Package ‘AoE’

April 11, 2008

Type Package

Title Analysis of Extremes

Version 1.0.1

Date 2008-04-11

Author Johan Segers

Maintainer Johan Segers <johan.segers@uclouvain.be>

Description Provides functions for univariate and multivariate extreme value analysis. Written as a complement to executive courses for the Dutch and Belgian Actuarial Associations.

R topics documented:

ABNING *Log-returns ING and ABN AMRO*

Description

Daily log-returns of stock prices ING and ABN AMRO for the period 1991-2003.

Usage

```
data(ABN)
data(ING)
```

Format

Numeric vectors of length 3283.

Source

Casper de Vries (Erasmus University Rotterdam), private communication.

Examples

```
data(ABN, ING)
plot(ABN, ING)
```

AngularMeasure *Angular Measure*

Description

Computes an estimate of the Pickands dependence function of the extreme-value attractor of a bivariate distribution based on a bivariate sample $(X_1, Y_1), \ldots, (X_n, Y_n)$ from that distribution.

Usage

```
AngularMeasure(data.x, data.y, data = NULL, k,
          method = "u", plot = TRUE)
```

Arguments

`data.x, data.y`
: Numeric vectors containing the data $X_1, \dots, X_n$ and $Y_1, \dots, Y_n$, respectively.

`data`
: Alternatively, the data may be provided in the form of a n-by-2 matrix. If provided, then the arguments `data.x` and `data.y` are ignored.

`k`
: An numeric vector of values for k in the definition of the empirical tail dependence function; see 'Details'.

`method`
: A character vector specifying the estimation method; possible choices are `"u"` for *unconstrained* and `"c"` for *constrained*. See 'Details'.

`plot`
: If `TRUE` (the default), the estimated distribution functions will be plotted.

Details

This function is an implementation of the following nonparametric estimator for the angular or spectral measure Φ (de Haan and Resnick, 1977) of the extreme-value attractor of an unknown distribution. For data $(X_1, Y_1), \dots, (X_n, Y_n)$, let R_i be the rank of X_i among $X_1, \dots, X_n$ and let S_i be the rank of Y_i among $Y_1, \dots, Y_n$. Define $X_i^* = n/(n + 1 - R_i)$ and $Y_i^* = n/(n + 1 - S_i)$. Write $(X_i^*, Y_i^*) = (\rho_i \cos\theta_i, \rho_i \sin\theta_i)$ in polecoordinates. For $0 < k < n$, let J be the set of $i = 1, \dots, n$ such that $\rho_i > n/k$. Then the estimate $\hat{\Phi}$ is the discrete measure with an atom of mass p_i at θ_i for all i in J. The masses or weights p_i depend on the method:

`method = "u"` ***unconstrained*** Then $p_i = 1/k$ for every i in J. This is called the empirical spectral measure and is a variant of the estimator considered for instance in Einmahl et al. (2001).

`method = "c"` ***constrained*** Then the weights p_i are determined by a variant of maximum empirical likelihood taking into account the moment constraints that a spectral measure should satisfy (Einmahl and Segers 2007).

The argument k may be a vector, in which case, provided `plot = TRUE`, the corresponding distribution function $\hat{\Phi}([0, \theta])$ will be drawn for every element of `k`. However, the value returned by the function corresponds only to the *final* element of `k`.

Value

A list with the `class` attribute `"AngularMeasure"`, which is a list containing the following components:

`angles`
: The angles θ_i for i in J.

`weights`
: The corresponding weights p_i.

`radii`
: The full vector of radii ρ_i for $i = 1, \dots, n$.

`indices`
: The set J.

References

Einmahl, J.H.J., de Haan, L. and Piterbarg, V.I. (2001). Nonparametric estimation of the spectral measure of an extreme value distribution. The Annals of Statistics 29, 1401-1423.

Einmahl, J.H.J. and Segers, J. (2007). Maximum empirical likelihood estimation of the spectral measure of an extreme value distribution. In preparation.

de Haan, L. and Resnick, S.I. (1977). Limit theory for multivariate sample extremes. Zeitschrift fuer Wahrscheinlichkeitstheorie und Verwandte Gebiete 40, 317-337.

See Also

ETDF, PickandsDF

Examples

```
# For the bivariate Cauchy distribution on the positive quadrant,
# the angular measure is known to be Phi([0, theta]) = theta.
AngularMeasure(data = rbivcauchy(1000), k = c(20, 50), method = "c")
abline(a = 0, b = 1, col = "red")
```

AoE-package *Analysis of Extremes*

Description

The package provides functions for some selected procedures in univariate and multivariate extreme value analysis. It has grown over a number of years as a complement to executive courses for the Dutch and Belgian Actuarial Associations. Its aim is mainly pedagogical, and no aim whatsoever is made to provide a comprehensive toolset for extreme value analysis.

Details

Package:	AoE
Type:	Package
Version:	1.0.1
Date:	2008-04-11
License:	Gnu General Public License version 2

The functions can be divided into a number of different categories; see below. Moreover, the package provides some data-sets as well.

Diagnostic plots
Hill.diagnostic
MEplot

Estimators of tail parameters
GPD_par
Hill
ML
Moment

Parametric distribution fitting
fitGPD
fitPareto

Estimators of tail-related risk measures
Burr.empirical
Burr.Weissman
EconomicCapital
ExcessLoss
Expectation
ExpectedShortfall
PHtransform

RiskMeasure
TailQuantile
Variance
Weissman.q

Threshold selection

ChooseK
TQ_ChooseK

Bivariate tail dependence

AngularMeasure
ETDF
PickandsDF
TailProb_sum
TailQuantile_sum

Temporal dependence of extremes

ExtremalIndex

Data-sets

ABN, ING
Loss, ALAE
norwegian
soa
top40

Random number generation

rbivcauchy
rbivnorm
rburr
UvT_Cat

Author(s)

Johan Segers ⟨johan.segers@uclouvain.be⟩, gratefully acknowledging valuable input and patient bug checking from John H.J. Einmahl (Tilburg University) as well as stimulating comments from course participants of the Actuarieel Instituut (the Netherlands) and the ARAB-KVBA (Belgium).

References

Beirlant, J., Goegebeur, Y., Segers, J., and Teugels, J. (2004). Statistics of Extremes. Wiley, Chichester. http://lstat.kuleuven.be/Wiley/index.html

Embrechts, P., Klueppelberg, C., and Mikosch, T. (1997). Modelling Extremal Events For Insurance And Finance. Springer.

de Haan, L. and Ferreira, A. (2006). Extreme Value Theory: An Introduction. Springer.

Examples

```
# Hill estimator
x <- 1/runif(100)
Hill(1/runif(100), CI.p = 0.95)
abline(h = 1, col = "blue")

# tail quantile and excess-of-loss net premium
data(Loss)
TailQuantile(Loss, p = 0.001, k = 25:200)
ExcessLoss(Loss, a = 1.2e6, k = 25:200)
```

```
# empirical tail dependence function
data(Loss, ALAE)
ETDF(data.x = Loss, data.y = ALAE, k = 10:100)

# angular measure
AngularMeasure(data = rbivcauchy(1000), k = c(20, 50), method = "c")
abline(a = 0, b = 1, col = "red")

# extremal index
data(ABN)
ExtremalIndex(-ABN[2000:2500], threshold = 0.05, plot = TRUE)
```

`BurrTailQuantile` *Tail Quantile Estimation for the Burr Distribution*

Description

The functions implement a small simulation study in order to assess the performance of tail quantile estimators based on random samples of the Burr distribution. The implemented estimators are the sample quantile and the Weissman estimator based on the moment estimator for the extreme-value index.

Usage

```
Burr.empirical(beta = 1, tau = 1, lambda = 1, n = 1000, p = 1/n,
        samples = 500, plot = TRUE)
Burr.Weissman(beta = 1, tau = 1, lambda = 1, n = 1000, p = 1/n,
        samples = 500, k = 20, plot = TRUE)
```

Arguments

`beta, tau, lambda`	Parameters of the Burr distribution. See 'Details'.
`n`	Sample size.
`p`	Tail probability of the quantile to be estimated.
`samples`	Number of samples.
`k`	Determines the thresholds at which the Weissman estimator will be computed.
`plot`	If `TRUE`, the results will be plotted.

Details

The *Burr distribution* is defined here by its distribution function

$$F(x) = 1 - \left(\frac{\beta}{\beta + x^\tau}\right)^\lambda$$

for $x \geq 0$, with shape parameters $\lambda, \tau > 0$ and scale parameter $\beta > 0$. The distribution is heavy-tailed with extreme-value index $\gamma = 1/(\tau\lambda)$. The quantile with excess probability $0 < p \leq 1$ is given by

$$Q(1-p) = \{\beta(p^{-1/\lambda} - 1)\}^{1/\tau}$$

Interest is in estimating this tail quantile for small p, say of the order $O(1/n)$, with n the sample size.

Let $X_{1:n} < \ldots < X_{n:n}$ be the ascending order statistics of the sample. The aim of the functions `Burr.empirical` and `Burr.Weissman` is to compare the performance of the following two estimators of $Q(1-p)$:

1. The empirical tail quantile $X_{i:n}$ with i equal to $n(1-p)$ rounded up.
2. The *Weissman* estimate $X_{n-k:n}\{k/(np)\}^{\hat{\gamma}}$, where $k = 1, \ldots, n-1$ is such that $X_{n-k:n} > 0$, and with $\hat{\gamma}$ an estimator of the extreme-value index γ, assumed to be positive. The estimator implemented here is the `Moment` estimator of Dekkers et al. (1989).

If `plot = TRUE`, the function `Burr.empirical` produces a kernel density estimate of the sampling distribution of the empirical quantile estimator. For `Burr.Weissman`, the produced plot depends on whether `k` is a single number or a vector:

- If `k` is a single number, the plot shows a kernel density estimate of the sampling distribution of the Weissman quantile estimator.
- If `k` is a vector, the plot shows the estimated 5/50/95 percentiles of the sampling distribution of the Weissman quantile estimator as a function of k. For comparison, the corresponding percentiles for the empirical quantile estimator are shown as well.

The use of the functions is mainly pedagogical. The following points stand out:

1. Purely nonparametric estimation of tail quantiles is *not* a good idea, especially not for heavy-tailed distributions, for which tail quantiles lie “far apart”.
2. Extreme value theory provides estimators which work reasonably well under very general assumptions, even for out-of-sample quantiles.
3. The performance of extreme-value estimators depends on the choice of the threshold. Their sampling variance decreases but their bias increases as the number of upper order statistics used increases.
4. The estimation uncertainty for these type of problems is quite large. Indeed, if one is not willing to rely on a parametric model, then one cannot reasonably expect a precise estimate in a region where there are no data.

Value

The functions are called mainly for their side-effect, which is to produce the plots described above. The function `Burr.empirical` silently returns a list with two components: `Q`, the true quantile, and `Quantile.empirical`, a vector of length `samples` with the estimates. The list `Burr.Weissman` silently returns a list with three components: the ones already mentioned and the additional component `Quantile.Weissman`, a matrix with at position (i, j) the Weissman quantile estimate for sample i and for k equal to `k[j]`.

References

Dekkers, A.L.M., Einmahl, J.H.J. and de Haan, L. (1989). A moment estimator for the index of an extreme-value distribution. The Annals of Statistics 17, 1833-1855.

See Also

`Moment`, `TailQuantile`, `Weissman.q`

Examples

```
# sampling distribution of empirical quantile estimator
Burr.empirical(beta = 2, tau = 2, lambda = 0.8,
        n = 1000, p = 0.001, samples = 500)

# sampling distribution of the
# Weissman quantile estimator
# based on the moment estimator
# for the extreme-value index
Burr.Weissman(beta = 2, tau = 2, lambda = 0.8,
        n = 1000, p = 0.001, samples = 500, k = 200)

# sampling distribution of the
# Weissman quantile estimator
# as a function of the threshold
Burr.Weissman(beta = 2, tau = 2, lambda = 0.8,
        n = 1000, p = 0.001, samples = 500, k = 50:400)
```

ChooseK *Automated Threshold Selection for Univariate Tail Estimation*

Description

The function is an implementation of an experimental method by the package author for the automated threshold selection (choice of k) for univariate tail estimation.

Usage

```
ChooseK(data = x, k = 10:(length(data) - 1), test = "s", alpha = 0.5,
         approx = "GPD", method = "ML", plot = TRUE)
```

Arguments

`data`	A numeric vector containing the data.
`k`	Vector of values of $k = 1, \ldots, n-1$, with n the sample size, among which to choose.
`test`	A character string specifying the test with which the goodness-of-fit of the exponential distribution to the residuals will be tested. See 'Details'.
`alpha`	The nominal level α of the test.
`approx`	A character string specifying the model which is fitted to the tail: the `"Weissman"` approximation or the `"GPD"`. See 'Details'.
`method`	In case `approx = "GPD"`, a character string specifying the estimators for the parameters of the generalized Pareto distribution fitted to high-threshold excesses: `"Hill"`, `"ML"`, or `"Moment"`. See `Hill`.
`plot`	If `TRUE` (the default), the results will be plotted. See 'Details'.

Details

Let $X_{1:n} \leq \ldots \leq X_{n:n}$ be the ascending order statistics of the sample. The residuals $Z_{1:k} \leq \ldots \leq Z_{k:k}$ are defined as follows:

- If `approx = "Weissman"`, then $Z_{i:k} = \log X_{n-k+i:n} - \log X_{n-k:n}$. This approach is suitable only for heavy-tailed distributions, that is, with extreme-value index $\gamma > 0$.
- If `approx = "GPD"`, then $Z_{i:k} = \log\{1 + \gamma(X_{n-k+i:n} - X_{n-k:n})/\sigma\}/\sigma$, with γ and σ the estimates of the parameters of the generalized Pareto distribution.

To this sample of k residuals, a goodness-of-fit test of the exponential distribution is performed. The largest k for which the null hypothesis is not rejected at level α is the selected value for k.

The argument `test` specifies which test will be used: `"Cox-Oakes"`, `"Gini"`, `"Anderson-Darling"`, `"Cramer-von Mises"`, `"correlation"`, `"score"`. See Henze and Meintanis (2005) and Stephens (1974) for more details on all of these tests but `"score"`. The test corresponding to `"score"` is the score test for $c = 0$ in the mixture model

$$F(x) = 1 - (1 - c)\exp(-\alpha x) - c\exp(-2\alpha x)$$

Essentially this is a test for the presence for a bias term of the form predicted by the theory of second-order regular variation.

If `plot = TRUE`, then two graphs are shown:

- Left: the p-values of the goodness-of-fit test as a function of k.
- Right: an exponential quantile-quantile plot of the residuals $Z_{i:k}$ at the selected value of k.

If in the functions `Hill`, `ML` or `Moment` the argument `choose.k` is set to `TRUE`, then a value of k is selected by a call to `ChooseK`. This is the main use of this function.

Simulation experience shows that the `"score"` test works best and that α should be chosen much larger than the usual values for the type-I error, lest the selected value for k is too large. This is why the default value is `alpha = 0.5`, which seems to give good results overall. But see 'Notes'.

Value

A list with the following components:

`p`	A numeric vector of the same lengths as `k` with the p-values of the test at the corresponding threshold.
`k0, i0`	The selected value of k, specified in two ways: `k0 = k[i0]`.
`g0, s0`	The estimated parameters of the generalized Pareto distribution at the selected threshold (only if `method = "GPD"`).
`z0`	The residuals at the selected value of k.
`test`	The name of the goodness-of-fit test.
`alpha`	The nominal level of the test.

Note

This method is still experimental. No theory is existing yet. For questions or suggestions, please feel free to write to ⟨johan.segers@uclouvain.be⟩.

Author(s)

Johan Segers

References

Henze, N. and Meintanis, S.G. (2005). Recent and classical tests for exponentiality: a partial review with comparisons. Metrika 61, 29-45.

Stephens, M.A. (1974). EDF statistics for goodness of fit and some comparisons. Journal of the American Statistical Association 69, 730-737.

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

Hill, ML, Moment

Examples

```
x <- rburr(n = 1000, gamma = 0.5, rho = -0.5)
Hill(x, k = 10:500, log = "x", choose.k = TRUE)
```

ETDF *Empirical Tail Dependence Function*

Description

Computes the empirical tail dependence function based on a bivariate sample $(X_1, Y_1), \ldots, (X_n, Y_n)$.

Usage

```
ETDF(data.x, data.y, data = NULL, v = c(1, 1), k,
                              method = "empirical", plot = TRUE)
```

Arguments

`data.x, data.y`	Numeric vectors containing the data $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_n$, respectively.
`data`	Alternatively, the data may be provided in the form of a n-by-2 matrix. If provided, then the arguments `data.x` and `data.y` are ignored.
`v`	The point in which the empirical tail dependence function is to be computed.
`k`	An numeric vector of values for k in the definition of the empirical tail dependence function; see 'Details'.
`method`	The estimation method, specified by a string. Currently, this argument is ignored since only the empirical method `"empirical"` is implemented.
`plot`	If `TRUE` (the default), the result will be plotted.

Details

The *empirical tail dependence function* for a bivariate sample $(X_1, Y_1), \ldots, (X_n, Y_n)$ is defined by

$$\hat{l}(x, y) = \frac{1}{k} \sum_{i=1}^{n} I(R_i \geq n + 1 - kx \text{ or } S_i \geq n + 1 - ky)$$

where $x, y \geq 0$, where R_i and S_i are the ranks of the data, and where $0 < k < n$ is a tuning parameter. The elements of the input vector `v` correspond to the values of x and y.

The function is an estimate of the *(stable) tail dependence function*

$$l(x, y) = \lim_{s \to 0} \frac{1 - C(1 - sx, 1 - sy)}{s}$$

where C is the copula of the underlying distribution. In order for the estimator to be consistent, we need $k = k(n)$ with k to infinity and k/n to zero.

A useful special case is when $(x, y) = (1, 1)$, for $\lambda = 2 - l(1, 1)$ is the *coefficient of tail dependence*:

$$\lambda = \lim_{s \to 0} P\{F_X(X) > 1 - s \mid F_Y(Y) > 1 - s)\}$$

In particular, $l(1, 1) = 1$ corresponds to *asymptotic complete dependence*, while $l(1, 1) = 2$ corresponds to *asymptotic independence*. More generally, low (high) values of $l(1, 1)$ indicate strong (weak) tail dependence.

Another special case is when $y = 1 - x$, yielding the *Pickands dependence function*

$$A(x) = l(x, 1 - x)$$

for $0 \leq x \leq 1$.

Value

A numeric vector of length `length(k)`, the elements being the corresponding estimates of the tail dependence function at the point specified by `v`. The result is returned invisibly.

References

Drees, H. and Huang, X. (1998). Best Attainable Rates of Convergence for Estimators of the Stable Tail Dependence Function. Journal of Multivariate Analysis 64, 25-47.

See Also

`AngularMeasure`, `PickandsDF`

Examples

```
# The bivariate normal distribution
# with arbitrary correlation not equal to one
# has an asymptotically independent upper tail:
ETDF(data = rbivnorm(1e5, cor = 0.9), k = 10:100)

# The Loss-ALAE data seem to exhibit asymptotic dependence:
data(Loss, ALAE)
ETDF(data.x = Loss, data.y = ALAE, k = 10:100)
```

`ExtremalIndex` *Intervals Estimator for the Extremal Index*

Description

This function is an implementation of the *intervals estimator* of Ferro and Segers (2003).

Usage

```
ExtremalIndex(data, threshold = NULL, k = NULL, plot = TRUE)
```

Arguments

`data`	A numeric vector containing the data.
`threshold`	The threshold above which excesses will be counted.
`k`	Alternatively, the threshold may be specified as the $(k+1)$th largest order statistic, so that in the abscence of ties there are exactly k excesses. If `threshold` is provided, then `k` is ignored.
`plot`	If `TRUE` (the default), the result will be plotted.

Details

The *extremal index* $0 \leq \theta \leq 1$ of a strictly stationary time series is a measure for tendency of high-threshold excesses to appear in clusters (Leadbetter, 1988). The extremal index can be thought of as the reciprocal of the mean of the number of excesses in such a cluster. In particular, $\theta = 1$ corresponds to no clustering, that is, asymptotic independence.

For a time series $X_1, \ldots, X_n$ and a threshold u, let

$$1 \leq S_1 < \ldots < S_N \leq n$$

be the ordered collection of random time instants $t = 1, \ldots, n$ such that $X_t > u$. The inter-arrival times are defined as

$$T_i = S_{i+1} - S_i$$

for $i = 1, \ldots, N-1$. Provided $\max_i T_i \geq 3$, the *intervals estimator* for the extremal index is defined as the minimum of 1 and

$$2 \frac{(\sum_i (T_i - 1))^2}{(N-1) \sum_i (T_i - 1)(T_i - 2)}$$

If the assumptions motivating the estimator are fulfilled, then the distribution of the inter-arrival times is a mixture of a point mass at zero and an exponential distribution. In an exponential quantile-quantile plot, the interarrival times should follow a broken-stick model, the location of the knot being determined by the extremal index. All inter-arrival times to the left of this knot correspond to *intra-cluster* inter-arrival times, that is, inter-arrival times *within* clusters of excesses; similarly, all inter-arrival times to the right of this knot correspond to *inter-cluster* inter-arrival times, that is, inter-arrival times *between* cluster of excesses. This heuristic also gives an automated way of partitioning high-threshold excesses into clusters.

If `plot = TRUE`, then two plots are being shown:

- Top: a time series plot with the high-threshold excesses being partitioned into clusters, the clusters being shown alternatingly by a red 'x' or a green 'o'.
- Bottom: an exponential quantile-quantile plot with the fitted broken-stick model.

Value

The extremal index estimate.

References

Ferro, C.A.T. and Segers, J. (2003). Inference for clusters of extreme values. Journal of the Royal Statistical Society, Series B 65, 545-556.

Leadbetter, M.R. (1983). Extremes and local dependence in stationary sequences. Zeitschrift fuer Wahrscheinlichkeitstheorie und Verwandte Gebiete 65, 291-306.

Examples

```
data(ABN)
ExtremalIndex(-ABN[2000:2500], threshold = 0.05, plot = TRUE)
```

`fitGPD` *Maximum likelihood estimation of the parameters of the (generalized) Pareto distribution*

Description

Given a sample of positive observations, the functions `fitGPD` and `fitPareto` compute the maximum likelihood estimators of the parameters of the (generalized) Pareto distributions.

Usage

```
fitGPD(z)
fitPareto(z)
```

Arguments

`z`	A numeric vector with positive elements.

Details

The *generalized Pareto distribution* with shape parameter γ and scale parameter $\sigma > 0$ is defined by its distribution function

$$F(z) = 1 - (1 + \gamma z/\sigma)^{-1/\gamma}$$

for all $z \geq 0$ such that $\sigma + \gamma z > 0$.

The *Pareto distribution* with shape parameter $\alpha > 0$ and scale parameter $\tau > 0$ is defined by its distribution function

$$F(z) = 1 - (1 + z/\tau)^{-\alpha}$$

for all $z \geq 0$. It is a reparametrization of the generalized Pareto distribution with positive shape parameter $\gamma = 1/\alpha$ and scale parameter $\sigma = \tau/\alpha$.

Value

For `fitGPD`, a list with components `gamma` and `sigma`. For `fitPareto`, a list with components `alpha` and `tau`.

Note

The function `fitGPD` is called within the function `GPD_par` if its argument `method` is set to `"ML"`.

References

Smith, R.L. (1987). Estimating tails of probability distributions. The Annals of Statistics 15, 1174-1207.

See Also

`GPD_par`, `ML`

Examples

```
# order statistics of random sample of size 100
# from the unit Frechet distribution:
x <- sort(- 1/log(runif(100)), decreasing = TRUE)
# fit GPD to sample of excesses over 21th largest observation:
fitGPD(x[1:20] - x[21])
```

`GPD_par` *Estimate GPD Parameters*

Description

Computes estimates of the parameters (γ, σ) of the generalized Pareto distribution fitted to excesses over a high threshold.

Usage

```
GPD_par(data, method = "ML", k = 5:(length(data) - 1))
```

Arguments

`data`	A numeric vector.
`method`	A character string determining which method will be used: `"Hill"`, `"ML"`, or `"Moment"`.
`k`	Integer vector. For each element of `k`, the parameter estimates will be computed based on the sample of excesses over the threshold u defined as the $(k+1)$th largest order statistic.

Details

Let $X_{1:n} < \ldots < X_{n:n}$ be a the increasing order statistics of the sample. Let $k = 1, \ldots, n-1$. The function fits the generalized Pareto distribution

$$H(z) = 1 - (1 + \gamma z/\sigma)^{-1/\gamma}$$

to the sample of excesses $X_{n-k+i:n} - X_{n-k:n}$, $i = 1, \ldots, k$ over the threshold $u = X_{n-k:n}$.

In case `method` is `"Hill"` or `"Moment"`, only those elements of `k` will be retained for which the corresponding order statistic is positive.

Value

A list with the `class` attribute `"GPD_par"`, which is a list containing the following components:

`gamma`	Numeric vector with the same length as `k` containing the estimates for γ.
`sigma`	Numeric vector with the same length as `k` containing the estimates for σ.
`threshold`	Numeric vector of thresholds corresponding to `k`.
`k`	Vector of k-values that have been used effectively.
`n`	The sample size.

References

Dekkers, A.L.M., Einmahl, J.H.J. and de Haan, L. (1989). A moment estimator for the index of an extreme-value distribution. The Annals of Statistics 17, 1833-1855.

Hill, B.M. (1975). A simple general approach to inference about the tail of a distribution. The Annals of Statistics 3, 1163-1174.

Smith, R.L. (1987). Estimating tails of probability distributions. The Annals of Statistics 15, 1174-1207.

See Also

`fitGPD`, `Hill`, `ML`, `Moment`

Examples

```
# random sample of size 100
# from the unit Frechet distribution:
x <- - 1/log(runif(100))
# fit GPD to sample of excesses over 21th largest observation:
out <- GPD_par(x)
# plot estimates of gamma as a function of k (on logarithmic scale)
# together with the true gamma (= 1)
plot(out$k, out$gamma, type = "l", log = "x"); abline(h = 1)
```

`Hill.diagnostic` *Diagnostic Plot for the Hill Estimator*

Description

Computes the Hill estimator at a certain threshold and shows a quantile-quantile plot of the log-excesses over the threshold versus the exponential distribution.

Usage

```
Hill.diagnostic(data, k)
```

Arguments

`data`	A numeric vector containing the observations.
`k`	An integer between 1 and $n - 1$, where n is the sample size, that is, the length of the `data` vector.

Details

For a sample $X_1, \ldots, X_n$ with order statistics $X_{1:n} \le \ldots \le X_{n:n}$ and for the given value of $k = 1, \ldots, n-1$, a scatterplot is shown of the points

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

for $i = 1, \ldots, k$. This is in fact a quantile-quantile plot of the excesses of the log-data over the threshold $\log X_{n-k:n}$ versus the standard exponential distribution.

The Hill estimator is defined as the mean of these log-excesses and is an estimator of the positive extreme-value index; see `Hill`. If the assumptions that justify the Hill estimator are justified, the points in the above scatterplot should be scattered around the line with intercept zero and slope equal to the Hill estimator. This heuristic can be transformed into an (experimental) method for automated threshold selection: see `ChooseK`.

Value

The function is primarily called for its side-effect, which is to show the above diagnostic plot. It silently returns a list with the following components:

`x`	The x-coordinates of the points in the scatterplot.
`y`	The y-coordinates of the points in the scatterplot.
`H`	The Hill estimate.

References

Hill, B.M. (1975). A simple general approach to inference about the tail of a distribution. The Annals of Statistics 3, 1163-1174.

See Also

`ChooseK`, `GPD_par`, `Hill`

Examples

```
x <- rburr(n = 1000, gamma = 0.5, rho = -0.5)
Hill.diagnostic(x, k = 20)  # high threshold, good fit
Hill.diagnostic(x, k = 200) # low threshold, bad fit
```

`extremevalueindex` *Estimators for the extreme-value index*

Description

Implementation of the Hill (1975) estimator, the moment estimator of Dekkers, Einmahl and de Haan (1989), and the maximum likelihood estimator of Smith (1987) of the extreme-value index. Allows for computation of confidence intervals and an experimental way to choose the threshold.

Usage

```
Hill(data, k = 5:(sum(data > 0) - 1), plot = TRUE,
        CI.type = c("Wald", "score", "none"), CI.p = NULL,
        choose.k = FALSE, test = "s", alpha = 0.5, ...)
Moment(data, k = 5:(sum(data > 0) - 1), plot = TRUE,
        CI.type = "Wald", CI.p = NULL,
        choose.k = FALSE, test = "s", alpha = 0.5, ...)
ML( data, k = 5:(length(data) - 1), plot = TRUE,
        CI.type = "Wald", CI.p = NULL,
        choose.k = FALSE, test = "s", alpha = 0.5, ...)
```

Arguments

`data`	The data vector
`k`	Vector of k values.
`plot`	Whether or not the results will be plotted. Defaults to `TRUE`.
`CI.type`	Type of confidence interval. For the functions `Moment` and `ML`, only the `"Wald"` symmetric confidence intervals are implemented.
`CI.p`	Nominal coverage probability of the confidence interval. If `NULL`, no confidence interval will be computed.
`choose.k`	Whether or not a choice for k will be suggested. Defaults to `FALSE`. See `ChooseK`.
`test`	If `choose.k = TRUE`, determines the test with which the goodness-of-fit of the exponential distribution to the residuals will be tested. See `ChooseK`.
`alpha`	If `choose.k = TRUE`, determines the nominal size of the test with which the goodness-of-fit of the exponential distribution to the log-excesses will be tested. See `ChooseK`.
`...`	Further arguments passed on to `plot`.

Details

By definition, the `Hill` estimator always returns a positive estimate, whereas the `Moment` and `ML` estimates can have either sign.

The `Wald` confidence interval is the usual symmetric interval centered around the estimator and based upon the estimated standard error and the normal approximation. In case of the `Hill` estimator, one can also choose the more accurate, asymmetric `score` confidence intervals (Haeusler and Segers, 2007).

For the `Hill` estimator and `Moment` estimator, values for `k` for which the $k+1$th order statistic is nonpositive will be ignored.

If `plot = TRUE`, the estimates are displayed as a function of k. Add the extra argument `log = "x"` to display the horizontal axis on a log-scale, as in the *altHill* plot (Resnick and Starica 1997; Drees et al. 2000).

If `choose.k = TRUE`, a value of k is selected using the function `ChooseK` with arguments `test` and `alpha`, and with argument `approx = Weissman` for the function `Hill` and with the arguments `approx = "GPD", method = "ML"` and `approx = "GPD", method = "Moment"` for the functions `ML` and `Moment`, respectively. In addition, if `plot = TRUE`, three plots are shown:

- Top: The estimates as a function of k.

- Bottom left: the p-values of the goodness-of-fit test as a function of k. See ChooseK.
- Bottom right: an exponential quantile-quantile plot of the residuals $Z_{i:k}$ at the selected value of k. See ChooseK.

Value

A list with the class attribute "EVI". If choose.k = TRUE, then the list gets the extra class attribute "ChooseK" and a number of extra fields related to the choice of k as in ChooseK.

n	The sample size.
k	Vector of values of k for which the Hill estimator has been computed.
threshold	Vector of thresholds corresponding to k.
estimate	Vector of corresponding Hill estimates.
CI	Matrix of upper and lower bounds of corresponding corresponding intervals.
CI.type	The type of confidence interval.
CI.p	Nominal coverage probability of the confidence intervals.
std.err	Vector of estimated standard errors.
data	The data
quantity	The string "gamma".
method	The name of the estimator used: "Hill", "ML" or "Moment".
...	If choose.k = TRUE, then the fields as in the output of ChooseK are present as well.

References

Dekkers, A.L.M., Einmahl, J.H.J. and de Haan, L. (1989). A moment estimator for the index of an extreme-value distribution. The Annals of Statistics 17, 1833-1855.

Drees, H., de Haan, L., and Resnick, S. (2000). How to make a Hill plot. The Annals of Statistics 28, 254-274.

Haeusler, E. and Segers, J. (2007). Assessing confidence intervals for the tail index by Edgeworth expansions for the Hill estimator. Bernoulli 13, 175-194.

Hill, B.M. (1975). A simple general approach to inference about the tail of a distribution. The Annals of Statistics 3, 1163-1174.

Resnick, S. and Starica, C. (1997). Smoothing the Hill estimator. Advances in Applied Probability 29, 271-293.

Smith, R.L. (1987). Estimating tails of probability distributions. The Annals of Statistics 15, 1174-1207.

See Also

ChooseK, GPD_par, Hill.diagnostic

Examples

```
# 1. power law with gamma = 1
x <- 1/runif(100)
Hill(x, CI.p = 0.95)
abline(h = 1, col = "blue")

# 2. altHill plot:
# display k on log-scale
# emphasise smaller k
x <- rburr(n = 1000, gamma = 0.5, rho = -0.75)
# linear scale
Hill(x, k = 10:500)
abline(h = 0.5, col = "blue")
# log scale
Hill(x, k = 10:500, log = "x")
abline(h = 0.5, col = "blue")

# 3. ML and Moment estimators
data(soa)
Moment(soa, k = 20:3000)
ML(soa, k = 20:3000)

# 4. choosing k
data(soa)
Moment(soa, k = 20:10000)
Moment(soa, k = 20:10000, choose.k = TRUE)
```

`LossALAE` *Liability Claims and Allocated Loss Adjustment Expenses*

Description

General liability claims `Loss` and allocated loss adjustement expenses `ALAE` provided by Insurance Services Office, Inc.

Usage

```
data(Loss)
data(ALAE)
```

Format

Numeric vectors of length 1500.

Details

Quoted from Frees and Valdez (1998), p. 12: "The data comprise 1500 general liability claims randomly chosen from late settlement lags and were provided by Insurance Services Office, Inc. Each claim consists of an indemnity payment `Loss` and an allocated loss adjustment expense `ALAE`. Here, `ALAE` are types of insurance company expenses that are specifically attributable to the settlement of individual claims such as lawyers' fees and claims investigation expenses."

Amongst others, the data are also analysed in Beirlant et al. (2004, chapter 9).

Source

Frees, E.W. en Valdez, E.A. (1998) Understanding relationships using copulas. North American Actuarial Journal 2, 1-15.

References

Beirlant, J., Goegebeur, Y., Segers, J., and Teugels, J. (2004). Statistics of Extremes. Wiley, Chichester. http://lstat.kuleuven.be/Wiley/index.html

Examples

```
data(Loss, ALAE)
plot(Loss, ALAE, xlog = TRUE, ylog = TRUE)
```

`MEplot` *Mean-Excess Plot*

Description

Draws the mean-excess plot for a given sample.

Usage

```
MEplot(data, omit = 0, ...)
```

Arguments

`data`	The sample.
`omit`	Number of largest observations for which the mean-excess function will *not* be plotted.
`...`	Further arguments passed on to `plot`.

Details

The *mean-excess function* of the distribution of the random variable X is defined as

$$m(x) = E[X - x \mid X > x]$$

Its empirical counterpart, the *empirical mean-excess function* $\hat{m}(x)$, is defined by taking expectations with respect to the empirical distribution: for $x < max_i(X_i)$,

$$\hat{m}(x) = \frac{\sum_i \max(X_i - x, 0)}{\sum_i I(X_i > x)}$$

The *mean-excess plot* is the plot of the pairs

$$(X_i, \hat{m}(X_i))$$

for $i = 1, \ldots, n-1$. Often, the points corresponding to the largest order statistics are omitted from the plot; this is the purpose of the argument `omit`.

For a distribution with extreme-value index $\gamma < 1$,

$$\lim_{x \to \infty} \frac{m(x)}{x} = \frac{\max(\gamma, 0)}{1 - \gamma}$$

As a consequence, if the empirical mean-excess function is increasing for large x, then this is an indication that the underlying distribution has a heavy tail.

Value

The function is mainly used for its side-effect, which is to plot the mean-excess function. The function invisibly returns a list with two components:

`x`	The x-coordinates of the points in the mean-excess plot.
`me`	The y-coordinates of the points in the mean-excess plot.

Examples

```
# for exponential data, the mean-excess function is approx. constant:
x <- rexp(n = 100, rate = 1)
MEplot(x)

# for heavy-tailed data, the mean-excess function is increasing:
x <- rburr(n = 100, gamma = 0.5, rho = -1)
MEplot(x, omit = 5)

# the Loss data look heavy-tailed:
data(Loss)
MEplot(Loss)
```

`norwegian` *Norwegian Fire Insurance Data*

Description

Norwegian fire insurance data treated in Beirlant et al. (1996a) and in Beirlant et al. (2004, Example 1.2). The data consists of fire insurance claims (times 1000 NOK) of the claims for the period 1972-1992. A priority of 500 units was in force.

Usage

```
data(norwegian)
```

Format

The data is a `list` with components `y72`, ..., `y92`. Each component is a numeric vector containing the claim values in increasing order. The number of claims may vary per year.

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. Wiley, Chichester. http://lstat.kuleuven.be/Wiley/index.html

Examples

```
data(norwegian)
hist(norwegian$y81)
Hill(norwegian$y81)
```

`PickandsDF` *Pickands Dependence Function*

Description

Computes an estimate of the Pickands dependence function of the extreme-value attractor of a bivariate distribution based on a bivariate sample $(X_1, Y_1), \ldots, (X_n, Y_n)$ from that distribution.

Usage

```
PickandsDF(data.x, data.y, data = NULL, w = (0:20)/20, k,
        method = "empirical", plot = TRUE)
```

Arguments

`data.x, data.y`	Numeric vectors containing the data $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_n$, respectively.
`data`	Alternatively, the data may be provided in the form of a n-by-2 matrix. If provided, then the arguments `data.x` and `data.y` are ignored.
`w`	A numeric vector giving the points in which the estimate of the Pickands dependence function will be computed. All the elements should be between 0 and 1.
`k`	A number specifying the tuning parameter k at which the estimate will be computed. See 'Details'.
`method`	A string specifying the estimation method; possible choices are `"empirical"` and `"angular"`. See 'Details'.
`plot`	If `TRUE` (the default), the result will be plotted.

Details

The *Pickands dependence function* of a bivariate extreme-value distribution is defined by

$$A(w) = l(w, 1 - w)$$

for $0 \leq w \leq 1$, where l is the (stable) tail dependence function; see `ETDF` for more details. Conversely,

$$l(x, y) = (x + y)A(x/(x + y))$$

for $x, y \geq 0$ and $x + y > 0$.

A function A from $[0, 1]$ to $[0, 1/2]$ is a Pickands dependence function if and only if (i) it is convex and (ii) $\max(w, 1 - w) \leq A(w) \leq 1$ for all $0 \leq w \leq 1$. The lower bound corresponds to asymptotic complete dependence, the upper bound to asymptotic independence. The coefficient of tail dependence is given by $\lambda = 2 - l(1, 1) = 2(1 - A(0.5))$.

If `method = "empirical"`, the estimate is computed by

$$\hat{A}(w) = \hat{l}(w, 1 - w)$$

where $\hat{l}$ is the empirical tail dependence function computed by `ETDF` at `k = k`. In general, the estimator does not satisfy any of the two requirements above.

If `method = "angular"`, the estimate is computed by

$$\hat{A}(w) = \int_0^{\pi/2} \max(w \cos\theta, (1-w)\sin\theta) d\hat{\Phi}(\theta)$$

where $\hat{\Phi}$ is an estimate of the angular measure; the estimator used is the estimator as implemented in `AngularMeasure` with `k = k` and `method = "c"`. In this way, the estimator is guaranteed to satisfy the two constraints mentioned above. By exploiting the above relations between A and l, a nonparametric estimator of l is obtained that is itself a stable tail dependence function.

Value

A `length(w)`-by-`length(methods)` matrix containing at position (i, j) the point estimate at `w[i]` computed by method `methods[j]`. The result is returned invisibly.

References

Pickands, J. (1981) Multivariate extreme value distributions. In: Bulletin of the International Statistical Institute, Proceedings of the 43rd Session, Buenos Aires, pp. 859-878.

See Also

`ETDF`, `AngularMeasure`

Examples

```
x <- rbivcauchy(1000)
w <- seq(from = 0, to = 1, by = 0.02)
PickandsDF(data = x, w = w, k = 20, method = c("empirical", "angular"))
lines(w, sqrt(w^2 + (1-w)^2), col = "red")
```

`rbivcauchy` — *Random Number Generation for the Bivariate Cauchy Distribution*

Description

Generates a random sample of the bivariate Cauchy distribution on the positive quadrant.

Usage

```
rbivcauchy(n)
```

Arguments

`n`	Sample size.

Details

The density of the bivariate Cauchy distribution on the positive quadrant is given by

$$f(x, y) = \frac{2}{\pi(1 + x^2 + y^2)^{3/2}}$$

for $x, y > 0$. Its marginal distributions are the standard Cauchy distribution restricted to the positive half-line.

The bivariate Cauchy distribution is *elliptic*: a random pair (X, Y) with this distribution can be represented as

$$(X, Y) = (R \cos \Theta, R \sin \Theta)$$

where $R > 0$ and $0 \leq \Theta \leq \pi/2$ are independent random variables, $P(R > r) = (1 + r^2)^{-1/2}$ for $r > 0$, and Θ is uniformly distributed on the interval $[0, \pi/2]$.

The bivariate Cauchy distribution is in the *bivariate max-domain of attraction* of the bivariate extreme-value distribution with unit Frechet margins and with stable tail dependence function

$$l(x, y) = (x^2 + y^2)^{1/2}$$

for $x, y > 0$; see Einmahl et al. (2001). The angular or spectral measure with respect to the Euclidean norm is simply

$$\Phi([0, \theta]) = \theta$$

for $0 \leq \theta \leq \pi/2$.

Value

An n-by-2 matrix containing the generated data.

References

Einmahl, J.H.J., de Haan, L. and Piterbarg, V.I. (2001). Nonparametric estimation of the spectral measure of an extreme value distribution. The Annals of Statistics 29, 1401-1423.

Examples

```
x <- rbivcauchy(1000)
AngularMeasure(data = x, k = c(20, 30, 50))
abline(a = 0, b = 1, col = "red")
```

---

rbivnorm — *Random Number Generation for the Bivariate Normal Distribution*

Description

Generates a random sample of the bivariate Normal distribution with given means, variances, and correlation.

Usage

```
rbivnorm(n, mean.x = 0, sd.x = 1, mean.y = 0, sd.y = 1, cor = 0)
```

Arguments

`n`	Sample size.
`mean.x, mean.y`	Means of the two components. Default value is 0.
`sd.x, sd.y`	Standard deviations of the two components. Should be positive. Default value is 1.
`cor`	Linear correlation between the two components. Should be between -1 and 1. Default value is 0.

Value

An n-by-2 matrix containing the generated data.

Examples

```
# Even for high correlation,
# the bivariate normal distribution has
# asymptotic independence in the tails:
x <- rbivnorm(10000, cor = 0.9)
plot(x[,1], x[,2])
plot(rank(x[,1]), rank(x[,2]))
```

`rburr` *Random number generation of the Burr distribution*

Description

Generates a random sample from the *Burr* distribution with parameters $\gamma > 0$ and $\rho < 0$, defined by its distribution function

$$F(z) = 1 - (z^{-\rho/\gamma} + 1)^{1/\rho}$$

for $z \geq 0$.

Usage

```
rburr(n, gamma, rho)
```

Arguments

`n`	Sample size.
`gamma, rho`	Parameters of the Burr distribution.

Details

The parametrization is chosen such that γ and ρ are the extreme-value index and second-order parameter of the distribution, respectively.

Value

A numeric vector of length `n`.

Examples

```
# rho far away from zero
# ==> fast convergence in domain-of-attraction asymptotics
# ==> estimates are stable as a function of the threshold
x <- rburr(n = 1e4, gamma = 0.5, rho = -1)
Hill(x, k = 10:1e3, ylim = c(0,1), log = "x")
abline(h = 0.5, col = "red")

# rho close to zero
# ==> slow convergence in domain-of-attraction asymptotics
# ==> only the highest thresholds can be used
x <- rburr(n = 1e4, gamma = 0.5, rho = -0.25)
Hill(x, k = 10:1e3, ylim = c(0,1), log = "x")
abline(h = 0.5, col = "red")
```

`RiskMeasure` *Risk Measure*

Description

Computes a number of (tail-related) risk measures for a number of distributions.

Usage

```
RiskMeasure(dist, par, rm, a = 0, b = Inf, eta = 1, p = 1)
```

Arguments

`dist`	A character string indicating the distribution. See *Details* below.
`par`	A numeric vector containing the parameters of the distribution. Parameters of the distribution. See *Details* below.
`rm`	A character string containing the parameter of the distribution.
`a, b`	Lower and upper level of an excess-of-loss contract; will be ignored unless `rm = "XL"`.
`eta`	Shape parameter in proportional hazards transform; will be ignored unless `rm = "PH"`.
`p`	Tail probability in case of expected shortfall, tail quantile, or economic capital. Will be ignored unless `rm` takes one of the values `"EC"`, `"ES"`, or `"TQ"`.

Details

The purpose of this function is primarily to investigate the performance of various estimators of tail-related risk measures implemented by providing the true values for a number of heavy-tailed distributions.

The arguments `dist` and `par` determine for which distribution the desired risk measure will be computed:

`"abs_t"` Distribution of the absolute value of a Student-*t* random variable with degrees of freedom nu determined by `par`.

`"Pareto"` The Pareto distribution with shape parameter $alpha$ determined by `par`.

The risk measure to be computed is determined by the value of the argument `rm` and, if relevant, the values of the arguments `a`, `b`, `nu`, and `p`:

"E" *Expectation* $E(X)$.

"EC" *Economic Capital*, defined as $Q(1-p) - E(X)$, with tail probability `p`.

"ES" *Expected Shortfall*, defined as $E[X|X > u]$, the threshold $u = Q(1-p)$ being determined by the tail probability `p`.

"PH" *Proportional Hazards transform*, defined as $\int_0^\infty \{P(X > x)\}^\eta dx$ with shape parameter `eta`.

"TQ" *Tail Quantile* $Q(1-p)$ with tail probability `p`.

"Var" *Variance* $\mathrm{var}(X)$.

"XL" *Excess-of-Loss* net premium $E[\max(min(X,b) - a)]$ with lower limit `a` and upper limit `b`. Set `b = Inf` in case of no upper limit.

Value

A number.

See Also

`EconomicCapital`, `ExcessLoss`, `Expectation`, `ExpectedShortfall`, `PHtransform`, `Variance`

Examples

```
# absolute values of random numbers from the t-distribution:
X <- abs(rt(n = 1000, df = 2))
# estimated tail quantile
TailQuantile(X, p = 0.001, k = (1:20)*5)
# true tail quantile
RiskMeasure(dist = "abs_t", par = 2, rm = "TQ", p = 0.001)
```

`RiskMeasureEstimators` *Risk Measure Estimators*

Description

Estimators of various of tail-related risk measures.

Usage

```
EconomicCapital(y, p, k = 5:(length(y) - 1),
         approx = "GPD", method = "ML", plot = TRUE, ...)
ExcessLoss(y, a, b = Inf, k = 5:(length(y) - 1),
         approx = "GPD", method = "ML", plot = TRUE, ...)
Expectation(y, k = 5:(length(y) - 1),
         approx = "GPD", method = "ML", plot = TRUE, ...)
ExpectedShortfall(y, p, k = 5:(length(y) - 1),
         approx = "GPD", method = "ML", plot = TRUE, ...)
PHtransform(y, eta = 1, k = 5:(length(y) - 1),
```

```
        approx = "GPD", method = "ML", plot = TRUE, ...)
TailQuantile(y, p, k = 5:(length(y) - 1),
        approx = "GPD", method = "Moment",
        choose.k = FALSE, B = 1000, leave.out = 20,
        k_rho = ceiling(length(y)^0.95),
        test = "s", alpha = 0.5, plot = TRUE, ...)
Variance(y, k = 5:(length(y) - 1),
        approx = "GPD", method = "ML", plot = TRUE, ...)
```

Arguments

- `y` — A numeric vector containing the data.
- `p` — Tail probability.
- `a,b` — Upper and lower limits of the excess-of-loss reinsurance contract.
- `eta` — Exponent η in the definition of the PH-transform.
- `k` — Vector of k values, determining at which threshold(s) the estimator will be computed.
- `approx` — Approximation method for the tail: `"GPD"` (default) or `"Weissman"`. The latter method is suitable only for heavy-tailed distributions, that is, with extreme-value index $\gamma > 0$.
- `method` — Estimation method for the tail parameters: `"Hill"`, `"ML"`, or `"Moment"`. Will be passed on to the function `GPD_par`.
- `plot` — Whether or not the results will be plotted. Defaults to `TRUE`.
- `choose.k` — If `FALSE` (the default), no automated threshold selection will be attempted. The other two possibilities are:
  - `"Bootstrap"`, in which case k will be chosen according to the bootstrap method of Ferreira et al. (2003);
  - `"Test"`, in which case k will be chosen according to an experimental method described in `ChooseK`.
- `B, leave.out, k_rho` — If `choose.k = "Bootstrap"`, these parameters are passed on to the boostrap procedure to select k; see 'Details'.
- `test, alpha` — If `choose.k = "Test"`, these arguments are passed on to `ChooseK` for the selection of k. If `plot = TRUE`, then the choice of k is illustrated through a number of extra graphs, see `ChooseK`.
- `...` — Further arguments passed on to `plot`. For instance, `log = "x"` puts the horizontal axis on a logarithmic scale, which sometimes facilitates the choice of the threshold via k.

Details

See `RiskMeasure` of a description of the risk measures above.

The risk measures are estimated as functionals of the estimated distribution. The latter is estimated in two pieces:

1. Nonparametrically up to the threshold $X_{n-k:n}$, the $(k+1)$-largest order statistic of the sample.
2. With extreme-value theory beyond $X_{n-k:n}$.

The argument `approx` determines which approximation is used for the tail beyond $X_{n-k:n}$:

- If `approx = "Weissman"`, the tail probability $1 - F(x)$ at $x > X_{n-k:n} > 0$ is estimated as
$$(k/n)(x/X_{n-k:n})^{\hat{\gamma}}$$
with γ a positive estimator of the extreme-value index.
- If `approx = "GPD"`, the tail probability $1 - F(x)$ at $x > X_{n-k:n}$ is estimated as
$$(k/n)(1 + \hat{\gamma}(x - X_{n-k:n})/\hat{\sigma})^{-1/\gamma}$$
with γ and σ estimators of the generalized Pareto distribution fitted to the excesses over the threshold $X_{n-k:n}$.

In both cases, the tail parameters are estimated by a call to the function `GPD_par` with arguments `approx` and `method`.

For the function `TailQuantile`, two methods for automated threshold selection are implemented:

- For `choose.k = "Bootstrap"`, the bootstrap method of Ferreira et al. (2003). This method is implemented in the function `TQ_ChooseK` [help file under construction], to which the additional arguments are passed on:
  - `B`, the number of bootstrap samples;
  - `leave.out`, the number of lowest and highest k-values to leave out in the search for the k that minimizes the estimated asymptotic mean squared error;
  - `k_rho`, determining the threshold at which to estimate the second-order parameter `\rho`.
- For `choose.k = "Test"`, an experimental method implemented in the function `ChooseK` and with arguments `test` and `alpha`.

Value

An object with `class` attribute `"EVI"`, that is, a list with the following components:

`n`	Sample size.
`k`	Number of threshold excesses.
`threshold`	Vector of thresholds.
`estimate`	Vector of point estimates.
`CI`	`NULL` (Confidence intervals are still to be implemented; however, for tail quantiles, see `Weissman.q`.)
`data`	A character string indicating the name of the data.
`quantity`	A character string describing the quantity being estimated.
`method`	A character string describing the estimator.
`gamma`	Vector of estimates of the extreme-value index.
`...`	If `choose.k` is `"Bootstrap"` or `"Test"`, the list contains a number of additional component providing diagnostics related to the choice of k [help file under construction].

References

A. Ferreira, L. de Haan and L. Peng (2003). On optimizing the estimation of high quantiles of a probability distribution. Statistics 37, 401-434.

Wang, S. (1995). Insurance Pricing and Increased Limits Ratemaking by Proportional Hazards Transforms. Insurance: Mathematics and Economics 17, 43-54.

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

Hill, ML, Moment, GPD_par, RiskMeasure, Weissman.q

Examples

```
# tail quantile and excess-of-loss net premium
# for Loss data of Frees and Valdez (1998)
data(Loss)
TailQuantile(Loss, p = 0.001, k = 25:200)
ExcessLoss(Loss, a = 1.2e6, k = 25:200)
```

---

soa *SOA Group Medical Insurance Large Claims Database*

Description

Claim data for 1991 of the Society of Actuaries' Group Medical Insurance Large Claims Database. Only claims over 25000 USD are in the database.

Usage

```
data(soa)
```

Format

Numeric vector of length 75789.

Details

See Grazier and G'Sell Associates (1997) for a thorough description of the data.

Amongst others, the data are also analysed in Cebrian et al. (2003) and in Beirlant et al. (2004, chapter 9).

Source

Grazier, K.L. and G'Sell Associates (1997). Group Medical Insurance Large Claims Database and Collection. SOA Monograph M-HB97-1, Society of Actuaries, Schaumburg.

http://www.soa.org

References

Beirlant, J., Goegebeur, Y., Segers, J., and Teugels, J. (2004). Statistics of Extremes. Wiley, Chichester. http://lstat.kuleuven.be/Wiley/index.html

Cebrian, A., Denuit, M., and Lambert, Ph. (2003). Generalized Pareto fit to the society of Actuaries' large claims database. North American Actuarial Journal 7, 18-36.

Examples

```
data(soa)
Hill(soa, k = (1:100)*20)
```

TailProb_sum *Tail Probability Estimation for a Sum of Random Variables*

Description

Computes an estimate of the probability that $w_x X + w_y Y > s$, where s is large.

Usage

```
TailProb_sum(s, w.x = 1, w.y = 1,
        tail.x, tail.y, Phi, plot = TRUE)
```

Arguments

`s`	The level for which the probability of excess is to be estimated.
`w.x`	A positive number; the weight w_x of X.
`w.y`	A positive number; the weight w_y of X.
`tail.x`	An object with `class` attribute `"GPD_par"`, i.e. the output of a call to the function `GPD_par` applied to the X data.
`tail.y`	Idem, now for the Y data.
`Phi`	An object with `class` attribute `"AngularMeasure"`, i.e. the output of a call to the function `AngularMeasure` applied to the data.
`plot`	If `TRUE` (the default), the results will be plotted.

Details

If `tail.x$k` and `tail.y$k` are vectors, the tail probability will be estimated for every possible combination of choices of k for X and Y.

If `plot` is `TRUE`, the tail probability estimates are plotted as a surface in function of `tail.x$k` and `tail.y$k`.

Value

The function invisibly returns a matrix with at position (i, j) the estimated tail probability when the tails of X and Y are estimated for k equal to `tail.x$k[i]` and `tail.x$k[j]`.

See Also

`AngularMeasure`, `GPD_par`, `TailQuantile_sum`

Examples

```
# estimate probability that the daily logreturn
# of a balanced portfolio of stocks ABN AMRO and ING
# is less than -0.10, i.e. a loss on the portfolio
# of more than about 10 percent
data(ABN, ING)
GPD.x <- GPD_par(-ABN, method = "Moment", k = (2:30)*10)
GPD.y <- GPD_par(-ING, method = "Moment", k = (2:30)*10)
Phi <- AngularMeasure(data.x = -ABN, data.y = -ING, k = 100)
TailProb_sum(s = 0.10, w.x = 0.5, w.y = 0.5,
        tail.x = GPD.x, tail.y = GPD.y, Phi = Phi)
```

TailQuantile_sum *Tail Quantile Estimation for a Sum of Random Variables*

Description

Computes an estimate of a tail quantile of a weighted sum $w_x X + w_y Y$.

Usage

```
TailQuantile_sum(p, w.x = NULL, w.y = NULL, lambda = NULL,
         tail.x, tail.y, Phi, lower, upper, plot = TRUE)
```

Arguments

p	Tail probability.
w.x, w.y	Weights $w_x \geq 0$ and $w_y \geq 0$ for X and Y, respectively; can be vectors (of the same length).
lambda	The weights may also be specified in the form $w_x = \lambda$ and $w_y = 1 - \lambda$.
tail.x	An object with class attribute "GPD_par", i.e. the output of a call to the function GPD_par applied to the X data with a *single* value for k.
tail.y	Similarly for Y.
Phi	An object with class attribute "AngularMeasure", i.e. the output of a call to the function AngularMeasure applied to the data.
lower	A priori lower bound for the tail quantiles.
upper	A priori upper bound for the tail quantiles.
plot	If TRUE (the default), the results will be plotted.

Details

A search is performed to find the value of s so that the tail probability estimated by TailProb_sum is equal to p.

If plot is TRUE, the estimated tail probabilities are plotted as a function of w.x or lambda.

Value

The function silently returns the vector of tail quantile estimates.

See Also

AngularMeasure, GPD_par, TailProb_sum

Examples

```
# determine a level s such that
# the probability that
# a portfolio of stocks ABN AMRO and ING
# has a daily logreturn of less than -s
# is equal to 0.001
data(ABN, ING)
GPD.x <- GPD_par(-ABN, method = "Moment", k = 100)
```

```
print(GPD.x)
GPD.y <- GPD_par(-ING, method = "Moment", k = 100)
print(GPD.y)
Phi <- AngularMeasure(data.x = -ABN, data.y = -ING, k = 100)
TailQuantile_sum(p = 0.001, lambda = (0:10)/10,
        lower = 0.05, upper = 0.15,
        tail.x = GPD.x, tail.y = GPD.y, Phi = Phi)
```

---

top40 *The 40 Most Costly Insurance Losses*

Description

The 40 most costly insurance losses (property and business interruption, excluding liability and life insurance losses) as provided in Sigma (2006, No. 2, p. 35). The amounts are indexed to 2005 and are expressed in million USD.

Usage

```
data(top40)
```

Format

A numeric vector of length 40.

Source

Sigma (2006, No. 2). Swiss Reinsurance Company, Zuerich. http://www.swissre.com/sigma.

Examples

```
data(top40)
top40
summary(top40)
hist(top40)
plot(top40, type = "h")
Hill(top40)
```

UvT_Cat *Random Number Generation from a Heavy-Tailed Distribution*

Description

Generates random samples of the convolution $F = G * H$ of the generalized Pareto distribution G with parameters $\gamma = 0.75$ and $\sigma = 1$ on the one hand and the Gamma(3, 1) distribution H on the other hand.

Usage

```
UvT_Cat(n)
```

Arguments

n Sample size.

Value

A numeric vector of length n.

See Also

rgamma

Examples

```
x <- UvT_Cat(1000)
Hill(x)
```

Weissman.q *Weissman Quantile Estimator*

Description

This function is an implementation of the Weissman (1978) estimator for a high tail quantile of a heavy-tailed distribution based on an estimate of the (positive) extreme-value index.

Usage

```
Weissman.q(EVI, p, plot = TRUE, ...)
```

Arguments

EVI	An object with class attribute "EVI", i.e. the output of one of the functions Hill, ML, or Moment. See 'Examples' below.
p	Tail probability of the quantile to be estimated.
plot	If TRUE (the default), the result will be plotted.
...	Further arguments passed on to plot provided plot = TRUE. For instance, log = "x", log = "y", and log = "xy" draw the horizontal and/or the vertical axis on logarithmic scale. The former is useful for selecting k, the latter is useful for estimating extreme quantiles of very heavy-tailed distributions.

Details

Let $X_{1:n} \leq \ldots \leq X_{n:n}$ be the ascending order statistics of a sample and let $k = 1, \ldots, n-1$ be such that $X_{n-k:n} > 0$. For $0 < p \leq k/n$, the *Weissman (1978) estimator* of the tail quantile $Q(1-p)$ is defined as

$$X_{n-k:n}(k/(np))^{\hat{\gamma}}$$

where $\hat{\gamma}$ is a positive estimate of the extreme-value index.

For random samples from a distribution with positive extreme-value index and if

- $k = k_n$ is an *intermediate* sequence, i.e., $k/n \to 0$ and $k \to \infty$,
- $0 < p = p_n \leq k/n$ tends to zero at a certain speed,

- $\hat{\gamma}_n$ is a consistent estimator sequence of the extreme-value index,

the Weissman quantile estimator is consistent in the sense that the relative error tends to zero. Under additional assumptions, the estimator is also asymptotically normal; see for instance Beirlant et al. (2004, section 4.6.1).

Value

An object with `class` attribute `"EVI"`, that is, a list with the following components:

`n`	Sample size.
`k`	Number of threshold excesses.
`threshold`	Vector of thresholds.
`estimate`	Vector of point estimates.
`CI`	Matrix with upper and lower endpoints of confidence intervals.
`CI.type`	A character string indicating the type of confidence interval.
`CI.p`	Nominal coverage probability of confidence interval.
`data`	A character string indicating the name of the data.
`quantity`	A list with two components: `name`, equal to `"Q"`, and `par`, equal to `1-p`.
`method`	A character string describing the estimator.

References

Beirlant, J., Goegebeur, Y., Segers, J., and Teugels, J. (2004). Statistics of Extremes. 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

`Burr.Weissman`, `Hill`, `ML`, `Moment`, `TailQuantile`

Examples

```
# norwegian fire insurance data:
data(norwegian)
# estimate gamma by the moment estimator:
gamma.M <- Moment(norwegian$y81, CI.p = 0.9, k = 10:100)
# use output gamma.M as input for the Weissman estimator:
Q.M <- Weissman.q(gamma.M, p = 0.01)
```

Index