Package ‘extremis’

November 27, 2020

Version 1.2

Date 2020-11-20

Title Statistics of Extremes

Description Conducts inference in statistical models for extreme values (de Carvalho et al (2012), <doi:10.1080/03610926.2012.709905>; de Carvalho and Davison (2014), <10.1080/01621459.2013.872651>; Einmahl et al (2016), <doi:10.1111/rssb.12099>).

Author Miguel de Carvalho [aut, cre],
Rodrigo Rubio [aut],
Vianey Palacios [aut],
Alejandro Jara [ctb],
Tim Hanson [ctb]

Depends R (>= 3.0.1)

Maintainer Miguel de Carvalho <miguel.decarvalho@ed.ac.uk>

Repository CRAN

Imports emplik,methods,MASS,evd

NeedsCompilation yes

LazyData true

Date/Publication 2020-11-27 02:20:13 UTC

R topics documented:

`angcdf` *Empirical-Likelihood Based Inference for the Angular Measure*

Description

This function computes empirical-likelihood based estimators for the angular distribution function of a bivariate extreme value distribution.

Usage

```
angcdf(Y, tau = 0.95, method = "euclidean", raw = TRUE)
```

Arguments

`Y`	data frame with two columns from which the estimate is to be computed.
`tau`	value used to threshold the data; by default it is set as the 0.95 quantile of the pseudo-radius `tau = 0.95`.
`method`	a character string setting the method to be used. By default `method = "euclidean"`, the other option being `method = "empirical"`. See details.
`raw`	logical; if `TRUE`, `Y` will be converted to unit Fréchet scale. If `FALSE`, `Y` will be understood as already in unit Fréchet scale.

Details

`method = "euclidean"` implements the maximum Euclidean likelihood spectral distribution function as introduced by de Carvalho et al (2013). `method = "empirical"` implements the maximum Empirical likelihood spectral distribution function as introduced by Einmahl and Segers (2009).

Value

`H`	angular distribution function.
`w`	pseudo-angles.
`Y`	data.

The `plot` method depicts the empirical likelihood-based angular distribution function.

Author(s)

Miguel de Carvalho

References

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

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

Examples

```
## de Carvalho et al (2013, Fig. 7)
data(beatenberg)
attach(beatenberg)
fit <- angcdf(beatenberg, tau = 0.98, raw = FALSE)
plot(fit)
rug(fit$w)
```

---

angdensity — *Empirical-Likelihood Based Inference for the Angular Density*

Description

This function computes empirical-likelihood based estimators for the angular distribution function of a bivariate extreme value distribution.

Usage

```
angdensity(Y, tau = 0.95, nu, grid = seq(0.01, 0.99, length = 2^8),
   method = "euclidean", raw = TRUE)
```

Arguments

`Y`	data frame with two columns from which the estimate is to be computed.
`tau`	value used to threshold the data; by default it is set as the 0.95 quantile of the pseudo-radius.
`nu`	concentration parameter of beta distribution which controls the amount of smoothing.
`grid`	grid with coordinates of the points where the angular density is estimated; by default `grid = seq(0.01,0.99,length = 2^8)`.
`method`	a character string setting the method to be used. By default `method = "euclidean"`, the other option being `method = "empirical"`. See details.
`raw`	logical; if `TRUE`, `Y` will be converted to unit Fréchet scale. If `FALSE`, `Y` will be understood as already in unit Fréchet scale.

Details

The smooth angular density was introduced in by de Carvalho et al (2013). `method = "euclidean"` implements the version of the method based on Euclidean likelihood weights, whereas `method = "empirical"` uses Empirical likelihood weights.

Value

h	the estimate angular density values.
grid	grid with coordinates of the points where the angular density is estimated.
w	pseudo-angles.
nu	concentration parameter of the Beta-kernel.
Y	raw data.

The `plot` method depicts the smooth angular density.

Author(s)

Miguel de Carvalho

References

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

Examples

```
## de Carvalho et al (2013, Fig. 7)
data(beatenberg)
attach(beatenberg)
fit <- angdensity(beatenberg, tau = 0.98, nu = 163, raw = FALSE)
plot(fit)
rug(fit$w)
```

---

angscdf — *Smooth Empirical-Likelihood Based Inference for the Angular Measure*

Description

This function computes smooth empirical-likelihood based estimators for the angular distribution function of a bivariate extreme value distribution.

Usage

```
angscdf(Y, tau = 0.95, nu, grid = seq(0.01, 0.99, length = 2^8),
method = "euclidean", raw = TRUE)
```

Arguments

`Y`	data frame with two columns from which the estimate is to be computed.
`tau`	value used to threshold the data; by default it is set as the 0.95 quantile of the pseudo-radius `tau = 0.95`.
`nu`	concentration parameter of beta distribution which controls the amount of smoothing.
`grid`	grid with coordinates of the points where the angular measure is estimated; by default `grid = seq(0.01,0.99,length = 2^8)`.
`method`	a character string setting the method to be used. By default `method = "euclidean"`, the other option being `method = "empirical"`. See details.
`raw`	logical; if `TRUE`, `Y` will be converted to unit Fréchet scale. If `FALSE`, `Y` will be understood as already in unit Fréchet scale.

Details

`method = "euclidean"` implements the maximum Euclidean likelihood spectral distribution function as introduced by de Carvalho et al (2013). `method = "empirical"` implements the maximum Empirical likelihood spectral distribution function as introduced by Einmahl and Segers (2009).

Value

`H`	the estimated angular distribution function values.
`grid`	grid with coordinates of the points where the angular measure is estimated.
`w`	pseudo-angles.
`nu`	concentration parameter of the Beta-kernel.
`Y`	raw data.

The `plot` method depicts the empirical likelihood-based angular distribution function.

Author(s)

Miguel de Carvalho

References

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

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

Examples

```
## de Carvalho et al (2013, Fig. 7)
data(beatenberg)
attach(beatenberg)
fit <- angscdf(beatenberg, tau = 0.98, nu = 163, raw = FALSE)
plot(fit)
rug(fit$w)
```

beatenberg *Beatenberg*

Description

Preprocessed pairs of temperatures in unit Fréchet scale from Beatenberg forest, registered under forest cover and in the open field. Preprocessing is conducted as described in Ferrez et al (2011).

Usage

```
beatenberg
```

Format

The `beatenberg` data frame has 2839 rows and 2 columns.

References

Ferrez, J., A. C. Davison, and Rebetez., M. (2011) Extreme temperature analysis under forest cover compared to an open field. *Agricultural and Forest Meteorology*, 151, 992–1001.

cdensity *Kernel Smoothed Joint Scedasis Density*

Description

This function computes a kernel joint scedasis density estimate.

Usage

```
cdensity(XY, tau = 0.95, raw = TRUE, ...)
```

Arguments

`XY`	data frame from which the estimate is to be computed; first column corresponds to time, the second and third columns correspond to the variables of interest `X` and `Y` respectively.
`tau`	value used to threshold the data `Z = min(X,Y)`; by default `threshold = quantile(Z,tau)`.
`raw`	logical; if `TRUE`, `X` and `Y` will be converted to unit Fréchet scale. If `FALSE`, `X` and `Y` will be understood as already in unit Fréchet scale. If a single variable is provided `raw` is automatically set to `TRUE`.
`...`	further arguments for `density` methods.

Details

This function learns about the joint scedasis function using kernel methods as discussed in Palacios and de Carvalho (2020). In the particular case where XY contains no third column, the function learns about the scedasis function of Einmahl et al (2016).

Value

`c`	scedasis density estimator.
`k`	number of exceedances above the threshold.
`w`	standardized indices of exceedances.
`XY`	raw data.

The `plot` method depicts the smooth scedasis density.

Author(s)

Miguel de Carvalho and Vianey Palacios

References

Einmahl, J. H., Haan, L., and Zhou, C. (2016) Statistics of heteroscedastic extremes. *Journal of the Royal Statistical Society: Ser. B*, 78(1), 31–51.

Palacios, V. and de Carvalho, M. (2020) Bayesian semiparametric modeling of jointly heteroscedastic extremes. *Preprint.*

Examples

```
data(lse)
attach(lse)
XY <- data.frame(DATE[-1], -diff(log(ROYAL.DUTCH.SHELL.B)))
T <- dim(XY)[1]
k <- floor((0.4258597) * T / (log(T)))
fit <- cdensity(XY, kernel = "biweight", bw = 0.1 / sqrt(7))
plot(fit)
plot(fit, original = FALSE)

## Example from Palacios and de Carvalho (2020, submitted)
library(evd)
T <- 5000
time <- seq(1/T, 1, by = 1/T)
set.seed(1263)
aux <- matrix(0,T,2)
for(i in 1:T) {
    aux[i,] <- rbvevd(1,dep=sin(time[i]*pi),model="log",
    mar1 = c(1, 1, 1), mar2 = c(1,1,1))}
XY <- cbind(time,aux)
fit <- cdensity(XY, kernel = "biweight", bw = 0.1)
plot(fit)
```

cdf *Empirical Scedasis Distribution Function*

Description

This function computes the empirical scedasis distribution function.

Usage

```
cdf(Y, threshold = quantile(Y[, 2], 0.95))
```

Arguments

`Y`	data frame from which the estimate is to be computed; first column corresponds to time and the second to the variable of interest.
`threshold`	value used to threshold the data `y`; by default `threshold = quantile(Y[,2],0.95)`.

Details

The empirical scedasis distribution function was introduced by Einmahl et al (2016).

Value

`C`	empirical scedasis distribution function.
`w`	standardized indices of exceedances.
`k`	number of exceedances above a threshold.
`Y`	raw data.

The `plot` method depicts the empirical cumulative scedasis function, and the reference line for the case of constant frequency of extremes over time (if `uniform = TRUE`).

Author(s)

Miguel de Carvalho

References

Einmahl, J. H., Haan, L., and Zhou, C. (2016) Statistics of heteroscedastic extremes. *Journal of the Royal Statistical Society: Ser. B*, 78(1), 31–51.

Examples

```
data(sp500)
attach(sp500)
Y <- data.frame(date[-1], -diff(log(close)))
fit <- cdf(Y)
plot(fit)
plot(fit, original = FALSE)
```

`cmodes` *Mode Mass Function*

Description

This function computes the mode mass function.

Usage

```
cmodes(Y, thresholds = apply(Y[, -1], 2, quantile, probs =
                 0.95), nu = 100, ...)
```

Arguments

`Y`	data frame from which the estimate is to be computed; first column corresponds to time and the second to the variable of interest.
`thresholds`	values used to threshold the data `y`; by default `threshold = quantile(y,0.95)`.
`nu`	concentration parameter of beta kernel used to smooth mode mass function.
`...`	further arguments for `density` methods.

Details

The scedasis functions on which the mode mass function is based are computed using the default `"nrd0"` option for bandwidth.

Value

`c`	scedasis density estimators.
`k`	number of exceedances above the threshold.
`w`	standardized indices of exceedances.
`Y`	raw data.

The `plot` method depicts the smooth mode mass function along with the smooth scedasis densities.

Author(s)

Miguel de Carvalho

References

Rubio, R., de Carvalho, M., and Huser, R. (2018) Similarity-Based Clustering of Extreme Losses from the London Stock Exchange. Submitted.

Examples

```
data(lse)
attach(lse)
nlr <- -apply(log(lse[, -1]), 2, diff)
Y <- data.frame(DATE[-1], nlr)
T <- dim(Y)[1]
k <- floor((0.4258597) * T / (log(T)))
fit <- cmodes(Y, thresholds = as.numeric(apply(nlr, 2, sort)[T - k, ]),
              kernel = "biweight", bw = 0.1 / sqrt(7), nu = 100)
plot(fit)
```

---

cPTdensity — *Nonparametric Bayesian Estimation of Joint Scedasis Density using Mixtures of Polya Trees*

Description

This function computes a Bayesian joint scedasis density estimate generating a posterior sample of a finite Mixture of Polya trees.

Usage

```
cPTdensity(XY, tau = 0.95, raw = TRUE, prior, mcmc, state,status, data =
          sys.frame(sys.parent()), na.action = na.fail)
```

Arguments

`XY`	data frame from which the estimate is to be computed; first column corresponds to time, the second and third columns correspond to the variables of interest `X` and `Y` respectively.
`tau`	value used to threshold the data `z = min(X,Y)`; by default `threshold = quantile(z,tau)`.
`raw`	logical; if `TRUE`, `X` and `Y` will be converted to unit Fréchet scale. If `FALSE`, `X` and `Y` will be understood as already in unit Fréchet scale. If a single variable is provided `raw` is automatically set to `TRUE`.
`prior`	a list giving the prior information. The list includes the following parameter: `a0` and `b0` giving the hyperparameters for prior distribution of the precision parameter of the Poly tree prior, `alpha` giving the value of the precision parameter (it must be specified if `alpha` is missing, see details below), optionally `M` giving the finite level to be considered (if `M` is specified, a partially specified mixture of Polya trees model is fitted), `tau1` and `tau2` giving the hyperparameters of the inverted gamma prior distribution for the centering Beta scale parameter,`m0` and `S0` giving the hyperparameters of the inverted gamma prior distribution for the centering Beta scale parameter, and `al`, `be` giving the value of the shape and scale parameter of the centering distribution.

mcmc a list giving the MCMC parameters. The list must include the following integers: nburn giving the number of burn-in scans, nskip giving the thinning interval, nsave giving the total number of scans to be saved, ndisplay giving the number of saved scans to be displayed on screen (the function reports on the screen when every ndisplay iterations have been carried out), tune1, tune2, and tune3 giving the positive Metropolis tuning parameter for the baseline shape, scale, and precision parameter, respectively (the default value is 1.1)

state a list giving the current value of the parameters. This list is used if the current analysis is the continuation of a previous analysis.

status a logical variable indicating whether this run is new (TRUE) or the continuation of a previous analysis (FALSE). In the latter case the current value of the parameters must be specified in the object state.

data data frame.

na.action a function that indicates what should happen when the data contain NAs. The default action (na.fail) causes PTdensity to print an error message and terminate if there are any incomplete observations.

Details

This function learns about the joint scedasis function using a Mixture of Polya Trees (MPT) prior as proposed in Palacios and de Carvalho (2020). In the particular case where XY contains no third column, the function learns about the scedasis function of Einmahl et al (2016) using an MPT prior. The details are as follows. Let

$$Z_i = min(X_i, Y_i)$$

. The model is given by:

$$Z_1, \dots, Z_n | G \sim G$$

$$G | \alpha, al, be \sim PT(\Pi^{al,be}, A)$$

where, the the PT is centered around a $Beta(al, be)$ distribution, by taking each m level of the partition $\Pi^{al,be}$ to coincide with the $k/2^m, k = 0, \dots, 2^m$ quantile of the $Beta(al, be)$ distribution. The family $A = \{\alpha_e : e \in E^*\}$, where $E^* = \bigcup_{m=0}^{m} E^m$ and E^m is the m-fold product of $E = \{0, 1\}$, was specified as $\alpha_{e_1 \dots e_m} = \alpha m^2$.

In the univariate case,the following proper priors can be assigned:

$$al | m_0, S_0 \sim LNormal(m_0, S_0)$$

$$be | \tau_1, \tau_2 \sim LNormal(\tau_1, \tau_2)$$

To complete the model specification, independent hyperpriors are assumed,

$$\alpha | a_0, b_0 \sim \Gamma(a_0, b_0)$$

The precision parameter, α, of the PT prior can be considered as random, having a gamma distribution, $\Gamma(a_0, b_0)$, or fixed at some particular value. To let α to be fixed at a particular value, set a_0 to NULL in the prior specification.

In the computational implementation of the model, Metropolis–Hastings steps are used to sample the posterior distribution of the baseline and precision parameters.

Value

c	Joint scedasis density estimator.
k	number of exceedances above the threshold.
w	standardized indices of exceedances.
Y	raw data.
al	giving the value of the baseline shape parameter.
be	giving the value of the baseline scale parameter.
alpha	giving the value of the precision parameter.

The `plot` method depicts the estimated joint scedasis density.

Author(s)

Miguel de Carvalho and Vianey Palacios

References

Palacios, V., de Carvalho, M. (2020) Bayesian semiparametric modeling of jointly heteroscedastic extremes. *Preprint.*

Examples

```
## Not run:
## Example from Palacios and de Carvalho (2020, submitted)
library(evd)
## Initial state
state <- NULL
## MCMC parameters
nburn <- 2000
nsave <- 1000
nskip <- 0
ndisplay <- 500
mcmc <- list(nburn=nburn,nsave=nsave,nskip=nskip,ndisplay=ndisplay,
 tune1=1.1,tune2=1.1,tune3=1.1)
## Prior information
prior<-list(a0=1,b0=1,M=8,m0=.01,S0=.01,tau1=.01,tau2=.01);

T <- 5000
time <- seq(1/T, 1, by = 1/T)
set.seed(12333)
aux <- matrix(0, T, 2)
for (i in 1:T) {
    aux[i,]<-rbvevd(1, dep =.5, asy=c(sin(pi*time[i]),time[i]),
    model="alog",mar1 = c(1, 1, 1), mar2 = c(1, 1, 1))
}
XY <- cbind(time, aux)
fit <- cPTdensity(XY, prior = prior, mcmc = mcmc, state = state, status =
TRUE)
plot(fit)

## End(Not run)
```

kgvar — *K-Geometric Means Algorithm for Value-at-Risk*

Description

This function performs k-geometric means for time-varying value-at-risk.

Usage

```
kgvar(y, centers, iter.max = 10, conf.level = 0.95)
```

Arguments

`y`	data frame from which the estimate is to be computed; first column corresponds to time and the second to the remainder of interest.
`centers`	the number of clusters or a set of initial (distinct) cluster centres. If a number, a random set of (distinct) rows in `y` is chosen as the initial centers.
`iter.max`	the maximum number of iterations allowed. The default is 10.
`conf.level`	the confidence level. The default is 0.95.

Details

The intermediate sequence κ_T is chosen proportional to $T/\log T$.

Value

kgvar returns an object of class `"kgvar"` which has a fitted method. It is a list with at least the following components:

`var.new`	cluster center value-at-risk function.
`clusters`	cluster allocation.
`Y`	raw data.
`n.clust`	number of clusters.
`scale.param`	the scale parameters in the Pareto-like tail specification.
`conf.level`	the confidence level.
`hill`	hill estimator of extreme value index.

The `plot` method depicts the k-geometric means algorithm for time-varying value-at-risk. If `c.c` is `TRUE`, the method displays the cluster means.

Author(s)

Miguel de Carvalho, Rodrigo Rubio.

References

Rubio, R., de Carvalho, M. and Huser, R. (2018) Similarity-Based Clustering of Extreme Losses from the London Stock Exchange. Submitted.

Examples

```
## Not run:
## Example (Overlapping version of Fig. 8 in Supplementary Materials)
data(lse)
attach(lse)
y <- -apply(log(lse[, -1]), 2, diff)
fit <- kgvar(y, centers = 3)
plot(fit, c.c = TRUE, ylim = c(0, 0.1))

## End(Not run)
```

---

khetmeans *K-Means Clustering for Heteroscedastic Extremes*

Description

This function performs k-means clustering for heteroscedastic extremes.

Usage

```
khetmeans(y, centers, iter.max = 10, alpha = 0.5)
```

Arguments

`y`	data frame from which the estimate is to be computed; first column corresponds to time and the second to the remainder of interest.
`centers`	the number of clusters or a set of initial (distinct) cluster centres. If a number, a random set of (distinct) rows in `y` is chosen as the initial centers.
`iter.max`	the maximum number of iterations allowed. The default is 10.
`alpha`	the tuning parameter. The default is 0.5.

Details

The intermediate sequence κ_T is chosen proportional to $T/\log T$.

Value

khetmeans returns an object of class "`khetmeans`" which has a fitted method. It is a list with at least the following components:

`mus.new`	cluster center scedasis density.
`mugamma.new`	cluster center extreme value index.

`clusters`	cluster allocation.
`Y`	raw data.
`n.clust`	number of clusters.

The `plot` method depicts the k-means clustering for heteroscedastic extremes. If `c.c` is `TRUE`, the method displays the cluster means.

Author(s)

Miguel de Carvalho, Rodrigo Rubio.

References

Rubio, R., de Carvalho, M. and Huser, R. (2018) Similarity-Based Clustering of Extreme Losses from the London Stock Exchange. Submitted.

Examples

```
## Not run:
## Example 1  (Scenario B, T = 5000)
## This example requires package evd
require(evd)
set.seed(12)
T <- 5000
n <- 30
b <- 0.1
gamma1 <- 0.7
gamma2 <- 1
grid <- seq(0, 1, length = 100)
c2 <- function(s)
    dbeta(s, 2, 5)
c3 <- function(s)
    dbeta(s, 5, 2)
X <- matrix(0, ncol = T, nrow = n)
for(i in 1:5)
  for(j in 1:T)
    X[i,  j] <- rgev(1, c2(j / T), c2(j / T), gamma1)
for(i in 6:15)
  for(j in 1:T)
    X[i,  j] <- rgev(1, c2(j / T), c2(j / T), gamma2)
for(i in 16:20)
  for(j in 1:T)
    X[i,  j] <- rgev(1, c3(j / T), c3(j / T), gamma1)
for(i in 21:30)
  for(j in 1:T)
    X[i,  j] <- rgev(1, c3(j / T), c3(j / T), gamma2)
Y <- t(X)
fit <- khetmeans(Y, centers = 4)
plot(fit, c.c = TRUE)
lines(grid, c2(grid), type = 'l', lwd = 8, col = 'black')
lines(grid, c3(grid), type = 'l', lwd = 8, col = 'black')
```

```
## End(Not run)

## Not run:
## Example 2 (Overlapping version of Fig. 5 in Supplementary Materials)
data(lse)
attach(lse)
y <- -apply(log(lse[, -1]), 2, diff)
fit <- khetmeans(y, centers = 3)
plot(fit, c.c = TRUE, ylim = c(0, 3))

## End(Not run)
```

---

lse *Selected Stocks from the London Stock Exchange*

Description

Prices at close from 26 selected stocks from the London stock exchange from 1989 till 2016.

Usage

```
lse
```

Format

The `lse` data frame has 6894 rows and 27 columns.

References

Rubio, R., de Carvalho, M., and Huser (2018) Similarity-based clustering of extreme losses from the London stock exchange.

---

plotFrechet *Unit Fréchet Scatterplot in Log-log Scale*

Description

This function depicts a scatterplot of bivariate data transformed to unit Fréchet scale.

Usage

```
plotFrechet(Y, tau = 0.95, raw = TRUE, ...)
```

Arguments

`Y`	list with data from which the estimates are to be computed.
`tau`	value used to threshold the data `y`; by default `treshold = quantile(y,0.95)`.
`raw`	logical; if `TRUE`, `Y` will be converted to unit Fréchet scale. If `FALSE`, `Y` will be understood as already in unit Fréchet scale.
`...`	other arguments to be passed to `plot`.

Details

The solid line corresponds to the boundary threshold in the log-log scale, with both axes being logarithmic.

Author(s)

Miguel de Carvalho

Examples

```
## de Carvalho et al (2013, Fig. 5)
data(beatenberg)
plotFrechet(beatenberg, xlab = "Forest Cover", ylab = "Open Field",
            raw = FALSE)
```

---

sp500 *Standard & Poor 500*

Description

Daily Standard and Poor's index at close from 1988 till 2007.

Usage

```
sp500
```

Format

The `sp500` data frame has 5043 rows and 2 columns.

References

de Carvalho, M. (2016) Statistics of extremes: Challenges and opportunities. In: *Handbook of EVT and its Applications to Finance and Insurance*. Eds F. Longin. Hoboken: Wiley.

Einmahl, J. H., Haan, L., and Zhou, C. (2016) Statistics of heteroscedastic extremes. *Journal of the Royal Statistical Society: Ser. B*, 78(1), 31–51.

Index