Package: heavytails (via r-universe)

November 11, 2026

Title Estimators and Algorithms for Heavy-Tailed Distributions

Version 0.2.0

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

Depends R (>= 3.5.0)

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports graphics, stats

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Repository https://0diraf.r-universe.dev

Date/Publication 2026-03-27 20:24:25 UTC

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

RemoteRef HEAD

RemoteSha 0531ed6c1f42521bd4598e1753db6eb24c73fa92

Contents

doublebootstrap 2
dpareto 4
hill_estimator 5
hill_plot 6
ks_gof 7
ks_xmin 8
lr_test_pareto 9

mle_pareto 10
moments_estimator 11
moments_plot 12
pareto_cdf 13
pickands_estimator 14
pickands_plot 15
plfit 16
pot_estimator 17
qq_pareto 19
rank_plot 20
rpareto 21
wls_pareto 22

Index **24**

`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

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

And a second moments-based estimator:

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

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:

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

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.

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

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)
```

`dpareto` *Pareto Density*

Description

Computes the probability density function of the Pareto(x_m, α) distribution:

$$f(x) = \frac{\alpha\, x_m^{\alpha}}{x^{\alpha+1}}, \quad x \geq x_m$$

Usage

```
dpareto(x, alpha, xm)
```

Arguments

`x`	A numeric vector of quantiles.
`alpha`	A positive numeric scalar: tail index.
`xm`	A positive numeric scalar: scale parameter (lower bound).

Value

A numeric vector of density values (zero for $x < x_m$).

Examples

```
dpareto(x = c(1, 2, 5), alpha = 2, xm = 1)
```

`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

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

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)
```

`hill_plot` *Hill Plot*

Description

Plots the Hill estimator of the tail index $\hat{\alpha}$ as a function of the number of top order statistics k. A stable plateau in this plot is used to visually select a suitable value of k.

Usage

```
hill_plot(data, k_range = NULL, alpha_true = NULL, na.rm = FALSE, ...)
```

Arguments

`data`	A numeric vector of i.i.d. observations.
`k_range`	An integer vector specifying which values of k to evaluate. If `NULL` (default), uses `2:floor(length(data)/2)`.
`alpha_true`	Optional numeric scalar. If supplied, a horizontal reference line at the true α is added to the plot.
`na.rm`	Logical. If `TRUE`, missing values (`NA`) are removed before analysis. Defaults to `FALSE`.
`...`	Additional arguments passed to `plot`.

Value

A `data.frame` with columns `k` and `alpha_hat`, returned invisibly. Users who prefer ggplot2 can capture this output and re-plot.

References

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

Examples

```
set.seed(1)
x <- rpareto(800, alpha = 2, xm = 1)
result <- hill_plot(x)
```

`ks_gof` *Goodness-of-Fit Test for the Pareto Distribution*

Description

Tests whether a Pareto(x_m, α) distribution is a good fit for the data by computing a bootstrap p-value for the Kolmogorov-Smirnov (KS) statistic (Step 2 of the Clauset et al. pipeline, §8.5).

Usage

```
ks_gof(data, alpha, xm, n_boot = 1000, na.rm = FALSE)
```

Arguments

`data`	A numeric vector of i.i.d. observations.
`alpha`	A positive numeric scalar: the Pareto tail index. Typically obtained from `mle_pareto` or `plfit`.
`xm`	A positive numeric scalar: the lower bound. Only `data[data >= xm]` is used.
`n_boot`	A positive integer: number of bootstrap replicates. Default `1000`.
`na.rm`	Logical. If `TRUE`, missing values (`NA`) are removed before analysis. Defaults to `FALSE`.

Details

The p-value is the fraction of bootstrap KS statistics that exceed the observed KS statistic. A large p-value (e.g., > 0.1) means the Pareto hypothesis cannot be rejected.

Value

A named list with elements:

- `ks_statistic`: Observed KS distance.
- `p_value`: Bootstrap p-value.
- `n_boot`: Number of bootstrap replicates used.
- `n`: Number of observations used (those $\geq x_m$).

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. 194-196) doi:10.1017/9781009053730

Examples

```
set.seed(1)
x <- rpareto(n = 500, alpha = 2, xm = 1)
fit <- mle_pareto(x)
ks_gof(x, alpha = fit$alpha, xm = fit$xm, n_boot = 100)
```

`ks_xmin` *Estimate the Power-Law Lower Bound via KS Minimization*

Description

Estimates the lower bound $\hat{x}_m$ of a power-law regime by finding the order statistic that minimizes the Kolmogorov-Smirnov distance between the empirical distribution and the fitted Pareto (Step 1 of the Clauset et al. pipeline, §8.5).

Usage

```
ks_xmin(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 to consider. If `-1` (default), uses `n - 1`.
`kmin`	Minimum number of top order statistics. Default `2`.
`na.rm`	Logical. If `TRUE`, missing values (`NA`) are removed before analysis. Defaults to `FALSE`.

Details

This function extracts and exposes the core loop from `plfit`, allowing $\hat{x}_m$ estimation as a standalone step — useful as input to `mle_pareto`, `wls_pareto`, or `ks_gof`.

Value

A named list with elements:

- `xm`: Estimated lower bound $\hat{x}_m = X_{(\hat{k})}$.
- `ks_distance`: Minimum KS distance achieved.
- `k_hat`: The optimal $\hat{k}$.

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

Examples

```
set.seed(1)
x <- rpareto(n = 500, alpha = 2, xm = 1)
ks_xmin(x)
```

---

lr_test_pareto *Likelihood Ratio Test: Pareto vs. Alternative Distributions*

Description

Compares the Pareto distribution fit against one or more alternative distributions using the Vuong likelihood ratio test for non-nested models (§8.5, Step 3; Clauset et al. 2009, §3.3).

Usage

```
lr_test_pareto(
  data,
  xm = NULL,
  alternatives = c("exponential", "lognormal", "weibull"),
  na.rm = FALSE
)
```

Arguments

`data`	A numeric vector of i.i.d. observations.
`xm`	A positive numeric scalar: lower bound. Only `data[data >= xm]` is used.
`alternatives`	A character vector naming the distributions to compare against. Supported: `"exponential"`, `"lognormal"`, `"weibull"`.
`na.rm`	Logical. If `TRUE`, missing values (`NA`) are removed before analysis. Defaults to `FALSE`.

Details

For each alternative, the log-likelihood ratio $LR = \ell_{\text{Pareto}} - \ell_{\text{alternative}}$ is computed. The Vuong test statistic checks whether the mean per-observation log-likelihood ratio is significantly different from zero. A positive LR with a small p-value indicates the Pareto is preferred; a negative LR with a small p-value indicates the alternative is preferred.

Value

A `data.frame` with one row per alternative and columns:

- `alternative`: Name of the alternative distribution.
- `ll_pareto`: Pareto log-likelihood.
- `ll_alternative`: Alternative log-likelihood.

- `lr_statistic`: Vuong test statistic (z-score).
- `p_value`: Two-sided p-value.
- `preferred`: `"pareto"` or the alternative name.

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

Vuong, Q. H. (1989). Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica*, **57**(2), 307-333.

Examples

```
set.seed(1)
x <- rpareto(n = 500, alpha = 2, xm = 1)
lr_test_pareto(x, xm = 1)
```

`mle_pareto` *Parametric MLE for the Pareto Distribution*

Description

Estimates the tail index α of a Pareto(x_m, α) distribution via maximum likelihood (Theorem 8.1 of Nair et al.).

Usage

```
mle_pareto(data, xm = NULL, bias_corrected = TRUE, na.rm = FALSE)
```

Arguments

`data`	A numeric vector of i.i.d. observations.
`xm`	Optional positive numeric scalar. Lower bound of the Pareto support. If `NULL` (default), `min(data)` is used.
`bias_corrected`	Logical. If `TRUE` (default), applies the finite-sample bias correction described in §8.3.
`na.rm`	Logical. If `TRUE`, missing values (`NA`) are removed before analysis. Defaults to `FALSE`.

Details

The MLE is:

$$\hat{\alpha} = \frac{n}{\sum_{i=1}^{n} \log(X_i / x_m)}$$

Unlike the Hill estimator (which uses only the top k order statistics), this estimator uses all n observations and assumes the entire sample follows a Pareto distribution with known lower bound x_m.

A finite-sample bias-corrected version (§8.3) uses $n - 1$ in the numerator:

$$\hat{\alpha}^* = \frac{n - 1}{\sum_{i=1}^{n} \log(X_i / x_m)}$$

Value

A named list with elements:

- `alpha`: Estimated tail index.
- `xm`: The lower bound used.
- `n`: Number of observations used (those $\geq x_m$).
- `bias_corrected`: Logical indicating whether bias correction was applied.

References

Nair, J., Wierman, A., & Zwart, B. (2022). *The Fundamentals of Heavy Tails: Properties, Emergence, and Estimation*. Cambridge University Press. (pp. 162-167) doi:10.1017/9781009053730

Examples

```
set.seed(1)
x <- rpareto(n = 1000, alpha = 2, xm = 1)
mle_pareto(x)
```

`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)
```

---

moments_plot *Moments Estimator Plot*

Description

Plots the Moments estimator of the shape parameter $\hat{\xi}$ as a function of the number of top order statistics k. A stable plateau indicates a suitable choice of k.

Usage

```
moments_plot(data, k_range = NULL, xi_true = NULL, na.rm = FALSE, ...)
```

Arguments

data	A numeric vector of i.i.d. observations.
k_range	An integer vector specifying which values of k to evaluate. If NULL (default), uses 2:floor(length(data)/2).
xi_true	Optional numeric scalar. If supplied, a horizontal reference line at the true ξ is added.
na.rm	Logical. If TRUE, missing values (NA) are removed before analysis. Defaults to FALSE.
...	Additional arguments passed to plot.

Value

A data.frame with columns k and xi_hat, returned invisibly.

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.

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

Examples

```
set.seed(1)
x <- rpareto(800, alpha = 2, xm = 1)
moments_plot(x)
```

---

pareto_cdf *Pareto CDF*

Description

Computes the cumulative distribution function of the Pareto(x_m, α) distribution:

$$F(x) = 1 - \left(\frac{x}{x_m}\right)^{-\alpha}, \quad x \geq x_m$$

Usage

```
pareto_cdf(x, xmin, alpha)
```

Arguments

x	A numeric vector of quantiles.
xmin	A positive numeric scalar: scale parameter (lower bound).
alpha	A positive numeric scalar: tail index.

Value

A numeric vector of CDF values in $[0, 1]$.

Examples

```
pareto_cdf(x = c(1, 2, 5), xmin = 1, alpha = 2)
```

---

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)
```

---

pickands_plot *Pickands Estimator Plot*

Description

Plots the Pickands estimator of the shape parameter $\hat{\xi}$ as a function of the number of top order statistics k. A stable plateau indicates a suitable choice of k.

Usage

```
pickands_plot(data, k_range = NULL, xi_true = NULL, na.rm = FALSE, ...)
```

Arguments

`data`	A numeric vector of i.i.d. observations.
`k_range`	An integer vector specifying which values of k to evaluate. If `NULL` (default), uses `2:floor(length(data)/4 - 1)`.
`xi_true`	Optional numeric scalar. If supplied, a horizontal reference line at the true ξ is added.
`na.rm`	Logical. If `TRUE`, missing values (`NA`) are removed before analysis. Defaults to `FALSE`.
`...`	Additional arguments passed to `plot`.

Details

The Pickands estimator requires $4k < n$, so the default `k_range` upper bound is `floor(n/4) - 1`.

Value

A `data.frame` with columns `k` and `xi_hat`, returned invisibly.

References

Pickands, J. (1975). Statistical Inference Using Extreme Order Statistics. *The Annals of Statistics*, **3**(1), 119–131.

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

Examples

```
set.seed(1)
x <- rpareto(800, alpha = 2, xm = 1)
pickands_plot(x)
```

`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)
```

`qq_pareto` *Pareto QQ Plot*

Description

Produces a QQ plot comparing the empirical quantiles of the data (filtered to $x \geq x_m$) against the theoretical quantiles of a Pareto(x_m, α) distribution. Points falling close to the 45-degree reference line indicate a good Pareto fit.

Usage

```
qq_pareto(data, alpha, xm = NULL, na.rm = FALSE, ...)
```

Arguments

`data`	A numeric vector of i.i.d. observations.
`alpha`	A positive numeric scalar: the Pareto tail index (as returned by `hill_estimator` or `mle_pareto`).
`xm`	Optional numeric scalar. Lower threshold; only data $\geq x_m$ are used. If `NULL` (default), `min(data)` is used.
`na.rm`	Logical. If `TRUE`, missing values (`NA`) are removed before analysis. Defaults to `FALSE`.
`...`	Additional arguments passed to `plot`.

Details

The theoretical quantile for the i-th order statistic is:

$$q_i = x_m \left(\frac{n - i + 1}{n + 1} \right)^{-1/\alpha}$$

Value

A `data.frame` with columns `empirical` and `theoretical`, returned invisibly.

References

Nair, J., Wierman, A., & Zwart, B. (2022). *The Fundamentals of Heavy Tails: Properties, Emergence, and Estimation*. Cambridge University Press. (pp. 191-194) doi:10.1017/9781009053730

Examples

```
set.seed(1)
x <- rpareto(800, alpha = 2, xm = 1)
qq_pareto(x, alpha = 2, xm = 1)
```

`rank_plot` *Rank Plot (Log-Log CCDF)*

Description

Plots the empirical complementary CDF (CCDF) of the data on a log-log scale. A power-law distribution appears as a straight line on this plot. If a fitted `plfit()` result is supplied, the theoretical Pareto CCDF is overlaid.

Usage

```
rank_plot(data, fit = NULL, log_scale = TRUE, na.rm = FALSE, ...)
```

Arguments

`data`	A numeric vector of i.i.d. observations.
`fit`	Optional. A list returned by `plfit`, used to overlay the fitted Pareto line. Must contain `alpha_hat` and `xmin_hat`.
`log_scale`	Logical. If `TRUE` (default), axes are log-transformed (i.e., $\log x$ vs $\log \hat{F}^c$).
`na.rm`	Logical. If `TRUE`, missing values (`NA`) are removed before analysis. Defaults to `FALSE`.
`...`	Additional arguments passed to `plot`.

Value

A `data.frame` with columns `x` and `ccdf`, returned invisibly.

References

Nair, J., Wierman, A., & Zwart, B. (2022). *The Fundamentals of Heavy Tails: Properties, Emergence, and Estimation*. Cambridge University Press. (pp. 176-179) doi:10.1017/9781009053730

Examples

```
set.seed(1)
x <- rpareto(800, alpha = 2, xm = 1)
fit <- plfit(x)
rank_plot(x, fit = fit)
```

`rpareto` *Generate Pareto Random Variates*

Description

Generates n random samples from a Pareto(x_m, α) distribution via inverse CDF: $x_m \cdot U^{-1/\alpha}$ where $U \sim \text{Uniform}(0, 1)$.

Usage

```
rpareto(n, alpha, xm)
```

Arguments

`n`	A non-negative integer: number of samples to generate.
`alpha`	A positive numeric scalar: tail index.
`xm`	A positive numeric scalar: scale parameter (lower bound).

Value

A numeric vector of length `n`.

References

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

Examples

```
x <- rpareto(n = 500, alpha = 2, xm = 1)
```

`wls_pareto` *Weighted Least Squares Estimator for the Pareto Tail Index*

Description

Estimates the Pareto tail index α via weighted least squares (WLS) regression on the log-log rank plot (Theorem 8.5 of Nair et al.). The WLS weights $w_i = 1/\log(X_{(i)}/x_m)$ downweight noisy tail observations relative to OLS, recovering the MLE asymptotically.

Usage

```
wls_pareto(data, xm = NULL, plot = TRUE, na.rm = FALSE, ...)
```

Arguments

`data`	A numeric vector of i.i.d. observations.
`xm`	Optional positive numeric scalar. Lower bound. If `NULL` (default), `min(data)` is used.
`plot`	Logical. If `TRUE` (default), draws the log-log rank plot with fitted WLS and OLS lines.
`na.rm`	Logical. If `TRUE`, missing values (`NA`) are removed before analysis. Defaults to `FALSE`.
`...`	Additional graphical arguments passed to `plot` (only used when `plot = TRUE`).

Details

The WLS estimate is:

$$\hat{\alpha}_{WLS} = -\frac{\sum_i w_i \log(\hat{F}_i^c)\log(X_{(i)}/x_m)}{\sum_i w_i (\log(X_{(i)}/x_m))^2}$$

If `plot = TRUE`, the rank plot is drawn with both the WLS and OLS fitted lines, reproducing Figure 8.9 of Nair et al.

Value

A named list with elements:

- `alpha_wls`: WLS estimate of the tail index.
- `alpha_ols`: OLS estimate (unweighted) for comparison.
- `xm`: The lower bound used.

References

Nair, J., Wierman, A., & Zwart, B. (2022). *The Fundamentals of Heavy Tails: Properties, Emergence, and Estimation*. Cambridge University Press. (pp. 167-173) doi:10.1017/9781009053730

Examples

```
set.seed(1)
x <- rpareto(n = 500, alpha = 2, xm = 1)
wls_pareto(x)
```

Index

doublebootstrap, 2
dpareto, 4

hill_estimator, 5, *19*
hill_plot, 6

ks_gof, 7, *8*
ks_xmin, 8

lr_test_pareto, 9

mle_pareto, *7, 8*, 10, *19*
moments_estimator, 11
moments_plot, 12

pareto_cdf, 13
pickands_estimator, 14
pickands_plot, 15
plfit, *7, 8*, 16, *20*
plot, *6*, *13*, *15*, *19, 20*, *22*
pot_estimator, 17

qq_pareto, 19

rank_plot, 20
rpareto, 21

wls_pareto, *8*, 22