Package 'ev.trawl'

December 3, 2018

Type Package

Title Extreme Value Trawls

Version 0.1.0

Author Valentin Courgeau

Maintainer Valentin Courgeau <valentin.courgeau15@imperial.ac.uk>

Description Implementation of trawl processes and an extension of such processes into a univariate latent model for extreme values.
Inference, simulation and initialization tools are available. See Noven et al. (2018) <DOI:10.21314/JEM.2018.179> which can be found on arXiv (<arXiv:1511.08190>) .

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

Depends R (>= 2.10)

Imports testthat, fExtremes, hypergeo, stats, ghyp, evir, eva

NeedsCompilation no

Repository CRAN

Date/Publication 2018-12-03 19:40:08 UTC

R topics documented:

CollectionTrawl

Creates a list of trawl functions given a type of trawl and a vector of times as peak times. Has the option to get primitives. Note that only exponential trawl are implemented so far.

Description

Creates a list of trawl functions given a type of trawl and a vector of times as peak times. Has the option to get primitives. Note that only exponential trawl are implemented so far.

Usage

```
CollectionTrawl(times, params, type, prim = F)
```

Arguments

times	Vector of times to evaluate the trawl function (or primitive) at.
params	List of trawl parameters.
type	Trawl type (so far, only "exp" is available).
prim	Boolean to use primitive or not. Default is False (F).

Value

Collection of trawl functions set on `times` given the type of trawl (`type`).

ComputeAExp

Wrapper to compute total area under exponential trawl function.

Description

Wrapper to compute total area under exponential trawl function.

Usage

```
ComputeAExp(rho)
```

Arguments

rho	Exponential trawl parameter (should be non-negative).

Value

Total area under exponential trawl function (1/rho).

Examples

```
ComputeAExp(1) # should be 1.0
ComputeAExp(0.2) # should be 5.0.
```

`ComputeB1Exp` *Wrapper to compute difference area between* `t2` *and* `t1` *under exponential trawl function.*

Description

Wrapper to compute difference area between `t2` and `t1` under exponential trawl function.

Usage

```
ComputeB1Exp(rho, t1, t2)
```

Arguments

`rho`	Exponential trawl parameter (should be non-negative).
`t1`	First timestamp.
`t2`	Second timestamp.

Value

Difference area between `t1` and `t2` under exponential trawl function. Relies on ComputeB3Exp.

Examples

```
ComputeB1Exp(1, t1 = 3, t2 = 5)
ComputeB1Exp(0.2, t1 = 7, t2 = 3)
```

`ComputeB3Exp` *Wrapper to compute difference area between* `t1` *and* `t2` *under exponential trawl function.*

Description

Wrapper to compute difference area between `t1` and `t2` under exponential trawl function.

Usage

```
ComputeB3Exp(rho, t1, t2)
```

Arguments

`rho`	Exponential trawl parameter (should be non-negative).
`t1`	First timestamp.
`t2`	Second timestamp.

Value

Difference area between t1 and t2 under exponential trawl function.

Examples

```
ComputeB3Exp(1, t1 = 3, t2 = 5)
ComputeB3Exp(0.2, t1 = 7, t2 = 3)
```

ComputeBInterExp — *Wrapper to compute intersection area between* t1 *and* t2 *under exponential trawl function.*

Description

Wrapper to compute intersection area between t1 and t2 under exponential trawl function.

Usage

```
ComputeBInterExp(rho, t1, t2)
```

Arguments

rho	Exponential trawl parameter (should be non-negative).
t1	First timestamp.
t2	Second timestamp.

Value

Intersection area between t1 and t2 under exponential trawl function.

Examples

```
ComputeBInterExp(1, t1 = 3, t2 = 5)
ComputeBInterExp(0.2, t1 = 7, t2 = 3)
```

DiffVal *Centered differentiation calculation.*

Description

Centered differentiation calculation.

Usage

```
DiffVal(value.p, value.m, epsilon)
```

Arguments

value.p	function value at x+epsilon.
value.m	function value at x-epsilon.
epsilon	offset value for x.

Value

(value.p - value.m) / (2*epsilon).

dlgpd *Computes the probability density function of Generalised Pareto Distribution at x, with shape parameter alpha and scale parameter beta.*

Description

Computes the probability density function of Generalised Pareto Distribution at x, with shape parameter alpha and scale parameter beta.

Usage

```
dlgpd(x, alpha, beta)
```

Arguments

x	value at which the pdf is evaluated.
alpha	Shape parameter.
beta	Scale parameter, should be positive.

Value

GPD pdf function evaluated at x with shape and scale parameters respectively alpha and beta.

Examples

```
dlgpd(2.34, alpha = 1, beta = 2)
```

`EvaluateF` *Evalutae function at given parameters.*

Description

Evalutae function at given parameters.

Usage

```
EvaluateF(f, params, epsilon)
```

Arguments

`f`	R function taking only a vector of parameters as input.
`params`	Vector of parameters.
`epsilon`	Offset value to add to all component of `params`.

Value

Evaluate of function `f` at parameters `params` with offset `epsilon`.

`FullPL` *Computes latent trawl FULL pairwise likelihood depending with exponential trawl function.*

Description

Computes latent trawl FULL pairwise likelihood depending with exponential trawl function.

Usage

```
FullPL(times, values, alpha, beta, kappa, rho, delta, logscale = T,
  transformation = F, trawl.function = "exp")
```

Arguments

`times`	Vector of timestamps.
`values`	Vector of target values.
`alpha`	Shape parameter. Should be positive.
`beta`	Latent Gamma scale parameter. Should be positive.
`kappa`	Exceedance probability parameter. Should be positive.
`rho`	Trawl parameter(s). For `trawl.function="exp"`, it should be positive.
`delta`	Maximum depth of pairwise likelihood blocks. Should be positive natural integer.

logscale	Boolean to use logscale (log-likelihood). Default T.
transformation	Boolean to use the Marginal Transform (MT) method.
trawl.function	Type of trawl function that should be used. Default NA.

Value

Full latent trawl pairwise likelihood.

Examples

```
FullPL(times=1:10, values = seq(from=0.1, to=5, by=0.5), alpha=0.3,
  beta=2, kappa=3, rho=0.2, delta=2, T, F, "exp")
```

GammaBox *Sample Gamma with small shape parameter* alpha*dx*dy.

Description

Sample Gamma with small shape parameter alpha*dx*dy.

Usage

```
GammaBox(alpha, beta, dx, dy, n)
```

Arguments

alpha	Shape parameter.
beta	Scale parameter.
dx	x-axis multiplier.
dy	y-axis multiplier.
n	Sample size.

Value

n Gamma(alpha*dx*dy, beta) samples.

`GammaSqBox` *Wrapper for GammaBox using a square to multiply the shape parameter.*

Description

Wrapper for GammaBox using a square to multiply the shape parameter.

Usage

```
GammaSqBox(alpha, beta, dt, n)
```

Arguments

`alpha`	Shape parameter.
`beta`	Scale parameter.
`dt`	Both x-axis and y-axis multiplier.
`n`	Sample size.

Value

n Gamma(alpha*dt*dt, beta) samples.

`GenerateParameters` *Computes initial guess for Univariate Latent-Trawl model. Uses* `fExtreme` *package method* `gpdFit` *to fit MLE.*

Description

Computes initial guess for Univariate Latent-Trawl model. Uses `fExtreme` package method `gpdFit` to fit MLE.

Usage

```
GenerateParameters(data, cluster.size)
```

Arguments

`data`	Exceedane timeseries to be used.
`cluster.size`	Lag at which ACF is negligeable.

Examples

```
# TODO ADD data
data(hourlyhourly_bloomsbury_air_pollution_2000_2017)
GenerateParameters(data = hourly_bloomsbury_air_pollution_2000_2017$O3[1:1000], cluster.size = 8)
```

GetEstimateRho — *Computes first-order estimate for* rho.

Description

Computes first-order estimate for rho.

Usage

```
GetEstimateRho(alpha, beta, kappa, cluster.size, data)
```

Arguments

alpha	Shape parameter.
beta	Scale parameter. Must be positive.
kappa	Exceedance probability parameter. Must be positive.
cluster.size	Lag at which ACF is negligeable i.e. cluster size.
data	Exceedane timeseries to be used.

Examples

```
# TODO ADD data
GetEstimateRho(alpha = 5, beta = 2, kappa = 3, cluster.size = 8, data = NA)
```

GPDFit — *Generalised Pareto likelihood maximisation using L-BFGS-B optimisation routine.*

Description

Generalised Pareto likelihood maximisation using L-BFGS-B optimisation routine.

Usage

```
GPDFit(values, initial_guess, lower = c(0.1, 0.1), upper = c(20, 20))
```

Arguments

values	Exceedance values.
initial_guess	(at least 2-d) Vector for GPD parameters starting values.
lower	Vector of lower bounds limits for optimisation procedure. Default c(0.1, 0.1).
upper	Vector of upper bounds limits for optimisation procedure. Default c(20, 20).

Value

Parameters of Log-likelihood maximisation of GPD distribued variables (i.e. non-zero exceedances).

Examples

```
GPDFit(c(2.0, 0.3, 6.15, 0, 0.31), c(2.1, 1.17, 0.52, 4.17))
```

`GradF` *Computes gradident of a function given parameters and offset value epsilon.*

Description

Computes gradident of a function given parameters and offset value epsilon.

Usage

```
GradF(f, params, epsilon = 1e-06)
```

Arguments

`f`	R function taking only a vector of parameter as input.
`params`	Vector of parameters.
`epsilon`	Offset value for all components.

Value

Gradient of function `f` at parameters `params` with offset `epsilon`.

`HessianF` *Computes Hessian matrix of a C^2 function given parameters and offset value epsilon*

Description

Computes Hessian matrix of a C^2 function given parameters and offset value epsilon

Usage

```
HessianF(f, params, epsilon = 1e-06)
```

Arguments

f	R function taking only a vector of parameter as input.
params	Vector of parameters.
epsilon	Offset value for all components.

Value

Hessian matrix of function f at parameters params with offset epsilon.

---

hourly_bloomsbury_air_pollution_2000_2017

Hourly measurements of 6 air pollutant concentrations in Bloomsbury, London, UK.

Description

A dataset containing the concentration of 6 pollutants in Bloomsbury, London, UK with hourly measurements from 1st January 2000 to 3st December 2017. Courtesy of King's College Air Pollution monitoring network. Values expressed in micrograms per cubic meter.

Usage

```
data(hourly_bloomsbury_air_pollution_2000_2017)
```

Format

A data frame with 157710 rows and 9 variables:

index index of entry

date date of entry in format dd/mm/yyyy

time time of day of entry in format HH:MM

O3 concentration in Ozone, in mu g / m^3

CO concentration in Carbone Monoxide, in mu g / m^3

NO concentration in Nitrogen Oxide, in mu g / m^3

NO2 concentration in Nitrogen Dioxyde, in mu g / m^3

PM10 concentration in particulate matter 10 micrometers or less in diameter, in mu g / m^3

SO2 concentration in Sulfur Dioxide, in mu g / m^3

Source

https://www.londonair.org.uk/ data selection tool.

Examples

```
data(hourly_bloomsbury_air_pollution_2000_2017)
plot(hourly_bloomsbury_air_pollution_2000_2017$O3)
```

MarginalGPDLikelihood *Computes Generalised Pareto (log-)likelihood on non-zero exceedances under independence.*

Description

Computes Generalised Pareto (log-)likelihood on non-zero exceedances under independence.

Usage

```
MarginalGPDLikelihood(values, fixed_names, fixed_params, params,
  model_vars_names, logscale = T, transformation = F, n.moments = 4)
```

Arguments

values	Vector of target values.
fixed_names	Vector of literal names of parameters to keep fixed.
fixed_params	Vector of numerical values of fixed parameters.
params	List of parameters.
model_vars_names	Vector of all parameters names in the model.
logscale	Logical; Default (TRUE) is to use logscale (log-likelihood).
transformation	Boolean to use the Marginal Transform (MT) method.
n.moments	Number of moments the transformed variables should have using the Marginal Transform (MT) method.

Value

Generalised Pareto (log-)likelihood on non-zero exceedances under independence.

Examples

```
times <- c(1,2,3,4,5)
values <- c(2,0,3,4,0)
delta <- 2
fixed_names <- c("alpha", "kappa")
params <- c(3.4, 0.1)
fixed_params <- c(2.0, 4.3)
model_vars_names <- c("alpha", "beta", "rho", "kappa")
MarginalGPDLikelihood(values, fixed_names, fixed_params, params, model_vars_names)
```

`MarginalSimpleLik` *Simplified marginal (GPD) log-likelihood function under independence in the exponential trawl case.*

Description

Simplified marginal (GPD) log-likelihood function under independence in the exponential trawl case.

Usage

```
MarginalSimpleLik(values, params)
```

Arguments

`values`	Exceedance values.
`params`	(at least) 2-d vector for GPD parameters

Value

Log-likelihood of GPD distribued variables (i.e. non-zero exceedances).

Examples

```
MarginalSimpleLik(c(2.0, 0.3, 6.15, 0, 0.31), c(2.1, 1.17, 0.52, 4.17)) # for GPD(2.1, 1.17)
```

`MoMGPD` *Method of moments (MoM) on a one or multiple exceedance timeseries for the latent trawl model using marginal GPD properties and probability of exceedance.*

Description

Method of moments (MoM) on a one or multiple exceedance timeseries for the latent trawl model using marginal GPD properties and probability of exceedance.

Usage

```
MoMGPD(values_array)
```

Arguments

`values_array`	Matrix of exceedance timeseries

Value

Parameters given by a second-order method of moments as well as standard deviation across time-series for each individual parameter.

Examples

```
exceed1 <- c(0.1, 0, 0.2, 0, 0, 0, 0.6, 1.5)
exceed2 <- c(0, 0.3, 5.2, 0, 0, 3.0, 0, 2.2)
val_array <- cbind(exceed1, exceed2)
MoMGPD(val_array)
```

PairwiseOneOne *Computes latent trawl pairwise likelihood with* (x,x) *where* x *and* y *positive with exponential trawl function.*

Description

Computes latent trawl pairwise likelihood with (x,x) where x and y positive with exponential trawl function.

Usage

```
PairwiseOneOne(x1, x2, alpha, beta, kappa, B1, B2, B3,
  transformation = F, n.moments = 4)
```

Arguments

x1	Positive value corresponding to t1.
x2	Positive value corresponding to t2.
alpha	Shape parameter. Should be positive.
beta	Latent Gamma scale parameter. Should be positive.
kappa	Exceedance probability parameter. Should be positive.
B1	Difference area between t1 and t2 (in this order).
B2	Intersection area between t1 and t2 (in this order).
B3	Difference area between t2 and t1 (in this order).
transformation	Boolean to use the Marginal Transform (MT) method.
n.moments	Number of moments achieved by transformed GPD marginals, if used.

Value

Latent trawl pairwise likelihood with (x,x) where x positive and y positive.

Examples

```
PairwiseOneOne(x1=0.5, x2=0.3, alpha=2, beta=3, B1 = 0.3, B2 = 0.7,
                        B3 = 0.3, kappa=3.5)
```

PairwiseOneOne1 *Computes first term in latent trawl pairwise likelihood with* (x,x) *where* x *positive and* y *positive.*

Description

Computes first term in latent trawl pairwise likelihood with (x,x) where x positive and y positive.

Usage

```
PairwiseOneOne1(x1, x2, alpha, beta, kappa, B1, B2, B3)
```

Arguments

x1	Positive value corresponding to t1.
x2	Positive value corresponding to t2.
alpha	Shape parameter. Should be positive.
beta	Latent Gamma scale parameter. Should be positive.
kappa	Exceedance probability parameter. Should be positive.
B1	Difference area between t1 and t2 (in this order).
B2	Intersection area between t1 and t2 (in this order).
B3	Difference area between t2 and t1 (in this order).

Value

First term in latent trawl pairwise likelihood with (x,x) where x positive and y positive.

PairwiseOneOne11 *Computes frst part of first term in latent trawl pairwise likelihood with* (x,x) *where* x *positive and* y *positive.*

Description

Computes frst part of first term in latent trawl pairwise likelihood with (x,x) where x positive and y positive.

Usage

```
PairwiseOneOne11(x1, x2, alpha, beta, kappa, B1, trawlA)
```

Arguments

x1	Positive value corresponding to t1.
x2	Positive value corresponding to t2.
alpha	Shape parameter. Should be positive.
beta	Latent Gamma scale parameter. Should be positive.
kappa	Exceedance probability parameter. Should be positive.
B1	Difference area between t1 and t2 (in this order).
trawlA	Total trawl set measure / area under trawl function.

Value

Frst part of first term in latent trawl pairwise likelihood with (x,x) where x positive and y positive.

PairwiseOneOne12 *Computes second part of first term in latent trawl pairwise likelihood with* (x,x) *where* x *positive and* y *positive.*

Description

Computes second part of first term in latent trawl pairwise likelihood with (x,x) where x positive and y positive.

Usage

```
PairwiseOneOne12(x1, x2, alpha, beta, kappa, B2, trawlA)
```

Arguments

x1	Positive value corresponding to t1.
x2	Positive value corresponding to t2.
alpha	Shape parameter. Should be positive.
beta	Latent Gamma scale parameter. Should be positive.
kappa	Exceedance probability parameter. Should be positive.
B2	Intersection area between t1 and t2 (in this order).
trawlA	Total trawl set measure / area under trawl function.

Value

Second part of first term in latent trawl pairwise likelihood with (x,x) where x positive and y positive.

PairwiseOneOne13 *Computes third part of first term in latent trawl pairwise likelihood with* (x,x) *where* x *positive and* y *positive.*

Description

Computes third part of first term in latent trawl pairwise likelihood with (x,x) where x positive and y positive.

Usage

```
PairwiseOneOne13(x1, x2, alpha, beta, kappa, B3, trawlA)
```

Arguments

x1	Positive value corresponding to t1.
x2	Positive value corresponding to t2.
alpha	Shape parameter. Should be positive.
beta	Latent Gamma scale parameter. Should be positive.
kappa	Exceedance probability parameter. Should be positive.
B3	Difference area between t2 and t1 (in this order).
trawlA	Total trawl set measure / area under trawl function.

Value

Second part of first term in latent trawl pairwise likelihood with (x,x) where x positive and y positive.

PairwiseOneOne2 *Computes second term in latent trawl pairwise likelihood with* (x,x) *where* x *positive and* y *positive.*

Description

Computes second term in latent trawl pairwise likelihood with (x,x) where x positive and y positive.

Usage

```
PairwiseOneOne2(x1, x2, alpha, beta, kappa, B1, B2, B3)
```

Arguments

x1	Positive value corresponding to t1.
x2	Positive value corresponding to t2.
alpha	Shape parameter. Should be positive.
beta	Latent Gamma scale parameter. Should be positive.
kappa	Exceedance probability parameter. Should be positive.
B1	Difference area between t1 and t2 (in this order).
B2	Intersection area between t1 and t2 (in this order).
B3	Difference area between t2 and t1 (in this order).

Value

Second term in latent trawl pairwise likelihood with (x,x) where x positive and y positive.

PairwiseOneOne21 *Computes second term in latent trawl pairwise likelihood with* (x,x) *where* x *positive and* y *positive.*

Description

Computes second term in latent trawl pairwise likelihood with (x,x) where x positive and y positive.

Usage

```
PairwiseOneOne21(x1, x2, alpha, beta, kappa, B1, B2)
```

Arguments

x1	Positive value corresponding to t1.
x2	Positive value corresponding to t2.
alpha	Shape parameter. Should be positive.
beta	Latent Gamma scale parameter. Should be positive.
kappa	Exceedance probability parameter. Should be positive.
B1	Difference area between t1 and t2 (in this order).
B2	Intersection area between t1 and t2 (in this order).

Value

Second term in latent trawl pairwise likelihood with (x,x) where x positive and y positive.

PairwiseOneOne22 *Computes second term in latent trawl pairwise likelihood with* (x,x) *where* x *positive and* y *positive.*

Description

Computes second term in latent trawl pairwise likelihood with (x,x) where x positive and y positive.

Usage

```
PairwiseOneOne22(x1, x2, alpha, beta, kappa, B1, B3)
```

Arguments

x1	Positive value corresponding to t1.
x2	Positive value corresponding to t2.
alpha	Shape parameter. Should be positive.
beta	Latent Gamma scale parameter. Should be positive.
kappa	Exceedance probability parameter. Should be positive.
B1	Difference area between t1 and t2 (in this order).
B3	Difference area between t2 and t1 (in this order).

Value

Second term in latent trawl pairwise likelihood with (x,x) where x positive and y positive.

PairwiseOneOne23 *Computes second term in latent trawl pairwise likelihood with* (x,x) *where* x *positive and* y *positive.*

Description

Computes second term in latent trawl pairwise likelihood with (x,x) where x positive and y positive.

Usage

```
PairwiseOneOne23(x1, x2, alpha, beta, kappa, B2, trawlA)
```

Arguments

x1	Positive value corresponding to t1.
x2	Positive value corresponding to t2.
alpha	Shape parameter. Should be positive.
beta	Latent Gamma scale parameter. Should be positive.
kappa	Exceedance probability parameter. Should be positive.
B2	Intersection area between t1 and t2 (in this order).
trawlA	Total trawl set measure / area under trawl function.

Value

Second term in latent trawl pairwise likelihood with (x,x) where x positive and y positive.

PairwiseOneOne24 *Computes second term in latent trawl pairwise likelihood with* (x,x) *where* x *positive and* y *positive.*

Description

Computes second term in latent trawl pairwise likelihood with (x,x) where x positive and y positive.

Usage

```
PairwiseOneOne24(x1, x2, alpha, beta, kappa, B2, B3)
```

Arguments

x1	Positive value corresponding to t1.
x2	Positive value corresponding to t2.
alpha	Shape parameter. Should be positive.
beta	Latent Gamma scale parameter. Should be positive.
kappa	Exceedance probability parameter. Should be positive.
B2	Intersection area between t1 and t2 (in this order).
B3	Difference area between t2 and t1 (in this order).

Value

Second term in latent trawl pairwise likelihood with (x,x) where x positive and y positive.

PairwiseOneZero *Computes term in latent trawl pairwise likelihood with* (x,0) *where* x *positive with exponential trawl function.*

Description

Computes term in latent trawl pairwise likelihood with (x,0) where x positive with exponential trawl function.

Usage

```
PairwiseOneZero(x1, alpha, beta, kappa, B1, B2, B3, transformation = F,
  n.moments = 4)
```

Arguments

x1	Positive value corresponding to t1.
alpha	Shape parameter. Should be positive.
beta	Latent Gamma scale parameter. Should be positive.
kappa	Exceedance probability parameter. Should be positive.
B1	Difference area between t1 and t2 (in this order).
B2	intersection area between t1 and t2 (in this order).
B3	Difference area between t2 and t1 (in this order).
transformation	Boolean to use the Marginal Transform (MT) method.
n.moments	Number of moments achieved by transformed GPD marginals, if used.

Value

Second term in latent trawl pairwise likelihood with (x,0) where x positive with exponential trawl function.

Examples

```
PairwiseOneZero(x1=0.5, alpha=0.3, beta=2, kappa=3, B1=0.3, B2=0.7, B3=0.3)
```

PairwiseOneZero1 — *Computes first part of latent trawl pairwise likelihood with* `(x,0)` *where* `x` *positive.*

Description

Computes first part of latent trawl pairwise likelihood with `(x,0)` where `x` positive.

Usage

```
PairwiseOneZero1(x1, alpha, beta, kappa, trawlA)
```

Arguments

`x1`	Positive value corresponding to `t1`.
`alpha`	Shape parameter. Should be positive.
`beta`	Latent Gamma scale parameter. Should be positive.
`kappa`	Exceedance probability parameter. Should be positive.
`trawlA`	Total trawl set measure / area under trawl function.

Value

first part of latent trawl pairwise likelihood with `(x,0)` where `x` positive.

PairwiseOneZero2 — *Computes second term in latent trawl pairwise likelihood with* `(x,0)` *where* `x` *positive.*

Description

Computes second term in latent trawl pairwise likelihood with `(x,0)` where `x` positive.

Usage

```
PairwiseOneZero2(x1, alpha, beta, kappa, B1, B2, B3, trawlA)
```

Arguments

`x1`	Positive value corresponding to `t1`.
`alpha`	Shape parameter. Should be positive.
`beta`	Latent Gamma scale parameter. Should be positive.
`kappa`	Exceedance probability parameter. Should be positive.
`B1`	Difference area between `t1` and `t2` (in this order).
`B2`	intersection area between `t1` and `t2` (in this order).
`B3`	Difference area between `t2` and `t1` (in this order).
`trawlA`	Total trawl set measure / area under trawl function.

Value

Second term in latent trawl pairwise likelihood with `(x,0)` where `x` positive.

`PairwiseOneZero21` *Computes partial part of second term in latent trawl pairwise likelihood with* `(x,0)` *where* `x` *positive.*

Description

Computes partial part of second term in latent trawl pairwise likelihood with `(x,0)` where `x` positive.

Usage

```
PairwiseOneZero21(x1, alpha, beta, kappa, B1, trawlA)
```

Arguments

`x1`	Positive value corresponding to `t1`.
`alpha`	Shape parameter. Should be positive.
`beta`	Latent Gamma scale parameter. Should be positive.
`kappa`	Exceedance probability parameter. Should be positive.
`B1`	Difference area between `t2` and `t1` (in this order).
`trawlA`	Total trawl set measure / area under trawl function.

Value

Partial part of second term in latent trawl pairwise likelihood with `(x,0)` where `x` positive.

`PairwiseOneZero22` *Computes partial part of second term in latent trawl pairwise likelihood with* `(x,0)` *where* `x` *positive.*

Description

Computes partial part of second term in latent trawl pairwise likelihood with `(x,0)` where `x` positive.

Usage

```
PairwiseOneZero22(x1, alpha, beta, kappa, B2, trawlA)
```

Arguments

x1	Positive value corresponding to t1.
alpha	Shape parameter. Should be positive.
beta	Latent Gamma scale parameter. Should be positive.
kappa	Exceedance probability parameter. Should be positive.
B2	Intersection area between t1 and t2.
trawlA	Total trawl set measure / area under trawl function.

Value

Partial part of second term in latent trawl pairwise likelihood with (x,0) where x positive.

PairwiseOneZero23 *Computes partial part of second term in latent trawl pairwise likelihood with* (x,0) *where* x *positive.*

Description

Computes partial part of second term in latent trawl pairwise likelihood with (x,0) where x positive.

Usage

```
PairwiseOneZero23(x1, alpha, beta, kappa, B3, trawlA)
```

Arguments

x1	Positive value corresponding to t1.
alpha	Shape parameter. Should be positive.
beta	Latent Gamma scale parameter. Should be positive.
kappa	Exceedance probability parameter. Should be positive.
B3	Difference area between t1 and t2 (in this order).
trawlA	Total trawl set measure / area under trawl function.

Value

Partial part of second term in latent trawl pairwise likelihood with (x,0) where x positive.

`PairwiseOneZero24` *Computes partial part of second term in latent trawl pairwise likelihood with* `(x,0)` *where* `x` *positive.*

Description

Computes partial part of second term in latent trawl pairwise likelihood with `(x,0)` where `x` positive.

Usage

```
PairwiseOneZero24(x1, alpha, beta, kappa, B1, trawlA)
```

Arguments

`x1`	Positive value corresponding to `t1`.
`alpha`	Shape parameter. Should be positive.
`beta`	Latent Gamma scale parameter. Should be positive.
`kappa`	Exceedance probability parameter. Should be positive.
`B1`	Difference area between `t1` and `t2` (in this order).
`trawlA`	Total trawl set measure / area under trawl function.

Value

Partial part of second term in latent trawl pairwise likelihood with `(x,0)` where `x` positive.

`PairwiseZeroZero` *Computes latent trawl pairwise likelihood with* `(x,y) = (0,0)` *for Exponential Trawl funciton.*

Description

Computes latent trawl pairwise likelihood with `(x,y) = (0,0)` for Exponential Trawl funciton.

Usage

```
PairwiseZeroZero(alpha, beta, kappa, B1, B2, B3, transformation = F,
  n.moments = 0)
```

Arguments

alpha	Shape parameter. Should be positive.
beta	Latent Gamma scale parameter. Should be positive.
kappa	Exceedance probability parameter. Should be positive.
B1	Difference area between times t1 and t2.
B2	Intersection area between times t1 and t2.
B3	Difference area between times t2 and t1.
transformation	Boolean to use the Marginal Transform (MT) method.
n.moments	Number of moments achieved by transformed GPD marginals, if used.

Value

Second part of latent trawl pairwise likelihood with (x,y) = (0,0).

Examples

```
PairwiseZeroZero(alpha = 3.2, beta = 2, kappa = 3, B1=0.3, B2=0.7, B3=0.3)
```

PairwiseZeroZero1 *Computes partial part of latent trawl pairwise likelihood with* (x,y) = (0,0).

Description

Computes partial part of latent trawl pairwise likelihood with (x,y) = (0,0).

Usage

```
PairwiseZeroZero1(alpha, beta, kappa)
```

Arguments

alpha	Shape parameter. Should be positive.
beta	Latent Gamma scale parameter. Should be positive.
kappa	Exceedance probability parameter. Should be positive.

Value

Partial part of latent trawl pairwise likelihood with (x,y) = (0,0).

`PairwiseZeroZero2` *Computes second part of latent trawl pairwise likelihood with* `(x,y) = (0,0)`.

Description

Computes second part of latent trawl pairwise likelihood with `(x,y) = (0,0)`.

Usage

```
PairwiseZeroZero2(alpha, beta, kappa, B1, B2, B3)
```

Arguments

`alpha`	Shape parameter. Should be positive.
`beta`	Latent Gamma scale parameter. Should be positive.
`kappa`	Exceedance probability parameter. Should be positive.
`B1`	Difference area between times `t1` and `t2`.
`B2`	Intersection area between times `t1` and `t2`.
`B3`	Difference area between times `t2` and `t1`.

Value

Second part of latent trawl pairwise likelihood with `(x,y) = (0,0)`.

`ParamsListFullPL` *Wrapper for FullPL using a list for parameters in the exponential trawl case.*

Description

Wrapper for FullPL using a list for parameters in the exponential trawl case.

Usage

```
ParamsListFullPL(times, values, delta, params, logscale = T,
  transformation = F)
```

Arguments

times	Vector of timestamps.
values	Vector of target values.
delta	Maximum depth of pairwise likelihood blocks. Should be positive natural integer.
params	List of parameters.
logscale	Boolean to use logscale (log-likelihood). Default T.
transformation	Boolean to use the Marginal Transform (MT) method.

Value

Pairwise Likelihood as per FullPL using a list of parameters instead.

Examples

```
ParamsListFullPL(c(1,2,3,4,5), c(0, 2.3, .3, 0, 5), delta=2,
  params=list("alpha"=2,"beta"=3,"kappa"=1.5, "rho"=0.2), TRUE, FALSE)
```

---

plgpd *Computes the cumulative distribution function of Generalised Pareto Distribution at x, with shape parameter alpha and scale parameter beta.*

Description

Computes the cumulative distribution function of Generalised Pareto Distribution at x, with shape parameter alpha and scale parameter beta.

Usage

```
plgpd(x, alpha, beta, lower.tail = T)
```

Arguments

x	value at which the pdf is evaluated.
alpha	Shape parameter.
beta	Scale parameter, should be positive.
lower.tail	logical; if TRUE (default), probabilities are P[X <= x] otherwise, P[X > x].

Value

GPD CDF function evaluated at x with shape and scale parameters respectively alpha and beta.

Examples

```
plgpd(2.34, alpha = 1, beta = 2)
```

rlexceed *Simulation of extreme value path using latent trawl process. Transformed marginals have scale parameter* 1+kappa.

Description

Simulation of extreme value path using latent trawl process. Transformed marginals have scale parameter 1+kappa.

Usage

```
rlexceed(alpha, beta, kappa, rho = NA, times, marg.dist, n,
  transformation, trawl.function = NA, trawl.fs = NA,
  trawl.fs.prim = NA, n_moments = 4, deep_cols = 30)
```

Arguments

alpha	Shape parameter.
beta	Scale parameter.
kappa	Additive constant to scale parameter beta.
rho	Trawl parameters. Must be positive if Exponential trawls are used.
times	Vectors of discret times.
marg.dist	Name of infinitely divisible distribution for latent trawls. Currenlty implemented: gamma, gaussian, generalised hyperbolic, generalised inverse gaussian.
n	Number of simulations (so far, only n=1 is implemented).
transformation	Boolean to apply marginal transform method. Default is False (F).
trawl.function	Type of trawl function that should be used. Default NA.
trawl.fs	collection of trawl functions indexed on times.
trawl.fs.prim	collection of trawl functions primitives indexed on times.
n_moments	Number of finite moments for transformed marginals.
deep_cols	Depth of reconstruction (columns). Default is 30.

Value

Simulated path (size the same as times) of latent-trawl extreme value process.

Examples

```
alpha <- 3
beta <- 2
kappa <- 0.95
rho <- 0.2
n.timestamps <- 200
times <- 1:n.timestamps
```

```
marg.dist <- "gamma"
n <- 1
transformation <- FALSE
trawl.function <- "exp"

rlexceed(alpha = alpha, beta = beta, kappa = kappa, rho = rho, times = times,
         marg.dist = marg.dist, n = n, transformation = transformation,
         trawl.function= trawl.function)
```

`rtrawl` *Simulation of trawl process path using Slice partition. If using customised trawl functions and primitives (i.e.* `trawl.function = NA`*), then, it is required that the user provides* `length(times) + deep_cols` *such functions and primitives where the* `deep_cols` *first one are used to reconstuct the first trawl value.*

Description

Simulation of trawl process path using Slice partition. If using customised trawl functions and primitives (i.e. `trawl.function = NA`), then, it is required that the user provides `length(times) + deep_cols` such functions and primitives where the `deep_cols` first one are used to reconstuct the first trawl value.

Usage

```
rtrawl(alpha, beta, times, marg.dist, trawl.function = NA,
  trawl.fs = NA, trawl.fs.prim = NA, n, rho = NA, kappa = 0,
  transformation = F, offset_shape = NULL, offset_scale = NULL,
  deep_cols = 30)
```

Arguments

`alpha`	Shape parameter.
`beta`	Scale parameter.
`times`	Vectors of discret times.
`marg.dist`	Name of infinitely divisible distribution. Currenlty implemented: gamma, gaussian, generalised hyperbolic, generalised inverse gaussian.
`trawl.function`	Type of trawl function that should be used. Default NA.
`trawl.fs`	collection of trawl functions indexed on `times`. Default NA. Default NA if no `trawl.function` is indicated and should contain as many as in `times`.
`trawl.fs.prim`	collection of trawl functions primitives indexed on `times`. Default NA if no `trawl.function` is indicated and should contain as many as in `times`.
`n`	Number of simulations (so far, only `n=1` is implemented).

rho	Trawl parameters. Must be positive if Exponential trawls are used.
kappa	Additive constant to scale parameter beta.
transformation	Boolean to apply marginal transform method. Default is False (F).
offset_shape	Transformed-marginal Shape parameter.
offset_scale	Transformed-marginal Scale parameter.
deep_cols	Depth of reconstruction (columns). Default is 30.

Value

Simulated path (size the same as times) of trawl process.

Examples

```
alpha <- 5
beta <- 3
times <- 1:150
rho <- 0.2
trawl.function <- "exp"
margi <- "gamma"
kappa <- 0
n <- 1
rtrawl(alpha = alpha, beta = beta, kappa = kappa, times = times, n = 1,
marg.dist = margi, rho = rho, trawl.function = trawl.function)
```

SecondOrderDiffVal *Second-order centered differentiation calculation.*

Description

Second-order centered differentiation calculation.

Usage

```
SecondOrderDiffVal(value.p, value.c, value.m, epsilon)
```

Arguments

value.p	function value at x+epsilon.
value.c	function value at x.
value.m	function value at x-epsilon.
epsilon	offset value for x.

Value

(value.p -2*value.c + value.m) / (2*epsilon).

SecondOrderMixedDiffVal *Second-order bivariate differentiation calculation.*

Description

Second-order bivariate differentiation calculation.

Usage

```
SecondOrderMixedDiffVal(value.c, value.x.p, value.x.m, value.y.p,
  value.y.m, value.xy.p, value.xy.m, epsilon)
```

Arguments

value.c	function value at (x,y).
value.x.p	function value at (x+epsilonx,y).
value.x.m	function value at (x-epsilon,y).
value.y.p	function value at (x,y+epsilon).
value.y.m	function value at (x,y-epsilon).
value.xy.p	function value at (x+epsilon,y+epsilon).
value.xy.m	function value at (x-epsilon,y-epsilon).
epsilon	offset value for both x and y.

Value

(value.p -2*value.c + value.m) / (2*epsilon).

SinglePairPL *Computes correct latent trawl SINGLE pairwise likelihood depending on the values of* (x1,x2) *with exponential trawl function.*

Description

Computes correct latent trawl SINGLE pairwise likelihood depending on the values of (x1,x2) with exponential trawl function.

Usage

```
SinglePairPL(x1, x2, alpha, beta, kappa, B1, B2, B3, transformation = F,
  n.moments = 0)
```

Arguments

x1	Positive value corresponding to t1.
x2	Positive value corresponding to t2.
alpha	Shape parameter. Should be positive.
beta	Latent Gamma scale parameter. Should be positive.
kappa	Exceedance probability parameter. Should be positive.
B1	Difference area between t1 and t2 (in this order).
B2	Intersection area between t1 and t2 (in this order).
B3	Difference area between t2 and t1 (in this order).
transformation	Boolean to use the Marginal Transform (MT) method.
n.moments	Number of finite moments for transformed marginals.

Value

SINGLE latent trawl pairwise likelihood depending on (x1,x2).

Examples

```
SinglePairPL(x1=0.5, x2= 0.3, alpha=0.3, beta=2, kappa=3, B1=0.3, B2=0.7, B3=0.3, FALSE)
SinglePairPL(x1=2.0, x2=1.0, alpha=-2, beta=3., kappa=3,  B1=0.3, B2=0.7, B3=0.3, TRUE)
```

SliceArea — *Computes the area of a slice for the Slice Partition method of trawl functions.*

Description

Computes the area of a slice for the Slice Partition method of trawl functions.

Usage

```
SliceArea(i, j, times, trawl.f.prim)
```

Arguments

i	main index.
j	secondary index.
times	vector of discret times at which the trawl function is partionned.
trawl.f.prim	Trawl function primitive.

Value

Area of slice S(i,j) in Slice Partition method for trawl functions.

TrawlExp — *Returns an exponential trawl function with base time t.*

Description

Returns an exponential trawl function with base time t.

Usage

```
TrawlExp(t, rho, max_value = 1, min_value = 0.01)
```

Arguments

t	trawl function peak time.
rho	exponential trawl parameter. Should be positive.
max_value	Miximal value of the trawl function (if known). Default 1.
min_value	Minimal value accepted for the trawl function. Default is 1e-2.

Value

(Vectorised) Exponential trawl function with peak time t and parameter rho. If this function is evaluated using NA, it yields a list of key components (rho, max time difference, total area of trawl set A).

TrawlExpPrimitive — *Returns a primitive of exponential trawl function with base time t. Such primitive has its zero at* zero_at.

Description

Returns a primitive of exponential trawl function with base time t. Such primitive has its zero at zero_at.

Usage

```
TrawlExpPrimitive(t, rho, zero_at = -Inf)
```

Arguments

t	trawl function peak time.
rho	exponential trawl parameter. Should be positive.
zero_at	Value at which primitive is zero. Default is -Inf.

Value

(Vectorised) Primitive of Exponential trawl function with peak time t and parameter rho. If this function is evaluated using NA, it yields a list of key components such as the trawl peak time t.

TrawlSliceReconstruct *Performs trawl slices reconstuction to get gamma samples using the independent measure scaterring.*

Description

Performs trawl slices reconstuction to get gamma samples using the independent measure scaterring.

Usage

```
TrawlSliceReconstruct(alpha, beta, times, marg.dist, n, trawl.fs,
  trawl.fs.prim, deep_cols = 30, ghyp.object = NA)
```

Arguments

alpha	(Gamma) Shape parameter.
beta	(Gamma) Scale parameter.
times	Vectors of discret times.
marg.dist	Name of infinitely divisible distribution. Currenlty implemented: gamma, gaussian
n	Number of simulations (so far, only n=1 is implemented).
trawl.fs	collection of trawl functions indexed on times.
trawl.fs.prim	collection of trawl functions primitives indexed on times.
deep_cols	Depth of reconstruction (columns). Default is 30 and must be positive.
ghyp.object	Object from ghyp package when using GH or GIG distributions.

Value

Samples using trawl slice reconstruction disributed using marginal.

TrfFindOffsetScale *Computes predefined transformed scale parameter.*

Description

Computes predefined transformed scale parameter.

Usage

```
TrfFindOffsetScale(alpha, beta, kappa, offset_shape)
```

Arguments

alpha	Shape parameter of output data.
beta	Part of scale parameter of output data (beta + kappa). Should be positive.
kappa	Part of scale parameter of output data. Should be positive.
offset_shape	Shape parameter of input data.

Value

Transformed scale parameter given original data parameters.

TrfG — *Computes g from the marginal transform method (vectorised version). That is from GPD(alpha, beta+kappa) to GPD(offset_shape, offset_scale).*

Description

Computes g from the marginal transform method (vectorised version). That is from GPD(alpha, beta+kappa) to GPD(offset_shape, offset_scale).

Usage

```
TrfG(x, alpha, beta, kappa, offset_scale, offset_shape)
```

Arguments

x	Data distributed as GPD(offset_shape, offset_scale).
alpha	Shape parameter of output data.
beta	Part of scale parameter of output data (beta + kappa). Should be positive.
kappa	Part of scale parameter of output data. Should be positive.
offset_scale	Scale parameter of input data. Should be positive.
offset_shape	Shape parameter of input data.

Value

GPD(offset_shape, offset_scale)-distributed data from GPD(alpha, beta+kappa)-distributed data input data.

`TrfInverseG` *Computes inverse of g from the marginal transform method (vectorised version). That is from GPD(offset_shape, offset_scale) to GPD(alpha, beta+kappa).*

Description

Computes inverse of g from the marginal transform method (vectorised version). That is from GPD(offset_shape, offset_scale) to GPD(alpha, beta+kappa).

Usage

```
TrfInverseG(z, alpha, beta, kappa, offset_scale, offset_shape)
```

Arguments

`z`	Data distributed as GPD(offset_shape, offset_scale).
`alpha`	Shape parameter of output data.
`beta`	Part of scale parameter of output data (beta + kappa). Should be positive.
`kappa`	Part of scale parameter of output data. Should be positive.
`offset_scale`	Scale parameter of input data. Should be positive.
`offset_shape`	Shape parameter of input data.

Value

GPD(alpha, beta+kappa)-distributed data from GPD(offset_shape, offset_scale)-distributed input data.

`TrfJacobian` *Computes determinant of Jacobian of Marginal Transform (MT) method.*

Description

Computes determinant of Jacobian of Marginal Transform (MT) method.

Usage

```
TrfJacobian(z, alpha, beta, kappa, offset_scale, offset_shape)
```

Arguments

z	GPD(offset_shape, offset_scale)-distributed data.
alpha	Shape parameter of output data.
beta	Part of scale parameter of output data (beta + kappa). Should be positive.
kappa	Part of scale parameter of output data. Should be positive.
offset_scale	Scale parameter of input data. Should be positive.
offset_shape	Shape parameter of input data.

Value

Transformed scale parameter given original data parameters.

UnivariateFullPL *Computes univariate latent trawl FULL pairwise likelihood depending with exponential trawl function with the option to fix some parameter values.*

Description

Computes univariate latent trawl FULL pairwise likelihood depending with exponential trawl function with the option to fix some parameter values.

Usage

```
UnivariateFullPL(times, values, delta, fixed_names, fixed_params, params,
  model_vars_names, logscale = T, transformation = F)
```

Arguments

times	Vector of timestamps.
values	Vector of target values.
delta	Maximum depth of pairwise likelihood blocks. Should be positive natural integer.
fixed_names	Vector of literal names of parameters to keep fixed.
fixed_params	Vector of numerical values of fixed parameters.
params	List of parameters.
model_vars_names	Vector of all parameters names in the model.
logscale	Boolean to use logscale (log-likelihood). Default T.
transformation	Boolean to use the Marginal Transform (MT) method.

Value

FULL latent trawl pairwise likelihood with some (or none) parameters fixed.

Examples

```
times <- c(1,2,3,4,5)
values <- c(2,0,3,4,0)
delta <- 2
fixed_names <- c("alpha", "beta")
fixed_params <- c(2.0, 3.4)
params <- c(0.1, 4.3)
model_vars_names <- c("alpha", "beta", "rho", "kappa")
UnivariateFullPL(times, values, delta, fixed_names, fixed_params,
params, model_vars_names, TRUE, FALSE)
```

Zeta *Zeta function from auto-correlation expansion for small lags.*

Description

Zeta function from auto-correlation expansion for small lags.

Usage

```
Zeta(alpha, beta, kappa)
```

Arguments

alpha	Shape parameter.
beta	Scale parameter. Must be positive.
kappa	Exceedance probability parameter. Must be positive.

Examples

```
Zeta(alpha = 5, beta = 4, kappa = 3)
```

Index