Package 'WASP'

January 20, 2025

Title Wavelet System Prediction

Version 1.4.4

Author Ze Jiang [aut, cre] (<https://orcid.org/0000-0002-3472-0829>),
Md. Mamunur Rashid [aut] (<https://orcid.org/0000-0002-0315-9055>),
Ashish Sharma [aut] (<https://orcid.org/0000-0002-6758-0519>),
Fiona Johnson [aut] (<https://orcid.org/0000-0001-5708-1807>)

Maintainer Ze Jiang <ze.jiang@unsw.edu.au>

Description The wavelet-based variance transformation method is used for system modelling and prediction. It refines predictor spectral representation using Wavelet Theory, which leads to improved model specifications and prediction accuracy. Details of methodologies used in the package can be found in Jiang, Z., Sharma, A., & Johnson, F. (2020) <doi:10.1029/2019WR026962>, Jiang, Z., Rashid, M. M., Johnson, F., & Sharma, A. (2020) <doi:10.1016/j.envsoft.2020.104907>, and Jiang, Z., Sharma, A., & Johnson, F. (2021) <doi:10.1016/J.JHYDROL.2021.126816>.

License GPL-3

Encoding UTF-8

Depends R (>= 3.6.0)

URL https://github.com/zejiang-unsw/WASP#readme

BugReports https://github.com/zejiang-unsw/WASP/issues

Imports waveslim, stats, tidyr, readr, ggplot2, sp, zoo, fitdistrplus

Suggests FNN, SPEI, knitr, dplyr, cowplot, gridGraphics, bookdown, rmarkdown, synthesis, kableExtra, devtools, testthat

RoxygenNote 7.3.2

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2024-07-20 06:40:02 UTC

Contents

WASP-package *WASP: WAvelet System Prediction*

Description

The package WASP (variance transformation) is used for system modelling and prediction.

Details

Package:	WASP
Type:	Package
Version:	1.0.1
Date:	2020-03-17
License:	GPL-3

WASP functions

Variance transformation functions: dwt.vt, modwt.vt, at.vt and associated

K-nearest neighbor function: knn

Synthetic data generator functions: data.gen.SW, data.gen.HL, data.gen.Rossler; data.gen.ar1, data.gen.ar4, data.gen.ar9, data.gen.tar1, data.gen.tar2.

_PACKAGE

Author(s)

Ze Jiang

Maintainer: Ze Jiang <ze.jiang@unsw.edu.au>

References

Jiang, Z., Sharma, A., & Johnson, F. (2020). Refining Predictor Spectral Representation Using Wavelet Theory for Improved Natural System Modeling. Water Resources Research, 56(3), e2019WR026962. doi:10.1029/2019wr026962

Percival, D. B. and A. T. Walden (2000) Wavelet Methods for Time Series Analysis, Cambridge: Cambridge University Press.

at.vt *Variance Transformation Operation - AT(a trous)*

Description

Variance Transformation Operation - AT(a trous)

Usage

```
at.vt(
  data,
  wf,
  J,
  boundary,
  cov.opt = "auto",
  flag = "biased",
  detrend = FALSE
)
```

Arguments

data	A list of response x and dependent variables dp.
wf	Name of the wavelet filter to use in the decomposition.
J	Specifies the depth of the decomposition. This must be a number less than or equal to log(length(x),2).
boundary	Character string specifying the boundary condition. If boundary=="periodic" the default, then the vector you decompose is assumed to be periodic on its defined interval, if boundary=="reflection", the vector beyond its boundaries is assumed to be a symmetric reflection of itself.
cov.opt	Options of Covariance matrix sign. Use "pos", "neg", or "auto".
flag	Biased or Unbiased variance transformation, c("biased","unbiased").
detrend	Detrend the input time series or just center, default (F).

Value

A list of 8 elements: wf, J, boundary, x (data), dp (data), dp.n (variance transformed dp), and S (covariance matrix).

References

Jiang, Z., Sharma, A., & Johnson, F. (2020). Refining Predictor Spectral Representation Using Wavelet Theory for Improved Natural System Modeling. Water Resources Research, 56(3), e2019WR026962.

Jiang, Z., Sharma, A., & Johnson, F. (2021). Variable transformations in the spectral domain – Implications for hydrologic forecasting. Journal of Hydrology, 126816.

Examples

```
data(rain.mon)
data(obs.mon)

## response SPI - calibration
SPI.cal <- SPI.calc(window(rain.mon, start=c(1949,1), end=c(1979,12)),sc=12)
#SPI.cal <- SPEI::spi(window(rain.mon, start = c(1949, 1), end = c(1979, 12)), scale = 12)$fitted

## create paired response and predictors dataset for each station
data.list <- list()
for (id in seq_len(ncol(SPI.cal))) {
  x <- window(SPI.cal[, id], start = c(1950, 1), end = c(1979, 12))
  dp <- window(obs.mon, start = c(1950, 1), end = c(1979, 12))
  data.list[[id]] <- list(x = as.numeric(x), dp = matrix(dp, nrow = nrow(dp)))
}

## variance transformation
dwt.list <- lapply(
  data.list,
  function(x) at.vt(x, wf = "d4", J = 7, boundary = "periodic", cov.opt = "auto")
)

## plot original and reconstrcuted predictors for each station
for (i in seq_len(length(dwt.list))) {
  # extract data
  dwt <- dwt.list[[i]]
  x <- dwt$x # response
  dp <- dwt$dp # original predictors
  dp.n <- dwt$dp.n # variance transformed predictors

  plot.ts(cbind(x, dp))
  plot.ts(cbind(x, dp.n))
}
```

`at.vt.val` *Variance Transformation Operation for Validation*

Description

Variance Transformation Operation for Validation

Usage

```
at.vt.val(data, J, dwt, detrend = FALSE)
```

Arguments

`data`	A list of response x and dependent variables dp.

J	Specifies the depth of the decomposition. This must be a number less than or equal to log(length(x),2).
dwt	A class of "at" data. Output from at.vt().
detrend	Detrend the input time series or just center, default (F).

Value

A list of 8 elements: wf, J, boundary, x (data), dp (data), dp.n (variance transformed dp), and S (covariance matrix).

References

Jiang, Z., Sharma, A., & Johnson, F. (2020). Refining Predictor Spectral Representation Using Wavelet Theory for Improved Natural System Modeling. Water Resources Research, 56(3), e2019WR026962. doi:10.1029/2019wr026962

Examples

```
data(rain.mon)
data(obs.mon)

## response SPI - calibration
SPI.cal <- SPI.calc(window(rain.mon, start=c(1949,1), end=c(1979,12)),sc=12)
#SPI.cal <- SPEI::spi(window(rain.mon, start = c(1949, 1), end = c(1979, 12)), scale = 12)$fitted

## create paired response and predictors dataset for each station
data.list <- list()
for (id in seq_len(ncol(SPI.cal))) {
  x <- window(SPI.cal[, id], start = c(1950, 1), end = c(1979, 12))
  dp <- window(obs.mon, start = c(1950, 1), end = c(1979, 12))
  data.list[[id]] <- list(x = as.numeric(x), dp = matrix(dp, nrow = nrow(dp)))
}

## variance transformation - calibration
dwt.list <- lapply(
  data.list,
  function(x) at.vt(x, wf = "d4", J = 7, boundary = "periodic", cov.opt = "auto")
)

## response SPI - validation
SPI.val <- SPI.calc(window(rain.mon, start=c(1979,1), end=c(2009,12)),sc=12)
#SPI.val <- SPEI::spi(window(rain.mon, start = c(1979, 1), end = c(2009, 12)), scale = 12)$fitted

## create paired response and predictors dataset for each station
data.list <- list()
for (id in seq_len(ncol(SPI.val))) {
  x <- window(SPI.val[, id], start = c(1980, 1), end = c(2009, 12))
  dp <- window(obs.mon, start = c(1980, 1), end = c(2009, 12))
  data.list[[id]] <- list(x = as.numeric(x), dp = matrix(dp, nrow = nrow(dp)))
}
```

```
# variance transformation - validation
dwt.list.val <- lapply(
  seq_len(length(data.list)),
  function(i) at.vt.val(data.list[[i]], J = 7, dwt.list[[i]])
)

## plot original and reconstrcuted predictors for each station
for (i in seq_len(length(dwt.list.val))) {
  # extract data
  dwt <- dwt.list.val[[i]]
  x <- dwt$x # response
  dp <- dwt$dp # original predictors
  dp.n <- dwt$dp.n # variance transformed predictors

  plot.ts(cbind(x, dp))
  plot.ts(cbind(x, dp.n))
}
```

`at.wd` *a trous (AT) based additive decompostion using Daubechies family wavelet*

Description

a trous (AT) based additive decompostion using Daubechies family wavelet

Usage

```
at.wd(x, wf, J, boundary = "periodic")
```

Arguments

`x`	The input time series.
`wf`	Name of the wavelet filter to use in the decomposition.
`J`	Specifies the depth of the decomposition. This must be a number less than or equal to log(length(x),2).
`boundary`	Character string specifying the boundary condition. If boundary=="periodic" the default, then the vector you decompose is assumed to be periodic on its defined interval, if boundary=="reflection", the vector beyond its boundaries is assumed to be a symmetric reflection of itself.

Value

A matrix of decomposed sub-time series.

References

Nason, G. P. (1996). Wavelet shrinkage using cross-validation. Journal of the Royal Statistical Society: Series B (Methodological), 58(2), 463-479.

Examples

```
data(obs.mon)

n <- nrow(obs.mon)
v <- 1
J <- floor(log(n / (2 * v - 1)) / log(2)) # (Kaiser, 1994)

names <- colnames(obs.mon)
at.atm <- vector("list", ncol(obs.mon))
for (i in seq_len(ncol(obs.mon))) {
  tmp <- as.numeric(scale(obs.mon[, i], scale = FALSE))
  at.atm <- do.call(cbind, at.wd(tmp, wf = "haar", J = J, boundary = "periodic"))

  plot.ts(cbind(obs.mon[1:n, i], at.atm[1:n, 1:9]), main = names[i])
  print(sum(abs(scale(obs.mon[1:n, i], scale = FALSE) - rowSums(at.atm[1:n, ]))))
}
```

---

aus.coast *Sample data: Australia map*

Description

A dataset containing the Australia map.

Usage

```
data(aus.coast)
```

---

data.AWAP.2.5 *Sample data: AWAP rainfall data over Australia*

Description

A dataset containing 1320 rows (data length) and 252 columns (grids).

Usage

```
data(data.AWAP.2.5)
```

data.CI *Sample data: Climate indices strongly influencing Australia climate*

Description

A dataset containing 1332 rows (data length) and 6 columns (indices).

Usage

```
data(data.CI)
```

data.gen.ar1 *Generate predictor and response data from AR1 model.*

Description

Generate predictor and response data from AR1 model.

Usage

```
data.gen.ar1(nobs, ndim = 9)
```

Arguments

nobs	The data length to be generated.
ndim	The number of potential predictors (default is 9).

Value

A list of 2 elements: a vector of response (x), and a matrix of potential predictors (dp) with each column containing one potential predictor.

Examples

```
# AR1 model from paper with 9 dummy variables
data.ar1 <- data.gen.ar1(500)
plot.ts(cbind(data.ar1$x, data.ar1$dp))
```

data.gen.ar4 *Generate predictor and response data from AR4 model.*

Description

Generate predictor and response data from AR4 model.

Usage

```
data.gen.ar4(nobs, ndim = 9)
```

Arguments

nobs	The data length to be generated.
ndim	The number of potential predictors (default is 9).

Value

A list of 2 elements: a vector of response (x), and a matrix of potential predictors (dp) with each column containing one potential predictor.

Examples

```
# AR4 model from paper with total 9 dimensions
data.ar4 <- data.gen.ar4(500)
plot.ts(cbind(data.ar4$x, data.ar4$dp))
```

data.gen.ar9 *Generate predictor and response data from AR9 model.*

Description

Generate predictor and response data from AR9 model.

Usage

```
data.gen.ar9(nobs, ndim = 9)
```

Arguments

nobs	The data length to be generated.
ndim	The number of potential predictors (default is 9).

Value

A list of 2 elements: a vector of response (x), and a matrix of potential predictors (dp) with each column containing one potential predictor.

Examples

```
# AR9 model from paper with total 9 dimensions
data.ar9 <- data.gen.ar9(500)
plot.ts(cbind(data.ar9$x, data.ar9$dp))
```

`data.gen.HL` *Generate predictor and response data: Hysteresis Loop*

Description

Generate predictor and response data: Hysteresis Loop

Usage

```
data.gen.HL(n = 3, m = 5, nobs = 512, fp = 25, fd, sd.x = 0.1, sd.y = 0.1)
```

Arguments

`n`	Positive integer for the split line parameter. If n=1, split line is linear; If n is even, split line has a u shape; If n is odd and higher than 1, split line has a chair or classical shape.
`m`	Positive odd integer for the bulging parameter, indicates degree of outward curving (1=highest level of bulging).
`nobs`	The data length to be generated.
`fp`	The frequency in the generated response.fp = 25 used in the WRR paper.
`fd`	A vector of frequencies for potential predictors. fd = c(3,5,10,15,25,30,55,70,95) used in the WRR paper.
`sd.x`	The noise level in the predictor.
`sd.y`	The noise level in the response.

Details

The Hysteresis is a common nonlinear phenomenon in natural systems and it can be numerical simulated by the following formulas:

$$x_t = a * cos(2pi * f * t)$$

$$y_t = b * cos(2pi * f * t)^n + c * sin(2pi * f * t)^m$$

The default selection for the system parameters (a = 0.8, b = 0.6, c = -0.2, n = 3, m = 5) is known to generate a classical hysteresis loop.

Value

A list of 3 elements: a vector of response (x), a matrix of potential predictors (dp) with each column containing one potential predictor, and a vector of true predictor numbers.

References

LAPSHIN, R. V. 1995. Analytical model for the approximation of hysteresis loop and its application to the scanning tunneling microscope. Review of Scientific Instruments, 66, 4718-4730.

Examples

```
###synthetic example - Hysteresis loop
#frequency, sampled from a given range
fd <- c(3,5,10,15,25,30,55,70,95)

data.HL <- data.gen.HL(n=3,m=5,nobs=512,fp=25,fd=fd)
plot.ts(cbind(data.HL$x,data.HL$dp))
```

---

data.gen.Rossler *Generate predictor and response data: Rossler system*

Description

Generates a 3-dimensional time series using the Rossler equations.

Usage

```
data.gen.Rossler(
  a = 0.2,
  b = 0.2,
  w = 5.7,
  start = c(-2, -10, 0.2),
  time = seq(0, 50, length.out = 5000)
)
```

Arguments

a	The *a* parameter. Default:0.2.
b	The *b* parameter. Default: 0.2.
w	The *w* parameter. Default: 5.7.
start	A 3-dimensional numeric vector indicating the starting point for the time series. Default: c(-2, -10, 0.2).
time	The temporal interval at which the system will be generated. Default: time=seq(0,50,length.out = 5000).

Details

The Rossler system is a system of ordinary differential equations defined as:

$$\dot{x} = -(y + z)$$

$$\dot{y} = x + a \cdot y$$

$$\dot{z} = b + z * (x - w)$$

The default selection for the system parameters (a = 0.2, b = 0.2, w = 5.7) is known to produce a deterministic chaotic time series.

Value

A list with four vectors named *time*, *x*, *y* and *z* containing the time, the x-components, the y-components and the z-components of the Rossler system, respectively.

Note

Some initial values may lead to an unstable system that will tend to infinity.

References

RÖSSLER, O. E. 1976. An equation for continuous chaos. Physics Letters A, 57, 397-398.

Examples

```
### synthetic example - Rossler
ts.r <- data.gen.Rossler(
  a = 0.2, b = 0.2, w = 5.7, start = c(-2, -10, 0.2),
  time = seq(0, 50, length.out = 1000)
)

# add noise
ts.r$x <- ts(ts.r$x + rnorm(length(ts.r$time), mean = 0, sd = 1))
ts.r$y <- ts(ts.r$y + rnorm(length(ts.r$time), mean = 0, sd = 1))
ts.r$z <- ts(ts.r$z + rnorm(length(ts.r$time), mean = 0, sd = 1))

ts.plot(ts.r$x, ts.r$y, ts.r$z, col = c("black", "red", "blue"))
```

---

data.gen.SW *Generate predictor and response data: Sinewave model*

Description

Generate predictor and response data: Sinewave model

Usage

```
data.gen.SW(nobs = 512, fp = 25, fd, sd.x = 0.1, sd.y = 0.1)
```

Arguments

nobs	The data length to be generated.
fp	The frequencies in the generated response.
fd	A vector of frequencies for potential predictors. fd = c(3,5,10,15,25,30,55,70,95) used in the WRR paper.
sd.x	The noise level in the predictor.
sd.y	The noise level in the response.

Value

A list of 3 elements: a vector of response (x), a matrix of potential predictors (dp) with each column containing one potential predictor, and a vector of true predictor numbers.

Examples

```
###synthetic example
#frequency, sampled from a given range
fd <- c(3,5,10,15,25,30,55,70,95)

data.SW1 <- data.gen.SW(nobs=512,fp=25,fd=fd)
data.SW3 <- data.gen.SW(nobs=512,fp=c(15,25,30),fd=fd)

ts.plot(ts(data.SW1$x),ts(data.SW3$x),col=c("black","red"))
plot.ts(cbind(data.SW1$x,data.SW1$dp))
plot.ts(cbind(data.SW3$x,data.SW3$dp))
```

---

data.gen.tar1 *Generate predictor and response data from TAR1 model.*

Description

Generate predictor and response data from TAR1 model.

Usage

```
data.gen.tar1(nobs, ndim = 9, noise = 0.1)
```

Arguments

nobs	The data length to be generated.
ndim	The number of potential predictors (default is 9).
noise	The white noise in the data

Value

A list of 2 elements: a vector of response (x), and a matrix of potential predictors (dp) with each column containing one potential predictor.

References

Sharma, A. (2000). Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1—A strategy for system predictor identification. Journal of Hydrology, 239(1-4), 232-239.

Examples

```
# TAR1 model from paper with total 9 dimensions
data.tar1 <- data.gen.tar1(500)
plot.ts(cbind(data.tar1$x, data.tar1$dp))
```

`data.gen.tar2` *Generate predictor and response data from TAR2 model.*

Description

Generate predictor and response data from TAR2 model.

Usage

```
data.gen.tar2(nobs, ndim = 9, noise = 0.1)
```

Arguments

`nobs`	The data length to be generated.
`ndim`	The number of potential predictors (default is 9).
`noise`	The white noise in the data

Value

A list of 2 elements: a vector of response (x), and a matrix of potential predictors (dp) with each column containing one potential predictor.

References

Sharma, A. (2000). Seasonal to interannual rainfall probabilistic forecasts for improved water supply management: Part 1—A strategy for system predictor identification. Journal of Hydrology, 239(1-4), 232-239.

Examples

```
# TAR2 model from paper with total 9 dimensions
data.tar2 <- data.gen.tar2(500)
plot.ts(cbind(data.tar2$x, data.tar2$dp))
```

data.HL *Sample data: Hysteresis loop*

Description

A dataset containing 3 lists: a vector of response (x), a matrix of 9 potential predictors (dp) with each column containing one potential predictor, and a vector of true predictor numbers.

Usage

```
data(data.HL)
```

data.SW1 *Sample data: Sinewave model 1 (SW1)*

Description

A dataset containing 3 lists: a vector of response (x), a matrix of 9 potential predictors (dp) with each column containing one potential predictor, and a vector of true predictor numbers. The Sinewave model 1 (SW1) is defined as:

$$x_t = sin(2pi * f * t) + eps$$

Usage

```
data(data.SW1)
```

data.SW3 *Sample data: Sinewave model 3 (SW3)*

Description

A dataset containing 3 lists: a vector of response (x), a matrix of 9 potential predictors (dp) with each column containing one potential predictor, and a vector of true predictor numbers. The Sinewave model 3 (SW3) is defined as:

$$x_t = \sum_{i=1}^{3} sin(2pi * f_i * t) + eps$$

Usage

```
data(data.SW3)
```

dwt.vt *Variance Transformation Operation - MRA*

Description

Variance Transformation Operation - MRA

Usage

```
dwt.vt(
  data,
  wf,
  J,
  method,
  pad,
  boundary,
  cov.opt = "auto",
  flag = "biased",
  detrend = FALSE,
  backward = FALSE,
  verbose = TRUE
)
```

Arguments

data	A list of response x and dependent variables dp.
wf	Name of the wavelet filter to use in the decomposition.
J	Specifies the depth of the decomposition. This must be a number less than or equal to log(length(x),2).
method	Either "dwt" or "modwt".
pad	The method used for extend data to dyadic size. Use "per", "zero", or "sym".
boundary	Character string specifying the boundary condition. If boundary=="periodic" the default, then the vector you decompose is assumed to be periodic on its defined interval, if boundary=="reflection", the vector beyond its boundaries is assumed to be a symmetric reflection of itself.
cov.opt	Options of Covariance matrix sign. Use "pos", "neg", or "auto".
flag	Biased or Unbiased variance transformation, c("biased","unbiased").
detrend	Detrend the input time series or just center, default (F).
backward	Detrend the input time series or just center, default (F).
verbose	A logical indicating if some “progress report” should be given.

Value

A list of 8 elements: wf, method, boundary, pad, x (data), dp (data), dp.n (variance trasnformed dp), and S (covariance matrix).

References

Jiang, Z., Sharma, A., & Johnson, F. (2020). Refining Predictor Spectral Representation Using Wavelet Theory for Improved Natural System Modeling. Water Resources Research, 56(3), e2019WR026962.

Examples

```
data(rain.mon)
data(obs.mon)

## response SPI - calibration
SPI.cal <- SPI.calc(window(rain.mon, start=c(1949,1), end=c(1979,12)),sc=12)
#SPI.cal <- SPEI::spi(window(rain.mon, start = c(1949, 1), end = c(1979, 12)), scale = 12)$fitted

## create paired response and predictors dataset for each station
data.list <- list()
for (id in seq_len(ncol(SPI.cal))) {
  x <- window(SPI.cal[, id], start = c(1950, 1), end = c(1979, 12))
  dp <- window(obs.mon, start = c(1950, 1), end = c(1979, 12))
  data.list[[id]] <- list(x = as.numeric(x), dp = matrix(dp, nrow = nrow(dp)))
}

## variance transformation
dwt.list <- lapply(data.list, function(x) {
 dwt.vt(x, wf = "d4", J = 7, method = "dwt", pad = "zero", boundary = "periodic", cov.opt = "auto")
})

## plot original and reconstrcuted predictors for each station
for (i in seq_len(length(dwt.list))) {
  # extract data
  dwt <- dwt.list[[i]]
  x <- dwt$x # response
  dp <- dwt$dp # original predictors
  dp.n <- dwt$dp.n # variance transformed predictors

  plot.ts(cbind(x, dp))
  plot.ts(cbind(x, dp.n))
}
```

---

dwt.vt.val *Variance Transformation Operation for Validation*

Description

Variance Transformation Operation for Validation

Usage

```
dwt.vt.val(data, J, dwt, detrend = FALSE, backward = FALSE, verbose = TRUE)
```

Arguments

data	A list of response x and dependent variables dp.
J	Specifies the depth of the decomposition. This must be a number less than or equal to log(length(x),2).
dwt	A class of "dwt" data. Output from dwt.vt().
detrend	Detrend the input time series or just center, default (F).
backward	Detrend the input time series or just center, default (F).
verbose	A logical indicating if some “progress report” should be given.

Value

A list of 8 elements: wf, method, boundary, pad, x (data), dp (data), dp.n (variance trasnformed dp), and S (covariance matrix).

References

Jiang, Z., Sharma, A., & Johnson, F. (2020). Refining Predictor Spectral Representation Using Wavelet Theory for Improved Natural System Modeling. Water Resources Research, 56(3), e2019WR026962. doi:10.1029/2019wr026962

Examples

```
data(rain.mon)
data(obs.mon)

## response SPI - calibration
SPI.cal <- SPI.calc(window(rain.mon, start=c(1949,1), end=c(1979,12)),sc=12)
#SPI.cal <- SPEI::spi(window(rain.mon, start = c(1949, 1), end = c(1979, 12)), scale = 12)$fitted

## create paired response and predictors dataset for each station
data.list <- list()
for (id in seq_len(ncol(SPI.cal))) {
  x <- window(SPI.cal[, id], start = c(1950, 1), end = c(1979, 12))
  dp <- window(obs.mon, start = c(1950, 1), end = c(1979, 12))
  data.list[[id]] <- list(x = as.numeric(x), dp = matrix(dp, nrow = nrow(dp)))
}

## variance transformation - calibration
dwt.list <- lapply(data.list, function(x) {
 dwt.vt(x, wf = "d4", J = 7, method = "dwt", pad = "zero", boundary = "periodic", cov.opt = "auto")
})

## response SPI - validation
SPI.val <- SPI.calc(window(rain.mon, start=c(1979,1), end=c(2009,12)),sc=12)
#SPI.val <- SPEI::spi(window(rain.mon, start = c(1979, 1), end = c(2009, 12)), scale = 12)$fitted

## create paired response and predictors dataset for each station
data.list <- list()
for (id in seq_len(ncol(SPI.val))) {
```

```
  x <- window(SPI.val[, id], start = c(1980, 1), end = c(2009, 12))
  dp <- window(obs.mon, start = c(1980, 1), end = c(2009, 12))
  data.list[[id]] <- list(x = as.numeric(x), dp = matrix(dp, nrow = nrow(dp)))
}

# variance transformation - validation
dwt.list.val <- lapply(
  seq_len(length(data.list)),
  function(i) dwt.vt.val(data.list[[i]], J = 7, dwt.list[[i]])
)

## plot original and reconstrcuted predictors for each station
for (i in seq_len(length(dwt.list.val))) {
  # extract data
  dwt <- dwt.list.val[[i]]
  x <- dwt$x # response
  dp <- dwt$dp # original predictors
  dp.n <- dwt$dp.n # variance transformed predictors

  plot.ts(cbind(x, dp))
  plot.ts(cbind(x, dp.n))
}
```

`fig.dwt.vt` *Plot function: Variance structure before and after variance transformation*

Description

Plot function: Variance structure before and after variance transformation

Usage

```
fig.dwt.vt(dwt.data)
```

Arguments

`dwt.data`	Output data from variance transformation function

Value

A plot with variance structure before and after variance transformation.

Examples

```
data("data.HL")
data("data.SW1")

# variance transfrom
dwt.SW1 <- dwt.vt(data.SW1[[1]],
```

```
  wf = "d4", J = 7, method = "dwt",
  pad = "zero", boundary = "periodic", cov.opt = "auto"
)

# plot
fig1 <- fig.dwt.vt(dwt.SW1)
fig1

# variance transfrom
dwt.HL <- dwt.vt(data.HL[[1]],
  wf = "d4", J = 7, method = "dwt",
  pad = "zero", boundary = "periodic", cov.opt = "auto"
)

# plot
fig2 <- fig.dwt.vt(dwt.HL)
fig2
```

`Ind_AWAP.2.5` *Sample data: Index of AWAP grids with no missing data*

Description

A dataset containing 145 numbers.

Usage

```
data(Ind_AWAP.2.5)
```

`knn` *Modified k-nearest neighbour conditional bootstrap or regression function estimation with extrapolation*

Description

Modified k-nearest neighbour conditional bootstrap or regression function estimation with extrapolation

Usage

```
knn(
  x,
  z,
  zout,
  k = 0,
  pw,
  reg = TRUE,
```

```
  nensemble = 100,
  tailcorrection = TRUE,
  tailprob = 0.25,
  tailfac = 0.2,
  extrap = TRUE
)
```

Arguments

x	A vector of response.
z	A matrix of existing predictors.
zout	A matrix of predictor values the response is to be estimated at.
k	The number of nearest neighbours used. The default value is 0, indicating Lall and Sharma default is used.
pw	A vector of partial weights of the same length of z.
reg	A logical operator to inform whether a conditional expectation should be output or not nensemble, Used if reg=F and represents the number of realisations that are generated Value.
nensemble	An integer the specifies the number of ensembles used. The default is 100.
tailcorrection	A logical value, T (default) or F, that denotes whether a reduced value of k (number of nearest neighbours) should be used in the tails of any conditioning plane. Whether one is in the tails or not is determined based on the nearest neighbour response value.
tailprob	A scalar that denotes the p-value of the cdf (on either extreme) the tailcorrection takes effect. The default value is 0.25.
tailfac	A scalar that specifies the lowest fraction of the default k that can be used in the tails. Depending on the how extreme one is in the tails, the actual k decreases linearly from k (for a p-value greater than tailprob) to tailfac*k proportional to the actual p-value of the nearest neighbour response, divided by tailprob. The default value is 0.2.
extrap	A logical value, T (default) or F, that denotes whether a kernel extraplation method is used to predict x.

Value

A matrix of responses having same rows as zout if reg=T, or having nensemble columns is reg=F.

References

Sharma, A., Tarboton, D.G. and Lall, U., 1997. Streamflow simulation: A nonparametric approach. Water resources research, 33(2), pp.291-308.

Sharma, A. and O'Neill, R., 2002. A nonparametric approach for representing interannual dependence in monthly streamflow sequences. Water resources research, 38(7), pp.5-1.

Examples

```
# AR9 model   x(i)=0.3*x(i-1)-0.6*x(i-4)-0.5*x(i-9)+eps
data.ar9 <- data.gen.ar9(500)
x <- data.ar9$x # response
z <- data.ar9$dp # possible predictors

zout <- ts(data.gen.ar9(500, ndim = ncol(z))$dp) # new input

xhat1 <- xhat2 <- x
xhat1 <- knn(x, z, zout, k = 5, reg = TRUE, extrap = FALSE) # without extrapolation
xhat2 <- knn(x, z, zout, k = 5, reg = TRUE, extrap = TRUE) # with extrapolation

ts.plot(ts(x), ts(xhat1), ts(xhat2), col = c("black", "red", "blue"),
ylim = c(-5, 5), lwd = c(2, 2, 1))
```

---

knnregl1cv *Leave one out cross validation.*

Description

Leave one out cross validation.

Usage

```
knnregl1cv(x, z, k = 0, pw)
```

Arguments

x	A vector of response.
z	A matrix of predictors.
k	The number of nearest neighbours used. The default is 0, indicating Lall and Sharma default is used.
pw	A vector of partial weights of the same length of z.

Value

A vector of L1CV estimates of the response.

References

Lall, U., Sharma, A., 1996. A Nearest Neighbor Bootstrap For Resampling Hydrologic Time Series. Water Resources Research, 32(3): 679-693.

Sharma, A., Mehrotra, R., 2014. An information theoretic alternative to model a natural system using observational information alone. Water Resources Research, 50(1): 650-660.

lat_lon.2.5 *Sample data: Latitude and longitude of AWAP grids*

Description

A dataset containing 252 rows (grids) and 2 columns (lat and lon).

Usage

```
data(lat_lon.2.5)
```

modwt.vt *Variance Transformation Operation - MODWT*

Description

Variance Transformation Operation - MODWT

Usage

```
modwt.vt(
  data,
  wf,
  J,
  boundary,
  cov.opt = "auto",
  flag = "biased",
  detrend = FALSE,
  backward = FALSE,
  verbose = TRUE
)
```

Arguments

data	A list of response x and dependent variables dp.
wf	Name of the wavelet filter to use in the decomposition.
J	Specifies the depth of the decomposition. This must be a number less than or equal to log(length(x),2).
boundary	Character string specifying the boundary condition. If boundary=="periodic" the default, then the vector you decompose is assumed to be periodic on its defined interval, if boundary=="reflection", the vector beyond its boundaries is assumed to be a symmetric reflection of itself.
cov.opt	Options of Covariance matrix sign. Use "pos", "neg", or "auto".
flag	Biased or Unbiased variance transformation, c("biased","unbiased").

detrend	Detrend the input time series or just center, default (F).
backward	Detrend the input time series or just center, default (F).
verbose	A logical indicating if some "progress report" should be given.

Value

A list of 8 elements: wf, J, boundary, x (data), dp (data), dp.n (variance transformed dp), and S (covariance matrix).

References

Jiang, Z., Sharma, A., & Johnson, F. (2020). Refining Predictor Spectral Representation Using Wavelet Theory for Improved Natural System Modeling. Water Resources Research, 56(3), e2019WR026962.

Jiang, Z., Rashid, M. M., Johnson, F., & Sharma, A. (2020). A wavelet-based tool to modulate variance in predictors: an application to predicting drought anomalies. Environmental Modelling & Software, 135, 104907.

Examples

```
### real-world example
data(Ind_AWAP.2.5)
data(obs.mon)
data(SPI.12)
x <- window(SPI.12, start = c(1950, 1), end = c(2009, 12))
dp <- window(obs.mon, start = c(1950, 1), end = c(2009, 12))

op <- par(mfrow = c(ncol(dp), 1), pty = "m", mar = c(1, 4, 1, 2))
for (id in sample(Ind_AWAP.2.5, 1)) {
  data <- list(x = x[, id], dp = dp)
  dwt <- modwt.vt(data, wf = "d4", J = 7, boundary = "periodic", cov.opt = "auto")

  for (i in 1:ncol(dp)) {
    ts.plot(dwt$dp[, i], dwt$dp.n[, i], xlab = NA, col = c("black", "red"), lwd = c(2, 1))
  }
}
par(op)

### synthetic example
# frequency, sampled from a given range
fd <- c(3, 5, 10, 15, 25, 30, 55, 70, 95)

data.SW1 <- data.gen.SW(nobs = 512, fp = 25, fd = fd)
dwt.SW1 <- modwt.vt(data.SW1, wf = "d4", J = 7, boundary = "periodic", cov.opt = "auto")

x.modwt <- waveslim::modwt(dwt.SW1$x, wf = "d4", n.levels = 7, boundary = "periodic")
dp.modwt <- waveslim::modwt(dwt.SW1$dp[, 1], wf = "d4", n.levels = 7, boundary = "periodic")
dp.vt.modwt <- waveslim::modwt(dwt.SW1$dp.n[, 1], wf = "d4", n.levels = 7, boundary = "periodic")

sum(sapply(dp.modwt, var))
var(dwt.SW1$dp[, 1])
```

```
sum(sapply(dp.vt.modwt, var))
var(dwt.SW1$dp.n[, 1])

data <- rbind(
  sapply(dp.modwt, var) / sum(sapply(dp.modwt, var)),
  sapply(dp.vt.modwt, var) / sum(sapply(dp.vt.modwt, var))
)

bar <- barplot(data, beside = TRUE, col = c("red", "blue"))
lines(x = bar[2, ], y = sapply(x.modwt, var) / sum(sapply(x.modwt, var)))
points(x = bar[2, ], y = sapply(x.modwt, var) / sum(sapply(x.modwt, var)))
```

`modwt.vt.val` *Variance Transformation Operation for Validation*

Description

Variance Transformation Operation for Validation

Usage

```
modwt.vt.val(data, J, dwt, detrend = FALSE, backward = FALSE, verbose = TRUE)
```

Arguments

`data`	A list of response x and dependent variables dp.
`J`	Specifies the depth of the decomposition. This must be a number less than or equal to log(length(x),2).
`dwt`	A class of "modwt" data. Output from modwt.vt().
`detrend`	Detrend the input time series or just center, default (F).
`backward`	Detrend the input time series or just center, default (F).
`verbose`	A logical indicating if some “progress report” should be given.

Value

A list of 8 elements: wf, J, boundary, x (data), dp (data), dp.n (variance transformed dp), and S (covariance matrix).

References

Jiang, Z., Sharma, A., & Johnson, F. (2020). Refining Predictor Spectral Representation Using Wavelet Theory for Improved Natural System Modeling. Water Resources Research, 56(3), e2019WR026962. doi:10.1029/2019wr026962

Examples

```
data(rain.mon)
data(obs.mon)

## response SPI - calibration
SPI.cal <- SPI.calc(window(rain.mon, start=c(1949,1), end=c(1979,12)),sc=12)
#SPI.cal <- SPEI::spi(window(rain.mon, start = c(1949, 1), end = c(1979, 12)), scale = 12)$fitted

## create paired response and predictors dataset for each station
data.list <- list()
for (id in 1:ncol(SPI.cal)) {
  x <- window(SPI.cal[, id], start = c(1950, 1), end = c(1979, 12))
  dp <- window(obs.mon, start = c(1950, 1), end = c(1979, 12))
  data.list[[id]] <- list(x = as.numeric(x), dp = matrix(dp, nrow = nrow(dp)))
}

## variance transformation - calibration
dwt.list <- lapply(data.list, function(x) {
  modwt.vt(x, wf = "d4", J = 7, boundary = "periodic", cov.opt = "auto")
})

## response SPI - validation
SPI.val <- SPI.calc(window(rain.mon, start=c(1979,1), end=c(2009,12)),sc=12)
#SPI.val <- SPEI::spi(window(rain.mon, start = c(1979, 1), end = c(2009, 12)), scale = 12)$fitted

## create paired response and predictors dataset for each station
data.list <- list()
for (id in 1:ncol(SPI.val)) {
  x <- window(SPI.val[, id], start = c(1980, 1), end = c(2009, 12))
  dp <- window(obs.mon, start = c(1980, 1), end = c(2009, 12))
  data.list[[id]] <- list(x = as.numeric(x), dp = matrix(dp, nrow = nrow(dp)))
}

# variance transformation - validation
dwt.list.val <- lapply(
  seq_along(data.list),
  function(i) modwt.vt.val(data.list[[i]], J = 7, dwt.list[[i]])
)

## plot original and reconstrcuted predictors for each station
for (i in seq_along(dwt.list.val)) {
  # extract data
  dwt <- dwt.list.val[[i]]
  x <- dwt$x # response
  dp <- dwt$dp # original predictors
  dp.n <- dwt$dp.n # variance transformed predictors

  plot.ts(cbind(x, dp))
  plot.ts(cbind(x, dp.n))
}
```

`mra.plot` *Plot function: Plot original time series and decomposed frequency components*

Description

Plot function: Plot original time series and decomposed frequency components

Usage

```
mra.plot(
  y,
  y.mra,
  limits.x,
  limits.y,
  type = c("details", "coefs"),
  ps = 12,
  ...
)
```

Arguments

`y`	Original time series (Y).
`y.mra`	Decomposed frequency components (d1,d2,..,aJ).
`limits.x`	x limit for plot.
`limits.y`	y limit for plot.
`type`	type of wavelet coefficients, details or approximations.
`ps`	integer; the point size of text (but not symbols).
`...`	arguments for plot().

Value

A plot with original time series and decomposed frequency components.

Examples

```
### synthetic example
# frequency, sampled from a given range
fd <- c(3, 5, 10, 15, 25, 30, 55, 70, 95)
data.SW3 <- data.gen.SW(nobs = 512, fp = c(15, 25, 30), fd = fd)

x <- data.SW3$x
xx <- padding(x, pad = "zero")
### wavelet transfrom
# wavelet family, extension mode and package
wf <- "d4" # wavelet family D8 or db4
```

```
boundary <- "periodic"
pad <- "zero"
if (wf != "haar") v <- as.integer(as.numeric(substr(wf, 2, 3)) / 2) else v <- 1

# Maximum decomposition level J
n <- length(x)
J <- ceiling(log(n / (2 * v - 1)) / log(2)) # (Kaiser, 1994)

### decomposition
x.mra <- waveslim::mra(xx, wf = wf, J = J, method = "dwt", boundary = "periodic")
x.mra.m <- matrix(unlist(x.mra), ncol = J + 1)

print(sum(abs(x - rowSums(x.mra.m[1:n, ])))) # additive check
var(x)
sum(apply(x.mra.m[1:n, ], 2, var)) # variance check

limits.x <- c(0, n)
limits.y <- c(-3, 3)
mra.plot(x, x.mra.m, limits.x, limits.y, type = "details")
```

---

non.bdy *Replace Boundary Wavelet Coefficients with Missing Values (NA).*

Description

Replace Boundary Wavelet Coefficients with Missing Values (NA).

Usage

```
non.bdy(x, wf, method = c("dwt", "modwt", "mra"))
```

Arguments

x	DWT/MODWT/AT object
wf	Character string; name of wavelet filter
method	Either dwt or modwt or mra

Value

Same object as x only with some missing values (NA).

References

Cornish, C. R., Bretherton, C. S., & Percival, D. B. (2006). Maximal overlap wavelet statistical analysis with application to atmospheric turbulence. Boundary-Layer Meteorology, 119(2), 339-374.

obs.mon *Sample data: NCEP reanalysis data averaged over Sydney region*

Description

A dataset containing 720 rows (data length) and 7 columns (atmospheric variables).

Usage

```
data(obs.mon)
```

padding *Padding data to dyadic sample size*

Description

Padding data to dyadic sample size

Usage

```
padding(x, pad = c("per", "zero", "sym"))
```

Arguments

x	A vector or time series containing the data be to decomposed.
pad	Method for padding, including periodic, zero and symetric padding.

Value

A dyadic length (power of 2) vector or time series.

Examples

```
x <- rnorm(360)
x1 <- padding(x, pad = "per")
x2 <- padding(x, pad = "zero")
x3 <- padding(x, pad = "sym")
ts.plot(cbind(x, x1, x2, x3), col = 1:4)
```

`pic.calc` *Calculate PIC*

Description

Calculate PIC

Usage

```
pic.calc(
  X,
  Y,
  Z,
  mode,
  wf,
  J,
  method = "dwt",
  pad = "zero",
  boundary = "periodic",
  cov.opt = "auto",
  flag = "biased",
  detrend = F
)
```

Arguments

`X`	A vector of response.
`Y`	A matrix of new predictors.
`Z`	A matrix of pre-existing predictors that could be NULL if no prior predictors exist.
`mode`	A mode of variance transfomration, i.e., MRA, MODWT, or AT
`wf`	Wavelet family
`J`	The maximum decomposition level
`method`	Either "dwt" or "modwt" of MRA.
`pad`	The method used for extend data to dyadic size. Use "per", "zero", or "sym".
`boundary`	Character string specifying the boundary condition. If boundary=="periodic" the default, then the vector you decompose is assumed to be periodic on its defined interval, if boundary=="reflection", the vector beyond its boundaries is assumed to be a symmetric reflection of itself.
`cov.opt`	Options of Covariance matrix sign. Use "pos", "neg", or "auto".
`flag`	Biased or Unbiased variance transformation.
`detrend`	Detrend the input time series or just center, default (F).

Value

A list of 2 elements: the partial mutual information (pmi), and partial informational correlation (pic).

References

Sharma, A., Mehrotra, R., 2014. An information theoretic alternative to model a natural system using observational information alone. Water Resources Research, 50(1): 650-660.

Galelli S., Humphrey G.B., Maier H.R., Castelletti A., Dandy G.C. and Gibbs M.S. (2014) An evaluation framework for input variable selection algorithms for environmental data-driven models, Environmental Modelling and Software, 62, 33-51, DOI: 10.1016/j.envsoft.2014.08.015.

`r2.boot` *R2 threshold by re-sampling approach*

Description

R2 threshold by re-sampling approach

Usage

```
r2.boot(z.vt, x, prob)
```

Arguments

`z.vt`	Identified independent variables
`x`	Response or dependent variable
`prob`	Probability with values in [0,1].

Value

A quantile assosciated with prob.

`rain.mon` *Sample data: Rainfall station data over Sydney region*

Description

A dataset containing 732 rows (data length) and 15 columns (stations).

Usage

```
data(rain.mon)
```

scal2freqM *Scale to frequency by Matlab*

Description

Scale to frequency by Matlab

Usage

```
scal2freqM(wf, scale, delta)
```

Arguments

wf	wavelet name
scale	a scale
delta	the sampling period.

Value

A vector of two numbers: frequency and period.

Examples

```
delta <- 1 / 12 # monthly data
scales <- 2^(1:7)

for (wf in c("haar", "d4", "d6", "d8", "d16")[1:5]) {
  df1 <- scal2freqM(wf, scales, delta)
  df2 <- scal2freqR(wf, scales, delta)

  print(cbind(df1$frequency, df2$frequency))
}
```

scal2freqR *Scale to frequency by R*

Description

Scale to frequency by R

Usage

```
scal2freqR(wf, scale, delta)
```

Arguments

wf	wavelet name
scale	a scale
delta	the sampling period.

Value

A vector of two numbers: frequency and period.

Examples

```
delta <- 1 / 12 # monthly data
scales <- 2^(1:7)

for (wf in c("haar", "d4", "d6", "d8", "d16")[1:5]) {
  df1 <- scal2freqM(wf, scales, delta)
  df2 <- scal2freqR(wf, scales, delta)

  print(cbind(df1$frequency, df2$frequency))
}
```

SPI.12 — *Sample data: Standardized Precipitation Index with 12 month accumulation period.*

Description

A dataset containing 1200 rows (data length) and 252 columns.

Usage

```
data(SPI.12)
```

SPI.calc — *Calculate Standardized Precipitation Index, SPI*

Description

Calculate Standardized Precipitation Index, SPI

Usage

```
SPI.calc(prec.zoo, sc = 24, method = "mle")
```

Arguments

prec.zoo	A zoo series contain date and rainfall vector/matrix.
sc	The accumulation period in months. Commonly 6, 12, 24, 36, and 48 months.
method	A character string coding for the fitting method: "mle" for 'maximum likelihood estimation', "mme" for 'moment matching estimation', "qme" for 'quantile matching estimation' and "mge" for 'maximum goodness-of-fit estimation'.

Value

A matrix of time series.

Examples

```
data(rain.mon)

## compute SPI
SPI <- SPI.calc(window(rain.mon, start = c(1949, 1), end = c(2009, 12)), sc = 12)

## plot
par(mfrow = c(3, 5))
for (i in seq_len(ncol(SPI))) plot(SPI[, i])
```

---

stepwise.VT *Calculate stepwise high order VT in calibration*

Description

Calculate stepwise high order VT in calibration

Usage

```
stepwise.VT(
  data,
  alpha = 0.1,
  nvarmax = 4,
  mode = c("MRA", "MODWT", "AT"),
  wf,
  J,
  method = "dwt",
  pad = "zero",
  boundary = "periodic",
  cov.opt = "auto",
  flag = "biased",
  detrend = FALSE
)
```

Arguments

data	A list of data, including response and predictors
alpha	The significance level used to judge whether the sample estimate is significant. A default alpha value is 0.1.
nvarmax	The maximum number of variables to be selected.
mode	A mode of variance transformation, i.e., MRA, MODWT, or AT
wf	Wavelet family
J	Specifies the depth of the decomposition. This must be a number less than or equal to log(length(x),2).
method	Either "dwt" or "modwt" of MRA.
pad	The method used for extend data to dyadic size. Use "per", "zero", or "sym".
boundary	Character string specifying the boundary condition. If boundary=="periodic" the default, then the vector you decompose is assumed to be periodic on its defined interval, if boundary=="reflection", the vector beyond its boundaries is assumed to be a symmetric reflection of itself.
cov.opt	Options of Covariance matrix sign. Use "pos", "neg", or "auto".
flag	Biased or Unbiased variance transformation.
detrend	Detrend the input time series or just center, default (F).

Value

A list of 2 elements: the column numbers of the meaningful predictors (cpy), and partial informational correlation (cpyPIC).

References

Sharma, A., Mehrotra, R., 2014. An information theoretic alternative to model a natural system using observational information alone. Water Resources Research, 50(1): 650-660.

Jiang, Z., Sharma, A., & Johnson, F. (2021). Variable transformations in the spectral domain – Implications for hydrologic forecasting. Journal of Hydrology, 126816.

Examples

```
### Real-world example
data("rain.mon")
data("obs.mon")
mode <- switch(1,
  "MRA",
  "MODWT",
  "AT"
)
wf <- "d4"
station.id <- 5 # station to investigate
#SPI.12 <- SPEI::spi(rain.mon, scale = 12)$fitted
SPI.12 <- SPI.calc(window(rain.mon, start=c(1949,1), end=c(2009,12)),sc=12)
lab.names <- colnames(obs.mon)
```

```
# plot.ts(SPI.12[,1:10])

x <- window(SPI.12[, station.id], start = c(1950, 1), end = c(1979, 12))
dp <- window(obs.mon[, lab.names], start = c(1950, 1), end = c(1979, 12))

data <- list(x = x, dp = matrix(dp, ncol = ncol(dp)))

dwt <- stepwise.VT(data, mode = mode, wf = wf, flag = "biased")

### plot transformed predictor before and after
cpy <- dwt$cpy
op <- par(mfrow = c(length(cpy), 1), mar = c(2, 3, 2, 1))
for (i in seq_along(cpy)) {
  ts.plot(cbind(dwt$dp[, i], dwt$dp.n[, i]), xlab = "NA", col = 1:2)
}
par(op)
```

`stepwise.VT.val` *Calculate stepwise high order VT in validation*

Description

Calculate stepwise high order VT in validation

Usage

```
stepwise.VT.val(data, J, dwt, mode = c("MRA", "MODWT", "AT"), detrend = FALSE)
```

Arguments

`data`	A list of data, including response and predictors
`J`	Specifies the depth of the decomposition. This must be a number less than or equal to log(length(x),2).
`dwt`	Output from dwt.vt(), including the transformation covariance
`mode`	A mode of variance transformation, i.e., MRA, MODWT, or AT
`detrend`	Detrend the input time series or just center, default (F)

Value

A list of objects, including transformed predictors

Examples

```
### Real-world example
data("rain.mon")
data("obs.mon")
mode <- switch(1,
  "MRA",
```

```
  "MODWT",
  "a trous"
)
wf <- "d4"
station.id <- 5 # station to investigate
#SPI.12 <- SPEI::spi(rain.mon, scale = 12)$fitted
SPI.12 <- SPI.calc(window(rain.mon, start=c(1949,1), end=c(2009,12)),sc=12)
lab.names <- colnames(obs.mon)
# plot.ts(SPI.12[,1:10])

#--------------------------------------
### calibration
x <- window(SPI.12[, station.id], start = c(1950, 1), end = c(1979, 12))
dp <- window(obs.mon[, lab.names], start = c(1950, 1), end = c(1979, 12))

data <- list(x = x, dp = matrix(dp, ncol = ncol(dp)))
dwt <- stepwise.VT(data, mode = mode, wf = wf, flag = "biased")
cpy <- dwt$cpy
#--------------------------------------
### validation
x <- window(SPI.12[, station.id], start = c(1980, 1), end = c(2009, 12))
dp <- window(obs.mon[, lab.names], start = c(1980, 1), end = c(2009, 12))

data.n <- list(x = x, dp = matrix(dp, ncol = ncol(dp)))
dwt.val <- stepwise.VT.val(data = data.n, dwt = dwt, mode = mode)

### plot transformed predictor before and after
op <- par(mfrow = c(length(cpy), 1), mar = c(0, 3, 2, 1))
for (i in seq_along(cpy))
{
  ts.plot(cbind(dwt.val$dp[, i], dwt.val$dp.n[, i]), xlab = "NA", col = 1:2)
}
par(op)
```

`wave.var` *Produces an estimate of the multiscale variance along with approximate confidence intervals.*

Description

Produces an estimate of the multiscale variance along with approximate confidence intervals.

Usage

```
wave.var(x, type = "eta3", p = 0.025)
```

Arguments

x	DWT/MODWT/AT object

type	character string describing confidence interval calculation; valid methods are gaussian, eta1, eta2, eta3, nongaussian
p	(one minus the) two-sided p-value for the confidence interval

Value

Dataframe with as many rows as levels in the wavelet transform object. The first column provides the point estimate for the wavelet variance followed by the lower and upper bounds from the confidence interval.

References

Percival, D. B. (1995) Biometrika, 82, No. 3, 619-631.

Index