Package ‘trawl’

October 14, 2022

Type Package

Title Estimation and Simulation of Trawl Processes

Version 0.2.2

Author Almut E. D. Veraart

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


License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends R (>= 4.0.0)

Imports DEoptim, ggplot2, ggpubr, graphics, MASS, rootSolve, Runuran, stats, squash, TSA

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2021-02-22 17:30:02 UTC

R topics documented:

  acf_DExp ......................................................... 2
  acf_Exp ......................................................... 3
  acf_LM .......................................................... 4
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:

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

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

Value

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

Examples

```r
acf_DExp(1, 0.3, 0.1, 2)
```

Description

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

Usage

```r
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

```r
acf_Exp(1, 0.1)
```

Description

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

Usage

```r
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 $x$
acf_supIG

Examples

acf.lm(1, 0.3, 1.5)

\[
acf_{\text{supIG}}
\]

\begin{tabular}{ll}
\hline
acf_{\text{supIG}} & \textit{Autocorrelation function of the supIG trawl function} \\
\hline
\end{tabular}

Description

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

Usage

acf_supIG(x, delta, gamma)

Arguments

\begin{itemize}
\item \textbf{x} \hspace{1cm} The argument (lag) at which the autocorrelation function associated with the supIG trawl function will be evaluated
\item \textbf{delta} \hspace{1cm} parameter in the supIG trawl
\item \textbf{gamma} \hspace{1cm} parameter in the supIG trawl
\end{itemize}

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 \( \delta \) and \( \gamma \) 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

acf_supIG(1, 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

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

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

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 $\kappa$ 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 $\kappa, p_1, p_2$ 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

---

**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
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

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 \( \delta \) 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
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 \( \delta \) 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

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 $\lambda_1$ above)
- lambda2: the second memory parameter (denoted by $\lambda_2$ above)
- LM: The Lebesgue measure of the trawl set associated with the double exponential trawl

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

fits an exponential trawl function to equidistant time series data
**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**

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

**Description**

Fist a negative binomial distribution as marginal law

**Usage**

\[
\text{fit\_marginalNB}(x, \ LM, \ \text{plotdiag} = \FALSE)
\]

**Arguments**

- \text{x}: vector of equidistant time series data
- \text{LM}: Lebesgue measure of the estimated trawl
- \text{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

\[
\hat{\theta} = 1 - \text{E}(X)/\text{Var}(X),
\]

and

\[
\hat{m} = \text{E}(X)(1 - \hat{\theta})/(\hat{LM}\hat{\theta}).
\]

**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

**fit_marginalPoisson**  
*Fits a Poisson distribution as marginal law*

**Description**

Fits a Poisson distribution as marginal law

**Usage**

\[
\text{fit\_marginalPoisson}(x, \ LM, \ \text{plotdiag} = \FALSE)
\]

**Arguments**

- \text{x}: vector of equidistant time series data
- \text{LM}: Lebesgue measure of the estimated trawl
- \text{plotdiag}: binary variable specifying whether or not diagnostic plots should be provided
**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

\[ \hat{\nu} = \frac{E(X)}{LM}. \]

**Value**

- **v**: the rate parameter in the Poisson marginal distribution

---

**Function: 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:

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

It is assumed that \( \delta \) and \( \gamma \) are not simultaneously equal to zero. The Lebesgue measure of the corresponding trawl set is given by \( \gamma/\delta \).

**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
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
fit_trawl_intersection_Exp

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

Description
Finds the intersection of two exponential trawl sets

Usage
fit_trawl_intersection_Exp(lambda1, lambda2, LM1, LM2, plotdiag = FALSE)

Arguments
lambda1, lambda2  parameters of the two exponential trawls
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
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**

```r
fit_trawl_intersection_LM(alpha1, H1, alpha2, H2, LM1, LM2, plotdiag = FALSE)
```

**Arguments**

- `alpha1, H1, alpha2, H2` parameters of the two long memory trawls
- `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

---

LSD_Mean

*Computes the mean of the logarithmic series distribution*

**Description**
Computes the mean of the logarithmic series distribution

**Usage**

```r
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

<table>
<thead>
<tr>
<th>LSD_Var</th>
<th>Computes the variance of the logarithmic series distribution</th>
</tr>
</thead>
</table>

Description

Computes the variance of the logarithmic series distribution

Usage

LSD_Var(p)

Arguments

$\text{p}$ 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

<table>
<thead>
<tr>
<th>ModLSD_Mean</th>
<th>Computes the mean of the modified logarithmic series distribution</th>
</tr>
</thead>
</table>

Description

Computes the mean of the modified logarithmic series distribution

Usage

ModLSD_Mean(delta, p)
Arguments

delta parameter $\delta$ 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

---

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 $\delta$ 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
**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

---

**plot_2and1hist_gg**

*Plots the bivariate histogram of two time series together with the univariate histograms using ggplot2*

**Description**

Plots the bivariate histogram of two time series together with the univariate histograms using ggplot2

**Usage**

plot_2and1hist_gg(x, y, bivbins = 50, xbins = 30, ybins = 30)

**Arguments**

- `x`: vector of equidistant time series data
- `y`: vector of equidistant time series data (of the same length as `x`)
- `bivbins`: number of bins in the bivariate histogram
- `xbins`: number of bins in the histogram of `x`
- `ybins`: number of bins in the histogram of `y`
Details

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

Value

no return value

---

**sim_BivariateTrawl**  
*Simulates a bivariate trawl process*

---

Description

Simulates a bivariate trawl process

Usage

```r
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)
```
**sim_UnivariateTrawl**

**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 \(\Delta\) 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.

**Description**

Simulates a univariate trawl process
Usage

```r
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 $\Delta$ 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 $\theta$ of the negative binomial distribution

lambda1 parameter $\lambda_1$ of the exponential (or double-exponential) trawl function

lambda2 parameter $\lambda_2$ of the double-exponential trawl function

w parameter of the double-exponential trawl function

delta parameter $\delta$ of the supIG trawl function

gamma parameter $\gamma$ of the supIG trawl function

alpha parameter $\alpha$ 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.
**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 $\lambda_1$ in the double exponential trawl
- `lambda2`: the parameter $\lambda_2$ 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:

\[
g(x) = we^{\lambda_1x} + (1 - w)e^{\lambda_2xz}, \text{ for } x \leq 0.
\]

**Value**

The double exponential trawl function evaluated at $x$

---

**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 $\lambda$ 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 $x$

---

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 $\alpha$ 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 $x$
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 \( \delta \) in the supIG trawl
gamma the parameter \( \gamma \) in the supIG trawl

Details
The trawl function is parametrised by the two parameters \( \delta \geq 0 \) and \( \gamma \geq 0 \) as follows:
\[
gd(x) = (1 - 2x\gamma^{-2})^{-1/2} \exp(\delta \gamma(1 - (1 - 2x\gamma^{-2})^{1/2})), \quad \text{for } x \leq 0.
\]
It is assumed that \( \delta \) and \( \gamma \) are not simultaneously equal to zero.

Value
The supIG trawl function evaluated at \( x \)

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 \( p_1 \) of the trivariate logarithmic series distribution
p2 parameter \( p_2 \) of the trivariate logarithmic series distribution
p3 parameter \( p_3 \) of the trivariate logarithmic series distribution
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.
Index

acf_DExp, 2
acf_Exp, 3
acf_LM, 4
acf_supIG, 5

Bivariate_LSDsim, 6
Bivariate_NBsim, 6
BivLSD_Cor, 7
BivLSD_Cov, 8
BivModLSD_Cor, 8
BivModLSD_Cov, 9

fit_DExptrawl, 9
fit_Exptrawl, 10
fit_LMtrawl, 11
fit_marginalNB, 12
fit_marginalPoisson, 12
fit_supIGtrawl, 13
fit_trawl_intersection, 14
fit_trawl_intersection_Exp, 15
fit_trawl_intersection_LM, 16

LSD_Mean, 16
LSD_Var, 17

ModLSD_Mean, 17
ModLSD_Var, 18

plot_2and1hist, 19
plot_2and1hist_gg, 19

sim_BivariateTrawl, 20
sim_UnivariateTrawl, 21

trawl_DExp, 23
trawl_Exp, 23
trawl_LM, 24
trawl_supIG, 25
Trivariate_LSDsim, 25