Package ‘ReIns’

January 20, 2025

Version 1.0.15

Date 2024-11-30

Title Functions from ``Reinsurance: Actuarial and Statistical Aspects''

Description Functions from the book ``Reinsurance: Actuarial and Statistical Aspects'' (2017) by Hansjoerg Albrecher, Jan Beirlant and Jef Teugels <https://www.wiley.com/en-us/Reinsurance%3A+Actuarial+and+Statistical+Aspects-p-9780470772683>.

Maintainer Tom Reynkens <tomreynkens.r@gmail.com>

Depends R (>= 3.4)

Imports stats, graphics, methods, survival, utils, parallel, foreach (>= 1.4.1), doParallel (>= 1.0.10), Rcpp (>= 0.12.12)

Suggests testthat, knitr, rmarkdown, interval, Icens

LinkingTo Rcpp

URL https://github.com/TReynkens/ReIns

BugReports https://github.com/TReynkens/ReIns/issues

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation yes

ByteCompile yes

Author Tom Reynkens [aut, cre] (<https://orcid.org/0000-0002-5516-5107>),
Roel Verbelen [aut] (R code for Mixed Erlang distribution, <https://orcid.org/0000-0002-2347-9240>),
Anastasios Bardoutsos [ctb] (Original R code for cEPD estimator),
Dries Cornilly [ctb] (Original R code for EVT estimators for truncated data),
Yuri Goegebeur [ctb] (Original S-Plus code for basic EVT estimators),
Klaus Herrmann [ctb] (Original R code for GPD estimator)

Repository CRAN

Date/Publication 2024-12-02 09:30:05 UTC

Contents

Burr . . . 4
cEPD . . . 6
cExpQQ . . . 7
cgenHill . . . 9
cGPDmle . . . 10
cHill . . . 12
cLognormalQQ . . . 14
cMoment . . . 15
cParetoQQ . . . 17
cProb . . . 18
cProbEPD . . . 20
cProbGH . . . 22
cProbGPD . . . 24
cProbMOM . . . 26
cQuant . . . 28
cQuantGH . . . 30
cQuantGPD . . . 32
cQuantMOM . . . 33
crHill . . . 35
crParetoQQ . . . 37
crSurv . . . 39
CTE . . . 41
cWeibullQQ . . . 42
EPD . . . 44
EPDfit . . . 46
EVTfit . . . 47
ExcessEPD . . . 48
ExcessGPD . . . 50
ExcessPareto . . . 51
ExcessSplice . . . 53
ExpQQ . . . 55
Extended Pareto . . . 56
Frechet . . . 57
genHill . . . 59
genQQ . . . 60
GPD . . . 62
GPDfit . . . 63
GPDmle . . . 64
GPDresiduals . . . 66
Hill . . . 67
Hill.2oQV . . . 69
Hill.kopt . . . 70
icHill . . . 72
icParetoQQ . . . 74
KaplanMeier . . . 76
LognormalQQ . . . 77

LognormalQQ_der 79
LStail 80
MeanExcess 82
MeanExcess_TB 83
MEfit 85
Moment 87
norwegianfire 88
Pareto 89
ParetoQQ 90
ParetoQQ_der 92
pClas 93
pEdge 95
pGC 97
Prob 99
ProbEPD 101
ProbGH 103
ProbGPD 104
ProbMOM 106
ProbReg 108
Quant 109
Quant.2oQV 111
QuantGH 113
QuantGPD 114
QuantMOM 116
QuantReg 117
Scale 119
Scale.2o 120
ScaleEPD 122
ScaleReg 124
secura 125
soa 126
Splice 127
SpliceECDF 129
SpliceFit 131
SpliceFitGPD 132
SpliceFiticPareto 135
SpliceFitPareto 137
SpliceLL 140
SpliceLL_TB 142
SplicePP 145
SplicePP_TB 147
SpliceQQ 150
SpliceQQ_TB 152
SpliceTB 154
stdf 157
tBurr 158
tExp 160
tFrechet 161

tGPD . . . 162
tlnorm . . . 164
tPareto . . . 165
trDT . . . 166
trDTMLE . . . 168
trEndpoint . . . 169
trEndpointMLE . . . 171
trHill . . . 172
trMLE . . . 174
trParetoQQ . . . 176
trProb . . . 178
trProbMLE . . . 179
trQuant . . . 181
trQuantMLE . . . 183
trTest . . . 185
trTestMLE . . . 187
Turnbull . . . 188
tWeibull . . . 190
VaR . . . 191
WeibullQQ . . . 192
WeibullQQ_der . . . 194

Index **196**

`Burr` *The Burr distribution*

Description

Density, distribution function, quantile function and random generation for the Burr distribution (type XII).

Usage

```
dburr(x, alpha, rho, eta = 1, log = FALSE)
pburr(x, alpha, rho, eta = 1, lower.tail = TRUE, log.p = FALSE)
qburr(p, alpha, rho, eta = 1, lower.tail = TRUE, log.p = FALSE)
rburr(n, alpha, rho, eta = 1)
```

Arguments

`x`	Vector of quantiles.
`p`	Vector of probabilities.
`n`	Number of observations.
`alpha`	The α parameter of the Burr distribution, a strictly positive number.
`rho`	The ρ parameter of the Burr distribution, a strictly negative number.

`eta`	The η parameter of the Burr distribution, a strictly positive number. The default value is `1`.
`log`	Logical indicating if the densities are given as $\log(f)$, default is `FALSE`.
`lower.tail`	Logical indicating if the probabilities are of the form $P(X \le x)$ (`TRUE`) or $P(X > x)$ (`FALSE`). Default is `TRUE`.
`log.p`	Logical indicating if the probabilities are given as $\log(p)$, default is `FALSE`.

Details

The Cumulative Distribution Function (CDF) of the Burr distribution is equal to $F(x) = 1 - ((\eta + x^{-\rho \times \alpha})/\eta)^{1/\rho}$ for all $x \ge 0$ and $F(x) = 0$ otherwise. We need that $\alpha > 0$, $\rho < 0$ and $\eta > 0$.

Beirlant et al. (2004) uses parameters η, τ, λ which correspond to η, $\tau = -\rho \times \alpha$ and $\lambda = -1/\rho$.

Value

`dburr` gives the density function evaluated in x, `pburr` the CDF evaluated in x and `qburr` the quantile function evaluated in p. The length of the result is equal to the length of x or p.

`rburr` returns a random sample of length n.

Author(s)

Tom Reynkens.

References

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

See Also

`tBurr`, `Distributions`

Examples

```
# Plot of the PDF
x <- seq(0, 10, 0.01)
plot(x, dburr(x, alpha=2, rho=-1), xlab="x", ylab="PDF", type="l")

# Plot of the CDF
x <- seq(0, 10, 0.01)
plot(x, pburr(x, alpha=2, rho=-1), xlab="x", ylab="CDF", type="l")
```

cEPD *EPD estimator for right censored data*

Description

Computes the EPD estimates adapted for right censored data.

Usage

```
cEPD(data, censored, rho = -1, beta = NULL, logk = FALSE,
     plot = FALSE, add = FALSE, main = "EPD estimates of the EVI", ...)
```

Arguments

data	Vector of n observations.
censored	A logical vector of length n indicating if an observation is censored.
rho	A parameter for the ρ-estimator of Fraga Alves et al. (2003) when strictly positive or choice(s) for ρ if negative. Default is -1.
beta	Parameter for EPD ($\beta = -\rho/\gamma$). If NULL (default), beta is estimated by $-\rho/H_{k,n}$ with $H_{k,n}$ the Hill estimator.
logk	Logical indicating if the estimates are plotted as a function of $\log(k)$ (logk=TRUE) or as a function of k. Default is FALSE.
plot	Logical indicating if the estimates of γ should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates of γ should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "EPD estimates of the EVI".
...	Additional arguments for the plot function, see plot for more details.

Details

The function EPD uses τ which is equal to $-\beta$.

This estimator is only suitable for right censored data.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
gamma1	Vector of the corresponding estimates for the γ parameter of the EPD.
kappa1	Vector of the corresponding MLE estimates for the κ parameter of the EPD.
beta	Vector of estimates for (or values of) the β parameter of the EPD.
Delta	Difference between gamma1 and the Hill estimator for censored data.

Author(s)

Tom Reynkens based on R code from Anastasios Bardoutsos.

References

Beirlant, J., Bardoutsos, A., de Wet, T. and Gijbels, I. (2016). "Bias Reduced Tail Estimation for Censored Pareto Type Distributions." *Statistics & Probability Letters*, 109, 78–88.

Fraga Alves, M.I. , Gomes, M.I. and de Haan, L. (2003). "A New Class of Semi-parametric Estimators of the Second Order Parameter." *Portugaliae Mathematica*, 60, 193–214.

See Also

EPD, cProbEPD, cGPDmle

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X, Y)

# Censoring indicator
censored <- (X>Y)

# EPD estimator adapted for right censoring
cepd <- cEPD(Z, censored=censored, plot=TRUE)
```

---

cExpQQ — *Exponential quantile plot for right censored data*

Description

Exponential QQ-plot adapted for right censored data.

Usage

```
cExpQQ(data, censored, plot = TRUE, main = "Exponential QQ-plot", ...)
```

Arguments

`data`	Vector of n observations.
`censored`	A logical vector of length n indicating if an observation is censored.
`plot`	Logical indicating if the quantiles should be plotted in an exponential QQ-plot, default is `TRUE`.
`main`	Title for the plot, default is `"Exponential QQ-plot"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The exponential QQ-plot adapted for right censoring is given by

$$(-\log(1 - F_{km}(Z_{j,n})), Z_{j,n})$$

for $j = 1, \ldots, n-1$, with $Z_{i,n}$ the i-th order statistic of the data and F_{km} the Kaplan-Meier estimator for the CDF. Hence, it has the same empirical quantiles as an ordinary exponential QQ-plot but replaces the theoretical quantiles $-\log(1 - j/(n+1))$ by $-\log(1 - F_{km}(Z_{j,n}))$.

This QQ-plot is only suitable for right censored data.

In Beirlant et al. (2007), only a Pareto QQ-plot adapted for right-censored data is proposed. This QQ-plot is constructed using the same ideas, but is not described in the paper.

Value

A list with following components:

`eqq.the`	Vector of the theoretical quantiles, see Details.
`eqq.emp`	Vector of the empirical quantiles from the data.

Author(s)

Tom Reynkens

References

Beirlant, J., Guillou, A., Dierckx, G. and Fils-Villetard, A. (2007). "Estimation of the Extreme Value Index and Extreme Quantiles Under Random Censoring." *Extremes*, 10, 151–174.

See Also

`ExpQQ`, `cLognormalQQ`, `cParetoQQ`, `cWeibullQQ`, `KaplanMeier`

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
X <- rpareto(500, shape=2)
```

```
# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X, Y)

# Censoring indicator
censored <- (X>Y)

# Exponential QQ-plot adapted for right censoring
cExpQQ(Z, censored=censored)
```

---

cgenHill *Generalised Hill estimator for right censored data*

Description

Computes the generalised Hill estimates adapted for right censored data.

Usage

```
cgenHill(data, censored, logk = FALSE, plot = FALSE, add = FALSE,
         main = "Generalised Hill estimates of the EVI", ...)
```

Arguments

`data`	Vector of n observations.
`censored`	A logical vector of length n indicating if an observation is censored.
`logk`	Logical indicating if the estimates are plotted as a function of $\log(k)$ (`logk=TRUE`) or as a function of k. Default is `FALSE`.
`plot`	Logical indicating if the estimates of γ_1 should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates of γ_1 should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Generalised Hill estimates of the EVI"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The generalised Hill estimator adapted for right censored data is equal to the ordinary generalised Hill estimator divided by the proportion of the k largest observations that is non-censored.

This estimator is only suitable for right censored data.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`gamma1`	Vector of the corresponding generalised Hill estimates.

Author(s)

Tom Reynkens

References

Einmahl, J.H.J., Fils-Villetard, A. and Guillou, A. (2008). "Statistics of Extremes Under Random Censoring." *Bernoulli*, 14, 207–227.

See Also

`genHill`, `cHill`, `cProbGH`, `cQuantGH`

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X, Y)

# Censoring indicator
censored <- (X>Y)

# Generalised Hill estimator adapted for right censoring
cghill <- cgenHill(Z, censored=censored, plot=TRUE)
```

`cGPDmle` *GPD-ML estimator for right censored data*

Description

Computes ML estimates of fitting GPD to peaks over a threshold adapted for right censoring.

Usage

```
cGPDmle(data, censored, start = c(0.1,1), warnings = FALSE, logk = FALSE,
        plot = FALSE, add = FALSE, main = "POT estimates of the EVI", ...)

cPOT(data, censored, start = c(0.1,1), warnings = FALSE, logk = FALSE,
     plot = FALSE, add = FALSE, main = "POT estimates of the EVI", ...)
```

Arguments

`data`	Vector of n observations.
`censored`	A logical vector of length n indicating if an observation is censored.
`start`	Vector of length 2 containing the starting values for the optimisation. The first element is the starting value for the estimator of γ_1 and the second element is the starting value for the estimator of σ_1. Default is `c(0.1,1)`.
`warnings`	Logical indicating if possible warnings from the optimisation function are shown, default is `FALSE`.
`logk`	Logical indicating if the estimates are plotted as a function of $\log(k)$ (`logk=TRUE`) or as a function of k. Default is `FALSE`.
`plot`	Logical indicating if the estimates of γ_1 should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates of γ_1 should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"POT estimates of the EVI"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The GPD-MLE estimator for the EVI adapted for right censored data is equal to the ordinary GPD-MLE estimator for the EVI divided by the proportion of the k largest observations that is non-censored. The estimates for σ are the ordinary GPD-MLE estimates for σ.

This estimator is only suitable for right censored data.

`cPOT` is the same function but with a different name for compatibility with `POT`.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`gamma1`	Vector of the corresponding MLE estimates for the γ_1 parameter of the GPD.
`sigma1`	Vector of the corresponding MLE estimates for the σ_1 parameter of the GPD.

Author(s)

Tom Reynkens

References

Einmahl, J.H.J., Fils-Villetard, A. and Guillou, A. (2008). "Statistics of Extremes Under Random Censoring." *Bernoulli*, 14, 207–227.

See Also

GPDmle, cProbGPD, cQuantGPD, cEPD

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X, Y)

# Censoring indicator
censored <- (X>Y)

# GPD-ML estimator adapted for right censoring
cpot <- cGPDmle(Z, censored=censored, plot=TRUE)
```

---

cHill — *Hill estimator for right censored data*

Description

Computes the Hill estimator for positive extreme value indices, adapted for right censoring, as a function of the tail parameter k (Beirlant et al., 2007). Optionally, these estimates are plotted as a function of k.

Usage

```
cHill(data, censored, logk = FALSE, plot = FALSE, add = FALSE,
      main = "Hill estimates of the EVI", ...)
```

Arguments

`data`	Vector of n observations.
`censored`	A logical vector of length n indicating if an observation is censored.
`logk`	Logical indicating if the estimates are plotted as a function of $\log(k)$ (`logk=TRUE`) or as a function of k. Default is `FALSE`.

plot	Logical indicating if the estimates of γ_1 should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates of γ_1 should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "Hill estimates of the EVI".
...	Additional arguments for the plot function, see plot for more details.

Details

The Hill estimator adapted for right censored data is equal to the ordinary Hill estimator $H_{k,n}$ divided by the proportion of the k largest observations that is non-censored.

This estimator is only suitable for right censored data, use icHill for interval censored data.

See Section 4.3.2 of Albrecher et al. (2017) for more details.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
gamma1	Vector of the corresponding Hill estimates.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant, J., Guillou, A., Dierckx, G. and Fils-Villetard, A. (2007). "Estimation of the Extreme Value Index and Extreme Quantiles Under Random Censoring." *Extremes*, 10, 151–174.

See Also

Hill, icHill, cParetoQQ, cProb, cQuant

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X, Y)
```

```
# Censoring indicator
censored <- (X>Y)

# Hill estimator adapted for right censoring
chill <- cHill(Z, censored=censored, plot=TRUE)
```

`cLognormalQQ` *Log-normal quantile plot for right censored data*

Description

Log-normal QQ-plot adapted for right censored data.

Usage

```
cLognormalQQ(data, censored, plot = TRUE, main = "Log-normal QQ-plot", ...)
```

Arguments

`data`	Vector of n observations.
`censored`	A logical vector of length n indicating if an observation is censored.
`plot`	Logical indicating if the quantiles should be plotted in a log-normal QQ-plot, default is `TRUE`.
`main`	Title for the plot, default is `"Log-normal QQ-plot"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The log-normal QQ-plot adapted for right censoring is given by

$$(\Phi^{-1}(F_{km}(Z_{j,n})), \log(Z_{j,n}))$$

for $j = 1, \ldots, n-1$, with $Z_{i,n}$ the i-th order statistic of the data, Φ^{-1} the quantile function of the standard normal distribution and F_{km} the Kaplan-Meier estimator for the CDF. Hence, it has the same empirical quantiles as an ordinary log-normal QQ-plot but replaces the theoretical quantiles $\Phi^{-1}(j/(n+1))$ by $\Phi^{-1}(F_{km}(Z_{j,n}))$.

This QQ-plot is only suitable for right censored data.

In Beirlant et al. (2007), only a Pareto QQ-plot adapted for right-censored data is proposed. This QQ-plot is constructed using the same ideas, but is not described in the paper.

Value

A list with following components:

`lqq.the`	Vector of the theoretical quantiles, see Details.
`lqq.emp`	Vector of the empirical quantiles from the log-transformed data.

Author(s)

Tom Reynkens

References

Beirlant, J., Guillou, A., Dierckx, G. and Fils-Villetard, A. (2007). "Estimation of the Extreme Value Index and Extreme Quantiles Under Random Censoring." *Extremes*, 10, 151–174.

See Also

LognormalQQ, cExpQQ, cParetoQQ, cWeibullQQ, KaplanMeier

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X, Y)

# Censoring indicator
censored <- (X>Y)

# Log-normal QQ-plot adapted for right censoring
cLognormalQQ(Z, censored=censored)
```

---

cMoment — *MOM estimator for right censored data*

Description

Computes the Method of Moment estimates adapted for right censored data.

Usage

```
cMoment(data, censored, logk = FALSE, plot = FALSE, add = FALSE,
        main = "Moment estimates of the EVI", ...)
```

Arguments

`data`	Vector of n observations.
`censored`	A logical vector of length n indicating if an observation is censored.
`logk`	Logical indicating if the estimates are plotted as a function of $\log(k)$ (`logk=TRUE`) or as a function of k. Default is `FALSE`.
`plot`	Logical indicating if the estimates of γ_1 should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates of γ_1 should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Moment estimates of the EVI"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The moment estimator adapted for right censored data is equal to the ordinary moment estimator divided by the proportion of the k largest observations that is non-censored.

This estimator is only suitable for right censored data.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`gamma1`	Vector of the corresponding moment estimates.

Author(s)

Tom Reynkens

References

Einmahl, J.H.J., Fils-Villetard, A. and Guillou, A. (2008). "Statistics of Extremes Under Random Censoring." *Bernoulli*, 14, 207–227.

See Also

`Moment`, `cProbMOM`, `cQuantMOM`

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)
```

```
# Observed sample
Z <- pmin(X, Y)

# Censoring indicator
censored <- (X>Y)

# Moment estimator adapted for right censoring
cmom <- cMoment(Z, censored=censored, plot=TRUE)
```

---

cParetoQQ *Pareto quantile plot for right censored data*

Description

Pareto QQ-plot adapted for right censored data.

Usage

```
cParetoQQ(data, censored, plot = TRUE, main = "Pareto QQ-plot", ...)
```

Arguments

`data`	Vector of n observations.
`censored`	A logical vector of length n indicating if an observation is censored.
`plot`	Logical indicating if the quantiles should be plotted in a Pareto QQ-plot, default is `TRUE`.
`main`	Title for the plot, default is `"Pareto QQ-plot"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The Pareto QQ-plot adapted for right censoring is given by

$$(-\log(1 - F_{km}(Z_{j,n})), \log Z_{j,n})$$

for $j = 1, \ldots, n-1$, with $Z_{i,n}$ the i-th order statistic of the data and F_{km} the Kaplan-Meier estimator for the CDF. Hence, it has the same empirical quantiles as an ordinary Pareto QQ-plot but replaces the theoretical quantiles $-\log(1 - j/(n+1))$ by $-\log(1 - F_{km}(Z_{j,n}))$.

This QQ-plot is only suitable for right censored data, use `icParetoQQ` for interval censored data.

Value

A list with following components:

`pqq.the`	Vector of the theoretical quantiles, see Details.
`pqq.emp`	Vector of the empirical quantiles from the log-transformed data.

Author(s)

Tom Reynkens

References

Beirlant, J., Guillou, A., Dierckx, G. and Fils-Villetard, A. (2007). "Estimation of the Extreme Value Index and Extreme Quantiles Under Random Censoring." *Extremes*, 10, 151–174.

See Also

ParetoQQ, icParetoQQ, cExpQQ, cLognormalQQ, cWeibullQQ, cHill, KaplanMeier

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X, Y)

# Censoring indicator
censored <- (X>Y)

# Pareto QQ-plot adapted for right censoring
cParetoQQ(Z, censored=censored)
```

---

cProb — *Estimator of small exceedance probabilities and large return periods using censored Hill*

Description

Computes estimates of a small exceedance probability $P(X > q)$ or large return period $1/P(X > q)$ using the estimates for the EVI obtained from the Hill estimator adapted for right censoring.

Usage

```
cProb(data, censored, gamma1, q, plot = FALSE, add = FALSE,
      main = "Estimates of small exceedance probability", ...)

cReturn(data, censored, gamma1, q, plot = FALSE, add = FALSE,
        main = "Estimates of large return period", ...)
```

Arguments

`data`	Vector of n observations.
`censored`	A logical vector of length n indicating if an observation is censored.
`gamma1`	Vector of $n-1$ estimates for the EVI obtained from `cHill`.
`q`	The used large quantile (we estimate $P(X > q)$ or $1/P(X > q)$ for q large).
`plot`	Logical indicating if the estimates should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Estimates of small exceedance probability"` for `cProb` and `"Estimates of large return period"` for `cReturn`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The probability is estimated as

$$\hat{P}(X > q) = (1 - km) \times (q/Z_{n-k,n})^{-1/H^c_{k,n}}$$

with $Z_{i,n}$ the i-th order statistic of the data, $H^c_{k,n}$ the Hill estimator adapted for right censoring and km the Kaplan-Meier estimator for the CDF evaluated in $Z_{n-k,n}$.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`P`	Vector of the corresponding probability estimates, only returned for `cProb`.
`R`	Vector of the corresponding estimates for the return period, only returned for `cReturn`.
`q`	The used large quantile.

Author(s)

Tom Reynkens

References

Beirlant, J., Guillou, A., Dierckx, G. and Fils-Villetard, A. (2007). "Estimation of the Extreme Value Index and Extreme Quantiles Under Random Censoring." *Extremes*, 10, 151–174.

See Also

`cHill`, `cQuant`, `Prob`, `KaplanMeier`

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X, Y)

# Censoring indicator
censored <- (X>Y)

# Hill estimator adapted for right censoring
chill <- cHill(Z, censored=censored, plot=TRUE)

# Small exceedance probability
q <- 10
cProb(Z, censored=censored, gamma1=chill$gamma1, q=q, plot=TRUE)

# Return period
cReturn(Z, censored=censored, gamma1=chill$gamma1, q=q, plot=TRUE)
```

---

cProbEPD — *Estimator of small exceedance probabilities and large return periods using censored EPD*

Description

Computes estimates of a small exceedance probability $P(X > q)$ or large return period $1/P(X > q)$ using the parameters from the EPD fit adapted for right censoring.

Usage

```
cProbEPD(data, censored, gamma1, kappa1, beta, q, plot = FALSE, add = FALSE,
         main = "Estimates of small exceedance probability", ...)

cReturnEPD(data, censored, gamma1, kappa1, beta, q, plot = FALSE, add = FALSE,
           main = "Estimates of large return period", ...)
```

Arguments

`data`	Vector of n observations.
`censored`	A logical vector of length n indicating if an observation is censored.
`gamma1`	Vector of $n-1$ estimates for the EVI obtained from `cEPD`.

kappa1	Vector of $n-1$ estimates for κ_1 obtained from cEPD.
beta	Vector of $n-1$ estimates for β obtained from cEPD.
q	The used large quantile (we estimate $P(X > q)$ or $1/P(X > q)$ for q large).
plot	Logical indicating if the estimates should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "Estimates of small exceedance probability" for cProbEPD and "Estimates of large return period" for cReturnEPD.
...	Additional arguments for the plot function, see plot for more details.

Details

The probability is estimated as

$$\hat{P}(X > q) = (1 - km) \times (1 - F(q))$$

with F the CDF of the EPD with estimated parameters $\hat{\gamma}_1$, $\hat{\kappa}_1$ and $\hat{\tau} = -\hat{\beta}$ and km the Kaplan-Meier estimator for the CDF evaluated in $Z_{n-k,n}$ (the $(k+1)$-th largest data point).

Value

A list with following components:

k	Vector of the values of the tail parameter k.
P	Vector of the corresponding probability estimates, only returned for cProbEPD.
R	Vector of the corresponding estimates for the return period, only returned for cReturnEPD.
q	The used large quantile.

Author(s)

Tom Reynkens.

References

Beirlant, J., Bardoutsos, A., de Wet, T. and Gijbels, I. (2016). "Bias Reduced Tail Estimation for Censored Pareto Type Distributions." *Statistics & Probability Letters*, 109, 78–88.

See Also

cEPD, ProbEPD, Prob, KaplanMeier

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X, Y)

# Censoring indicator
censored <- (X>Y)

# EPD estimator adapted for right censoring
cepd <- cEPD(Z, censored=censored, plot=TRUE)

# Small exceedance probability
q <- 10
cProbEPD(Z, censored=censored, gamma1=cepd$gamma1,
        kappa1=cepd$kappa1, beta=cepd$beta, q=q, plot=TRUE)

# Return period
cReturnEPD(Z, censored=censored, gamma1=cepd$gamma1,
        kappa1=cepd$kappa1, beta=cepd$beta, q=q, plot=TRUE)
```

`cProbGH` *Estimator of small exceedance probabilities and large return periods using censored generalised Hill*

Description

Computes estimates of a small exceedance probability $P(X > q)$ or large return period $1/P(X > q)$ using the estimates for the EVI obtained from the generalised Hill estimator adapted for right censoring.

Usage

```
cProbGH(data, censored, gamma1, q, plot = FALSE, add = FALSE,
        main = "Estimates of small exceedance probability", ...)

cReturnGH(data, censored, gamma1, q, plot = FALSE, add = FALSE,
          main = "Estimates of large return period", ...)
```

Arguments

data	Vector of n observations.
censored	A logical vector of length n indicating if an observation is censored.
gamma1	Vector of $n-1$ estimates for the EVI obtained from cgenHill.
q	The used large quantile (we estimate $P(X > q)$ or $1/P(X > q)$ for q large).
plot	Logical indicating if the estimates should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "Estimates of small exceedance probability" for cProbGH and "Estimates of large return period" for cReturnGH.
...	Additional arguments for the plot function, see plot for more details.

Details

The probability is estimated as

$$\hat{P}(X > q) = (1 - km) \times (1 + \hat{\gamma}_1 / a_{k,n} \times (q - Z_{n-k,n}))^{-1/\hat{\gamma}_1}$$

with $Z_{i,n}$ the i-th order statistic of the data, $\hat{\gamma}_1$ the generalised Hill estimator adapted for right censoring and km the Kaplan-Meier estimator for the CDF evaluated in $Z_{n-k,n}$. The value a is defined as

$$a_{k,n} = Z_{n-k,n} H_{k,n} (1 - \min(\hat{\gamma}_1, 0)) / \hat{p}_k$$

with $H_{k,n}$ the ordinary Hill estimator and $\hat{p}_k$ the proportion of the k largest observations that is non-censored.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
P	Vector of the corresponding probability estimates, only returned for cProbGH.
R	Vector of the corresponding estimates for the return period, only returned for cReturnGH.
q	The used large quantile.

Author(s)

Tom Reynkens

References

Einmahl, J.H.J., Fils-Villetard, A. and Guillou, A. (2008). "Statistics of Extremes Under Random Censoring." *Bernoulli*, 14, 207–227.

See Also

cQuantGH, cgenHill, ProbGH, cProbMOM, KaplanMeier

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X, Y)

# Censoring indicator
censored <- (X>Y)

# Generalised Hill estimator adapted for right censoring
cghill <- cgenHill(Z, censored=censored, plot=TRUE)

# Small exceedance probability
q <- 10
cProbGH(Z, censored=censored, gamma1=cghill$gamma1, q=q, plot=TRUE)

# Return period
cReturnGH(Z, censored=censored, gamma1=cghill$gamma1, q=q, plot=TRUE)
```

---

cProbGPD — *Estimator of small exceedance probabilities and large return periods using censored GPD-MLE*

Description

Computes estimates of a small exceedance probability $P(X > q)$ or large return period $1/P(X > q)$ using the GPD-ML estimator adapted for right censoring.

Usage

```
cProbGPD(data, censored, gamma1, sigma1, q, plot = FALSE, add = FALSE,
         main = "Estimates of small exceedance probability", ...)

cReturnGPD(data, censored, gamma1, sigma1, q, plot = FALSE, add = FALSE,
           main = "Estimates of large return period", ...)
```

Arguments

`data`	Vector of n observations.
`censored`	A logical vector of length n indicating if an observation is censored.
`gamma1`	Vector of $n-1$ estimates for the EVI obtained from `cGPDmle`.
`sigma1`	Vector of $n-1$ estimates for σ_1 obtained from `cGPDmle`.
`q`	The used large quantile (we estimate $P(X > q)$ or $1/P(X > q)$ for q large).
`plot`	Logical indicating if the estimates should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Estimates of small exceedance probability"` for `cProbGPD` and `"Estimates of large return period"` for `cReturnGPD`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The probability is estimated as

$$\hat{P}(X > q) = (1 - km) \times (1 + \hat{\gamma}_1 / a_{k,n} \times (q - Z_{n-k,n}))^{-1/\hat{\gamma}_1}$$

with $Z_{i,n}$ the i-th order statistic of the data, $\hat{\gamma}_1$ the generalised Hill estimator adapted for right censoring and km the Kaplan-Meier estimator for the CDF evaluated in $Z_{n-k,n}$. The value a is defined as

$$a_{k,n} = \hat{\sigma}_1 / \hat{p}_k$$

with $\hat{\sigma}_1$ the ML estimate for σ_1 and $\hat{p}_k$ the proportion of the k largest observations that is non-censored.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`P`	Vector of the corresponding probability estimates, only returned for `cProbGPD`.
`R`	Vector of the corresponding estimates for the return period, only returned for `cReturnGPD`.
`q`	The used large quantile.

Author(s)

Tom Reynkens

References

Einmahl, J.H.J., Fils-Villetard, A. and Guillou, A. (2008). "Statistics of Extremes Under Random Censoring." *Bernoulli*, 14, 207–227.

See Also

cQuantGPD, cGPDmle, ProbGPD, Prob, KaplanMeier

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X, Y)

# Censoring indicator
censored <- (X>Y)

# GPD-MLE estimator adapted for right censoring
cpot <- cGPDmle(Z, censored=censored, plot=TRUE)

# Exceedance probability
q <- 10
cProbGPD(Z, gamma1=cpot$gamma1, sigma1=cpot$sigma1,
         censored=censored, q=q, plot=TRUE)

# Return period
cReturnGPD(Z, gamma1=cpot$gamma1, sigma1=cpot$sigma1,
           censored=censored, q=q, plot=TRUE)
```

---

cProbMOM — *Estimator of small exceedance probabilities and large return periods using censored MOM*

Description

Computes estimates of a small exceedance probability $P(X > q)$ or large return period $1/P(X > q)$ using the Method of Moments estimates for the EVI adapted for right censoring.

Usage

```
cProbMOM(data, censored, gamma1, q, plot = FALSE, add = FALSE,
         main = "Estimates of small exceedance probability", ...)

cReturnMOM(data, censored, gamma1, q, plot = FALSE, add = FALSE,
           main = "Estimates of large return period", ...)
```

Arguments

`data`	Vector of n observations.
`censored`	A logical vector of length n indicating if an observation is censored.
`gamma1`	Vector of $n-1$ estimates for the EVI obtained from `cMoment`.
`q`	The used large quantile (we estimate $P(X > q)$ or $1/P(X > q)$ for q large).
`plot`	Logical indicating if the estimates should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Estimates of small exceedance probability"` for `cProbMOM` and `"Estimates of large return period"` for `cReturnMOM`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The probability is estimated as

$$\hat{P}(X > q) = (1 - km) \times (1 + \hat{\gamma}_1/a_{k,n} \times (q - Z_{n-k,n}))^{-1/\hat{\gamma}_1}$$

with $Z_{i,n}$ the i-th order statistic of the data, $\hat{\gamma}_1$ the MOM estimator adapted for right censoring and km the Kaplan-Meier estimator for the CDF evaluated in $Z_{n-k,n}$. The value a is defined as

$$a_{k,n} = Z_{n-k,n} H_{k,n} (1 - \min(\hat{\gamma}_1, 0))/\hat{p}_k$$

with $H_{k,n}$ the ordinary Hill estimator and $\hat{p}_k$ the proportion of the k largest observations that is non-censored.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`P`	Vector of the corresponding probability estimates, only returned for `cProbMOM`.
`R`	Vector of the corresponding estimates for the return period, only returned for `cReturnMOM`.
`q`	The used large quantile.

Author(s)

Tom Reynkens

References

Einmahl, J.H.J., Fils-Villetard, A. and Guillou, A. (2008). "Statistics of Extremes Under Random Censoring." *Bernoulli*, 14, 207–227.

See Also

cQuantMOM, cMoment, ProbMOM, Prob, KaplanMeier

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X, Y)

# Censoring indicator
censored <- (X>Y)

# Moment estimator adapted for right censoring
cmom <- cMoment(Z, censored=censored, plot=TRUE)

# Small exceedance probability
q <- 10
cProbMOM(Z, censored=censored, gamma1=cmom$gamma1, q=q, plot=TRUE)

# Return period
cReturnMOM(Z, censored=censored, gamma1=cmom$gamma1, q=q, plot=TRUE)
```

---

cQuant *Estimator of large quantiles using censored Hill*

Description

Computes estimates of large quantiles $Q(1-p)$ using the estimates for the EVI obtained from the Hill estimator adapted for right censoring.

Usage

```
cQuant(data, censored, gamma1, p, plot = FALSE, add = FALSE,
       main = "Estimates of extreme quantile", ...)
```

Arguments

data	Vector of n observations.
censored	A logical vector of length n indicating if an observation is censored.
gamma1	Vector of $n-1$ estimates for the EVI obtained from cHill.

p	The exceedance probability of the quantile (we estimate $Q(1-p)$ for p small).
plot	Logical indicating if the estimates should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "Estimates of extreme quantile".
...	Additional arguments for the plot function, see plot for more details.

Details

The quantile is estimated as

$$\hat{Q}(1-p) = Z_{n-k,n} \times ((1-km)/p)^{H^c_{k,n}}$$

with $Z_{i,n}$ the i-th order statistic of the data, $H^c_{k,n}$ the Hill estimator adapted for right censoring and km the Kaplan-Meier estimator for the CDF evaluated in $Z_{n-k,n}$.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
Q	Vector of the corresponding quantile estimates.
p	The used exceedance probability.

Author(s)

Tom Reynkens.

References

Beirlant, J., Guillou, A., Dierckx, G. and Fils-Villetard, A. (2007). "Estimation of the Extreme Value Index and Extreme Quantiles Under Random Censoring." *Extremes*, 10, 151–174.

See Also

cHill, cProb, Quant, KaplanMeier

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
```

```
Z <- pmin(X, Y)

# Censoring indicator
censored <- (X>Y)

# Hill estimator adapted for right censoring
chill <- cHill(Z, censored=censored, plot=TRUE)

# Large quantile
p <- 10^(-4)
cQuant(Z, gamma1=chill$gamma, censored=censored, p=p, plot=TRUE)
```

---

cQuantGH *Estimator of large quantiles using censored Hill*

Description

Computes estimates of large quantiles $Q(1-p)$ using the estimates for the EVI obtained from the generalised Hill estimator adapted for right censoring.

Usage

```
cQuantGH(data, censored, gamma1, p, plot = FALSE, add = FALSE,
         main = "Estimates of extreme quantile", ...)
```

Arguments

data	Vector of n observations.
censored	A logical vector of length n indicating if an observation is censored.
gamma1	Vector of $n-1$ estimates for the EVI obtained from cgenHill.
p	The exceedance probability of the quantile (we estimate $Q(1-p)$ for p small).
plot	Logical indicating if the estimates should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "Estimates of extreme quantile".
...	Additional arguments for the plot function, see plot for more details.

Details

The quantile is estimated as

$$\hat{Q}(1-p) = Z_{n-k,n} + a_{k,n}(((1-km)/p)^{\hat{\gamma}_1} - 1)/\hat{\gamma}_1)$$

with $Z_{i,n}$ the i-th order statistic of the data, $\hat{\gamma}_1$ the generalised Hill estimator adapted for right censoring and km the Kaplan-Meier estimator for the CDF evaluated in $Z_{n-k,n}$. The value a is defined as

$$a_{k,n} = Z_{n-k,n} H_{k,n} (1 - S_{Z,k,n})/\hat{p}_k$$

with $H_{k,n}$ the ordinary Hill estimator and $\hat{p}_k$ the proportion of the k largest observations that is non-censored, and

$$S_{Z,k,n} = 1 - (1 - M_1^2/M_2)^(- 1)/2$$

with

$$M_l == 1/k \sum_{j=1}^{k} (\log X_{n-j+1,n} - \log X_{n-k,n})^l.$$

Value

A list with following components:

- `k` Vector of the values of the tail parameter k.
- `Q` Vector of the corresponding quantile estimates.
- `p` The used exceedance probability.

Author(s)

Tom Reynkens

References

Einmahl, J.H.J., Fils-Villetard, A. and Guillou, A. (2008). "Statistics of Extremes Under Random Censoring." *Bernoulli*, 14, 207–227.

See Also

`cProbGH`, `cgenHill`, `QuantGH`, `Quant`, `KaplanMeier`

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X, Y)

# Censoring indicator
censored <- (X>Y)

# Generalised Hill estimator adapted for right censoring
cghill <- cgenHill(Z, censored=censored, plot=TRUE)

# Large quantile
p <- 10^(-4)
cQuantGH(Z, gamma1=cghill$gamma, censored=censored, p=p, plot=TRUE)
```

cQuantGPD — *Estimator of large quantiles using censored GPD-MLE*

Description

Computes estimates of large quantiles $Q(1-p)$ using the estimates for the EVI obtained from the GPD-ML estimator adapted for right censoring.

Usage

```
cQuantGPD(data, censored, gamma1, sigma1, p, plot = FALSE, add = FALSE,
          main = "Estimates of extreme quantile", ...)
```

Arguments

`data`	Vector of n observations.
`censored`	A logical vector of length n indicating if an observation is censored.
`gamma1`	Vector of $n-1$ estimates for the EVI obtained from `cGPDmle`.
`sigma1`	Vector of $n-1$ estimates for σ_1 obtained from `cGPDmle`.
`p`	The exceedance probability of the quantile (we estimate $Q(1-p)$ for p small).
`plot`	Logical indicating if the estimates should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Estimates of extreme quantile"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The quantile is estimated as

$$\hat{Q}(1-p) = Z_{n-k,n} + a_{k,n}(((1-km)/p)^{\hat{\gamma}_1} - 1)/\hat{\gamma}_1)$$

ith $Z_{i,n}$ the i-th order statistic of the data, $\hat{\gamma}_1$ the generalised Hill estimator adapted for right censoring and km the Kaplan-Meier estimator for the CDF evaluated in $Z_{n-k,n}$. The value a is defined as

$$a_{k,n} = \hat{\sigma}_1/\hat{p}_k$$

with $\hat{\sigma}_1$ the ML estimate for σ_1 and $\hat{p}_k$ the proportion of the k largest observations that is non-censored.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`Q`	Vector of the corresponding quantile estimates.
`p`	The used exceedance probability.

Author(s)

Tom Reynkens

References

Einmahl, J.H.J., Fils-Villetard, A. and Guillou, A. (2008). "Statistics of Extremes Under Random Censoring." *Bernoulli*, 14, 207–227.

See Also

cProbGPD, cGPDmle, QuantGPD, Quant, KaplanMeier

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X, Y)

# Censoring indicator
censored <- (X>Y)

# GPD-MLE estimator adapted for right censoring
cpot <- cGPDmle(Z, censored=censored, plot=TRUE)

# Large quantile
p <- 10^(-4)
cQuantGPD(Z, gamma1=cpot$gamma1, sigma1=cpot$sigma1,
         censored=censored, p=p, plot=TRUE)
```

---

cQuantMOM *Estimator of large quantiles using censored MOM*

Description

Computes estimates of large quantiles $Q(1-p)$ using the estimates for the EVI obtained from the MOM estimator adapted for right censoring.

Usage

```
cQuantMOM(data, censored, gamma1, p, plot = FALSE, add = FALSE,
          main = "Estimates of extreme quantile", ...)
```

Arguments

data	Vector of n observations.
censored	A logical vector of length n indicating if an observation is censored.
gamma1	Vector of $n-1$ estimates for the EVI obtained from cMoment.
p	The exceedance probability of the quantile (we estimate $Q(1-p)$ for p small).
plot	Logical indicating if the estimates should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "Estimates of extreme quantile".
...	Additional arguments for the plot function, see plot for more details.

Details

The quantile is estimated as

$$\hat{Q}(1-p) = Z_{n-k,n} + a_{k,n}(((1-km)/p)^{\hat{\gamma}_1} - 1)/\hat{\gamma}_1)$$

ith $Z_{i,n}$ the i-th order statistic of the data, $\hat{\gamma}_1$ the MOM estimator adapted for right censoring and km the Kaplan-Meier estimator for the CDF evaluated in $Z_{n-k,n}$. The value a is defined as

$$a_{k,n} = Z_{n-k,n}H_{k,n}(1-\min(\hat{\gamma}_1, 0))/\hat{p}_k$$

with $H_{k,n}$ the ordinary Hill estimator and $\hat{p}_k$ the proportion of the k largest observations that is non-censored.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
Q	Vector of the corresponding quantile estimates.
p	The used exceedance probability.

Author(s)

Tom Reynkens

References

Einmahl, J.H.J., Fils-Villetard, A. and Guillou, A. (2008). "Statistics of Extremes Under Random Censoring." *Bernoulli*, 14, 207–227.

See Also

cProbMOM, cMoment, QuantMOM, Quant, KaplanMeier

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X, Y)

# Censoring indicator
censored <- (X>Y)

# Moment estimator adapted for right censoring
cmom <- cMoment(Z, censored=censored, plot=TRUE)

# Large quantile
p <- 10^(-4)
cQuantMOM(Z, censored=censored, gamma1=cmom$gamma1, p=p, plot=TRUE)
```

---

crHill *Hill-type estimator for the conditional EVI*

Description

Hill-type estimator for the conditional Extreme Value Index (EVI) adapted for censored data.

Usage

```
crHill(x, Xtilde, Ytilde, censored, h,
      kernel = c("biweight", "normal", "uniform", "triangular", "epanechnikov"),
       logk = FALSE, plot = FALSE, add = FALSE, main = "", ...)
```

Arguments

`x`	Value of the conditioning variable X to estimate the EVI at.
`Xtilde`	Vector of length n containing the censored sample of the conditioning variable X.
`Ytilde`	Vector of length n containing the censored sample of the variable Y.
`censored`	A logical vector of length n indicating if an observation is censored.
`h`	Bandwidth of the non-parametric estimator.
`kernel`	Kernel of the non-parametric estimator. One of `"biweight"` (default), `"normal"`, `"uniform"`, `"triangular"` and `"epanechnikov"`.

logk	Logical indicating if the Hill-type estimates are plotted as a function of $\log(k)$ (logk=TRUE) or as a function of k. Default is FALSE.
plot	Logical indicating if the estimates should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "" (no title).
...	Additional arguments for the plot function, see plot for more details.

Details

This is a Hill-type estimator of the EVI of Y given $X = x$. The estimator uses the censored sample $(\tilde{X}_i, \tilde{Y}_i)$, for $i = 1, \ldots, n$, where X and Y are censored at the same time. We assume that Y and the censoring variable are conditionally independent given X.

See Section 4.4.3 in Albrecher et al. (2017) for more details.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
gamma	Vector of the corresponding Hill-type estimates.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

See Also

crParetoQQ, crSurv, cHill

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
Y <- rpareto(200, shape=2)

# Censoring variable
C <- rpareto(200, shape=1)

# Observed (censored) sample of variable Y
Ytilde <- pmin(Y, C)
```

```
# Censoring indicator
censored <- (Y>C)

# Conditioning variable
X <- seq(1, 10, length.out=length(Y))

# Observed (censored) sample of conditioning variable
Xtilde <- X
Xtilde[censored] <- X[censored] - runif(sum(censored), 0, 1)


# Conditional Pareto QQ-plot
crParetoQQ(x=1, Xtilde=Xtilde, Ytilde=Ytilde, censored=censored, h=2)

# Plot Hill-type estimates
crHill(x=1, Xtilde, Ytilde, censored, h=2, plot=TRUE)
```

---

crParetoQQ   *Conditional Pareto quantile plot for right censored data*

Description

Conditional Pareto QQ-plot adapted for right censored data.

Usage

```
crParetoQQ(x, Xtilde, Ytilde, censored, h,
        kernel = c("biweight", "normal", "uniform", "triangular", "epanechnikov"),
           plot = TRUE, add = FALSE, main = "Pareto QQ-plot", type = "p", ...)
```

Arguments

x	Value of the conditioning variable X at which to make the conditional Pareto QQ-plot.
Xtilde	Vector of length n containing the censored sample of the conditioning variable X.
Ytilde	Vector of length n containing the censored sample of the variable Y.
censored	A logical vector of length n indicating if an observation is censored.
h	Bandwidth of the non-parametric estimator for the conditional survival function (crSurv).
kernel	Kernel of the non-parametric estimator for the conditional survival function (crSurv). One of "biweight" (default), "normal", "uniform", "triangular" and "epanechnikov".
plot	Logical indicating if the quantiles should be plotted in a Pareto QQ-plot, default is TRUE.

add	Logical indicating if the quantiles should be added to an existing plot, default is `FALSE`.
main	Title for the plot, default is `"Pareto QQ-plot"`.
type	Type of the plot, default is `"p"` meaning points are plotted, see `plot` for more details.
...	Additional arguments for the `plot` function, see `plot` for more details.

Details

We construct a Pareto QQ-plot for Y conditional on $X = x$ using the censored sample $(\tilde{X}_i, \tilde{Y}_i)$, for $i = 1, \ldots, n$, where X and Y are censored at the same time. We assume that Y and the censoring variable are conditionally independent given X.

The conditional Pareto QQ-plot adapted for right censoring is given by

$$(-\log(1 - \hat{F}_{Y|X}(\tilde{Y}_{j,n}|x)), \log \tilde{Y}_{j,n})$$

for $j = 1, \ldots, n-1$, with $\tilde{Y}_{i,n}$ the i-th order statistic of the censored data and $\hat{F}_{Y|X}(y|x)$ the non-parametric estimator for the conditional CDF of Akritas and Van Keilegom (2003), see `crSurv`.

See Section 4.4.3 in Albrecher et al. (2017) for more details.

Value

A list with following components:

pqq.the	Vector of the theoretical quantiles, see Details.
pqq.emp	Vector of the empirical quantiles from the log-transformed Y data.

Author(s)

Tom Reynkens

References

Akritas, M.G. and Van Keilegom, I. (2003). "Estimation of Bivariate and Marginal Distributions With Censored Data." *Journal of the Royal Statistical Society: Series B*, 65, 457–471.

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

See Also

`crSurv`, `crHill`, `cParetoQQ`

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
Y <- rpareto(200, shape=2)
```

```
# Censoring variable
C <- rpareto(200, shape=1)

# Observed (censored) sample of variable Y
Ytilde <- pmin(Y, C)

# Censoring indicator
censored <- (Y>C)

# Conditioning variable
X <- seq(1, 10, length.out=length(Y))

# Observed (censored) sample of conditioning variable
Xtilde <- X
Xtilde[censored] <- X[censored] - runif(sum(censored), 0, 1)


# Conditional Pareto QQ-plot
crParetoQQ(x=1, Xtilde=Xtilde, Ytilde=Ytilde, censored=censored, h=2)

# Plot Hill-type estimates
crHill(x=1, Xtilde, Ytilde, censored, h=2, plot=TRUE)
```

---

crSurv *Non-parametric estimator of conditional survival function*

Description

Non-parametric estimator of the conditional survival function of Y given X for censored data, see Akritas and Van Keilegom (2003).

Usage

```
crSurv(x, y, Xtilde, Ytilde, censored, h,
      kernel = c("biweight", "normal", "uniform", "triangular", "epanechnikov"))
```

Arguments

x	The value of the conditioning variable X to evaluate the survival function at. x needs to be a single number or a vector with the same length as y.
y	The value(s) of the variable Y to evaluate the survival function at.
Xtilde	Vector of length n containing the censored sample of the conditioning variable X.
Ytilde	Vector of length n containing the censored sample of the variable Y.
censored	A logical vector of length n indicating if an observation is censored.
h	Bandwidth of the non-parametric estimator.
kernel	Kernel of the non-parametric estimator. One of "biweight" (default), "normal", "uniform", "triangular" and "epanechnikov".

Details

We estimate the conditional survival function

$$1 - F_{Y|X}(y|x)$$

using the censored sample $(\tilde{X}_i, \tilde{Y}_i)$, for $i = 1, \ldots, n$, where X and Y are censored at the same time. We assume that Y and the censoring variable are conditionally independent given X.

The estimator is given by

$$1 - \hat{F}_{Y|X}(y|x) = \prod_{\tilde{Y}_i \le y} (1 - W_{n,i}(x; h_n)/(\sum_{j=1}^{n} W_{n,j}(x; h_n) I\{\tilde{Y}_j \ge \tilde{Y}_i\}))^{\Delta_i}$$

where $\Delta_i = 1$ when $(\tilde{X}_i, \tilde{Y}_i)$ is censored and 0 otherwise. The weights are given by

$$W_{n,i}(x; h_n) = K((x - \tilde{X}_i)/h_n) / \sum_{\Delta_j = 1} K((x - \tilde{X}_j)/h_n)$$

when $\Delta_i = 1$ and 0 otherwise.

See Section 4.4.3 in Albrecher et al. (2017) for more details.

Value

Estimates for $1 - F_{Y|X}(y|x)$ as described above.

Author(s)

Tom Reynkens

References

Akritas, M.G. and Van Keilegom, I. (2003). "Estimation of Bivariate and Marginal Distributions With Censored Data." *Journal of the Royal Statistical Society: Series B*, 65, 457–471.

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

See Also

crParetoQQ, crHill

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
Y <- rpareto(200, shape=2)

# Censoring variable
C <- rpareto(200, shape=1)
```

```
# Observed (censored) sample of variable Y
Ytilde <- pmin(Y, C)

# Censoring indicator
censored <- (Y>C)

# Conditioning variable
X <- seq(1, 10, length.out=length(Y))

# Observed (censored) sample of conditioning variable
Xtilde <- X
Xtilde[censored] <- X[censored] - runif(sum(censored), 0, 1)

# Plot estimates of the conditional survival function
x <- 5
y <- seq(0, 5, 1/100)
plot(y, crSurv(x, y, Xtilde=Xtilde, Ytilde=Ytilde, censored=censored, h=5), type="l",
     xlab="y", ylab="Conditional survival function")
```

CTE *Conditional Tail Expectation*

Description

Compute Conditional Tail Expectation (CTE) CTE_{1-p} of the fitted spliced distribution.

Usage

```
CTE(p, splicefit)

ES(p, splicefit)
```

Arguments

p	The probability associated with the CTE (we estimate CTE_{1-p}).
splicefit	A `SpliceFit` object, e.g. output from `SpliceFitPareto`, `SpliceFiticPareto` or `SpliceFitGPD`.

Details

The Conditional Tail Expectation is defined as

$$CTE_{1-p} = E(X|X > Q(1-p)) = E(X|X > VaR_{1-p}) = VaR_{1-p} + \Pi(VaR_{1-p})/p,$$

where $\Pi(u) = E((X-u)_+)$ is the premium of the excess-loss insurance with retention u.

If the CDF is continuous in p, we have $CTE_{1-p} = TVaR_{1-p} = 1/p \int_0^p VaR_{1-s} ds$ with $TVaR$ the Tail Value-at-Risk.

See Reynkens et al. (2017) and Section 4.6 of Albrecher et al. (2017) for more details.

The `ES` function is the same function as `CTE` but is deprecated.

Value

Vector with the CTE corresponding to each element of p.

Author(s)

Tom Reynkens with R code from Roel Verbelen for the mixed Erlang quantiles.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Reynkens, T., Verbelen, R., Beirlant, J. and Antonio, K. (2017). "Modelling Censored Losses Using Splicing: a Global Fit Strategy With Mixed Erlang and Extreme Value Distributions". *Insurance: Mathematics and Economics*, 77, 65–77.

Verbelen, R., Gong, L., Antonio, K., Badescu, A. and Lin, S. (2015). "Fitting Mixtures of Erlangs to Censored and Truncated Data Using the EM Algorithm." *Astin Bulletin*, 45, 729–758

See Also

qSplice, ExcessSplice, SpliceFit, SpliceFitPareto, SpliceFiticPareto, SpliceFitGPD

Examples

```
## Not run:

# Pareto random sample
X <- rpareto(1000, shape = 2)

# Splice ME and Pareto
splicefit <- SpliceFitPareto(X, 0.6)

p <- seq(0.01, 0.99, 0.01)
# Plot of CTE
plot(p, CTE(p, splicefit), type="l", xlab="p", ylab=bquote(CTE[1-p]))

## End(Not run)
```

---

cWeibullQQ *Weibull quantile plot for right censored data*

Description

Weibull QQ-plot adapted for right censored data.

Usage

```
cWeibullQQ(data, censored, plot = TRUE, main = "Weibull QQ-plot", ...)
```

Arguments

`data`	Vector of n observations.
`censored`	A logical vector of length n indicating if an observation is censored.
`plot`	Logical indicating if the quantiles should be plotted in a Weibull QQ-plot, default is `TRUE`.
`main`	Title for the plot, default is `"Weibull QQ-plot"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The Weibull QQ-plot adapted for right censoring is given by

$$(\log(-\log(1 - F_{km}(Z_{j,n}))), \log(Z_{j,n}))$$

for $j = 1, \ldots, n-1$, with $Z_{i,n}$ the i-th order statistic of the data and F_{km} the Kaplan-Meier estimator for the CDF. Hence, it has the same empirical quantiles as an ordinary Weibull QQ-plot but replaces the theoretical quantiles $\log(-\log(1 - j/(n+1)))$ by $\log(-\log(1 - F_{km}(Z_{j,n})))$.

This QQ-plot is only suitable for right censored data.

In Beirlant et al. (2007), only a Pareto QQ-plot adapted for right-censored data is proposed. This QQ-plot is constructed using the same ideas, but is not described in the paper.

Value

A list with following components:

`wqq.the`	Vector of the theoretical quantiles, see Details.
`wqq.emp`	Vector of the empirical quantiles from the log-transformed data.

Author(s)

Tom Reynkens

References

Beirlant, J., Guillou, A., Dierckx, G. and Fils-Villetard, A. (2007). "Estimation of the Extreme Value Index and Extreme Quantiles Under Random Censoring." *Extremes*, 10, 151–174.

See Also

`WeibullQQ`, `cExpQQ`, `cLognormalQQ`, `cParetoQQ`, `KaplanMeier`

Examples

```
# Set seed
set.seed(29072016)

# Pareto random sample
X <- rpareto(500, shape=2)
```

```
# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X, Y)

# Censoring indicator
censored <- (X>Y)

# Weibull QQ-plot adapted for right censoring
cWeibullQQ(Z, censored=censored)
```

EPD *EPD estimator*

Description

Fit the Extended Pareto Distribution (GPD) to the exceedances (peaks) over a threshold. Optionally, these estimates are plotted as a function of k.

Usage

```
EPD(data, rho = -1, start = NULL, direct = FALSE, warnings = FALSE,
   logk = FALSE, plot = FALSE, add = FALSE, main = "EPD estimates of the EVI", ...)
```

Arguments

`data`	Vector of n observations.
`rho`	A parameter for the ρ-estimator of Fraga Alves et al. (2003) when strictly positive or choice(s) for ρ if negative. Default is `-1`.
`start`	Vector of length 2 containing the starting values for the optimisation. The first element is the starting value for the estimator of γ and the second element is the starting value for the estimator of κ. This argument is only used when `direct=TRUE`. Default is `NULL` meaning the initial value for γ is the Hill estimator and the initial value for κ is 0.
`direct`	Logical indicating if the parameters are obtained by directly maximising the log-likelihood function, see Details. Default is `FALSE`.
`warnings`	Logical indicating if possible warnings from the optimisation function are shown, default is `FALSE`.
`logk`	Logical indicating if the estimates are plotted as a function of $\log(k)$ (`logk=TRUE`) or as a function of k. Default is `FALSE`.
`plot`	Logical indicating if the estimates of γ should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates of γ should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"EPD estimates of the EVI"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

We fit the Extended Pareto distribution to the relative excesses over a threshold (X/u). The EPD has distribution function $F(x) = 1 - (x(1+\kappa-\kappa x^\tau))^{-1/\gamma}$ with $\tau = \rho/\gamma < 0 < \gamma$ and $\kappa > \max(-1, 1/\tau)$.

The parameters are determined using MLE and there are two possible approaches: maximise the log-likelihood directly (`direct=TRUE`) or follow the approach detailed in Beirlant et al. (2009) (`direct=FALSE`). The latter approach uses the score functions of the log-likelihood.

See Section 4.2.1 of Albrecher et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`gamma`	Vector of the corresponding estimates for the γ parameter of the EPD.
`kappa`	Vector of the corresponding MLE estimates for the κ parameter of the EPD.
`tau`	Vector of the corresponding estimates for the τ parameter of the EPD using Hill estimates and values for ρ.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant, J., Joossens, E. and Segers, J. (2009). "Second-Order Refined Peaks-Over-Threshold Modelling for Heavy-Tailed Distributions." *Journal of Statistical Planning and Inference*, 139, 2800–2815.

Fraga Alves, M.I. , Gomes, M.I. and de Haan, L. (2003). "A New Class of Semi-parametric Estimators of the Second Order Parameter." *Portugaliae Mathematica*, 60, 193–214.

See Also

`GPDmle`, `ProbEPD`

Examples

```
data(secura)

# EPD estimates for the EVI
epd <- EPD(secura$size, plot=TRUE)

# Compute return periods
ReturnEPD(secura$size, 10^10, gamma=epd$gamma, kappa=epd$kappa,
          tau=epd$tau, plot=TRUE)
```

EPDfit *Fit EPD using MLE*

Description

Fit the Extended Pareto Distribution (EPD) to data using Maximum Likelihood Estimation (MLE).

Usage

```
EPDfit(data, tau, start = c(0.1, 1), warnings = FALSE)
```

Arguments

`data`	Vector of n observations.
`tau`	Value for the τ parameter of the EPD.
`start`	Vector of length 2 containing the starting values for the optimisation. The first element is the starting value for the estimator of γ and the second element is the starting value for the estimator of κ. Default is `c(0.1,1)`.
`warnings`	Logical indicating if possible warnings from the optimisation function are shown, default is `FALSE`.

Details

See Section 4.2.1 of Albrecher et al. (2017) for more details.

Value

A vector with the MLE estimate for the γ parameter of the EPD as the first component and the MLE estimate for the κ parameter of the EPD as the second component.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant, J., Joossens, E. and Segers, J. (2009). "Second-Order Refined Peaks-Over-Threshold Modelling for Heavy-Tailed Distributions." *Journal of Statistical Planning and Inference*, 139, 2800–2815.

See Also

`EPD`, `GPDfit`

Examples

```
data(soa)

# Look at last 500 observations of SOA data
SOAdata <- sort(soa$size)[length(soa$size)-(0:499)]

# Fit EPD to last 500 observations
res <- EPDfit(SOAdata/sort(soa$size)[500], tau=-1)
```

EVTfit *EVT fit*

Description

Create an S3 object using an EVT (Extreme Value Theory) fit.

Usage

```
EVTfit(gamma, endpoint = NULL, sigma = NULL)
```

Arguments

gamma	Vector of estimates for γ.
endpoint	Vector of endpoints (with the same length as gamma). When NULL (default), a vector containing Inf for each value of gamma will be used.
sigma	Vector of scale estimates for the GPD (with the same length as gamma). When NULL (default), not included in the object.

Details

See Reynkens et al. (2017) and Section 4.3 of Albrecher et al. (2017) for details.

Value

An S3 object containing the above input arguments.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Reynkens, T., Verbelen, R., Beirlant, J. and Antonio, K. (2017). "Modelling Censored Losses Using Splicing: a Global Fit Strategy With Mixed Erlang and Extreme Value Distributions". *Insurance: Mathematics and Economics*, 77, 65–77.

See Also

SpliceFit, SpliceFitPareto, SpliceFiticPareto, SpliceFitGPD

Examples

```
# Create MEfit object
mefit <- MEfit(p=c(0.65,0.35), shape=c(39,58), theta=16.19, M=2)

# Create EVTfit object
evtfit <- EVTfit(gamma=c(0.76,0.64), endpoint=c(39096, Inf))

# Create SpliceFit object
splicefit <- SpliceFit(const=c(0.5,0.996), trunclower=0, t=c(1020,39096), type=c("ME","TPa","Pa"),
                       MEfit=mefit, EVTfit=evtfit)

# Show summary
summary(splicefit)
```

ExcessEPD *Estimates for excess-loss premiums using EPD estimates*

Description

Estimate premiums of excess-loss reinsurance with retention R and limit L using EPD estimates.

Usage

```
ExcessEPD(data, gamma, kappa, tau, R, L = Inf, warnings = TRUE, plot = TRUE, add = FALSE,
          main = "Estimates for premium of excess-loss insurance", ...)
```

Arguments

`data`	Vector of n observations.
`gamma`	Vector of $n-1$ estimates for the EVI, obtained from EPD.
`kappa`	Vector of $n-1$ estimates for κ, obtained from EPD.
`tau`	Vector of $n-1$ estimates for τ, obtained from EPD.
`R`	The retention level of the (re-)insurance.
`L`	The limit of the (re-)insurance, default is `Inf`.
`warnings`	Logical indicating if warnings are displayed, default is `TRUE`.
`plot`	Logical indicating if the estimates should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Estimates for premium of excess-loss insurance"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

We need that $u \geq X_{n-k,n}$, the $(k+1)$-th largest observation. If this is not the case, we return `NA` for the premium. A warning will be issued in that case if `warnings=TRUE`.

The premium for the excess-loss insurance with retention R and limit L is given by

$$E(\min{(X-R)_+, L)} = \Pi(R) - \Pi(R+L)$$

where $\Pi(u) = E((X-u)_+) = \int_u^\infty (1-F(z))dz$ is the premium of the excess-loss insurance with retention u. When $L = \infty$, the premium is equal to $\Pi(R)$.

We estimate Π by

$$\hat{\Pi}(u) = (k+1)/(n+1) \times (X_{n-k,n})^{1/\hat{\gamma}} \times ((1-\hat{\kappa}/\hat{\gamma})(1/\hat{\gamma}-1)^{-1}u^{1-1/\hat{\gamma}} + \hat{\kappa}/(\hat{\gamma}X_{n-k,n}^{\hat{\tau}})(1/\hat{\gamma}-\hat{\tau}-1)^{-1}u^{1+\hat{\tau}-1/\hat{\gamma}})$$

with $\hat{\gamma}$, $\hat{\kappa}$ and $\hat{\tau}$ the estimates for the parameters of the EPD.

See Section 4.6 of Albrecher et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`premium`	The corresponding estimates for the premium.
`R`	The retention level of the (re-)insurance.
`L`	The limit of the (re-)insurance.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

See Also

`EPD`, `ExcessHill`, `ExcessGPD`

Examples

```
data(secura)

# EPD estimator
epd <- EPD(secura$size)

# Premium of excess-loss insurance with retention R
R <- 10^7
ExcessEPD(secura$size, gamma=epd$gamma, kappa=epd$kappa, tau=epd$tau, R=R, ylim=c(0,2*10^4))
```

ExcessGPD *Estimates for excess-loss premiums using GPD-MLE estimates*

Description

Estimate premiums of excess-loss reinsurance with retention R and limit L using GPD-MLE estimates.

Usage

```
ExcessGPD(data, gamma, sigma, R, L = Inf, warnings = TRUE, plot = TRUE, add = FALSE,
          main = "Estimates for premium of excess-loss insurance", ...)
```

Arguments

`data`	Vector of n observations.
`gamma`	Vector of $n-1$ estimates for the EVI obtained from `GPDmle`.
`sigma`	Vector of $n-1$ estimates for σ obtained from `GPDmle`.
`R`	The retention level of the (re-)insurance.
`L`	The limit of the (re-)insurance, default is `Inf`.
`warnings`	Logical indicating if warnings are displayed, default is `TRUE`.
`plot`	Logical indicating if the estimates should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Estimates for premium of excess-loss insurance"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

We need that $u \geq X_{n-k,n}$, the $(k+1)$-th largest observation. If this is not the case, we return `NA` for the premium. A warning will be issued in that case if `warnings=TRUE`. One should then use global fits: `ExcessSplice`.

The premium for the excess-loss insurance with retention R and limit L is given by

$$E(\min{(X-R)_+, L)} = \Pi(R) - \Pi(R+L)$$

where $\Pi(u) = E((X-u)_+) = \int_u^\infty (1-F(z))dz$ is the premium of the excess-loss insurance with retention u. When $L = \infty$, the premium is equal to $\Pi(R)$.

We estimate Π by

$$\hat{\Pi}(u) = (k+1)/(n+1) \times \hat{\sigma}_k/(1-\hat{\gamma}_k) \times (1+\hat{\gamma}_k/\hat{\sigma}_k(u-X_{n-k,n}))^{1-1/\hat{\gamma}_k},$$

with $\hat{\gamma}_k$ and $\hat{\sigma}_k$ the estimates for the parameters of the GPD.

See Section 4.6 of Albrecher et al. (2017) for more details.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
premium	The corresponding estimates for the premium.
R	The retention level of the (re-)insurance.
L	The limit of the (re-)insurance.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

See Also

GPDmle, ExcessHill, ExcessEPD

Examples

```
data(secura)

# GPDmle estimator
mle <- GPDmle(secura$size)

# Premium of excess-loss insurance with retention R
R <- 10^7
ExcessGPD(secura$size, gamma=mle$gamma, sigma=mle$sigma, R=R, ylim=c(0,2*10^4))
```

ExcessPareto *Estimates for excess-loss premiums using a Pareto model*

Description

Estimate premiums of excess-loss reinsurance with retention R and limit L using a (truncated) Pareto model.

Usage

```
ExcessPareto(data, gamma, R, L = Inf, endpoint = Inf, warnings = TRUE, plot = TRUE,
       add = FALSE, main = "Estimates for premium of excess-loss insurance", ...)

ExcessHill(data, gamma, R, L = Inf, endpoint = Inf, warnings = TRUE, plot = TRUE,
       add = FALSE, main = "Estimates for premium of excess-loss insurance", ...)
```

Arguments

`data`	Vector of n observations.
`gamma`	Vector of $n-1$ estimates for the EVI, obtained from `Hill` or `trHill`.
`R`	The retention level of the (re-)insurance.
`L`	The limit of the (re-)insurance, default is `Inf`.
`endpoint`	Endpoint for the truncated Pareto distribution. When `Inf`, the default, the ordinary Pareto model is used.
`warnings`	Logical indicating if warnings are displayed, default is `TRUE`.
`plot`	Logical indicating if the estimates should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Estimates for premium of excess-loss insurance"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

We need that $u \geq X_{n-k,n}$, the $(k+1)$-th largest observation. If this is not the case, we return `NA` for the premium. A warning will be issued in that case if `warnings=TRUE`. One should then use global fits: `ExcessSplice`.

The premium for the excess-loss insurance with retention R and limit L is given by

$$E(\min{(X-R)_+, L}) = \Pi(R) - \Pi(R+L)$$

where $\Pi(u) = E((X-u)_+) = \int_u^\infty (1-F(z))dz$ is the premium of the excess-loss insurance with retention u. When $L = \infty$, the premium is equal to $\Pi(R)$.

We estimate Π (for the untruncated Pareto distribution) by

$$\hat{\Pi}(u) = (k+1)/(n+1)/(1/H_{k,n} - 1) \times (X_{n-k,n}^{1/H_{k,n}} u^{1-1/H_{k,n}}),$$

with $H_{k,n}$ the Hill estimator.

The `ExcessHill` function is the same function but with a different name for compatibility with old versions of the package.

See Section 4.6 of Albrecher et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`premium`	The corresponding estimates for the premium.
`R`	The retention level of the (re-)insurance.
`L`	The limit of the (re-)insurance.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

See Also

Hill, ExcessEPD, ExcessGPD, ExcessSplice

Examples

```
data(secura)

# Hill estimator
H <- Hill(secura$size)

# Premium of excess-loss insurance with retention R
R <- 10^7
ExcessPareto(secura$size, H$gamma, R=R)
```

ExcessSplice *Estimates for excess-loss premiums using splicing*

Description

Estimate premiums of excess-loss reinsurance with retention R and limit L using fitted spliced distribution.

Usage

```
ExcessSplice(R, L=Inf, splicefit)
```

Arguments

R	The retention level of the (re-)insurance or a vector of retention levels for the (re-)insurance.
L	The limit for the (re-)insurance or a vector of limits for the (re-)insurance, default is `Inf`.
splicefit	A SpliceFit object, e.g. output from SpliceFitPareto, SpliceFiticPareto or SpliceFitGPD.

Details

The premium for the excess-loss insurance with retention R and limit L is given by

$$E(\min{(X-R)_+,L}) = \Pi(R) - \Pi(R+L)$$

where $\Pi(u) = E((X-u)_+) = \int_u^{\infty}(1-F(z))dz$ is the premium of the excess-loss insurance with retention u. When $L = \infty$, the premium is equal to $\Pi(R)$.

See Reynkens et al. (2017) and Section 4.6 of Albrecher et al. (2017) for more details.

Value

An estimate for the premium is returned (for every value of `R`).

Author(s)

Tom Reynkens with `R` code from Roel Verbelen for the estimates for the excess-loss premiums using the mixed Erlang distribution.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Reynkens, T., Verbelen, R., Beirlant, J. and Antonio, K. (2017). "Modelling Censored Losses Using Splicing: a Global Fit Strategy With Mixed Erlang and Extreme Value Distributions". *Insurance: Mathematics and Economics*, 77, 65–77.

Verbelen, R., Gong, L., Antonio, K., Badescu, A. and Lin, S. (2015). "Fitting Mixtures of Erlangs to Censored and Truncated Data Using the EM Algorithm." *Astin Bulletin*, 45, 729–758

See Also

`SpliceFit`, `SpliceFitPareto`, `SpliceFiticPareto`, `SpliceFitGPD`

Examples

```
## Not run:

# Pareto random sample
X <- rpareto(1000, shape = 2)

# Splice ME and Pareto
splicefit <- SpliceFitPareto(X, 0.8)

# Excess-loss premium
ExcessSplice(R=2, splicefit=splicefit)

## End(Not run)
```

ExpQQ *Exponential quantile plot*

Description

Computes the empirical quantiles of a data vector and the theoretical quantiles of the standard exponential distribution. These quantiles are then plotted in an exponential QQ-plot with the theoretical quantiles on the x-axis and the empirical quantiles on the y-axis.

Usage

```
ExpQQ(data, plot = TRUE, main = "Exponential QQ-plot", ...)
```

Arguments

`data`	Vector of n observations.
`plot`	Logical indicating if the quantiles should be plotted in an Exponential QQ-plot, default is `TRUE`.
`main`	Title for the plot, default is `"Exponential QQ-plot"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The exponential QQ-plot is defined as

$$(-\log(1-i/(n+1)), X_{i,n})$$

for $i = 1, ..., n$, with $X_{i,n}$ the i-th order statistic of the data.

Note that the mean excess plot is the derivative plot of the Exponential QQ-plot.

See Section 4.1 of Albrecher et al. (2017) for more details.

Value

A list with following components:

`eqq.the`	Vector of the theoretical quantiles from a standard exponential distribution.
`eqq.emp`	Vector of the empirical quantiles from the data.

Author(s)

Tom Reynkens based on `S-Plus` code from Yuri Goegebeur.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

See Also

MeanExcess, LognormalQQ, ParetoQQ, WeibullQQ

Examples

```
data(norwegianfire)

# Exponential QQ-plot for Norwegian Fire Insurance data for claims in 1976.
ExpQQ(norwegianfire$size[norwegianfire$year==76])

# Pareto QQ-plot for Norwegian Fire Insurance data for claims in 1976.
ParetoQQ(norwegianfire$size[norwegianfire$year==76])
```

Extended Pareto *The Extended Pareto Distribution*

Description

Density, distribution function, quantile function and random generation for the Extended Pareto Distribution (EPD).

Usage

```
depd(x, gamma, kappa, tau = -1, log = FALSE)
pepd(x, gamma, kappa, tau = -1, lower.tail = TRUE, log.p = FALSE)
qepd(p, gamma, kappa, tau = -1, lower.tail = TRUE, log.p = FALSE)
repd(n, gamma, kappa, tau = -1)
```

Arguments

x	Vector of quantiles.
p	Vector of probabilities.
n	Number of observations.
gamma	The γ parameter of the EPD, a strictly positive number.
kappa	The κ parameter of the EPD. It should be larger than $\max\{-1, 1/\tau\}$.
tau	The τ parameter of the EPD, a strictly negative number. Default is -1.
log	Logical indicating if the densities are given as $\log(f)$, default is FALSE.
lower.tail	Logical indicating if the probabilities are of the form $P(X \le x)$ (TRUE) or $P(X > x)$ (FALSE). Default is TRUE.
log.p	Logical indicating if the probabilities are given as $\log(p)$, default is FALSE.

Details

The Cumulative Distribution Function (CDF) of the EPD is equal to $F(x) = 1 - (x(1 + \kappa - \kappa x^{\tau}))^{-1/\gamma}$ for all $x > 1$ and $F(x) = 0$ otherwise.

Note that an EPD random variable with $\tau = -1$ and $\kappa = \gamma/\sigma - 1$ is GPD distributed with $\mu = 1$, γ and σ.

Value

depd gives the density function evaluated in x, pepd the CDF evaluated in x and qepd the quantile function evaluated in p. The length of the result is equal to the length of x or p.

repd returns a random sample of length n.

Author(s)

Tom Reynkens.

References

Beirlant, J., Joossens, E. and Segers, J. (2009). "Second-Order Refined Peaks-Over-Threshold Modelling for Heavy-Tailed Distributions." *Journal of Statistical Planning and Inference*, 139, 2800–2815.

See Also

Pareto, GPD, Distributions

Examples

```
# Plot of the PDF
x <- seq(0, 10, 0.01)
plot(x, depd(x, gamma=1/2, kappa=1, tau=-1), xlab="x", ylab="PDF", type="l")

# Plot of the CDF
x <- seq(0, 10, 0.01)
plot(x, pepd(x, gamma=1/2, kappa=1, tau=-1), xlab="x", ylab="CDF", type="l")
```

Frechet *The Frechet distribution*

Description

Density, distribution function, quantile function and random generation for the Fréchet distribution (inverse Weibull distribution).

Usage

```
dfrechet(x, shape, loc = 0, scale = 1, log = FALSE)
pfrechet(x, shape, loc = 0, scale = 1, lower.tail = TRUE, log.p = FALSE)
qfrechet(p, shape, loc = 0, scale = 1, lower.tail = TRUE, log.p = FALSE)
rfrechet(n, shape, loc = 0, scale = 1)
```

Arguments

x	Vector of quantiles.
p	Vector of probabilities.
n	Number of observations.
shape	Shape parameter of the Fréchet distribution.
loc	Location parameter of the Fréchet distribution, default is 0.
scale	Scale parameter of the Fréchet distribution, default is 1.
log	Logical indicating if the densities are given as $\log(f)$, default is FALSE.
lower.tail	Logical indicating if the probabilities are of the form $P(X \leq x)$ (TRUE) or $P(X > x)$ (FALSE). Default is TRUE.
log.p	Logical indicating if the probabilities are given as $\log(p)$, default is FALSE.

Details

The Cumulative Distribution Function (CDF) of the Fréchet distribution is equal to $F(x) = \exp(-((x-loc)/scale)^{-shape})$ for all $x \geq loc$ and $F(x) = 0$ otherwise. Both shape and scale need to be strictly positive.

Value

dfrechet gives the density function evaluated in x, pfrechet the CDF evaluated in x and qfrechet the quantile function evaluated in p. The length of the result is equal to the length of x or p.

rfrechet returns a random sample of length n.

Author(s)

Tom Reynkens.

See Also

tFréchet, Distributions

Examples

```
# Plot of the PDF
x <- seq(1,10,0.01)
plot(x, dfrechet(x, shape=2), xlab="x", ylab="PDF", type="l")

# Plot of the CDF
x <- seq(1,10,0.01)
plot(x, pfrechet(x, shape=2), xlab="x", ylab="CDF", type="l")
```

`genHill` *Generalised Hill estimator*

Description

Computes the generalised Hill estimator for real extreme value indices as a function of the tail parameter k. Optionally, these estimates are plotted as a function of k.

Usage

```
genHill(data, gamma, logk = FALSE, plot = FALSE, add = FALSE,
        main = "Generalised Hill estimates of the EVI", ...)
```

Arguments

`data`	Vector of n observations.
`gamma`	Vector of $n-1$ estimates for the EVI, typically Hill estimates are used.
`logk`	Logical indicating if the estimates are plotted as a function of $\log(k)$ (`logk=TRUE`) or as a function of k. Default is `FALSE`.
`plot`	Logical indicating if the estimates should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Generalised Hill estimates of the EVI"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The generalised Hill estimator is an estimator for the slope of the k last points of the generalised QQ-plot:

$$\hat{\gamma}_{k,n}^{GH} = 1/k \sum_{j=1}^{k} \log UH_{j,n} - \log UH_{k+1,n}$$

with $UH_{j,n} = X_{n-j,n}H_{j,n}$ the UH scores and $H_{j,n}$ the Hill estimates. This is analogous to the (ordinary) Hill estimator which is the estimator of the slope of the k last points of the Pareto QQ-plot when using constrained least squares.

See Section 4.2.2 of Albrecher et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`gamma`	Vector of the corresponding generalised Hill estimates.

Author(s)

Tom Reynkens based on S-Plus code from Yuri Goegebeur.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

Beirlant, J., Vynckier, P. and Teugels, J.L. (1996). "Excess Function and Estimation of the Extreme-value Index". *Bernoulli*, 2, 293–318.

See Also

Hill, genQQ, Moment

Examples

```
data(soa)

# Hill estimator
H <- Hill(soa$size, plot=FALSE)
# Moment estimator
M <- Moment(soa$size)
# Generalised Hill estimator
gH <- genHill(soa$size, gamma=H$gamma)

# Plot estimates
plot(H$k[1:5000], M$gamma[1:5000], xlab="k", ylab=expression(gamma), type="l", ylim=c(0.2,0.5))
lines(H$k[1:5000], gH$gamma[1:5000], lty=2)
legend("topright", c("Moment", "Generalised Hill"), lty=1:2)
```

---

genQQ *Generalised quantile plot*

Description

Computes the empirical quantiles of the UH scores of a data vector and the theoretical quantiles of the standard exponential distribution. These quantiles are then plotted in a generalised QQ-plot with the theoretical quantiles on the x-axis and the empirical quantiles on the y-axis.

Usage

```
genQQ(data, gamma, plot = TRUE, main = "Generalised QQ-plot", ...)

generalizedQQ(data, gamma, plot = TRUE, main = "Generalised QQ-plot", ...)
```

Arguments

`data`	Vector of n observations.
`gamma`	Vector of $n-1$ estimates for the EVI, typically Hill estimates are used.
`plot`	Logical indicating if the quantiles should be plotted in a generalised QQ-plot, default is `TRUE`.
`main`	Title for the plot, default is `"Generalised QQ-plot"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The `generalizedQQ` function is the same function but with a different name for compatibility with the old `S-Plus` code.

The UH scores are defined as $UH_{j,n} = X_{n-j,n} H_{j,n}$ with $H_{j,n}$ the Hill estimates, but other positive estimates for the EVI can also be used. The appropriate positive estimates for the EVI need to be specified in `gamma`. The generalised QQ-plot then plots

$$(\log((n+1)/(k+1)), \log(X_{n-k,n} H_{k,n}))$$

for $k = 1, \ldots, n-1$.

See Section 4.2.2 of Albrecher et al. (2017) for more details.

Value

A list with following components:

`gqq.the`	Vector of the theoretical quantiles from a standard exponential distribution.
`gqq.emp`	Vector of the empirical quantiles from the logarithm of the UH scores.

Author(s)

Tom Reynkens based on `S-Plus` code from Yuri Goegebeur.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

Beirlant, J., Vynckier, P. and Teugels, J.L. (1996). "Excess Function and Estimation of the Extreme-value Index." *Bernoulli*, 2, 293–318.

See Also

`ParetoQQ`, `Hill`

Examples

```
data(soa)

# Compute Hill estimator
H <- Hill(soa$size[1:5000], plot=FALSE)$gamma

# Generalised QQ-plot
genQQ(soa$size[1:5000], gamma=H)
```

GPD *The generalised Pareto distribution*

Description

Density, distribution function, quantile function and random generation for the Generalised Pareto Distribution (GPD).

Usage

```
dgpd(x, gamma, mu = 0, sigma, log = FALSE)
pgpd(x, gamma, mu = 0, sigma, lower.tail = TRUE, log.p = FALSE)
qgpd(p, gamma, mu = 0, sigma, lower.tail = TRUE, log.p = FALSE)
rgpd(n, gamma, mu = 0, sigma)
```

Arguments

`x`	Vector of quantiles.
`p`	Vector of probabilities.
`n`	Number of observations.
`gamma`	The γ parameter of the GPD, a real number.
`mu`	The μ parameter of the GPD, a strictly positive number. Default is 0.
`sigma`	The σ parameter of the GPD, a strictly positive number.
`log`	Logical indicating if the densities are given as $\log(f)$, default is `FALSE`.
`lower.tail`	Logical indicating if the probabilities are of the form $P(X \leq x)$ (`TRUE`) or $P(X > x)$ (`FALSE`). Default is `TRUE`.
`log.p`	Logical indicating if the probabilities are given as $\log(p)$, default is `FALSE`.

Details

The Cumulative Distribution Function (CDF) of the GPD for $\gamma \neq 0$ is equal to $F(x) = 1 - (1 + \gamma(x-\mu)/\sigma)^{-1/\gamma}$ for all $x \geq \mu$ and $F(x) = 0$ otherwise. When $\gamma = 0$, the CDF is given by $F(x) = 1 - \exp((x-\mu)/\sigma)$ for all $x \geq \mu$ and $F(x) = 0$ otherwise.

Value

dgpd gives the density function evaluated in x, pgpd the CDF evaluated in x and qgpd the quantile function evaluated in p. The length of the result is equal to the length of x or p.

rgpd returns a random sample of length n.

Author(s)

Tom Reynkens.

References

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

See Also

tGPD, Pareto, EPD, Distributions

Examples

```
# Plot of the PDF
x <- seq(0, 10, 0.01)
plot(x, dgpd(x, gamma=1/2, sigma=5), xlab="x", ylab="PDF", type="l")

# Plot of the CDF
x <- seq(0, 10, 0.01)
plot(x, pgpd(x, gamma=1/2, sigma=5), xlab="x", ylab="CDF", type="l")
```

GPDfit *Fit GPD using MLE*

Description

Fit the Generalised Pareto Distribution (GPD) to data using Maximum Likelihood Estimation (MLE).

Usage

```
GPDfit(data, start = c(0.1, 1), warnings = FALSE)
```

Arguments

data	Vector of n observations.
start	Vector of length 2 containing the starting values for the optimisation. The first element is the starting value for the estimator of γ and the second element is the starting value for the estimator of σ. Default is c(0.1,1).
warnings	Logical indicating if possible warnings from the optimisation function are shown, default is FALSE.

Details

See Section 4.2.2 in Albrecher et al. (2017) for more details.

Value

A vector with the MLE estimate for the γ parameter of the GPD as the first component and the MLE estimate for the σ parameter of the GPD as the second component.

Author(s)

Tom Reynkens based on `S-Plus` code from Yuri Goegebeur and `R` code from Klaus Herrmann.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

See Also

`GPDmle`, `EPDfit`

Examples

```
data(soa)

# Look at last 500 observations of SOA data
SOAdata <- sort(soa$size)[length(soa$size)-(0:499)]

# Fit GPD to last 500 observations
res <- GPDfit(SOAdata-sort(soa$size)[500])
```

`GPDmle` *GPD-ML estimator*

Description

Fit the Generalised Pareto Distribution (GPD) to the exceedances (peaks) over a threshold using Maximum Likelihood Estimation (MLE). Optionally, these estimates are plotted as a function of k.

Usage

```
GPDmle(data, start = c(0.1,1), warnings = FALSE, logk = FALSE,
       plot = FALSE, add = FALSE, main = "POT estimates of the EVI", ...)

POT(data, start = c(0.1,1), warnings = FALSE, logk = FALSE,
    plot = FALSE, add = FALSE, main = "POT estimates of the EVI", ...)
```

Arguments

`data`	Vector of n observations.
`start`	Vector of length 2 containing the starting values for the optimisation. The first element is the starting value for the estimator of γ and the second element is the starting value for the estimator of σ. Default is `c(0.1,1)`.
`warnings`	Logical indicating if possible warnings from the optimisation function are shown, default is `FALSE`.
`logk`	Logical indicating if the estimates are plotted as a function of $\log(k)$ (`logk=TRUE`) or as a function of k. Default is `FALSE`.
`plot`	Logical indicating if the estimates of γ should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates of γ should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"POT estimates of the EVI"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The `POT` function is the same function but with a different name for compatibility with the old `S-Plus` code.

For each value of `k`, we look at the exceedances over the $(k+1)$th largest observation: $X_{n-k+j,n} - X_{n-k,n}$ for $j = 1, ..., k$, with $X_{j,n}$ the jth largest observation and n the sample size. The GPD is then fitted to these `k` exceedances using MLE which yields estimates for the parameters of the GPD: γ and σ.

See Section 4.2.2 in Albrecher et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`gamma`	Vector of the corresponding MLE estimates for the γ parameter of the GPD.
`sigma`	Vector of the corresponding MLE estimates for the σ parameter of the GPD.

Author(s)

Tom Reynkens based on `S-Plus` code from Yuri Goegebeur and `R` code from Klaus Herrmann.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

See Also

GPDfit, GPDresiduals, EPD

Examples

```
data(soa)

# Look at last 500 observations of SOA data
SOAdata <- sort(soa$size)[length(soa$size)-(0:499)]

# Plot GPD-ML estimates as a function of k
GPDmle(SOAdata, plot=TRUE)
```

GPDresiduals *GPD residual plot*

Description

Residual plot to check GPD fit for peaks over a threshold.

Usage

```
GPDresiduals(data, t, gamma, sigma, plot = TRUE,
             main = "GPD residual plot", ...)
```

Arguments

data	Vector of n observations.
t	The used threshold.
gamma	Estimate for the EVI obtained from GPDmle.
sigma	Estimate for σ obtained from GPDmle.
plot	Logical indicating if the residuals should be plotted, default is FALSE.
main	Title for the plot, default is "GPD residual plot".
...	Additional arguments for the plot function, see plot for more details.

Details

Consider the POT values $Y = X - t$ and the transformed variable

$$R = 1/\gamma \log(1 + \gamma/\sigma Y),$$

when $\gamma \neq 0$ and

$$R = Y/\sigma,$$

otherwise. We can assess the goodness-of-fit of the GPD when modelling POT values $Y = X - t$ by constructing an exponential QQ-plot of the transformed variable R since R is standard exponentially distributed if Y follows the GPD.

See Section 4.2.2 in Albrecher et al. (2017) for more details.

Value

A list with following components:

`res.the`	Vector of the theoretical quantiles from a standard exponential distribution.
`res.emp`	Vector of the empirical quantiles of R, see Details.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

See Also

`GPDfit`, `ExpQQ`

Examples

```
data(soa)

# Look at last 500 observations of SOA data
SOAdata <- sort(soa$size)[length(soa$size)-(0:499)]

# Plot POT-MLE estimates as a function of k
pot <- GPDmle(SOAdata, plot=TRUE)

# Residual plot
k <- 200
GPDresiduals(SOAdata, sort(SOAdata)[length(SOAdata)-k], pot$gamma[k], pot$sigma[k])
```

`Hill` *Hill estimator*

Description

Computes the Hill estimator for positive extreme value indices (Hill, 1975) as a function of the tail parameter k. Optionally, these estimates are plotted as a function of k.

Usage

```
Hill(data, k = TRUE, logk = FALSE, plot = FALSE, add = FALSE,
     main = "Hill estimates of the EVI", ...)
```

Arguments

`data`	Vector of n observations.
`k`	Logical indicating if the Hill estimates are plotted as a function of the tail parameter k (`k=TRUE`) or as a function of $\log(X_{n-k})$. Default is `TRUE`.
`logk`	Logical indicating if the Hill estimates are plotted as a function of $\log(k)$ (`logk=TRUE`) or as a function of k (`logk=FALSE`) when `k=TRUE`. Default is `FALSE`.
`plot`	Logical indicating if the estimates should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Hill estimates of the EVI"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The Hill estimator can be seen as the estimator of slope in the upper right corner (k last points) of the Pareto QQ-plot when using constrained least squares (the regression line has to pass through the point $(-\log((k+1)/(n+1)), \log X_{n-k})$). It is given by

$$H_{k,n} = 1/k \sum_{j=1}^{k} \log X_{n-j+1,n} - \log X_{n-k,n}.$$

See Section 4.2.1 of Albrecher et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`gamma`	Vector of the corresponding Hill estimates.

Author(s)

Tom Reynkens based on `S-Plus` code from Yuri Goegebeur.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

Hill, B. M. (1975). "A simple general approach to inference about the tail of a distribution." *Annals of Statistics*, 3, 1163–1173.

See Also

`ParetoQQ`, `Hill.2oQV`, `genHill`

Examples

```
data(norwegianfire)

# Plot Hill estimates as a function of k
Hill(norwegianfire$size[norwegianfire$year==76],plot=TRUE)
```

`Hill.2oQV` *Bias-reduced MLE (Quantile view)*

Description

Computes bias-reduced ML estimates of gamma based on the quantile view.

Usage

```
Hill.2oQV(data, start = c(1,1,1), warnings = FALSE, logk = FALSE,
          plot = FALSE, add = FALSE, main = "Estimates of the EVI", ...)
```

Arguments

`data`	Vector of n observations.
`start`	A vector of length 3 containing starting values for the first numerical optimisation (see Details). The elements are the starting values for the estimators of γ, μ and σ, respectively. Default is `c(1,1,1)`.
`warnings`	Logical indicating if possible warnings from the optimisation function are shown, default is `FALSE`.
`logk`	Logical indicating if the estimates are plotted as a function of $\log(k)$ (`logk=TRUE`) or as a function of k. Default is `FALSE`.
`plot`	Logical indicating if the estimates of γ should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates of γ should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Estimates of the EVI"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

See Section 4.2.1 of Albrecher et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`gamma`	Vector of the ML estimates for the EVI for each value of k.
`b`	Vector of the ML estimates for the parameter b in the regression model for each value of k.
`beta`	Vector of the ML estimates for the parameter β in the regression model for each value of k.

Author(s)

Tom Reynkens based on `S-Plus` code from Yuri Goegebeur and `R` code from Klaus Herrmann.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Dierckx, G., Goegebeur Y. and Matthys, G. (1999). "Tail Index Estimation and an Exponential Regression Model." *Extremes*, 2, 177–200.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

Examples

```
data(norwegianfire)

# Plot bias-reduced MLE (QV) as a function of k
Hill.2oQV(norwegianfire$size[norwegianfire$year==76],plot=TRUE)
```

`Hill.kopt` *Select optimal threshold for Hill estimator*

Description

Select optimal threshold for the Hill estimator by minimising the Asymptotic Mean Squared Error (AMSE).

Usage

```
Hill.kopt(data, start = c(1, 1, 1), warnings = FALSE, logk = FALSE,
          plot = FALSE, add = FALSE, main = "AMSE plot", ...)
```

Arguments

data	Vector of n observations.
start	A vector of length 3 containing starting values for the first numerical optimisation (see Hill.2oQV for more details). Default is c(1,1,1).
warnings	Logical indicating if possible warnings from the optimisation function are shown, default is FALSE.
logk	Logical indicating if the AMSE values are plotted as a function of $\log(k)$ (logk=TRUE) or as a function of k. Default is FALSE.
plot	Logical indicating if the AMSE values should be plotted as a function of k, default is FALSE.
add	Logical indicating if the optimal value for k should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "AMSE plot".
...	Additional arguments for the plot function, see plot for more details.

Details

See Section 4.2.1 of Albrecher et al. (2017) for more details.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
AMSE	Vector of the AMSE values for each value of k.
kopt	Optimal value of k corresponding to minimal $AMSE$ value.
gammaopt	Optimal value of the Hill estimator corresponding to minimal $AMSE$ value.

Author(s)

Tom Reynkens based on S-Plus code from Yuri Goegebeur.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

Beirlant J., Vynckier, P. and Teugels, J. (1996). "Tail Index Estimation, Pareto Quantile Plots, and Regression Diagnostics." *Journal of the American Statistical Association*, 91, 1659–1667.

See Also

Hill, Hill.2oQV

Examples

```
data(norwegianfire)

# Plot Hill estimator as a function of k
Hill(norwegianfire$size[norwegianfire$year==76],plot=TRUE)

# Add optimal value of k
Hill.kopt(norwegianfire$size[norwegianfire$year==76],add=TRUE)
```

---

icHill *Hill estimator for interval censored data*

Description

Computes the Hill estimator for positive extreme value indices, adapted for interval censoring, as a function of the tail parameter k. Optionally, these estimates are plotted as a function of k.

Usage

```
icHill(L, U, censored, trunclower = 0, truncupper = Inf,
     logk = FALSE, plot = TRUE, add = FALSE, main = "Hill estimates of the EVI", ...)
```

Arguments

L	Vector of length n with the lower boundaries of the intervals for interval censored data or the observed data for right censored data.
U	Vector of length n with the upper boundaries of the intervals.
censored	A logical vector of length n indicating if an observation is censored.
trunclower	Lower truncation point. Default is 0.
truncupper	Upper truncation point. Default is `Inf` (no upper truncation).
logk	Logical indicating if the estimates are plotted as a function of $\log(k)$ (`logk=TRUE`) or as a function of k. Default is `FALSE`.
plot	Logical indicating if the estimates of γ should be plotted as a function of k, default is `FALSE`.
add	Logical indicating if the estimates of γ should be added to an existing plot, default is `FALSE`.
main	Title for the plot, default is `"Hill estimates of the EVI"`.
...	Additional arguments for the `plot` function, see `plot` for more details.

Details

This estimator is given by

$$H^{TB}(x) = (\int_x^\infty (1 - \hat{F}^{TB}(u))/udu)/(1 - \hat{F}^{TB}(x)),$$

where $\hat{F}^{TB}$ is the Turnbull estimator for the CDF. More specifically, we use the values $x = \hat{Q}^{TB}(p)$ for $p = 1/(n+1), \ldots, (n-1)/(n+1)$ where $\hat{Q}^{TB}(p)$ is the empirical quantile function corresponding to the Turnbull estimator. We then denote

$$H_{k,n}^{TB} = H^{TB}(x_{n-k,n})$$

with

$$x_{n-k,n} = \hat{Q}^{TB}((n-k)/(n+1)) = \hat{Q}^{TB}(1 - (k+1)/(n+1)).$$

Right censored data should be entered as `L=l` and `U=truncupper`, and left censored data should be entered as `L=trunclower` and `U=u`.

If the **interval** package is installed, the `icfit` function is used to compute the Turnbull estimator. Otherwise, `survfit.formula` from **survival** is used.

Use `Hill` for non-censored data or `cHill` for right censored data.

See Section 4.3 in Albrecher et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`gamma`	Vector of the corresponding Hill estimates.
`X`	Vector of thresholds $x_{n-k,n}$ used when estimating γ.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

See Also

`cHill`, `Hill`, `MeanExcess_TB`, `icParetoQQ`, `Turnbull`, `icfit`

Examples

```
# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X,Y)

# Censoring indicator
censored <- (X>Y)

# Right boundary
U <- Z
U[censored] <- Inf

# Hill estimator adapted for interval censoring
icHill(Z, U, censored, ylim=c(0,1))

# Hill estimator adapted for right censoring
cHill(Z, censored, lty=2, add=TRUE)

# True value of gamma
abline(h=1/2, lty=3, col="blue")

# Legend
legend("topright", c("icHill", "cHill"), lty=1:2)
```

---

icParetoQQ *Pareto quantile plot for interval censored data*

Description

Pareto QQ-plot adapted for interval censored data using the Turnbull estimator.

Usage

```
icParetoQQ(L, U = L, censored, trunclower = 0, truncupper = Inf,
           plot = TRUE, main = "Pareto QQ-plot", ...)
```

Arguments

L	Vector of length n with the lower boundaries of the intervals for interval censored data or the observed data for right censored data.
U	Vector of length n with the upper boundaries of the intervals.
censored	A logical vector of length n indicating if an observation is censored.
trunclower	Lower truncation point. Default is 0.

`truncupper`	Upper truncation point. Default is `Inf` (no upper truncation).
`plot`	Logical indicating if the quantiles should be plotted in a Pareto QQ-plot, default is `TRUE`.
`main`	Title for the plot, default is `"Pareto QQ-plot"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The Pareto QQ-plot adapted for interval censoring is given by

$$(-\log(1 - F^{TB}(x_{j,n})), \log x_{j,n})$$

for $j = 1, \ldots, n-1$, where $\hat{F}^{TB}$ is the Turnbull estimator for the CDF and $x_{i,n} = \hat{Q}^{TB}(i/(n+1))$ with $\hat{Q}^{TB}(p)$ the empirical quantile function corresponding to the Turnbull estimator.

Right censored data should be entered as `L=l` and `U=truncupper`, and left censored data should be entered as `L=trunclower` and `U=u`.

If the **interval** package is installed, the `icfit` function is used to compute the Turnbull estimator. Otherwise, `survfit.formula` from **survival** is used.

Use `ParetoQQ` for non-censored data or `cParetoQQ` for right censored data.

See Section 4.3 in Albrecher et al. (2017) for more details.

Value

A list with following components:

`pqq.the`	Vector of the theoretical quantiles, see Details.
`pqq.emp`	Vector of the empirical quantiles from the log-transformed data.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

See Also

`cParetoQQ`, `ParetoQQ`, `icHill`, `Turnbull`, `icfit`

Examples

```
# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)
```

```
# Observed sample
Z <- pmin(X,Y)

# Censoring indicator
censored <- (X>Y)

# Right boundary
U <- Z
U[censored] <- Inf

# Pareto QQ-plot adapted for interval censoring
icParetoQQ(Z, U, censored)

# Pareto QQ-plot adapted for right censoring
cParetoQQ(Z, censored)
```

KaplanMeier *Kaplan-Meier estimator*

Description

Computes the Kaplan-Meier estimator for the survival function of right censored data.

Usage

```
KaplanMeier(x, data, censored, conf.type="plain", conf.int = 0.95)
```

Arguments

`x`	Vector with points to evaluate the estimator in.
`data`	Vector of n observations.
`censored`	Vector of n logicals indicating if an observation is right censored.
`conf.type`	Type of confidence interval, see `survfit.formula`. Default is `"plain"`.
`conf.int`	Confidence level of the two-sided confidence interval, see `survfit.formula`. Default is `0.95`.

Details

We consider the random right censoring model where one observes $Z = \min(X, C)$ where X is the variable of interest and C is the censoring variable.

This function is merely a wrapper for `survfit.formula` from **survival**.

This estimator is only suitable for right censored data. When the data are interval censored, one can use the Turnbull estimator implemented in `Turnbull`.

Value

A list with following components:

surv	A vector of length `length(x)` containing the Kaplan-Meier estimator evaluated in the elements of `x`.
fit	The output from the call to `survfit.formula`, an object of class `survfit`.

Author(s)

Tom Reynkens

References

Kaplan, E. L. and Meier, P. (1958). "Nonparametric Estimation from Incomplete Observations." *Journal of the American Statistical Association*, 53, 457–481.

See Also

`survfit.formula`, `Turnbull`

Examples

```
data <- c(1, 2.5, 3, 4, 5.5, 6, 7.5, 8.25, 9, 10.5)
censored <- c(0, 1, 0, 0, 1, 0, 1, 1, 0, 1)

x <- seq(0, 12, 0.1)

# Kaplan-Meier estimator
plot(x, KaplanMeier(x, data, censored)$surv, type="s", ylab="Kaplan-Meier estimator")
```

LognormalQQ *Log-normal quantile plot*

Description

Computes the empirical quantiles of the log-transform of a data vector and the theoretical quantiles of the standard normal distribution. These quantiles are then plotted in a log-normal QQ-plot with the theoretical quantiles on the x-axis and the empirical quantiles on the y-axis.

Usage

```
LognormalQQ(data, plot = TRUE, main = "Log-normal QQ-plot", ...)
```

Arguments

`data`	Vector of n observations.
`plot`	Logical indicating if the quantiles should be plotted in a log-normal QQ-plot, default is `TRUE`.
`main`	Title for the plot, default is `"Log-normal QQ-plot"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

By definition, a log-transformed log-normal random variable is normally distributed. We can thus obtain a log-normal QQ-plot from a normal QQ-plot by replacing the empirical quantiles of the data vector by the empirical quantiles from the log-transformed data. We hence plot

$$(\Phi^{-1}(i/(n+1)), \log(X_{i,n}))$$

for $i = 1, \ldots, n$, where Φ is the standard normal CDF.

See Section 4.1 of Albrecher et al. (2017) for more details.

Value

A list with following components:

`lnqq.the`	Vector of the theoretical quantiles from a standard normal distribution.
`lnqq.emp`	Vector of the empirical quantiles from the log-transformed data.

Author(s)

Tom Reynkens.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

See Also

`ExpQQ`, `ParetoQQ`, `WeibullQQ`

Examples

```
data(norwegianfire)

# Log-normal QQ-plot for Norwegian Fire Insurance data for claims in 1976.
LognormalQQ(norwegianfire$size[norwegianfire$year==76])
```

LognormalQQ_der *Derivative plot of the log-normal QQ-plot*

Description

Computes the derivative plot of the log-normal QQ-plot. These values can be plotted as a function of the data or as a function of the tail parameter k.

Usage

```
LognormalQQ_der(data, k = FALSE, plot = TRUE,
                main = "Derivative plot of log-normal QQ-plot", ...)
```

Arguments

data	Vector of n observations.
plot	Logical indicating if the derivative values should be plotted, default is TRUE.
k	Logical indicating if the derivative values are plotted as a function of the tail parameter k (k=TRUE) or as a function of the logarithm of the data (k=FALSE). Default is FALSE.
main	Title for the plot, default is "Derivative plot of log-normal QQ-plot".
...	Additional arguments for the plot function, see plot for more details.

Details

The derivative plot of a log-normal QQ-plot is

$$(k, H_{k,n}/N_{k,n})$$

or

$$(\log X_{n-k,n}, H_{k,n}/N_{k,n})$$

with $H_{k,n}$ the Hill estimates and

$$N_{k,n} = (n+1)/(k+1)\phi(\Phi^{-1}(a)) - \Phi^{-1}(a).$$

Here is $a = 1 - (k+1)/(n+1)$, ϕ the standard normal PDF and Φ the standard normal CDF.

See Section 4.1 of Albrecher et al. (2017) for more details.

Value

A list with following components:

xval	Vector of the x-values of the plot (k or $\log X_{n-k,n}$).
yval	Vector of the derivative values.

Author(s)

Tom Reynkens.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

See Also

LognormalQQ, Hill, MeanExcess, ParetoQQ_der, WeibullQQ_der

Examples

```
data(norwegianfire)

# Log-normal QQ-plot for Norwegian Fire Insurance data for claims in 1976.
LognormalQQ(norwegianfire$size[norwegianfire$year==76])

# Derivate plot
LognormalQQ_der(norwegianfire$size[norwegianfire$year==76])
```

LStail *Least Squares tail estimator*

Description

Computes the Least Squares (LS) estimates of the EVI based on the last k observations of the generalised QQ-plot.

Usage

```
LStail(data, rho = -1, lambda = 0.5, logk = FALSE, plot = FALSE, add = FALSE,
       main = "LS estimates of the EVI", ...)

TSfraction(data, rho = -1, lambda = 0.5, logk = FALSE, plot = FALSE, add = FALSE,
           main = "LS estimates of the EVI", ...)
```

Arguments

`data`	Vector of n observations.
`rho`	Estimate for ρ, or `NULL` when ρ needs to be estimated using the method of Beirlant et al. (2002). Default is `-1`.
`lambda`	Parameter used in the method of Beirlant et al. (2002), only used when `rho=NULL`. Default is `0.5`.
`logk`	Logical indicating if the estimates are plotted as a function of $\log(k)$ (`logk=TRUE`) or as a function of k. Default is `FALSE`.

plot	Logical indicating if the estimates of γ should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates of γ should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "LS estimates of the EVI".
...	Additional arguments for the plot function, see plot for more details.

Details

We estimate γ (EVI) and b using least squares on the following regression model (Beirlant et al., 2005): $Z_j = \gamma + b(n/k)(j/k)^{-\rho} + \epsilon_j$ with $Z_j = (j+1)\log(UH_{j,n}/UH_{j+1,n})$ and $UH_{j,n} = X_{n-j,n}H_{j,n}$, where $H_{j,n}$ is the Hill estimator with threshold $X_{n-j,n}$.

See Section 5.8 of Beirlant et al. (2004) for more details.

The function TSfraction is included for compatibility with the old S-Plus code.

Value

k	Vector of the values of the tail parameter k.
gamma	Vector of the corresponding LS estimates for the EVI.
b	Vector of the corresponding LS estimates for b.
rho	Vector of the estimates for ρ when rho=NULL or the given input for rho otherwise.

Author(s)

Tom Reynkens based on S-Plus code from Yuri Goegebeur.

References

Beirlant, J., Dierckx, G. and Guillou, A. (2005). "Estimation of the Extreme Value Index and Regression on Generalized Quantile Plots." *Bernoulli*, 11, 949–970.

Beirlant, J., Dierckx, G., Guillou, A. and Starica, C. (2002). "On Exponential Representations of Log-spacing of Extreme Order Statistics." *Extremes*, 5, 157–180.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

See Also

genQQ

Examples

```
data(soa)

# LS tail estimator
LStail(soa$size, plot=TRUE, ylim=c(0,0.5))
```

`MeanExcess` *Mean excess function*

Description

Computes the mean excess values for a vector of observations. These mean excess values can then be plotted as a function of the data or as a function of the tail parameter k.

Usage

```
MeanExcess(data, plot = TRUE, k = FALSE, main = "Mean excess plot", ...)
```

Arguments

`data`	Vector of n observations.
`plot`	Logical indicating if the mean excess values should be plotted in a mean excess plot, default is `TRUE`.
`k`	Logical indicating if the mean excess scores are plotted as a function of the tail parameter k (`k=TRUE`) or as a function of the data (`k=FALSE`). Default is `FALSE`.
`main`	Title for the plot, default is `"Mean excess plot"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The mean excess plot is

$$(k, e_{k,n})$$

or

$$(X_{n-k,n}, e_{k,n})$$

with

$$e_{k,n} = 1/k \sum_{j=1}^{k} X_{n-j+1,n} - X_{n-k,n}.$$

Note that the mean excess plot is the derivative plot of the Exponential QQ-plot.

See Section 4.1 of Albrecher et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter `k`.
`X`	Vector of the order statistics `data[n-k]` corresponding to the tail parameters in `k`.
`e`	Vector of the mean excess values corresponding to the tail parameters in `k`.

Author(s)

Tom Reynkens based on S-Plus code from Yuri Goegebeur.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

See Also

ExpQQ, LognormalQQ_der, ParetoQQ_der, WeibullQQ_der

Examples

```
data(norwegianfire)

# Mean excess plots for Norwegian Fire Insurance data for claims in 1976.

# Mean excess values as a function of k
MeanExcess(norwegianfire$size[norwegianfire$year==76], k=TRUE)

# Mean excess values as a function of the data
MeanExcess(norwegianfire$size[norwegianfire$year==76], k=FALSE)
```

MeanExcess_TB *Mean excess function using Turnbull estimator*

Description

Computes mean excess values using the Turnbull estimator. These mean excess values can then be plotted as a function of the empirical quantiles (computed using the Turnbull estimator) or as a function of the tail parameter k.

Usage

```
MeanExcess_TB(L, U = L, censored, trunclower = 0, truncupper = Inf,
              plot = TRUE, k = FALSE, intervalpkg = TRUE,
              main = "Mean excess plot", ...)
```

Arguments

`L`	Vector of length n with the lower boundaries of the intervals for interval censored data or the observed data for right censored data.
`U`	Vector of length n with the upper boundaries of the intervals. By default, they are equal to `L`.
`censored`	A logical vector of length n indicating if an observation is censored.
`trunclower`	Lower truncation point, default is 0.
`truncupper`	Upper truncation point, default is `Inf`.
`plot`	Logical indicating if the mean excess values should be plotted in a mean excess plot, default is `TRUE`.
`k`	Logical indicating if the mean excess values are plotted as a function of the tail parameter k (`k=TRUE`) or as a function of the empirical quantiles computed using the Turnbull estimator (`k=FALSE`). Default is `FALSE`.
`intervalpkg`	Logical indicating if the Turnbull estimator is computed using the implementation in the **interval** package if this package is installed. Default is `TRUE`.
`main`	Title for the plot, default is `"Mean excess plot"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The mean excess values are given by

$$\hat{e}^{TB}(v) = (\int_v^\infty 1 - \hat{F}^{TB}(u)du)/(1 - \hat{F}^{TB}(v))$$

where $\hat{F}^{TB}$ is the Turnbull estimator for the CDF. More specifically, we use the values $v = \hat{Q}^{TB}(p)$ for $p = 1/(n+1), \dots, (n-1)/(n+1)$ where $\hat{Q}^{TB}(p)$ is the empirical quantile function corresponding to the Turnbull estimator.

Right censored data should be entered as `L=l` and `U=truncupper`, and left censored data should be entered as `L=trunclower` and `U=u`.

If the **interval** package is installed and `intervalpkg=TRUE`, the `icfit` function is used to compute the Turnbull estimator. Otherwise, `survfit.formula` from **survival** is used.

Use `MeanExcess` for non-censored data.

See Section 4.3 in Albrecher et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter `k`.
`X`	Vector of the empirical quantiles, computed using the Turnbull estimator, corresponding to `(n-k)/(n+1)=1-(k+1)/(n+1)`.
`e`	Vector of the mean excess values corresponding to the tail parameters in `k`.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

See Also

MeanExcess, Turnbull, icfit

Examples

```
# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X, Y)

# Censoring indicator
censored <- (X>Y)

# Right boundary
U <- Z
U[censored] <- Inf

# Mean excess plot
MeanExcess_TB(Z, U, censored, k=FALSE)
```

MEfit *Mixed Erlang fit*

Description

Create an S3 object using a Mixed Erlang (ME) fit.

Usage

```
MEfit(p, shape, theta, M, M_initial = NULL)
```

Arguments

p	Vector of mixing weights, denoted by α in Verbelen et al. (2015).
shape	Vector of shape parameters r.
theta	Scale parameter θ.
M	Number of mixture components.
M_initial	Initial value provided for M. When NULL (default), not included in the object.

Details

The rate parameter λ used in Albrecher et al. (2017) is equal to $1/\theta$.

See Reynkens et al. (2017) and Section 4.3 of Albrecher et al. (2017) for more details

Value

An S3 object which contains the input arguments in a list.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Reynkens, T., Verbelen, R., Beirlant, J. and Antonio, K. (2017). "Modelling Censored Losses Using Splicing: a Global Fit Strategy With Mixed Erlang and Extreme Value Distributions". *Insurance: Mathematics and Economics*, 77, 65–77.

Verbelen, R., Gong, L., Antonio, K., Badescu, A. and Lin, S. (2015). "Fitting Mixtures of Erlangs to Censored and Truncated Data Using the EM Algorithm." *Astin Bulletin*, 45, 729–758

See Also

SpliceFit, SpliceFitPareto, SpliceFiticPareto, SpliceFitGPD

Examples

```
# Create MEfit object
mefit <- MEfit(p=c(0.65,0.35), shape=c(39,58), theta=16.19, M=2)

# Create EVTfit object
evtfit <- EVTfit(gamma=c(0.76,0.64), endpoint=c(39096, Inf))

# Create SpliceFit object
splicefit <- SpliceFit(const=c(0.5,0.996), trunclower=0, t=c(1020,39096), type=c("ME","TPa","Pa"),
                       MEfit=mefit, EVTfit=evtfit)

# Show summary
summary(splicefit)
```

`Moment` *Moment estimator*

Description

Compute the moment estimates for real extreme value indices as a function of the tail parameter k. Optionally, these estimates are plotted as a function of k.

Usage

```
Moment(data, logk = FALSE, plot = FALSE, add = FALSE,
       main = "Moment estimates of the EVI", ...)
```

Arguments

`data`	Vector of n observations.
`logk`	Logical indicating if the estimates are plotted as a function of $\log(k)$ (`logk=TRUE`) or as a function of k. Default is `FALSE`.
`plot`	Logical indicating if the estimates should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Moment estimates of the EVI"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The moment estimator for the EVI is introduced by Dekkers et al. (1989) and is a generalisation of the Hill estimator.

See Section 4.2.2 of Albrecher et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`gamma`	Vector of the corresponding moment estimates.

Author(s)

Tom Reynkens based on `S-Plus` code from Yuri Goegebeur.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

Dekkers, A.L.M, Einmahl, J.H.J. and de Haan, L. (1989). "A Moment Estimator for the Index of an Extreme-value Distribution." *Annals of Statistics*, 17, 1833–1855.

See Also

Hill, genHill

Examples

```
data(soa)

# Hill estimator
H <- Hill(soa$size, plot=FALSE)
# Moment estimator
M <- Moment(soa$size)
# Generalised Hill estimator
gH <- genHill(soa$size, gamma=H$gamma)

# Plot estimates
plot(H$k[1:5000], M$gamma[1:5000], xlab="k", ylab=expression(gamma), type="l", ylim=c(0.2,0.5))
lines(H$k[1:5000], gH$gamma[1:5000], lty=2)
legend("topright", c("Moment", "Generalised Hill"), lty=1:2)
```

---

norwegianfire *Norwegian fire insurance data*

Description

Fire insurance claims for a Norwegian insurance company for the period 1972 to 1992 as studied in Beirlant et al. (1996).
A priority of 500 units was in force.

Usage

```
data("norwegianfire")
```

Format

A data frame with 9181 observations on the following 2 variables:

`size` Size of fire insurance claim (in 1000 NOK).

`year` Year of claim occurence (expressed as yy instead of $19yy$).

Source

Beirlant, J., Teugels, J. L. and Vynckier, P. (1996). *Practical Analysis of Extreme Values*, Leuven University Press.

References

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

Examples

```
data(norwegianfire)

# Exponential QQ-plot for Norwegian Fire Insurance data for claims in 1976.
ExpQQ(norwegianfire$size[norwegianfire$year==76])

# Pareto QQ-plot for Norwegian Fire Insurance data for claims in 1976.
ParetoQQ(norwegianfire$size[norwegianfire$year==76])
```

Pareto — *The Pareto distribution*

Description

Density, distribution function, quantile function and random generation for the Pareto distribution (type I).

Usage

```
dpareto(x, shape, scale = 1, log = FALSE)
ppareto(x, shape, scale = 1, lower.tail = TRUE, log.p = FALSE)
qpareto(p, shape, scale = 1, lower.tail = TRUE, log.p = FALSE)
rpareto(n, shape, scale = 1)
```

Arguments

`x`	Vector of quantiles.
`p`	Vector of probabilities.
`n`	Number of observations.
`shape`	The shape parameter of the Pareto distribution, a strictly positive number.
`scale`	The scale parameter of the Pareto distribution, a strictly positive number. Its default value is `1`.
`log`	Logical indicating if the densities are given as $\log(f)$, default is `FALSE`.
`lower.tail`	Logical indicating if the probabilities are of the form $P(X \leq x)$ (`TRUE`) or $P(X > x)$ (`FALSE`). Default is `TRUE`.
`log.p`	Logical indicating if the probabilities are given as $\log(p)$, default is `FALSE`.

Details

The Cumulative Distribution Function (CDF) of the Pareto distribution is equal to $F(x) = 1 - (x/scale)^{-shape}$ for all $x \geq scale$ and $F(x) = 0$ otherwise. Both `shape` and `scale` need to be strictly positive.

Value

`dpareto` gives the density function evaluated in x, `ppareto` the CDF evaluated in x and `qpareto` the quantile function evaluated in p. The length of the result is equal to the length of x or p.

`rpareto` returns a random sample of length n.

Author(s)

Tom Reynkens.

See Also

`tPareto`, `GPD`, `Distributions`

Examples

```
# Plot of the PDF
x <- seq(1, 10, 0.01)
plot(x, dpareto(x, shape=2), xlab="x", ylab="PDF", type="l")

# Plot of the CDF
x <- seq(1, 10, 0.01)
plot(x, ppareto(x, shape=2), xlab="x", ylab="CDF", type="l")
```

`ParetoQQ` *Pareto quantile plot*

Description

Computes the empirical quantiles of the log-transform of a data vector and the theoretical quantiles of the standard exponential distribution. These quantiles are then plotted in a Pareto QQ-plot with the theoretical quantiles on the x-axis and the empirical quantiles on the y-axis.

Usage

```
ParetoQQ(data, plot = TRUE, main = "Pareto QQ-plot", ...)
```

Arguments

data	Vector of n observations.
plot	Logical indicating if the quantiles should be plotted in a Pareto QQ-plot, default is TRUE.
main	Title for the plot, default is "Pareto QQ-plot".
...	Additional arguments for the plot function, see plot for more details.

Details

It can be easily seen that a log-transformed Pareto random variable is exponentially distributed. We can hence obtain a Pareto QQ-plot from an exponential QQ-plot by replacing the empirical quantiles from the data vector by the empirical quantiles from the log-transformed data. We hence plot

$$(-\log(1 - i/(n + 1)), \log X_{i,n})$$

for $i = 1, ..., n$, with $X_{i,n}$ the i-th order statistic of the data.

See Section 4.1 of Albrecher et al. (2017) for more details.

Value

A list with following components:

pqq.the	Vector of the theoretical quantiles from a standard exponential distribution.
pqq.emp	Vector of the empirical quantiles from the log-transformed data.

Author(s)

Tom Reynkens based on S-Plus code from Yuri Goegebeur.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

See Also

ParetoQQ_der, ExpQQ, genQQ, LognormalQQ, WeibullQQ

Examples

```
data(norwegianfire)

# Exponential QQ-plot for Norwegian Fire Insurance data for claims in 1976.
ExpQQ(norwegianfire$size[norwegianfire$year==76])

# Pareto QQ-plot for Norwegian Fire Insurance data for claims in 1976.
ParetoQQ(norwegianfire$size[norwegianfire$year==76])
```

ParetoQQ_der *Derivative plot of the Pareto QQ-plot*

Description

Computes the derivative plot of the Pareto QQ-plot. These values can be plotted as a function of the data or as a function of the tail parameter k.

Usage

```
ParetoQQ_der(data, k = FALSE, plot = TRUE,
             main = "Derivative plot of Pareto QQ-plot", ...)
```

Arguments

`data`	Vector of n observations.
`plot`	Logical indicating if the derivative values should be plotted, default is `TRUE`.
`k`	Logical indicating if the derivative values are plotted as a function of the tail parameter k (`k=TRUE`) or as a function of the logarithm of the data (`k=FALSE`). Default is `FALSE`.
`main`	Title for the plot, default is `"Derivative plot of Pareto QQ-plot"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The derivative plot of a Pareto QQ-plot is

$$(k, H_{k,n})$$

or

$$(\log X_{n-k,n}, H_{k,n})$$

with $H_{k,n}$ the Hill estimates.

See Section 4.1 of Albrecher et al. (2017) for more details.

Value

A list with following components:

`xval`	Vector of the x-values of the plot (k or $\log X_{n-k,n}$).
`yval`	Vector of the derivative values $H_{k,n}$.

Author(s)

Tom Reynkens.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

See Also

ParetoQQ, Hill, MeanExcess, LognormalQQ_der, WeibullQQ_der

Examples

```
data(norwegianfire)

# Pareto QQ-plot for Norwegian Fire Insurance data for claims in 1976.
ParetoQQ(norwegianfire$size[norwegianfire$year==76])

# Derivate plot
ParetoQQ_der(norwegianfire$size[norwegianfire$year==76])
```

---

pClas — *Classical estimators for the CDF*

Description

Compute approximations of the CDF using the normal approximation, normal-power approximation, shifted Gamma approximation or normal approximation to the shifted Gamma distribution.

Usage

```
pClas(x, mean = 0, variance = 1, skewness = NULL,
    method = c("normal", "normal-power", "shifted Gamma", "shifted Gamma normal"),
      lower.tail = TRUE, log.p = FALSE)
```

Arguments

x	Vector of points to approximate the CDF in.
mean	Mean of the distribution, default is 0.
variance	Variance of the distribution, default is 1.
skewness	Skewness coefficient of the distribution, this argument is not used for the normal approximation. Default is NULL meaning no skewness coefficient is provided.
method	Approximation method to use, one of "normal", "normal-power", "shifted Gamma" or "shifted Gamma normal". Default is "normal".
lower.tail	Logical indicating if the probabilities are of the form $P(X \le x)$ (TRUE) or $P(X > x)$ (FALSE). Default is TRUE.
log.p	Logical indicating if the probabilities are given as $\log(p)$, default is FALSE.

Details

- The *normal approximation* for the CDF of the r.v. X is defined as

  $$F_X(x) \approx \Phi((x-\mu)/\sigma)$$

  where μ and σ^2 are the mean and variance of X, respectively.
- This approximation can be improved when the skewness parameter

  $$\nu = E((X-\mu)^3)/\sigma^3$$

  is available. The *normal-power approximation* of the CDF is then given by

  $$F_X(x) \approx \Phi(\sqrt{9/\nu^2 + 6z/\nu + 1} - 3/\nu)$$

  for $z = (x-\mu)/\sigma \geq 1$ and $9/\nu^2 + 6z/\nu + 1 \geq 0$.
- The *shifted Gamma approximation* uses the approximation

  $$X \approx \Gamma(4/\nu^2, 2/(\nu \times \sigma)) + \mu - 2\sigma/\nu.$$

  Here, we need that $\nu > 0$.
- The *normal approximation to the shifted Gamma distribution* approximates the CDF of X as

  $$F_X(x) \approx \Phi(\sqrt{16/\nu^2 + 8z/\nu} - \sqrt{16/\nu^2 - 1})$$

  for $z = (x-\mu)/\sigma \geq 1$. We need again that $\nu > 0$.

See Section 6.2 of Albrecher et al. (2017) for more details.

Value

Vector of estimates for the probabilities $F(x) = P(X \leq x)$.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

See Also

pEdge, pGC

Examples

```
# Chi-squared sample
X <- rchisq(1000, 2)

x <- seq(0, 10, 0.01)

# Classical approximations
p1 <- pClas(x, mean(X), var(X))
p2 <- pClas(x, mean(X), var(X), mean((X-mean(X))^3)/sd(X)^3, method="normal-power")
p3 <- pClas(x, mean(X), var(X), mean((X-mean(X))^3)/sd(X)^3, method="shifted Gamma")
p4 <- pClas(x, mean(X), var(X), mean((X-mean(X))^3)/sd(X)^3, method="shifted Gamma normal")

# True probabilities
p <- pchisq(x, 2)


# Plot true and estimated probabilities
plot(x, p, type="l", ylab="F(x)", ylim=c(0,1), col="red")
lines(x, p1, lty=2)
lines(x, p2, lty=3, col="green")
lines(x, p3, lty=4)
lines(x, p4, lty=5, col="blue")

legend("bottomright", c("True CDF", "normal approximation", "normal-power approximation",
                    "shifted Gamma approximation", "shifted Gamma normal approximation"),
      lty=1:5, col=c("red", "black", "green", "black", "blue"), lwd=2)
```

---

pEdge *Edgeworth approximation*

Description

Edgeworth approximation of the CDF using the first four moments.

Usage

```
pEdge(x, moments = c(0, 1, 0, 3), raw = TRUE, lower.tail = TRUE, log.p = FALSE)
```

Arguments

`x`	Vector of points to approximate the CDF in.
`moments`	The first four raw moments if `raw=TRUE`. By default the first four raw moments of the standard normal distribution are used. When `raw=FALSE`, the mean $\mu = E(X)$, variance $\sigma^2 = E((X-\mu)^2)$, skewness (third standardised moment, $\nu = E((X-\mu)^3)/\sigma^3$) and kurtosis (fourth standardised moment, $k = E((X-\mu)^4)/\sigma^4$).
`raw`	When `TRUE` (default), the first four raw moments are provided in `moments`. Otherwise, the mean, variance, skewness and kurtosis are provided in `moments`.

`lower.tail` Logical indicating if the probabilities are of the form $P(X \leq x)$ (`TRUE`) or $P(X > x)$ (`FALSE`). Default is `TRUE`.

`log.p` Logical indicating if the probabilities are given as $\log(p)$, default is `FALSE`.

Details

Denote the standard normal PDF and CDF respectively by ϕ and Φ. Let μ be the first moment, $\sigma^2 = E((X-\mu)^2)$ the variance, $\mu_3 = E((X-\mu)^3)$ the third central moment and $\mu_4 = E((X-\mu)^4)$ the fourth central moment of the random variable X. The corresponding cumulants are given by $\kappa_1 = \mu$, $\kappa_2 = \sigma^2$, $\kappa_3 = \mu_3$ and $\kappa_4 = \mu_4 - 3\sigma^4$.

Now consider the random variable $Z = (X-\mu)/\sigma$, which has cumulants 0, 1, $\nu = \kappa_3/\sigma^3$ and $k = \kappa_4/\sigma^4 = \mu_4/\sigma^4 - 3$.

The Edgeworth approximation for the CDF of X ($F(x)$) is given by

$$\hat{F}_E(x) = \Phi(z) + \phi(z)(-\nu/6h_2(z) - (3k \times h_3(z) + \gamma_3^2 h_5(z))/72)$$

with $h_2(z) = z^2 - 1$, $h_3(z) = z^3 - 3z$, $h_5(z) = z^5 - 10z^3 + 15z$ and $z = (x-\mu)/\sigma$.

See Section 6.2 of Albrecher et al. (2017) for more details.

Value

Vector of estimates for the probabilities $F(x) = P(X \leq x)$.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Cheah, P.K., Fraser, D.A.S. and Reid, N. (1993). "Some Alternatives to Edgeworth." *The Canadian Journal of Statistics*, 21(2), 131–138.

See Also

pGC, pEdge

Examples

```
# Chi-squared sample
X <- rchisq(1000, 2)


x <- seq(0, 10, 0.01)

# Empirical moments
moments = c(mean(X), mean(X^2), mean(X^3), mean(X^4))
```

```
# Gram-Charlier approximation
p1 <- pGC(x, moments)

# Edgeworth approximation
p2 <- pEdge(x, moments)

# Normal approximation
p3 <- pClas(x, mean(X), var(X))

# True probabilities
p <- pchisq(x, 2)


# Plot true and estimated probabilities
plot(x, p, type="l", ylab="F(x)", ylim=c(0,1), col="red")
lines(x, p1, lty=2)
lines(x, p2, lty=3)
lines(x, p3, lty=4, col="blue")

legend("bottomright", c("True CDF", "GC approximation",
                        "Edgeworth approximation", "Normal approximation"),
       col=c("red", "black", "black", "blue"), lty=1:4, lwd=2)
```

---

pGC *Gram-Charlier approximation*

Description

Gram-Charlier approximation of the CDF using the first four moments.

Usage

```
pGC(x, moments = c(0, 1, 0, 3), raw = TRUE, lower.tail = TRUE, log.p = FALSE)
```

Arguments

`x`	Vector of points to approximate the CDF in.
`moments`	The first four raw moments if `raw=TRUE`. By default the first four raw moments of the standard normal distribution are used. When `raw=FALSE`, the mean $\mu = E(X)$, variance $\sigma^2 = E((X-\mu)^2)$, skewness (third standardised moment, $\nu = E((X-\mu)^3)/\sigma^3$) and kurtosis (fourth standardised moment, $k = E((X-\mu)^4)/\sigma^4$).
`raw`	When `TRUE` (default), the first four raw moments are provided in `moments`. Otherwise, the mean, variance, skewness and kurtosis are provided in `moments`.
`lower.tail`	Logical indicating if the probabilities are of the form $P(X \leq x)$ (`TRUE`) or $P(X > x)$ (`FALSE`). Default is `TRUE`.
`log.p`	Logical indicating if the probabilities are given as $\log(p)$, default is `FALSE`.

Details

Denote the standard normal PDF and CDF respectively by ϕ and Φ. Let μ be the first moment, $\sigma^2 = E((X-\mu)^2)$ the variance, $\mu_3 = E((X-\mu)^3)$ the third central moment and $\mu_4 = E((X-\mu)^4)$ the fourth central moment of the random variable X. The corresponding cumulants are given by $\kappa_1 = \mu$, $\kappa_2 = \sigma^2$, $\kappa_3 = \mu_3$ and $\kappa_4 = \mu_4 - 3\sigma^4$.

Now consider the random variable $Z = (X-\mu)/\sigma$, which has cumulants 0, 1, $\nu = \kappa_3/\sigma^3$ and $k = \kappa_4/\sigma^4 = \mu_4/\sigma^4 - 3$.

The Gram-Charlier approximation for the CDF of X ($F(x)$) is given by

$$\hat{F}_{GC}(x) = \Phi(z) + \phi(z)(-\nu/6h_2(z) - k/24h_3(z))$$

with $h_2(z) = z^2 - 1$, $h_3(z) = z^3 - 3z$ and $z = (x-\mu)/\sigma$.

See Section 6.2 of Albrecher et al. (2017) for more details.

Value

Vector of estimates for the probabilities $F(x) = P(X \le x)$.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Cheah, P.K., Fraser, D.A.S. and Reid, N. (1993). "Some Alternatives to Edgeworth." *The Canadian Journal of Statistics*, 21(2), 131–138.

See Also

pEdge, pClas

Examples

```
# Chi-squared sample
X <- rchisq(1000, 2)


x <- seq(0, 10, 0.01)

# Empirical moments
moments = c(mean(X), mean(X^2), mean(X^3), mean(X^4))

# Gram-Charlier approximation
p1 <- pGC(x, moments)

# Edgeworth approximation
p2 <- pEdge(x, moments)
```

```
# Normal approximation
p3 <- pClas(x, mean(X), var(X))

# True probabilities
p <- pchisq(x, 2)


# Plot true and estimated probabilities
plot(x, p, type="l", ylab="F(x)", ylim=c(0,1), col="red")
lines(x, p1, lty=2)
lines(x, p2, lty=3)
lines(x, p3, lty=4, col="blue")

legend("bottomright", c("True CDF", "GC approximation",
                        "Edgeworth approximation", "Normal approximation"),
       col=c("red", "black", "black", "blue"), lty=1:4, lwd=2)
```

`Prob` *Weissman estimator of small exceedance probabilities and large return periods*

Description

Compute estimates of a small exceedance probability $P(X > q)$ or large return period $1/P(X > q)$ using the approach of Weissman (1978).

Usage

```
Prob(data, gamma, q, plot = FALSE, add = FALSE,
     main = "Estimates of small exceedance probability", ...)

Return(data, gamma, q, plot = FALSE, add = FALSE,
       main = "Estimates of large return period", ...)

Weissman.p(data, gamma, q, plot = FALSE, add = FALSE,
           main = "Estimates of small exceedance probability", ...)

Weissman.r(data, gamma, q, plot = FALSE, add = FALSE,
           main = "Estimates of large return period", ...)
```

Arguments

`data`	Vector of n observations.
`gamma`	Vector of $n-1$ estimates for the EVI, typically Hill estimates are used.
`q`	The used large quantile (we estimate $P(X > q)$ or $1/P(X > q)$ for q large).
`plot`	Logical indicating if the estimates should be plotted as a function of k, default is `FALSE`.

add	Logical indicating if the estimates should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "Estimates of extreme quantile" for Prob and "Estimates of large return period" for Return.
...	Additional arguments for the plot function, see plot for more details.

Details

See Section 4.2.1 of Albrecher et al. (2017) for more details.

Weissman.p and Weissman.r are the same functions as Prob and Return but with a different name for compatibility with the old S-Plus code.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
P	Vector of the corresponding probability estimates, only returned for Prob.
R	Vector of the corresponding estimates for the return period, only returned for Return.
q	The used large quantile.

Author(s)

Tom Reynkens based on S-Plus code from Yuri Goegebeur.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

Weissman, I. (1978). "Estimation of Parameters and Large Quantiles Based on the *k* Largest Observations." *Journal of the American Statistical Association*, 73, 812–815.

See Also

Quant

Examples

```
data(soa)

# Look at last 500 observations of SOA data
SOAdata <- sort(soa$size)[length(soa$size)-(0:499)]

# Hill estimator
H <- Hill(SOAdata)
```

```
# Exceedance probability
q <- 10^6
# Weissman estimator
Prob(SOAdata,gamma=H$gamma,q=q,plot=TRUE)

# Return period
q <- 10^6
# Weissman estimator
Return(SOAdata,gamma=H$gamma,q=q,plot=TRUE)
```

ProbEPD — *Estimator of small exceedance probabilities and large return periods using EPD*

Description

Computes estimates of a small exceedance probability $P(X > q)$ or large return period $1/P(X > q)$ using the parameters from the EPD fit.

Usage

```
ProbEPD(data, q, gamma, kappa, tau, plot = FALSE, add = FALSE,
        main = "Estimates of small exceedance probability", ...)

ReturnEPD(data, q, gamma, kappa, tau, plot = FALSE, add = FALSE,
          main = "Estimates of large return period", ...)
```

Arguments

`data`	Vector of n observations.
`q`	The used large quantile (we estimate $P(X > q)$ or $1/P(X > q)$ for q large).
`gamma`	Vector of $n - 1$ estimates for the EVI obtained from `EPD`.
`kappa`	Vector of $n - 1$ estimates for κ obtained from `EPD`.
`tau`	Vector of $n - 1$ estimates for τ obtained from `EPD`.
`plot`	Logical indicating if the estimates should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Estimates of small exceedance probability"` for `ProbEPD` and `"Estimates of large return period"` for `ReturnEPD`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

See Section 4.2.1 of Albrecher et al. (2017) for more details.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
P	Vector of the corresponding probability estimates, only returned for `ProbEPD`.
R	Vector of the corresponding estimates for the return period, only returned for `ReturnEPD`.
q	The used large quantile.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant, J., Joossens, E. and Segers, J. (2009). "Second-Order Refined Peaks-Over-Threshold Modelling for Heavy-Tailed Distributions." *Journal of Statistical Planning and Inference*, 139, 2800–2815.

See Also

`EPD`, `Prob`

Examples

```
data(secura)

# EPD estimates for the EVI
epd <- EPD(secura$size, plot=TRUE)

# Compute exceedance probabilities
q <- 10^7
ProbEPD(secura$size, q=q, gamma=epd$gamma, kappa=epd$kappa, tau=epd$tau, plot=TRUE)

# Compute return periods
ReturnEPD(secura$size, q=q, gamma=epd$gamma, kappa=epd$kappa, tau=epd$tau,
          plot=TRUE, ylim=c(0,10^4))
```

ProbGH *Estimator of small exceedance probabilities and large return periods using generalised Hill*

Description

Computes estimates of a small exceedance probability $P(X > q)$ or large return period $1/P(X > q)$ using the generalised Hill estimates for the EVI.

Usage

```
ProbGH(data, gamma, q, plot = FALSE, add = FALSE,
       main = "Estimates of small exceedance probability", ...)

ReturnGH(data, gamma, q, plot = FALSE, add = FALSE,
         main = "Estimates of large return period", ...)
```

Arguments

data	Vector of n observations.
gamma	Vector of $n - 2$ estimates for the EVI obtained from genHill.
q	The used large quantile (we estimate $P(X > q)$ or $1/P(X > q)$ for q large).
plot	Logical indicating if the estimates should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "Estimates of small exceedance probability" for ProbGH and "Estimates of large return period" for ReturnGH.
...	Additional arguments for the plot function, see plot for more details.

Details

See Section 4.2.2 of Albrecher et al. (2017) for more details.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
P	Vector of the corresponding probability estimates, only returned for ProbGH.
R	Vector of the corresponding estimates for the return period, only returned for ReturnGH.
q	The used large quantile.

Author(s)

Tom Reynkens.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

Beirlant, J., Vynckier, P. and Teugels, J.L. (1996). "Excess Function and Estimation of the Extreme-value Index". *Bernoulli*, 2, 293–318.

See Also

QuantGH, genHill, ProbMOM, Prob

Examples

```
data(soa)

# Look at last 500 observations of SOA data
SOAdata <- sort(soa$size)[length(soa$size)-(0:499)]

# Hill estimator
H <- Hill(SOAdata)
# Generalised Hill estimator
gH <- genHill(SOAdata, H$gamma)

# Exceedance probability
q <- 10^7
ProbGH(SOAdata, gamma=gH$gamma, q=q, plot=TRUE)

# Return period
q <- 10^7
ReturnGH(SOAdata, gamma=gH$gamma, q=q, plot=TRUE)
```

ProbGPD — *Estimator of small exceedance probabilities and large return periods using GPD-MLE*

Description

Computes estimates of a small exceedance probability $P(X > q)$ or large return period $1/P(X > q)$ using the GPD fit for the peaks over a threshold.

Usage

```
ProbGPD(data, gamma, sigma, q, plot = FALSE, add = FALSE,
        main = "Estimates of small exceedance probability", ...)

ReturnGPD(data, gamma, sigma, q, plot = FALSE, add = FALSE,
          main = "Estimates of large return period", ...)
```

Arguments

`data`	Vector of n observations.
`gamma`	Vector of $n-1$ estimates for the EVI obtained from `GPDmle`.
`sigma`	Vector of $n-1$ estimates for σ obtained from `GPDmle`.
`q`	The used large quantile (we estimate $P(X > q)$ or $1/P(X > q)$ for q large).
`plot`	Logical indicating if the estimates should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Estimates of small exceedance probability"` for `ProbGPD` and `"Estimates of large return period"` for `ReturnGPD`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

See Section 4.2.2 in Albrecher et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`P`	Vector of the corresponding probability estimates, only returned for `ProbGPD`.
`R`	Vector of the corresponding estimates for the return period, only returned for `ReturnGPD`.
`q`	The used large quantile.

Author(s)

Tom Reynkens.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

See Also

QuantGPD, GPDmle, Prob

Examples

```
data(soa)

# Look at last 500 observations of SOA data
SOAdata <- sort(soa$size)[length(soa$size)-(0:499)]

# GPD-ML estimator
pot <- GPDmle(SOAdata)

# Exceedance probability
q <- 10^7
ProbGPD(SOAdata, gamma=pot$gamma, sigma=pot$sigma, q=q, plot=TRUE)

# Return period
q <- 10^7
ReturnGPD(SOAdata, gamma=pot$gamma, sigma=pot$sigma, q=q, plot=TRUE)
```

ProbMOM *Estimator of small exceedance probabilities and large return periods using MOM*

Description

Computes estimates of a small exceedance probability $P(X > q)$ or large return period $1/P(X > q)$ using the Method of Moments estimates for the EVI.

Usage

```
ProbMOM(data, gamma, q, plot = FALSE, add = FALSE,
        main = "Estimates of small exceedance probability", ...)

ReturnMOM(data, gamma, q, plot = FALSE, add = FALSE,
          main = "Estimates of large return period", ...)
```

Arguments

data	Vector of n observations.
gamma	Vector of $n-1$ estimates for the EVI obtained from Moment.
q	The used large quantile (we estimate $P(X > q)$ or $1/P(X > q)$ for q large).
plot	Logical indicating if the estimates should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates should be added to an existing plot, default is FALSE.

`main`	Title for the plot, default is `"Estimates of small exceedance probability"` for `ProbMOM` and `"Estimates of large return period"` for `ReturnMOM`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

See Section 4.2.2 of Albrecher et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`P`	Vector of the corresponding probability estimates, only returned for `ProbMOM`.
`R`	Vector of the corresponding estimates for the return period, only returned for `ReturnMOM`.
`q`	The used large quantile.

Author(s)

Tom Reynkens.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

Dekkers, A.L.M, Einmahl, J.H.J. and de Haan, L. (1989). "A Moment Estimator for the Index of an Extreme-value Distribution." *Annals of Statistics*, 17, 1833–1855.

See Also

`QuantMOM`, `Moment`, `ProbGH`, `Prob`

Examples

```
data(soa)

# Look at last 500 observations of SOA data
SOAdata <- sort(soa$size)[length(soa$size)-(0:499)]

# MOM estimator
M <- Moment(SOAdata)

# Exceedance probability
q <- 10^7
ProbMOM(SOAdata, gamma=M$gamma, q=q, plot=TRUE)
```

```
# Return period
q <- 10^7
ReturnMOM(SOAdata, gamma=M$gamma, q=q, plot=TRUE)
```

ProbReg *Estimator of small tail probability in regression*

Description

Estimator of small tail probability $1 - F_i(q)$ in the regression case where γ is constant and the regression modelling is thus only solely placed on the scale parameter.

Usage

```
ProbReg(Z, A, q, plot = FALSE, add = FALSE,
        main = "Estimates of small exceedance probability", ...)
```

Arguments

Z	Vector of n observations (from the response variable).
A	Vector of $n-1$ estimates for $A(i/n)$ obtained from ScaleReg.
q	The used large quantile (we estimate $P(X_i > q)$) for q large).
plot	Logical indicating if the estimates should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "Estimates of small exceedance probability".
...	Additional arguments for the plot function, see plot for more details.

Details

The estimator is defined as

$$1 - \hat{F}_i(q) = \hat{A}(i/n)(k+1)/(n+1)(q/Z_{n-k,n})^{-1/H_{k,n}},$$

with $H_{k,n}$ the Hill estimator. Here, it is assumed that we have equidistant covariates $x_i = i/n$.

See Section 4.4.1 in Albrecher et al. (2017) for more details.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
P	Vector of the corresponding probability estimates.
q	The used large quantile.

Author(s)

Tom Reynkens.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

See Also

QuantReg, ScaleReg, Prob

Examples

```
data(norwegianfire)

Z <- norwegianfire$size[norwegianfire$year==76]

i <- 100
n <- length(Z)

# Scale estimator in i/n
A <- ScaleReg(i/n, Z, h=0.5, kernel = "epanechnikov")$A

# Small exceedance probability
q <- 10^6
ProbReg(Z, A, q, plot=TRUE)

# Large quantile
p <- 10^(-5)
QuantReg(Z, A, p, plot=TRUE)
```

Quant *Weissman estimator of extreme quantiles*

Description

Compute estimates of an extreme quantile $Q(1-p)$ using the approach of Weissman (1978).

Usage

```
Quant(data, gamma, p, plot = FALSE, add = FALSE,
      main = "Estimates of extreme quantile", ...)

Weissman.q(data, gamma, p, plot = FALSE, add = FALSE,
           main = "Estimates of extreme quantile", ...)
```

Arguments

data	Vector of n observations.
gamma	Vector of $n-1$ estimates for the EVI, typically Hill estimates are used.
p	The exceedance probability of the quantile (we estimate $Q(1-p)$ for p small).
plot	Logical indicating if the estimates should be plotted as a function of k, default is `FALSE`.
add	Logical indicating if the estimates as a function of k should be added to an existing plot, default is `FALSE`.
main	Title for the plot, default is `"Estimates of extreme quantile"`.
...	Additional arguments for the `plot` function, see `plot` for more details.

Details

See Section 4.2.1 of Albrecher et al. (2017) for more details.

`Weissman.q` is the same function but with a different name for compatibility with the old `S-Plus` code.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
Q	Vector of the corresponding quantile estimates.
p	The used exceedance probability.

Author(s)

Tom Reynkens based on `S-Plus` code from Yuri Goegebeur.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

Weissman, I. (1978). "Estimation of Parameters and Large Quantiles Based on the *k* Largest Observations." *Journal of the American Statistical Association*, 73, 812–815.

See Also

`Prob`, `Quant.2oQV`

Examples

```
data(soa)

# Look at last 500 observations of SOA data
SOAdata <- sort(soa$size)[length(soa$size)-(0:499)]

# Hill estimator
H <- Hill(SOAdata)
# Bias-reduced estimator (QV)
H_QV <- Hill.2oQV(SOAdata)

# Exceedance probability
p <- 10^(-5)
# Weissman estimator
Quant(SOAdata, gamma=H$gamma, p=p, plot=TRUE)

# Second order Weissman estimator (QV)
Quant.2oQV(SOAdata, gamma=H_QV$gamma, beta=H_QV$beta, b=H_QV$b, p=p,
           add=TRUE, lty=2)
```

`Quant.2oQV` *Second order refined Weissman estimator of extreme quantiles (QV)*

Description

Compute second order refined Weissman estimator of extreme quantiles $Q(1-p)$ using the quantile view.

Usage

```
Quant.2oQV(data, gamma, b, beta, p, plot = FALSE, add = FALSE,
           main = "Estimates of extreme quantile", ...)

Weissman.q.2oQV(data, gamma, b, beta, p, plot = FALSE, add = FALSE,
                main = "Estimates of extreme quantile", ...)
```

Arguments

`data`	Vector of n observations.
`gamma`	Vector of $n-1$ estimates for the EVI obtained from `Hill.2oQV`.
`b`	Vector of $n-1$ estimates for b obtained from `Hill.2oQV`.
`beta`	Vector of $n-1$ estimates for β obtained from `Hill.2oQV`.
`p`	The exceedance probability of the quantile (we estimate $Q(1-p)$ for p small).
`plot`	Logical indicating if the estimates should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates should be added to an existing plot, default is `FALSE`.

main	Title for the plot, default is "Estimates of extreme quantile".
...	Additional arguments for the plot function, see plot for more details.

Details

See Section 4.2.1 of Albrecher et al. (2017) for more details.

Weissman.q.2oQV is the same function but with a different name for compatibility with the old S-Plus code.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
Q	Vector of the corresponding quantile estimates.
p	The used exceedance probability.

Author(s)

Tom Reynkens based on S-Plus code from Yuri Goegebeur.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

See Also

Quant, Hill.2oQV

Examples

```
data(soa)

# Look at last 500 observations of SOA data
SOAdata <- sort(soa$size)[length(soa$size)-(0:499)]

# Hill estimator
H <- Hill(SOAdata)
# Bias-reduced estimator (QV)
H_QV <- Hill.2oQV(SOAdata)

# Exceedance probability
p <- 10^(-5)
# Weissman estimator
Quant(SOAdata, gamma=H$gamma, p=p, plot=TRUE)

# Second order Weissman estimator (QV)
```

```
Quant.2oQV(SOAdata, gamma=H_QV$gamma, beta=H_QV$beta, b=H_QV$b, p=p,
          add=TRUE, lty=2)
```

QuantGH *Estimator of extreme quantiles using generalised Hill*

Description

Compute estimates of an extreme quantile $Q(1-p)$ using generalised Hill estimates of the EVI.

Usage

```
QuantGH(data, gamma, p, plot = FALSE, add = FALSE,
        main = "Estimates of extreme quantile", ...)
```

Arguments

data	Vector of n observations.
gamma	Vector of $n-2$ estimates for the EVI obtained from genHill.
p	The exceedance probability of the quantile (we estimate $Q(1-p)$ for p small).
plot	Logical indicating if the estimates should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "Estimates of extreme quantile".
...	Additional arguments for the plot function, see plot for more details.

Details

See Section 4.2.2 of Albrecher et al. (2017) for more details.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
Q	Vector of the corresponding quantile estimates.
p	The used exceedance probability.

Author(s)

Tom Reynkens.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

Beirlant, J., Vynckier, P. and Teugels, J.L. (1996). "Excess Function and Estimation of the Extreme-value Index". *Bernoulli*, 2, 293–318.

See Also

ProbGH, genHill, QuantMOM, Quant

Examples

```
data(soa)

# Look at last 500 observations of SOA data
SOAdata <- sort(soa$size)[length(soa$size)-(0:499)]

# Hill estimator
H <- Hill(SOAdata)
# Generalised Hill estimator
gH <- genHill(SOAdata, H$gamma)

# Large quantile
p <- 10^(-5)
QuantGH(SOAdata, p=p, gamma=gH$gamma, plot=TRUE)
```

QuantGPD *Estimator of extreme quantiles using GPD-MLE*

Description

Computes estimates of an extreme quantile $Q(1-p)$ using the GPD fit for the peaks over a threshold.

Usage

```
QuantGPD(data, gamma, sigma, p, plot = FALSE, add = FALSE,
         main = "Estimates of extreme quantile", ...)
```

Arguments

data	Vector of n observations.
gamma	Vector of $n-1$ estimates for the EVI obtained from GPDmle.
sigma	Vector of $n-1$ estimates for σ obtained from GPDmle.
p	The exceedance probability of the quantile (we estimate $Q(1-p)$ for p small).

plot	Logical indicating if the estimates should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "Estimates of extreme quantile".
...	Additional arguments for the plot function, see plot for more details.

Details

See Section 4.2.2 in Albrecher et al. (2017) for more details.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
Q	Vector of the corresponding quantile estimates.
p	The used exceedance probability.

Author(s)

Tom Reynkens.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

See Also

ProbGPD, GPDmle, Quant

Examples

```
data(soa)

# Look at last 500 observations of SOA data
SOAdata <- sort(soa$size)[length(soa$size)-(0:499)]

# GPD-ML estimator
pot <- GPDmle(SOAdata)

# Large quantile
p <- 10^(-5)
QuantGPD(SOAdata, p=p, gamma=pot$gamma, sigma=pot$sigma, plot=TRUE)
```

QuantMOM *Estimator of extreme quantiles using MOM*

Description

Compute estimates of an extreme quantile $Q(1-p)$ using the Method of Moments estimates of the EVI.

Usage

```
QuantMOM(data, gamma, p, plot = FALSE, add = FALSE,
         main = "Estimates of extreme quantile", ...)
```

Arguments

data	Vector of n observations.
gamma	Vector of $n-1$ estimates for the EVI obtained from Moment.
p	The exceedance probability of the quantile (we estimate $Q(1-p)$ for p small).
plot	Logical indicating if the estimates should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "Estimates of extreme quantile".
...	Additional arguments for the plot function, see plot for more details.

Details

See Section 4.2.2 of Albrecher et al. (2017) for more details.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
Q	Vector of the corresponding quantile estimates.
p	The used exceedance probability.

Author(s)

Tom Reynkens.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

Dekkers, A.L.M, Einmahl, J.H.J. and de Haan, L. (1989). "A Moment Estimator for the Index of an Extreme-value Distribution." *Annals of Statistics*, 17, 1833–1855.

See Also

ProbMOM, Moment, QuantGH, Quant

Examples

```
data(soa)

# Look at last 500 observations of SOA data
SOAdata <- sort(soa$size)[length(soa$size)-(0:499)]

# MOM estimator
M <- Moment(SOAdata)

# Large quantile
p <- 10^(-5)
QuantMOM(SOAdata, p=p, gamma=M$gamma, plot=TRUE)
```

QuantReg *Estimator of extreme quantiles in regression*

Description

Estimator of extreme quantile $Q_i(1-p)$ in the regression case where γ is constant and the regression modelling is thus only solely placed on the scale parameter.

Usage

```
QuantReg(Z, A, p, plot = FALSE, add = FALSE,
          main = "Estimates of extreme quantile", ...)
```

Arguments

Z	Vector of n observations (from the response variable).
A	Vector of $n-1$ estimates for $A(i/n)$ obtained from ScaleReg.
p	The exceedance probability of the quantile (we estimate $Q_i(1-p)$ for p small).
plot	Logical indicating if the estimates should be plotted as a function of k, default is FALSE.

`add`	Logical indicating if the estimates should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Estimates of extreme quantile"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The estimator is defined as

$$\hat{Q}_i(1-p) = Z_{n-k,n}((k+1)/((n+1)\times p)\hat{A}(i/n))^{H_{k,n}},$$

with $H_{k,n}$ the Hill estimator. Here, it is assumed that we have equidistant covariates $x_i = i/n$.

See Section 4.4.1 in Albrecher et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`Q`	Vector of the corresponding quantile estimates.
`p`	The used exceedance probability.

Author(s)

Tom Reynkens.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

See Also

`ProbReg`, `ScaleReg`, `Quant`

Examples

```
data(norwegianfire)

Z <- norwegianfire$size[norwegianfire$year==76]

i <- 100
n <- length(Z)

# Scale estimator in i/n
A <- ScaleReg(i/n, Z, h=0.5, kernel = "epanechnikov")$A

# Small exceedance probability
q <- 10^6
ProbReg(Z, A, q, plot=TRUE)
```

```
# Large quantile
p <- 10^(-5)
QuantReg(Z, A, p, plot=TRUE)
```

`Scale` *Scale estimator*

Description

Computes the estimator for the scale parameter as described in Beirlant et al. (2016).

Usage

```
Scale(data, gamma = NULL, logk = FALSE, plot = FALSE, add = FALSE,
      main = "Estimates of scale parameter", ...)
```

Arguments

`data`	Vector of n observations.
`gamma`	Vector of $n-1$ estimates for the EVI. When `NULL` (the default value), Hill estimates are computed.
`logk`	Logical indicating if the estimates are plotted as a function of $\log(k)$ (`logk=TRUE`) or as a function of k. Default is `FALSE`.
`plot`	Logical indicating if the estimates should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Estimates of scale parameter"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The scale estimates are computed based on the following model for the CDF: $1-F(x) = Ax^{-1/\gamma}$, where $A := C^{1/\gamma}$ is the scale parameter:

$$\hat{A}_{k,n} = (k+1)/(n+1)X_{n-k,n}^{1/H_{k,n}}$$

where $H_{k,n}$ are the Hill estimates.

See Section 4.2.1 of Albrecher et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`A`	Vector of the corresponding scale estimates.
`C`	Vector of the corresponding estimates for C, see Details.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant, J., Schoutens, W., De Spiegeleer, J., Reynkens, T. and Herrmann, K. (2016). "Hunting for Black Swans in the European Banking Sector Using Extreme Value Analysis." In: Jan Kallsen and Antonis Papapantoleon (eds.), *Advanced Modelling in Mathematical Finance*, Springer International Publishing, Switzerland, pp. 147–166.

See Also

`ScaleEPD`, `Scale.2o`, `Hill`

Examples

```
data(secura)

# Hill estimator
H <- Hill(secura$size)

# Scale estimator
S <- Scale(secura$size, gamma=H$gamma, plot=FALSE)

# Plot logarithm of scale
plot(S$k,log(S$A), xlab="k", ylab="log(Scale)", type="l")
```

`Scale.2o` *Bias-reduced scale estimator using second order Hill estimator*

Description

Computes the bias-reduced estimator for the scale parameter using the second-order Hill estimator.

Usage

```
Scale.2o(data, gamma, b, beta, logk = FALSE, plot = FALSE, add = FALSE,
         main = "Estimates of scale parameter", ...)
```

Arguments

`data`	Vector of n observations.
`gamma`	Vector of $n-1$ estimates for the EVI obtained from `Hill.2oQV`.
`b`	Vector of $n-1$ estimates for B obtained from `Hill.2oQV`.
`beta`	Vector of $n-1$ estimates for β obtained from `Hill.2oQV`.

logk	Logical indicating if the estimates are plotted as a function of $\log(k)$ (logk=TRUE) or as a function of k. Default is FALSE.
plot	Logical indicating if the estimates should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "Estimates of scale parameter".
...	Additional arguments for the plot function, see plot for more details.

Details

The scale estimates are computed based on the following model for the CDF: $1-F(x) = Ax^{-1/\gamma}(1+bx^{-\beta}(1+o(1)))$, where $A := C^{1/\gamma}$ is the scale parameter.

See Section 4.2.1 of Albrecher et al. (2017) for more details.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
A	Vector of the corresponding scale estimates.
C	Vector of the corresponding estimates for C, see details.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant, J., Schoutens, W., De Spiegeleer, J., Reynkens, T. and Herrmann, K. (2016). "Hunting for Black Swans in the European Banking Sector Using Extreme Value Analysis." In: Jan Kallsen and Antonis Papapantoleon (eds.), *Advanced Modelling in Mathematical Finance*, Springer International Publishing, Switzerland, pp. 147–166.

See Also

Scale, ScaleEPD, Hill.2oQV

Examples

```
data(secura)

# Hill estimator
H <- Hill(secura$size)
# Bias-reduced Hill estimator
H2o <- Hill.2oQV(secura$size)
```

```
# Scale estimator
S <- Scale(secura$size, gamma=H$gamma, plot=FALSE)
# Bias-reduced scale estimator
S2o <- Scale.2o(secura$size, gamma=H2o$gamma, b=H2o$b,
          beta=H2o$beta, plot=FALSE)

# Plot logarithm of scale
plot(S$k,log(S$A), xlab="k", ylab="log(Scale)", type="l")
lines(S2o$k,log(S2o$A), lty=2)
```

ScaleEPD *Bias-reduced scale estimator using EPD estimator*

Description

Computes the bias-reduced estimator for the scale parameter using the EPD estimator (Beirlant et al., 2016).

Usage

```
ScaleEPD(data, gamma, kappa, logk = FALSE, plot = FALSE, add = FALSE,
         main = "Estimates of scale parameter", ...)
```

Arguments

data	Vector of n observations.
gamma	Vector of $n-1$ estimates for the EVI obtained from EPD.
kappa	Vector of $n-1$ estimates for κ obtained from EPD.
logk	Logical indicating if the estimates are plotted as a function of $\log(k)$ (logk=TRUE) or as a function of k. Default is FALSE.
plot	Logical indicating if the estimates should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "Estimates of scale parameter".
...	Additional arguments for the plot function, see plot for more details.

Details

The scale estimates are computed based on the following model for the CDF: $1-F(x) = Ax^{-1/\gamma}(1+bx^{-\beta}(1+o(1)))$, where $A := C^{1/\gamma}$ is the scale parameter. Using the EPD approach we replace $bx^{-\beta}$ by κ/γ.

See Section 4.2.1 of Albrecher et al. (2017) for more details.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
A	Vector of the corresponding scale estimates.
C	Vector of the corresponding estimates for C, see details.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant, J., Schoutens, W., De Spiegeleer, J., Reynkens, T. and Herrmann, K. (2016). "Hunting for Black Swans in the European Banking Sector Using Extreme Value Analysis." In: Jan Kallsen and Antonis Papapantoleon (eds.), *Advanced Modelling in Mathematical Finance*, Springer International Publishing, Switzerland, pp. 147–166.

See Also

Scale, Scale.2o, EPD

Examples

```
data(secura)

# Hill estimator
H <- Hill(secura$size)
# EPD estimator
epd <- EPD(secura$size)

# Scale estimator
S <- Scale(secura$size, gamma=H$gamma, plot=FALSE)
# Bias-reduced scale estimator
Sepd <- ScaleEPD(secura$size, gamma=epd$gamma, kappa=epd$kappa, plot=FALSE)

# Plot logarithm of scale
plot(S$k,log(S$A), xlab="k", ylab="log(Scale)", type="l")
lines(Sepd$k,log(Sepd$A), lty=2)
```

`ScaleReg` *Scale estimator in regression*

Description

Estimator of the scale parameter in the regression case where γ is constant and the regression modelling is thus placed solely on the scale parameter.

Usage

```
ScaleReg(s, Z, kernel = c("normal", "uniform", "triangular", "epanechnikov", "biweight"),
         h, plot = TRUE, add = FALSE, main = "Estimates of scale parameter", ...)
```

Arguments

`s`	Point to evaluate the scale estimator in.
`Z`	Vector of n observations (from the response variable).
`kernel`	The kernel used in the estimator. One of `"normal"` (default), `"uniform"`, `"triangular"`, `"epanechnikov"` and `"biweight"`.
`h`	The bandwidth used in the kernel function.
`plot`	Logical indicating if the estimates should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Estimates of scale parameter"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The scale estimator is computed as

$$\hat{A}(s) = 1/(k+1)\sum_{i=1}^{n} 1_{Z_i > Z_{n-k,n}} K_h(s - i/n)$$

with $K_h(x) = K(x/h)/h$, K the kernel function and h the bandwidth. Here, it is assumed that we have equidistant covariates $x_i = i/n$.

See Section 4.4.1 in Albrecher et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`A`	Vector of the corresponding scale estimates.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

See Also

ProbReg, QuantReg, scale, Hill

Examples

```
data(norwegianfire)

Z <- norwegianfire$size[norwegianfire$year==76]

i <- 100
n <- length(Z)

# Scale estimator in i/n
A <- ScaleReg(i/n, Z, h=0.5, kernel = "epanechnikov")$A

# Small exceedance probability
q <- 10^6
ProbReg(Z, A, q, plot=TRUE)

# Large quantile
p <- 10^(-5)
QuantReg(Z, A, p, plot=TRUE)
```

---

secura *Secura dataset*

Description

Automobile claims from 1988 to 2001, gathered from several European insurance companies, exceeding 1 200 000 Euro. Note that the data were, among others, corrected for inflation.

Usage

```
data("secura")
```

Format

A data frame with 371 observations on the following 2 variables:

`year` Year of claim occurence.

`size` Size of automobile insurance claim (in EUR).

References

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

Examples

```
data(secura)

# Exponential QQ-plot of Secura data
ExpQQ(secura$size)

# Pareto QQ-plot of Secura data
ParetoQQ(secura$size)

# Mean excess plot of Secura data (function of k)
MeanExcess(secura$size, k=TRUE)

# Mean excess plot of Secura data (function of order statistics)
MeanExcess(secura$size, k=FALSE)
```

`soa` *SOA Group Medical Insurance Large Claims Database*

Description

The SOA Group Medical Insurance Large Claims Database records, among others, all the claim amounts exceeding 25,000 USD in the year 1991.

Usage

```
data("soa")
```

Format

A data frame with 75789 observations on the following variable:

`size` Claim size (in USD).

Source

Grazier, K. L. and G'Sell Associates (1997). *Group Medical Insurance Large Claims Database Collection and Analysis*. SOA Monograph M-HB97-1, Society of Actuaries, Schaumburg.

Society of Actuaries, https://www.soa.org/resources/experience-studies/2000-2004/91-92-group-medical-cla

References

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

Examples

```
data(soa)

# Histogram of log-claim amount
hist(log(soa$size),breaks=seq(10,16,0.2),xlab="log(Claim size)")

# Exponential QQ-plot of claim amount
ExpQQ(soa$size)

# Mean excess plot of claim amount (function of k)
MeanExcess(soa$size, k=TRUE)

# Mean excess plot of claim amount (function of order statistics)
MeanExcess(soa$size, k=FALSE)
```

`Splice` *Spliced distribution*

Description

Density, distribution function, quantile function and random generation for the fitted spliced distribution.

Usage

```
dSplice(x, splicefit, log = FALSE)

pSplice(x, splicefit, lower.tail = TRUE, log.p = FALSE)

qSplice(p, splicefit, lower.tail = TRUE, log.p = FALSE)

rSplice(n, splicefit)
```

Arguments

x	Vector of points to evaluate the CDF or PDF in.
p	Vector of probabilities.
n	Number of observations.
splicefit	A `SpliceFit` object, e.g. output from `SpliceFitPareto`, `SpliceFiticPareto` or `SpliceFitGPD`.
log	Logical indicating if the densities are given as $\log(f)$, default is `FALSE`.
lower.tail	Logical indicating if the probabilities are of the form $P(X \leq x)$ (`TRUE`) or $P(X > x)$ (`FALSE`). Default is `TRUE`.
log.p	Logical indicating if the probabilities are given as $\log(p)$, default is `FALSE`.

Details

See Reynkens et al. (2017) and Section 4.3 in Albrecher et al. (2017) for details.

Value

`dSplice` gives the density function evaluated in x, `pSplice` the CDF evaluated in x and `qSplice` the quantile function evaluated in p. The length of the result is equal to the length of x or p.

`rSplice` returns a random sample of length n.

Author(s)

Tom Reynkens with `R` code from Roel Verbelen for the mixed Erlang PDF, CDF and quantiles.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Reynkens, T., Verbelen, R., Beirlant, J. and Antonio, K. (2017). "Modelling Censored Losses Using Splicing: a Global Fit Strategy With Mixed Erlang and Extreme Value Distributions". *Insurance: Mathematics and Economics*, 77, 65–77.

Verbelen, R., Gong, L., Antonio, K., Badescu, A. and Lin, S. (2015). "Fitting Mixtures of Erlangs to Censored and Truncated Data Using the EM Algorithm." *Astin Bulletin*, 45, 729–758.

See Also

`VaR`, `SpliceFit`, `SpliceFitPareto`, `SpliceFiticPareto`, `SpliceFitGPD`, `SpliceECDF`, `SpliceLL`, `SplicePP`

Examples

```
## Not run:

# Pareto random sample
X <- rpareto(1000, shape = 2)

# Splice ME and Pareto
splicefit <- SpliceFitPareto(X, 0.6)



x <- seq(0, 20, 0.01)

# Plot of spliced CDF
plot(x, pSplice(x, splicefit), type="l", xlab="x", ylab="F(x)")

# Plot of spliced PDF
plot(x, dSplice(x, splicefit), type="l", xlab="x", ylab="f(x)")
```

```
p <- seq(0, 1, 0.01)

# Plot of splicing quantiles
plot(p, qSplice(p, splicefit), type="l", xlab="p", ylab="Q(p)")

# Plot of VaR
plot(p, VaR(p, splicefit), type="l", xlab="p", ylab=bquote(VaR[p]))



# Random sample from spliced distribution
x <- rSplice(1000, splicefit)


## End(Not run)
```

SpliceECDF *Plot of fitted and empirical survival function*

Description

This function plots the fitted survival function of the spliced distribution together with the empirical survival function (determined using the Empirical CDF (ECDF)). Moreover, $100(1-\alpha)\%$ confidence bands are added.

Usage

```
SpliceECDF(x, X, splicefit, alpha = 0.05, ...)
```

Arguments

`x`	Vector of points to plot the functions at.
`X`	Data used for fitting the distribution.
`splicefit`	A `SpliceFit` object, e.g. output from `SpliceFitPareto` or `SpliceFitGPD`.
`alpha`	$100(1-\alpha)\%$ is the confidence level for the confidence bands. Default is $\alpha = 0.05$.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

Use `SpliceTB` for censored data.

Confidence bands are determined using the Dvoretzky-Kiefer-Wolfowitz inequality (Massart, 1990).

See Reynkens et al. (2017) and Section 4.3.1 in Albrecher et al. (2017) for more details.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Massart, P. (1990). The Tight Constant in the Dvoretzky-Kiefer-Wolfowitz Inequality. *Annals of Probability*, 18, 1269–1283.

Reynkens, T., Verbelen, R., Beirlant, J. and Antonio, K. (2017). "Modelling Censored Losses Using Splicing: a Global Fit Strategy With Mixed Erlang and Extreme Value Distributions". *Insurance: Mathematics and Economics*, 77, 65–77.

Verbelen, R., Gong, L., Antonio, K., Badescu, A. and Lin, S. (2015). "Fitting Mixtures of Erlangs to Censored and Truncated Data Using the EM Algorithm." *Astin Bulletin*, 45, 729–758.

See Also

SpliceTB, pSplice, ecdf, SpliceFitPareto, SpliceFitGPD, SpliceLL, SplicePP, SpliceQQ

Examples

```
## Not run:

# Pareto random sample
X <- rpareto(1000, shape = 2)

# Splice ME and Pareto
splicefit <- SpliceFitPareto(X, 0.6)



x <- seq(0, 20, 0.01)

# Plot of spliced CDF
plot(x, pSplice(x, splicefit), type="l", xlab="x", ylab="F(x)")

# Plot of spliced PDF
plot(x, dSplice(x, splicefit), type="l", xlab="x", ylab="f(x)")



# Fitted survival function and empirical survival function
SpliceECDF(x, X, splicefit)

# Log-log plot with empirical survival function and fitted survival function
SpliceLL(x, X, splicefit)

# PP-plot of empirical survival function and fitted survival function
SplicePP(X, splicefit)

# PP-plot of empirical survival function and
# fitted survival function with log-scales
SplicePP(X, splicefit, log=TRUE)
```

```
# Splicing QQ-plot
SpliceQQ(X, splicefit)

## End(Not run)
```

SpliceFit *Splicing fit*

Description

Create an S3 object using ME-Pa or ME-GPD splicing fit obtained from SpliceFitPareto, SpliceFiticPareto or SpliceFitGPD.

Usage

```
SpliceFit(const, trunclower, t, type, MEfit, EVTfit, loglik = NULL, IC = NULL)
```

Arguments

const	Vector of splicing constants or a single splicing constant.
trunclower	Lower truncation point.
t	Vector of splicing points or a single splicing point.
type	Vector of types of the distributions: "ME" and then for each fitted EVT distribution: "Pa" (Pareto), "TPa" (truncated Pareto) or "GPD" (GPD).
MEfit	MEfit object with details on the mixed Erlang fit.
EVTfit	EVTfit object with details on the EVT fit.
loglik	Log-likelihood of the fitted model. When NULL (default), not included in the object.
IC	Information criteria of the fitted model. When NULL (default), not included in the object. This vector should have length 1, 2 or 3 when included.

Details

See Reynkens et al. (2017) and Section 4.3 in Albrecher et al. (2017) for details.

Value

An S3 object containing the above input arguments and values for π, the splicing weights. These splicing weights are equal to

$$\pi_1 = const_1, \pi_2 = const_2 - const_1, ..., \pi_{l+1} = 1 - const_l = 1 - (\pi_1 + ... + \pi_l)$$

when $l \geq 2$ and

$$\pi_1 = const_1, \pi_2 = 1 - const_1 = 1 - \pi_1$$

when $l = 1$, where l is the length of const.

A summary method is available.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Reynkens, T., Verbelen, R., Beirlant, J. and Antonio, K. (2017). "Modelling Censored Losses Using Splicing: a Global Fit Strategy With Mixed Erlang and Extreme Value Distributions". *Insurance: Mathematics and Economics*, 77, 65–77.

Verbelen, R., Gong, L., Antonio, K., Badescu, A. and Lin, S. (2015). "Fitting Mixtures of Erlangs to Censored and Truncated Data Using the EM Algorithm." *Astin Bulletin*, 45, 729–758

See Also

MEfit, EVTfit, SpliceFitPareto, SpliceFiticPareto, SpliceFitGPD

Examples

```
# Create MEfit object
mefit <- MEfit(p=c(0.65,0.35), shape=c(39,58), theta=16.19, M=2)

# Create EVTfit object
evtfit <- EVTfit(gamma=c(0.76,0.64), endpoint=c(39096, Inf))

# Create SpliceFit object
splicefit <- SpliceFit(const=c(0.5,0.996), trunclower=0, t=c(1020,39096), type=c("ME","TPa","Pa"),
                       MEfit=mefit, EVTfit=evtfit)

# Show summary
summary(splicefit)
```

SpliceFitGPD — *Splicing of mixed Erlang and GPD using POT-MLE*

Description

Fit spliced distribution of a mixed Erlang distribution and a Generalised Pareto Distribution (GPD). The parameters of the GPD are determined using the POT-MLE approach.

Usage

```
SpliceFitGPD(X, const = NULL, tsplice = NULL, M = 3, s = 1:10, trunclower = 0,
             ncores = NULL, criterium = c("BIC","AIC"), reduceM = TRUE,
             eps = 10^(-3), beta_tol = 10^(-5), maxiter = Inf)
```

Arguments

- `X` Data used for fitting the distribution.
- `const` The probability of the quantile where the ME distribution will be spliced with the GPD distribution. Default is `NULL` meaning the input from `tsplice` is used.
- `tsplice` The point where the ME distribution will be spliced with the GPD distribution. Default is `NULL` meaning the input from `const` is used.
- `M` Initial number of Erlang mixtures, default is 3. This number can change when determining an optimal mixed Erlang fit using an information criterion.
- `s` Vector of spread factors for the EM algorithm, default is `1:10`. We loop over these factors when determining an optimal mixed Erlang fit using an information criterion, see Verbelen et al. (2016).
- `trunclower` Lower truncation point. Default is 0.
- `ncores` Number of cores to use when determining an optimal mixed Erlang fit using an information criterion. When `NULL` (default), `max(nc-1,1)` cores are used where `nc` is the number of cores as determined by `detectCores`.
- `criterium` Information criterion used to select the number of components of the ME fit and `s`. One of `"AIC"` and `"BIC"` (default).
- `reduceM` Logical indicating if M should be reduced based on the information criterion, default is `TRUE`.
- `eps` Covergence threshold used in the EM algorithm (ME part). Default is `10^(-3)`.
- `beta_tol` Threshold for the mixing weights below which the corresponding shape parameter vector is considered neglectable (ME part). Default is `10^(-5)`.
- `maxiter` Maximum number of iterations in a single EM algorithm execution (ME part). Default is `Inf` meaning no maximum number of iterations.

Details

See Reynkens et al. (2017), Section 4.3.1 of Albrecher et al. (2017) and Verbelen et al. (2015) for details. The code follows the notation of the latter. Initial values follow from Verbelen et al. (2016).

Value

A `SpliceFit` object.

Author(s)

Tom Reynkens with `R` code from Roel Verbelen for fitting the mixed Erlang distribution.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Reynkens, T., Verbelen, R., Beirlant, J. and Antonio, K. (2017). "Modelling Censored Losses Using Splicing: a Global Fit Strategy With Mixed Erlang and Extreme Value Distributions". *Insurance: Mathematics and Economics*, 77, 65–77.

Verbelen, R., Gong, L., Antonio, K., Badescu, A. and Lin, S. (2015). "Fitting Mixtures of Erlangs to Censored and Truncated Data Using the EM Algorithm." *Astin Bulletin*, 45, 729–758.

Verbelen, R., Antonio, K. and Claeskens, G. (2016). "Multivariate Mixtures of Erlangs for Density Estimation Under Censoring." *Lifetime Data Analysis*, 22, 429–455.

See Also

SpliceFitPareto, SpliceFiticPareto, Splice, GPDfit

Examples

```
## Not run:

# GPD random sample
X <- rgpd(1000, gamma = 0.5, sigma = 2)

# Splice ME and GPD
splicefit <- SpliceFitGPD(X, 0.6)



x <- seq(0, 20, 0.01)

# Plot of spliced CDF
plot(x, pSplice(x, splicefit), type="l", xlab="x", ylab="F(x)")

# Plot of spliced PDF
plot(x, dSplice(x, splicefit), type="l", xlab="x", ylab="f(x)")



# Fitted survival function and empirical survival function
SpliceECDF(x, X, splicefit)

# Log-log plot with empirical survival function and fitted survival function
SpliceLL(x, X, splicefit)

# PP-plot of empirical survival function and fitted survival function
SplicePP(X, splicefit)

# PP-plot of empirical survival function and
# fitted survival function with log-scales
SplicePP(X, splicefit, log=TRUE)

# Splicing QQ-plot
SpliceQQ(X, splicefit)

## End(Not run)
```

`SpliceFiticPareto` *Splicing of mixed Erlang and Pareto for interval censored data*

Description

Fit spliced distribution of a mixed Erlang distribution and a Pareto distribution adapted for interval censoring and truncation.

Usage

```
SpliceFiticPareto(L, U, censored, tsplice, M = 3, s = 1:10, trunclower = 0,
                  truncupper = Inf, ncores = NULL, criterium = c("BIC", "AIC"),
                 reduceM = TRUE, eps = 10^(-3), beta_tol = 10^(-5), maxiter = Inf,
                  cpp = FALSE)
```

Arguments

`L`	Vector of length n with the lower boundaries of the intervals for interval censored data or the observed data for right censored data.
`U`	Vector of length n with the upper boundaries of the intervals.
`censored`	A logical vector of length n indicating if an observation is censored.
`tsplice`	The splicing point t.
`M`	Initial number of Erlang mixtures, default is 3. This number can change when determining an optimal mixed Erlang fit using an information criterion.
`s`	Vector of spread factors for the EM algorithm, default is `1:10`. We loop over these factors when determining an optimal mixed Erlang fit using an information criterion, see Verbelen et al. (2016).
`trunclower`	Lower truncation point. Default is 0.
`truncupper`	Upper truncation point. Default is `Inf` (no upper truncation).
`ncores`	Number of cores to use when determining an optimal mixed Erlang fit using an information criterion. When `NULL` (default), `max(nc-1,1)` cores are used where `nc` is the number of cores as determined by `detectCores`.
`criterium`	Information criterion used to select the number of components of the ME fit and `s`. One of `"AIC"` and `"BIC"` (default).
`reduceM`	Logical indicating if M should be reduced based on the information criterion, default is `TRUE`.
`eps`	Covergence threshold used in the EM algorithm. Default is `10^(-3)`.
`beta_tol`	Threshold for the mixing weights below which the corresponding shape parameter vector is considered neglectable (ME part). Default is `10^(-5)`.
`maxiter`	Maximum number of iterations in a single EM algorithm execution. Default is `Inf` meaning no maximum number of iterations.
`cpp`	Use `C++` implementation (`cpp=TRUE`) or `R` implementation (`cpp=FALSE`) of the algorithm. Default is `FALSE` meaning the plain `R` implementation is used.

Details

See Reynkens et al. (2017), Section 4.3.2 of Albrecher et al. (2017) and Verbelen et al. (2015) for details. The code follows the notation of the latter. Initial values follow from Verbelen et al. (2016).

Right censored data should be entered as `L=l` and `U=truncupper`, and left censored data should be entered as `L=trunclower` and `U=u`.

Value

A `SpliceFit` object.

Author(s)

Tom Reynkens based on `R` code from Roel Verbelen for fitting the mixed Erlang distribution (without splicing).

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Reynkens, T., Verbelen, R., Beirlant, J. and Antonio, K. (2017). "Modelling Censored Losses Using Splicing: a Global Fit Strategy With Mixed Erlang and Extreme Value Distributions". *Insurance: Mathematics and Economics*, 77, 65–77.

Verbelen, R., Gong, L., Antonio, K., Badescu, A. and Lin, S. (2015). "Fitting Mixtures of Erlangs to Censored and Truncated Data Using the EM Algorithm." *Astin Bulletin*, 45, 729–758.

Verbelen, R., Antonio, K. and Claeskens, G. (2016). "Multivariate Mixtures of Erlangs for Density Estimation Under Censoring." *Lifetime Data Analysis*, 22, 429–455.

See Also

`SpliceFitPareto`, `SpliceFitGPD`, `Splice`

Examples

```
## Not run:

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X,Y)

# Censoring indicator
censored <- (X>Y)

# Right boundary
U <- Z
```

```
U[censored] <- Inf

# Splice ME and Pareto
splicefit <- SpliceFiticPareto(L=Z, U=U, censored=censored, tsplice=quantile(Z,0.9))



x <- seq(0,20,0.1)

# Plot of spliced CDF
plot(x, pSplice(x, splicefit), type="l", xlab="x", ylab="F(x)")

# Plot of spliced PDF
plot(x, dSplice(x, splicefit), type="l", xlab="x", ylab="f(x)")


# Fitted survival function and Turnbull survival function
SpliceTB(x, L=Z, U=U, censored=censored, splicefit=splicefit)


# Log-log plot with Turnbull survival function and fitted survival function
SpliceLL_TB(x, L=Z, U=U, censored=censored, splicefit=splicefit)


# PP-plot of Turnbull survival function and fitted survival function
SplicePP_TB(L=Z, U=U, censored=censored, splicefit=splicefit)

# PP-plot of Turnbull survival function and
# fitted survival function with log-scales
SplicePP_TB(L=Z, U=U, censored=censored, splicefit=splicefit, log=TRUE)

# QQ-plot using Turnbull survival function and fitted survival function
SpliceQQ_TB(L=Z, U=U, censored=censored, splicefit=splicefit)

## End(Not run)
```

SpliceFitPareto *Splicing of mixed Erlang and Pareto*

Description

Fit spliced distribution of a mixed Erlang distribution and Pareto distribution(s). The shape parameter(s) of the Pareto distribution(s) is determined using the Hill estimator.

Usage

```
SpliceFitPareto(X, const = NULL, tsplice = NULL, M = 3, s = 1:10, trunclower = 0,
                truncupper = Inf, EVTtruncation = FALSE, ncores = NULL,
                criterium = c("BIC","AIC"), reduceM = TRUE,
                eps = 10^(-3), beta_tol = 10^(-5), maxiter = Inf)
```

```
SpliceFitHill(X, const = NULL, tsplice = NULL, M = 3, s = 1:10, trunclower = 0,
              truncupper = Inf, EVTtruncation = FALSE, ncores = NULL,
              criterium = c("BIC","AIC"), reduceM = TRUE,
              eps = 10^(-3), beta_tol = 10^(-5), maxiter = Inf)
```

Arguments

X	Data used for fitting the distribution.
const	Vector of length l containing the probabilities of the quantiles where the distributions will be spliced (splicing points). The ME distribution will be spliced with l Pareto distributions. Default is NULL meaning the input from tsplice is used.
tsplice	Vector of length l containing the splicing points. The ME distribution will be spliced with l Pareto distributions. Default is NULL meaning the input from const is used.
M	Initial number of Erlang mixtures, default is 3. This number can change when determining an optimal mixed Erlang fit using an information criterion.
s	Vector of spread factors for the EM algorithm, default is 1:10. We loop over these factors when determining an optimal mixed Erlang fit using an information criterion, see Verbelen et al. (2016).
trunclower	Lower truncation point. Default is 0.
truncupper	Upper truncation point. Default is Inf (no upper truncation). When truncupper=Inf and EVTtruncation=TRUE, the truncation point is estimated using the approach of Beirlant et al. (2016).
EVTtruncation	Logical indicating if the l-th Pareto distribution is a truncated Pareto distribution. Default is FALSE.
ncores	Number of cores to use when determining an optimal mixed Erlang fit using an information criterion. When NULL (default), max(nc-1,1) cores are used where nc is the number of cores as determined by detectCores.
criterium	Information criterion used to select the number of components of the ME fit and s. One of "AIC" and "BIC" (default).
reduceM	Logical indicating if M should be reduced based on the information criterion, default is TRUE.
eps	Covergence threshold used in the EM algorithm (ME part). Default is 10^(-3).
beta_tol	Threshold for the mixing weights below which the corresponding shape parameter vector is considered neglectable (ME part). Default is 10^(-5).
maxiter	Maximum number of iterations in a single EM algorithm execution (ME part). Default is Inf meaning no maximum number of iterations.

Details

See Reynkens et al. (2017), Section 4.3.1 of Albrecher et al. (2017) and Verbelen et al. (2015) for details. The code follows the notation of the latter. Initial values follow from Verbelen et al. (2016).

The SpliceFitHill function is the same function but with a different name for compatibility with old versions of the package.

Use SpliceFiticPareto when censoring is present.

Value

A SpliceFit object.

Author(s)

Tom Reynkens with R code from Roel Verbelen for fitting the mixed Erlang distribution.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant, J., Fraga Alves, M.I. and Gomes, M.I. (2016). "Tail fitting for Truncated and Non-truncated Pareto-type Distributions." *Extremes*, 19, 429–462.

Reynkens, T., Verbelen, R., Beirlant, J. and Antonio, K. (2017). "Modelling Censored Losses Using Splicing: a Global Fit Strategy With Mixed Erlang and Extreme Value Distributions". *Insurance: Mathematics and Economics*, 77, 65–77.

Verbelen, R., Gong, L., Antonio, K., Badescu, A. and Lin, S. (2015). "Fitting Mixtures of Erlangs to Censored and Truncated Data Using the EM Algorithm." *Astin Bulletin*, 45, 729–758.

Verbelen, R., Antonio, K. and Claeskens, G. (2016). "Multivariate Mixtures of Erlangs for Density Estimation Under Censoring." *Lifetime Data Analysis*, 22, 429–455.

See Also

SpliceFiticPareto, SpliceFitGPD, Splice, Hill, trHill

Examples

```
## Not run:

# Pareto random sample
X <- rpareto(1000, shape = 2)

# Splice ME and Pareto
splicefit <- SpliceFitPareto(X, 0.6)



x <- seq(0, 20, 0.01)

# Plot of spliced CDF
plot(x, pSplice(x, splicefit), type="l", xlab="x", ylab="F(x)")

# Plot of spliced PDF
plot(x, dSplice(x, splicefit), type="l", xlab="x", ylab="f(x)")
```

```
# Fitted survival function and empirical survival function
SpliceECDF(x, X, splicefit)

# Log-log plot with empirical survival function and fitted survival function
SpliceLL(x, X, splicefit)

# PP-plot of empirical survival function and fitted survival function
SplicePP(X, splicefit)

# PP-plot of empirical survival function and
# fitted survival function with log-scales
SplicePP(X, splicefit, log=TRUE)

# Splicing QQ-plot
SpliceQQ(X, splicefit)

## End(Not run)
```

SpliceLL *LL-plot with fitted and empirical survival function*

Description

This function plots the logarithm of the empirical survival function (determined using the Empirical CDF (ECDF)) versus the logarithm of the data. Moreover, the logarithm of the fitted survival function of the spliced distribution is added.

Usage

```
SpliceLL(x = sort(X), X, splicefit, plot = TRUE, main = "Splicing LL-plot", ...)
```

Arguments

`x`	Vector of points to plot the fitted survival function at. By default we plot it at the data points.
`X`	Data used for fitting the distribution.
`splicefit`	A `SpliceFit` object, e.g. output from `SpliceFitPareto` or `SpliceFitGPD`.
`plot`	Logical indicating if the splicing LL-plot should be made, default is `TRUE`.
`main`	Title for the plot, default is `"Splicing LL-plot"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The LL-plot consists of the points

$$(\log(x_{i,n}), \log(1 - \hat{F}(x_{i,n})))$$

for $i = 1, \ldots, n$ with n the length of the data, $x_{i,n}$ the i-th smallest observation and $\hat{F}$ the empirical distribution function. Then, the line

$$(\log(x), \log(1 - \hat{F}_{spliced}(x))),$$

with $\hat{F}_{spliced}$ the fitted spliced distribution function, is added.

Use `SpliceLL_TB` for censored data.

See Reynkens et al. (2017) and Section 4.3.1 in Albrecher et al. (2017) for more details.

Value

A list with following components:

`logX`	Vector of the logarithms of the sorted data.
`sll.the`	Vector of the theoretical log-probabilities $\log(1 - \hat{F}_{spliced}(x))$.
`logx`	Vector of the logarithms of the points to plot the fitted survival function at.
`sll.emp`	Vector of the empirical log-probabilities $\log(1 - \hat{F}(x_{i,n}))$.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Reynkens, T., Verbelen, R., Beirlant, J. and Antonio, K. (2017). "Modelling Censored Losses Using Splicing: a Global Fit Strategy With Mixed Erlang and Extreme Value Distributions". *Insurance: Mathematics and Economics*, 77, 65–77.

Verbelen, R., Gong, L., Antonio, K., Badescu, A. and Lin, S. (2015). "Fitting Mixtures of Erlangs to Censored and Truncated Data Using the EM Algorithm." *Astin Bulletin*, 45, 729–758

See Also

`SpliceLL_TB`, `pSplice`, `ecdf`, `SpliceFitPareto`, `SpliceFitGPD`, `SpliceECDF`, `SplicePP`, `SpliceQQ`

Examples

```
## Not run:

# Pareto random sample
X <- rpareto(1000, shape = 2)
```

```
# Splice ME and Pareto
splicefit <- SpliceFitPareto(X, 0.6)



x <- seq(0, 20, 0.01)

# Plot of spliced CDF
plot(x, pSplice(x, splicefit), type="l", xlab="x", ylab="F(x)")

# Plot of spliced PDF
plot(x, dSplice(x, splicefit), type="l", xlab="x", ylab="f(x)")



# Fitted survival function and empirical survival function
SpliceECDF(x, X, splicefit)

# Log-log plot with empirical survival function and fitted survival function
SpliceLL(x, X, splicefit)

# PP-plot of empirical survival function and fitted survival function
SplicePP(X, splicefit)

# PP-plot of empirical survival function and
# fitted survival function with log-scales
SplicePP(X, splicefit, log=TRUE)

# Splicing QQ-plot
SpliceQQ(X, splicefit)

## End(Not run)
```

SpliceLL_TB *LL-plot with fitted and Turnbull survival function*

Description

This function plots the logarithm of the Turnbull survival function (which is suitable for interval censored data) versus the logarithm of the data. Moreover, the logarithm of the fitted survival function of the spliced distribution is added.

Usage

```
SpliceLL_TB(x = sort(L), L, U = L, censored, splicefit, plot = TRUE,
            main = "Splicing LL-plot", ...)
```

Arguments

x	Vector of points to plot the fitted survival function at. By default we plot it at the points L.
L	Vector of length n with the lower boundaries of the intervals for interval censored data or the observed data for right censored data.
U	Vector of length n with the upper boundaries of the intervals. By default, they are equal to L.
censored	A logical vector of length n indicating if an observation is censored.
splicefit	A SpliceFit object, e.g. output from SpliceFiticPareto.
plot	Logical indicating if the splicing LL-plot should be made, default is TRUE.
main	Title for the plot, default is "Splicing LL-plot".
...	Additional arguments for the plot function, see plot for more details.

Details

The LL-plot consists of the points

$$(\log(L_{i,n}), \log(1 - \hat{F}^{TB}(L_{i,n})))$$

for $i = 1, \ldots, n$ with n the length of the data, $x_{i,n}$ the i-th smallest observation and $\hat{F}^{TB}$ the Turnbull estimator for the distribution function. Then, the line

$$(\log(x), \log(1 - \hat{F}_{spliced}(x))),$$

with $\hat{F}_{spliced}$ the fitted spliced distribution function, is added.

Right censored data should be entered as L=l and U=truncupper, and left censored data should be entered as L=trunclower and U=u. The limits trunclower and truncupper are obtained from the SpliceFit object.

If the **interval** package is installed, the icfit function is used to compute the Turnbull estimator. Otherwise, survfit.formula from **survival** is used.

Use SpliceLL for non-censored data.

See Reynkens et al. (2017) and Section 4.3.2 in Albrecher et al. (2017) for more details.

Value

A list with following components:

logX	Vector of the logarithms of the sorted left boundaries of the intervals.
sll.the	Vector of the theoretical log-probabilities $\log(1 - \hat{F}_{spliced}(x)$.
logx	Vector of the logarithms of the points to plot the fitted survival function at.
sll.emp	Vector of the empirical log-probabilities $\log(1 - \hat{F}^{TB}(x_{i,n}))$.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Reynkens, T., Verbelen, R., Beirlant, J. and Antonio, K. (2017). "Modelling Censored Losses Using Splicing: a Global Fit Strategy With Mixed Erlang and Extreme Value Distributions". *Insurance: Mathematics and Economics*, 77, 65–77.

Verbelen, R., Gong, L., Antonio, K., Badescu, A. and Lin, S. (2015). "Fitting Mixtures of Erlangs to Censored and Truncated Data Using the EM Algorithm." *Astin Bulletin*, 45, 729–758

See Also

SpliceLL, pSplice, Turnbull, icfit, SpliceFiticPareto, SpliceTB, SplicePP_TB, SpliceQQ_TB

Examples

```
## Not run:

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X,Y)

# Censoring indicator
censored <- (X>Y)

# Right boundary
U <- Z
U[censored] <- Inf

# Splice ME and Pareto
splicefit <- SpliceFiticPareto(L=Z, U=U, censored=censored, tsplice=quantile(Z,0.9))



x <- seq(0,20,0.1)

# Plot of spliced CDF
plot(x, pSplice(x, splicefit), type="l", xlab="x", ylab="F(x)")

# Plot of spliced PDF
plot(x, dSplice(x, splicefit), type="l", xlab="x", ylab="f(x)")


# Fitted survival function and Turnbull survival function
SpliceTB(x, L=Z, U=U, censored=censored, splicefit=splicefit)
```

```
# Log-log plot with Turnbull survival function and fitted survival function
SpliceLL_TB(x, L=Z, U=U, censored=censored, splicefit=splicefit)


# PP-plot of Turnbull survival function and fitted survival function
SplicePP_TB(L=Z, U=U, censored=censored, splicefit=splicefit)

# PP-plot of Turnbull survival function and
# fitted survival function with log-scales
SplicePP_TB(L=Z, U=U, censored=censored, splicefit=splicefit, log=TRUE)

# QQ-plot using Turnbull survival function and fitted survival function
SpliceQQ_TB(L=Z, U=U, censored=censored, splicefit=splicefit)

## End(Not run)
```

SplicePP *PP-plot with fitted and empirical survival function*

Description

This function plots the fitted survival function of the spliced distribution versus the empirical survival function (determined using the Empirical CDF (ECDF)).

Usage

```
SplicePP(X, splicefit, x = sort(X), log = FALSE, plot = TRUE,
         main = "Splicing PP-plot", ...)
```

Arguments

`X`	Data used for fitting the distribution.
`splicefit`	A `SpliceFit` object, e.g. output from `SpliceFitPareto` or `SpliceFitGPD`.
`x`	Vector of points to plot the functions at. By default we plot them at the data points.
`log`	Logical indicating if minus the logarithms of the survival probabilities are plotted versus each other, default is `FALSE`.
`plot`	Logical indicating if the splicing PP-plot should be made, default is `TRUE`.
`main`	Title for the plot, default is `"Splicing PP-plot"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The PP-plot consists of the points

$$(1 - \hat{F}(x_{i,n}), 1 - \hat{F}_{spliced}(x_{i,n})))$$

for $i = 1, \ldots, n$ with n the length of the data, $x_{i,n}$ the i-th smallest observation, $\hat{F}$ the empirical distribution function and $\hat{F}_{spliced}$ the fitted spliced distribution function. The minus-log version of the PP-plot consists of

$$(-\log(1 - \hat{F}(x_{i,n})), -\log(1 - \hat{F}_{spliced}(x_{i,n})))).$$

Use `SplicePP_TB` for censored data.

See Reynkens et al. (2017) and Section 4.3.1 in Albrecher et al. (2017) for more details.

Value

A list with following components:

`spp.the` Vector of the theoretical probabilities $1 - \hat{F}_{spliced}(x_{i,n})$ (when `log=FALSE`) or $-\log(1 - \hat{F}_{spliced}(x_{i,n}))$ (when `log=TRUE`).

`spp.emp` Vector of the empirical probabilities $1-\hat{F}(x_{i,n})$ (when `log=FALSE`) or $-\log(1-\hat{F}(x_{i,n}))$ (when `log=TRUE`).

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Reynkens, T., Verbelen, R., Beirlant, J. and Antonio, K. (2017). "Modelling Censored Losses Using Splicing: a Global Fit Strategy With Mixed Erlang and Extreme Value Distributions". *Insurance: Mathematics and Economics*, 77, 65–77.

Verbelen, R., Gong, L., Antonio, K., Badescu, A. and Lin, S. (2015). "Fitting Mixtures of Erlangs to Censored and Truncated Data Using the EM Algorithm." *Astin Bulletin*, 45, 729–758

See Also

`SplicePP_TB`, `pSplice`, `ecdf`, `SpliceFitPareto`, `SpliceFitGPD`, `SpliceECDF`, `SpliceLL`, `SpliceQQ`

Examples

```
## Not run:

# Pareto random sample
X <- rpareto(1000, shape = 2)

# Splice ME and Pareto
```

```
splicefit <- SpliceFitPareto(X, 0.6)



x <- seq(0, 20, 0.01)

# Plot of spliced CDF
plot(x, pSplice(x, splicefit), type="l", xlab="x", ylab="F(x)")

# Plot of spliced PDF
plot(x, dSplice(x, splicefit), type="l", xlab="x", ylab="f(x)")



# Fitted survival function and empirical survival function
SpliceECDF(x, X, splicefit)

# Log-log plot with empirical survival function and fitted survival function
SpliceLL(x, X, splicefit)

# PP-plot of empirical survival function and fitted survival function
SplicePP(X, splicefit)

# PP-plot of empirical survival function and
# fitted survival function with log-scales
SplicePP(X, splicefit, log=TRUE)

# Splicing QQ-plot
SpliceQQ(X, splicefit)

## End(Not run)
```

SplicePP_TB *PP-plot with fitted and Turnbull survival function*

Description

This function plots the fitted survival function of the spliced distribution versus the Turnbull survival function (which is suitable for interval censored data).

Usage

```
SplicePP_TB(L, U = L, censored, splicefit, x = NULL, log = FALSE, plot = TRUE,
            main = "Splicing PP-plot", ...)
```

Arguments

L Vector of length n with the lower boundaries of the intervals for interval censored data or the observed data for right censored data.

U	Vector of length n with the upper boundaries of the intervals. By default, they are equal to L.
censored	A logical vector of length n indicating if an observation is censored.
splicefit	A SpliceFit object, e.g. output from SpliceFiticPareto.
x	Vector of points to plot the functions at. When NULL, the default, the empirical quantiles for $1/(n+1), \dots, n/(n+1)$, obtained using the Turnbull estimator, are used.
log	Logical indicating if minus the logarithms of the survival probabilities are plotted versus each other, default is FALSE.
plot	Logical indicating if the splicing PP-plot should be made, default is TRUE.
main	Title for the plot, default is "Splicing PP-plot".
...	Additional arguments for the plot function, see plot for more details.

Details

The PP-plot consists of the points

$$(1 - \hat{F}^{TB}(x_i), 1 - \hat{F}_{spliced}(x_i)))$$

for $i = 1, \dots, n$ with n the length of the data, $x_i = \hat{Q}^{TB}(p_i)$ where $p_i = i/(n+1)$, $\hat{Q}^{TB}$ is the quantile function obtained using the Turnbull estimator, $\hat{F}^{TB}$ the Turnbull estimator for the distribution function and $\hat{F}_{spliced}$ the fitted spliced distribution function. The minus-log version of the PP-plot consists of

$$(-\log(1 - \hat{F}^{TB}(x_i)), -\log(1 - \hat{F}_{spliced}(x_i))).$$

Right censored data should be entered as L=l and U=truncupper, and left censored data should be entered as L=trunclower and U=u. The limits trunclower and truncupper are obtained from the SpliceFit object.

If the **interval** package is installed, the icfit function is used to compute the Turnbull estimator. Otherwise, survfit.formula from **survival** is used.

Use SplicePP for non-censored data.

See Reynkens et al. (2017) and Section 4.3.2 in Albrecher et al. (2017) for more details.

Value

A list with following components:

spp.the	Vector of the theoretical probabilities $1 - \hat{F}_{spliced}(x_i)$ (when log=FALSE) or $-\log(1 - \hat{F}_{spliced}(x_i))$ (when log=TRUE).
spp.emp	Vector of the empirical probabilities $1 - \hat{F}^{TB}(x_i)$ (when log=FALSE) or $-\log(1 - \hat{F}^{TB}(x_i))$ (when log=TRUE).

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Reynkens, T., Verbelen, R., Beirlant, J. and Antonio, K. (2017). "Modelling Censored Losses Using Splicing: a Global Fit Strategy With Mixed Erlang and Extreme Value Distributions". *Insurance: Mathematics and Economics*, 77, 65–77.

Verbelen, R., Gong, L., Antonio, K., Badescu, A. and Lin, S. (2015). "Fitting Mixtures of Erlangs to Censored and Truncated Data Using the EM Algorithm." *Astin Bulletin*, 45, 729–758

See Also

SplicePP, pSplice, Turnbull, icfit, SpliceFiticPareto, SpliceTB, SpliceLL_TB, SpliceQQ_TB

Examples

```
## Not run:

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X,Y)

# Censoring indicator
censored <- (X>Y)

# Right boundary
U <- Z
U[censored] <- Inf

# Splice ME and Pareto
splicefit <- SpliceFiticPareto(L=Z, U=U, censored=censored, tsplice=quantile(Z,0.9))



x <- seq(0,20,0.1)

# Plot of spliced CDF
plot(x, pSplice(x, splicefit), type="l", xlab="x", ylab="F(x)")

# Plot of spliced PDF
plot(x, dSplice(x, splicefit), type="l", xlab="x", ylab="f(x)")


# Fitted survival function and Turnbull survival function
SpliceTB(x, L=Z, U=U, censored=censored, splicefit=splicefit)
```

```
# Log-log plot with Turnbull survival function and fitted survival function
SpliceLL_TB(x, L=Z, U=U, censored=censored, splicefit=splicefit)


# PP-plot of Turnbull survival function and fitted survival function
SplicePP_TB(L=Z, U=U, censored=censored, splicefit=splicefit)

# PP-plot of Turnbull survival function and
# fitted survival function with log-scales
SplicePP_TB(L=Z, U=U, censored=censored, splicefit=splicefit, log=TRUE)

# QQ-plot using Turnbull survival function and fitted survival function
SpliceQQ_TB(L=Z, U=U, censored=censored, splicefit=splicefit)

## End(Not run)
```

SpliceQQ *Splicing quantile plot*

Description

Computes the empirical quantiles of a data vector and the theoretical quantiles of the fitted spliced distribution. These quantiles are then plotted in a splicing QQ-plot with the theoretical quantiles on the x-axis and the empirical quantiles on the y-axis.

Usage

```
SpliceQQ(X, splicefit, p = NULL, plot = TRUE, main = "Splicing QQ-plot", ...)
```

Arguments

X	Vector of n observations.
splicefit	A SpliceFit object, e.g. output from SpliceFitPareto or SpliceFitGPD.
p	Vector of probabilities used in the QQ-plot. If NULL, the default, we take p equal to 1/(n+1),...,n/(n+1).
plot	Logical indicating if the quantiles should be plotted in a splicing QQ-plot, default is TRUE.
main	Title for the plot, default is "Splicing QQ-plot".
...	Additional arguments for the plot function, see plot for more details.

Details

This QQ-plot is given by

$$(Q(p_j), \hat{Q}(p_j)),$$

for $j = 1, \dots, n$ where Q is the quantile function of the fitted splicing model and $\hat{Q}$ is the empirical quantile function and $p_j = j/(n + 1)$.

See Reynkens et al. (2017) and Section 4.3.1 in Albrecher et al. (2017) for more details.

Value

A list with following components:

sqq.the	Vector of the theoretical quantiles of the fitted spliced distribution.
sqq.emp	Vector of the empirical quantiles from the data.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Reynkens, T., Verbelen, R., Beirlant, J. and Antonio, K. (2017). "Modelling Censored Losses Using Splicing: a Global Fit Strategy With Mixed Erlang and Extreme Value Distributions". *Insurance: Mathematics and Economics*, 77, 65–77.

Verbelen, R., Gong, L., Antonio, K., Badescu, A. and Lin, S. (2015). "Fitting Mixtures of Erlangs to Censored and Truncated Data Using the EM Algorithm." *Astin Bulletin*, 45, 729–758

See Also

SpliceQQ_TB, qSplice, SpliceFitPareto, SpliceFitGPD, SpliceECDF, SpliceLL, SplicePP

Examples

```
## Not run:

# Pareto random sample
X <- rpareto(1000, shape = 2)

# Splice ME and Pareto
splicefit <- SpliceFitPareto(X, 0.6)



x <- seq(0, 20, 0.01)

# Plot of spliced CDF
plot(x, pSplice(x, splicefit), type="l", xlab="x", ylab="F(x)")

# Plot of spliced PDF
plot(x, dSplice(x, splicefit), type="l", xlab="x", ylab="f(x)")



# Fitted survival function and empirical survival function
SpliceECDF(x, X, splicefit)

# Log-log plot with empirical survival function and fitted survival function
```

```
SpliceLL(x, X, splicefit)

# PP-plot of empirical survival function and fitted survival function
SplicePP(X, splicefit)

# PP-plot of empirical survival function and
# fitted survival function with log-scales
SplicePP(X, splicefit, log=TRUE)

# Splicing QQ-plot
SpliceQQ(X, splicefit)

## End(Not run)
```

SpliceQQ_TB *Splicing quantile plot using Turnbull estimator*

Description

This function plots the fitted quantile function of the spliced distribution versus quantiles based on the Turnbull survival function (which is suitable for interval censored data).

Usage

```
SpliceQQ_TB(L, U = L, censored, splicefit, p = NULL,
            plot = TRUE, main = "Splicing QQ-plot", ...)
```

Arguments

L	Vector of length n with the lower boundaries of the intervals for interval censored data or the observed data for right censored data.
U	Vector of length n with the upper boundaries of the intervals. By default, they are equal to L.
censored	A logical vector of length n indicating if an observation is censored.
splicefit	A SpliceFit object, e.g. output from SpliceFiticPareto.
p	Vector of probabilities used in the QQ-plot. If NULL, the default, we take p equal to 1/(n+1),...,n/(n+1).
plot	Logical indicating if the quantiles should be plotted in a splicing QQ-plot, default is TRUE.
main	Title for the plot, default is "Splicing QQ-plot".
...	Additional arguments for the plot function, see plot for more details.

Details

This QQ-plot is given by

$$(Q(p_j), \hat{Q}^{TB}(p_j)),$$

for $j = 1, \ldots, n$ where Q is the quantile function of the fitted splicing model, $\hat{Q}^{TB}$ the quantile function obtained using the Turnbull estimator and $p_j = j/(n + 1)$.

If the **interval** package is installed, the `icfit` function is used to compute the Turnbull estimator. Otherwise, `survfit.formula` from **survival** is used.

Right censored data should be entered as `L=l` and `U=truncupper`, and left censored data should be entered as `L=trunclower` and `U=u`. The limits `trunclower` and `truncupper` are obtained from the `SpliceFit` object.

Use `SpliceQQ` for non-censored data.

See Reynkens et al. (2017) and Section 4.3.2 in Albrecher et al. (2017) for more details.

Value

A list with following components:

`sqq.the`	Vector of the theoretical quantiles of the fitted spliced distribution.
`sqq.emp`	Vector of the empirical quantiles from the data (based on the Turnbull estimator).

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Reynkens, T., Verbelen, R., Beirlant, J. and Antonio, K. (2017). "Modelling Censored Losses Using Splicing: a Global Fit Strategy With Mixed Erlang and Extreme Value Distributions". *Insurance: Mathematics and Economics*, 77, 65–77.

Verbelen, R., Gong, L., Antonio, K., Badescu, A. and Lin, S. (2015). "Fitting Mixtures of Erlangs to Censored and Truncated Data Using the EM Algorithm." *Astin Bulletin*, 45, 729–758

See Also

`SpliceQQ`, `qSplice`, `Turnbull`, `icfit`, `SpliceFiticPareto`, `SpliceTB`, `SplicePP_TB`, `SpliceLL_TB`

Examples

```
## Not run:

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)
```

```
# Observed sample
Z <- pmin(X,Y)

# Censoring indicator
censored <- (X>Y)

# Right boundary
U <- Z
U[censored] <- Inf

# Splice ME and Pareto
splicefit <- SpliceFiticPareto(L=Z, U=U, censored=censored, tsplice=quantile(Z,0.9))



x <- seq(0,20,0.1)

# Plot of spliced CDF
plot(x, pSplice(x, splicefit), type="l", xlab="x", ylab="F(x)")

# Plot of spliced PDF
plot(x, dSplice(x, splicefit), type="l", xlab="x", ylab="f(x)")


# Fitted survival function and Turnbull survival function
SpliceTB(x, L=Z, U=U, censored=censored, splicefit=splicefit)


# Log-log plot with Turnbull survival function and fitted survival function
SpliceLL_TB(x, L=Z, U=U, censored=censored, splicefit=splicefit)


# PP-plot of Turnbull survival function and fitted survival function
SplicePP_TB(L=Z, U=U, censored=censored, splicefit=splicefit)

# PP-plot of Turnbull survival function and
# fitted survival function with log-scales
SplicePP_TB(L=Z, U=U, censored=censored, splicefit=splicefit, log=TRUE)

# QQ-plot using Turnbull survival function and fitted survival function
SpliceQQ_TB(L=Z, U=U, censored=censored, splicefit=splicefit)

## End(Not run)
```

SpliceTB *Plot of fitted and Turnbull survival function*

Description

This function plots the fitted survival function of the spliced distribution together with the Turnbull survival function (which is suitable for interval censored data). Moreover, $100(1-\alpha)\%$ confidence intervals are added.

Usage

```
SpliceTB(x = sort(L), L, U = L, censored, splicefit, alpha = 0.05, ...)
```

Arguments

`x`	Vector of points to plot the functions at. By default we plot it at the points `L`.
`L`	Vector of length n with the lower boundaries of the intervals for interval censored data or the observed data for right censored data.
`U`	Vector of length n with the upper boundaries of the intervals. By default, they are equal to `L`.
`censored`	A logical vector of length n indicating if an observation is censored.
`splicefit`	A `SpliceFit` object, e.g. output from `SpliceFiticPareto`.
`alpha`	$100(1-\alpha)\%$ is the confidence level for the confidence intervals. Default is $\alpha = 0.05$.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

Right censored data should be entered as `L=l` and `U=truncupper`, and left censored data should be entered as `L=trunclower` and `U=u`. The limits `trunclower` and `truncupper` are obtained from the `SpliceFit` object.

Use `SpliceECDF` for non-censored data.

See Reynkens et al. (2017) and Section 4.3.2 in Albrecher et al. (2017) for more details.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Reynkens, T., Verbelen, R., Beirlant, J. and Antonio, K. (2017). "Modelling Censored Losses Using Splicing: a Global Fit Strategy With Mixed Erlang and Extreme Value Distributions". *Insurance: Mathematics and Economics*, 77, 65–77.

Verbelen, R., Gong, L., Antonio, K., Badescu, A. and Lin, S. (2015). "Fitting Mixtures of Erlangs to Censored and Truncated Data Using the EM Algorithm." *Astin Bulletin*, 45, 729–758

See Also

`SpliceECDF`, `pSplice`, `Turnbull`, `SpliceFiticPareto`, `SpliceLL_TB`, `SplicePP_TB`, `SpliceQQ_TB`

Examples

```
## Not run:

# Pareto random sample
X <- rpareto(500, shape=2)

# Censoring variable
Y <- rpareto(500, shape=1)

# Observed sample
Z <- pmin(X,Y)

# Censoring indicator
censored <- (X>Y)

# Right boundary
U <- Z
U[censored] <- Inf

# Splice ME and Pareto
splicefit <- SpliceFiticPareto(L=Z, U=U, censored=censored, tsplice=quantile(Z,0.9))



x <- seq(0,20,0.1)

# Plot of spliced CDF
plot(x, pSplice(x, splicefit), type="l", xlab="x", ylab="F(x)")

# Plot of spliced PDF
plot(x, dSplice(x, splicefit), type="l", xlab="x", ylab="f(x)")


# Fitted survival function and Turnbull survival function
SpliceTB(x, L=Z, U=U, censored=censored, splicefit=splicefit)


# Log-log plot with Turnbull survival function and fitted survival function
SpliceLL_TB(x, L=Z, U=U, censored=censored, splicefit=splicefit)


# PP-plot of Turnbull survival function and fitted survival function
SplicePP_TB(L=Z, U=U, censored=censored, splicefit=splicefit)

# PP-plot of Turnbull survival function and
# fitted survival function with log-scales
SplicePP_TB(L=Z, U=U, censored=censored, splicefit=splicefit, log=TRUE)

# QQ-plot using Turnbull survival function and fitted survival function
SpliceQQ_TB(L=Z, U=U, censored=censored, splicefit=splicefit)

## End(Not run)
```

stdf *Non-parametric estimators of the STDF*

Description

Non-parametric estimators of the stable tail dependence function (STDF): $\hat{l}_k(x)$ and $\tilde{l}_k(x)$.

Usage

```
stdf(x, k, X, alpha = 0.5)

stdf2(x, k, X)
```

Arguments

x	A d-dimensional point to estimate the STDF in.
k	Value of the tail index k.
X	A data matrix of dimensions n by d with observations in the rows.
alpha	The parameter α of the estimator $\hat{l}_k(x)$ (`stdf`), default is 0.5. This argument is not used in `stdf2`.

Details

The stable tail dependence function in x can be estimated by

$$\hat{l}_k(x) = 1/k \sum_{i=1}^{n} 1_{\{\exists j \in \{1,\dots,d\}: \hat{F}_j(X_{i,j}) > 1-k/nx_j\}}$$

with

$$\hat{F}_j(X_{i,j}) = (R_{i,j} - \alpha)/n$$

where $R_{i,j}$ is the rank of $X_{i,j}$ among the n observations in the j-th dimension:

$$R_{i,j} = \sum_{m=1}^{n} 1_{\{X_{m,j} \le X_{i,j}\}}.$$

This estimator is implemented in `stdf`.

The second estimator is given by

$$\tilde{l}_k(x) = 1/k \sum_{i=1}^{n} 1_{\{X_{i,1} \ge X^{(1)}_{n-[kx_1]+1,n} or ... or X_{i,d} \ge X^{(d)}_{n-[kx_d]+1,n}\}}$$

where $X^{(j)}_{i,n}$ is the i-th smallest observation in the j-th dimension. This estimator is implemented in `stdf2`.

See Section 4.5 of Beirlant et al. (2016) for more details.

Value

`stdf` returns the estimate $\hat{l}_k(x)$ and `stdf2` returns the estimate $\tilde{l}_k(x)$.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant J., Goegebeur Y., Segers, J. and Teugels, J. (2004). *Statistics of Extremes: Theory and Applications*, Wiley Series in Probability, Wiley, Chichester.

Examples

```
# Generate data matrix
X <- cbind(rpareto(100,2), rpareto(100,3))

# Tail index
k <- 20

# Point to evaluate the STDF in
x <- c(2,3)

# First estimate
stdf(x, k, X)

# Second estimate
stdf2(x, k, X)
```

`tBurr` *The truncated Burr distribution*

Description

Density, distribution function, quantile function and random generation for the truncated Burr distribution (type XII).

Usage

```
dtburr(x, alpha, rho, eta = 1, endpoint = Inf, log = FALSE)
ptburr(x, alpha, rho, eta = 1, endpoint = Inf, lower.tail = TRUE, log.p = FALSE)
qtburr(p, alpha, rho, eta = 1, endpoint = Inf, lower.tail = TRUE, log.p = FALSE)
rtburr(n, alpha, rho, eta = 1, endpoint = Inf)
```

Arguments

- `x` — Vector of quantiles.
- `p` — Vector of probabilities.
- `n` — Number of observations.
- `alpha` — The α parameter of the truncated Burr distribution, a strictly positive number.
- `rho` — The ρ parameter of the truncated Burr distribution, a strictly negative number.
- `eta` — The η parameter of the truncated Burr distribution, a strictly positive number. The default value is `1`.
- `endpoint` — Endpoint of the truncated Burr distribution. The default value is `Inf` for which the truncated Burr distribution corresponds to the ordinary Burr distribution.
- `log` — Logical indicating if the densities are given as $\log(f)$, default is `FALSE`.
- `lower.tail` — Logical indicating if the probabilities are of the form $P(X \leq x)$ (`TRUE`) or $P(X > x)$ (`FALSE`). Default is `TRUE`.
- `log.p` — Logical indicating if the probabilities are given as $\log(p)$, default is `FALSE`.

Details

The Cumulative Distribution Function (CDF) of the truncated Burr distribution is equal to $F_T(x) = F(x)/F(T)$ for $x \leq T$ where F is the CDF of the ordinary Burr distribution and T is the endpoint (truncation point) of the truncated Burr distribution.

Value

`dtburr` gives the density function evaluated in x, `ptburr` the CDF evaluated in x and `qtburr` the quantile function evaluated in p. The length of the result is equal to the length of x or p.

`rtburr` returns a random sample of length n.

Author(s)

Tom Reynkens.

See Also

`Burr`, `Distributions`

Examples

```
# Plot of the PDF
x <- seq(0, 10, 0.01)
plot(x, dtburr(x, alpha=2, rho=-1, endpoint=9), xlab="x", ylab="PDF", type="l")

# Plot of the CDF
x <- seq(0, 10, 0.01)
plot(x, ptburr(x, alpha=2, rho=-1, endpoint=9), xlab="x", ylab="CDF", type="l")
```

tExp *The truncated exponential distribution*

Description

Density, distribution function, quantile function and random generation for the truncated exponential distribution.

Usage

```
dtexp(x, rate = 1, endpoint = Inf, log = FALSE)
ptexp(x, rate = 1, endpoint = Inf, lower.tail = TRUE, log.p = FALSE)
qtexp(p, rate = 1, endpoint = Inf, lower.tail = TRUE, log.p = FALSE)
rtexp(n, rate = 1, endpoint = Inf)
```

Arguments

`x`	Vector of quantiles.
`p`	Vector of probabilities.
`n`	Number of observations.
`rate`	The rate parameter for the exponential distribution, default is 1.
`endpoint`	Endpoint of the truncated exponential distribution. The default value is `Inf` for which the truncated exponential distribution corresponds to the ordinary exponential distribution.
`log`	Logical indicating if the densities are given as $\log(f)$, default is `FALSE`.
`lower.tail`	Logical indicating if the probabilities are of the form $P(X \le x)$ (`TRUE`) or $P(X > x)$ (`FALSE`). Default is `TRUE`.
`log.p`	Logical indicating if the probabilities are given as $\log(p)$, default is `FALSE`.

Details

The Cumulative Distribution Function (CDF) of the truncated exponential distribution is equal to $F_T(x) = F(x)/F(T)$ for $x \le T$ where F is the CDF of the ordinary exponential distribution and T is the endpoint (truncation point) of the truncated exponential distribution.

Value

`dtexp` gives the density function evaluated in x, `ptexp` the CDF evaluated in x and `qtexp` the quantile function evaluated in p. The length of the result is equal to the length of x or p.

`rtexp` returns a random sample of length n.

Author(s)

Tom Reynkens.

See Also

Exponential, Distributions

Examples

```
# Plot of the PDF
x <- seq(0, 10, 0.01)
plot(x, dtexp(x, rate = 2, endpoint=5), xlab="x", ylab="PDF", type="l")

# Plot of the CDF
x <- seq(0, 10, 0.01)
plot(x, ptexp(x, rate = 2, endpoint=5), xlab="x", ylab="CDF", type="l")
```

---

tFrechet *The truncated Frechet distribution*

Description

Density, distribution function, quantile function and random generation for the truncated Fréchet distribution.

Usage

```
dtfrechet(x, shape, loc = 0, scale = 1, endpoint = Inf, log = FALSE)
ptfrechet(x, shape, loc = 0, scale = 1, endpoint = Inf, lower.tail = TRUE, log.p = FALSE)
qtfrechet(p, shape, loc = 0, scale = 1, endpoint = Inf, lower.tail = TRUE, log.p = FALSE)
rtfrechet(n, shape, loc = 0, scale = 1, endpoint = Inf)
```

Arguments

x	Vector of quantiles.
p	Vector of probabilities.
n	Number of observations.
shape	Shape parameter of the Fréchet distribution.
loc	Location parameter of the Fréchet distribution, default is 0.
scale	Scale parameter of the Fréchet distribution, default is 1.
endpoint	Endpoint of the truncated Fréchet distribution. The default value is Inf for which the truncated Fréchet distribution corresponds to the ordinary Fréchet distribution.
log	Logical indicating if the densities are given as $\log(f)$, default is FALSE.
lower.tail	Logical indicating if the probabilities are of the form $P(X \leq x)$ (TRUE) or $P(X > x)$ (FALSE). Default is TRUE.
log.p	Logical indicating if the probabilities are given as $\log(p)$, default is FALSE.

Details

The Cumulative Distribution Function (CDF) of the truncated Fréchet distribution is equal to $F_T(x) = F(x)/F(T)$ for $x \leq T$ where F is the CDF of an ordinary Fréchet distribution and T is the endpoint (truncation point) of the truncated Fréchet distribution.

Value

`dtfrechet` gives the density function evaluated in x, `ptfrechet` the CDF evaluated in x and `qtfrechet` the quantile function evaluated in p. The length of the result is equal to the length of x or p.

`rtfrechet` returns a random sample of length n.

Author(s)

Tom Reynkens.

See Also

`Fréchet`, `Distributions`

Examples

```
# Plot of the PDF
x <- seq(1, 10, 0.01)
plot(x, dtfrechet(x, shape=2, endpoint=5), xlab="x", ylab="PDF", type="l")

# Plot of the CDF
x <- seq(1, 10, 0.01)
plot(x, ptfrechet(x, shape=2, endpoint=5), xlab="x", ylab="CDF", type="l")
```

`tGPD` *The truncated generalised Pareto distribution*

Description

Density, distribution function, quantile function and random generation for the truncated Generalised Pareto Distribution (GPD).

Usage

```
dtgpd(x, gamma, mu = 0, sigma, endpoint = Inf, log = FALSE)
ptgpd(x, gamma, mu = 0, sigma, endpoint = Inf, lower.tail = TRUE, log.p = FALSE)
qtgpd(p, gamma, mu = 0, sigma, endpoint = Inf, lower.tail = TRUE, log.p = FALSE)
rtgpd(n, gamma, mu = 0, sigma, endpoint = Inf)
```

Arguments

x	Vector of quantiles.
p	Vector of probabilities.
n	Number of observations.
gamma	The γ parameter of the GPD, a real number.
mu	The μ parameter of the GPD, a strictly positive number. Default is 0.
sigma	The σ parameter of the GPD, a strictly positive number.
endpoint	Endpoint of the truncated GPD. The default value is `Inf` for which the truncated GPD corresponds to the ordinary GPD.
log	Logical indicating if the densities are given as $\log(f)$, default is `FALSE`.
lower.tail	Logical indicating if the probabilities are of the form $P(X \leq x)$ (`TRUE`) or $P(X > x)$ (`FALSE`). Default is `TRUE`.
log.p	Logical indicating if the probabilities are given as $\log(p)$, default is `FALSE`.

Details

The Cumulative Distribution Function (CDF) of the truncated GPD is equal to $F_T(x) = F(x)/F(T)$ for $x \leq T$ where F is the CDF of the ordinary GPD and T is the endpoint (truncation point) of the truncated GPD.

Value

`dtgpd` gives the density function evaluated in x, `ptgpd` the CDF evaluated in x and `qtgpd` the quantile function evaluated in p. The length of the result is equal to the length of x or p.

`rtgpd` returns a random sample of length n.

Author(s)

Tom Reynkens

See Also

`tGPD`, `Pareto`, `Distributions`

Examples

```
# Plot of the PDF
x <- seq(0, 10, 0.01)
plot(x, dtgpd(x, gamma=1/2, sigma=5, endpoint=8), xlab="x", ylab="PDF", type="l")

# Plot of the CDF
x <- seq(0, 10, 0.01)
plot(x, ptgpd(x, gamma=1/2, sigma=5, endpoint=8), xlab="x", ylab="CDF", type="l")
```

tlnorm *The truncated log-normal distribution*

Description

Density, distribution function, quantile function and random generation for the truncated log-normal distribution.

Usage

```
dtlnorm(x, meanlog = 0, sdlog = 1, endpoint = Inf, log = FALSE)
ptlnorm(x, meanlog = 0, sdlog = 1, endpoint = Inf, lower.tail = TRUE, log.p = FALSE)
qtlnorm(p, meanlog = 0, sdlog = 1, endpoint = Inf, lower.tail = TRUE, log.p = FALSE)
rtlnorm(n, meanlog = 0, sdlog = 1, endpoint = Inf)
```

Arguments

`x`	Vector of quantiles.
`p`	Vector of probabilities.
`n`	Number of observations.
`meanlog`	Mean of the distribution on the log scale, default is 0.
`sdlog`	Standard deviation of the distribution on the log scale, default is 1.
`endpoint`	Endpoint of the truncated log-normal distribution. The default value is `Inf` for which the truncated log-normal distribution corresponds to the ordinary log-normal distribution.
`log`	Logical indicating if the densities are given as $\log(f)$, default is `FALSE`.
`lower.tail`	Logical indicating if the probabilities are of the form $P(X \le x)$ (`TRUE`) or $P(X > x)$ (`FALSE`). Default is `TRUE`.
`log.p`	Logical indicating if the probabilities are given as $\log(p)$, default is `FALSE`.

Details

The Cumulative Distribution Function (CDF) of the truncated log-normal distribution is equal to $F_T(x) = F(x)/F(T)$ for $x \le T$ where F is the CDF of the ordinary log-normal distribution and T is the endpoint (truncation point) of the truncated log-normal distribution.

Value

`dtlnorm` gives the density function evaluated in x, `ptlnorm` the CDF evaluated in x and `qtlnorm` the quantile function evaluated in p. The length of the result is equal to the length of x or p.

`rtlnorm` returns a random sample of length n.

Author(s)

Tom Reynkens.

See Also

Lognormal, Distributions

Examples

```
# Plot of the PDF
x <- seq(0, 10, 0.01)
plot(x, dtlnorm(x, endpoint=9), xlab="x", ylab="PDF", type="l")

# Plot of the CDF
x <- seq(0, 10, 0.01)
plot(x, ptlnorm(x, endpoint=9), xlab="x", ylab="CDF", type="l")
```

---

tPareto *The truncated Pareto distribution*

Description

Density, distribution function, quantile function and random generation for the truncated Pareto distribution.

Usage

```
dtpareto(x, shape, scale = 1, endpoint = Inf, log = FALSE)
ptpareto(x, shape, scale = 1, endpoint = Inf, lower.tail = TRUE, log.p = FALSE)
qtpareto(p, shape, scale = 1, endpoint = Inf, lower.tail = TRUE, log.p = FALSE)
rtpareto(n, shape, scale = 1, endpoint = Inf)
```

Arguments

`x`	Vector of quantiles.
`p`	Vector of probabilities.
`n`	Number of observations.
`shape`	The shape parameter of the truncated Pareto distribution, a strictly positive number.
`scale`	The scale parameter of the truncated Pareto distribution, a strictly positive number. Its default value is `1`.
`endpoint`	Endpoint of the truncated Pareto distribution. The default value is `Inf` for which the truncated Pareto distribution corresponds to the ordinary Pareto distribution.
`log`	Logical indicating if the densities are given as $\log(f)$, default is `FALSE`.
`lower.tail`	Logical indicating if the probabilities are of the form $P(X \le x)$ (`TRUE`) or $P(X > x)$ (`FALSE`). Default is `TRUE`.
`log.p`	Logical indicating if the probabilities are given as $\log(p)$, default is `FALSE`.

Details

The Cumulative Distribution Function (CDF) of the truncated Pareto distribution is equal to $F_T(x) = F(x)/F(T)$ for $x \le T$ where F is the CDF of an ordinary Pareto distribution and T is the endpoint (truncation point) of the truncated Pareto distribution.

Value

dtpareto gives the density function evaluated in x, ptpareto the CDF evaluated in x and qtpareto the quantile function evaluated in p. The length of the result is equal to the length of x or p.

rtpareto returns a random sample of length n.

Author(s)

Tom Reynkens

See Also

Pareto, Distributions

Examples

```
# Plot of the PDF
x = seq(1,10,0.01)
plot(x, dtpareto(x, shape=2, endpoint=10), xlab="x", ylab="PDF", type="l")

# Plot of the CDF
x = seq(1,10,0.01)
plot(x, ptpareto(x, shape=2, endpoint=10), xlab="x", ylab="CDF", type="l")
```

---

trDT *Truncation odds*

Description

Estimates of truncation odds of the truncated probability mass under the untruncated distribution using truncated Hill.

Usage

```
trDT(data, r = 1, gamma, plot = FALSE, add = FALSE, main = "Estimates of DT", ...)
```

Arguments

data	Vector of n observations.
r	Trimming parameter, default is 1 (no trimming).
gamma	Vector of $n-1$ estimates for the EVI obtained from trHill.

plot	Logical indicating if the estimates of D_T should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates of D_T should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "Estimates of DT".
...	Additional arguments for the plot function, see plot for more details.

Details

The truncation odds is defined as

$$D_T = (1 - F(T))/F(T)$$

with T the upper truncation point and F the CDF of the untruncated distribution (e.g. Pareto distribution).

We estimate this truncation odds as

$$\hat{D}_T = \max\{(k+1)/(n+1)(R_{r,k,n}^{1/\gamma_k} - 1/(k+1))/(1 - R_{r,k,n}^{1/\gamma_k}), 0\}$$

with $R_{r,k,n} = X_{n-k,n}/X_{n-r+1,n}$.

See Beirlant et al. (2016) or Section 4.2.3 of Albrecher et al. (2017) for more details.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
DT	Vector of the corresponding estimates for the truncation odds D_T.

Author(s)

Tom Reynkens based on R code of Dries Cornilly.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant, J., Fraga Alves, M.I. and Gomes, M.I. (2016). "Tail fitting for Truncated and Non-truncated Pareto-type Distributions." *Extremes*, 19, 429–462.

See Also

trHill, trEndpoint, trQuant, trDTMLE

Examples

```
# Sample from truncated Pareto distribution.
# truncated at 99% quantile
shape <- 2
X <- rtpareto(n=1000, shape=shape, endpoint=qpareto(0.99, shape=shape))

# Truncated Hill estimator
trh <- trHill(X, plot=TRUE, ylim=c(0,2))

# Truncation odds
dt <- trDT(X, gamma=trh$gamma, plot=TRUE, ylim=c(0,0.05))
```

`trDTMLE` *Truncation odds*

Description

Estimates of truncation odds of the truncated probability mass under the untruncated distribution using truncated MLE.

Usage

```
trDTMLE(data, gamma, tau, plot = FALSE, add = FALSE, main = "Estimates of DT", ...)
```

Arguments

`data`	Vector of n observations.
`gamma`	Vector of $n-1$ estimates for the EVI obtained from `trMLE`.
`tau`	Vector of $n-1$ estimates for the τ obtained from `trMLE`.
`plot`	Logical indicating if the estimates of D_T should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates of D_T should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Estimates of DT"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The truncation odds is defined as

$$D_T = (1 - F(T))/F(T)$$

with T the upper truncation point and F the CDF of the untruncated distribution (e.g. GPD).

We estimate this truncation odds as

$$\hat{D}_T = \max\{(k+1)/(n+1)((1+\hat{\tau}_k E_{1,k})^{-1/\hat{\xi}_k} - 1/(k+1))/(1-(1+\hat{\tau}_k E_{1,k})^{-1/\hat{\xi}_k}), 0\}$$

with $E_{1,k} = X_{n,n} - X_{n-k,n}$.

See Beirlant et al. (2017) for more details.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
DT	Vector of the corresponding estimates for the truncation odds D_T.

Author(s)

Tom Reynkens.

References

Beirlant, J., Fraga Alves, M. I. and Reynkens, T. (2017). "Fitting Tails Affected by Truncation". *Electronic Journal of Statistics*, 11(1), 2026–2065.

See Also

trMLE, trEndpointMLE, trProbMLE, trQuantMLE, trTestMLE, trDT

Examples

```
# Sample from GPD truncated at 99% quantile
gamma <- 0.5
sigma <- 1.5
X <- rtgpd(n=250, gamma=gamma, sigma=sigma, endpoint=qgpd(0.99, gamma=gamma, sigma=sigma))

# Truncated ML estimator
trmle <- trMLE(X, plot=TRUE, ylim=c(0,2))

# Truncation odds
dtmle <- trDTMLE(X, gamma=trmle$gamma, tau=trmle$tau, plot=TRUE, ylim=c(0,0.05))
```

---

trEndpoint *Estimator of endpoint*

Description

Estimator of endpoint using truncated Hill estimates.

Usage

```
trEndpoint(data, r = 1, gamma, plot = FALSE, add = FALSE,
           main = "Estimates of endpoint", ...)
```

Arguments

data	Vector of n observations.
r	Trimming parameter, default is 1 (no trimming).
gamma	Vector of $n-1$ estimates for the EVI obtained from trHill.
plot	Logical indicating if the estimates of T should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates of T should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "Estimates of endpoint".
...	Additional arguments for the plot function, see plot for more details.

Details

The endpoint is estimated as

$$\hat{T}_{k,n} = \max\{X_{n-k,n}(((X_{n-k,n}/X_{n,n})^{1/\hat{\gamma}_k} - 1/(k+1))/(1-1/(k+1)))^{-\hat{\gamma}_k}, X_{n,n}\}$$

with $\hat{\gamma}_k$ the Hill estimates adapted for truncation.

See Beirlant et al. (2016) or Section 4.2.3 of Albrecher et al. (2017) for more details.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
Tk	Vector of the corresponding estimates for the endpoint T.

Author(s)

Tom Reynkens based on R code of Dries Cornilly.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant, J., Fraga Alves, M.I. and Gomes, M.I. (2016). "Tail fitting for Truncated and Non-truncated Pareto-type Distributions." *Extremes*, 19, 429–462.

See Also

trHill, trDT, trEndpointMLE

Examples

```
# Sample from truncated Pareto distribution.
# truncated at 99% quantile
shape <- 2
X <- rtpareto(n=1000, shape=shape, endpoint=qpareto(0.99, shape=shape))

# Truncated Hill estimator
trh <- trHill(X, plot=TRUE, ylim=c(0,2))

# Endpoint
trEndpoint(X, gamma=trh$gamma, plot=TRUE, ylim=c(8,12))
abline(h=qpareto(0.99, shape=shape), lty=2)
```

`trEndpointMLE` *Estimator of endpoint*

Description

Estimator of endpoint using truncated ML estimates.

Usage

```
trEndpointMLE(data, gamma, tau, plot = FALSE, add = FALSE,
              main = "Estimates of endpoint", ...)
```

Arguments

`data`	Vector of n observations.
`gamma`	Vector of $n-1$ estimates for the EVI obtained from `trMLE`.
`tau`	Vector of $n-1$ estimates for the τ obtained from `trMLE`.
`plot`	Logical indicating if the estimates of T should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates of T should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Estimates of endpoint"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The endpoint is estimated as

$$\hat{T}_k = X_{n-k,n} + 1/\hat{\tau}_k[((1-1/k)/((1+\hat{\tau}_k(X_{n,n}-X_{n-k,n}))^{-1/\hat{\xi}_k}-1/k))^{\hat{\xi}_k}-1]$$

with $\hat{\gamma}_k$ and $\hat{\tau}_k$ the truncated ML estimates for γ and τ.

See Beirlant et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`Tk`	Vector of the corresponding estimates for the endpoint T.

Author(s)

Tom Reynkens.

References

Beirlant, J., Fraga Alves, M. I. and Reynkens, T. (2017). "Fitting Tails Affected by Truncation". *Electronic Journal of Statistics*, 11(1), 2026–2065.

See Also

`trMLE`, `trDTMLE`, `trProbMLE`, `trQuantMLE`, `trTestMLE`, `trEndpoint`

Examples

```
# Sample from GPD truncated at 99% quantile
gamma <- 0.5
sigma <- 1.5
X <- rtgpd(n=250, gamma=gamma, sigma=sigma, endpoint=qgpd(0.99, gamma=gamma, sigma=sigma))

# Truncated ML estimator
trmle <- trMLE(X, plot=TRUE, ylim=c(0,2))

# Endpoint
trEndpointMLE(X, gamma=trmle$gamma, tau=trmle$tau, plot=TRUE, ylim=c(0,50))
abline(h=qgpd(0.99, gamma=gamma, sigma=sigma), lty=2)
```

`trHill` *Hill estimator for upper truncated data*

Description

Computes the Hill estimator for positive extreme value indices, adapted for upper truncation, as a function of the tail parameter k (Aban et al. 2006; Beirlant et al., 2016). Optionally, these estimates are plotted as a function of k.

Usage

```
trHill(data, r = 1, tol = 1e-08, maxiter = 100, logk = FALSE,
       plot = FALSE, add = FALSE, main = "Estimates of the EVI", ...)
```

Arguments

data	Vector of n observations.
r	Trimming parameter, default is 1 (no trimming).
tol	Numerical tolerance for stopping criterion used in Newton-Raphson iterations, default is 1e-08.
maxiter	Maximum number of Newton-Raphson iterations, default is 100.
logk	Logical indicating if the estimates are plotted as a function of $\log(k)$ (logk=TRUE) or as a function of k. Default is FALSE.
plot	Logical indicating if the estimates of γ should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates of γ should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "Estimates of the EVI".
...	Additional arguments for the plot function, see plot for more details.

Details

The truncated Hill estimator is the MLE for γ under the truncated Pareto distribution.

To estimate the EVI using the truncated Hill estimator an equation needs to be solved. Beirlant et al. (2016) propose to use Newton-Raphson iterations to solve this equation. We take the trimmed Hill estimates as starting values for this algorithm. The trimmed Hill estimator is defined as

$$H_{r,k,n} = 1/(k-r+1)\sum_{j=r}^{k}\log(X_{n-j+1,n}) - \log(X_{n-k,n})$$

for $1 \le r < k < n$ and is a basic extension of the Hill estimator for upper truncated data (the ordinary Hill estimator is obtained for $r = 1$).

The equation that needs to be solved is

$$H_{r,k,n} = \gamma + R_{r,k,n}^{1/\gamma}\log(R_{r,k,n})/(1 - R_{r,k,n}^{1/\gamma})$$

with $R_{r,k,n} = X_{n-k,n}/X_{n-r+1,n}$.

See Beirlant et al. (2016) or Section 4.2.3 of Albrecher et al. (2017) for more details.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
gamma	Vector of the corresponding estimates for γ.
H	Vector of corresponding trimmed Hill estimates.

Author(s)

Tom Reynkens based on R code of Dries Cornilly.

References

Aban, I.B., Meerschaert, M.M. and Panorska, A.K. (2006). "Parameter Estimation for the Truncated Pareto Distribution." *Journal of the American Statistical Association*, 101, 270–277.

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant, J., Fraga Alves, M.I. and Gomes, M.I. (2016). "Tail fitting for Truncated and Non-truncated Pareto-type Distributions." *Extremes*, 19, 429–462.

See Also

Hill, trDT, trEndpoint, trProb, trQuant, trMLE

Examples

```
# Sample from truncated Pareto distribution.
# truncated at 99% quantile
shape <- 2
X <- rtpareto(n=1000, shape=shape, endpoint=qpareto(0.99, shape=shape))

# Truncated Hill estimator
trh <- trHill(X, plot=TRUE, ylim=c(0,2))
```

---

trMLE *MLE estimator for upper truncated data*

Description

Computes the ML estimator for the extreme value index, adapted for upper truncation, as a function of the tail parameter k (Beirlant et al., 2017). Optionally, these estimates are plotted as a function of k.

Usage

```
trMLE(data, start = c(1, 1), eps = 10^(-10),
      plot = TRUE, add = FALSE, main = "Estimates for EVI", ...)
```

Arguments

`data`	Vector of n observations.
`start`	Starting values for γ and τ for the numerical optimisation.
`eps`	Numerical tolerance, see Details. By default it is equal to `10^(-10)`.
`plot`	Logical indicating if the estimates of γ should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates of γ should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Estimates of the EVI"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

We compute the MLE for the γ and σ parameters of the truncated GPD. For numerical reasons, we compute the MLE for $\tau = \gamma/\sigma$ and transform this estimate to σ.

The log-likelihood is given by

$$(k-1)\ln\tau - (k-1)\ln\xi - (1+1/\xi)\sum_{j=2}^{k}\ln(1+\tau E_{j,k}) - (k-1)\ln(1-(1+\tau E_{1,k})^{-1/\xi})$$

with $E_{j,k} = X_{n-j+1,n} - X_{n-k,n}$.

In order to meet the restrictions $\sigma = \xi/\tau > 0$ and $1 + \tau E_{j,k} > 0$ for $j = 1, \ldots, k$, we require the estimates of these quantities to be larger than the numerical tolerance value `eps`.

See Beirlant et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`gamma`	Vector of the corresponding estimates for γ.
`tau`	Vector of the corresponding estimates for τ.
`sigma`	Vector of the corresponding estimates for σ.
`conv`	Convergence indicator of `optim`.

Author(s)

Tom Reynkens.

References

Beirlant, J., Fraga Alves, M. I. and Reynkens, T. (2017). "Fitting Tails Affected by Truncation". *Electronic Journal of Statistics*, 11(1), 2026–2065.

See Also

`trDTMLE`, `trEndpointMLE`, `trProbMLE`, `trQuantMLE`, `trTestMLE`, `trHill`, `GPDmle`

Examples

```
# Sample from GPD truncated at 99% quantile
gamma <- 0.5
sigma <- 1.5
X <- rtgpd(n=250, gamma=gamma, sigma=sigma, endpoint=qgpd(0.99, gamma=gamma, sigma=sigma))

# Truncated ML estimator
trmle <- trMLE(X, plot=TRUE, ylim=c(0,2))
```

trParetoQQ *Truncated Pareto quantile plot*

Description

Extension of the Pareto QQ-plot as described in Beirlant et al. (2016).

Usage

```
trParetoQQ(data, r = 1, DT, kstar = NULL, plot = TRUE, main = "TPa QQ-plot", ...)
```

Arguments

`data`	Vector of n observations.
`r`	Trimming parameter, default is `1` (no trimming).
`DT`	Vector of $n-1$ estimates for the truncation odds D_T obtained from `trDT`.
`kstar`	Value for k used to construct the plot. When `NULL` (default), a value will be chosen by maximising the correlation between the empirical and theoretical quantiles (see Details).
`plot`	Logical indicating if the quantiles should be plotted in a Pareto QQ-plot, default is `TRUE`.
`main`	Title for the plot, default is `"TPa QQ-plot"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The Pareto QQ-plot for truncated data plots

$$(-\log(\hat{D}_{T,r,k^*,n} + j/(n+1)), \log(X_{n-j+1,n}))$$

for $j = 1, \dots, n$.

The value for k^* can be be given by the user or can be determined automatically. In the latter case, we use the k^* that maximises the absolute value of the correlation between $-\log(\hat{D}_{T,r,k^*,n} + j/(n+1))$ and $\log(X_{n-j+1,n})$ for $j = 1, \dots, k$ and $k^* > 10$.

When taking $D_T = 0$, one obtains the ordinary Pareto QQ-plot.

Note that the definition here differs slightly from the one in Beirlant et al. (2016). We plot the empirical and theoretical quantiles the other way around and therefore have to add a minus (before the log).

See Beirlant et al. (2016) for more details.

Value

A list with following components:

- `pqq.the` — Vector of theoretical quantiles $-\log(\hat{D}_{T,r,k^*,n} + j/(n+1))$, see Details.
- `pqq.emp` — Vector of the empirical quantiles from the log-transformed data.
- `kstar` — Optimal value for k or input argument `kstar`, see Details.
- `DTstar` — Estimate of D_T corresponding to `kstar`.

Author(s)

Tom Reynkens.

References

Beirlant, J., Fraga Alves, M.I. and Gomes, M.I. (2016). "Tail fitting for Truncated and Non-truncated Pareto-type Distributions." *Extremes*, 19, 429–462.

See Also

ParetoQQ, trDT

Examples

```
# Endpoint of truncated Pareto distribution
endpoint <- qpareto(0.99, shape=2)

# Generate sample from truncated Pareto distribution
X <- rtpareto(1000, shape=2, endpoint=endpoint)

# Ordinary Pareto QQ-plot
ParetoQQ(X)

# Truncated Hill estimates
gamma <- trHill(X)$gamma

# Estimates for truncation odds
dt <- trDT(X, gamma=gamma)$DT

# Truncated Pareto QQ-plot
trParetoQQ(X, DT=dt)
```

trProb *Estimator of small exceedance probabilities using truncated Hill*

Description

Computes estimates of a small exceedance probability $P(X > q)$ using the estimates for the EVI obtained from the Hill estimator adapted for upper truncation.

Usage

```
trProb(data, r = 1, gamma, q, warnings = TRUE, plot = FALSE, add = FALSE,
       main = "Estimates of small exceedance probability", ...)
```

Arguments

`data`	Vector of n observations.
`r`	Trimming parameter, default is `1` (no trimming).
`gamma`	Vector of $n-1$ estimates for the EVI obtained from `trHill`.
`q`	The used large quantile (we estimate $P(X > q)$ for q large).
`warnings`	Logical indicating if warnings are shown, default is `TRUE`.
`plot`	Logical indicating if the estimates should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Estimates of small exceedance probability"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The probability is estimated as

$$\hat{P}(X > q) = (k+1)/(n+1)((q/X_{n-k,n})^{-1/\gamma_k} - R_{r,k,n}^{1/\hat{\gamma}_k})/(1 - R_{r,k,n}^{1/\hat{\gamma}_k})$$

with $R_{r,k,n} = X_{n-k,n}/X_{n-r+1,n}$ and $\hat{\gamma}_k$ the Hill estimates adapted for truncation.

See Beirlant et al. (2016) or Section 4.2.3 of Albrecher et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`P`	Vector of the corresponding probability estimates.
`q`	The used large quantile.

Author(s)

Tom Reynkens based on R code of Dries Cornilly.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant, J., Fraga Alves, M.I. and Gomes, M.I. (2016). "Tail fitting for Truncated and Non-truncated Pareto-type Distributions." *Extremes*, 19, 429–462.

See Also

trHill, trQuant, Prob, trProbMLE

Examples

```
# Sample from truncated Pareto distribution.
# truncated at 99% quantile
shape <- 2
X <- rtpareto(n=1000, shape=shape, endpoint=qpareto(0.99, shape=shape))

# Truncated Hill estimator
trh <- trHill(X, plot=TRUE, ylim=c(0,2))

# Small probability
trProb(X, gamma=trh$gamma, q=8, plot=TRUE)
```

---

trProbMLE — *Estimator of small exceedance probabilities using truncated MLE*

Description

Computes estimates of a small exceedance probability $P(X > q)$ using the estimates for the EVI obtained from the ML estimator adapted for upper truncation.

Usage

```
trProbMLE(data, gamma, tau, DT, q, plot = FALSE, add = FALSE,
          main = "Estimates of small exceedance probability", ...)
```

Arguments

data	Vector of n observations.
gamma	Vector of $n-1$ estimates for the EVI obtained from trMLE.
tau	Vector of $n-1$ estimates for the τ obtained from trMLE.
DT	Vector of $n-1$ estimates for the truncation odds obtained from trDTMLE.
q	The used large quantile (we estimate $P(X > q)$ for q large).

`plot`	Logical indicating if the estimates should be plotted as a function of k, default is `FALSE`.
`add`	Logical indicating if the estimates should be added to an existing plot, default is `FALSE`.
`main`	Title for the plot, default is `"Estimates of small exceedance probability"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The probability is estimated as

$$\hat{p}_{T,k}(q) = (1 + \hat{D}_{T,k})(k+1)/(n+1)(1 + \hat{\tau}_k(q - X_{n-k,n}))^{-1/\hat{\xi}_k} - \hat{D}_{T,k}$$

with $\hat{\gamma}_k$ and $\hat{\tau}_k$ the ML estimates adapted for truncation and $\hat{D}_T$ the estimates for the truncation odds.

See Beirlant et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`P`	Vector of the corresponding probability estimates.
`q`	The used large quantile.

Author(s)

Tom Reynkens.

References

Beirlant, J., Fraga Alves, M. I. and Reynkens, T. (2017). "Fitting Tails Affected by Truncation". *Electronic Journal of Statistics*, 11(1), 2026–2065.

See Also

`trMLE`, `trDTMLE`, `trQuantMLE`, `trEndpointMLE`, `trTestMLE`, `trProb`, `Prob`

Examples

```
# Sample from GPD truncated at 99% quantile
gamma <- 0.5
sigma <- 1.5
X <- rtgpd(n=250, gamma=gamma, sigma=sigma, endpoint=qgpd(0.99, gamma=gamma, sigma=sigma))

# Truncated ML estimator
trmle <- trMLE(X, plot=TRUE, ylim=c(0,2))

# Truncation odds
```

```
dtmle <- trDTMLE(X, gamma=trmle$gamma, tau=trmle$tau, plot=FALSE)

# Small exceedance probability
trProbMLE(X, gamma=trmle$gamma, tau=trmle$tau, DT=dtmle$DT, plot=TRUE, q=26, ylim=c(0,0.005))
```

---

trQuant *Estimator of large quantiles using truncated Hill*

Description

trQuant computes estimates of large quantiles $Q(1-p)$ of the truncated distribution using the estimates for the EVI obtained from the Hill estimator adapted for upper truncation. trQuantW computes estimates of large quantiles $Q_W(1-p)$ of the parent distribution W which is unobserved.

Usage

```
trQuant(data, r = 1, rough = TRUE, gamma, DT, p, plot = FALSE, add = FALSE,
        main = "Estimates of extreme quantile", ...)

trQuantW(data, gamma, DT, p, plot = FALSE, add = FALSE,
         main = "Estimates of extreme quantile", ...)
```

Arguments

data	Vector of n observations (truncated data).
r	Trimming parameter, default is 1 (no trimming).
rough	Logical indicating if rough truncation is present, default is TRUE.
gamma	Vector of $n-1$ estimates for the EVI obtained from trHill.
DT	Vector of $n-1$ estimates for the truncation odds obtained from trDT.
p	The exceedance probability of the quantile (we estimate $Q(1-p)$ for p small).
plot	Logical indicating if the estimates should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "Estimates of extreme quantile".
...	Additional arguments for the plot function, see plot for more details.

Details

We observe the truncated r.v. $X =_d W | W < T$ where T is the truncation point and W the untruncated r.v.

Under rough truncation, the quantiles for X are estimated using

$$\hat{Q}(1-p) = X_{n-k,n}((\hat{D}_T + (k+1)/(n+1))/(\hat{D}_T + p))^{\hat{\gamma}_k},$$

with $\hat{\gamma}_k$ the Hill estimates adapted for truncation and $\hat{D}_T$ the estimates for the truncation odds.

Under light truncation, the quantiles are estimated using the Weissman estimator with the Hill estimates replaced by the truncated Hill estimates:

$$\hat{Q}(1-p) = X_{n-k,n}((k+1)/((n+1)p))^{\hat{\gamma}_k}.$$

To decide between light and rough truncation, one can use the test implemented in trTest.

The quantiles for W are estimated using

$$\hat{Q}_W(1-p) = X_{n-k,n}((\hat{D}_T + (k+1)/(n+1))/(p(1+\hat{D}_T))^{\hat{\gamma}_k}.$$

See Beirlant et al. (2016) or Section 4.2.3 of Albrecher et al. (2017) for more details.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
Q	Vector of the corresponding quantile estimates.
p	The used exceedance probability.

Author(s)

Tom Reynkens based on R code of Dries Cornilly.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant, J., Fraga Alves, M.I. and Gomes, M.I. (2016). "Tail fitting for Truncated and Non-truncated Pareto-type Distributions." *Extremes*, 19, 429–462.

See Also

trHill, trDT, trProb, trEndpoint, trTest, Quant, trQuantMLE

Examples

```
# Sample from truncated Pareto distribution.
# truncated at 99% quantile
shape <- 2
X <- rtpareto(n=1000, shape=shape, endpoint=qpareto(0.99, shape=shape))

# Truncated Hill estimator
trh <- trHill(X, plot=TRUE, ylim=c(0,2))

# Truncation odds
dt <- trDT(X, gamma=trh$gamma, plot=TRUE, ylim=c(0,2))

# Large quantile
```

```
p <- 10^(-5)
# Truncated distribution
trQuant(X, gamma=trh$gamma, DT=dt$DT, p=p, plot=TRUE)
# Original distribution
trQuantW(X, gamma=trh$gamma, DT=dt$DT, p=p, plot=TRUE, ylim=c(0,1000))
```

---

trQuantMLE *Estimator of large quantiles using truncated MLE*

Description

This function computes estimates of large quantiles $Q(1-p)$ of the truncated distribution using the ML estimates adapted for upper truncation. Moreover, estimates of large quantiles $Q_Y(1-p)$ of the original distribution Y, which is unobserved, are also computed.

Usage

```
trQuantMLE(data, gamma, tau, DT, p, Y = FALSE, plot = FALSE, add = FALSE,
           main = "Estimates of extreme quantile", ...)
```

Arguments

data	Vector of n observations.
gamma	Vector of $n-1$ estimates for the EVI obtained from trMLE.
tau	Vector of $n-1$ estimates for the τ obtained from trMLE.
DT	Vector of $n-1$ estimates for the truncation odds obtained from trDTMLE.
p	The exceedance probability of the quantile (we estimate $Q(1-p)$ or $Q_Y(1-p)$ for p small).
Y	Logical indicating if quantiles from the truncated distribution ($Q(1-p)$) or from the parent distribution ($Q_Y(1-p)$) are computed. Default is TRUE.
plot	Logical indicating if the estimates should be plotted as a function of k, default is FALSE.
add	Logical indicating if the estimates should be added to an existing plot, default is FALSE.
main	Title for the plot, default is "Estimates of extreme quantile".
...	Additional arguments for the plot function, see plot for more details.

Details

We observe the truncated r.v. $X =_d Y|Y < T$ where T is the truncation point and Y the untruncated r.v.

Under rough truncation, the quantiles for X are estimated using

$$\hat{Q}_{T,k}(1-p) = X_{n-k,n} + 1/(\hat{\tau}_k)([(\hat{D}_{T,k} + (k+1)/(n+1))/(\hat{D}_{T,k} + p)]^{\hat{\xi}_k} - 1),$$

with $\hat{\gamma}_k$ and $\hat{\tau}_k$ the ML estimates adapted for truncation and $\hat{D}_T$ the estimates for the truncation odds.

The quantiles for Y are estimated using

$$\hat{Q}_{Y,k}(1-p) = X_{n-k,n} + 1/(\hat{\tau}_k)([(\hat{D}_{T,k} + (k+1)/(n+1))/(p(\hat{D}_{T,k}+1))]^{\hat{\xi}_k} - 1).$$

See Beirlant et al. (2017) for more details.

Value

A list with following components:

k	Vector of the values of the tail parameter k.
Q	Vector of the corresponding quantile estimates.
p	The used exceedance probability.

Author(s)

Tom Reynkens.

References

Beirlant, J., Fraga Alves, M. I. and Reynkens, T. (2017). "Fitting Tails Affected by Truncation". *Electronic Journal of Statistics*, 11(1), 2026–2065.

See Also

trMLE, trDTMLE, trProbMLE, trEndpointMLE, trTestMLE, trQuant, Quant

Examples

```
# Sample from GPD truncated at 99% quantile
gamma <- 0.5
sigma <- 1.5
X <- rtgpd(n=250, gamma=gamma, sigma=sigma, endpoint=qgpd(0.99, gamma=gamma, sigma=sigma))

# Truncated ML estimator
trmle <- trMLE(X, plot=TRUE, ylim=c(0,2))

# Truncation odds
dtmle <- trDTMLE(X, gamma=trmle$gamma, tau=trmle$tau, plot=FALSE)

# Large quantile of X
trQuantMLE(X, gamma=trmle$gamma, tau=trmle$tau, DT=dtmle$DT, plot=TRUE, p=0.005, ylim=c(15,30))

# Large quantile of Y
trQuantMLE(X, gamma=trmle$gamma, tau=trmle$tau, DT=dtmle$DT, plot=TRUE, p=0.005, ylim=c(0,300),
           Y=TRUE)
```

`trTest` *Test for truncated Pareto-type tails*

Description

Test between non-truncated Pareto-type tails *(light truncation)* and truncated Pareto-type tails *(rough truncation)*.

Usage

```
trTest(data, alpha = 0.05, plot = TRUE, main = "Test for truncation", ...)
```

Arguments

`data`	Vector of n observations.
`alpha`	The used significance level, default is `0.05`.
`plot`	Logical indicating if the P-values should be plotted as a function of k, default is `FALSE`.
`main`	Title for the plot, default is `"Test for truncation"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

We want to test $H_0 : X$ has non-truncated Pareto tails vs. $H_1 : X$ has truncated Pareto tails. Let

$$E_{k,n}(\gamma) = 1/k \sum_{j=1}^{k} (X_{n-k,n}/X_{n-j+1,n})^{1/\gamma},$$

with $X_{i,n}$ the i-th order statistic. The test statistic is then

$$T_{k,n} = \sqrt{12k}(E_{k,n}(H_{k,n}) - 1/2)/(1 - E_{k,n}(H_{k,n}))$$

which is asymptotically standard normally distributed. We reject H_0 on level α if

$$T_{k,n} < -z_\alpha$$

where z_α is the $100(1-\alpha)\%$ quantile of a standard normal distribution. The corresponding P-value is thus given by

$$\Phi(T_{k,n})$$

with Φ the CDF of a standard normal distribution.

See Beirlant et al. (2016) or Section 4.2.3 of Albrecher et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`testVal`	Corresponding test values.
`critVal`	Critical value used for the test, i.e. `qnorm(1-alpha/2)`.
`Pval`	Corresponding P-values.
`Reject`	Logical vector indicating if the null hypothesis is rejected for a certain value of `k`.

Author(s)

Tom Reynkens.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Beirlant, J., Fraga Alves, M.I. and Gomes, M.I. (2016). "Tail fitting for Truncated and Non-truncated Pareto-type Distributions." *Extremes*, 19, 429–462.

See Also

`trHill`, `trTestMLE`

Examples

```
# Sample from truncated Pareto distribution.
# truncated at 95% quantile
shape <- 2
X <- rtpareto(n=1000, shape=shape, endpoint=qpareto(0.95, shape=shape))

# Test for truncation
trTest(X)


# Sample from truncated Pareto distribution.
# truncated at 99% quantile
shape <- 2
X <- rtpareto(n=1000, shape=shape, endpoint=qpareto(0.99, shape=shape))

# Test for truncation
trTest(X)
```

`trTestMLE` *Test for truncated GPD tails*

Description

Test between non-truncated GPD tails *(light truncation)* and truncated GPD tails *(rough truncation)*.

Usage

```
trTestMLE(data, gamma, tau, alpha = 0.05, plot = TRUE, main = "Test for truncation", ...)
```

Arguments

`data`	Vector of n observations.
`gamma`	Vector of $n-1$ estimates for the EVI obtained from `trMLE`.
`tau`	Vector of $n-1$ estimates for the τ obtained from `trMLE`.
`alpha`	The used significance level, default is `0.05`.
`plot`	Logical indicating if the P-values should be plotted as a function of k, default is `FALSE`.
`main`	Title for the plot, default is `"Test for truncation"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

We want to test $H_0 : X$ has non-truncated GPD tails vs. $H_1 : X$ has truncated GPD tails. Let $\hat{\gamma}_k$ and $\hat{\tau}_k$ be the truncated MLE estimates for γ and τ. The test statistic is then

$$T_{k,n} = k(1 + \hat{\tau}(X_{n,n} - X_{-k,n}))^{-1/\hat{\xi}_k}$$

which is asymptotically standard exponentially distributed. We reject H_0 on level α if $T_{k,n} > \ln(1/\alpha)$. The corresponding P-value is given by $\exp(-T_{k,n})$.

See Beirlant et al. (2017) for more details.

Value

A list with following components:

`k`	Vector of the values of the tail parameter k.
`testVal`	Corresponding test values.
`critVal`	Critical value used for the test, i.e. $\ln(1/\alpha)$.
`Pval`	Corresponding P-values.
`Reject`	Logical vector indicating if the null hypothesis is rejected for a certain value of `k`.

Author(s)

Tom Reynkens.

References

Beirlant, J., Fraga Alves, M. I. and Reynkens, T. (2017). "Fitting Tails Affected by Truncation". *Electronic Journal of Statistics*, 11(1), 2026–2065.

See Also

trMLE, trDTMLE, trProbMLE, trEndpointMLE, trTestMLE, trTest

Examples

```
# Sample from GPD truncated at 99% quantile
gamma <- 0.5
sigma <- 1.5
X <- rtgpd(n=250, gamma=gamma, sigma=sigma, endpoint=qgpd(0.99, gamma=gamma, sigma=sigma))

# Truncated ML estimator
trmle <- trMLE(X, plot=TRUE, ylim=c(0,2))

# Test for truncation
trTestMLE(X, gamma=trmle$gamma, tau=trmle$tau)
```

Turnbull *Turnbull estimator*

Description

Computes the Turnbull estimator for the survival function of interval censored data.

Usage

```
Turnbull(x, L, R, censored, trunclower = 0, truncupper = Inf,
         conf.type = "plain", conf.int = 0.95)
```

Arguments

x	Vector with points to evaluate the estimator in.
L	Vector of length n with the lower boundaries of the intervals.
R	Vector of length n with the upper boundaries of the intervals.
censored	Vector of n logicals indicating if an observation is interval censored.
trunclower	Lower truncation point, default is 0.
truncupper	Upper truncation point, default is Inf.
conf.type	Type of confidence interval, see survfit.formula. Default is "plain".
conf.int	Confidence level of the two-sided confidence interval, see survfit.formula. Default is 0.95.

Details

We consider the random interval censoring model where one observes $L \leq R$ and where the variable of interest X lies between L and R.

Right censored data should be entered as `L=l` and `R=truncupper`, and right censored data should be entered as `L=trunclower` and `R=r`.

This function calls `survfit.formula` from **survival**.

See Section 4.3.2 in Albrecher et al. (2017) for more details.

Value

A list with following components:

`surv`	A vector of length `length(x)` containing the Turnbull estimator evaluated in the elements of `x`.
`fit`	The output from the call to `survfit.formula`, an object of class `survfit`.

Author(s)

Tom Reynkens

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Turnbull, B. W. (1974). "Nonparametric Estimation of a Survivorship Function with Doubly Censored Data." *Journal of the American Statistical Association*, 69, 169–173.

Turnbull, B. W. (1976). "The Empirical Distribution Function with Arbitrarily Grouped, Censored and Truncated Data." *Journal of the Royal Statistical Society: Series B (Methodological)*, 38, 290–295.

See Also

`survfit.formula`, `KaplanMeier`

Examples

```
L <- 1:10
R <- c(1, 2.5, 3, 4, 5.5, 6, 7.5, 8.25, 9, 10.5)
censored <- c(0, 1, 0, 0, 1, 0, 1, 1, 0, 1)

x <- seq(0, 12, 0.1)

# Turnbull estimator
plot(x, Turnbull(x, L, R, censored)$cdf, type="s", ylab="Turnbull estimator")
```

tWeibull *The truncated Weibull distribution*

Description

Density, distribution function, quantile function and random generation for the truncated Weibull distribution.

Usage

```
dtweibull(x, shape, scale = 1, endpoint = Inf, log = FALSE)
ptweibull(x, shape, scale = 1, endpoint = Inf, lower.tail = TRUE, log.p = FALSE)
qtweibull(p, shape, scale = 1, endpoint = Inf, lower.tail = TRUE, log.p = FALSE)
rtweibull(n, shape, scale = 1, endpoint = Inf)
```

Arguments

x	Vector of quantiles.
p	Vector of probabilities.
n	Number of observations.
shape	The shape parameter of the Weibull distribution, a strictly positive number.
scale	The scale parameter of the Weibull distribution, a strictly positive number, default is 1.
endpoint	Endpoint of the truncated Weibull distribution. The default value is `Inf` for which the truncated Weibull distribution corresponds to the ordinary Weibull distribution.
log	Logical indicating if the densities are given as $\log(f)$, default is `FALSE`.
lower.tail	Logical indicating if the probabilities are of the form $P(X \leq x)$ (`TRUE`) or $P(X > x)$ (`FALSE`). Default is `TRUE`.
log.p	Logical indicating if the probabilities are given as $\log(p)$, default is `FALSE`.

Details

The Cumulative Distribution Function (CDF) of the truncated Weibull distribution is equal to $F_T(x) = F(x)/F(T)$ for $x \leq T$ where F is the CDF of the ordinary Weibull distribution and T is the endpoint (truncation point) of the truncated Weibull distribution.

Value

`dtweibull` gives the density function evaluated in x, `ptweibull` the CDF evaluated in x and `qtweibull` the quantile function evaluated in p. The length of the result is equal to the length of x or p.

`rtweibull` returns a random sample of length n.

Author(s)

Tom Reynkens.

See Also

Weibull, Distributions

Examples

```
# Plot of the PDF
x <- seq(0, 10, 0.01)
plot(x, dtweibull(x, shape=2, scale=0.5, endpoint=1), xlab="x", ylab="PDF", type="l")

# Plot of the CDF
x <- seq(0, 10, 0.01)
plot(x, ptweibull(x, shape=2, scale=0.5, endpoint=1), xlab="x", ylab="CDF", type="l")
```

VaR — *VaR of splicing fit*

Description

Compute Value-at-Risk ($VaR_{1-p} = Q(1-p)$) of the fitted spliced distribution.

Usage

```
VaR(p, splicefit)
```

Arguments

p	The exceedance probability (we estimate $VaR_{1-p} = Q(1-p)$).
splicefit	A SpliceFit object, e.g. output from SpliceFitPareto, SpliceFiticPareto or SpliceFitGPD.

Details

See Reynkens et al. (2017) and Section 4.6 of Albrecher et al. (2017) for details.

Note that `VaR(p, splicefit)` corresponds to `qSplice(p, splicefit, lower.tail = FALSE)`.

Value

Vector of quantiles $VaR_{1-p} = Q(1-p)$.

Author(s)

Tom Reynkens with `R` code from Roel Verbelen for the mixed Erlang quantiles.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

Reynkens, T., Verbelen, R., Beirlant, J. and Antonio, K. (2017). "Modelling Censored Losses Using Splicing: a Global Fit Strategy With Mixed Erlang and Extreme Value Distributions". *Insurance: Mathematics and Economics*, 77, 65–77.

Verbelen, R., Gong, L., Antonio, K., Badescu, A. and Lin, S. (2015). "Fitting Mixtures of Erlangs to Censored and Truncated Data Using the EM Algorithm." *Astin Bulletin*, 45, 729–758

See Also

qSplice, CTE, SpliceFit, SpliceFitPareto, SpliceFiticPareto, SpliceFitGPD

Examples

```
## Not run:

# Pareto random sample
X <- rpareto(1000, shape = 2)

# Splice ME and Pareto
splicefit <- SpliceFitPareto(X, 0.6)



p <- seq(0,1,0.01)

# Plot of quantiles
plot(p, qSplice(p, splicefit), type="l", xlab="p", ylab="Q(p)")

# Plot of VaR
plot(p, VaR(p, splicefit), type="l", xlab="p", ylab=bquote(VaR[1-p]))

## End(Not run)
```

WeibullQQ *Weibull quantile plot*

Description

Computes the empirical quantiles of the log-transform of a data vector and the theoretical quantiles of the standard Weibull distribution. These quantiles are then plotted in a Weibull QQ-plot with the theoretical quantiles on the x-axis and the empirical quantiles on the y-axis.

Usage

```
WeibullQQ(data, plot = TRUE, main = "Weibull QQ-plot", ...)
```

Arguments

`data`	Vector of n observations.
`plot`	Logical indicating if the quantiles should be plotted in a Weibull QQ-plot, default is `TRUE`.
`main`	Title for the plot, default is `"Weibull QQ-plot"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The Weibull QQ-plot is given by

$$(\log(-\log(1 - i/(n+1))), \log X_{i,n})$$

for $i = 1, ..., n$, with $X_{i,n}$ the i-th order statistic of the data.

See Section 4.1 of Albrecher et al. (2017) for more details.

Value

A list with following components:

`wqq.the`	Vector of the theoretical quantiles from a standard Weibull distribution.
`wqq.emp`	Vector of the empirical quantiles from the log-transformed data.

Author(s)

Tom Reynkens.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

See Also

`WeibullQQ_der`, `ExpQQ`, `LognormalQQ`, `ParetoQQ`

Examples

```
data(norwegianfire)

# Weibull QQ-plot for Norwegian Fire Insurance data for claims in 1976.
WeibullQQ(norwegianfire$size[norwegianfire$year==76])

# Derivative of Weibull QQ-plot for Norwegian Fire Insurance data for claims in 1976.
WeibullQQ_der(norwegianfire$size[norwegianfire$year==76])
```

`WeibullQQ_der` *Derivative plot of the Weibull QQ-plot*

Description

Computes the derivative plot of the Weibull QQ-plot. These values can be plotted as a function of the data or as a function of the tail parameter k.

Usage

```
WeibullQQ_der(data, k = FALSE, plot = TRUE,
              main = "Derivative plot of Weibull QQ-plot", ...)
```

Arguments

`data`	Vector of n observations.
`plot`	Logical indicating if the derivative values should be plotted, default is `TRUE`.
`k`	Logical indicating if the derivative values are plotted as a function of the tail parameter k (`k=TRUE`) or as a function of the logarithm of the data (`k=FALSE`). Default is `FALSE`.
`main`	Title for the plot, default is `"Derivative plot of Weibull QQ-plot"`.
`...`	Additional arguments for the `plot` function, see `plot` for more details.

Details

The derivative plot of a Weibull QQ-plot is

$$(k, H_{k,n}/W_{k,n})$$

or

$$(\log X_{n-k,n}, H_{k,n}/W_{k,n})$$

with $H_{k,n}$ the Hill estimates and

$$W_{k,n} = 1/k \sum_{j=1}^{k} \log(\log((n+1)/j)) - \log(\log((n+1)/(k+1))).$$

See Section 4.1 of Albrecher et al. (2017) for more details.

Value

A list with following components:

`xval`	Vector of the x-values of the plot (k or $\log X_{n-k,n}$).
`yval`	Vector of the derivative values.

Author(s)

Tom Reynkens.

References

Albrecher, H., Beirlant, J. and Teugels, J. (2017). *Reinsurance: Actuarial and Statistical Aspects*, Wiley, Chichester.

See Also

WeibullQQ, Hill, MeanExcess, LognormalQQ_der, ParetoQQ_der

Examples

```
data(norwegianfire)

# Weibull QQ-plot for Norwegian Fire Insurance data for claims in 1976.
WeibullQQ(norwegianfire$size[norwegianfire$year==76])

# Derivative of Weibull QQ-plot for Norwegian Fire Insurance data for claims in 1976.
WeibullQQ_der(norwegianfire$size[norwegianfire$year==76])
```

Index

∗ **classes**
EVTfit, 47
MEfit, 85
SpliceFit, 131
∗ **datasets**
norwegianfire, 88
secura, 125
soa, 126
∗ **distribution**
Burr, 4
Extended Pareto, 56
Frechet, 57
GPD, 62
Pareto, 89
Splice, 127
tBurr, 158
tExp, 160
tFrechet, 161
tGPD, 162
tlnorm, 164
tPareto, 165
tWeibull, 190
∗ **htest**
trTest, 185

Burr, 4, *159*

cEPD, 6, *12*, *20, 21*
cExpQQ, 7, *15*, *18*, *43*
cgenHill, 9, *23, 24*, *30, 31*
cGPDmle, *7*, 10, *25, 26*, *32, 33*
cHill, *10*, 12, *18, 19*, *28, 29*, *36*, *73*
cLognormalQQ, *8*, 14, *18*, *43*
cMoment, 15, *27, 28*, *34*
cParetoQQ, *8*, *13*, *15*, 17, *38*, *43*, *75*
cPOT *(*cGPDmle*)*, 10
cProb, *13*, 18, *29*
cProbEPD, *7*, 20
cProbGH, *10*, 22, *31*
cProbGPD, *12*, 24, *33*
cProbMOM, *16*, *24*, 26, *34*
cQuant, *13*, *19*, 28
cQuantGH, *10*, *24*, 30
cQuantGPD, *12*, *26*, 32
cQuantMOM, *16*, *28*, 33
cReturn *(*cProb*)*, 18
cReturnEPD *(*cProbEPD*)*, 20
cReturnGH *(*cProbGH*)*, 22
cReturnGPD *(*cProbGPD*)*, 24
cReturnMOM *(*cProbMOM*)*, 26
crHill, 35, *38*, *40*
crParetoQQ, *36*, 37, *40*
crSurv, *36–38*, 39
CTE, 41, *192*
cWeibullQQ, *8*, *15*, *18*, 42

dburr *(*Burr*)*, 4
depd *(*Extended Pareto*)*, 56
detectCores, *133*, *135*, *138*
dfrechet *(*Frechet*)*, 57
dgpd *(*GPD*)*, 62
Distributions, *5*, *57, 58*, *63*, *90*, *159*, *161–163*, *165, 166*, *191*
dpareto *(*Pareto*)*, 89
dSplice *(*Splice*)*, 127
dtburr *(*tBurr*)*, 158
dtexp *(*tExp*)*, 160
dtfrechet *(*tFrechet*)*, 161
dtgpd *(*tGPD*)*, 162
dtlnorm *(*tlnorm*)*, 164
dtpareto *(*tPareto*)*, 165
dtweibull *(*tWeibull*)*, 190

ecdf, *130*, *141*, *146*
EPD, *7*, 44, *46*, *48, 49*, *63*, *66*, *101, 102*, *123*
EPDfit, 46, *64*
ES *(*CTE*)*, 41
EVTfit, 47, *131, 132*
ExcessEPD, 48, *51*, *53*
ExcessGPD, *49*, 50, *53*

ExcessHill, *49*, *51*
ExcessHill *(*ExcessPareto*)*, 51
ExcessPareto, 51
ExcessSplice, *42*, *50*, *52*, *53*, 53
Exponential, *161*
ExpQQ, *8*, 55, *67*, *78*, *83*, *91*, *193*
Extended Pareto, 56

Frechet, 57
Fréchet, *162*

generalizedQQ *(*genQQ*)*, 60
genHill, *10*, 59, *68*, *88*, *103*, *104*, *113*, *114*
genQQ, *60*, 60, *81*, *91*
GPD, *57*, 62, *90*
GPDfit, *46*, 63, *66*, *67*, *134*
GPDmle, *12*, *45*, *50*, *51*, *64*, 64, *105*, *106*, *114*, *115*, *175*
GPDresiduals, *66*, 66

Hill, *13*, *52*, *53*, *60*, *61*, 67, *71*, *73*, *80*, *88*, *93*, *120*, *125*, *139*, *174*, *195*
Hill.2oQV, *68*, 69, *71*, *111*, *112*, *121*
Hill.kopt, 70

icfit, *73*, *75*, *84*, *85*, *143*, *144*, *148*, *149*, *153*
icHill, *13*, 72, *75*
icParetoQQ, *17*, *18*, *73*, 74

KaplanMeier, *8*, *15*, *18*, *19*, *21*, *24*, *26*, *28*, *29*, *31*, *33*, *34*, *43*, 76, *189*

Lognormal, *165*
LognormalQQ, *15*, *56*, 77, *80*, *91*, *193*
LognormalQQ_der, 79, *83*, *93*, *195*
LStail, 80

MeanExcess, *56*, *80*, 82, *84*, *85*, *93*, *195*
MeanExcess_TB, *73*, 83
MEfit, 85, *131*, *132*
Moment, *16*, *60*, 87, *106*, *107*, *116*, *117*

norwegianfire, 88

optim, *175*

Pareto, *57*, *63*, 89, *163*, *166*
ParetoQQ, *18*, *56*, *61*, *68*, *75*, *78*, 90, *93*, *177*, *193*
ParetoQQ_der, *80*, *83*, *91*, 92, *195*

pburr *(*Burr*)*, 4
pClas, 93, *98*
pEdge, *94*, 95, *96*, *98*
pepd *(*Extended Pareto*)*, 56
pfrechet *(*Frechet*)*, 57
pGC, *94*, *96*, 97
pgpd *(*GPD*)*, 62
plot, *6*, *8*, *9*, *11*, *13*, *14*, *16*, *17*, *19*, *21*, *23*, *25*, *27*, *29*, *30*, *32*, *34*, *36*, *38*, *43*, *44*, *48*, *50*, *52*, *55*, *59*, *61*, *65*, *66*, *68*, *69*, *71*, *72*, *75*, *78*, *79*, *81*, *82*, *84*, *87*, *91*, *92*, *100*, *101*, *103*, *105*, *107*, *108*, *110*, *112*, *113*, *115*, *116*, *118*, *119*, *121*, *122*, *124*, *129*, *140*, *143*, *145*, *148*, *150*, *152*, *155*, *167*, *168*, *170*, *171*, *173*, *174*, *176*, *178*, *180*, *181*, *183*, *185*, *187*, *193*, *194*
POT, *11*
POT *(*GPDmle*)*, 64
ppareto *(*Pareto*)*, 89
Prob, *19*, *21*, *26*, *28*, 99, *102*, *104*, *106*, *107*, *109*, *110*, *179*, *180*
ProbEPD, *21*, *45*, 101
ProbGH, *24*, 103, *107*, *114*
ProbGPD, *26*, 104, *115*
ProbMOM, *28*, *104*, 106, *117*
ProbReg, 108, *118*, *125*
pSplice, *130*, *141*, *144*, *146*, *149*, *155*
pSplice *(*Splice*)*, 127
ptburr *(*tBurr*)*, 158
ptexp *(*tExp*)*, 160
ptfrechet *(*tFrechet*)*, 161
ptgpd *(*tGPD*)*, 162
ptlnorm *(*tlnorm*)*, 164
ptpareto *(*tPareto*)*, 165
ptweibull *(*tWeibull*)*, 190

qburr *(*Burr*)*, 4
qepd *(*Extended Pareto*)*, 56
qfrechet *(*Frechet*)*, 57
qgpd *(*GPD*)*, 62
qpareto *(*Pareto*)*, 89
qSplice, *42*, *151*, *153*, *192*
qSplice *(*Splice*)*, 127
qtburr *(*tBurr*)*, 158
qtexp *(*tExp*)*, 160
qtfrechet *(*tFrechet*)*, 161
qtgpd *(*tGPD*)*, 162
qtlnorm *(*tlnorm*)*, 164

qtpareto *(tPareto)*, 165
qtweibull *(tWeibull)*, 190
Quant, *29*, *31*, *33, 34*, *100*, 109, *112*, *114, 115*, *117, 118*, *182*, *184*
Quant.2oQV, *110*, 111
QuantGH, *31*, *104*, 113, *117*
QuantGPD, *33*, *106*, 114
QuantMOM, *34*, *107*, *114*, 116
QuantReg, *109*, 117, *125*

rburr *(Burr)*, 4
repd *(Extended Pareto)*, 56
Return *(Prob)*, 99
ReturnEPD *(ProbEPD)*, 101
ReturnGH *(ProbGH)*, 103
ReturnGPD *(ProbGPD)*, 104
ReturnMOM *(ProbMOM)*, 106
rfrechet *(Frechet)*, 57
rgpd *(GPD)*, 62
rpareto *(Pareto)*, 89
rSplice *(Splice)*, 127
rtburr *(tBurr)*, 158
rtexp *(tExp)*, 160
rtfrechet *(tFrechet)*, 161
rtgpd *(tGPD)*, 162
rtlnorm *(tlnorm)*, 164
rtpareto *(tPareto)*, 165
rtweibull *(tWeibull)*, 190

Scale, 119, *121*, *123*
scale, *125*
Scale.2o, *120*, 120, *123*
ScaleEPD, *120, 121*, 122
ScaleReg, *108, 109*, *117, 118*, 124
secura, 125
soa, 126
Splice, 127, *134*, *136*, *139*
SpliceECDF, *128*, 129, *141*, *146*, *151*, *155*
SpliceFit, *41, 42*, *48*, *53, 54*, *86*, *127–129*, 131, *133*, *136*, *139, 140*, *143*, *145*, *148*, *150*, *152, 153*, *155*, *191, 192*
SpliceFitGPD, *41, 42*, *48*, *53, 54*, *86*, *127–132*, 132, *136*, *139–141*, *145*, *146*, *150, 151*, *191, 192*
SpliceFitHill *(SpliceFitPareto)*, 137
SpliceFiticPareto, *41, 42*, *48*, *53, 54*, *86*, *127, 128*, *131, 132*, *134*, 135, *139*, *143, 144*, *148, 149*, *152, 153*, *155*, *191, 192*
SpliceFitPareto, *41, 42*, *48*, *53, 54*, *86*, *127–132*, *134*, *136*, 137, *140, 141*, *145, 146*, *150, 151*, *191, 192*
SpliceLL, *128*, *130*, 140, *143, 144*, *146*, *151*
SpliceLL_TB, *141*, 142, *149*, *153*, *155*
SplicePP, *128*, *130*, *141*, 145, *148, 149*, *151*
SplicePP_TB, *144*, *146*, 147, *153*, *155*
SpliceQQ, *130*, *141*, *146*, 150, *153*
SpliceQQ_TB, *144*, *149*, *151*, 152, *155*
SpliceTB, *129, 130*, *144*, *149*, *153*, 154
stdf, 157
stdf2 *(stdf)*, 157
survfit.formula, *73*, *75–77*, *84*, *143*, *148*, *153*, *188, 189*

tBurr, *5*, 158
tExp, 160
tFrechet, 161
tFréchet, *58*
tGPD, *63*, 162, *163*
tlnorm, 164
tPareto, *90*, 165
trDT, 166, *169, 170*, *174*, *176, 177*, *181, 182*
trDTMLE, *167*, 168, *172*, *175*, *179, 180*, *183, 184*, *188*
trEndpoint, *167*, 169, *172*, *174*, *182*
trEndpointMLE, *169, 170*, 171, *175*, *180*, *184*, *188*
trHill, *52*, *139*, *166, 167*, *170*, 172, *175*, *178, 179*, *181, 182*, *186*
trMLE, *168, 169*, *171, 172*, *174*, 174, *179, 180*, *183, 184*, *187, 188*
trParetoQQ, 176
trProb, *174*, 178, *180*, *182*
trProbMLE, *169*, *172*, *175*, *179*, 179, *184*, *188*
trQuant, *167*, *174*, *179*, 181, *184*
trQuantMLE, *169*, *172*, *175*, *180*, *182*, 183
trQuantW *(trQuant)*, 181
trTest, *182*, 185, *188*
trTestMLE, *169*, *172*, *175*, *180*, *184*, *186*, 187, *188*
TSfraction *(LStail)*, 80
Turnbull, *73*, *75–77*, *85*, *144*, *149*, *153*, *155*, 188
tWeibull, 190

VaR, *128*, 191

Weibull, *191*

WeibullQQ, *43*, *56*, *78*, *91*, 192, *195*
WeibullQQ_der, *80*, *83*, *93*, *193*, 194
Weissman.p *(*Prob*)*, 99
Weissman.q *(*Quant*)*, 109
Weissman.q.2oQV *(*Quant.2oQV*)*, 111
Weissman.r *(*Prob*)*, 99