

Package ‘BLRPM’

July 21, 2025

Type Package

Title Stochastic Rainfall Generator Bartlett-Lewis Rectangular Pulse Model

Version 1.0

Date 2017-10-04

Author Christoph Ritschel

Maintainer Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

Depends R6

Description Due to a limited availability of observed high-resolution precipitation records with adequate length, simulations with stochastic precipitation models are used to generate series for subsequent studies [e.g. Khaliq and Cunmae, 1996, <doi:10.1016/0022-1694(95)02894-3>, Vandenberghe et al., 2011, <doi:10.1029/2009WR008388>]. This package contains an R implementation of the original Bartlett-Lewis rectangular pulse model (BLRPM), developed by Rodriguez-Iturbe et al. (1987) <doi:10.1098/rspa.1987.0039>. It contains a function for simulating a precipitation time series based on storms and cells generated by the model with given or estimated model parameters. Additionally BLRPM parameters can be estimated from a given or simulated precipitation time series. The model simulations can be plotted in a three-layer plot including an overview of generated storms and cells by the model (which can also be plotted individually), a continuous step-function and a discrete precipitation time series at a chosen aggregation level.

License GPL (>= 2)

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-10-06 11:42:37 UTC

Contents

Beta.fun	2
BL.acc	2
BL.sim	3
BL.stepfun	5

BLRPM.class	6
BLRPM.est	6
BLRPM.OF	8
BLRPM.sim	9
Delta.fun	10
plot.BLRPM	11
TS.acc	12
TS.stats	13
Index	14

Beta.fun	<i>Beta function needed in objective function</i>
----------	---

Description

Beta.fun is a help function for OF

Usage

Beta.fun(a, b)

Arguments

- a value specifying Parameter a
- b value specifying Parameter b

Value

beta returns value of Beta.fun for parameters a and b

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

BL.acc	<i>Accumulation of a precipitation stepfunction</i>
--------	---

Description

BL.acc accumulates the BLRPM stepfunction for a given accumulation time t.acc at a given accumulation level acc.val. An off se can be defined. The unit is typically hours.

Usage

BL.acc(sfn, t.acc = 240, acc.val = 1, offset = 0)

Arguments

sfn	stepfunction of precipitation
t.acc	value specifying the length of accumulated time series [h]
acc.val	value specifying the accumulation level [h]
offset	value specifying the offset of the accumulated time series [h]

Value

p.acc data.frame

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

Examples

```
lambda <- 4/240
gamma <- 1/10
beta <- 0.3
eta <- 2
mux <- 4
t.sim <- 240
t.acc <- t.sim
acc.val <- 1
offset <- 0

simulation <- BL.sim(lambda,gamma,beta,eta,mux,t.sim)
sfn <- BL.stepfun(simulation$cells)
ts <- BL.acc(sfn,t.acc,acc.val,offset)
```

BL.sim

Simulating storms and cells

Description

Bartlett-Lewis Rectangular Pulse Model

Usage

```
BL.sim(lambda = 4/240, gamma = 1/10, beta = 0.3, eta = 2, mux = 4,
        t.sim = 240)
```

Arguments

<code>lambda</code>	value specifying the generation rate of storms [1/h]
<code>gamma</code>	value specifying the storm duration [1/h]
<code>beta</code>	value specifying the generation rate of cells [1/h]
<code>eta</code>	value specifying the cell duration [1/h]
<code>mux</code>	value specifying the cell intensity [mm/h]
<code>t.sim</code>	value specifying the simulation time [h]

Details

Model description (Rodriguez-Iturbe et al., 1987):

The model is a combination of 2 poisson processes and simulates storms and cells. During the given simulation time storms are generated in a poisson process with rate `lambda`. Those storms are given a exponentially distributed duration with parameter `gamma`. During its duration the storm generates in a second poisson process cells with rate `beta`. The first cell has to be instantaneous at the time of the storm arrival. The cell duration is exponentially distributed with parameter `eta`. For the whole lifetime each cell is given a constant intensity which is exponentially distributed with parameter `1/mux`.

Aggregation:

The intensities of all cells alive at time `t` are summed up for total precipitation at time `t`.

Parameter estimation:

The model parameters (`lambda, gamma, beta, eta, mux`) can be estimated from simulated or observed precipitation time series using the method of moments. Certain moments, e.g. mean, variance can be calculated from the time series at different aggregation levels. These moments can also be calculated theoretically from model parameters. Both sets of statistics can be compared in an objective function, similar to a squared error estimator. By numerical optimization the model parameters can be tuned to match the time series characteristics. `BL.sim` generates model realisations of storms and cells by using given model parameters `lambda, gamma, beta, eta, mux` for a given simulation time `t.sim`

Value

`BL.sim` returns storms; data.frame of all storms containing information about occurrence time, end time and number of cells

`BL.sim` returns cells; data.frame of all cells containing information about occurrence time, end time, intensity and storm index

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

Examples

```
lambda <- 4/240
gamma <- 1/10
beta <- 0.3
eta <- 2
mux <- 4
t.sim <- 240
simulation <- BL.sim(lambda,gamma,beta,eta,mux,t.sim)
```

BL.stepfun*BLRPM continous stepfunction of precipitation*

Description

BL.stepfun calculates a continous stepfunction of precipitation from the data.frame cells

Usage

```
BL.stepfun(cells)
```

Arguments

cells	data.frame of all cells containing information about occurence time, end time, intensity and storm index
-------	--

Value

sfn returns stepfunction of precipitation

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

Examples

```
lambda <- 4/240
gamma <- 1/10
beta <- 0.3
eta <- 2
mux <- 4
t.sim <- 240
simulation <- BL.sim(lambda,gamma,beta,eta,mux,t.sim)
stepfun <- BL.stepfun(simulation$cells)
```

BLRPM.class

BLRPM class

Description

BLRPM.class defines a new class for objects of type BLRPM containing the information about storms, cells, stepfunction and the precipitation time series.

Usage

```
BLRPM.class
```

Format

An object of class R6ClassGenerator of length 24.

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

BLRPM.est

BLRPM Parameter Estimation function

Description

BLRPM.est estimates the five Bartlett-Lewis rectangular pulse model parameters λ , γ , β , η , μ_x for a given time series data. At first the time series statistics at given accumulation levels `acc.vals` are calculated. These statistics are given over to the parameter estimation algorithm together with parameter starting values `par`. An objective function `O.Fun` can be specified, default is `BLRPM.OF`. In addition the weights for different statistics and accumulation levels `weights.mean`, `weights.var`, `weights.cov`, `weights.pz` can be specified. For the BLRPM objective function the user can select the measure of distance between observation and model with `OF`: =1 quadratic, =2: quad extended, =3: absolute, =4: abs extended. A scale parameter controls different cases in the objective function for differences in the scale of duration parameters γ and η . If a debugging is wished, `debug` can be set to `TRUE` and a log file is created in working directory. Several `optim` parameters can be also defined. For specifics see `?optim`.

Usage

```
BLRPM.est(RR, acc.vals, pars.in, O.Fun,
weights.mean, weights.var, weights.cov, weights.pz, OF, debug,
scale, method, lower, upper, use.log, maxit, ndeps, trace)
```

Arguments

RR	vector of a precipitation time series
acc.vals	vector of different accumulation levels at which statistics are to be calculated
pars.in	vector specifying starting values of lambda,gamma,beta,eta,mux for optimization
O.Fun	objective function to be used during optimization
weights.mean	value for weight for mean value at first accumulation level
weights.var	vecotr of weights for variances, has to have length(acc.vals)
weights.cov	vecotr of weights for covariances, has to have length(acc.vals)
weights.pz	vecotr of weights for probability of zero rainfall, has to have length(acc.vals)
OF	value specifying the type of objective function. 1: quadratic, 2: quad symmetrized, 3: absolute, 4: abs symmetrized Note: quadratic symmetrized proofed to be most effective and fastest
debug	set TRUE if debugging is wished, default FALSE. Creates a log file in working directory
scale	value specifying the scaling between gamma and eta in the objective function
method	character defining the method to be used in optim, preferences are: "Nelder-Mead", "BFGS", "L-BFGS-B"
lower	vector specifying the lower boundary of parameters for "L-BFGS-B" method
upper	vector specifying the upper boundary of parameters for "L-BFGS-B" method
use.log	logical, set TRUE if logarithmic parameters during optimization should be used. Advantage: zero as lower boundary for parameters
maxit	value specifying the maximum number of itereations durion optimization
ndeps	vector specifying the change for each parameter during one iteration step
trace	value specifying output information of optim

Value

\$est returns vector of estimated parameters lambda,gamma,beta,eta,mux
 \$conv returns value of convergence of optimization, see optim for details
 \$mess returns character message about optimization if using "L-BFGS-B" method
 \$Z returns value of objective function for estimated parameters

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

Examples

```
t.sim=240

lambda <- 4/240
gamma <- 1/10
beta <- 0.3
eta <- 2
mux <- 4

pars <- c(lambda,gamma,beta,eta,mux)

sim <- BLRPM.sim(lambda,gamma,beta,eta,mux,t.sim)
est <- BLRPM.est(sim$RR,pars.in=pars,method="BFGS",use.log=TRUE)
```

BLRPM.OF

BLRPM objective function for parameter estimation

Description

BLRPM.OF is the objective function used for parameter estimation of the BLRPM parameters. Given a set of BLRPM parameters `par` this function calculates a set of model statistics at given accumulation time steps `acc.vals`. These model statistics are compared with given time series statistics `stats` in the objective function. The user is able to define weights for each statistic (has to be the same length as statistics input vector). Option for debugging is given. A scale parameter defines a criterium for which different kinds of model statistics are calculated. This criterium is mainly based on the timescale difference between storm duration parameter `gamma` and cell duration parameter `eta`. If `use.log` is true, the objective function needs logarithmic input parameters. The value of `OF` defines the kind of objective function to be used: 1= quadratic 2= quadratic extended 3= absolute 4= absolute extended.

Usage

```
BLRPM.OF(par, stats, acc.vals = c(1, 3, 12, 24), weights = rep(1,
  length(stats)), debug = FALSE, scale = 1, use.log = TRUE, OF = 2)
```

Arguments

<code>par</code>	vector specifying the five model parameters (<code>lambda,gamma,beta,eta,mux</code>) at which the objective function is to be calculated
<code>stats</code>	vector specifying the time series statistics to which the model is compared to
<code>acc.vals</code>	vector specifying the accumulation time steps at which the model statistics are calculated
<code>weights</code>	vector specifying the weight of each statistic in the objective function. Note: has to have the same length as <code>stats</code>
<code>debug</code>	logical defining if debugging of function has to be done

scale	value specifying the scale factor for comparison between duration parameters gamma and eta
use.log	logical defining if input parameters are logarithmic
OF	value specifying the type of objective function. 1: quadratic, 2: quad extended, 3: absolute, 4: abs extended

Value

Z returns value of objective function for input parameters and input statistics

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

BLRPM.sim

Simulating precipitation with the BLRPM

Description

BLRPM.sim is the main function for simulating precipitation with the Bartlett-Lewis rectangular pulse model. It generates storms and cells using the given five BLRPM parameters lambda, gamma, beta, eta, mux for a given simulation time t.sim. The function BLRPM.sim then accumulates a precipitation time series of length t.akk (typically the same as t.sim) with an accumulation time step interval from the generated storms and cells. An offset can be used to delay the precipitation time series for initialization reasons. BLRPM.sim returns a list of different variables and data.frames: Storms, Cells, Stepfun, Precip, time.

Usage

```
BLRPM.sim(lambda,gamma,beta,eta,mux,t.sim,t.acc,interval,offset)
```

Arguments

lambda	value specifying the expected storm generation rate [1/units.time]
gamma	value specifying the expected storm duration[1/units.time]
beta	value specifying the expected cell generation rate [1/units.time]
eta	value specifying the expected cell duration [1/units.time]
mux	value specifying the expected cell intensity [mm/unit.time]
t.sim	value specifying the simulation length [units.time]
t.acc	value specifying the length of the accumulated time series [units.time]. Note: if longer than t.sim only zeros are added after t.sim.
interval	value specifying the accumulation time step [units.time]
offset	value specifying the offset of the accumulated time series with respect to the start time of the simulation [units.time]. Note: negative values are not allowed.

Value

\$storms returns data.frame containing information about storms: start, end, number of cells
 \$cells returns data.frame containing information about cells: start, end, intensity, storm index
 \$sfm returns stepfunction used to accumulate precipitation time series
 \$RR returns vector of accumulated precipitation with time step interval [mm/interval]
 \$time returns vector of time steps [interval]

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

Examples

```
lambda <- 4/240
gamma <- 1/10
beta <- 0.3
eta <- 2
mux <- 4
t.sim <- 240
t.acc <- t.sim
interval <- 1
offset <- 0
simulation <- BLRPM.sim(lambda,gamma,beta,eta,mux,t.sim,t.acc=t.sim,interval,offset)
```

Delta.fun

Delta function needed in objective function

Description

Delta.fun is a help function for OF

Usage

```
Delta.fun(kappa, MStrich)
```

Arguments

kappa	value specifying Parameter kappa
MStrich	value specifying dimension of error correction in objective function

Value

Delta returns value of Delta.fun for kappa and MStrich

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

plot.BLRPM*Plotting of an object of class BLRPM*

Description

plot.BLRPM plots an object of class BLRPM returned by the function BLRPM.sim with an option to plot either only the storms and cells or to additionally plot the stepfunction and the precipitation time series in a multiframe plot.

Usage

```
## S3 method for class 'BLRPM'
plot(x, ..., OSC = FALSE, start.time = NULL,
     end.time = NULL, legend = TRUE, c.axis = 1.5, c.lab = 1.5,
     c.legend = 1.5)
```

Arguments

x	class BLRPM object which is returned by function BLRPM.sim
...	Arguments to be passed to methods, such as graphical parameters (see par).
OSC	logical determing type of plot. OSC=True only storms and cells are plotted. OSC=FALSE storms, cells, stepfunction and precipitation time series plotted.
start.time	numerical value setting the starting time of a time window to be plotted. Default is NULL, therefore start time is 0
end.time	numerical value setting the end time of a time window to be plotted. Default is NULL, meaning the plot will end with the last active cell
legend	logical setting the option for legend to be plotted or not
c.axis	numerical value for axis label size, default is 1.5
c.lab	numerical value for plot label size, default is 1.5
c.legend	numerical value for legend font size, default is 1.5

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

See Also

[plot](#)

Examples

```

lambda <- 4/240
gamma <- 1/10
beta <- 0.3
eta <- 2
mux <- 4
t.sim <- 240
t.acc <- t.sim
interval <- 1
offset <- 0
simulation <- BLRPM.sim(lambda,gamma,beta,eta,mux,t.sim,t.acc=t.sim,interval,offset)
plot(simulation,OSC=FALSE)

plot(simulation,OSC=TRUE,start.time=1,end.time=24)

```

TS.acc	<i>Accumulation of a time series</i>
--------	--------------------------------------

Description

TS.acc accumulates a given time series `x` at a given accumulation level `acc.val`. Minimum value for `acc.val` is 2 [unit time]

Usage

```
TS.acc(x, acc.val)
```

Arguments

<code>x</code>	vector of a time series
<code>acc.val</code>	value specifying the accumulation level, minimum value is 2

Value

`x.acc` TS.acc returns a vector of an accumulated time series

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

Examples

```

x <- rgamma(1000,1)
x.2 <- TS.acc(x,acc.val=2)

```

TS.stats	<i>calculating statistics of a time series needed for parameter estimation</i>
----------	--

Description

TS.stats calculates statistics of a given time series `x` at given accumulation levels `acc.vals`. The calculated statistics are the mean of the first accumulation level, the variance, auto-covariance lag-1 and the probability of zero rainfall of all given accumulation levels of the time series. These statistics are needed for estimating the BLRPM parameters.

Usage

```
TS.stats(x, acc.vals)
```

Arguments

<code>x</code>	vector of a time series
<code>acc.vals</code>	vector of accumulation levels, first value should be 1

Value

stats TS.stats returns a vector of statistics calculated at given accumulation levels

Author(s)

Christoph Ritschel <christoph.ritschel@met.fu-berlin.de>

Examples

```
time.series <- rgamma(1000, shape=1)
statistics <- TS.stats(time.series, acc.vals=c(1, 3, 12, 24))
```

Index

* **datasets**

BLRPM.class, [6](#)

Beta.fun, [2](#)

BL.acc, [2](#)

BL.sim, [3](#)

BL.stepfun, [5](#)

BLRPM.class, [6](#)

BLRPM.est, [6](#)

BLRPM.OF, [8](#)

BLRPM.sim, [9](#)

Delta.fun, [10](#)

graphical parameters, [11](#)

par, [11](#)

plot, [11](#)

plot.BLRPM, [11](#)

TS.acc, [12](#)

TS.stats, [13](#)