Package ‘anySim’

May 13, 2019

Type Package

Title Stochastic Simulation of Processes with any Marginal Distribution and Correlation Structure

Version 0.1.0

Description anySim is an R package for the stochastic simulation of processes with any marginal distribution and dependence structure. Currently, the package provides models for the simulation of univariate stationary and cyclostationary processes, exhibiting continuous, discrete and mixed-type marginal distributions as well as any valid (i.e., positive definite) short-range or long-range autocorrelation structure. Furthermore, it implements a multivariate stationary stochastic model with similar capabilities, preserving also the lag-0 crosscorrelation coefficients among the processes. The package can be used for the generation of synthetic time series (e.g., rainfall, runoff, temperature, wind speed etc.) with the desired marginal and stochastic properties.

Author Ioannis Tsoukalas [aut, cre, ctb],
Panagiotis Kossieris [aut, ctb]

Maintainer Ioannis Tsoukalas <itsoukal@mail.ntua.gr>

Imports pracma, moments, DEoptim, cubature, Matrix, psych, matrixcalc, dplyr, ExtDist, VGAM, Rsolnp, parallel, actuar

Depends R (>= 3.5.0), pracma, moments, DEoptim, cubature, Matrix, psych, matrixcalc, dplyr, ExtDist, VGAM, Rsolnp, parallel, actuar

Suggests knitr, rmarkdown

License GPL-3

URL www.itia.ntua.gr/en/softinfo/33/

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

`anySim-package` *anySim: Stochastic Simulation of Processes with any Marginal Distribution and Correlation Structure*

Description

anySim is an R package for the stochastic simulation of processes with any marginal distribution and dependence structure. Currently, the package provides models for the simulation of univariate stationary and cyclostationary processes, exhibiting continuous, discrete and mixed-type marginal distributions as well as any valid (i.e., positive definite) short-range or long-range autocorrelation structure. Furthermore, it implements a multivariate stationary stochastic model with similar capabilities, preserving also the lag-0 cross-correlation coefficients among the processes. The package can be used for the generation of synthetic time series (e.g., rainfall, runoff, temperature, wind speed etc.) with the desired marginal and stochastic properties.

Details

The methodology is based on the concept of Nataf's joint distribution model (Nataf, 1962; Liu and Der Kiureghian 1986) according to which the joint distribution of random variables with any target arbitrary marginal distributions can be obtained by mapping an auxiliary multivariate standard Gaussian distribution via the inverse cumulative distribution functions (ICDFs). It exploits the link that exists between correlation coefficients in the Gaussian and the target domain, reproducing also the target correlations. Moving to stochastic process simulation, anySim employs a similar concept (for more details see, Kossieris et al., 2019; Tsoukalas et al., 2017, 2018a, 2018b, 2018c, 2019; Tsoukalas 2019) that is based on the mapping (through the ICDF) of an auxiliary Gaussian process (Gp) through the ICDF in order to establish processes with the target marginal distribution and correlation structure. The package comprises the following stochastic simulation models:

- Autoregressive To Anything model of order p - ARTA(p): This model uses an appropriately parameterised univariate AR(p) to simulate an auxiliary Gp to establish the target correlation structure. It is noted that a similar low-order (p=1) model has been proposed by Cario and Nelson (1996).

- Sum of n Autoregressive To Anything models of order 1 - nARTA(1): This model uses the sum of n, appropriately parameterised, univariate AR(1) models to simulate an auxiliary Gp to establish the target correlation structure (see, Papalexiou, 2018).
- Symmetric Moving Average To Anything - SMARTA(q): This model uses an appropriately parameterised SMA(q) model to simulate an auxiliary Gp to establish the target correlation structure. In the final step, the Gp realisation is mapped to the actual domain through the ICDF of the target distribution (Tsoukalas et al., 2018b, 2019; Tsoukalas, 2019).
- Stochastic Periodic Autoregressive To Anything model of order 1 - SPARTA: This model uses an appropriately parameterised univariate PAR(1) model to simulate a cyclostationary auxiliary Gp to establish the target season-to-season correlation structure (Tsoukalas et al., 2017, 2018a, 2019; Tsoukalas, 2019).

Author(s)

Developed by: Ioannis Tsoukalas <itsoukal@mail.ntua.gr>
Panagiotis Kossieris <pkossier@cental.ntua.gr>

References

- Cario, M., and Nelson, B., 1996. Autoregressive to anything: Time-series input processes for simulation. Operations Research Letters, 19(2), 51–58. (link)
- Kossieris, P., Tsoukalas, I., Makropoulos, C., and Savic D., 2019. Simulating marginal and dependence behaviour of water demand processes at any fine time scale, Water, 11(5), 885.(link)
- Liu, P., and Der Kiureghian, A., 1986. Multivariate distribution models with prescribed marginals and covariances. Probabilistic Engineering Mechanics, 1(2), 105–112. (link)
- Nataf, A., 1962. Statistique mathematique-determination des distributions de probabilites dont les marges sont donnees. Comptes Rendus de l'Academie Des Sciences, 255(1), 42–43.
- Papalexiou, S., 2018. Unified theory for stochastic modelling of hydroclimatic processes: Preserving marginal distributions, correlation structures, and intermittency. Advances in Water Resources. (link)
- Tsoukalas, I., Efstratiadis, A., and Makropoulos, C., 2017. Stochastic simulation of periodic processes with arbitrary marginal distributions. 15th International Conference on Environmental Science and Technology (CEST2017), Rhodes, Greece. (link)
- Tsoukalas, I., Efstratiadis, A., and Makropoulos, C., 2018a. Stochastic periodic autoregressive to anything (SPARTA): Modeling and simulation of cyclostationary processes with arbitrary marginal distributions. Water Resources Research 54 (1), 161-185. (link)
- Tsoukalas, I., Makropoulos, C., and Koutsoyiannis, D., 2018b. Simulation of Stochastic Processes Exhibiting Any-Range Dependence and Arbitrary Marginal Distributions. Water Resources Research 54(11), 9484-9513. (link)
- Tsoukalas, I., Papalexiou, S.M, Efstratiadis, A., and Makropoulos, C., 2018c. A cautionary note on the reproduction of dependencies through linear stochastic models with non-Gaussian white noise. Water 10 (6), 771. (link)
- Tsoukalas, I., Efstratiadis, A., and Makropoulos, C., 2019. Building a puzzle to solve a riddle: A multi-scale disaggregation approach for multivariate stochastic processes with any marginal distribution and correlation structure. Journal of hydrology. (link).
- Tsoukalas, I., 2019. Modelling and simulation of non-Gaussian stochastic processes for optimization of water-systems under uncertainty, PhD thesis, 339 pages, National Technical University of Athens, December 2018. (link)

acsCAS	*The Cauchy-type autocorrelation structure (CAS)*

Description

The Cauchy-type autocorrelation structure (CAS)

Usage

```
acsCAS(param, lag, var = 1)
```

Arguments

`param`	A two-dimenstional vector, containing the parameters of CAS. First position is for b, the second for k.
`lag`	A scalar indicating the maximum lag, up to which CAS is estimated.
`var`	A scalar indicating the variance of the process. If var!=1 then,the autocovariance structure is returned, instead of the autocorrelation structure.

Value

A vector of length (lag+1) with the values of CAS autocorrelation structure.

Examples

```
##  Estimate and plot a CAS structure, with b=1 and k=0.6, up to lag 500.
ACF=acsCAS(param=c(1, 0.6), lag=500, var=1)
plot(0:(length(ACF)-1), ACF)
```

acsHurst	*The Hurst or fGn autocorrelation structure*

Description

The Hurst or fGn autocorrelation structure

Usage

```
acsHurst(H, lag, var = 1)
```

Arguments

`H`	A scalar indicating the Hurst coefficient.
`lag`	A scalar indicating the maximum lag, up to which the fGn is estimated.
`var`	A scalar indicating the variance of the process. If var!=1 then, the autocovariance structure is returned, instead of the autocorrelation structure.

Value

A vector of length (lag+1) with the values of fGn autocorrelation structure.

Examples

```
## Estimate and plot an fGn (i.e., Hurst) structure with H=0.7, up to lag 500.
ACF=acsHurst(H=0.7, lag=500, var=1)
plot(0:(length(ACF)-1), ACF)
```

DistrStats — *Estimation of the theoretical Mean, Variance, Skewness and Kurtosis coefficients of a distribution function (from the distribution function).*

Description

Estimate the theoretical Mean, Variance, Skewness and Kurtosis coefficients of a distribution function, based on the distribution function.

Usage

```
DistrStats(dDistr, lb = -Inf, ub = Inf, subdiv = 90000,
  rel.tol = 10^-8, abs.tol = 10^-8, ...)
```

Arguments

`dDistr`	The density function (as a function) of the distribution.
`lb`	A scalar indicating the lower bound of distribution.
`ub`	A scalar indicating the upper bound of distribution.
`subdiv`	A scalar indicating the maximum number of subintervals.
`rel.tol`	A scalar indicating the relative accuracy requested.
`abs.tol`	A scalar indicating the absolute accuracy requested.
`...`	Additional named arguments containing the distribution parameters.

Value

A 4-dimensional vector containing the theoretical Mean, Variance, Skewness and Kurtosis coefficients of the distribution.

Examples

```
## Gamma distribution with shape=0.5 and scale=2
DistrStats(dDistr = dgamma, shape=0.5,scale=2,lb = 0)

## Zero-inflated (i.e., mixed) distribution with p0=0.7 and
## continuous part given by Gamma distribution with shape=0.5 and scale=2

DistrStats(dDistr = function(x) dmixed(x,Distr = dgamma, p0=0.7, shape=0.5, scale=2),lb = 0)
```

DistrStats2 *Estimation of the theoretical Mean, Variance, Skewness and Kurtosis coefficients of a distribution function (from the quantile function).*

Description

Estimate the theoretical Mean, Variance, Skewness and Kurtosis coefficients of a distribution function, based on the quantile function.

Usage

```
DistrStats2(qDistr, lb = 0, ub = 1, subdiv = 90000,
  rel.tol = 10^-8, abs.tol = 10^-8, ...)
```

Arguments

qDistr	The quantile function (as a function) of the distribution.
lb	A scalar indicating the lower bound of distribution.
ub	A scalar indicating the upper bound of distribution.
subdiv	A scalar indicating the maximum number of subintervals.
rel.tol	A scalar indicating the relative accuracy requested.
abs.tol	A scalar indicating the absolute accuracy requested.
...	Additional named arguments containing the distribution parameters.

Value

A 4-dimensional vector containing the theoretical Mean, Variance, Skewness and Kurtosis coefficients of the distribution.

Examples

```
## Gamma distribution with shape=0.5 and scale=2
DistrStats2(qDistr = qgamma, shape=1,scale=1, lb = 0,ub = 1)

## Zero-inflated (i.e., mixed) distribution with p0=0.7 and
## continuous part given by Gamma distribution with shape=0.5 and scale=2

DistrStats2(qDistr = function(x) qmixed(x,Distr = qgamma, p0=0.7, shape=1, scale=1),lb = 0,ub=1)
```

EstARTAp *Estimation of the auxiliary AR(p) model parameters*

Description

Estimation of parameters of the auxiliary AR(p) model to simulate the auxiliary Gaussian process.

Usage

```
EstARTAp(ACF, maxlag = 0, dist, params, NatafIntMethod = "GH",
  NoEval = 9, polydeg = 8, ...)
```

Arguments

ACF	A vector with the target autocorrelation structure (including lag-0, i.e., 1).
maxlag	A scalar incating the order of the AR(p) model. If maxlag=0, then the order of the model is p=(length(ACF)-1)
dist	A string indicating the quantile function of the target marginal distribution (i.e., the ICDF).
params	A named list with the parameters of the target distribution.
NatafIntMethod	A string ("GH", "Int", or "MC"), indicating the intergation method, to resolve the Nataf integral.
NoEval	A scalar indicating (default: 9) the number of evaluation points for the integration methods.
polydeg	A scalar indicating the order of the fitted polynomial. If polydeg=0, then another curve is fitted.
...	Additional named arguments for the selected "NatafIntMethod" method.

Value

A list with the parameters of the auxiliary Gaussian AR(p) model.

Note

Avoid the use of the "GH" method (i.e., NatafIntMethod='GH'), when the marginal(s) are discrete.

Examples

```
## Parameter estimation for a process with zero-inflated (i.e., mixed) marginal distribution,
## with p0=0.9, and a Gamma distribution
## for the continuous part with shape=0.1 and scale=1.
## In this case, the Autocorrelation strucure is a simple AR(1) with rho=0.8.
## Not run:
ACF=as.vector(ARMAacf(ar = 0.8, lag.max = 100))
fx='qmixed'
pfx=list(Distr=qgamma, p0=0.9, shape=0.1, scale=1)

ARTApar=EstARTAp(ACF=ACF, maxlag=0, dist=fx, params=pfx,
NatafIntMethod ='GH', NoEval=9, polydeg=0)

## End(Not run)
```

EstnAR1 *Estimation of the auxiliary n-AR(1) model parameters*

Description

Estimation of parameters of the the sum of n univariate AR(1) models to simulate the auxiliary Gaussian process.

Usage

```
EstnAR1(ACF, Ar1Num, dist = "qgamma", params, NatafIntMethod = "GH",
  NoEval = 9, polydeg = 8, ...)
```

Arguments

ACF	A vector with the target autocorrelation structure (including lag-0, i.e., 1).
Ar1Num	A scalar (>=2) indicating the number (n) of AR(1) models.
dist	A string indicating the quantile function of the target marginal distribution (i.e., the ICDF).
params	A named list with the parameters of the target distribution.
NatafIntMethod	A string ("GH", "Int", or "MC"), indicating the intergation method, to resolve the Nataf integral.
NoEval	A scalar indicating (default: 9) the number of evaluation points for the integration methods.
polydeg	A scalar indicating the order of the fitted polynomial. If polydeg=0, then another curve is fitted.
...	Additional named arguments for the selected "NatafIntMethod" method.

Value

A list with the parameters of the auxiliary Gaussian n-AR(1) model.

Note

Avoid the use of the "GH" method (i.e., NatafIntMethod='GH'), when the marginal(s) are discrete.

Examples

```
## Parameter estimation for a process with zero-inflated (i.e., mixed) marginal distribution,
## with p0=0.8, and a Gamma distribution
## for the continuous part with shape=0.5 and scale=1.
## In this case, the Autocorrelation strucure is a CAS ACS with b=2 and k=0.5.
## Not run:
ACF=acsCAS(param = c(2, 0.5), lag = 500, var = 1)
dist='qmixed'
params=list(Distr=qgamma, p0=0.8, shape=0.5, scale=1)
nAR1param=EstnAR1(ACF = ACF, Ar1Num = 3, dist = dist, params = params,
                  NatafIntMethod = 'GH', NoEval = 9, polydeg = 8)

## End(Not run)
```

EstSMARTA — *Estimation of the auxiliary SMA model parameters*

Description

Estimation of parameters of the auxiliary SMA model to simulate the auxiliary Gaussian process.

Usage

```
EstSMARTA(dist, params, ACFs, Cmat, DecoMethod = "cor.smooth",
  FFTLag = 512, NatafIntMethod = "GH", NoEval = 9, polydeg = 8,
  ...)
```

Arguments

`dist`	A k-dimensional string vector indicating the quantile function of the target marginal distribution (i.e., the ICDF).
`params`	A k-dimensional named list with the parameters of the target distributions.
`ACFs`	A k-dimensional list with the target autocorrelation structure (including lag-0, i.e., 1).
`Cmat`	A matrix (k x k) containing the lag-0 cross-correlation coefficients of the processes.
`DecoMethod`	A string indicating the decomposition method, in case of a non-positive definite matrix (options: 'cor.smooth' and 'nearPD')
`FFTLag`	A scalar indicating the length of the Fast Fourrier Transform (required to estimate the internal parameters of SMA model). Default value=512.
`NatafIntMethod`	A string ("GH", "Int", or "MC"), indicating the intergation method, to resolve the Nataf integral.
`NoEval`	A scalar (power of 2) indicating (default: 9) the number of evaluation points for the integration methods.
`polydeg`	A scalar indicating the order of the fitted polynomial. If polydeg=0, then another curve is fitted.
`...`	Additional named arguments for the selected "NatafIntMethod" method.

Value

A list with the parameters of the auxiliary Gaussian SMA model.

Note

Avoid the use of the "GH" method (i.e., NatafIntMethod='GH'), when the marginal(s) are discrete.

Examples

```
## Simulation example of a bivariate process with
## zero-inflated marginal distributions.
# Define the simulation parameters ----------------------------------------
## Not run:
LAG=2^6
```

```
FFTLag=2^7
SMALAG=2^6
steps=2^14

PFXs=list()
FXs=c('qmixed','qmixed')
# Gamma distribution: Gamma(shape=2, rate=1)
PFXs[[1]]=list(Distr=qgamma, p0=0.9, shape=1, scale=1)
# Weibull distribution: Weibull(shape=1, scale=2)
PFXs[[2]]=list(Distr=qweibull, p0=0.85, shape=1, scale=2)

ACFs=list()
ACFs[[1]]=acsCAS(param = c(0.1, 0.6), lag = LAG)
ACFs[[2]]=acsCAS(param = c(0.2, 0.3), lag = LAG)

Cmat=matrix(c(1,0.6,0.6,1), ncol=2, nrow=2)

# Calculate SMARTAâ€™s parameters -------------------------------------------
SMAparam=EstSMARTA(dist = FXs, params = PFXs, ACFs = ACFs,
Cmat = Cmat, DecoMethod = 'cor.smooth',
FFTLag = FFTLag, NatafIntMethod = 'GH', NoEval = 9, polydeg = 8)

## End(Not run)
```

EstSPARTA *Estimation of the auxiliary PAR(1) model parameters*

Description

Estimation of parameters of the auxiliary PAR(1) model to simulate the auxiliary cyclostationary Gaussian process to establish the target season-to-season correlation structure.

Usage

```
EstSPARTA(s2srtarget, dist, params, NatafIntMethod = "GH", NoEval = 9,
  polydeg = 6, ...)
```

Arguments

s2srtarget	A k-dimensional vector with the lag-1 season-to-season correlation coefficients.
dist	A k-dimensional string vector indicating the quantile function of the target marginal distributions (i.e., the ICDFs).
params	A k-dimensional named list with the parameters of the target distributions.
NatafIntMethod	A string ("GH", "Int", or "MC") indicating the integration method to resolve the Nataf integral.
NoEval	A scalar indicating (default: 9) the number of evaluation points for the integration methods.
polydeg	A scalar indicating the order of the fitted polynomial. If polydeg=0, then another curve is fitted.
...	Additional named arguments for the selected "NatafIntMethod" method.

Value

A list with the parameters of the cyclostationary auxiliary Gaussian PAR(1) model.

Note

Avoid the use of the "GH" method (i.e., NatafIntMethod='GH'), when the marginal(s) are discrete.

Examples

```
## Parameter estimation of a cyclostationary process with 12 seasons.
## Not run:
rtarget<-c(0.5, 0.7, 0.6, 0.4, 0.5, 0.7, 0.8, 0.7, 0.6, 0.4, 0.5, 0.7)
NumOfSeasons=length(rtarget)

FXs<-rep('qmixed',NumOfSeasons)
PFXs<-vector("list",NumOfSeasons)
PFXs<-lapply(PFXs,function(x) x<-list(p0=0.4,Distr=qexp,rate=0.5))

SPARTApar<-EstSPARTA(s2srtarget = rtarget, dist = FXs,
params = PFXs, NatafIntMethod = 'GH', NoEval = 9, polydeg = 8)

## End(Not run)
```

---

mixed — *Zero-Inflated distribution model (i.e., mixed)*

Description

Density, distribution function, quantile function and random generation for the zero-inflated (i.e., mixed) distribution model. This model is composed by two parts, 1) the discrete part, which regards an atom at zero, and 2) the continuous part, which regards a continuous distribution model (J-shaped and with left support >0).

Usage

```
dmixed(x, Distr, p0, ...)

pmixed(q, Distr, p0, ...)

qmixed(p, Distr, p0, ...)

rmixed(n, Distr, p0, ...)
```

Arguments

`x, q`	Vector of quantiles.
`Distr`	The name (as a function) of the continuous distribution model.
`p0`	Probability of zero values (i.e., zero-inflation).
`...`	Additional named arguments containing the continuous distribution parameters
`p`	Vector of probabilities.
`n`	Number of observations. If length(n) > 1, the length is taken to be the number required.

Value

dmixed gives the density of the zero-inflated (i.e., mixed) distribution model. pmixed gives the cdf of the zero-inflated (i.e., mixed) distribution model. qmixed gives the quantile (ICDF) of the zero-inflated (i.e., mixed) distribution model. rmixed gives random variates from the zero-inflated (i.e., mixed) distribution model.

Examples

```
## Plot the CDF of a Gamma distribution.
p=seq(0,1,0.01)
x=qmixed(p, Distr = qgamma, p0=0.7, shape=0.5, scale=1)
plot(x, p)

## Generate 100000 random variables with p0=0.7 and
## Gamma distribution for the continoous part.
X=rmixed(1000, qgamma, p0=0.7, shape=0.5, scale=1)
hist(X)

## Generate 100000 random variables with p0=0.7 and
## Burr type XII distribution for the continuous part.
## The actuar package is required, since it contains
## the d,p,q,r functions of the Burr type XII distribution.
require(actuar)
X=rmixed(1000, qburr, p0=0.7, shape1=5, shape2=1, scale=1)
hist(X)
plot(sort(X[X>0]), 1-ppoints(X[X>0], a=0), log='xy',
xlab = 'x', ylab = 'P[X>x]', main='Prob of exceedance plot')
```

NatafGH — *Solve the Nataf integral with the Gauss-Hermite integration method*

Description

Estimation of the resulting (i.e., in the actual domain) correlation coefficients, given the equivalent correlation coefficients (i.e., in the Gaussian domain).

Usage

```
NatafGH(rho, fx, fy, paramlistfx, paramlistfy, nodes = 21, prune = 0)
```

Arguments

rho	A scalar or vector of correlation coefficients (i.e., in seq(from=0, to=1, by=0.1)).
fx	A string indicating the quantile function of the distribution (i.e., the ICDF).
fy	A string indicating the quantile function of the distribution (i.e., the ICDF).
paramlistfx	A named list with the parameters of the distribution.
paramlistfy	A named list with parameters of the distribution.
nodes	A scalar indicating the number of nodes for Gauss-Hermite integration (default: nodes=21).
prune	A scalar in (0,1) indicating the percentage of pruning for Gauss-Hermite integration (default: nodes=0).

Value

A vector of correlation coefficients in the actual domain

Note

Avoid the use of this function, when the marginal(s) are discrete.

Examples

```
## The case of two identrical Gamma distributions, with shape=1 and scale=1.
## Not run:
fx=fy='qgamma'
pfx=pfy=list(shape=1, scale=1)
rhoz=seq(from=0, to=1 , by=0.05)
rhox=NatafGH(rho = rhoz, fx = fx, fy = fy, paramlistfx = pfx,
paramlistfy = pfy, nodes = 10, prune = 0)
plot(rhoz,rhox); abline(0,1)

## The case of two identrical zero-inflated (i.e., mixed) distributions,
with p0=0.7 a Gamma distribution
## for the continuous part with shape=1 and scale=1.

fx=fy='qmixed'
pfx=pfy=list(Distr=qgamma, p0=0.7, shape=0.5, scale=1)
rhoz=seq(from=0, to=1 , by=0.05)
rhox=NatafGH(rho = rhoz, fx = fx, fy = fy, paramlistfx = pfx,
paramlistfy = pfy, nodes = 21, prune = 0)
plot(rhoz,rhox); abline(0,1)

## End(Not run)
```

NatafInt — *Solve the Nataf integral with an alternative integration method*

Description

Estimation of the resulting (i.e., in the actual domain) correlation coefficients, given the equivalent correlation coefficients (i.e., in the Gaussian domain).

Usage

```
NatafInt(rho, fx, fy, paramlistfx, paramlistfy)
```

Arguments

`rho`	A scalar or vector of correlation coefficients (i.e., in seq(from=0, to=1, by=0.1)).
`fx`	A string indicating the quantile function of the distribution (i.e., the ICDF).
`fy`	A string indicating the quantile function of the distribution (i.e., the ICDF).
`paramlistfx`	A named list with the parameters of the distribution.
`paramlistfy`	A named list with parameters of the distribution.

Value

A vector of correlation coefficients in the actual domain.

Examples

```
## The case of two identrical Gamma distributions, with shape=1 and scale=1.
## Not run:
fx=fy='qgamma'
pfx=pfy=list(shape=1, scale=1)
rhoz=seq(from=0, to=1 , by=0.2)
rhox=NatafInt(rho = rhoz, fx = fx, fy = fy, paramlistfx = pfx, paramlistfy = pfy)
plot(rhoz,rhox); abline(0,1)

## The case identrical Bernoulli distributions, with size=1 and prob=0.3.
fx=fy='qbinom'
pfx=pfy=list(size=1, prob=0.3)
rhoz=seq(from=0, to=1 , by=0.2)
rhox=NatafInt(rho = rhoz, fx = fx, fy = fy, paramlistfx = pfx, paramlistfy = pfy)
plot(rhoz,rhox); abline(0,1)

## The case of two identrical zero-inflated (i.e., mixed) distributions,
## with p0=0.7 a Gamma distribution
## for the continuous part with shape=1 and scale=1.

fx=fy='qmixed'
pfx=pfy=list(Distr=qgamma, p0=0.7, shape=1, scale=1)
rhoz=seq(from=0, to=1 , by=0.2)
rhox=NatafInt(rho = rhoz, fx = fx, fy = fy, paramlistfx = pfx, paramlistfy = pfy)
plot(rhoz,rhox);abline(0,1)
# compare with the Gauss-Hermite integration method
rhox=NatafGH(rho = rhoz, fx = fx, fy = fy, paramlistfx = pfx, paramlistfy = pfy, nodes = 21)
points(rhoz,rhox,col='red',pch=19)

## End(Not run)
```

`NatafInvD` *Direct estimation of equivalent correlation coefficients.*

Description

Direct estimation of equivalent correlation coefficients (i.e., in the Gaussian domain).

Usage

```
NatafInvD(targetrho, fx, fy, paramlistfx, paramlistfy,
  NatafIntMethod = "GH", NoEval = 19, polydeg = 8, ...)
```

Arguments

`targetrho`	A scalar or vector of target correlation coefficients.
`fx`	A string indicating the quantile function of the distribution (i.e., the ICDF).
`fy`	A string indicating the quantile function of the distribution (i.e., the ICDF).

paramlistfx	A named list with the parameters of the distribution.
paramlistfy	A named list with parameters of the distribution.
NatafIntMethod	A string ("GH", "Int", or "MC"), indicating the integration method, to resolve the Nataf integral.
NoEval	A scalar indicating (default: 9) the number of evaluation points for the integration methods.
polydeg	A scalar indicating the order of the fitted polynomial. If polydeg=0, then an alternative parametric curve is fitted (see, Papalexiou, 2018).
...	Additional named arguments for the selected "NatafIntMethod" method.

Value

A named list with two elements: dfnataf: A dataframe that contains the pairs of Gaussian and resulting correlation coefficients, upon which the curve (polynomial or other) was fitted. rzEq: A vector with the equivalent correlation coefficients, that result into the target ones (i.e., targetrho).

Note

Avoid the use of the "GH" method (i.e., NatafIntMethod='GH'), when the marginal(s) are discrete.

Examples

```
## The case of two identrical zero-inflated (i.e., mixed) distributions,
## with p0=0.9 a Gamma distribution
## for the continuous part with shape=0.1 and scale=1.
## Not run:
fx=fy='qmixed'
pfx=pfy=list(Distr=qgamma, p0=0.9, shape=0.1, scale=1)
rhoz=seq(from=0, to=1 , length.out = 21)
rhox=NatafInt(rho = rhoz, fx = fx, fy = fy, paramlistfx = pfx, paramlistfy = pfy)
plot(rhox,rhoz, col='red', pch=19); abline(0,1)
rhotarget=seq(from=0.0001, to=0.9999 , length.out = 210)
req=NatafInvD(targetrho = rhotarget, fx = fx, fy = fy, paramlistfx = pfx, paramlistfy = pfy,
              NatafIntMethod = 'GH', polydeg=8, NoEval = 9)
points(rhotarget, req, col='blue', pch=17);

req=NatafInvD(targetrho = rhotarget, fx = fx, fy = fy, paramlistfx = pfx, paramlistfy = pfy,
              NatafIntMethod = 'GH', polydeg=8, NoEval = 9)
points(rhotarget, req, col='green')

## End(Not run)
## The case with identical Bernoulli distributions, with size=1 and prob=0.2.

## Not run:
fx=fy='qbinom'
pfx=pfy=list(size=1, prob=0.2)
rhoz=seq(from=0, to=1 , length.out = 21)
rhox=NatafInt(rho = rhoz, fx = fx, fy = fy, paramlistfx = pfx, paramlistfy = pfy)
plot(rhox,rhoz, col='red', pch=19); abline(0,1)
rhotarget=seq(from=0.0001, to=0.9999 , length.out = 210)
req=NatafInvD(targetrho = rhotarget, fx = fx, fy = fy, paramlistfx = pfx, paramlistfy = pfy,
              NatafIntMethod = 'Int', polydeg=8, NoEval = 9)$rzEq
points(rhotarget, req, col='blue', pch=17);
```

```
req2=NatafInvD(targetrho = rhotarget, fx = fx, fy = fy, paramlistfx = pfx, paramlistfy = pfy,
              NatafIntMethod = 'Int', polydeg=8, NoEval = 9)$rzEq
points(rhotarget, req2, col='green')

## End(Not run)
```

NatafMC *Solve the Nataf integral with the Monte-Carlo method*

Description

Estimation of the resulting (i.e., in the actual domain) correlation coefficients, given the equivalent correlation coefficients (i.e., in the Gaussian domain).

Usage

```
NatafMC(rho, fx, fy, paramlistfx, paramlistfy, MCsize = 10^5)
```

Arguments

rho	A scalar or vector of correlation coefficients (i.e., in seq(from=0, to=1, by=0.1)).
fx	A string indicating the quantile function of the distribution (i.e., the ICDF).
fy	A string indicating the quantile function of the distribution (i.e., the ICDF).
paramlistfx	A named list with the parameters of the distribution.
paramlistfy	A named list with parameters of the distribution.
MCsize	A scalar determining the number of Monte-Carlo trials (default: 10^-5)

Value

A vector of correlation coefficients in the actual domain.

Examples

```
## The case of two identical Gamma distributions, with shape=1 and scale=1.
## Not run:
fx=fy='qgamma'
pfx=pfy=list(shape=1, scale=1)
rhoz=seq(from=0, to=1 , by=0.2)
rhoxMC=NatafMC(rho = rhoz, fx = fx, fy = fy, paramlistfx = pfx, paramlistfy = pfy)
plot(rhoz,rhoxMC, col='blue', pch=19); abline(0,1)
rhoxGH=NatafGH(rho = rhoz, fx = fx, fy = fy, paramlistfx = pfx, paramlistfy = pfy)
points(rhoz, rhoxGH, col='red', pch=5)

## The case identrical Bernoulli distributions, with size=1 and prob=0.3.

fx=fy='qbinom'
pfx=pfy=list(size=1, prob=0.3)
rhoz=seq(from=0, to=1 , by=0.2)
rhoxMC=NatafMC(rho = rhoz, fx = fx, fy = fy, paramlistfx = pfx, paramlistfy = pfy)
plot(rhoz,rhoxMC, col='blue', pch=19); abline(0,1)
```

```
## End(Not run)

## Not run:
rhoxInt=NatafInt(rho = rhoz, fx = fx, fy = fy, paramlistfx = pfx, paramlistfy = pfy)
points(rhoz, rhoxInt, col='red', pch=5)

## End(Not run)
```

---

s2scor *Estimation of lag-1 season-to-season correlation coefficients*

Description

Estimation of lag-1 seasson-to-season correlation coefficients.

Usage

```
s2scor(data)
```

Arguments

data	A matrix of dimensions k x m, where m denotes the number of sub-seasons and k the number of periods. For instance k may refer to years and m to months (i.e., 12). Another example could regard k as days (e.g., 31 x years) and m to hours (i.e., 24).

Value

A k-dimensional vector with the lag-1 season-to-season correlations coefficients.

Examples

```
## Simulation of cyclostationary process with 12 seasons and zero-inflated marginal distributions.
rtarget<-c(0.5, 0.7, 0.6, 0.4, 0.5, 0.7, 0.8, 0.7, 0.6, 0.4, 0.5, 0.7)
NumOfSeasons=length(rtarget)
FXs<-rep('qmixed',NumOfSeasons)
PFXs<-vector("list",NumOfSeasons)
PFXs<-lapply(PFXs,function(x) x<-list(p0=0.4, Distr=qexp, rate=0.5))

SPARTApar<-EstSPARTA(s2srtarget = rtarget, dist = FXs, params = PFXs,
NatafIntMethod = 'GH', NoEval = 9, polydeg = 8)

sim<-SimSPARTA(SPARTApar = SPARTApar, steps=100000, stand=0)
s2scor(sim$X)
plot(s2scor(sim$X)); lines(rtarget, col='red')
```

SimARTAp *Simulation of the target stationary process using the ARTA(p) model.*

Description

Simulation of the target stationary process using an ARTA(p) model to simulate the auxiliary Gaussian process and establish the target correlation structure.

Usage

```
SimARTAp(ARTApar, burn = 1000, steps = 10000, stand = F)
```

Arguments

ARTApar	A list containing the parameters of the model. The list is constructed by the function "EstARTAp".
burn	A scalar specifying the length of burn-out sample.
steps	A scalar specifying the length of the time series to be generated.
stand	A boolean (T or F) indicating whether to standardize (or not) the auxiliary Gaussian time series prior to their mapping to the actual domain. The default value is FALSE.

Value

A list of the 3 generated time series (in vector format): X: The final time series at the actual domain with the target marginal distribution and correlation structure; Z: The auxiliary Gaussian time series at the Gaussian domain and; U: The auxiliary uniform time series at the Copula domain (i.e., in [0,1]).

Examples

```
## Simulation of a process with a zero-inflated (i.e., mixed)
## marginal distribution, with p0=0.9, and a Gamma distribution
## for the continuous part with shape=0.1 and scale=1.
## In this case, the target autocorrelation strucure is from
## the CAS ACS with b=1 and k=0.6.
## Not run:
ACF=acsCAS(param=c(1, 0.6), lag=100, var=1)
fx='qmixed'
pfx=list(Distr=qgamma, p0=0.9, shape=0.1, scale=1)

ARTApar=EstARTAp(ACF=ACF, maxlag=0, dist=fx, params=pfx,
NatafIntMethod ='GH', NoEval=9, polydeg=8)

Sim=SimARTAp(ARTApar = ARTApar, burn = 1000, steps = 10^5, stand = 0)
acf(Sim$X)
lines(0:(length(ACF)-1), ACF)
plot(Sim$X[1:1000], type='l', col='red')

## Simulation of a process with a bernoulli marginal distribution,
## with size=1, and prob=0.2.
```

```
## In this case, the target autocorrelation strucure
## is from an fGn process (i.e., Hurst) with H=0.7.

ACF=acsHurst(H=0.7, lag=500, var=1)
fx='qbinom'
pfx=list(size=1, prob=0.2)

ARTApar=EstARTAp(ACF=ACF, maxlag=0, dist=fx, params=pfx,
NatafIntMethod ='Int', NoEval=9, polydeg=0)

Sim=SimARTAp(ARTApar = ARTApar, burn = 1000, steps = 10^5, stand = 0)
acf(Sim$X)
lines(0:(length(ACF)-1), ACF)
plot(Sim$X[1:1000], type='l', col='red')

## Simulation of a process with a Beta marginal distribution,
## with shape=2, and shape2=10.
## In this case, the target autocorrelation strucure is from
## an fGn process (i.e., Hurst) with H=0.7.

ACF=acsHurst(H=0.7, lag=500, var=1)
fx='qbeta'
pfx=list(shape1=2, shape2=10)

ARTApar=EstARTAp(ACF=ACF, maxlag=0, dist=fx, params=pfx,
NatafIntMethod ='Int', NoEval=9, polydeg=0)

Sim=SimARTAp(ARTApar = ARTApar, burn = 1000, steps = 10^5, stand = 0)
acf(Sim$X)
lines(0:(length(ACF)-1), ACF)
plot(Sim$X[1:1000], type='l', col='red')

## End(Not run)
```

SimnAR1	*Simulation of the target stationary process using the n-ARTA(1) model.*

Description

Simulation of the target stationary process using an n-ARTA(1) model to simulate an auxiliary Gaussian process and establish the target correlation structure.

Usage

```
SimnAR1(nAR1param, steps)
```

Arguments

nAR1param	A list containing the parameters of the model. The list is constructed by the function "estnAR1".
steps	A scalar specifying the length of the time series to be generated.

Value

A list with 3 generated time series (in vector format): X: The final time series at the actual domain with the target marginal distribution and correlation structure; Z: The auxiliary Gaussian time series at the Gaussian domain and; U: The auxiliary uniform time series at the Copula domain (i.e., in [0,1]).

Examples

```
## Parameter estimation for a process with zero-inflated (i.e., mixed) marginal distribution,
## with p0=0.8, and a Gamma distribution
## for the continuous part with shape=0.5 and scale=1.
## In this case, the Autocorrelation strucure is a CAS ACS with b=2 and k=0.5.
## Not run:
ACF=acsCAS(param = c(2, 0.5), lag = 500, var = 1)
dist='qmixed'
params=list(Distr=qgamma, p0=0.8, shape=0.5, scale=1)
nAR1param=EstnAR1(ACF = ACF, Ar1Num = 3, dist = dist, params = params,
                  NatafIntMethod = 'GH', NoEval = 9, polydeg = 8)

Sim=SimnAR1(nAR1param = nAR1param, steps = 10^6)

acf(Sim$X)
lines(0:500,ACF,col='red')
plot.ecdf(Sim$X)

## End(Not run)
```

SimSMARTA — *Simulation of the target stationary processes using the SMARTA model.*

Description

Simulation of the target stationary processes using an SMARTA model to simulate the auxiliary Gaussian process and establish the target correlation structure. This model allows also the simulation of a multivariate process.

Usage

```
SimSMARTA(SMARTApar, steps, SMALAG = 512)
```

Arguments

SMARTApar	A list containing the parameters of the model. The list is constructed by the function "EstSMARTA".
steps	A scalar specifying the length of the time series to be generated.
SMALAG	A scalar specifying the order of the SMARTA model (must be equal to FFTAG and the lengths of ACFs).

Value

A list of the 3 simulated time series (in matrix format - i.e., matrix of dimensions steps x k, where steps denotes the length of generated time series and k the number of processes). X: The final time series at the actual domain with the target marginal distribution and correlation structure; Z: The auxiliary Gaussian time series at the Gaussian domain and; U: The auxiliary uniform time series at the Copula domain (i.e., in [0,1]).

Examples

```
## Simulation example of a bivariate process with zero-inflated marginal distributions.
# Define the simulation parameters ----------------------------------------
## Not run:
LAG=2^6
FFTLag=2^7
SMALAG=2^6
steps=2^14

PFXs=list()
FXs=c('qmixed','qmixed')
# Gamma distribution: Gamma(shape=2, rate=1)
PFXs[[1]]=list(Distr=qgamma, p0=0.9, shape=1, scale=1)
# Weibull distribution: Weibull(shape=1, scale=2)
PFXs[[2]]=list(Distr=qweibull, p0=0.85, shape=1, scale=2)

ACFs=list()
ACFs[[1]]=acsCAS(param = c(0.1, 0.6), lag = LAG)
ACFs[[2]]=acsCAS(param = c(0.2, 0.3), lag = LAG)

Cmat=matrix(c(1,0.6,0.6,1), ncol=2, nrow=2)

# Calculate SMARTA's parameters -------------------------------------------
SMAparam=EstSMARTA(dist = FXs, params = PFXs, ACFs = ACFs, Cmat = Cmat,
DecoMethod = 'cor.smooth',FFTLag = FFTLag,
NatafIntMethod = 'GH', NoEval = 9, polydeg = 8)

# Simulate a SMARTA process -----------------------------------------------
Sim=SimSMARTA(SMARTApar = SMAparam, steps = steps, SMALAG = SMALAG)

# Draw some basic plots ---------------------------------------------------
for (i in 1:2) {
 par(mfrow=c(2,2))
  plot(Sim$X[1:1001,i], type='l')
 acf(Sim$X[,i], lag.max = 20); lines(0:20,ACFs[[i]][1:21], col='red', type='o')
 plot(ecdf(Sim$X[,i]))
  hist(Sim$X[,i],probability = TRUE)
}

## End(Not run)

## Not run:
## Simulation example of a bivariate process with Bernoulli marginal distributions.
# Define the simulation parameters ----------------------------------------
PFXs=list()
FXs=c('qbinom','qbinom')
PFXs[[1]]=list(size=1, prob=0.2)# Gamma distribution: Gamma(shape=2, rate=1)
PFXs[[2]]=list(size=1, prob=0.25) # Weibull distribution: Weibull(shape=1, scale=2)
```

```
ACFs=list()
ACFs[[1]]=acsCAS(param = c(0.1, 0.6), lag = LAG)
ACFs[[2]]=acsCAS(param = c(0.2, 0.3), lag = LAG)

Cmat=matrix(c(1,0.6,0.6,1), ncol=2, nrow=2)

# Calculate SMARTAâ€™s parameters ------------------------------------------
SMAparam=EstSMARTA(dist = FXs, params = PFXs, ACFs = ACFs, Cmat = Cmat, DecoMethod = 'cor.smooth',
                   FFTLag = FFTLag, NatafIntMethod = 'Int', NoEval = 9, polydeg = 8)
# Simulate a SMARTA process ---------------------------------------------
Sim=SimSMARTA(SMARTApar = SMAparam, steps = steps, SMALAG = SMALAG)

# Draw some basic plots ---------------------------------------------------
for (i in 1:2) {
  par(mfrow=c(2,2))
  plot(Sim$X[1:1001,i], type='l')
  acf(Sim$X[,i], lag.max = 20); lines(0:20,ACFs[[i]][1:21], col='red', type='o')
  plot(ecdf(Sim$X[,i]))
  hist(Sim$X[,i],probability = TRUE)
}

## End(Not run)
```

SimSPARTA — *Simulation of the target cyclostationary process using the SPARTA model of order 1.*

Description

Simulation of the target cyclostationary process using a PAR(p) model (i.e., the SPARTA model of order 1) to simulate the auxiliary cyclostationary Gaussian process to establish the target season-to-season correlation structure.

Usage

```
SimSPARTA(SPARTApar, steps = 1000, stand = 0)
```

Arguments

`SPARTApar`	A list containing the parameters of the model. The list is constructed by the function "EstSPARTA".
`steps`	A scalar specifying the length of the time series to be generated.
`stand`	A boolean (T or F) indicating whether to standardize (or not) the auxiliary Gaussian time series prior to their mapping to the actual domain. The default value is FALSE.

Value

A list of 3 generated time series (in matrix format - i.e., matrix of dimensions k x m, where m denotes the number of sub-seasons and k the number of periods.): X: The final time series at the actual domain with the target marginal distribution and correlation structure; Z: The auxiliary Gaussian time series at the Gaussian domain and, U: The auxiliary uniform time series at the Copula domain (i.e., in [0,1]).

Examples

```
## Simulation of cyclostationary process with 12 seasons and zero-inflated marginal distributions.
## Not run:
rtarget<-c(0.5, 0.7, 0.6, 0.4, 0.5, 0.7, 0.8, 0.7, 0.6, 0.4, 0.5, 0.7)
NumOfSeasons=length(rtarget)
FXs<-rep('qmixed',NumOfSeasons)
PFXs<-vector("list",NumOfSeasons)
PFXs<-lapply(PFXs,function(x) x<-list(p0=0.4, Distr=qexp, rate=0.5))

SPARTApar<-EstSPARTA(s2srtarget = rtarget, dist = FXs, params = PFXs,
NatafIntMethod = 'GH', NoEval = 9, polydeg = 8)

sim<-SimSPARTA(SPARTApar = SPARTApar, steps=100000, stand=0)
s2scor(sim$X)
plot(s2scor(sim$X)); lines(rtarget, col='red')

## End(Not run)
```

Index