Package 'trawl'

August 16, 2018

Type Package

Title Estimation and Simulation of Trawl Processes

Version 0.2.1

Author Almut E. D. Veraart

Maintainer Almut E. D. Veraart <a.veraart@imperial.ac.uk>

Description Contains R functions for simulating and estimating integer-valued trawl processes as described in the article ``Modelling, simulation and inference for multivariate time series of counts using trawl processes'' by A. E. D. Veraart (Journal of Multivariate Analysis, 2018, to appear, preprint available at: <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3100076>) and for simulating random vectors from the bivariate negative binomial and the bi- and trivariate logarithmic series distributions.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

Depends R (>= 3.4.0)

Imports DEoptim, graphics, MASS, rootSolve, Runuran, squash, stats, TSA

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2018-08-16 11:50:03 UTC

R topics documented:

`acf_DExp` *Autocorrelation function of the double exponential trawl function*

Description

This function computes the autocorrelation function associated with the double exponential trawl function.

Usage

```
acf_DExp(x, w, lambda1, lambda2)
```

Arguments

`x`	The argument (lag) at which the autocorrelation function associated with the double exponential trawl function will be evaluated
`w`	parameter in the double exponential trawl
`lambda1`	parameter in the double exponential trawl
`lambda2`	parameter in the double exponential trawl

Details

The trawl function is parametrised by parameters $0 \leq w \leq 1$ and $\lambda_1, \lambda_2 > 0$ as follows:

$$g(x) = we^{\lambda_1 x} + (1-w)e^{\lambda_2 x}, \text{ for } x \leq 0.$$

Its autocorrelation function is given by:

$$r(x) = (we^{-\lambda_1 x}/\lambda_1 + (1-w)e^{-\lambda_2 x}/\lambda_2)/c, \text{ for } x \geq 0,$$

where

$$c = w/\lambda_1 + (1-w)/\lambda_2.$$

Value

The autocorrelation function of the double exponential trawl function evaluated at x

Examples

```
#Evaluate the trawl autocorrelation function at x=1
acf_DExp(1,0.3,0.1,2)
#Plot the trawl autocorrelation function
plot(acf_DExp((0:10),0.3,0.1,2))
```

`acf_Exp` *Autocorrelation function of the exponential trawl function*

Description

This function computes the autocorrelation function associated with the exponential trawl function.

Usage

```
acf_Exp(x, lambda)
```

Arguments

`x`	The argument (lag) at which the autocorrelation function associated with the exponential trawl function will be evaluated
`lambda`	parameter in the exponential trawl

Details

The trawl function is parametrised by the parameter $\lambda > 0$ as follows:

$$g(x) = e^{\lambda x}, \text{ for } x \leq 0.$$

Its autocorrelation function is given by:

$$r(x) = e^{-\lambda x}, \text{ for } x \geq 0.$$

Value

The autocorrelation function of the exponential trawl function evaluated at x

Examples

```
#Evaluate the trawl autocorrelation function at x=1
acf_Exp(1,0.1)
#Plot the trawl autocorrelation function
plot(acf_Exp((0:10),0.1))
```

---

acf_LM *Autocorrelation function of the long memory trawl function*

Description

This function computes the autocorrelation function associated with the long memory trawl function.

Usage

```
acf_LM(x, alpha, H)
```

Arguments

x	The argument (lag) at which the autocorrelation function associated with the long memory trawl function will be evaluated
alpha	parameter in the long memory trawl
H	parameter in the long memory trawl

Details

The trawl function is parametrised by the two parameters $H > 1$ and $\alpha > 0$ as follows:

$$g(x) = (1 - x/\alpha)^{-H}, \text{ for } x \leq 0.$$

Its autocorrelation function is given by

$$r(x) = (1 + x/\alpha)^{(1-H)}, \text{ for } x \geq 0.$$

Value

The autocorrelation function of the long memory trawl function evaluated at x

Examples

```
#Evaluate the trawl autocorrelation function at x=1
acf_LM(1,0.3,1.5)
#Plot the trawl autocorrelation function
plot(acf_LM((0:10),0.3,1.5))
```

---

acf_supIG — *Autocorrelation function of the supIG trawl function*

Description

This function computes the autocorrelation function associated with the supIG trawl function.

Usage

```
acf_supIG(x, delta, gamma)
```

Arguments

x	The argument (lag) at which the autocorrelation function associated with the supIG trawl function will be evaluated
delta	parameter in the supIG trawl
gamma	parameter in the supIG trawl

Details

The trawl function is parametrised by the two parameters $\delta \geq 0$ and $\gamma \geq 0$ as follows:

$$g(x) = (1 - 2x\gamma^{-2})^{-1/2} \exp(\delta\gamma(1 - (1 - 2x\gamma^{-2})^{1/2})), \text{ for } x \leq 0.$$

It is assumed that δ and γ are not simultaneously equal to zero. Its autocorrelation function is given by:

$$r(x) = \exp(\delta\gamma(1 - \sqrt{1 + 2x/\gamma^2})), \text{ for } x \geq 0.$$

Value

The autocorrelation function of the supIG trawl function evaluated at x

Examples

```
#Evaluate the trawl autocorrelation function at x=1
acf_supIG(1,0.3,0.1)
#Plot the trawl autocorrelation function
plot(acf_supIG((0:10),0.3,0.1))
```

`Bivariate_LSDsim` *Simulates from the bivariate logarithmic series distribution*

Description

Simulates from the bivariate logarithmic series distribution

Usage

```
Bivariate_LSDsim(N, p1, p2)
```

Arguments

`N`	number of data points to be simulated
`p1`	parameter p_1 of the bivariate logarithmic series distribution
`p2`	parameter p_2 of the bivariate logarithmic series distribution

Details

The probability mass function of a random vector $X = (X_1, X_2)'$ following the bivariate logarithmic series distribution with parameters $0 < p_1, p_2 < 1$ with $p := p_1 + p_2 < 1$ is given by

$$P(X_1 = x_1, X_2 = x_2) = \frac{\Gamma(x_1 + x_2)}{x_1!x_2!} \frac{p_1^{x_1} p_2^{x_2}}{(-\log(1-p))},$$

for $x_1, x_2 = 0, 1, 2, \ldots$ such that $x_1 + x_2 > 0$. The simulation proceeds in two steps: First, X_1 is simulated from the modified logarithmic distribution with parameters $\tilde{p}_1 = p_1/(1 - p_2)$ and $\delta_1 = \log(1 - p_2)/\log(1 - p)$. Then we simulate X_2 conditional on X_1. We note that $X_2|X_1 = x_1$ follows the logarithmic series distribution with parameter p_2 when $x_1 = 0$, and the negative binomial distribution with parameters (x_1, p_2) when $x_1 > 0$.

Value

An $N \times 2$ matrix with N simulated values from the bivariate logarithmic series distribution

Examples

```
set.seed(1)
p1 <- 0.15
p2 <- 0.3
N <- 100
#Simulate N realisations from the bivariate LSD
y <- Bivariate_LSDsim(N, p1, p2)
```

`Bivariate_NBsim` *Simulates from the bivariate negative binomial distribution*

Description

Simulates from the bivariate negative binomial distribution

Usage

```
Bivariate_NBsim(N, kappa, p1, p2)
```

Arguments

N	number of data points to be simulated
kappa	parameter κ of the bivariate negative binomial distribution
p1	parameter p_1 of the bivariate negative binomial distribution
p2	parameter p_2 of the bivariate negative binomial distribution

Details

A random vector $\mathbf{X} = (X_1, X_2)'$ is said to follow the bivariate negative binomial distribution with parameters κ, p_1, p_2 if its probability mass function is given by

$$P(\mathbf{X} = \mathbf{x}) = \frac{\Gamma(x_1 + x_2 + \kappa)}{x_1!x_2!\Gamma(\kappa)} p_1^{x_1} p_2^{x_2} (1 - p_1 - p_2)^{\kappa},$$

where, for $i = 1, 2$, $x_i \in \{0, 1, \ldots\}$, $0 < p_i < 1$ such that $p_1 + p_2 < 1$ and $\kappa > 0$.

Value

An $N \times 2$ matrix with N simulated values from the bivariate negative binomial distribution

Examples

```
set.seed(1)
kappa <- 3
p1 <- 0.1
p2 <- 0.85
N <- 100
#Simulate N realisations from the bivariate negative binomial distribution
y <- Bivariate_NBsim(N,kappa,p1,p2)
```

BivLSD_Cor — *Computes the correlation of the components of a bivariate vector following the bivariate logarithmic series distribution*

Description

Computes the correlation of the components of a bivariate vector following the bivariate logarithmic series distribution

Usage

```
BivLSD_Cor(p1, p2)
```

Arguments

p1	parameter p_1 of the bivariate logarithmic series distribution
p2	parameter p_2 of the bivariate logarithmic series distribution

Value

Correlation of the components of a bivariate vector following the bivariate logarithmic series distribution

Examples

```
BivLSD_Cor(0.2,0.5)
```

BivLSD_Cov — *Computes the covariance of the components of a bivariate vector following the bivariate logarithmic series distribution*

Description

Computes the covariance of the components of a bivariate vector following the bivariate logarithmic series distribution

Usage

```
BivLSD_Cov(p1, p2)
```

Arguments

p1	parameter p_1 of the bivariate logarithmic series distribution
p2	parameter p_2 of the bivariate logarithmic series distribution

Value

Covariance of the components of a bivariate vector following the bivariate logarithmic series distribution

Examples

```
BivLSD_Cov(0.2,0.5)
```

`BivModLSD_Cor` *Computes the correlation of the components of a bivariate vector following the bivariate modified logarithmic series distribution*

Description

Computes the correlation of the components of a bivariate vector following the bivariate modified logarithmic series distribution

Usage

```
BivModLSD_Cor(delta, p1, p2)
```

Arguments

`delta`	parameter δ of the bivariate modified logarithmic series distribution
`p1`	parameter p_1 of the bivariate modified logarithmic series distribution
`p2`	parameter p_2 of the bivariate modified logarithmic series distribution

Value

Covariance of the components of a bivariate vector following the bivariate modified logarithmic series distribution

Examples

```
BivModLSD_Cor(0.2, 0.3, 0.5)
```

BivModLSD_Cov — *Computes the covariance of the components of a bivariate vector following the bivariate modified logarithmic series distribution*

Description

Computes the covariance of the components of a bivariate vector following the bivariate modified logarithmic series distribution

Usage

```
BivModLSD_Cov(delta, p1, p2)
```

Arguments

delta	parameter δ of the bivariate modified logarithmic series distribution
p1	parameter p_1 of the bivariate modified logarithmic series distribution
p2	parameter p_2 of the bivariate modified logarithmic series distribution

Value

Covariance of the components of a bivariate vector following the bivariate modified logarithmic series distribution

Examples

```
BivModLSD_Cov(0.2, 0.3, 0.5)
```

fit_DExptrawl — *Fits the trawl function consisting of the weighted sum of two exponential functions*

Description

Fits the trawl function consisting of the weighted sum of two exponential functions

Usage

```
fit_DExptrawl(x, Delta = 1, GMMlag = 5, plotacf = FALSE,
  lags = 100)
```

Arguments

x	vector of equidistant time series data
Delta	interval length of the time grid used in the time series, the default is 1
GMMlag	lag length used in the GMM estimation, the default is 5
plotacf	binary variable specifying whether or not the empirical and fitted autocorrelation function should be plotted
lags	number of lags to be used in the plot of the autocorrelation function

Details

The trawl function is parametrised by the three parameters $0 \leq w \leq 1$ and $\lambda_1, \lambda_2 > 0$ as follows:

$$g(x) = we^{\lambda_1 x} + (1-w)e^{\lambda_2 x}, \text{ for } x \leq 0.$$

The Lebesgue measure of the corresponding trawl set is given by $w/\lambda_1 + (1-w)/\lambda_2$.

Value

w: the weight parameter (restricted to be in [0,0.5] for identifiability reasons)

lambda1: the first memory parameter (denoted by λ_1 above)

lambda2: the second memory parameter (denoted by λ_2 above)

LM: The Lebesgue measure of the trawl set associated with the double exponential trawl

Examples

```
#Simulate a univariate trawl process and fit the double exponential trawl
#function
set.seed(1)
t <- 1000
Delta <- 1
v <- 250
w <- 0.1
lambda1 <- 0.1
lambda2 <- 1
#Simulate a univariate trawl process with double exponential trawl function
#and Poisson marginal law
trawl <- sim_UnivariateTrawl(t,Delta,burnin=50,marginal =c("Poi"),trawl
="DExp",v=v, w=w,lambda1=lambda1,lambda2=lambda2)
#Fit the double exponential trawl function to the simulated data
fittrawlfct <- fit_DExptrawl(trawl,Delta, plotacf=TRUE,lags=500)
#Print the results
print(paste("w: estimated:", fittrawlfct$w, ", theoretical:", w))
print(paste("lambda1: estimated:", fittrawlfct$lambda1, ", theoretical:",
lambda1))
print(paste("lambda2: estimated:", fittrawlfct$lambda2, ", theoretical:",
lambda2))
```

fit_Exptrawl — *Fits an exponential trawl function to equidistant time series data*

Description

Fits an exponential trawl function to equidistant time series data

Usage

```
fit_Exptrawl(x, Delta = 1, plotacf = FALSE, lags = 100)
```

Arguments

x	vector of equidistant time series data
Delta	interval length of the time grid used in the time series, the default is 1
plotacf	binary variable specifying whether or not the empirical and fitted autocorrelation function should be plotted
lags	number of lags to be used in the plot of the autocorrelation function

Details

The trawl function is parametrised by the parameter $\lambda > 0$ as follows:

$$g(x) = e^{\lambda x}, \text{ for } x \leq 0.$$

The Lebesgue measure of the corresponding trawl set is given by $1/\lambda$.

Value

lambda: the memory parameter λ in the exponential trawl

LM: the Lebesgue measure of the trawl set associated with the exponential trawl, i.e. $1/\lambda$.

Examples

```
#Simulate a univariate trawl process and fit the exponential trawl function
set.seed(1)
t <- 1000
Delta <- 1
v <- 250
lambda <- 0.25
#Simulate a univariate trawl process with exponential trawl function and
#Poisson marginal law
trawl <- sim_UnivariateTrawl(t,Delta,burnin=50,marginal =c("Poi"),trawl
="Exp",v=v, lambda1=lambda)
#Fit the exponential trawl function to the simulated data
fittrawlfct <- fit_Exptrawl(trawl,Delta, plotacf=TRUE,lags=500)
#Print the results
```

```
print(paste("lambda: estimated:", fittrawlfct$lambda, ", theoretical:",
lambda))
```

---

fit_LMtrawl *Fits a long memory trawl function to equidistant univariate time series data*

Description

Fits a long memory trawl function to equidistant univariate time series data

Usage

```
fit_LMtrawl(x, Delta = 1, GMMlag = 5, plotacf = FALSE, lags = 100)
```

Arguments

x	vector of equidistant time series data
Delta	interval length of the time grid used in the time series, the default is 1
GMMlag	lag length used in the GMM estimation, the default is 5
plotacf	binary variable specifying whether or not the empirical and fitted autocorrelation function should be plotted
lags	number of lags to be used in the plot of the autocorrelation function

Details

The trawl function is parametrised by the two parameters $H > 1$ and $\alpha > 0$ as follows:

$$g(x) = (1 - x/\alpha)^{-H}, \text{ for } x \leq 0.$$

The Lebesgue measure of the corresponding trawl set is given by $\alpha/(1 - H)$.

Value

alpha: parameter in the long memory trawl

H: parameter in the long memory trawl

LM: The Lebesgue measure of the trawl set associated with the long memory trawl

Examples

```
#Simulate a univariate trawl process and fit the long memory trawl function
set.seed(1)
t <- 1000
Delta <- 1
v <- 250
alpha <- 0.01
H <- 1.3
#Simulate a univariate trawl process with LM trawl function and Poisson
#marginal law
trawl <- sim_UnivariateTrawl(t,Delta,burnin=50,marginal =c("Poi"),trawl
="LM",v=v, alpha=alpha, H=H)
#Fit the LM trawl function to the simulated data
fittrawlfct <- fit_LMtrawl(trawl,Delta, plotacf=TRUE,lags=500)
#Print the results
print(paste("alpha: estimated:", fittrawlfct$alpha, ", theoretical:", alpha))
print(paste("H: estimated:", fittrawlfct$H, ", theoretical:", H))
```

`fit_marginalNB` *Fist a negative binomial distribution as marginal law*

Description

Fist a negative binomial distribution as marginal law

Usage

```
fit_marginalNB(x, LM, plotdiag = FALSE)
```

Arguments

`x`	vector of equidistant time series data
`LM`	Lebesgue measure of the estimated trawl
`plotdiag`	binary variable specifying whether or not diagnostic plots should be provided

Details

The moment estimator for the parameters of the negative binomial distribution are given by

$$\hat{\theta} = 1 - \mathrm{E}(X)/\mathrm{Var}(X),$$

and

$$\hat{m} = \mathrm{E}(X)(1-\hat{\theta})/(\widehat{\mathrm{LM}}\hat{\theta}).$$

Value

m: parameter in the negative binomial marginal distribution

theta: parameter in the negative binomial marginal distribution

a: Here $a = \theta/(1-\theta)$. This is given for an alternative parametrisation of the negative binomial marginal distribution

Examples

```
#Simulate a univariate trawl process and fit the exponential trawl function
#and the marginal negative binomial law
set.seed(1)
t <- 1000
Delta <- 1
m<-200
theta<-0.5
lambda <- 0.25
#Simulate a univariate trawl process with exponential trawl function and
#negative binomial marginal law
trawl <- sim_UnivariateTrawl(t,Delta,burnin=50,marginal =c("NegBin"),trawl
="Exp",m=m, theta=theta, lambda1=lambda)
#Fit the exponential trawl function to the simulated data
fittrawlfct <- fit_Exptrawl(trawl,Delta, plotacf=TRUE,lags=500)
#Fit the Poisson marginal law
fitmarginallaw <- fit_marginalNB(trawl, fittrawlfct$LM, plotdiag=TRUE)
#Print the results
print(paste("lambda: estimated:", fittrawlfct$lambda, ", theoretical:",
lambda))
print(paste("m: estimated:", fitmarginallaw$m, ", theoretical:", m))
print(paste("theta: estimated:", fitmarginallaw$theta, ", theoretical:",
theta))
```

---

fit_marginalPoisson *Fits a Poisson distribution as marginal law*

Description

Fits a Poisson distribution as marginal law

Usage

```
fit_marginalPoisson(x, LM, plotdiag = FALSE)
```

Arguments

x	vector of equidistant time series data
LM	Lebesgue measure of the estimated trawl
plotdiag	binary variable specifying whether or not diagnostic plots should be provided

Details

The moment estimator for the Poisson rate parameter is given by

$$\hat{v} = \mathrm{E}(X)/\widehat{\mathrm{LM}}.$$

Value

v: the rate parameter in the Poisson marginal distribution

Examples

```
#Simulate a univariate trawl process and fit the exponential trawl function
#and the marginal Poisson law
set.seed(1)
t <- 1000
Delta <- 1
v <- 250
lambda <- 0.25
#Simulate a univariate trawl process with exponential trawl function and
#Poisson marginal law
trawl <- sim_UnivariateTrawl(t,Delta,burnin=50,marginal =c("Poi"),trawl
="Exp",v=v, lambda1=lambda)
#Fit the exponential trawl function to the simulated data
fittrawlfct <- fit_Exptrawl(trawl,Delta, plotacf=TRUE,lags=500)
#Fit the Poisson marginal law
fitmarginallaw <- fit_marginalPoisson(trawl, fittrawlfct$LM, plotdiag=TRUE)
#Print the results
print(paste("lambda: estimated:", fittrawlfct$lambda, ", theoretical:",
lambda))
print(paste("v: estimated:", fitmarginallaw$v, ", theoretical:", v))
```

---

fit_supIGtrawl *Fits a supIG trawl function to equidistant univariate time series data*

Description

Fits a supIG trawl function to equidistant univariate time series data

Usage

```
fit_supIGtrawl(x, Delta = 1, GMMlag = 5, plotacf = FALSE,
  lags = 100)
```

Arguments

`x`	vector of equidistant time series data
`Delta`	interval length of the time grid used in the time series, the default is 1
`GMMlag`	lag length used in the GMM estimation, the default is 5
`plotacf`	binary variable specifying whether or not the empirical and fitted autocorrelation function should be plotted
`lags`	number of lags to be used in the plot of the autocorrelation function

Details

The trawl function is parametrised by the two parameters $\delta \geq 0$ and $\gamma \geq 0$ as follows:

$$g(x) = (1 - 2x\gamma^{-2})^{-1/2} \exp(\delta\gamma(1 - (1 - 2x\gamma^{-2})^{1/2})), \text{ for } x \leq 0.$$

It is assumed that δ and γ are not simultaneously equal to zero. The Lebesgue measure of the corresponding trawl set is given by γ/δ.

Value

delta: parameter in the supIG trawl

gamma: parameter in the supIG trawl

LM: The Lebesgue measure of the trawl set associated with the supIG trawl

Examples

```
#Simulate a univariate trawl process and fit the supIG trawl function
set.seed(1)
t <- 1000
Delta <- 1
v <- 250
delta <- 0.5
gamma <- 1
#Simulate a univariate trawl process with supIG trawl function and
#Poisson marginal law
trawl <- sim_UnivariateTrawl(t,Delta,burnin=50,marginal =c("Poi"),trawl
="supIG",v=v, delta=delta,gamma=gamma)
#Fit the supIG trawl function to the simulated data
fittrawlfct <- fit_supIGtrawl(trawl,Delta, plotacf=TRUE,lags=500)
#Print the results
print(paste("delta: estimated:", fittrawlfct$delta, ", theoretical:", delta))
print(paste("gamma: estimated:", fittrawlfct$gamma, ", theoretical:", gamma))
```

fit_trawl_intersection

Finds the intersection of two trawl sets

Description

Finds the intersection of two trawl sets

Usage

```
fit_trawl_intersection(fct1 = base::c("Exp", "DExp", "supIG", "LM"),
  fct2 = base::c("Exp", "DExp", "supIG", "LM"), lambda11 = 0,
  lambda12 = 0, w1 = 0, delta1 = 0, gamma1 = 0, alpha1 = 0,
  H1 = 0, lambda21 = 0, lambda22 = 0, w2 = 0, delta2 = 0,
  gamma2 = 0, alpha2 = 0, H2 = 0, LM1, LM2, plotdiag = FALSE)
```

Arguments

`fct1`	specifies the type of the first trawl function
`fct2`	specifies the type of the second trawl function
`lambda11, lambda12, w1`	parameters of the (double) exponential trawl functions of the first process
`delta1, gamma1`	parameters of the supIG trawl functions of the first process
`alpha1, H1`	parameters of the long memory trawl function of the first process
`lambda21, lambda22, w2`	parameters of the (double) exponential trawl functions of the second process
`delta2, gamma2`	parameters of the supIG trawl functions of the second process
`alpha2, H2`	parameters of the long memory trawl function of the second process
`LM1`	Lebesgue measure of the first trawl
`LM2`	Lebesgue measure of the second trawl
`plotdiag`	binary variable specifying whether or not diagnostic plots should be provided

Details

Computes $R_{12}(0) = \text{Leb}(A_1 \cap A_2)$ based on two trawl functions g_1 and g_2.

Value

The Lebesgue measure of the intersection of the two trawl sets

Examples

```
#Compute the intersection of two exponential trawls
fit_trawl_intersection(fct1="Exp",fct2="Exp",lambda11=0.1,lambda21=0.5,LM1=1/0.1,LM2=1/0.5)
```

`fit_trawl_intersection_Exp` *Finds the intersection of two exponential trawl sets*

Description

Finds the intersection of two exponential trawl sets

Usage

```
fit_trawl_intersection_Exp(lambda1, lambda2, plotdiag = FALSE)
```

Arguments

`lambda1, lambda2`	parameters of the two exponential trawls
`plotdiag`	binary variable specifying whether or not diagnostic plots should be provided

Details

Computes $R_{12}(0) = \text{Leb}(A_1 \cap A_2)$ based on two trawl functions g_1 and g_2.

Value

The Lebesgue measure of the intersection of the two trawl sets

Examples

```
#Compute the intersection of two exponential trawls
fit_trawl_intersection_Exp(0.1,0.5)
```

`fit_trawl_intersection_LM` *Finds the intersection of two long memory (LM) trawl sets*

Description

Finds the intersection of two long memory (LM) trawl sets

Usage

```
fit_trawl_intersection_LM(alpha1, H1, alpha2, H2, plotdiag = FALSE)
```

Arguments

`alpha1, H1, alpha2, H2`	parameters of the two long memory trawls
`plotdiag`	binary variable specifying whether or not diagnostic plots should be provided

Details

Computes $R_{12}(0) = \text{Leb}(A_1 \cap A_2)$ based on two trawl functions g_1 and g_2.

Value

the Lebesgue measure of the intersection of the two trawl sets

Examples

```
#Compute the intersection of two long memory trawls
fit_trawl_intersection_LM(0.1,1.1,0.2,1.2)
```

`LSD_Mean` *Computes the mean of the logarithmic series distribution*

Description

Computes the mean of the logarithmic series distribution

Usage

```
LSD_Mean(p)
```

Arguments

`p`	parameter of the logarithmic series distribution

Details

A random variable X has logarithmic series distribution with parameter $0 < p < 1$ if

$$P(X = x) = (-1)/(\log(1-p))p^x/x, \text{ for } x = 1, 2, \ldots.$$

Value

Mean of the logarithmic series distribution

Examples

```
LSD_Mean(0.5)
```

LSD_Var *Computes the variance of the logarithmic series distribution*

Description

Computes the variance of the logarithmic series distribution

Usage

```
LSD_Var(p)
```

Arguments

p	parameter of the logarithmic series distribution

Details

A random variable X has logarithmic series distribution with parameter $0 < p < 1$ if

$$P(X = x) = (-1)/(\log(1 - p))p^x/x, \text{ for } x = 1, 2, \ldots.$$

Value

Variance of the logarithmic series distribution

Examples

```
LSD_Var(0.5)
```

ModLSD_Mean *Computes the mean of the modified logarithmic series distribution*

Description

Computes the mean of the modified logarithmic series distribution

Usage

```
ModLSD_Mean(delta, p)
```

Arguments

delta	parameter δ of the modified logarithmic series distribution
p	parameter of the modified logarithmic series distribution

Details

A random variable X has modified logarithmic series distribution with parameters $0 \leq \delta < 1$ and $0 < p < 1$ if $P(X = 0) = (1 - \delta)$ and

$$P(X = x) = (1 - \delta)(-1)/(\log(1 - p))p^x/x, \text{ for } x = 1, 2, \ldots.$$

Value

Mean of the modified logarithmic series distribution

Examples

```
ModLSD_Mean(0.2,0.5)
```

`ModLSD_Var` *Computes the variance of the modified logarithmic series distribution*

Description

Computes the variance of the modified logarithmic series distribution

Usage

```
ModLSD_Var(delta, p)
```

Arguments

`delta`	parameter δ of the modified logarithmic series distribution
`p`	parameter of the modified logarithmic series distribution

Details

A random variable X has modified logarithmic series distribution with parameters $0 \leq \delta < 1$ and $0 < p < 1$ if $P(X = 0) = (1 - \delta)$ and

$$P(X = x) = (1 - \delta)(-1)/(\log(1 - p))p^x/x, \text{ for } x = 1, 2, \ldots.$$

Value

Mean of the modified logarithmic series distribution

Examples

```
ModLSD_Var(0.2,0.5)
```

plot_2and1hist *Plots the bivariate histogram of two time series together with the univariate histograms*

Description

Plots the bivariate histogram of two time series together with the univariate histograms

Usage

```
plot_2and1hist(x, y)
```

Arguments

x	vector of equidistant time series data
y	vector of equidistant time series data (of the same length as x)

Details

This function plots the bivariate histogram of two time series together with the univariate histograms

Value

no return value

Examples

```
#Plot a bivariate histogram of two samples from
#independent standard normal random vectors
#Fix the seed
set.seed(1)
plot_2and1hist(stats::rnorm(100,0,1),stats::rnorm(100,0,1))
##############################
#A more interesting example:
#'#Simulate a bivariate negative binomial trawl process with exponential trawl
#functions and plot the bivariate histogram
#Parameters of the exponential trawls:
lambda1 <- 1.2
lambda2 <- 1.5
#Parameters of the negative binomial marginal law:
m1 <- 2.1
theta1 <- 0.9
a1 <- 27.3
m2 <- 2.3
theta2 <- 0.9
a2 <- 35.3
kappa12 <- m1
kappa1 <- 0
kappa2 <- m2 - kappa12
```

```
#Specify the time period and grid
t <- 720
Delta <- 1
#Fix the seed
set.seed(1)
#Simulate the bivariate trawl process with common factor
#and independent components ("dep") and negative binomial
# marginal law. Both trawl functions are chosen as exponentials.
simdata <- sim_BivariateTrawl(t, Delta, burnin=10,marginal ="NegBin",
dependencetype="dep",trawl1 ="Exp", trawl2 ="Exp",
kappa1=kappa1,kappa2=kappa2,kappa12=kappa12,a1=a1,a2=a2,lambda11=lambda1,
lambda21 =lambda2)
plot_2and1hist(simdata[,1],simdata[,2])
```

---

sim_BivariateTrawl *Simulates a bivariate trawl process*

Description

Simulates a bivariate trawl process

Usage

```
sim_BivariateTrawl(t, Delta = 1, burnin = 10,
  marginal = base::c("Poi", "NegBin"),
  dependencetype = base::c("fullydep", "dep"), trawl1 = base::c("Exp",
  "DExp", "supIG", "LM"), trawl2 = base::c("Exp", "DExp", "supIG", "LM"),
  v1 = 0, v2 = 0, v12 = 0, kappa1 = 0, kappa2 = 0, kappa12 = 0,
  a1 = 0, a2 = 0, lambda11 = 0, lambda12 = 0, w1 = 0,
  delta1 = 0, gamma1 = 0, alpha1 = 0, H1 = 0, lambda21 = 0,
  lambda22 = 0, w2 = 0, delta2 = 0, gamma2 = 0, alpha2 = 0,
  H2 = 0)
```

Arguments

t	parameter which specifying the length of the time interval $[0, t]$ for which a simulation of the trawl process is required.
Delta	parameter Δ specifying the length of the time step, the default is 1
burnin	parameter specifying the length of the burn-in period at the beginning of the simulation
marginal	parameter specifying the marginal distribution of the trawl
dependencetype	Parameter specifying the type of dependence
trawl1	parameter specifying the type of the first trawl function
trawl2	parameter specifying the type of the second trawl function
v1, v2, v12	parameters of the Poisson distribution

`kappa1, kappa2, kappa12, a1, a2`
: parameters of the (possibly bivariate) negative binomial distribution

`lambda11, lambda12, w1`
: parameters of the exponential (or double-exponential) trawl function of the first process

`delta1, gamma1`
: parameters of the supIG trawl function of the first process

`alpha1, H1`
: parameter of the long memory trawl of the first process

`lambda21, lambda22, w2`
: parameters of the exponential (or double-exponential) trawl function of the second process

`delta2, gamma2`
: parameters of the supIG trawl function of the second process

`alpha2, H2`
: parameter of the long memory trawl of the second process

Details

This function simulates a bivariate trawl process with either Poisson or negative binomial marginal law. For the trawl function there are currently four choices: exponential, double-exponential, supIG or long memory. More details on the precise simulation algorithm is available in the vignette.

Examples

```
#Simulate a bivariate negative binomial trawl process with exponential trawl
#functions
#Parameters of the exponential trawls:
lambda1 <- 1.2
lambda2 <- 1.5
#Parameters of the negative binomial marginal law:
m1 <- 2.1
theta1 <- 0.9
a1 <- 27.3
m2 <- 2.3
theta2 <- 0.9
a2 <- 35.3
kappa12 <- m1
kappa1 <- 0
kappa2 <- m2 - kappa12
#Specify the time period and grid
t <- 720
Delta <- 1
#Fix the seed
set.seed(1)
#Simulate the bivariate trawl process with common factor
#and independent components ("dep") and negative binomial
# marginal law. Both trawl functions are chosen as exponentials.
simdata <- sim_BivariateTrawl(t, Delta, burnin=10,marginal ="NegBin",
dependencetype="dep", trawl1 ="Exp", trawl2 ="Exp",
kappa1=kappa1,kappa2=kappa2,kappa12=kappa12,a1=a1,a2=a2,lambda11=lambda1,
lambda21 =lambda2)
```

sim_UnivariateTrawl *Simulates a univariate trawl process*

Description

Simulates a univariate trawl process

Usage

```
sim_UnivariateTrawl(t, Delta = 1, burnin = 10,
  marginal = base::c("Poi", "NegBin"), trawl = base::c("Exp", "DExp",
  "supIG", "LM"), v = 0, m = 0, theta = 0, lambda1 = 0,
  lambda2 = 0, w = 0, delta = 0, gamma = 0, alpha = 0, H = 0)
```

Arguments

`t`	parameter which specifying the length of the time interval $[0, t]$ for which a simulation of the trawl process is required.
`Delta`	parameter Δ specifying the length of the time step, the default is 1
`burnin`	parameter specifying the length of the burn-in period at the beginning of the simulation
`marginal`	parameter specifying the marginal distribution of the trawl
`trawl`	parameter specifying the type of trawl function
`v`	parameter of the Poisson distribution
`m`	parameter of the negative binomial distribution
`theta`	parameter θ of the negative binomial distribution
`lambda1`	parameter λ_1 of the exponential (or double-exponential) trawl function
`lambda2`	parameter λ_2 of the double-exponential trawl function
`w`	parameter of the double-exponential trawl function
`delta`	parameter δ of the supIG trawl function
`gamma`	parameter γ of the supIG trawl function
`alpha`	parameter α of the long memory trawl function
`H`	parameter of the long memory trawl function

Details

This function simulates a univariate trawl process with either Poisson or negative binomial marginal law. For the trawl function there are currently four choices: exponential, double-exponential, supIG or long memory. More details on the precise simulation algorithm is available in the vignette.

Examples

```
set.seed(1)
t <- 100
Delta <- 1
v <- 250
lambda <- 0.25
#Simulate a univariate trawl process with exponential trawl function and
#Poisson marginal law
trawl <- sim_UnivariateTrawl(t,Delta,burnin=50,marginal =c("Poi"),trawl
="Exp",v=v, lambda1=lambda)
#Plot the sample path of the simulated process
plot(trawl,type="p")
```

`trawl_DExp` *Evaluates the double exponential trawl function*

Description

Evaluates the double exponential trawl function

Usage

```
trawl_DExp(x, w, lambda1, lambda2)
```

Arguments

`x`	the argument at which the double exponential trawl function will be evaluated
`w`	parameter in the double exponential trawl
`lambda1`	the parameter λ_1 in the double exponential trawl
`lambda2`	the parameter λ_2 in the double exponential trawl

Details

The trawl function is parametrised by parameters $0 \leq w \leq 1$ and $\lambda_1, \lambda_2 > 0$ as follows:

$$g(x) = we^{\lambda_1 x} + (1-w)e^{\lambda_2 xz}, \text{ for } x \leq 0.$$

Value

The double exponential trawl function evaluated at x

Examples

```
#Evaluate the trawl function at x=-2
trawl_DExp(-2,0.4,0.1,0.9)
#Plot the trawl function
plot(trawl_DExp(-(0:10),0.4,0.1,0.9))
```

trawl_Exp *Evaluates the exponential trawl function*

Description

Evaluates the exponential trawl function

Usage

```
trawl_Exp(x, lambda)
```

Arguments

x	the argument at which the exponential trawl function will be evaluated
lambda	the parameter λ in the exponential trawl

Details

The trawl function is parametrised by parameter $\lambda > 0$ as follows:

$$g(x) = e^{\lambda x}, \text{ for } x \leq 0.$$

Value

The exponential trawl function evaluated at x

Examples

```
#Evaluate the trawl function at x=-2
trawl_Exp(-2,0.4)
#Plot the trawl function
plot(trawl_Exp(-(0:10),0.4))
```

trawl_LM *Evaluates the long memory trawl function*

Description

Evaluates the long memory trawl function

Usage

```
trawl_LM(x, alpha, H)
```

Arguments

`x`	the argument at which the long memory trawl function will be evaluated
`alpha`	the parameter α in the long memory trawl
`H`	the parameter H in the long memory trawl

Details

The trawl function is parametrised by the two parameters $H > 1$ and $\alpha > 0$ as follows:

$$g(x) = (1 - x/\alpha)^{-H}, \text{ for } x \leq 0.$$

Value

the long memory trawl function evaluated at x

Examples

```
#Evaluate the trawl function at x=-2
trawl_LM(-2,0.9,1.5)
#Plot the trawl function
plot(trawl_LM(-(0:10),0.9,1.5))
```

`trawl_supIG` *Evaluates the supIG trawl function*

Description

Evaluates the supIG trawl function

Usage

```
trawl_supIG(x, delta, gamma)
```

Arguments

`x`	the argument at which the supIG trawl function will be evaluated
`delta`	the parameter δ in the supIG trawl
`gamma`	the parameter γ in the supIG trawl

Details

The trawl function is parametrised by the two parameters $\delta \geq 0$ and $\gamma \geq 0$ as follows:

$$gd(x) = (1 - 2x\gamma^{-2})^{-1/2} \exp(\delta\gamma(1 - (1 - 2x\gamma^{-2})^{1/2})), \text{ for } x \leq 0.$$

It is assumed that δ and γ are not simultaneously equal to zero.

Value

The supIG trawl function evaluated at x

Examples

```
#Evaluate the trawl function at x=-2
trawl_supIG(-2,0.4,0.8)
#Plot the trawl function
plot(trawl_supIG(-(0:10),0.4,0.8))
```

Trivariate_LSDsim *Simulates from the trivariate logarithmic series distribution*

Description

Simulates from the trivariate logarithmic series distribution

Usage

```
Trivariate_LSDsim(N, p1, p2, p3)
```

Arguments

N	number of data points to be simulated
p1	parameter $p1$ of the trivariate logarithmic series distribution
p2	parameter $p2$ of the trivariate logarithmic series distribution
p3	parameter $p3$ of the trivariate logarithmic series distribution

Details

The probability mass function of a random vector $X = (X_1, X_2, X_3)'$ following the trivariate logarithmic series distribution with parameters $0 < p_1, p_2, p_3 < 1$ with $p := p_1 + p_2 + p_3 < 1$ is given by

$$P(X_1 = x_1, X_2 = x_2, X_3 = x_3) = \frac{\Gamma(x_1 + x_2 + x_3)}{x_1!x_2!x_3!} \frac{p_1^{x_1} p_2^{x_2} p_3^{x_3}}{(-\log(1-p))},$$

for $x_1, x_2, x_3 = 0, 1, 2, \ldots$ such that $x_1 + x_2 + x_3 > 0$.

The simulation proceeds in two steps: First, X_1 is simulated from the modified logarithmic distribution with parameters $\tilde{p}_1 = p_1/(1 - p_2 - p_3)$ and $\delta_1 = \log(1 - p_2 - p_3)/\log(1 - p)$. Then we simulate $(X_2, X_3)'$ conditional on X_1. We note that $(X_2, X_3)'|X_1 = x_1$ follows the bivariate logarithmic series distribution with parameters (p_2, p_3) when $x_1 = 0$, and the bivariate negative binomial distribution with parameters (x_1, p_2, p_3) when $x_1 > 0$.

Value

An $N \times 3$ matrix with N simulated values from the trivariate logarithmic series distribution

Examples

```
set.seed(1)
p1 <- 0.15
p2 <- 0.25
p3 <- 0.55
N <- 100
#Simulate N realisations from the bivariate LSD
y <- Trivariate_LSDsim(N, p1, p2, p3)
```

Index