of the given uncertainty bounds, given by lband and uband, divided by the standard deviation of the observations x

If sim and obs are matrixes, the returned value is a vector, with the R-factor between each column of sim and obs.
Note

So far, the argument `na.rm` is not being taken into account.

Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

References


See Also

`pfactor`, `plotbands`

Examples

```r
x <- 1:10
lband <- x - 0.1
uband <- x + 0.1
rfactor(x, lband, uband)

lband <- x - rnorm(10)
uband <- x + rnorm(10)
rfactor(x, lband, uband)
```

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Selecting only the daily values belonging to the year 1961
obs <- window(obs, end=as.Date("1961-12-31"))

# Generating the lower and upper uncertainty bounds, centred at the observations
lm <- obs - 5
um <- obs + 5
rfactor(obs, lm, um)

# Randomly generating the lower and upper uncertainty bounds
um <- obs + rnorm(length(obs))
lm <- obs - rnorm(length(obs))
rfactor(obs, lm, um)

---

**rmse**  
*Root Mean Square Error*

**Description**

Root Mean Square Error (RMSE) between `sim` and `obs`, in the same units of `sim` and `obs`, with treatment of missing values. RMSE gives the standard deviation of the model prediction error. A smaller value indicates better model performance.

**Usage**

```r
rmse(sim, obs, ...)
```

```
## Default S3 method:
rmse(sim, obs, na.rm=TRUE, ...)

## S3 method for class 'data.frame'
rmse(sim, obs, na.rm=TRUE, ...)

## S3 method for class 'matrix'
rmse(sim, obs, na.rm=TRUE, ...)

## S3 method for class 'zoo'
rmse(sim, obs, na.rm=TRUE, ...)
```

**Arguments**

- `sim` numeric, zoo, matrix or data.frame with simulated values
- `obs` numeric, zoo, matrix or data.frame with observed values
- `na.rm` a logical value indicating whether 'NA' should be stripped before the computation proceeds.
  When an 'NA' value is found at the i-th position in `obs` OR `sim`, the i-th value of `obs` AND `sim` are removed before the computation.
- `...` further arguments passed to or from other methods.
Details

\[ \text{rmse} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (S_i - O_i)^2} \]

Value

Root mean square error (rmse) between sim and obs.

If sim and obs are matrixes, the returned value is a vector, with the RMSE between each column of sim and obs.

Note

obs and sim has to have the same length/dimension

The missing values in obs and sim are removed before the computation proceeds, and only those positions with non-missing values in obs and sim are considered in the computation.

Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

References

http://en.wikipedia.org/wiki/Root_mean_square_deviation

See Also

nrmse, ssq, gof, ggof

Examples

```r
obs <- 1:10
sim <- 1:10
rmse(sim, obs)

obs <- 1:10
sim <- 2:11
rmse(sim, obs)
```

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970

data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Generating a simulated daily time series, initially equal to the observed series
sim <- obs
# Computing the root mean squared error for the "best" (unattainable) case
rmse(sim=sim, obs=obs)

# Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').

# Computing the new root mean squared error
rmse(sim=sim, obs=obs)

---

## rNSE

**Relative Nash-Sutcliffe efficiency**

### Description

Relative Nash-Sutcliffe efficiency between sim and obs, with treatment of missing values.

### Usage

```r
rNSE(sim, obs, ...)
```

**## Default S3 method:**

```r
rNSE(sim, obs, na.rm=TRUE, ...)
```

**## S3 method for class 'data.frame'

```r
rNSE(sim, obs, na.rm=TRUE, ...)
```

**## S3 method for class 'matrix'

```r
rNSE(sim, obs, na.rm=TRUE, ...)
```

**## S3 method for class 'zoo'

```r
rNSE(sim, obs, na.rm=TRUE, ...)
```

### Arguments

- **sim**: numeric, zoo, matrix or data.frame with simulated values
- **obs**: numeric, zoo, matrix or data.frame with observed values
- **na.rm**: a logical value indicating whether 'NA' should be stripped before the computation proceeds. When an 'NA' value is found at the i-th position in obs OR sim, the i-th value of obs AND sim are removed before the computation.
- **...**: further arguments passed to or from other methods.
Details

\[ rNSE = 1 - \frac{\sum_{i=1}^{N} (S_i - O_i)^2}{\sum_{i=1}^{N} (O_i - \bar{O})^2} \]

Value

Relative Nash-Sutcliffe efficiency between \texttt{sim} and \texttt{obs}.

If \texttt{sim} and \texttt{obs} are matrices, the returned value is a vector, with the relative Nash-Sutcliffe efficiency between each column of \texttt{sim} and \texttt{obs}.

Note

\texttt{obs} and \texttt{sim} has to have the same length/dimension.

The missing values in \texttt{obs} and \texttt{sim} are removed before the computation proceeds, and only those positions with non-missing values in \texttt{obs} and \texttt{sim} are considered in the computation.

If some of the observed values are equal to zero (at least one of them), this index can not be computed.

Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

References


See Also

\texttt{NSE}, \texttt{mNSE}, \texttt{gof}, \texttt{ggof}

Examples

```r
sim <- 1:10
obs <- 1:10
rNSE(sim, obs)

sim <- 2:11
obs <- 1:10
rNSE(sim, obs)
```

#---------------------
# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Generating a simulated daily time series, initially equal to the observed series
sim <- obs

# Computing the 'rNSE' for the "best" (unattainable) case
rNSE(sim, obs)

# Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').

# Computing the new 'rNSE'
rNSE(sim, obs)

rPearson

Mean Squared Error

Description

Correlation of sim and obs if these are vectors, with treatment of missing values. If sim and obs are matrices then the covariances (or correlations) between the columns of sim and the columns of obs are computed. It is a wrapper to the cor function.

Usage

rPearson(sim, obs, ...)

## Default S3 method:
rPearson(sim, obs, ...)

## S3 method for class 'matrix'
rPearson(sim, obs, na.rm=TRUE, ...)

## S3 method for class 'data.frame'
rPearson(sim, obs, na.rm=TRUE, ...)

## S3 method for class 'zoo'
rPearson(sim, obs, na.rm=TRUE, ...)

Arguments

- **sim**: numeric, zoo, matrix or data.frame with simulated values
- **obs**: numeric, zoo, matrix or data.frame with observed values
na.rm a logical value indicating whether 'NA' should be stripped before the computation proceeds. When an 'NA' value is found at the i-th position in obs OR sim, the i-th value of obs AND sim are removed before the computation.

... further arguments passed to or from other methods.

Details
It is a wrapper to the cor function.

Value
Mean squared error between sim and obs.

If sim and obs are matrixes, the returned value is a vector, with the mean squared error between each column of sim and obs.

Note
obs and sim has to have the same length/dimension

The missing values in obs and sim are removed before the computation proceeds, and only those positions with non-missing values in obs and sim are considered in the computation

Author(s)
Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

See Also
cor

Examples
obs <- 1:10
sim <- 1:10
rPearson(sim, obs)

obs <- 1:10
sim <- 2:11
rPearson(sim, obs)

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Generating a simulated daily time series, initially equal to the observed series
sim <- obs
# Computing the linear correlation for the "best" case
rPearson(sim=sim, obs=obs)

# Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').

# Computing the new correlation value
rPearson(sim=sim, obs=obs)

---

**rSD**  
*Ratio of Standard Deviations*

**Description**

Ratio of standard deviations between *sim* and *obs*, with treatment of missing values.

**Usage**

```r
rSD(sim, obs, ...)  
```

- **# Default S3 method:**
  ```r
  rSD(sim, obs, na.rm=TRUE, ...)
  ```

- **# S3 method for class 'data.frame'**
  ```r
  rSD(sim, obs, na.rm=TRUE, ...)
  ```

- **# S3 method for class 'matrix'**
  ```r
  rSD(sim, obs, na.rm=TRUE, ...)
  ```

- **# S3 method for class 'zoo'**
  ```r
  rSD(sim, obs, na.rm=TRUE, ...)
  ```

**Arguments**

- **sim**  
  numeric, zoo, matrix or data.frame with simulated values

- **obs**  
  numeric, zoo, matrix or data.frame with observed values

- **na.rm**  
  a logical value indicating whether 'NA' should be stripped before the computation proceeds.  
  When an 'NA' value is found at the i-th position in *obs* **OR** *sim*, the i-th value of *obs* **AND** *sim* are removed before the computation.

- **...**  
  further arguments passed to or from other methods.
Value

Ratio of standard deviations between `sim` and `obs`.

If `sim` and `obs` are matrixes, the returned value is a vector, with the ratio of standard deviations between each column of `sim` and `obs`.

Note

`obs` and `sim` has to have the same length/dimension

The missing values in `obs` and `sim` are removed before the computation proceeds, and only those positions with non-missing values in `obs` and `sim` are considered in the computation.

Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

See Also

`sd`, `rser`, `gof`, `ggof`

Examples

```r
sim <- 1:10
obs <- 1:10
rSD(sim, obs)

sim <- 2:11
obs <- 1:10
rSD(sim, obs)
```

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Generating a simulated daily time series, initially equal to the observed series
sim <- obs

# Computing the 'rSD' for the "best" (unattainable) case
rSD(sim=sim, obs=obs)

# Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').

# Computing the new 'rSD'
rSD(sim=sim, obs=obs)
```
**rsr**  
*Ratio of RMSE to the standard deviation of the observations*

**Description**

Ratio of the RMSE between simulated and observed values to the standard deviation of the observations.

**Usage**

```r
rsr(sim, obs, ...)  
```

## Default S3 method:
```r
rsr(sim, obs, na.rm=TRUE, ...)
```

## S3 method for class 'data.frame'
```r
rsr(sim, obs, na.rm=TRUE, ...)
```

## S3 method for class 'matrix'
```r
rsr(sim, obs, na.rm=TRUE, ...)
```

## S3 method for class 'zoo'
```r
rsr(sim, obs, na.rm=TRUE, ...)
```

**Arguments**

- `sim` numeric, zoo, matrix or data.frame with simulated values
- `obs` numeric, zoo, matrix or data.frame with observed values
- `na.rm` a logical value indicating whether 'NA' should be stripped before the computation proceeds.  
  When an 'NA' value is found at the i-th position in `obs` OR `sim`, the i-th value of `obs` AND `sim` are removed before the computation.
  
  ... further arguments passed to or from other methods.

**Value**

Ratio of RMSE to the standard deviation of the observations.

If `sim` and `obs` are matrixes, the returned value is a vector, with the RSR between each column of `sim` and `obs`.

**Note**

`obs` and `sim` has to have the same length/dimension

The missing values in `obs` and `sim` are removed before the computation proceeds, and only those positions with non-missing values in `obs` and `sim` are considered in the computation.
Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

References


See Also

sd, rSD, gof, ggof

Examples

```r
sim <- 1:10
obs <- 1:10
rsr(sim, obs)

sim <- 2:11
obs <- 1:10
rsr(sim, obs)
```

Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
```r
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts
```

Generating a simulated daily time series, initially equal to the observed series
```r
sim <- obs
```

Computing the 'rsr' for the "best" (unattainable) case
```r
rsr(sim=sim, obs=obs)
```

Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').
```r
```

Computing the new 'rsr'
```r
rsr(sim=sim, obs=obs)
```

ssq Sum of the Squared Residuals

Description

Sum of the Squared Residuals between `sim` and `obs`, with treatment of missing values. Its units are the squared measurement units of `sim` and `obs`. 
Usage

ssq(sim, obs, ...)  
## Default S3 method:  
ssq(sim, obs, na.rm = TRUE, ...)  
## S3 method for class 'data.frame'  
ssq(sim, obs, na.rm=TRUE, ...)  
## S3 method for class 'matrix'  
ssq(sim, obs, na.rm=TRUE, ...)

Arguments

sim numeric, zoo, matrix or data.frame with simulated values  
obs numeric, zoo, matrix or data.frame with observed values  
na.rm a logical value indicating whether 'NA' should be stripped before the computation proceeds.  
When an 'NA' value is found at the i-th position in obs OR sim, the i-th value of obs AND sim are removed before the computation.  
... further arguments passed to or from other methods.

Value

Sum of the squared residuals between sim and obs.

If sim and obs are matrixes, the returned value is a vector, with the SSR between each column of sim and obs.

Note

obs and sim has to have the same length/dimension

The missing values in obs and sim are removed before the computation proceeds, and only those positions with non-missing values in obs and sim are considered in the computation.

Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

Examples

obs <- 1:10  
sim <- 1:10  
ssq(sim, obs)

obs <- 1:10  
sim <- 2:11
valindex

ssq(sim, obs)

# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Generating a simulated daily time series, initially equal to the observed series
sim <- obs

# Computing the \textquoteleft rNSeff\textquoteright for the \textquoteleft best\textquoteright (unattainable) case
ssq(sim=sim, obs=obs)

# Randomly changing the first 2000 elements of \textquotesingle sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of \textquoteleft rnorm\textquoteright).

# Computing the new \textquoteleft rNSeff\textquoteright
ssq(sim=sim, obs=obs)

---

valindex | Valid Indexes
---------|-------------------------

**Description**

Identify the indexes that are simultaneously valid (not missing) in sim and obs.

**Usage**

valindex(sim, obs, ...)

## Default S3 method:
valindex(sim, obs, ...)

## S3 method for class 'matrix'
valindex(sim, obs, ...)

**Arguments**

- **sim**: zoo, xts, numeric, matrix or data.frame with simulated values
- **obs**: zoo, xts, numeric, matrix or data.frame with observed values
- **...**: further arguments passed to or from other methods.

**Value**

A vector with the indexes that are simultaneously valid (not missing) in obs and sim.
Note

This function is used in the functions of this package for removing missing values from the observed and simulated time series.

Author(s)

Mauricio Zambrano Bigiarini <mauricio.zambrano@ing.unitn.it>

See Also

is.na, which

Examples

```r
sim <- 1:5
obs <- c(1, NA, 3, NA, 5)
valindex(sim, obs)
```

---

Volumetric Efficiency

Description

Volumetric efficiency between sim and obs, with treatment of missing values.

Usage

```r
VE(sim, obs, ...)
```

## Default S3 method:
`VE(sim, obs, na.rm=TRUE, ...)`

## S3 method for class 'data.frame'
`VE(sim, obs, na.rm=TRUE, ...)`

## S3 method for class 'matrix'
`VE(sim, obs, na.rm=TRUE, ...)`

## S3 method for class 'zoo'
`VE(sim, obs, na.rm=TRUE, ...)`
Arguments

- **sim**: numeric, zoo, matrix or data.frame with simulated values
- **obs**: numeric, zoo, matrix or data.frame with observed values
- **na.rm**: a logical value indicating whether 'NA' should be stripped before the computation proceeds.

When an 'NA' value is found at the i-th position in obs OR sim, the i-th value of obs AND sim are removed before the computation.

... further arguments passed to or from other methods.

Details

\[
VE = 1 - \frac{\sum_{i=1}^{N} |S_i - O_i|}{\sum_{i=1}^{N} (O_i)}
\]

Volumetric efficiency was proposed in order to circumvent some problems associated to the Nash-Sutcliffe efficiency. It ranges from 0 to 1 and represents the fraction of water delivered at the proper time; its compliment represents the fractional volumetric mistmach (Criss and Winston, 2008).

Value

Volumetric efficiency between sim and obs.

If sim and obs are matrixes, the returned value is a vector, with the Volumetric efficiency between each column of sim and obs.

Note

obs and sim have to have the same length/dimension

The missing values in obs and sim are removed before the computation proceeds, and only those positions with non-missing values in obs and sim are considered in the computation.

Author(s)

Mauricio Zambrano Bigiarini <mzb.devel@gmail.com>

References


See Also
gof, ggof, NSE
Examples

```r
obs <- 1:10
sim <- 1:10
VE(sim, obs)
```

```r
obs <- 1:10
sim <- 2:11
VE(sim, obs)
```

```
# Loading daily streamflows of the Ega River (Spain), from 1961 to 1970
require(zoo)
data(EgaEnEstellaQts)
obs <- EgaEnEstellaQts

# Generating a simulated daily time series, initially equal to the observed series
sim <- obs

# Computing the volumetric efficiency for the "best" case
VE(sim=sim, obs=obs)

# Randomly changing the first 2000 elements of 'sim', by using a normal distribution
# with mean 10 and standard deviation equal to 1 (default of 'rnorm').

# Computing the new volumetric efficiency
VE(sim=sim, obs=obs)
```
Index

*Topic **datasets**
  EgaEnEstellaQts, 12

*Topic **dplot**
  ggof, 13
  plot2, 48
  plotbands, 51
  plotbandsonly, 54

*Topic **math**
  br2, 5
  cp, 7
  d, 10
  ggof, 13
  gof, 18
  KGE, 22
  mae, 26
  md, 28
  me, 31
  mNSE, 32
  mse, 35
  nrmse, 37
  NSE, 39
  pbias, 41
  pbiasfdc, 43
  pfactor, 46
  rd, 55
  rfactor, 58
  rmse, 60
  rNSE, 62
  rPearson, 64
  rSD, 66
  rsr, 68
  ssq, 69
  valindex, 71
  ve, 72

*Topic **package**
  hydroGOF-package, 2

as.Date, 13, 51, 54
br2, 5, 17, 21

cor, 7, 64, 65

cp, 7, 17, 21

d, 10, 17, 21, 30, 57
drawTimeAxis, 14, 49, 52
EgaEnEstellaQts, 12

fdc, 44, 45

ggof, 7, 11, 13, 25, 28, 30, 32, 34, 38, 41, 43, 50, 57, 61, 63, 67, 69, 73
gof, 7, 9, 11, 17, 18, 25, 28, 30, 32, 34, 36, 38, 41, 43, 45, 57, 61, 63, 67, 69, 73

hydroGOF (hydroGOF-package), 2
hydroGOF-package, 2

is.na, 72

KGE, 17, 19, 21, 22, 41

lm, 7

mae, 17, 21, 26, 32, 36
md, 11, 17, 21, 28, 57
me, 17, 21, 28, 31, 36
mNSE, 17, 19, 21, 32, 41, 63
mNSEFF (mNSE), 32
mse, 35

nrmse, 17, 19, 21, 37, 61
NSE, 17, 21, 23, 25, 34, 39, 63, 73
NSEFF (NSE), 39

par, 15, 49
pbias, 17, 21, 41
pbiasfdc, 17, 19, 21, 43
pfactor, 46, 53, 54, 59
plot.default, 52, 53
plot.zoo, 50
plot2, 17, 48
plotbands, 47, 51, 59
plotbandsonly, 54
points, 53
polygon, 52, 54

rd, 11, 17, 21, 30, 55
rfactor, 47, 53, 54, 58
rmse, 17, 21, 38, 60
rNSE, 17, 21, 34, 41, 62
rNSeff (rNSE), 62
rPearson, 64
rSD, 17, 21, 66, 69
rsr, 67, 68

sd, 67, 69
ssq, 38, 61, 69

title, 49

valindex, 71
VE, 17, 21
VE (ve), 72
ve, 72

which, 72