Package ‘Pareto’

March 3, 2021

Type Package

Title The Pareto, Piecewise Pareto and Generalized Pareto Distribution

Version 2.4.2

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 7.1.1

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 2021-03-03 09:30:02 UTC

R topics documented:

  dGenPareto ....................................................... 3
  dPareto .......................................................... 4
  dPiecewisePareto ................................................ 4
  Example1_AP ..................................................... 5
<table>
<thead>
<tr>
<th>R topics documented</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example_EL</td>
<td>6</td>
</tr>
<tr>
<td>Excess_Frequency</td>
<td>6</td>
</tr>
<tr>
<td>Excess_Frequency.PGP_Model</td>
<td>7</td>
</tr>
<tr>
<td>Excess_Frequency.PPP_Model</td>
<td>7</td>
</tr>
<tr>
<td>Fit_PML_Curve</td>
<td>8</td>
</tr>
<tr>
<td>Fit_References</td>
<td>9</td>
</tr>
<tr>
<td>GenPareto_Layer_Mean</td>
<td>12</td>
</tr>
<tr>
<td>GenPareto_Layer_SM</td>
<td>13</td>
</tr>
<tr>
<td>GenPareto_Layer_Var</td>
<td>14</td>
</tr>
<tr>
<td>GenPareto_ML_Estimator_Alpha</td>
<td>15</td>
</tr>
<tr>
<td>is.PGP_Model</td>
<td>16</td>
</tr>
<tr>
<td>is.PPP_Model</td>
<td>17</td>
</tr>
<tr>
<td>is.valid.PGP_Model</td>
<td>18</td>
</tr>
<tr>
<td>is.valid.PPP_Model</td>
<td>18</td>
</tr>
<tr>
<td>Layer_Mean</td>
<td>19</td>
</tr>
<tr>
<td>Layer_Mean.PGP_Model</td>
<td>20</td>
</tr>
<tr>
<td>Layer_Mean.PPP_Model</td>
<td>20</td>
</tr>
<tr>
<td>Layer_Sd</td>
<td>21</td>
</tr>
<tr>
<td>Layer_Sd.PGP_Model</td>
<td>22</td>
</tr>
<tr>
<td>Layer_Sd.PPP_Model</td>
<td>23</td>
</tr>
<tr>
<td>Layer_Var</td>
<td>23</td>
</tr>
<tr>
<td>Layer_Var.PGP_Model</td>
<td>24</td>
</tr>
<tr>
<td>Layer_Var.PPP_Model</td>
<td>25</td>
</tr>
<tr>
<td>Local_Pareto_Alpha</td>
<td>26</td>
</tr>
<tr>
<td>Pareto_CDF</td>
<td>27</td>
</tr>
<tr>
<td>Pareto_Extrapolation</td>
<td>27</td>
</tr>
<tr>
<td>Pareto_Find_Alpha_btw_FQs</td>
<td>29</td>
</tr>
<tr>
<td>Pareto_Find_Alpha_btw_FQ_Layer</td>
<td>30</td>
</tr>
<tr>
<td>Pareto_Find_Alpha_btw_Layers</td>
<td>31</td>
</tr>
<tr>
<td>Pareto_Layer_Mean</td>
<td>32</td>
</tr>
<tr>
<td>Pareto_Layer_SM</td>
<td>33</td>
</tr>
<tr>
<td>Pareto_Layer_Var</td>
<td>33</td>
</tr>
<tr>
<td>Pareto_ML_Estimator_Alpha</td>
<td>34</td>
</tr>
<tr>
<td>Pareto_PDF</td>
<td>36</td>
</tr>
<tr>
<td>pGenPareto</td>
<td>36</td>
</tr>
<tr>
<td>PGP_Model</td>
<td>37</td>
</tr>
<tr>
<td>PiecewisePareto_CDF</td>
<td>38</td>
</tr>
<tr>
<td>PiecewisePareto_Layer_Mean</td>
<td>39</td>
</tr>
<tr>
<td>PiecewisePareto_Layer_SM</td>
<td>40</td>
</tr>
<tr>
<td>PiecewisePareto_Layer_Var</td>
<td>41</td>
</tr>
<tr>
<td>PiecewisePareto_Match_Layer_Losses</td>
<td>42</td>
</tr>
<tr>
<td>PiecewisePareto_ML_Estimator_Alpha</td>
<td>45</td>
</tr>
<tr>
<td>PiecewisePareto_PDF</td>
<td>46</td>
</tr>
<tr>
<td>pPareto</td>
<td>47</td>
</tr>
<tr>
<td>pPiecewisePareto</td>
<td>48</td>
</tr>
<tr>
<td>PPP_Model</td>
<td>49</td>
</tr>
<tr>
<td>PPP_Model_Excess_Frequency</td>
<td>50</td>
</tr>
<tr>
<td>PPP_Model_Exp_Layer_Loss</td>
<td>50</td>
</tr>
</tbody>
</table>
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**

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

**Arguments**
- `x`: Numeric. The function evaluates the density at `x`.
- `t`: Numeric. Threshold of the Pareto distribution.
- `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**
```r
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**

```r
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")

Description

Example data: Attachment Points

Usage

Example1_AP

Format

An object of class numeric of length 5.
Excess_Frequency

Example data: Expected Losses

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

```r
## 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

```r
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

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

Arguments

- `CollectiveModel` 
  - PPP_Model object.
- `x` 
  - Numeric. Threshold.
**Fit_PML_Curve**

**Arguments**

- **CollectiveModel**
  - PPP_Model object.

- **x**
  - Numeric. Threshold.

**Value**

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

**Examples**

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

---

**Fit_PML_Curve**

*Fits a Collective Model to a PML Curve*

**Description**

Fits a PPP_Model that matches the values of a PML curve

**Usage**

```r
Fit_PML_Curve(
  return_periods,
  amounts,
  tail_alpha = 2,
  truncation = NULL,
  truncation_type = "lp",
  dispersion = 1
)
```

**Arguments**

- **return_periods**
  - Numeric vector. Vector containing the return periods of the PML curve.

- **amounts**
  - Numeric vector. Vector containing the loss amounts corresponding to the return periods.

- **tail_alpha**
  - Numerical. Pareto alpha that is used above the highest amount of the PML curve.

- **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.

- **dispersion**
  - Numerical. Dispersion of the claim count distribution in the resulting PPP_Model.
Fit References

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

```r
return_periods <- c(1, 5, 10, 20, 50, 100)
amounts <- c(1000, 4000, 7000, 10000, 13000, 14000)

fit <- Fit_PML_Curve(return_periods, amounts)
1 / Excess_Frequency(fit, amounts)

fit <- Fit_PML_Curve(return_periods, amounts, tail_alpha = 1.5,
                      truncation = 20000, truncation_type = "wd")
1 / Excess_Frequency(fit, amounts)
```

Fit References

Fit a Collective Model to a Wishlist of References

Description

The function fits a collective model to a wishlist of references (expected layer losses and excess frequencies). The function allows to specify the family of the severity distribution that is used. Depending on this distribution family the function works slightly differently:

• For the severity distribution PiecewisePareto the function returns a PPP_Model that satisfies all the references
• For the severity distribution Pareto the function returns a PPP_Model that minimizes the squared relative deviations from the references
• For the severity distribution GenPareto the function returns a PGP_Model that minimizes the squared relative deviations from the references
Usage

```r
Fit_References(
  Covers = NULL,
  Attachment_Points = NULL,
  Expected_Layer_Losses = NULL,
  Thresholds = NULL,
  Frequencies = NULL,
  model_threshold = min(c(Attachment_Points, Thresholds)),
  default_alpha = 2,
  dispersion = 1,
  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 wishlist for which excess frequencies are given.
- **Frequencies**: Numeric vector. Expected frequencies excess the Thresholds from the wishlist.
- **model_threshold**: Numerical. Lowest threshold of the fitted piecewise Pareto distribution.
- **default_alpha**: Numerical. Default alpha for situations where an alpha has to be selected.
- **dispersion**: Numerical. Dispersion of the claim count distribution in the resulting PPP_Model.
- **alpha_max**: Numerical. Maximum alpha to be used for the matching.
- **severity_distribution**: Character. Implemented distributions: "PiecewisePareto" (default), "Pareto" and "GenPareto".
- **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

For `severity_distribution = "PiecewisePareto"` or "Pareto": 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
• \( \text{truncation} \) Numerical. If \( \text{truncation} \) is not \( \text{NULL} \) and \( \text{truncation} > \max(t) \), then the distribution is truncated at \( \text{truncation} \).
• \( \text{truncation}\_\text{type} \) Character. If \( \text{truncation}\_\text{type} = "wd" \) then the whole distribution is truncated. If \( \text{truncation}\_\text{type} = "lp" \) then a truncated Pareto is used for the last piece.
• \( \text{dispersion} \) Numerical. Dispersion of the Panjer distribution (i.e. variance to mean ratio).
• \( \text{Status} \) Numerical indicator: 0 = success, 1 = some information has been ignored, 2 = no solution found
• \( \text{Comment} \) Character. Information on whether the fit was successful

For \( \text{severity}\_\text{distribution} = "\text{GenPareto}\" \): A \texttt{PGP}\_\texttt{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:

• \( \text{FQ} \) Expected claim count of the collective model.
• \( t \) Numeric. Threshold of the Pareto distribution.
• \( \alpha\_\text{ini} \) Numeric. Initial Pareto alpha (at \( t \)).
• \( \alpha\_\text{tail} \) Numeric. Tail Pareto alpha.
• \( \text{truncation} \) If \( \text{truncation} \) is not \( \text{NULL} \) and \( \text{truncation} > t \), then the Pareto distribution is truncated at \( \text{truncation} \).
• \( \text{dispersion} \) Numerical. Dispersion of the Panjer distribution (i.e. variance to mean ratio).
• \( \text{Status} \) Numerical indicator: 0 = success, 1 = some information has been ignored, 2 = no solution found
• \( \text{Comment} \) Character. Information on whether the fit was successful

Examples

```r
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)
fit <- Fit_References(covers, att_points, exp_losses, thresholds, fqs, severity_distribution = "Pareto")
Layer_Mean(fit, covers, att_points)
Excess_Frequency(fit, thresholds)
fit <- Fit_References(covers, att_points, exp_losses, severity_distribution = "GenPareto")
Layer_Mean(fit, covers, att_points)
```
**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**

```r
GenPareto_Layer_Mean(
  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 = NULL  # Numeric. If truncation is not NULL and truncation > t, then the Pareto distribution is truncated at truncation.
)
```

**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**

```r
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
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover</td>
<td>Numeric. Cover of the reinsurance layer. Use Inf for unlimited layers.</td>
</tr>
<tr>
<td>AttachmentPoint</td>
<td>Numeric. Attachment point of the reinsurance layer.</td>
</tr>
<tr>
<td>t</td>
<td>Numeric. Threshold of the Pareto distribution. If t is NULL (default) then t&lt;AttachmentPoint is used</td>
</tr>
<tr>
<td>alpha_ini</td>
<td>Numeric. Initial Pareto alpha (at t).</td>
</tr>
<tr>
<td>alpha_tail</td>
<td>Numeric. Tail Pareto alpha.</td>
</tr>
<tr>
<td>truncation</td>
<td>Numeric. If truncation is not NULL and truncation &gt; t, then the Pareto distribution is truncated at truncation.</td>
</tr>
</tbody>
</table>

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**
```r
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 x 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 with known threshold and (if applicable) known truncation.

Usage

GenPareto_ML_Estimator_Alpha(
  losses,
  t,
  truncation = NULL,
  reporting_thresholds = NULL,
  is.censored = NULL,
  weights = NULL,
  alpha_min = 0.001,
  alpha_max = 10
)

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.
reporting_thresholds Numeric vector. Allows to enter loss specific reporting thresholds. If NULL then all reporting thresholds are assumed to be less than or equal to t.
is.censored Logical vector. TRUE indicates that a loss has been censored by the policy limit. The assumption is that the uncensored losses are Generalized Pareto distributed with the alphas we are looking for. is.censored = NULL means that no losses are censored.
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].
alpha_min Numeric. Lower bound for the estimated alphas.
alpha_max Numeric. Upper bound for the estimated alphas.
is.PGP_Model

Check if an object is a PGP_Model

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

```r
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)

# Set threshold and alpha parameters

t <- 1000
alpha_ini <- 1
alpha_tail <- 3

# Generate losses
losses <- rGenPareto(5000, t, alpha_ini, alpha_tail)
reported_thresholds <- rPareto(5000, 1000, 3)
reported <- losses > reported_thresholds
losses <- losses[reported]
reported_thresholds <- reported_thresholds[reported]
GenPareto_ML_Estimator_Alpha(losses, t)
GenPareto_ML_Estimator_Alpha(losses, t, reporting_thresholds = reported_thresholds)

limit <- 3000
censored <- losses > limit
losses[censored] <- limit
reported <- losses > reported_thresholds
losses <- losses[reported]
censored <- censored[reported]
reported_thresholds <- reported_thresholds[reported]
GenPareto_ML_Estimator_Alpha(losses, t, reporting_thresholds = reported_thresholds)
GenPareto_ML_Estimator_Alpha(losses, t, reporting_thresholds = reported_thresholds, is.censored = censored)

# Additional examples

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)
is.PPP_Model

Arguments

x Object to be checked.

Examples

```r
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

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

Arguments

x Object to be checked.

Examples

```r
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**

```r
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**

```r
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**

```r
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**

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

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

Examples

```r
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

```r
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

```r
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  

**Description**

Calculates the expected loss of a reinsurance layer for a PGP_Model

**Usage**

```r
## 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 \( x \) AttachmentPoint for the given CollectiveModel

**Examples**

```r
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  

**Description**

Calculates the expected loss of a reinsurance layer for a PPP_Model

**Usage**

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

**Examples**

```r
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_Sd

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
Layer_Sd.PGP_Model

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**

```r
## 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**

```r
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**

```r
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
Layer_Var.PPP_Model

Examples

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

Description

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

Usage

```r
## 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

```r
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

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)
  - 'weibull' for weibull distribution (arguments: shape, scale)
  - 'exp' for exponential distribution (arguments: rate)
  - 'Pareto' for Pareto distribution (arguments: t, alpha, truncation = NULL)
  - 'GenPareto' for exp distribution (arguments: t, alpha_ini, alpha_tail, truncation = NULL)
  - 'PiecewisePareto' for exp distribution (arguments: t, alpha, truncation = NULL, truncation_type = 'lp')
- **...**: Arguments for the selected distribution

Value
Local Pareto alpha of the selected distribution at x

References

Examples
```r
x <- 1:10 * 1e6
Local_Pareto_Alpha(x, "norm", mean = 5e6, sd = 2e6)
Local_Pareto_Alpha(x, "lnorm", meanlog = 0, sdlog = 4)
Local_Pareto_Alpha(x, "gamma", shape = 5, rate = 1e-6)
Local_Pareto_Alpha(x, "weibull", shape = 0.5, scale = 1e6)
Local_Pareto_Alpha(x, "exp", rate = 1e-6)
Local_Pareto_Alpha(x, "Pareto", t = 1e6, alpha = 1, truncation = 20e6)
Local_Pareto_Alpha(x, "GenPareto", t = 1e6, alpha_ini = 1, alpha_tail = 2)
Local_Pareto_Alpha(x, "PiecewisePareto", t = c(1e6, 3e6, 5e6), alpha = c(1, 2, 3),
  truncation = 20e6, truncation_type = "wd")
```
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**

```r
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**

```r
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


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


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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threshold</td>
<td>Numeric. Threshold</td>
</tr>
<tr>
<td>Frequency</td>
<td>Numeric. Expected frequency in excess of Threshold</td>
</tr>
<tr>
<td>Cover</td>
<td>Numeric. Cover of the second layer.</td>
</tr>
<tr>
<td>AttachmentPoint</td>
<td>Numeric. Attachment point of the layer.</td>
</tr>
<tr>
<td>ExpLoss</td>
<td>Numeric. Expected loss of the layer.</td>
</tr>
<tr>
<td>max_alpha</td>
<td>Numeric. Upper limit for the alpha that is returned.</td>
</tr>
<tr>
<td>tolerance</td>
<td>Numeric. Accuracy of the result.</td>
</tr>
<tr>
<td>truncation</td>
<td>Numeric. If truncation is not NULL then the Pareto distribution is truncated at truncation.</td>
</tr>
</tbody>
</table>

Value

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

References

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)

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


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

**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)`
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 \( t \) \(
\leq\) 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 \times 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 Alpha of a Pareto distribution*

Description

Calculates the maximum likelihood estimator for the parameter \( \alpha \) of a Pareto distribution with a known threshold and (if applicable) a known truncation.

Usage

Pareto_ML_Estimator_Alpha(
    losses,
    t,
    truncation = NULL,
    reporting_thresholds = NULL,
    is.censored = NULL,
    weights = NULL,
    alpha_min = 0.001,
    alpha_max = 10
)
**Arguments**

- **losses**: Numeric vector. Losses that are used for the ML estimation.
- **t**: Numeric. Threshold of the Pareto distribution.
- **truncation**: Numeric. If truncation is not NULL, then the Pareto distribution is truncated at truncation.
- **reporting_thresholds**: Numeric vector. Allows to enter loss specific reporting thresholds. If NULL then all reporting thresholds are assumed to be less than or equal to t.
- **is.censored**: Logical vector. TRUE indicates that a loss has been censored by the policy limit. The assumption is that the uncensored losses are Pareto distributed with the alpha we are looking for. is.censored = NULL means that no losses are censored.
- **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].
- **alpha_min**: Numeric. Lower bound for alpha (only used in truncated case).
- **alpha_max**: Numeric. Upper bound for alpha (only used in truncated case).

**Value**

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

**Examples**

```r
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 <- 100
alpha <- 2
losses <- rPareto(10000, t, alpha)
reporting_thresholds <- rPareto(10000, t, 5)
index <- losses > reporting_thresholds
losses <- losses[index]
reporting_thresholds <- reporting_thresholds[index]
Pareto_ML_Estimator_Alpha(losses, t)
Pareto_ML_Estimator_Alpha(losses, t, reporting_thresholds = reporting_thresholds)

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

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

Description

Constructor function for the PGP_Model object

Usage

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

**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**
  Character. An optional comment.

**Examples**

```r
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**

```r
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


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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cover</td>
<td>Numeric. Cover of the reinsurance layer.</td>
</tr>
<tr>
<td>AttachmentPoint</td>
<td>Numeric. Attachment point of the reinsurance layer.</td>
</tr>
<tr>
<td>t</td>
<td>Numeric vector. Thresholds of the piecewise Pareto distribution.</td>
</tr>
<tr>
<td>alpha</td>
<td>Numeric vector. ( \alpha[i] ) is the Pareto alpha in excess of ( t[i] ).</td>
</tr>
<tr>
<td>truncation</td>
<td>Numeric. If truncation is not NULL and truncation &gt; t, then the Pareto distribution is truncated at truncation.</td>
</tr>
</tbody>
</table>
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

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.
**PiecewisePareto_Layer_Var**

- **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**

```r
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")
```

---

**Description**

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

**Usage**

```r
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
```r
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
```r
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 $1 - \text{RoL}\_\text{tolerance}$.

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:

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

References

Examples

```r
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)
```
Description

Calculates the maximum likelihood estimator of the parameter vector alpha for a piecewise Pareto distribution with given vector t and (if applicable) a known truncation.

Usage

```r
PiecewisePareto_ML_Estimator_Alpha(
  losses,
  t,
  truncation = NULL,
  truncation_type = "lp",
  reporting_thresholds = NULL,
  is.censored = NULL,
  weights = NULL,
  alpha_min = 0.001,
  alpha_max = 10
)
```

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.
- **reporting_thresholds**: Numeric vector. Allows to enter loss specific reporting thresholds. If NULL then all reporting thresholds are assumed to be less than or equal to t[1].
- **is.censored**: Logical vector. TRUE indicates that a loss has been censored by the policy limit. The assumption is that the uncensored losses are piecewise Pareto distributed with the alphas we are looking for. is.censored = NULL means that no losses are censored.
- **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].
- **alpha_min**: Numeric. Lower bound for the estimated alphas (only used in truncated case).
- **alpha_max**: Numeric. Upper bound for the estimated alphas (only used in truncated case).
Value

Maximum likelihood estimator for the parameter alpha of a Pareto distribution with threshold \( t \) given the observations \( \text{losses} \).

Examples

```r
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 = "wd")
PiecewisePareto_ML_Estimator_Alpha(losses, c(100,200,300),
   truncation = 500, truncation_type = "wd")
reporting_thresholds <- rPareto(10000, 100, 3)
index <- losses > reporting_thresholds
losses <- losses[index]
reporting_thresholds <- reporting_thresholds[index]
PiecewisePareto_ML_Estimator_Alpha(losses, c(100,200,300),
   truncation = 500, truncation_type = "wd")
PiecewisePareto_ML_Estimator_Alpha(losses, c(100,200,300),
   truncation = 500, truncation_type = "wd",
   reporting_thresholds = reporting_thresholds)

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**

```r
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

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

---

**Description**

Calculates the cumulative distribution function of a Piecewise Pareto Distribution

**Usage**

```r
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**


**Examples**

```r
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 (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**  
  Character. An optional comment.

**Examples**

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

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.
PPP_Model_Layer_Sd

Value

The expected loss of the layer \( \text{Cover} \times \text{AttachmentPoint} \) for the given \( \text{PPP_Model} \).

Examples

```r
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)
```

Description

Calculates the standard deviation of the loss in a reinsurance layer for a \( \text{PPP_Model} \).

Usage

```r
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**
  \( \text{PPP_Model} \) object.

Value

The standard deviation of the loss in the layer \( \text{Cover} \times \text{AttachmentPoint} \) for the given \( \text{PPP_Model} \).

Examples

```r
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**

```r
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**

```r
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**

```r
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

```r
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**

```r
## 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**

```r
## 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
t <- 1000
qGenPareto(p, t, alpha_ini = 2, alpha_tail = 3)
qGenPareto(p, t, alpha_ini = 2, alpha_tail = 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)
qPiecewisePareto

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

```r
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

```r
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
)

Simulate Losses

Simulate Losses with a Collective Model

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.
**Simulate_Losses.PGP_Model**

**Value**

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

**Examples**

```r
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**

```r
## 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**

```r
PGPM <- PGP_Model(2, 1000, 1, 2, dispersion = 2)
PGPM
Simulate_Losses(PGPM, 100)
```
Simulate Losses with a PPP_Model

Description
Simulates losses with a PPP_Model

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

* datasets
  Example1_AP, 5
  Example1_EL, 6

  dGenPareto, 3
  dPareto, 4
  dPiecewisePareto, 4

  Example1_AP, 5
  Example1_EL, 6
  Excess_Frequency, 6
  Excess_Frequency.PGP_Model, 7
  Excess_Frequency.PPP_Model, 7

  Fit_PML_Curve, 8
  Fit_References, 9

  GenPareto_Layer_Mean, 12
  GenPareto_Layer_SM, 13
  GenPareto_Layer_Var, 14
  GenPareto_ML_Estimator_Alpha, 15

  is.PGP_Model, 16
  is.PPP_Model, 17
  is.valid.PGP_Model, 18
  is.valid.PPP_Model, 18

  Layer_Mean, 19
  Layer_Mean.PGP_Model, 20
  Layer_Mean.PPP_Model, 20
  Layer_Sd, 21
  Layer_Sd.PGP_Model, 22
  Layer_Sd.PPP_Model, 23
  Layer_Var, 23
  Layer_Var.PGP_Model, 24
  Layer_Var.PPP_Model, 25
  Local_Pareto_Alpha, 26

  Pareto_CDF, 27
  Pareto_Extrapolation, 27
  Pareto_Find_Alpha_btw_FQ_Layer, 30

  Pareto_Find_Alpha_btw_FQs, 29
  Pareto_Find_Alpha_btw_Layers, 31
  Pareto_Layer_Mean, 32
  Pareto_Layer_SM, 33
  Pareto_Layer_Var, 33
  Pareto_ML_Estimator_Alpha, 34
  Pareto_PDF, 36
  pGenPareto, 36
  PGP_Model, 37
  PiecewisePareto_CDF, 38
  PiecewisePareto_Layer_Mean, 39
  PiecewisePareto_Layer_SM, 40
  PiecewisePareto_Layer_Var, 41
  PiecewisePareto_Match_Layer_Losses, 42
  PiecewisePareto_ML_Estimator_Alpha, 45
  PiecewisePareto_PDF, 46
  pPareto, 47
  pPiecewisePareto, 48
  PPP_Model, 49
  PPP_Model_Excess_Frequency, 50
  PPP_Model_Exp_Layer_Loss, 50
  PPP_Model_Layer_Sd, 51
  PPP_Model_Layer_Var, 52
  PPP_Model_Simulate, 52
  print.PGP_Model, 53
  print.PPP_Model, 53

  qGenPareto, 54
  qPareto, 54
  qPiecewisePareto, 55

  rGenPareto, 56
  rPareto, 57
  rPiecewisePareto, 57

  Simulate_Losses, 58
  Simulate_Losses.PGP_Model, 59
  Simulate_Losses.PPP_Model, 60

61