Package 'tea'

May 8, 2026

Type Package

Title Threshold Estimation Approaches

Version 1.1

Date 2020-04-17

Author Johannes Ossberger

Maintainer Johannes Ossberger <johannes.ossberger@gmail.com>

Description Different approaches for selecting the threshold in generalized Pareto distributions. Most of them are based on minimizing the AMSE-criterion or at least by reducing the bias of the assumed GPD-model. Others are heuristically motivated by searching for stable sample paths, i.e. a nearly constant region of the tail index estimator with respect to k, which is the number of data in the tail. The third class is motivated by graphical inspection. In addition, a sequential testing procedure for GPD-GoF-tests is also implemented here.

License GPL-3

Imports Matrix, stats, graphics

RoxygenNote 6.1.1

NeedsCompilation no

Repository CRAN

Date/Publication 2020-04-19 15:00:06 UTC

Contents

`tea-package` *Threshold Estimation Approaches*

Description

This package contains implementations of many of the threshold estimation approaches proposed in the literature. The estimation of the threshold is of great interest in statistics of extremes. Estimating the threshold is equivalent to choose the optimal sample fraction in tail index estimation. The sample fraction is given by `k/n` with `n` the sample size and `k` the number of extremes in the data or, if you wish, the exceedances over a high unknown threshold `u`.

Details

Package:	tea
Type:	Package
Version:	1.1
Date:	2020-04-17
License:	GPL-3

Author(s)

Johannes Ossberger

Maintainer: Johannes Ossberger <johannes.ossberger@gmail.com>

References

Caeiro and Gomes (2016) <doi:10.1201/b19721-5>

Cebrian et al. (2003) <doi:10.1080/10920277.2003.10596098>

Danielsson et al. (2001) <doi:10.1006/jmva.2000.1903>

Danielsson et al. (2016) <doi:10.2139/ssrn.2717478>

De Sousa and Michailidis (2004) <doi:10.1198/106186004X12335>

Drees and Kaufmann (1998) <doi:10.1016/S0304-4149(98)00017-9>

Hall (1990) <doi:10.1016/0047-259X(90)90080-2>

Hall and Welsh (1985) <doi:10.1214/aos/1176346596>

Kratz and Resnick (1996) <doi:10.1080/15326349608807407>

Gomes et al. (2011) <doi:10.1080/03610918.2010.543297>

Gomes et al. (2012) <doi:10.1007/s10687-011-0146-6>

Gomes et al. (2013) <doi:10.1080/00949655.2011.652113>

G'Sell et al. (2016) <doi:10.1111/rssb.12122>

Guillou and Hall <doi:10.1111/1467-9868.00286>

Reiss and Thomas (2007) <doi:10.1007/978-3-0348-6336-0>

Resnick and Starica (1997) <doi:10.1017/S0001867800027889>

Thompson et al. (2009) <doi:10.1016/j.coastaleng.2009.06.003>

`althill` *Alternative Hill Plot*

Description

Plots the Alternative Hill Plot and an averaged version of it against the upper order statistics.

Usage

```
althill(data, u = 2, kmin = 5, conf.int = FALSE)
```

Arguments

`data`	vector of sample data
`u`	gives the amount of which the Hill estimator is averaged. Default ist set to `u=2`.
`kmin`	gives the minimal `k` for which the graph is plotted. Default ist set to `kmin=5`.
`conf.int`	`logical`. If FALSE (default) no confidence intervals are plotted

Details

The Alternative Hill Plot is just a normal Hill Plot scaled to the `[0,1]` interval which can make interpretation much easier. See references for more information.

Value

The normal black line gives a simple Hill Plot scaled to `[0,1]`. The red dotted line is an averaged version that smoothes the Hill Plot by taking the mean of `k(u-1)` subsequent Hill estimations with respect to `k`. See references for more information.

References

Resnick, S. and Starica, C. (1997). Smoothing the Hill estimator. *Advances in Applied Probability*, 271–293.

Examples

```
data=rexp(500)
althill(data)
```

`avhill` *Averaged Hill Plot*

Description

Plots an averaged version of the classical Hill Plot

Usage

```
avhill(data, u = 2, kmin = 5, conf.int = FALSE)
```

Arguments

`data`	vector of sample data
`u`	gives the amount of which the Hill estimator is averaged. Default ist set to `u=2`.
`kmin`	gives the minimal `k` for which the graph is plotted. Default ist set to `kmin=5`.
`conf.int`	`logical`. If FALSE (default) no confidence intervals are plotted

Details

The Averaged Hill Plot is a smoothed version of the classical Hill Plot by taking the mean of values of the Hill estimator for subsequent `k`, i.e. upper order statistics. For more information see references.

Value

The normal black line gives the classical Hill Plot. The red dotted line is an averaged version that smoothes the Hill Plot by taking the mean of `k(u-1)` subsequent Hill estimations with respect to `k`. See references for more information.

References

Resnick, S. and Starica, C. (1997). Smoothing the Hill estimator. *Advances in Applied Probability*, 271–293.

Examples

```
data(danish)
avhill(danish)
```

---

dAMSE *Minimizing the AMSE of the Hill estimator with respect to k*

Description

Gives the optimal number of upper order statistics k for the Hill estimator by minimizing the AMSE-criterion.

Usage

```
dAMSE(data)
```

Arguments

data	vector of sample data

Details

The optimal number of upper order statistics is equivalent to the number of extreme values or, if you wish, the number of exceedances in the context of a POT-model like the generalized Pareto distribution. This number is identified by minimizing the AMSE criterion with respect to k. The optimal number, denoted k0 here, can then be associated with the unknown threshold u of the GPD by choosing u as the n-k0th upper order statistic. For more information see references.

Value

second.order.par	gives an estimation of the second order parameter beta and rho.
k0	optimal number of upper order statistics, i.e. number of exceedances or data in the tail
threshold	the corresponding threshold
tail.index	the corresponding tail index

References

Caeiro, J. and Gomes, M.I. (2016). Threshold selection in extreme value analysis. *Extreme Value Modeling and Risk Analysis:Methids and Applications*, 69–86.

Examples

```
data(danish)
dAMSE(danish)
```

`danielsson` *A Double Bootstrap Procedure for Choosing the Optimal Sample Fraction*

Description

An Implementation of the procedure proposed in Danielsson et al. (2001) for selecting the optimal sample fraction in tail index estimation.

Usage

```
danielsson(data, B = 500, epsilon = 0.9)
```

Arguments

`data`	vector of sample data
`B`	number of Bootstrap replications
`epsilon`	gives the amount of the first resampling size `n1` by choosing `n1 = n^epsilon`. Default is set to `epsilon=0.9`

Details

The Double Bootstrap procedure simulates the AMSE criterion of the Hill estimator using an auxiliary statistic. Minimizing this statistic gives a consistent estimator of the sample fraction `k/n` with `k` the optimal number of upper order statistics. This number, denoted `k0` here, is equivalent to the number of extreme values or, if you wish, the number of exceedances in the context of a POT-model like the generalized Pareto distribution. `k0` can then be associated with the unknown threshold `u` of the GPD by choosing `u` as the `n-k0`th upper order statistic. For more information see references.

Value

`second.order.par`	gives an estimation of the second order parameter `rho`.
`k0`	optimal number of upper order statistics, i.e. number of exceedances or data in the tail
`threshold`	the corresponding threshold
`tail.index`	the corresponding tail index

References

Danielsson, J. and Haan, L. and Peng, L. and Vries, C.G. (2001). Using a bootstrap method to choose the sample fraction in tail index estimation. *Journal of Multivariate analysis*, **2**, 226-248.

Examples

```
data=rexp(100)
danielsson(data, B=200)
```

---

danish *Danish Fire Insurance Claims*

Description

These data describe large fire insurance claims in Denmark from Thursday 3rd January 1980 until Monday 31st December 1990. The data are contained in a numeric vector. They were supplied by Mette Rytgaard of Copenhagen Re

Usage

```
data("danish")
```

Format

The format is: atomic [1:2167] 1.68 2.09 1.73 1.78 4.61 ... - attr(*, "times")= POSIXt[1:2167], format: "1980-01-03 01:00:00" "1980-01-04 01:00:00" ...

Source

The data is taken from package `evir`.

Examples

```
data(danish)
```

DK *A Bias-based procedure for Choosing the Optimal Sample Fraction*

Description

An Implementation of the procedure proposed in Drees & Kaufmann (1998) for selecting the optimal sample fraction in tail index estimation.

Usage

```
DK(data, r = 1)
```

Arguments

`data`	vector of sample data
`r`	tuning parameter for the stopping criterion. `default` is set to `1`. Change only if recommended by the output.

Details

The procedure proposed in Drees & Kaufmann (1998) is based on bias reduction. A stopping criterion with respect to k is implemented to find the optimal tail fraction, i.e. k/n with k the optimal number of upper order statistics. This number, denoted k0 here, is equivalent to the number of extreme values or, if you wish, the number of exceedances in the context of a POT-model like the generalized Pareto distribution. k0 can then be associated with the unknown threshold u of the GPD by choosing u as the n-k0th upper order statistic. If the above mentioned stopping criterion exceedes a certain value r, the bias of the assumed extreme model has become prominent and therefore k should not be chosen higher. For more information see references.

Value

second.order.par	gives an estimation of the second order parameter rho.
k0	optimal number of upper order statistics, i.e. number of exceedances or data in the tail
threshold	the corresponding threshold
tail.index	the corresponding tail

References

Drees, H. and Kaufmann, E. (1998). Selecting the optimal sample fraction in univariate extreme value estimation. *Stochastic Processes and their Applications*, **75(2)**, 149–172.

Examples

```
data(danish)
DK(danish)
```

---

eye *Automated Approach for Interpreting the Hill-Plot*

Description

An Implementation of the so called Eye-balling Technique proposed in Danielsson et al. (2016)

Usage

```
eye(data, ws = 0.01, epsilon = 0.3, h = 0.9)
```

Arguments

data	vector of sample data
ws	size of the moving window. Default is one percent of the data
epsilon	size of the range in which the estimates can vary
h	percentage of data inside the moving window that should lie in the tolerable range

Details

The procedure searches for a stable region in the Hill-Plot by defining a moving window. Inside this window the estimates of the Hill estimator with respect to k have to be in a pre-defined range around the first estimate within this window. It is sufficient to claim that only h percent of the estimates within this window lie in this range. The smallest k that accomplishes this is then the optimal number of upper order statistics, i.e. data in the tail.

Value

k0	optimal number of upper order statistics, i.e. number of exceedances or data in the tail
threshold	the corresponding threshold
tail.index	the corresponding tail index by plugging in k0 into the hill estimator

References

Danielsson, J. and Ergun, L.M. and de Haan, L. and de Vries, C.G. (2016). Tail Index Estimation: Quantile Driven Threshold Selection.

Examples

```
data(danish)
eye(danish)
```

---

ggplot *Gerstengarbe Plot*

Description

Performs a sequential Mann-Kendall Plot also known as Gerstengarbe Plot.

Usage

```
ggplot(data, nexceed = min(data) - 1)
```

Arguments

data	vector of sample data
nexceed	number of exceedances. Default is the minimum of the data to make sure the whole dataset is considered.

Details

The Gerstengarbe Plot, referring to Gerstengarbe and Werner (1989), is a sequential version of the Mann-Kendall-Test. This test searches for change points within a time series. This method is adopted for finding a threshold in a POT-model. The basic idea is that the differences of order statistics of a given dataset behave different between the body and the tail of a heavy-tailed distribution. So there should be a change point if the POT-model holds. To identify this change point the sequential test is done twice, for the differences from start to the end of the dataset and vice versa. The intersection point of these two series can then be associated with the change point of the sample data. For more informations see references.

Value

`k0`	optimal number of upper order statistics, i.e. the change point of the dataset
`threshold`	the corresponding threshold
`tail.index`	the corresponding tail index

Authors

Ana Cebrian Johannes Ossberger

Acknowledgements

Great thanks to A. Cebrian for providing a basic version of this code.

References

Gerstengarbe, F.W. and Werner, P.C. (1989). A method for statistical definition of extreme-value regions and their application to meteorological time series. *Zeitschrift fuer Meteorologie*, **39(4)**, 224–226.

Cebrian, A., and Denuit, M. and Lambert, P. (2003). Generalized pareto fit to the society of actuaries large claims database. *North American Actuarial Journal*, **7(3)**, 18–36.

Examples

```
data(danish)
ggplot(danish)
```

`GH` *A Bias-based procedure for Choosing the Optimal Threshold*

Description

An Implementation of the procedure proposed in Guillou & Hall(2001) for selecting the optimal threshold in extreme value analysis.

Usage

```
GH(data)
```

Arguments

data	vector of sample data

Details

The procedure proposed in Guillou & Hall (2001) is based on bias reduction. Due to the fact that the log-spacings of the order statistics are approximately exponentially distributed if the tail of the underlying distribution follows a Pareto distribution, an auxilliary statistic with respect to k is implemented with the same properties. The method then behaves like an asymptotic test for mean 0. If some critical value crit is exceeded the hypothesis of zero mean is rejected. Thus the bias has become too large and the assumed exponentiality and therefore the assumed Pareto tail can not be hold. From this an optimal number of k can be found such that the critical value is not exceeded. This optimal number, denoted k0 here, is equivalent to the number of extreme values or, if you wish, the number of exceedances in the context of a POT-model like the generalized Pareto distribution. k0 can then be associated with the unknown threshold u of the GPD by coosing u as the n-k0th upper order statistic. For more information see references.

Value

k0	optimal number of upper order statistics, i.e. number of exceedances or data in the tail
threshold	the corresponding threshold
tail.index	the corresponding tail index

References

Guillou, A. and Hall, P. (2001). A Diagnostic for Selecting the Threshold in Extreme Value Analysis. *Journal of the Royal Statistical Society*, **63(2)**, 293–305.

Examples

```
data(danish)
GH(danish)
```

---

gomes *A Double Bootstrap Procedure for Choosing the Optimal Sample Fraction*

Description

An Implementation of the procedure proposed in Gomes et al. (2012) and Caeiro et al. (2016) for selecting the optimal sample fraction in tail index estimation.

Usage

```
gomes(data, B = 1000, epsilon = 0.995)
```

Arguments

data	vector of sample data
B	number of Bootstrap replications
epsilon	gives the amount of the first resampling size n1 by choosing n1 = n^epsilon. Default is set to epsilon=0.995

Details

The Double Bootstrap procedure simulates the AMSE criterion of the Hill estimator using an auxiliary statistic. Minimizing this statistic gives a consistent estimator of the sample fraction k/n with k the optimal number of upper order statistics. This number, denoted k0 here, is equivalent to the number of extreme values or, if you wish, the number of exceedances in the context of a POT-model like the generalized Pareto distribution. k0 can then be associated with the unknown threshold u of the GPD by choosing u as the n-k0th upper order statistic. For more information see references.

Value

second.order.par	gives an estimation of the second order parameter rho.
k0	optimal number of upper order statistics, i.e. number of exceedances or data in the tail
threshold	the corresponding threshold
tail.index	the corresponding tail

References

Gomes, M.I. and Figueiredo, F. and Neves, M.M. (2012). Adaptive estimation of heavy right tails: resampling-based methods in action. *Extremes*, **15**, 463–489.

Caeiro, F. and Gomes, I. (2016). Threshold selection in extreme value analysis. *Extreme Value Modeling and Risk Analysis: Methods and Applications*, 69–86.

Examples

```
data(danish)
gomes(danish)
```

gpd *The Generalized Pareto Distribution (GPD)*

Description

Density, distribution function, quantile function and random number generation for the Generalized Pareto distribution with location, scale, and shape parameters.

Usage

```
dgpd(x, loc = 0, scale = 1, shape = 0, log.d = FALSE)

rgpd(n, loc = 0, scale = 1, shape = 0)

qgpd(p, loc = 0, scale = 1, shape = 0, lower.tail = TRUE,
  log.p = FALSE)

pgpd(q, loc = 0, scale = 1, shape = 0, lower.tail = TRUE,
  log.p = FALSE)
```

Arguments

x	Vector of observations.
loc, scale, shape	Location, scale, and shape parameters. Can be vectors, but the lengths must be appropriate.
log.d	Logical; if TRUE, the log density is returned.
n	Number of observations.
p	Vector of probabilities.
lower.tail	Logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
log.p	Logical; if TRUE, probabilities p are given as log(p).
q	Vector of quantiles.

Details

The Generalized Pareto distribution function is given (Pickands, 1975) by

$$H(y) = 1 - \left[1 + \frac{\xi(y-\mu)}{\sigma}\right]^{-1/\xi}$$

defined on $\{y : y > 0, (1 + \xi(y - \mu)/\sigma) > 0\}$, with location μ, scale $\sigma > 0$, and shape parameter ξ.

References

Brian Bader, Jun Yan. "eva: Extreme Value Analysis with Goodness-of-Fit Testing." R package version (2016)

Pickands III, J. (1975). Statistical inference using extreme order statistics. Annals of Statistics, 119-131.

Examples

```
dgpd(2:4, 1, 0.5, 0.01)
dgpd(2, -2:1, 0.5, 0.01)
pgpd(2:4, 1, 0.5, 0.01)
qgpd(seq(0.9, 0.6, -0.1), 2, 0.5, 0.01)
rgpd(6, 1, 0.5, 0.01)

## Generate sample with linear trend in location parameter
rgpd(6, 1:6, 0.5, 0.01)

## Generate sample with linear trend in location and scale parameter
rgpd(6, 1:6, seq(0.5, 3, 0.5), 0.01)

p <- (1:9)/10
pgpd(qgpd(p, 1, 2, 0.8), 1, 2, 0.8)
## [1] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9

## Incorrect syntax (parameter vectors are of different lengths other than 1)
# rgpd(1, 1:8, 1:5, 0)

## Also incorrect syntax
# rgpd(10, 1:8, 1, 0.01)
```

---

gpdFit *Parameter estimation for the Generalized Pareto Distribution (GPD)*

Description

Fits exceedances above a chosen threshold to the Generalized Pareto model. Various estimation procedures can be used, including maximum likelihood, probability weighted moments, and maximum product spacing. It also allows generalized linear modeling of the parameters.

Usage

```
gpdFit(data, threshold = NA, nextremes = NA, npp = 365,
  method = c("mle", "mps", "pwm"), information = c("expected",
  "observed"), scalevars = NULL, shapevars = NULL, scaleform = ~1,
  shapeform = ~1, scalelink = identity, shapelink = identity,
  start = NULL, opt = "Nelder-Mead", maxit = 10000, ...)
```

Arguments

`data`	Data should be a numeric vector from the GPD.
`threshold`	A threshold value or vector of the same length as the data.
`nextremes`	Number of upper extremes to be used (either this or the threshold must be given, but not both).
`npp`	Length of each period (typically year). Is used in return level estimation. Defaults to 365.
`method`	Method of estimation - maximum likelihood (mle), maximum product spacing (mps), and probability weighted moments (pwm). Uses mle by default. For pwm, only the stationary model can be fit.
`information`	Whether standard errors should be calculated via observed or expected (default) information. For probability weighted moments, only expected information will be used if possible. For non-stationary models, only observed information is used.
`scalevars, shapevars`	A dataframe of covariates to use for modeling of the each parameter. Parameter intercepts are automatically handled by the function. Defaults to NULL for the stationary model.
`scaleform, shapeform`	An object of class 'formula' (or one that can be coerced into that class), specifying the model of each parameter. By default, assumes stationary (intercept only) model. See details.
`scalelink, shapelink`	A link function specifying the relationship between the covariates and each parameter. Defaults to the identity function. For the stationary model, only the identity link should be used.
`start`	Option to provide a set of starting parameters to optim; a vector of scale and shape, in that order. Otherwise, the routine attempts to find good starting parameters. See details.
`opt`	Optimization method to use with optim.
`maxit`	Number of iterations to use in optimization, passed to optim. Defaults to 10,000.
`...`	Additional arguments to pass to optim.

Details

The base code for finding probability weighted moments is taken from the R package evir. See citation. In the stationary case (no covariates), starting parameters for mle and mps estimation are the probability weighted moment estimates. In the case where covariates are used, the starting intercept parameters are the probability weighted moment estimates from the stationary case and the parameters based on covariates are initially set to zero. For non-stationary parameters, the first reported estimate refers to the intercept term. Covariates are centered and scaled automatically to speed up optimization, and then transformed back to original scale.

Formulas for generalized linear modeling of the parameters should be given in the form '~ var1 + var2 + $\cdots$'. Essentially, specification here is the same as would be if using function 'lm' for only the right hand side of the equation. Interactions, polynomials, etc. can be handled as in the

'formula' class.
Intercept terms are automatically handled by the function. By default, the link functions are the identity function and the covariate dependent scale parameter estimates are forced to be positive. For some link function $f(\cdot)$ and for example, scale parameter σ, the link is written as $\sigma = f(\sigma_1 x_1 + \sigma_2 x_2 + \cdots + \sigma_k x_k)$.
Maximum likelihood estimation and maximum product spacing estimation can be used in all cases. Probability weighted moments can only be used for stationary models.

Value

A class object 'gpdFit' describing the fit, including parameter estimates and standard errors.

References

Brian Bader, Jun Yan. "eva: Extreme Value Analysis with Goodness-of-Fit Testing." R package version (2016)

Examples

```
## Fit data using the three different estimation procedures
set.seed(7)
x <- rgpd(2000, loc = 0, scale = 2, shape = 0.2)
## Set threshold at 4
mle_fit <- gpdFit(x, threshold = 4, method = "mle")
pwm_fit <- gpdFit(x, threshold = 4, method = "pwm")
mps_fit <- gpdFit(x, threshold = 4, method = "mps")
## Look at the difference in parameter estimates and errors
mle_fit$par.ests
pwm_fit$par.ests
mps_fit$par.ests

mle_fit$par.ses
pwm_fit$par.ses
mps_fit$par.ses

## A linear trend in the scale parameter
set.seed(7)
n <- 300
x2 <- rgpd(n, loc = 0, scale = 1 + 1:n / 200, shape = 0)

covs <- as.data.frame(seq(1, n, 1))
names(covs) <- c("Trend1")

result1 <- gpdFit(x2, threshold = 0, scalevars = covs, scaleform = ~ Trend1)

## Show summary of estimates
result1
```

`hall` *A Single Bootstrap Procedure for Choosing the Optimal Sample Fraction*

Description

An Implementation of the procedure proposed in Hall (1990) for selecting the optimal sample fraction in tail index estimation

Usage

```
hall(data, B = 1000, epsilon = 0.955, kaux = 2 * sqrt(length(data)))
```

Arguments

`data`	vector of sample data
`B`	number of Bootstrap replications
`epsilon`	gives the amount of the first resampling size `n1` by choosing `n1 = n^epsilon`. Default is set to `epsilon=0.955`
`kaux`	tuning parameter for the hill estimator

Details

The Bootstrap procedure simulates the AMSE criterion of the Hill estimator. The unknown theoretical parameter of the inverse tail index `gamma` is replaced by a consistent estimation using a tuning parameter `kaux` for the Hill estimator. Minimizing this statistic gives a consistent estimator of the sample fraction `k/n` with `k` the optimal number of upper order statistics. This number, denoted `k0` here, is equivalent to the number of extreme values or, if you wish, the number of exceedances in the context of a POT-model like the generalized Pareto distribution. `k0` can then be associated with the unknown threshold `u` of the GPD by choosing `u` as the `n-k0`th upper order statistic. For more information see references.

Value

`k0`	optimal number of upper order statistics, i.e. number of exceedances or data in the tail
`threshold`	the corresponding threshold
`tail.index`	the corresponding tail index

References

Hall, P. (1990). Using the Bootstrap to Estimate Mean Squared Error and Select Smoothing Parameter in Nonparametric Problems. *Journal of Multivariate Analysis*, **32**, 177–203.

Examples

```
data(danish)
hall(danish)
```

Himp *A Single Bootstrap Procedure for Choosing the Optimal Sample Fraction*

Description

An Implementation of the procedure proposed in Caeiro & Gomes (2012) for selecting the optimal sample fraction in tail index estimation

Usage

```
Himp(data, B = 1000, epsilon = 0.955)
```

Arguments

data	vector of sample data
B	number of Bootstrap replications
epsilon	gives the amount of the first resampling size n1 by choosing n1 = n^epsilon. Default is set to epsilon=0.955

Details

This procedure is an improvement of the one introduced in Hall (1990) by overcoming the restrictive assumptions through estimation of the necessary parameters. The Bootstrap procedure simulates the AMSE criterion of the Hill estimator using an auxiliary statistic. Minimizing this statistic gives a consistent estimator of the sample fraction k/n with k the optimal number of upper order statistics. This number, denoted k0 here, is equivalent to the number of extreme values or, if you wish, the number of exceedances in the context of a POT-model like the generalized Pareto distribution. k0 can then be associated with the unknown threshold u of the GPD by choosing u as the n-k0th upper order statistic. For more information see references.

Value

second.order.par	gives an estimation of the second order parameter rho.
k0	optimal number of upper order statistics, i.e. number of exceedances or data in the tail
threshold	the corresponding threshold
tail.index	the corresponding tail index

References

Hall, P. (1990). Using the Bootstrap to Estimate Mean Squared Error and Select Smoothing Parameter in Nonparametric Problems. *Journal of Multivariate Analysis*, **32**, 177–203.

Caeiro, F. and Gomes, M.I. (2014). On the bootstrap methodology for the estimation of the tail sample fraction. *Proceedings of COMPSTAT*, 545–552.

Examples

```
data(danish)
Himp(danish)
```

HW *Minimizing the AMSE of the Hill estimator with respect to k*

Description

An Implementation of the procedure proposed in Hall & Welsh (1985) for obtaining the optimal number of upper order statistics k for the Hill estimator by minimizing the AMSE-criterion.

Usage

```
HW(data)
```

Arguments

data	vector of sample data

Details

The optimal number of upper order statistics is equivalent to the number of extreme values or, if you wish, the number of exceedances in the context of a POT-model like the generalized Pareto distribution. This number is identified by minimizing the AMSE criterion with respect to k. The optimal number, denoted k0 here, can then be associated with the unknown threshold u of the GPD by choosing u as the n-k0th upper order statistic. For more information see references.

Value

second.order.par	gives an estimation of the second order parameter rho.
k0	optimal number of upper order statistics, i.e. number of exceedances or data in the tail
threshold	the corresponding threshold
tail.index	the corresponding tail index

References

Hall, P. and Welsh, A.H. (1985). Adaptive estimates of parameters of regular variation. *The Annals of Statistics*, **13(1)**, 331–341.

Examples

```
data(danish)
HW(danish)
```

`mindist` *Minimizing the distance between the empirical tail and a theoretical Pareto tail with respect to k.*

Description

An Implementation of the procedure proposed in Danielsson et al. (2016) for selecting the optimal threshold in extreme value analysis.

Usage

```
mindist(data, ts = 0.15, method = "mad")
```

Arguments

`data`	vector of sample data
`ts`	size of the upper tail the procedure is applied to. Default is 15 percent of the data
`method`	should be one of `ks` for the "Kolmogorov-Smirnov" distance metric or `mad` for the mean absolute deviation (default)

Details

The procedure proposed in Danielsson et al. (2016) minimizes the distance between the largest upper order statistics of the dataset, i.e. the empirical tail, and the theoretical tail of a Pareto distribution. The parameter of this distribution are estimated using Hill's estimator. Therefor one needs the optimal number of upper order statistics `k`. The distance is then minimized with respect to this `k`. The optimal number, denoted `k0` here, is equivalent to the number of extreme values or, if you wish, the number of exceedances in the context of a POT-model like the generalized Pareto distribution. `k0` can then be associated with the unknown threshold `u` of the GPD by saying `u` is the `n-k0`th upper order statistic. For the distance metric in use one could choose the mean absolute deviation called `mad` here, or the maximum absolute deviation, also known as the "Kolmogorov-Smirnov" distance metric (`ks`). For more information see references.

Value

`k0`	optimal number of upper order statistics, i.e. number of exceedances or data in the tail
`threshold`	the corresponding threshold
`tail.index`	the corresponding tail index by plugging in `k0` into the hill estimator

References

Danielsson, J. and Ergun, L.M. and de Haan, L. and de Vries, C.G. (2016). Tail Index Estimation: Quantile Driven Threshold Selection.

Examples

```
data(danish)
mindist(danish,method="mad")
```

PS *Sample Path Stability Algorithm*

Description

An Implementation of the heuristic algorithm for choosing the optimal sample fraction proposed in Caeiro & Gomes (2016), among others.

Usage

```
PS(data, j = 1)
```

Arguments

data	vector of sample data
j	digits to round to. Should be 0 or 1 (default)

Details

The algorithm searches for a stable region of the sample path, i.e. the plot of a tail index estimator with respect to k. This is done in two steps. First the estimation of the tail index for every k is rounded to j digits and the longest set of equal consecutive values is chosen. For this set the estimates are rounded to *j+2* digits and the mode of this subset is determined. The corresponding biggest k-value, denoted k0 here, is the optimal number of data in the tail.

Value

k0	optimal number of upper order statistics, i.e. number of exceedances or data in the tail
threshold	the corresponding threshold
tail.index	the corresponding tail index

References

Caeiro, J. and Gomes, M.I. (2016). Threshold selection in extreme value analysis. *Extreme Value Modeling and Risk Analysis:Methids and Applications*, 69–86.

Gomes, M.I. and Henriques-Rodrigues, L. and Fraga Alves, M.I. and Manjunath, B. (2013). Adaptive PORT-MVRB estimation: an empirical comparison of two heuristic algorithms. *Journal of Statistical Computation and Simulation*, **83**, 1129–1144.

Gomes, M.I. and Henriques-Rodrigues, L. and Miranda, M.C. (2011). Reduced-bias location-invariant extreme value index estimation: a simulation study. *Communications in Statistic-Simulation and Computation*, **40**, 424–447.

Examples

```
data(danish)
PS(danish)
```

---

qqestplot *QQ-Estimator-Plot*

Description

Plots the QQ-Estimator against the upper order statistics

Usage

```
qqestplot(data, kmin = 5, conf.int = FALSE)
```

Arguments

data	vector of sample data
kmin	gives the minimal k for which the graph is plotted. Default ist set to kmin=5
conf.int	logical. If FALSE (default) no confidence intervals are plotted

Details

The QQ-Estimator is a Tail Index Estimator based on regression diagnostics. Assuming a Pareto tail behaviour of the data at hand a QQ-Plot of the theoretical quantiles of an exponential distribution against the empirical quantiles of the log-data should lead to a straight line above some unknown upper order statistic k. The slope of this line is an estimator for the tail index. Computing this estimator via linear regression for every k the plot should stabilize for the correct number of upper order statistics, denoted k0 here.

Value

The plot shows the values of the QQ-Estimator with respect to k. See references for more information.

References

Kratz, M. and Resnick, S.I. (1996). The QQ-estimator and heavy tails. *Stochastic Models*, **12(4)**, 699–724.

Examples

```
data(danish)
qqestplot(danish)
```

qqgpd *QQ-Plot against the generalized Pareto distribution for given number of exceedances*

Description

Plots the empirical observations above a given threshold against the theoretical quantiles of a generalized Pareto distribution.

Usage

```
qqgpd(data, nextremes, scale, shape)
```

Arguments

data	vector of sample data
nextremes	number of exceedances
scale	scale parameter of GPD
shape	shape parameter of GPD

Details

If the fitted GPD model provides a reasonable approximation of the underlying sample data the empirical and theoretical quantiles should coincide. So plotting them against each other should result in a straight line. Deviations from that line speak for a bad model fit and against a GPD assumption.

Value

The straight red line gives the line of agreement. The dashed lines are simulated 95 percent confidence intervals. Therefor the fitted GPD model is simulated 1000 times using Monte Carlo. The sample size of each simulation equals the number of exceedances.

Examples

```
data=rexp(1000) #GPD with scale=1, shape=0
qqgpd(data,1000,1,0)
```

RT *Adaptive choice of the optimal sample fraction in tail index estimation*

Description

An implementation of the minimization criterion proposed in Reiss & Thomas (2007).

Usage

```
RT(data, beta = 0, kmin = 2)
```

Arguments

data	vector of sample data
beta	a factor for weighting the expression below. Default is set to beta=0
kmin	gives a minimum value for k. Default ist set to kmin=2

Details

The procedure proposed in Reiss & Thomas (2007) chooses the lowest upper order statistic k to minimize the expression 1/k sum_i=1^k i^beta |gamma_i-median(gamma_1,...,gamma_k)| or an alternative of that by replacing the absolute deviation with a squared deviation and the median just with gamma_k, where gamma denotes the Hill estimator

Value

k0	optimal number of upper order statistics, i.e. number of exceedances or data in the tail for both metrics, i.e. the absolute and squared deviation.
threshold	the corresponding thresholds.
tail.index	the corresponding tail indices

References

Reiss, R.-D. and Thomas, M. (2007). Statistical Analysis of Extreme Values: With Applications to Insurance, Finance, Hydrology and Other Fields. *Birkhauser, Boston*.

Examples

```
data(danish)
RT(danish)
```

sumplot *Sum Plot*

Description

An implementation of the so called sum plot proposed in de Sousa & Michailidis (2004)

Usage

```
sumplot(data, kmin = 5)
```

Arguments

data	vector of sample data
kmin	gives the minimal k for which the graph is plotted. Default ist set to kmin=5.

Details

The sum plot is based on the plot (k,S_k) with S_k:=k*gamma_k where gamma_k denotes the Hill estimator. So the sum plot and the Hill plot are statistically equivalent. The sum plot should be approximately linear for the k-values where gamma_k=gamma. So the linear part of the graph can be used as an estimator of the (inverse) tail index. The sum plot leads to the estimation of the slope while the classical Hill plot leads to estimation of the intercept. The optimal number of order statistics, also known as the threshold, can then be derived as the value k where the plot differs from a straight line with slope gamma. See references for more information.

Value

The plot shows the values of S_k=k*gamma_k for different k. See references for more information.

References

De Sousa, Bruno and Michailidis, George (2004). A diagnostic plot for estimating the tail index of a distribution. *Journal of Computational and Graphical Statistics* **13(4)**, 1–22.

Examples

```
data(danish)
sumplot(danish)
```

TH *Sequential Goodness of Fit Testing for the Generalized Pareto Distribution*

Description

An implementation of the sequential testing procedure proposed in Thompson et al. (2009) for automated threshold selection

Usage

```
TH(data, thresholds)
```

Arguments

data	vector of sample data
thresholds	a sequence of pre-defined thresholds to check for GPD assumption

Details

The procedure proposed in Thompson et al. (2009) is based on sequential goodness of fit testing. First, one has to choose a equally spaced grid of posssible thresholds. The authors recommend 100 thresholds between the 50 percent and 98 percent quantile of the data, provided there are enough observations left (about 100 observations above the last pre-defined threshold). Then the parameters of a GPD for each threshold are estimated. One can show that the differences of subsequent scale parameters are approximately normal distributed. So a Pearson chi-squared test for normality is applied to all the differences, striking the smallest thresholds out until the test is not rejected anymore.

Value

threshold	the threshold used for the test
num.above	the number of observations above the given threshold
p.values	raw p-values for the thresholds tested
ForwardStop	transformed p-values according to the ForwardStop criterion. See G'Sell et al (2016) for more information
StrongStop	transformed p-values according to the StrongStop criterion. See G'Sell et al (2016) for more information
est.scale	estimated scale parameter for the given threshold
est.shape	estimated shape parameter for the given threshold

References

Thompson, P. and Cai, Y. and Reeve, D. (2009). Automated threshold selection methods for extreme wave analysis. *Coastal Engineering*, **56**(10), 1013–1021.

G'Sell, M.G. and Wager, S. and Chouldechova, A. and Tibshirani, R. (2016). Sequential selection procedures and false discovery rate control. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **78**(2), 423–444.

Examples

```
data=rexp(1000)
u=seq(quantile(data,.1),quantile(data,.9),,100)
A=TH(data,u);A
```

Index