Package 'heavytails'

December 1, 2025

Title Estimators and Algorithms for Heavy-Tailed Distributions

Version 0.1.1

Description Implements the estimators and algorithms described in Chapters 8 and 9 of the book ``The Fundamentals of Heavy Tails: Properties, Emergence, and Estimation'' by Nair et al. (2022, ISBN:9781009053730). These include the Hill estimator, Moments estimator, Pickands estimator, Peaks-over-Threshold (POT) method, Power-law fit, and the Double Bootstrap algorithm.

License MIT + file LICENSE

URL https://github.com/0diraf/heavytails

Depends R (>= 3.5.0)

Encoding UTF-8

RoxygenNote 7.3.3

Imports stats

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Farid Rohan [aut, cre]

Maintainer Farid Rohan <frohan@ur.rochester.edu>

Repository CRAN

Date/Publication 2025-12-01 14:30:02 UTC

Contents

doublebootstrap *Double Bootstrap algorithm*

Description

This function implements the Double Bootstrap algorithm as described by in Chapter 9 by *Nair et al.* It applies bootstrapping to two samples of different sizes to choose the value of k that minimizes the mean square error.

Usage

```
doublebootstrap(
  data,
  n1 = -1,
  n2 = -1,
  r = 50,
  k_max_prop = 0.5,
  kvalues = 20,
  na.rm = FALSE
)
```

Arguments

`data`	A numeric vector of i.i.d. observations.
`n1`	A numeric scalar specifying the first bootstrap sample size, *Nair et al.* describe this as $n_1 = O(n^{1-\epsilon})$ for $\epsilon \in (0, 1/2)$. Hence, default value (if n1 = -1) is chosen as 0.9.
`n2`	A numeric scalar specifying the second bootstrap sample size
`r`	A numeric scalar specifying the number of bootstraps
`k_max_prop`	A numeric scalar. The max k as a proportion of the sample size. It might be computationally expensive to consider all possible k values from the data. Furthermore, lower k values can be noisy, while higher values can be biased. Hence, k here is limited to a specific proportion (by default 50%) of the data
`kvalues`	An integer specifying the length of sequence of candidate k values
`na.rm`	Logical. If `TRUE`, missing values (`NA`) are removed before analysis. Defaults to `FALSE`.

Details

Chapter 9 of *Nair et al.* specifically describes the Double Bootstrap algorithm for the Hill estimator.

The Hill Double Bootstrap method uses the Hill estimator as the first estimator

$$\hat{\xi}^{(1)}_{n,k} := \frac{1}{k}\sum_{i=1}^{k} \log\left(\frac{X_{(i)}}{X_{(k+1)}}\right)$$

And a second moments-based estimator:

$$\hat{\xi}^{(2)}_{n,k} = \frac{M_{n,k}}{2\hat{\xi}^{H}_{n,k}}$$

Where

$$M_{n,k} := \frac{1}{k}\sum_{i=1}^{k}\left(\log\left(\frac{X_{(i)}}{X_{(k+1)}}\right)\right)^2$$

The difference between these two $\hat{\xi}$ is given by:

$$|\hat{\xi}^{(1)}_{n,k} - \hat{\xi}^{(2)}_{n,k}| = \frac{|M_{n,k} - 2(\hat{\xi}^{H}_{n,k})^2|}{2|\hat{\xi}^{H}_{n,k}|}$$

The Hill bootstrap method selects $\hat{\kappa}$ in a way that minimizes the mean square error in the numerator by going through r bootstrap samples of different sizes n_1 and n_2.

$$\hat{\kappa}^*_1 := \text{arg min}_k \frac{1}{r}\sum_{j=1}^{r}(M_{n_1,k}(j) - 2(\hat{\xi}^{(1)}_{n_1,k}(j))^2)^2$$

This process is repeated to determine $\hat{\kappa}_2$ with the bootstrap sample of size n_2. The final $\hat{\kappa}$ is given by:

$$\hat{\kappa}^* = \frac{(\hat{\kappa}^*_1)^2}{\hat{\kappa}^*_2}\left(\frac{\log\hat{\kappa}^*_1}{2\log n_1 - \log\hat{\kappa}^*_1}\right)^{\frac{2(\log n_1 - \log\hat{\kappa}^*_1)}{\log n_1}}$$

Value

A named list containing the final results of the Double Bootstrap algorithm:

- `k`: The optimal number of top-order statistics $\hat{k}$ selected by minimizing the MSE.
- `alpha`: The estimated tail index $\hat{\alpha}$ (Hill estimator) corresponding to $\hat{k}$.

References

Danielsson, J., de Haan, L., Peng, L., & de Vries, C. G. (2001). Using a bootstrap method to choose the sample fraction in tail index estimation. *Journal of Multivariate Analysis*, **76**(2), 226–248. doi:10.1006/jmva.2000.1903

Nair, J., Wierman, A., & Zwart, B. (2022). *The Fundamentals of Heavy Tails: Properties, Emergence, and Estimation*. Cambridge University Press. (pp. 229-233) doi:10.1017/9781009053730

Examples

```
xmin <- 1
alpha <- 2
r <- runif(800, 0, 1)
x <- (xmin * r^(-1/(alpha)))
db_kalpha <- doublebootstrap(data = x, n1 = -1, n2 = -1, r = 5, k_max_prop = 0.5, kvalues = 20)
```

---

gpd_lg_likelihood *Negative Log likelihood of Generalized Pareto Distribution*

Description

Helper function for pot_estimator(). Returns the ξ and β that minimize the negative log-likelihood of the Generalized Pareto Distribution (GPD).

Usage

```
gpd_lg_likelihood(params, data)
```

Arguments

`params`	Vector containing initial values of ξ and β
`data`	Original dataset

Details

$$l(\xi, \beta) = -n\log(\beta) - (\frac{1}{\xi} + 1)\sum_{i=1}^{n}\log(1 + \xi\frac{x_i}{\beta})$$

Value

Negative log-likelihood of the GPD.

Examples

```
x <- rweibull(n=2000, shape = 0.8, scale = 1)
u <- 2 # Threshold
y <- x[x > u] - u
log_lik <- gpd_lg_likelihood(params = c(xi = 0.1, beta = 2), data = y)
```

`hill_estimator` *Hill Estimator*

Description

Hill estimator used to calculate the tail index (alpha) of input data.

Usage

```
hill_estimator(data, k, na.rm = FALSE)
```

Arguments

`data`	A numeric vector of i.i.d. observations.
`k`	An integer specifying the number of top order statistics to use (the size of the tail). Must be strictly less than the sample size.
`na.rm`	Logical. If `TRUE`, missing values (`NA`) are removed before analysis. Defaults to `FALSE`.

Details

$$\hat{\alpha}_H = \frac{1}{\frac{1}{k}\sum_{i=1}^{k} log(\frac{X_{(i)}}{X_{(k)}})}$$

where $X_{(1)} \geq X_{(2)} \geq \cdots \geq X_{(n)}$ are the order statistics of the data (descending order).

Value

A single numeric scalar: Hill estimator calculation of the tail index α.

References

Hill, B. M. (1975). A Simple General Approach to Inference About the Tail of a Distribution. *The Annals of Statistics*, **3**(5), 1163–1174. http://www.jstor.org/stable/2958370

Nair, J., Wierman, A., & Zwart, B. (2022). *The Fundamentals of Heavy Tails: Properties, Emergence, and Estimation*. Cambridge University Press. (pp. 203-205) doi:10.1017/9781009053730

Examples

```
xmin <- 1
alpha <- 2
r <- runif(800, 0, 1)
x <- (xmin * r^(-1/(alpha)))
hill <- hill_estimator(data = x, k = 5)
```

`moments_estimator` *Moments Estimator*

Description

Moments estimator to calculate ξ for the input data.

Usage

```
moments_estimator(data, k, na.rm = FALSE, eps = 1e-12)
```

Arguments

`data`	A numeric vector of i.i.d. observations.
`k`	An integer specifying the number of top order statistics to use (the size of the tail). Must be strictly less than the sample size.
`na.rm`	Logical. If `TRUE`, missing values (`NA`) are removed before analysis. Defaults to `FALSE`.
`eps`	numeric, factor added to the denominator to avoid division by zero. Default value is 1e-12.

Details

$$\hat{\xi}_{ME} = \underbrace{\hat{\xi}_{k,n}^{H,1}}_{T_1} + \underbrace{1 - \frac{1}{2}(1 - \frac{(\hat{\xi}_{k,n}^{H,1})^2}{\hat{\xi}_{k,n}^{H,2}})^{-1}}_{T_2}$$

Value

A single numeric scalar: Moments estimator calculation of the shape parameter ξ.

References

Dekkers, A. L. M., Einmahl, J. H. J., & De Haan, L. (1989). A Moment Estimator for the Index of an Extreme-Value Distribution. *The Annals of Statistics*, **17**(4), 1833–1855. http://www.jstor.org/stable/2241667

Nair, J., Wierman, A., & Zwart, B. (2022). *The Fundamentals of Heavy Tails: Properties, Emergence, and Estimation*. Cambridge University Press. (pp. 216-219) doi:10.1017/9781009053730

Examples

```
xmin <- 1
alpha <- 2
r <- runif(800, 0, 1)
x <- (xmin * r^(-1/(alpha)))
moments <- moments_estimator(data = x, k = 5)
```

`pickands_estimator` *Pickands Estimator*

Description

Pickands estimator to calculate ξ for the input data.

Usage

```
pickands_estimator(data, k, na.rm = FALSE)
```

Arguments

`data`	A numeric vector of i.i.d. observations.
`k`	An integer specifying the number of top order statistics to use (the size of the tail). Must be strictly less than the sample size.
`na.rm`	Logical. If `TRUE`, missing values (`NA`) are removed before analysis. Defaults to `FALSE`.

Details

$$\hat{\xi}_P = \frac{1}{\log 2} \log(\frac{X_{(k)} - X_{(2k)}}{X_{(2k)} - X_{(4k)}})$$

Value

A single numeric scalar: Pickands estimator calculation of the shape parameter ξ.

References

Pickands, J. (1975). Statistical Inference Using Extreme Order Statistics. *The Annals of Statistics*, **3**(1), 119–131. `http://www.jstor.org/stable/2958083`

Nair, J., Wierman, A., & Zwart, B. (2022). *The Fundamentals of Heavy Tails: Properties, Emergence, and Estimation*. Cambridge University Press. (pp. 219-221) doi:10.1017/9781009053730

Examples

```
xmin <- 1
alpha <- 2
r <- runif(800, 0, 1)
x <- (xmin * r^(-1/(alpha)))
pickands <- pickands_estimator(data = x, k = 5)
```

plfit *Power-law fit (PLFIT) Algorithm*

Description

This function implements the PLFIT algorithm as described by *Clauset et al.* to determine the value of $\hat{k}$. It minimizes the Kolmorogorov-Smirnoff (KS) distance between the empirical cumulative distribution function and the fitted power law.

Usage

```
plfit(data, kmax = -1, kmin = 2, na.rm = FALSE)
```

Arguments

`data`	A numeric vector of i.i.d. observations.
`kmax`	Maximum number of top-order statistics. If kmax = -1, then kmax=(n-1) where n is the length of dataset
`kmin`	Minimum number of top-order statistics to start with
`na.rm`	Logical. If `TRUE`, missing values (`NA`) are removed before analysis. Defaults to `FALSE`.

Details

$$D_{n,k} := \sup_{y \geq 1} | \frac{1}{k-1} \sum_{i=1}^{k-1} I(\frac{X_{(i)}}{X_{(k)}} > y) - y^{-\hat{\alpha}^H_{n,k}} |$$

The above equation, as described by *Nair et al.*, is implemented in this function with an Empirical CDF instead of the empirical survival function, which is mathematical equivalent since they are both complements of each other.

$$D_{n,k} := \sup_{y \geq 1} | \underbrace{\frac{1}{k-1} \sum_{i=1}^{k-1} I(\frac{X_{(i)}}{X_{(k)}} \leq y)}_{\text{Empirical CDF}} - \underbrace{(1 - y^{-\hat{\alpha}_{n,k}})}_{\text{Theoretical CDF}} |$$

$$\hat{k} = \text{argmin}(D_{n,k})$$

Value

A named list containing the results of the PLFIT algorithm:

- `k_hat`: The optimal number of top-order statistics $\hat{k}$.
- `alpha_hat`: The estimated power-law exponent $\hat{\alpha}$ corresponding to $\hat{k}$.
- `xmin_hat`: The minimum value $x_{\min} = X_{(\hat{k})}$ above which the power law is fitted.
- `ks_distance`: The minimum Kolmogorov-Smirnov distance $D_{n,k}$ found.

References

Clauset, A., Shalizi, C. R., & Newman, M. E. (2009). Power-law distributions in empirical data. *SIAM Review*, **51**(4), 661-703. doi:10.1137/070710111

Nair, J., Wierman, A., & Zwart, B. (2022). *The Fundamentals of Heavy Tails: Properties, Emergence, and Estimation*. Cambridge University Press. (pp. 227-229) doi:10.1017/9781009053730

Examples

```
xmin <- 1
alpha <- 2
r <- runif(800, 0, 1)
x <- (xmin * r^(-1/(alpha)))
plfit_values <- plfit(data = x, kmax = -1, kmin = 2)
```

`pot_estimator` *Peaks-over-threshold (POT) Estimator*

Description

This function chooses the $\hat{\xi}_k$ and $\hat{\beta}$ that minimize the negative log likelihood of the Generalized Pareto Distribution (GPD).

Usage

```
pot_estimator(data, u, start_xi = 0.1, start_beta = NULL, na.rm = FALSE)
```

Arguments

`data`	A numeric vector of i.i.d. observations.
`u`	A numeric scalar that specifies the threshold value to calculate excesses
`start_xi`	Initial value of ξ to pass to the optimizer
`start_beta`	Initial value of β to pass to the optimizer
`na.rm`	Logical. If `TRUE`, missing values (`NA`) are removed before analysis. Defaults to `FALSE`.

Details

The PDF of a excess data point x_i is given by:

$$f(x_i; \xi, \beta) = \frac{1}{\beta}\left(1 + \xi\frac{x_i}{\beta}\right)^{-\left(\frac{1}{\xi}+1\right)}$$

If we apply log to the above equation we get:

$$l(x_i;\xi,\beta) = -\log(\beta) - (\frac{1}{\xi}+1)\log(1+\xi\frac{x_i}{\beta})$$

For all excess data points n:

$$l(\xi,\beta) = \sum_{i=1}^{n}(-\log(\beta) - (\frac{1}{\xi}+1)\log(1+\xi\frac{x_i}{\beta}))$$

$$l(\xi,\beta) = -n\log(\beta) - (\frac{1}{\xi}+1)\sum_{i=1}^{n}\log(1+\xi\frac{x_i}{\beta})$$

We can thus minimize $-l(\xi,\beta)$. The parameters ξ and β that minimize the negative log likelihood are the same that maximize the log likelihood. Hence, by using the excesses, we are able to determine ξ and β that best fit the tail of the data.

There is also the case to consider when $\xi = 0$ which results in an exponential distribution. The total log likelihood in such a case is:

$$l(0,\beta) = -n\log(\beta) - \frac{1}{\beta}\sum_{i=1}^{n}x_i$$

Value

An unnamed numeric vector of length 2 containing the estimated Generalized Pareto Distribution (GPD) parameters that minimize the negative log likelihood: ξ (shape/tail index) and β (scale parameter).

References

Davison, A. C., & Smith, R. L. (1990). Models for exceedances over high thresholds. *Journal of the Royal Statistical Society: Series B (Methodological)*, **52**(3), 393-425. doi:10.1111/j.2517-6161.1990.tb01796.x

Balkema, A. A., & de Haan, L. (1974). Residual life time at great age. *The Annals of Probability*, **2**(5), 792-804. doi:10.1214/aop/1176996548

Pickands, J. (1975). Statistical Inference Using Extreme Order Statistics. *The Annals of Statistics*, **3**(1), 119–131. http://www.jstor.org/stable/2958083

Nair, J., Wierman, A., & Zwart, B. (2022). *The Fundamentals of Heavy Tails: Properties, Emergence, and Estimation*. Cambridge University Press. (pp. 221-226) doi:10.1017/9781009053730

Examples

```
x <- rweibull(n=800, shape = 0.8, scale = 1)
values <- pot_estimator(data = x, u = 2, start_xi = 0.1, start_beta = NULL)
```

Index