Package 'mev'

August 29, 2016

Type Package

Title Multivariate Extreme Value Distributions

Version 1.8.2

Date 2016-08-22

Description Exact simulation from max-stable processes and multivariate extreme value distributions for various parametric models. Threshold selection methods.

License GPL-3

Imports boot, evd, gmm, ismev, methods, nleqslv, numDeriv, Rcpp (>= 0.11.6), rootSolve, stats

Suggests cobs

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 5.0.1

NeedsCompilation yes

Author Leo Belzile [aut, cre],
Jennifer L. Wadsworth [aut],
Paul J. Northrop [aut],
Scott D. Grimshaw [aut],
Jin. Zhang [ctb],
Michael A. Stephens [ctb],
Art B. Owen [ctb],
Anthony C. Davison [ctb],
Raphael Huser [aut]

Maintainer Leo Belzile <leo.belzile@epfl.ch>

Repository CRAN

Date/Publication 2016-08-24 18:38:38

R topics documented:

`mev-package` *Multivariate Extreme Value Distributions*

Description

Exact simulation from max-stable processes and multivariate extreme value distributions for various parametric models.

Details

The package allows exact generation of multivariate extreme value vectors or max-stable processes. For the latter, the user can provide a variogram function along with a set of locations that serve as input. Models implemented include the 1-parameter logistic and negative logistic as described in the article, the bilogistic and Coles and Tawn extremal Dirichlet model using the algorithm of Boldi (2009) and the Dirichlet mixture. The extremal Student and Husler-Reiss (Brown-Resnick) models are also implemented.

Other features of the package include threshold diagnostic tests and other tools not found otherwise in packages evd and ismev.

Author(s)

Leo Belzile

Maintainer: <leo.belzile@epfl.ch>

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.

`angmeas` *Rank-based transformation to angular measure*

Description

The method uses the pseudo-polar transformation for suitable norms, transforming the data to pseudo-observations, than marginally to unit Frechet or unit Pareto. Empirical or Euclidean weights are computed and return alongside of the angular and radial sample

Usage

```
angmeas(x, Rnorm = c("l1", "l2", "linf"), Anorm = c("l1", "l2", "linf",
  "arctan"), marg = c("Frechet", "Pareto"), wgt = c("Euclidean",
  "Empirical"))
```

Arguments

`x`	an `n` by `d` sample matrix
`Rnorm`	the norm for the radial component
`Anorm`	the norm for the angular component. `arctan` is only implemented for $d = 2$
`marg`	choice of marginal transformation, either to Frechet or Pareto scale
`wgt`	weighting function for the equation. Can be based on Euclidean or empirical likelihood for the mean

Value

a list with arguments `ang` for the $d - 1$ pseudo-angular sample, `rad` with the radial component and `wts` if `Rnorm` is set to `"l1"` (default).

a list with components

- `ang` matrix of pseudo-angular observations
- `rad` vector of radial contributions
- `wts` empirical or Euclidean likelihood weights for 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(x=x, Rnorm="l1", Anorm="l1", marg="Frechet", wgt="Empirical")
wts2 <- angmeas(x=x, Rnorm="l2", Anorm="l2", marg="Pareto", wgt="Euclidean")
```

`confint.fr` *Confidence intervals for profile likelihood derived from TEM*

Description

This function uses spline interpolation to derive `level` confidence intervals using the output of either gev.tem or gpd.tem.

Usage

```
## S3 method for class 'fr'
confint(object, parm, level = 0.95, ...)
```

Arguments

`object`	an object of class `fr`, normally the output of gpd.tem or gev.tem.
`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 0.95
`...`	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

a 2 by 3 matrix containing point estimates, lower and upper confidence intervals based on `r` and `rstar`

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). The latter include an additional parameter, κ. All three families share the same tail index than the GP model, while allowing for lower thresholds. In the case $\kappa = 1$, the models reduce to the generalised Pareto.

`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
`p`	extreme event probability; `p` must be greater than the rate of exceedance for the calculation to make sense. See **Details**.
`plot`	boolean indicating whether or not to plot 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.

`egp.retlev` returns a plot of the return levels if `plot=TRUE` and a matrix of return levels.

Usage

```
egp.ll(xdat, thresh, par, model=c("egp1","egp2","egp3"))
egp.retlev(xdat, thresh, par, model=c("egp1","egp2","egp3"), 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.

Examples

```
set.seed(123)
xdat <- evd::rgpd(1000, loc=0, scale=1, shape=0.5)
par <- egp.fit(xdat, thresh=0, model="egp3")$par
p <- c(1/1000,1/1500,1/2000)
egp.retlev(xdat, 0, par, "egp3",p)
#With multiple thresholds
th <- c(0,0.1,0.2,1)
opt <- egp.fitrange(xdat, th, model="egp1",plots=NA)
egp.retlev(xdat, opt$thresh, opt$par, "egp1",p=p)
opt <- egp.fitrange(xdat, th, model="egp2",plots=NA)
egp.retlev(xdat, opt$thresh, opt$par, "egp2",p=p)
opt <- egp.fitrange(xdat, th, model="egp3",plots=NA)
egp.retlev(xdat, opt$thresh, opt$par, "egp3",p=p)
```

`egp.fit` *Fit of extended GP models and parameter stability plots*

Description

The function `egp.fitrange` provides classical parameter stability plot for (κ, σ, ξ). The fitted parameter values are displayed with pointwise normal 95% confidence intervals. 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. `egp.fitrange` can also be used to fit the model to multiple thresholds.

Usage

```
egp.fit(xdat, thresh, model = c("egp1", "egp2", "egp3"), init)

egp.fitrange(xdat, thresh, model = c("egp1", "egp2", "egp3"), plots = 1:3,
  umin, umax, nint)
```

Arguments

`xdat`	vector of observations, greater than the threshold
`thresh`	threshold value
`model`	a string indicating which extended family to fit
`init`	vector of initial values, with $\log(\kappa)$ and $\log(\sigma)$; can be omitted.
`plots`	vector of integers specifying which parameter stability to plot (if any); passing `NA` results in no plots
`umin`	optional minimum value considered for threshold (if `thresh` is not provided)
`umax`	optional maximum value considered for threshold (if `thresh` is not provided)
`nint`	optional integer number specifying the number of thresholds to test.

Details

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

Value

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

`egp.fitrange` returns a plot(s) of the parameters fit over the range of provided thresholds, with pointwise 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.

`egp2` *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
`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

```
pegp2(q, prob=NA, kappa=NA, delta=NA, sigma=NA, xi=NA, type=1)
degp2(x, prob=NA, kappa=NA, delta=NA, sigma=NA, xi=NA, type=1, log=FALSE)
qegp2(p, prob=NA, kappa=NA, delta=NA, sigma=NA, xi=NA, type=1)
regp2(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.

`egp2.fit` *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

```
egp2.fit(data, model = 1, method = c("mle", "pwm"), init, censoring = c(0,
  Inf), rounded = 0, CI = FALSE, R = 1000, ncpus = 1, plots = TRUE)
```

Arguments

`data`	data vector.
`model`	integer ranging from 0 to 4 indicating the model to select (see `egp2`).
`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.
`CI`	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`, `egp2`

Examples

```
library(ismev)
data(rain)
egp2.fit(rain[rain>0], model=1, method="mle", init=c(0.9, gp.fit(rain,0, show=FALSE)$est),
 rounded=0.1,CI=TRUE, R=20)
```

`egp2.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 (`degp2.G`), probabilities (`qegp2.G`) or quantiles (`pegp2.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

```
pegp2.G(u, type=1,prob, kappa, delta)
degp2.G(u,type=1, prob=NA, kappa=NA, delta=NA, log=FALSE)
qegp2.G(u,type=1, prob=NA,kappa=NA,delta=NA)
regp2.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

`egp2`

`emplik` *Self-concordant empirical likelihood for a vector mean*

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

`dat`	`n` by `d` matrix of `d`-variate observations
`mu`	`d` vector of hypothesized mean of `dat`
`lam`	starting values for Lagrange multiplier vector, default to zero vector
`eps`	lower cutoff for $-\log$, with default `1/nrow(dat)`
`M`	upper cutoff for $-\log$.
`thresh`	convergence threshold for log likelihood (default of `1e-30` is agressive)
`itermax`	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.

`ext.index` *Extremal index estimators based on interexceedance time and gap of exceedances*

Description

The function implements the maximum likelihood estimator and iteratively reweigthed 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. sMultiple thresholds can be supplied

Usage

```
ext.index(x, q = 0.95, method = c("wls", "mle", "intervals"),
  plot = FALSE)
```

Arguments

`x`	a vector containing of data points
`q`	a vector of quantile levels in (0,1). Defaults to 0.95
`method`	a string specifying the chosen method. Must be either `wls` for weighted least squares, `mle` for maximum likelihood estimation or `intervals` for the intervals estimator of Ferro and Segers (2003). Partial match is allowed.
`plot`	a boolean specifying whether to plot the extremal index as a function of `q`

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 vector or matrix of estimated extremal index of dimension `length(method)` by `length(q)`.

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 <- evd::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)
ext.index(x=x,q=q,method=c("wls","mle"))
```

---

gev *Generalized extreme value distribution*

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)
```

Note

The Gumbel case is not currently handled.

Functions

- `gev.ll`: log-likelihood
- `gev.ll.optim`: negative log-likelihood parametrized in terms of location, `log(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.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", "VaR"), dat, psi = NULL,
  p = NULL, n.psi = 50, plot = TRUE)
```

Arguments

param	parameter over which to profile
dat	sample vector for the GEV distribution
psi	scalar or ordered vector of values for the interest parameter. If NULL (default), a grid of values centered at the MLE is selected
p	probability associated with the (1-p)th quantile for return levels if param="VaR".
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
plot	logical indiating whether plot.fr should be called upon exit

Value

an object of class fr (see tem) with elements

- normal: maximum likelihood estimate and standard error of the interest parameter psi
- 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
- param: parameter

Author(s)

Leo Belzile, from code by A. Davison from the hoa package

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

Description

Likelihood, score function and information matrix, bias, 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
V	vector calculated by gevr.Vfun
p	probability, corresponding to $(1-p)$th quantile for z

Usage

```
gevr.ll(par, dat, p)
gevr.ll.optim(par, dat, p)
gevr.score(par, dat, p)
gevr.infomat(par, dat, p)
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(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

gp.fit *Peaks-over-threshold modelling using the generalized Pareto distribution*

Description

Numerical optimization of the Generalized Pareto distribution over a high threshold.

Usage

```
gp.fit(xdat, threshold, method = c("Grimshaw", "nlm", "optim", "ismev", "zs",
  "zhang"), show = FALSE, MCMC = NULL)
```

Arguments

xdat	a numeric vector of data to be fitted.
threshold	the chosen threshold.
method	the method to be used. See **Details**. Can be abbreviated.
show	logical; if TRUE (the default), print details of the fit.
MCMC	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 niter, burnin and thin to the sampler as a list.

Details

The default method is "Grimshaw", consisting in maximization of the profile likelihood for the scale. Other options for maximization of the profile likelihood are nlm and optim, which use respectively nlm and optim. Method "ismev" is the two-dimensional optimization routine gpd.fit from the ismev 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 all parameters (optimized and fixed).
- std.err a vector containing the standard errors.
- var.cov 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.
- deviance the deviance at the maximum likelihood estimates.

- 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.

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 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 other frequentist functions. Zhang and Stephens (2009) and Zhang (2010) for the zs and zhang approximate methods and L. Belzile for the wrapper and MCMC samplers.

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.

See Also

fpot and gpd.fit

Examples

```
library(ismev)
data(rain)
threshold <- quantile(rain,0.9)
gp.fit(rain, threshold, method="Grimshaw")
gp.fit(rain, threshold, method="zs")
```

---

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.infomat(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.infomat`: 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.bcor` *Bias correction using Firth's modified score function or bias substraction*

Description

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` - additionally, passing a `par` argument with shape less than -1/3 will return an error if `method="subtract"` is selected, as the bias correction does not exist then. For small samples, expected and observed information can return very different estimates.

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"

Value

vector of bias-corrected parameters

Examples

```
set.seed(1)
dat <- evd::rgpd(n=40, scale=1, shape=-0.2)
par <- mev::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.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 fail may fit and return an error.

Usage

```
gpd.tem(param = c("scale", "shape", "VaR", "ES"), psi = NULL, m = NULL,
  dat, n.psi = 50, plot = FALSE)
```

Arguments

param	parameter over which to profile
psi	scalar or ordered vector of values for the interest parameter. If NULL (default), a grid of values centered at the MLE is selected
m	number of observations (year) of interest. Required only for param="VaR" or param="ES".
dat	sample vector for the GP distribution

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
plot	logical indiating whether plot.fr should be called upon exit

Value

an object of class fr (see tem) with elements

- normal: maximum likelihood estimate and standard error of the interest parameter psi
- 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
- param: parameter

Author(s)

Leo Belzile, from code by A. Davison from the hoa package

Examples

```
set.seed(123)
dat <- evd::rgpd(n=40, scale=1, shape=-0.1)
#with plots
m1 = gpd.tem(param="shape", n.psi=50, dat=dat,plot=TRUE)
m2 = gpd.tem(param="scale", n.psi=50, dat=dat)
m3 = gpd.tem(param="VaR", n.psi=50, dat=dat, m=100)
#Providing psi
m4 = gpd.tem(param="ES", dat=dat, m=100, psi = c(seq(2,5,length=15),seq(5, 35, length=45)))
plot(m4, c(2,4)) #displays numerical instability for values of r around zero
plot(fr4 <- spline.corr(m4), which=c(2,4))
confint(m1)
confint(m4, parm=2, warn=FALSE)
```

---

gpde *Generalized Pareto distribution (expected shortfall parametrization)*

Description

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

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
V	vector calculated by gpde.Vfun

Details

The observed information matrix was calculated using the second Bartlett identity as the negative of the second derivative of the log-likelihood function, 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

```
gpde.ll(par, dat, m, tol=1e-5)
gpde.ll.optim(par, dat, m, tol=1e-5)
gpde.score(par, dat, m)
gpde.infomat(par, dat, m
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.infomat: 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

gpdr *Generalized Pareto distribution (return level parametrization)*

Description

Likelihood, score function and information matrix, bias, 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
`V`	vector calculated by `gpdr.Vfun`

Details

The observed information matrix was calculated using the second Bartlett identity as the negative of the second derivative of the log-likelihood function, 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
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(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` was adapted to deal 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

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

Details

The list `fitted` must contain

- `model` a string; see `bvevd` 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

Examples

```
y <- evd::rgpd(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))
```

---

infomat.test *Information matrix test statistic and MLE for the extremal index*

Description

The Information Matrix Test (IMT), proposed by Suveges and Davison (2010) based on White (1982), 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

```
infomat.test(x, q, K, plot = TRUE)
```

Arguments

x	data vector
q	vector of thresholds
K	int specifying the largest K-gap
plot	boolean: should the graphical diagnostic be plotted?

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 matrices containing

- `IMT` a matrix of test statistics
- `pvals` a matrix of approximate p-values (corresponding to probabilities under a χ^2_1 distribution)
- `mle` a matrix of maximum likelihood estimates for each given pair (`u`, `K`)
- `loglik` a matrix of log-likelihood values at MLE for each given pair (`u`, `K`)
- `threshold` a vector of thresholds based on empirical quantiles at supplied levels.
- `q` the vector `q` supplied by the user

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

```
library(evd)
infomat.test(x <- rgpd(n=1000), q = seq(0.9,0.995,length=10), K<- 3)
```

---

mvrnorm *Multivariate Normal distribution sampler*

Description

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

Usage

```
mvrnorm(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 p.s.d., so use at own risk

Value

an n sample from a multivariate Normal distribution

Examples

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

NC.diag *Score and likelihood ratio tests fit of equality of shape over multiple thresholds*

Description

The function returns a P-value path for the score testand/or likelihood ratio test for equality of the shape parameters over multiple thresholds under the generalized Pareto model.

Usage

```
NC.diag(x, u, GP.fit = c("Grimshaw", "nlm", "optim", "ismev"),
  do.LRT = FALSE, size = NULL, my.xlab = NULL, xi.tol = 0.001)
```

Arguments

x	raw data
u	m-vector of thresholds (sorted from smallest to largest)
GP.fit	function used to optimize the generalized Pareto model.
do.LRT	boolean indicating whether to perform the likelihood ratio test (in addition to the score test)
size	level at which a horizontal line is drawn on multiple threshold plot
my.xlab	(optional) x-axis label
xi.tol	numerical tolerance for threshold distance; if the absolute value of xi1.hat is less than xi.tol use linear interpolation to evaluate score vectors, expected Fisher information matrices, Hessians

Details

The default method is "Grimshaw" using the reduction of the parameters to a one-dimensional maximization. Other options are one-dimensional maximization of the profile the nlm function or optim. Two-dimensional optimisation using 2D-optimization ismev using the routine from gpd.fit from the ismev library, with the addition of the algebraic gradient. The choice of GP.fit should make no difference but the options were kept. **Warning**: the function is not robust and will not recover from failure of the maximization routine, returning various error messages.

Value

a plot of P-values for the test at the different thresholds u

Author(s)

Paul J. Northrop and Claire L. Coleman

References

Grimshaw (1993). Computing Maximum Likelihood Estimates for the Generalized Pareto Distribution, *Technometrics*, **35**(2), 185–191.

Northrop & Coleman (2014). Improved threshold diagnostic plots for extreme value analyses, *Extremes*, **17**(2), 289–303.

Wadsworth & Tawn (2012). Likelihood-based procedures for threshold diagnostics and uncertainty in extreme value modelling, *J. R. Statist. Soc. B*, **74**(3), 543–567.

Examples

```
## Not run:
library(ismev)
data(rain)
u <- quantile(rain, seq(0.85,0.99,by=0.01))
NC.diag(rain, u, size=0.05)

## End(Not run)
```

---

plot.fr *Plot of tangent exponential model profile likelihood*

Description

This function is adapted from plot.fr. It differs 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 rstar and the likelihood modification q as functions of the parameter psi. 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 removing the incorrect corrections and providing smoothed 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.

---

rdir *Random variate generation for Dirichlet distribution on S_d*

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)
```

`rmev` *Exact simulations of multivariate extreme value distributions*

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",
  "dirmix"), alg = c("ef", "sm"), weights, vario, loc, grid = FALSE, ...)
```

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 `rmvevd`, 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.
`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. Must sum to one
`vario`	function specifying the variogram. Used only if provided in conjonction with `loc` and if `sigma` is missing
`loc`	`d` by `k` matrix of location, 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).
`...`	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`: `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 `alpha`
- `dirmix`: `d` by `m`-dimensional matrix of positive (a)symmetry parameters

Stephenson points out that the multivariate asymmetric negative logistic model is not a valid distribution function. The implementation in `mev` uses the same construction as the asymmetric logistic distribution, and 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 conjonction with variograms. The parametrization differs between the two models.

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 variogram, from Clement Dombry
#Variogram gamma(h) = scale*||h||^alpha
scale <- 0.5; alpha <- 1
vario <- function(x) scale*sqrt(sum(x^2))^alpha
#grid specification
grid.loc <- as.matrix(expand.grid(runif(4), runif(4)))
rmev(n=100, vario=vario,loc=grid.loc, model="br")
#Example with a grid (generating an array)
rmev(n=10, sigma=cbind(c(2,1),c(1,3)), loc=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")
```

`rmevspec` *Random samples from spectral distributions of multivariate extreme value models.*

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", "ct", "dirmix"), weights, vario, loc, ...)
```

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.
`weights`	vector of length `m` for the `m` mixture components. Must sum to one

vario	function specifying the variogram. Used only if provided in conjonction with loc and if sigma is missing
loc	d by k matrix of location, 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

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: 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 alpha
- dirmix: d by m-dimensional matrix of positive (a)symmetry 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 (by multiplying by a standard Pareto variable the output). Other functionals are not currently implemented.

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="ct") #with additional scale parameter
```

```
#Variogram gamma(h) = scale*||h||^alpha
scale <- 0.5; alpha <- 1
vario <- function(x) scale*sqrt(sum(x^2))^alpha
#grid specification
grid.loc <- as.matrix(expand.grid(runif(4), runif(4)))
rmevspec(n=100, vario=vario,loc=grid.loc, 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")
```

`smith.penult` *Smith (1987) penultimate approximations*

Description

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

Usage

```
smith.penult(densF, distF, ddensF = NULL, model = c("bm", "pot"), u, m, ...)
```

Arguments

`densF`	density function; for standard statistical family `family`, given by `dfamily`
`distF`	distribution function, for standard statistical family `family`, given by `pfamily`
`ddensF`	derivative of the density function (optional)
`model`	either block maxima (`"bm"`) or peaks-over-threshold (`"pot"`) are supported
`u`	vector of thresholds for method `"pot"`
`m`	vector of block sizes for method `"bm"`
`...`	additional arguments passed to `densF` and `distF`

Value

a matrix containing the penultimate location (if `method="bm"`), scale and shape parameters

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
qu <- seq(1,5,by=0.02)
penult <- smith.penult(densF=dnorm, distF=pnorm,
   ddensF=function(x){-x*dnorm(x)}, model="pot", u=qu)
plot(qu, penult[,2], type="l",xlab="Quantile",
   ylab="Penultimate shape",ylim=c(-0.5,0))
#Block maxima for Gamma variables -
#Use must provide arguments for shape (or rate)
m <- seq(30,3650,by=30)
penult <- smith.penult(densF=dgamma, distF=pgamma,
model="bm", m=m, shape=0.1)
#First column is location
#second is scale (both scaling constants) and
#third is penultimate shape
plot(m, penult[,3], type="l",
 xlab="Quantile", ylab="Penultimate shape")
```

`smith.penult.fn` *Smith (1987) third penultimate approximation*

Description

This function returns the density and distribution functions of the 3rd penultimate approximation for extremes of Smith (1987). It requires knowledge of the exact constants ϵ and ρ.

Usage

```
smith.penult.fn(loc, scale, shape, eps, rho = NULL, model = c("bm", "pot"),
  mdaGumbel = FALSE)
```

Arguments

`loc`	location parameter returned by `smith.penult` or threshold vector
`scale`	scale parameter returned by `smith.penult`
`shape`	shape parameter returned by `smith.penult`
`eps`	parameter vector, see **Details**.
`rho`	second-order parameter, model dependent
`model`	one of `pot` for the generalized Pareto or `bm` for the generalized extreme value distribution
`mdaGumbel`	logical indicating whether the function H_ρ should be replaced by $x^3/6$; see **Details**.

Details

Let F, f denote respectively the distribution and density functions and define the function $\phi(x)$ as

$$\phi(x) = -\frac{F(x)\log F(x)}{f(x)}$$

for block maxima. The sequence `loc` corresponds to b_n otherwise, defined as the solution of $F(b_n) = \exp(-1/n)$.

The `scale` is given by $a_n = \phi(b_n)$, the `shape` as $\gamma_n = \phi'(b_n)$. These are returned by a call to smith.penult.

For threshold exceedances, b_n is replaced by the sequence of thresholds u and we take instead $\phi(x)$ to be the reciprocal hazard function $\phi(x) = (1 - F(x))/f(x)$.

In cases where the distribution function is in the maximum domain of attraction of the Gumbel distribution, ρ is possibly undetermined and ϵ can taken to be $\phi(b_n)\phi''(b_n)$.

For distributions in the maximum domain of attraction of the Gumbel distribution, it is also possible to abstract from the ρ parameter by substituting the function H_ρ by $x^3/6$ without affecting the rate of convergence. This can be done by setting `mdaGumbel=TRUE` in the function call.

Warning

The third penultimate approximation does not yield a valid distribution function over the whole range of the original distribution, but is rather valid in a neighborhood of the true support of the distribution of maxima/threshold exceedance. The function handles the most standard failure (decreasing distribution function and negative densities), but any oscillatory behaviour will not be captured. This is inherent to the method and can be resolved by 'not' evaluating the functions F and f at the faulty points.

References

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

Examples

```
#Normal maxima example from Smith (1987)
m <- 100 #block of size 100
p <- c(smith.penult(densF=dnorm, distF=pnorm,
   ddensF=function(x){-x*dnorm(x)},model="bm",m=m))
approx <- smith.penult.fn(loc=p[1], scale=p[2], shape=p[3],
   eps=p[3]^2+p[3]+p[2]^2, mdaGumbel=TRUE, model="bm")
x <- seq(0.5,6,by=0.001)
#First penultimate approximation
plot(x, exp(m*pnorm(x, log.p=TRUE)),type="l", ylab="CDF",
main="Distribution of the maxima of\n 100 standard normal variates")
lines(x, evd::pgev(x,loc=p[1], scale=p[2], shape=0),col=2)
lines(x, evd::pgev(x,loc=p[1], scale=p[2], shape=p[3]),col=3)
lines(x, approx$F(x),col=4)
legend(x="bottomright",lty=c(1,1,1,1),col=c(1,2,3,4),
   legend=c("Exact","1st approx.","2nd approx.","3rd approx"),bty="n")
```

```
plot(x, 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, evd::dgev(x,loc=p[1], scale=p[2], shape=0),col=2)
lines(x, evd::dgev(x,loc=p[1], scale=p[2], shape=p[3]),col=3)
lines(x, approx$f(x),col=4)
legend(x="topright",lty=c(1,1,1,1),col=c(1,2,3,4),
 legend=c("Exact","1st approx.","2nd approx.","3rd approx"),bty="n")

#Threshold exceedances
p <- c(4,smith.penult(densF=dnorm, distF=pnorm,
 ddensF=function(x){-x*dnorm(x)},model="pot",u=4))
approx <- smith.penult.fn(loc=p[1], scale=p[2], shape=p[3],
 eps=p[3]^2+p[3]+p[2]^2, mdaGumbel=TRUE, model="pot")
x <- seq(4.01,7,by=0.01)
#Distribution function
plot(x, 1-(1-pnorm(x))/(1-pnorm(p[1])),type="l", ylab="Conditional CDF",
main="Exceedances of 4\n for standard normal variates")
lines(x, evd::pgpd(x, loc=p[1], scale=p[2], shape=0),col=2)
lines(x, evd::pgpd(x, loc=p[1], scale=p[2], shape=p[3]),col=3)
lines(x, approx$F(x),col=4)
#Density
plot(x, dnorm(x)/(1-pnorm(p[1])),type="l", ylab="Conditional density",
main="Exceedances of 4\n for standard normal variates")
lines(x, evd::dgpd(x, loc=p[1], scale=p[2], shape=0),col=2)
lines(x, evd::dgpd(x, loc=p[1], scale=p[2], shape=p[3]),col=3)
lines(x, approx$f(x),col=4)
```

`spline.corr` *Spline correction for Fraser-Reid approximations*

Description

The tangent exponential model can be numerically unstable for values close to $r = 0$. This function corrects these incorrect values, which are interpolated using splines. The function takes as input an object of class `fr` and returns the same object with different r^* values.

Usage

```
spline.corr(fr)
```

Arguments

`fr`	an object of class `fr`, normally the output of gpd.tem or gev.tem.

Details

If available, the function uses `cobs` from the eponym package. The latter handles constraints and smoothness penalties, and is more robust than the equivalent `smooth.spline`.

Value

an object of class `fr`, containing as additional arguments `spline` and a modified `rstar` argument.

Warning

While penalized (robust) splines often do a good job at capturing and correcting for numerical outliers and `NA`, it may also be driven by unusual curves or fail at detecting outliers (or falsely identifying 'correct' values as outliers). The user should always validate by comparing the plots of both the uncorrected (raw) output of the `gpd.tem` or `gev.tem` with that of `spline.corr`.

`W.diag` *Wadsworth's univariate and bivariate exponential threshold diagnostics*

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

Usage

```
W.diag(xdat, model = c("nhpp", "exp", "invexp"), u = NULL, k, q1 = 0,
  q2 = 1, par = NULL, M = NULL, nbs = 1000, alpha = 0.05,
  plots = c("LRT", "WN", "PS"), UseQuantiles = TRUE, pmar = c(5.5, 7, 3,
  3), tikz = FALSE, ...)
```

Arguments

`xdat`	a numeric vector of data to be fitted.
`model`	string specifying whether the univariate or bivariate diagnostic should be used. Either `nhpp` for the univariate model, `exp` (`invexp`) for the bivariate exponential model with rate (inverse rate) parametrization. See details.
`u`	optional; vector of candidate thresholds.
`k`	number of thresholds to consider (if `u` unspecified).
`q1`	lowest quantile for the threshold sequence.
`q2`	upper quantile limit for the threshold sequence (`q2` itself is not used as a threshold, but rather the uppermost threshold will be at the $(q_2 - 1/k)$ quantile).
`par`	parameters of the NHPP likelihood. If `missing`, the `fpot` routine will be run to obtain values
`M`	number of superpositions or "blocks" / "years" the process corresponds to (can affect the optimization)
`nbs`	number of simulations used to assess the null distribution of the LRT, and produce the p-value
`alpha`	significance level of the LRT

plots	vector of strings indicating which plots to produce; LRT= likelihood ratio test, WN = white noise, PS = parameter stability
UseQuantiles	logical; use quantiles as the thresholds in the plot?
pmar	vector of length 4 giving the arguments for the plot margins in par(mar=c(*,*,*,*)).
tikz	logical; if TRUE, axis labels are replaced with LaTeX code
...	additional parameters passed to plot.

Details

The function is a wrapper for the univariate (non-homogeneous Poisson process model) and bivariate exponential dependence model. For the latter, the user can select either the rate or inverse rate parameter (the inverse rate parametrization works better for uniformity of the p-value distribution under the LR test.

There are two options for the bivariate diagnostic: either provide pairwise minimum of marginally exponentially distributed margins or provide a n times 2 matrix with the original data, which is transformed to exponential margins using the empirical distribution function.

Value

plots of the requested diagnostics and a list with components

- MLE maximum likelihood estimates from all thresholds
- Cov joint asymptotic covariance matrix for ξ, η or η^{-1}.
- WN values of the white noise process.
- LRT values of the likelihood ratio test statistic vs threshold.
- pval P-value of the likelihood ratio test.
- k final number of thresholds used.
- thresh threshold selected by the likelihood ratio procedure.
- mle.u maximum likelihood estimates from selected threshold.

Author(s)

Jennifer L. Wadsworth

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)
W.diag(rexp(1000), model="nhpp", k=30, q1=0)
# Parameter Stability only
W.diag(abs(rnorm(5000)), model="nhpp", k=30, q1=0, plots=c("PS"))
library(mvtnorm)
```

```
xbvn<-rmvnorm(6000, sigma=matrix(c(1,0.7,0.7,1),2,2))
# Transform margins to exponential manually
xbvn.exp<- -log(1-pnorm(xbvn))
W.diag(apply(xbvn.exp,1,min), model="exp", k=30, q1=0) #rate parametrization
W.diag(xbvn, model="exp", k=30, q1=0)
W.diag(apply(xbvn.exp,1,min), model="invexp", k=30, q1=0) #inverse rate parametrization

## End(Not run)
## Not run:
library(ismev)
data(rain)
u <- quantile(rain, seq(0.85,0.99,by=0.01))
W.diag(xdat=rain, u=u, plots="PS")

## End(Not run)
```

Index