Package ‘Pareto’

September 11, 2020

Type Package

Title The Pareto, Piecewise Pareto and Generalized Pareto Distribution

Version 2.2.1

Description Utilities for the Pareto, piecewise Pareto and generalized Pareto distribution that are useful for reinsurance pricing. In particular, the package provides a non-trivial algorithm that can be used to match the expected losses of a tower of reinsurance layers with a layer-independent collective risk model. The theoretical background of the matching algorithm and most other methods are described in Ulrich Riegel (2018) <doi:10.1007/s13385-018-0177-3>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

VignetteBuilder knitr

Depends R (>= 2.10)

Suggests testthat, knitr, rmarkdown, lpSolve

URL https://github.com/ulrichriegel/Pareto#pareto

Language en-US

NeedsCompilation no

Author Ulrich Riegel [aut, cre]

Maintainer Ulrich Riegel <ulrich.riegel@gmx.de>

Repository CRAN

Date/Publication 2020-09-11 13:30:03 UTC

R topics documented:

`dGenPareto` *Density of the generalized Pareto Distribution*

Description

Calculates the density function of the generalized Pareto distribution

Usage

```
dGenPareto(x, t, alpha_ini, alpha_tail, truncation = NULL)
```

Arguments

`x`	Numeric. The function evaluates the density at x.
`t`	Numeric. Threshold of the Pareto distribution.
`alpha_ini`	Numeric. Initial Pareto alpha.
`alpha_tail`	Numeric. Tail Pareto alpha.
`truncation`	Numeric. If truncation is not NULL and truncation > t, then the generalized Pareto distribution is truncated at truncation.

Value

Density function of the Pareto distribution with parameters `t`, `alpha_ini` and `alpha_tail` evaluated at `x`

Examples

```
x <- 0:10 * 1000
dGenPareto(x, 1000, 1, 3)
dGenPareto(x, 1000, 1, 3, truncation = 5000)
```

dPareto *Density of the Pareto Distribution*

Description

Calculates the density function of the Pareto distribution

Usage

```
dPareto(x, t, alpha, truncation = NULL)
```

Arguments

`x`	Numeric. The function evaluates the density at x.
`t`	Numeric. Threshold of the Pareto distribution.
`alpha`	Numeric. Pareto alpha.
`truncation`	Numeric. If truncation is not NULL and truncation > t, then the Pareto distribution is truncated at truncation.

Value

Density function of the Pareto distribution with parameters `t` and `alpha` evaluated at `x`

Examples

```
x <- 0:10 * 1000
dPareto(x, 1000, 2)
dPareto(x, 1000, 2, truncation = 5000)
```

dPiecewisePareto *Density of the Piecewise Pareto Distribution*

Description

Calculates the density function of the piecewise Pareto distribution

Usage

```
dPiecewisePareto(x, t, alpha, truncation = NULL, truncation_type = "lp")
```

Arguments

x	Numeric. The function evaluates the density at x.
t	Numeric vector. Thresholds of the piecewise Pareto distribution.
alpha	Numeric vector. alpha[i] is the Pareto alpha in excess of t[i].
truncation	Numeric. If truncation is not NULL and truncation > t, then the Pareto distribution is truncated at truncation.
truncation_type	Character. If truncation_type = "wd" then the whole distribution is truncated. If truncation_type = "lp" then a truncated Pareto is used for the last piece.

Value

Density function of the piecewise Pareto distribution with parameter vectors t and alpha evaluated at x

Examples

```
t <- c(1000, 2000, 3000)
alpha <- c(1, 1.5, 2)
x <- 0:10 * 1000
dPiecewisePareto(x, t, alpha)
dPiecewisePareto(x, t, alpha, truncation = 5000, truncation_type = "lp")
dPiecewisePareto(x, t, alpha, truncation = 5000, truncation_type = "wd")
```

Example1_AP *Example data: Attachment Points*

Description

Example data: Attachment Points

Usage

```
Example1_AP
```

Format

An object of class numeric of length 5.

Example1_EL — *Example data: Expected Losses*

Description

Example data: Expected Losses

Usage

```
Example1_EL
```

Format

An object of class `numeric` of length 5.

Excess_Frequency — *Expected Frequency in Excess of a Threshold*

Description

Calculates the expected frequency in excess of a threshold for a collective model

Usage

```
Excess_Frequency(CollectiveModel, x = 0)
```

Arguments

`CollectiveModel`
: A collective model object. Currently only `PPP_Models` are handled.

`x`
: Numeric. Threshold.

Value

The expected frequency in excess of `x` for the given `CollectiveModel`

Examples

```
PPPM <- PiecewisePareto_Match_Layer_Losses(Example1_AP, Example1_EL)
PPPM
Excess_Frequency(PPPM, c(-Inf, 0, 1000, 2000, 3000, Inf))
```

`Excess_Frequency.PGP_Model`
Expected Frequency in Excess of a Threshold

Description

Calculates the expected frequency in excess of a threshold for a PGP_model

Usage

```
## S3 method for class 'PGP_Model'
Excess_Frequency(CollectiveModel, x = 0)
```

Arguments

`CollectiveModel`	PGP_Model object.
`x`	Numeric. Threshold.

Value

The expected frequency in excess of `x` for the given `CollectiveModel`

Examples

```
PGPM <- PGP_Model(2, 1000, 1, 2, dispersion = 2)
PGPM
Excess_Frequency(PGPM, c(-Inf, 0, 1000, 2000, 3000, Inf))
```

`Excess_Frequency.PPP_Model`
Expected Frequency in Excess of a Threshold

Description

Calculates the expected frequency in excess of a threshold for a PPP_model

Usage

```
## S3 method for class 'PPP_Model'
Excess_Frequency(CollectiveModel, x = 0)
```

Arguments

`CollectiveModel`
: PPP_Model object.

`x`
: Numeric. Threshold.

Value

The expected frequency in excess of `x` for the given `CollectiveModel`

Examples

```
PPPM <- PiecewisePareto_Match_Layer_Losses(Example1_AP, Example1_EL)
PPPM
Excess_Frequency(PPPM, c(-Inf, 0, 1000, 2000, 3000, Inf))
```

Fit_References — *Fit a Tailor-Made Collective Model that Satisfies a Wishlist of Conditions*

Description

Fits a PPP_Model that fulfils a wishlist of conditions (expected layer losses and excess frequencies)

Usage

```
Fit_References(
  Covers = NULL,
  Attachment_Points = NULL,
  Expected_Layer_Losses = NULL,
  Thresholds = NULL,
  Frequencies = NULL,
  t_1 = min(c(Attachment_Points, Thresholds)),
  default_alpha = 2,
  alpha_max = 100,
  severity_distribution = "PiecewisePareto",
  ignore_inconsistent_references = FALSE
)
```

Arguments

`Covers`
: Numeric vector. Vector containing the covers of the layers from the wishlist.

`Attachment_Points`
: Numeric vector. Vector containing the attachment points of the layers from the wishlist.

`Expected_Layer_Losses`
: Numeric vector. Vector containing the expected losses of the layers from the wishlist.

`Thresholds`	Numeric vector. Contains the thresholds from the whishlist for which excess frequencies are given.
`Frequencies`	Numeric vector. Expected frequencies excess the `Thresholds` from the wishlist.
`t_1`	Numerical. Lowest threshold of the piecewise Pareto distribution.
`default_alpha`	Numerical. Default alpha for situations where an alpha has to be selected.
`alpha_max`	Numerical. Maximum alpha to be used for the matching.
`severity_distribution`	Character. Currently only "PiecewisePareto" is supported.
`ignore_inconsistent_references`	Logical. If TRUE then inconsistent references are ignored in case of the piecewise Pareto distribution and the other references are used to fit the model

Value

A PPP_Model object that contains the information about a collective model with a Panjer distributed claim count and a Piecewise Pareto distributed severity. The object contains the following elements:

- `FQ` Numerical. Frequency in excess of the lowest threshold of the piecewise Pareto distribution
- `t` Numeric vector. Vector containing the thresholds for the piecewise Pareto distribution
- `alpha` Numeric vector. Vector containing the Pareto alphas of the piecewise Pareto distribution
- `truncation` Numerical. If `truncation` is not `NULL` and `truncation` > `max(t)`, then the distribution is truncated at `truncation`.
- `truncation_type` Character. If `truncation_type` = `"wd"` then the whole distribution is truncated. If `truncation_type` = `"lp"` then a truncated Pareto is used for the last piece.
- `dispersion` Numerical. Dispersion of the Panjer distribution (i.e. variance to mean ratio).
- `Status` Numerical indicator: 0 = success, 1 = some information has been ignored, 2 = no solution found
- `Comment` Character. Information on whether the fit was successful

Examples

```
covers <- c(1000, 1000, 1000)
att_points <- c(1000, 2000, 5000)
exp_losses <- c(100, 50, 10)
thresholds <- c(4000, 10000)
fqs <- c(0.04, 0.005)
fit <- Fit_References(covers, att_points, exp_losses, thresholds, fqs)
Layer_Mean(fit, covers, att_points)
Excess_Frequency(fit, thresholds)
```

GenPareto_Layer_Mean *Layer Mean of the generalized Pareto Distribution*

Description

Calculates the expected loss of a generalized Pareto distribution in a reinsurance layer

Usage

```
GenPareto_Layer_Mean(
  Cover,
  AttachmentPoint,
  t,
  alpha_ini,
  alpha_tail,
  truncation = NULL
)
```

Arguments

`Cover`	Numeric. Cover of the reinsurance layer. Use `Inf` for unlimited layers.
`AttachmentPoint`	Numeric. Attachment point of the reinsurance layer.
`t`	Numeric. Threshold of the Pareto distribution. If `t` is `NULL` (default) then `t <-Attachment Point` is used.
`alpha_ini`	Numeric. Initial Pareto alpha (at `t`).
`alpha_tail`	Numeric. Tail Pareto alpha.
`truncation`	Numeric. If `truncation` is not `NULL` and `truncation` > `t`, then the Pareto distribution is truncated at `truncation`.

Value

The expected loss of the (truncated) Pareto distribution with parameters `t` and `alpha` in the layer `Cover` xs `AttachmentPoint`

Examples

```
GenPareto_Layer_Mean(4000, 1000, 1000, 1, 3)
GenPareto_Layer_Mean(4000, 1000, t = 1000, alpha_ini = 1, alpha_tail = 3)
GenPareto_Layer_Mean(4000, 1000, t = 5000, alpha_ini = 1, alpha_tail = 3)
GenPareto_Layer_Mean(4000, 1000, t = 1000, alpha_ini = 1, alpha_tail = 3, truncation = 5000)
GenPareto_Layer_Mean(9000, 1000, t = 1000, alpha_ini = 1, alpha_tail = 3, truncation = 5000)
```

GenPareto_Layer_SM *Second Layer Moment of the Generalized Pareto Distribution*

Description

Calculates the second moment of a generalized Pareto distribution in a reinsurance layer

Usage

```
GenPareto_Layer_SM(
  Cover,
  AttachmentPoint,
  t,
  alpha_ini,
  alpha_tail,
  truncation = NULL
)
```

Arguments

Cover	Numeric. Cover of the reinsurance layer. Use Inf for unlimited layers.
AttachmentPoint	Numeric. Attachment point of the reinsurance layer.
t	Numeric. Threshold of the Pareto distribution. If t is NULL (default) then t <-Attachment Point is used
alpha_ini	Numeric. Initial Pareto alpha (at t).
alpha_tail	Numeric. Tail Pareto alpha.
truncation	Numeric. If truncation is not NULL and truncation > t, then the Pareto distribution is truncated at truncation.

Value

The second moment of the (truncated) generalized Pareto distribution with parameters t, alpha_ini and alpha_tail in the layer Cover xs AttachmentPoint

Examples

```
GenPareto_Layer_SM(4000, 1000, 1000, 1, 2)
GenPareto_Layer_SM(4000, 1000, t = 1000, alpha_ini = 1, alpha_tail = 3)
GenPareto_Layer_SM(4000, 1000, t = 5000, alpha_ini = 1, alpha_tail = 3)
GenPareto_Layer_SM(4000, 1000, t = 1000, alpha_ini = 1, alpha_tail = 3, truncation = 5000)
GenPareto_Layer_SM(9000, 1000, t = 1000, alpha_ini = 1, alpha_tail = 3, truncation = 5000)
```

GenPareto_Layer_Var *Layer Variance of the Generalized Pareto Distribution*

Description

Calculates the variance of a generalized Pareto distribution in a reinsurance layer

Usage

```
GenPareto_Layer_Var(
  Cover,
  AttachmentPoint,
  t,
  alpha_ini,
  alpha_tail,
  truncation = NULL
)
```

Arguments

`Cover`	Numeric. Cover of the reinsurance layer. Use `Inf` for unlimited layers.
`AttachmentPoint`	Numeric. Attachment point of the reinsurance layer.
`t`	Numeric. Threshold of the Pareto distribution. If `t` is `NULL` (default) then `t <-Attachment Point` is used
`alpha_ini`	Numeric. Initial Pareto alpha (at `t`).
`alpha_tail`	Numeric. Tail Pareto alpha.
`truncation`	Numeric. If `truncation` is not `NULL` and `truncation` > `t`, then the Pareto distribution is truncated at `truncation`.

Value

Variance of the (truncated) generalized Pareto distribution with parameters `t`, `alpha_ini` and `alpha_tail` in the layer `Cover` xs `AttachmentPoint`

Examples

```
GenPareto_Layer_Var(4000, 1000, 1000, 1, 2)
GenPareto_Layer_Var(4000, 1000, t = 1000, alpha_ini = 1, alpha_tail = 3)
GenPareto_Layer_Var(4000, 1000, t = 5000, alpha_ini = 1, alpha_tail = 3)
GenPareto_Layer_Var(4000, 1000, t = 1000, alpha_ini = 1, alpha_tail = 3, truncation = 5000)
GenPareto_Layer_Var(9000, 1000, t = 1000, alpha_ini = 1, alpha_tail = 3, truncation = 5000)
```

GenPareto_ML_Estimator_Alpha

Maximum Likelihood Estimation of the Pareto Alphas of a Generalized Pareto Distribution

Description

Calculates the maximum likelihood estimators of the parameters alpha_ini and alpha_tail of a generalized Pareto distribution

Usage

```
GenPareto_ML_Estimator_Alpha(
  losses,
  t,
  truncation = NULL,
  weights = NULL,
  tol = 1e-07,
  max_iterations = 1000,
  alpha_min = 0.01,
  alpha_max = 100
)
```

Arguments

losses	Numeric vector. Losses that are used for the ML estimation.
t	Numeric or numeric vector. Threshold of the generalized Pareto distribution. Alternatively, t can be a vector of same length as losses. In this case t[i] is the reporting threshold of losses[i].
truncation	Numeric. If truncation is not NULL and truncation > t, then the generalized Pareto distribution is truncated at truncation.
weights	Numeric vector. Weights for the losses. For instance weights[i] = 2 and weights[j] = 1 for j != i has the same effect as adding another loss of size loss[i].
tol	Numeric. Desired accuracy (only relevant in the truncated case).
max_iterations	Numeric. Maximum number of iteration in the case truncation < Inf (only relevant in the truncated case).
alpha_min	Numeric. Lower bound for the estimated alphas.
alpha_max	Numeric. Upper bound for the estimated alphas.

Value

Maximum likelihood estimator for the parameters alpha_ini and alpha_tail of a generalized Pareto distribution with threshold t given the observations losses

Examples

```
losses <- rGenPareto(1000, 1000, 2,3)
GenPareto_ML_Estimator_Alpha(losses, 1000)
losses <- rGenPareto(1000, 1000, 2, 1, truncation = 10000)
GenPareto_ML_Estimator_Alpha(losses, 1000)
GenPareto_ML_Estimator_Alpha(losses, 1000, truncation = 10000)

t <- rPareto(10000, 100, 2)
alpha_ini <- 1
alpha_tail <- 3
losses <- rGenPareto(10000, t, alpha_ini, alpha_tail)
GenPareto_ML_Estimator_Alpha(losses, t)
losses <- rGenPareto(10000, t, alpha_ini, alpha_tail, truncation = 2 * max(t))
GenPareto_ML_Estimator_Alpha(losses, t, truncation = 2 * max(t))

losses <- c(190, 600, 120, 270, 180, 120)
w <- rep(1, length(losses))
w[1] <- 3
losses2 <- c(losses, losses[1], losses[1])
GenPareto_ML_Estimator_Alpha(losses, 100, weights = w)
GenPareto_ML_Estimator_Alpha(losses2, 100)
```

---

is.PGP_Model *Check if an object is a PGP_Model*

Description

Checks if the class of an object is 'PGP_Model'

Usage

```
is.PGP_Model(x)
```

Arguments

x	Object to be checked.

Examples

```
PGPM <- PGP_Model(2, 1000, 1, 2, dispersion = 2)
PGPM
is.valid.PGP_Model(PGPM)
is.valid.PGP_Model(PGPM, comment = TRUE)

PGPM$alpha_tail <- -2
is.PGP_Model(PGPM)
is.valid.PGP_Model(PGPM)
is.valid.PGP_Model(PGPM, comment = TRUE)
```

is.PPP_Model *Check if an object is a PPP_Model*

Description

Checks if the class of an object is 'PPP_Model'

Usage

```
is.PPP_Model(x)
```

Arguments

x	Object to be checked.

Examples

```
PPPM <- PPP_Model(2, c(1000,2000), c(1,2), dispersion = 2)
PPPM
is.valid.PPP_Model(PPPM)

PPPM$alpha <- 2
is.valid.PPP_Model(PPPM)
is.PPP_Model(PPPM)
```

is.valid.PGP_Model *Check if an object is a valid PGP_Model*

Description

Checks if an object is a PGP_Model object and whether it is valid for the use in functions like `Layer_Mean`

Usage

```
is.valid.PGP_Model(x, comment = FALSE)
```

Arguments

x	Object to be checked.
comment	If FALSE then the function returns a boolean indicating whether `x` is a valid PGP_Model. If TRUE then the function returns a comment instead.

Examples

```
PGPM <- PGP_Model(2, 1000, 1, 2, dispersion = 2)
PGPM
is.valid.PGP_Model(PGPM)
is.valid.PGP_Model(PGPM, comment = TRUE)

PGPM$alpha_tail <- -2
is.valid.PGP_Model(PGPM)
is.valid.PGP_Model(PGPM, comment = TRUE)
```

---

is.valid.PPP_Model *Check if an object is a valid PPP_Model*

Description

Checks if an object is a PPP_Model object and whether it is valid for the use in functions like `Layer_Mean`

Usage

```
is.valid.PPP_Model(x, comment = FALSE)
```

Arguments

x	Object to be checked.
comment	If FALSE then the function returns a boolean indicating whether x is a valid PPP_Model. If TRUE then the function returns a comment instead.

Examples

```
PPPM <- PPP_Model(2, c(1000,2000), c(1,2), dispersion = 2)
PPPM
is.valid.PPP_Model(PPPM)
is.valid.PPP_Model(PPPM, comment = TRUE)

PPPM$alpha <- 2
is.valid.PPP_Model(PPPM)
is.valid.PPP_Model(PPPM, comment = TRUE)
```

Layer_Mean — *Expected Loss of a Reinsurance Layer*

Description

Calculates the expected loss of a reinsurance layer for a collective model

Usage

```
Layer_Mean(CollectiveModel, Cover = Inf, AttachmentPoint = 0)
```

Arguments

`CollectiveModel`
: A collective model object. Currently only `PPP_Models` are handled.

`Cover`
: Numeric. Cover of the reinsurance layer. Use `Inf` for unlimited layers.

`AttachmentPoint`
: Numeric. Attachment point of the reinsurance layer.

Value

The expected loss of the layer `Cover` xs `AttachmentPoint` for the given `CollectiveModel`

Examples

```
PPPM <- PiecewisePareto_Match_Layer_Losses(Example1_AP, Example1_EL)
PPPM
Example1_Cov <- c(diff(Example1_AP), Inf)
Example1_AP
Example1_Cov
Example1_EL
Layer_Mean(PPPM, Example1_Cov, Example1_AP)
```

Layer_Mean.PGP_Model — *Expected Loss of a Reinsurance Layer*

Description

Calculates the expected loss of a reinsurance layer for a PGP_Model

Usage

```
## S3 method for class 'PGP_Model'
Layer_Mean(CollectiveModel, Cover = Inf, AttachmentPoint = 0)
```

Arguments

CollectiveModel
: PGP_Model object.

Cover
: Numeric. Cover of the reinsurance layer. Use Inf for unlimited layers.

AttachmentPoint
: Numeric. Attachment point of the reinsurance layer.

Value

The expected loss of the layer Cover xs AttachmentPoint for the given CollectiveModel

Examples

```
PGPM <- PGP_Model(2, 1000, 1, 2, dispersion = 2)
PGPM
Example1_Cov <- c(diff(Example1_AP), Inf)
Example1_AP
Example1_Cov
Example1_EL
Layer_Mean(PGPM, Example1_Cov, Example1_AP)
```

Layer_Mean.PPP_Model *Expected Loss of a Reinsurance Layer*

Description

Calculates the expected loss of a reinsurance layer for a PPP_Model

Usage

```
## S3 method for class 'PPP_Model'
Layer_Mean(CollectiveModel, Cover = Inf, AttachmentPoint = 0)
```

Arguments

CollectiveModel
: PPP_Model object.

Cover
: Numeric. Cover of the reinsurance layer. Use Inf for unlimited layers.

AttachmentPoint
: Numeric. Attachment point of the reinsurance layer.

Value

The expected loss of the layer Cover xs AttachmentPoint for the given CollectiveModel

Examples

```
PPPM <- PiecewisePareto_Match_Layer_Losses(Example1_AP, Example1_EL)
PPPM
Example1_Cov <- c(diff(Example1_AP), Inf)
Example1_AP
Example1_Cov
Example1_EL
Layer_Mean(PPPM, Example1_Cov, Example1_AP)
```

Layer_Sd *Standard Deviation of a Reinsurance Layer*

Description

Calculates the standard deviation of the loss in a reinsurance layer for a collective model

Usage

```
Layer_Sd(CollectiveModel, Cover = Inf, AttachmentPoint = 0)
```

Arguments

`CollectiveModel`
: A collective model object. Currently only `PPP_Models` are handled.

`Cover`
: Numeric. Cover of the reinsurance layer. Use `Inf` for unlimited layers.

`AttachmentPoint`
: Numeric. Attachment point of the reinsurance layer.

Value

The standard deviation of the loss in the layer `Cover` xs `AttachmentPoint` for the given `CollectiveModel`

Examples

```
PPPM <- PiecewisePareto_Match_Layer_Losses(Example1_AP, Example1_EL)
PPPM
Example1_Cov <- c(diff(Example1_AP), Inf)
Layer_Sd(PPPM, Example1_Cov, Example1_AP)
```

`Layer_Sd.PGP_Model` *Standard Deviation of a Reinsurance Layer*

Description

Calculates the standard deviation of the loss in a reinsurance layer for a PGP_model

Usage

```
## S3 method for class 'PGP_Model'
Layer_Sd(CollectiveModel, Cover = Inf, AttachmentPoint = 0)
```

Arguments

`CollectiveModel`
: PGP_Model object.

`Cover`
: Numeric. Cover of the reinsurance layer. Use `Inf` for unlimited layers.

`AttachmentPoint`
: Numeric. Attachment point of the reinsurance layer.

Value

The standard deviation of the loss in the layer `Cover` xs `AttachmentPoint` for the given `CollectiveModel`

Examples

```
PGPM <- PGP_Model(2, 1000, 1, 2, dispersion = 2)
PGPM
Example1_Cov <- c(diff(Example1_AP), Inf)
Layer_Sd(PGPM, Example1_Cov, Example1_AP)
```

`Layer_Sd.PPP_Model` *Standard Deviation of a Reinsurance Layer*

Description

Calculates the standard deviation of the loss in a reinsurance layer for a PPP_model

Usage

```
## S3 method for class 'PPP_Model'
Layer_Sd(CollectiveModel, Cover = Inf, AttachmentPoint = 0)
```

Arguments

`CollectiveModel`
: PPP_Model object.

`Cover`
: Numeric. Cover of the reinsurance layer. Use `Inf` for unlimited layers.

`AttachmentPoint`
: Numeric. Attachment point of the reinsurance layer.

Value

The standard deviation of the loss in the layer `Cover` xs `AttachmentPoint` for the given `CollectiveModel`

Examples

```
PPPM <- PiecewisePareto_Match_Layer_Losses(Example1_AP, Example1_EL)
PPPM
Example1_Cov <- c(diff(Example1_AP), Inf)
Layer_Sd(PPPM, Example1_Cov, Example1_AP)
```

`Layer_Var` *Variance of a Reinsurance Layer*

Description

Calculates the variance of the loss in a reinsurance layer for a collective model

Usage

```
Layer_Var(CollectiveModel, Cover = Inf, AttachmentPoint = 0)
```

Arguments

`CollectiveModel`
: A collective model object. Currently only `PPP_Models` are handled.

`Cover`
: Numeric. Cover of the reinsurance layer. Use `Inf` for unlimited layers.

`AttachmentPoint`
: Numeric. Attachment point of the reinsurance layer.

Value

The variance of the loss in the layer `Cover` xs `AttachmentPoint` for the given `CollectiveModel`

Examples

```
PPPM <- PiecewisePareto_Match_Layer_Losses(Example1_AP, Example1_EL)
PPPM
Example1_Cov <- c(diff(Example1_AP), Inf)
Layer_Var(PPPM, Example1_Cov, Example1_AP)
```

Layer_Var.PGP_Model *Variance of a Reinsurance Layer*

Description

Calculates the variance of the loss in a reinsurance layer for a PGP_model

Usage

```
## S3 method for class 'PGP_Model'
Layer_Var(CollectiveModel, Cover = Inf, AttachmentPoint = 0)
```

Arguments

`CollectiveModel`	PGP_Model object.
`Cover`	Numeric. Cover of the reinsurance layer. Use `Inf` for unlimited layers.
`AttachmentPoint`	Numeric. Attachment point of the reinsurance layer.

Value

The variance of the loss in the layer `Cover` xs `AttachmentPoint` for the given `CollectiveModel`

Examples

```
PGPM <- PGP_Model(2, 1000, 1, 2, dispersion = 2)
PGPM
Example1_Cov <- c(diff(Example1_AP), Inf)
Layer_Var(PGPM, Example1_Cov, Example1_AP)
```

Layer_Var.PPP_Model *Variance of a Reinsurance Layer*

Description

Calculates the variance of the loss in a reinsurance layer for a PPP_model

Usage

```
## S3 method for class 'PPP_Model'
Layer_Var(CollectiveModel, Cover = Inf, AttachmentPoint = 0)
```

Arguments

`CollectiveModel`
: PPP_Model object.

`Cover`
: Numeric. Cover of the reinsurance layer. Use `Inf` for unlimited layers.

`AttachmentPoint`
: Numeric. Attachment point of the reinsurance layer.

Value

The variance of the loss in the layer `Cover` xs `AttachmentPoint` for the given `CollectiveModel`

Examples

```
PPPM <- PiecewisePareto_Match_Layer_Losses(Example1_AP, Example1_EL)
PPPM
Example1_Cov <- c(diff(Example1_AP), Inf)
Layer_Var(PPPM, Example1_Cov, Example1_AP)
```

`Local_Pareto_Alpha` *Local Pareto Alpha*

Description

Calculates the local Pareto alpha of the normal, lognormal and gamma distribution

Usage

```
Local_Pareto_Alpha(x, distribution, ...)
```

Arguments

`x`
: Numeric. Vector of thresholds at which the local Pareto alpha is calculated.

`distribution`
: Character.
  - `'lnorm'` for lognormal distribution (arguments: `meanlog`, `sdlog`)
  - `'norm'` for normal distribution (arguments: `mean`, `sd`)
  - `'gamma'` for gamma distribution (arguments: `shape`, `rate`, `scale`)

`...`
: Arguments for the selected distribution

Value

Local Pareto alpha of the selected distribution at `x`

References

Riegel, U. (2008) Generalizations of common ILF models. Blaetter der DGVFM 29: 45–71

Examples

```
x <- 1:10
Local_Pareto_Alpha(x, "norm", mean = 1, sd = 5)
x <- 1:10 * 1000000
Local_Pareto_Alpha(x, "lnorm", meanlog = 1, sdlog = 5)
```

Pareto_CDF *Distribution Function of the Pareto Distribution*

Description

Calculates the cumulative distribution function of a Pareto distribution. This function is deprecated. Use `pPareto` instead.

Usage

```
Pareto_CDF(x, t, alpha, truncation = NULL)
```

Arguments

`x`	Numeric. The function evaluates the CDF at `x`.
`t`	Numeric. Threshold of the Pareto distribution.
`alpha`	Numeric. Pareto alpha.
`truncation`	Numeric. If `truncation` is not `NULL` and `truncation` > `t`, then the Pareto distribution is truncated at `truncation`.

Value

Distribution function of the Pareto distribution with parameters `t` and `alpha` evaluated at `x`

Examples

```
x <- 0:10 * 1000
pPareto(x, 1000, 2)
pPareto(x, 1000, 2, truncation = 5000)
```

Pareto_Extrapolation *Pareto Extrapolation*

Description

Uses a Pareto distribution to derive the expected loss of a layer from the expected loss of another layer

Usage

```
Pareto_Extrapolation(
  Cover_1,
  AttachmentPoint_1,
  Cover_2,
  AttachmentPoint_2,
  alpha,
  ExpLoss_1 = NULL,
  truncation = NULL
)
```

Arguments

`Cover_1`	Numeric. Cover of the layer from which we extrapolate. Use `Inf` for unlimited layers.
`AttachmentPoint_1`	Numeric. Attachment point of the layer from which we extrapolate.
`Cover_2`	Numeric. Cover of the layer to which we extrapolate. Use `Inf` for unlimited layers.
`AttachmentPoint_2`	Numeric. Attachment point of the layer to which we extrapolate.
`alpha`	Numeric. Pareto alpha used for the extrapolation.
`ExpLoss_1`	Numeric. Expected loss of the layer from which we extrapolate. If `NULL` (default) then the function provides only the ratio between the expected losses of the layers.
`truncation`	Numeric. If `truncation` is not `NULL` and `truncation` > `AttachmentPoint_1`, then the Pareto distribution is truncated at `truncation`.

Value

The expected loss of the layer `Cover_2` xs `AttachmentPoint_2` given that `Cover_1` xs `AttachmentPoint_1` has expected loss `ExpLoss_1` and assuming a (truncated) Pareto distribution with parameters `t` and `alpha`. If missing then `ExpLoss_1 == 1` is assumed.

References

Riegel, U. (2018) Matching tower information with piecewise Pareto. European Actuarial Journal 8(2): 437–460

Examples

```
Pareto_Extrapolation(1000, 1000, 2000, 2000, 2, ExpLoss_1 = 100)
Pareto_Extrapolation(1000, 1000, 2000, 2000, 2) * 100
Pareto_Extrapolation(1000, 1000, 2000, 2000, 2, truncation = 5000, ExpLoss_1 = 100)
Pareto_Extrapolation(1000, 1000, 2000, 2000, 2, truncation = 5000) * 100
```

Pareto_Find_Alpha_btw_FQs

Pareto Alpha Between Two Frequencies

Description

Finds the Pareto alpha between two excess frequencies

Usage

```
Pareto_Find_Alpha_btw_FQs(
  Threshold_1,
  Frequency_1,
  Threshold_2,
  Frequency_2,
  max_alpha = 100,
  tolerance = 1e-10,
  truncation = NULL
)
```

Arguments

Threshold_1	Numeric. Threshold 1
Frequency_1	Numeric. Expected frequency in excess of Threshold_1
Threshold_2	Numeric. Threshold 2
Frequency_2	Numeric. Expected frequency in excess of Threshold_2
max_alpha	Numeric. Upper limit for the alpha that is returned.
tolerance	Numeric. Accuracy of the result.
truncation	Numeric. If truncation is not NULL then the Pareto distribution is truncated at truncation.

Value

The Pareto alpha between the expected number of claims Frequency_1 excess Threshold_1 and the expected number of claims Frequency_2 excess Threshold_2

References

Riegel, U. (2018) Matching tower information with piecewise Pareto. European Actuarial Journal 8(2): 437–460

Examples

```
Pareto_Find_Alpha_btw_FQs(1000, 1, 2000, 0.5)
Pareto_Find_Alpha_btw_FQs(1000, 1, 2000, 0.5, truncation = 5000)
```

Pareto_Find_Alpha_btw_FQ_Layer

Pareto Alpha Between a Frequency and a Layer

Description

Finds the Pareto alpha between an excess frequency and the expected loss of a layer

Usage

```
Pareto_Find_Alpha_btw_FQ_Layer(
  Threshold,
  Frequency,
  Cover,
  AttachmentPoint,
  ExpLoss,
  max_alpha = 100,
  tolerance = 1e-10,
  truncation = NULL
)
```

Arguments

`Threshold`	Numeric. Threshold
`Frequency`	Numeric. Expected frequency in excess of `Thershold`
`Cover`	Numeric. Cover of the second layer.
`AttachmentPoint`	Numeric. Attachment point of the layer.
`ExpLoss`	Numeric. Expected loss of the layer.
`max_alpha`	Numeric. Upper limit for the alpha that is returned.
`tolerance`	Numeric. Accuracy of the result.
`truncation`	Numeric. If `truncation` is not `NULL` then the Pareto distribution is truncated at `truncation`.

Value

The Pareto alpha between the expected number of claims `Frequency` excess `Threshold` and the layer `Cover` xs `AttachmentPoint` with expected loss `ExpLoss`

References

Riegel, U. (2018) Matching tower information with piecewise Pareto. European Actuarial Journal 8(2): 437–460

Examples

```
Pareto_Find_Alpha_btw_FQ_Layer(1000, 1, 1000, 1000, 500)
Pareto_Find_Alpha_btw_FQ_Layer(1000, 1, 1000, 1000, 500, truncation = 5000)
```

`Pareto_Find_Alpha_btw_Layers`

Pareto Alpha Between Two Layers

Description

Finds the Pareto alpha between two layers

Usage

```
Pareto_Find_Alpha_btw_Layers(
  Cover_1,
  AttachmentPoint_1,
  ExpLoss_1,
  Cover_2,
  AttachmentPoint_2,
  ExpLoss_2,
  max_alpha = 100,
  tolerance = 1e-10,
  truncation = NULL
)
```

Arguments

`Cover_1`	Numeric. Cover of the first layer.
`AttachmentPoint_1`	Numeric. Attachment point of the first layer.
`ExpLoss_1`	Numeric. Expected loss of the first layer.
`Cover_2`	Numeric. Cover of the second layer.
`AttachmentPoint_2`	Numeric. Attachment point of the second layer.
`ExpLoss_2`	Numeric. Expected loss of the second layer.
`max_alpha`	Numeric. Upper limit for the alpha that is returned.
`tolerance`	Numeric. Accuracy of the result.
`truncation`	Numeric. If `truncation` is not `NULL` then the Pareto distribution is truncated at `truncation`.

Value

The Pareto alpha between the layer `Cover_1` xs `AttachmentPoint_1` with expected loss `ExpLoss_1` and the layer `Cover_2` xs `AttachmentPoint_2` with expected loss `ExpLoss_2`

References

Riegel, U. (2018) Matching tower information with piecewise Pareto. European Actuarial Journal 8(2): 437–460

Examples

```
Pareto_Find_Alpha_btw_Layers(100, 100, 100, 200, 200, 50)
Pareto_Find_Alpha_btw_Layers(100, 100, 100, 200, 200, 50, truncation = 500)
```

`Pareto_Layer_Mean` *Layer Mean of the Pareto Distribution*

Description

Calculates the expected loss of a Pareto distribution in a reinsurance layer

Usage

```
Pareto_Layer_Mean(Cover, AttachmentPoint, alpha, t = NULL, truncation = NULL)
```

Arguments

`Cover`	Numeric. Cover of the reinsurance layer. Use `Inf` for unlimited layers.
`AttachmentPoint`	Numeric. Attachment point of the reinsurance layer.
`alpha`	Numeric. Pareto alpha.
`t`	Numeric. Threshold of the Pareto distribution. If `t` is `NULL` (default) then `t <-Attachment Point` is used.
`truncation`	Numeric. If `truncation` is not `NULL` and `truncation` > `t`, then the Pareto distribution is truncated at `truncation`.

Value

The expected loss of the (truncated) Pareto distribution with parameters `t` and `alpha` in the layer `Cover` xs `AttachmentPoint`

Examples

```
Pareto_Layer_Mean(4000, 1000, 2)
Pareto_Layer_Mean(4000, 1000, alpha = 2, t = 1000)
Pareto_Layer_Mean(4000, 1000, alpha = 2, t = 5000)
Pareto_Layer_Mean(4000, 1000, alpha = 2, t = 1000, truncation = 5000)
Pareto_Layer_Mean(9000, 1000, alpha = 2, t = 1000, truncation = 5000)
```

Pareto_Layer_SM *Second Layer Moment of the Pareto Distribution*

Description

Calculates the second moment of a Pareto distribution in a reinsurance layer

Usage

```
Pareto_Layer_SM(Cover, AttachmentPoint, alpha, t = NULL, truncation = NULL)
```

Arguments

`Cover`	Numeric. Cover of the reinsurance layer. Use `Inf` for unlimited layers.
`AttachmentPoint`	Numeric. Attachment point of the reinsurance layer.
`alpha`	Numeric. Pareto alpha.
`t`	Numeric. Threshold of the Pareto distribution. If `t` is `NULL` (default) then `t <-Attachment Point` is used
`truncation`	Numeric. If `truncation` is not `NULL` and `truncation` > `t`, then the Pareto distribution is truncated at `truncation`.

Value

The second moment of the (truncated) Pareto distribution with parameters `t` and `alpha` in the layer `Cover` xs `AttachmentPoint`

Examples

```
Pareto_Layer_SM(4000, 1000, 2)
Pareto_Layer_SM(4000, 1000, alpha = 2, t = 1000)
Pareto_Layer_SM(4000, 1000, alpha = 2, t = 5000)
Pareto_Layer_SM(4000, 1000, alpha = 2, t = 1000, truncation = 5000)
Pareto_Layer_SM(9000, 1000, alpha = 2, t = 1000, truncation = 5000)
```

Pareto_Layer_Var *Layer Variance of the Pareto Distribution*

Description

Calculates the variance of a Pareto distribution in a reinsurance layer

Usage

```
Pareto_Layer_Var(Cover, AttachmentPoint, alpha, t = NULL, truncation = NULL)
```

Arguments

Cover	Numeric. Cover of the reinsurance layer. Use Inf for unlimited layers.
AttachmentPoint	Numeric. Attachment point of the reinsurance layer.
alpha	Numeric. Pareto alpha.
t	Numeric. Threshold of the Pareto distribution. If t is NULL (default) then t <-Attachment Point is used.
truncation	Numeric. If truncation is not NULL and truncation > t, then the Pareto distribution is truncated at truncation.

Value

The variance of the(truncated) Pareto distribution with parameters t and alpha in the layer Cover xs AttachmentPoint

Examples

```
Pareto_Layer_Var(4000, 1000, 2)
Pareto_Layer_Var(4000, 1000, alpha = 2, t = 1000)
Pareto_Layer_Var(4000, 1000, alpha = 2, t = 5000)
Pareto_Layer_Var(4000, 1000, alpha = 2, t = 1000, truncation = 5000)
Pareto_Layer_Var(9000, 1000, alpha = 2, t = 1000, truncation = 5000)
```

Pareto_ML_Estimator_Alpha *Maximum Likelihood Estimation of the Pareto Alpha*

Description

Calculates the maximum likelihood estimator of the parameter alpha of a Pareto distribution

Usage

```
Pareto_ML_Estimator_Alpha(
  losses,
  t,
  truncation = NULL,
  weights = NULL,
  tol = 1e-07,
  max_iterations = 1000,
  alpha_min = 0,
  alpha_max = Inf
)
```

Arguments

losses	Numeric vector. Losses that are used for the ML estimation.
t	Numeric or numeric vector. Threshold of the Pareto distribution. Alternatively, t can be a vector of same length as losses. In this case t[i] is the reporting threshold of losses[i].
truncation	Numeric. If truncation is not NULL and truncation > t, then the Pareto distribution is truncated at truncation.
weights	Numeric vector. Weights for the losses. For instance weights[i] = 2 and weights[j] = 1 for j != i has the same effect as adding another loss of size loss[i].
tol	Numeric. Desired accuracy (only relevant in the truncated case).
max_iterations	Numeric. Maximum number of iteration in the case truncation < Inf (only relevant in the truncated case).
alpha_min	Numeric. Deprecated.
alpha_max	Numeric. Deprecated.

Value

Maximum likelihood estimator for the parameter alpha of a Pareto distribution with threshold t given the observations losses

Examples

```
losses <- rPareto(100, 1000, 2)
Pareto_ML_Estimator_Alpha(losses, 1000)
losses <- rPareto(100, 1000, 2, truncation = 2000)
Pareto_ML_Estimator_Alpha(losses, 1000)
Pareto_ML_Estimator_Alpha(losses, 1000, truncation = 2000)

t <- rPareto(10000, 100, 2)
alpha <- 2
losses <- rPareto(10000, t, alpha)
Pareto_ML_Estimator_Alpha(losses, t)
losses <- rPareto(10000, t, alpha, truncation = 2 * max(t))
```

```
Pareto_ML_Estimator_Alpha(losses, t, truncation = 2 * max(t))

losses <- rPareto(10, 1000, 2)
w <- rep(1, 10)
w[1] <- 3
losses2 <- c(losses, losses[1], losses[1])
Pareto_ML_Estimator_Alpha(losses, 1000, weights = w)
Pareto_ML_Estimator_Alpha(losses2, 1000)
```

Pareto_PDF *Density of the Pareto Distribution*

Description

Calculates the density function of the Pareto distribution. This function is deprecated. Use `dPareto` instead.

Usage

```
Pareto_PDF(x, t, alpha, truncation = NULL)
```

Arguments

x	Numeric. The function evaluates the density at x
t	Numeric. Threshold of the Pareto distribution.
alpha	Numeric. Pareto alpha.
truncation	Numeric. If truncation is not NULL and truncation > t, then the Pareto distribution is truncated at truncation.

Value

Density function of the Pareto distribution with parameters t and alpha evaluated at x

Examples

```
x <- 0:10 * 1000
dPareto(x, 1000, 2)
dPareto(x, 1000, 2, truncation = 5000)
```

pGenPareto — *Distribution Function of the generalized Pareto Distribution*

Description

Calculates the cumulative distribution function of a generalized Pareto distribution

Usage

```
pGenPareto(x, t, alpha_ini, alpha_tail, truncation = NULL)
```

Arguments

`x`	Numeric. The function evaluates the CDF at `x`.
`t`	Numeric. Threshold of the generalized Pareto distribution.
`alpha_ini`	Numeric. Initial Pareto alpha.
`alpha_tail`	Numeric. Tail Pareto alpha.
`truncation`	Numeric. If `truncation` is not `NULL` and `truncation` > `t`, then the generalized Pareto distribution is truncated at `truncation`.

Value

Distribution function of the generalized Pareto distribution with parameters `t`, `alpha_ini` and `alpha_tail` evaluated at `x`

Examples

```
x <- 0:10 * 1000
pGenPareto(x, 1000, 1, 3)
pGenPareto(x, 1000, 1, 3, truncation = 5000)
```

PGP_Model — *PGP_Model (Collective Panjer & Generalized Pareto Model) Object*

Description

Constructor function for the PGP_Model object

Usage

```
PGP_Model(
  FQ = NULL,
  t = NULL,
  alpha_ini = NULL,
  alpha_tail = NULL,
  truncation = NULL,
  dispersion = 1,
  Status = 0,
  Comment = "OK"
)
```

Arguments

FQ	Numerical. Expected claim count of the collective model.
t	Numeric. Threshold of the Pareto distribution. If t is NULL (default) then t <-Attachment Point is used
alpha_ini	Numeric. Initial Pareto alpha (at t).
alpha_tail	Numeric. Tail Pareto alpha.
truncation	Numeric. If truncation is not NULL and truncation > t, then the Pareto distribution is truncated at truncation.
dispersion	Numerical. Dispersion of the Panjer distribution (i.e. variance to mean ratio).
Status	Numerical indicator if a function returns a PGP_Model object: 0 = success, 1 = some information has been ignored, 2 = no solution found
Comment	Charakter. An optional comment.

Examples

```
PGPM <- PGP_Model(2, t = 1000, alpha_ini = 1, alpha_tail = 2 , dispersion = 2)
PGPM
```

PiecewisePareto_CDF *Distribution Function of the Piecewise Pareto Distribution*

Description

Calculates the cumulative distribution function of a Piecewise Pareto Distribution. This function is deprecated. Use pPiecewisePareto instead.

Usage

```
PiecewisePareto_CDF(x, t, alpha, truncation = NULL, truncation_type = "lp")
```

Arguments

x	Numeric. The function evaluates the CDF at x.
t	Numeric vector. Thresholds of the piecewise Pareto distribution.
alpha	Numeric vector. alpha[i] is the Pareto alpha in excess of t[i].
truncation	Numeric. If truncation is not NULL and truncation > t, then the distribution is truncated at truncation.
truncation_type	Character. If truncation_type = "wd" then the whole distribution is truncated. If truncation_type = "lp" then a truncated Pareto is used for the last piece.

Value

Distribution function of the piecewise Pareto distribution with parameter vectors t and alpha evaluated at x

References

Riegel, U. (2018) Matching tower information with piecewise Pareto. European Actuarial Journal 8(2): 437–460

Examples

```
t <- c(1000, 2000, 3000)
alpha <- c(1, 1.5, 2)
x <- 0:10 * 1000
pPiecewisePareto(x, t, alpha)
pPiecewisePareto(x, t, alpha, truncation = 5000, truncation_type = "lp")
pPiecewisePareto(x, t, alpha, truncation = 5000, truncation_type = "wd")
```

PiecewisePareto_Layer_Mean

Layer Mean of the Piecewise Pareto Distribution

Description

Calculates the expected loss of a piecewise Pareto distribution in a reinsurance layer

Usage

```
PiecewisePareto_Layer_Mean(
  Cover,
  AttachmentPoint,
  t,
  alpha,
  truncation = NULL,
  truncation_type = "lp"
)
```

Arguments

- `Cover` Numeric. Cover of the reinsurance layer.
- `AttachmentPoint` Numeric. Attachment point of the reinsurance layer.
- `t` Numeric vector. Thresholds of the piecewise Pareto distribution.
- `alpha` Numeric vector. `alpha[i]` is the Pareto alpha in excess of `t[i]`.
- `truncation` Numeric. If `truncation` is not `NULL` and `truncation` > `t`, then the Pareto distribution is truncated at `truncation`.
- `truncation_type` Character. If `truncation_type` = `"wd"` then the whole distribution is truncated. If `truncation_type` = `"lp"` then a truncated Pareto is used for the last piece.

Value

The expected loss of the (truncated) piecewise Pareto distribution with parameter vectors `t` and `alpha` in the layer `Cover` xs `AttachmentPoint`

References

Riegel, U. (2018) Matching tower information with piecewise Pareto. European Actuarial Journal 8(2): 437–460

Examples

```
t <- c(1000, 2000, 3000)
alpha <- c(1, 1.5, 2)
PiecewisePareto_Layer_Mean(4000, 1000, t, alpha)
PiecewisePareto_Layer_Mean(4000, 1000, t, alpha, truncation = 5000)
PiecewisePareto_Layer_Mean(4000, 1000, t, alpha, truncation = 5000, truncation_type = "lp")
PiecewisePareto_Layer_Mean(4000, 1000, t, alpha, truncation = 5000, truncation_type = "wd")
```

`PiecewisePareto_Layer_SM` *Second Layer Moment of the Piecewise Pareto Distribution*

Description

Calculates the second moment of a piecewise Pareto distribution in a reinsurance layer

Usage

```
PiecewisePareto_Layer_SM(
  Cover,
  AttachmentPoint,
  t,
  alpha,
  truncation = NULL,
  truncation_type = "lp"
)
```

Arguments

`Cover`	Numeric. Cover of the reinsurance layer.
`AttachmentPoint`	Numeric. Attachment point of the reinsurance layer.
`t`	Numeric vector. Thresholds of the piecewise Pareto distribution.
`alpha`	Numeric vector. `alpha[i]` is the Pareto alpha in excess of `t[i]`.
`truncation`	Numeric. If `truncation` is not `NULL` and `truncation` > `t`, then the Pareto distribution is truncated at `truncation`.
`truncation_type`	Character. If `truncation_type` = `"wd"` then the whole distribution is truncated. If `truncation_type` = `"lp"` then a truncated Pareto is used for the last piece.

Value

The second moment of the (truncated) piecewise Pareto distribution with parameter vectors `t` and `alpha` in the layer `Cover` xs `AttachmentPoint`

Examples

```
t <- c(1000, 2000, 3000)
alpha <- c(1, 1.5, 2)
PiecewisePareto_Layer_SM(4000, 1000, t, alpha)
PiecewisePareto_Layer_SM(4000, 1000, t, alpha, truncation = 5000)
PiecewisePareto_Layer_SM(4000, 1000, t, alpha, truncation = 5000, truncation_type = "lp")
PiecewisePareto_Layer_SM(4000, 1000, t, alpha, truncation = 5000, truncation_type = "wd")
```

PiecewisePareto_Layer_Var

Layer Variance of the Piecewise Pareto Distribution

Description

Calculate the variance of a piecewise Pareto distribution in a reinsurance layer

Usage

```
PiecewisePareto_Layer_Var(
  Cover,
  AttachmentPoint,
  t,
  alpha,
  truncation = NULL,
  truncation_type = "lp"
)
```

Arguments

- `Cover` Numeric. Cover of the reinsurance layer.
- `AttachmentPoint` Numeric. Attachment point of the reinsurance layer.
- `t` Numeric vector. Thresholds of the piecewise Pareto distribution.
- `alpha` Numeric vector. `alpha[i]` is the Pareto alpha in excess of `t[i]`.
- `truncation` Numeric. If `truncation` is not `NULL` and `truncation` > `t`, then the Pareto distribution is truncated at `truncation`.
- `truncation_type` Character. If `truncation_type` = `"wd"` then the whole distribution is truncated. If `truncation_type` = `"lp"` then a truncated Pareto is used for the last piece.

Value

The variance of the (truncated) piecewise Pareto distribution with parameter vectors `t` and `alpha` in the layer `Cover` xs `AttachmentPoint`

Examples

```
t <- c(1000, 2000, 3000)
alpha <- c(1, 1.5, 2)
PiecewisePareto_Layer_Var(4000, 1000, t, alpha)
PiecewisePareto_Layer_SM(4000, 1000, t, alpha) - PiecewisePareto_Layer_Mean(4000, 1000, t, alpha)^2
PiecewisePareto_Layer_Var(4000, 1000, t, alpha, truncation = 5000)
PiecewisePareto_Layer_Var(4000, 1000, t, alpha, truncation = 5000, truncation_type = "lp")
PiecewisePareto_Layer_Var(4000, 1000, t, alpha, truncation = 5000, truncation_type = "wd")
```

`PiecewisePareto_Match_Layer_Losses` *Match a Tower of Expected Layers Losses*

Description

Matches the expected losses of a tower of reinsurance layers using a piecewise Pareto severity

Usage

```
PiecewisePareto_Match_Layer_Losses(
  Attachment_Points,
  Expected_Layer_Losses,
  Unlimited_Layers = FALSE,
  Frequencies = NULL,
  FQ_at_lowest_AttPt = NULL,
  FQ_at_highest_AttPt = NULL,
  TotalLoss_Frequencies = NULL,
  minimize_ratios = TRUE,
  Use_unlimited_Layer_for_FQ = TRUE,
  truncation = NULL,
  truncation_type = "lp",
  dispersion = 1,
  tolerance = 1e-10,
  alpha_max = 100,
  merge_tolerance = 1e-06,
  RoL_tolerance = 1e-06
)
```

Arguments

`Attachment_Points`
: Numeric vector. Vector containing the attachment points of consecutive layers in increasing order

`Expected_Layer_Losses`
: Numeric vector. Vector containing the expected losses of layers xs the attachment points.

`Unlimited_Layers`
: Logical. If `TRUE`, then `Expected_Layer_Losses[i]` contains the expected loss of `Inf` xs `Attachment_Points[i]`. If `FALSE` then `Expected_Layer_Losses[i]` contains the expected loss of the layer `Attachment_Points[i+1]` xs `Attachment_Points[i]`

`Frequencies`
: Numeric vector. Expected frequencies excess the attachment points. The vector may contain NAs. If `NULL` then the function calculates frequencies.

`FQ_at_lowest_AttPt`
: Numerical. Expected frequency excess `Attachment_Points[1]`. Overrules first entry in Frequencies.

`FQ_at_highest_AttPt`
: Numerical. Expected frequency excess `Attachment_Points[k]`. Overrules last entry in Frequencies.

`TotalLoss_Frequencies`
: Numeric vector. `TotalLoss_Frequencies[i]` is the frequency of total losses to layer `i` (i.e. `Attachment_Points[i+1] -Attachment_Points[i]` xs `Attachment_Points[i]`). `TotalLoss_Frequencies[i]` is the frequency for losses larger than or equal to `Attachment_Points[i+1]`, whereas `Frequencies[i]` is the frequency of losses larger than `Attachment_Points[i]`. `TotalLoss_Frequencies[i]` > `Frequencies[i+1]` means that there is a point mass of the severity at `Attachment_Points[i+1]`.

minimize_ratios
: Logical. If TRUE then ratios between alphas are minimized.

Use_unlimited_Layer_for_FQ
: Logical. Only relevant if no frequency is provided for the highest attachment point by the user. If TRUE then the frequency is calculated using the Pareto alpha between the last two layers.

truncation
: Numeric. If truncation is not NULL, then the distribution is truncated at truncation.

truncation_type
: Character. If truncation_type = "wd" then the whole distribution is truncated. If truncation_type = "lp" then a truncated Pareto is used for the last piece.

dispersion
: Numerical. Dispersion of the claim count distribution in the resulting PPP_Model.

tolerance
: Numeric. Numerical tolerance.

alpha_max
: Numerical. Maximum alpha to be used for the matching.

merge_tolerance
: Numerical. Consecutive Pareto pieces are merged if the alphas deviate by less than merge_tolerance.

RoL_tolerance
: Numerical. Consecutive layers are merged if RoL decreases less than factor 1 -RoL_tolerance.

Value

A PPP_Model object that contains the information about a collective model with a Panjer distributed claim count and a Piecewise Pareto distributed severity. The object contains the following elements:

- FQ Numerical. Frequency in excess of the lowest threshold of the piecewise Pareto distribution
- t Numeric vector. Vector containing the thresholds for the piecewise Pareto distribution
- alpha Numeric vector. Vector containing the Pareto alphas of the piecewise Pareto distribution
- truncation Numerical. If truncation is not NULL and truncation > max(t), then the distribution is truncated at truncation.
- truncation_type Character. If truncation_type = "wd" then the whole distribution is truncated. If truncation_type = "lp" then a truncated Pareto is used for the last piece.
- dispersion Numerical. Dispersion of the Panjer distribution (i.e. variance to mean ratio).
- Status Numerical indicator: 0 = success, 1 = some information has been ignored, 2 = no solution found
- Comment Character. Information on whether the fit was successful

References

Riegel, U. (2018) Matching tower information with piecewise Pareto. European Actuarial Journal 8(2): 437–460

Examples

```
AP <- Example1_AP
EL <- Example1_EL
PiecewisePareto_Match_Layer_Losses(AP, EL)
EL_unlimited <- rev(cumsum(rev(Example1_EL)))
PiecewisePareto_Match_Layer_Losses(AP, EL_unlimited, Unlimited_Layers = TRUE)
PiecewisePareto_Match_Layer_Losses(AP, EL, FQ_at_lowest_AttPt = 0.5)
Example1_FQ <- c(0.3, 0.15, 0.08, 0.02, 0.005)
PiecewisePareto_Match_Layer_Losses(AP, EL, Frequencies = Example1_FQ)
```

`PiecewisePareto_ML_Estimator_Alpha`

Maximum Likelihood Estimation of the Alphas of the Piecewise Pareto Distribution

---

Description

Calculates the maximum likelihood estimator of the parameter vector alpha of a piecewise Pareto distribution

Usage

```
PiecewisePareto_ML_Estimator_Alpha(
  losses,
  t,
  truncation = NULL,
  truncation_type = "lp",
  weights = NULL,
  tol = 1e-07,
  max_iterations = 1000
)
```

Arguments

`losses`	Numeric vector. Losses that are used for the ML estimation.
`t`	Numeric vector. Thresholds of the piecewise Pareto distribution.
`truncation`	Numeric. If `truncation` is not `NULL` and `truncation` > `max(t)`, then the distribution is truncated at `truncation`.
`truncation_type`	Character. If `truncation_type` = `"wd"` then the whole distribution is truncated. If `truncation_type` = `"lp"` then a truncated Pareto is used for the last piece.
`weights`	Numeric vector. Weights for the losses. For instance `weights[i]` = `2` and `weights[j]` = `1` for `j` != `i` has the same effect as adding another loss of size `loss[i]`.
`tol`	Numeric. Desired accuracy (only relevant in the truncated case).
`max_iterations`	Numeric. Maximum number of iteration in the case `truncation` < `Inf` (only relevant in the truncated case).

Value

Maximum likelihood estimator for the parameter alpha of a Pareto distribution with threshold t given the observations losses

Examples

```
losses <- rPiecewisePareto(10000, t = c(100,200,300), alpha = c(1,2,3))
PiecewisePareto_ML_Estimator_Alpha(losses, c(100,200,300))
losses <- rPiecewisePareto(10000, t = c(100,200,300), alpha = c(1,2,3),
                           truncation = 500, truncation_type = "lp")
PiecewisePareto_ML_Estimator_Alpha(losses, c(100,200,300))
PiecewisePareto_ML_Estimator_Alpha(losses, c(100,200,300),
                                   truncation = 500, truncation_type = "lp")
losses <- rPiecewisePareto(10000, t = c(100,200,300), alpha = c(1,2,3),
                           truncation = 500, truncation_type = "wd")
PiecewisePareto_ML_Estimator_Alpha(losses, c(100,200,300))
PiecewisePareto_ML_Estimator_Alpha(losses, c(100,200,300),
                                   truncation = 500, truncation_type = "wd")

losses <- c(140, 240, 490, 200, 110, 710, 120, 190, 210, 310)
w <- rep(1, length(losses))
w[1] <- 2
losses2 <- c(losses, losses[1])
PiecewisePareto_ML_Estimator_Alpha(losses, c(100,200,300), weights = w)
PiecewisePareto_ML_Estimator_Alpha(losses2, c(100,200,300))
```

PiecewisePareto_PDF *Density of the Piecewise Pareto Distribution*

Description

Calculates the density function of the piecewise Pareto distribution. This function is deprecated. Use dPiecewisePareto instead.

Usage

```
PiecewisePareto_PDF(x, t, alpha, truncation = NULL, truncation_type = "lp")
```

Arguments

x	Numeric. The function evaluates the density at x.
t	Numeric vector. Thresholds of the piecewise Pareto distribution.
alpha	Numeric vector. alpha[i] is the Pareto alpha in excess of t[i].
truncation	Numeric. If truncation is not NULL and truncation > t, then the distribution is truncated at truncation.
truncation_type	Character. If truncation_type = "wd" then the whole distribution is truncated. If truncation_type = "lp" then a truncated Pareto is used for the last piece.

Value

Density function of the piecewise Pareto distribution with parameter vectors t and alpha evaluated at x

Examples

```
t <- c(1000, 2000, 3000)
alpha <- c(1, 1.5, 2)
x <- 0:10 * 1000
dPiecewisePareto(x, t, alpha)
dPiecewisePareto(x, t, alpha, truncation = 5000, truncation_type = "lp")
dPiecewisePareto(x, t, alpha, truncation = 5000, truncation_type = "wd")
```

---

pPareto — *Distribution Function of the Pareto Distribution*

Description

Calculates the cumulative distribution function of a Pareto distribution

Usage

```
pPareto(x, t, alpha, truncation = NULL)
```

Arguments

x	Numeric. The function evaluates the CDF at x.
t	Numeric. Threshold of the Pareto distribution.
alpha	Numeric. Pareto alpha.
truncation	Numeric. If truncation is not NULL and truncation > t, then the Pareto distribution is truncated at truncation.

Value

Distribution function of the Pareto distribution with parameters t and alpha evaluated at x

Examples

```
x <- 0:10 * 1000
pPareto(x, 1000, 2)
pPareto(x, 1000, 2, truncation = 5000)
```

pPiecewisePareto — *Distribution Function of the Piecewise Pareto Distribution*

Description

Calculates the cumulative distribution function of a Piecewise Pareto Distribution

Usage

```
pPiecewisePareto(x, t, alpha, truncation = NULL, truncation_type = "lp")
```

Arguments

`x`	Numeric. The function evaluates the CDF at `x`.
`t`	Numeric vector. Thresholds of the piecewise Pareto distribution.
`alpha`	Numeric vector. `alpha[i]` is the Pareto alpha in excess of `t[i]`.
`truncation`	Numeric. If `truncation` is not `NULL` and `truncation` > `t`, then the distribution is truncated at `truncation`.
`truncation_type`	Character. If `truncation_type` = `"wd"` then the whole distribution is truncated. If `truncation_type` = `"lp"` then a truncated Pareto is used for the last piece.

Value

Distribution function of the piecewise Pareto distribution with parameter vectors `t` and `alpha` evaluated at `x`

References

Riegel, U. (2018) Matching tower information with piecewise Pareto. European Actuarial Journal 8(2): 437–460

Examples

```
t <- c(1000, 2000, 3000)
alpha <- c(1, 1.5, 2)
x <- 0:10 * 1000
pPiecewisePareto(x, t, alpha)
pPiecewisePareto(x, t, alpha, truncation = 5000, truncation_type = "lp")
pPiecewisePareto(x, t, alpha, truncation = 5000, truncation_type = "wd")
```

PPP_Model *PPP_Model (Collective Panjer & Piecewise Pareto Model) Object*

Description

Constructor function for the PPP_Model object

Usage

```
PPP_Model(
  FQ = NULL,
  t = NULL,
  alpha = NULL,
  truncation = NULL,
  truncation_type = "lp",
  dispersion = 1,
  Status = 0,
  Comment = "OK"
)
```

Arguments

FQ	Numerical. Expected claim count of the collective model.
t	Numeric vector. Vector containing the thresholds of the Piecewise Pareto distribution.
alpha	Numeric vector. Vector containing the alphas of the Piecewise Pareto distribution.
truncation	Numerical. If truncation is not NULL and truncation > max(t), then the distribution is truncated at truncation.
truncation_type	Character. If truncation_type = "wd" then the whole distribution is truncated. If truncation_type = "lp" then a truncated Pareto is used for the last piece.
dispersion	Numerical. Dispersion of the Panjer distribution (i.e. variance to mean ratio).
Status	Numerical indicator if a function returns a PPP_Model object: 0 = success, 1 = some information has been ignored, 2 = no solution found
Comment	Charakter. An optional comment.

Examples

```
PPPM <- PPP_Model(2, c(1000,2000), c(1,2), dispersion = 2)
PPPM
```

PPP_Model_Excess_Frequency

Expected Frequency in Excess of a Threshold

Description

Calculates the expected frequency in excess of a threshold for a PPP_Model

Usage

```
PPP_Model_Excess_Frequency(x, PPP_Model)
```

Arguments

`x`	Numeric. Threshold.
`PPP_Model`	PPP_Model object.

Value

The expected frequency in excess of `x` for the given `PPP_Model`

Examples

```
PPPM <- PiecewisePareto_Match_Layer_Losses(Example1_AP, Example1_EL)
PPPM
Excess_Frequency(PPPM, c(-Inf, 0, 1000, 2000, 3000, Inf))
```

PPP_Model_Exp_Layer_Loss

Expected Loss of a Reinsurance Layer

Description

Calculates the expected loss of a reinsurance layer for a PPP_Model. This function is deprecated. Use `Layer_Mean` instead.

Usage

```
PPP_Model_Exp_Layer_Loss(Cover, AttachmentPoint, PPP_Model)
```

Arguments

`Cover`	Numeric. Cover of the reinsurance layer. Use `Inf` for unlimited layers.
`AttachmentPoint`	Numeric. Attachment point of the reinsurance layer.
`PPP_Model`	PPP_Model object.

Value

The expected loss of the layer `Cover` xs `AttachmentPoint` for the given `PPP_Model`

Examples

```
PPPM <- PiecewisePareto_Match_Layer_Losses(Example1_AP, Example1_EL)
PPPM
Example1_Cov <- c(diff(Example1_AP), Inf)
Example1_AP
Example1_Cov
Example1_EL
Layer_Mean(PPPM, Example1_Cov, Example1_AP)
```

PPP_Model_Layer_Sd *Standard Deviation of a Reinsurance Layer*

Description

Calculates the standard deviation of the loss in a reinsurance layer for a PPP_Model

Usage

```
PPP_Model_Layer_Sd(Cover, AttachmentPoint, PPP_Model)
```

Arguments

`Cover`	Numeric. Cover of the reinsurance layer. Use `Inf` for unlimited layers.
`AttachmentPoint`	Numeric. Attachment point of the reinsurance layer.
`PPP_Model`	PPP_Model object.

Value

The standard deviation of the loss in the layer `Cover` xs `AttachmentPoint` for the given `PPP_Model`

Examples

```
PPPM <- PiecewisePareto_Match_Layer_Losses(Example1_AP, Example1_EL)
PPPM
Example1_Cov <- c(diff(Example1_AP), Inf)
Layer_Sd(PPPM, Example1_Cov, Example1_AP)
```

PPP_Model_Layer_Var *Variance of a Reinsurance Layer*

Description

Calculates the variance of the loss in a reinsurance layer for a PPP_Model. This function is deprecated. Use `Layer_Var` instead.

Usage

```
PPP_Model_Layer_Var(Cover, AttachmentPoint, PPP_Model)
```

Arguments

`Cover`	Numeric. Cover of the reinsurance layer. Use `Inf` for unlimited layers.
`AttachmentPoint`	Numeric. Attachment point of the reinsurance layer.
`PPP_Model`	PPP_Model object.

Value

The variance of the loss in the layer `Cover` xs `AttachmentPoint` for the given `PPP_Model`

Examples

```
PPPM <- PiecewisePareto_Match_Layer_Losses(Example1_AP, Example1_EL)
PPPM
Example1_Cov <- c(diff(Example1_AP), Inf)
Layer_Var(PPPM, Example1_Cov, Example1_AP)
```

PPP_Model_Simulate *Simulate Losses with a PPP_Model*

Description

Simulates losses of a PPP_Model

Usage

```
PPP_Model_Simulate(n, PPP_Model)
```

Arguments

`n`	Integer. Number of Simulations.
`PPP_Model`	PPP_Model object.

Value

A matrix where row k contains the simulated losses of the kth simulation.

Examples

```
PPPM <- PiecewisePareto_Match_Layer_Losses(c(1000, 2000, 3000), c(2000, 1000, 500),
                                            truncation = 10000, truncation_type = "wd")
PPPM
Simulate_Losses(PPPM, 100)
```

`print.PGP_Model` *Print a PGP_Model Object(Collective Panjer & Generalized Pareto Model) Object*

Description

Print method for PGP_Model objects

Usage

```
## S3 method for class 'PGP_Model'
print(x, ...)
```

Arguments

x	PGP_Model object.
...	Other arguments, all currently ignored.

`print.PPP_Model` *Print a PPP_Model Object(Collective Panjer & Piecewise Pareto Model) Object*

Description

Print method for PPP_Model objects

Usage

```
## S3 method for class 'PPP_Model'
print(x, ...)
```

Arguments

x	PPP_Model object.
...	Other arguments, all currently ignored.

qGenPareto *Quantile Function of the generalized Pareto Distribution*

Description

Calculates the quantile function of a generalized Pareto distribution

Usage

```
qGenPareto(p, t, alpha_ini, alpha_tail, truncation = NULL)
```

Arguments

p	Numeric. The function evaluates the inverse CDF at p.
t	Numeric. Threshold of the piecewise Pareto distribution.
alpha_ini	Numeric. Initial Pareto alpha.
alpha_tail	Numeric. Tail Pareto alpha.
truncation	Numeric. If truncation is not NULL and truncation > t, then the generalized Pareto distribution is truncated at truncation.

Value

Quantile function of the Pareto distribution with parameters t, alpha_ini and alpha_tail, evaluated at p

Examples

```
p <- 0:10 * 0.1
qGenPareto(p, 1000, 2, 3)
qGenPareto(p, 1000, 2, 3, truncation = 5000)
```

qPareto *Quantile Function of the Pareto Distribution*

Description

Calculates the quantile function of a Pareto distribution

Usage

```
qPareto(p, t, alpha, truncation = NULL)
```

Arguments

p	Numeric. The function evaluates the inverse CDF at p.
t	Numeric. Threshold of the piecewise Pareto distribution.
alpha	Numeric. Pareto alpha.
truncation	Numeric. If truncation is not NULL and truncation > t, then the Pareto distribution is truncated at truncation.

Value

Quantile function of the Pareto distribution with parameters t and alpha, evaluated at p

Examples

```
p <- 0:10 * 0.1
qPareto(p, 1000, 2)
qPareto(p, 1000, 2, truncation = 5000)
```

---

qPiecewisePareto *Quantile Function of the Piecewise Pareto Distribution*

Description

Calculates the quantile function of a piecewise Pareto distribution

Usage

```
qPiecewisePareto(p, t, alpha, truncation = NULL, truncation_type = "lp")
```

Arguments

p	Numeric. The function evaluates the quantile function at p.
t	Numeric vector. Thresholds of the piecewise Pareto distribution.
alpha	Numeric vector. alpha[i] is the Pareto alpha in excess of t[i].
truncation	Numeric. If truncation is not NULL and truncation > t, then the distribution is truncated at truncation.
truncation_type	Character. If truncation_type = "wd" then the whole distribution is truncated. If truncation_type = "lp" then a truncated Pareto is used for the last piece.

Value

Quantile function of the piecewise Pareto distribution with parameter vectors t and alpha evaluated at p

Examples

```
t <- c(1000, 2000, 3000)
alpha <- c(1, 1.5, 2)
p <- 0:10 * 0.1
qPiecewisePareto(p, t, alpha)
qPiecewisePareto(p, t, alpha, truncation = 5000, truncation_type = "lp")
qPiecewisePareto(p, t, alpha, truncation = 5000, truncation_type = "wd")
```

---

rGenPareto — *Simulation of the generalized Pareto Distribution*

Description

Generates random deviates of a generalized Pareto distribution

Usage

```
rGenPareto(n, t, alpha_ini, alpha_tail, truncation = NULL)
```

Arguments

n	Numeric. Number of observations.
t	Numeric vector. Thresholds of the generalized Pareto distributions
alpha_ini	Numeric vector. Initial Pareto alphas of the generalized Pareto distributions.
alpha_tail	Numeric vector. Tail Pareto alphas of the generalized Pareto distributions.
truncation	NULL or Numeric vector. If truncation is not NULL and truncation > t, then the generalized Pareto distributions are truncated at truncation (resampled generalized Pareto)

Value

A vector of n samples from the (truncated) generalized Pareto distribution with parameters t, alpha_ini and alpha_tail

Examples

```
rGenPareto(100, 1000, 2, 3)
rGenPareto(100, 1000, 2, 3, truncation = 2000)
rGenPareto(100, t = c(1, 10, 100, 1000), alpha_ini = 1, alpha_tail = c(2, 5))
```

rPareto *Simulation of the Pareto Distribution*

Description

Generates random deviates of a Pareto distribution

Usage

```
rPareto(n, t, alpha, truncation = NULL)
```

Arguments

n	Numeric. Number of observations.
t	Numeric vector. Thresholds of the Pareto distributions
alpha	Numeric vector. Pareto alphas of the Pareto distributions.
truncation	NULL or Numeric vector. If `truncation` is not `NULL` and `truncation` > `t`, then the Pareto distribution is truncated at `truncation` (resampled Pareto)

Value

A vector of `n` samples from the (truncated) Pareto distribution with parameters `t` and `alpha`

Examples

```
rPareto(100, 1000, 2)
rPareto(100, 1000, 2, truncation = 2000)
rPareto(100, t = c(1, 10, 100, 1000, 10000), alpha = c(1, 2, 4, 8, 16))
```

rPiecewisePareto *Simulation of the Piecewise Pareto Distribution*

Description

Generates random deviates of a piecewise Pareto distribution

Usage

```
rPiecewisePareto(
  n,
  t,
  alpha,
  truncation = NULL,
  truncation_type = "lp",
  scale_pieces = NULL
)
```

Arguments

- `n` — Numeric. Number of simulations
- `t` — Numeric vector. Thresholds of the piecewise Pareto distribution.
- `alpha` — Numeric vector. `alpha[i]` is the Pareto alpha in excess of `t[i]`.
- `truncation` — Numeric. If `truncation` is not `NULL` and `truncation` > `t`, then the distribution is truncated at `truncation`.
- `truncation_type` — Character. If `truncation_type` = `"wd"` then the whole distribution is truncated. If `truncation_type` = `"lp"` then a truncated Pareto is used for the last piece.
- `scale_pieces` — Numeric vector. If not `NULL` then the density of the i-th Pareto piece (on the interval (`t[i]`,`t[i+1]`)) is scaled with the factor `const` * `scale_pieces[i]` (where `const` is a normalization constant)

Value

A vector of `n` samples from the (truncated) piecewise Pareto distribution with parameter vectors `t` and `alpha`

Examples

```
t <- c(1000, 2000, 3000)
alpha <- c(1, 1.5, 2)
rPiecewisePareto(100, t, alpha)
rPiecewisePareto(100, t, alpha, truncation = 5000)
rPiecewisePareto(100, t, alpha, truncation = 5000, truncation_type = "lp")
rPiecewisePareto(100, t, alpha, truncation = 5000, truncation_type = "wd")
```

`Simulate_Losses` *Simulate Losses with a Collective Model*

Description

Simulates losses with a collective model

Usage

```
Simulate_Losses(CollectiveModel, nyears = 1)
```

Arguments

- `CollectiveModel` — A collective model object. Currently only `PPP_Models` are handled.
- `nyears` — Integer. Number of simulated years.

Value

A matrix where row k contains the simulated losses of the kth simulated year.

Examples

```
PPPM <- PiecewisePareto_Match_Layer_Losses(c(1000, 2000, 3000), c(2000, 1000, 500),
                                           truncation = 10000, truncation_type = "wd")
PPPM
Simulate_Losses(PPPM, 100)
```

Simulate_Losses.PGP_Model

Simulate Losses with a PGP_Model

Description

Simulates losses with a PGP_Model

Usage

```
## S3 method for class 'PGP_Model'
Simulate_Losses(CollectiveModel, nyears = 1)
```

Arguments

`CollectiveModel`
PGP_Model object.

`nyears` Integer. Number of simulated years.

Value

A matrix where row k contains the simulated losses of the kth simulated year.

Examples

```
PGPM <- PGP_Model(2, 1000, 1, 2, dispersion = 2)
PGPM
Simulate_Losses(PGPM, 100)
```

Simulate_Losses.PPP_Model
Simulate Losses with a PPP_Model

Description

Simulates losses with a PPP_Model

Usage

```
## S3 method for class 'PPP_Model'
Simulate_Losses(CollectiveModel, nyears = 1)
```

Arguments

`CollectiveModel`	PPP_Model object.
`nyears`	Integer. Number of simulated years.

Value

A matrix where row k contains the simulated losses of the kth simulated year.

Examples

```
PPPM <- PiecewisePareto_Match_Layer_Losses(c(1000, 2000, 3000), c(2000, 1000, 500),
                                           truncation = 10000, truncation_type = "wd")
PPPM
Simulate_Losses(PPPM, 100)
```

Index