Package 'GWEX'

February 2, 2024

Type Package

Date 2024-02-02

License GPL-3

Title Multi-Site Stochastic Models for Daily Precipitation and Temperature

Version 1.1.3

Author Guillaume Evin [aut, cre]

Maintainer Guillaume Evin <guillaume.evin@inrae.fr>

Imports Rcpp (>= 1.0.11), EnvStats, MASS, mvtnorm, nleqslv, fGarch, parallel, abind, foreach, doParallel, Renext, lmomco, methods, stats

LinkingTo Rcpp, RcppArmadillo

Description Application of multi-site models for daily precipitation and temperature data.
This package is designed for an application to 105 precipitation and 26 temperature gauges located in Switzerland.
It applies fitting procedures and provides weather generators described in the following references:
- Evin, G., A.-C. Favre, and B. Hingray. (2018) <doi:10.5194/hess-22-655-2018>.
- Evin, G., A.-C. Favre, and B. Hingray. (2018) <doi:10.1007/s00704-018-2404-x>.

Depends R (>= 2.10)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

NeedsCompilation yes

Repository CRAN

Date/Publication 2024-02-02 09:00:02 UTC

R topics documented:

`agg.matrix` *agg.matrix*

Description

Simple accumulation of a matrix of precipitation

Usage

```
agg.matrix(mat, k, average = F)
```

Arguments

`mat`	matrix nDates x nStations to be aggregated
`k`	number of days for the accumulation
`average`	logical: should we average over the different periods (default=F)

Value

aggregated matrix

Author(s)

Guillaume Evin

`autocor.emp.int` *autocor.emp.int*

Description

Finds empirical autocorrelations (lag-1) between intensities corresponding to a degree of autocorrelation of an AR(1) process

Usage

```
autocor.emp.int(rho, nChainFit, Xt, parMargin, typeMargin)
```

Arguments

`rho`	autocorrelation of the AR(1) process
`nChainFit`	number of simulated variates
`Xt`	simulated occurrences, nChainFit x 2 matrix
`parMargin`	parameters of the margins 2 x 3
`typeMargin`	type of marginal distribution: 'EGPD' or 'mixExp'

Value

`scalar`	correlation between simulated intensities

Author(s)

Guillaume Evin

`cor.emp.int` *cor.emp.int*

Description

Finds observed correlations between intensities corresponding to a degree of correlation of Gaussian multivariate random numbers

Usage

```
cor.emp.int(zeta, nChainFit, Xt, parMargin, typeMargin)
```

Arguments

zeta	correlation of Gaussian multivariates
nChainFit	number of simulated variates
Xt	simulated occurrences, n x 2 matrix
parMargin	parameters of the margins 2 x 3
typeMargin	type of marginal distribution: 'EGPD' or 'mixExp'

Value

scalar	correlation between simulated intensities

Author(s)

Guillaume Evin

---

cor.emp.occ *cor.emp.occ*

Description

Finds observed correlations between occurrences corresponding to a degree of correlation of Gaussian multivariate random numbers

Usage

```
cor.emp.occ(w, Qtrans.mat, mat.comb, nLag, nChainFit, myseed = 1)
```

Arguments

w	correlation of Gaussian multivariates
Qtrans.mat	transition probabilities, 2 x ncomb matrix
mat.comb	matrix of logical: ncomb x nlag
nLag	order of the Markov chain
nChainFit	number of simulated variates
myseed	seed of random variates

Value

scalar	correlation between occurrences

Author(s)

Guillaume Evin

cor.obs.occ *cor.obs.occ*

Description

provide observed correlations between occurrences for all pairs of stations see Mhanna et al. (2012)

Usage

```
cor.obs.occ(pi00, pi0, pi1)
```

Arguments

pi00	joint probability of having dry states
pi0	probability of having a dry state
pi1	probability of having a wet state

Value

scalar	matrix of observed correlations

Author(s)

Guillaume Evin

References

Mhanna, Muamaraldin, and Willy Bauwens. "A Stochastic Space-Time Model for the Generation of Daily Rainfall in the Gaza Strip." International Journal of Climatology 32, no. 7 (June 15, 2012): 1098–1112. doi:10.1002/joc.2305.

dailyPrecipGWEX *daily observations of precipitation data*

Description

Example of daily observations of precipitation (mm) for three fictive stations, for a period of ten years.

Usage

```
data(dailyPrecipGWEX)
```

Format

matrix of Observed precipitation: 3652 days x 3 stations

Author(s)

Guillaume Evin <guillaume.evin@irstea.fr>

References

Evin, G., A.-C. Favre, and B. Hingray. 2018. “Stochastic Generation of Multi-Site Daily Precipitation Focusing on Extreme Events". Hydrol. Earth Syst. Sci. 22 (1): 655–672.

---

dailyTemperGWEX *daily observations of temperature data*

Description

Example of daily observations of temperature (mm) for three fictive stations, for a period of ten years.

Usage

```
data(dailyTemperGWEX)
```

Format

matrix of Observed temperature: 3652 days x 3 stations

Author(s)

Guillaume Evin <guillaume.evin@irstea.fr>

References

Evin G., A.C. Favre, and B. Hingray. 2018. Stochastic Generators of Multi Site Daily Temperature: Comparison of Performances in Various Applications. Theoretical and Applied Climatology.

---

disag.3D.to.1D *disag.3D.to.1D*

Description

disag.3D.to.1D

Usage

```
disag.3D.to.1D(Yobs, YObsAgg, mObsAgg, YSimAgg, mSimAgg, prob.class)
```

Arguments

Yobs	matrix of observed intensities at 24h: (nTobs*3) x nStation
YObsAgg	matrix of observed 3-day intensities: nTobs x nStation
mObsAgg	vector of season corresponding to YobsAgg
YSimAgg	matrix of simulated intensities per 3-day period: nTsim x nStation
mSimAgg	vector of season corresponding to the period simulated
prob.class	vector of probabilities indicating class of "similar" mean intensities

Value

list	Ysim matrix of disagregated daily precipitation, codeDisag matrix of disagregation codes

Author(s)

Guillaume Evin

---

dist.functions.EGPD.GI

dEGPD.GI, pEGPD.GI, qEGPD.GI, rEGPD.GI

Description

Density function, distribution function, quantile function, random generation for the unified EGPD distribution

Usage

```
dEGPD.GI(x, kappa, sig, xi)

pEGPD.GI(x, kappa, sig, xi)

qEGPD.GI(p, kappa, sig, xi)

rEGPD.GI(n, kappa, sig, xi)
```

Arguments

x	Vector of quantiles
kappa	transformation parameter greater than 0
sig	Scale parameter
xi	Shape parameter
p	Vector of probabilities
n	Number of observations

Value

dEGPD.GI gives the density function, pEGPD.GI gives the distribution function, qEGPD.GI gives the quantile function, and rEGPD.GI generates random deviates.

Author(s)

Guillaume Evin

`dry.day.frequency` *dry.day.frequency*

Description

Estimate the dry day frequency (proportion of dry days) for all stations

Usage

```
dry.day.frequency(mat.prec, th)
```

Arguments

`mat.prec`	matrix of precipitation (possibly for one month/period)
`th`	threshold above which we consider that a day is wet (e.g. 0.2 mm)

Value

`vector of numeric`
dry day frequencies

Author(s)

Guillaume Evin

`EGPD.GI.fit.PWM` *EGPD.GI.fit.PWM*

Description

Parameter estimation of the unified EGPD distribution with the PWM method. Numerical solver of the system of nonlinear equations

Usage

```
EGPD.GI.fit.PWM(x, xi = 0.05)
```

Arguments

x	vector of parameters kappa,sig
xi	shape parameter

Value

estimated parameters kappa, sig, xi

Author(s)

Guillaume Evin

`EGPD.GI.fPWM` *EGPD.GI.fPWM*

Description

Parameter estimation of the unified EGPD distribution with the PWM method. Set of equations which have to be equal to zero

Usage

```
EGPD.GI.fPWM(par, pwm, xi)
```

Arguments

par	vector of parameters kappa,sig,xi
pwm	set of probability weighted moments of order 0, 1 and 2
xi	shape parameter

Value

differences between expected and target weighted moments

Author(s)

Guillaume Evin

find.autocor *find.autocor*

Description

finds the autocorrelation leading to observed autocorrelation

Usage

```
find.autocor(autocor.emp, nChainFit, Xt, parMargin, typeMargin)
```

Arguments

autocor.emp	target correlation between intensities
nChainFit	number of simulations
Xt	simulated occurrences, nChainFit x 2 matrix
parMargin	parameters of the margins 2 x 3
typeMargin	type of marginal distribution: 'EGPD' or 'mixExp'

Value

scalar	needed correlation

Author(s)

Guillaume Evin

find.omega *find.omega*

Description

finds the correlation between normal variates leading to correlation between occurrences

Usage

```
find.omega(rho.emp, Qtrans.mat, mat.comb, nLag, nChainFit)
```

Arguments

rho.emp	target correlation between occurences
Qtrans.mat	transition probabilities, 2 x ncomb matrix
mat.comb	matrix of logical: ncomb x nlag
nLag	order of the Markov chain
nChainFit	length of the simulated chains used during the fitting

Value

scalar	needed correlation

Author(s)

Guillaume Evin

---

find.zeta *find.zeta*

Description

finds the correlation between normal variates leading to correlation between intensities

Usage

```
find.zeta(eta.emp, nChainFit, Xt, parMargin, typeMargin)
```

Arguments

eta.emp	target correlation between intensities
nChainFit	number of simulations
Xt	simulated occurrences, n x 2 matrix
parMargin	parameters of the margins 2 x 3
typeMargin	type of marginal distribution: 'EGPD' or 'mixExp'

Value

scalar	needed correlation

Author(s)

Guillaume Evin

`fit.copula.amount` *fit.copula.amount*

Description

estimate parameters which control the spatial dependence between intensities using a copula

Usage

```
fit.copula.amount(P.mat, isPeriod, th, copulaInt, M0)
```

Arguments

`P.mat`	precipitation matrix
`isPeriod`	vector of logical n x 1 indicating the days concerned by a 3-month period
`th`	threshold above which we consider that a day is wet (e.g. 0.2 mm)
`copulaInt`	type of dependence between inter-site amounts: 'Gaussian' or 'Student'
`M0`	covariance matrix of gaussianized prec. amounts for all pairs of stations

Value

`list`	list of estimates (e.g., M0, dfStudent)

Author(s)

Guillaume Evin

`fit.GWex.prec` *fit.GWex.prec*

Description

estimate all the parameters for the G-Wex model of precipitation

Usage

```
fit.GWex.prec(objGwexObs, parMargin, listOption = NULL)
```

Arguments

objGwexObs	object of class GwexObs
parMargin	if not NULL, list where each element parMargin[[iM]] corresponds to a month iM=1...12 and contains a matrix nStation x 3 of estimated parameters of the marginal distributions (EGPD or mixture of exponentials)
listOption	list with the following fields:

- **th**: threshold value in mm above which precipitation observations are considered to be non-zero (=0.2 by default)
- **nLag**: order of he Markov chain for the transitions between dry and wet states (=2 by default)
- **typeMargin**: 'EGPD' (Extended GPD) or 'mixExp' (Mixture of Exponentials). 'EGPD' by default
- **copulaInt**: 'Gaussian' or 'Student': type of dependence for amounts (='Student' by default)
- **isMAR**: logical value, do we apply a Autoregressive Multivariate Autoregressive model (order 1) =TRUE by default
- **is3Damount**: logical value, do we apply the model on 3D-amount. =FALSE by default
- **nChainFit**: integer, length of the runs used during the fitting procedure. =100000 by default
- **nCluster**: integer, number of clusters which can be used for the parallel computation

Value

a list containing the list of options listOption and the list of estimated parameters listPar. The parameters of the occurrence process are contained in parOcc and the parameters related to the precipitation amounts are contained in parInt. Each type of parameter is a list containing the estimates for each month. In parOcc, we find:

- **p01**: For each station, the probability of transition from a dry state to a wet state.
- **p11**: For each station, the probability of staying in a wet state.
- **list.pr.state**: For each station, the probabilities of transitions for a Markov chain with lag p.
- **list.mat.omega**: The spatial correlation matrix of occurrences Ω (see Evin et al., 2018).

In parInt, we have:

- **parMargin**: list of matrices nStation x nPar of parameters for the marginal distributions (one element per Class).
- **cor.int**: Matrices nStation x nStation M_0, A, Ω_Z representing the spatial and temporal correlations between all the stations (see Evin et al., 2018). For the Student copula, dfStudent indicates the ν parameter.

Author(s)

Guillaume Evin

References

Evin, G., A.-C. Favre, and B. Hingray. 2018. 'Stochastic Generation of Multi-Site Daily Precipitation Focusing on Extreme Events.' Hydrol. Earth Syst. Sci. 22 (1): 655-672. doi.org/10.5194/hess-22-655-2018.

`fit.MAR1.amount` *fit.MAR1.amount*

Description

estimate parameters which control the dependence between intensities with a MAR(1) process

Usage

```
fit.MAR1.amount(P.mat, isPeriod, th, copulaInt, M0, A)
```

Arguments

`P.mat`	precipitation matrix
`isPeriod`	vector of logical n x 1 indicating the days concerned by a 3-month period
`th`	threshold above which we consider that a day is wet (e.g. 0.2 mm)
`copulaInt`	type of dependance between inter-site amounts: 'Gaussian' or 'Student'
`M0`	covariance matrix of gaussianized prec. amounts for all pairs of stations
`A`	Matrix containing the autocorrelation (temporal) correlations

Value

list with the following items

- **M0** covariance matrix of gaussianized prec. amounts for all pairs of stations
- **A** omega correlations for all pairs of stations
- **covZ** covariance matrix of the MAR(1) process
- **sdZ** standard deviation of the diagonal elements
- **corZ** correlation matrix of the MAR(1) process
- **dfStudent** degrees of freedom for the Student copula if CopulaInt is equal to "Student"

Author(s)

Guillaume Evin

References

Matalas, N. C. 1967. "Mathematical Assessment of Synthetic Hydrology." Water Resources Research 3 (4): 937–45. https://doi.org/10.1029/WR003i004p00937.

Bárdossy, A., and G. G. S. Pegram. 2009. "Copula Based Multisite Model for Daily Precipitation Simulation." Hydrology and Earth System Sciences 13 (12): 2299–2314. https://doi.org/10.5194/hess-13-2299-2009.

`fit.margin.cdf` *fit.margin.cdf*

Description

estimate parameters which control the marginal distribution of precipitation amounts

Usage

```
fit.margin.cdf(P.mat, isPeriod, th, type = c("EGPD", "mixExp"))
```

Arguments

`P.mat`	precipitation matrix
`isPeriod`	vector of logical n x 1 indicating the days concerned by a 3-month period
`th`	threshold above which we consider that a day is wet (e.g. 0.2 mm)
`type`	distribution: 'EGPD' or 'mixExp'

Value

`matrix`	matrix of estimates p x 3

Author(s)

Guillaume Evin

`fitGwexModel` *fitGwexModel: fit a GWex model to observations.*

Description

fitGwexModel: fit a GWex model to observations.

Usage

```
fitGwexModel(objGwexObs, parMargin = NULL, listOption = NULL)
```

Arguments

objGwexObs	an object of class GwexObs
parMargin	(not required for temperature) list parMargin where and each element corresponds to a month (1...12) and contains a matrix nStation x 3 of pre-estimated parameters of the marginal distributions (EGPD or Mixture of Exponentials)
listOption	for precipitation, a list with the following fields: • **th**: threshold value in mm above which precipitation observations are considered to be non-zero (=0.2 by default) • **nLag**: order of the Markov chain for the transitions between dry and wet states (=2 by default) • **typeMargin**: 'EGPD' (Extended GPD) or 'mixExp' (Mixture of Exponentials). 'mixExp' by default • **copulaInt**: 'Gaussian' or 'Student': type of dependence for amounts (='Gaussian' by default) • **isMAR**: logical value, do we apply a Autoregressive Multivariate Autoregressive model (order 1) = FALSE by default • **is3Damount**: logical value, do we apply the model on 3D-amount. =FALSE by default • **nChainFit**: integer, length of the runs which are generated during the fitting procedure. =100000 by default • **nCluster**: integer, number of clusters which can be used for the parallel computation and for temperature, a list with the following fields: • **hasTrend**: logical value, do we fit a linear trend for the long-term change, =FALSE by default • **objGwexPrec**: object of class GwexObs containing precipitation observations. If provided, we assume that temperature must be modelled and simulated according to the precipitation states 'dry' and 'wet'. For each state, a seasonal cycle is fitted (mean and sd). • **typeMargin**: 'SGED' (default) or 'Gaussian': type of marginal distribution. • **depStation**: 'MAR1' (default) or 'Gaussian': MAR1 (Multivariate Autoregressive model order 1) for the spatial and temporal dependence or 'Gaussian' for the spatial dependence only.

Value

Return an object of class GwexFit with:

- **p**: The number of station,
- **version**: package version,
- **variable**: the type of variable,
- **fit**: a list containing the list of options listOption and the list of estimated parameters listPar.

Author(s)

Guillaume Evin

Examples

```
# Format dates corresponding to daily observations of precipitation and temperature
vecDates = seq(from=as.Date("01/01/2005",format="%d/%m/%Y"),
to=as.Date("31/12/2014",format="%d/%m/%Y"),by='day')

################################################################
#               FIT THE PRECIPITATION MODEL
################################################################

# Format observations: create a Gwex object for one station only to show a quick
# example. The syntax is similar for multi-site applications.
myObsPrec = GwexObs(variable='Prec',date=vecDates,obs=dailyPrecipGWEX[,1,drop=FALSE])

# Fit precipitation model with a threshold of 0.5 mm to distinguish wet and dry
# states (th) and keep default options otherwise, e.g. a Gaussian
# copula for the spatial dependence (copulaInt) and a mixExp distribution for
# marginal intensities ('typeMargin')
myParPrec = fitGwexModel(myObsPrec,listOption=list(th=0.5))
myParPrec # print object

################################################################
#     FIT THE TEMPERATURE MODEL, COND. TO PRECIPITATION
################################################################
# Format observations: create a G-Wex object
myObsTemp = GwexObs(variable='Temp',date=vecDates,obs=dailyTemperGWEX)

# Fit temperature model with a long-term linear trend ('hasTrend'), Gaussian margins
# ('typeMargin') and Gaussian spatial dependence ('depStation')
myParTemp = fitGwexModel(myObsTemp,listOption=list(hasTrend=TRUE,typeMargin='Gaussian',
depStation='Gaussian'))
myParTemp # print object
```

---

functions.EGPD.GI *EGPD.pGI, EGPD.dGI, EGPD.qGI*

Description

First parametric family for G(v) = v^kappa: distribution, density and quantile function

Usage

```
EGPD.pGI(v, kappa)

EGPD.dGI(v, kappa)

EGPD.qGI(p, kappa)
```

Arguments

v	probability
kappa	transformation parameter greater than 0
p	probability

Value

distribution, density and quantile of EGPD

Author(s)

Guillaume Evin

---

get.df.Student *get.df.Student*

Description

Estimates the nu parameter (degrees of freedom) of the multivariate Student distribution when the correlation matrix Sig is given

Usage

```
get.df.Student(P, Sig, max.df = 20)
```

Arguments

P	matrix of non-zero precipitation (zero precipitation are set to NA)
Sig	correlation matrix
max.df	maximum degrees of freedom tested (default=20)

Value

nu estimate

Author(s)

Guillaume Evin

References

McNeil et al. (2005) "Quantitative Risk Management"

`get.emp.cdf.matrix` *get.df.Student*

Description

get the cdf values (empirical distribution) of positive precipitation

Usage

```
get.emp.cdf.matrix(X)
```

Arguments

`X`	matrix of positive precipitation

Value

matrix with cdf values (NA if zero precipitation)

Author(s)

Guillaume Evin

`get.list.month` *get.list.month*

Description

return a vector of 3-char tags of the 12 months

Usage

```
get.list.month()
```

`get.list.season` *get.list.season*

Description

get the vector of the four seasons c('DJF','MAM','JJA','SON')

Usage

```
get.list.season()
```

Author(s)

Guillaume Evin

`get.listOption` *get.listOption*

Description

get default options and check values proposed by the user

Usage

```
get.listOption(listOption)
```

Arguments

`listOption`	list containing fields corr. to the different options. Can be NULL if no options are set

Value

`listOption`	list of options

Author(s)

Guillaume Evin

`get.M0` *get.M0*

Description

find matrix of correlations leading to estimates cor between intensities

Usage

```
get.M0(
  cor.obs,
  infer.mat.omega.out,
  nLag,
  parMargin,
  typeMargin,
  nChainFit,
  isParallel
)
```

Arguments

cor.obs	matrix p x p of observed correlations between intensities for all pairs of stations
infer.mat.omega.out	output of infer.mat.omega
nLag	order of the Markov chain
parMargin	parameters of the margins p x 3
typeMargin	type of marginal distribution: 'EGPD' or 'mixExp'
nChainFit	integer indicating the length of simulated chains
isParallel	logical: indicate computation in parallel or not (easier for debugging)

Value

list with two items

- **Xt** long simulation of the wet/dry states according to the model
- **M0** covariance matrix of gaussianized prec. amounts for all pairs of stations

Author(s)

Guillaume Evin

---

get.mat.omega *get.mat.omega*

Description

find omega correlation leading to estimates cor between occurrences

Usage

```
get.mat.omega(cor.obs, Qtrans.mat, mat.comb, nLag, nChainFit, isParallel)
```

Arguments

cor.obs	matrix p x p of observed correlations between occurrences for all pairs of stations
Qtrans.mat	transition probabilities, 2 x ncomb matrix
mat.comb	matrix of logical: ncomb x nlag
nLag	order of the Markov chain
nChainFit	length of the simulated chains used during the fitting
isParallel	logical: indicate computation in parallel or not (easier for debugging)

Value

matrix	omega correlations for all pairs of stations

Author(s)

Guillaume Evin

`get.period.fitting.month` *get.period.fitting.month*

Description

get.period.fitting.month

Usage

```
get.period.fitting.month(m.char)
```

Arguments

`m.char`	3-letter name of a month (e.g. 'JAN') return the 3 indices corresponding to the 3-month period of a month ('JAN')

`get.vec.autocor` *get.vec.autocor*

Description

find rho autocorrelation leading to empirical estimates

Usage

```
get.vec.autocor(vec.ar1.obs, Xt, parMargin, typeMargin, nChainFit, isParallel)
```

Arguments

`vec.ar1.obs`	vector of observed autocorrelations for all stations
`Xt`	simulated occurrences given model parameters of wet/dry states
`parMargin`	parameters of the margins p x 3
`typeMargin`	type of marginal distribution: 'EGPD' or 'mixExp'
`nChainFit`	integer indicating the length of the simulated chains
`isParallel`	logical: indicate computation in parallel or not (easier for debugging)

Value

`vector`	vector of rho parameters to simulate the MAR process

Author(s)

Guillaume Evin

getGwexFitPrec *getGwexFitPrec*

Description

get object GwexFit derived from the parameters replicated for each month

Usage

```
getGwexFitPrec(
  listOption = NULL,
  p,
  condProbaWDstates,
  parMargin,
  vec.ar1 = NULL,
  M0 = NULL,
  mat.omega = NULL
)
```

Arguments

listOption	list of options (see fitGwexModel)
p	number of stations
condProbaWDstates	vector of length nLag^2 of transition probabilities corresponding to the nlag possible transitions between dry/wet states expand.grid(lapply(numeric(nLag), function(x) c(F,T)))
parMargin	parameters of the margins: vector of length 3
vec.ar1	vector of observed autocorrelations for all stations
M0	**M0**: covariance matrix of gaussianized prec. amounts for all pairs of stations
mat.omega	**mat.omega**: The spatial correlation matrix of occurrences Ω

Value

Return an object of class GwexFit with:

- **p**: The number of station,
- **version**: package version,
- **variable**: the type of variable,
- **fit**: a list containing the list of options listOption and the list of estimated parameters listPar.

Examples

```
exFitGwexPrec = getGwexFitPrec(p=2,condProbaWDstates=c(0.7,0.3,0.2,0.1),
parMargin=c(0.5,0.1,0.4),vec.ar1=rep(0.7,2),M0=rbind(c(1,0.6),c(0.6,1)),
mat.omega=rbind(c(1,0.8),c(0.8,1)))
```

Gwex-class *Class Gwex*

Description

Defines a generic Gwex object. GWex objects contain two slots: - the version ('vX.X.X') - the type of variable ('Prec' or 'Temp')

Author(s)

Guillaume Evin

GwexFit-class *Class GwexFit*

Description

Defines a GwexFit object which is a Gwex object containing 'fit', a list containing the fitted parameters, and 'p', the number of stations. See fitGwexModel for some examples.

Author(s)

Guillaume Evin

GwexObs *Constructor*

Description

Constructor of class [GwexObs]

Usage

```
GwexObs(variable, date, obs)
```

Arguments

variable	'Prec' or 'Temp'
date	vector of class 'Date'
obs	matrix nTime x nStations of observations

Value

An object of class [GwexObs]

Examples

```
# Format dates corresponding to daily observations of precipitation and temperature
vecDates = seq(from=as.Date("01/01/2005",format="%d/%m/%Y"),
to=as.Date("31/12/2014",format="%d/%m/%Y"),by='day')

# build GwexObs object with precipitation data
myObsPrec = GwexObs(variable='Prec',date=vecDates,obs=dailyPrecipGWEX)

# print GwexObs object
myObsPrec

# build GwexObs object with temperature data
myObsTemp = GwexObs(variable='Temp',date=vecDates,obs=dailyTemperGWEX)

# print GwexObs object
myObsTemp
```

GwexObs-class *Class* GwexObs

Description

Defines a GwexObs object which is a Gwex object containing dates and a matrix of observations.

Author(s)

Guillaume Evin

Examples

```
# Format dates corresponding to daily observations of precipitation and temperature
vecDates = seq(from=as.Date("01/01/2005",format="%d/%m/%Y"),
to=as.Date("31/12/2014",format="%d/%m/%Y"),by='day')

# build GwexObs object with precipitation data
myObsPrec = GwexObs(variable='Prec',date=vecDates,obs=dailyPrecipGWEX)

# print GwexObs object
myObsPrec

# build GwexObs object with temperature data
myObsTemp = GwexObs(variable='Temp',date=vecDates,obs=dailyTemperGWEX)

# print GwexObs object
myObsTemp
```

`GwexSim-class` *Defines a* `GwexSim` *object which is a* `Gwex` *object containing 'sim', an array containing the simulations, and 'dates', a vector of dates. See simGwexModel for some examples.*

Description

Defines a `GwexSim` object which is a `Gwex` object containing 'sim', an array containing the simulations, and 'dates', a vector of dates. See simGwexModel for some examples.

Author(s)

Guillaume Evin

`infer.autocor.amount` *infer.autocor.amount*

Description

special case of `infer.dep.amount` where there is only one station

Usage

```
infer.autocor.amount(
  P.mat,
  pr.state,
  isPeriod,
  nLag,
  th,
  parMargin,
  typeMargin,
  nChainFit,
  isMAR,
  isParallel
)
```

Arguments

`P.mat`	precipitation matrix
`pr.state`	probabilities of transitions for a Markov chain with lag p.
`isPeriod`	vector of logical n x 1 indicating the days concerned by a 3-month period
`nLag`	order of he Markov chain for the transitions between dry and wet states (=2 by default)
`th`	threshold above which we consider that a day is wet (e.g. 0.2 mm)

parMargin	parameters of the margins 2 x 3
typeMargin	'EGPD' (Extended GPD) or 'mixExp' (Mixture of Exponentials). 'EGPD' by default
nChainFit	integer, length of the runs used during the fitting procedure. =100000 by default
isMAR	logical value, do we apply a Autoregressive Multivariate Autoregressive model (order 1) =TRUE by default
isParallel	logical: indicate computation in parallel or not (easier for debugging)

Value

list	list of estimates (e.g., M0, dfStudent)

Author(s)

Guillaume Evin

`infer.dep.amount` *infer.dep.amount*

Description

estimate parameters which control the spatial dependence between intensities using a copula

Usage

```
infer.dep.amount(
  P.mat,
  isPeriod,
  infer.mat.omega.out,
  nLag,
  th,
  parMargin,
  typeMargin,
  nChainFit,
  isMAR,
  copulaInt,
  isParallel
)
```

Arguments

P.mat	precipitation matrix
isPeriod	vector of logical n x 1 indicating the days concerned by a 3-month period
infer.mat.omega.out	output of `infer.mat.omega`

nLag	order of he Markov chain for the transitions between dry and wet states (=2 by default)
th	threshold above which we consider that a day is wet (e.g. 0.2 mm)
parMargin	parameters of the margins 2 x 3
typeMargin	'EGPD' (Extended GPD) or 'mixExp' (Mixture of Exponentials). 'EGPD' by default
nChainFit	integer, length of the runs used during the fitting procedure. =100000 by default
isMAR	logical value, do we apply a Autoregressive Multivariate Autoregressive model (order 1) =TRUE by default
copulaInt	'Gaussian' or 'Student': type of dependence for amounts (='Student' by default)
isParallel	logical: indicate computation in parallel or not (easier for debugging)

Value

list	list of estimates (e.g., M0, dfStudent)

Author(s)

Guillaume Evin

`infer.mat.omega` *infer.mat.omega*

Description

find omega correlation leading to estimates cor between occurrences

Usage

```
infer.mat.omega(P.mat, isPeriod, th, nLag, pr.state, nChainFit, isParallel)
```

Arguments

P.mat	matrix of precipitation n x p
isPeriod	vector of logical n x 1 indicating the days concerned by a 3-month period
th	threshold above which we consider that a day is wet (e.g. 0.2 mm)
nLag	order of the Markov chain
pr.state	output of function `lagTransProbaMatrix`
nChainFit	length of the simulated chains used during the fitting
isParallel	logical: indicate computation in parallel or not (easier for debugging)

Value

A list with different objects

- **Qtrans.mat**: matrix nStation x n.comb of transition probabilites
- **mat.comb**: matrix of possible combination n.comb x nLag
- **mat.omega**: The spatial correlation matrix of occurrences Ω (see Evin et al., 2018).

Author(s)

Guillaume Evin

`joint.proba.occ` *joint.proba.occ*

Description

joint probabilities of occurrences for all pairs of stations

Usage

```
joint.proba.occ(P, th)
```

Arguments

`P`	matrix of precipitation
`th`	threshold above which we consider that a day is wet (e.g. 0.2 mm)

Value

`list`	list of joint probabilities

Author(s)

Guillaume Evin

`lagTransProbaMatrix` *lagTransProbaMatrix*

Description

Estimate the transition probabilities between wet and dry states, for nlag previous days, for all stations

Usage

```
lagTransProbaMatrix(mat.prec, isPeriod, th, nlag)
```

Arguments

`mat.prec`	matrix of precipitation
`isPeriod`	vector of logical n x 1 indicating the days concerned by a 3-month period
`th`	threshold above which we consider that a day is wet (e.g. 0.2 mm)
`nlag`	number of lag days

Value

`list`	list with one item per station, where each item is a matrix nLag^2 x (nLag+1) of transition probability between dry/wet state. The first nLag columns indicate the wet/dry states for the previous nLag days

Author(s)

Guillaume Evin

`lagTransProbaVector` *lagTransProbaVector*

Description

Estimate the transition probabilities between wet and dry states, for nlag previous days, for one station

Usage

```
lagTransProbaVector(vec.prec, isPeriod, th, nlag)
```

Arguments

vec.prec	vector nx1 of precipitation for one station
isPeriod	vector of logical n x 1 indicating the days concerned by a 3-month period
th	threshold above which we consider that a day is wet (e.g. 0.2 mm)
nlag	number of lag days

Value

matrix	matrix nLag^2 x (nLag+1) of transition probability between dry/wet state. The first nLag columns indicate the wet/dry states for the previous nLag days

Author(s)

Guillaume Evin

---

mask.GWex.Yt *mask.GWex.Yt*

Description

Mask intensities where there is no occurrence

Usage

```
mask.GWex.Yt(Xt, Yt)
```

Arguments

Xt	simulated occurrences
Yt	simulated intensities

Value

matrix	matrix n x p of simulated precipitations

Author(s)

Guillaume Evin

`modify.cor.matrix` *modify.cor.matrix*

Description

Modify a non-positive definite correlation matrix in order to have a positive definite matrix

Usage

```
modify.cor.matrix(cor.matrix)
```

Arguments

`cor.matrix`	possibly non-positive definite correlation matrix

Value

positive definite correlation matrix

Author(s)

Guillaume Evin

References

Rousseeuw, P. J. and G. Molenberghs. 1993. Transformation of non positive semidefinite correlation matrices. Communications in Statistics: Theory and Methods 22(4):965-984.

Rebonato, R., & Jackel, P. (2000). The most general methodology to create a valid correlation matrix for risk management and option pricing purposes. J. Risk, 2(2), 17-26.

`month2season` *month2season*

Description

transform vector of months to seasons

Usage

```
month2season(vecMonth)
```

Arguments

`vecMonth`	a vector of months given as integers 1:12

Author(s)

Guillaume Evin

print,Gwex-method *print-methods: Create a method to print Gwex objects.*

Description

print-methods: Create a method to print Gwex objects.

Usage

```
## S4 method for signature 'Gwex'
print(x)

## S4 method for signature 'GwexObs'
print(x)

## S4 method for signature 'GwexFit'
print(x)

## S4 method for signature 'GwexSim'
print(x)
```

Arguments

x	Gwex object

Examples

```
# Format dates corresponding to daily observations of precipitation and temperature
vecDates = seq(from=as.Date("01/01/2005",format="%d/%m/%Y"),
to=as.Date("31/12/2014",format="%d/%m/%Y"),by='day')

# build GwexObs object with temperature data
myObsTemp = GwexObs(variable='Temp',date=vecDates,obs=dailyTemperGWEX)

# print GwexObs object
myObsTemp
```

PWM.EGPD.GI *EGPD.GI.mu0, EGPD.GI.mu1, EGPD.GI.mu2*

Description

Probability Weighted Moments of order 0, 1 and 2 of the unified EGPD distribution

Usage

```
EGPD.GI.mu0(kappa, sig, xi)

EGPD.GI.mu1(kappa, sig, xi)

EGPD.GI.mu2(kappa, sig, xi)
```

Arguments

kappa	transformation parameter greater than 0
sig	Scale parameter
xi	Shape parameter

Value

Probability Weighted Moments

Author(s)

Guillaume Evin

QtransMat2Array *QtransMat2Array*

Description

reshape Qtrans.mat to an array

Usage

```
QtransMat2Array(n, p, Qtrans.mat)
```

Arguments

n	matrix of precipitation
p	number of stations
Qtrans.mat	transition probabilities, 2 x ncomb matrix

Value

array	array of transition probabilities with dimension n x p x n.comb

Author(s)

Guillaume Evin

show,Gwex-method *show-methods: Create a method to show Gwex objects.*

Description

show-methods: Create a method to show Gwex objects.

Usage

```
## S4 method for signature 'Gwex'
show(object)

## S4 method for signature 'GwexObs'
show(object)

## S4 method for signature 'GwexFit'
show(object)

## S4 method for signature 'GwexSim'
show(object)
```

Arguments

object	Gwex object

Examples

```
# Format dates corresponding to daily observations of precipitation and temperature
vecDates = seq(from=as.Date("01/01/2005",format="%d/%m/%Y"),
to=as.Date("31/12/2014",format="%d/%m/%Y"),by='day')

# build GwexObs object with temperature data
myObsTemp = GwexObs(variable='Temp',date=vecDates,obs=dailyTemperGWEX)

# show GwexObs object
myObsTemp
```

sim.GWex.occ *sim.GWex.occ*

Description

generate boolean variates which describe the dependence between intersite occurrence correlations and wet/dry persistence

Usage

```
sim.GWex.occ(objGwexFit, vecMonth)
```

Arguments

objGwexFit	object of class GwexFit
vecMonth	vector n x 1 of integers indicating the months

Value

matrix of logical
: occurrences simulated

Author(s)

Guillaume Evin

---

sim.GWex.prec.1it *sim.GWex.prec.1it*

Description

Simulate one scenario of precipitation from the GWex model

Usage

```
sim.GWex.prec.1it(objGwexFit, vecDates, myseed, objGwexObs, prob.class)
```

Arguments

objGwexFit	object of class GwexFit
vecDates	vector of continuous dates
myseed	seed of the random generation, to be fixed if the results need to be replicated
objGwexObs	optional: necessary if we need observations to simulate (e.g. disaggregation of 3-day periods)
prob.class	vector of probabilities indicating class of "similar" mean intensities

Value

matrix	Precipitation simulated for the dates contained in vec.Dates at the different stations

Author(s)

Guillaume Evin

sim.GWex.Yt *sim.GWex.Yt*

Description

Inverse PIT: from the probability space to the precipitation space

Usage

```
sim.GWex.Yt(objGwexFit, vecMonth, Yt.Pr)
```

Arguments

objGwexFit	object of class GwexFit
vecMonth	vector of integer indicating the months
Yt.Pr	uniform variates describing dependence between inter-site amounts

Value

matrix	matrix n x p of simulated non-zero precipitation intensities

Author(s)

Guillaume Evin

sim.GWex.Yt.Pr *sim.GWex.Yt.Pr*

Description

generate uniform variates which describe the dependence between intersite amount correlations

Usage

```
sim.GWex.Yt.Pr(objGwexFit, vecMonth)
```

Arguments

objGwexFit	object of class GwexFit
vecMonth	vector n x 1 of integer indicating the months

Value

matrix	matrix n x p of uniform dependent variates

Author(s)

Guillaume Evin

sim.GWex.Yt.Pr.get.param *sim.GWex.Yt.Pr.get.param*

Description

get relevant parameters

Usage

```
sim.GWex.Yt.Pr.get.param(objGwexFit, iM)
```

Arguments

objGwexFit	object of class GwexFit
iM	integer indicating the month

Value

list	list of parameters

Author(s)

Guillaume Evin

sim.Zt.MAR *sim.Zt.MAR*

Description

generate gaussian variates which describe the spatial and temporal dependence between the sites (MAR(1) process)

Usage

```
sim.Zt.MAR(PAR, copulaInt, Zprev, p)
```

Arguments

PAR	parameters for this class
copulaInt	'Gaussian' or 'Student'
Zprev	previous Gaussian variate
p	number of stations

Value

matrix	matrix n x p of uniform dependent variates

Author(s)

Guillaume Evin

---

sim.Zt.Spatial *sim.Zt.Spatial*

Description

generate gaussian variates which describe the spatial dependence between the sites

Usage

```
sim.Zt.Spatial(PAR, copulaInt, p)
```

Arguments

PAR	parameters for a class
copulaInt	'Gaussian' or 'Student'
p	number of stations

Value

matrix	matrix n x p of uniform dependent variates

Author(s)

Guillaume Evin

---

simGwexModel *simGwexModel*

Description

Simulate from a GWex model

Usage

```
simGwexModel(
  objGwexFit,
  nb.rep = 10,
  d.start = as.Date("01011900", "%d%m%Y"),
  d.end = as.Date("31121999", "%d%m%Y"),
  objGwexObs = NULL,
  prob.class = c(0.5, 0.75, 0.9, 0.99),
  objGwexSim = NULL,
  nCluster = 1
)
```

Arguments

objGwexFit	an object of class GwexFit
nb.rep	number of repetitions of scenarios
d.start	a starting date for the simulation
d.end	an ending date for the simulation
objGwexObs	optional: an object of class GwexObs if we need the observations to simulate (disaggregation prec 3D -> 1D)
prob.class	vector of probabilities indicating class of "similar" mean intensities
objGwexSim	optional: an object of class GwexSim if we need simulations to simulate (temp conditional to prec)
nCluster	optional, number of clusters which can be used for the parallel computation

Value

GwexSim	an object of class GwexSim. Contains sim (3D-array with the simulations) and a vector of dates

Author(s)

Guillaume Evin

Examples

```
# vector of dates
vecDates = seq(from=as.Date("01/01/2005",format="%d/%m/%Y"),
to=as.Date("31/12/2014",format="%d/%m/%Y"),by='day')

############################################################
#               FIT AND SIMULATE FROM THE PRECIPITATION MODEL
############################################################
# Format observations: create a G-Wex object
myObsPrec = GwexObs(variable='Prec',date=vecDates,obs=dailyPrecipGWEX[,1,drop=FALSE])

# Fit GWEX precipitation model, default options except for the threshold th
myParPrec = fitGwexModel(myObsPrec,listOption=list(th=0.5)) # fit model
```

```
# Generate 2 scenarios for one year, using the 'GwexFit' object
mySimPrec = simGwexModel(objGwexFit=myParPrec, nb.rep=2, d.start=vecDates[1],
d.end=vecDates[10])
mySimPrec # print object

############################################################
#     FIT AND SIMULATE FROM THE TEMPERATURE MODEL
############################################################
# Format observations: create a G-Wex object
myObsTemp = GwexObs(variable='Temp',date=vecDates,obs=dailyTemperGWEX)

# Fit GWEX temperature model
myParTemp = fitGwexModel(myObsTemp,listOption=list(hasTrend=TRUE,typeMargin='Gaussian',
depStation='Gaussian'))

# Generate 2 scenarios for one year, using an existing 'GwexFit' object
mySimTemp = simGwexModel(objGwexFit=myParTemp, nb.rep=2, d.start=vecDates[1],
                         d.end=vecDates[365],objGwexObs=myObsPrec)
mySimTemp # print object
```

`simPrecipOcc` *simPrecipOcc*

Description

find matrix of correlations leading to estimates cor between intensities

Usage

```
simPrecipOcc(nLag, n, pr)
```

Arguments

`nLag`	order of the Markov chain
`n`	integer indicating the length of simulated chains
`pr`	vector of probabilies corr. to the conditional transition probabilities

Value

a vector Xt of length n with values 0/1 corr. to dry/wet states

Author(s)

Guillaume Evin

`unif.to.prec` *unif.to.prec*

Description

from uniform variates to precipitation variates

Usage

```
unif.to.prec(pI, typeMargin, U)
```

Arguments

`pI`	vector of three parameters of the marginal distributions
`typeMargin`	type of marginal distribution: 'EGPD' or 'mixExp'
`U`	vector of uniform variates

Value

`matrix`	matrix of estimates p x 3

Author(s)

Guillaume Evin

`wet.day.frequency` *wet.day.frequency*

Description

Estimate the wet day frequency (proportion of wet days) for all stations

Usage

```
wet.day.frequency(mat.prec, th)
```

Arguments

`mat.prec`	matrix of precipitation (possibly for one month/period)
`th`	threshold above which we consider that a day is wet (e.g. 0.2 mm)

Value

`vector of numeric`
wet day frequencies

Author(s)

Guillaume Evin

Index