

Package ‘mev’

October 22, 2025

Type Package

Title Modelling of Extreme Values

Version 2.0

Description Various tools for the analysis of univariate, multivariate and functional extremes. Exact simulation from max-stable processes (Dombry, Engelke and Oestling, 2016, <[doi:10.1093/biomet/asw008](https://doi.org/10.1093/biomet/asw008)>, R-Pareto processes for various parametric models, including Brown-Resnick (Wadsworth and Tawn, 2014, <[doi:10.1093/biomet/ast042](https://doi.org/10.1093/biomet/ast042)>) and Extremal Student (Thibaud and Opitz, 2015, <[doi:10.1093/biomet/asv045](https://doi.org/10.1093/biomet/asv045)>). Threshold selection methods, including Wadsworth (2016) <[doi:10.1080/00401706.2014.998345](https://doi.org/10.1080/00401706.2014.998345)>, and Northrop and Coleman (2014) <[doi:10.1007/s10687-014-0183-z](https://doi.org/10.1007/s10687-014-0183-z)>. Multivariate extreme diagnostics. Estimation and likelihoods for univariate extremes, e.g., Coles (2001) <[doi:10.1007/978-1-4471-3675-0](https://doi.org/10.1007/978-1-4471-3675-0)>.

License GPL-3

URL <https://lbelzile.github.io/mev/>, <https://github.com/lbelzile/mev/>

BugReports <https://github.com/lbelzile/mev/issues/>

Depends R (>= 3.4)

Imports alabama, methods, nleqslv, numDeriv, Rcpp (>= 0.12.16),
Rsolnp, stats,

Suggests boot, cobs, evd, expint, knitr, MASS, mvPot (>= 0.1.4),
mvtnorm, gmm, revdbayes, rmarkdown, ismev, tinytest,
TruncatedNormal (>= 1.1)

LinkingTo Rcpp, RcppArmadillo

LazyData true

RoxygenNote 7.3.3

VignetteBuilder knitr

Encoding UTF-8

ByteCompile true

NeedsCompilation yes

Author Leo Belzile [aut, cre] (ORCID: <<https://orcid.org/0000-0002-9135-014X>>),
 Jennifer L. Wadsworth [aut],
 Paul J. Northrop [aut] (ORCID: <<https://orcid.org/0000-0002-1992-4882>>),
 Raphael Huser [aut] (ORCID: <<https://orcid.org/0000-0002-1228-2071>>),
 Scott D. Grimshaw [aut] (ORCID:
 <<https://orcid.org/0000-0002-6326-9360>>),
 Jin Zhang [ctb],
 Michael A. Stephens [ctb],
 Art B. Owen [ctb]

Maintainer Leo Belzile <belzilel@gmail.com>

Repository CRAN

Date/Publication 2025-10-22 09:50:02 UTC

Contents

abisko	5
adf	5
angmeas	6
angmeasdir	8
confint.eprof	9
dgeoaniso	10
egp	11
egp.pll	12
egpdist	13
emplik	15
eskrain	16
expme	17
extcoef	18
extgp	21
extgp.G	23
fit.egp	24
fit.extgp	26
fit.gev	28
fit.gpd	29
fit.pp	32
fit.rho	33
fit.rlarg	34
fit.shape	35
fit.wgpd	36
frwind	37
geomagnetic	38
gev	38
gev.abias	40
gev.bcor	40
gev.mle	41
gev.Nyr	42
gev.pll	43

gev.tem	45
gevdist	47
gevN	48
gevr	49
gpd	50
gpd.abias	51
gpd.bcor	52
gpd.boot	53
gpd.lmom	53
gpd.mle	54
gpd.pll	55
gpd.tem	57
gpde	59
gpdist	60
gpdN	61
gpdr	62
ibvpot	63
kjtail	65
leedspollution	66
likmgp	67
maiquetia	68
maxstable	69
nidd	69
nutrients	70
pandemics	72
penultimate	72
plot.eprof	74
plot.fr	75
plot.mev_shape_rbm	76
plot.mev_thselect_vmetric	77
plot.mev_thselect_wadsworth	77
plot.mev_tstab_mrl	78
pp	79
qweissman	80
rdir	81
rgparp	81
rho.dk	83
rho.fagh	84
rho.gbw	85
rho.ghp	85
rlarg	86
rmev	87
rmevspec	90
rmnorm	92
rparp	93
rparpcs	94
rparpcshr	96
rrlarg	97

shape.erm	98
shape.genjack	99
shape.genquant	99
shape.hill	100
shape.lthill	101
shape.moment	102
shape.osz	102
shape.pickands	103
shape.rbm	104
shape.trimhill	104
shape.vries	105
spunif	106
taildep	107
test.maxstab	109
test.scoreindep	110
thselect.alrs	111
thselect.bab	112
thselect.cv	113
thselect.expgqt	114
thselect.gbw	116
thselect.ksmd	117
thselect.mdps	118
thselect.mrl	118
thselect.ncpgp	120
thselect.pec	120
thselect.pickands	121
thselect.rbm	122
thselect.samsee	123
thselect.sdinfo	123
thselect.vmetric	125
thselect.wseq	126
tstab.cv	129
tstab.egp	130
tstab.gpd	131
tstab.hill	132
tstab.lthill	133
tstab.mrl	134
venice	135
w1500m	136
xdep.asym	136
xdep.chi	138
xdep.chibar	140
xdep.eta	142
xdep.xindex	144

abisko	<i>Abisko rainfall</i>
--------	------------------------

Description

Daily non-zero rainfall measurements in Abisko (Sweden) from January 1913 until December 2014.

Arguments

date	Date of the measurement
precip	rainfall amount (in mm)

Format

a data frame with 15132 rows and two variables

Source

Abisko Scientific Research Station

References

A. Kiriliouk, H. Rootzen, J. Segers and J.L. Wadsworth (2019), *Peaks over thresholds modeling with multivariate generalized Pareto distributions*, *Technometrics*, **61**(1), 123–135, <doi:10.1080/00401706.2018.1462738>

adf	<i>Estimation of the bivariate angular dependence function</i>
-----	--

Description

Estimation of the bivariate angular dependence function

Usage

```
adf(  
  xdat,  
  qlev = 0.95,  
  estimator = c("hill", "mle", "bayes"),  
  level = 0.95,  
  ties.method = "random",  
  angles = seq(0, 1, by = 0.02),  
  plot = TRUE  
)
```

Arguments

<code>xdat</code>	an n by 2 matrix of multivariate observations
<code>qlav</code>	quantile level on uniform scale at which to threshold data. Default to 0.95
<code>estimator</code>	string indicating the estimation method
<code>level</code>	level for confidence intervals, default to 0.95
<code>ties.method</code>	method for handling of ties in rank transformation
<code>angles</code>	vector of angles at which to evaluate the angular dependence function The confidence intervals are based on normal quantiles. The standard errors for the hill are based on the asymptotic covariance and that of the mle derived using the delta-method. Bayesian posterior predictive interval estimates are obtained using ratio-of-uniform sampling with flat priors: the shape parameters are constrained to lie within the triangle, as are frequentist point estimates which are adjusted post-inference.
<code>plot</code>	logical indicating whether to plot the function, defaults to TRUE

Value

a plot of the angular dependence function if `plot=TRUE`, plus an invisible list with components

- `angle` the sequence of angles in (0,1) at which the lambda values are evaluated
- `coef` point estimates of the angular dependence function
- `lower` `level%` confidence interval for lambda (lower bound)
- `upper` `level%` confidence interval for lambda (upper bound)

References

J.L. Wadsworth and J.A. Tawn (2013). A new representation for multivariate tail probabilities, *Bernoulli*, 19(5B), 2689-2714.

Examples

```
set.seed(12)
dat <- mev::rmev(n = 1000, d = 2, model = "log", param = 0.1)
adf(xdat = dat, estimator = 'hill')
```

angmeas

Rank-based transformation to angular measure

Description

The method uses the pseudo-polar transformation for suitable norms, transforming the data to pseudo-observations, then marginally to unit Frechet or unit Pareto. Empirical or Euclidean weights are computed and returned alongside with the angular and radial sample for values above threshold(s) `thresh`, specified in terms of quantiles of the radial component `R` or marginal quantiles. Only complete tuples are kept.

Usage

```
angmeas(
  xdat,
  thresh,
  Rnorm = c("l1", "l2", "linf"),
  Anorm = c("l1", "l2", "linf", "arctan"),
  marg = c("frechet", "pareto"),
  wgt = c("empirical", "euclidean"),
  region = c("sum", "min", "max"),
  is.angle = FALSE,
  ...
)
```

Arguments

xdat	an n by d sample matrix
thresh	threshold of length 1 for 'sum', or d marginal thresholds otherwise.
Rnorm	character string indicating the norm for the radial component.
Anorm	character string indicating the norm for the angular component. arctan is only implemented for $d = 2$
marg	character string indicating choice of marginal transformation, either to Frechet or Pareto scale
wgt	character string indicating weighting function for the equation. Can be based on Euclidean or empirical likelihood for the mean
region	character string specifying which observations to consider (and weight). 'sum' corresponds to a radial threshold $\sum x_i > \text{thresh}$, 'min' to $\min x_i > \text{thresh}$ and 'max' to $\max x_i > \text{thresh}$.
is.angle	logical indicating whether observations are already angle with respect to region. Default to FALSE.
...	additional arguments

Details

The empirical likelihood weighted mean problem is implemented for all thresholds, while the Euclidean likelihood is only supported for diagonal thresholds specified via `region=sum`.

Value

a list with arguments `ang` for the $d - 1$ pseudo-angular sample, `rad` with the radial component and possibly `wt`s if `Rnorm='l1'` and the empirical likelihood algorithm converged. The Euclidean algorithm always returns weights even if some of these are negative.

a list with components

- `ang` matrix of pseudo-angular observations
- `rad` vector of radial contributions
- `wt`s empirical or Euclidean likelihood weights for angular observations

Author(s)

Leo Belzile

References

Einmahl, J.H.J. and J. Segers (2009). Maximum empirical likelihood estimation of the spectral measure of an extreme-value distribution, *Annals of Statistics*, **37**(5B), 2953–2989.

de Carvalho, M. and B. Oumow and J. Segers and M. Warchol (2013). A Euclidean likelihood estimator for bivariate tail dependence, *Comm. Statist. Theory Methods*, **42**(7), 1176–1192.

Owen, A.B. (2001). *Empirical Likelihood*, CRC Press, 304p.

Examples

```
x <- rmev(n = 25, d = 3, param = 0.5, model = 'log')
wts <- angmeas(xdat = x, Rnorm = 'l1', Anorm = 'l1', marg = 'frechet', wgt = 'empirical')
wts2 <- angmeas(xdat = x, Rnorm = 'l2', Anorm = 'l2', marg = 'pareto')
```

angmeasdir

Dirichlet mixture smoothing of the angular measure

Description

This function computes the empirical or Euclidean likelihood estimates of the spectral measure and uses the points returned from a call to angmeas to compute the Dirichlet mixture smoothing of de Carvalho, Warchol and Segers (2012), placing a Dirichlet kernel at each observation.

Usage

```
angmeasdir(
  xdat,
  thresh,
  Rnorm = c("l1", "l2", "linf"),
  Anorm = c("l1", "l2", "linf", "arctan"),
  marg = c("frechet", "pareto"),
  wgt = c("empirical", "euclidean"),
  region = c("sum", "min", "max"),
  is.angle = FALSE,
  ...
)
```

Arguments

xdat	an n by d sample matrix
thresh	threshold of length 1 for 'sum', or d marginal thresholds otherwise.
Rnorm	character string indicating the norm for the radial component.

Anorm	character string indicating the norm for the angular component. arctan is only implemented for $d = 2$
marg	character string indicating choice of marginal transformation, either to Frechet or Pareto scale
wgt	character string indicating weighting function for the equation. Can be based on Euclidean or empirical likelihood for the mean
region	character string specifying which observations to consider (and weight). 'sum' corresponds to a radial threshold $\sum x_i > \text{thresh}$, 'min' to $\min x_i > \text{thresh}$ and 'max' to $\max x_i > \text{thresh}$.
is.angle	logical indicating whether observations are already angle with respect to region. Default to FALSE.
...	additional arguments

Details

The cross-validation bandwidth is the solution of

$$\max_{\nu} \sum_{i=1}^n \log \left\{ \sum_{k=1, k \neq i}^n p_{k,-i} f(\mathbf{w}_i; \nu \mathbf{w}_k) \right\},$$

where f is the density of the Dirichlet distribution, $p_{k,-i}$ is the Euclidean weight obtained from estimating the Euclidean likelihood problem without observation i .

Value

an invisible list with components

- nu bandwidth parameter obtained by cross-validation;
- dirparmat n by d matrix of Dirichlet parameters for the mixtures;
- wts mixture weights.

Examples

```
set.seed(123)
x <- rmev(n = 100, d = 2L, param = 0.5, model = 'log')
out <- angmeasdir(x)
```

Description

Computes confidence intervals for the parameter ψ for profile likelihood objects. This function uses spline interpolation to derive level confidence intervals

Usage

```
## S3 method for class 'eprof'
confint(
  object,
  parm,
  level = 0.95,
  prob = c((1 - level)/2, 1 - (1 - level)/2),
  print = FALSE,
  method = c("cobs", "smooth.spline"),
  boundary = FALSE,
  ...
)
```

Arguments

object	an object of class eprof, normally the output of gpd.pll or gev.pll .
parm	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
level	confidence level, with default value of 0.95
prob	percentiles, with default giving symmetric 95% confidence intervals
print	should a summary be printed. Default to FALSE.
method	string for the method, either cobs (constrained robust B-spline from eponym package) or smooth.spline
boundary	logical; if TRUE, the null distribution is assumed to be a mixture of a point mass and half a chi-square with one degree of freedom.
...	additional arguments passed to functions. Providing a logical warn=FALSE turns off warning messages when the lower or upper confidence interval for psi are extrapolated beyond the provided calculations.

Value

returns a 2 by 3 matrix containing point estimates, lower and upper confidence intervals based on the likelihood root and modified version thereof

dgeoaniso

Distance matrix with geometric anisotropy

Description

The function computes the distance between locations, with geometric anisotropy. Consider real parameters θ_1 and θ_2 , and the transformation $\psi = \arctan(\theta_1/\theta_2)/2$ and $r = 1 + \theta_1^2 + \theta_2^2$. The dilation and rotation matrix is

$$\begin{pmatrix} \sqrt{r} \cos(\rho) & -\sqrt{r} \sin(\rho) \\ \sin(\rho)/\sqrt{r} & \cos(\rho)/\sqrt{r} \end{pmatrix}.$$

The parametrization is convenient for optimization purposes, as the parameter vector is unconstrained and the transformation has unit Jacobian.

Usage

```
dgeoaniso(loc, theta)
```

Arguments

loc	a d by 2 matrix of locations giving the coordinates of a site per row.
theta	numeric vector of length 2, real parameters

Value

a d by d square matrix of pairwise distance

References

Rai, K. and Brown, P.E. (2025), A parameter transformation of the anisotropic Matérn covariance function. Canadian Journal of Statistics e11839. doi:[10.1002/cjs.11839](https://doi.org/10.1002/cjs.11839)

egp

Extended generalised Pareto families

Description

This function provides the log-likelihood and quantiles for the three different families presented in Papastathopoulos and Tawn (2013) and the two proposals of Gamet and Jalbert (2022), plus exponential tilting. All of the models contain an additional parameter, $\kappa \geq 0$. All families share the same tail index as the generalized Pareto distribution, while allowing for lower thresholds. For most models, the distribution reduce to the generalised Pareto when $\kappa = 1$ (for models gj-tnorm and logist, on the boundary of the parameter space when $\kappa \rightarrow 0$).

egp.retlev gives the return levels for the extended generalised Pareto distributions

Arguments

xdat	vector of observations, greater than the threshold
thresh	threshold value
par	parameter vector (κ, σ, ξ).
model	a string indicating which extended family to fit
show	logical; if TRUE, print the results of the optimization
p	extreme event probability; p must be greater than the rate of exceedance for the calculation to make sense. See Details .
plot	logical; if TRUE, a plot of the return levels

Details

For return levels, the p argument can be related to T year exceedances as follows: if there are n_y observations per year, than take p to equal $1/(Tn_y)$ to obtain the T -years return level.

Value

`egp.ll` returns the log-likelihood value, while `egp.retlev` returns a plot of the return levels if `plot=TRUE` and a list with tail probabilities p , return levels `retlev`, thresholds `thresh` and model name `model`.

Usage

```
egp.ll(xdat, thresh, model, par)
egp.retlev(xdat, thresh, par, model, p, plot=TRUE)
```

Author(s)

Leo Belzile

References

Papastathopoulos, I. and J. Tawn (2013). Extended generalised Pareto models for tail estimation, *Journal of Statistical Planning and Inference* **143**(3), 131–143, <doi:10.1016/j.jspi.2012.07.001>.

Gamet, P. and Jalbert, J. (2022). A flexible extended generalized Pareto distribution for tail estimation. *Environmetrics*, **33**(6), <doi:10.1002/env.2744>.

Examples

```
set.seed(123)
xdat <- rgp(1000, loc = 0, scale = 2, shape = 0.5)
par <- fit.egp(xdat, thresh = 0, model = 'gj-beta')$par
p <- c(1/1000, 1/1500, 1/2000)
# With multiple thresholds
th <- c(0, 0.1, 0.2, 1)
opt <- tstab.egp(xdat, thresh = th, model = 'gj-beta')
egp.retlev(xdat = xdat, thresh = th, model = 'gj-beta', p = p)
opt <- tstab.egp(xdat, th, model = 'pt-power', plots = NA)
egp.retlev(xdat = xdat, thresh = th, model = 'pt-power', p = p)
```

egp.pll

Profile log likelihood for extended generalized Pareto models

Description

Computes the profile log likelihood over a grid of values of ψ for various parameters, including return levels.

Usage

```

egp.pll(
  psi,
  model = c("pt-beta", "pt-gamma", "pt-power", "gj-tnorm", "gj-beta", "exptilt",
    "logist"),
  param = c("kappa", "scale", "shape", "retlev"),
  mle = NULL,
  xdat,
  thresh = NULL,
  plot = FALSE,
  method = c("Nelder", "nlminb", "BFGS"),
  p,
  ...
)

```

Arguments

<code>psi</code>	grid of values for the parameter to profile
<code>model</code>	string; choice of extended eneralized Pareto model.
<code>param</code>	string; parameter to profile
<code>mle</code>	a vector or matrix with maximum likelihood estimates of kappa, scale, shape. This can be a matrix if there are multiple threshold
<code>xdat</code>	vector of observations
<code>thresh</code>	vector of positive thresholds. If NULL, defaults to zero.
<code>plot</code>	logical; if TRUE, returns a plot of the profile log likelihood
<code>method</code>	string giving the optimization method for the outer optimization in the augmented Lagrangian routine; one of nlminb or BFGS
<code>p</code>	tail probability for return level if param="retlev".
<code>...</code>	additional arguments, currently ignored

Value

an object of class eprof

egpdist

Extended generalized Pareto distribution

Description

Density function, distribution function, quantile function and random number generation for various extended generalized Pareto distributions

Usage

```

pegp(
  q,
  scale,
  shape,
  kappa,
  model = c("pt-beta", "pt-gamma", "pt-power", "gj-tnorm", "gj-beta", "exptilt",
    "logist"),
  lower.tail = TRUE,
  log.p = FALSE
)

degp(
  x,
  scale,
  shape,
  kappa,
  model = c("pt-beta", "pt-gamma", "pt-power", "gj-tnorm", "gj-beta", "exptilt",
    "logist"),
  log = FALSE
)

qegp(
  p,
  scale,
  shape,
  kappa,
  model = c("pt-beta", "pt-gamma", "pt-power", "gj-tnorm", "gj-beta", "exptilt",
    "logist"),
  lower.tail = TRUE
)

regp(
  n,
  scale,
  shape,
  kappa,
  model = c("pt-beta", "pt-gamma", "pt-power", "gj-tnorm", "gj-beta", "exptilt",
    "logist")
)

```

Arguments

scale	scale parameter, strictly positive.
shape	shape parameter.
kappa	shape parameter for the tilting distribution.
model	string giving the distribution of the model

<code>lower.tail</code>	logical; if TRUE (default), the lower tail probability $\Pr(X \leq x)$ is returned.
<code>log.p, log</code>	logical; if FALSE (default), values are returned on the probability scale.
<code>x, q</code>	vector of quantiles
<code>p</code>	vector of probabilities
<code>n</code>	scalar number of observations

References

Papastathopoulos, I. and J. Tawn (2013). Extended generalised Pareto models for tail estimation, *Journal of Statistical Planning and Inference* **143**(3), 131–143, <doi:10.1016/j.jspi.2012.07.001>.

Gamet, P. and Jalbert, J. (2022). A flexible extended generalized Pareto distribution for tail estimation. *Environmetrics*, **33**(6), <doi:10.1002/env.2744>.

emplik	<i>Self-concordant empirical likelihood for a vector mean</i>
--------	---

Description

Self-concordant empirical likelihood for a vector mean

Usage

```
emplik(
  dat,
  mu = rep(0, ncol(dat)),
  lam = rep(0, ncol(dat)),
  eps = 1/nrow(dat),
  M = 1e+30,
  thresh = 1e-30,
  itermax = 100
)
```

Arguments

<code>dat</code>	n by d matrix of d-variate observations
<code>mu</code>	d vector of hypothesized mean of dat
<code>lam</code>	starting values for Lagrange multiplier vector, default to zero vector
<code>eps</code>	lower cutoff for $-\log$, with default $1/\text{nrow}(\text{dat})$
<code>M</code>	upper cutoff for $-\log$.
<code>thresh</code>	convergence threshold for log likelihood (default of $1e-30$ is aggressive)
<code>itermax</code>	upper bound on number of Newton steps.

Value

a list with components

- logelr log empirical likelihood ratio.
- lam Lagrange multiplier (vector of length d).
- wts n vector of observation weights (probabilities).
- conv boolean indicating convergence.
- niter number of iteration until convergence.
- ndec Newton decrement.
- gradnorm norm of gradient of log empirical likelihood.

Author(s)

Art Owen, C++ port by Leo Belzile

References

Owen, A.B. (2013). Self-concordance for empirical likelihood, *Canadian Journal of Statistics*, **41**(3), 387–397.

eskrain

Eskdalemuir Observatory Daily Rainfall

Description

This dataset contains exceedances of 30mm for daily cumulated rainfall observations over the period 1970-1986. These data were aggregated from hourly series.

Format

a vector with 93 daily cumulated rainfall measurements exceeding 30mm.

Details

The station is one of the rainiest of the whole UK, with an average 1554mm of cumulated rainfall per year. The data consisted of 6209 daily observations, of which 4409 were non-zero. Only the 93 largest observations are provided.

Source

Met Office.

expme

*Exponent measure for multivariate generalized Pareto distributions***Description**

Integrated intensity over the region defined by $[0, z]^c$ for logistic, Huesler-Reiss, Brown-Resnick and extremal Student processes.

Usage

```
expme(
  z,
  par,
  model = c("log", "neglog", "hr", "br", "xstud"),
  method = c("TruncatedNormal", "mvtnorm", "mvPot")
)
```

Arguments

<code>z</code>	vector at which to estimate exponent measure
<code>par</code>	list of parameters
<code>model</code>	string indicating the model family
<code>method</code>	string indicating the package from which to extract the numerical integration routine

Value

numeric giving the measure of the complement of $[0, z]$.

Note

The list `par` must contain different arguments depending on the model. For the Brown–Resnick model, the user must supply the conditionally negative definite matrix `Lambda` following the parametrization in Engelke *et al.* (2015) or the covariance matrix `Sigma`, following Wadsworth and Tawn (2014). For the Husler–Reiss model, the user provides the mean and covariance matrix, `m` and `Sigma`. For the extremal student, the covariance matrix `Sigma` and the degrees of freedom `df`. For the logistic model, the strictly positive dependence parameter `alpha`.

Examples

```
## Not run:
# Extremal Student
Sigma <- stats::rWishart(n = 1, df = 20, Sigma = diag(10))[, , 1]
expme(z = rep(1, ncol(Sigma)), par = list(Sigma = cov2cor(Sigma), df = 3), model = "xstud")
# Brown-Resnick model
D <- 5L
loc <- cbind(runif(D), runif(D))
```

```

di <- as.matrix(dist(rbind(c(0, ncol(loc)), loc)))
semivario <- function(d, alpha = 1.5, lambda = 1) {
  (d / lambda)^alpha
}
Vmat <- semivario(di)
Lambda <- Vmat[-1, -1] / 2
expme(z = rep(1, ncol(Lambda)), par = list(Lambda = Lambda), model = "br", method = "mvPot")
Sigma <- outer(Vmat[-1, 1], Vmat[1, -1], "+") - Vmat[-1, -1]
expme(z = rep(1, ncol(Lambda)), par = list(Lambda = Lambda), model = "br", method = "mvPot")

## End(Not run)

```

extcoef

Estimators of the extremal coefficient

Description

These functions estimate the extremal coefficient using an approximate sample from the Frechet distribution.

Usage

```

extcoef(
  xdat,
  coord = NULL,
  thresh = NULL,
  estimator = c("schlather", "smith", "fmado"),
  margtrans = c("emp", "gev", "none"),
  ties.method = "random",
  prob = 0,
  plot = TRUE,
  ...
)

```

Arguments

xdat	an n by D matrix of unit Frechet observations
coord	an optional d by D matrix of location coordinates
thresh	threshold parameter (default is to keep all data if prob = 0).
estimator	string indicating which estimator to compute, one of smith, schlather or fmado.
margtrans	string indicating which method to use to transform the margins to unit Frechet scale, either emp for nonparametric transformation via rank transform, gev for fit of generalized extreme value distribution to marginals, or none
ties.method	method for handling of ties in rank transformation
prob	probability of not exceeding threshold thresh
plot	logical; should cloud or matrix of pairwise empirical estimates be plotted? Default to TRUE.
...	additional parameters passed to the function, currently ignored.

Details

The **Smith** estimator: suppose $Z(x)$ is simple max-stable vector (i.e., with unit Frechet marginals). Then $1/Z$ is unit exponential and $1/\max(Z(s_1), Z(s_2))$ is exponential with rate $\theta = \max\{Z(s_1), Z(s_2)\}$. The extremal index for the pair can therefore be calculated using the reciprocal mean.

The **Schlather and Tawn** estimator: the likelihood of the naive estimator for a pair of two sites A is

$$\text{card} \left\{ j : \max_{i \in A} X_i^{(j)} \bar{X}_i > z \right\} \log(\theta_A) - \theta_A \sum_{j=1}^n \left[\max \left\{ z, \max_{i \in A} (X_i^{(j)} \bar{X}_i) \right\} \right]^{-1},$$

where $\bar{X}_i = n^{-1} \sum_{j=1}^n 1/X_i^{(j)}$ is the harmonic mean and z is a threshold on the unit Frechet scale. The search for the maximum likelihood estimate for every pair A is restricted to the interval $[1, 3]$. A binned version of the extremal coefficient cloud is also returned. The Schlather estimator is not self-consistent. The Schlather and Tawn estimator includes as special case the Smith estimator if we do not censor the data ($p = 0$) and do not standardize observations by their harmonic mean.

The **F-madogram** estimator is a non-parametric estimate based on a stationary process Z ; the extremal coefficient satisfies

$$\theta(h) = \frac{1 + 2\nu(h)}{1 - 2\nu(h)},$$

where

$$\nu(h) = \frac{1}{2} \mathbb{E}[|F(Z(s+h)) - F(Z(s))|]$$

The implementation only uses complete pairs to calculate the relative ranks.

All estimators are coded in plain R and computations are not optimized. The estimation time can therefore be large for large data sets. If there are no missing observations, the routine `fmadogram` from the `SpatialExtremes` package should be preferred as it is noticeably faster.

The data will typically consist of max-stable vectors or block maxima. Both of the Smith and the Schlather–Tawn estimators require unit Frechet margins; the margins will be standardized to the unit Frechet scale, either parametrically or nonparametrically unless `margtrans="none"`. If `margtrans="gev"`, a parametric GEV model is fitted to each column of `dat` using maximum likelihood estimation and transformed back using the probability integral transform. If `method="emp"`, using the empirical distribution function. The latter is the default, as it is appreciably faster.

Value

an invisible list with vectors `dist` if `coord` is non-null or else a matrix of pairwise indices `ind`, `extcoef` and the supplied estimator, `fmado` and `binned`. If `estimator == "schlather"`, an additional matrix with 2 columns containing the binned distance `binned` with the `h` and the binned extremal coefficient.

References

- Schlather, M. and J. Tawn (2003). A dependence measure for multivariate and spatial extremes, *Biometrika*, **90**(1), pp.139–156.
- Cooley, D., P. Naveau and P. Poncet (2006). Variograms for spatial max-stable random fields, In: Bertail P., Soulier P., Doukhan P. (eds) *Dependence in Probability and Statistics*. Lecture Notes in Statistics, vol. 187. Springer, New York, NY

R. J. Erhardt, R. L. Smith (2012), Approximate Bayesian computing for spatial extremes, *Computational Statistics and Data Analysis*, **56**, pp.1468–1481.

Examples

```
## Not run:
coord <- 10 * cbind(runif(50), runif(50))
di <- as.matrix(dist(coord))
dat <- rmev(
  n = 1000,
  d = 100,
  param = 3,
  sigma = exp(-di / 2),
  model = 'xstud'
)
res <- extcoef(xdat = dat, coord = coord)
# Extremal Student extremal coefficient function

XT.extcoeffun <- function(h, nu, corrfun, ...) {
  if (!is.function(corrfun)) {
    stop('Invalid function \"corrfun\".')
  }
  h <- unique(as.vector(h))
  rhoh <- sapply(h, corrfun, ...)
  cbind(
    h = h,
    extcoef = 2 * pt(sqrt((nu + 1) * (1 - rhoh) / (1 + rhoh)), nu + 1)
  )
}
#This time, only one graph with theoretical extremal coef
plot(res$dist, res$extcoef, ylim = c(1, 2), pch = 20)
abline(v = 2, col = 'gray')
extcoefxt <- XT.extcoeffun(
  seq(0, 10, by = 0.1),
  nu = 3,
  corrfun = function(x) {
    exp(-x / 2)
  }
)
lines(
  extcoefxt[, 'h'],
  extcoefxt[, 'extcoef'],
  type = 'l',
  col = 'blue',
  lwd = 2
)
# Brown--Resnick extremal coefficient function
BR.extcoeffun <- function(h, vario, ...) {
  if (!is.function(vario)) {
    stop('Invalid function \"vario\".')
  }
  h <- unique(as.vector(h))
  gammah <- sapply(h, vario, ...)
```

```

    cbind(h = h, extcoef = 2 * pnorm(sqrt(gammah / 4)))
  }
  extcoefbr <- BR.extcoeffun(
    seq(0, 20, by = 0.25),
    vario = function(x) {
      2 * x^0.7
    }
  )
  lines(
    extcoefbr[, 'h'],
    extcoefbr[, 'extcoef'],
    type = 'l',
    col = 'orange',
    lwd = 2
  )

  coord <- 10 * cbind(runif(20), runif(20))
  di <- as.matrix(dist(coord))
  dat <- rmev(
    n = 1000,
    d = 20,
    param = 3,
    sigma = exp(-di / 2),
    model = 'xstud'
  )
  res <- extcoef(
    xdat = dat,
    coord = coord,
    estimator = "smith"
  )

  ## End(Not run)

```

extgp

Extended generalised Pareto families of Naveau et al. (2016)

Description

Density function, distribution function, quantile function and random generation for the extended generalized Pareto distribution (GPD) with scale and shape parameters.

Arguments

q	vector of quantiles
x	vector of observations
p	vector of probabilities
n	sample size
prob	mixture probability for model type 4

kappa	shape parameter for type 1, 3 and 4
delta	additional parameter for type 2, 3 and 4
sigma	scale parameter
xi	shape parameter
type	integer between 0 to 5 giving the model choice
step	function of step size for discretization with default 0, corresponding to continuous quantiles
log	logical; should the log-density be returned (default to FALSE)?
unifsamp	sample of uniform; if provided, the data will be used in place of new uniform random variates
censoring	numeric vector of length 2 containing the lower and upper bound for censoring

Details

The extended generalized Pareto families proposed in Naveau *et al.* (2016) retain the tail index of the distribution while being compliant with the theoretical behavior of extreme low rainfall. There are five proposals, the first one being equivalent to the GP distribution.

- type 0 corresponds to uniform carrier, $G(u) = u$.
- type 1 corresponds to a three parameters family, with carrier $G(u) = u^\kappa$.
- type 2 corresponds to a three parameters family, with carrier $G(u) = 1 - V_\delta((1 - u)^\delta)$.
- type 3 corresponds to a four parameters family, with carrier

$$G(u) = 1 - V_\delta((1 - u)^\delta))^{\kappa/2}$$

- type 4 corresponds to a five parameter model (a mixture of type 2, with $G(u) = pu^\kappa + (1 - p) * u^\delta$

Usage

```
pextgp(q, prob=NA, kappa=NA, delta=NA, sigma=NA, xi=NA, type=1)
dextgp(x, prob=NA, kappa=NA, delta=NA, sigma=NA, xi=NA, type=1, log=FALSE)
qextgp(p, prob=NA, kappa=NA, delta=NA, sigma=NA, xi=NA, type=1)
rextgp(n, prob=NA, kappa=NA, delta=NA, sigma=NA, xi=NA, type=1, unifsamp=NULL, censoring=c(0,Inf))
```

Author(s)

Raphael Huser and Philippe Naveau

References

Naveau, P., R. Huser, P. Ribereau, and A. Hannart (2016), Modeling jointly low, moderate, and heavy rainfall intensities without a threshold selection, *Water Resour. Res.*, 52, 2753-2769, doi : 10.1002/2015WR018552.

extgp.G

Carrier distribution for the extended GP distributions of Naveau et al.

Description

Density, distribution function, quantile function and random number generation for the carrier distributions of the extended Generalized Pareto distributions.

Arguments

u	vector of observations (dextgp.G), probabilities (qextgp.G) or quantiles (pextgp.G), in $[0, 1]$
prob	mixture probability for model type 4
kappa	shape parameter for type 1, 3 and 4
delta	additional parameter for type 2, 3 and 4
type	integer between 0 to 5 giving the model choice
log	logical; should the log-density be returned (default to FALSE)?
n	sample size
unifsamp	sample of uniform; if provided, the data will be used in place of new uniform random variates
censoring	numeric vector of length 2 containing the lower and upper bound for censoring
direct	logical; which method to use for sampling in model of type 4?

Usage

```
pextgp.G(u, type=1, prob, kappa, delta)
dextgp.G(u, type=1, prob=NA, kappa=NA, delta=NA, log=FALSE)
qextgp.G(u, type=1, prob=NA, kappa=NA, delta=NA)
rextgp.G(n, prob=NA, kappa=NA, delta=NA, type=1, unifsamp=NULL, direct=FALSE, censoring=c(0,1))
```

Author(s)

Raphael Huser and Philippe Naveau

See Also

[extgp](#)

fit.egp

Parameter stability plot and maximum likelihood routine for extended GP models

Description

The function `tstab.egp` provides classical threshold stability plot for (κ, σ, ξ) . The fitted parameter values are displayed with pointwise normal 95% confidence intervals. The function returns an invisible list with parameter estimates and standard errors, and p-values for the Wald test that $\kappa = 1$. The plot is for the modified scale (as in the generalised Pareto model) and as such it is possible that the modified scale be negative. `tstab.egp` can also be used to fit the model to multiple thresholds.

Usage

```
fit.egp(
  xdat,
  thresh = 0,
  model = c("pt-beta", "pt-gamma", "pt-power", "gj-tnorm", "gj-beta", "exptilt",
    "logist"),
  start = NULL,
  method = c("Nelder", "nlminb", "BFGS"),
  fpar = NULL,
  show = FALSE,
  ...
)
```

Arguments

<code>xdat</code>	vector of observations, greater than the threshold
<code>thresh</code>	threshold value
<code>model</code>	a string indicating which extended family to fit
<code>start</code>	optional named list of initial values, with κ , <i>sigma</i> or <i>xi</i> .
<code>method</code>	the method to be used. See Details . Can be abbreviated.
<code>fpar</code>	a named list with fixed parameters, either scale or shape
<code>show</code>	logical; if TRUE, print the results of the optimization
<code>...</code>	additional parameters, for backward compatibility purposes

Details

`fit.egp` is a numerical optimization routine to fit the extended generalised Pareto models of Papastathopoulos and Tawn (2013), using maximum likelihood estimation.

Value

`fit.egp` outputs the list returned by `optim`, which contains the parameter values, the hessian and in addition the standard errors

`tstab.egp` returns a plot(s) of the parameters fit over the range of provided thresholds, with point-wise normal confidence intervals; the function also returns an invisible list containing notably the matrix of point estimates (`par`) and standard errors (`se`).

Author(s)

Leo Belzile

References

Papastathopoulos, I. and J. Tawn (2013). Extended generalised Pareto models for tail estimation, *Journal of Statistical Planning and Inference* **143**(3), 131–143.

Examples

```
xdat <- mev::rgp(
  n = 100,
  loc = 0,
  scale = 1,
  shape = 0.5)
fitted <- fit.egp(
  xdat = xdat,
  thresh = 1,
  model = "pt-gamma",
  show = TRUE)
thresh <- mev::qgp(seq(0.1, 0.5, by = 0.05), 0, 1, 0.5)
tstab.egp(
  xdat = xdat,
  thresh = thresh,
  model = "pt-gamma")
xdat <- regp(
  n = 100,
  scale = 1,
  shape = 0.1,
  kappa = 0.5,
  model = "pt-power"
)
fit.egp(
  xdat = xdat,
  model = "pt-power",
  show = TRUE,
  fpar = list(kappa = 1),
  method = "Nelder"
)
```

fit.extgp

Fit an extended generalized Pareto distribution of Naveau et al.

Description

This is a wrapper function to obtain PWM or MLE estimates for the extended GP models of Naveau et al. (2016) for rainfall intensities. The function calculates confidence intervals by means of nonparametric percentile bootstrap and returns histograms and QQ plots of the fitted distributions. The function handles both censoring and rounding.

Usage

```
fit.extgp(
  data,
  model = 1,
  method = c("mle", "pwm"),
  init,
  censoring = c(0, Inf),
  rounded = 0,
  confint = FALSE,
  R = 1000,
  ncpus = 1,
  plots = TRUE
)
```

Arguments

data	data vector.
model	integer ranging from 0 to 4 indicating the model to select (see extgp).
method	string; either 'mle' for maximum likelihood, or 'pwm' for probability weighted moments, or both.
init	vector of initial values, comprising of p , κ , δ , σ , ξ (in that order) for the optimization. All parameters may not appear depending on model.
censoring	numeric vector of length 2 containing the lower and upper bound for censoring; censoring=c(0, Inf) is equivalent to no censoring.
rounded	numeric giving the instrumental precision (and rounding of the data), with default of 0.
confint	logical; should confidence interval be returned (percentile bootstrap).
R	integer; number of bootstrap replications.
ncpus	integer; number of CPUs for parallel calculations (default: 1).
plots	logical; whether to produce histogram and density plots.

Details

The different models include the following transformations:

- model 0 corresponds to uniform carrier, $G(u) = u$.
- model 1 corresponds to a three parameters family, with carrier $G(u) = u^\kappa$.
- model 2 corresponds to a three parameters family, with carrier $G(u) = 1 - V_\delta((1 - u)^\delta)$.
- model 3 corresponds to a four parameters family, with carrier

$$G(u) = 1 - V_\delta((1 - u)^\delta))^{\kappa/2}$$

.

- model 4 corresponds to a five parameter model (a mixture of type 2, with $G(u) = pu^\kappa + (1 - p) * u^\delta$

Author(s)

Raphael Huser and Philippe Naveau

References

Naveau, P., R. Huser, P. Ribereau, and A. Hannart (2016), Modeling jointly low, moderate, and heavy rainfall intensities without a threshold selection, *Water Resour. Res.*, 52, 2753-2769, doi : 10.1002/2015WR018552.

See Also

[egp.fit](#), [egp](#), [extgp](#)

Examples

```
## Not run:
data(rain, package = "ismev")
fit.extgp(
  rain[rain > 0],
  model = 1,
  method = 'mle',
  init = c(0.9, fit.gpd(rain)$est),
  rounded = 0.1,
  confint = TRUE,
  R = 20
)

## End(Not run)
```

fit.gev	<i>Maximum likelihood estimation for the generalized extreme value distribution</i>
---------	---

Description

This function returns an object of class `mev_gev`, with default methods for printing and quantile-quantile plots. The default starting values are the solution of the probability weighted moments.

Usage

```
fit.gev(
  xdat,
  start = NULL,
  method = c("nllminb", "BFGS"),
  show = FALSE,
  fpar = NULL,
  warnSE = FALSE
)
```

Arguments

<code>xdat</code>	a numeric vector of data to be fitted.
<code>start</code>	named list of starting values
<code>method</code>	string indicating the outer optimization routine for the augmented Lagrangian. One of <code>nllminb</code> or <code>BFGS</code> .
<code>show</code>	logical; if TRUE (the default), print details of the fit.
<code>fpar</code>	a named list with optional fixed components <code>loc</code> , <code>scale</code> and <code>shape</code>
<code>warnSE</code>	logical; if TRUE, a warning is printed if the standard errors cannot be returned from the observed information matrix when the shape is less than -0.5.

Value

a list containing the following components:

- `estimate` a vector containing the maximum likelihood estimates.
- `std.err` a vector containing the standard errors.
- `vcov` the variance covariance matrix, obtained as the numerical inverse of the observed information matrix.
- `method` the method used to fit the parameter.
- `nllh` the negative log-likelihood evaluated at the parameter estimate.
- `convergence` components taken from the list returned by [auglag](#). Values other than 0 indicate that the algorithm likely did not converge.
- `counts` components taken from the list returned by [auglag](#).
- `xdat` vector of data

Examples

```
xdat <- mev::rgev(n = 100)
fit.gev(xdat, show = TRUE)
# Example with fixed parameter
fit.gev(xdat, show = TRUE, fpar = list(shape = 0))
```

fit.gpd

Maximum likelihood estimation for the generalized Pareto distribution

Description

Numerical optimization of the generalized Pareto distribution for data exceeding threshold. This function returns an object of class `mev_gpd`, with default methods for printing and quantile-quantile plots.

Usage

```
fit.gpd(
  xdat,
  threshold = 0,
  method = "Grimshaw",
  show = FALSE,
  MCMC = NULL,
  k = 4,
  tol = 1e-08,
  fpar = NULL,
  warnSE = FALSE
)
```

Arguments

<code>xdat</code>	a numeric vector of data to be fitted.
<code>threshold</code>	the chosen threshold.
<code>method</code>	the method to be used. See Details . Can be abbreviated.
<code>show</code>	logical; if TRUE (the default), print details of the fit.
<code>MCMC</code>	NULL for frequentist estimates, otherwise a boolean or a list with parameters passed. If TRUE, runs a Metropolis-Hastings sampler to get posterior mean estimates. Can be used to pass arguments <code>niter</code> , <code>burnin</code> and <code>thin</code> to the sampler as a list.
<code>k</code>	bound on the influence function (<code>method = "obre"</code>); the constant <code>k</code> is a robustness parameter (higher bounds are more efficient, low bounds are more robust). Default to 4, must be larger than $\sqrt{2}$.
<code>tol</code>	numerical tolerance for OBRE weights iterations (<code>method = "obre"</code>). Default to $1e-8$.
<code>fpar</code>	a named list with fixed parameters, either scale or shape
<code>warnSE</code>	logical; if TRUE, a warning is printed if the standard errors cannot be returned from the observed information matrix when the shape is less than -0.5.

Details

The default method is 'Grimshaw', which maximizes the profile likelihood for the ratio scale/shape. Other options include 'obre' for optimal B -robust estimator of the parameter of Dupuis (1998), vanilla maximization of the log-likelihood using constrained optimization routine 'auglag', 1-dimensional optimization of the profile likelihood using `nlm` and `optim`. Method 'isnev' performs the two-dimensional optimization routine `gpd.fit` from the `isnev` library, with in addition the algebraic gradient. The approximate Bayesian methods ('zs' and 'zhang') are extracted respectively from Zhang and Stephens (2009) and Zhang (2010) and consists of a approximate posterior mean calculated via importance sampling assuming a GPD prior is placed on the parameter of the profile likelihood.

Value

If method is neither 'zs' nor 'zhang', a list containing the following components:

- `estimate` a vector containing the scale and shape parameters (optimized and fixed).
- `std.err` a vector containing the standard errors. For method = "obre", these are Huber's robust standard errors.
- `vcov` the variance covariance matrix, obtained as the numerical inverse of the observed information matrix. For method = "obre", this is the sandwich Godambe matrix inverse.
- `threshold` the threshold.
- `method` the method used to fit the parameter. See details.
- `nllh` the negative log-likelihood evaluated at the parameter estimate.
- `nat` number of points lying above the threshold.
- `pat` proportion of points lying above the threshold.
- `convergence` components taken from the list returned by `optim`. Values other than 0 indicate that the algorithm likely did not converge (in particular 1 and 50).
- `counts` components taken from the list returned by `optim`.
- `exceedances` excess over the threshold.

Additionally, if method = "obre", a vector of OBRE weights.

Otherwise, a list containing

- `threshold` the threshold.
- `method` the method used to fit the parameter. See **Details**.
- `nat` number of points lying above the threshold.
- `pat` proportion of points lying above the threshold.
- `approx.mean` a vector containing the approximate posterior mean estimates.

and in addition if MCMC is neither FALSE, nor NULL

- `post.mean` a vector containing the posterior mean estimates.
- `post.se` a vector containing the posterior standard error estimates.
- `accept.rate` proportion of points lying above the threshold.
- `niter` length of resulting Markov Chain
- `burnin` amount of discarded iterations at start, capped at 10000.
- `thin` thinning integer parameter describing

Note

Some of the internal functions (which are hidden from the user) allow for modelling of the parameters using covariates. This is not currently implemented within `gp.fit`, but users can call internal functions should they wish to use these features.

Author(s)

Scott D. Grimshaw for the Grimshaw option. Paul J. Northrop and Claire L. Coleman for the methods `optim`, `nlm` and `ismev`. J. Zhang and Michael A. Stephens (2009) and Zhang (2010) for the `zs` and `zhang` approximate methods and L. Belzile for methods `auglag` and `obre`, the wrapper and MCMC samplers.

If `show = TRUE`, the optimal B robust estimated weights for the largest observations are printed alongside with the p -value of the latter, obtained from the empirical distribution of the weights. This diagnostic can be used to guide threshold selection: small weights for the r -largest order statistics indicate that the robust fit is driven by the lower tail and that the threshold should perhaps be increased.

References

- Davison, A.C. (1984). Modelling excesses over high thresholds, with an application, in *Statistical extremes and applications*, J. Tiago de Oliveira (editor), D. Reidel Publishing Co., 461–482.
- Grimshaw, S.D. (1993). Computing Maximum Likelihood Estimates for the Generalized Pareto Distribution, *Technometrics*, **35**(2), 185–191.
- Northrop, P.J. and C. L. Coleman (2014). Improved threshold diagnostic plots for extreme value analyses, *Extremes*, **17**(2), 289–303.
- Zhang, J. (2010). Improving on estimation for the generalized Pareto distribution, *Technometrics* **52**(3), 335–339.
- Zhang, J. and M. A. Stephens (2009). A new and efficient estimation method for the generalized Pareto distribution. *Technometrics* **51**(3), 316–325.
- Dupuis, D.J. (1998). Exceedances over High Thresholds: A Guide to Threshold Selection, *Extremes*, **1**(3), 251–261.

See Also

[fpot](#) and [gpd.fit](#)

Examples

```
data(eskrain)
fit.gpd(eskrain, threshold = 35, method = 'Grimshaw', show = TRUE)
fit.gpd(eskrain, threshold = 30, method = 'zs', show = TRUE)
```

fit.pp

*Maximum likelihood estimation of the point process of extremes***Description**

Data above threshold is modelled using the limiting point process of extremes.

Usage

```
fit.pp(
  xdat,
  threshold = 0,
  npp = 1,
  np = NULL,
  method = c("nllminb", "BFGS"),
  start = NULL,
  show = FALSE,
  fpar = NULL,
  warnSE = FALSE
)
```

Arguments

xdat	a numeric vector of data to be fitted.
threshold	the chosen threshold.
npp	number of observation per period. See Details
np	number of periods of data, if xdat only contains exceedances.
method	the method to be used. See Details . Can be abbreviated.
start	named list of starting values
show	logical; if TRUE (the default), print details of the fit.
fpar	a named list with optional fixed components loc, scale and shape
warnSE	logical; if TRUE, a warning is printed if the standard errors cannot be returned from the observed information matrix when the shape is less than -0.5.

Details

The parameter npp controls the frequency of observations. If data are recorded on a daily basis, using a value of npp = 365.25 yields location and scale parameters that correspond to those of the generalized extreme value distribution fitted to block maxima.

Value

a list containing the following components:

- `estimate` a vector containing all parameters (optimized and fixed).
- `std.err` a vector containing the standard errors.
- `vcov` the variance covariance matrix, obtained as the numerical inverse of the observed information matrix.
- `threshold` the threshold.
- `method` the method used to fit the parameter. See details.
- `nllh` the negative log-likelihood evaluated at the parameter estimate.
- `nat` number of points lying above the threshold.
- `pat` proportion of points lying above the threshold.
- `convergence` components taken from the list returned by `optim`. Values other than 0 indicate that the algorithm likely did not converge (in particular 1 and 50).
- `counts` components taken from the list returned by `optim`.

References

Coles, S. (2001), An introduction to statistical modelling of extreme values. Springer : London, 208p.

Examples

```
data(eskrain)
pp_mle <- fit.pp(eskrain, threshold = 30, np = 6201)
plot(pp_mle)
```

fit.rho

Estimator of the second order tail index parameter

Description

Estimator of the second order tail index parameter

Usage

```
fit.rho(xdat, k, method = c("fagh", "dk", "ghp", "gbw"), ...)
```

Arguments

<code>xdat</code>	vector of positive observations
<code>k</code>	number of highest order statistics to use for estimation
<code>method</code>	string for the estimator
<code>...</code>	additional arguments passed to individual routines currently ignored.

Examples

```
# Example with rho = -0.2
n <- 1000
xdat <- mev::rgp(n = n, shape = 0.2)
kmin <- floor(n^0.995)
kmax <- ceiling(n^0.999)
rho_est <- fit.rho(
  xdat = xdat,
  k = n - kmin:kmax)
rho_med <- mean(rho_est$rho)
```

fit.rlarg	<i>Maximum likelihood estimates of point process for the r-largest observations</i>
-----------	---

Description

This uses a constrained optimization routine to return the maximum likelihood estimate based on an n by r matrix of observations. Observations should be ordered, i.e., the r -largest should be in the last column.

Usage

```
fit.rlarg(
  xdat,
  start = NULL,
  method = c("nlnmb", "BFGS"),
  show = FALSE,
  fpar = NULL,
  warnSE = FALSE
)
```

Arguments

xdat	a numeric vector of data to be fitted.
start	named list of starting values
method	the method to be used. See Details . Can be abbreviated.
show	logical; if TRUE (the default), print details of the fit.
fpar	a named list with fixed parameters, either scale or shape
warnSE	logical; if TRUE, a warning is printed if the standard errors cannot be returned from the observed information matrix when the shape is less than -0.5.

Value

a list containing the following components:

- `estimate` a vector containing all the maximum likelihood estimates.
- `std.err` a vector containing the standard errors.
- `vcov` the variance covariance matrix, obtained as the numerical inverse of the observed information matrix.
- `method` the method used to fit the parameter.
- `nllh` the negative log-likelihood evaluated at the parameter estimate.
- `convergence` components taken from the list returned by [auglag](#). Values other than 0 indicate that the algorithm likely did not converge.
- `counts` components taken from the list returned by [auglag](#).
- `xdat` an n by r matrix of data

Examples

```
xdat <- rrlarg(n = 10, loc = 0, scale = 1, shape = 0.1, r = 4)
fit.rlarg(xdat)
```

fit.shape

Shape parameter estimates

Description

Wrapper to estimate the tail index or shape parameter of an extreme value distribution. Each function has similar sets of arguments, a vector or scalar number of order statistics k and a vector of positive observations `xdat`. The `method` argument allows users to choose between different indicators, including the Hill estimator (`hill`, for positive observations and shape only), the moment estimator of Dekkers and de Haan (`mom` or `dekkers`), the de Vries estimator of de Haan and Peng (`vries`), the generalized jackknife estimator of Gomes et al. (`genjack`), the Beirlant, Vynckier and Teugels generalized quantile estimator (`bvt` or `genquant`), the Pickands estimator (`pickands`), the extreme U -statistics estimator of Oorschot, Segers and Zhou (`osz`), or the exponential regression model of Beirlant et al. (`erm`).

Usage

```
fit.shape(
  xdat,
  k,
  method = c("hill", "rbm", "osz", "vries", "genjack", "mom", "dekkers", "genquant",
    "pickands", "erm"),
  ...
)
```

Arguments

xdat	vector of positive observations of length n
k	number of largest order statistics
method	estimation method.
...	additional parameters passed to functions

Value

a data frame with the number of order statistics k and the shape parameter estimate `shape`, or a single numeric value if k is a scalar.

<code>fit.wgpd</code>	<i>Maximum likelihood estimation for weighted generalized Pareto distribution</i>
-----------------------	---

Description

Weighted maximum likelihood estimation, with user-specified vector of weights.

Usage

```
fit.wgpd(xdat, threshold = 0, weightfun = Stein_weights, start = NULL, ...)
```

Arguments

xdat	vector of observations
threshold	numeric, value of the threshold
weightfun	function whose first argument is the length of the weight vector
start	optional vector of scale and shape parameters for the optimization routine, defaults to NULL
...	additional arguments passed to the weighting function <code>weightfun</code>

Value

a list with components

- `estimate` a vector containing the scale and shape parameters (optimized and fixed).
- `std.err` a vector containing the standard errors.
- `vcov` the variance covariance matrix, obtained as the numerical inverse of the observed information matrix.
- `threshold` the threshold.
- `method` the method used to fit the parameter. See details.
- `nllh` the negative log-likelihood evaluated at the parameter estimate.
- `nat` number of points lying above the threshold.

- pat proportion of points lying above the threshold.
- convergence logical indicator of convergence.
- weights vector of weights for exceedances.
- exceedances excess over the threshold, sorted in decreasing order.

frwind

French wind data

Description

Daily mean wind speed (in km/h) at four stations in the south of France, namely Cap Cepet (S1), Lyon St-Exupery (S2), Marseille Marignane (S3) and Montelimar (S4). The data includes observations from January 1976 until April 2023; days containing missing values are omitted.

Format

A data frame with 17209 observations and 8 variables:

date date of measurement

S1 wind speed (in km/h) at Cap Cepet

S2 wind speed (in km/h) at Lyon Saint-Exupery

S3 wind speed (in km/h) at Marseille Marignane

S4 wind speed (in km/h) at Montelimar

H2 humidity (in percentage) at Lyon Saint-Exupery

T2 mean temperature (in degree Celcius) at Lyon Saint-Exupery

The metadata attribute includes latitude and longitude (in degrees, minutes, seconds), altitude (in m), station name and station id.

Source

European Climate Assessment and Dataset project <https://www.ecad.eu/>

References

Klein Tank, A.M.G. and Coauthors, 2002. Daily dataset of 20th-century surface air temperature and precipitation series for the European Climate Assessment. Int. J. of Climatol., 22, 1441-1453.

Examples

```
data(frwind, package = "mev")
head(frwind)
attr(frwind, which = "metadata")
```

geomagnetic	<i>Magnetic storms</i>
-------------	------------------------

Description

Absolute magnitude of 373 geomagnetic storms lasting more than 48h with absolute magnitude (dst) larger than 100 in 1957-2014.

Format

a vector of size 373

Note

For a detailed article presenting the derivation of the Dst index, see <http://wdc.kugi.kyoto-u.ac.jp/dstdir/dst2/onDst>

Source

Aki Vehtari

References

World Data Center for Geomagnetism, Kyoto, M. Nose, T. Iyemori, M. Sugiura, T. Kamei (2015), *Geomagnetic Dst index*, <doi:10.17593/14515-74000>.

gev	<i>Generalized extreme value distribution</i>
-----	---

Description

Likelihood, score function and information matrix, bias, approximate ancillary statistics and sample space derivative for the generalized extreme value distribution

Arguments

par	vector of loc, scale and shape
dat	sample vector
method	string indicating whether to use the expected ('exp') or the observed ('obs' - the default) information matrix.
V	vector calculated by gev.Vfun
n	sample size
p	vector of probabilities

Usage

```
gev.ll(par, dat)
gev.ll.optim(par, dat)
gev.score(par, dat)
gev.infomat(par, dat, method = c('obs', 'exp'))
gev.retlev(par, p)
gev.bias(par, n)
gev.Fscore(par, dat, method=c('obs', 'exp'))
gev.Vfun(par, dat)
gev.phi(par, dat, V)
gev.dphi(par, dat, V)
```

Functions

- `gev.ll`: log likelihood
- `gev.ll.optim`: negative log likelihood parametrized in terms of location, $\log(\text{scale})$ and shape in order to perform unconstrained optimization
- `gev.score`: score vector
- `gev.infomat`: observed or expected information matrix
- `gev.retlev`: return level, corresponding to the $(1 - p)$ th quantile
- `gev.bias`: Cox-Snell first order bias
- `gev.Fscore`: Firth's modified score equation
- `gev.Vfun`: vector implementing conditioning on approximate ancillary statistics for the TEM
- `gev.phi`: canonical parameter in the local exponential family approximation
- `gev.dphi`: derivative matrix of the canonical parameter in the local exponential family approximation

References

- Firth, D. (1993). Bias reduction of maximum likelihood estimates, *Biometrika*, **80**(1), 27–38.
- Coles, S. (2001). *An Introduction to Statistical Modeling of Extreme Values*, Springer, 209 p.
- Cox, D. R. and E. J. Snell (1968). A general definition of residuals, *Journal of the Royal Statistical Society: Series B (Methodological)*, **30**, 248–275.
- Cordeiro, G. M. and R. Klein (1994). Bias correction in ARMA models, *Statistics and Probability Letters*, **19**(3), 169–176.

 gev.abias

Asymptotic bias of block maxima for fixed sample sizes

Description

Asymptotic bias of block maxima for fixed sample sizes

Usage

```
gev.abias(shape, rho)
```

Arguments

shape	shape parameter
rho	second-order parameter, non-positive

Value

a vector of length three containing the bias for location, scale and shape (in this order)

References

Dombry, C. and A. Ferreira (2017). Maximum likelihood estimators based on the block maxima method. <https://arxiv.org/abs/1705.00465>

 gev.bcor

Bias correction for GEV distribution

Description

Bias corrected estimates for the generalized extreme value distribution using Firth's modified score function or implicit bias subtraction.

Usage

```
gev.bcor(par, dat, corr = c("subtract", "firth"), method = c("obs", "exp"))
```

Arguments

par	parameter vector (scale, shape)
dat	sample of observations
corr	string indicating which correction to employ either subtract or firth
method	string indicating whether to use the expected ('exp') or the observed ('obs' — the default) information matrix. Used only if corr='firth'

Details

Method subtractsolves

$$\tilde{\theta} = \hat{\theta} + b(\tilde{\theta})$$

for $\tilde{\theta}$, using the first order term in the bias expansion as given by `gev.bias`.

The alternative is to use Firth's modified score and find the root of

$$U(\tilde{\theta}) - i(\tilde{\theta})b(\tilde{\theta}),$$

where U is the score vector, b is the first order bias and i is either the observed or Fisher information.

The routine uses the MLE (bias-corrected) as starting values and proceeds to find the solution using a root finding algorithm. Since the bias-correction is not valid for $\xi < -1/3$, any solution that is unbounded will return a vector of NA as the solution does not exist then.

Value

vector of bias-corrected parameters

Examples

```
set.seed(1)
dat <- mev::rgev(n=40, loc = 1, scale=1, shape=-0.2)
par <- mev::fit.gev(dat)$estimate
gev.bcor(par, dat, 'subtract')
gev.bcor(par, dat, 'firth') #observed information
gev.bcor(par, dat, 'firth','exp')
```

gev.mle	<i>Generalized extreme value maximum likelihood estimates for various quantities of interest</i>
---------	--

Description

This function calls the `fit.gev` routine on the sample of block maxima and returns maximum likelihood estimates for all quantities of interest, including location, scale and shape parameters, quantiles and mean and quantiles of maxima of N blocks.

Usage

```
gev.mle(
  xdat,
  args = c("loc", "scale", "shape", "quant", "Nmean", "Nquant"),
  N,
  p,
  q
)
```

Arguments

xdat	sample vector of maxima
args	vector of strings indicating which arguments to return the maximum likelihood values for.
N	size of block over which to take maxima. Required only for args Nmean and Nquant.
p	tail probability. Required only for arg quant.
q	level of quantile for maxima of N exceedances. Required only for args Nquant.

Value

named vector with maximum likelihood estimated parameter values for arguments args

Examples

```
dat <- mev::rgev(n = 100, shape = 0.2)
gev.mle(xdat = dat, N = 100, p = 0.01, q = 0.5)
```

gev.Nyr

N-year return levels, median and mean estimate

Description

N-year return levels, median and mean estimate

Usage

```
gev.Nyr(par, nobs, N, type = c("retlev", "median", "mean"), p = 1/N)
```

Arguments

par	vector of location, scale and shape parameters for the GEV distribution
nobs	integer number of observation on which the fit is based
N	integer number of observations for return level. See Details
type	string indicating the statistic to be calculated (can be abbreviated).
p	probability indicating the return level, corresponding to the quantile at $1-1/p$

Details

If there are n_y observations per year, the L-year return level is obtained by taking N equal to $n_y L$.

Value

a list with components

- est point estimate
- var variance estimate based on delta-method
- type statistic

gev.pll

Profile log-likelihood for the generalized extreme value distribution

Description

This function calculates the profile likelihood along with two small-sample corrections based on Severini's (1999) empirical covariance and the Fraser and Reid tangent exponential model approximation.

Usage

```
gev.pll(
  psi,
  param = c("loc", "scale", "shape", "quant", "Nmean", "Nquant"),
  mod = "profile",
  dat,
  N = NULL,
  p = NULL,
  q = NULL,
  correction = TRUE,
  plot = TRUE,
  ...
)
```

Arguments

psi	parameter vector over which to profile (unidimensional)
param	string indicating the parameter to profile over
mod	string indicating the model, one of profile, tem or modif. See Details .
dat	sample vector
N	size of block over which to take maxima. Required only for param Nmean and Nquant.
p	tail probability. Required only for param quant.
q	probability level of quantile. Required only for param Nquant.
correction	logical indicating whether to use spline.corr to smooth the tem approximation.
plot	logical; should the profile likelihood be displayed? Default to TRUE
...	additional arguments such as output from call to Vfun if mode='tem'.

Details

The two additional mod available are `tem`, the tangent exponential model (TEM) approximation and `modif` for the penalized profile likelihood based on p^* approximation proposed by Severini. For the latter, the penalization is based on the TEM or an empirical covariance adjustment term.

Value

a list with components

- `mle`: maximum likelihood estimate
- `psi.max`: maximum profile likelihood estimate
- `param`: string indicating the parameter to profile over
- `std.error`: standard error of `psi.max`
- `psi`: vector of parameter ψ given in `psi`
- `pll`: values of the profile log likelihood at `psi`
- `maxpll`: value of maximum profile log likelihood

In addition, if `mod` includes `tem`

- `normal`: maximum likelihood estimate and standard error of the interest parameter ψ
- `r`: values of likelihood root corresponding to ψ
- `q`: vector of likelihood modifications
- `rstar`: modified likelihood root vector
- `rstar.old`: uncorrected modified likelihood root vector
- `tem.psimax`: maximum of the tangent exponential model likelihood

In addition, if `mod` includes `modif`

- `tem.mle`: maximum of tangent exponential modified profile log likelihood
- `tem.profl1`: values of the modified profile log likelihood at `psi`
- `tem.maxpll`: value of maximum modified profile log likelihood
- `empcov.mle`: maximum of Severini's empirical covariance modified profile log likelihood
- `empcov.profl1`: values of the modified profile log likelihood at `psi`
- `empcov.maxpll`: value of maximum modified profile log likelihood

References

- Fraser, D. A. S., Reid, N. and Wu, J. (1999), A simple general formula for tail probabilities for frequentist and Bayesian inference. *Biometrika*, **86**(2), 249–264.
- Severini, T. (2000) Likelihood Methods in Statistics. Oxford University Press. ISBN 9780198506508.
- Brazzale, A. R., Davison, A. C. and Reid, N. (2007) Applied asymptotics: case studies in small-sample statistics. Cambridge University Press, Cambridge. ISBN 978-0-521-84703-2

Examples

```
## Not run:
set.seed(123)
dat <- rgev(n = 100, loc = 0, scale = 2, shape = 0.3)
gev.pll(psi = seq(0,0.5, length = 50), param = 'shape', dat = dat)
gev.pll(psi = seq(-1.5, 1.5, length = 50), param = 'loc', dat = dat)
gev.pll(psi = seq(10, 40, length = 50), param = 'quant', dat = dat, p = 0.01)
gev.pll(psi = seq(12, 100, length = 50), param = 'Nmean', N = 100, dat = dat)
gev.pll(psi = seq(12, 90, length = 50), param = 'Nquant', N = 100, dat = dat, q = 0.5)

## End(Not run)
```

 gev.tem

Tangent exponential model approximation for the GEV distribution

Description

The function `gev.tem` provides a tangent exponential model (TEM) approximation for higher order likelihood inference for a scalar parameter for the generalized extreme value distribution. Options include location scale and shape parameters as well as value-at-risk (or return levels). The function attempts to find good values for `psi` that will cover the range of options, but the fail may fit and return an error.

Usage

```
gev.tem(
  param = c("loc", "scale", "shape", "quant", "Nmean", "Nquant"),
  dat,
  psi = NULL,
  p = NULL,
  q = 0.5,
  N = NULL,
  n.psi = 50,
  plot = TRUE,
  correction = TRUE
)
```

Arguments

<code>param</code>	parameter over which to profile
<code>dat</code>	sample vector for the GEV distribution
<code>psi</code>	scalar or ordered vector of values for the interest parameter. If <code>NULL</code> (default), a grid of values centered at the MLE is selected
<code>p</code>	tail probability for the (1-p)th quantile (return levels). Required only if <code>param = 'retlev'</code>
<code>q</code>	probability level of quantile. Required only for <code>param Nquant</code> .

N	size of block over which to take maxima. Required only for param Nmean and Nquant.
n.psi	number of values of psi at which the likelihood is computed, if psi is not supplied (NULL). Odd values are more prone to give rise to numerical instabilities near the MLE. If psi is a vector of length 2 and n.psi is greater than 2, these are taken to be endpoints of the sequence.
plot	logical indicating whether plot.fr should be called upon exit
correction	logical indicating whether spline.corr should be called.

Value

an invisible object of class fr (see tem in package hoa) with elements

- normal: maximum likelihood estimate and standard error of the interest parameter ψ
- par.hat: maximum likelihood estimates
- par.hat.se: standard errors of maximum likelihood estimates
- th.rest: estimated maximum profile likelihood at $(\psi, \hat{\lambda})$
- r: values of likelihood root corresponding to ψ
- psi: vector of interest parameter
- q: vector of likelihood modifications
- rstar: modified likelihood root vector
- rstar.old: uncorrected modified likelihood root vector
- param: parameter

Author(s)

Leo Belzile

Examples

```
## Not run:
set.seed(1234)
dat <- rgev(n = 40, loc = 0, scale = 2, shape = -0.1)
gev.tem('shape', dat = dat, plot = TRUE)
gev.tem('quant', dat = dat, p = 0.01, plot = TRUE)
gev.tem('scale', psi = seq(1, 4, by = 0.1), dat = dat, plot = TRUE)
dat <- rgev(n = 40, loc = 0, scale = 2, shape = 0.2)
gev.tem('loc', dat = dat, plot = TRUE)
gev.tem('Nmean', dat = dat, p = 0.01, N=100, plot = TRUE)
gev.tem('Nquant', dat = dat, q = 0.5, N=100, plot = TRUE)

## End(Not run)
```

gevdist

*Generalized extreme value distribution***Description**

Density function, distribution function, quantile function and random number generation for the generalized extreme value distribution.

Usage

```
qgev(p, loc = 0, scale = 1, shape = 0, lower.tail = TRUE, log.p = FALSE)
```

```
rgev(n, loc = 0, scale = 1, shape = 0)
```

```
dgev(x, loc = 0, scale = 1, shape = 0, log = FALSE)
```

```
pgev(q, loc = 0, scale = 1, shape = 0, lower.tail = TRUE, log.p = FALSE)
```

Arguments

<code>p</code>	vector of probabilities
<code>loc</code>	scalar or vector of location parameters whose length matches that of the input
<code>scale</code>	scalar or vector of positive scale parameters whose length matches that of the input
<code>shape</code>	scalar shape parameter
<code>lower.tail</code>	logical; if TRUE (default), returns the distribution function, otherwise the survival function
<code>n</code>	scalar number of observations
<code>x, q</code>	vector of quantiles
<code>log, log.p</code>	logical; if TRUE, probabilities p are given as $\log(p)$.

Details

The distribution function of a GEV distribution with parameters $\text{loc} = \mu$, $\text{scale} = \sigma$ and $\text{shape} = \xi$ is

$$F(x) = \exp\{-[1 + \xi(x - \mu)/\sigma]^{-1/\xi}\}$$

for $1 + \xi(x - \mu)/\sigma > 0$. If $\xi = 0$ the distribution function is defined as the limit as ξ tends to zero.

The quantile function, when evaluated at zero or one, returns the lower and upper endpoint, whether the latter is finite or not.

Author(s)

Leo Belzile, with code adapted from Paul Northrop

References

- Jenkinson, A. F. (1955) The frequency distribution of the annual maximum (or minimum) of meteorological elements. *Quart. J. R. Met. Soc.*, **81**, 158-171. Chapter 3: [doi:10.1002/qj.49708134804](https://doi.org/10.1002/qj.49708134804)
- Coles, S. G. (2001) *An Introduction to Statistical Modeling of Extreme Values*, Springer-Verlag, London. [doi:10.1007/9781447136750_3](https://doi.org/10.1007/9781447136750_3)

gevN	<i>Generalized extreme value distribution (quantile/mean of N-block maxima parametrization)</i>
------	---

Description

Likelihood, score function and information matrix, approximate ancillary statistics and sample space derivative for the generalized extreme value distribution parametrized in terms of the quantiles/mean of N-block maxima parametrization z , scale and shape.

Arguments

par	vector of loc, quantile/mean of N-block maximum and shape
dat	sample vector
V	vector calculated by gevN.Vfun
q	probability, corresponding to q th quantile of the N-block maximum
qty	string indicating whether to calculate the q quantile or the mean

Usage

```
gevN.ll(par, dat, N, q, qty = c('mean', 'quantile'))
gevN.ll.optim(par, dat, N, q = 0.5, qty = c('mean', 'quantile'))
gevN.score(par, dat, N, q = 0.5, qty = c('mean', 'quantile'))
gevN.infomat(par, dat, qty = c('mean', 'quantile'), method = c('obs', 'exp'), N, q = 0.5, nobs = length(dat))
gevN.Vfun(par, dat, N, q = 0.5, qty = c('mean', 'quantile'))
gevN.phi(par, dat, N, q = 0.5, qty = c('mean', 'quantile'), V)
gevN.dphi(par, dat, N, q = 0.5, qty = c('mean', 'quantile'), V)
```

Functions

- gevN.ll: log likelihood
- gevN.score: score vector
- gevN.infomat: expected and observed information matrix
- gevN.Vfun: vector implementing conditioning on approximate ancillary statistics for the TEM
- gevN.phi: canonical parameter in the local exponential family approximation
- gevN.dphi: derivative matrix of the canonical parameter in the local exponential family approximation

Author(s)

Leo Belzile

gevr

*Generalized extreme value distribution (return level parametrization)***Description**

Likelihood, score function and information matrix, approximate ancillary statistics and sample space derivative for the generalized extreme value distribution parametrized in terms of the return level z , scale and shape.

Arguments

par	vector of retlev, scale and shape
dat	sample vector
p	tail probability, corresponding to $(1 - p)$ th quantile for z
method	string indicating whether to use the expected ('exp') or the observed ('obs' - the default) information matrix.
nobs	number of observations
V	vector calculated by <code>gevr.Vfun</code>

Usage

```
gevr.ll(par, dat, p)
gevr.ll.optim(par, dat, p)
gevr.score(par, dat, p)
gevr.infomat(par, dat, p, method = c('obs', 'exp'), nobs = length(dat))
gevr.Vfun(par, dat, p)
gevr.phi(par, dat, p, V)
gevr.dphi(par, dat, p, V)
```

Functions

- `gevr.ll`: log likelihood
- `gevr.ll.optim`: negative log likelihood parametrized in terms of return levels, $\log(\text{scale})$ and shape in order to perform unconstrained optimization
- `gevr.score`: score vector
- `gevr.infomat`: observed information matrix
- `gevr.Vfun`: vector implementing conditioning on approximate ancillary statistics for the TEM
- `gevr.phi`: canonical parameter in the local exponential family approximation
- `gevr.dphi`: derivative matrix of the canonical parameter in the local exponential family approximation

Author(s)

Leo Belzile

gpd

*Generalized Pareto distribution***Description**

Likelihood, score function and information matrix, bias, approximate ancillary statistics and sample space derivative for the generalized Pareto distribution

Arguments

par	vector of scale and shape
dat	sample vector
tol	numerical tolerance for the exponential model
method	string indicating whether to use the expected ('exp') or the observed ('obs' - the default) information matrix.
V	vector calculated by gpd.Vfun
n	sample size

Usage

```
gpd.ll(par, dat, tol=1e-5)
gpd.ll.optim(par, dat, tol=1e-5)
gpd.score(par, dat)
gpd.informat(par, dat, method = c('obs','exp'))
gpd.bias(par, n)
gpd.Fscore(par, dat, method = c('obs','exp'))
gpd.Vfun(par, dat)
gpd.phi(par, dat, V)
gpd.dphi(par, dat, V)
```

Functions

- gpd.ll: log likelihood
- gpd.ll.optim: negative log likelihood parametrized in terms of log(scale) and shape in order to perform unconstrained optimization
- gpd.score: score vector
- gpd.informat: observed or expected information matrix
- gpd.bias: Cox-Snell first order bias
- gpd.Fscore: Firth's modified score equation
- gpd.Vfun: vector implementing conditioning on approximate ancillary statistics for the TEM
- gpd.phi: canonical parameter in the local exponential family approximation
- gpd.dphi: derivative matrix of the canonical parameter in the local exponential family approximation

Author(s)

Leo Belzile

References

- Firth, D. (1993). Bias reduction of maximum likelihood estimates, *Biometrika*, **80**(1), 27–38.
- Coles, S. (2001). *An Introduction to Statistical Modeling of Extreme Values*, Springer, 209 p.
- Cox, D. R. and E. J. Snell (1968). A general definition of residuals, *Journal of the Royal Statistical Society: Series B (Methodological)*, **30**, 248–275.
- Cordeiro, G. M. and R. Klein (1994). Bias correction in ARMA models, *Statistics and Probability Letters*, **19**(3), 169–176.
- Giles, D. E., Feng, H. and R. T. Godwin (2016). Bias-corrected maximum likelihood estimation of the parameters of the generalized Pareto distribution, *Communications in Statistics - Theory and Methods*, **45**(8), 2465–2483.

gpd.abias

Asymptotic bias of threshold exceedances for k order statistics

Description

The formula given in de Haan and Ferreira, 2007 (Springer). Note that the latter differs from that found in Drees, Ferreira and de Haan.

Usage

```
gpd.abias(shape, rho)
```

Arguments

shape	shape parameter
rho	second-order parameter, non-positive

Value

a vector of length containing the bias for scale and shape (in this order)

References

- Dombry, C. and A. Ferreira (2017). Maximum likelihood estimators based on the block maxima method. <https://arxiv.org/abs/1705.00465>

gpd.bcor

*Bias correction for GP distribution***Description**

Bias corrected estimates for the generalized Pareto distribution using Firth's modified score function or implicit bias subtraction.

Usage

```
gpd.bcor(par, dat, corr = c("subtract", "firth"), method = c("obs", "exp"))
```

Arguments

par	parameter vector (scale, shape)
dat	sample of observations
corr	string indicating which correction to employ either subtract or firth
method	string indicating whether to use the expected ('exp') or the observed ('obs' — the default) information matrix. Used only if corr='firth'

Details

Method subtract solves

$$\tilde{\theta} = \hat{\theta} + b(\tilde{\theta})$$

for $\tilde{\theta}$, using the first order term in the bias expansion as given by [gpd.bias](#).

The alternative is to use Firth's modified score and find the root of

$$U(\tilde{\theta}) - i(\tilde{\theta})b(\tilde{\theta}),$$

where U is the score vector, b is the first order bias and i is either the observed or Fisher information.

The routine uses the MLE as starting value and proceeds to find the solution using a root finding algorithm. Since the bias-correction is not valid for $\xi < -1/3$, any solution that is unbounded will return a vector of NA as the bias correction does not exist then.

Value

vector of bias-corrected parameters

Examples

```
set.seed(1)
dat <- rgp(n=40, scale=1, shape=-0.2)
par <- gp.fit(dat, threshold=0, show=FALSE)$estimate
gpd.bcor(par, dat, 'subtract')
gpd.bcor(par, dat, 'firth') #observed information
gpd.bcor(par, dat, 'firth', 'exp')
```

gpd.boot

*Bootstrap approximation for generalized Pareto parameters***Description**

Given an object of class `mev_gpd`, returns a matrix of parameter values to mimic the estimation uncertainty.

Usage

```
gpd.boot(object, B = 1000L, method = c("post", "norm"))
```

Arguments

<code>object</code>	object of class <code>mev_gpd</code>
<code>B</code>	number of pairs to sample
<code>method</code>	string; one of 'norm' for the normal approximation or 'post' (default) for posterior sampling

Details

Two options are available: a normal approximation to the scale and shape based on the maximum likelihood estimates and the observed information matrix. This method uses forward sampling to simulate from a bivariate normal distribution that satisfies the support and positivity constraints

The second approximation uses the ratio-of-uniforms method to obtain samples from the posterior distribution with uninformative priors, thus mimicking the joint distribution of maximum likelihood. The benefit of the latter is that it is more reliable in small samples and when the shape is negative.

Value

a matrix of size `B` by 2 whose columns contain scale and shape parameters

gpd.lmom

*Estimation of generalized Pareto parameters via L-moments***Description**

Given a sample of exceedances, compute the first four L-moments and use either the first two to obtain the scale and shape (default), or else use L-skewness and L-scale to compute the scale and shape of the generalized Pareto distribution

Usage

```
gpd.lmom(xdat, thresh, sorted = FALSE, Lskew = FALSE)
```

Arguments

xdat	[numeric] vector of observations
thresh	[numeric] optional threshold argument
sorted	[logical] if TRUE, observations are sorted in increasing order
Lskew	[logical]; if TRUE, shape is obtained from L-skewness rather than first two moments.

Value

a vector of length two with the scale and shape estimates

gpd.mle	<i>Generalized Pareto maximum likelihood estimates for various quantities of interest</i>
---------	---

Description

This function calls the `fit.gpd` routine on the sample of excesses and returns maximum likelihood estimates for all quantities of interest, including scale and shape parameters, quantiles and value-at-risk, expected shortfall and mean and quantiles of maxima of N threshold exceedances

Usage

```
gpd.mle(
  xdat,
  args = c("scale", "shape", "quant", "VaR", "ES", "Nmean", "Nquant"),
  m,
  N,
  p,
  q
)
```

Arguments

xdat	sample vector of excesses
args	vector of strings indicating which arguments to return the maximum likelihood values for
m	number of observations of interest for return levels. Required only for args values 'VaR' or 'ES'
N	size of block over which to take maxima. Required only for args Nmean and Nquant.
p	tail probability, equivalent to $1/m$. Required only for args quant.
q	level of quantile for N -block maxima. Required only for args Nquant.

Value

named vector with maximum likelihood values for arguments args

Examples

```
xdat <- mev::rgp(n = 30, shape = 0.2)
gpd.mle(xdat = xdat, N = 100, p = 0.01, q = 0.5, m = 100)
```

gpd.pll

*Profile log-likelihood for the generalized Pareto distribution***Description**

This function calculates the (modified) profile likelihood based on the p^* formula. There are two small-sample corrections that use a proxy for $\ell_{\lambda;\hat{\lambda}}$, which are based on Severini's (1999) empirical covariance and the Fraser and Reid tangent exponential model approximation.

Usage

```
gpd.pll(
  psi,
  param = c("scale", "shape", "quant", "retlev", "VaR", "ES", "Nmean", "Nquant"),
  mod = "profile",
  mle = NULL,
  dat,
  m = NULL,
  N = NULL,
  p = NULL,
  q = NULL,
  correction = TRUE,
  thresh = NULL,
  plot = TRUE,
  ...
)
```

Arguments

psi	parameter vector over which to profile (unidimensional)
param	string indicating the parameter to profile over
mod	string indicating the model. See Details .
mle	maximum likelihood estimate in (ψ, ξ) parametrization if $\psi \neq \xi$ and (σ, ξ) otherwise (optional).
dat	sample vector of excesses, unless thresh is provided (in which case user provides original data)
m	number of observations of interest for return levels. Required only for args values 'VaR' or 'ES'

N	size of block over which to take maxima. Required only for args Nmean and Nquant.
p	tail probability, equivalent to $1/m$. Required only for args quant.
q	level of quantile for N-block maxima. Required only for args Nquant.
correction	logical indicating whether to use <code>spline.corr</code> to smooth the tem approximation.
thresh	numerical threshold above which to fit the generalized Pareto distribution
plot	logical; should the profile likelihood be displayed? Default to TRUE
...	additional arguments such as output from call to <code>Vfun</code> if <code>mode='tem'</code> .

Details

The three mod available are `profile` (the default), `tem`, the tangent exponential model (TEM) approximation and `modif` for the penalized profile likelihood based on p^* approximation proposed by Severini. For the latter, the penalization is based on the TEM or an empirical covariance adjustment term.

Value

a list with components

- `mle`: maximum likelihood estimate
- `psi.max`: maximum profile likelihood estimate
- `param`: string indicating the parameter to profile over
- `std.error`: standard error of `psi.max`
- `psi`: vector of parameter ψ given in `psi`
- `pll`: values of the profile log likelihood at `psi`
- `maxpll`: value of maximum profile log likelihood
- `family`: a string indicating "gpd"
- `thresh`: value of the threshold, by default zero

In addition, if `mod` includes `tem`

- `normal`: maximum likelihood estimate and standard error of the interest parameter ψ
- `r`: values of likelihood root corresponding to ψ
- `q`: vector of likelihood modifications
- `rstar`: modified likelihood root vector
- `rstar.old`: uncorrected modified likelihood root vector
- `tem.psimax`: maximum of the tangent exponential model likelihood

In addition, if `mod` includes `modif`

- `tem.mle`: maximum of tangent exponential modified profile log likelihood
- `tem.profl1`: values of the modified profile log likelihood at `psi`

- `tem.maxpll`: value of maximum modified profile log likelihood
- `empcov.mle`: maximum of Severini's empirical covariance modified profile log likelihood
- `empcov.profl1`: values of the modified profile log likelihood at `psi`
- `empcov.maxpll`: value of maximum modified profile log likelihood

Examples

```
## Not run:
dat <- rgp(n = 100, scale = 2, shape = 0.3)
gpd.pll(psi = seq(-0.5, 1, by=0.01), param = 'shape', dat = dat)
gpd.pll(psi = seq(0.1, 5, by=0.1), param = 'scale', dat = dat)
gpd.pll(psi = seq(20, 35, by=0.1), param = 'quant', dat = dat, p = 0.01)
gpd.pll(psi = seq(20, 80, by=0.1), param = 'ES', dat = dat, m = 100)
gpd.pll(psi = seq(15, 100, by=1), param = 'Nmean', N = 100, dat = dat)
gpd.pll(psi = seq(15, 90, by=1), param = 'Nquant', N = 100, dat = dat, q = 0.5)

## End(Not run)
```

gpd.tem

Tangent exponential model approximation for the GP distribution

Description

The function `gpd.tem` provides a tangent exponential model (TEM) approximation for higher order likelihood inference for a scalar parameter for the generalized Pareto distribution. Options include scale and shape parameters as well as value-at-risk (also referred to as quantiles, or return levels) and expected shortfall. The function attempts to find good values for `psi` that will cover the range of options, but the fit may fail and return an error. In such cases, the user can try to find good grid of starting values and provide them to the routine.

Usage

```
gpd.tem(
  dat,
  param = c("scale", "shape", "quant", "VaR", "retlev", "ES", "Nmean", "Nquant"),
  psi = NULL,
  m = NULL,
  thresh = 0,
  n.psi = 50,
  N = NULL,
  p = NULL,
  q = NULL,
  plot = FALSE,
  correction = TRUE,
  ...
)
```

Arguments

<code>dat</code>	sample vector for the GP distribution
<code>param</code>	parameter over which to profile
<code>psi</code>	scalar or ordered vector of values for the interest parameter. If NULL (default), a grid of values centered at the MLE is selected. If <code>psi</code> is of length 2 and <code>n.psi</code> >2, it is assumed to be the minimal and maximal values at which to evaluate the profile log likelihood.
<code>m</code>	number of observations of interest for return levels. See Details . Required only for <code>param = 'VaR'</code> or <code>param = 'ES'</code> .
<code>thresh</code>	threshold value corresponding to the lower bound of the support or the location parameter of the generalized Pareto distribution.
<code>n.psi</code>	number of values of <code>psi</code> at which the likelihood is computed, if <code>psi</code> is not supplied (NULL). Odd values are more prone to give rise to numerical instabilities near the MLE
<code>N</code>	size of block over which to take maxima. Required only for args <code>Nmean</code> and <code>Nquant</code> .
<code>p</code>	tail probability, equivalent to $1/m$. Required only for args <code>quant</code> .
<code>q</code>	level of quantile for N-block maxima. Required only for args <code>Nquant</code> .
<code>plot</code>	logical indicating whether <code>plot.fr</code> should be called upon exit
<code>correction</code>	logical indicating whether spline.cor should be called.
<code>...</code>	additional arguments, for backward compatibility

Details

As of version 1.11, this function is a wrapper around `gpd.pll`.

The interpretation for `m` is as follows: if there are on average m_y observations per year above the threshold, then $m = Tm_y$ corresponds to T -year return level.

Value

an invisible object of class `fr` (see `tem` in package `hoa`) with elements

- `normal`: maximum likelihood estimate and standard error of the interest parameter ψ
- `par.hat`: maximum likelihood estimates
- `par.hat.se`: standard errors of maximum likelihood estimates
- `th.rest`: estimated maximum profile likelihood at $(\psi, \hat{\lambda})$
- `r`: values of likelihood root corresponding to ψ
- `psi`: vector of interest parameter
- `q`: vector of likelihood modifications
- `rstar`: modified likelihood root vector
- `rstar.old`: uncorrected modified likelihood root vector
- `param`: parameter

Author(s)

Leo Belzile

Examples

```

set.seed(123)
dat <- rgp(n = 40, scale = 1, shape = -0.1)
#with plots
m1 <- gpd.tem(param = 'shape', n.psi = 50, dat = dat, plot = TRUE)
## Not run:
m2 <- gpd.tem(param = 'scale', n.psi = 50, dat = dat)
m3 <- gpd.tem(param = 'VaR', n.psi = 50, dat = dat, m = 100)
#Providing psi
psi <- c(seq(2, 5, length = 15), seq(5, 35, length = 45))
m4 <- gpd.tem(param = 'ES', dat = dat, m = 100, psi = psi, correction = FALSE)
mev::plot.fr(m4, which = c(2, 4))
plot(fr4 <- spline.corr(m4))
confint(m1)
confint(m4, parm = 2, warn = FALSE)
m5 <- gpd.tem(param = 'Nmean', dat = dat, N = 100, psi = psi, correction = FALSE)
m6 <- gpd.tem(param = 'Nquant', dat = dat, N = 100, q = 0.7, correction = FALSE)

## End(Not run)

```

gpde

*Generalized Pareto distribution (expected shortfall parametrization)***Description**

Likelihood, score function and information matrix, approximate ancillary statistics and sample space derivative for the generalized Pareto distribution parametrized in terms of expected shortfall.

The parameter m corresponds to $\zeta_u/(1-\alpha)$, where ζ_u is the rate of exceedance over the threshold u and α is the percentile of the expected shortfall. Note that the actual parametrization is in terms of excess expected shortfall, meaning expected shortfall minus threshold.

Arguments

par	vector of length 2 containing e_m and ξ , respectively the expected shortfall at probability $1/(1-\alpha)$ and the shape parameter.
dat	sample vector
m	number of observations of interest for return levels. See Details
tol	numerical tolerance for the exponential model
method	string indicating whether to use the expected ('exp') or the observed ('obs' - the default) information matrix.
nobs	number of observations
V	vector calculated by gpde.Vfun

Details

The observed information matrix was calculated from the Hessian using symbolic calculus in Sage.

Usage

```
gpde.ll(par, dat, m, tol=1e-5)
gpde.ll.optim(par, dat, m, tol=1e-5)
gpde.score(par, dat, m)
gpde.informat(par, dat, m, method = c('obs', 'exp'), nobs = length(dat))
gpde.Vfun(par, dat, m)
gpde.phi(par, dat, V, m)
gpde.dphi(par, dat, V, m)
```

Functions

- `gpde.ll`: log likelihood
- `gpde.ll.optim`: negative log likelihood parametrized in terms of log expected shortfall and shape in order to perform unconstrained optimization
- `gpde.score`: score vector
- `gpde.informat`: observed information matrix for GPD parametrized in terms of rate of expected shortfall and shape
- `gpde.Vfun`: vector implementing conditioning on approximate ancillary statistics for the TEM
- `gpde.phi`: canonical parameter in the local exponential family approximation
- `gpde.dphi`: derivative matrix of the canonical parameter in the local exponential family approximation

Author(s)

Leo Belzile

gpdist

Generalized Pareto distribution

Description

Density function, distribution function, quantile function and random number generation for the generalized Pareto distribution.

Usage

```
pgp(q, loc = 0, scale = 1, shape = 0, lower.tail = TRUE, log.p = FALSE)

dgp(x, loc = 0, scale = 1, shape = 0, log = FALSE)

qgp(p, loc = 0, scale = 1, shape = 0, lower.tail = TRUE)

rgp(n, loc = 0, scale = 1, shape = 0)
```

Arguments

loc	location parameter.
scale	scale parameter, strictly positive.
shape	shape parameter.
lower.tail	logical; if TRUE (default), the lower tail probability $\Pr(X \leq x)$ is returned.
log.p, log	logical; if FALSE (default), values are returned on the probability scale.
x, q	vector of quantiles
p	vector of probabilities
n	scalar number of observations

References

Coles, S. G. (2001) *An Introduction to Statistical Modeling of Extreme Values*, Springer-Verlag, London. doi:[10.1007/9781447136750_3](https://doi.org/10.1007/9781447136750_3)

gpdN	<i>Generalized Pareto distribution (mean of maximum of N exceedances parametrization)</i>
------	---

Description

Likelihood, score function and information matrix, approximate ancillary statistics and sample space derivative for the generalized Pareto distribution parametrized in terms of average maximum of N exceedances.

The parameter N corresponds to the number of threshold exceedances of interest over which the maxima is taken. z is the corresponding expected value of this block maxima. Note that the actual parametrization is in terms of excess expected mean, meaning expected mean minus threshold.

Arguments

par	vector of length 2 containing z and ξ , respectively the mean excess of the maxima of N exceedances above the threshold and the shape parameter.
dat	sample vector
N	block size for threshold exceedances.
tol	numerical tolerance for the exponential model
V	vector calculated by gpdN.Vfun

Details

The observed information matrix was calculated from the Hessian using symbolic calculus in Sage.

Usage

```

gpdrN.ll(par, dat, N, tol=1e-5)
gpdrN.score(par, dat, N)
gpdrN.informat(par, dat, N, method = c('obs', 'exp'), nobs = length(dat))
gpdrN.Vfun(par, dat, N)
gpdrN.phi(par, dat, N, V)
gpdrN.dphi(par, dat, N, V)

```

Functions

- gpdrN.ll: log likelihood
- gpdrN.score: score vector
- gpdrN.informat: observed information matrix for GP parametrized in terms of mean of the maximum of N exceedances and shape
- gpdrN.Vfun: vector implementing conditioning on approximate ancillary statistics for the TEM
- gpdrN.phi: canonical parameter in the local exponential family approximation
- gpdrN.dphi: derivative matrix of the canonical parameter in the local exponential family approximation

Author(s)

Leo Belzile

gpdr

Generalized Pareto distribution (return level parametrization)

Description

Likelihood, score function and information matrix, approximate ancillary statistics and sample space derivative for the generalized Pareto distribution parametrized in terms of return levels.

Arguments

par	vector of length 2 containing y_m and ξ , respectively the m -year return level and the shape parameter.
dat	sample vector
m	number of observations of interest for return levels. See Details
tol	numerical tolerance for the exponential model
method	string indicating whether to use the expected ('exp') or the observed ('obs' - the default) information matrix.
nobs	number of observations
V	vector calculated by gpdr.Vfun

Details

The observed information matrix was calculated from the Hessian using symbolic calculus in Sage.

The interpretation for m is as follows: if there are on average m_y observations per year above the threshold, then $m = Tm_y$ corresponds to T -year return level.

Usage

```
gpdr.ll(par, dat, m, tol=1e-5)
gpdr.ll.optim(par, dat, m, tol=1e-5)
gpdr.score(par, dat, m)
gpdr.infomat(par, dat, m, method = c('obs', 'exp'), nobs = length(dat))
gpdr.Vfun(par, dat, m)
gpdr.phi(par, V, dat, m)
gpdr.dphi(par, V, dat, m)
```

Functions

- `gpdr.ll`: log likelihood
- `gpdr.ll.optim`: negative log likelihood parametrized in terms of $\log(\text{scale})$ and shape in order to perform unconstrained optimization
- `gpdr.score`: score vector
- `gpdr.infomat`: observed information matrix for GPD parametrized in terms of rate of m -year return level and shape
- `gpdr.Vfun`: vector implementing conditioning on approximate ancillary statistics for the TEM
- `gpdr.phi`: canonical parameter in the local exponential family approximation
- `gpdr.dphi`: derivative matrix of the canonical parameter in the local exponential family approximation

Author(s)

Leo Belzile

ibvpot

Interpret bivariate threshold exceedance models

Description

This is an adaptation of the `evir` package `interpret.gpdbiv` function. `interpret.fbvpot` deals with the output of a call to `fbvpot` from the **evd** and to handle families other than the logistic distribution. The likelihood derivation comes from expression 2.10 in Smith et al. (1997).

Usage

```
ibvpot(fitted, q, silent = FALSE)
```

Arguments

<code>fitted</code>	the output of <code>fbvpot</code> or a list. See Details.
<code>q</code>	a vector of quantiles to consider, on the data scale. Must be greater than the thresholds.
<code>silent</code>	boolean; whether to print the interpretation of the result. Default to FALSE.

Details

The list `fitted` must contain

- `model` a string; see `bvevd` from package `evd` for options
- `param` a named vector containing the parameters of the model, as well as parameters `scale1`, `shape1`, `scale2` and `shape2`, corresponding to marginal GPD parameters.
- `threshold` a vector of length 2 containing the two thresholds.
- `pat` the proportion of observations above the corresponding threshold

Value

an invisible numeric vector containing marginal, joint and conditional exceedance probabilities.

Author(s)

Leo Belzile, adapting original S code by Alexander McNeil

References

Smith, Tawn and Coles (1997), Markov chain models for threshold exceedances. *Biometrika*, **84**(2), 249–268.

See Also

`interpret.gpdbiv` in package `evir`

Examples

```
if (requireNamespace("evd", quietly = TRUE)) {
  y <- rgp(1000,1,1,1)
  x <- y*rmevspec(n=1000,d=2,sigma=cbind(c(0,0.5),c(0.5,0)), model='hr')
  mod <- evd::fbvpot(x, threshold = c(1,1), model = 'hr', likelihood='censored')
  ibvpot(mod, c(20,20))
}
```

kjtail	<i>Estimators of the tail coefficient</i>
--------	---

Description

Estimators proposed by Krupskii and Joe under second order expansion for the coefficient of tail dependence η and the joint tail orthant probability

Usage

```
kjtail(
  xdat,
  qlev,
  ptail = NULL,
  mqu = NULL,
  type = 1,
  ties.method = eval(formals(rank)$ties.method),
  ...
)
```

Arguments

xdat	a matrix of observations
qlev	vector of quantile levels
ptail	tail probability smaller than qlev. Default to NULL
mqu	marginal quantile levels for semiparametric estimation; data above this are modelled using a generalized Pareto distribution. If NULL, empirical estimation is used throughout
type	integer indicating the estimator type
ties.method	method for handling of ties in rank transformation
...	additional arguments, for backward compatibility

Value

a list with elements

- p quantile level for estimation
- eta matrix of estimated coefficient of tail dependence η , and standard errors
- k1 parameter of the tail expansion
- pat proportion of observations above the threshold
- lambda tail dependence coefficient (sic)
- tailprob tail probability, if ptail is provided

Note

EXPERIMENTAL. The numerical optimization of the likelihood surface is difficult, as the function is ill-behaved. Visual inspection of estimates is often necessary to check for possible lack of convergence.

Examples

```
d <- 2
rho <- 0.9
Sigma <- matrix(rho, d, d) + diag(1 - rho, d)
eta_true <- 1/sum(Sigma)
data <- rmnorm(
  n = 1e4,
  mu = rep(0, d),
  Sigma = Sigma)
q <- seq(0.95, 0.995, by = 0.005)
kj <- kjtail(xdat = data, qlev = q)
plot(kj)
abline(h = (1+rho)/2, col = 2)
```

leedspollution

*Leeds air pollution***Description**

Daily maximum data (hourly for PM10) on air pollution for the Leeds Centre station in Yorkshire and Humberside station. The data goes from January 1st, 1993, until December 31st, 2024. Data show seasonality and there are some outliers. From December 2nd, 2008 onwards, particulate matters (PM10 and PM2.5) are measured using a tapered element oscillating microbalance (TEOM) and Filter Dynamics Measurement System (FDMS). The data for PM2.5 is missing before the change of instrumentation. A total of 231 daily measurements with only missing values were removed during preprocessing.

Usage

```
leedspollution
```

Format

A data frame with 11455 rows and 8 variables:

```
date [character] a date with format yyyy-mm-dd
O3 [integer] ozone (in nanograms per cubic meter)
NO [integer] nitrogen oxyde (in nanograms per cubic meter)
CO [double] carbon monoxyde (in micrograms per cubic meter)
NO2 nitrogen dioxyde (in nanograms per cubic meter)
SO2 sulphur dioxide (in nanograms per cubic meter)
PM10 [integer] particulate matter 10, (in nanograms per cubic meter)
PM2.5 [integer] particulate matter 2.5, (in nanograms per cubic meter)
```

Source

Crown 2025 copyright Defra via uk-air.defra.gov.uk, licenced under the Open Government Licence (OGL).

likmgp	<i>Likelihood for multivariate peaks over threshold models</i>
--------	--

Description

Likelihood for the various parametric limiting models over region determined by

$$\{y \in F : \max_{j=1}^D \sigma_j \frac{y_j^\xi - 1}{\xi_j} + \mu_j > u\};$$

where μ is loc, σ is scale and ξ is shape.

Usage

```
likmgp(  
  dat,  
  thresh,  
  loc,  
  scale,  
  shape,  
  par,  
  model = c("log", "br", "xstud"),  
  likt = c("mgp", "pois", "binom"),  
  lambdau = 1,  
  ...  
)
```

Arguments

dat	matrix of observations
thresh	functional threshold for the maximum
loc	vector of location parameter for the marginal generalized Pareto distribution
scale	vector of scale parameter for the marginal generalized Pareto distribution
shape	vector of shape parameter for the marginal generalized Pareto distribution
par	list of parameters: alpha for the logistic model, Lambda for the Brown–Resnick model or else Sigma and df for the extremal Student.
model	string indicating the model family, one of "log", "neglog", "br" or "xstud"
likt	string indicating the type of likelihood, with an additional contribution for the non-exceeding components: one of "mgp", "binom" and "pois".
lambdau	vector of marginal rate of marginal threshold exceedance.
...	additional arguments (see Details)

Details

Optional arguments can be passed to the function via . . .

- `cl` cluster instance created by `makeCluster` (default to `NULL`)
- `ncors` number of cores for parallel computing of the likelihood
- `mmax` maximum per column
- `B1` number of replicates for quasi Monte Carlo integral for the exponent measure
- `genvec1` generating vector for the quasi Monte Carlo routine (exponent measure), associated with `B1`

Value

the value of the log-likelihood with attributes `expme`, giving the exponent measure

Note

The location and scale parameters are not identifiable unless one of them is fixed.

maiquetia

Maiquetia Daily Rainfall

Description

Daily cumulated rainfall (in mm) at Maiquetia airport, Venezuela. The observations cover the period from January 1961 to December 1999. The original series had missing days in February 1996 (during which there were 2 days with 1hr each of light rain) and January 1998 (no rain). These were replaced by zeros.

Format

a vector of size 14244 containing daily rainfall (in mm),

Source

J.R. Cordova and M. González, accessed 25.11.2018 from <<https://rss.onlinelibrary.wiley.com/hub/journal/14679876/series-c-datasets>>

References

Coles, S. and L.R. Pericchi (2003). Anticipating Catastrophes through Extreme Value Modelling, *Applied Statistics*, **52**(4), 405-416.

Coles, S., Pericchi L.R. and S. Sisson (2003). A fully probabilistic approach to extreme rainfall modeling, *Journal of Hydrology*, **273**, 35-50.

Examples

```
## Not run:
data(maiquetia, package = "mev")
day <- seq.Date(from = as.Date("1961-01-01"), to = as.Date("1999-12-31"), by = "day")
nzrain <- maiquetia[substr(day, 1, 4) < 1999 & maiquetia > 0]
fit.gpd(nzrain, threshold = 30, show = TRUE)

## End(Not run)
```

maxstable	<i>Transform arguments using max stability</i>
-----------	--

Description

Given a vector of location, scale and shape parameters, compute the corresponding parameters for block of size m assuming a generalized extreme value distribution.

Usage

```
maxstable(pars, m = 1L, inverse = FALSE)
```

Arguments

<code>pars</code>	vector of location, scale and shape parameters
<code>m</code>	[integer] block size
<code>inverse</code>	[logical] whether to compute the parameters for the inverse relationship (defaults to FALSE)

Examples

```
maxstable(pars = maxstable(pars = c(1,2,0), m = 10), m = 10, inv = TRUE)
maxstable(pars = maxstable(pars = c(1,2,0.1), m = 5), m = 1/5)
```

nidd	<i>River Nidd Flow</i>
------	------------------------

Description

The data consists of exceedances over the threshold 65 cubic meter per second of the River Nidd at Hunsingore Weir, for 35 years of data between 1934 and 1969.

Format

a vector of size 154

Source

Natural Environment Research Council (1975). *Flood Studies Report*, volume 4. pp. 235–236.

References

Davison, A.C. and R.L. Smith (1990). Models for Exceedances over High Thresholds (with discussion), *Journal of the Royal Statistical Society. Series B (Methodological)*, **52**(3), 393–442.

See Also

nidd.thresh from the evir package

nutrients

Nutrient data

Description

Interview component of survey 'What we eat in America'. These are extracted from the 2015–2016 National Health and Nutrition Examination Survey (NHANES, <https://wwwn.cdc.gov/nchs/nhanes/Default.aspx>) report and consist of the total nutrients for all food and beverage intake ingested over a 24 hours period.

Usage

nutrients

Format

A data frame with 9544 rows and 38 variables:

prot proteins (in grams)

carb carbohydrate (in gram)

sugr total sugars (in gram)

fibe dietary fibers (in grams)

tfat total fat (in grams)

sfat saturated fat (in grams)

mfat monounsaturated fat (in grams)

pfat polyunsaturated fat (in grams)

chol cholesterol (in milligrams)

atoc vitamin E as alpha-tocopherol (in milligrams)

ret retinol (in micrograms)

vara Vitamin A as retinol activity equivalents (in micrograms).

acar alpha-carotene (in micrograms)

bcar beta-carotene (in micrograms)

cryp beta-cryptoxanthin (in micrograms)
lyco lycopene (in micrograms)
lz lutein and zeaxanthin (in micrograms).
vb1 thiamin (vitamin B1, in milligrams)
vb2 riboflavin (vitamin B2, in milligrams)
niac niacin (in milligrams)
vb6 vitamin B5 (in milligrams)
fola total folate (in micrograms)
fa folic acid (in micrograms)
ff food folate (in micrograms)
chl total choline (in milligrams)
vb12 vitamin B12 (in micrograms)
vc vitamin C (in milligrams)
vd vitamin D (comprising D2 and D3, in micrograms)
vk vitamin K (in micrograms)
calc calcium (in milligrams)
phos phosphorus (in milligrams)
magn magnesium (in milligrams)
iron iron (in milligrams)
zinc zinc (in milligrams)
copp copper (in milligrams)
sodi sodium (in milligrams)
pota potassium (in milligrams)
sele selenium (in micrograms)

Details

Note that the sample design oversampled specific population targets and that only respondents are provided. The website contains more information about sampling weights. There are multiple missing records.

Note

These data are subject to a data user agreement, available at <https://www.cdc.gov/nchs/policy/data-user-agreement.html>

Source

National Center for Health Statistics, now available from the Wayback Machine via https://web.archive.org/web/20201029113801/https://wwwn.cdc.gov/Nchs/Nhanes/2015-2016/DR1TOT_I.XPT

pandemics	<i>Deaths from pandemics</i>
-----------	------------------------------

Description

The data base contains estimated records of the number of deaths from pandemics.

Usage

pandemics

Format

A data frame with 72 rows and 8 variables:

event name of the event
 startyear start year of the event
 endyear end year of the event
 lower lower bound on estimated deaths (in thousands)
 average average estimated deaths (in thousands)
 upper upper bound on estimated deaths (in thousands)
 saverage scaled average of estimated deaths (in thousands)
 population estimated population at risk (in thousands)

Source

Cirillo, P. and N.N. Taleb (2020). *Tail risk of contagious diseases*. Nat. Phys. **16**, 606–613 (2020).
 <doi:10.1038/s41567-020-0921-x>

penultimate	<i>Smith's penultimate approximations</i>
-------------	---

Description

The function takes as arguments the distribution and density functions. There are two options: method='bm' yields block maxima and method='pot' threshold exceedances. For method='bm', the user should provide in such case the block sizes via the argument m, whereas if method='pot', a vector of threshold values should be provided. The other argument (thresh or m depending on the method) is ignored.

Usage

```
penultimate(family, method = c("bm", "pot"), thresh, qlev, m, ...)
```

Arguments

family	the name of the parametric family. Will be used to obtain dfamily, pfamily, qfamily
method	either block maxima ('bm') or peaks-over-threshold ('pot') are supported
thresh	vector of thresholds for method 'pot'
qlev	vector of quantile levels for method 'pot', e.g., 0.9, 0.95, ... Ignored if argument thresh is provided.
m	vector of block sizes for method 'bm'
...	additional arguments passed to densF and distF

Details

Alternatively, the user can provide functions densF, quantF and distF for the density, quantile function and distribution functions, respectively. The user can also supply the derivative of the density function, ddensF. If the latter is missing, it will be approximated using finite-differences.

For method = "pot", the function computes the reciprocal hazard and its derivative on the log scale to avoid numerical overflow. Thus, the density function should have argument log and the distribution function arguments log.p and lower.tail, respectively.

Value

a data frame containing

- loc: location parameters (method='bm')
- scale: scale parameters
- shape: shape parameters
- thresh: thresholds (if method='pot'), percentile corresponding to threshold (if method='pot')
- m: block sizes (if method='bm')

Author(s)

Leo Belzile

References

Smith, R.L. (1987). Approximations in extreme value theory. *Technical report 205*, Center for Stochastic Process, University of North Carolina, 1–34.

Examples

```
# Threshold exceedance for Normal variables
quants <- seq(1, 5, by = 0.02)
penult <- penultimate(
  family = "norm",
  method = 'pot',
  thresh = quants,
  ddensF = function(x){-x*dnorm(x)}, # optional argument
```

```

    )
plot(x = quants,
     y = penult$shape,
     type = 'l',
     xlab = 'quantile',
     ylab = 'Penultimate shape',
     ylim = c(-0.5, 0))
# Block maxima for Gamma variables
# User must provide arguments for shape (or rate), for which there is no default
m <- seq(30, 3650, by = 30)
penult <- penultimate(family = 'gamma', method = 'bm', m = m, shape = 0.1)
plot(x = m,
     y = penult$shape,
     type = 'l',
     xlab = 'quantile',
     ylab = 'penultimate shape')

# Comparing density of GEV approximation with true density of maxima
m <- 100 # block of size 100
p <- penultimate(
  family = 'norm',
  ddensF = function(x){-x*dnorm(x)},
  method = 'bm',
  m = m)
x <- seq(1, 5, by = 0.01)
plot(
  x = x,
  y = m * dnorm(x) * exp((m-1) * pnorm(x, log.p = TRUE)),
  type = 'l',
  ylab = 'density',
  main = 'Distribution of the maxima of\n 100 standard normal variates')
lines(x, mev::dgev(x, loc = p$loc, scale = p$scale, shape = 0), col = 2)
lines(x, mev::dgev(x, loc = p$loc, scale = p$scale, shape = p$shape), col = 4)
legend(
  x = 'topright',
  lty = c(1, 1, 1),
  col = c(1, 2, 4),
  legend = c('exact', 'ultimate', 'penultimate'),
  bty = 'n')

```

plot.eprof

Plot of (modified) profile likelihood

Description

The function plots the (modified) profile likelihood and the tangent exponential profile likelihood

Usage

```
## S3 method for class 'eprof'
plot(x, ...)
```

Arguments

`x` an object of class `eprof` returned by `gpd.pll` or `gev.pll`.
`...` further arguments to `plot`.

Value

a graph of the (modified) profile likelihoods

References

Brazzale, A. R., Davison, A. C. and Reid, N. (2007). *Applied Asymptotics: Case Studies in Small-Sample Statistics*. Cambridge University Press, Cambridge.
 Severini, T. A. (2000). *Likelihood Methods in Statistics*. Oxford University Press, Oxford.

plot.fr	<i>Plot of tangent exponential model profile likelihood</i>
---------	---

Description

This function is adapted from the `plot.fr` function from the `hoa` package bundle. It differs from the latter mostly in the placement of legends.

Usage

```
## S3 method for class 'fr'
plot(x, ...)
```

Arguments

`x` an object of class `fr` returned by `gpd.tem` or `gev.tem`.
`...` further arguments to `plot` currently ignored. Providing a numeric vector which allows for custom selection of the plots. A logical `all`. See **Details**.

Details

Plots produced depend on the integers provided in `which`. 1 displays the Wald pivot, the likelihood root r , the modified likelihood root r_{star} and the likelihood modification q as functions of the parameter ψ . 2 gives the renormalized profile log likelihood and adjusted form, with the maximum likelihood having ordinate value of zero. 3 provides the significance function, a transformation of 1. Lastly, 4 plots the correction factor as a function of the likelihood root; it is a diagnostic plot aimed for detecting failure of the asymptotic approximation, often due to poor numerics in a neighborhood of $r=0$; the function should be smooth. The function `spline.corr` is designed to handle this by correcting numerically unstable estimates, replacing outliers and missing values with the fitted values from the fit.

Value

graphs depending on argument which

References

Brazzale, A. R., Davison, A. C. and Reid, N. (2007). *Applied Asymptotics: Case Studies in Small-Sample Statistics*. Cambridge University Press, Cambridge.

plot.mev_shape_rbm	<i>Plots for random block maximum estimator</i>
--------------------	---

Description

The function returns plot of the shape estimator along with the value (and 95% Wald-based confidence interval) at the selected threshold, or a plot of the empirical Bayes risk.

Usage

```
## S3 method for class 'mev_shape_rbm'
plot(x, type = c("shape", "risk"), log = TRUE, ...)
```

Arguments

x	object of class mev_shape_rbm returned by shape.rbm
type	[string] type of plot, either "shape" for the tail index or "risk" for the empirical Bayes risk
log	[logical] if TRUE (default), the x-axis for the number of exceedances is displayed on the log scale.
...	additional arguments, currently ignored

Value

one or more plots

plot.mev_thselect_vmetric

Plots for Varty and al. metric-based threshold selection

Description

This S3 method produces quantile-quantile plots with confidence and tolerance bands on various scale (uniform, exponential, generalized Pareto), or a plot of the metric as a function of the threshold.

Usage

```
## S3 method for class 'mev_thselect_vmetric'
plot(
  x,
  type = c("qq", "pp", "exp", "metric"),
  B = 1000L,
  probs = c(0.025, 0.975),
  ...
)
```

Arguments

x	an object of class <code>mev_thselect_vmetric</code> produced by a call to <code>thselect.vmetric</code>
type	string; a single string indicating the choice of plot
B	number of simulations for variability of estimation
probs	quantile levels for intervals.
...	additional arguments, currently ignored

Value

NULL; the function is used to produce a plot

plot.mev_thselect_wadsworth

Sequential analysis diagnostic plots for threshold selection

Description

Sequential analysis diagnostic plots for threshold selection

Usage

```
## S3 method for class 'mev_thselect_wadsworth'
plot(x, type = c("wn", "ps"), ...)
```

Arguments

x	object returned by a call to <code>thselect.wseq</code>
type	string giving the plots to produce
...	additional arguments passed to plotting function

Value

NULL; the method is used to generate plots

plot.mev_tstab_mrl	<i>Mean residual life parameter stability plot</i>
--------------------	--

Description

Mean residual life parameter stability plot

Usage

```
## S3 method for class 'mev_tstab_mrl'
plot(
  x,
  xlab = c("thresh", "nexc"),
  level = 0.95,
  type = c("band", "ptwise"),
  ...
)
```

Arguments

x	object resulting from a call to <code>tstab.mrl</code>
xlab	[string]; whether to plot mean residual life plot as a function of threshold value of number of exceedances
level	[numeric] level of Wald confidence intervals
type	[string] whether to plot pointwise confidence intervals using segments ("ptwise") or using dashed lines ("band")
...	additional arguments, currently ignored

Value

NULL; use to produce plots

pp	<i>Poisson process of extremes.</i>
----	-------------------------------------

Description

Likelihood, score function and information matrix for the Poisson process likelihood.

Arguments

par	vector of loc, scale and shape
dat	sample vector
u	threshold
method	string indicating whether to use the expected ('exp') or the observed ('obs' - the default) information matrix.
np	number of periods of observations. This is a <i>post hoc</i> adjustment for the intensity so that the parameters of the model coincide with those of a generalized extreme value distribution with block size $\text{length}(\text{dat})/\text{np}$.
nobs	number of observations for the expected information matrix. Default to $\text{length}(\text{dat})$ if dat is provided.

Usage

```
pp.ll(par, dat)
pp.ll(par, dat, u, np)
pp.score(par, dat)
pp.infomat(par, dat, method = c('obs', 'exp'))
```

Functions

- `pp.ll`: log likelihood
- `pp.score`: score vector
- `pp.infomat`: observed or expected information matrix

Author(s)

Leo Belzile

References

- Coles, S. (2001). *An Introduction to Statistical Modeling of Extreme Values*, Springer, 209 p.
- Wadsworth, J.L. (2016). Exploiting Structure of Maximum Likelihood Estimators for Extreme Value Threshold Selection, *Technometrics*, **58**(1), 116-126, <http://dx.doi.org/10.1080/00401706.2014.998345>.
- Sharkey, P. and J.A. Tawn (2017). A Poisson process reparameterisation for Bayesian inference for extremes, *Extremes*, **20**(2), 239-263, <http://dx.doi.org/10.1007/s10687-016-0280-2>.

qweissman

*Weissman's quantile estimator***Description**

Given a small probability of exceedance p , the number of exceedances k out of n observation above the threshold u (thresh) (corresponding typically to the $(k + 1)$ th order statistic, compute the tail quantile at level $Q(1 - p)$ using the estimator of Weissman (1978) under the assumption of Pareto tail (positive shape ξ), viz.

$$Q(1 - p) = u \left(\frac{k}{pn} \right)^\xi.$$

Usage

```
qweissman(p, k, n, thresh, shape)
```

Arguments

p	tail probability, must be larger than the proportion of exceedances k/n .
k	vector of the number of exceedances above thresh
n	integer, total sample size
thresh	vector of thresholds
shape	vector of positive shape parameters

Value

a vector of tail quantiles

References

Weissman, I. (1978). Estimation of Parameters and Larger Quantiles Based on the k Largest Observations. *Journal of the American Statistical Association*, 73(364), 812–815. <doi:10.2307/2286285>.

Examples

```
set.seed(2025)
p <- 1/100
xdat <- rgp(n = 1000, loc = 2, scale = 2, shape = 0.4)
hill <- shape.hill(xdat, k = seq(20L, 100L, by = 10L))
thresh <- sort(xdat, decreasing = TRUE)[hill$k+1]
qweissman(
  p = 1/100,
  k = hill$k,
  n = length(xdat),
  thresh = thresh,
  shape = hill$shape)
# Compare with true quantile
qgp(1/100, loc = 2, scale = 2, shape = 0.4, lower.tail = FALSE)
```

rdir	<i>Random variate generation for Dirichlet distribution on S_d</i>
------	---

Description

A function to sample Dirichlet random variables, based on the representation as ratios of Gamma. Note that the RNG will generate on the full simplex and the sum to one constraint is respected here

Usage

```
rdir(n, alpha, normalize = TRUE)
```

Arguments

n	sample size
alpha	vector of parameter
normalize	boolean. If FALSE, the function returns Gamma variates with parameter alpha.

Value

sample of dimension d (size of alpha) from the Dirichlet distribution.

Examples

```
rdir(n=100, alpha=c(0.5,0.5,2),TRUE)
rdir(n=100, alpha=c(3,1,2),FALSE)
```

rgparp	<i>Simulation from generalized R-Pareto processes</i>
--------	---

Description

The generalized R-Pareto process is supported on $(loc - scale / shape, Inf)$ if $shape > 0$, or $(-Inf, loc - scale / shape)$ for negative shape parameters, conditional on $(X - r(loc))/r(scale) > 0$. The standard Pareto process corresponds to $scale = loc = rep(1, d)$.

Usage

```
rgparp(
  n,
  shape = 1,
  thresh = 1,
  risk = c("mean", "sum", "site", "max", "min", "l2"),
  siteindex = NULL,
  d,
```

```

    loc,
    scale,
    param,
    sigma,
    model = c("log", "neglog", "bilog", "negbilog", "hr", "br", "xstud", "smith",
              "schlather", "ct", "sdir", "dirmix"),
    weights,
    vario,
    coord = NULL,
    ...
)

```

Arguments

<code>n</code>	number of observations
<code>shape</code>	shape parameter of the generalized Pareto variable
<code>thresh</code>	univariate threshold for the exceedances of risk functional
<code>risk</code>	string indicating the risk functional.
<code>siteindex</code>	integer between 1 and d specifying the index of the site or variable
<code>d</code>	dimension of sample
<code>loc</code>	location vector
<code>scale</code>	scale vector
<code>param</code>	parameter vector for the logistic, bilogistic, negative bilogistic and extremal Dirichlet (Coles and Tawn) model. Parameter matrix for the Dirichlet mixture. Degree of freedoms for extremal student model. See Details .
<code>sigma</code>	covariance matrix for Brown-Resnick and extremal Student-t distributions. Symmetric matrix of squared coefficients λ^2 for the Husler-Reiss model, with zero diagonal elements.
<code>model</code>	for multivariate extreme value distributions, users can choose between 1-parameter logistic and negative logistic, asymmetric logistic and negative logistic, bilogistic, Husler-Reiss, extremal Dirichlet model (Coles and Tawn) or the Dirichlet mixture. Spatial models include the Brown-Resnick, Smith, Schlather and extremal Student max-stable processes. Max linear models are also supported
<code>weights</code>	vector of length m for the m mixture components that sum to one. For the "maxlin" model, weights should be a matrix with d columns that represent the weight of the components and whose column sum to one (if provided, this argument overrides asy).
<code>vario</code>	semivariogram function whose first argument must be distance. Used only if provided in conjunction with coord and if sigma is missing
<code>coord</code>	d by k matrix of coordinates, used as input in the variogram vario or as parameter for the Smith model. If grid is TRUE, unique entries should be supplied.
<code>...</code>	additional arguments for the vario function

Value

an n by d sample from the generalized R-Pareto process, with attributes `accept.rate` if the procedure uses rejection sampling.

Examples

```
rgparp(n = 10, risk = 'site', siteindex = 2, d = 3, param = 2.5,
       model = 'log', scale = c(1, 2, 3), loc = c(2, 3, 4), shape = 0.5)
rgparp(n = 10, risk = 'max', d = 4, param = c(0.2, 0.1, 0.9, 0.5),
       scale = 1:4, loc = 1:4, model = 'bilog')
rgparp(n = 10, risk = 'sum', d = 3, param = c(0.8, 1.2, 0.6, -0.5),
       scale = 1:3, loc = 1:3, model = 'sdir')
vario <- function(x, scale = 0.5, alpha = 0.8){ scale*x^alpha }
grid.coord <- as.matrix(expand.grid(runif(4), runif(4)))
rgparp(n = 10, risk = 'max', vario = vario, coord = grid.coord,
       model = 'br', scale = runif(16), loc = rnorm(16))
```

rho.dk

*Second order tail index estimator of Drees and Kaufmann***Description**

Estimator of the second order regular variation parameter $\rho \leq 0$ parameter for heavy-tailed data proposed by Drees and Kaufmann (1998)

Usage

```
rho.dk(xdat, k, tau = 0.5)
```

Arguments

<code>xdat</code>	vector of positive observations
<code>k</code>	number of highest order statistics to use for estimation
<code>tau</code>	tuning parameter $\tau \in (0, 1)$

References

Drees, H. and E. Kaufmann (1998). Selecting the optimal sample fraction in univariate extreme value estimation, *Stochastic Processes and their Applications*, 75(2), 149-172, <doi:10.1016/S0304-4149(98)00017-9>.

rho.fagh	<i>Second order tail index estimator of Fraga Alves et al. Estimator of the second order regular variation parameter $\rho \leq 0$ parameter for heavy-tailed data proposed by Fraga Alves et al. (2003)</i>
----------	---

Description

Second order tail index estimator of Fraga Alves et al. Estimator of the second order regular variation parameter $\rho \leq 0$ parameter for heavy-tailed data proposed by Fraga Alves et al. (2003)

Usage

```
rho.fagh(xdat, k, tau = 0)
```

Arguments

xdat	vector of positive observations
k	number of highest order statistics to use for estimation
tau	scalar real tuning parameter. Default values is 0, which is typically chosen whenever $\rho \geq -1$. The choice $\tau = 1$ otherwise.

References

Fraga Alves, M.I., Gomes, M. Ivette, and de Haan, Laurens (2003). A new class of semi-parametric estimators of the second order parameter. *Portugaliae Mathematica*. Nova Serie 60(2), 193-213. <<http://eudml.org/doc/50867>>.

Examples

```
# Example with rho = -0.2
n <- 1000
xdat <- mev::rgp(n = n, shape = 0.2)
kmin <- floor(n^0.995)
kmax <- ceiling(n^0.999)
rho_est <- rho.fagh(
  xdat = xdat,
  k = n - kmin:kmax)
rho_med <- mean(rho_est$rho)
```

rho.gbw

*Second order tail index estimator of Goegebeur et al. (2008)***Description**

Estimator of the second order regular variation parameter $\rho \leq 0$ parameter for heavy-tailed data based on ratio of kernel goodness-of-fit statistics.

Usage

```
rho.gbw(xdat, k)
```

Arguments

xdat	vector of positive observations
k	number of highest order statistics to use for estimation

References

Goegebeur, Y., J. Beirlant and T. de Wet (2008). Linking Pareto-tail kernel goodness-of-fit statistics with tail index at optimal threshold and second order estimation. *REVSTAT-Statistical Journal*, 6(1), 51-69. <doi:10.57805/revstat.v6i1.57>

rho.ghp

*Second order tail index estimator of Gomes et al.***Description**

Estimator of the second order regular variation parameter $\rho \leq 0$ parameter for heavy-tailed data proposed by Gomes et al. (2003)

Usage

```
rho.ghp(xdat, k, alpha = 2)
```

Arguments

xdat	vector of positive observations
k	number of highest order statistics to use for estimation
alpha	positive scalar tuning parameter

References

Gomes, M.I., de Haan, L. & Peng, L. (2002). Semi-parametric Estimation of the Second Order Parameter in Statistics of Extremes. *Extremes* 5, 387–414. <doi:10.1023/A:1025128326588>

rlarg

Distribution of the r-largest observations

Description

Likelihood, score function and information matrix for the r-largest observations likelihood.

Arguments

par	vector of loc, scale and shape
dat	an n by r sample matrix, ordered from largest to smallest in each row
method	string indicating whether to use the expected ('exp') or the observed ('obs' - the default) information matrix.
nobs	number of observations for the expected information matrix. Default to nrow(dat) if dat is provided.
r	number of order statistics kept. Default to ncol(dat)

Usage

```
rlarg.ll(par, dat, u, np)
rlarg.score(par, dat)
rlarg.infomat(par, dat, method = c('obs', 'exp'), nobs = nrow(dat), r = ncol(dat))
```

Functions

- rlarg.ll: log likelihood
- rlarg.score: score vector
- rlarg.infomat: observed or expected information matrix

Author(s)

Leo Belzile

References

- Coles, S. (2001). *An Introduction to Statistical Modeling of Extreme Values*, Springer, 209 p.
- Smith, R.L. (1986). Extreme value theory based on the r largest annual events, *Journal of Hydrology*, **86**(1-2), 27–43, [http://dx.doi.org/10.1016/0022-1694\(86\)90004-1](http://dx.doi.org/10.1016/0022-1694(86)90004-1).

Description

Implementation of the random number generators for multivariate extreme-value distributions and max-stable processes based on the two algorithms described in Dombry, Engelke and Oesting (2016).

Usage

```
rmev(
  n,
  d,
  param,
  asy,
  sigma,
  model = c("log", "alog", "neglog", "aneglog", "bilog", "negbilog", "hr", "br", "xstud",
    "smith", "schlather", "ct", "sdir", "dirmix", "pairbeta", "pairexp", "wdirbs",
    "wexpbs", "maxlin"),
  alg = c("ef", "sm"),
  weights = NULL,
  vario = NULL,
  coord = NULL,
  grid = FALSE,
  dist = NULL,
  ...
)
```

Arguments

n	number of observations
d	dimension of sample
param	parameter vector for the logistic, bilogistic, negative bilogistic and extremal Dirichlet (Coles and Tawn) model. Parameter matrix for the Dirichlet mixture. Degree of freedoms for extremal student model. See Details .
asy	list of asymmetry parameters, as in function <code>rmvevd</code> from package <code>evd</code> , of $2^d - 1$ vectors of size corresponding to the power set of d , with sum to one constraints.
sigma	covariance matrix for Brown-Resnick and extremal Student-t distributions. Symmetric matrix of squared coefficients λ^2 for the Husler-Reiss model, with zero diagonal elements.
model	for multivariate extreme value distributions, users can choose between 1-parameter logistic and negative logistic, asymmetric logistic and negative logistic, bilogistic, Husler-Reiss, extremal Dirichlet model (Coles and Tawn) or the Dirichlet mixture. Spatial models include the Brown-Resnick, Smith, Schlather and extremal Student max-stable processes. Max linear models are also supported

alg	algorithm, either simulation via extremal function ('ef') or via the spectral measure ('sm'). Default to ef.
weights	vector of length m for the m mixture components that sum to one. For the "maxlin" model, weights should be a matrix with d columns that represent the weight of the components and whose column sum to one (if provided, this argument overrides asy).
vario	semivariogram function whose first argument must be distance. Used only if provided in conjunction with coord and if sigma is missing
coord	d by k matrix of coordinates, used as input in the variogram vario or as parameter for the Smith model. If grid is TRUE, unique entries should be supplied.
grid	Logical. TRUE if the coordinates are two-dimensional grid points (spatial models).
dist	symmetric matrix of pairwise distances. Default to NULL.
...	additional arguments for the vario function

Details

The vector param differs depending on the model

- log: one dimensional parameter greater than 1
- alog: $2^d - d - 1$ dimensional parameter for dep. Values are recycled if needed.
- neglog: one dimensional positive parameter
- aneglog: $2^d - d - 1$ dimensional parameter for dep. Values are recycled if needed.
- bilog: d-dimensional vector of parameters in $[0, 1]$
- negbilog: d-dimensional vector of negative parameters
- ct, dir, negdir, sdir: d-dimensional vector of positive (a)symmetry parameters. For dir and negdir, a $d + 1$ vector consisting of the d Dirichlet parameters and the last entry is an index of regular variation in $(-\min(\alpha_1, \dots, \alpha_d), 1]$ treated as shape parameter
- xstud: one dimensional parameter corresponding to degrees of freedom alpha
- dirmix: d by m-dimensional matrix of positive (a)symmetry parameters
- pairbeta, pairexp: $d(d-1)/2+1$ vector of parameters, containing the concentration parameter and the coefficients of the pairwise beta, in lexicographical order e.g., $\beta_{12}, \beta_{13}, \dots$
- wdirbs, wexpbs: 2d vector of d concentration parameters followed by the d Dirichlet parameters

Stephenson points out that the multivariate asymmetric negative logistic model given in e.g. Coles and Tawn (1991) is not a valid distribution function in dimension $d > 3$ unless additional constraints are imposed on the parameter values. The implementation in mev uses the same construction as the asymmetric logistic distribution (see the vignette). As such it does not match the bivariate implementation of [rbvevd](#).

The dependence parameter of the evd package for the Husler-Reiss distribution can be recovered taking for the Brown-Resnick model $2/r = \sqrt{(2\gamma(h))}$ where h is the lag vector between sites and $r = 1/\lambda$ for the Husler-Reiss.

Value

an n by d exact sample from the corresponding multivariate extreme value model

Warning

As of version 1.8 (August 16, 2016), there is a distinction between models `hr` and `br`. The latter is meant to be used in conjunction with variograms. The parametrization differs between the two models.

The family of scaled Dirichlet is now parametrized by a parameter in $-\min(\alpha)$ appended to the `d` vector `param` containing the parameter `alpha` of the Dirichlet model. Arguments `model='dir'` and `model='negdir'` are still supported internally, but not listed in the options.

Author(s)

Leo Belzile

References

Dombry, Engelke and Oesting (2016). Exact simulation of max-stable processes, *Biometrika*, **103**(2), 303–317.

See Also

[rmevspec](#), [rmvevd](#), [rbvevd](#)

Examples

```
set.seed(1)
rmev(n=100, d=3, param=2.5, model='log', alg='ef')
rmev(n=100, d=4, param=c(0.2,0.1,0.9,0.5), model='bilog', alg='sm')
## Spatial example using power variogram
#NEW: Semi-variogram must take distance as argument
semivario <- function(x, scale, alpha){ scale*x^alpha }
#grid specification
grid.coord <- as.matrix(expand.grid(runif(4), runif(4)))
rmev(n=100, vario=semivario, coord=grid.coord, model='br', scale = 0.5, alpha = 1)
#using the Brown-Resnick model with a covariance matrix
vario2cov <- function(coord, semivario,...){
  sapply(1:nrow(coord), function(i) sapply(1:nrow(coord), function(j)
    semivario(sqrt(sum((coord[i,]-coord[j,])^2)), ...) +
    semivario(sqrt(sum((coord[j,]-coord[i,])^2)), ...) -
    semivario(sqrt(sum((coord[i,]-coord[j,])^2)), ...)))
}
rmev(n=100, sigma=vario2cov(grid.coord, semivario = semivario, scale = 0.5, alpha = 1), model='br')
# asymmetric logistic model - see function 'rmvevd' from package 'evd'
asy <- list(0, 0, 0, 0, c(0,0), c(0,0), c(0,0), c(0,0), c(0,0), c(0,0),
  c(.2,.1,.2), c(.1,.1,.2), c(.3,.4,.1), c(.2,.2,.2), c(.4,.6,.2,.5))
rmev(n=1, d=4, param=0.3, asy=asy, model="alog")
#Example with a grid (generating an array)
rmev(n=10, sigma=cbind(c(2,1), c(1,3)), coord=cbind(runif(4), runif(4)), model='smith', grid=TRUE)
## Example with Dirichlet mixture
```

```
alpha.mat <- cbind(c(2,1,1),c(1,2,1),c(1,1,2))
rmev(n=100, param=alpha.mat, weights=rep(1/3,3), model='dirmix')
rmev(n=10, param=c(0.1,1,2,3), d=3, model='pairbeta')
```

rmevspec	<i>Random samples from spectral distributions of multivariate extreme value models.</i>
----------	---

Description

Generate from Q_i , the spectral measure of a given multivariate extreme value model based on the L1 norm.

Usage

```
rmevspec(
  n,
  d,
  param,
  sigma,
  model = c("log", "neglog", "bilog", "negbilog", "hr", "br", "xstud", "smith",
    "schlather", "ct", "sdir", "dirmix", "pairbeta", "pairexp", "wdirbs", "wexpbs",
    "maxlin"),
  weights = NULL,
  vario = NULL,
  coord = NULL,
  grid = FALSE,
  dist = NULL,
  ...
)
```

Arguments

n	number of observations
d	dimension of sample
param	parameter vector for the logistic, bilogistic, negative bilogistic and extremal Dirichlet (Coles and Tawn) model. Parameter matrix for the Dirichlet mixture. Degree of freedoms for extremal student model. See Details .
sigma	covariance matrix for Brown-Resnick and extremal Student-t distributions. Symmetric matrix of squared coefficients λ^2 for the Husler-Reiss model, with zero diagonal elements.
model	for multivariate extreme value distributions, users can choose between 1-parameter logistic and negative logistic, asymmetric logistic and negative logistic, bilogistic, Husler-Reiss, extremal Dirichlet model (Coles and Tawn) or the Dirichlet mixture. Spatial models include the Brown-Resnick, Smith, Schlather and extremal Student max-stable processes. Max linear models are also supported

weights	vector of length m for the m mixture components that sum to one. For the "maxlin" model, weights should be a matrix with d columns that represent the weight of the components and whose column sum to one (if provided, this argument overrides <code>asy</code>).
vario	semivariogram function whose first argument must be distance. Used only if provided in conjunction with <code>coord</code> and if <code>sigma</code> is missing
coord	d by k matrix of coordinates, used as input in the variogram <code>vario</code> or as parameter for the Smith model. If <code>grid</code> is TRUE, unique entries should be supplied.
grid	Logical. TRUE if the coordinates are two-dimensional grid points (spatial models).
dist	symmetric matrix of pairwise distances. Default to NULL.
...	additional arguments for the <code>vario</code> function

Details

The vector `param` differs depending on the model

- `log`: one dimensional parameter greater than 1
- `neglog`: one dimensional positive parameter
- `bilog`: d -dimensional vector of parameters in $[0, 1]$
- `negbilog`: d -dimensional vector of negative parameters
- `ct`, `dir`, `negdir`: d -dimensional vector of positive (a)symmetry parameters. Alternatively, a $d + 1$ vector consisting of the d Dirichlet parameters and the last entry is an index of regular variation in $(0, 1]$ treated as scale
- `xstud`: one dimensional parameter corresponding to degrees of freedom α
- `dirmix`: d by m -dimensional matrix of positive (a)symmetry parameters
- `pairbeta`, `pairexp`: $d(d-1)/2+1$ vector of parameters, containing the concentration parameter and the coefficients of the pairwise beta, in lexicographical order e.g., $\beta_{1,2}, \beta_{1,3}, \dots$
- `wdirbs`, `wexpbs`: $2d$ vector of d concentration parameters followed by the d Dirichlet parameters

Value

an n by d exact sample from the corresponding multivariate extreme value model

Note

This functionality can be useful to generate for example Pareto processes with marginal exceedances.

Author(s)

Leo Belzile

References

Dombry, Engelke and Oesting (2016). Exact simulation of max-stable processes, *Biometrika*, **103**(2), 303–317.

Boldi (2009). A note on the representation of parametric models for multivariate extremes. *Extremes* **12**, 211–218.

Examples

```
set.seed(1)
rmevspec(n=100, d=3, param=2.5, model='log')
rmevspec(n=100, d=3, param=2.5, model='neglog')
rmevspec(n=100, d=4, param=c(0.2,0.1,0.9,0.5), model='bilog')
rmevspec(n=100, d=2, param=c(0.8,1.2), model='ct') #Dirichlet model
rmevspec(n=100, d=2, param=c(0.8,1.2,0.5), model='sdir') #with additional scale parameter
#Variogram gamma(h) = scale*||h||^alpha
#NEW: Variogram must take distance as argument
vario <- function(x, scale=0.5, alpha=0.8){ scale*x^alpha }
#grid specification
grid.coord <- as.matrix(expand.grid(runif(4), runif(4)))
rmevspec(n=100, vario=vario, coord=grid.coord, model='br')
## Example with Dirichlet mixture
alpha.mat <- cbind(c(2,1,1),c(1,2,1),c(1,1,2))
rmevspec(n=100, param=alpha.mat, weights=rep(1/3,3), model='dirmix')
```

rmnorm

Multivariate Normal distribution sampler

Description

Sampler derived using the eigendecomposition of the covariance matrix Sigma. The function uses the Armadillo random normal generator

Usage

```
rmnorm(n, mu, Sigma)
```

Arguments

n	sample size
mu	mean vector. Will set the dimension
Sigma	a square covariance matrix, of same dimension as mu. No sanity check is performed to validate that the matrix is positive definite, so use at own risk

Value

an n sample from a multivariate Normal distribution

Examples

```
rmnorm(n = 10, mu = c(0,2), Sigma = diag(2))
```

rparp

Simulation from R-Pareto processes

Description

Simulation from R-Pareto processes

Usage

```
rparp(
  n,
  shape = 1,
  risk = c("sum", "site", "max", "min", "l2"),
  siteindex = NULL,
  d,
  param,
  sigma,
  model = c("log", "neglog", "bilog", "negbilog", "hr", "br", "xstud", "smith",
    "schlather", "ct", "sdir", "dirmix"),
  weights,
  vario,
  coord = NULL,
  ...
)
```

Arguments

n	number of observations
shape	shape tail index of Pareto variable
risk	string indicating risk functional.
siteindex	integer between 1 and d specifying the index of the site or variable
d	dimension of sample
param	parameter vector for the logistic, bilogistic, negative bilogistic and extremal Dirichlet (Coles and Tawn) model. Parameter matrix for the Dirichlet mixture. Degree of freedoms for extremal student model. See Details .
sigma	covariance matrix for Brown-Resnick and extremal Student-t distributions. Symmetric matrix of squared coefficients λ^2 for the Husler-Reiss model, with zero diagonal elements.
model	for multivariate extreme value distributions, users can choose between 1-parameter logistic and negative logistic, asymmetric logistic and negative logistic, bilogistic, Husler-Reiss, extremal Dirichlet model (Coles and Tawn) or the Dirichlet mixture. Spatial models include the Brown-Resnick, Smith, Schlather and extremal Student max-stable processes. Max linear models are also supported

weights	vector of length m for the m mixture components that sum to one. For the "maxlin" model, weights should be a matrix with d columns that represent the weight of the components and whose column sum to one (if provided, this argument overrides asy).
vario	semivariogram function whose first argument must be distance. Used only if provided in conjunction with coord and if sigma is missing
coord	d by k matrix of coordinates, used as input in the variogram vario or as parameter for the Smith model. If grid is TRUE, unique entries should be supplied.
...	additional arguments for the vario function

Details

For riskf=max and riskf=min, the procedure uses rejection sampling based on Pareto variates sampled from sum and may be slow if d is large.

Value

an n by d sample from the R-Pareto process, with attributes `accept.rate` if the procedure uses rejection sampling.

Examples

```
rparp(n=10, risk = 'site', siteindex=2, d=3, param=2.5, model='log')
rparp(n=10, risk = 'min', d=3, param=2.5, model='neglog')
rparp(n=10, risk = 'max', d=4, param=c(0.2,0.1,0.9,0.5), model='bilog')
rparp(n=10, risk = 'sum', d=3, param=c(0.8,1.2,0.6, -0.5), model='sdir')
vario <- function(x, scale=0.5, alpha=0.8){ scale*x^alpha }
grid.coord <- as.matrix(expand.grid(runif(4), runif(4)))
rparp(n=10, risk = 'max', vario=vario, coord=grid.coord, model='br')
```

rparpcs

Simulation from Pareto processes using composition sampling

Description

The algorithm performs forward sampling by simulating first from a mixture, then sample angles conditional on them being less than (max) or greater than (min) one. The resulting sample from the angular distribution is then multiplied by Pareto variates with tail index shape.

Usage

```
rparpcs(
  n,
  model = c("log", "neglog", "br", "xstud"),
  risk = c("max", "min"),
  param = NULL,
  d,
```

```

    Lambda = NULL,
    Sigma = NULL,
    df = NULL,
    shape = 1,
    ...
)

```

Arguments

n	sample size.
model	string indicating the model family.
risk	string indicating the risk functional. Only max and min are currently supported.
param	parameter value for the logistic or negative logistic model
d	dimension of the multivariate model, only needed for logistic or negative logistic models
Lambda	parameter matrix for the Brown–Resnick model. See Details .
Sigma	correlation matrix if model = 'xstud', otherwise the covariance matrix formed from the stationary Brown-Resnick process.
df	degrees of freedom for extremal Student process.
shape	tail index of the Pareto variates (reciprocal shape parameter). Must be strictly positive.
...	additional parameters, currently ignored

Details

For the moment, only exchangeable models and models based on elliptical processes are handled.

The parametrization of the Brown–Resnick is in terms of the matrix Lambda, which is formed by evaluating the semivariogram γ at sites s_i, s_j , meaning that $\Lambda_{i,j} = \gamma(s_i, s_j)/2$.

The argument Sigma is ignored for the Brown-Resnick model if Lambda is provided by the user.

Value

an n by d matrix of samples, where d = ncol(Sigma), with attributes mixt.weights.

Author(s)

Leo Belzile

See Also

[rparp](#) for general simulation of Pareto processes based on an accept-reject algorithm.

Examples

```
## Not run:
#Brown-Resnick, Wadsworth and Tawn (2014) parametrization
D <- 20L
coord <- cbind(runif(D), runif(D))
semivario <- function(d, alpha = 1.5, lambda = 1){0.5 * (d/lambda)^alpha}
Lambda <- semivario(as.matrix(dist(coord))) / 2
rparpcs(n = 10, Lambda = Lambda, model = 'br', shape = 0.1)
#Extremal Student
Sigma <- stats::rWishart(n = 1, df = 20, Sigma = diag(10))[, , 1]
rparpcs(n = 10, Sigma = cov2cor(Sigma), df = 3, model = 'xstud')

## End(Not run)
```

rparpcshr

Simulation of generalized Huesler-Reiss Pareto vectors via composition sampling

Description

Sample from the generalized Pareto process associated to Huesler-Reiss spectral profiles. For the Huesler-Reiss Pareto vectors, the matrix Sigma is utilized to build Q viz.

$$Q = \Sigma^{-1} - \frac{\Sigma^{-1} \mathbf{1}_d \mathbf{1}_d^\top \Sigma^{-1}}{\mathbf{1}_d^\top \Sigma^{-1} \mathbf{1}_d}.$$

The location vector $\mathbf{m}$ and Sigma are the parameters of the underlying log-Gaussian process.

Usage

```
rparpcshr(n, u, alpha, Sigma, m)
```

Arguments

n	sample size
u	vector of marginal location parameters (must be strictly positive)
alpha	vector of shape parameters (must be strictly positive).
Sigma	covariance matrix of process, used to define Q . See Details .
m	location vector of Gaussian distribution.

Value

n by d matrix of observations

References

Ho, Z. W. O and C. Dombry (2019), Simple models for multivariate regular variations and the Huesler-Reiss Pareto distribution, Journal of Multivariate Analysis (**173**), p. 525-550, [doi:10.1016/j.jmva.2019.04.008](https://doi.org/10.1016/j.jmva.2019.04.008)

Examples

```

D <- 20L
coord <- cbind(runif(D), runif(D))
di <- as.matrix(dist(rbind(c(0, ncol(coord)), coord)))
semivario <- function(d, alpha = 1.5, lambda = 1){(d/lambda)^alpha}
Vmat <- semivario(di)
Sigma <- outer(Vmat[-1, 1], Vmat[1, -1], '+') - Vmat[-1, -1]
m <- Vmat[-1,1]
## Not run:
samp <- rparpcshr(n = 100, u = c(rep(1, 10), rep(2, 10)),
                 alpha = seq(0.1, 1, length = 20), Sigma = Sigma, m = m)

## End(Not run)

```

rrlarg

*Simulate r-largest observations from point process of extremes***Description**

Simulate the r -largest observations from a Poisson point process with intensity

$$\Lambda(x) = (1 + \xi(x - \mu)/\sigma)^{-1/\xi}$$

.

Usage

```
rrlarg(n, r, loc = 0, scale = 1, shape = 0)
```

Arguments

<code>n</code>	sample size
<code>r</code>	number of observations per block
<code>loc</code>	location parameter
<code>scale</code>	scale parameter
<code>shape</code>	shape parameter

Value

an n by r matrix of samples from the point process, ordered from largest to smallest in each row.

shape.erm	<i>Exponential regression estimator</i>
-----------	---

Description

This function implements the exponential regression estimator of the shape parameter for the case of Pareto tails with positive shape index.

Usage

```
shape.erm(xdat, k, method = c("bdgm", "fh"), bounds = NULL)
```

Arguments

xdat	vector of observations
k	vector of integer, the number of largest observations to consider
method	string; one of bdgm for the approach of Beirlant, Dierckx, Goegebeur and Matthys (1999) or fh for Feuerverger and Hall (1999)
bounds	vector of length 2 giving the bounds for ρ , the second order parameter. Default to $\rho \in [-5, -0.5]$

Details

The second-order parameter is difficult to pin down, and while values within $[-1, 0)$ are most logical under Hall model, the model parameters become unidentifiable when $\rho \rightarrow 0$. The default constraint restrict $-5 < \rho < -0.5$, with the upper bound changed to -0.25 for sample of sizes larger than 5000 observations. Users can set the value of the bounds for ρ via argument bounds. The optimization is initialized at the Hill estimator.

Value

a data frame with columns

- k number of exceedances
- shape estimate of the shape parameter
- rho estimate of the second-order regular variation index
- scale estimate of the scale parameter

References

Feuerverger, A. and P. Hall (1999), Estimating a tail exponent by modelling departure from a Pareto distribution, *The Annals of Statistics* 27(2), 760-781. <doi:10.1214/aos/1018031215>

Beirlant, J., Dierckx, G., Goegebeur, Y. G. Matthys (1999). Tail Index Estimation and an Exponential Regression Model. *Extremes*, 2, 177–200 (1999). <doi:10.1023/A:1009975020370>

shape.genjack	<i>Generalized jackknife shape estimator</i>
---------------	--

Description

Generalized jackknife shape estimator

Usage

```
shape.genjack(xdat, k)
```

Arguments

xdat	vector of positive observations
k	vector of order statistics; if missing, a vector going from 10 to sample size minus one.

Value

a data frame with the number of order statistics k and the shape parameter estimate shape, or a single numeric value if k is a scalar integer.

References

Gomes, I.M., João Martins, M. and Neves, M. (2000) Alternatives to a Semi-Parametric Estimator of Parameters of Rare Events-The Jackknife Methodology. *Extremes*, 3, 207–229. doi:[10.1023/A:1011470010228](https://doi.org/10.1023/A:1011470010228)

shape.genquant	<i>Beirlant et al. generalized quantile shape estimator</i>
----------------	---

Description

This estimator estimates the real shape parameter based on generalized quantile plots based on mean excess functions, generalized median excesses or trimmed mean excesses.

Usage

```
shape.genquant(
  xdat,
  k,
  type = c("genmean", "genmed", "trimmean"),
  weight,
  p = 0.5
)
```

Arguments

xdat	n vector of observations
k	number of upper order statistics
type	string indicating the estimator choice, one of genmean, genmed and trimmean.
weight	weight a kernel function on $[0, 1]$
p	number between zero and one giving the proportion of order statistics for the second threshold

References

Beirlant, J., Vynckier P. and J.L. Teugels (1996). *Excess functions and estimation of the extreme-value index*. Bernoulli, 2(4), 293-318.

shape.hill	<i>Hill's estimator of the shape parameter</i>
------------	--

Description

Given a sample of positive observations, calculate the tail index or shape parameter. The shape estimate returned is positive.

Usage

```
shape.hill(xdat, k)
```

Arguments

xdat	vector of positive observations
k	vector of order statistics; if missing, a vector going from 10 to sample size minus one.

Value

a data frame with the number of order statistics k and the shape parameter estimate shape, or a single numeric value if k is a scalar integer.

References

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

Examples

```
xdat <- mev::rgp(n = 200, loc = 1, scale = 0.5, shape = 0.5)
shape.hill(xdat)
```

shape.lthill	<i>Lower-trimmed Hill estimator for the shape parameter</i>
--------------	---

Description

Given a sample of Pareto-tailed samples (positive tail index), compute the lower-trimmed Hill estimator. If $k_0 = k$, the estimator reduces to Hill's estimator for the shape index

Usage

```
shape.lthill(xdat, k, k0 = k, sorted = FALSE, ...)
```

Arguments

xdat	[numeric] vector of positive observations
k	[integer] number of order statistics for the threshold
k0	[integer] vector of number of largest order statistics, no greater than k
sorted	[logical] if TRUE, data are assumed to be sorted in decreasing order.
...	additional arguments for other routines (notably vectorize)

Value

a scalar with the shape parameter estimate if k_0 is a scalar, otherwise a data frame with columns k_0 for the number of exceedances and shape for the tail index.

References

Bladt, M., Albrecher, H. & Beirlant, J. (2020) *Threshold selection and trimming in extremes*. *Extremes*, 23, 629-665 . doi:[10.1007/s10687020003850](https://doi.org/10.1007/s10687020003850)

Examples

```
# Pareto sample
n <- 200
xdat <- 10/(1 - runif(n)) - 10
shape.lthill(xdat = xdat, k = 100, k0 = 5:100)
```

shape.moment	<i>Dekkers and de Haan moment estimator for the shape</i>
--------------	---

Description

Given a sample of exceedances, compute the moment estimator of the real shape parameter.

Usage

```
shape.moment(xdat, k)
```

Arguments

xdat	vector of positive observations of length n
k	number of largest order statistics

Value

a data frame with the number of order statistics k and the shape parameter estimate `shape`, or a single numeric value if k is a scalar.

References

Dekkers, A.L.M. and de Haan, L. (1989). *On the estimation of the extreme-value index and large quantile estimation.*, Annals of Statistics, **17**, 1795-1833.

shape.osz	<i>Extreme U-statistic Pickands estimator</i>
-----------	---

Description

Given a random sample of n exceedances, the estimator returns an estimator of the shape parameter or extreme value index using a kernel of order 3, based on k largest exceedances of `xdat`. Note that the method does not allow for ties.

Usage

```
shape.osz(xdat, k, ...)
```

Arguments

xdat	vector of observations of length n
k	number of largest order statistics $3 \leq k < n$.
...	additional arguments for backward compatibility

Details

The calculations are based on the recursions provided in Lemma 4.3 of Oorschot et al.

References

Oorschot, J, J. Segers and C. Zhou (2023), Tail inference using extreme U-statistics, Electronic Journal of Statistics 17(1): 1113-1159. doi:[10.1214/23EJS2129](https://doi.org/10.1214/23EJS2129)

Examples

```
xdat <- rgp(n = 1000, shape = 0.2)
shape.osz(xdat, k = 10)
```

shape.pickands	<i>Pickand's shape estimator</i>
----------------	----------------------------------

Description

Given a sample of size n of positive exceedances, compute the real shape parameter ξ based on the k largest order statistics.

Usage

```
shape.pickands(xdat, k)
```

Arguments

xdat	vector of positive observations of length n
k	number of largest order statistics

Value

a data frame with the number of order statistics k and the shape parameter estimate shape, or a single numeric value if k is a scalar.

References

Pickands, III, J. (1975). *Statistical inference using extreme order statistics*. Annals of Statistics, **3**, 119-131.

shape.rbm	<i>Random block maxima shape estimator of Wager</i>
-----------	---

Description

Computes the shape estimator for varying k up to sample size of maximum $k_{\max}$ largest observations

Usage

```
shape.rbm(xdat, k = 10:floor(length(xdat)/2), ...)
```

Arguments

xdat	[vector] sample exceedances
k	[int] vector of integers for which to compute the estimator
...	additional parameters, currently ignored

Value

a list with elements

- k** number of exceedances
- shape** tail index estimate, strictly positive
- risk** empirical Bayes estimate of risk
- thresh** threshold given by the smallest order statistic considered in the sample

References

Wager, S. (2014). Subsampling extremes: From block maxima to smooth tail estimation, *Journal of Multivariate Analysis*, 130, 335-353, doi:[10.1016/j.jmva.2014.06.010](https://doi.org/10.1016/j.jmva.2014.06.010)

shape.trimhill	<i>Trimmed Hill estimator for the shape parameter</i>
----------------	---

Description

Given a sample of Pareto-tailed samples (positive tail index), compute the trimmed Hill estimator. If $k_0 = k$, the estimator reduces to Hill's estimator for the shape index

Usage

```
shape.trimhill(xdat, k, k0, sorted = FALSE)
```

Arguments

xdat	[numeric] vector of positive observations
k	[integer] number of order statistics for the threshold
k0	[integer] number of largest order statistics, strictly less than k
sorted	[logical] if TRUE, data are assumed to be sorted in decreasing order

Value

a scalar with the shape parameter estimate

References

Bhattacharya, S., Kallitsis, M. and S. Stoev, (2019) Data-adaptive trimming of the Hill estimator and detection of outliers in the extremes of heavy-tailed data. *Electronic Journal of Statistics* 13, 1872–1925

shape.vries	<i>de Vries shape estimator</i>
-------------	---------------------------------

Description

Given a sample of exceedances, compute the moment estimator of the positive shape parameter using the ratio of log ratio of exceedance and it's square.

Usage

```
shape.vries(xdat, k)
```

Arguments

xdat	vector of positive observations of length n
k	number of largest order statistics

Value

a data frame with the number of order statistics k and the shape parameter estimate shape, or a single numeric value if k is a scalar.

References

de Haan, L. and Peng, L. (1998). *Comparison of tail index estimators*, *Statistica Neerlandica* 52, 60-70.

de Haan, L. and Peng, L. (1998). Comparison of tail index estimators. *Statistica Neerlandica*, 52: 60-70. doi:[10.1111/14679574.00068](https://doi.org/10.1111/14679574.00068)

spunif	<i>Semi-parametric marginal transformation to uniform</i>
--------	---

Description

The function `spunif` transforms a matrix or vector of data `x` to the pseudo-uniform scale using a semiparametric transform. Data below the threshold are transformed to pseudo-uniforms using a rank transform, while data above the threshold are assumed to follow a generalized Pareto distribution. The parameters of the latter are estimated using maximum likelihood if either `scale = NULL` or `shape = NULL`.

Usage

```
spunif(x, thresh, scale = NULL, shape = NULL)
```

Arguments

<code>x</code>	matrix or vector of data
<code>thresh</code>	vector of marginal thresholds
<code>scale</code>	vector of marginal scale parameters for the generalized Pareto
<code>shape</code>	vector of marginal shape parameters for the generalized Pareto

Value

a matrix or vector of the same dimension as `x`, with pseudo-uniform observations

Author(s)

Leo Belzile

Examples

```
x <- rmev(1000, d = 3, param = 2, model = 'log')
thresh <- apply(x, 2, quantile, 0.95)
spunif(x, thresh)
```

taildep

*Coefficient of tail correlation and tail dependence***Description**

For data with unit Pareto margins, the coefficient of tail dependence η is defined via

$$\Pr(\min(X) > x) = L(x)x^{-1/\eta},$$

where $L(x)$ is a slowly varying function. Ignoring the latter, several estimators of η can be defined. In unit Pareto margins, η is a nonnegative shape parameter that can be estimated by fitting a generalized Pareto distribution above a high threshold. In exponential margins, η is a scale parameter and the maximum likelihood estimator of the latter is the Hill estimator. Both methods are based on peaks-over-threshold and the user can choose between pointwise confidence obtained through a likelihood ratio test statistic ("lrt") or the Wald statistic ("wald").

Usage

```
taildep(
  xdat,
  qlev = NULL,
  nq = 40,
  qlim = c(0.8, 0.99),
  depmeas = c("eta", "chi"),
  estimator = list(eta = c("emp", "betacop", "gpd", "hill"), chi = c("emp", "betacop")),
  confint = c("wald", "lrt"),
  level = 0.95,
  trunc = TRUE,
  margtrans = c("emp", "none"),
  ties.method = "random",
  plot = TRUE,
  ...
)
```

Arguments

xdat	an n by d matrix of multivariate observations
qlev	vector of percentiles between 0 and 1
nq	number of quantiles of the structural variable at which to form a grid; only used if $u = \text{NULL}$.
qlim	limits for the sequence u of the structural variable
depmeas	dependence measure, either of "eta" or "chi"
estimator	named list giving the estimation method for eta and chi. Default to "emp" for both.
confint	string indicating the type of confidence interval for η , one of "wald" or "lrt"

level	the confidence level required (default to 0.95).
trunc	logical indicating whether the estimates and confidence intervals should be truncated in $[0, 1]$
margtrans	marginal transformation; if "none", data are assumed to be in uniform margins
ties.method	string indicating the type of method for rank; see rank for a list of options. Default to "random"
plot	logical; should graphs be plotted?
...	additional arguments passed to plot; current support for main, xlab, ylab, add and further pch, lty, type, col for points; additional arguments for confidence intervals are handled via cipch, cilty, cotype, cicol.

Details

The most common approach for estimation is the empirical survival copula, by evaluating the proportion of sample minima with uniform margins that exceed a given x . An alternative estimator uses a smoothed estimator of the survival copula using Bernstein polynomial, resulting in the so-called betacop estimator. Approximate pointwise confidence intervals for the latter are obtained by assuming the proportion of points is binomial.

The coefficient of tail correlation χ is

$$\chi = \lim_{u \rightarrow 1} \frac{\Pr(F_1(X_1) > u, \dots, F_D(X_D) > u)}{1 - u}.$$

Asymptotically independent vectors have $\chi = 0$. The estimator uses an estimator of the survival copula

Value

a named list with elements

- qlev: a K vector of percentile levels
- eta: a K by 3 matrix with point estimates, lower and upper confidence intervals
- chi: a K by 3 matrix with point estimates, lower and upper confidence intervals

Note

As of version 1.15, the percentiles used are from the minimum variable. This ensures that, regardless of the number of variables, there is no error message returned because the quantile levels are too low for there to be observations

See Also

[chplot](#) for bivariate empirical estimates of χ and $\bar{\chi}$.

Examples

```
## Not run:
set.seed(765)
# Max-stable model
dat <- rmev(n = 1000, d = 4, param = 0.7, model = "log")
taildep(dat, confint = 'wald')

## End(Not run)
```

test.maxstab

P-P plot for testing max stability

Description

The diagnostic, proposed by Gabda, Towe, Wadsworth and Tawn, relies on the fact that, for max-stable vectors on the unit Gumbel scale, the distribution of the maxima is Gumbel distribution with a location parameter equal to the exponent measure. One can thus consider tuples of size m and estimate the location parameter via maximum likelihood and transforming observations to the standard Gumbel scale. Replicates are then pooled and empirical quantiles are defined. The number of combinations of m vectors can be prohibitively large, hence only $n_{\max}$ randomly selected tuples are selected from all possible combinations. The confidence intervals are obtained by a nonparametric bootstrap, by resampling observations with replacement observations for the selected tuples and re-estimating the location parameter. The procedure can be computationally intensive as a result.

Usage

```
test.maxstab(
  xdat,
  m = prod(dim(dat)[-1]),
  nmax = 500L,
  B = 1000L,
  ties.method = "random",
  plot = TRUE,
  ...
)
```

Arguments

xdat	matrix or array of max-stable observations, typically block maxima. The first dimension should consist of replicates
m	integer indicating how many tuples should be aggregated.
nmax	maximum number of pairs. Default to 500L.
B	number of nonparametric bootstrap replications. Default to 1000L.
ties.method	string indicating the method for rank . Default to "random".
plot	logical indicating whether a graph should be produced (default to TRUE).
...	additional arguments for backward compatibility

Value

a Tukey probability-probability plot with 95% confidence intervals obtained using a nonparametric bootstrap

References

Gabda, D.; Towe, R. Wadsworth, J. and J. Tawn, Discussion of “Statistical Modeling of Spatial Extremes” by A. C. Davison, S. A. Padoan and M. Ribatet. *Statist. Sci.* **27** (2012), no. 2, 189–192.

Examples

```
## Not run:
xdat <- mev::rmev(n = 250, d = 100, param = 0.5, model = "log")
test.maxstab(xdat, m = 100)
xdat <- rmnorm(n = 250, Sigma = diag(0.5, 10) + matrix(0.5, 10, 10), mu = rep(0, 10))
test.maxstab(xdat, m = 2, nmax = 100)
test.maxstab(xdat, m = ncol(xdat))

## End(Not run)
```

test.scoreindep	<i>Ramos and Ledford test of independence</i>
-----------------	---

Description

The Ramos and Ledford (2005) score test of independence is a modification of tests by Tawn (1988) and Ledford and Tawn (1996) for a logistic model parameter $\alpha = 1$; the latter two have scores with zero expectation, but the variance of the score are infinite, which produces non-regularity and yield test, once suitably normalized, that converge slowly to their asymptotic null distribution. The test, designed for bivariate samples, transforms observations to have unit Frechet margins and considers a bivariate censored likelihood approach for the logistic distribution.

Usage

```
test.scoreindep(xdat, p, test = c("ledford", "tawn"))
```

Arguments

xdat	a n by 2 matrix of observations
p	probability level for the marginal threshold
test	string; if tawn, only censor observations in the upper quadrant when both variables are large as in Tawn (1988), otherwise censor marginally for ledford as in Ledford and Tawn (1996).

Value

a list with elements

stat value of the score test statistic

pval asymptotic p-value

test test argument

Examples

```
samp <- rmev(n = 1000, d = 2,
  param = 0.99, model = "log")
(test.scoreindep(samp, p = 0.9))
```

thselect.alrs	<i>Automatic L-moment ratio selection method</i>
---------------	--

Description

Given a sample of observations, calculate the L-skewness and L-kurtosis over a set of candidate thresholds. For each threshold candidate, we find the L-skewness that minimizes the sum of squared distance between the theoretical L-skewness and L-kurtosis of the generalized Pareto distribution,

$$\min_{\tau_3} (t_3 - \tau_3)^2 + [t_4 - \tau_3(1 + 5\tau_3)/(5 + \tau_3)]^2.$$

The function returns the threshold with the minimum distance.

Usage

```
thselect.alrs(xdat, thresh, plot = FALSE)
```

Arguments

xdat [numeric] vector of observations

thresh [numeric] vector of candidate thresholds. If missing, 20 sample quantiles starting at the 0.25 quantile in increments of 3.75 percent.

plot [logical] if TRUE, return a plot of the sample L-kurtosis against the L-skewness, along with the theoretical generalized Pareto curve.

Value

scalar for the chosen numeric threshold

References

Silva Lomba, J., Fraga Alves, M.I. (2020). *L-moments for automatic threshold selection in extreme value analysis*. Stoch Environ Res Risk Assess, 34, 465–491. doi:[10.1007/s0047702001789](https://doi.org/10.1007/s0047702001789)

thselect.bab

*Lower truncated Hill threshold selection***Description**

Given a sample of positive data with Pareto tail, the algorithm computes the optimal number of order statistics that minimizes the variance of the average left truncated tail index estimator, and uses the relationship to the Hill estimator for the Hall class of distributions to derive the optimal number (minimizing the asymptotic mean squared error) of the Hill estimator. The default value for the second order regular variation index is taken to be $\rho = -1$.

Usage

```
thselect.bab(
  xdat,
  kmin = floor(0.2 * length(xdat)),
  kmax = length(xdat) - 1L,
  rho = -1,
  test = FALSE,
  nsim = 999L,
  level = 0.95
)
```

Arguments

xdat	[vector] positive vector of exceedances
kmin	[int] minimum number of exceedances
kmax	[int] maximum number of exceedances for the estimation of the shape parameter.
rho	[double] scalar for the second order regular variation index, a negative number.
test	[logical] if TRUE, computes the goodness-of-fit statistic for the model using Monte Carlo
nsim	[int] number of replications for Monte Carlo test, used only if test=TRUE.
level	[double] confidence level for test

Value

a list with the number of order statistics for the Hill estimator, `k0` and the corresponding shape estimate `shape`, the average lower-trimmed Hill estimator `shape._1th` and the number of order statistics upon which the latter is based, `k0_1th`.

References

Bladt, M., Albrecher, H. & Beirlant, J. (2020) *Threshold selection and trimming in extremes*. *Extremes*, 23, 629-665 . doi:[10.1007/s10687020003850](https://doi.org/10.1007/s10687020003850)

thselect.cv

*Threshold selection via coefficient of variation***Description**

This function computes the empirical coefficient of variation and computes a weighted statistic comparing the squared distance with the theoretical coefficient variation corresponding to a specific shape parameter (estimated from the data using a moment estimator as the value minimizing the test statistic, or using maximum likelihood). The procedure stops if there are no more than 10 exceedances above the highest threshold.

Usage

```
thselect.cv(
  xdat,
  thresh,
  method = c("mle", "wcv", "cv"),
  nsim = 999L,
  nthresh = 10L,
  level = 0.05,
  lazy = FALSE,
  plot = FALSE
)
```

Arguments

xdat	[vector] vector of observations
thresh	[vector] vector of threshold. If missing, set to p^k for $k = 0$ to $k = \text{nthresh}$
method	[string], either moment estimator for the (weighted) coefficient of variation (wcv and cv) or maximum likelihood (mle)
nsim	[integer] number of bootstrap replications
nthresh	[integer] number of thresholds, if thresh is not supplied by the user
level	[numeric] probability level for sequential testing procedure
lazy	[logical] compute the bootstrap p-value until the test stops rejecting at level level? Default to FALSE
plot	[logical] if TRUE, returns a plot of the p-value path

Value

a list with elements

- thresh: value of threshold returned by the procedure, NA if the hypothesis is rejected at all thresholds
- thresh0: sorted vector of candidate thresholds
- cindex: index of selected threshold among thresh0 or NA if none returned

- pval: bootstrap p-values, with NA if lazy and the p-value exceeds level at lower thresholds
- shape: shape parameter estimates
- nexc: number of exceedances of each threshold thresh0
- method: estimation method for the shape parameter

Note

The authors suggest transformation of

$$Y = -1/(X + c) + 1/c,$$

where X are exceedances and $c = \sigma/\xi$ is the ratio of estimated scale and shape parameters. For heavy-tailed distributions with $\xi > 0.25$, this may be preferable, but must be conducted outside of the function.

References

del Castillo, J. and M. Padilla (2016). *Modelling extreme values by the residual coefficient of variation*, SORT, 40(2), pp. 303–320.

Examples

```
thselect.cv(
  xdat = rgp(1000),
  thresh = qgp(seq(0,0.9, by = 0.1)),
  nsim = 99,
  lazy = TRUE,
  plot = TRUE)
```

thselect.expgqt

Generalized quantile threshold selection

Description

The methodology proposed by Beirlant, Vynckier and Teugels (1996) uses an asymptotic expansion of the mean squared error for Hill's estimator given a random sample with Pareto tails and positive shape, using an exponential regression. The value of k is selected to minimize the mean squared error given optimal weighting scheme. This depends on the order of regular variation ρ , which is obtained based on the slope of the difference in Hill estimators, suitably reweighted. The iterative procedure of Beirlant et al. alternates between parameter estimation until convergence. It returns the Hill shape estimate, the number of higher order statistic, the parameter ρ and estimates of the standard error of the shape and the mean squared error, based on the ultimate parameter values. Since the weights can become negative, there is no guarantee that the mean squared error estimate is positive, nor that the estimated value of ρ is nonpositive.

Usage

```
thselect.expgqt(
  xdat,
  maxiter = 10L,
  tol = 2,
  kmin = max(10, floor(length(xdat)/100)),
  kmax = floor(0.8 * length(xdat)),
  ...
)
```

Arguments

xdat	[vector] sample of exceedances
maxiter	[int] maximum number of iteration
tol	[double] tolerance for difference in value of k for the fixed point
kmin	[int] minimum number of exceedances for the estimator
kmax	[int] maximum number of exceedances for the estimator
...	additional arguments, currently ignored

Value

a list with components

- shape the Hill estimator of the shape, based on the k largest order statistics
- k_0 number of high order statistics for estimation of the shape using Hill's estimator
- rho estimate of the second order regular variation parameter
- mse mean squared error estimate of the shape parameter
- se standard error of the shape parameter
- convergence logical; if TRUE, indicates that the method converged to a fixed point within tol before reaching the maximum number of iterations maxiter

References

Beirlant, J., Vynckier, P., & Teugels, J. L. (1996). Excess Functions and Estimation of the Extreme-Value Index. *Bernoulli*, 2(4), 293–318. doi:10.2307/3318416

Examples

```
# Simulate Pareto data - log(xdat) is exponential with rate 2
xdat <- rgp(n = 200, loc = 1, scale = 0.5, shape = 0.5)
(thselect.expgqt(xdat))
```

thselect.gbw	<i>Kernel-based threshold selection of Goegebeur, Beirlant and de Wet (2008)</i>
--------------	--

Description

Kernel-based threshold selection of Goegebeur, Beirlant and de Wet (2008)

Usage

```
thselect.gbw(
  xdat,
  kmax,
  kernel = c("Jackson", "Lewis"),
  rho = c("gbw", "ghp", "fagh", "dk"),
  ...
)
```

Arguments

xdat	[vector] sample exceedances
kmax	[int] maximum number of exceedances considered
kernel	[string] kernel choice, one of Jackson or Lewis
rho	string for the estimator of the second order regular variation. Can also be a negative scalar
...	additional arguments, for backward compatibility purposes

Value

a list with elements

- k0: number of exceedances
- shape: Hill's shape estimate
- rho: second-order regular variation parameter estimate
- gof: goodness-of-fit statistic for the chosen threshold.

References

Goegebeur , Y., Beirlant , J., and de Wet , T. (2008). Linking Pareto-Tail Kernel Goodness-of-fit Statistics with Tail Index at Optimal Threshold and Second Order Estimation. REVSTAT-Statistical Journal, 6(1), 51–69. <doi:10.57805/revstat.v6i1.57>

Examples

```
xdat <- rgp(n = 1000, scale = 2, shape = 0.5)
(thselect.gbw(xdat, kmax = 500))
```

thselect.ksmd*Mahalanobis distance-based methodology*

Description

Compute the Mahalanobis distance-based threshold method over a grid of thresholds by transforming data from generalized Pareto to unit exponential based on probability weighted moment estimates, then computing the first L-moment and the L-skewness. The latter are compared to the theoretical counterparts from a unit exponential sample of the same size, which is used to compute the Mahalanobis distance. The threshold returned is the one which minimizes the distance.

Usage

```
thselect.ksmd(xdat, thresh, approx = c("asymptotic", "mc"), nsim = 1000L)
```

Arguments

xdat	[numeric] vector of observations
thresh	[numeric] vector of candidate thresholds. If missing, 20 sample quantiles starting at the 0.25 quantile in increments of 3.75 percent.
approx	[string] method to use to obtain moments of first L-moment
nsim	[integer] number of replications for Monte Carlo approximation

Value

a list with components

- thresh0: selected threshold returned by the procedure
- thresh: vector of candidate thresholds
- pval: scalar p -value for the chi-square approximation to the test statistic for the selected threshold
- dist: vector of Mahalanobis distance
- approx: type of approximation

References

Kiran, K. G. and Srivinas, V.V. (2021). *A Mahalanobis distance-based automatic threshold selection method for peaks over threshold model*. Water Resources Research 57. <doi:10.1029/2020WR027534>

thselect.mdps	<i>Minimum distance threshold selection procedure</i>
---------------	---

Description

Minimum distance threshold selection procedure

Usage

```
thselect.mdps(xdat)
```

Arguments

xdat vector of positive exceedances

Value

a list with components

k0 order statistic corresponding to threshold (number of exceedances)

shape Hill's estimator of the tail index based on k0 exceedances

thresh0 numerical value of the threshold, the n-k0+1 order statistic of the original sample

References

Clauset, A., Shalizi, C.R. and Newman, M.E.J. (2009). *Power-Law Distributions in Empirical Data*. SIAM Review. Society for Industrial and Applied Mathematics, **51**, 661-703, doi:[10.1137/070710111](https://doi.org/10.1137/070710111)

thselect.mrl	<i>Automated mean residual life plots</i>
--------------	---

Description

This function implements the automated proposal from Section 2.2 of Langousis et al. (2016) for mean residual life plots. It returns the threshold that minimize the weighted mean square error and moment estimators for the scale and shape parameter based on weighted least squares.

Usage

```
thselect.mrl(xdat, thresh, kmax, plot = TRUE, ...)
```

Arguments

xdat	[numeric] vector of observations
thresh	[numeric] vector of thresholds; if missing, uses all order statistics from the 20th largest until kmax as candidates
kmax	[integer] maximum number of order statistics
plot	[logical] if TRUE (default), return a plot of the mean residual life plot with the fitted slope and the chosen threshold
...	additional arguments, currently ignored

Details

The procedure consists in estimating the usual mean residual life as a function of the threshold, and looking for an order statistic or threshold value above which the fit is more or less linear.

Value

a list containing

- thresh: candidate threshold vector
- thresh0: selected threshold
- scale: scale parameter estimate
- shape: shape parameter estimate
- mrl: empirical mean excess values
- xdat: ordered observations
- intercept: intercept for mean excess value at chosen threshold
- slope: slope for mean excess value at chosen threshold
- tmanual: logical; TRUE if the user passed a vector of thresholds

References

Langousis, A., A. Mamalakis, M. Puliga and R. Deidda (2016). *Threshold detection for the generalized Pareto distribution: Review of representative methods and application to the NOAA NCDC daily rainfall database*, Water Resources Research, **52**, 2659–2681.

Examples

```
thselect.mrl(rgp(n = 100))
```

thselect.ncpgp	<i>Northop and Coleman piecewise generalized Pareto threshold selection diagnostic</i>
----------------	--

Description

The model tests the null hypothesis of a generalized Pareto above each threshold in `thresh` against the alternative of a piecewise generalized Pareto model with continuity constraints.

Usage

```
thselect.ncpgp(xdat, thresh, test = "score", plot = FALSE, level = 0.95, ...)
```

Arguments

<code>xdat</code>	[vector] observations
<code>thresh</code>	[vector] candidate thresholds
<code>test</code>	[string] indicating whether to perform score test or likelihood ratio (lr) test. The latter requires fitting the alternative model, and so is more computationally expensive.
<code>plot</code>	[logical]; if TRUE, return a plot with the p-value path.
<code>level</code>	[double] confidence level for confidence interval, defaults to 0.95
<code>...</code>	additional arguments, for backward compatibility purposes

Value

an object of class `mev_thselect_ncpgp` containing the test statistic (`stat`), the p-values (`pval`), the threshold candidates (`thresh`) and the selected threshold (`thresh0`).

thselect.pec	<i>Prediction error C-criterion threshold selection method</i>
--------------	--

Description

This function computes the non-robust Pareto prediction error of Dupuis and Victoria-Feser (2003), termed C-criterion, for the Hill estimator of the shape parameter. The threshold returned is the value of the threshold, taken from order statistics, that minimizes the average prediction error.

Usage

```
thselect.pec(xdat, kmax)
```

Arguments

xdat	vector of observations
kmax	maximum number of order statistics to consider. Default to sample size if left unspecified.

Value

a list with the number of exceedances k , the chosen threshold thresh0 and the corresponding Hill estimator shape estimate shape.

References

Dupuis, D.J. and M.-P. Victoria-Feser (2003). A Prediction Error Criterion for Choosing the Lower Quantile in Pareto Index Estimation, University of Geneva, technical report, <https://archive-ouverte.unige.ch/unige:5789>.

thselect.pickands	<i>Pickands' order statistics threshold selection method</i>
-------------------	--

Description

Restricting to the largest fourth of the data, returns the number of exceedances that minimizes the Kolmogorov-Smirnov statistic, i.e., the maximum absolute difference between the estimated generalized Pareto and the empirical distribution of exceedances. Relative to the paper, different estimation methods are proposed.

Usage

```
thselect.pickands(xdat, thresh, method = c("mle", "lmom", "quartiles"))
```

Arguments

xdat	[numeric] vector of observations
thresh	[numeric] vector of candidate thresholds. If missing, defaults to order statistics from the 10th to a quarter of the sample size.
method	[string] estimation method, either the quartiles of Pickands (1975), maximum likelihood, probability weighted moments or L-moments

Value

a list with components

- $k0$: number of exceedances
- thresh0 : selected threshold returned by the procedure
- thresh : vector of candidate thresholds
- dist : vector of Kolmogorov-Smirnoff distance

- method; string for the estimation method
- scale: estimated scale parameter at the chosen threshold
- shape: estimated shape parameter at the chosen threshold

Note

The quartiles estimator of Pickands is robust, but very inefficient. It is provided for historical reasons.

References

James Pickands III (1975). *Statistical inference using extreme order statistics*, Annals of Statistics, 3(1) 119-131, doi:[10.1214/aos/1176343003](https://doi.org/10.1214/aos/1176343003)

thselect.rbm

Threshold selection for the random block maxima method

Description

Threshold selection for the random block maxima method

Usage

```
thselect.rbm(xdat, kmax = length(xdat))
```

Arguments

xdat	[vector] sample exceedances
kmax	maximum number of exceedances to consider.

Value

a list with elements

thselect.samsee	<i>Threshold selection via SAMSEE</i>
-----------------	---------------------------------------

Description

Smooth asymptotic mean squared error estimator of Schneider et al. (2021) for threshold selection. The implementation uses a second-order regular variation index of -1

Usage

```
thselect.samsee(xdat)
```

Arguments

xdat vector of positive exceedances

Value

a list with elements

k0 optimal number of exceedances

shape Hill estimator of the tail index

thresh0 selected threshold

References

Schneider, L.F., Krajina, A. & Krivobokova, T. (2021). *Threshold selection in univariate extreme value analysis*, *Extremes*, **24**, 881–913 [doi:10.1007/s10687021004057](https://doi.org/10.1007/s10687021004057)

thselect.sdinfo	<i>Threshold selection diagnostic of Suveges and Davison</i>
-----------------	--

Description

The information matrix test (IMT), proposed by Suveges and Davison (2010), is based on the difference between the expected quadratic score and the second derivative of the log-likelihood. The asymptotic distribution for each threshold u and gap K is asymptotically χ^2 with one degree of freedom. The approximation is good for $N > 80$ and conservative for smaller sample sizes. The test assumes independence between gaps.

Usage

```
thselect.sdinfo(xdat, thresh, qllev, plot = FALSE, kmax = 1)
```

Arguments

xdat	[vector] vector of observations
thresh	[vector] candidate thresholds
qllev	[vector] probability levels to define threshold if thresh is missing.
plot	[logical]; should the graphical diagnostic be plotted?
kmax	[int] the largest K-gap under consideration for clusters

Details

The procedure proposed in Suveges & Davison (2010) was corrected for erratas. The maximum likelihood is based on the limiting mixture distribution of the intervals between exceedances (an exponential with a point mass at zero). The condition $D^{(K)}(u_n)$ should be checked by the user.

Fukutome et al. (2015) propose an ad hoc automated procedure

1. Calculate the interexceedance times for each K-gap and each threshold, along with the number of clusters
2. Select the (u, K) pairs for which $IMT < 0.05$ (corresponding to a P-value of 0.82)
3. Among those, select the pair (u, K) for which the number of clusters is the largest

Value

an invisible list of class with elements

- thresh a vector of thresholds based on empirical quantiles at supplied levels.
- stat a matrix of test statistics
- pval a matrix of approximate p-values (corresponding to probabilities under a χ_1^2 distribution)
- mle a matrix of maximum likelihood estimates for each given pair of thresholds and gaps
- loglik a matrix of log-likelihood values at MLE for each given pair of elements in thresh and gap in $0, \dots, kmax$
- quantile quantile levels for thresholds, if supplied by the user
- kmax the largest gap number

Author(s)

Leo Belzile

References

- Fukutome, Liniger and Suveges (2015), Automatic threshold and run parameter selection: a climatology for extreme hourly precipitation in Switzerland. *Theoretical and Applied Climatology*, **120**(3), 403-416.
- Suveges and Davison (2010), Model misspecification in peaks over threshold analysis. *Annals of Applied Statistics*, **4**(1), 203-221.
- White (1982), Maximum Likelihood Estimation of Misspecified Models. *Econometrica*, **50**(1), 1-25.

Examples

```
thselect.sinfo(
  xdat = rgp(n = 10000),
  qlev = seq(0.1, 0.9, length = 10),
  kmax = 3)
```

thselect.vmetric	<i>Metric-based threshold selection</i>
------------------	---

Description

Adaptation of Varty et al.'s metric-based threshold automated diagnostic for the independent and identically distributed case with no rounding.

Usage

```
thselect.vmetric(
  xdat,
  thresh,
  B = 199L,
  type = c("eqd", "exp", "qq", "pp"),
  dist = c("l1", "l2"),
  uq = FALSE,
  neval = 1000L,
  ci = 0.95
)
```

Arguments

xdat	vector of observations
thresh	vector of thresholds
B	number of bootstrap replications
type	string indicating scale, either exp for exponential quantile-quantile plot as in Varty et al. (2021) or pp for probability-probability plot (uniform). The method qq or expected quantile discrepancy (eqd) corresponds to Murphy et al. (2024) with the quantiles on the generalized Pareto scale.
dist	string indicating norm, either l1 for absolute error or l2 for quadratic error
uq	logical; if TRUE, generate bootstrap samples accounting for the sampling distribution of parameters
neval	number of points at which to estimate the metric. Default to 1000L
ci	level of symmetric confidence interval. Default to 0.95

Details

The algorithm proceeds by first computing the maximum likelihood algorithm and then simulating datasets from replication with parameters drawn from a bivariate normal approximation to the maximum likelihood estimator distribution.

For each bootstrap sample, we refit the model and convert the quantiles to exponential or uniform variates. The mean absolute or mean squared distance is calculated on these. The threshold returned is the one with the lowest value of the metric.

Value

an invisible list with components

- thresh: scalar threshold minimizing criterion
- thresh0: vector of candidate thresholds
- metric: value of the metric criterion evaluated at each threshold
- type: argument type
- dist: argument dist

References

Varty, Z. and J.A. Tawn and P.M. Atkinson and S. Bierman (2021+), Inference for extreme earthquake magnitudes accounting for a time-varying measurement process.

Murphy, C., Tawn, J. A., & Varty, Z. (2024). *Automated Threshold Selection and Associated Inference Uncertainty for Univariate Extremes*. *Technometrics*, 67(2), 215–224. <doi:10.1080/00401706.2024.2421744>

Examples

```
xdat <- rexp(1000, rate = 1/2)
thresh <- quantile(xdat, prob = c(0.25, 0.5, 0.75))
```

thselect.wseq

Wadsworth's sequential analysis threshold selection

Description

Function to produce diagnostic plots and test statistics for the threshold diagnostics exploiting structure of maximum likelihood estimators based on the non-homogeneous Poisson process likelihood or the coefficient of tail dependence

Usage

```
thselect.wseq(
  xdat,
  thresh,
  qlev,
  model = c("nhpp", "taildep", "rtaildep"),
  npp = 1,
  nsim = 1000L,
  level = 0.95,
  plot = FALSE,
  ...
)
```

Arguments

xdat	a numeric vector or matrix of data to be fitted.
thresh	vector of candidate thresholds.
qlev	vector of probabilities for empirical quantiles used in place of the threshold, used if argument thresh is missing.
model	string specifying whether the univariate or multivariate diagnostic should be used. Either nhpp for the univariate model, or exp (invexp) for the bivariate exponential model with rate (inverse rate) parametrization. See details.
npp	number of observations per period for the non-homogeneous point process model. Default to 1.
nsim	number of Monte Carlo simulations used to assess the null distribution of the test statistic
level	confidence level of intervals, defaults to 0.95
plot	logical; if TRUE, calls the plot routine
...	additional parameters passed to internal routine

Details

The function is a wrapper for the univariate (non-homogeneous Poisson process model) and exponential dependence model applied to the minimum component (tail dependence coefficient). For the latter, the user can select either the rate ("taildep" or inverse rate parameter ("rtaildep"). The inverse rate parametrization works better for uniformity of the p-value distribution under the likelihood ratio test for the changepoint.

For the coefficient of tail dependence, users must provide pairwise minimum of marginally exponentially distributed margins (see example)

Value

an object of class invisible list with components

- thresh0: threshold selected by the likelihood ratio procedure
- thresh: vector of candidate thresholds

- coef: maximum likelihood estimates from all thresholds
- vcov: joint asymptotic covariance matrix for shape ξ or coefficient of tail dependence η , or its reciprocal.
- wn: values of the white noise process
- stat: value of the likelihood ratio test statistic for the changepoint test
- pval: P -value of the likelihood ratio test
- mle: maximum likelihood estimates for the selected threshold
- model: model fitted, either nhpp, exp or invexp
- nsim: number of Monte Carlo simulations for changepoint test
- xdat: vector of observations

Author(s)

Jennifer L. Wadsworth, Léo Belzile

References

Wadsworth, J.L. (2016). Exploiting Structure of Maximum Likelihood Estimators for Extreme Value Threshold Selection, *Technometrics*, **58**(1), 116-126, <http://dx.doi.org/10.1080/00401706.2014.998345>.

Examples

```
## Not run:
set.seed(123)
xdat <- abs(rnorm(5000))
thresh <- quantile(xdat, seq(0, 0.9, by = 0.1))
(diag <- thselect.wseq(
  xdat = xdat,
  thresh = thresh,
  plot = TRUE,
  type = "ps"))
# Multivariate example, with coefficient of tail dependence
xbvn <- rmnorm(n = 6000L,
               mu = rep(0, 2),
               Sigma = cbind(c(1, 0.7), c(0.7, 1)))
thselect.wseq(
  xdat = xbvn,
  qlev = seq(0, 0.9, length.out = 30),
  model = 'taildep',
  plot = TRUE)

## End(Not run)
```

tstab.cv

*Coefficient of variation threshold stability plot***Description**

This function calculates parametric estimates of the coefficient of variation with pointwise Wald confidence intervals along with empirical estimates and returns a threshold stability plot.

Usage

```
tstab.cv(
  xdat,
  thresh,
  method = c("empirical", "mle", "wcv", "cv"),
  nthresh = 10L,
  nsim = 99L,
  plot = TRUE,
  level = 0.95,
  ...
)
```

Arguments

xdat	[vector] vector of observations
thresh	[vector] vector of threshold. If missing, set to p^k for $k = 0$ to $k = \text{nthresh}$
method	[string], either moment estimator for the (weighted) coefficient of variation (wcv and cv) or maximum likelihood (mle)
nthresh	[integer] number of thresholds, if thresh is not supplied by the user
nsim	[integer] number of bootstrap replications
plot	[logical] if TRUE, returns a plot of the p-value path
level	[numeric] probability level for sequential testing procedure
...	additional parameters, notably for package boot, for the type of confidence intervals.

Examples

```
tstab.cv(
  xdat = rgp(1000),
  thresh = qgp(seq(0,0.9, by = 0.1)),
  method = "cv")
tstab.cv(
  xdat = rgp(1000),
  thresh = qgp(seq(0,0.9, by = 0.1)),
  method = "empirical")
```

tstab.egp

*Threshold stability plots for extended generalized Pareto models***Description**

Threshold stability plots for extended generalized Pareto models

Usage

```
tstab.egp(
  xdat,
  thresh,
  model = c("pt-beta", "pt-gamma", "pt-power", "gj-tnorm", "gj-beta", "exptilt",
    "logist"),
  param = c("shape", "kappa"),
  type = c("wald", "profile"),
  transform = FALSE,
  level = 0.95,
  plot = TRUE,
  ...
)
```

Arguments

xdat	vector of observations, greater than the threshold
thresh	threshold value
model	a string indicating which extended family to fit
param	[string] parameter, either shape or additional parameter kappa
type	[string] confidence interval type, either wald or profile.
transform	logical; if TRUE and type="wald", intervals for kappa are computed on the log-scale and back-transformed.
level	[double] confidence interval level, default to 0.95.
plot	[logical] if TRUE (default), return a threshold stability plot
...	additional arguments for the plot function, currently ignored.

Examples

```
xdat <- rgp(n = 1000)
tstab.egp(
  xdat = xdat,
  thresh = c(0, quantile(xdat, 0.5)),
  model = "gj-tnorm",
  param = "kappa",
  transform = TRUE)
```

tstab.gpd

*Parameter stability plots for peaks-over-threshold***Description**

This function computes the maximum likelihood estimate at each provided threshold and plots the estimates (pointwise), along with 95% confidence/credible intervals obtained using Wald or profile confidence intervals, or else from 1000 independent draws from the posterior distribution under vague independent normal prior on the log-scale and shape. The latter two methods better reflect the asymmetry of the estimates than the Wald confidence intervals.

Usage

```
tstab.gpd(
  xdat,
  thresh,
  method = c("wald", "profile", "post"),
  level = 0.95,
  plot = TRUE,
  which = c("scale", "shape"),
  changepar = TRUE,
  ...
)
```

Arguments

xdat	a vector of observations
thresh	a vector of candidate thresholds at which to compute the estimates.
method	string indicating the method for computing confidence or credible intervals. Must be one of "wald", "profile" or "post".
level	confidence level of the intervals. Default to 0.95.
plot	logical; should parameter stability plots be displayed? Default to TRUE.
which	character vector with elements scale or shape
changepar	logical; if TRUE, changes the graphical parameters.
...	additional arguments passed to plot.

Value

a list with components

- threshold: vector of numerical threshold values.
- mle: matrix of modified scale and shape maximum likelihood estimates.
- lower: matrix of lower bounds for the confidence or credible intervals.
- upper: matrix of lower bounds for the confidence or credible intervals.

- method: method for the confidence or coverage intervals.

plots of the modified scale and shape parameters, with pointwise confidence/credible intervals and an invisible data frame containing the threshold thresh and the modified scale and shape parameters.

Note

The function is hard coded to prevent fitting a generalized Pareto distribution to samples of size less than 10. If the estimated shape parameters are all on the boundary of the parameter space (meaning $\hat{\xi} = -1$), then the plots return one-sided confidence intervals for both the modified scale and shape parameters: these typically suggest that the chosen thresholds are too high for estimation to be reliable.

Author(s)

Leo Belzile

See Also

[gpd.fitrange](#)

Examples

```
dat <- abs(rnorm(10000))
u <- qnorm(seq(0.9,0.99, by= 0.01))
par(mfrow = c(1,2))
tstab.gpd(xdat = dat, thresh = u, changepar = FALSE)
## Not run:
tstab.gpd(xdat = dat, thresh = u, method = "profile")
tstab.gpd(xdat = dat, thresh = u, method = "post")

## End(Not run)
```

tstab.hill

Threshold stability plot for Hill estimator

Description

Threshold stability plot for Hill estimator

Usage

```
tstab.hill(xdat, kmax, method = "hill", ..., log = TRUE)
```

Arguments

xdat	[vector] sample exceedances
kmax	[int] maximum number of order statistics
method	[string] name of estimator for shape parameter. Default to hill.
...	additional arguments passed to <code>fit.shape</code> for certain methods.
log	[logical] should the x-axis for the number of order statistics used for estimation be displayed on the log scale? Default to TRUE

Value

a plot of shape estimates as a function of the number of exceedances

Examples

```
xdat <- rgp(n = 250, loc = 1, scale = 2, shape = 0.5)
tstab.hill(xdat)
```

tstab.lthill	<i>Threshold stability plots for left-truncated Hill estimators</i>
--------------	---

Description

Given a vector of exceedances and some potential choices of k for the threshold, compute the left-truncated Hill estimators for each value of k and use these to compute the variance and slope of the estimator

Usage

```
tstab.lthill(xdat, k, which = c("lthill", "var", "slope"), log = TRUE, ...)
```

Arguments

xdat	[numeric] vector of positive observations
k	[integer] number of order statistics for the threshold
which	[string] the type of plot, showing the left-truncated Hill plot on the log, the log of the variance of the estimator, or the log slope
log	[logical] if TRUE (default), shows the Hill plot on the log-scale
...	additional parameters for color, etc. to be passed to plot

Value

an invisible list with `lthill`, order statistics, the log variance and the log scale.

References

Bladt, M., Albrecher, H. & Beirlant, J. (2020) *Threshold selection and trimming in extremes*. *Extremes*, 23, 629-665 . doi:[10.1007/s10687020003850](https://doi.org/10.1007/s10687020003850)

Examples

```
xdat <- 10/(1 - runif(n = 1000)) - 10
tstab.lthill(xdat = xdat, k = c(50,100,200))
```

tstab.mrl	<i>Mean residual life plot</i>
-----------	--------------------------------

Description

Computes mean of sample exceedances over a range of thresholds or for a pre-specified number of largest order statistics, and returns a plot with 95% Wald-based confidence intervals as a function of either the threshold or the number of exceedances. The main purpose is the plotting method, which generates the so-called mean residual life plot. The latter should be approximately linear over the threshold for a generalized Pareto distribution

Usage

```
tstab.mrl(
  xdat,
  thresh,
  kmin = 10L,
  kmax = length(xdat),
  plot = TRUE,
  level = 0.95,
  xlab = c("thresh", "nexc"),
  type = c("band", "ptwise"),
  ...
)
```

Arguments

xdat	vector of sample observations
thresh	vector of thresholds
kmin	integer giving the minimum number of exceedances; ignored if thresh is provided. Default to 10
kmax	integer giving the maximum number of exceedances; ignored if thresh is provided. Default to sample size.
plot	logical; if TRUE, call the plot method
level	double giving the level of confidence intervals for the plot, default to 0.95

xlab	string indicating whether to use thresholds (thresh) or number of largest order statistics (nexc) for the x-axis
type	string whether to plot pointwise confidence intervals using segments ("ptwise") or using dashed lines ("band")
...	additional arguments, currently ignored

Value

an invisible list with mean sample exceedances and standard deviation, number of exceedances, threshold

References

Davison, A.C. and R.L. Smith (1990). Models for Exceedances over High Thresholds (with discussion), *Journal of the Royal Statistical Society. Series B (Methodological)*, **52**(3), 393–442.

Examples

```
tstab.mrl(
  xdat = rgp(n = 100, shape = -0.5),
  xlab = "thresh",
  kmax = 50)
tstab.mrl(
  rexp(100),
  thresh = qexp(seq(0, 0.9, by = 0.01)))
```

venice

*Venice Sea Levels***Description**

The venice data contains the 10 largest yearly sea levels (in cm) from 1887 until 2019. Only the yearly maximum is available for 1922 and the six largest observations for 1936.

Format

a data frame with 133 rows and 11 columns containing the year of the measurement (first column) and ordered 10-largest yearly observations, reported in decreasing order from largest (r1) to smallest (r10).

Note

Smith (1986) notes that the annual maxima seems to fluctuate around a constant sea level up to 1930 or so, after which there is potential linear trend. Records of threshold exceedances above 80 cm (reported on the website) indicate that observations are temporally clustered.

The observations from 1931 until 1981 can be found in Table 1 in Smith (1986), who reported data from Pirazzoli (1982). The values from 1983 until 2019 were extracted by Anthony Davison from the City of Venice website (accessed in May 2020) and are licensed under the CC BY-NC-SA 3.0 license. The Venice City website indicates that later measurements were recorded by an instrument located in Punta Salute.

Source

City of Venice, Historical archive <<https://www.comune.venezia.it/node/6214>>. Last accessed November 5th, 2020.

References

Smith, R. L. (1986) Extreme value theory based on the r largest annual events. *Journal of Hydrology* **86**, 27–43.

Pirazzoli, P., 1982. Maree estreme a Venezia (periodo 1872-1981). *Acqua Aria* **10**, 1023-1039.

Coles, S. G. (2001) *An Introduction to Statistical Modelling of Extreme Values*. London: Springer.

See Also

[venice](#)

w1500m

Best 200 times of Women 1500m Track

Description

200 all-time best performance (in seconds) of women 1500-meter run.

Format

a vector of size 200

Source

<http://www.alltime-athletics.com/w_1500ok.htm>, accessed 14.08.2018

xdep.asym

Coefficient of extremal asymmetry

Description

This function implements estimators of the bivariate coefficient of extremal asymmetry proposed in Semadeni's (2021) PhD thesis. Two estimators are implemented: one based on empirical distributions, the second using empirical likelihood.

Usage

```
xdep.asym(
  xdat,
  qlev = NULL,
  nq = 40,
  qlim = c(0.8, 0.99),
  estimator = c("emp", "elik"),
  confint = c("none", "wald", "bootstrap"),
  level = 0.95,
  B = 999L,
  ties.method = "random",
  plot = TRUE,
  ...
)
```

Arguments

xdat	an n by 2 matrix of observations
qlev	vector of quantile levels at which to evaluate extremal asymmetry
nq	integer; number of quantiles at which to evaluate the coefficient if u is NULL
qlim	a vector of length 2 with the probability limits for the quantiles
estimator	string indicating the estimation method, one of emp or empirical likelihood (elik)
confint	string for the method used to derive confidence intervals, either none (default) or a nonparametric bootstrap
level	probability level for confidence intervals, default to 0.95 or bounds for the interval
B	integer; number of bootstrap replicates (if applicable)
ties.method	string; method for handling ties. See the documentation of rank for available options.
plot	logical; if TRUE, return a plot.
...	additional arguments for backward compatibility

Details

Let U, V be uniform random variables and define the partial extremal dependence coefficients

$$\varphi_+(u) = \Pr(V > U \mid U > u, V > u),$$

,

$$\varphi_-(u) = \Pr(V < U \mid U > u, V > u),$$

$$\varphi_0(u) = \Pr(V = U \mid U > u, V > u).$$

Define

$$\varphi(u) = \frac{\varphi_+ - \varphi_-}{\varphi_+ + \varphi_-}$$

and the coefficient of extremal asymmetry as $\varphi = \lim_{u \rightarrow 1} \varphi(u)$.

The empirical likelihood estimator, derived for max-stable vectors with unit Frechet margins, is

$$\hat{\varphi}_{\text{el}} = \frac{\sum_i p_i \mathbf{I}(w_i \leq 0.5) - 0.5}{0.5 - 2 \sum_i p_i (0.5 - w_i) \mathbf{I}(w_i \leq 0.5)}$$

where p_i is the empirical likelihood weight for observation i , $\mathbf{I}$ is an indicator function and w_i is the pseudo-angle associated to the first coordinate, derived based on exceedances above u .

Value

an invisible data frame with columns

qllev quantile level of thresholds

coef extremal asymmetry coefficient estimates

lower either NULL or a vector containing the lower bound of the confidence interval

upper either NULL or a vector containing the lower bound of the confidence interval

References

Semadeni, C. (2020). Inference on the Angular Distribution of Extremes, PhD thesis, EPFL, no. 8168.

Examples

```
## Not run:
samp <- rmev(n = 1000,
            d = 2,
            param = 0.2,
            model = "log")
xdep.asym(samp, confint = "wald")
xdep.asym(samp, method = "emplik", confint = "none")

## End(Not run)
```

xdep.chi

Coefficient of tail correlation

Description

The coefficient of tail correlation χ is

$$\chi = \lim_{u \rightarrow 1} \frac{\Pr(F_1(X_1) > u, \dots, F_D(X_D) > u)}{1 - u}.$$

Asymptotically independent vectors have $\chi = 0$. The estimator uses an estimator of the survival copula

Usage

```
xdep.chi(
  xdat,
  qlev = NULL,
  nq = 40,
  qlim = c(0.8, 0.99),
  estimator = c("emp", "betacop", "gpd", "hill"),
  confint = c("wald", "lrt"),
  level = 0.95,
  margtrans = c("emp", "none"),
  ties.method = "random",
  plot = TRUE,
  ...
)
```

Arguments

<code>xdat</code>	an n by d matrix of multivariate observations
<code>qlev</code>	vector of percentiles between 0 and 1
<code>nq</code>	number of quantiles of the structural variable at which to form a grid; only used if <code>u = NULL</code> .
<code>qlim</code>	limits for the sequence <code>u</code> of the structural variable
<code>estimator</code>	string giving estimator to employ
<code>confint</code>	string indicating the type of confidence interval, one of "wald" or "lrt"
<code>level</code>	the confidence level required (default to 0.95).
<code>margtrans</code>	string giving the marginal transformation, one of emp for rank-based transformation or none if data are already on the uniform scale
<code>ties.method</code>	string indicating the type of method for rank; see rank for a list of options. Default to "random"
<code>plot</code>	logical; if TRUE, return a plot
<code>...</code>	additional arguments to <code>taildep</code> , currently ignored

Value

a data frame

- `qlev`: quantile level of estimates
- `coef`: point estimates
- `lower`: lower bound of confidence interval
- `upper`: lower bound of confidence interval

Examples

```
## Not run:
set.seed(765)
# Max-stable model
dat <- rmev(n = 1000, d = 2, param = 0.7, model = "log")
xdep.chi(dat, confint = 'wald')

## End(Not run)
```

xdep.chibar	<i>Coefficient chi-bar</i>
-------------	----------------------------

Description

For data with unit Pareto margins, the coefficient $\bar{\chi} = 2\eta - 1$ is defined via

$$\Pr(\min(X) > x) = L(x)x^{-1/\eta},$$

where $L(x)$ is a slowly varying function. Ignoring the latter, several estimators of η can be defined. In unit Pareto margins, η is a nonnegative shape parameter that can be estimated by fitting a generalized Pareto distribution above a high threshold. In exponential margins, η is a scale parameter and the maximum likelihood estimator of the latter is the Hill estimator. Both methods are based on peaks-over-threshold and the user can choose between pointwise confidence obtained through a likelihood ratio test statistic ("lrt") or the Wald statistic ("wald").

Usage

```
xdep.chibar(
  xdat,
  qlev = NULL,
  nq = 40,
  qlim = c(0.8, 0.99),
  estimator = c("emp", "betacop"),
  confint = c("wald", "lrt"),
  level = 0.95,
  margtrans = c("emp", "none"),
  ties.method = "random",
  plot = TRUE,
  ...
)
```

Arguments

xdat	an n by d matrix of multivariate observations
qlev	vector of percentiles between 0 and 1
nq	number of quantiles of the structural variable at which to form a grid; only used if u = NULL.

qlim	limits for the sequence u of the structural variable
estimator	string giving estimator to employ
confint	string indicating the type of confidence interval, one of "wald" or "lrt"
level	the confidence level required (default to 0.95).
margtrans	string giving the marginal transformation, one of emp for rank-based transformation or none if data are already on the uniform scale
ties.method	string indicating the type of method for rank; see rank for a list of options. Default to "random"
plot	logical; if TRUE, return a plot
...	additional arguments to taildep, currently ignored

Details

The most common approach for estimation is the empirical survival copula, by evaluating the proportion of sample minima with uniform margins that exceed a given x . An alternative estimator uses a smoothed estimator of the survival copula using Bernstein polynomial, resulting in the so-called betacop estimator. Approximate pointwise confidence intervals for the latter are obtained by assuming the proportion of points is binomial.

Value

a data frame

- qlav: quantile level of estimates
- coef: point estimates
- lower: lower bound of confidence interval
- upper: upper bound of confidence interval

References

Ledford, A.W. and J. A. Tawn (1996), Statistics for near independence in multivariate extreme values. *Biometrika*, **83**(1), 169–187.

Ledford, A.W. and J. A. Tawn (1996), Statistics for near independence in multivariate extreme values. *Biometrika*, **83**(1), 169–187.

Examples

```
## Not run:
set.seed(765)
# Max-stable model
dat <- rmev(n = 1000, d = 2, param = 0.7, model = "log")
xdep.chibar(dat, confint = 'wald')

## End(Not run)
```

xdep.eta

*Coefficient of tail dependence***Description**

For data with unit Pareto margins, the coefficient of tail dependence η is defined via

$$\Pr(\min(X) > x) = L(x)x^{-1/\eta},$$

where $L(x)$ is a slowly varying function. Ignoring the latter, several estimators of η can be defined. In unit Pareto margins, η is a nonnegative shape parameter that can be estimated by fitting a generalized Pareto distribution above a high threshold. In exponential margins, η is a scale parameter and the maximum likelihood estimator of the latter is the Hill estimator. Both methods are based on peaks-over-threshold and the user can choose between pointwise confidence obtained through a likelihood ratio test statistic ("lrt") or the Wald statistic ("wald").

Usage

```
xdep.eta(
  xdat,
  qlev = NULL,
  nq = 40,
  qlim = c(0.8, 0.99),
  estimator = c("emp", "betacop", "gpd", "hill", "kj"),
  confint = c("wald", "lrt"),
  level = 0.95,
  margtrans = c("emp", "sp", "none"),
  ties.method = "random",
  plot = TRUE,
  mqlev = NULL,
  ...
)
```

Arguments

xdat	an n by d matrix of multivariate observations
qlev	vector of percentiles between 0 and 1
nq	number of quantiles of the structural variable at which to form a grid; only used if $u = \text{NULL}$.
qlim	limits for the sequence u of the structural variable
estimator	string giving estimator to employ
confint	string indicating the type of confidence interval, one of "wald" or "lrt"
level	the confidence level required (default to 0.95).
margtrans	string giving the marginal transformation, one of emp for rank-based transformation or none if data are already on the uniform scale

<code>ties.method</code>	string indicating the type of method for rank; see rank for a list of options. Default to "random"
<code>plot</code>	logical; if TRUE, return a plot
<code>mqllev</code>	marginal quantile levels for semiparametric estimation for estimator kj; data above this are modelled using a generalized Pareto distribution. If missing, empirical estimation is used throughout
<code>...</code>	additional arguments to <code>taildep</code> , currently ignored

Details

The most common approach for estimation is the empirical survival copula, by evaluating the proportion of sample minima with uniform margins that exceed a given x . An alternative estimator uses a smoothed estimator of the survival copula using Bernstein polynomial, resulting in the so-called betacop estimator. Approximate pointwise confidence intervals for the latter are obtained by assuming the proportion of points is binomial.

Value

a data frame

- `qllev`: quantile level of estimates
- `coef`: point estimates
- `lower`: lower bound of confidence interval
- `upper`: lower bound of confidence interval

References

Ledford, A.W. and J. A. Tawn (1996), Statistics for near independence in multivariate extreme values. *Biometrika*, **83**(1), 169–187.

Examples

```
## Not run:
set.seed(765)
# Max-stable model
dat <- rmev(n = 1000, d = 2, param = 0.7, model = "log")
xdep.eta(dat, confint = 'wald')

## End(Not run)
```

xdep.xindex

*Extremal index estimators***Description**

The function implements estimators of the extremal index based on interexceedance time and gap of exceedances. The maximum likelihood estimator and iteratively reweighted least square estimators of Suveges (2007) as well as the intervals estimator. The implementation differs from the presentation of the paper in that an iteration limit is enforced to make sure the iterative procedure terminates. Multiple thresholds can be supplied.

Usage

```
xdep.xindex(
  xdat,
  qlev = 0.95,
  estimator = c("wls", "mle", "intervals"),
  confint = c("none", "wald", "lrt"),
  level = 0.95,
  plot = FALSE,
  warn = FALSE,
  ...
)
```

Arguments

<code>xdat</code>	numeric vector of observations
<code>qlev</code>	a vector of quantile levels in (0,1) for estimation of the extremal index. Defaults to 0.95
<code>estimator</code>	a string specifying the chosen method (only one allowed). Must be either <code>wls</code> for weighted least squares, <code>mle</code> for maximum likelihood estimation or <code>intervals</code> for the intervals estimator of Ferro and Segers (2003). Partial match is allowed.
<code>confint</code>	string indicating the type of confidence interval, one of <code>"wald"</code> or <code>"lrt"</code> for <code>estimator="mle"</code> , else <code>"none"</code>
<code>level</code>	the confidence level required (default to 0.95).
<code>plot</code>	logical; if TRUE, plot the extremal index as a function of <code>q</code>
<code>warn</code>	logical; if TRUE, receive a warning when the sample size is too small
<code>...</code>	additional arguments, for backward compatibility

Details

The iteratively reweighted least square is a procedure based on the gaps of exceedances $S_n = T_n - 1$. The model is first fitted to non-zero gaps, which are rescaled to have unit exponential scale. The slope between the theoretical quantiles and the normalized gap of exceedances is $b = 1/\theta$, with intercept $a = \log(\theta)/\theta$. As such, the estimate of the extremal index is based on $\hat{\theta} = \exp(\hat{a}/\hat{b})$. The

weights are chosen in such a way as to reduce the influence of the smallest values. The estimator exploits the dual role of θ as the parameter of the mean for the interexceedance time as well as the mixture proportion for the non-zero component.

The maximum likelihood is based on an independence likelihood for the rescaled gap of exceedances, namely $\bar{F}(u_n)S(u_n)$. The score equation is equivalent to a quadratic equation in θ and the maximum likelihood estimate is available in closed form. Its validity requires however condition $D^{(2)}(u_n)$ to apply; this should be checked by the user beforehand.

A warning is emitted if the effective sample size is less than 50 observations.

Value

a data frame

- qlév: quantile level of estimates
- coef: point estimates
- lower: lower bound of confidence interval
- upper: lower bound of confidence interval

Author(s)

Leo Belzile

References

- Ferro and Segers (2003). Inference for clusters of extreme values, *JRSS: Series B*, **65**(2), 545-556.
- Suveges (2007) Likelihood estimation of the extremal index. *Extremes*, **10**(1), 41-55.
- Suveges and Davison (2010), Model misspecification in peaks over threshold analysis. *Annals of Applied Statistics*, **4**(1), 203-221.
- Fukutome, Liniger and Suveges (2015), Automatic threshold and run parameter selection: a climatology for extreme hourly precipitation in Switzerland. *Theoretical and Applied Climatology*, **120**(3), 403-416.

Examples

```
set.seed(234)
# Moving maxima model with theta=0.5
a <- 1; theta <- 1/(1+a)
sim <- rgev(10001, loc=1/(1+a), scale=1/(1+a), shape=1)
x <- pmax(sim[-length(sim)]*a, sim[-1])
q <- seq(0.9, 0.99, by = 0.01)
xdep.xindex(
  xdat = x,
  qlév = q,
  estimator = "mle",
  confint = "wald")
```

Index

* datasets

- leedspollution, 66
- nutrients, 70
- pandemics, 72

- abisko, 5
- adf, 5
- angmeas, 6
- angmeasdir, 8
- auglag, 28, 35

- chipplot, 108
- confint.eprof, 9

- degp (egpdist), 13
- dgeoaniso, 10
- dgev (gevdist), 47
- dgp (gpdist), 60

- egp, 11, 27
- egp.fit, 27
- egp.pll, 12
- egpdist, 13
- emplik, 15
- eskrain, 16
- expme, 17
- extcoef, 18
- extgp, 21, 23, 26, 27
- extgp.G, 23

- fbvpot, 64
- fit.egp, 24
- fit.extgp, 26
- fit.gev, 28
- fit.gpd, 29
- fit.pp, 32
- fit.rho, 33
- fit.rlarg, 34
- fit.shape, 35
- fit.wgpd, 36
- fpot, 31

- frwind, 37

- geomagnetic, 38
- gev, 38
- gev.abias, 40
- gev.bcor, 40
- gev.bias, 41
- gev.mle, 41
- gev.Nyr, 42
- gev.pll, 10, 43, 75
- gev.tem, 45, 75
- gevdist, 47
- gevN, 48
- gevr, 49
- gpd, 50
- gpd.abias, 51
- gpd.bcor, 52
- gpd.bias, 52
- gpd.boot, 53
- gpd.fit, 30, 31
- gpd.fitrangle, 132
- gpd.lmom, 53
- gpd.mle, 54
- gpd.pll, 10, 55, 75
- gpd.tem, 57, 75
- gpde, 59
- gpdist, 60
- gpdN, 61
- gpdr, 62

- ibvpot, 63
- ismev, 30

- kjtail, 65

- leedspollution, 66
- likmgp, 67

- maiquetia, 68
- maxstable, 69

nidd, 69
 nlm, 30
 nutrients, 70

 optim, 25, 30, 33

 pandemics, 72
 pegp (egpdist), 13
 penultimate, 72
 pgev (gevdist), 47
 pgp (gpdist), 60
 plot.eprof, 74
 plot.fr, 75
 plot.mev_shape_rbm, 76
 plot.mev_thselect_vmetric, 77
 plot.mev_thselect_wadsworth, 77
 plot.mev_tstab_mrl, 78
 pp, 79

 qegp (egpdist), 13
 qgev (gevdist), 47
 qgp (gpdist), 60
 qweissman, 80

 rank, 108, 109, 137, 139, 141, 143
 rbvevd, 88, 89
 rdir, 81
 regp (egpdist), 13
 rgev (gevdist), 47
 rgp (gpdist), 60
 rgparp, 81
 rho.dk, 83
 rho.fagh, 84
 rho.gbw, 85
 rho.ghp, 85
 rlarg, 86
 rmev, 87
 rmevspec, 89, 90
 rmnorm, 92
 rmvevd, 89
 rparp, 93, 95
 rparpcs, 94
 rparpcshr, 96
 rrlarg, 97

 shape.erm, 98
 shape.genjack, 99
 shape.genquant, 99
 shape.hill, 100
 shape.lthill, 101
 shape.moment, 102
 shape.osz, 102
 shape.pickands, 103
 shape.rbm, 104
 shape.trimhill, 104
 shape.vries, 105
 spline.corr, 46, 58, 75
 spunif, 106

 taildep, 107
 test.maxstab, 109
 test.scoreindep, 110
 thselect.alrs, 111
 thselect.bab, 112
 thselect.cv, 113
 thselect.expgqt, 114
 thselect.gbw, 116
 thselect.ksmd, 117
 thselect.mdps, 118
 thselect.mrl, 118
 thselect.ncpgp, 120
 thselect.pec, 120
 thselect.pickands, 121
 thselect.rbm, 122
 thselect.samsee, 123
 thselect.sdinfo, 123
 thselect.vmetric, 125
 thselect.wseq, 126
 tstab.cv, 129
 tstab.egp, 130
 tstab.gpd, 131
 tstab.hill, 132
 tstab.lthill, 133
 tstab.mrl, 134

 venice, 135, 136

 w1500m, 136

 xdep.asym, 136
 xdep.chi, 138
 xdep.chibar, 140
 xdep.eta, 142
 xdep.xindex, 144