# Package 'mev'

January 27, 2020

**Type** Package

**Title** Multivariate Extreme Value Distributions

**Version** 1.13.1

**Date** 2020-01-26

**Description** Various tools for the analysis of univariate, multivariate and functional extremes. Exact simulation from max-stable processes [Dombry, Engelke and Oesting (2016) <doi:10.1093/biomet/asw008>, R-Pareto processes for various parametric models, including Brown-Resnick (Wadsworth and Tawn, 2014, <doi:10.1093/biomet/ast042>) and Extremal Student (Thibaud and Opitz, 2015, <doi:10.1093/biomet/asv045>). Threshold selection methods, including Wadsworth (2016) <doi:10.1080/00401706.2014.998345>, and Northrop and Coleman (2014) <doi: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>.

**License** GPL-3

**BugReports** https://github.com/lbelzile/mev/issues

**URL** https://github.com/lbelzile/mev/

**Depends** R (>= 2.10)

**Imports** alabama, boot, evd, methods, nleqslv, nloptr (>= 1.2.0), Rcpp (>= 0.12.16), stats, TruncatedNormal (>= 1.1)

**Suggests** cobs, knitr, mvPot (>= 0.1.4), mvtnorm, gmm, revdbayes, ismev

**LinkingTo** Rcpp, RcppArmadillo

**LazyData** true

**RoxygenNote** 7.0.2

**VignetteBuilder** knitr

**Encoding** UTF-8

**ByteCompile** true

**NeedsCompilation** yes

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

**Maintainer** Leo Belzile <belzilel@gmail.com>

**Repository** CRAN

**Date/Publication** 2020-01-27 06:10:02 UTC

# R topics documented:

---

`angextrapo` *Bivariate angular function for extrapolation based on rays*

---

## Description

The scale parameter $g(w)$ in the Ledford and Tawn approach is estimated empirically for $x$ large as

$$\frac{\Pr(X_P > xw, Y_P > x(1-w))}{\Pr(X_P > x, Y_P > x)}$$

where the sample $(X_P, Y_P)$ are observations on a common unit Pareto scale. The coefficient $\eta$ is estimated using maximum likelihood as the shape parameter of a generalized Pareto distribution on $\min(X_P, Y_P)$.

## Usage

```
angextrapo(dat, qu = 0.95, w = seq(0.05, 0.95, length = 20))
```

## Arguments

| | |
|---|---|
| `dat` | an $n$ by 2 matrix of multivariate observations |
| `qu` | quantile level on uniform scale at which to threshold data. Default to 0.95 |
| `w` | vector of unique angles between 0 and 1 at which to evaluate scale empirically. |

## Value

a list with elements

- `w`: angles between zero and one
- `g`: scale function at a given value of `w`
- `eta`: Ledford and Tawn tail dependence coefficient

## References

Ledford, A.W. and J. A. Tawn (1996), Statistics for near independence in multivariate extreme values. *Biometrika*, **83**(1), 169–187.

## Examples

```
angextrapo(rmev(n = 1000, model = 'log', d = 2, param = 0.5))
```

---

`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 returned alongside with the angular and radial sample for values above threshold(s) `th`, specified in terms of quantiles of the radial component `R` or marginal quantiles. Only complete tuples are kept.

## Usage

```
angmeas(
  x,
  th,
  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

| | |
|---|---|
| x | an n by d sample matrix |
| th | 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 $\sum x_i >$th, 'min' to $\min x_i >$th and 'max' to $\max x_i >$th. |
| is.angle | logical indicating whether observations are already angle with respect to region. Default to FALSE. |

## 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 wts 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
- wts 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(x=x, th=0, Rnorm='l1', Anorm='l1', marg='Frechet', wgt='Empirical')
wts2 <- angmeas(x=x, Rnorm='l2', Anorm='l2', marg='Pareto', th=0)
```

---

`angmeasdir` *Dirichlet mixture model for the spectral density*

---

## 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(
  x,
  th,
  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

| | |
|---|---|
| `x` | an `n` by `d` sample matrix |
| `th` | 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 $\sum x_i >$th, 'min' to $\min x_i >$th and 'max' to $\max x_i >$th. |
| is.angle | logical indicating whether observations are already angle with respect to region. Default to FALSE. |

## 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=2, param=0.5, model='log')
out <- angmeasdir(x=x, th=0, Rnorm='l1', Anorm='l1', marg='Frechet', wgt='Empirical')
```

---

# clikmgp *Censored likelihood for multivariate generalized Pareto distributions*

---

## Description

Censored likelihood for the logistic distribution and the Brown–Resnick and extremal Student processes.

## Usage

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

## Arguments

| | |
|---|---|
| dat | matrix of observations |
| thresh | functional threshold for the maximum |
| mthresh | vector of individuals thresholds under which observations are censored |
| 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", "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 ...

- censored matrix of booleans and NA indicating whether observations dat fall below the mthreshold mthresh
- cl cluster instance created by makeCluster (default to NULL)
- ncors number of cores for parallel computing of the likelihood
- numAbovePerRow number of observations above mthreshold (non-missing) per row
- numAbovePerCol number of observations above mthreshold (non-missing) per column
- mmax maximum per column
- B1 number of replicates for quasi Monte Carlo integral for the exponent measure

- B2 number of replicates for quasi Monte Carlo integral for the censored intensity contribution
- genvec1 generating vector for the quasi Monte Carlo routine (exponent measure), associated with B1
- genvec2 generating vector for the quasi Monte Carlo routine (individual obs contrib), associated with B2

## 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.

---

confint.eprof *Confidence intervals for profile likelihood objects*

---

## Description

Computes confidence intervals for the parameter psi 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,
  ...
)
```

## 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. |
| ... | 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

---

`distg` *Distance matrix with geometric anisotropy*

---

## Description

The function computes the distance between locations, with geometric anisotropy. The parametrization assumes there is a scale parameter, so that `scale` is the distortion for the second component only. The angle `rho` must lie in $[-\pi/2, \pi/2]$.

## Usage

```
distg(loc, scale, rho)
```

## Arguments

| | |
|---|---|
| `loc` | a `d` by 2 matrix of locations giving the coordinates of a site per row. |
| `scale` | numeric vector of length 1, greater than 1. |
| `rho` | angle for the anisotropy, must be larger than $\pi/2$ in modulus. |

## Value

a `d` by `d` square matrix of pairwise distance

---

`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, $\kappa$. All three families share the same tail index as the generalized Pareto distribution, 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 ($\kappa$, $\sigma$,$\xi$). |
| 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 | 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 = 2, shape = 0.5)
par <- fit.egp(xdat, thresh = 0, model = 'egp3')$par
p <- c(1/1000, 1/1500, 1/2000)
#With multiple thresholds
th <- c(0, 0.1, 0.2, 1)
opt <- tstab.egp(xdat, th, model = 'egp1')
egp.retlev(xdat, opt$thresh, opt$par, 'egp1', p = p)
opt <- tstab.egp(xdat, th, model = 'egp2', plots = NA)
egp.retlev(xdat, opt$thresh, opt$par, 'egp2', p = p)
opt <- tstab.egp(xdat, th, model = 'egp3', plots = NA)
egp.retlev(xdat, opt$thresh, opt$par, 'egp3', p = p)
```

`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 aggressive) |
| `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.

---

`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 1554m 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", "hr", "br", "xstud"),
  method = c("TruncatedNormal", "mvtnorm", "mvPot")
)
```

## Arguments

| | |
|---|---|
| z | vector at which to estimate exponent measure |
| par | list of parameters |
| model | string indicating the model family |
| method | 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)
```

---

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

---

## Description

The function implements 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

```
ext.index(
  x,
  q = 0.95,
  method = c("wls", "mle", "intervals"),
  plot = FALSE,
  warn = 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 | logical; if TRUE, plot the extremal index as a function of q |
| warn | logical; if TRUE, receive a warning when the sample size is too small |

## 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 $\theta$ 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 $\theta$ 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'))
```

---

`extcoef` *Estimators of the extremal coefficient*

---

## Description

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

## Usage

```
extcoef(
  dat,
  coord = NULL,
  thresh = NULL,
  estimator = c("schlather", "smith", "fmado"),
  standardize = TRUE,
  method = c("nonparametric", "parametric"),
  prob = 0,
  plot = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| `dat` | 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 model estimates to compute, one of `smith`, `schlather` or `fmado`. |
| `standardize` | logical; should observations be transformed to unit Frechet scale? Default is to transform |
| `method` | string indicating which method to use to transform the margins. See **Details** |
| `prob` | probability of not exceeding threshold `thresh` |
| `plot` | logical; should cloud 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

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

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

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

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 significant for large data sets. If there are no missing observations, the routine `fmadogram` from the `SpatialExtremes` package should be prefered 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 `standardize = FALSE`. If `method = "parametric"`, 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 = "nonparametric"`, 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(dat = 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(dat = 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 |
| 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

---

extremo *Pairwise extremogram for max-risk functional*

---

### Description

The function computes the pairwise $chi$ estimates and plots them as a function of the distance between sites.

### Usage

```
extremo(dat, margp, coord, scale = 1, rho = 0, plot = FALSE, ...)
```

### Arguments

| | |
|---|---|
| dat | data matrix |
| margp | marginal probability above which to threshold observations |
| coord | matrix of coordinates (one site per row) |
| scale | geometric anisotropy scale parameter |
| rho | geometric anisotropy angle parameter |
| plot | logical; should a graph of the pairwise estimates against distance? Default to FALSE |
| ... | additional arguments passed to plot |

### Value

an invisible matrix with pairwise estimates of chi along with distance (unsorted)

### Examples

```
## Not run:
lon <- seq(650, 720, length = 10)
lat <- seq(215, 290, length = 10)
# Create a grid
grid <- expand.grid(lon,lat)
coord <- as.matrix(grid)
dianiso <- distg(coord, 1.5, 0.5)
sgrid <- scale(grid, scale = FALSE)
# Specify marginal parameters `loc` and `scale` over grid
eta <- 26 + 0.05*sgrid[,1] - 0.16*sgrid[,2]
```

```
tau <- 9 + 0.05*sgrid[,1] - 0.04*sgrid[,2]
# Parameter matrix of Huesler--Reiss
# associated to power variogram
Lambda <- ((dianiso/30)^0.7)/4
# Regular Euclidean distance between sites
di <- distg(coord, 1, 0)
# Simulate generalized max-Pareto field
set.seed(345)
simu1 <- rgparp(n = 1000, thresh = 50, shape = 0.1, riskf = "max",
                scale = tau, loc = eta, sigma = Lambda, model = "hr")
extdat <- extremo(dat = simu1, margp = 0.98, coord = coord,
                  scale = 1.5, rho = 0.5, plot = TRUE)

# Constrained optimization
# Minimize distance between extremal coefficient from fitted variogram
mindistpvario <- function(par, emp, coord){
alpha <- par[1]; if(!isTRUE(all(alpha > 0, alpha < 2))){return(1e10)}
scale <- par[2]; if(scale <= 0){return(1e10)}
a <- par[3]; if(a<1){return(1e10)}
rho <- par[4]; if(abs(rho) >= pi/2){return(1e10)}
semivariomat <- mgp::power.vario(distg(coord, a, rho), alpha = alpha, scale = scale)
  sum((2*(1-pnorm(sqrt(semivariomat[lower.tri(semivariomat)]/2))) - emp)^2)
}

hin <- function(par, ...){
  c(1.99-par[1], -1e-5 + par[1],
    -1e-5 + par[2],
    par[3]-1,
    pi/2 - par[4],
    par[4]+pi/2)
  }
opt <- alabama::auglag(par = c(0.7, 30, 1, 0),
                       hin = hin,
                        fn = function(par){
                          mindistpvario(par, emp = extdat[,'prob'], coord = coord)})
stopifnot(opt$kkt1, opt$kkt2)
# Plotting the extremogram in the deformed space
distfa <- distg(loc = coord, opt$par[3], opt$par[4])
plot(c(distfa[lower.tri(distfa)]), extdat[,2], pch = 20,
col = scales::alpha(1,0.1), yaxs = "i", xaxs = "i", bty = 'l',
     xlab = "distance", ylab= "cond. prob. of exceedance", ylim = c(0,1))
lines(x = (distvec <- seq(0,200, length = 1000)), col = 2, lwd = 2,
      2*(1-pnorm(sqrt(power.vario(distvec, alpha = opt$par[1], scale = opt$par[2])/2))))

## End(Not run)
```

---

`fit.egp` *Parameter stability plot and maximum likelihood routine for extended GP models*

---

## Description

The function `tstab.egp` provides classical threshold stability plot for $(\kappa, \sigma, \xi)$. 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. `tstab.egp` can also be used to fit the model to multiple thresholds.

## Usage

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

tstab.egp(
  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. |
| `show` | logical; if `TRUE`, print the results of the optimization |
| `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

`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 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.

## Examples

```
xdat <- evd::rgpd(n = 100, loc = 0, scale = 1, shape = 0.5)
fitted <- fit.egp(xdat = xdat, thresh = 1, model = "egp2", show = TRUE)
thresh <- evd::qgpd(seq(0.1, 0.5, by = 0.05), 0, 1, 0.5)
tstab.egp(xdat = xdat, thresh = thresh, model = "egp2", plots = 1:3)
```

---

`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 $p$, $\kappa$, $\delta$,$\sigma$,$\xi$ (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, gp.fit(rain, 0)$est),
 rounded = 0.1, confint = TRUE, R = 20)

## End(Not run)
```

`fit.gev` *Maximum likelihood estimation for the generalized extreme value distribution*

## Description

This function returns an object of class `mev_gev`, with default methods for printing and quantile-quantile plots.

## Usage

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

## Arguments

| | |
|---|---|
| `xdat` | a numeric vector of data to be fitted. |
| `start` | numeric vector of starting values |
| `method` | string indicating the outer optimization routine for the augmented Lagrangian. One of `nlminb` or `BFGS`. |
| `show` | logical; if `TRUE` (the default), print details of the fit. |

## 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

`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
)
```

## 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. |
| `k` | bound on the influence function (`method = "obre"`); the constant `k` is a robustness parameter (higher bounds are more efficient, low bounds are more robust). Default to 4, must be larger than $\sqrt{2}$. |
| `tol` | numerical tolerance for OBRE weights iterations (`method = "obre"`). Default to `1e-8`. |

## 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 `'ismev'` performs 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 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 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 `nlm`, `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, show = 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. |
| show | logical; if TRUE (the default), print details of the fit. |

## 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.rlarg` *Maximum likelihood estimates of point process for the r-largest observations*

## 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("nlminb", "BFGS"), show = FALSE)
```

## Arguments

| | |
|---|---|
| `xdat` | a numeric vector of data to be fitted. |
| `start` | numeric vector 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. |

## 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)
```

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)
```

## 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.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

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

for $\tilde{\boldsymbol{\theta}}$, 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

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

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

gev.mle *Generalized extreme value maximum likelihood estimates for various quantities of interest*

## 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 <- evd::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 $psi$ 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 $psi$
- `r`:values of likelihood root corresponding to $\psi$
- `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.profll`: 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.profll`: 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 <- evd::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, by = 0.1), param = 'quant', dat = dat, p = 0.01)
gev.pll(psi = seq(12, 100, by=1), param = 'Nmean', N = 100, dat = dat)
gev.pll(psi = seq(12, 90, by=1), 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

| | |
|---|---|
| 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 | tail probability for the (1-p)th quantile (return levels). Required only if param = 'retlev' |
| q | probability level of quantile. Required only for param Nquant. |
| 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`) 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 $\psi$
- `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 <- evd::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 <- evd::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)
```

---

# gevN *Generalized extreme value distribution (quantile/mean of N-block maxima 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 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(d
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, 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 |
| 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 gevr.Vfun |

### 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(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.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.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

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

for $\tilde{\boldsymbol{\theta}}$, 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

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

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 <- evd::rgpd(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.mle *Generalized Pareto maximum likelihood estimates for various quantities of interest*

---

## 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 <- evd::rgpd(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", "VaR", "ES", "Nmean", "Nquant"),
  mod = "profile",
  mle = NULL,
  dat,
  m = NULL,
  N = NULL,
  p = NULL,
  q = NULL,
  correction = TRUE,
  threshold = 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 $(\psi, \xi)$ parametrization if $\psi \neq \xi$ and $(\sigma, \xi)$ otherwise (optional). |
| dat | sample vector of excesses, unless threshold 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 spline.corr to smooth the tem approximation. |
| threshold | 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 Vfun if mode='tem'. |

## 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 $psi$ given in `psi`
- `pll`: values of the profile log likelihood at `psi`
- `maxpll`: value of maximum profile log likelihood
- `family`: a string indicating "gpd"
- `threshold`: value of the threshold, by default zero

In addition, if `mod` includes `tem`

- `normal`:maximum likelihood estimate and standard error of the interest parameter $psi$
- `r`:values of likelihood root corresponding to $\psi$
- 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.profll`: 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.profll`: values of the modified profile log likelihood at `psi`
- `empcov.maxpll`: value of maximum modified profile log likelihood

## Examples

```
## Not run:
dat <- evd::rgpd(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", "ES", "Nmean", "Nquant"),
  psi = NULL,
  m = NULL,
  threshold = 0,
  n.psi = 50,
  N = NULL,
  p = NULL,
  q = NULL,
  plot = FALSE,
  correction = TRUE
)
```

## Arguments

| | |
|---|---|
| dat | sample vector for the GP distribution |
| 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. If psi is of length 2 and n.psi>2, it is assumed to be the minimal and maximal values at which to evaluate the profile log likelihood. |
| m | number of observations of interest for return levels. See **Details**. Required only for param = 'VaR' or param = 'ES'. |
| threshold | threshold value corresponding to the lower bound of the support or the location parameter of the generalized Pareto 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 |
| 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. |
| plot | logical indicating whether plot.fr should be called upon exit |
| correction | logical indicating whether spline.corr should be called. |

## 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) 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 $\psi$
- 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 <- evd::rgpd(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, bias, 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 $\zeta_u$ is the rate of exceedance over the threshold `u` and $\alpha$ 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 $\xi$, 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.infomat(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.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

---

## gpdN — *Generalized Pareto distribution (mean of maximum of N exceedances 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 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 $\xi$, 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

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

### Functions

- `gpdN.ll`: log likelihood
- `gpdN.score`: score vector
- `gpdN.infomat`: observed information matrix for GP parametrized in terms of mean of the maximum of `N` exceedances and shape
- `gpdN.Vfun`: vector implementing conditioning on approximate ancillary statistics for the TEM
- `gpdN.phi`: canonical parameter in the local exponential family approximation
- `gpdN.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 $\xi$, 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(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), 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 $\chi^2$ 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` | logical: 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 $\chi^2_1$ 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

```
infomat.test(x <- evd::rgpd(n = 10000), q = seq(0.1, 0.9, length = 10), K <- 3)
```

---

## lambdadep — *Estimation of the bivariate lambda function of Wadsworth and Tawn (2013)*

---

### Description

Estimation of the bivariate lambda function of Wadsworth and Tawn (2013)

### Usage

```
lambdadep(dat, qu = 0.95, method = c("hill", "mle", "bayes"), plot = TRUE)
```

### Arguments

| | |
|---|---|
| `dat` | an $n$ by 2 matrix of multivariate observations |
| `qu` | quantile level on uniform scale at which to threshold data. Default to 0.95 |
| `method` | string indicating the estimation method |
| `plot` | logical indicating whether to return the graph of `lambda` |
| | 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. |

## Value

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

- `w` the sequence of angles in (0,1) at which the `lambda` values are evaluated
- `lambda` point estimates of lambda
- `lower.confint` 95
- `upper.confint` 95

## Examples

```
set.seed(12)
dat <- evd::rbvevd(n=1000, dep = 0.1)
lambdadep(dat, method = 'hill')
## Not run:
lambdadep(dat, method = 'bayes')
lambdadep(dat, method = 'mle')
dat <- matrix(runif(n = 2000), ncol = 2)
lambdadep(dat, method = 'hill')

## End(Not run)
```

---

`likmgp` *Likelihood for multivariate generalized Pareto distribution*

---

## Description

Likelihood for the Brown–Resnick, extremal Student or logistic vectors over region determined by

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

where $\mu$ is `loc`, $\sigma$ is `scale` and $\xi$ is `shape`.

## Usage

```
likmgp(
  dat,
  thresh,
  loc,
  scale,
  shape,
  par,
  model = c("br", "xstud", "log"),
  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", "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)
```

---

# maxstabtest *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 `nmax` 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

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

## Arguments

| | |
|---|---|
| dat | 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). |

## Value

a Tukey probability-probability plot with 95

## 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:
dat <- mev::rmev(n = 250, d = 100, param = 0.5, model = "log")
maxstabtest(dat, m = 100)
maxstabtest(dat, m = 2, nmax = 100)
dat <- mev::mvrnorm(n = 250, Sigma = diag(0.5, 10) + matrix(0.5, 10, 10), mu = rep(0, 10))
maxstabtest(dat, m = 2, nmax = 100)
maxstabtest(dat, m = ncol(dat))

## End(Not run)
```

## 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:
data(nidd)
u <- quantile(nidd, seq(0.85, 0.99, by = 0.01))
NC.diag(nidd, u, size = 0.05)

## End(Not run)
```

## nidd *River Nidd Flow*

### 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, *Journal of the Royal Statistical Society. Series B (Methodological)*, **52**(3), 393–442. With discussion.

### See Also

`nidd.thresh`

## 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` *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 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.

pp *Poisson process of extremes.*

## 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 *post hoc* adjustment for the intensity so that the parameters of the model coincide with those of a generalized extreme value distribution with block size length(dat)/np. |
| nobs | number of observations for the expected information matrix. Default to length(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.

## 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)
```

## rgparp *Simulation from generalized R-Pareto processes*

### 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,
  riskf = 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

| | |
|---|---|
| n | number of observations |
| shape | shape parameter of the generalized Pareto variable |
| thresh | univariate threshold for the exceedances of risk functional |
| riskf | string indicating the risk functional. |
| siteindex | integer between 1 and d specifying the index of the site or variable |
| d | dimension of sample |
| loc | location vector |
| scale | scale vector |
| 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 $\lambda^2$ 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 | 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 |

## 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, riskf = 'site', siteindex = 2, d = 3, param = 2.5,
   model = 'log', scale = c(1, 2, 3), loc = c(2, 3, 4))
rgparp(n = 10, riskf = 'max', d = 4, param = c(0.2, 0.1, 0.9, 0.5),
   scale = 1:4, loc = 1:4, model = 'bilog')
rgparp(n = 10, riskf = '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, riskf = 'max', vario = vario, coord = grid.coord,
   model = 'br', scale = runif(16), loc = rnorm(16))
```

---

# 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.

---

`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", "sdir", "dirmix"),
  alg = c("ef", "sm"),
  weights,
  vario,
  coord = NULL,
  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 $\lambda^2$ 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 | 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). |
| ... | 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 $(-\min(\alpha_1, \ldots, \alpha_d), 1]$ 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

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 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)
vario2cov <- function(coord, semivario,...){
 sapply(1:nrow(coord), function(i) sapply(1:nrow(coord), function(j)
  semivario(sqrt(sum((coord[i,])^2)), ...) +
  semivario(sqrt(sum((coord[j,])^2)), ...) -
  semivario(sqrt(sum((coord[i,]-coord[j,])^2)), ...)))
}
# asymmetric logistic model - see evd::rmvevd
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")
rmev(n=100, sigma=vario2cov(grid.coord, semivario = semivario, scale = 0.5, alpha = 1), model='br')
#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')
```

`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", "smith",
    "schlather", "ct", "sdir", "dirmix"),
  weights,
  vario,
  coord = NULL,
  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**. |
| `sigma` | covariance matrix for Brown-Resnick and extremal Student-t distributions. Symmetric matrix of squared coefficients $\lambda^2$ 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` | 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). |
| ... | 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, 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 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 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')
```

---

rparp *Simulation from R-Pareto processes*

---

## Description

Simulation from R-Pareto processes

## Usage

```
rparp(
  n,
  shape = 1,
  riskf = 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 |
| riskf | 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 $\lambda^2$ 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 | 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, riskf='site', siteindex=2, d=3, param=2.5, model='log')
rparp(n=10, riskf='min', d=3, param=2.5, model='neglog')
rparp(n=10, riskf='max', d=4, param=c(0.2,0.1,0.9,0.5), model='bilog')
rparp(n=10, riskf='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, riskf='max', vario=vario, coord=grid.coord, model='br')
```

---

# rparpcs *Simulation from Pareto processes (max) using composition sampling*

---

## Description

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

## Usage

```
rparpcs(
  n,
  Lambda = NULL,
  Sigma = NULL,
  df = NULL,
  model = c("br", "xstud"),
  riskf = c("max", "min"),
  shape = 1
)
```

## Arguments

| | |
|---|---|
| n | sample size. |
| 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. |
| model | string indicating the model family. |
| riskf | string indicating the risk functional. Only max and min are currently supported. |
| shape | tail index of the Pareto variates (reciprocal shape parameter). Must be strictly positive. |

## Details

Only extreme value models based on elliptical processes are handled. The Lambda matrix is formed by evaluating the semivariogram $\gamma$ 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 `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 (2017), Simple models for multivariate regular variations and the Huesler-Reiss Pareto distribution, http://arxiv.org/abs/1712.09225v1

## 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, scale, shape)
```

## Arguments

| | |
|---|---|
| n | sample size |
| r | number of observations per block |
| loc | location parameter |
| scale | scale parameter |
| shape | shape parameter |

## Value

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

`smith.penult` *Smith's 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(family, method = c("bm", "pot"), u, qu, m, returnList = TRUE, ...)
```

## 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 |
| `u` | vector of thresholds for method `'pot'` |
| `qu` | vector of quantiles for method `'pot'`. Ignored if argument `u` is provided. |
| `m` | vector of block sizes for method `'bm'` |
| `returnList` | logical; should the arguments be returned as a list or as a matrix of parameter |
| `...` | 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.

## Value

either a vector, a matrix if either `length(m)>1` or `length(u)>1` or a list (if `returnList`) containing

- `loc`: location parameters (`method='bm'`)
- `scale`: scale parameters
- `shape`: shape parameters
- `u`: thresholds (if `method='pot'`)
- `u`: 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
qu <- seq(1,5,by=0.02)
penult <- smith.penult(family = "norm", ddensF=function(x){-x*dnorm(x)},
   method = 'pot', u = qu)
plot(qu, penult$shape, type='l', xlab='Quantile',
   ylab='Penultimate shape', ylim=c(-0.5,0))
#Block maxima for Gamma variables -
#User must provide arguments for shape (or rate)
m <- seq(30, 3650, by=30)
penult <- smith.penult(family = 'gamma', method = 'bm', m=m, shape=0.1)
plot(m, penult$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 <- smith.penult(family='norm',
   ddensF=function(x){-x*dnorm(x)}, method='bm', m=m, returnList=FALSE)
x <- seq(1, 5, by = 0.01)
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)
legend(x = 'topright',lty = c(1,1,1,1), col = c(1,2,3,4),
   legend = c('exact', 'ultimate', 'penultimate'), bty = 'n')
```

---

`smith.penult.fn` *Smith's 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 $\epsilon$ and $\rho$ described in the paper.

## Usage

```
smith.penult.fn(
  loc,
  scale,
  shape,
  eps,
  rho = NULL,
  method = 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 |
| `method` | one of `pot` for the generalized Pareto or `bm` for the generalized extreme value distribution |
| `mdaGumbel` | logical indicating whether the function $H_\rho$ should be replaced by $x^3/6$; see **Details**. |
| `...` | additional parameters, currently ignored. These are used for backward compatibility due to a change in the names of the arguments. |

## 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, $\rho$ is possibly undetermined and $\epsilon$ can be equal to $\phi(b_n)\phi''(b_n)$.

For distributions in the maximum domain of attraction of the Gumbel distribution and that are class N, it is also possible to abstract from the $\rho$ parameter by substituting the function $H_\rho$ 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 necessarily 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 <- smith.penult(family='norm',
   ddensF=function(x){-x*dnorm(x)}, method='bm', m=m, returnList=FALSE)
approx <- smith.penult.fn(loc=p[1], scale=p[2], shape=p[3],
   eps=p[3]^2+p[3]+p[2]^2, mdaGumbel=TRUE, method='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
par <- smith.penult(family = "norm", ddensF=function(x){-x*dnorm(x)},
method='pot', u=4, returnList=FALSE)
approx <- smith.penult.fn(loc=par[1], scale=par[2], shape=par[3],
 eps=par[3]^2+par[3]+par[2]^2, mdaGumbel=TRUE, method='pot')
x <- seq(4.01,7,by=0.01)
#Distribution function
plot(x, 1-(1-pnorm(x))/(1-pnorm(par[1])),type='l', ylab='Conditional CDF',
main='Exceedances over 4\n for standard normal variates')
lines(x, evd::pgpd(x, loc=par[1], scale=par[2], shape=0),col=2)
lines(x, evd::pgpd(x, loc=par[1], scale=par[2], shape=par[3]),col=3)
lines(x, approx$F(x),col=4)
#Density
plot(x, dnorm(x)/(1-pnorm(par[1])),type='l', ylab='Conditional density',
main='Exceedances over 4\n for standard normal variates')
lines(x, evd::dgpd(x, loc=par[1], scale=par[2], shape=0),col=2)
lines(x, evd::dgpd(x, loc=par[1], scale=par[2], shape=par[3]),col=3)
lines(x, approx$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 `rstar` 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 values lying on the profile log-likelihood the curve or fail to detect 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 object with that of `spline.corr`.

---

# spunif *Semi-parametric marginal transformation to uniform*

---

## 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

| | |
|---|---|
| x | matrix or vector of data |
| thresh | vector of marginal thresholds |
| scale | vector of marginal scale parameters for the generalized Pareto |
| shape | 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 $\eta$ is defined via

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

where $L(x)$ is a slowly varying function; $0 < \eta$. Ignoring the latter, several estimators of $\eta$ can be defined. In unit Pareto margins, $\eta$ is a shape parameter that can be estimated by fitting a generalized Pareto distribution above a high threshold. In exponential margins, $\eta$ 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 confint obtained through a likelihood ratio test statistic ("lrt") or the Wald statistic ("wald").

### Usage

```
taildep(
  data,
  u = NULL,
  nq = 40,
  qlim = c(0.8, 0.99),
  depmeas = c("eta", "chi"),
 method = list(eta = c("emp", "betacop", "gpd", "hill"), chi = c("emp", "betacop")),
  confint = c("wald", "lrt"),
  level = 0.95,
```

```
  trunc = TRUE,
  ties.method = "random",
  plot = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| data | an $n$ by $d$ matrix of multivariate observations |
| u | vector of percentiles between 0 and 1 at which to evaluate the plot |
| nq | number of quantiles at which to form a grid; only used if u = NULL. |
| qlim | limits for the sequence u |
| depmeas | dependence measure, either of "eta" or "chi" |
| method | named list giving the estimation method for eta and chi. Default to "emp" for both. |
| confint | string indicating the type of confidence interval for $\eta$, 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 $[0, 1]$ |
| 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, citype, 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 confint for the latter are obtained by assuming the proportion of points is binomial.

The coefficient of tail correlation $\chi$ is

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

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

## Value

a named list with elements

- u: 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

## See Also

chiplot for bivariate empirical estimates of $\chi$ 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)
```

---

# 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 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(
  dat,
  thresh,
  method = c("wald", "profile", "post"),
  level = 0.95,
  plot = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| dat | 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. |
| ... | 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.

## Author(s)

Leo Belzile

## See Also

`gpd.fitrange`

## Examples

```
dat <- abs(rnorm(10000))
u <- qnorm(seq(0.9,0.99, by= 0.01))
tstab.gpd(dat = dat, thresh = u)
## Not run:
tstab.gpd(dat = dat, thresh = u, method = "profile")
tstab.gpd(dat = dat, thresh = u, method = "post")

## End(Not run)
```

---

# venice *Venice Sea Levels*

---

## Description

The `venice` data contains the 10 largest yearly sea levels (in cm) from 1887 until 2017. Only the yearly maximum is available for 1922 and the six largest observations for 1936.

## Format

a data frame with 131 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 2017 were extracted by Anthony Davison from the City of Venice website (accessed October 2018) 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 <http://archive.comune.venezia.it/flex/cm/pages/ServeBLOB.php/L/EN/IDPagina/3045>. Last accessed October 2018.

## 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

---

`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, 1, 1),
  ...
)
```

## 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 fit.pp 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(*,*,*,*)). |
| ... | 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 an invisible list with components

- `MLE` maximum likelihood estimates from all thresholds
- `Cov` joint asymptotic covariance matrix for $\xi$, $\eta$ or $\eta^{-1}$.
- `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 = 20, q1 = 0)
# Parameter stability only
W.diag(abs(rnorm(5000)), model = 'nhpp', k = 30, q1 = 0, plots = "PS")
xbvn <- mvrnorm(6000, mu = rep(0, 2), Sigma = cbind(c(1, 0.7), c(0.7, 1)))
# 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)
```

---

# 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

# Index