Package ‘exdex’

May 8, 2026

Type Package

Title Estimation of the Extremal Index

Version 1.2.4

Date 2026-01-10

Description Performs frequentist inference for the extremal index of a stationary time series. Two types of methodology are used. One type is based on a model that relates the distribution of block maxima to the marginal distribution of series and leads to the semiparametric maxima estimators described in Northrop (2015) <doi:10.1007/s10687-015-0221-5> and Berghaus and Bucher (2018) <doi:10.1214/17-AOS1621>. Sliding block maxima are used to increase precision of estimation. A graphical block size diagnostic is provided. The other type of methodology uses a model for the distribution of threshold inter-exceedance times (Ferro and Segers (2003) <doi:10.1111/1467-9868.00401>). Three versions of this type of approach are provided: the iterated weight least squares approach of Suveges (2007) <doi:10.1007/s10687-007-0034-2>, the K-gaps model of Suveges and Davison (2010) <doi:10.1214/09-AOAS292> and a similar approach of Holesovsky and Fusek (2020) <doi:10.1007/s10687-020-00374-3> that we refer to as D-gaps. For the K-gaps and D-gaps models this package allows missing values in the data, can accommodate independent subsets of data, such as monthly or seasonal time series from different years, and can incorporate information from right-censored inter-exceedance times. Graphical diagnostics for the threshold level and the respective tuning parameters K and D are provided.

Imports chandwich, graphics, methods, Rcpp, RcppRoll, stats

License GPL (>= 2)

Depends R (>= 3.3.0)

Suggests knitr, revdbayes, rmarkdown, testthat, zoo (>= 1.6.4)

LazyData true

Encoding UTF-8

RoxygenNote 7.2.3

VignetteBuilder knitr

URL https://github.com/paulnorthrop/exdex,
https://paulnorthrop.github.io/exdex/

BugReports https://github.com/paulnorthrop/exdex/issues

LinkingTo Rcpp, RcppArmadillo

Config/testthat/edition 3

NeedsCompilation yes

Author Paul J. Northrop [aut, cre, cph],
Constantinos Christodoulides [aut, cph]

Maintainer Paul J. Northrop <p.northrop@ucl.ac.uk>

Repository CRAN

Date/Publication 2026-01-11 10:10:08 UTC

Contents

`exdex-package` *exdex: Estimation of the Extremal Index*

Description

The extremal index θ is a measure of the degree of local dependence in the extremes of a stationary process. The *exdex* package performs frequentist inference about θ using the methodologies proposed in Northrop (2015), Berghaus and Bucher (2018), Suveges (2007), Suveges and Davison (2010) and Holesovsky and Fusek (2020).

Details

Functions to implement four estimators of the extremal index are provided, namely

- `spm`: semiparametric maxima estimator, using block maxima: (Northrop, 2015; Berghaus and Bucher, 2018)
- `kgaps`: K-gaps estimator, using threshold inter-exceedance times (Suveges and Davison, 2010)
- `dgaps`: D-gaps estimator, using threshold inter-exceedance times (Holesovsky and Fusek, 2020))
- `iwls`: iterated weighted least squares estimator, using threshold inter-exceedance times: (Suveges, 2007)

The functions `choose_b`, `choose_uk` and `choose_ud` provide graphical diagnostics for choosing the respective tuning parameters of the semiparametric maxima, K-gaps and D-gaps estimators.

For the K-gaps and D-gaps models the ‘exdex‘ package allows missing values in the data, can accommodate independent subsets of data, such as monthly or seasonal time series from different years, and can incorporate information from censored inter-exceedance times.

See `vignette("exdex-vignette", package = "exdex")` for an overview of the package.

Author(s)

Maintainer: Paul J. Northrop <p.northrop@ucl.ac.uk> [copyright holder]

Authors:

- Constantinos Christodoulides [copyright holder]

References

Berghaus, B., Bucher, A. (2018) Weak convergence of a pseudo maximum likelihood estimator for the extremal index. *Ann. Statist.* **46**(5), 2307-2335. doi:10.1214/17AOS1621

Holesovsky, J. and Fusek, M. Estimation of the extremal index using censored distributions. Extremes 23, 197-213 (2020). doi:10.1007/s10687020003743

Northrop, P. J. (2015) An efficient semiparametric maxima estimator of the extremal index. *Extremes* **18**(4), 585-603. doi:10.1007/s1068701502215

Suveges, M. (2007) Likelihood estimation of the extremal index. *Extremes*, **10**, 41-55. doi:10.1007/s1068700700342

Suveges, M. and Davison, A. C. (2010) Model misspecification in peaks over threshold analysis, *Annals of Applied Statistics*, **4**(1), 203-221. doi:10.1214/09AOAS292

See Also

`spm`: semiparametric maxima estimator.

`kgaps`: K-gaps estimator.

`dgaps`: D-gaps estimator.

`iwls`: iterated weighted least squares estimator.

`choose_b`, `choose_ud` and `choose_ud` for choosing tuning parameters.

`newlyn`, `sp500` and `cheeseboro` for example datasets.

`all_max_rcpp` *Sliding and disjoint block maxima*

Description

Calculates the (sliding) maxima of all blocks of `b` contiguous values and all sets of the maxima of disjoint blocks of `b` contiguous values in the vector `x`. This provides the first step of computations in `spm`.

Usage

```
all_max_rcpp(x, b = 1, which_dj = c("all", "first", "last"), ...)
```

Arguments

`x`	A numeric vector of raw observations.
`b`	A numeric scalar. The block size.
`which_dj`	A character scalar. Determines Which sets of disjoint maxima are calculated: `"all"`, all sets; `"first"`, only the set whose first block starts on the first observation in `x`; `"last"`, only the set whose last block end on the last observation in `x`.
`...`	Further arguments to be passed to `roll_max`.

Details

Sliding maxima. The function roll_max in the RcppRoll package is used.

Disjoint maxima. If n = length(x) is an integer multiple of b, or if which_dj = "first" or which_dj = "last" then only one set of n / b disjoint block maxima are returned. Otherwise, n - floor(n / b) * b + 1 sets of floor(n / b) disjoint block maxima are returned. Set i are the disjoint maxima of x[i:(i + floor(n / b) * b - 1)]. That is, all possible sets of contiguous disjoint maxima achieving the maxima length of floor(n / b) are calculated.

In both instances na.rm = TRUE is passed to max so that blocks containing missing values produce a non-missing result.

Also returned are the values in x that contribute to each set of block maxima.

Value

A list containing

- ys — a numeric vector containing one set of sliding block maxima.
- xs — a numeric vector containing the values that contribute to ys, that is, the whole input vector x.
- yd — if which_dj = "all" a floor(n / b) by n - floor(n / b) * b + 1 numeric matrix. Each column contains a set of disjoint maxima. Otherwise, a floor(n / b) by 1 numeric matrix containing one set of block maxima.
- xd — if which_dj = "all" a floor(n / b) * b by n - floor(n / b) * b + 1 numeric matrix. Each column contains the values in x that contribute to the corresponding column in yd. Otherwise, a floor(n / b) by 1 numeric matrix containing one the one set of the values in x that contribute to yd.

See Also

spm for semiparametric estimation of the extremal index based on block maxima.

Examples

```
x <- 1:11
all_max_rcpp(x, 3)
all_max_rcpp(x, 3, which_dj = "first")
all_max_rcpp(x, 3, which_dj = "last")
```

---

cheeseboro — *Cheeseboro hourly maximum wind gusts*

Description

The matrix cheeseboro contains hourly maximum wind gusts (in miles per hour) recorded at the Cheeseboro weather station near Thousand Oaks, Southern California, USA during the month of January over the period 2000-2009. These data are analysed in Reich and Shaby (2016).

Usage

```
cheeseboro
```

Format

A 744 by 10 numeric matrix. Column `i` contains the hourly maximum wind gusts (in miles per hour) from Cheeseboro in the year 2000 + `i` - 1. The columns are named 2000, 2001, ..., 2009 and the rows are named `dayjhourk`, where `j` is the day of the month and `k` the hour of the day.

Note

There are 42 missing values, located in 6 of the 10 years, namely 2000-2003 and 2005-2006.

Source

The Remote Automated Weather Stations USA Climate Archive at `https://raws.dri.edu/`, more specifically the Daily Summaries of the Cheeseboro page.

References

Reich, B. J. and Shaby, B. A. (2016). 'Time series of Extremes', in Dey, D. K. and Yan, J. (eds.) Extreme Value Modeling and Risk Analysis. New York: Chapman and Hall/CRC, pp. 131-151.

---

choose_b *Block length diagnostic for the semiparametric maxima estimator*

Description

Creates data for a plot to aid the choice of the block length `b` to supply to `spm`. The general idea is to select the smallest value of `b` above which estimates of the extremal index θ appear to be constant with respect to `b`, taking into account sampling variability. `plot.choose_b` creates the plot.

Usage

```
choose_b(
  data,
  b,
  bias_adjust = c("BB3", "BB1", "N", "none"),
  constrain = TRUE,
  varN = TRUE,
  level = 0.95,
  interval_type = c("norm", "lik"),
  conf_scale = c("theta", "log"),
  type = c("vertical", "cholesky", "spectral", "none")
)
```

Arguments

- `data` A numeric vector of raw data. No missing values are allowed.
- `b` A numeric scalar. The block size.
- `bias_adjust` A character scalar. Is bias-adjustment of the raw estimate of θ performed using the bias-reduced estimator (`bias_adjust = "BB3"`), derived in Section 5 of Berghaus and Bucher (2018); or a simpler version (`bias_adjust = "BB1"`), in which the raw estimate is multiplied by $(k-1)/k$, where k is the number of blocks; or the bias-adjustment of the empirical distribution function used to calculate the estimate, as detailed in Section 2 of Northrop (2015). When disjoint maxima are used `bias_adjust = "BB1"` and `bias_adjust = "N"` give identical estimates of the Berghaus and Bucher (2018) variant, as explained at the end of Section 5 of Berghaus and Bucher (2018). If `bias_adjust = "none"` then no bias-adjustment is performed.
- `constrain` A logical scalar. If `constrain = TRUE` then any estimates that are greater than 1 are set to 1, that is, they are constrained to lie in (0, 1]. This is carried out *after* any bias-adjustment. Otherwise, estimates that are greater than 1 may be obtained.
- `varN` A logical scalar. If `varN = TRUE` then the estimation of the sampling variance of the Northrop (2015) estimator is tailored to that estimator. Otherwise, the sampling variance derived in Berghaus and Bucher (2018) is used. See **Details** for further information.
- `level` A numeric scalar in (0, 1). The confidence level required.
- `interval_type` A character scalar: `"norm"` for intervals of type (a), `"lik"` for intervals of type (b).
- `conf_scale` A character scalar. If `interval_type = "norm"` then `conf_scale` determines the scale on which we use approximate large-sample normality of the estimators to estimate confidence intervals of type (a).

  If `conf_scale = "theta"` then confidence intervals are estimated for θ directly. If `conf_scale = "log"` then confidence intervals are first estimated for $\log \theta$ and then transformed back to the θ-scale.

  Any bias-adjustment requested in the original call to `spm`, using it's `bias_adjust` argument, is automatically applied here.
- `type` A character scalar. The argument `type` to be passed to `conf_intervals` in the `chandwich` package in order to estimate the likelihood-based intervals. Using `type = "none"` is *not* advised because then the intervals are based on naive estimated standard errors. In particular, if (the default) `sliding = TRUE` was used in the call to `spm` then the unadjusted likelihood-based confidence intervals provide *vast* underestimates of uncertainty.

Details

For each block size in `b` the extremal index θ is estimated using `spm`. The estimates of θ approximate `conf`% confidence intervals for θ are stored for plotting (by `plot.choose_b`) to produce a simple graphical diagnostic to inform the choice of block size. This plot is used to choose a block size above which the underlying value of θ may be approximately constant. This is akin to a threshold stability plot: see Chapter 4 of Coles (2001), for example.

The nature of the calculation of the sampling variances of the estimates of θ (see spm for details) means that choose_b may be a little slow to run if b contains many values, particularly if some of them are small.

For very small block sizes it may not be possible to estimate the confidence intervals. See **Details** in spm. For any such block sizes the intervals will be missing from the plot.

Value

An object of class c("choose_b", "exdex") containing

theta_sl, theta_dj
: numeric b by 3 matrices of estimates of θ using sliding and disjoint blocks. Columns 1-3 relate to the estimators N2015, BB2018 and BB2018b.

lower_sl, lower_dj
: Similarly for the lower limits of the confidence intervals.

upper_sl, upper_dj
: Similarly for the upper limits of the confidence intervals.

b
: the input b

call
: the call to choose_b.

References

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

Northrop, P. J. (2015) An efficient semiparametric maxima estimator of the extremal index. *Extremes* **18**(4), 585-603. doi:10.1007/s1068701502215

Berghaus, B., Bucher, A. (2018) Weak convergence of a pseudo maximum likelihood estimator for the extremal index. *Ann. Statist.* **46**(5), 2307-2335. doi:10.1214/17AOS1621

See Also

plot.choose_b to produce the block length diagnostic plot.

Examples

```
# Newlyn seas surges
# Plot like the top left of Northrop (2015)
# Remove the last 14 values because 2880 has lots of factors
b_vals <- c(2,3,4,5,6,8,9,10,12,15,16,18,20,24,30,32,36,40,45,48,54,60)
res <- choose_b(newlyn[1:2880], b_vals)
# Some b are too small for the sampling variance of the sliding blocks
# estimator to be estimated
plot(res)
plot(res, estimator = "BB2018")
plot(res, maxima = "disjoint")

# S&P 500 index: similar to Berghaus and Bucher (2018), Fig 4 top left
b_vals <- c(10, seq(from = 25, to = 350, by = 25), 357)
res500 <- choose_b(sp500, b_vals)
```

```
plot(res500, ylim = c(0, 1))
plot(res500, estimator = "BB2018", ylim = c(0, 1))
```

---

choose_ud *Threshold u and runs parameter D diagnostic for the D-gaps estimator*

Description

Creates data for a plot to aid the choice of the threshold and run parameter D for the D-gaps estimator (see dgaps). plot.choose_ud creates the plot.

Usage

```
choose_ud(data, u, D = 1, inc_cens = TRUE)
```

Arguments

data	A numeric vector or numeric matrix of raw data. If data is a matrix then the log-likelihood is constructed as the sum of (independent) contributions from different columns. A common situation is where each column relates to a different year. If data contains missing values then split_by_NAs is used to divide the data into sequences of non-missing values.
u, D	Numeric vectors. u is a vector of extreme value thresholds applied to data. D is a vector of values of the run parameter D, as defined in Holesovsky and Fusek (2020). See dgaps for more details. Any values in u that are greater than all the observations in data will be removed without a warning being given.
inc_cens	A logical scalar indicating whether or not to include contributions from right-censored inter-exceedance times, relating to the first and last observations. It is known that these times are greater than or equal to the time observed. If data has multiple columns then there will be right-censored first and last inter-exceedance times for each column. See **Details** in dgaps.

Details

For each combination of threshold in u and D in D the functions dgaps and dgaps_imt are called in order to estimate θ and to perform the information matrix test of Holesovsky and Fusek (2020).

Value

An object (a list) of class c("choose_ud", "exdex") containing

imt	an object of class c("dgaps_imt", "exdex") returned from dgaps_imt.
theta	a length(u) by length(D) matrix. Element (i,j) of theta contains an object (a list) of class c("dgaps", "exdex"), a result of a call dgaps(data, u[j], D[i]) to dgaps.

References

Holesovsky, J. and Fusek, M. Estimation of the extremal index using censored distributions. Extremes 23, 197-213 (2020). doi:10.1007/s10687020003743

See Also

dgaps for maximum likelihood estimation of the extremal index θ using the D-gaps model.

dgaps_imt for the information matrix test under the D-gaps model

plot.choose_ud to produce the diagnostic plot.

Examples

```
### S&P 500 index

# Multiple thresholds and left-censoring parameters
u <- quantile(sp500, probs = seq(0.2, 0.9, by = 0.1))
imt_theta <- choose_ud(sp500, u = u, D = 1:5)
plot(imt_theta)
plot(imt_theta, uprob = TRUE)
plot(imt_theta, y = "theta")

# One left-censoring parameter D, many thresholds u
u <- quantile(sp500, probs = seq(0.2, 0.9, by = 0.1))
imt_theta <- choose_ud(sp500, u = u, D = 1)
plot(imt_theta)
plot(imt_theta, y = "theta")

# One threshold u, many left-censoring parameters D
u <- quantile(sp500, probs = 0.9)
imt_theta <- choose_ud(sp500, u = u, D = 1:5)
plot(imt_theta)
plot(imt_theta, y = "theta")

### Newlyn sea surges

u <- quantile(newlyn, probs = seq(0.1, 0.9, by = 0.1))
imt_theta <- choose_ud(newlyn, u = u, D = 1:5)
plot(imt_theta, uprob = TRUE)

### Cheeseboro wind gusts (a matrix containing some NAs)

probs <- c(seq(0.5, 0.95, by = 0.05), 0.99)
u <- quantile(cheeseboro, probs = probs, na.rm = TRUE)
imt_theta <- choose_ud(cheeseboro, u, D = 1:6)
plot(imt_theta, uprob = FALSE, lwd = 2)

### Uccle July temperatures

probs <- c(seq(0.7, 0.95, by = 0.05), 0.99)
u <- quantile(uccle720m, probs = probs, na.rm = TRUE)
imt_theta <- choose_ud(uccle720m, u, D = 1:5)
```

```
plot(imt_theta, uprob = TRUE, lwd = 2)
```

---

choose_uk — *Threshold u and runs parameter K diagnostic for the K-gaps estimator*

Description

Creates data for a plot to aid the choice of the threshold and run parameter K for the K-gaps estimator (see kgaps). plot.choose_uk creates the plot.

Usage

```
choose_uk(data, u, k = 1, inc_cens = TRUE)
```

Arguments

data
: A numeric vector or numeric matrix of raw data. If data is a matrix then the log-likelihood is constructed as the sum of (independent) contributions from different columns. A common situation is where each column relates to a different year.

  If data contains missing values then split_by_NAs is used to divide the data into sequences of non-missing values.

u, k
: Numeric vectors. u is a vector of extreme value thresholds applied to data. k is a vector of values of the run parameter K, as defined in Suveges and Davison (2010). See kgaps for more details.

  Any values in u that are greater than all the observations in data will be removed without a warning being given.

inc_cens
: A logical scalar indicating whether or not to include contributions from censored inter-exceedance times, relating to the first and last observations. See Attalides (2015) for details.

Details

For each combination of threshold in u and K in k the functions kgaps and kgaps_imt are called in order to estimate θ and to perform the information matrix test of Suveges and Davison (2010).

Value

An object (a list) of class c("choose_uk", "exdex") containing

imt
: an object of class c("kgaps_imt", "exdex") returned from kgaps_imt.

theta
: a length(u) by length(k) matrix. Element (i,j) of theta contains an object (a list) of class c("kgaps", "exdex"), a result of a call kgaps(data, u[j], k[i]) to kgaps.

References

Suveges, M. and Davison, A. C. (2010) Model misspecification in peaks over threshold analysis, *Annals of Applied Statistics*, **4**(1), 203-221. doi:10.1214/09AOAS292

Attalides, N. (2015) Threshold-based extreme value modelling, PhD thesis, University College London. https://discovery.ucl.ac.uk/1471121/1/Nicolas_Attalides_Thesis.pdf

See Also

kgaps for maximum likelihood estimation of the extremal index θ using the K-gaps model.

kgaps_imt for the information matrix test under the K-gaps model

plot.choose_uk to produce the diagnostic plot.

Examples

```
### S&P 500 index

# Multiple thresholds and run parameters
u <- quantile(sp500, probs = seq(0.1, 0.9, by = 0.1))
imt_theta <- choose_uk(sp500, u = u, k = 1:5)
plot(imt_theta)
plot(imt_theta, uprob = TRUE)
plot(imt_theta, y = "theta")

# One run parameter K, many thresholds u
u <- quantile(sp500, probs = seq(0.1, 0.9, by = 0.1))
imt_theta <- choose_uk(sp500, u = u, k = 1)
plot(imt_theta)
plot(imt_theta, y = "theta")

# One threshold u, many run parameters K
u <- quantile(sp500, probs = 0.9)
imt_theta <- choose_uk(sp500, u = u, k = 1:5)
plot(imt_theta)
plot(imt_theta, y = "theta")

### Newlyn sea surges

u <- quantile(newlyn, probs = seq(0.1, 0.9, by = 0.1))
imt_theta <- choose_uk(newlyn, u = u, k = 1:5)
plot(imt_theta, uprob = TRUE)

### Cheeseboro wind gusts (a matrix containing some NAs)

probs <- c(seq(0.5, 0.95, by = 0.05), 0.99)
u <- quantile(cheeseboro, probs = probs, na.rm = TRUE)
imt_theta <- choose_uk(cheeseboro, u, k = 1:6)
plot(imt_theta, uprob = FALSE, lwd = 2)

### Uccle July temperatures
```

```
probs <- c(seq(0.7, 0.95, by = 0.05), 0.99)
u <- quantile(uccle720m, probs = probs, na.rm = TRUE)
imt_theta <- choose_uk(uccle720m, u, k = 1:5)
plot(imt_theta, uprob = TRUE, lwd = 2)
```

---

dgaps — *Maximum likelihood estimation using left-censored inter-exceedances times*

Description

Calculates maximum likelihood estimates of the extremal index θ based on a model for threshold inter-exceedances times of Holesovsky and Fusek (2020). We refer to this as the D-gaps model, because it uses a tuning parameter D, whereas the related K-gaps model of Suveges and Davison (2010) has a tuning parameter K.

Usage

```
dgaps(data, u, D = 1, inc_cens = TRUE)
```

Arguments

`data`	A numeric vector or numeric matrix of raw data. If `data` is a matrix then the log-likelihood is constructed as the sum of (independent) contributions from different columns. A common situation is where each column relates to a different year. If `data` contains missing values then `split_by_NAs` is used to divide the data further into sequences of non-missing values, stored in different columns in a matrix. Again, the log-likelihood is constructed as a sum of contributions from different columns.
`u`	A numeric scalar. Extreme value threshold applied to data.
`D`	A numeric scalar. The censoring parameter D. Threshold inter-exceedances times that are not larger than `D` units are left-censored, occurring with probability $\log(1 - \theta e^{-\theta d})$, where $d = qD$ and q is the probability with which the threshold u is exceeded.
`inc_cens`	A logical scalar indicating whether or not to include contributions from right-censored inter-exceedance times, relating to the first and last observations. It is known that these times are greater than or equal to the time observed. If `data` has multiple columns then there will be right-censored first and last inter-exceedance times for each column.

Details

If `inc_cens = FALSE` then the maximum likelihood estimate of the extremal index θ under the D-gaps model of Holesovsky and Fusek (2020) is calculated. Under this model inter-exceedance times that are less than or equal to D are left-censored, as a strategy to mitigate model mis-specification resulting from the fact that inter-exceedance times that are equal to 0 are expected under the model but only positive inter-exceedance times can be observed in practice.

If inc_cens = TRUE then information from the right-censored first and last inter-exceedance times are also included in the likelihood to be maximized. For an explanation of the idea see Attalides (2015). The form of the log-likelihood is given in the **Details** section of dgaps_stat.

It is possible that the estimate of θ is equal to 1, and also possible that it is equal to 0. dgaps_stat explains the respective properties of the data that cause these events to occur.

Value

An object (a list) of class c("dgaps", "exdex") containing

theta	The maximum likelihood estimate (MLE) of θ.
se	The estimated standard error of the MLE, calculated using an algebraic expression for the observed information. If the estimate of θ is 0 then se is NA.
se_exp	The estimated standard error of the MLE, calculated using an algebraic expression for the expected information. If the estimate of θ is 0 then se_exp is NA. This is provided because cases may be encountered where the observed information is not positive.
ss	The list of summary statistics returned from dgaps_stat.
D, u, inc_cens	The input values of D, u and inc_cens.
max_loglik	The value of the log-likelihood at the MLE.
call	The call to dgaps.

References

Holesovsky, J. and Fusek, M. Estimation of the extremal index using censored distributions. Extremes 23, 197-213 (2020). doi:10.1007/s10687020003743

Suveges, M. and Davison, A. C. (2010) Model misspecification in peaks over threshold analysis, *Annals of Applied Statistics*, **4**(1), 203-221. doi:10.1214/09AOAS292

See Also

dgaps_confint to estimate confidence intervals for θ.

dgaps_methods for S3 methods for "dgaps" objects.

dgaps_imt for the information matrix test, which may be used to inform the choice of the pair (u, D).

choose_ud for a diagnostic plot based on dgaps_imt.

dgaps_stat for the calculation of sufficient statistics for the D-gaps model.

Examples

```
### S&P 500 index

u <- quantile(sp500, probs = 0.60)
theta <- dgaps(sp500, u = u, D = 1)
```

```
theta
summary(theta)
coef(theta)
nobs(theta)
vcov(theta)
logLik(theta)

### Newlyn sea surges

u <- quantile(newlyn, probs = 0.60)
theta <- dgaps(newlyn, u = u, D = 2)
theta
summary(theta)

### Uccle July temperatures

# Using vector input, which merges data from different years
u <- quantile(uccle720$temp, probs = 0.9, na.rm = TRUE)
theta <- dgaps(uccle720$temp, u = u, D = 2)
theta

# Using matrix input to separate data from different years
u <- quantile(uccle720m, probs = 0.9, na.rm = TRUE)
theta <- dgaps(uccle720m, u = u, D = 2)
theta
```

`dgaps_confint` *Confidence intervals for the extremal index θ for* `"dgaps"` *objects*

Description

`confint` method for objects of class `c("dgaps", "exdex")`. Computes confidence intervals for θ based on an object returned from `dgaps`. Two types of interval may be returned: (a) intervals based on approximate large-sample normality of the estimator of θ, which are symmetric about the point estimate, and (b) likelihood-based intervals. The `plot` method plots the log-likelihood for θ, with the required confidence interval indicated on the plot.

Usage

```
## S3 method for class 'dgaps'
confint(
  object,
  parm = "theta",
  level = 0.95,
  interval_type = c("both", "norm", "lik"),
  conf_scale = c("theta", "log"),
  constrain = TRUE,
  se_type = c("observed", "expected"),
  ...
```

```
)

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

## S3 method for class 'confint_dgaps'
print(x, ...)
```

Arguments

`object`	An object of class `c("dgaps", "exdex")`, returned by `dgaps`.
`parm`	Specifies which parameter is to be given a confidence interval. Here there is only one option: the extremal index θ.
`level`	The confidence level required. A numeric scalar in (0, 1).
`interval_type`	A character scalar: `"norm"` for intervals of type (a), `"lik"` for intervals of type (b).
`conf_scale`	A character scalar. If `interval_type = "norm"` then `conf_scale` determines the scale on which we use approximate large-sample normality of the estimator to estimate confidence intervals. If `conf_scale = "theta"` then confidence intervals are estimated for θ directly. If `conf_scale = "log"` then confidence intervals are first estimated for $\log \theta$ and then transformed back to the θ-scale.
`constrain`	A logical scalar. If `constrain = TRUE` then any confidence limits that are greater than 1 are set to 1, that is, they are constrained to lie in (0, 1]. Otherwise, limits that are greater than 1 may be obtained. If `constrain = TRUE` then any lower confidence limits that are less than 0 are set to 0.
`se_type`	A character scalar. Should the confidence intervals for the `interval_type = "norm"` use the estimated standard error based on the observed information or based on the expected information?
`...`	`plot.confint_dgaps`: further arguments passed to `plot.confint`. `print.confint_dgaps`: further arguments passed to `print.default`.
`x`	an object of class `c("confint_dgaps", "exdex")`, a result of a call to `confint.dgaps`.

Details

Two type of interval are calculated: (a) an interval based on the approximate large sample normality of the estimator of θ (if `conf_scale = "theta"`) or of $\log \theta$ (if `conf_scale = "log"`) and (b) a likelihood-based interval, based on the approximate large sample chi-squared, with 1 degree of freedom, distribution of the log-likelihood ratio statistic.

`print.confint_dgaps` prints the matrix of confidence intervals for θ.

Value

A list of class c("confint_dgaps", "exdex") containing the following components.

`cis` A matrix with columns giving the lower and upper confidence limits. These are labelled as (1 - level)/2 and 1 - (1 - level)/2 in % (by default 2.5% and 97.5%). The row names indicate the type of interval: `norm` for intervals based on large sample normality and `lik` for likelihood-based intervals.

`call` The call to `spm`.

`object` The input object `object`.

`level` The input `level`.

`plot.confint_dgaps`: nothing is returned.

`print.confint_dgaps`: the argument `x`, invisibly.

References

Holesovsky, J. and Fusek, M. Estimation of the extremal index using censored distributions. Extremes 23, 197-213 (2020). doi:10.1007/s10687020003743

See Also

`dgaps` for estimation of the extremal index θ using a semiparametric maxima method.

Examples

```
u <- quantile(newlyn, probs = 0.90)
theta <- dgaps(newlyn, u)
cis <- confint(theta)
cis
plot(cis)
```

`dgaps_imt` *Information matrix test under the D-gaps model*

Description

Performs an information matrix test (IMT) to diagnose misspecification of the D-gaps model of Holesovsky and Fusek (2020).

Usage

```
dgaps_imt(data, u, D = 1, inc_cens = TRUE)
```

Arguments

`data` A numeric vector or numeric matrix of raw data. If `data` is a matrix then the log-likelihood is constructed as the sum of (independent) contributions from different columns. A common situation is where each column relates to a different year.

If `data` contains missing values then `split_by_NAs` is used to divide the data into sequences of non-missing values.

u, D
: Numeric vectors. u is a vector of extreme value thresholds applied to data. D is a vector of values of the left-censoring parameter D, as defined in Holesovsky and Fusek (2020). See dgaps.

  Any values in u that are greater than all the observations in data will be removed without a warning being given.

inc_cens
: A logical scalar indicating whether or not to include contributions from right-censored inter-exceedance times, relating to the first and last observations. See dgaps.

Details

The general approach follows Suveges and Davison (2010). The D-gaps IMT is performed a over grid of all combinations of threshold and D in the vectors u and D. If the estimate of θ is 0 then the IMT statistic, and its associated p-value is NA.

Value

An object (a list) of class c("dgaps_imt", "exdex") containing

imt
: A length(u) by length(D) numeric matrix. Column i contains, for D = D[i], the values of the information matrix test statistic for the set of thresholds in u. The column names are the values in D. The row names are the approximate empirical percentage quantile levels of the thresholds in u.

p
: A length(u) by length(D) numeric matrix containing the corresponding p-values for the test.

theta
: A length(u) by length(D) numeric matrix containing the corresponding estimates of θ.

u, D
: The input u and D.

References

Holesovsky, J. and Fusek, M. Estimation of the extremal index using censored distributions. Extremes 23, 197-213 (2020). doi:10.1007/s10687020003743

Suveges, M. and Davison, A. C. (2010) Model misspecification in peaks over threshold analysis, *Annals of Applied Statistics*, **4**(1), 203-221. doi:10.1214/09AOAS292

See Also

dgaps for maximum likelihood estimation of the extremal index θ using the D-gaps model.

Examples

```
### Newlyn sea surges

u <- quantile(newlyn, probs = seq(0.1, 0.9, by = 0.1))
imt <- dgaps_imt(newlyn, u = u, D = 1:5)

### S&P 500 index
```

```
u <- quantile(sp500, probs = seq(0.1, 0.9, by = 0.1))
imt <- dgaps_imt(sp500, u = u, D = 1:5)

### Cheeseboro wind gusts (a matrix containing some NAs)

probs <- c(seq(0.5, 0.98, by = 0.025), 0.99)
u <- quantile(cheeseboro, probs = probs, na.rm = TRUE)
imt <- dgaps_imt(cheeseboro, u = u, D = 1:5)

### Uccle July temperatures

probs <- c(seq(0.7, 0.98, by = 0.025), 0.99)
u <- quantile(uccle720m, probs = probs, na.rm = TRUE)
imt <- dgaps_imt(uccle720m, u = u, D = 1:5)
```

---

dgaps_imt_stat *Statistics for the D-gaps information matrix test*

Description

Calculates the components required to calculate the value of the information matrix test under the D-gaps model, using vector data input. Called by `dgaps_imt`.

Usage

```
dgaps_imt_stat(data, theta, u, D = 1, inc_cens = TRUE)
```

Arguments

`data`	A numeric vector of raw data. Missing values are allowed, but they should not appear between non-missing values, that is, they only be located at the start and end of the vector. Missing values are omitted using `na.omit`.
`theta`	A numeric scalar. An estimate of the extremal index θ, produced by `dgaps`.
`u`	A numeric scalar. Extreme value threshold applied to data.
`D`	A numeric scalar. The censoring parameter D. Threshold inter-exceedances times that are not larger than `D` units are left-censored, occurring with probability $\log(1-\theta e^{-\theta d})$, where $d = qD$ and q is the probability with which the threshold u is exceeded.
`inc_cens`	A logical scalar indicating whether or not to include contributions from right-censored inter-exceedance times, relating to the first and last observations. See `dgaps`.

Value

A list relating the quantities given on pages 18-19 of Suveges and Davison (2010). All but the last component are vectors giving the contribution to the quantity from each D-gap, evaluated at the input value `theta` of θ.

ldj	the derivative of the log-likelihood with respect to θ (the score)
Ij	the observed information
Jj	the square of the score
dj	Jj - Ij
Ddj	the derivative of Jj - Ij with respect to θ
n_dgaps	the number of D-gaps that contribute to the log-likelihood.

References

Holesovsky, J. and Fusek, M. Estimation of the extremal index using censored distributions. Extremes 23, 197-213 (2020). doi:10.1007/s10687020003743

---

dgaps_methods *Methods for objects of class* "dgaps"

Description

Methods for objects of class c("dgaps", "exdex") returned from dgaps.

Usage

```
## S3 method for class 'dgaps'
coef(object, ...)

## S3 method for class 'dgaps'
vcov(object, type = c("observed", "expected"), ...)

## S3 method for class 'dgaps'
nobs(object, ...)

## S3 method for class 'dgaps'
logLik(object, ...)

## S3 method for class 'dgaps'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'dgaps'
summary(
  object,
  se_type = c("observed", "expected"),
  digits = max(3, getOption("digits") - 3L),
  ...
)

## S3 method for class 'summary.dgaps'
print(x, ...)
```

Arguments

object	and object of class c("dgaps", "exdex") returned from dgaps.
...	For print.summary.dgaps, additional arguments passed to print.default.
type	A character scalar. Should the estimate of the variance be based on the observed information or the expected information?
x	print.dgaps. An object of class c("dgaps", "exdex"), a result of a call to dgaps. print.summary.dgaps. An object of class "summary.dgaps", a result of a call to summary.dgaps.
digits	print.dgaps. The argument digits to print.default. summary.dgaps. An integer. Used for number formatting with signif.
se_type	A character scalar. Should the estimate of the standard error be based on the observed information or the expected information?

Value

coef.dgaps. A numeric scalar: the estimate of the extremal index θ.

vcov.dgaps. A 1×1 numeric matrix containing the estimated variance of the estimator.

nobs.dgaps. A numeric scalar: the number of inter-exceedance times used in the fit. If x$inc_cens = TRUE then this includes up to 2 censored observations.

logLik.dgaps. An object of class "logLik": a numeric scalar with value equal to the maximised log-likelihood. The returned object also has attributes nobs, the numbers of K-gaps that contribute to the log-likelihood and "df", which is equal to the number of total number of parameters estimated (1).

print.dgaps. The argument x, invisibly.

summary.dgaps. Returns a list containing the list element object$call and a numeric matrix summary giving the estimate of the extremal index θ and the estimated standard error (Std. Error).

print.summary.dgaps. The argument x, invisibly.

Examples

See the examples in dgaps.

See Also

dgaps for maximum likelihood estimation of the extremal index θ using the K-gaps model.

confint.dgaps for confidence intervals for θ.

`dgaps_stat` *Sufficient statistics for the left-censored inter-exceedances time model*

Description

Calculates sufficient statistics for the the left-censored inter-exceedances time D-gaps model for the extremal index θ.

Usage

```
dgaps_stat(data, u, q_u, D = 1, inc_cens = TRUE)
```

Arguments

`data`	A numeric vector of raw data. No missing values are allowed.
`u`	A numeric scalar. Extreme value threshold applied to data.
`q_u`	A numeric scalar. An estimate of the probability with which the threshold `u` is exceeded. If `q_u` is missing then it is calculated using `mean(data > u, na.rm = TRUE)`.
`D`	A numeric scalar. Run parameter K, as defined in Suveges and Davison (2010). Threshold inter-exceedances times that are not larger than `k` units are assigned to the same cluster, resulting in a K-gap equal to zero. Specifically, the K-gap S corresponding to an inter-exceedance time of T is given by $S = \max(T-K, 0)$.
`inc_cens`	A logical scalar indicating whether or not to include contributions from right-censored inter-exceedance times relating to the first and last observation. It is known that these times are greater than or equal to the time observed. See Attalides (2015) for details.

Details

The sample inter-exceedance times are $T_0, T_1, ..., T_{N-1}, T_N$, where $T_1, ..., T_{N-1}$ are uncensored and T_0 and T_N are right-censored. Under the assumption that the inter-exceedance times are independent, the log-likelihood of the D-gaps model is given by

$$l(\theta; T_0, \ldots, T_N) = N_0 \log(1 - \theta e^{-\theta d}) + 2N_1 \log \theta - \theta q(I_0 T_0 + \cdots + I_N T_N),$$

where

- q is the threshold exceedance probability, estimated by the proportion of threshold exceedances,
- $d = qD$,
- $I_j = 1$ if $T_j > D$ and $I_j = 0$ otherwise,
- N_0 is the number of sample inter-exceedance times that are left-censored, that is, are less than or equal to D,
- (apart from an adjustment for the contributions of T_0 and T_N) N_1 is the number of inter-exceedance times that are uncensored, that is, are greater than D,

- specifically, if `inc_cens = TRUE` then N_1 is equal to the number of $T_1, ..., T_{N-1}$ that are uncensored plus $(I_0 + I_N)/2$.

The differing treatment of uncensored and censored K-gaps reflects differing contributions to the likelihood. Right-censored inter-exceedance times whose observed values are less than or equal to D add no information to the likelihood because we do not know to which part of the likelihood they should contribute.

If $N_1 = 0$ then we are in the degenerate case where there is one cluster (all inter-exceedance times are left-censored) and the likelihood is maximized at $\theta = 0$.

If $N_0 = 0$ then all exceedances occur singly (no inter-exceedance times are left-censored) and the likelihood is maximized at $\theta = 1$.

Value

A list containing the sufficient statistics, with components

`N0`	the number of left-censored inter-exceedance times.
`N1`	contribution from inter-exceedance times that are not left-censored (see **Details**).
`sum_qtd`	the sum of the (scaled) inter-exceedance times that are not left-censored, that is, $q(I_0 T_0 + \cdots + I_N T_N)$, where q is estimated by the proportion of threshold exceedances.
`n_dgaps`	the number of inter-exceedances that contribute to the log-likelihood.
`q_u`	the sample proportion of values that exceed the threshold.
`D`	the input value of `D`.

References

Holesovsky, J. and Fusek, M. Estimation of the extremal index using censored distributions. Extremes 23, 197-213 (2020). doi:10.1007/s10687020003743

Attalides, N. (2015) Threshold-based extreme value modelling, PhD thesis, University College London. `https://discovery.ucl.ac.uk/1471121/1/Nicolas_Attalides_Thesis.pdf`

See Also

`dgaps` for maximum likelihood estimation of the extremal index θ using the D-gaps model.

Examples

```
u <- quantile(newlyn, probs = 0.90)
dgaps_stat(newlyn, u = u, D = 1)
```

`iwls` *Iterated weighted least squares estimation of the extremal index*

Description

Estimates the extremal index θ using the iterated weighted least squares method of Suveges (2007). At the moment no estimates of uncertainty are provided.

Usage

```
iwls(data, u, maxit = 100)
```

Arguments

`data`	A numeric vector of raw data. No missing values are allowed.
`u`	A numeric scalar. Extreme value threshold applied to data.
`maxit`	A numeric scalar. The maximum number of iterations.

Details

The iterated weighted least squares algorithm on page 46 of Suveges (2007) is used to estimate the value of the extremal index. This approach uses the time *gaps* between successive exceedances in the data `data` of the threshold `u`. The ith gap is defined as $T_i - 1$, where T_i is the difference in the occurrence time of exceedance i and exceedance $i + 1$. Therefore, threshold exceedances at adjacent time points produce a gap of zero.

The model underlying this approach is an exponential-point mas mixture for *scaled gaps*, that is, gaps multiplied by the proportion of values in `data` that exceed `u`. Under this model scaled gaps are zero ('within-cluster' inter-exceedance times) with probability $1 - \theta$ and otherwise ('between-cluster' inter-exceedance times) follow an exponential distribution with mean $1/\theta$. The estimation method is based on fitting the 'broken stick' model of Ferro (2003) to an exponential quantile-quantile plot of all of the scaled gaps. Specifically, the broken stick is a horizontal line and a line with gradient $1/\theta$ which intersect at $(-\log\theta, 0)$. The algorithm on page 46 of Suveges (2007) uses a weighted least squares minimization applied to the exponential part of this model to seek a compromise between the role of θ as the proportion of inter-exceedance times that are between-cluster and the reciprocal of the mean of an exponential distribution for these inter-exceedance times. The weights (see Ferro (2003)) are based on the variances of order statistics of a standard exponential sample: larger order statistics have larger sampling variabilities and therefore receive smaller weight than smaller order statistics.

Note that in step (1) of the algorithm on page 46 of Suveges there is a typo: $N_c + 1$ should be N, where N is the number of threshold exceedances. Also, the gaps are scaled as detailed above, not by their mean.

Value

An object (a list) of class `"iwls", "exdex"` containing

`theta`	The estimate of θ.

conv	A convergence indicator: 0 indicates successful convergence; 1 indicates that maxit has been reached.
niter	The number of iterations performed.
n_gaps	The number of time gaps between successive exceedances.
call	The call to iwls.

References

Suveges, M. (2007) Likelihood estimation of the extremal index. *Extremes*, **10**, 41-55. doi:10.1007/s1068700700342

Ferro, C.A.T. (2003) Statistical methods for clusters of extreme values. Ph.D. thesis, Lancaster University.

See Also

iwls_methods for S3 methods for "iwls" objects.

Examples

```
### S&P 500 index

u <- quantile(sp500, probs = 0.60)
theta <- iwls(sp500, u)
theta
coef(theta)
nobs(theta)

### Newlyn sea surges

u <- quantile(newlyn, probs = 0.90)
theta <- iwls(newlyn, u)
theta
```

---

iwls_methods *Methods for objects of class* "iwls"

Description

Methods for objects of class c("iwls", "exdex") returned from iwls.

Usage

```
## S3 method for class 'iwls'
coef(object, ...)

## S3 method for class 'iwls'
nobs(object, ...)
```

```
## S3 method for class 'iwls'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

`object`	and object of class `c("iwls", "exdex")` returned from `iwls`.
`...`	Further arguments. None are used.
`x`	an object of class `c("iwls", "exdex")`, a result of a call to `iwls`.
`digits`	The argument `digits` to `print.default`.

Details

`print.iwls` prints the original call to `iwls` and the estimate of the extremal index θ.

Value

`coef.iwls`. A numeric scalar: the estimate of the extremal index θ.

`nobs.iwls`. A numeric scalar: the number of inter-exceedance times used in the fit.

`print.iwls`. The argument `x`, invisibly.

Examples

See the examples in `iwls`.

See Also

`iwls` for maximum likelihood estimation of the extremal index θ using the K-gaps model.

---

kgaps — *Maximum likelihood estimation for the K-gaps model*

Description

Calculates maximum likelihood estimates of the extremal index θ based on the K-gaps model for threshold inter-exceedances times of Suveges and Davison (2010).

Usage

```
kgaps(data, u, k = 1, inc_cens = TRUE)
```

Arguments

data	A numeric vector or numeric matrix of raw data. If data is a matrix then the log-likelihood is constructed as the sum of (independent) contributions from different columns. A common situation is where each column relates to a different year. If data contains missing values then split_by_NAs is used to divide the data further into sequences of non-missing values, stored in different columns in a matrix. Again, the log-likelihood is constructed as a sum of contributions from different columns.
u	A numeric scalar. Extreme value threshold applied to data.
k	A non-negative numeric scalar. Run parameter K, as defined in Suveges and Davison (2010). Threshold inter-exceedances times that are not larger than k units are assigned to the same cluster, resulting in a K-gap equal to zero. Specifically, the K-gap S corresponding to an inter-exceedance time of T is given by $S = \max(T - K, 0)$. In practice, k should be no smaller than 1, because when k is less than 1 the estimate of θ is always equal to 1.
inc_cens	A logical scalar indicating whether or not to include contributions from right-censored inter-exceedance times, relating to the first and last observations. It is known that these times are greater than or equal to the time observed. See Attalides (2015) for details. If data has multiple columns then there will be right-censored first and last inter-exceedance times for each column.

Details

If inc_cens = FALSE then the maximum likelihood estimate of the extremal index θ under the K-gaps model of Suveges and Davison (2010) is calculated.

If inc_cens = TRUE then information from right-censored first and last inter-exceedance times is also included in the likelihood to be maximized, following Attalides (2015). The form of the log-likelihood is given in the **Details** section of kgaps_stat.

It is possible that the estimate of θ is equal to 1, and also possible that it is equal to 0. kgaps_stat explains the respective properties of the data that cause these events to occur.

Value

An object (a list) of class c("kgaps", "exdex") containing

theta	The maximum likelihood estimate (MLE) of θ.
se	The estimated standard error of the MLE, calculated using an algebraic expression for the observed information. If k = 0 then se is returned as 0.
se_exp	The estimated standard error of the MLE, calculated using an algebraic expression for the expected information. If the estimate of θ is 0 or 1 then se_exp is NA.
ss	The list of summary statistics returned from kgaps_stat.
k, u, inc_cens	The input values of k, u and inc_cens.
max_loglik	The value of the log-likelihood at the MLE.
call	The call to kgaps.

References

Suveges, M. and Davison, A. C. (2010) Model misspecification in peaks over threshold analysis, *Annals of Applied Statistics*, **4**(1), 203-221. doi:10.1214/09AOAS292

Attalides, N. (2015) Threshold-based extreme value modelling, PhD thesis, University College London. https://discovery.ucl.ac.uk/1471121/1/Nicolas_Attalides_Thesis.pdf

See Also

kgaps_confint to estimate confidence intervals for θ.

kgaps_methods for S3 methods for "kgaps" objects.

kgaps_imt for the information matrix test, which may be used to inform the choice of the pair (u, k).

choose_uk for a diagnostic plot based on kgaps_imt.

kgaps_stat for the calculation of sufficient statistics for the K-gaps model.

kgaps_post in the revdbayes package for Bayesian inference about θ using the K-gaps model.

Examples

```
### S&P 500 index

u <- quantile(sp500, probs = 0.60)
theta <- kgaps(sp500, u)
theta
summary(theta)
coef(theta)
nobs(theta)
vcov(theta)
logLik(theta)

### Newlyn sea surges

u <- quantile(newlyn, probs = 0.60)
theta <- kgaps(newlyn, u, k = 2)
theta
summary(theta)

### Cheeseboro wind gusts

theta <- kgaps(cheeseboro, 45, k = 3)
theta
summary(theta)
```

kgaps_confint — *Confidence intervals for the extremal index θ for "kgaps" objects*

Description

confint method for objects of class c("kgaps", "exdex"). Computes confidence intervals for θ based on an object returned from kgaps. Two types of interval may be returned: (a) intervals based on approximate large-sample normality of the estimator of θ, which are symmetric about the point estimate, and (b) likelihood-based intervals. The plot method plots the log-likelihood for θ, with the required confidence interval indicated on the plot.

Usage

```
## S3 method for class 'kgaps'
confint(
  object,
  parm = "theta",
  level = 0.95,
  interval_type = c("both", "norm", "lik"),
  conf_scale = c("theta", "log"),
  constrain = TRUE,
  se_type = c("observed", "expected"),
  ...
)

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

## S3 method for class 'confint_kgaps'
print(x, ...)
```

Arguments

object — An object of class c("kgaps", "exdex"), returned by kgaps.

parm — Specifies which parameter is to be given a confidence interval. Here there is only one option: the extremal index θ.

level — The confidence level required. A numeric scalar in (0, 1).

interval_type — A character scalar: "norm" for intervals of type (a), "lik" for intervals of type (b).

conf_scale — A character scalar. If interval_type = "norm" then conf_scale determines the scale on which we use approximate large-sample normality of the estimator to estimate confidence intervals.

If conf_scale = "theta" then confidence intervals are estimated for θ directly. If conf_scale = "log" then confidence intervals are first estimated for $\log\theta$ and then transformed back to the θ-scale.

constrain — A logical scalar. If constrain = TRUE then any confidence limits that are greater than 1 are set to 1, that is, they are constrained to lie in (0, 1]. Otherwise, limits that are greater than 1 may be obtained. If constrain = TRUE then any lower confidence limits that are less than 0 are set to 0.

se_type — A character scalar. Should the confidence intervals for the interval_type = "norm" use the estimated standard error based on the observed information or based on the expected information?

... — plot.confint_kgaps: further arguments passed to plot.confint.

print.confint_kgaps: further arguments passed to print.default.

x — an object of class c("confint_kgaps", "exdex"), a result of a call to confint.kgaps.

Details

Two type of interval are calculated: (a) an interval based on the approximate large sample normality of the estimator of θ (if conf_scale = "theta") or of $\log\theta$ (if conf_scale = "log") and (b) a likelihood-based interval, based on the approximate large sample chi-squared, with 1 degree of freedom, distribution of the log-likelihood ratio statistic.

print.confint_kgaps prints the matrix of confidence intervals for θ.

Value

A list of class c("confint_kgaps", "exdex") containing the following components.

cis — A matrix with columns giving the lower and upper confidence limits. These are labelled as (1 - level)/2 and 1 - (1 - level)/2 in % (by default 2.5% and 97.5%). The row names indicate the type of interval: norm for intervals based on large sample normality and lik for likelihood-based intervals. If object$k = 0 then both confidence limits are returned as being equal to the point estimate of θ.

call — The call to spm.

object — The input object object.

level — The input level.

plot.confint_kgaps: nothing is returned. If x$object$k = 0 then no plot is produced.

print.confint_kgaps: the argument x, invisibly.

References

Suveges, M. and Davison, A. C. (2010) Model misspecification in peaks over threshold analysis, *Annals of Applied Statistics*, **4**(1), 203-221. doi:10.1214/09AOAS292

See Also

kgaps for estimation of the extremal index θ using a semiparametric maxima method.

Examples

```
u <- quantile(newlyn, probs = 0.90)
theta <- kgaps(newlyn, u)
cis <- confint(theta)
cis
plot(cis)
```

---

kgaps_imt *Information matrix test under the K-gaps model*

Description

Performs the information matrix test (IMT) of Suveges and Davison (2010) to diagnose misspecification of the K-gaps model.

Usage

```
kgaps_imt(data, u, k = 1, inc_cens = TRUE)
```

Arguments

`data`	A numeric vector or numeric matrix of raw data. If `data` is a matrix then the log-likelihood is constructed as the sum of (independent) contributions from different columns. A common situation is where each column relates to a different year. If `data` contains missing values then `split_by_NAs` is used to divide the data into sequences of non-missing values.
`u, k`	Numeric vectors. `u` is a vector of extreme value thresholds applied to data. `k` is a vector of values of the run parameter K, as defined in Suveges and Davison (2010). See `kgaps` for more details. Any values in `u` that are greater than all the observations in `data` will be removed without a warning being given.
`inc_cens`	A logical scalar indicating whether or not to include contributions from censored inter-exceedance times, relating to the first and last observations. See Attalides (2015) for details.

Details

The K-gaps IMT is performed a over grid of all combinations of threshold and K in the vectors `u` and `k`. If the estimate of θ is 0 then the IMT statistic, and its associated p-value is `NA`.

For details of the IMT see Suveges and Davison (2010). There are some typing errors on pages 18-19 that have been corrected in producing the code: the penultimate term inside `{...}` in the middle equation on page 18 should be $(c_j(K))^2$, as should the penultimate term in the first equation on page 19; the `{...}` bracket should be squared in the 4th equation on page 19; the factor n should be $N - 1$ in the final equation on page 19.

Value

An object (a list) of class `c("kgaps_imt", "exdex")` containing

`imt`	A `length(u)` by `length(k)` numeric matrix. Column i contains, for K = `k[i]`, the values of the information matrix test statistic for the set of thresholds in `u`. The column names are the values in `k`. The row names are the approximate empirical percentage quantile levels of the thresholds in `u`.
`p`	A `length(u)` by `length(k)` numeric matrix containing the corresponding p-values for the test.
`theta`	A `length(u)` by `length(k)` numeric matrix containing the corresponding estimates of θ.
`u, k`	The input `u` and `k`.

References

Suveges, M. and Davison, A. C. (2010) Model misspecification in peaks over threshold analysis, *Annals of Applied Statistics*, **4**(1), 203-221. doi:10.1214/09AOAS292

Attalides, N. (2015) Threshold-based extreme value modelling, PhD thesis, University College London. https://discovery.ucl.ac.uk/1471121/1/Nicolas_Attalides_Thesis.pdf

See Also

`kgaps` for maximum likelihood estimation of the extremal index θ using the K-gaps model.

`choose_uk` for graphical diagnostic to aid the choice of the threshold u and the run parameter K.

Examples

```
### Newlyn sea surges

u <- quantile(newlyn, probs = seq(0.1, 0.9, by = 0.1))
imt <- kgaps_imt(newlyn, u = u, k = 1:5)

### S&P 500 index

u <- quantile(sp500, probs = seq(0.1, 0.9, by = 0.1))
imt <- kgaps_imt(sp500, u = u, k = 1:5)

### Cheeseboro wind gusts (a matrix containing some NAs)

probs <- c(seq(0.5, 0.98, by = 0.025), 0.99)
u <- quantile(cheeseboro, probs = probs, na.rm = TRUE)
imt <- kgaps_imt(cheeseboro, u = u, k = 1:5)
```

kgaps_imt_stat *Statistics for the information matrix test*

Description

Calculates the components required to calculate the value of the information matrix test under the K-gaps model, using vector data input. Called by kgaps_imt.

Usage

```
kgaps_imt_stat(data, theta, u, k = 1, inc_cens = TRUE)
```

Arguments

data	A numeric vector of raw data. Missing values are allowed, but they should not appear between non-missing values, that is, they only be located at the start and end of the vector. Missing values are omitted using na.omit.
theta	A numeric scalar. An estimate of the extremal index θ, produced by kgaps.
u	A numeric scalar. Extreme value threshold applied to data.
k	A numeric scalar. Run parameter K, as defined in Suveges and Davison (2010). Threshold inter-exceedances times that are not larger than k units are assigned to the same cluster, resulting in a K-gap equal to zero. Specifically, the K-gap S corresponding to an inter-exceedance time of T is given by $S = \max(T - K, 0)$.
inc_cens	A logical scalar indicating whether or not to include contributions from censored inter-exceedance times relating to the first and last observation. See Attalides (2015) for details.

Value

A list relating the quantities given on pages 18-19 of Suveges and Davison (2010). All but the last component are vectors giving the contribution to the quantity from each K-gap, evaluated at the input value theta of θ.

ldj	the derivative of the log-likelihood with respect to θ (the score)
Ij	the observed information
Jj	the square of the score
dj	Jj - Ij
Ddj	the derivative of Jj - Ij with respect to θ
n_kgaps	the number of K-gaps that contribute to the log-likelihood.

References

Suveges, M. and Davison, A. C. (2010) Model misspecification in peaks over threshold analysis, *Annals of Applied Statistics*, **4**(1), 203-221. doi:10.1214/09AOAS292

Attalides, N. (2015) Threshold-based extreme value modelling, PhD thesis, University College London. https://discovery.ucl.ac.uk/1471121/1/Nicolas_Attalides_Thesis.pdf

kgaps_methods *Methods for objects of class* "kgaps"

Description

Methods for objects of class c("kgaps", "exdex") returned from kgaps.

Usage

```
## S3 method for class 'kgaps'
coef(object, ...)

## S3 method for class 'kgaps'
vcov(object, type = c("observed", "expected"), ...)

## S3 method for class 'kgaps'
nobs(object, ...)

## S3 method for class 'kgaps'
logLik(object, ...)

## S3 method for class 'kgaps'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'kgaps'
summary(
  object,
  se_type = c("observed", "expected"),
  digits = max(3, getOption("digits") - 3L),
  ...
)

## S3 method for class 'summary.kgaps'
print(x, ...)
```

Arguments

object — and object of class c("kgaps", "exdex") returned from kgaps.

... — For print.summary.kgaps, additional arguments passed to print.default.

type — A character scalar. Should the estimate of the variance be based on the observed information or the expected information?

x — print.kgaps. An object of class c("kgaps", "exdex"), a result of a call to kgaps.

print.summary.kgaps. An object of class "summary.kgaps", a result of a call to summary.kgaps.

digits
: print.kgaps. The argument digits to print.default.

  summary.kgaps. An integer. Used for number formatting with signif.

se_type
: A character scalar. Should the estimate of the standard error be based on the observed information or the expected information?

Value

coef.kgaps. A numeric scalar: the estimate of the extremal index θ.

vcov.kgaps. A 1×1 numeric matrix containing the estimated variance of the estimator.

nobs.kgaps. A numeric scalar: the number of inter-exceedance times used in the fit. If x$inc_cens = TRUE then this includes up to 2 censored observations.

logLik.kgaps. An object of class "logLik": a numeric scalar with value equal to the maximised log-likelihood. The returned object also has attributes nobs, the numbers of K-gaps that contribute to the log-likelihood and "df", which is equal to the number of total number of parameters estimated (1).

print.kgaps. The argument x, invisibly.

summary.kgaps. Returns a list containing the list element object$call and a numeric matrix summary giving the estimate of the extremal index θ and the estimated standard error (Std. Error).

print.summary.kgaps. The argument x, invisibly.

Examples

See the examples in kgaps.

See Also

kgaps for maximum likelihood estimation of the extremal index θ using the K-gaps model.

confint.kgaps for confidence intervals for θ.

---

kgaps_stat *Sufficient statistics for the K-gaps model*

Description

Calculates sufficient statistics for the K-gaps model for the extremal index θ. Called by kgaps.

Usage

```
kgaps_stat(data, u, q_u, k = 1, inc_cens = TRUE)
```

Arguments

data	A numeric vector of raw data.
u	A numeric scalar. Extreme value threshold applied to data.
q_u	A numeric scalar. An estimate of the probability with which the threshold u is exceeded. If q_u is missing then it is calculated using mean(data > u, na.rm = TRUE).
k	A numeric scalar. Run parameter K, as defined in Suveges and Davison (2010). Threshold inter-exceedances times that are not larger than k units are assigned to the same cluster, resulting in a K-gap equal to zero. Specifically, the K-gap S corresponding to an inter-exceedance time of T is given by $S = \max(T - K, 0)$.
inc_cens	A logical scalar indicating whether or not to include contributions from right-censored inter-exceedance times relating to the first and last observation. It is known that these times are greater than or equal to the time observed. See Attalides (2015) for details.

Details

The sample K-gaps are $S_0, S_1, ..., S_{N-1}, S_N$, where $S_1, ..., S_{N-1}$ are uncensored and S_0 and S_N are right-censored. Under the assumption that the K-gaps are independent, the log-likelihood of the K-gaps model is given by

$$l(\theta; S_0, \ldots, S_N) = N_0 \log(1-\theta) + 2N_1 \log\theta - \theta q(S_0 + \cdots + S_N),$$

where

- q is the threshold exceedance probability, estimated by the proportion of threshold exceedances,
- N_0 is the number of uncensored sample K-gaps that are equal to zero,
- (apart from an adjustment for the contributions of S_0 and S_N) N_1 is the number of positive sample K-gaps,
- specifically, if inc_cens = TRUE then N_1 is equal to the number of $S_1, ..., S_{N-1}$ that are positive plus $(I_0 + I_N)/2$, where $I_0 = 1$ if S_0 is greater than zero and $I_0 = 0$ otherwise, and similarly for I_N.

The differing treatment of uncensored and right-censored K-gaps reflects differing contributions to the likelihood. Right-censored K-gaps that are equal to zero add no information to the likelihood. For full details see Suveges and Davison (2010) and Attalides (2015).

If $N_1 = 0$ then we are in the degenerate case where there is one cluster (all K-gaps are zero) and the likelihood is maximized at $\theta = 0$.

If $N_0 = 0$ then all exceedances occur singly (all K-gaps are positive) and the likelihood is maximized at $\theta = 1$.

Value

A list containing the sufficient statistics, with components

N0	the number of zero K-gaps.
N1	contribution from non-zero K-gaps (see **Details**).

sum_qs — the sum of the (scaled) K-gaps, that is, $q(S_0 + \cdots + S_N)$, where q is estimated by the proportion of threshold exceedances.

n_kgaps — the number of K-gaps that contribute to the log-likelihood.

References

Suveges, M. and Davison, A. C. (2010) Model misspecification in peaks over threshold analysis, *Annals of Applied Statistics*, **4**(1), 203-221. doi:10.1214/09AOAS292

Attalides, N. (2015) Threshold-based extreme value modelling, PhD thesis, University College London. https://discovery.ucl.ac.uk/1471121/1/Nicolas_Attalides_Thesis.pdf

See Also

kgaps for maximum likelihood estimation of the extremal index θ using the K-gaps model.

Examples

```
u <- quantile(newlyn, probs = 0.90)
kgaps_stat(newlyn, u)
```

---

newlyn — *Newlyn sea surges*

Description

The vector newlyn contains 2894 maximum sea-surges measured at Newlyn, Cornwall, UK over the period 1971-1976. The observations are the maximum hourly sea-surge heights over contiguous 15-hour time periods.

Usage

```
newlyn
```

Format

A vector of length 2894.

Source

Coles, S.G. (1991) Modelling extreme multivariate events. PhD thesis, University of Sheffield, U.K.

References

Fawcett, L. and Walshaw, D. (2012) Estimating return levels from serially dependent extremes. *Environmetrics*, **23**(3), 272-283. doi:10.1002/env.2133

Northrop, P. J. (2015) An efficient semiparametric maxima estimator of the extremal index. *Extremes*, **18**, 585-603. doi:10.1007/s1068701502215

`plot.choose_b` *Plot block length diagnostic for the semiparametric maxima estimator*

Description

`plot` method for objects inheriting from class `"choose_b"`, returned from `choose_b`

Usage

```
## S3 method for class 'choose_b'
plot(
  x,
  y,
  ...,
  estimator = c("N2015", "BB2018"),
  maxima = c("sliding", "disjoint")
)
```

Arguments

`x`	an object of class `c("choose_b", "exdex")`, a result of a call to `choose_b`.
`y`	Not used.
`...`	Additional arguments passed on to `matplot` and/or `axis`.
`estimator`	Choice of estimator: `"N2015"` for Northrop (2015), `"BB2018"` for Berghaus and Bucher (2018). See `spm` for details.
`maxima`	Should the estimator be based on sliding or disjoint maxima?

Details

Produces a simple diagnostic plot to aid the choice of block length b based on the object returned from `choose_b`. Estimates of b and approximate `conf`% confidence intervals are plotted against the value of b used to produce each estimate. The type of confidence interval is determined by the arguments `interval_type`, `conf_scale` and `type` provided in the call to `choose_b`.

Value

Nothing is returned.

Examples

See the examples in `choose_b`.

References

Northrop, P. J. (2015) An efficient semiparametric maxima estimator of the extremal index. *Extremes* **18**(4), 585-603. doi:10.1007/s1068701502215

Berghaus, B., Bucher, A. (2018) Weak convergence of a pseudo maximum likelihood estimator for the extremal index. *Ann. Statist.* **46**(5), 2307-2335. doi:10.1214/17AOS1621

See Also

choose_b.

---

plot.choose_ud *Plot threshold u and runs parameter D diagnostic for the D-gaps estimator*

Description

plot method for objects inheriting from class "choose_ud", returned from choose_ud

Usage

```
## S3 method for class 'choose_ud'
plot(
  x,
  y = c("imts", "theta"),
  level = 0.95,
  interval_type = c("norm", "lik"),
  conf_scale = c("theta", "log"),
  alpha = 0.05,
  constrain = TRUE,
  for_abline = list(lty = 2, lwd = 1, col = 1),
  digits = 3,
  uprob = FALSE,
  leg_pos = if (y == "imts") "topright" else "topleft",
  ...
)
```

Arguments

x	an object of class c("choose_ud", "exdex"), a result of a call to choose_ud.
y	A character scalar indicating what should be plotted on the vertical axes of the plot: information matrix test statistics (IMTS) if y = "imts" and estimates of θ if y = "theta". If y = "theta", and either x$u or x$D have length one, then 100level% confidence intervals are added to the plot.
level	A numeric scalar in (0, 1). The confidence level used in calculating confidence intervals for θ. Only relevant if y = "theta" and either x$u or x$D have length one.
interval_type	A character scalar. The type of confidence interval to be plotted, if y = "theta". See confint.dgaps.
conf_scale	A character scalar. If interval_type = "norm" then conf_scale determines the scale on which we use approximate large-sample normality of the estimator to estimate confidence intervals. See confint.dgaps.
alpha	A numeric vector with entries in (0, 1). The size of the test to be performed.

constrain	A logical scalar. The argument constrain to confint.dgaps.
for_abline	Only relevant when y = "imts" and at one of u or D is scalar. A list of graphical parameters to be passed to abline to indicate the critical value of the information matrix test (IMT) implied by alpha.
digits	An integer. Used for formatting the value of the threshold with signif before adding its value to a plot.
uprob	A logical scalar. Should we plot x$u on the horizontal axis (uprob = FALSE) or the approximate sample quantile to which x$u corresponds (uprob = TRUE)?
leg_pos	A character scalar. The position of any legend added to a plot. Only relevant when both the arguments u and D in the call to choose_ud have length greater than one.
...	Additional arguments passed to matplot.

Details

The type of plot produced depends mainly on y.

If y = "imts" then the values of IMTS are plotted against the thresholds in x$u (or their corresponding approximate sample quantile levels if uprob = TRUE) for each value of D in x$D. Horizontal lines are added to indicate the critical values of the IMT for the significance levels in alpha. We would not reject at the 100alpha% level combinations of threshold and D corresponding to values of the IMTS that fall below the line.

If y = "theta" then estimates of θ are plotted on the vertical axis. If both x$u and x$D$ have length greater than one then only these estimates are plotted. If either x$u or x$D have length one then approximate 100level% confidence intervals are added to the plot and the variable, x$u or x$D that has length greater than one is plotted on the horizontal axis.

Value

Nothing is returned.

Examples

See the examples in choose_ud.

See Also

choose_ud.

---

plot.choose_uk   *Plot threshold u and runs parameter K diagnostic for the K-gaps estimator*

Description

plot method for objects inheriting from class "choose_uk", returned from choose_uk

Usage

```
## S3 method for class 'choose_uk'
plot(
  x,
  y = c("imts", "theta"),
  level = 0.95,
  interval_type = c("norm", "lik"),
  conf_scale = c("theta", "log"),
  alpha = 0.05,
  constrain = TRUE,
  for_abline = list(lty = 2, lwd = 1, col = 1),
  digits = 3,
  uprob = FALSE,
  leg_pos = if (y == "imts") "topright" else "topleft",
  ...
)
```

Arguments

`x`	an object of class `c("choose_uk", "exdex")`, a result of a call to `choose_uk`.
`y`	A character scalar indicating what should be plotted on the vertical axes of the plot: information matrix test statistics (IMTS) if `y = "imts"` and estimates of θ if `y = "theta"`. If `y = "theta"`, and either `x$u` or `x$k` have length one, then `100level%` confidence intervals are added to the plot.
`level`	A numeric scalar in (0, 1). The confidence level used in calculating confidence intervals for θ. Only relevant if `y = "theta"` and either `x$u` or `x$k` have length one.
`interval_type`	A character scalar. The type of confidence interval to be plotted, if `y = "theta"`. See `confint.kgaps`.
`conf_scale`	A character scalar. If `interval_type = "norm"` then `conf_scale` determines the scale on which we use approximate large-sample normality of the estimator to estimate confidence intervals. See `confint.kgaps`.
`alpha`	A numeric vector with entries in (0, 1). The size of the test to be performed.
`constrain`	A logical scalar. The argument `constrain` to `confint.kgaps`.
`for_abline`	Only relevant when `y = "imts"` and at one of `u` or `k` is scalar. A list of graphical parameters to be passed to `abline` to indicate the critical value of the information matrix test (IMT) implied by `alpha`.
`digits`	An integer. Used for formatting the value of the threshold with `signif` before adding its value to a plot.
`uprob`	A logical scalar. Should we plot `x$u` on the horizontal axis (`uprob = FALSE`) or the approximate sample quantile to which `x$u` corresponds (`uprob = TRUE`)?
`leg_pos`	A character scalar. The position of any legend added to a plot. Only relevant when both the arguments `u` and `k` in the call to `choose_uk` have length greater than one.
`...`	Additional arguments passed to `matplot`.

Details

The type of plot produced depends mainly on y.

If y = "imts" then the values of IMTS are plotted against the thresholds in x$u (or their corresponding approximate sample quantile levels if uprob = TRUE) for each value of K in x$k. Horizontal lines are added to indicate the critical values of the IMT for the significance levels in alpha. We would not reject at the 100alpha% level combinations of threshold and K corresponding to values of the IMTS that fall below the line.

If y = "theta" then estimates of θ are plotted on the vertical axis. If both x$u and x$k$ have length greater than one then only these estimates are plotted. If either x$u or x$k have length one then approximate 100level% confidence intervals are added to the plot and the variable, x$u or x$k that has length greater than one is plotted on the horizontal axis.

Value

Nothing is returned.

Examples

See the examples in choose_uk.

See Also

choose_uk.

---

sp500 *Daily log returns of the Standard and Poor (S&P) 500 index*

Description

Daily log returns of the S&P 500 index, that is, the log of the ratio of successive daily closing prices, from 3rd January 1990 to 9th October 2018.

Usage

```
sp500
```

Format

A vector of length 7250, created using zoo with an "index" attribute giving the date of the corresponding negated log return.

Source

Yahoo finance: https://finance.yahoo.com/quote/^SPX/history/

split_by_NAs — *Divides data into parts that contain no missing values*

Description

Splits the values in a numeric matrix column-wise into sequences of non-missing values.

Usage

```
split_by_NAs(x)
```

Arguments

x	A vector or matrix.

Details

For each column in x, split_by_NAs finds runs of values that contain no missing values and assigns them to a column in the matrix that is returned. Different columns are treated separately. If there are no missing values in a column then that column appears unmodified in the output matrix. Please see the **Examples** for illustrations.

Value

A matrix containing a column for each run of non-missing values in x. The number of rows is equal to the longest run of non-missing values in x and will therefore be at most nrow{x}. The matrix is padded with NA values at the end of each column, where necessary.

The returned object has an attribute called split_by_NAs_done whose value is TRUE, so that in programming one can avoid calling split_by_NAs more than once.

Examples

```
# Create a simple numeric matrix and insert some NAs
x <- matrix(1:50, 10, 5)
x[c(3, 8), 1] <- NA
x[c(1:2, 5, 10), 3] <- NA
x[1:3, 4] <- NA
x[7:10, 5] <- NA
x

res <- split_by_NAs(x)
res

# An example of a character matrix
x <- matrix(c(letters, letters[1:18]), 11, 4)
x[c(1:2, 5:11), 2] <- NA
x[c(2:4, 6:11), 3] <- NA
x[1:10, 4] <- NA
```

```
res <- split_by_NAs(x)
res
```

spm *Semiparametric maxima estimator of the extremal index*

Description

Estimates the extremal index θ using a semiparametric block maxima estimator of Northrop (2015) ("N2015") and a variant of this estimator studied by Berghaus and Bucher (2018) ("BB2018"), using both sliding (overlapping) block maxima and disjoint (non-overlapping) block maxima. A simple modification (subtraction of $1/b$, where b is the block size) of the Berghaus and Bucher (2018) estimator ("BB2018b") is also calculated. Estimates of uncertainty are calculated using the asymptotic theory developed by Berghaus and Bucher (2018).

Usage

```
spm(
  data,
  b,
  bias_adjust = c("BB3", "BB1", "N", "none"),
  constrain = TRUE,
  varN = TRUE,
  which_dj = c("last", "first")
)
```

Arguments

data	A numeric vector of raw data. No missing values are allowed.
b	A numeric scalar. The block size.
bias_adjust	A character scalar. Is bias-adjustment of the raw estimate of θ performed using the bias-reduced estimator (bias_adjust = "BB3"), derived in Section 5 of Berghaus and Bucher (2018); or a simpler version (bias_adjust = "BB1"), in which the raw estimate is multiplied by $(k-1)/k$, where k is the number of blocks; or the bias-adjustment of the empirical distribution function used to calculate the estimate, as detailed in Section 2 of Northrop (2015). When disjoint maxima are used bias_adjust = "BB1" and bias_adjust = "N" give identical estimates of the Berghaus and Bucher (2018) variant, as explained at the end of Section 5 of Berghaus and Bucher (2018). If bias_adjust = "none" then no bias-adjustment is performed.
constrain	A logical scalar. If constrain = TRUE then any estimates that are greater than 1 are set to 1, that is, they are constrained to lie in (0, 1]. This is carried out *after* any bias-adjustment. Otherwise, estimates that are greater than 1 may be obtained.

`varN` A logical scalar. If `varN = TRUE` then the estimation of the sampling variance of the Northrop (2015) estimator is tailored to that estimator. Otherwise, the sampling variance derived in Berghaus and Bucher (2018) is used. See **Details** for further information.

`which_dj` A character scalar. Determines which set of disjoint maxima are used to calculate an estimate of θ: `"first"`, only the set whose first block starts on the first observation in `x`; `"last"`, only the set whose last block ends on the last observation in `x`.

Details

The extremal index θ is estimated using the semiparametric maxima estimator of Northrop (2015) and variant of this studied by Berghaus and Bucher (2018). In each case a sample of 'data' is derived from the input data `data`, based on the empirical distribution function of these data evaluated at the maximum values of of blocks of `b` contiguous values in `data`.

The estimators are based on an assumption that these 'data' are sampled approximately from an exponential distribution with mean $1/\theta$. For details see page 2309 of Berghaus and Bucher (2018), where the 'data' for the Northrop (2015) estimator are denoted Y and those for the Berghaus and Bucher (2018) are denoted Z. For convenience, we will refer to these values as the Y-data and the Z-data.

The approximate nature of the model for the Y-data arises from the estimation of the distribution function F. A further approximation motivates the use of the Z-data. If F is known then the variable Z/b has a beta(1, $b\theta$) distribution, so that that is, Z has mean $1/(\theta + 1/b)$. Therefore, an exponential distribution with mean $1/(\theta + 1/b)$ may provide a better approximate model, which provides the motivation for subtracting $1/b$ from the Berghaus and Bucher (2018) estimator. Indeed, the resulting estimates are typically close to those of the Northrop (2015) estimator.

If `sliding = TRUE` then the function uses sliding block maxima, that is, the largest value observed in *all* (`length(data) - b + 1`) blocks of `b` observations. If `sliding = FALSE` then disjoint block maxima, that is, the largest values in (`floor(n / b)`) disjoint blocks of `b` observations, are used.

Estimation of the sampling variances of the estimators is based on Proposition 4.1 on page 2319 of Berghaus and Bucher (2018). For the Northrop (2015) variant the user has the choice either to use the sampling variance based on the Berghaus and Bucher (2018) estimator, i.e. the Z-data (`varN = FALSE`) or an analogous version tailored to the Northrop (2015) estimator that uses the Y-data (`varN = TRUE`).

The estimates of the sampling variances of the sliding blocks estimators are inferred from those of the disjoint blocks estimators (see page 2319 of Berhaus and Bucher (2018)). The calculation of the latter uses a set of disjoint block maxima. If `length(data)` is not an integer multiple of `b` then there will be more than one set of these, and all are equally valid. In this event we perform the calculation for all such sets and use the mean of the resulting estimates. This reduces the sampling variability of the estimates at the expense of slowing down the calculation somewhat, particularly if `b` is small. This may become apparent when calling `spm` repeatedly in `choose_b`.

This estimator of the sampling variance of the sliding blocks estimator is not constrained to be positive: a negative estimate may result if the block size is small. In this event **no warning will be given until the returned object is printed** and, for the affected estimator (`"N2015"` or `"BB2018/BB2018b"`),

- the corresponding estimated standard errors using sliding blocks will be missing in se_sl in the returned object,
- if bias_adjust == "BB3" then bias-adjustment based on bias_adjust == "BB1" will instead be performed when using sliding blocks, because the former relies on the estimated variances of the estimators.

Similarly, bias adjustment under adjust = "BB3" and/or subtraction of $1/b$ in the "BB2018b" case may, rare cases, produce a negative estimate of θ. In these instances an estimate of zero is returned, but the values returned in bias_dj and bias_sl are not changed.

Value

A list of class c("spm", "exdex") containing the components listed below. The components that are vectors are labelled to indicate the estimator to which the constituent values relate: "N2015" for Northrop (2015), "BB2018" for Berghaus and Bucher (2018) and "BB2018b" for the modified version.

theta_sl, theta_dj
: Vectors containing the estimates of θ resulting from sliding maxima and disjoint maxima respectively.

se_sl, se_dj
: The estimated standard errors associated with the estimates in theta_sl and theta_dj. The values for "BB2018" and "BB2018b" are identical.

bias_sl, bias_dj
: The respective values of the bias-adjustment applied to the raw estimates, that is, the values subtracted from the raw estimates. For estimator BB2018b this includes a contribution for the subtraction of 1 / b. If bias_adjust = "N" or "none" then bias_sl and bias_dj are c(0, 0, 1 / b).

raw_theta_sl, raw_theta_dj
: Vectors containing the raw estimates of θ, prior to any bias_adjustment.

uncon_theta_sl, uncon_theta_dj
: The (bias_adjusted) estimates of θ before the constraint that they lie in (0, 1] has been applied.

data_sl, data_dj
: Matrices containing the Y-data and Z-data for the sliding an disjoint maxima respectively. The first columns are the Y-data, the second columns the Z-data.

sigma2dj, sigma2dj_for_sl
: Estimates of the variance σ^2_{dj} defined on pages 2314-2315 of Berghaus and Bucher (2018). The form of the estimates is given on page 2319. sigma2dj is used in estimating the standard error se_dj, sigma2dj_for_sl in estimating the standard error se_sl.

sigma2sl
: Estimates of the variance σ^2_{sl}. defined on pages 2314-2315 of Berghaus and Bucher (2018). The form of the estimates is given on page 2319. sigma2dj_for_sl is used to estimate σ^2_{dj} for this purpose.

b
: The input value of b.

bias_adjust
: The input value of bias_adjust.

call
: The call to spm.

References

Northrop, P. J. (2015) An efficient semiparametric maxima estimator of the extremal index. *Extremes* **18**(4), 585-603. doi:10.1007/s1068701502215

Berghaus, B., Bucher, A. (2018) Weak convergence of a pseudo maximum likelihood estimator for the extremal index. *Ann. Statist.* **46**(5), 2307-2335. doi:10.1214/17AOS1621

See Also

`spm_confint` to estimate confidence intervals for θ.

`spm_methods` for S3 methods for `"spm"` objects.

Examples

```
### Newlyn sea surges
theta <- spm(newlyn, 20)
theta
summary(theta)
coef(theta)
nobs(theta)
vcov(theta)

### S&P 500 index
theta <- spm(sp500, 100)
theta
summary(theta)
```

---

spm_confint *Confidence intervals for the extremal index θ for "spm" objects*

Description

`confint` method for objects of class `c("spm", "exdex")`. Computes confidence intervals for θ based on an object returned from `spm`. Two types of interval may be returned: (a) intervals that are based on approximate large-sample normality of the estimators of θ (or of $\log\theta$ if `conf_scale = "log"`), and which are symmetric about the respective point estimates, and (b) likelihood-based intervals based on an adjustment of a naive (pseudo-) loglikelihood, using the `adjust_loglik` function in the `chandwich` package. The `plot` method plots the log-likelihood for θ, with the required confidence interval(s) indicated on the plot.

Usage

```
## S3 method for class 'spm'
confint(
  object,
  parm = "theta",
  level = 0.95,
  maxima = c("sliding", "disjoint"),
```

```
  interval_type = c("norm", "lik", "both"),
  conf_scale = c("theta", "log"),
  constrain = TRUE,
  bias_adjust = TRUE,
  type = c("vertical", "cholesky", "spectral", "none"),
  ...
)

## S3 method for class 'confint_spm'
plot(x, estimator = "all", ndec = 2, ...)

## S3 method for class 'confint_spm'
print(x, ...)
```

Arguments

`object`	An object of class `c("spm", "exdex")`, returned by `spm`.
`parm`	Specifies which parameter is to be given a confidence interval. Here there is only one option: the extremal index θ.
`level`	A numeric scalar in (0, 1). The confidence level required.
`maxima`	A character scalar specifying whether to estimate confidence intervals based on sliding maxima or disjoint maxima.
`interval_type`	A character scalar: `"norm"` for intervals of type (a), `"lik"` for intervals of type (b).
`conf_scale`	A character scalar. If `interval_type = "norm"` then `conf_scale` determines the scale on which we use approximate large-sample normality of the estimators to estimate confidence intervals of type (a). If `conf_scale = "theta"` then confidence intervals are estimated for θ directly. If `conf_scale = "log"` then confidence intervals are first estimated for $\log\theta$ and then transformed back to the θ-scale. Any bias-adjustment requested in the original call to `spm`, using it's `bias_adjust` argument, is automatically applied here.
`constrain`	A logical scalar. If `constrain = TRUE` then any confidence limits that are greater than 1 are set to 1, that is, they are constrained to lie in (0, 1]. Otherwise, limits that are greater than 1 may be obtained. If `constrain = TRUE` then any lower confidence limits that are less than 0 are set to 0.
`bias_adjust`	A logical scalar. If `bias_adjust = TRUE` then, if appropriate, bias-adjustment is also applied to the loglikelihood before it is adjusted using `adjust_loglik`. This is performed only if, in the call to `spm`, `bias_adjust = "BB3"` or `"BB1"` was specified, that is, we have `object$bias_adjust = "BB3"` or `"BB1"`. In these cases the relevant component of `object$bias_sl` or `object$bias_dj` is used to scale θ so that the location of the maximum of the loglikelihood lies at the bias-adjusted estimate of θ. If `bias_adjust = FALSE` or `object$bias_adjust = "none"` or `"N"` then no bias-adjustment of the intervals is performed. In the latter case this is because the bias-adjustment is applied in the creation of the data in `object$N2015_data` and `object$BB2018_data`, on which the naive likelihood is based.

type	A character scalar. The argument type to be passed to conf_intervals in the chandwich package in order to estimate the likelihood-based intervals. Using type = "none" is *not* advised because then the intervals are based on naive estimated standard errors. In particular, if (the default) sliding = TRUE was used in the call to spm then the unadjusted likelihood-based confidence intervals provide *vast* underestimates of uncertainty.
...	Additional optional arguments to be passed to print.default
x	an object of class c("confint_spm", "exdex"), a result of a call to confint.spm.
estimator	A character vector specifying which of the three variants of the semiparametric maxima estimator to include in the plot: "N2015", "BB2018" or "BB2018b". See spm for details. If estimator = "all" then all three are included.
ndec	An integer scalar. The legend (if included on the plot) contains the confidence limits rounded to ndec decimal places.

Details

The likelihood-based intervals are estimated using the adjust_loglik function in the chandwich package, followed by a call to conf_intervals. This adjusts the naive (pseudo-)loglikelihood so that the curvature of the adjust loglikelihood agrees with the estimated standard errors of the estimators. The option type = "none" should not be used in practice because the resulting confidence intervals will be wrong. In particular, in the intervals based on sliding maxima will provide *vast* underestimates of uncertainty.

If object$se contains NAs, because the block size b was too small or too large in the call to spm then confidence intervals cannot be estimated. A matrix of NAs will be returned. See the **Details** section of the spm documentation for more information.

print.confint_spm prints the matrix of confidence intervals for θ.

Value

A list of class c("confint_spm", "exdex") containing the following components.

cis	A matrix with columns giving the lower and upper confidence limits. These are labelled as (1 - level)/2 and 1 - (1 - level)/2 in % (by default 2.5% and 97.5%). The row names are a concatenation of the variant of the estimator (N2015 for Northrop (2015), BB2018 for Berghaus and Bucher (2018)), BB2018b for the modified (by subtracting 1 / b) Berghaus and Bucher (2018) and the type of interval (norm for symmetric and lik for likelihood-based).
ciN	The object returned from conf_intervals that contains information about the adjusted loglikelihood for the Northrop (2015) variant of the estimator.
ciBB	The object returned from conf_intervals that contains information about the adjusted loglikelihood for the Berghaus and Bucher (2018) variant of the estimator.
ciBBb	The object returned from conf_intervals that contains information about the adjusted loglikelihood for the modified Berghaus and Bucher (2018) variant of the estimator.
call	The call to spm.

object	The input object.
maxima	The input maxima.
theta	The relevant estimates of θ returned from adjust_loglik. These are equal to object$theta_sl if maxima = "sliding", object$theta_dj if maxima = "disjoint", which provides a check that the results are correct.
level	The input level.

plot.confint_spm: nothing is returned.

print.confint_spm: the argument x, invisibly.

References

Northrop, P. J. (2015) An efficient semiparametric maxima estimator of the extremal index. *Extremes* **18**(4), 585-603. doi:10.1007/s1068701502215

Berghaus, B., Bucher, A. (2018) Weak convergence of a pseudo maximum likelihood estimator for the extremal index. *Ann. Statist.* **46**(5), 2307-2335. doi:10.1214/17AOS1621

See Also

spm for estimation of the extremal index θ using a semiparametric maxima method.

Examples

```
# Newlyn sea surges
theta <- spm(newlyn, 20)
confint(theta)
cis <- confint(theta, interval_type = "lik")
cis
plot(cis)

# S&P 500 index
theta <- spm(sp500, 100)
confint(theta)
cis <- confint(theta, interval_type = "lik")
cis
plot(cis)
```

---

spm_methods *Methods for objects of class* "spm"

Description

Methods for objects of class c("spm", "exdex") returned from spm.

Usage

```
## S3 method for class 'spm'
coef(
  object,
  maxima = c("sliding", "disjoint"),
  estimator = "all",
  constrain = FALSE,
  ...
)

## S3 method for class 'spm'
vcov(object, maxima = c("sliding", "disjoint"), estimator = "all", ...)

## S3 method for class 'spm'
nobs(object, maxima = c("sliding", "disjoint"), ...)

## S3 method for class 'spm'
print(x, digits = max(3L, getOption("digits") - 3L), ...)

## S3 method for class 'spm'
summary(object, digits = max(3, getOption("digits") - 3L), ...)

## S3 method for class 'summary.spm'
print(x, ...)
```

Arguments

`object`	an object of class "spm", a result of a call to `spm`.
`maxima`	A character scalar specifying whether to return the number of observed sliding maxima or disjoint maxima.
`estimator`	A character vector specifying which of the three variants of the semiparametric maxima estimator to use: "N2015", "BB2018" or "BB2018b". See `spm` for details. If `estimator = "all"` then the estimated variances of all variants are returned.
`constrain`	A logical scalar. If `constrain = TRUE` then any estimates that are greater than 1 are set to 1, that is, they are constrained to lie in (0, 1]. Otherwise, estimates that are greater than 1 may be obtained.
`...`	For `print.summary.spm`, additional arguments passed to `print.default`.
`x`	`print.spm`. An object of class `c("spm", "exdex")`, a result of a call to `spm`. `print.summary.spm`. An object of class "summary.spm", a result of a call to `summary.spm`.
`digits`	An integer. Used for number formatting with `signif`.

Details

`print.spm` prints the original call to `spm` and the estimates of the extremal index θ, based on all three variants of the semiparametric maxima estimator and both sliding and disjoint blocks.

Value

coef.spm. A numeric scalar (or a vector of length 3 if estimator = "all"): the required estimate(s) of the extremal index θ.

vcov.spm. A 1×1 numeric matrix if estimator = "N2015" or "BB2018" and a vector of length 3 if estimator = "all", containing the estimated variance(s) of the estimator(s).

nobs.spm. A numeric scalar: the number of observations used in the fit.

print.spm. The argument x, invisibly.

summary.spm. Returns an object (a list) of class "summary.spm" containing the list element object$call and a numeric matrix matrix giving, for all three variants of the semiparametric estimator and both sliding and disjoint blocks, the (bias-adjusted) Estimate of the extremal index θ, the estimated standard error (Std. Error), and the bias adjustment (Bias adj.) applied to obtain the estimate, i.e. the value subtracted from the raw estimate. If any of the (bias-adjusted) estimates are greater than 1 then a column containing the unconstrained estimates (Uncon. estimate) is added.

print.summary.spm. The argument x, invisibly.

Examples

See the examples in spm.

See Also

spm for semiparametric estimation of the extremal index based on block maxima.

---

uccle *Uccle maximum daily temperatures*

Description

The dataframe uccle contains daily maximum temperatures in degrees C recorded at the Uccle, Belgium from 1/1/1833 to 23/1/2011. The Station identifier in the source file is 17 and the Source identifier is 117882.

Usage

```
uccle
```

Format

A data frame with 65036 observations on the following and 5 variables.

- temp: daily maximum temperature in degrees C.
- year: the year.
- month: the month of the year.
- day: day of the month.
- date: date with the Date class, in the format YYYY-MM-DD.

Note

There are 5336 missing values.

Source

Klein Tank, A.M.G. and Coauthors, 2002. Daily dataset of 20th-century surface air temperature and precipitation series for the European Climate Assessment. *Int. J. of Climatol.*, **22**, 1441-1453 doi:10.1002/joc.773. Data and metadata available at https://www.ecad.eu. The data were downloaded on 27/3/2022 using a Custom query (ASCII), selecting "non-blend" for type of series.

`uccle720` *20th century Uccle maximum daily temperatures in July - data frame*

Description

The dataframe `uccle720` contains daily maximum temperatures in degrees C recorded at the Uccle, Belgium during July for the years 1901 to 1999. The Station identifier in the source file is 17 and the Source identifier is 117882. These data are analysed in Holesovsky and Fusek (2020).

Usage

```
uccle720
```

Format

A data frame with 3100 observations on the following and 5 variables.

- `temp`: daily maximum temperature in degrees C.
- `year`: the year.
- `month`: the month of the year.
- `day`: day of the month.
- `date`: date with the `Date` class, in the format YYYY-MM-DD.

Note

There are 6 missing values, one located in each of the years 1925, 1926, 1956, 1963, 1969 and 1976.

Source

Klein Tank, A.M.G. and Coauthors, 2002. Daily dataset of 20th-century surface air temperature and precipitation series for the European Climate Assessment. *Int. J. of Climatol.*, **22**, 1441-1453 doi:10.1002/joc.773. Data and metadata available at https://www.ecad.eu. The data were downloaded on 27/3/2022 using a Custom query (ASCII), selecting "non-blend" for type of series.

References

Holesovsky, J. and Fusek, M. Estimation of the extremal index using censored distributions. *Extremes*, **23**, 197-213 (2020). doi:10.1007/s10687020003743

Examples

```
uccle720_ts <- ts(uccle720$temp, start = c(1901, 1), frequency = 31)
plot(uccle720_ts, ylab = "daily maximum temperature in July / degrees C",
     xlab = "year")
```

`uccle720m` *20th century Uccle maximum daily temperatures in July - matrix*

Description

The matrix `uccle720m` contains daily maximum temperatures in degrees C recorded at the Uccle, Belgium during July for the years 1901 to 1999. The Station identifier in the source file is 17 and the Source identifier is 117882. These data are analysed in Holesovsky and Fusek (2020).

Usage

```
uccle720m
```

Format

A 31 by 100 numeric matrix. Column `i` contains the maximum daily temperature in degrees C at Uccle in the year 1900 + `i` - 1. The columns are named 1900, 1901, ..., 1999 and the rows are named after the day of the month: 1, 2, .., 31.

Note

There are 6 missing values, one located in each of the years 1925, 1926, 1956, 1963, 1969 and 1976.

Source

Klein Tank, A.M.G. and Coauthors, 2002. Daily dataset of 20th-century surface air temperature and precipitation series for the European Climate Assessment. *Int. J. of Climatol.*, **22**, 1441-1453 doi:10.1002/joc.773. Data and metadata available at https://www.ecad.eu. The data were downloaded on 27/3/2022 using a Custom query (ASCII), selecting "non-blend" for type of series.

References

Holesovsky, J. and Fusek, M. Estimation of the extremal index using censored distributions. *Extremes*, **23**, 197-213 (2020). doi:10.1007/s10687020003743

Examples

```
uccle720_ts <- ts(uccle720$temp, start = c(1901, 1), frequency = 31)
plot(uccle720_ts, ylab = "daily maximum temperature in July / degrees C",
     xlab = "year")
```

Index