Package ‘cpcens’

August 2, 2019

Type Package

Title Changepoint Analysis using Censored Time Series Data

Version 0.1.0

Date 2019-07-01

Author Hajra Siddiqa<hajrasiddiqa92@gmail.com>, Sajid Ali<sajidali@qau.edu.pk>, Ismail Shah<ishah@qau.edu.pk>

Maintainer Sajid Ali<sajidali@qau.edu.pk>

Depends R (>= 2.10)

Description To detect the changepoint, this package uses most recent changepoint, double cumulative sum binary segmentation, multiple changepoints in multivariate time series, analyzing each series in the panel independently, and analyzing aggregated data methods. This package is useful to simulate censored time series to detect the most recent changepoint in censored panel data as well as to assess prediction accuracy.

LazyLoad yes

LazyData yes

License GPL (>= 2)

Imports stats, utils, Rdpack, cents, tbart

RdMacros Rdpack

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-08-02 11:40:02 UTC

RoxygenNote 6.1.1

Encoding UTF-8

Suggests testthat

R topics documented:

`AR1.data` *Most recent changepoints*

Description

Detecting most recent changepoints using censored data generated from AR model.

Usage

```
AR1.data(n = 500, N = 100, K = 5, eps = 1, rho = 0.6, mu = 0,
  siga = 1, rates = c(NA, 0.2), Mrate = 0)
```

Arguments

`n`	length of series, default 500.The size of series (n) should be greater than 200.
`N`	number of series, default 100.
`K`	number of most recent changepoints, default 5.
`eps`	size of the mean change at the most recent changepoint.
`rho`	ar coefficients
`mu`	mean
`siga`	standard deviation of innovations

rates	either a vector of length 2 or a matrix with n rows and 2 columns. In the vector case, the first element indicates the left-censor rate and the second element indicates the right-censor rate. Set to NA if there is no censoring. Interval censored data corresponds to setting both a left-censor rate and right-censor rate. The default setting indicates a right-censor rate 0.2 with no left censoring. The vector case handles single censoring and the matrix case is for multiple censor points. In this case each column indicates the corresponding censoring for each observation.
Mrate	fraction of missing values. Default is 0

Value

an object of class 'censored' which is a list with four elements. First element, 'data', is the censored time series. Second element, 'mrc',indicates most recent changepoints. Third element, 'series.mrc', indicates which series is affecting from which most recent changepoint . Fourth element, 'series.chpts' indicates the changepoints in each series.

Examples

```
#Default example
library(cpcens)
ans<-AR1.data()
#example (right censoring)
out = AR1.data ( n=500 , N = 100 , K = 5 , eps = 1 , rho=0.2,
mu = 0,  siga = 1, rates = c(NA,0.4), Mrate=0 )
#example (left censoring)
out = AR1.data( n=500 , N = 100 , K = 5 , eps = 1 , rho=0.4,
mu = 0,  siga = 1, rates = c(0.3,NA), Mrate=0 )
#example (interval censoring)
out = AR1.data ( n=500 , N = 100 , K = 5 , eps = 1 , rho=0.4,
mu = 0,  siga = 1, rates = c(0.4,0.5), Mrate=0 )
```

Bin_segAR — *Most recent changepoints from dcbs method using censored AR time-series.*

Description

Detecting most recent changepoints from double CUSUM binary segmentation (DCBS) method (Cho, 2016) after generating censored data from AR model. The DCBS method focuses on finding the position of detected changepoints. The consistency of the double CUSUM statistic based on the binary segmentation algorithm that is established in terms of both number and locations of estimated changepoints.

Usage

```
Bin_segAR(data, beta = 10)
```

Arguments

data	a censored data matrix obtained from AR1.data.
beta	threshold for testing whether or not a change is significant, default 10.

Value

indicates the most recent changepoint in each series.

See Also

AR1.data

Examples

```
library(cpcens)
#example(right censoring)
# The length of series(n) should be greater than 200.
sim=AR1.data(n = 500, N = 100, K = 5, eps = 1,
rho = 0.4, mu = 0, siga = 1, rates = c(NA, 0.4), Mrate = 0)
data=sim$data
ans = Bin_segAR(data,beta=10)
```

Bin_segMA *Most recent changepoints from dcbs method using censored MA time-series.*

Description

Detecting most recent changepoints from double CUSUM binary segmentation algorithm (DCBS) method (Cho, 2016) after generating censored data from MA model.

Usage

```
Bin_segMA(data, beta)
```

Arguments

data	a censored data matrix obtained from MA1.data.
beta	threshold for testing whether or not a change is significant, default 10.

Value

indicates the most recent changepoint in each series.

References

Cho, H. (2016). Change-point detection in panel data via double CUSUM statistic. Electronic Journal of Statistics, 10(2):2000–2038.

See Also

MA1.data

Examples

```
library(cpcens)
#example(right censoring)
sim=MA1.data(n = 500, N = 100, K = 5, eps = 1,
rho = 0.4, mu = 0, siga = 1, rates = c(NA, 0.4), Mrate = 0)
data=sim$data
ans = Bin_segMA(data,beta=10)
```

---

censoredex — *Successive readings of a toxic substance in the Niagara River near Fort Erie, Ontario and then converting it into panel data.*

Description

Niagara River at Fort Erie, successive readings of 12-Dichloro in units of ng/L measured approximately biweekly.

Usage

```
data("censoredex")
```

Format

A data matrix with 100 rows annd 144 columns.

Details

Dr. Abdel El-Shaarwai provided through Environment Canada some a special water quality time series that is of great practical interest. The time series is from Station ON02HA0019 (Fort Erie) on the water quality of the Niagara River. There are more than 500 water quality parameters or variables of interest in this river. The water quality in this river is montiored by a joint U.S./Canada committee. The time series data are about water quality of the Niagara River where toxicity is of great interest,i.e., the chemical known as 12-Dichloro which is measured in units of ng/L (nanogram/liter) when dissolved in water. The data are measured approximately every two weeks over the period from March 1, 2001 to March 22, 2007, and in-total we have 144 left censored values. The observed censoring rate is CR = (21*100)/144 = 14%. The detection level for 12 Dichloro after March 24, 2005 dropped from 0.214 to 0.0878. There is only one censored value at Julian day number 1807, 75 complete observations and 20 censored ones before the change in censoring while there were 48 complete observations and only one censored observation from March 24, 2005 to the last observation on March 22, 2007. Although the data are time series, not panel data and to convert it into panel we simulated the panel data.

Source

Dr. Abdel El-Shaarwai, Environment Canada

References

N. M. Mohammad (2014). Censored time series analysis. Ph.D. Thesis, Western University

Examples

```
library(cpcens)
data("censoredex")
data=censoredex
dim(censoredex)
```

`ind` *Most recent changepoint using ind method*

Description

Analyzing each series in the panel independently (IND) method that is the simplest one to analyze all the series independently in the panel data and in each given series estimate the most recent changepoint. We use PELT for segmenting a time series into changing means, assumes normally distributed observations with changing mean but constant variance

Usage

```
ind(data, pen = 0)
```

Arguments

`data`	a censored data matrix
`pen`	(penalty term) default 2*log(n). If pen is equal to zero, penalty term will be equal to 2*log(n)

Value

indicates the most recent changepoint in each series .

Examples

```
#Default example
library(cpcens)
data("censoredex")
data=censoredex
n=144
N=100
out=ind(data, pen=0)
```

indAR	*Most recent changepoints from ind method using censored AR time-series.*

Description

Detecting most recent changepoints from ind method (analyzes all the series independently in the panel data and in each given series estimate the most recent changepoint) after generating censored data from AR model. We use PELT for segmenting a time series into changing means, assumes normally distributed observations with changing mean but constant variance.

Usage

```
indAR(data, pen = 0)
```

Arguments

`data`	a censored data matrix obtained from AR1.data .
`pen`	default 2*log(n). If pen is equal to zero, penalty term will be equal to 2*log(n).

Value

indicates the most recent changepoint in each series .

See Also

AR1.data

Examples

```
#Default example
library(cpcens)
sim=AR1.data()
data=sim$data
ans = indAR(data,pen)
#example(right censoring)
#The length of series(n) should be greater than 200.
sim=AR1.data(n = 500, N = 100, K = 5, eps = 1,
rho = 0.6, mu = 0, siga = 1, rates = c(NA, 0.2), Mrate = 0)
data=sim$data
ans = indAR(data,pen=0)
```

indMA — *Most recent changepoints from ind method using censored MA time-series.*

Description

Detecting most recent changepoints from ind method (analyzes all the series independently in the panel data and in each given series estimate the most recent changepoint) after generating censored data from MA model. PELT is used for segmenting a time series into changing means, assumes normally distributed observations with changing mean but constant variance

Usage

```
indMA(data, pen = 0)
```

Arguments

`data`	a censored data matrix obtained from MA1.data.
`pen`	default 2*log(n). If pen is equal to zero , penalty term will be equal to 2*log(n)

Value

indicates the most recent changepoint in each series.

See Also

MA1.data

Examples

```
#Default example
library(cpcens)
sim=MA1.data()
data=sim$data
ans = indMA(data,pen)
#example(right censoring)
# The size of series(n) should be greater than 200.
sim=MA1.data(n = 500, N = 100, K = 5, eps = 1,
rho = 0.6, mu = 0, siga = 1, rates = c(NA, 0.2), Mrate = 0)
data=sim$data
ans = indMA(data,pen=0)
```

`MA1.data` *Most recent changepoints*

Description

Detecting most recent changepoints using censored data generated from MA model.

Usage

```
MA1.data(n = 500, N = 100, K = 5, eps = 1, rho = 0.6, mu = 0,
  siga = 1, rates = c(NA, 0.2), Mrate = 0)
```

Arguments

`n`	length of series, default 500. The size of series(n) should be greater than 200.
`N`	number of series, default 100.
`K`	number of most recent changepoints, default 5.
`eps`	size of the mean change at the most recent changepoint.
`rho`	ma coefficients
`mu`	mean
`siga`	standard deviation of innovations
`rates`	either a vector of length 2 or a matrix with n rows and 2 columns. In the vector case, the first element indicates the left-censor rate and the second element indicates the right-censor rate. Set to NA if there is no censoring. Interval censored data corresponds to setting both a left-censor rate and right-censor rate. The default setting indicates a right-censor rate 0.2 with no left censoring. The vector case handles single censoring and the matrix case is for multiple censor points. In this case each column indicates the corresponding censoring for each observation.
`Mrate`	fraction of missing values. Default is 0

Value

an object of class 'censored' which is a list with four elements. First element, 'data', is the censored time series. Second element, 'mrc', indicates the most recent changepoints. Third element, 'series.mrc' indicates which series is affecting from the most recent changepoint. Fourth element, 'series.chpts' indicates the changepoints in each series.

Examples

```
#Default example
library(cpcens)
ans<-MA1.data()
#example (right censoring)
# The size of series(n) should be greater than 200.
ans<-MA1.data ( n=500 , N = 100 , K = 5 , eps = 1 , rho=0.2,
mu = 0,  siga = 1, rates = c(NA,0.4), Mrate=0 )
```

`mrc.mean` *Segmenting data generated from AR1.data/MA1.data using PELT function.*

Description

To find changepoints using mrc method, segmenting the data (obtained from AR1.data/MA1.data) using PELT (Killick, Fearnhead and Eckley 2012) function in such a way that cost is minimum .

Usage

```
mrc.mean(data, beta = 1.5 * log(n))
```

Arguments

`data`	a censored data matrix obtained from AR1.data/ MA1.data .
`beta`	default 1.5*log(n).

Value

data

References

Killick, R., Fearnhead, P., and Eckley, I. A. (2012). Optimal detection of changepoints with a linear computational cost. Journal of the American Statistical Association, 107(500):1590–1598.

See Also

AR1.data, MA1.data

Examples

```
#example(right censoring)
library(cpcens)
n=500
N=100
# Generate censored data using AR model
# The size of series(n) should be greater than 200.
sim=AR1.data(n = 500, N = 100, K = 5, eps = 1,
rho = 0.4, mu = 0, siga = 1, rates = c(NA, 0.4), Mrate = 0)
data=sim$data
mrc = mrc.mean( data , beta = 1.5*log(n) )
mrc
#example(left censoring)
library(cpcens)
n=500
N=100
# Generate censored data using MA model
```

```
# The size of series(n) should be greater than 200.
sim=MA1.data(n = 500, N = 100, K = 5, eps = 1,
rho = 0.4, mu = 0, siga = 1, rates = c(0.6,NA), Mrate = 0)
data=sim$data
mrc = mrc.mean( data , beta = 1.5*log(n) )
mrc
```

---

mrc.mean1 *Segmenting data using PELT funtion.*

Description

To find changepoints using mrc method consisting of many related univariate timeseries, segmenting the data using PELT function (Killick, Fearnhead, and Eckley 2012) in such a way that cost is minimum.

Usage

```
mrc.mean1(data, beta = 1.5 * log(n))
```

Arguments

data	a censored data matrix.
beta	default 1.5*log(n).

Value

data

Examples

```
#example(right censoring)
library(cpcens)
data("censoredex")
data=censoredex
n=144
mrc = mrc.mean1( data , beta = 1.5*log(n) )
mrc
```

`multiple.mrc` *Most recent changepoints from mrc method using censored AR/MA timeseries.*

Description

Detecting most recent changepoints from mrc method consisting of many related univariate time-series (Bardwell, Eckley, Fearnhead, and Smith, 2016) after generating censored data from AR/MA model and pools information across the time-series by solving the K-median problem using tb.raw (Teitz and Bart, 1968).

Usage

```
multiple.mrc(mrc, pmax = 10, alpha = 2, elbow.thresh = 0.5,
  n = 500)
```

Arguments

`mrc`	data obtained from mrc.mean
`pmax`	Maximum number of most recent changepoints to search for. Default value pmax=10.
`alpha`	The variable specific penalty used to penalise the addition of a given changepoint into a given variable. Default value alpha = 2.
`elbow.thresh`	default 0.5.
`n`	length of series

Value

indicates the most recent changepoint in each series .

References

Teitz, M. B. and Bart, P. (1968). Heuristic methods for estimating the generalized vertex median of a weighted graph. Operations Research, 16(5):955–961.

Bardwell, L., Fearnhead, P., Eckley, I. A., Smith, S., and Spott, M. (2019). Most recent changepoint detection in panel data. Technometrics, 61(1):88–98.

See Also

mrc

Examples

```
#'#example(left censoring)
library(cpcens)
n=300
N=100
# Generate censored data using MA model
sim=MA1.data(n = 300, N = 100, K = 5, eps = 1,
rho = 0.4, mu = 0, siga = 1, rates = c(0.6,NA), Mrate = 0)
data=sim$data
mrc = mrc.mean( data , beta = 1.5*log(n) )
c = multiple.mrc(  mrc , pmax=10, alpha = 2 , elbow.thresh = 0.5, n=500 )
p.hat = c$MDL
mrc.chpts = c$locs[[p.hat]][ c$affected[[p.hat]] ]
mrc.chpts
```

---

multiple.mrc1 *Most recent changepoints from mrc method.*

Description

Detecting most recent changepoints using mrc method consisting of many related univariate time-series (Bardwell, Eckley, Fearnhead, and Smith, 2016) and pools information across the time-series by solving the K-median problem using tb.raw (Teitz and Bart, 1968).

Usage

```
multiple.mrc1(mrc, pmax = 10, alpha = 2, elbow.thresh = 0.5,
  n = 144)
```

Arguments

mrc	data obtained from mrc.mean1
pmax	Maximum number of most recent changepoints to search for. Default value pmax=10.
alpha	The variable penalty used to penalise the addition of a given changepoint into a given variable. Default value alpha = 2.
elbow.thresh	default 0.5
n	length of series

Value

indicates the most recent changepoint in each series .

See Also

mrc.mean1

Examples

```
#example(right censoring)
library(cpcens)
data("censoredex")
data=censoredex
n=144
N=100
mrc = mrc.mean1( data , beta = 1.5*log(n) )
c = multiple.mrc1( mrc , pmax=10, alpha = 2 , elbow.thresh = 0.5 , n=144)
p.hat = c$MDL
mrc.chpts = c$locs[[p.hat]][ c$affected[[p.hat]] ]
mrc.chpts
```

PELT *Most recent changepoints from AGG method.*

Description

Detecting most recent changepoints from AGG method (detect changepoint in univariate time series). We use PELT for segmenting a time series into changing means, assuming normally distributed observations with changing mean but constant variance.

Usage

```
PELT(data, pen)
```

Arguments

data	a censored data matrix. And then we add this data matrix column wise and resulting data use as first argument in PELT function.
pen	(penalty term) default 200*log(dim(data)[2]. Here dim(data)[2] means length of series (n).

Value

indicates the most recent changepoint in each series .

Examples

```
#example
library(cpcens)
data("censoredex")
data=censoredex
n=144
N=100
agg = apply( data , 2 , sum )
pagg = PELT( agg , 200*log(dim(data)[2]) )
agg.chpts = rep( rev( pagg$cpts )[1] , N )
```

PELT.ar *Most recent changepoints from AGG method using censored AR time-series.*

Description

Detecting most recent changepoints uing AGG method (detect changepoint in univariate time series) after generating censored data from AR model. We use PELT for segmenting a time series into changing mean, assuming normally distributed observations with changing mean but constant variance.

Usage

```
PELT.ar(data, pen = 200 * log(dim(data)[2]))
```

Arguments

data	a censored data matrix obtained from AR1.data . And then we add this data matrix column wise and use this as first argument in PELT.ar function.
pen	penalty term, default 200*log(dim(data)[2]). Here dim(data)[2] means consider length of series (n). The PELT function return cpts (Vector of changepoints in segmentation) and F (optimal cost of segmenting series upto time t).

Value

indicates the most recent changepoint in each series .

See Also

AR1.data

Examples

```
#example
library(cpcens)
# The size of series(n) should be greater than 200.
sim=AR1.data(n = 500, N = 100, K = 5, eps = 1,
  rho = 0.6, mu = 0, siga = 1, rates = c(NA, 0.2), Mrate = 0)
data=sim$data
N=100
agg = apply( data , 2 , sum )
pagg = PELT.ar( agg , 200*log(dim(data)[2]) )
agg.chpts = rep( rev( pagg$cpts )[1] , N )
```

PELT.ma *Most recent changepoints from AGG method using censored MA time-series.*

Description

Detecting most recent changepoints from AGG method (detect changepoint in univariate time series) after generating censored data from MA model. We use PELT for segmenting a time series into changing mean, assuming normally distributed observations with changing mean but constant variance.

Usage

```
PELT.ma(data, pen = 200 * log(dim(data)[2]))
```

Arguments

data	a censored data matrix obtained from MA1.data . And then we add this data matrix column wise and use this as a first argument in PELT.ma function.
pen	penalty term, default 200*log(dim(data)[2]). Here dim(data)[2] means length of series (n).

Value

indicates the most recent changepoint in each series .

See Also

MA1.data

Examples

```
#example
library(cpcens)
# The size of series(n) should be greater than 200.
sim=MA1.data(n = 500, N = 100, K = 5, eps = 1,
    rho = 0.6, mu = 0, siga = 1, rates = c(NA, 0.2), Mrate = 0)
data=sim$data
N=100
agg = apply( data , 2 , sum )
pagg = PELT.ma( agg , 200*log(dim(data)[2]) )
agg.chpts = rep( rev( pagg$cpts )[1] , N )
```

PELT.MV *Most recent changepoints from MV method.*

Description

Detecting most recent changepoints from MV methd (Lavielle and Teyssiere, 2006) deal with multivariate data which is modeling the data within each segment as a multivariate (MV) Gaussian having a given covariance.

Usage

```
PELT.MV(data, beta = 101 * log(dim(data)[2]))
```

Arguments

`data`	a censored data matrix.
`beta`	default 101*log(dim(data)[2])). Here dim(data)[2] means consider size(length) of series (n).

Value

indicates the most recent changepoint in each series .

Examples

```
# example
library(cpcens)
data("censoredex")
data=censoredex
N=100
n=144
pmv = PELT.MV( data , 101*log(dim(data)[2]) )
mv.chpts =  rep( rev( pmv$cpts )[1] , N )
```

PELT.MVar *Most recent changepoints from MV method using censored AR time-series.*

Description

Detecting most recent changepoints from MV methd (Lavielle and Teyssiere, 2006) deal with multivariate data which is modeling the data within each segment as a multivariate (MV) Gaussian having a given covariance after generating censored data from AR model.

Usage

```
PELT.MVar(data, beta = 101 * log(dim(data)[2]))
```

Arguments

`data`	a censored data matrix obtained from AR1.data .
`beta`	default 101*log(dim(data)[2])). Here dim(data)[2] means consider size(length) of series (n).

Value

indicates the most recent changepoint in each series .

References

Lavielle, M. and Teyssiere, G. (2006). Detection of multiple changepoints in multivariate time series.Lithuanian Mathematical Journal, 46(3):287-306

See Also

AR1.data

Examples

```
# example (Right censoring)
library(cpcens)
# The size of series(n) should be greater than 200.
sim=AR1.data(n = 500, N = 100, K = 5, eps = 1,
rho = 0.6, mu = 0, siga = 1, rates = c(NA, 0.2), Mrate = 0)
data=sim$data
N=100
pmv = PELT.MVar( data , 101*log(dim(data)[2]) )
mv.chpts =  rep( rev( pmv$cpts )[1] , N )
```

PELT.MVma — *Most recent changepoints from MV method using censored MA time-series.*

Description

Detecting most recent changepoints from MV methd (Lavielle and Teyssiere, 2006) deal with multivariate data which is modeling the data within each segment as a multivariate (MV) Gaussian having a given covariance after generating censored data from MA model.

Usage

```
PELT.MVma(data, beta = 101 * log(dim(data)[2]))
```

Arguments

`data`	a censored data matrix obtained from MA1.data .
`beta`	default 101*log(dim(data)[2])). Here dim(data)[2] means consider size(length) of series (n).

Value

indicates the most recent changepoint in each series .

See Also

MA1.data

Examples

```
# example
library(cpcens)
# The size of series(n) shoul be greater than 200.
sim=MA1.data(n = 500, N = 100, K = 5, eps = 1,
 rho = 0.6, mu = 0, siga = 1, rates = c(NA, 0.2), Mrate = 0)
data=sim$data
N=100
pmv = PELT.MVma( data , 101*log(dim(data)[2]) )
mv.chpts =  rep( rev( pmv$cpts )[1] , N )
```

---

predar.mse *Mean Squared Error using censored data generated from AR model*

Description

(Accuracy of prediction) One can find the mean squared error (MSE) to check how these different methods (ind, dcbs, mrc, agg, mv) perform if the objective is to make prediction.

Usage

```
predar.mse(chpts, data.train, data.test)
```

Arguments

`chpts`	changepoints that are obtained using the discussed method (ind, dcbs, mrc, agg, mv)
`data.train`	divide generated censored data from AR model into data.train and data.test. Here, we consider n=500 (size of each series) and N=100 (number of series) so we have a matrix of N*n. In data.train we leave out the five data points at the end of each series.
`data.test`	Remaining dataset (five time points at the end of each series) will be considered as data.test.

Value

return mean squared error (MSE)

See Also

AR1.data, indAR, Bin_segAR, PELT.MVar

Examples

```
# example
#mean squared error to check the accuracy of ind method using
#censored data generated from AR model.
# data generated through AR model considering 60% censoring rate
#(Left censoring) and missing rate is equal to zero
library(cpcens)
sim = AR1.data ( n=500 , N = 100 , K = 5 , eps = 1 , rho=0.6,
mu = 0,  siga = 1, rates = c(0.6,NA), Mrate=0 )
data=sim$data
n=500
N=100
# training and test
data.train = sim$data[,1:(n-5)]
data.test = sim$data[,(n-4):n]
##If pen is equal to zero, penalty term will be equal to 2*log(n)
indar.chpts=indAR(data.train, pen=0)
indar.mse = predar.mse( indar.chpts , data.train , data.test )
indar.mse
#example
#mean squared error to check the accuracy of dcbs method using
#censored data generated from AR model.
library(cpcens)
# data generated through AR model considering 20% censoring rate
#(Right censoring) and missing rate is equal to zero
sim = AR1.data ( n=500 , N = 100 , K = 5 , eps = 1 , rho=0.4,
mu = 0,  siga = 1, rates = c(NA,0.2), Mrate=0 )
data=sim$data
n=500
N=100
# training and test
data.train = sim$data[,1:(n-5)]
data.test = sim$data[,(n-4):n]
dcbsar.chpts= Bin_segAR(data.train, 10)
dcbsar.mse = predar.mse( dcbsar.chpts , data.train , data.test )
dcbsar.mse
```

---

predma.mse *Mean Squared Error using censored data generated from MA model*

Description

(Accuracy of prediction) One can find the mean squared error (MSE) to check how these different methods (ind, dcbs, mrc, agg, mv) perform if the objective is to make prediction.

Usage

```
predma.mse(chpts, data.train, data.test)
```

Arguments

chpts	changepoints that are obtained using any discussed method (ind, dcbs, mrc, agg, mv)
data.train	divide generated censored data from MA model into data.train and data.test. Here we consider n=500 (size of each series) and N=100 (number of series) so we have a matrix of N*n. In data.train we leave out the five data points at the end of each series.
data.test	Remaining dataset (five time points at the end of each series) will be considered as data.test.

Value

return mean squared error (MSE)

See Also

MA1.data, indMA, Bin_segMA, PELT.MVma

Examples

```
# example
#mean squared error to check the accuracy of ind method using
#censored data generated from MA model.
# data generated through MA model considering 60% censoring rate
#(Left censoring) and missing rate is equal to zero
library(cpcens)
sim = MA1.data ( n=500 , N = 100 , K = 5 , eps = 1 , rho=0.6,
mu = 0,  siga = 1, rates = c(0.6,NA), Mrate=0 )
data=sim$data
n=500
N=100
# training and test
data.train = sim$data[,1:(n-5)]
data.test = sim$data[,(n-4):n]
##If pen is equal to zero, penalty term will be equal to 2*log(n)
indma.chpts=indMA(data.train, pen=0)
indma.mse = predma.mse( indma.chpts , data.train , data.test )
indma.mse
#example
#mean squared error to check the accuracy of dcbs method using
#censored data generated from MA model.
library(cpcens)
# data generated through MA model considering 20% censoring rate
#(Right censoring) and missing rate is equal to zero
sim = MA1.data ( n=500 , N = 100 , K = 5 , eps = 1 , rho=0.4,
mu = 0,  siga = 1, rates = c(NA,0.2), Mrate=0 )
```

```
data=sim$data
n=500
N=100
# training and test
data.train = sim$data[,1:(n-5)]
data.test = sim$data[,(n-4):n]
dcbsma.chpts= Bin_segMA(data.train, 10)
dcbsma.mse = predma.mse( dcbsma.chpts , data.train , data.test )
dcbsma.mse
#example
#mean squared error to check the accuracy of mv method using
#censored data generated from MA model.
library(cpcens)
# data generated through MA model considering 60% censoring rate
#(Right censoring) and missing rate is equal to zero
sim = MA1.data ( n=500 , N = 100 , K = 5 , eps = 1 , rho=0.4,
mu = 0,  siga = 1, rates = c(NA,0.6), Mrate=0 )
data=sim$data
n=500
N=100
# training and test
data.train = sim$data[,1:(n-5)]
data.test = sim$data[,(n-4):n]
pmv = PELT.MVma( data.train , 101*log(dim(data.train)[2]) )
mv.chpts =  rep( rev( pmv$cpts )[1] , N )
mv.mse = predma.mse( mv.chpts , data.train , data.test )
```

`rcar` *Simulate censored AR time series*

Description

Randomly Generate Censored AR

Usage

```
rcar(n = 100, ar = 0.6, mu = 0, siga = 1, rates = c(NA, 0.2),
  Mrate = 0)
```

Arguments

`n`	length of series, default 100.
`ar`	ar coefficients
`mu`	mean
`siga`	standard deviation of innovations
`rates`	either a vector of length 2 or a matrix with n rows and 2 columns. In the vector case, the first element indicates the left-censor rate and the second element

indicates the right-censor rate. Set to NA if there is no censoring. Interval censored data corresponds to setting both a left-censor rate and right-censor rate. The default setting indicates a right-censor rate 0.2 with no left censoring. The vector case handles single censoring and the matrix case is for multiple censor points. In this case each column indicates the corresponding censoring for each observation.

Mrate	fraction of missing values. Default is 0

Value

an object of class 'censored' which is a list with three elements. First element, 'y', is the censored time series. Second element, 'iy', indicates for each observed valued "o", "L", "R", NA according to whether the value is fully observed, left-censored, right-censored, or missing. Third element, 'censorPts', is a matrix with 2 columns indicating the censor point or NA if no censoring is applicable. Note that censorPts does not indicate if the observation was actually censored since this depends on the unknown latent variable. An observation is censored if and only if the corresponding entry in iy is either "L" or "R". See example below

Examples

```
#Default example
library(cpcens)
ans<-rcar()
#example (right censoring)
ans = rcar (n=100 , ar=0.2, mu=0 , siga=1, rates=c(NA,0.7), Mrate=0 )
#example (left censoring)
ans = rcar (n=100 , ar=0.7, mu=0 , siga=1, rates=c(0.3,NA), Mrate=0 )
#example (interval censoring)
ans = rcar (n=100 , ar=0.7, mu=0 , siga=1, rates=c(0.1,0.1), Mrate=0 )
```

---

rcarma — *Simulate censored ARMA time series*

Description

Randomly Generate Censored ARMA

Usage

```
rcarma(n = 100, ar = 0.6, ma = 0.4, mu = 0, siga = 1,
  rates = c(NA, 0.2), Mrate = 0)
```

Arguments

n	length of series, default 100.
ar	ar coefficients
ma	ma coefficients

mu	mean
siga	standard deviation of innovations
rates	either a vector of length 2 or a matrix with n rows and 2 columns. In the vector case, the first element indicates the left-censor rate and the second element indicates the right-censor rate. Set to NA if there is no censoring. Interval censored data corresponds to setting both a left-censor rate and right-censor rate. The default setting indicates a right-censor rate 0.2 with no left censoring. The vector case handles single censoring and the matrix case is for multiple censor points. In this case each column indicates the corresponding censoring for each observation.
Mrate	fraction of missing values. Default is 0

Value

an object of class 'censored' which is a list with three elements. First element, 'y', is the censored time series. Second element, 'iy', indicates for each observed valued "o", "L", "R", NA according to whether the value is fully observed, left-censored, right-censored, or missing. Third element, 'censorPts', is a matrix with 2 columns indicating the censor point or NA if no censoring is applicable. Note that censorPts does not indicate if the observation was actually censored since this depends on the unknown latent variable. An observation is censored if and only if the corresponding entry in iy is either "L" or "R". See example below

Examples

```
#Default example
library(cpcens)
ans<-rcarma()
#example (right censoring)
ans = rcarma (n=100 , ar=0.2, ma = 0.6, mu=0 , siga=1, rates=c(NA,0.7), Mrate=0 )
#example (left censoring)
ans = rcarma (n=100 , ar=0.7, ma = 0.3,  mu=0 , siga=1, rates=c(0.3,NA), Mrate=0 )
#example (interval censoring)
ans = rcarma (n=100 , ar=0.7, ma = 0.2, mu=0 , siga=1, rates=c(0.25,0.25), Mrate=0 )
```

---

rcma *Simulate censored MA time series*

Description

Randomly Generate Censored MA

Usage

```
rcma(n = 100, ma = 0.6, mu = 0, siga = 1, rates = c(NA, 0.2),
  Mrate = 0)
```

Arguments

`n`	length of series, default 100.
`ma`	ma coefficients
`mu`	mean
`siga`	standard deviation of innovations
`rates`	either a vector of length 2 or a matrix with n rows and 2 columns. In the vector case, the first element indicates the left-censor rate and the second element indicates the right-censor rate. Set to NA if there is no censoring. Interval censored data corresponds to setting both a left-censor rate and right-censor rate. The default setting indicates a right-censor rate 0.2 with no left censoring. The vector case handles single censoring and the matrix case is for multiple censor points. In this case each column indicates the corresponding censoring for each observation.
`Mrate`	fraction of missing values. Default is 0

Value

an object of class 'censored' which is a list with three elements. First element, 'y', is the censored time series. Second element, 'iy', indicates for each observed valued "o", "L", "R", NA according to whether the value is fully observed, left-censored, right-censored, or missing. Third element, 'censorPts', is a matrix with 2 columns indicating the censor point or NA if no censoring is applicable. Note that censorPts does not indicate if the observation was actually censored since this depends on the unknown latent variable. An observation is censored if and only if the corresponding entry in iy is either "L" or "R".

Examples

```
#Default example
library(cpcens)
ans<-rcma()
#example (right censoring)
ans = rcma (n=100 , ma=0.2, mu=0 , siga=1, rates=c(NA,0.7), Mrate=0 )
#example (left censoring)
ans = rcma (n=100 , ma=0.7, mu=0 , siga=1, rates=c(0.3,NA), Mrate=0 )
#example (interval censoring)
ans = rcma (n=100 , ma=0.7, mu=0 , siga=1, rates=c(0.1,0.3), Mrate=0 )
```

Index