Package ‘ExtremeRisks’

May 12, 2026

Version 0.0.6

Date 2026-05-08

Title Extreme Risk Measures

Author Simone Padoan [cre, aut],
Gilles Stupfler [aut],
Carlotta Pacifici [aut]

Maintainer Simone Padoan <simone.padoan@unibocconi.it>

Imports evd, copula, mvtnorm, plot3D, tmvtnorm, pracma

Depends R (>= 3.5.0)

Description A set of procedures for estimating risks related to extreme events via risk measures such as Expectile, Value-at-Risk, etc. is provided. Estimation methods for univariate independent observations and temporal dependent observations are available. The methodology is extended to the case of independent multidimensional observations. The statistical inference is performed through parametric and non-parametric estimators. Inferential procedures such as confidence intervals, confidence regions and hypothesis testing are obtained by exploiting the asymptotic theory. Adapts the methodologies derived in Padoan and Stupfler (2022) <doi:10.3150/21-BEJ1375>, Davison et al. (2023) <doi:10.1080/07350015.2022.2078332>, Daouia et al. (2018) <doi:10.1111/rssb.12254>, Drees (2000) <do
reira (2006) <doi:10.1007/0-387-34471-3>, de Haan et al. (2016) <doi:10.1007/s00780-015-0287-6>, Padoan and Rizzelli (2024) <doi:10.3150/23-BEJ1668>, Daouia et al. (2024) <doi:10.3150/23-BEJ1632>.

License GPL (>= 2)

URL https://www.unibocconi.it/en/faculty/simone-padoan

NeedsCompilation yes

Repository CRAN

Repository/R-Forge/Project extremerisks

Encoding UTF-8

Date/Publication 2026-05-12 05:10:27 UTC

Contents

`cpost_stat` *Estimation of the scedasis function*

Description

Kernel-based method for the estimation of the scedasis function. Given the values of the complete and concomitant covariate, defined as $X \mid Y > t$, with t being the threshold, it returns a matrix containing a posterior sample of the scedasis function for each covariate value.

Usage

```
cpost_stat(N, x, xs, xg, bw, k, C = 5L)
```

Arguments

`N`	integer, number of samples to draw from the distribution of the concomitant covariate
`x`	one-dimensional vector of in-sample covariate in [0,1]
`xs`	one-dimensional vector of concomitant covariate
`xg`	one-dimensional vector of length m containing the grid of in-sample and possibly out-sample covariate in [0,1]
`bw`	double, bandwidth for the computation of the kernel
`k`	integer, number of exceedances for the generalized Pareto
`C`	integer, hyperparameter entering the posterior distribution of the law of the concomitant covariate. Default: 5

Value

an `N` by `m` matrix containing the values of the posterior samples of the scedasis function (rows) for each value of `xg` (columns)

Examples

```
## Not run:
# generate data
set.seed(1234)
n <- 500
samp <- evd::rfrechet(n, 0, 1:n, 4)
# set effective sample size and threshold
k <- 50
threshold <- sort(samp, decreasing = TRUE)[k+1]
# preliminary mle estimates of scale and shape parameters
mlest <- evd::fpot(
  samp,
  threshold,
  control = list(maxit = 500))
# empirical bayes procedure
proc <- estPOT(
  samp,
  k = k,
  pn = c(0.01, 0.005),
  type = "continuous",
  method = "bayesian",
  prior = "empirical",
  start = as.list(mlest$estimate),
  sig0 = 0.1)
# conditional predictive density estimation
yg <- seq(0, 50, by = 2)
nyg <- length(yg)
```

```
# estimation of scedasis function
# setting
M <- 1e3
C <- 5
alpha <- 0.05
bw <- .5
nsim <- 5000
burn <- 1000
# create covariate
# in sample obs
n_in = n
# number of years ahead
nY = 1
n_out = 365 * nY
# total obs
n_tot = n_in + n_out
# total covariate (in+out sample period)
x <- seq(0, 1, length = n_tot)
# in sample grid dimension for covariate
ng_in <- 150
xg <- seq(0, x[n_in], length = ng_in)
# in+out of sample grid
xg <- c(xg, seq(x[n_in + 1], x[(n_tot)], length = ng_in))
# in+out sample grid dimension
nxg <- length(xg)
xg <- array(xg, c(nxg, 1))
# in sample observations
samp_in <- samp[1:n_in]
ssamp_in <- sort(samp_in, decreasing = TRUE, index = TRUE)
x_in <- x[1:n_in] # in sample covariate
xs <- x_in[ssamp_in$ix[1:k]]
# in sample concomitant covariate
# estimate scedasis function over the in and out of sample period
res_stat <- apply(
  xg,
  1,
  cpost_stat,
  N = nsim - burn,
  x = x_in,
  xs = xs,
  bw = bw,
  k = k,
  C = C
)

## End(Not run)
```

`dowjones` *Negative log-returns of DOW JONES.*

Description

Series of negative log-returns of the U.S. stock market index Dow Jones.

Format

A $8784 * 2$ data frame.

Details

From the series of n = 8785 closing prices S_t, $t = 1, 2, ...$, for the Dow Jones stock market index, recorded from January 29, 1985 to December 12, 2019, the series of negative log-returns.

$X_{t+1} = -\log(S_{t+1}/S_t), \quad 1 \leq t \leq n-1$

is available. Hence the dataset (negative log-returns) contains 8784 observations.

`EBTailIndex` *Expectile Based Tail Index Estimation*

Description

Computes a point estimate of the tail index based on the Expectile Based (EB) estimator.

Usage

```
EBTailIndex(data, tau, est=NULL)
```

Arguments

`data`	A vector of $(1 \times n)$ observations.
`tau`	A real in $(0, 1)$ specifying the intermediate level τ_n. See **Details**\.
`est`	A real specifying the estimate of the expectile at the intermediate level `tau`.

Details

For a dataset `data` of sample size n, the tail index γ of its (marginal) distribution is estimated using the EB estimator:

$\hat{\gamma}_n^E = \left(1 + \frac{\hat{\bar{F}}_n(\tilde{\xi}_{\tau_n})}{1-\tau_n}\right)^{-1}$,

where $\hat{\bar{F}}_n$ is the empirical survival function of the observations, $\tilde{\xi}_{\tau_n}$ is an estimate of the τ_n*-th* expectile. The observations can be either independent or temporal dependent. See Padoan and Stupfler (2020) and Daouia et al. (2018) for details.

- The so-called intermediate level `tau` or τ_n is a sequence of positive reals such that $\tau_n \to 1$ as $n \to \infty$. Practically, $\tau_n \in (0, 1)$ is the ratio between the empirical mean distance of the τ_n*-th* expectile from the smaller observations and the empirical mean distance of of the τ_n*-th* expectile from all the observations. An estimate of τ_n*-th* expectile is computed and used in turn to estimate γ.

- The value est, if provided, is meant to be an esitmate of the τ_n-*th* expectile which is used to estimate γ. On the contrary, if est=NULL, then the routine EBTailIndex estimate first the τ_n-*th* expectile expectile and then use it to estimate γ.

Value

An estimate of the tain index γ.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://www.unibocconi.it/en/faculty/simone-padoan; Gilles Stupfler, <gilles.stupfler@univ-angers.fr>, https://math.univ-angers.fr/~stupfler/

References

Anthony C. Davison, Simone A. Padoan and Gilles Stupfler (2023). Tail Risk Inference via Expectiles in Heavy-Tailed Time Series, *Journal of Business & Economic Statistics*, **41**(3) 876-889.

Daouia, A., Girard, S. and Stupfler, G. (2018). Estimation of tail risk based on extreme expectiles. *Journal of the Royal Statistical Society: Series B*, **80**, 263-292.

See Also

HTailIndex, MomTailIndex, MLTailIndex,

Examples

```
# Tail index estimation based on the Expectile based estimator obtained with data
# simulated from an AR(1) with 1-dimensional Student-t distributed innovations

tsDist <- "studentT"
tsType <- "AR"

# parameter setting
corr <- 0.8
df <- 3
par <- c(corr, df)

# Big- small-blocks setting
bigBlock <- 65
smallblock <- 15

# Intermediate level (or sample tail probability 1-tau)
tau <- 0.97

# sample size
ndata <- 2500

# Simulates a sample from an AR(1) model with Student-t innovations
data <- rtimeseries(ndata, tsDist, tsType, par)
```

```
# tail index estimation
gammaHat <- EBTailIndex(data, tau)
gammaHat
```

estExpectiles *High Expectile Estimation*

Description

Computes a point and interval estimate of the expectile at the intermediate level.

Usage

```
estExpectiles(data, tau, method="LAWS", tailest="Hill", var=FALSE, varType="asym-Dep-Adj",
              bigBlock=NULL, smallBlock=NULL, k=NULL, alpha=0.05)
```

Arguments

`data`	A vector of $(1 \times n)$ observations.
`tau`	A real in $(0, 1)$ specifying the intermediate level τ_n. See **Details**.
`method`	A string specifying the method used to estimate the expecile. By default `est="LAWS"` specifies the use of the direct LAWS estimator. See **Details**.
`tailest`	A string specifying the type of tail index estimator. By default `tailest="Hill"` specifies the use of Hill estimator. See **Details**.
`var`	If `var=TRUE` then an estimate of the variance of the expectile estimator is computed.
`varType`	A string specifying the asymptotic variance to compute. By default `varType="asym-Dep-Adj"` specifies the variance estimator for serial dependent observations implemented with a suitable adjustment. See **Details**.
`bigBlock`	An interger specifying the size of the big-block used to estimaste the asymptotic variance. See **Details**.
`smallBlock`	An interger specifying the size of the small-block used to estimaste the asymptotic variance. See **Details**.
`k`	An integer specifying the value of the intermediate sequence k_n. See **Details**.
`alpha`	A real in $(0, 1)$ specifying the confidence level $(1-\alpha)100\%$ of the approximate confidence interval for the expecile at the intermedite level.

Details

For a dataset `data` of sample size n, an estimate of the τ_n-*th* expectile is computed. Two estimators are available: the so-called direct Least Asymmetrically Weighted Squares (LAWS) and indirect Quantile-Based (QB). The definition of the QB estimator depends on the estimation of the tail index γ. Here, γ is estimated using the Hill estimation (see HTailIndex) or in alternative using the the expectile based estimator (see EBTailIndex). The observations can be either independent or temporal dependent. See Section 3.1 in Padoan and Stupfler (2020) for details.

- The so-called intermediate level `tau` or τ_n is a sequence of positive reals such that $\tau_n \to 1$ as $n \to \infty$. Practically, $\tau_n \in (0,1)$ is the ratio between *N* (Numerator) and *D* (Denominator). Where *N* is the empirical mean distance of the τ_n-*th* expectile from the observations smaller than it, and *D* is the empirical mean distance of τ_n-*th* expectile from all the observations.
- If `method='LAWS'`, then the expectile at the intermediate level τ_n is estimated applying the direct LAWS estimator. Instead, If `method='QB'` the indirect QB esimtator is used to estimate the expectile. See Section 3.1 in Padoan and Stupfler (2020) for details.
- When the expectile is estimated by the indirect QB esimtator (`method='QB'`), an estimate of the tail index γ is needed. If `tailest='Hill'` then γ is estimated using the Hill estimator (see also HTailIndex). If `tailest='ExpBased'` then γ is estimated using the expectile based estimator (see EBTailIndex). See Section 3.1 in Padoan and Stupfler (2020) for details.
- `k` or k_n is the value of the so-called intermediate sequence k_n, $n = 1, 2, \ldots$. Its represents a sequence of positive integers such that $k_n \to \infty$ and $k_n/n \to 0$ as $n \to \infty$. Practically, when `method='LAWS'` and `tau=NULL`, k_n specifies by $\tau_n = 1 - k_n/n$ the intermediate level of the expectile. Instead, when `method='QB'`, if `tailest="Hill"` then the value k_n specifies the number of `k`$+1$ larger order statistics to be used to estimate γ by the Hill estimator and if `tau=NULL` then it also specifies by $\tau_n = 1 - k_n/n$ the confidence level τ_n of the quantile to estimate. Finally, if `tailest="ExpBased"` and `tau=NULL` then it also specifies by $\tau_n = 1 - k_n/n$ the intermediate level expectile based esitmator of γ (see EBTailIndex).
- If `var=TRUE` then the asymptotic variance of the expecile estimator is computed. With independent observations the asymptotic variance is computed by the formula Theorem 3.1 of Padoan and Stupfler (2020). This is achieved through `varType="asym-Ind"`. With serial dependent observations the asymptotic variance is estimated by the formula in Theorem 3.1 of Padoan and Stupfler (2020). This is achieved through `varType="asym-Dep"`. In this latter case the computation of the asymptotic variance is based on the "big blocks seperated by small blocks" techinque which is a standard tools in time series, see Leadbetter et al. (1986). See also Section C.1 in Appendix of Padoan and Stupfler (2020). The size of the big and small blocks are specified by the parameters `bigblock` and `smallblock`, respectively. Still with serial dependent observations, If `varType="asym-Dep-Adj"`, then the asymptotic variance is estimated using formula (C.79) in Padoan and Stupfler (2020), see Section C.1 of the Appendix for details.
- Given a small value $\alpha \in (0,1)$ then an asymptotic confidence interval for the τ_n-*th* expectile, with approximate nominal confidence level $(1-\alpha)100\%$ is computed. See Sections 3.1 and C.1 in the Appendix of Padoan and Stupfler (2020).

Value

A list with elements:

- `ExpctHat`: a point estimate of the τ_n-*th* expecile;
- `VarExpHat`: an estimate of the asymptotic variance of the expectile estimator;
- `CIExpct`: an estimate of the approximate $(1-\alpha)100\%$ confidence interval for τ_n-*th* expecile.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://www.unibocconi.it/en/faculty/simone-padoan; Gilles Stupfler, <gilles.stupfler@univ-angers.fr>, https://math.univ-angers.fr/~stupfler/

References

Anthony C. Davison, Simone A. Padoan and Gilles Stupfler (2023). Tail Risk Inference via Expectiles in Heavy-Tailed Time Series, *Journal of Business & Economic Statistics*, **41**(3) 876-889.

Daouia, A., Girard, S. and Stupfler, G. (2018). Estimation of tail risk based on extreme expectiles. *Journal of the Royal Statistical Society: Series B*, **80**, 263-292.

Leadbetter, M.R., Lindgren, G. and Rootzen, H. (1989). Extremes and related properties of random sequences and processes. *Springer*.

See Also

HTailIndex, EBTailIndex, predExpectiles, extQuantile

Examples

```
# Extreme expectile estimation at the intermediate level tau obtained with
# 1-dimensional data simulated from an AR(1) with Student-t innovations

tsDist <- "studentT"
tsType <- "AR"

# parameter setting
corr <- 0.8
df <- 3
par <- c(corr, df)

# Big- small-blocks setting
bigBlock <- 65
smallBlock <- 15

# Intermediate level (or sample tail probability 1-tau)
tau <- 0.99

# sample size
ndata <- 2500

# Simulates a sample from an AR(1) model with Student-t innovations
data <- rtimeseries(ndata, tsDist, tsType, par)

# High expectile (intermediate level) estimation
expectHat <- estExpectiles(data, tau, var=TRUE, bigBlock=bigBlock, smallBlock=smallBlock)
expectHat$ExpctHat
expectHat$CIExpct
```

---

estExtLevel *Extreme Level Estimation*

Description

Estimates the expectile's extreme level corresponding to a quantile's extreme level.

Usage

```
estExtLevel(alpha_n, data=NULL, gammaHat=NULL, VarGamHat=NULL, tailest="Hill", k=NULL,
        var=FALSE, varType="asym-Dep", bigBlock=NULL, smallBlock=NULL, alpha=0.05)
```

Arguments

alpha_n	A real in $(0, 1)$ specifying the extreme level α_n for the quantile. See **Details**.
data	A vector of $(1 \times n)$ observations to be used to estimate the tail index in the case it is not provided. By default data=NULL specifies that no data are given.
gammaHat	A real specifying an estimate of the tail index. By default gammaHat=NULL specifies that no estimate is given. See **Details**.
VarGamHat	A real specifying an estimate of the variance of the tail index estimate. By default VarGamHat=NULL specifies that no estimate is given. See **Details**.
tailest	A string specifying the type of tail index estimator to be used. By default tailest="Hill" specifies the use of Hill estimator. See **Details**.
k	An integer specifying the value of the intermediate sequence k_n. See **Details**.
var	If var=TRUE then an estimate of the variance of the extreme level estimator is computed.
varType	A string specifying the asymptotic variance to compute. By default varType="asym-Dep" specifies the variance estimator for serial dependent observations. See **Details**.
bigBlock	An interger specifying the size of the big-block used to estimaste the asymptotic variance. See **Details**.
smallBlock	An interger specifying the size of the small-block used to estimaste the asymptotic variance. See **Details**.
alpha	A real in $(0, 1)$ specifying the confidence level $(1-\alpha)100\%$ of the approximate confidence interval for the expecile at the intermedite level.

Details

For a given extreme level α_n for the α_n-*th* quantile, an estimate of the extreme level $\tau_n'(\alpha_n)$ is computed such that $\xi_{\tau_n'(\alpha_n)} = q_{\alpha_n}$. The estimator is defined by

$$\hat{\tau}_n'(\alpha_n) = 1 - (1-\alpha_n)\frac{\hat{\gamma}_n}{1-\hat{\gamma}_n}$$

where $\hat{\gamma}_n$ is a consistent estimator of the tail index γ. If a value for the parameter gammaHat is given, then such a value is used to compute $\hat{\tau}_n'$. If gammaHat is NULL and a dataset is provided through the parameter data, then the tail index γ is estimated by a suitable estimator $\hat{\gamma}_n$. See Section 6 in Padoan and Stupfler (2020) for more details.

- If VarGamHat is specified, i.e. the variance of the tail index estimator, then the variance of the extreme level estimator $\hat{\tau}_n'$ is computed by using such value.
- When estimating the tail index, if tailest='Hill' then γ is estimated using the Hill estimator (see also HTailIndex). If tailest='ML' then γ is estimated using the Maximum Likelihood estimator (see MLTailIndex). If tailest='ExpBased' then γ is estimated using the expectile based estimator (see EBTailIndex). If tailest='Moment' then γ is estimated using the moment based estimator (see MomTailIndex). See Padoan and Stupfler (2020) for details.

- k or k_n is the value of the so-called intermediate sequence k_n, $n = 1, 2, \ldots$. Its represents a sequence of positive integers such that $k_n \to \infty$ and $k_n/n \to 0$ as $n \to \infty$. Practically, when `tailest="Hill"` then the value k_n specifies the number of k+1 larger order statistics to be used to estimate γ by the Hill estimator. See MLTailIndex, EBTailIndex and MomTailIndex for the other estimators.
- If `var=TRUE` then the asymptotic variance of the extreme level estimator is computed by applying the delta method, i.e.
  $Var(\tau_n') = Var(\hat{\gamma}_n) * (\alpha_n - 1)^2/(1 - \hat{\gamma}_n)^4$
  where $Var(\hat{\gamma}_n$ is provided by `VarGamHat` or is estimated when esitmating the tail index through `tailest='Hill'` and `tailest='ML'`. See HTailIndex and MLTailIndex for details on how the variance is computed.
- Given a small value $\alpha \in (0, 1)$ then an asymptotic confidence interval for the extreme level, $\tau_n'(\alpha_n)$, with approximate nominal confidence level $(1 - \alpha)100\%$ is computed.

Value

A list with elements:

- `tauHat`: an estimate of the extreme level τ_n';
- `tauVar`: an estimate of the asymptotic variance of the extreme level estimator $\hat{\tau}_n'(\alpha_n)$;
- `tauCI`: an estimate of the approximate $(1 - \alpha)100\%$ confidence interval for the extreme level $\tau_n'(\alpha_n)$.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://www.unibocconi.it/en/faculty/simone-padoan; Gilles Stupfler, <gilles.stupfler@univ-angers.fr>, https://math.univ-angers.fr/~stupfler/

References

Anthony C. Davison, Simone A. Padoan and Gilles Stupfler (2023). Tail Risk Inference via Expectiles in Heavy-Tailed Time Series, *Journal of Business & Economic Statistics*, **41**(3) 876-889.

Daouia, A., Girard, S. and Stupfler, G. (2018). Estimation of tail risk based on extreme expectiles. *Journal of the Royal Statistical Society: Series B*, **80**, 263-292.

See Also

estExpectiles, predExpectiles, extQuantile

Examples

```
# Extreme level estimation for a given quantile's extreme level alpha_n
# obtained with 1-dimensional data simulated from an AR(1) with Student-t innovations

tsDist <- "studentT"
tsType <- "AR"

# parameter setting
```

```
corr <- 0.8
df <- 3
par <- c(corr, df)

# Big- small-blocks setting
bigBlock <- 65
smallBlock <- 15

# quantile's extreme level
alpha_n <- 0.999

# sample size
ndata <- 2500

# Simulates a sample from an AR(1) model with Student-t innovations
data <- rtimeseries(ndata, tsDist, tsType, par)

# expectile's extreme level estimation
tau1Hat <- estExtLevel(alpha_n, data, var=TRUE, k=150, bigBlock=bigBlock,
                       smallBlock=smallBlock)
tau1Hat
```

`estMultiExpectiles` *Multidimensional High Expectile Estimation*

Description

Computes point estimates and $(1-\alpha)100\%$ confidence regions for d-dimensional expectiles at the intermediate level.

Usage

```
estMultiExpectiles(data, tau, method="LAWS", tailest="Hill", var=FALSE,
                   varType="asym-Ind-Adj", k=NULL, alpha=0.05, plot=FALSE)
```

Arguments

`data`	A matrix of $(n \times d)$ observations.
`tau`	A real in $(0, 1)$ specifying the intermediate level τ_n. See **Details**.
`method`	A string specifying the method used to estimate the expecile. By default `est="LAWS"` specifies the use of the direct LAWS estimator. See **Details**.
`tailest`	A string specifying the type of tail index estimator. By default `tailest="Hill"` specifies the use of Hill estimator. See **Details**.
`var`	If `var=TRUE` then an estimate of the variance of the expectile estimator is computed.

varType A string specifying the asymptotic variance-covariance matrix to compute. By default varType="asym-Ind-Adj" specifies that the variance-covariance matrix is computed assuming dependent variables and exploiting a suitable adjustment. See **Details**.

k An integer specifying the value of the intermediate sequence k_n. See **Details**.

alpha A real in $(0, 1)$ specifying the confidence level $(1-\alpha)100\%$ of the approximate confidence region for the d-dimensional expecile at the intermedite level.

plot A logical value. By default plot=FALSE specifies that no graphical representation of the estimates is not provided. See **Details**.

Details

For a dataset data of d-dimensional observations and sample size n, an estimate of the τ_n*-th* d-dimensional is computed. Two estimators are available: the so-called direct Least Asymmetrically Weighted Squares (LAWS) and indirect Quantile-Based (QB). The QB estimator depends on the estimation of the d-dimensional tail index γ. Here, γ is estimated using the Hill estimator (see MultiHTailIndex). The data are regarded as d-dimensional temporal independent observations coming from dependent variables. See Padoan and Stupfler (2020) for details.

- The so-called intermediate level tau or τ_n is a sequence of positive reals such that $\tau_n \to 1$ as $n \to \infty$. Practically, for each individual marginal distribution $\tau_n \in (0, 1)$ is the ratio between *N* (Numerator) and *D* (Denominator). Where *N* is the empirical mean distance of the τ_n*-th* expectile from the observations smaller than it, and *D* is the empirical mean distance of τ_n*-th* expectile from all the observations.
- If method='LAWS', then the expectile at the intermediate level τ_n is estimated applying the direct LAWS estimator. Instead, If method='QB' the indirect QB esimtator is used to estimate the expectile. See Section 2.1 in Padoan and Stupfler (2020) for details.
- When the expectile is estimated by the indirect QB esimtator (method='QB'), an estimate of the d-dimensional tail index γ is needed. Here the d-dimensional tail index γ is estimated using the d-dimensional Hill estimator (tailest='Hill', see MultiHTailIndex). This is the only available option so far (soon more results will be available).
- k or k_n is the value of the so-called intermediate sequence k_n, $n = 1, 2, \ldots$. Its represents a sequence of positive integers such that $k_n \to \infty$ and $k_n/n \to 0$ as $n \to \infty$. Practically, for each marginal distribution, when method='LAWS' and tau=NULL, k_n specifies by $\tau_n = 1 - k_n/n$ the intermediate level of the expectile. Instead, for each marginal distribution, when method='QB', then the value k_n specifies the number of k+1 larger order statistics to be used to estimate γ by the Hill estimator and if tau=NULL then it also specifies by $\tau_n = 1 - k_n/n$ the confidence level τ_n of the quantile to estimate.
- If var=TRUE then an estimate of the asymptotic variance-covariance matrix of the d-dimensional expecile estimator is computed. If the data are regarded as d-dimensional temporal independent observations coming from dependent variables. Then, the asymptotic variance-covariance matrix is estimated by the formulas in section 3.1 of Padoan and Stupfler (2020). In particular, the variance-covariance matrix is computed exploiting the asymptotic behaviour of the relative explectile estimator appropriately normalized and using a suitable adjustment. This is achieved through varType="asym-Ind-Adj". The data can also be regarded as d-dimensional temporal independent observations coming from independent variables. In this

case the asymptotic variance-covariance matrix is diagonal and is also computed exploiting the formulas in section 3.1 of Padoan and Stupfler (2020). This is achieved through `varType="asym-Ind"`.

- Given a small value $\alpha \in (0, 1)$ then an asymptotic confidence region for the τ_n-*th* expectile, with approximate nominal confidence level $(1 - \alpha)100\%$ is computed. In particular, a "symmetric" confidence regions is computed exploiting the asymptotic behaviour of the relative explectile estimator appropriately normalized. See Sections 3.1 of Padoan and Stupfler (2020) for detailed.
- If `plot=TRUE` then a graphical representation of the estimates is not provided.

Value

A list with elements:

- `ExpctHat`: an point estimate of the τ_n-*th* d-dimensional expecile;
- `biasTerm`: an point estimate of the bias term of the estimated expecile;
- `VarCovEHat`: an estimate of the asymptotic variance of the expectile estimator;
- `EstConReg`: an estimate of the approximate $(1 - \alpha)100\%$ confidence region for τ_n-*th* d-dimensional expecile.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://www.unibocconi.it/en/faculty/simone-padoan; Gilles Stupfler, <gilles.stupfler@univ-angers.fr>, https://math.univ-angers.fr/~stupfler/

References

Simone A. Padoan and Gilles Stupfler (2022). Joint inference on extreme expectiles for multivariate heavy-tailed distributions, *Bernoulli* **28**(2), 1021-1048.

See Also

MultiHTailIndex, predMultiExpectiles, extMultiQuantile

Examples

```
# Extreme expectile estimation at the intermediate level tau obtained with
# d-dimensional observations simulated from a joint distribution with
# a Gumbel copula and equal Frechet marginal distributions.
library(plot3D)
library(copula)
library(evd)

# distributional setting
copula <- "Gumbel"
dist <- "Frechet"

# parameter setting
```

```
dep <- 3
dim <- 3
scale <- rep(1, dim)
shape <- rep(3, dim)
par <- list(dep=dep, scale=scale, shape=shape, dim=dim)

# Intermediate level (or sample tail probability 1-tau)
tau <- .95

# sample size
ndata <- 1000

# Simulates a sample from a multivariate distribution with equal Frechet
# marginals distributions and a Gumbel copula
data <- rmdata(ndata, dist, copula, par)
scatter3D(data[,1], data[,2], data[,3])

# High d-dimensional expectile (intermediate level) estimation
expectHat <- estMultiExpectiles(data, tau, var=TRUE)
expectHat$ExpctHat
expectHat$VarCovEHat
# run the following command to see the graphical representation
## Not run:
 expectHat <- estMultiExpectiles(data, tau, var=TRUE, plot=TRUE)

## End(Not run)
```

---

estPOT *Estimation of generalized Pareto distributions*

Description

Bayesian or frequentist estimation of the scale and shape parameters of the continuous or discrete generalized Pareto distribution.

Usage

```
estPOT(
  data,
  k = 10L,
  pn = NULL,
  type = c("continuous", "discrete"),
  method = c("bayesian", "frequentist"),
  prior = "empirical",
  start = NULL,
  sig0 = NULL,
  nsim = 5000L,
  burn = 1000L,
  p = 0.234,
```

```
  optim.method = "Nelder-Mead",
  control = NULL,
  ...
)
```

Arguments

`data`	numeric vector of length n containing complete data values (under and above threshold)
`k`	double indicating the effective sample size. Default: 10
`pn`	numeric vector containing one or more percentile level at which extreme quantiles are estimated, with $p < k/n$. Default: NULL
`type`	string indicating distribution types. Default: `c('continuous','discrete')`
`method`	string indicating estimation methods. Default: `c('bayesian', 'frequentist')`
`prior`	string indicating prior distribution (uniform or empirical). Default: `'empirical'`
`start`	list of 2 containing starting values for scale and shape parameters. Default: `NULL`
`sig0`	double indicating the initial value for the update of the variance in the MCMC algorithm. Default: `NULL`
`nsim`	double indicating the total number of iterations of the MCMC algorithm in the Bayesian estimation case. Default: 5000L
`burn`	double indicating the number of iterations to exclude in the MCMC algorithm of the Bayesian estimation case. Default: 1000L
`p`	double indicating the desired overall sampler acceptance probability. Default: 0.234
`optim.method`	string indicating the optimization method in the frequentist estimation case. Default: `'Nelder-Mead'`
`control`	list containing additional parameters for the minimization function optim. Default: `NULL`
`...`	other arguments passed to the function

Value

a list with the following elements

- Bayesian estimation case
  - `Q.est` matrix with `nsim-burn` rows and `length(pn)` columns containing the posterior sample of the extreme quantile estimated at level given in pn
  - `post_sample` matrix with `nsim-burn` rows and 2 columns containing the posterior sample of the scale and shape parameters for the continuous or discrete generalized Pareto distribution
  - `burn` double indicating the number of iterations excluded in the MCMC algorithm
  - `straight.reject` vector of length `nsim-burn+1` indicating the iterations in which the proposed parameters do not respect basic constraints
  - `sig.vec` vector of length `nsim`$-\lfloor(5/(p(1-p)))\rfloor+1$ containing the values of the variance updated at each iteration of the MCMC algorithm

  - accept.prob matrix containing the values of acceptance probability (second column) corresponding to specific iterations (first column)
  - msg character string containing an output message on the result of the Bayesian estimation procedure
  - mle vector of length 2 containing the maximum likelihood estimates of the scale and shape parameters of the continuous or discrete generalized Pareto distribution
  - t double indicating the threshold for the generalized Pareto model, corresponding to the $n-k$th order statistic of the sample
- Frequentist estimation case
  - est vector of length 2 containing the maximum likelihood estimates of the scale and shape parameters of the continuous or discrete generalized Pareto distribution
  - t double indicating the threshold
  - Q.est vector of dimension length(pn) containing the estimates of the extreme quantile at level given in pn
  - VarCov 2×2 variance-covariance matrix of (gamma, sigma)
  - Q.VC variance of Q.est
  - msg character string containing an output message on the result of the frequentist estimation procedure

References

Dombry, C., S. Padoan and S. Rizzelli (2025). Asymptotic theory for Bayesian inference and prediction: from the ordinary to a conditional Peaks-Over-Threshold method, arXiv:2310.06720v2.

See Also

fpot

Examples

```
## Not run:
# generate data
set.seed(1234)
n <- 500
samp <- evd::rfrechet(n,0,3,4)
# set effective sample size and threshold
k <- 50
threshold <- sort(samp,decreasing = TRUE)[k+1]
# preliminary mle estimates of scale and shape parameters
mlest <- evd::fpot(samp, threshold)
# empirical bayes procedure
proc <- estPOT(
  samp,
  k = k,
  pn = c(0.01, 0.005),
  type = "continuous",
  method = "bayesian",
  prior = "empirical",
  start = as.list(mlest$estimate),
```

```
  sig0 = 0.1)

## End(Not run)
```

---

expectiles *Expectile Computation*

Description

Computes the true expectile for some families of parametric models.

Usage

```
expectiles(par, tau, tsDist="gPareto", tsType="IID", trueMethod="true",
           estMethod="LAWS", nrep=1e+05, ndata=1e+06, burnin=1e+03)
```

Arguments

`par`	A vector of $(1 \times p)$ parameters of the time series parametric family. See **Details**.
`tau`	A real in $(0, 1)$ specifying the level τ of the expectile to be computed. See **Details**.
`tsDist`	A string specifying the parametric family of the innovations distribution. By default `tsDist="gPareto"` specifies a Pareto family of distributions. See **Details**.
`tsType`	A string specifying the type of time series. By default `tsType="IID"` specifies a sequence of independent and indentically distributed random variables. See **Details**.
`trueMethod`	A string specifying the method used to computed the expecile. By default `trueMethod="true"` specifies that the true analytical expression to computed the expectile is used. See **Details**.
`estMethod`	A string specifying the method used to estimate the expecile. By default `est="LAWS"` specifies the use of the direct LAWS estimator. See **Details**.
`nrep`	A positive interger specifying the number of simulations to use for computing an approximation of the expectile. See **Details**.
`ndata`	A positive interger specifying the number of observations to genreated for each simulation. See **Details**.
`burnin`	A positive interger specifying the number of initial observations to discard from the simulated sample.

Details

For a parametric family of time series models or a parametric family of distributions (for the case of independent observations) the τ-*th* expectile (or expectile of level `tau`) is computed.

- There are two methods to compute the τ-*th* expectile. For the Generalised Pareto and Student-*t* parametric families of distributions, the analytical epxression of the expectile is available. This is used to compute the τ-*th* expectile if the parameter `trueMethod="true"` is specified. For most of parametric family of distributions or parametric families of time series models the analytical epxression of the expectile is not available. In this case an approximate value of the τ-*th* expectile is computed via a Monte Carlo method if the parameter `trueMethod=="approx"` is specified. In particular, `ndata` observations from a family of time series models (e.g. `tsType="AR"` and `tsDist="studentT"`) or a sequence of independent and indentically distributed random variables with common family of distributions (e.g. `tsType="IID"` and `tsDist="gPareto"`) are simulated `nrep` times. For each simulation the τ-*th* expectile is estimate by the estimation method specified by `estMethod`. The mean of such estimate provides an approximate value of the τ-*th* expectile. The available estimator to esitmate the expecile are the direct LAWS (`estMethod="LAWS"`) and the indirect QB (`estMethod="QB"`), see estExpectiles for details. The available families of distributions are: Generalised Pareto (`tsDist="gPareto"`), Student-*t* (`tsDist="studentT"`) and Frechet (`tsDist="Frechet"`). The available classes of time series with parametric innovations families of distributions are specified in rtimeseries.

Value

The τ-*th* expectile.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://www.unibocconi.it/en/faculty/simone-padoan; Gilles Stupfler, <gilles.stupfler@univ-angers.fr>, https://math.univ-angers.fr/~stupfler/

References

Anthony C. Davison, Simone A. Padoan and Gilles Stupfler (2023). Tail Risk Inference via Expectiles in Heavy-Tailed Time Series, *Journal of Business & Economic Statistics*, **41**(3) 876-889.

See Also

rtimeseries

Examples

```
# Derivation of the true tau-th expectile for the Pareto distribution
# via accurate simulation

# parameter value
par <- c(1, 0.3)

# Intermediate level (or sample tail probability 1-tau)
tau <- 0.99

trueExp <- expectiles(par, tau)
trueExp
```

```
## Not run:
# tau-th expectile of the AR(1) with Student-t innovations
tsDist <- "studentT"
tsType <- "AR"

# Approximation via Monte Carlo methods
trueMethod <- "approx"

# parameter setting
corr <- 0.8
df <- 3
par <- c(corr, df)

# Intermediate level (or sample tail probability 1-tau)
tau <- 0.99

trueExp <- expectiles(par, tau, tsDist, tsType, trueMethod)
trueExp

## End(Not run)
```

ExpectMES *Marginal Expected Shortfall Expectile Based Estimation*

Description

Computes a point and interval estimate of the Marginal Expected Shortfall (MES) using an expectile based approach.

Usage

```
ExpectMES(data, tau, tau1, method="LAWS", var=FALSE, varType="asym-Dep", bias=FALSE,
          bigBlock=NULL, smallBlock=NULL, k=NULL, alpha_n=NULL, alpha=0.05)
```

Arguments

data	A vector of $(1 \times n)$ observations.
tau	A real in $(0, 1)$ specifying the intermediate level τ_n. See **Details**.
tau1	A real in $(0, 1)$ specifying the extreme level τ'_n. See **Details**.
method	A string specifying the method used to estimate the expecile. By default `est="LAWS"` specifies the use of the LAWS based estimator. See **Details**.
var	If `var=TRUE` then an estimate of the asymptotic variance of the MES estimator is computed.
varType	A string specifying the type of asymptotic variance to compute. By default `varType="asym-Dep"` specifies the variance estimator for serial dependent observations. See **Details**.

`bias`	A logical value. By default `bias=FALSE` specifies that no bias correction is computed. See **Details**.
`bigBlock`	An interger specifying the size of the big-block used to estimaste the asymptotic variance. See **Details**.
`smallBlock`	An interger specifying the size of the small-block used to estimaste the asymptotic variance. See **Details**.
`k`	An integer specifying the value of the intermediate sequence k_n. See **Details**.
`alpha_n`	A real in $(0, 1)$ specifying the quantile's extreme level to be use in order to estimate the expectile's extreme level.
`alpha`	A real in $(0, 1)$ specifying the confidence level $(1 - \alpha)100\%$ of the approximate confidence interval for the expecile at the intermedite level.

Details

For a dataset `data` of sample size n, an estimate of the τ_n'-*th* MES is computed. The estimation of the MES at the extreme level `tau1` (τ_n') is indeed meant to be a prediction. Two estimators are available: the so-called Least Asymmetrically Weighted Squares (LAWS) based estimator and the Quantile-Based (QB) estimator. The definition of both estimators depends on the estimation of the tail index γ. Here, γ is estimated using the Hill estimation (see HTailIndex for details). The observations can be either independent or temporal dependent. See Section 4 in Padoan and Stupfler (2020) for details.

- The so-called intermediate level `tau` or τ_n is a sequence of positive reals such that $\tau_n \to 1$ as $n \to \infty$. See predExpectiles for details.
- The so-called extreme level `tau1` or τ_n' is a sequence of positive reals such that $\tau_n' \to 1$ as $n \to \infty$. See predExpectiles for details.
- When `method='LAWS'`, then the τ_n'-*th* MES is estimated using the LAWS based estimator. When `method='QB'`, the expectile is instead estimated using the QB esimtator. See Sectino 4 in Padoan and Stupfler (2020) and in particular Corollary 4.3 and 4.4 for details. The definition of both estimators depend on the estimation of the tail index γ. In particular, the tail index γ is estimated using the Hill estimator (see HTailIndex).
- If `var=TRUE` then an esitmate of the asymptotic variance of the tau_n'-*th* MES is computed. Notice that the estimation of the asymptotic variance **is only available** when γ is estimated using the Hill estimator (see HTailIndex). With independent observations the asymptotic variance is estimated by $\hat{\gamma}^2$, see Corollary 4.3 in Padoan and Stupfler (2020). This is achieved through `varType="asym-Ind"`. With serial dependent observations the asymptotic variance is estimated by the formula in Corollary 4.3 of Padoan and Stupfler (2020). This is achieved through `varType="asym-Dep"`. See Section 4 adn 5 in Padoan and Stupfler (2020) for details. In this latter case the computation of the serial dependence is based on the "big blocks seperated by small blocks" techinque which is a standard tools in time series, see e.g. Leadbetter et al. (1986). The size of the big and small blocks are specified by the parameters `bigBlock` and `smallBlock`, respectively.
- If `bias=TRUE` then γ is estimated using formula (4.2) of Haan et al. (2016). This is used by the LAWS and QB estimators. Furthermore, the τ_n'–*th* quantile is estimated using the formula in page 330 of de Haan et al. (2016). This provides a bias corrected version of the Weissman estimator. This is used by the QB estimator. However, in this case the asymptotic variance

is not estimated using the formula in Haan et al. (2016) Theorem 4.2. Instead, for simplicity the asymptotic variance is estimated by the formula in Corollary 3.8, with serial dependent observations, and $\hat{\gamma}^2$ with independent observation (see e.g. de Drees 2000, for the details).

- `k` or k_n is the value of the so-called intermediate sequence k_n, $n = 1, 2, \ldots$. Its represents a sequence of positive integers such that $k_n \to \infty$ and $k_n/n \to 0$ as $n \to \infty$. Practically, when `tau=NULL` and `method='LAWS'`, then $\tau_n = 1 - k_n/n$ is the intermediate level of the expectile to be stimated. k_n also specifies the number of `k`+1 larger order statistics used in the definition of the Hill estimator (see HTailIndex for detail). Differently, When `tau=NULL` and `method='QB'`, then $\tau_n = 1 - k_n/n$ is the intermediate level of the quantile to be stimated.
- If the quantile's extreme level is provided by `alpha_n`, then expectile's extreme level tau'_n is replaced by $tau'_n(\alpha_n)$ which is estimated by the method described in Section 6 of Padoan and Stupfler (2020). See estExtLevel for details.
- Given a small value $\alpha \in (0, 1)$ then an estimate of an asymptotic confidence interval for tau'_n-*th* expectile, with approximate nominal confidence level $(1 - \alpha)100\%$, is computed. The confidence intervals are computed exploiting formula in Corollary 4.3, 4.4 and Theorem 6.2 of Padoan and Stupfler (2020) and (46) in Drees (2003). See Sections 4-6 in Padoan and Stupfler (2020) for details. When `biast=TRUE` confidence intervals are computed in the same way but after correcting the tail index estimate by an estimate of the bias term, see formula (4.2) in de Haan et al. (2016) for details.

Value

A list with elements:

- `HatXMES`: an estimate of the τ'_n-*th* expectile based MES;
- `VarHatXMES`: an estimate of the asymptotic variance of the expectile based MES estimator;
- `CIHatXMES`: an estimate of the approximate $(1 - \alpha)100\%$ confidence interval for τ'_n-*th* MES.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://www.unibocconi.it/en/faculty/simone-padoan; Gilles Stupfler, <gilles.stupfler@univ-angers.fr>, https://math.univ-angers.fr/~stupfler/

References

Anthony C. Davison, Simone A. Padoan and Gilles Stupfler (2023). Tail Risk Inference via Expectiles in Heavy-Tailed Time Series, *Journal of Business & Economic Statistics*, **41**(3) 876-889.

Daouia, A., Girard, S. and Stupfler, G. (2018). Estimation of tail risk based on extreme expectiles. *Journal of the Royal Statistical Society: Series B*, **80**, 263-292.

de Haan, L., Mercadier, C. and Zhou, C. (2016). Adapting extreme value statistics to financial time series: dealing with bias and serial dependence. *Finance and Stochastics*, **20**, 321-354.

Drees, H. (2003). Extreme quantile estimation for dependent data, with applications to finance. *Bernoulli*, **9**, 617-657.

Drees, H. (2000). Weighted approximations of tail processes for β-mixing random variables. *Annals of Applied Probability*, **10**, 1274-1301.

Leadbetter, M.R., Lindgren, G. and Rootzen, H. (1989). Extremes and related properties of random sequences and processes. *Springer*.

See Also

QuantMES, HTailIndex, predExpectiles, extQuantile

Examples

```
# Marginl Expected Shortfall expectile based estimation at the extreme level
# obtained with 2-dimensional data simulated from an AR(1) with bivariate
# Student-t distributed innovations

tsDist <- "AStudentT"
tsType <- "AR"
tsCopula <- "studentT"

# parameter setting
corr <- 0.8
dep <- 0.8
df <- 3
par <- list(corr=corr, dep=dep, df=df)

# Big- small-blocks setting
bigBlock <- 65
smallBlock <- 15

# quantile's extreme level
alpha_n <- 0.999

# sample size
ndata <- 2500

# Simulates a sample from an AR(1) model with Student-t innovations
data <- rbtimeseries(ndata, tsDist, tsType, tsCopula, par)


# Extreme MES expectile based estimation
MESHat <- ExpectMES(data, NULL, NULL, var=TRUE, k=150, bigBlock=bigBlock,
                    smallBlock=smallBlock, alpha_n=alpha_n)
MESHat
```

---

extBQuant *Bayesian extreme quantile*

Description

Given posterior samples for the parameters of the continuous or discrete generalized Pareto distribution, return the posterior mean and $1-\alpha$ level credibility intervals of the extreme quantile

Usage

```
extBQuant(
  threshold,
  postsamp,
  k,
  n,
  retp,
  alpha = 0.05,
  type = c("continuous", "discrete")
)
```

Arguments

`threshold`	threshold for the generalized Pareto model, corresponding to the $n-k$th order statistic of the sample
`postsamp`	a `m` by `2` matrix with columns containing the posterior samples of scale and shape parameters of the generalized Pareto distribution, respectively
`k`	integer, number of exceedances for the generalized Pareto (only used if `extrapolation=TRUE`)
`n`	integer, number of observations in the full sample. Must be greater than `k` (only used if `extrapolation=TRUE`)
`retp`	double indicating the value of the return period
`alpha`	level for credibility interval. Default: 0.05 giving 95% credibility intervals
`type`	string indicating distribution types. Default: `c('continuous','discrete')`

Value

a list with components

- `mQ` posterior mean of the extreme quantile
- `ciQ` vector of dimension 2 returning the $\alpha/2$ and $1-\alpha/2$ empirical quantiles of the posterior distribution of the extreme quantile

Examples

```
## Not run:
# generate data
set.seed(1234)
n <- 500
samp <- evd::rfrechet(500,0,3,4)
# set effective sample size and threshold
k <- 50
threshold <- sort(samp,decreasing = TRUE)[k+1]
# preliminary mle estimates of scale and shape parameters
mlest <- evd::fpot(samp, threshold)
# empirical bayes procedure
proc <- estPOT(
  samp,
  k = k,
```

```
  pn = c(0.01, 0.005),
  type = "continuous",
  method = "bayesian",
  prior = "empirical",
  start = as.list(mlest$estimate),
  sig0 = 0.1)
# extreme quantile corresponding to a return period of 100
extBQuant(
  proc$t,proc$post_sample,
  k,
  n,
  100,
  0.05,
  type = "continuous")

## End(Not run)
```

---

extBQuantx *Conditional Bayesian extreme quantile*

Description

Given posterior samples for the parameters of the continuous or discrete generalized Pareto distribution and scedasis function for a set of covariates, return the posterior mean and $1 - \alpha$ level credibility intervals of the extreme quantile for each value of the scedasis function

Usage

```
extBQuantx(
  cx,
  postsamp,
  threshold,
  n,
  qlev,
  k,
  type = c("continuous", "discrete"),
  confint = c("asymmetric", "symmetric"),
  alpha = 0.05,
  ...
)
```

Arguments

cx	an `m` by `p` matrix, obtained by evaluating the scedasis function for every of the `p` covariate values and every `m` posterior draw
postsamp	a `m` by 2 matrix with columns containing the posterior samples of scale and shape parameters of the generalized Pareto distribution, respectively

threshold	threshold for the generalized Pareto model, corresponding to the $n-k$th order statistic of the sample
n	integer, number of observations in the full sample. Must be greater than k
qlev	double indicating the percentile level at which the extreme quantile is estimated. Must be smaller than k/n
k	integer, number of exceedances for the generalized Pareto (only used if extrapolation=TRUE)
type	string indicating distribution types. Default: c('continuous','discrete')
confint	type of confidence interval. Default: c('asymmetric', 'symmetric')
alpha	level for credibility interval. Default 0.05, giving 95% credibility intervals
...	additional arguments, for back-compatilibity

Value

a list with components

- mQ posterior mean of the extreme quantile
- ciQ vector of dimension 2 returning the $\alpha/2$ and $1-\alpha/2$ empirical quantiles of the posterior distribution of the extreme quantile

Examples

```
## Not run:
# generate data
set.seed(1234)
n <- 500
samp <- evd::rfrechet(n,0,1:n,4)
# set effective sample size and threshold
k <- 50
threshold <- sort(samp,decreasing = TRUE)[k+1]
# preliminary mle estimates of scale and shape parameters
mlest <- evd::fpot(
  samp,
  threshold,
  control = list(maxit = 500))
# empirical bayes procedure
proc <- estPOT(
  samp,
  k = k,
  pn = c(0.01, 0.005),
  type = "continuous",
  method = "bayesian",
  prior = "empirical",
  start = as.list(mlest$estimate),
  sig0 = 0.1)
# conditional predictive density estimation
yg <- seq(0, 50, by = 2)
nyg <- length(yg)
# estimation of scedasis function
# setting
```

```
M <- 1e3
C <- 5
alpha <- 0.05
bw <- .5
nsim <- 5000
burn <- 1000
# create covariate
# in sample obs
n_in = n
# number of years ahead
nY = 1
n_out = 365 * nY
# total obs
n_tot = n_in + n_out
# total covariate (in+out sample period)
x <- seq(0, 1, length = n_tot)
# in sample grid dimension for covariate
ng_in <- 150
xg <- seq(0, x[n_in], length = ng_in)
# in+out of sample grid
xg <- c(xg, seq(x[n_in + 1], x[(n_tot)], length = ng_in))
# in+out sample grid dimension
nxg <- length(xg)
xg <- array(xg, c(nxg, 1))
# in sample observations
samp_in <- samp[1:n_in]
ssamp_in <- sort(samp_in, decreasing = TRUE, index = TRUE)
x_in <- x[1:n_in] # in sample covariate
xs <- x_in[ssamp_in$ix[1:k]] # in sample concomitant covariate
# estimate scedasis function over the in and out of sample period
res_stat <- apply(
  xg,
  1,
  cpost_stat,
  N = nsim - burn,
  x = x_in,
  xs = xs,
  bw = bw,
  k = k,
  C = C
)
# conditional predictive posterior density
probq = 1 - 0.99
res_AQ <- extBQuantx(
 cx = res_stat,
 postsamp = proc$post_sample,
 threshold = proc$t,
 n = n,
 qlev = probq,
 k = k,
 type = "continuous",
 confint = "asymmetric")
```

```
## End(Not run)
```

`extMultiQuantile` *Multidimensional Value-at-Risk (VaR) or Extreme Quantile (EQ) Estimation*

Description

Computes point estimates and $(1-\alpha)100\%$ confidence regions for d-dimensional VaR based on the Weissman estimator.

Usage

```
extMultiQuantile(data, tau, tau1, var=FALSE, varType="asym-Ind-Log", bias=FALSE,
                 k=NULL, alpha=0.05, plot=FALSE)
```

Arguments

`data`	A matrix of $(n \times d)$ observations.
`tau`	A real in $(0, 1)$ specifying the intermediate level τ_n. See **Details**.
`tau1`	A real in $(0, 1)$ specifying the extreme level τ_n'. See **Details**.
`var`	If `var=TRUE` then an estimate of the asymptotic variance-covariance matrix of the d-dimensional VaR estimator is computed.
`varType`	A string specifying the type of asymptotic variance-covariance matrix to compute. By default `varType="asym-Ind-Log"` specifies that the variance-covariance matrix is obtained assuming dependent variables and exploiting the logarithmic scale. See **Details**.
`bias`	A logical value. By default `biast=FALSE` specifies that no bias correction is computed. See **Details**.
`k`	An integer specifying the value of the intermediate sequence k_n. See **Details**.
`alpha`	A real in $(0, 1)$ specifying the confidence level $(1-\alpha)100\%$ of the approximate confidence region for the d-dimensional VaR.
`plot`	A logical value. By default `plot=FALSE` specifies that no graphical representation of the estimates is not provided. See **Details**.

Details

For a dataset `data` of d-dimensional observations and sample size n, the VaR or EQ, correspoding to the extreme level `tau1`, is computed by applying the d-dimensional Weissman estimator. The definition of the Weissman estimator depends on the estimation of the d-dimensional tail index γ. Here, γ is estimated using the Hill estimation (see MultiHTailIndex). The data are regarded as d-dimensional temporal independent observations coming from dependent variables. See Padoan and Stupfler (2020) for details.

- The so-called intermediate level `tau` or τ_n is a sequence of positive reals such that $\tau_n \to 1$ as $n \to \infty$. Practically, for each variable, $(1 - \tau_n) \in (0, 1)$ is a small proportion of observations in the observed data sample that exceed the tau_n*-th* empirical quantile. Such proportion of observations is used to estimate the individual tau_n*-th* quantile and tail index γ.
- The so-called extreme level `tau1` or τ'_n is a sequence of positive reals such that $\tau'_n \to 1$ as $n \to \infty$. For each variable, the value $(1 - tau'_n) \in (0, 1)$ is meant to be a small tail probability such that $(1 - \tau'_n) = 1/n$ or $(1 - \tau'_n) < 1/n$. It is also assumed that $n(1 - \tau'_n) \to C$ as $n \to \infty$, where C is a positive finite constant. The value C is the expected number of exceedances of the individual τ'_n*-th* quantile. Typically, $C \in (0, 1)$ which means that it is expected that there are no observations in a data sample exceeding the individual quantile of level $(1 - \tau'_n)$.
- If `var=TRUE` then an estimate of the asymptotic variance-covariance matrix of the tau'_n*-th* `d`-dimensional quantile is computed. The data are regarded as temporal independent observations coming from dependent variables. The asymptotic variance-covariance matrix is estimated exploiting the formula in Section 5 of Padoan and Stupfler (2020). In particular, the variance-covariance matrix is computed exploiting the asymptotic behaviour of the normalized quantile estimator which is expressed in logarithmic scale. This is achieved through `varType="asym-Ind-Log"`. If `varType="asym-Ind"` then the variance-covariance matrix is computed exploiting the asymptotic behaviour of the `d`-dimensional relative quantile estimator appropriately normalized (see formula in Section 5 of Padoan and Stupfler (2020)).
- If `bias=TRUE` then an estimate of each individual τ'_n*–th* quantile is estimated using the formula in page 330 of de Haan et al. (2016), which provides a bias corrected version of the Weissman estimator. However, in this case the asymptotic variance is not estimated using the formula in Haan et al. (2016) Theorem 4.2. For simplicity standard the variance-covariance matrix is still computed using formula in Section 5 of Padoan and Stupfler (2020).
- `k` or k_n is the value of the so-called intermediate sequence k_n, $n = 1, 2, \ldots$. Its represents a sequence of positive integers such that $k_n \to \infty$ and $k_n/n \to 0$ as $n \to \infty$. Practically, for each marginal distribution, the value k_n specifies the number of `k`$+1$ larger order statistics to be used to estimate the individual τ_n*-th* empirical quantile and individual tail index γ_j for $j = 1, \ldots, d$. The intermediate level τ_n can be seen defined as $\tau_n = 1 - k_n/n$.
- Given a small value $\alpha \in (0, 1)$ then an estimate of an asymptotic confidence region for tau'_n*-th* `d`-dimensional quantile, with approximate nominal confidence level $(1 - \alpha)100\%$, is computed. The confidence regions are computed exploiting the asymptotic behaviour of the normalized quantile estimator in logarithmic scale. This is an "asymmetric" region and it is achieved through `varType="asym-Ind-Log"`. A "symmetric" region is obtained exploiting the asymptotic behaviour of the relative quantile estimator appropriately normalized, see formula in Section 5 of Padoan and Stupfler (2020). This is achieved through `varType="asym-Ind"`.
- If `plot=TRUE` then a graphical representation of the estimates is not provided.

Value

A list with elements:

- `ExtQHat`: an estimate of the `d`-dimensional VaR or τ'_n*-th* `d`-dimensional quantile;
- `VarCovExQHat`: an estimate of the asymptotic variance-covariance of the `d`-dimensional VaR estimator;
- `EstConReg`: an estimate of the approximate $(1 - \alpha)100\%$ confidence regions for the `d`-dimensional VaR.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://www.unibocconi.it/en/faculty/simone-padoan; Gilles Stupfler, <gilles.stupfler@univ-angers.fr>, https://math.univ-angers.fr/~stupfler/

References

Simone A. Padoan and Gilles Stupfler (2022). Joint inference on extreme expectiles for multivariate heavy-tailed distributions, *Bernoulli* **28**(2), 1021-1048.

de Haan, L., Mercadier, C. and Zhou, C. (2016). Adapting extreme value statistics to financial time series: dealing with bias and serial dependence. *Finance and Stochastics*, **20**, 321-354.

de Haan, L. and Ferreira, A. (2006). Extreme Value Theory: An Introduction. *Springer-Verlag*, New York.

See Also

MultiHTailIndex, estMultiExpectiles, predMultiExpectiles

Examples

```
# Extreme quantile estimation at the extreme level tau1 obtained with
# d-dimensional observations simulated from a joint distribution with
# a Gumbel copula and equal Frechet marginal distributions.
library(plot3D)
library(copula)
library(evd)

# distributional setting
copula <- "Gumbel"
dist <- "Frechet"

# parameter setting
dep <- 3
dim <- 3
scale <- rep(1, dim)
shape <- rep(3, dim)
par <- list(dep=dep, scale=scale, shape=shape, dim=dim)

# Intermediate level (or sample tail probability 1-tau)
tau <- 0.95
# Extreme level (or tail probability 1-tau1 of unobserved quantile)
tau1 <- 0.9995

# sample size
ndata <- 1000

# Simulates a sample from a multivariate distribution with equal Frechet
# marginals distributions and a Gumbel copula
data <- rmdata(ndata, dist, copula, par)
scatter3D(data[,1], data[,2], data[,3])
```

```
# High d-dimensional expectile (intermediate level) estimation
extQHat <- extMultiQuantile(data, tau, tau1, TRUE)

extQHat$ExtQHat
extQHat$VarCovExQHat
# run the following command to see the graphical representation
## Not run:
 extQHat <- extMultiQuantile(data, tau, tau1, TRUE, plot=TRUE)

## End(Not run)
```

`extQuantile` *Value-at-Risk (VaR) or Extreme Quantile (EQ) Estimation*

Description

Computes a point and interval estimate of the VaR based on the Weissman estimator.

Usage

```
extQuantile(data, tau, tau1, var=FALSE, varType="asym-Dep", bias=FALSE, bigBlock=NULL,
            smallBlock=NULL, k=NULL, alpha=0.05)
```

Arguments

`data`	A vector of $(1 \times n)$ observations.
`tau`	A real in $(0, 1)$ specifying the intermediate level τ_n. See **Details**.
`tau1`	A real in $(0, 1)$ specifying the extreme level τ'_n. See **Details**.
`var`	If `var=TRUE` then an estimate of the asymptotic variance of the VaR estimator is computed.
`varType`	A string specifying the type of asymptotic variance to compute. By default `varType="asym-Dep"` specifies the variance estimator for serial dependent observations. See **Details**.
`bias`	A logical value. By default `biast=FALSE` specifies that no bias correction is computed. See **Details**.
`bigBlock`	An interger specifying the size of the big-block used to estimaste the asymptotic variance. See **Details**.
`smallBlock`	An interger specifying the size of the small-block used to estimaste the asymptotic variance. See **Details**.
`k`	An integer specifying the value of the intermediate sequence k_n. See **Details**.
`alpha`	A real in $(0, 1)$ specifying the confidence level $(1-\alpha)100\%$ of the approximate confidence interval for the VaR.

Details

For a dataset `data` of sample size n, the VaR or EQ, correspoding to the extreme level `tau1`, is computed by applying the Weissman estimator. The definition of the Weissman estimator depends on the estimation of the tail index γ. Here, γ is estimated using the Hill estimation (see HTailIndex). The observations can be either independent or temporal dependent (see e.g. de Haan and Ferreira 2006; Drees 2003; de Haan et al. 2016 for details).

- The so-called intermediate level `tau` or τ_n is a sequence of positive reals such that $\tau_n \to 1$ as $n \to \infty$. Practically, $(1 - \tau_n) \in (0, 1)$ is a small proportion of observations in the observed data sample that exceed the tau_n*-th* empirical quantile. Such proportion of observations is used to estimate the tau_n*-th* quantile and γ.
- The so-called extreme level `tau1` or τ_n' is a sequence of positive reals such that $\tau_n' \to 1$ as $n \to \infty$. The value $(1 - tau_n') \in (0, 1)$ is meant to be a small tail probability such that $(1 - \tau_n') = 1/n$ or $(1 - \tau_n') < 1/n$. It is also assumed that $n(1 - \tau_n') \to C$ as $n \to \infty$, where C is a positive finite constant. The value C is the expected number of exceedances of the τ_n'*-th* quantile. Typically, $C \in (0, 1)$ which means that it is expected that there are no observations in a data sample exceeding the quantile of level $(1 - \tau_n')$.
- If `var=TRUE` then an estimate of the asymptotic variance of the tau_n'*-th* quantile is computed. With independent observations the asymptotic variance is estimated by the formula $\hat{\gamma}^2$ (see e.g. de Drees 2000, 2003, for details). This is achieved through `varType="asym-Ind"`. With serial dependent data the asymptotic variance is estimated by the formula in 1288 in Drees (2000). This is achieved through `varType="asym-Dep"`. In this latter case the computation of the serial dependence is based on the "big blocks seperated by small blocks" techinque which is a standard tools in time series, see e.g. Leadbetter et al. (1986). The size of the big and small blocks are specified by the parameters `bigBlock` and `smallBlock`, respectively. With serial dependent data the asymptotic variance can also be estimated by formula (32) of Drees (2003). This is achieved through `varType="asym-Alt-Dep"`.
- If `bias=TRUE` then an estimate of the τ_n'–*th* quantile is computed using the formula in page 330 of de Haan et al. (2016), which provides a bias corrected version of the Weissman estimator. However, in this case the asymptotic variance is not estimated using the formula in Haan et al. (2016) Theorem 4.2. Instead, for simplicity standard formula in Drees (2000) page 1288 is used.
- `k` or k_n is the value of the so-called intermediate sequence k_n, $n = 1, 2, \ldots$. Its represents a sequence of positive integers such that $k_n \to \infty$ and $k_n/n \to 0$ as $n \to \infty$. Practically, the value k_n specifies the number of `k`+1 larger order statistics to be used to estimate the τ_n*-th* empirical quantile and γ. The intermediate level τ_n can be seen defined as $\tau_n = 1 - k_n/n$.
- Given a small value $\alpha \in (0, 1)$ then an estimate of an asymptotic confidence interval for tau_n'*-th* quantile, with approximate nominal confidence level $(1 - \alpha)100\%$, is computed. The confidence intervals are computed exploiting the formulas (33) and (46) of Drees (2003). When `biast=TRUE` confidence intervals are computed in the same way but after correcting the tail index estimate by an estimate of the bias term, see formula (4.2) in de Haan et al. (2016) for details. Furthermore, in this case with serial dependent data the asymptotic variance is estimated using the formula in Drees (2000) page 1288.

Value

A list with elements:

- `ExtQHat`: an estimate of the VaR or τ_n'-*th* quantile;
- `VarExQHat`: an estimate of the asymptotic variance of the VaR estimator;
- `CIExtQ`: an estimate of the approximate $(1-\alpha)100\%$ confidence interval for the VaR.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://www.unibocconi.it/en/faculty/simone-padoan; Gilles Stupfler, <gilles.stupfler@univ-angers.fr>, https://math.univ-angers.fr/~stupfler/

References

Anthony C. Davison, Simone A. Padoan and Gilles Stupfler (2023). Tail Risk Inference via Expectiles in Heavy-Tailed Time Series, *Journal of Business & Economic Statistics*, **41**(3) 876-889.

de Haan, L., Mercadier, C. and Zhou, C. (2016). Adapting extreme value statistics to financial time series: dealing with bias and serial dependence. *Finance and Stochastics*, **20**, 321-354.

de Haan, L. and Ferreira, A. (2006). Extreme Value Theory: An Introduction. *Springer-Verlag*, New York.

Drees, H. (2000). Weighted approximations of tail processes for β-mixing random variables. *Annals of Applied Probability*, **10**, 1274-1301.

Drees, H. (2003). Extreme quantile estimation for dependent data, with applications to finance. *Bernoulli*, **9**, 617-657.

Leadbetter, M.R., Lindgren, G. and Rootzen, H. (1989). Extremes and related properties of random sequences and processes. *Springer*.

See Also

HTailIndex, EBTailIndex, estExpectiles

Examples

```
# Extreme quantile estimation at the level tau1 obtained with 1-dimensional data
# simulated from an AR(1) with univariate Student-t distributed innovations

tsDist <- "studentT"
tsType <- "AR"

# parameter setting
corr <- 0.8
df <- 3
par <- c(corr, df)

# Big- small-blocks setting
bigBlock <- 65
smallBlock <- 15

# Intermediate level (or sample tail probability 1-tau)
tau <- 0.97
# Extreme level (or tail probability 1-tau1 of unobserved quantile)
```

```
tau1 <- 0.9995

# sample size
ndata <- 2500

# Simulates a sample from an AR(1) model with Student-t innovations
data <- rtimeseries(ndata, tsDist, tsType, par)

# VaR (extreme quantile) estimation
extQHat1 <- extQuantile(data, tau, tau1, TRUE, bigBlock=bigBlock, smallBlock=smallBlock)
extQHat1$ExtQHat
extQHat1$CIExtQ

# VaR (extreme quantile) estimation with bias correction
extQHat2 <- extQuantile(data, tau, tau1, TRUE, bias=TRUE, bigBlock=bigBlock, smallBlock=smallBlock)
extQHat2$ExtQHat
extQHat2$CIExtQ
```

`fitdGPD` *Maximum likelihood estimation of the parameters of the discrete generalized Pareto distribution*

Description

Given a sample of exceedances, estimate the parameters via maximum likelihood along with $1-\alpha$ level confidence intervals.

Usage

```
fitdGPD(excess, alpha = 0.05)
```

Arguments

`excess`	vector of positive exceedances, i.e., $Y-t \mid Y>t$, with t being the threshold
`alpha`	level for confidence interval of scale and shape parameters. Default: 0.05, giving 95% confidence intervals

Value

a list with elements

- `mle` vector of dimension 2 containing estimated scale and shape parameters
- `CI` matrix of dimension 2 by 2 containing the $1-\alpha$ level confidence intervals for scale and shape

References

Hitz, A.S., G. Samorodnistsky and R.A. Davis (2024). Discrete Extremes, *Journal of Data Science*, 22(**4**), pp. 524-536.

Examples

```
fitdGPD(rpois(1000,2))
```

`HTailIndex` *Hill Tail Index Estimation*

Description

Computes a point and interval estimate of the tail index based on the Hill's estimator.

Usage

```
HTailIndex(data, k, var=FALSE, varType="asym-Dep", bias=FALSE, bigBlock=NULL,
           smallBlock=NULL, alpha=0.05)
```

Arguments

`data`	A vector of $(1 \times n)$ observations.
`k`	An integer specifying the value of the intermediate sequence k_n. See **Details**.
`var`	If `var=TRUE` then an estimate of the variance of the tail index estimator is computed.
`varType`	A string specifying the asymptotic variance to compute. By default `varType="asym-Dep"` specifies the variance estimator for serial dependent observations. See **Details**.
`bias`	A logical value. By default `biast=FALSE` specifies that no bias correction is computed. See **Details**.
`bigBlock`	An interger specifying the size of the big-block used to estimaste the asymptotic variance. See **Details**.
`smallBlock`	An interger specifying the size of the small-block used to estimaste the asymptotic variance. See **Details**.
`alpha`	A real in $(0, 1)$ specifying the confidence level $(1 - \alpha)100\%$ of the approximate confidence interval for the tail index.

Details

For a dataset `data` of sample size n, the tail index γ of its (marginal) distribution is computed by applying the Hill estimator. The observations can be either independent or temporal dependent.

- `k` or k_n is the value of the so-called intermediate sequence k_n, $n = 1, 2, \ldots$. Its represents a sequence of positive integers such that $k_n \to \infty$ and $k_n/n \to 0$ as $n \to \infty$. Practically, the value k_n specifies the number of `k`+1 larger order statistics to be used to estimate γ.
- If `var=TRUE` then an estimate of the asymptotic variance of the Hill estimator is computed. With independent observations the asymptotic variance is estimated by the formula $\hat{\gamma}^2$, see Theorem 3.2.5 of de Haan and Ferreira (2006). This is achieved through `varType="asym-Ind"`. With serial dependent observations the asymptotic variance is estimated by the formula in 1288 in Drees (2000). This is achieved through `varType="asym-Dep"`. In this latter case the

serial dependence is estimated by exploiting the "big blocks seperated by small blocks" technique which is a standard tools in time series, see Leadbetter et al. (1986). See also formula (11) in Drees (2003). The size of the big and small blocks are specified by the parameters bigBlock and smallBlock, respectively.

- If bias=TRUE then an estimate of the bias term of the Hill estimator is computed implementing using formula (4.2) in de Haan et al. (2016). However, in this case the asymptotic variance is not estimated using the formula in Haan et al. (2016) Theorem 4.1. Instead for simplicity standard formulas have been used (see de Haan and Ferreira 2006 Theorem 3.2.5 and Drees 2000 page 1288).
- Given a small value $\alpha \in (0,1)$ then an estimate of an asymptotic confidence interval for γ, with approximate nominal confidence level $(1-\alpha)100\%$, is computed. The confidence intervals are computed exploiting the formulas in de Haan and Ferreira (2006) Theorem 3.2.5 and Drees (2000) page 1288. When biast=TRUE the confidence intervals are computed in the same way but after correcting the tail index estimate by an estimate of the bias term, see formula (4.2) in de Haan et al. (2016) for details.

Value

A list with elements:

- gammaHat: an estimate of tail index γ;
- VarGamHat: an estimate of the asymptotic variance of the Hill estimator;
- BiasGamHat: an estimate of bias term of the Hill estimator;
- AdjExtQHat: the adjustment to correct the Weissman estimator of an extreme quantile.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://www.unibocconi.it/en/faculty/simone-padoan; Gilles Stupfler, <gilles.stupfler@univ-angers.fr>, https://math.univ-angers.fr/~stupfler/

References

Anthony C. Davison, Simone A. Padoan and Gilles Stupfler (2023). Tail Risk Inference via Expectiles in Heavy-Tailed Time Series, *Journal of Business & Economic Statistics*, **41**(3) 876-889.

de Haan, L., Mercadier, C. and Zhou, C. (2016). Adapting extreme value statistics to financial time series: dealing with bias and serial dependence. *Finance and Stochastics*, **20**, 321-354.

de Haan, L. and Ferreira, A. (2006). Extreme Value Theory: An Introduction. *Springer-Verlag*, New York.

Drees, H. (2000). Weighted approximations of tail processes for β-mixing random variables. *Annals of Applied Probability*, **10**, 1274-1301.

Leadbetter, M.R., Lindgren, G. and Rootzen, H. (1989). Extremes and related properties of random sequences and processes. *Springer*.

See Also

MLTailIndex, MomTailIndex, EBTailIndex

Examples

```
# Tail index estimation based on the Hill estimator obtained with
# 1-dimensional data simulated from an AR(1) with univariate Student-t
# distributed innovations

tsDist <- "studentT"
tsType <- "AR"

# parameter setting
corr <- 0.8
df <- 3
par <- c(corr, df)

# Big- small-blocks setting
bigBlock <- 65
smallBlock <- 15

# Number of larger order statistics
k <- 150

# sample size
ndata <- 2500

# Simulates a sample from an AR(1) model with Student-t innovations
data <- rtimeseries(ndata, tsDist, tsType, par)

# tail index estimation
gammaHat1 <- HTailIndex(data, k, TRUE, bigBlock=bigBlock, smallBlock=smallBlock)
gammaHat1$gammaHat
gammaHat1$CIgamHat

# tail index estimation with bias correction
gammaHat2 <- HTailIndex(data, 2*k, TRUE, bias=TRUE, bigBlock=bigBlock, smallBlock=smallBlock)
gammaHat2$gammaHat-gammaHat2$BiasGamHat
gammaHat2$CIgamHat
```

HypoTesting *Wald-Type Hypothesis Testing*

Description

Wald-type hypothesis tes for testing equality of high or extreme expectiles and quantiles

Usage

```
HypoTesting(data, tau, tau1=NULL, type="ExpectRisks", level="extreme",
            method="LAWS", bias=FALSE, k=NULL, alpha=0.05)
```

Arguments

data	A matrix of $(n \times d)$ observations.
tau	A real in $(0, 1)$ specifying the intermediate level τ_n. See **Details**.
tau1	A real in $(0, 1)$ specifying the extreme level τ'_n. See **Details**.
type	A string specifying the type of test. By default `type="ExpectRisks"` specifies the test for testing the equality of expectiles. See **Details**.
level	A string specifying the level of the expectile. This make sense when `type="ExpectRisks"`. By default `level="extreme"` specifies that the test concerns expectiles at the extreme level. See **Details**.
method	A string specifying the method used to estimate the expecile. By default `est="LAWS"` specifies the use of the LAWS based estimator. See **Details**.
bias	A logical value. By default `bias=FALSE` specifies that no bias correction is computed. See **Details**.
k	An integer specifying the value of the intermediate sequence k_n. See **Details**.
alpha	A real in $(0, 1)$ specifying the significance level of the test.

Details

With a dataset `data` of `d`-dimensional observations and sample size n, a Wald-type hypothesis testing is performed in order to check whether the is empirical evidence against the null hypothesis. The null hypothesis concerns the equality among the expectiles or quantiles or tail indices of the marginal distributions. The three tests depend on the depends on the estimation of the `d`-dimensional tail index γ. Here, γ is estimated using the Hill estimation (see MultiHTailIndex for details). The data are regarded as `d`-dimensional temporal independent observations coming from dependent variables. See Padoan and Stupfler (2020) for details.

- The so-called intermediate level `tau` or τ_n is a sequence of positive reals such that $\tau_n \to 1$ as $n \to \infty$. Practically, for each marginal distribution, $\tau_n \in (0, 1)$ is the ratio between *N* (Numerator) and *D* (Denominator). Where *N* is the empirical mean distance of the τ_n*-th* expectile from the observations smaller than it, and *D* is the empirical mean distance of τ_n*-th* expectile from all the observations.
- The so-called extreme level `tau1` or τ'_n is a sequence of positive reals such that $\tau'_n \to 1$ as $n \to \infty$. For each marginal distribution, the value $(1 - tau'_n) \in (0, 1)$ is meant to be a small tail probability such that $(1 - \tau'_n) = 1/n$ or $(1 - \tau'_n) < 1/n$. It is also assumed that $n(1 - \tau'_n) \to C$ as $n \to \infty$, where C is a positive finite constant. Typically, $C \in (0, 1)$ so it is expected that there are no observations in a data sample that are greater than the expectile at the extreme level τ'_n.
- When `type="ExpectRisks"`, the null hypothesis of the hypothesis testing concerns the equality among the expectiles of the marginal distributions. See Section 3.3 of Padoan and Stupfler (2020) for details. When `type="QuantRisks"`, the null hypothesis of the hypothesis testing concerns the equality among the quantiles of the marginal distributions. See Section 5 of Padoan and Stupfler (2020) for details. Note that in this case the test is based on the asymptotic distribution of normalized quantile estimator in the logarithmic scale. When `type="tails"`, the null hypothesis of the hypothesis testing concerns the equality among the tail indices of the marginal distributions. See Sections 3.2 and 3.3 of Padoan and Stupfler (2020) for details.

- When `type="ExpectRisks"`, the null hypothesis concerns the equality among the expectiles of the marginal distributions at the intermediate level and this is achieved through `level=="inter"`. In this case the test is obtained exploiting the asymptotic distribution of relative expectile appropriately normalised. See Section 2.1, 3.1 and 3.3 of Padoan and Stupfler (2020) for details. Instead, if `level=="extreme"` the null hypothesis concerns the equality among the expectiles of the marginal distributions at the extreme level.
- When `method='LAWS'`, then the τ_n'-*th* d-dimensional expectile is estimated using the LAWS based estimator. When `method='QB'`, the expectile is instead estimated using the QB esimtator. The definition of both estimators depend on the estimation of the `d`-dimensional tail index γ. The `d`-dimensional tail index γ is estimated using the `d`-dimensional Hill estimator (`tailest='Hill'`), see MultiHTailIndex). See Section 2.2 in Padoan and Stupfler (2020) for details.
- If `bias=TRUE` then `d`-dimensional γ is estimated using formula (4.2) of Haan et al. (2016). This is used by the LAWS and QB estimators. Furthermore, the τ_n'–*th* quantile is estimated using the formula in page 330 of de Haan et al. (2016). This provides a bias corrected version of the Weissman estimator. This is used by the QB estimator. However, in this case the asymptotic variance is not estimated using the formula in Haan et al. (2016) Theorem 4.2. Instead, for simplicity the asymptotic variance-covariance matrix is estimated by the formulas Section 3.2 of Padoan and Stupfler (2020).
- `k` or k_n is the value of the so-called intermediate sequence k_n, $n = 1, 2, \ldots$. Its represents a sequence of positive integers such that $k_n \to \infty$ and $k_n/n \to 0$ as $n \to \infty$. Practically, for each marginal distribution when `tau=NULL` and `method='LAWS'` or `method='QB'`, then $\tau_n = 1 - k_n/n$ is the intermediate level of the expectile to be stimated. When `tailest='Hill'`, for each marginal distributions, then k_n specifies the number of `k`$+1$ larger order statistics used in the definition of the Hill estimator.
- A small value $\alpha \in (0, 1)$ specifies the significance level of Wald-type hypothesis testing.

Value

A list with elements:

- `logLikR`: the observed value of log-likelihood ratio statistic test;
- `critVal`: the quantile (critical level) of a chi-square distribution with `d` degrees of freedom and confidence level α.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://www.unibocconi.it/en/faculty/simone-padoan; Gilles Stupfler, <gilles.stupfler@univ-angers.fr>, https://math.univ-angers.fr/~stupfler/

References

Simone A. Padoan and Gilles Stupfler (2022). Joint inference on extreme expectiles for multivariate heavy-tailed distributions, *Bernoulli* **28**(2), 1021-1048.

See Also

MultiHTailIndex, predMultiExpectiles, extMultiQuantile

Examples

```
# Hypothesis testing on the equality extreme expectiles based on a sample of
# d-dimensional observations simulated from a joint distribution with
# a Gumbel copula and equal Frechet marginal distributions.
library(plot3D)
library(copula)
library(evd)

# distributional setting
copula <- "Gumbel"
dist <- "Frechet"

# parameter setting
dep <- 3
dim <- 3
scale <- rep(1, dim)
shape <- rep(3, dim)
par <- list(dep=dep, scale=scale, shape=shape, dim=dim)

# Intermediate level (or sample tail probability 1-tau)
tau <- 0.95
# Extreme level (or tail probability 1-tau1 of unobserved expectile)
tau1 <- 0.9995

# sample size
ndata <- 1000

# Simulates a sample from a multivariate distribution with equal Frechet
# marginals distributions and a Gumbel copula
data <- rmdata(ndata, dist, copula, par)
scatter3D(data[,1], data[,2], data[,3])

# Performs Wald-type hypothesis testing
HypoTesting(data, tau, tau1)

# Hypothesis testing on the equality extreme expectiles based on a sample of
# d-dimensional observations simulated from a joint distribution with
# a Clayton copula and different Frechet marginal distributions.

# distributional setting
copula <- "Clayton"
dist <- "Frechet"

# parameter setting
dim <- 3
dep <- 2
scale <- rep(1, dim)
shape <- c(2.1, 3, 4.5)
par <- list(dep=dep, scale=scale, shape=shape, dim=dim)

# Intermediate level (or sample tail probability 1-tau)
tau <- 0.95
```

```
# Extreme level (or tail probability 1-tau1 of unobserved expectile)
tau1 <- 0.9995

# sample size
ndata <- 1000

# Simulates a sample from a multivariate distribution with equal Frechet
# marginals distributions and a Gumbel copula
data <- rmdata(ndata, dist, copula, par)
scatter3D(data[,1], data[,2], data[,3])

# Performs Wald-type hypothesis testing
HypoTesting(data, tau, tau1)
```

MLTailIndex *Maximum Likelihood Tail Index Estimation*

Description

Computes a point and interval estimate of the tail index based on the Maximum Likelihood (ML) estimator.

Usage

```
MLTailIndex(data, k, var=FALSE, varType="asym-Dep", bigBlock=NULL,
            smallBlock=NULL, alpha=0.05)
```

Arguments

`data`	A vector of $(1 \times n)$ observations.
`k`	An integer specifying the value of the intermediate sequence k_n. See **Details**.
`var`	If `var=TRUE` then an estimate of the asymptotic variance of the tail index estimator is computed.
`varType`	A string specifying the asymptotic variance to compute. By default `varType="asym-Dep"` specifies the variance estimator for serial dependent observations. See **Details**.
`bigBlock`	An interger specifying the size of the big-block used to estimaste the asymptotic variance. See **Details**.
`smallBlock`	An interger specifying the size of the small-block used to estimaste the asymptotic variance. See **Details**.
`alpha`	A real in $(0, 1)$ specifying the confidence level $(1-\alpha)100\%$ of the approximate confidence interval for the tail index.

Details

For a dataset `data` of sample size n, the tail index γ of its (marginal) distribution is computed by applying the ML estimator. The observations can be either independent or temporal dependent.

- `k` or k_n is the value of the so-called intermediate sequence k_n, $n = 1, 2, \ldots$. Its represents a sequence of positive integers such that $k_n \to \infty$ and $k_n/n \to 0$ as $n \to \infty$. Practically, the value k_n specifies the numer of `k`+1 larger order statistics to be used to estimate γ.
- If `var=TRUE` then the asymptotic variance of the Hill estimator is computed. With independent observations the asymptotic variance is estimated by the formula in Theorem 3.4.2 of de Haan and Ferreira (2006). This is achieved through `varType="asym-Ind"`. With serial dependent observations the asymptotic variance is estimated by the formula in 1288 in Drees (2000). This is achieved through `varType="asym-Dep"`. In this latter case the serial dependence is estimated by exploiting the "big blocks seperated by small blocks" techinque which is a standard tools in time series, see Leadbetter et al. (1986). See also formula (11) in Drees (2003). The size of the big and small blocks are specified by the parameters `bigBlock` and `smallBlock`, respectively.
- Given a small value $\alpha \in (0, 1)$ then an asymptotic confidence interval for the tail index, with approximate nominal confidence level $(1 - \alpha)100\%$ is computed.

Value

A list with elements:

- `gammaHat`: an estimate of tail index γ;
- `VarGamHat`: an estimate of the variance of the ML estimator;
- `CIgamHat`: an estimate of the approximate $(1 - \alpha)100\%$ confidence interval for γ.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://www.unibocconi.it/en/faculty/simone-padoan; Gilles Stupfler, <gilles.stupfler@univ-angers.fr>, https://math.univ-angers.fr/~stupfler/

References

Anthony C. Davison, Simone A. Padoan and Gilles Stupfler (2023). Tail Risk Inference via Expectiles in Heavy-Tailed Time Series, *Journal of Business & Economic Statistics*, **41**(3) 876-889.

Drees, H. (2000). Weighted approximations of tail processes for β-mixing random variables. *Annals of Applied Probability*, **10**, 1274-1301.

de Haan, L. and Ferreira, A. (2006). Extreme Value Theory: An Introduction. *Springer-Verlag*, New York.

Leadbetter, M.R., Lindgren, G. and Rootzen, H. (1989). Extremes and related properties of random sequences and processes. *Springer*.

See Also

HTailIndex, MomTailIndex, EBTailIndex

Examples

```
# Tail index estimation based on the Maximum Likelihood estimator obtained with
# 1-dimensional data simulated from an AR(1) with univariate Student-t
# distributed innovations

tsDist <- "studentT"
tsType <- "AR"

# parameter setting
corr <- 0.8
df <- 3
par <- c(corr, df)

# Big- small-blocks setting
bigBlock <- 65
smallBlock <- 15

# Number of larger order statistics
k <- 150

# sample size
ndata <- 2500

# Simulates a sample from an AR(1) model with Student-t innovations
data <- rtimeseries(ndata, tsDist, tsType, par)

# tail index estimation
gammaHat <- MLTailIndex(data, k, TRUE, bigBlock=bigBlock, smallBlock=smallBlock)
gammaHat$gammaHat
gammaHat$CIgamHat
```

MomTailIndex *Moment based Tail Index Estimation*

Description

Computes a point estimate of the tail index based on the Moment Based (MB) estimator.

Usage

```
MomTailIndex(data, k)
```

Arguments

data	A vector of $(1 \times n)$ observations.
k	An integer specifying the value of the intermediate sequence k_n. See **Details**.

Details

For a dataset data of sample size n, the tail index γ of its (marginal) distribution is computed by applying the MB estimator. The observations can be either independent or temporal dependent. For details see de Haan and Ferreira (2006).

- k or k_n is the value of the so-called intermediate sequence k_n, $n = 1, 2, \ldots$. Its represents a sequence of positive integers such that $k_n \to \infty$ and $k_n/n \to 0$ as $n \to \infty$. Practically, the value k_n specifies the number of k+1 larger order statistics to be used to estimate γ.

Value

An estimate of the tail index γ.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://www.unibocconi.it/en/faculty/simone-padoan; Gilles Stupfler, <gilles.stupfler@univ-angers.fr>, https://math.univ-angers.fr/~stupfler/

References

de Haan, L. and Ferreira, A. (2006). Extreme Value Theory: An Introduction. *Springer-Verlag*, New York.

See Also

HTailIndex, MLTailIndex, EBTailIndex

Examples

```
# Tail index estimation based on the Moment estimator obtained with
# 1-dimensional data simulated from an AR(1) with univariate Student-t
# distributed innovations

tsDist <- "studentT"
tsType <- "AR"

# parameter setting
corr <- 0.8
df <- 3
par <- c(corr, df)

# Big- small-blocks setting
bigBlock <- 65
smallblock <- 15

# Number of larger order statistics
k <- 150

# sample size
ndata <- 2500
```

```
# Simulates a sample from an AR(1) model with Student-t innovations
data <- rtimeseries(ndata, tsDist, tsType, par)

# tail index estimation
gammaHat <- MomTailIndex(data, k)
gammaHat
```

`MultiHTailIndex` *Multidimensional Hill Tail Index Estimation*

Description

Computes point estimates and $(1-\alpha)100\%$ confidence regions estimate of d-dimensional tail indices based on the Hill's estimator.

Usage

```
MultiHTailIndex(data, k, var=FALSE, varType="asym-Dep", bias=FALSE,
                alpha=0.05, plot=FALSE)
```

Arguments

`data`	A matrix of $(n \times d)$ observations.
`k`	An integer specifying the value of the intermediate sequence k_n. See **Details**.
`var`	If `var=TRUE` then an estimate of the variance-covariance matrix of the tail indices estimators is computed.
`varType`	A string specifying the asymptotic variance to compute. By default `varType="asym-Dep"` specifies the variance estimator for d dependent marginal variables. See **Details**.
`bias`	A logical value. By default `biast=FALSE` specifies that no bias correction is computed. See **Details**.
`alpha`	A real in $(0,1)$ specifying the confidence level $(1-\alpha)100\%$ of the approximate confidence interval for the tail index.
`plot`	A logical value. By default `plot=FALSE` specifies that no graphical representation of the estimates is provided. See **Details**.

Details

For a dataset `data` of $(n \times d)$ observations, where d is the number of variables and n is the sample size, the tail index γ of the d marginal distributions is estimated by applying the Hill estimator. Together with a point estimate a $(1-\alpha)100\%$ confidence region is computed. The data are regarded as `d`-dimensional temporal independent observations coming from dependent variables.

- `k` or k_n is the value of the so-called intermediate sequence k_n, $n = 1, 2, \ldots$. Its represents a sequence of positive integers such that $k_n \to \infty$ and $k_n/n \to 0$ as $n \to \infty$. Practically, the value k_n specifies the number of `k`+1 larger order statistics to be used to estimate each marginal tail index γ_j for $j = 1, \ldots, d$.

- If `var=TRUE` then an estimate of the asymptotic variance-covariance matrix of the multivariate Hill estimator is computed. With independent observations the asymptotic variance-covariance matrix is estimated by the matrix $\hat{\Sigma}^{LAWS}_{j,\ell}(\gamma, R)(1, 1)$, see bottom formula in page 14 of Padoan and Stupfler (2020). This is achieved through `varType="asym-Dep"` which means d dependent marginal variables. When `varType="asym-Ind"` d marginal variables are regarded as independent and the returned variance-covariance matrix $\hat{\Sigma}^{LAWS}_{j,\ell}(\gamma, R)(1, 1)$ is a diagonal matrix with only variance terms.
- If `bias=TRUE` then an estimate of the bias term of the Hill estimator is computed implementing using formula (4.2) in de Haan et al. (2016). In this case the asymptotic variance is not estimated using the formula in Haan et al. (2016) Theorem 4.1 but instead for simplicity the formula at the bottom of page 14 in Padoan and Stupfler (2020) is still used.
- Given a small value $\alpha \in (0, 1)$ then an estimate of an asymptotic confidence region for γ_j, for $j = 1, \ldots, d$, with approximate nominal confidence level $(1 - \alpha)100\%$, is computed. The confidence intervals are computed exploiting the asymptotic normality of multivariate Hill estimator appropriately normalized (the logarithmic scale is not used), see Padoan and Stupfler (2020) for details.
- If `plot=TRUE` then a graphical representation of the estimates is not provided.

Value

A list with elements:

- `gammaHat`: an estimate of the d tail indices γ_j, for $j = 1, \ldots, d$;
- `VarCovGHat`: an estimate of the asymptotic variance-covariance matrix of the multivariate Hill estimator;
- `biasTerm`: an estimate of bias term of the multivariate Hill estimator;
- `EstConReg`: an estimate of the $(1 - \alpha)100\%$ confidence region.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://www.unibocconi.it/en/faculty/simone-padoan; Gilles Stupfler, <gilles.stupfler@univ-angers.fr>, https://math.univ-angers.fr/~stupfler/

References

Simone A. Padoan and Gilles Stupfler (2022). Joint inference on extreme expectiles for multivariate heavy-tailed distributions, *Bernoulli* **28**(2), 1021-1048.

de Haan, L., Mercadier, C. and Zhou, C. (2016). Adapting extreme value statistics to financial time series: dealing with bias and serial dependence. *Finance and Stochastics*, **20**, 321-354.

de Haan, L. and Ferreira, A. (2006). Extreme Value Theory: An Introduction. *Springer-Verlag*, New York.

See Also

HTailIndex, rmdata

Examples

```
# Tail index estimation based on the multivariate Hill estimator obtained with
# n observations simulated from a d-dimensional random vector with a multivariate
# distribution with equal Frechet margins and a Clayton copula.
library(plot3D)
library(copula)
library(evd)

# distributional setting
copula <- "Clayton"
dist <- "Frechet"

# parameter setting
dep <- 3
dim <- 3
scale <- rep(1, dim)
shape <- rep(3, dim)
par <- list(dep=dep, scale=scale, shape=shape, dim=dim)

# Number of larger order statistics
k <- 150

# sample size
ndata <- 1000

# Simulates a sample from a multivariate distribution with equal Frechet
# marginals distributions and a Clayton copula
data <- rmdata(ndata, dist, copula, par)
scatter3D(data[,1], data[,2], data[,3])

# tail indices estimation
est <- MultiHTailIndex(data, k, TRUE)
est$gammaHat
est$VarCovGHat
# run the following command to see the graphical representation
## Not run:
 est <- MultiHTailIndex(data, k, TRUE, plot=TRUE)

## End(Not run)
```

---

plotBayes — *Plot empirical Bayes inference results for continuous and discrete generalized Pareto distribution*

Description

Given a sample of posterior draws of the scale or shape parameter, return a histogram on the density scale for the posterior distribution along with a theoretical prior and posterior comparison curve based on the MLE, pointwise Wald-based normal 95% confidence intervals for the mean of the sample, and pointwise credible intervals (asymmetric by design).

Usage

```
plotBayes(
  x,
  mle,
  alpha = 0.05,
  param = c("scale", "shape"),
  cols = c("mediumseagreen", "goldenrod", "gold4"),
  ...
)
```

Arguments

`x`	a vector of posterior samples
`mle`	vector of length 2 containing the maximum likelihood estimator for the scale and shape parameters, respectively (only used if `param="scale"`)
`alpha`	level for intervals. Default to 0.05 giving 95% confidence intervals
`param`	character string indicating the parameter. Default: `c("scale", "shape")`
`cols`	vector of length three containing colors for posterior mean, confidence intervals, and credible intervals. Default to `c("mediumseagreen", "goldenrod", "gold4")`
`...`	additional arguments for plotting function; only `xlab` is allowed

Value

`NULL`; used to create a plot

Examples

```
## Not run:
# generate data
set.seed(1234)
n <- 500
samp <- evd::rfrechet(n, 0, 3, 4)
# set effective sample size and threshold
k <- 50
threshold <- sort(samp, decreasing = TRUE)[k+1]
# preliminary mle estimates of scale and shape parameters
mlest <- evd::fpot(samp, threshold)
# empirical bayes procedure
proc <- estPOT(
 samp,
 k = k,
 pn = c(0.01, 0.005),
 type = "continuous",
 method = "bayesian",
 prior = "empirical",
 start = as.list(mlest$estimate),
 sig0 = 0.1)
plotBayes(
```

```
    proc$post_sample[,1],
    mlest$estimate,
    param = "scale")
  plotBayes(
    proc$post_sample[,2],
    param = "shape")

## End(Not run)
```

predDens *Predictive posterior density of peak-over-threshold models*

Description

Given posterior samples for the parameters of the continuous or discrete generalized Pareto distribution, return the predictive posterior density of a peak above an intermediate or extreme threshold using the threshold stability property.

Usage

```
predDens(
  x,
  postsamp,
  threshold,
  type = c("continuous", "discrete"),
  extrapolation = FALSE,
  p,
  k,
  n
)
```

Arguments

x	vector of length r containing the points at which to evaluate the density
postsamp	a m by 2 matrix with columns containing the posterior samples of scale and shape parameters of the generalized Pareto distribution, respectively
threshold	threshold for the generalized Pareto model, corresponding to the $n-k$th order statistic of the sample
type	data type, either "continuous" or "discrete" for count data.
extrapolation	logical; if TRUE, extrapolate using the threshold stability property
p	scalar tail probability for the extrapolation. Must be smaller than k/n (only used if extrapolation=TRUE)
k	integer, number of exceedances for the generalized Pareto (only used if extrapolation=TRUE)
n	integer, number of observations in the full sample. Must be greater than k (only used if extrapolation=TRUE)

Value

a vector of length r of posterior predictive density values associated to x

Examples

```
## Not run:
# generate data
set.seed(1234)
n <- 500
samp <- evd::rfrechet(n,0,3,4)
# set effective sample size and threshold
k <- 50
threshold <- sort(samp,decreasing = TRUE)[k+1]
# preliminary mle estimates of scale and shape parameters
mlest <- evd::fpot(samp, threshold)
# empirical bayes procedure
proc <- estPOT(
  samp,
  k = k,
  pn = c(0.01, 0.005),
  type = "continuous",
  method = "bayesian",
  prior = "empirical",
  start = as.list(mlest$estimate),
  sig0 = 0.1)
# predictive density estimation
yg <- seq(0, 50, by = 2)
nyg <- length(yg)
predDens_int <- predDens(
  yg,
  proc$post_sample,
  proc$t,
  "continuous",
  extrapolation = FALSE)
predDens_ext <- predDens(
  yg,
  proc$post_sample,
  proc$t,
  "continuous",
  extrapolation = TRUE,
  p = 0.001,
  k = k,
  n = n)
# plot
plot(
 x = yg,
 y = predDens_int,
 type = "l",
 lwd = 2,
 col = "dodgerblue",
 ylab = "",
 main = "Predictive posterior density")
```

```
lines(
 x = yg,
 y = predDens_ext,
 lwd = 2,
 col = "violet")
legend(
  "topright",
  legend = c("Intermediate threshold", "Extreme threshold"),
  lwd = 2,
  col = c("dodgerblue", "violet"))

## End(Not run)
```

predDensx   *Conditional predictive posterior density of peaks-over-threshold models*

Description

Given posterior samples for the parameters of the continuous or discrete generalized Pareto distribution and scedasis function for a set of covariates, evaluated at every draw of the latter, return the predictive posterior density of a peak above an intermediate or extreme threshold using the threshold stability property.

Usage

```
predDensx(
  x,
  postsamp,
  scedasis,
  threshold,
  type = c("continuous", "discrete"),
  extrapolation = FALSE,
  p,
  k,
  n
)
```

Arguments

`x`	vector of length `r` containing the points at which to evaluate the density
`postsamp`	a `m` by `2` matrix with columns containing the posterior samples of scale and shape parameters of the generalized Pareto distribution, respectively
`scedasis`	an `m` by `p` matrix, obtained by evaluating the scedasis function for every of the `p` covariate values and every `m` posterior draw
`threshold`	threshold for the generalized Pareto model, corresponding to the $n-k$th order statistic of the sample

type	data type, either "continuous" or "discrete" for count data.
extrapolation	logical; if TRUE, extrapolate using the threshold stability property
p	scalar tail probability for the extrapolation. Must be smaller than k/n (only used if extrapolation=TRUE)
k	integer, number of exceedances for the generalized Pareto (only used if extrapolation=TRUE)
n	integer, number of observations in the full sample. Must be greater than k (only used if extrapolation=TRUE)

Value

a list with components

- x the vector at which the posterior density is evaluated
- preddens an r by p matrix of predictive density corresponding to each combination of x and scedasis value

Examples

```
## Not run:
# generate data
set.seed(1234)
n <- 500
samp <- evd::rfrechet(n,0,1:n,4)
# set effective sample size and threshold
k <- 50
threshold <- sort(samp,decreasing = TRUE)[k+1]
# preliminary mle estimates of scale and shape parameters
mlest <- evd::fpot(samp, threshold, control=list(maxit = 500))
# empirical bayes procedure
proc <- estPOT(
  samp,
  k = k,
  pn = c(0.01, 0.005),
  type = "continuous",
  method = "bayesian",
  prior = "empirical",
  start = as.list(mlest$estimate),
  sig0 = 0.1)
# conditional predictive density estimation
yg <- seq(0, 50, by = 2)
nyg <- length(yg)
# estimation of scedasis function
# setting
M <- 1e3
C <- 5
alpha <- 0.05
bw <- .5
nsim <- 5000
burn <- 1000
# create covariate
# in sample obs
```

```
n_in = n
# number of years ahead
nY = 1
n_out = 365 * nY
# total obs
n_tot = n_in + n_out
# total covariate (in+out sample period)
x <- seq(0, 1, length = n_tot)
# in sample grid dimension for covariate
ng_in <- 150
xg <- seq(0, x[n_in], length = ng_in)
# in+out of sample grid
xg <- c(xg, seq(x[n_in + 1], x[(n_tot)], length = ng_in))
# in+out sample grid dimension
nxg <- length(xg)
xg <- array(xg, c(nxg, 1))
# in sample observations
samp_in <- samp[1:n_in]
ssamp_in <- sort(samp_in, decreasing = TRUE, index = TRUE)
x_in <- x[1:n_in] # in sample covariate
xs <- x_in[ssamp_in$ix[1:k]] # in sample concomitant covariate
# estimate scedasis function over the in and out of sample period
res_stat <- apply(
  xg,
  1,
  cpost_stat,
  N = nsim - burn,
  x = x_in,
  xs = xs,
  bw = bw,
  k = k,
  C = C
)
# conditional predictive posterior density
yg <- seq(500, 5000, by = 100)
nyg = length(yg)
# intermediate threshold
predDens_intx <- predDensx(
  x = yg,
  postsamp = proc$post_sample,
  scedasis = res_stat,
  threshold = proc$t,
  "continuous",
  extrapolation = FALSE)
# extreme threshold
predDens_extx <- predDensx(
  x = yg,
  postsamp = proc$post_sample,
  scedasis = res_stat,
  threshold = proc$t,
  "continuous",
  extrapolation = TRUE,
  p = 0.001,
```

```
  k = k,
  n = n)
# plot intermediate and extreme density conditional on a specific value of scedasis function
# disclaimer: to speed up the procedure, we specify a coarse grid
plot(
 x = predDens_intx$x,
 y = predDens_intx$preddens[,20],
 type = "l",
 lwd = 2,
 col="dodgerblue",
 ylab = "",
 xlab = "yg",
 main = "Conditional predictive posterior density")
lines(
  x = predDens_extx$x,
  y = predDens_extx$preddens[,20],
  lwd = 2,
  col = "violet")
legend("topright",
  legend = c("Intermediate threshold","Extreme threshold"),
  lwd = 2,
  col = c("dodgerblue", "violet"))

## End(Not run)
```

---

predExpectiles *Extreme Expectile Estimation*

Description

Computes a point and interval estimate of the expectile at the extreme level (Expectile Prediction).

Usage

```
predExpectiles(data, tau, tau1, method="LAWS", tailest="Hill", var=FALSE,
                varType="asym-Dep", bias=FALSE, bigBlock=NULL, smallBlock=NULL,
                k=NULL, alpha_n=NULL, alpha=0.05)
```

Arguments

`data`	A vector of $(1 \times n)$ observations.
`tau`	A real in $(0, 1)$ specifying the intermediate level τ_n. See **Details**.
`tau1`	A real in $(0, 1)$ specifying the extreme level τ'_n. See **Details**.
`method`	A string specifying the method used to estimate the expecile. By default `est="LAWS"` specifies the use of the LAWS based estimator. See **Details**.
`tailest`	A string specifying the tail index estimator. By default `tailest="Hill"` specifies the use of Hill estimator. See **Details**.

var	If `var=TRUE` then an estimate of the asymptotic variance of the expectile estimator is computed.
varType	A string specifying the type of asymptotic variance to compute. By default `varType="asym-Dep"` specifies the variance estimator for serial dependent observations. See **Details**.
bias	A logical value. By default `bias=FALSE` specifies that no bias correction is computed. See **Details**.
bigBlock	An interger specifying the size of the big-block used to estimaste the asymptotic variance. See **Details**.
smallBlock	An interger specifying the size of the small-block used to estimaste the asymptotic variance. See **Details**.
k	An integer specifying the value of the intermediate sequence k_n. See **Details**.
alpha_n	A real in $(0, 1)$ specifying the quantile's extreme level to be use in order to estimate the expectile's extreme level.
alpha	A real in $(0, 1)$ specifying the confidence level $(1-\alpha)100\%$ of the approximate confidence interval for the expecile at the intermedite level.

Details

For a dataset `data` of sample size n, an estimate of the τ'_n*-th* expectile is computed. The estimation of the expectile at the extreme level `tau1` (τ'_n) is meant to be a prediction beyond the observed sample. Two estimators are available: the so-called Least Asymmetrically Weighted Squares (LAWS) based estimator and the Quantile-Based (QB) estimator. The definition of both estimators depends on the estimation of the tail index γ. Here, γ is estimated using the Hill estimation (see HTailIndex for details) or in alternative using the the expectile based estimator (see EBTailIndex). The observations can be either independent or temporal dependent. See Section 3.2 in Padoan and Stupfler (2020) for details.

- The so-called intermediate level `tau` or τ_n is a sequence of positive reals such that $\tau_n \to 1$ as $n \to \infty$. Practically, $\tau_n \in (0, 1)$ is the ratio between *N* (Numerator) and *D* (Denominator). Where *N* is the empirical mean distance of the τ_n*-th* expectile from the observations smaller than it, and *D* is the empirical mean distance of τ_n*-th* expectile from all the observations.
- The so-called extreme level `tau1` or τ'_n is a sequence of positive reals such that $\tau'_n \to 1$ as $n \to \infty$. The value $(1 - tau'_n) \in (0, 1)$ is meant to be a small tail probability such that $(1 - \tau'_n) = 1/n$ or $(1 - \tau'_n) < 1/n$. It is also assumed that $n(1 - \tau'_n) \to C$ as $n \to \infty$, where C is a positive finite constant. Typically, $C \in (0, 1)$ so it is expected that there are no observations in a data sample that are greater than the expectile at the extreme level τ'_n.
- When `method='LAWS'`, then the τ'_n*-th* expectile is estimated using the LAWS based estimator. When `method='QB'`, the expectile is instead estimated using the QB esimtator. The definition of both estimators depend on the estimation of the tail index γ. When `tailest='Hill'` then γ is estimated using the Hill estimator (see HTailIndex). When `tailest='ExpBased'`, then γ is estimated using the expectile based estimator (see EBTailIndex). See Section 3.2 in Padoan and Stupfler (2020) for details.
- If `var=TRUE` then an esitmate of the asymptotic variance of the tau'_n*-th* expectile is computed. Notice that the estimation of the asymptotic variance **is only available** when γ is estimated using the Hill estimator (see HTailIndex). With independent observations the asymptotic variance is estimated by $\hat{\gamma}^2$, see the remark below Theorem 3.5 in Padoan and Stupfler

(2020). This is achieved through `varType="asym-Ind"`. With serial dependent observations the asymptotic variance is estimated by the formula in Throrem 3.5 of Padoan and Stupfler (2020). This is achieved through `varType="asym-Dep"`. See Section 3.2 in Padoan and Stupfler (2020) for details. In this latter case the computation of the serial dependence is based on the "big blocks seperated by small blocks" techinque which is a standard tools in time series, see e.g. Leadbetter et al. (1986). The size of the big and small blocks are specified by the parameters `bigBlock` and `smallBlock`, respectively.

- If `bias=TRUE` then γ is estimated using formula (4.2) of Haan et al. (2016). This is used by the LAWS and QB estimators. Furthermore, the τ_n'*–th* quantile is estimated using the formula in page 330 of de Haan et al. (2016). This provides a bias corrected version of the Weissman estimator. This is used by the QB estimator. However, in this case the asymptotic variance is not estimated using the formula in Haan et al. (2016) Theorem 4.2. Instead, for simplicity the asymptotic variance is estimated by the formula in Corollary 3.8, with serial dependent observations, and $\hat{\gamma}^2$ with independent observation (see e.g. de Drees 2000, for the details).
- `k` or k_n is the value of the so-called intermediate sequence k_n, $n = 1, 2, \ldots$. Its represents a sequence of positive integers such that $k_n \to \infty$ and $k_n/n \to 0$ as $n \to \infty$. Practically, when `tau=NULL` and `method='LAWS'`, then $\tau_n = 1 - k_n/n$ is the intermediate level of the expectile to be stimated. The latter is also used to estimate the tail index when `tailest='ExpBased'`. Instead, if `tailest='Hill'`, then k_n specifies the number of `k`+1 larger order statistics used in the definition of the Hill estimator. Differently, When `tau=NULL` and `method='QB'`, then $\tau_n = 1 - k_n/n$ is the intermediate level of the quantile to be stimated and of the expectile to be stimated when `tailest='ExpBased'`. Instead, when `tailest='Hill'` it is the numer of `k`+1 larger order statistics used in the definition of the Hill estimator.
- If quantile's extreme level is provided by `alpha_n`, then expectile's extreme level $tau_n'(\alpha_n)$ is replaced by $tau_n'(\alpha_n)$ which is esitmated using the method described in Section 6 of Padoan and Stupfler (2020). See estExtLevel for details.
- Given a small value $\alpha \in (0, 1)$ then an estimate of an asymptotic confidence interval for tau_n'-*th* expectile, with approximate nominal confidence level $(1 - \alpha)100\%$, is computed. The confidence intervals are computed exploiting formula (10) and (11) in Padoan and Stupfler (2020) and (46) in Drees (2003). See Section 5 in Padoan and Stupfler (2020) for details. When `biast=TRUE` confidence intervals are computed in the same way but after correcting the tail index estimate by an estimate of the bias term, see formula (4.2) in de Haan et al. (2016) for details.

Value

A list with elements:

- `EExpcHat`: an estimate of the τ_n'-*th* expecile;
- `VarExtHat`: an estimate of the asymptotic variance of the expectile estimator;
- `CIExpct`: an estimate of the approximate $(1 - \alpha)100\%$ confidence interval for τ_n'-*th* expecile.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://www.unibocconi.it/en/faculty/simone-padoan; Gilles Stupfler, <gilles.stupfler@univ-angers.fr>, https://math.univ-angers.fr/~stupfler/

References

Anthony C. Davison, Simone A. Padoan and Gilles Stupfler (2023). Tail Risk Inference via Expectiles in Heavy-Tailed Time Series, *Journal of Business & Economic Statistics*, **41**(3) 876-889.

Daouia, A., Girard, S. and Stupfler, G. (2018). Estimation of tail risk based on extreme expectiles. *Journal of the Royal Statistical Society: Series B*, **80**, 263-292.

de Haan, L., Mercadier, C. and Zhou, C. (2016). Adapting extreme value statistics to financial time series: dealing with bias and serial dependence. *Finance and Stochastics*, **20**, 321-354.

Drees, H. (2003). Extreme quantile estimation for dependent data, with applications to finance. *Bernoulli*, **9**, 617-657.

Drees, H. (2000). Weighted approximations of tail processes for β-mixing random variables. *Annals of Applied Probability*, **10**, 1274-1301.

Leadbetter, M.R., Lindgren, G. and Rootzen, H. (1989). Extremes and related properties of random sequences and processes. *Springer*.

See Also

HTailIndex, EBTailIndex, estExpectiles, extQuantile

Examples

```
# Extreme expectile estimation at the extreme level tau1 obtained with
# 1-dimensional data simulated from an AR(1) with univariate
# Student-t distributed innovations

tsDist <- "studentT"
tsType <- "AR"

# parameter setting
corr <- 0.8
df <- 3
par <- c(corr, df)

# Big- small-blocks setting
bigBlock <- 65
smallBlock <- 15

# Intermediate level (or sample tail probability 1-tau)
tau <- 0.95
# Extreme level (or tail probability 1-tau1 of unobserved expectile)
tau1 <- 0.9995

# sample size
ndata <- 2500

# Simulates a sample from an AR(1) model with Student-t innovations
data <- rtimeseries(ndata, tsDist, tsType, par)

# Extreme expectile estimation
expectHat1 <- predExpectiles(data, tau, tau1, var=TRUE, bigBlock=bigBlock,
```

```
                        smallBlock=smallBlock)
expectHat1$EExpcHat
expectHat1$CIExpct
# Extreme expectile estimation with bias correction
tau <- 0.80
expectHat2 <- predExpectiles(data, tau, tau1, "QB", var=TRUE, bias=TRUE, bigBlock=bigBlock,
 smallBlock=smallBlock)
expectHat2$EExpcHat
expectHat2$CIExpct
```

`predMultiExpectiles` *Multidimensional Extreme Expectile Estimation*

Description

Computes point estimates and $(1-\alpha)100\%$ confidence regions for d-dimensional expectile at the extreme level (Expectile Prediction).

Usage

```
predMultiExpectiles(data, tau, tau1, method="LAWS", tailest="Hill", var=FALSE,
                    varType="asym-Ind-Adj-Log", bias=FALSE, k=NULL, alpha=0.05,
                    plot=FALSE)
```

Arguments

`data`	A matrix of $(n \times d)$ observations.
`tau`	A real in $(0, 1)$ specifying the intermediate level τ_n. See **Details**.
`tau1`	A real in $(0, 1)$ specifying the extreme level τ_n'. See **Details**.
`method`	A string specifying the method used to estimate the expecile. By default `est="LAWS"` specifies the use of the LAWS based estimator. See **Details**.
`tailest`	A string specifying the tail index estimator. By default `tailest="Hill"` specifies the use of Hill estimator. See **Details**.
`var`	If `var=TRUE` then an estimate of the asymptotic variance of the expectile estimator is computed.
`varType`	A string specifying the type of asymptotic variance-covariance matrix to compute. By default `varType="asym-Ind-Adj-Log"` specifies that the variance-covariance matrix is computed assuming dependent variables and exploiting the log scale and a suitable adjustment. See **Details**.
`bias`	A logical value. By default `bias=FALSE` specifies that no bias correction is computed. See **Details**.
`k`	An integer specifying the value of the intermediate sequence k_n. See **Details**.
`alpha`	A real in $(0, 1)$ specifying the confidence level $(1-\alpha)100\%$ of the approximate confidence region for the d-dimensional expecile at the extreme level.
`plot`	A logical value. By default `plot=FALSE` specifies that no graphical representation of the estimates is provided. See **Details**.

Details

For a dataset data of d-dimensional observations and sample size n, an estimate of the τ_n'*-th* d-dimensional expectile is computed. The estimation of the d-dimensional expectile at the extreme level tau1 (τ_n') is meant to be a prediction beyond the observed sample. Two estimators are available: the so-called Least Asymmetrically Weighted Squares (LAWS) based estimator and the Quantile-Based (QB) estimator. The definition of both estimators depends on the estimation of the d-dimensional tail index γ. Here, γ is estimated using the Hill estimation (see MultiHTailIndex for details). The data are regarded as d-dimensional temporal independent observations coming from dependent variables. See Padoan and Stupfler (2020) for details.

- The so-called intermediate level tau or τ_n is a sequence of positive reals such that $\tau_n \to 1$ as $n \to \infty$. Practically, for each marginal distribution, $\tau_n \in (0,1)$ is the ratio between *N* (Numerator) and *D* (Denominator). Where *N* is the empirical mean distance of the τ_n*-th* expectile from the observations smaller than it, and *D* is the empirical mean distance of τ_n*-th* expectile from all the observations.
- The so-called extreme level tau1 or τ_n' is a sequence of positive reals such that $\tau_n' \to 1$ as $n \to \infty$. For each marginal distribution, the value $(1 - tau_n') \in (0,1)$ is meant to be a small tail probability such that $(1-\tau_n') = 1/n$ or $(1-\tau_n') < 1/n$. It is also assumed that $n(1-\tau_n') \to C$ as $n \to \infty$, where C is a positive finite constant. Typically, $C \in (0,1)$ so it is expected that there are no observations in a data sample that are greater than the expectile at the extreme level τ_n'.
- When method='LAWS', then the τ_n'*-th* d-dimensional expectile is estimated using the LAWS based estimator. When method='QB', the expectile is instead estimated using the QB esimtator. The definition of both estimators depend on the estimation of the d-dimensional tail index γ. The d-dimensional tail index γ is estimated using the d-dimensional Hill estimator (tailest='Hill'), see MultiHTailIndex). This is the only available option so far (soon more results will be available). See Section 2.2 in Padoan and Stupfler (2020) for details.
- If var=TRUE then an estimate of the asymptotic variance-covariance matrix of the tau_n'*-th* d-dimensional expectile is computed. Notice that the estimation of the asymptotic variance-covariance matrix **is only available** when γ is estimated using the Hill estimator (see MultiHTailIndex). The data are regarded as temporal independent observations coming from dependent variables. The asymptotic variance-covariance matrix is estimated exploiting the formulas in Section 3.2 of Padoan and Stupfler (2020). The variance-covariance matrix is computed exploiting the asymptotic behaviour of the normalized expectile estimator which is expressed in logarithmic scale. In addition, a suitable adjustment is considered. This is achieved through varType="asym-Ind-Adj-Log". The data can also be regarded as d-dimensional temporal independent observations coming from independent variables. In this case the asymptotic variance-covariance matrix is diagonal and is also computed exploiting the formulas in Section 3.2 of Padoan and Stupfler (2020). This is achieved through varType="asym-Ind-Log". If varType="asym-Ind-Adj", then the variance-covariance matrix is computed exploiting the asymptotic behaviour of the relative expectile estimator appropriately normalized and exploiting a suitable adjustment. This concerns the case of dependent variables. The case of independent variables is achieved through varType="asym-Ind".
- If bias=TRUE then d-dimensional γ is estimated using formula (4.2) of Haan et al. (2016). This is used by the LAWS and QB estimators. Furthermore, the τ_n'*–th* quantile is estimated using the formula in page 330 of de Haan et al. (2016). This provides a bias corrected version of the Weissman estimator. This is used by the QB estimator. However, in this case the

asymptotic variance is not estimated using the formula in Haan et al. (2016) Theorem 4.2. Instead, for simplicity the asymptotic variance-covariance matrix is estimated by the formulas Section 3.2 of Padoan and Stupfler (2020).

- k or k_n is the value of the so-called intermediate sequence k_n, $n = 1, 2, \ldots$. Its represents a sequence of positive integers such that $k_n \to \infty$ and $k_n/n \to 0$ as $n \to \infty$. Practically, for each marginal distribution when tau=NULL and method='LAWS' or method='QB', then $\tau_n = 1 - k_n/n$ is the intermediate level of the expectile to be stimated. When tailest='Hill', for each marginal distributions, then k_n specifies the number of k+1 larger order statistics used in the definition of the Hill estimator.
- Given a small value $\alpha \in (0, 1)$ then an estimate of an asymptotic confidence region for tau'_n*-th* d-dimensional expectile, with approximate nominal confidence level $(1 - \alpha)100\%$, is computed. The confidence regions are computed exploiting the formulas in Section 3.2 of Padoan and Stupfler (2020). If varType="asym-Ind-Adj-Log", then an "asymmetric" confidence regions is computed exploiting the asymptotic behaviour of the normalized expectile estimator in logarithmic scale and using a suitable adjustment. This choice is recommended. If varType="asym-Ind-Adj", then the a "symmetric" confidence regions is computed exploiting the asymptotic behaviour of the relative explectile estimator appropriately normalized.
- If plot=TRUE then a graphical representation of the estimates is not provided.

Value

A list with elements:

- ExpctHat: an estimate of the τ'_n*-th* d-dimensional expecile;
- biasTerm: an estimate of the bias term of yje τ'_n*-th* d-dimensional expecile;
- VarCovEHat: an estimate of the asymptotic variance-covariance of the d-dimensional expectile estimator;
- EstConReg: an estimate of the approximate $(1 - \alpha)100\%$ confidence regions for τ'_n*-th* d-dimensional expecile.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://www.unibocconi.it/en/faculty/simone-padoan; Gilles Stupfler, <gilles.stupfler@univ-angers.fr>, https://math.univ-angers.fr/~stupfler/

References

Simone A. Padoan and Gilles Stupfler (2022). Joint inference on extreme expectiles for multivariate heavy-tailed distributions, *Bernoulli* **28**(2), 1021-1048.

See Also

MultiHTailIndex, estMultiExpectiles, extMultiQuantile

Examples

```
# Extreme expectile estimation at the extreme level tau1 obtained with
# d-dimensional observations simulated from a joint distribution with
# a Gumbel copula and equal Frechet marginal distributions.
library(plot3D)
library(copula)
library(evd)

# distributional setting
copula <- "Gumbel"
dist <- "Frechet"

# parameter setting
dep <- 3
dim <- 3
scale <- rep(1, dim)
shape <- rep(3, dim)
par <- list(dep=dep, scale=scale, shape=shape, dim=dim)

# Intermediate level (or sample tail probability 1-tau)
tau <- 0.95
# Extreme level (or tail probability 1-tau1 of unobserved expectile)
tau1 <- 0.9995

# sample size
ndata <- 1000

# Simulates a sample from a multivariate distribution with equal Frechet
# marginals distributions and a Gumbel copula
data <- rmdata(ndata, dist, copula, par)
scatter3D(data[,1], data[,2], data[,3])

# High d-dimensional expectile (intermediate level) estimation
expectHat <- predMultiExpectiles(data, tau, tau1, var=TRUE)

expectHat$ExpctHat
expectHat$VarCovEHat
# run the following command to see the graphical representation
## Not run:
 expectHat <- predMultiExpectiles(data, tau, tau1, var=TRUE, plot=TRUE)

## End(Not run)
```

---

predQuant *Predictive quantile based on the generalized Pareto model*

Description

Bayesian Generalize Pareto-based predictive quantile for continuous and discrete predictive distribution conditioned on intermediate and extreme levels.

Usage

```
predQuant(
  qlev,
  postsamp,
  threshold,
  lb,
  ub,
  type = c("continuous", "discrete"),
  extrapolation = FALSE,
  p,
  k,
  n
)
```

Arguments

qlev	double, quantile level
postsamp	a m by 2 matrix with columns containing the posterior samples of scale and shape parameters of the generalized Pareto distribution, respectively
threshold	threshold for the generalized Pareto model, corresponding to the $n-k$th order statistic of the sample
lb	double, the lower bound of the admissible region for the quantile value
ub	double, the upper bound of the admissible region for the quantile value
type	data type, either "continuous" or "discrete" for count data.
extrapolation	logical; if TRUE, extrapolate using the threshold stability property
p	scalar tail probability for the extrapolation. Must be smaller than k/n (only used if extrapolation=TRUE)
k	integer, number of exceedances for the generalized Pareto (only used if extrapolation=TRUE)
n	integer, number of observations in the full sample. Must be greater than k (only used if extrapolation=TRUE)

Value

a double indicating the value of the quantile

Examples

```
## Not run:
# generate data
set.seed(1234)
n <- 500
samp <- evd::rfrechet(n,0,3,4)
# set effective sample size and threshold
k <- 50
threshold <- sort(samp,decreasing = TRUE)[k+1]
# preliminary mle estimates of scale and shape parameters
mlest <- evd::fpot(samp, threshold)
```

```
# empirical bayes procedure
proc <- estPOT(
  samp,
  k = k,
  pn = c(0.01, 0.005),
  type = "continuous",
  method = "bayesian",
  prior = "empirical",
  start = as.list(mlest$estimate),
  sig0 = 0.1)
# predictive density estimation
yg <- seq(0, 50, by = 2)
nyg <- length(yg)
predDens_int <- predDens(
  yg,
  proc$post_sample,proc$t,
  "continuous",
  extrapolation=FALSE)
predQuant_int <- predQuant(
  0.5,
  proc$post_sample,
  proc$t,
  proc$t + 0.01,
  50,
  "continuous",
  extrapolation = FALSE)
predDens_ext <- predDens(
  yg,
  proc$post_sample,
  proc$t,
  "continuous",
  extrapolation = TRUE,
  p = 0.001,
  k = k,
  n = n)
predQuant_ext <- predQuant(
  0.5,
  proc$post_sample,
  proc$t,
  proc$t + 0.01,
  100,
  "continuous",
  extrapolation = TRUE,
  p = 0.005,
  k = k,
  n = n)

## End(Not run)
```

QuantMES — *Marginal Expected Shortfall Quantile Based Estimation*

Description

Computes a point and interval estimate of the Marginal Expected Shortfall (MES) using a quantile based approach.

Usage

```
QuantMES(data, tau, tau1, var=FALSE, varType="asym-Dep", bias=FALSE, bigBlock=NULL,
         smallBlock=NULL, k=NULL, alpha=0.05)
```

Arguments

`data`	A vector of $(1 \times n)$ observations.
`tau`	A real in $(0, 1)$ specifying the intermediate level τ_n. See **Details**.
`tau1`	A real in $(0, 1)$ specifying the extreme level τ'_n. See **Details**.
`var`	If `var=TRUE` then an estimate of the asymptotic variance of the MES estimator is computed.
`varType`	A string specifying the type of asymptotic variance to compute. By default `varType="asym-Dep"` specifies the variance estimator for serial dependent observations. See **Details**.
`bias`	A logical value. By default `bias=FALSE` specifies that no bias correction is computed. See **Details**.
`bigBlock`	An interger specifying the size of the big-block used to estimaste the asymptotic variance. See **Details**.
`smallBlock`	An interger specifying the size of the small-block used to estimaste the asymptotic variance. See **Details**.
`k`	An integer specifying the value of the intermediate sequence k_n. See **Details**.
`alpha`	A real in $(0, 1)$ specifying the confidence level $(1-\alpha)100\%$ of the approximate confidence interval for the expecile at the intermedite level.

Details

For a dataset `data` of sample size n, an estimate of the τ'_n*-th* MES is computed. The estimation of the MES at the extreme level `tau1` (τ'_n) is indeed meant to be a prediction. Estimates are obtained through the quantile based estimator defined in page 12 of Padoan and Stupfler (2020). Such an estimator depends on the estimation of the tail index γ. Here, γ is estimated using the Hill estimation (see HTailIndex for details). The observations can be either independent or temporal dependent. See Section 4 in Padoan and Stupfler (2020) for details.

- The so-called intermediate level `tau` or τ_n is a sequence of positive reals such that $\tau_n \to 1$ as $n \to \infty$. See predExpectiles for details.
- The so-called extreme level `tau1` or τ'_n is a sequence of positive reals such that $\tau'_n \to 1$ as $n \to \infty$. See predExpectiles for details.
- If `var=TRUE` then an esitmate of the asymptotic variance of the tau'_n*-th* MES is computed. Notice that the estimation of the asymptotic variance **is only available** when γ is estimated using the Hill estimator (see HTailIndex). With independent observations the asymptotic variance is estimated by $\hat{\gamma}^2$, see Corollary 4.3 in Padoan and Stupfler (2020). This is achieved

through `varType="asym-Ind"`. With serial dependent observations the asymptotic variance is estimated by the formula in Corollary 4.2 of Padoan and Stupfler (2020). This is achieved through `varType="asym-Dep"`. See Section 4 and 5 in Padoan and Stupfler (2020) for details. In this latter case the computation of the serial dependence is based on the "big blocks seperated by small blocks" techinque which is a standard tools in time series, see e.g. Leadbetter et al. (1986). The size of the big and small blocks are specified by the parameters `bigBlock` and `smallBlock`, respectively.

- If `bias=TRUE` then γ is estimated using formula (4.2) of Haan et al. (2016). This is used by the LAWS and QB estimators. Furthermore, the τ_n'–*th* quantile is estimated using the formula in page 330 of de Haan et al. (2016). This provides a bias corrected version of the Weissman estimator. This is used by the QB estimator. However, in this case the asymptotic variance is not estimated using the formula in Haan et al. (2016) Theorem 4.2. Instead, for simplicity the asymptotic variance is estimated by the formula in Corollary 3.8, with serial dependent observations, and $\hat{\gamma}^2$ with independent observation (see e.g. de Drees 2000, for the details).
- `k` or k_n is the value of the so-called intermediate sequence k_n, $n = 1, 2, \ldots$. Its represents a sequence of positive integers such that $k_n \to \infty$ and $k_n/n \to 0$ as $n \to \infty$. k_n specifies the number of `k`$+1$ larger order statistics used in the definition of the Hill estimator (see HTailIndex for details).
- If the quantile's extreme level is provided by `alpha_n`, then expectile's extreme level tau_n' is replaced by $tau_n'(\alpha_n)$ which is estimated by the method described in Section 6 of Padoan and Stupfler (2020). See estExtLevel for details.
- Given a small value $\alpha \in (0, 1)$ then an estimate of an asymptotic confidence interval for tau_n'-*th* expectile, with approximate nominal confidence level $(1 - \alpha)100\%$, is computed. The confidence intervals are computed exploiting formula in Corollary 4.2, Theorem 6.2 of Padoan and Stupfler (2020) and (46) in Drees (2003). See Sections 4-6 in Padoan and Stupfler (2020) for details. When `biast=TRUE` confidence intervals are computed in the same way but after correcting the tail index estimate by an estimate of the bias term, see formula (4.2) in de Haan et al. (2016) for details.

Value

A list with elements:

- `HatQMES`: an estimate of the τ_n'-*th* quantile based MES;
- `VarHatQMES`: an estimate of the asymptotic variance of the quantile based MES estimator;
- `CIHatQMES`: an estimate of the approximate $(1 - \alpha)100\%$ confidence interval for τ_n'-*th* MES.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://www.unibocconi.it/en/faculty/simone-padoan; Gilles Stupfler, <gilles.stupfler@univ-angers.fr>, https://math.univ-angers.fr/~stupfler/

References

Anthony C. Davison, Simone A. Padoan and Gilles Stupfler (2023). Tail Risk Inference via Expectiles in Heavy-Tailed Time Series, *Journal of Business & Economic Statistics*, **41**(3) 876-889.

Daouia, A., Girard, S. and Stupfler, G. (2018). Estimation of tail risk based on extreme expectiles. *Journal of the Royal Statistical Society: Series B*, **80**, 263-292.

de Haan, L., Mercadier, C. and Zhou, C. (2016). Adapting extreme value statistics to financial time series: dealing with bias and serial dependence. *Finance and Stochastics*, **20**, 321-354.

Drees, H. (2003). Extreme quantile estimation for dependent data, with applications to finance. *Bernoulli*, **9**, 617-657.

Drees, H. (2000). Weighted approximations of tail processes for β-mixing random variables. *Annals of Applied Probability*, **10**, 1274-1301.

Leadbetter, M.R., Lindgren, G. and Rootzen, H. (1989). Extremes and related properties of random sequences and processes. *Springer*.

See Also

ExpectMES, HTailIndex, predExpectiles, extQuantile

Examples

```
# Marginl Expected Shortfall quantile based estimation at the extreme level
# obtained with 2-dimensional data simulated from an AR(1) with bivariate
# Student-t distributed innovations


tsDist <- "AStudentT"
tsType <- "AR"
tsCopula <- "studentT"

# parameter setting
corr <- 0.8
dep <- 0.8
df <- 3
par <- list(corr=corr, dep=dep, df=df)

# Big- small-blocks setting
bigBlock <- 65
smallBlock <- 15

# quantile's extreme level
tau1 <- 0.9995

# sample size
ndata <- 2500

# Simulates a sample from an AR(1) model with Student-t innovations
data <- rbtimeseries(ndata, tsDist, tsType, tsCopula, par)


# Extreme MES expectile based estimation
MESHat <- QuantMES(data, NULL, tau1, var=TRUE, k=150, bigBlock=bigBlock,
                   smallBlock=smallBlock)
MESHat
```

rbtimeseries *Simulation of Two-Dimensional Temporally Dependent Observations*

Description

Simulates samples from parametric families of bivariate time series models.

Usage

```
rbtimeseries(ndata, dist="studentT", type="AR", copula="Gumbel", par, burnin=1e+03)
```

Arguments

ndata	A positive interger specifying the number of observations to simulate.
dist	A string specifying the parametric family of the innovations distribution. By default dist="studentT" specifies a Student-*t* family of distributions. See **Details**.
type	A string specifying the type of time series. By default type="AR" specifies a linear Auto-Regressive time series. See **Details**.
copula	A string specifying the type copula to be used. By default copula="Gumbel" specifies the Gumbel copula. See **Details**.
par	A list of p parameters to be specified for the bivariate time series parametric family. See **Details**.
burnin	A positive interger specifying the number of initial observations to discard from the simulated sample.

Details

For a time series class (type), with a parametric family (dist) for the innovations, a sample of size ndata is simulated. See for example Brockwell and Davis (2016).

- The available categories of bivariate time series models are: Auto-Regressive (type="AR"), Auto-Regressive and Moving-Average (type="ARMA"), Generalized-Autoregressive-Conditional-Heteroskedasticity (type="GARCH") and Auto-Regressive.
- With AR(1) times series the available families of distributions for the innovations and the dependence structure (copula) are:
  - Student-*t* (dist="studentT" and copula="studentT") with marginal parameters (equal for both distributions): $\phi \in (-1, 1)$ (autoregressive coefficient), $\nu > 0$ (degrees of freedom) and dependence parameter $dep \in (-1, 1)$. The parameters are specified as par <- list(corr, df, dep);
  - Asymmetric Student-*t* (dist="AStudentT" and copula="studentT") with marginal parameters (equal for both distributions): $\phi \in (-1, 1)$ (autoregressive coefficient), $\nu > 0$ (degrees of freedom) and dependence parameter $dep \in (-1, 1)$. The paraters are specified as par <- list(corr, df, dep). Note that in this case the tail index of the lower and upper tail of the first marginal are different, see Padoan and Stupfler (2020) for details;

- With ARMA(1,1) times series the available families of distributions for the innovations and the dependence structure (copula) are:
  - symmetric Pareto (`dist="double-Pareto"` and `copula="Gumbel"` or `copula="Gaussian"`) with marginal parameters (equal for both distributions): $\phi \in (-1, 1)$ (autoregressive coefficient), $\sigma > 0$ (scale), $\alpha > 0$ (shape), θ (movingaverage coefficient), and dependence parameter dep ($dep > 0$ if `copula="Gumbel"` or $dep \in (-1, 1)$ if `copula="Gaussian"`). The parameters are specified as `par <- list(corr, scale, shape, smooth, dep)`.
  - symmetric Pareto (`dist="double-Pareto"` and `copula="Gumbel"` or `copula="Gaussian"`) with marginal parameters (equal for both distributions): $\phi \in (-1, 1)$ (autoregressive coefficient), $\sigma > 0$ (scale), $\alpha > 0$ (shape), θ (movingaverage coefficient), and dependence parameter dep ($dep > 0$ if `copula="Gumbel"` or $dep \in (-1, 1)$ if `copula="Gaussian"`). The parameters are specified as `par <- list(corr, scale, shape, smooth, dep)`. Note that in this case the tail index of the lower and upper tail of the first marginal are different, see Padoan and Stupfler (2020) for details;
- With ARCH(1)/GARCH(1,1) time series the distribution of the innovations are symmetric Gaussian (`dist="Gaussian"`) or asymmetric Gaussian `dist="AGaussian"`. In both cases the marginal parameters (equal for both distributions) are: α_0, α_1, β. In the asymmetric Gaussian case the tail index of the lower and upper tail of the first marginal are different, see Padoan and Stupfler (2020) for details. The available copulas are: Gaussian (`copula="Gaussian"`) with dependence parameter $dep \in (-1, 1)$, Student-*t* (`copula="studentT"`) with dependence parameters $dep \in (-1, 1)$ and $\nu > 0$ (degrees of freedom), Gumbel (`copula="Gumbel"`) with dependence parameter $dep > 0$. The parameters are specified as `par <- list(alpha0, alpha1, beta, dep)` or `par <- list(alpha0, alpha1, beta, dep, df)`.

Value

A vector of $(2 \times n)$ observations simulated from a specified bivariate time series model.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://www.unibocconi.it/en/faculty/simone-padoan; Gilles Stupfler, <gilles.stupfler@univ-angers.fr>, https://math.univ-angers.fr/~stupfler/

References

Brockwell, Peter J., and Richard A. Davis. (2016). Introduction to time series and forecasting. *Springer*.

Anthony C. Davison, Simone A. Padoan and Gilles Stupfler (2023). Tail Risk Inference via Expectiles in Heavy-Tailed Time Series, *Journal of Business & Economic Statistics*, **41**(3) 876-889.

See Also

rtimeseries, expectiles

Examples

```
# Data simulation from a 2-dimensional AR(1) with bivariate Student-t distributed
# innovations, with one marginal distribution whose lower and upper tail indices
```

```
# that are different

tsDist <- "AStudentT"
tsType <- "AR"
tsCopula <- "studentT"

# parameter setting
corr <- 0.8
dep <- 0.8
df <- 3
par <- list(corr=corr, dep=dep, df=df)

# sample size
ndata <- 2500

# Simulates a sample from an AR(1) model with Student-t innovations
data <- rbtimeseries(ndata, tsDist, tsType, tsCopula, par)

# Extreme expectile estimation
plot(data, pch=21)
plot(data[,1], type="l")
plot(data[,2], type="l")
```

---

rmdata *Simulation of d-Dimensional Temporally Independent Observations*

Description

Simulates samples of independent d-dimensional observations from parametric families of joint distributions with a given copula and equal marginal distributions.

Usage

```
rmdata (ndata, dist="studentT", copula="studentT", par)
```

Arguments

`ndata`	A positive interger specifying the number of observations to simulate.
`dist`	A string specifying the parametric family of equal marginal distributions. By default `dist="studentT"` specifies a Student-*t* family of distributions. See **Details**.
`copula`	A string specifying the type copula to be used. By default `copula="studentT"` specifies the Student-*t* copula. See **Details**.
`par`	A list of p parameters to be specified for the multivariate parametric family of distributions. See **Details**.

Details

For a joint multivariate distribution with a given parametric copula class (copula) and a given parametric family of equal marginal distributions (dist), a sample of size ndata is simulated.

- The available copula classes are: Student-*t* (copula="studentT") with $\nu > 0$ degrees of freedom (df) and scale parameters $\rho_{i,j} \in (-1, 1)$ for $i \neq j = 1, \ldots, d$ (sigma), Gaussian (copula="Gaussian") with correlation parameters $\rho_{i,j} \in (-1, 1)$ for $i \neq j = 1, \ldots, d$ (sigma), Clayton (copula="Clayton") with dependence parameter $\theta > 0$ (dep), Gumbel (copula="Gumbel") with dependence parameter $\theta \geq 1$ (dep) and Frank (copula="Frank") with dependence parameter $\theta > 0$ (dep).
- The available families of marginal distributions are:
  - Student-*t* (dist="studentT") with $\nu > 0$ degrees of freedom (df);
  - Asymmetric Student-*t* (dist="AStudentT") with $\nu > 0$ degrees of freedom (df). In this case all the observations are only positive;
  - Frechet (dist="Frechet") with scale $\sigma > 0$ (scale) and shape $\alpha > 0$ (shape) parameters.
  - Frechet (dist="double-Frechet") with scale $\sigma > 0$ (scale) and shape $\alpha > 0$ (shape) parameters. In this case positive and negative observations are allowed;
  - symmetric Pareto (dist="double-Pareto") with scale $\sigma > 0$ (scale) and shape $\alpha > 0$ (shape) parameters. In this case positive and negative observations are allowed.
- The available classes of multivariate joint distributions are:
  - studentT-studentT (dist="studentT" and copula="studentT") with parameters par <- list(df, sigma);
  - studentT (dist="studentT" and copula="None" with parameters par <- list(df, dim). In this case the d variables are regarded as independent;
  - studentT-AstudentT (dist="AstudentT" and copula="studentT") with parameters par <- list(df, sigma, shape);
  - Gaussian-studentT (dist="studentT" and copula="Gaussian") with parameters par <- list(df, sigma);
  - Gaussian-AstudentT (dist="AstudentT" and copula="Gaussian") with parameters par <- list(df, sigma, shape);
  - Frechet (dist="Frechet" and copula="None") with parameters par <- list(shape, dim). In this case the d variables are regarded as independent;
  - Clayton-Frechet (dist="Frechet" and copula="Clayton") with parameters par <- list(dep, dim, scale, shape);
  - Gumbel-Frechet (dist="Frechet" and copula="Gumbel") with parameters par <- list(dep, dim, scale, shape);
  - Frank-Frechet (dist="Frechet" and copula="Frank") with parameters par <- list(dep, dim, scale, shape);
  - Clayton-double-Frechet (dist="double-Frechet" and copula="Clayton") with parameters par <- list(dep, dim, scale, shape);
  - Gumbel-double-Frechet (dist="double-Frechet" and copula="Gumbel") with parameters par <- list(dep, dim, scale, shape);
  - Frank-double-Frechet (dist="double-Frechet" and copula="Frank") with parameters par <- list(dep, dim, scale, shape);

– Clayton-double-Pareto (dist="double-Pareto" and copula="Clayton") with parameters par <- list(dep, dim, scale, shape);
– Gumbel-double-Pareto (dist="double-Pareto" and copula="Gumbel") with parameters par <- list(dep, dim, scale, shape);
– Frank-double-Pareto (dist="double-Pareto" and copula="Frank") with parameters par <- list(dep, dim, scale, shape).

Note that above dim indicates the number of d marginal variables.

Value

A matrix of $(n \times d)$ observations simulated from a specified multivariate parametric joint distribution.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://www.unibocconi.it/en/faculty/simone-padoan; Gilles Stupfler, <gilles.stupfler@univ-angers.fr>, https://math.univ-angers.fr/~stupfler/

References

Joe, H. (2014). Dependence Modeling with Copulas. Chapman & Hall/CRC Press, Boca Raton, USA.

Simone A. Padoan and Gilles Stupfler (2022). Joint inference on extreme expectiles for multivariate heavy-tailed distributions, *Bernoulli* **28**(2), 1021-1048.

See Also

rtimeseries, rbtimeseries

Examples

```
library(plot3D)
library(copula)
library(evd)

# Data simulation from a 3-dimensional random vector a with multivariate distribution
# given by a Gumbel copula and three equal Frechet marginal distributions

# distributional setting
copula <- "Gumbel"
dist <- "Frechet"

# parameter setting
dep <- 3
dim <- 3
scale <- rep(1, dim)
shape <- rep(3, dim)
par <- list(dep=dep, scale=scale, shape=shape, dim=dim)
```

```
# sample size
ndata <- 1000

# Simulates a sample from a multivariate distribution with equal Frechet
# marginal distributions and a Gumbel copula
data <- rmdata(ndata, dist, copula, par)
scatter3D(data[,1], data[,2], data[,3])


# Data simulation from a 3-dimensional random vector a with multivariate distribution
# given by a Gaussian copula and three equal Student-t marginal distributions

# distributional setting
dist <- "studentT"
copula <- "Gaussian"

# parameter setting
rho <- c(0.9, 0.8, 0.7)
sigma <- c(1, 1, 1)
Sigma <- sigma^2 * diag(dim)
Sigma[lower.tri(Sigma)] <- rho
Sigma <- t(Sigma)
Sigma[lower.tri(Sigma)] <- rho
df <- 3
par <- list(sigma=Sigma, df=df)

# sample size
ndata <- 1000

# Simulates a sample from a multivariate distribution with equal Student-t
# marginal distributions and a Gaussian copula
data <- rmdata(ndata, dist, copula, par)
scatter3D(data[,1], data[,2], data[,3])
```

`rtimeseries` *Simulation of One-Dimensional Temporally Dependent Observations*

Description

Simulates samples from parametric families of time series models.

Usage

```
rtimeseries(ndata, dist="studentT", type="AR", par, burnin=1e+03)
```

Arguments

`ndata` A positive interger specifying the number of observations to simulate.

`dist` A string specifying the parametric family of the innovations distribution. By default `dist="studentT"` specifies a Student-*t* family of distributions. See **Details**.

`type` A string specifying the type of time series. By default `type="AR"` specifies a linear Auto-Regressive time series. See **Details**.

`par` A vector of $(1 \times p)$ parameters to be specified for the univariate time series parametric family. See **Details**.

`burnin` A positive interger specifying the number of initial observations to discard from the simulated sample.

Details

For a time series class (`type`) with a parametric family (`dist`) for the innovations, a sample of size `ndata` is simulated. See for example Brockwell and Davis (2016).

- The available categories of time series models are: Auto-Regressive (`type="AR"`), Auto-Regressive and Moving-Average (`type="ARMA"`), Generalized-Autoregressive-Conditional-Heteroskedasticity (`type="GARCH"`) and Auto-Regressive and Moving-Maxima (`type="ARMAX"`).
- With AR(1) and ARMA(1,1) times series the available families of distributions for the innovations are:
  - Student-*t* (`dist="studentT"`) with parameters: $\phi \in (-1, 1)$ (autoregressive coefficient), $\nu > 0$ (degrees of freedom) specified by `par=c(corr, df)`;
  - symmetric Frechet (`dist="double-Frechet"`) with parameters $\phi \in (-1, 1)$ (autoregressive coefficient), $\sigma > 0$ (scale), $\alpha > 0$ (shape), θ (movingaverage coefficient), specified by `par=c(corr, scale, shape, smooth)`;
  - symmetric Pareto (`dist="double-Pareto"`) with parameters $\phi \in (-1, 1)$ (autoregressive coefficient), $\sigma > 0$ (scale), $\alpha > 0$ (shape), θ (movingaverage coefficient), specified by `par=c(corr, scale, shape, smooth)`.

  With ARCH(1)/GARCH(1,1) time series the Gaussian family of distributions is available for the innovations (`dist="Gaussian"`) with parameters, α_0, α_1, β specified by `par=c(alpha0, alpha1, beta)`. Finally, with ARMAX(1) times series the Frechet families of distributions is available for the innovations (`dist="Frechet"`) with parameters, $\phi \in (-1, 1)$ (autoregressive coefficient), $\sigma > 0$ (scale), $\alpha > 0$ (shape) specified by `par=c(corr, scale, shape)`.

Value

A vector of $(1 \times n)$ observations simulated from a specified time series model.

Author(s)

Simone Padoan, <simone.padoan@unibocconi.it>, https://www.unibocconi.it/en/faculty/simone-padoan; Gilles Stupfler, <gilles.stupfler@univ-angers.fr>, https://math.univ-angers.fr/~stupfler/

References

Brockwell, Peter J., and Richard A. Davis. (2016). Introduction to time series and forecasting. *Springer*.

Anthony C. Davison, Simone A. Padoan and Gilles Stupfler (2023). Tail Risk Inference via Expectiles in Heavy-Tailed Time Series, *Journal of Business & Economic Statistics*, **41**(3) 876-889.

See Also

expectiles

Examples

```
# Data simulation from a 1-dimensional AR(1) with univariate Student-t
# distributed innovations

tsDist <- "studentT"
tsType <- "AR"

# parameter setting
corr <- 0.8
df <- 3
par <- c(corr, df)

# sample size
ndata <- 2500

# Simulates a sample from an AR(1) model with Student-t innovations
data <- rtimeseries(ndata, tsDist, tsType, par)

# Graphic representation
plot(data, type="l")
acf(data)
```

`scedastic.test` *Test on the effect of concomitant covariate on the extremes of the response variable*

Description

Given observed data, perform a Kolmogorov-Smirnov type test comparing the cumulative distribution function of the concomitant covariate, defined as $X \mid Y > t$, with t being the threshold, against the cumulative distribution function of the random vector of covariate.

Usage

```
scedastic.test(data, k, M = 1000L, xg, ng, bayes = TRUE, C = 5L, alpha = 0.05)
```

Arguments

`data`	design matrix of dimension `n` by `2` containing the complete data for the dependent variable (first column) and covariate (second column) in [0,1]
`k`	integer, number of exceedances for the generalized Pareto
`M`	integer, number of samples to draw from the posterior distrinution of the law of the concomitant covariate. Default: 1000
`xg`	vector of covariate grid of dimension `ng` by `1` containing a sequence between zero and the last value of the corresponding covariate
`ng`	length of covariate grid
`bayes`	logical indicating the bootstrap method. If `FALSE`, a frequentist bootstrap on the empirical cumulative distribution function of the concomitant covariate is performed. Default to `TRUE`
`C`	integer, hypermparameter entering the posterior distributyion of the law of the concomitant covariate. Default: 5
`alpha`	double, significance level for the critical value of the test, computed as the $(1 - alpha)$ level empirical quantile of the sample of distances between the empirical cumulative distribution function of the concomitant and complete covariate. Default: 0.05

Value

a list with components

- `Delta` maximum observed distance between the empirical distribution functions of the concomitant and complete covariate
- `DeltaM` vector of length M containing the sample of maximum distances between the empirical distribution function of the concomitant complete covariate
- `critical` double, critical value for the test statistic, computed as the $(1 - alpha)$ level empirical quantile of DeltaM
- `pval` double, p-value

References

Dombry, C., S. Padoan and S. Rizzelli (2025). Asymptotic theory for Bayesian inference and prediction: from the ordinary to a conditional Peaks-Over-Threshold method, arXiv:2310.06720v2.

Examples

```
## Not run:
# generate data
set.seed(1234)
n <- 500
samp <- evd::rfrechet(n,0,1:n,4)
# set effective sample size and threshold
k <- 50
threshold <- sort(samp,decreasing = TRUE)[k+1]
# preliminary mle estimates of scale and shape parameters
```

```
mlest <- evd::fpot(samp,
 threshold,
 control=list(maxit = 500))
# empirical bayes procedure
proc <- estPOT(
  samp,
  k = k,
  pn = c(0.01, 0.005),
  type = "continuous",
  method = "bayesian",
  prior = "empirical",
  start = as.list(mlest$estimate),
  sig0 = 0.1)
# conditional predictive density estimation
yg <- seq(0, 50, by = 2)
nyg <- length(yg)
# estimation of scedasis function
# setting
M <- 1e3
C <- 5
alpha <- 0.05
bw <- .5
nsim <- 5000
burn <- 1000
# create covariate
# in sample obs
n_in = n
# number of years ahead
nY = 1
n_out = 365 * nY
# total obs
n_tot = n_in + n_out
# total covariate (in+out sample period)
x <- seq(0, 1, length = n_tot)
# in sample grid dimension for covariate
ng_in <- 150
xg <- seq(0, x[n_in], length = ng_in)
# in+out of sample grid
xg <- c(xg,
 seq(x[n_in + 1],
     x[(n_tot)],
     length = ng_in))
# in+out sample grid dimension
nxg <- length(xg)
xg <- array(xg, c(nxg, 1))
# in sample observations
samp_in <- samp[1:n_in]
ssamp_in <- sort(samp_in, decreasing = TRUE, index = TRUE)
x_in <- x[1:n_in] # in sample covariate
xs <- x_in[ssamp_in$ix[1:k]] # in sample concomitant covariate
# test on covariate effect
test <- scedastic.test(
  cbind(samp, x[1:n]),
```

```
  k,
  M,
  array(xg[1:ng_in], c(ng_in, 1)),
  ng_in,
  TRUE,
  C,
  0.05
)

## End(Not run)
```

---

sp500 *Negative log-returns of S&P 500.*

Description

Series of negative log-returns of the U.S. stock market index Standard and Poor 500.

Format

A $8784 * 2$ data frame.

Details

From the series of n = 8785 closing prices S_t, $t = 1, 2, ...$, for the Standard and Poor 500 stock market index, recorded from January 29, 1985 to December 12, 2019, the series of negative log-returns.

$X_{t+1} = -\log(S_{t+1}/S_t), \quad 1 \le t \le n-1$

is available. Hence the dataset (negative log-returns) contains 8784 observations.

---

testTailHomo *Test on tail homogeneity*

Description

Given observed samples and effective sample size, return the results for a likelihood ratio-type test on tail homogeneity.

Usage

```
testTailHomo(y, k, alpha = 0.05)
```

Arguments

y	list, containing the samples on which the test is to be performed
k	integer, number of exceedances for the generalized Pareto
alpha	double indicating the confidence level for the test. Default: 0.05

Value

list of 7 containing

- `gamHatP` the pooled tail index
- `VarGamHatP` the variance of `gamHatP`
- `CIGamHatP` $(1-\alpha)$ level confidence interval for `gamHatP`
- `BiasGamHatP` bias term of `gamHatP`
- `logLikR` value of the likelihood ratio-type of test statistic
- `PVal` p-value of the test

References

Daouia, A., S.A. Padoan and G. Stupfler (2024). Optimal weighted pooling for inference about the tail index and extreme quantiles, *Bernoulli*, 30(**2**), pp. 1287–1312.

Examples

```
## Not run:
# generate two samples of data
set.seed(1234)
y1 <- evd::rgpd(500, 0, 1, 0.2)
y2 <- evd::rgpd(700, 0, 2, 0.7)
y <- list(y1 = y1, y2 = y2)
# set effective sample size
k <- 50
# perform test
test <- testTailHomo(y,k)

## End(Not run)
```

Index