Package: Rsubbotools (via r-universe)

November 3, 2026

Title Fast Estimation of Subbottin and AEP Distributions (Generalized Error Distribution)

Version 0.0.1

Description Create densities, probabilities, random numbers, quantiles, and maximum likelihood estimation for several distributions, mainly the symmetric and asymmetric power exponential (AEP), a.k.a. the Subbottin family of distributions, also known as the generalized error distribution. Estimation is made using the design of Bottazzi (2004) <https://ideas.repec.org/p/ssa/lemwps/2004-14.html>, where the likelihood is maximized by several optimization procedures using the 'GNU Scientific Library (GSL)', translated to 'C++' code, which makes it both fast and accurate. The package also provides methods for the gamma, Laplace, and Asymmetric Laplace distributions.

License GPL-3

Encoding UTF-8

BugReports https://github.com/nettoyoussef/Rsubbotools/issues

RoxygenNote 7.3.2

Suggests testthat (>= 2.1.0), usethis, rmarkdown

Imports Rcpp

LinkingTo Rcpp, RcppGSL

NeedsCompilation yes

Author Elias Haddad [aut, cre] (ORCID: <https://orcid.org/0000-0003-1863-1022>), Giulio Bottazzi [aut] (ORCID: <https://orcid.org/0000-0003-2565-4732>), Fernando J. Von Zuben [aut, ths] (ORCID: <https://orcid.org/0000-0002-4128-5415>), Jochen Voss [ctb], J.D. Lamb [ctb], Brian Gough [ctb]

Maintainer Elias Haddad <eliasynetto@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2025-07-22 15:41:26 UTC

RemoteUrl https://github.com/cran/Rsubbotools

RemoteRef HEAD

RemoteSha 5eb2171ed0893985a5c7aaa657d8d2b85166aa01

Contents

alaplafit . . . 3
dalaplace . . . 4
dasubbo . . . 5
dlaplace . . . 6
dpower . . . 6
dsep . . . 7
dsubbo . . . 8
laplafit . . . 8
palaplace . . . 9
pasubbo . . . 10
plaplace . . . 11
ppower . . . 12
psep . . . 12
psubbo . . . 13
qalaplace . . . 14
qasubbo . . . 15
qlaplace . . . 16
qpower . . . 16
qsep . . . 17
qsubbo . . . 18
ralaplace . . . 19
rasubbo . . . 20
rasubbo_orig . . . 21
rgamma_c . . . 22
rlaplace . . . 23
rpower . . . 23
rsubbo . . . 24
sepfit . . . 25
subboafish . . . 27
subboafit . . . 28
subbofish . . . 30
subbofit . . . 31
subbolafit . . . 33

Index **36**

`alaplafit` *Fit an Asymmetric Laplace Distribution via maximum likelihood*

Description

`alaplafit` returns the parameters, standard errors. negative log-likelihood and covariance matrix of the Asymmetric Laplace Distribution for a sample. See details below.

Usage

```
alaplafit(data, verb = 0L, interv_step = 10L, provided_m_ = NULL)
```

Arguments

`data`	(NumericVector) - the sample used to fit the distribution.
`verb`	(int) - the level of verbosity. Select one of: • 0 just the final result • 1 details of optim. routine
`interv_step`	int - the number of intervals to be explored after the last minimum was found in the interval optimization. Default is 10.
`provided_m_`	NumericVector - if NULL, the m parameter is estimated by the routine. If numeric, the estimation fixes m to the given value.

Details

The Asymmetric Laplace distribution is a distribution controlled by three parameters, with formula:

$$f(x; a_l, a_r, m) = \frac{1}{A} e^{-|\frac{x-m}{a_l}|}, x < m$$

$$f(x; a_l, a_r, m) = \frac{1}{A} e^{-|\frac{x-m}{a_r}|}, x > m$$

with:

$$A = a_l + a_r$$

where $a*$ are scale parameters, and m is a location parameter. It is basically derived from the Asymmetric Exponential Power distribution by setting $b_l = b_r = b$. The estimations are produced by maximum likelihood, where analytical formulas are available for the $a*$ parameters. The m parameter is found by an iterative method, using the median as the initial guess. The method explore intervals around the last minimum found, similar to the `subboafit` routine. Details on the method can be found on the package vignette.

Value

a list containing the following items:

- "dt" - dataset containing parameters estimations and standard deviations.
- "log-likelihood" - negative log-likelihood value.
- "matrix" - the covariance matrix for the parameters.

Examples

```
sample_subbo <- rpower(1000, 1, 1)
alaplafit(sample_subbo)
```

`dalaplace` *Returns density from Asymmetric Laplace Distribution*

Description

The `dalaplace` returns the density at point x for the Asymmetric Laplace distribution with parameters $a*$ and m.

Usage

```
dalaplace(x, m = 0, al = 1, ar = 1)
```

Arguments

`x`	(numeric) - value in the range $(-\infty, \infty)$ to evaluate the density.
`m`	(numeric) - location parameter.
`al`, `ar`	(numeric) - scale parameters. Must be in the range $(0, \infty)$.

Details

The Asymmetric Laplace distribution is a distribution controlled by three parameters, with formula:

$$f(x; a_l, a_r, m) = \frac{1}{A} e^{-|\frac{x-m}{a_l}|}, x < m$$

$$f(x; a_l, a_r, m) = \frac{1}{A} e^{-|\frac{x-m}{a_r}|}, x > m$$

with:

$$A = a_l + a_r$$

where $a*$ are scale parameters, and m is a location parameter. It is basically derived from the Asymmetric Exponential Power distribution by setting $b_l = b_r = b$.

Value

a vector containing the values for the densities.

dasubbo *Returns density from the AEP Distribution*

Description

The dasubbo returns the density at point x for the AEP distribution with parameters $a*$, $b*$, m. Notice that the function can generate RNGs for both the subboafit and subbolafit routines.

Usage

```
dasubbo(x, m = 0, al = 1, ar = 1, bl = 2, br = 2)
```

Arguments

x	(numeric) - value in the range $(-\infty, \infty)$ to evaluate the density.
m	(numeric) - location parameter.
al, ar	(numeric) - scale parameters. Must be in the range $(0, \infty)$.
bl, br	(numeric) - shape parameters. Must be in the range $(0, \infty)$.

Details

The AEP is a exponential power distribution controlled by five parameters, with formula:

$$f(x; a_l, a_r, b_l, b_r, m) = \frac{1}{A} e^{-\frac{1}{b_l}|\frac{x-m}{a_l}|^{b_l}}, x < m$$

$$f(x; a_l, a_r, b_l, b_r, m) = \frac{1}{A} e^{-\frac{1}{b_r}|\frac{x-m}{a_r}|^{b_r}}, x > m$$

with:

$$A = a_l b_l^{1/b_l} \Gamma(1 + 1/b_l) + a_r b_r^{1/b_r} \Gamma(1 + 1/b_r)$$

where l and r represent left and right tails, $a*$ are scale parameters, $b*$ control the tails (lower values represent fatter tails), and m is a location parameter.

Value

a vector containing the values for the densities.

dlaplace *Returns density from Laplace Distribution*

Description

The dlaplace returns the density at point x for the Laplace distribution with parameters a and m.

Usage

```
dlaplace(x, m = 0, a = 1)
```

Arguments

x	(numeric) - value in the range $(-\infty, \infty)$ to evaluate the density.
m	(numeric) - location parameter.
a	(numeric) - scale parameter. Must be in the range $(0, \infty)$.

Details

The Laplace distribution is a distribution controlled by two parameters, with formula:

$$f(x; a, m) = \frac{1}{2a} e^{-|\frac{x-m}{a}|}$$

where a is a scale parameter, and m is a location parameter.

Value

a vector containing the values for the densities.

dpower *Returns density from EP Distribution*

Description

The dpower returns the density at point x for the Exponential Power distribution with parameters a, b and m.

Usage

```
dpower(x, m = 0, a = 1, b = 2)
```

Arguments

x	(numeric) - value in the range $(-\infty, \infty)$ to evaluate the density.
m	(numeric) - location parameter. Must be in the range $(-\infty, \infty)$.
a	(numeric) - scale parameter. Must be in the range $(0, \infty)$.
b	(numeric) - shape parameter. Must be in the range $(0, \infty)$.

Details

The Exponential Power distribution (EP) is given by the function:

$$f(a,b) = \frac{1}{2a\Gamma(1+1/b)} e^{-|(x-m)/a|^b}, -\infty < x < \infty$$

. where b is a shape parameter, a is a scale parameter, m is a location parameter and Γ represents the gamma function.

Value

a vector containing the values for the densities.

`dsep` *Returns density from Skewed Exponential Power distribution*

Description

The `dsep` returns the density at point x for the Gamma distribution with parameters a, b.

Usage

```
dsep(x, m = 0, a = 1, b = 1, lambda = 1)
```

Arguments

`x`	(numeric) - value in the range $(-\infty, \infty)$ to evaluate the density.
`m`	(numeric) - location parameter. Must be in the range $(-\infty, \infty)$.
`a`	(numeric) - scale parameter. Must be in the range $(0, \infty)$.
`b`	(numeric) - shape parameter. Must be in the range $(0, \infty)$.
`lambda`	(numeric) - skewness parameter. Must be in the range $(-\infty, \infty)$.

Details

The SEP is a exponential power distribution controlled by four parameters, with formula:

$$f(x; m, b, a, \lambda) = 2\Phi(w)e^{-|z|^b/b}/(c)$$

where:

$$z = (x-m)/a$$
$$w = sign(z)|z|^{(b/2)}\lambda\sqrt{2/b}$$
$$c = 2ab^{(1/b)-1}\Gamma(1/b)$$

with Φ the cumulative normal distribution with mean zero and variance one.

Value

a vector containing the values for the densities.

dsubbo *Returns density from Subbotin Distribution*

Description

The dsubbo returns the density at point x for the Subbotin distribution with parameters a, b, m.

Usage

```
dsubbo(x, m = 0, a = 1, b = 2)
```

Arguments

x	(numeric) - value in the range $(-\infty, \infty)$ to evaluate the density.
m	(numeric) - location parameter.
a	(numeric) - scale parameter. Must be in the range $(0, \infty)$.
b	(numeric) - shape parameter. Must be in the range $(0, \infty)$.

Details

The Subbotin distribution is a exponential power distribution controlled by three parameters, with formula:

$$f(x; a, b, m) = \frac{1}{A} e^{-\frac{1}{b}|\frac{x-m}{a}|^b}$$

with:

$$A = 2ab^{1/b}\Gamma(1 + 1/b)$$

where a is a scale parameter, b controls the tails (lower values represent fatter tails), and m is a location parameter.

Value

a vector containing the values for the densities.

laplafit *Fit a Laplace Distribution via maximum likelihood*

Description

laplafit returns the parameters, standard errors. negative log-likelihood and covariance matrix of the Laplace Distribution for a sample. See details below.

Usage

```
laplafit(data, verb = 0L, interv_step = 10L, provided_m_ = NULL)
```

Arguments

`data`	(NumericVector) - the sample used to fit the distribution.
`verb`	(int) - the level of verbosity. Select one of: • 0 just the final result • 1 details of optim. routine
`interv_step`	int - the number of intervals to be explored after the last minimum was found in the interval optimization. Default is 10.
`provided_m_`	NumericVector - if NULL, the m parameter is estimated by the routine. If numeric, the estimation fixes m to the given value.

Details

The Laplace distribution is a distribution controlled by two parameters, with formula:

$$f(x; a, m) = \frac{1}{2a} e^{-\left|\frac{x-m}{a}\right|}$$

where a is a scale parameter, and m is a location parameter. The estimations are produced by maximum likelihood, where analytical formulas are available. Details on the method can be found on the package vignette.

Value

a list containing the following items:

- "dt" - dataset containing parameters estimations and standard deviations.
- "log-likelihood" - negative log-likelihood value.
- "matrix" - the covariance matrix for the parameters.

Examples

```
sample_subbo <- rpower(1000, 1, 1)
laplafit(sample_subbo)
```

---

palaplace *Returns CDF from Asymmetric Laplace Distribution*

Description

The `palaplace` returns the Cumulative Distribution Function at point x for the Asymmetric Laplace distribution with parameters $a*$ and m.

Usage

```
palaplace(x, m = 0, al = 1, ar = 1)
```

Arguments

x	(numeric) - value in the range $(-\infty, \infty)$ to evaluate the density.
m	(numeric) - location parameter.
al, ar	(numeric) - scale parameters. Must be in the range $(0, \infty)$.

Details

The Asymmetric Laplace distribution is a distribution controlled by three parameters, with formula:

$$f(x; a_l, a_r, m) = \frac{1}{A} e^{-|\frac{x-m}{a_l}|}, x < m$$

$$f(x; a_l, a_r, m) = \frac{1}{A} e^{-|\frac{x-m}{a_r}|}, x > m$$

with:

$$A = a_l + a_r$$

where $a*$ are scale parameters, and m is a location parameter. It is basically derived from the Asymmetric Exponential Power distribution by setting $b_l = b_r = b$.

Value

a vector containing the values for the probabilities.

---

pasubbo *Returns CDF from the AEP Distribution*

Description

The pasubbo returns the Cumulative Distribution Function at point x for the AEP distribution with parameters $a*$, $b*$, m.

Usage

```
pasubbo(x, m = 0, al = 1, ar = 1, bl = 2, br = 2)
```

Arguments

x	(numeric) - value in the range $(-\infty, \infty)$ to evaluate the density.
m	(numeric) - location parameter.
al, ar	(numeric) - scale parameters. Must be in the range $(0, \infty)$.
bl, br	(numeric) - shape parameters. Must be in the range $(0, \infty)$.

Details

The AEP is a exponential power distribution controlled by five parameters, with formula:

$$f(x; a_l, a_r, b_l, b_r, m) = \frac{1}{A} e^{-\frac{1}{b_l}|\frac{x-m}{a_l}|^{b_l}}, x < m$$

$$f(x; a_l, a_r, b_l, b_r, m) = \frac{1}{A} e^{-\frac{1}{b_r}|\frac{x-m}{a_r}|^{b_r}}, x > m$$

with:

$$A = a_l b_l^{1/b_l} \Gamma(1 + 1/b_l) + a_r b_r^{1/b_r} \Gamma(1 + 1/b_r)$$

where l and r represent left and right tails, $a*$ are scale parameters, $b*$ control the tails (lower values represent fatter tails), and m is a location parameter.

Value

a vector containing the values for the probabilities.

`plaplace` *Returns CDF from the Laplace Distribution*

Description

The `plaplace` returns the Cumulative Distribution Function at point x for the Laplace distribution with parameters a and m.

Usage

```
plaplace(x, m = 0, a = 1)
```

Arguments

`x`	(numeric) - value in the range $(-\infty, \infty)$ to evaluate the density.
`m`	(numeric) - location parameter.
`a`	(numeric) - scale parameter. Must be in the range $(0, \infty)$.

Details

The Laplace distribution is a distribution controlled by two parameters, with formula:

$$f(x; a, m) = \frac{1}{2a} e^{-|\frac{x-m}{a}|}$$

where a is a scale parameter, and m is a location parameter.

Value

a vector containing the values for the probabilities.

ppower *Returns CDF from EP Distribution*

Description

The ppower returns the Cumulative Distribution Function at point x for the Exponential Power distribution with parameters a, b and m.

Usage

```
ppower(x, m = 0, a = 1, b = 2)
```

Arguments

x	(numeric) - value in the range $(-\infty, \infty)$ to evaluate the density.
m	(numeric) - location parameter. Must be in the range
a	(numeric) - scale parameter. Must be in the range $(0, \infty)$.
b	(numeric) - shape parameter. Must be in the range $(0, \infty)$. $(-\infty, \infty)$.

Details

The Exponential Power distribution (EP) is given by the function:

$$f(a, b) = \frac{1}{2a\Gamma(1 + 1/b)} e^{-|(x-m)/a|^b}, -\infty < x < \infty$$

. where b is a shape parameter, a is a scale parameter, m is a location parameter and Γ represents the gamma function.

Value

a vector containing the values for the probabilities.

psep *Returns CDF from the Skewed Exponential Power distribution*

Description

The psep returns the Cumulative Distribution Function at point x for the Skewed Exponential Power distribution with parameters a, b.

Usage

```
psep(x, m = 0, a = 2, b = 1, lambda = 0)
```

Arguments

x	• vector with values to evaluate CDF.
m	• the location parameter.
a	• the scale parameter.
b	• the shape parameter
lambda	• the skewness parameter.

Details

The SEP is a exponential power distribution controlled by four parameters, with formula:

$$f(x; m, b, a, \lambda) = 2\Phi(w)e^{-|z|^b/b}/(c)$$

where:

$$z = (x - m)/a$$

$$w = sign(z)|z|^{(b/2)}\lambda\sqrt{2/b}$$

$$c = 2ab^{(1/b)-1}\Gamma(1/b)$$

with Φ the cumulative normal distribution with mean zero and variance one. The CDF is calculated through numerical integration using the 'GSL' suite.

Value

a vector containing the values for the probabilities.

---

psubbo *Returns CDF from Subbotin Distribution*

Description

The psubbo returns the Cumulative Distribution Function (CDF) from the the Subbotin evaluated at a and return z, such that $P(X < a) = z$.

Usage

```
psubbo(x, m = 0, a = 1, b = 2)
```

Arguments

x	(numeric) - value in the range $(-\infty, \infty)$ to evaluate the density.
m	(numeric) - location parameter.
a	(numeric) - scale parameter. Must be in the range $(0, \infty)$.
b	(numeric) - shape parameter. Must be in the range $(0, \infty)$.

Details

The Subbotin cumulative distribution function is given by:

$$F(x; a, b, m) = 0.5 + 0.5\text{sign}(x - m)P(x, 1/b)$$

where P is the normalized incomplete gamma function:

$$P(x, 1/b) = 1 - \frac{1}{\Gamma(1/b)} \int_0^x t^{1/b-1} e^{-t}$$

and a is a scale parameter, b controls the tails (lower values represent fatter tails), and m is a location parameter.

Value

a vector containing the values for the probabilities.

`qalaplace` *Returns quantile from Asymmetric Laplace Distribution*

Description

The `qalaplace` returns the density at point x for the Asymmetric Laplace distribution with parameters $a*$ and m.

Usage

```
qalaplace(x, m = 0, al = 1, ar = 1)
```

Arguments

`x`	(numeric) - value in the range $(-\infty, \infty)$ to evaluate the density.
`m`	(numeric) - location parameter.
`al`, `ar`	(numeric) - scale parameters. Must be in the range $(0, \infty)$.

Details

The Asymmetric Laplace distribution is a distribution controlled by three parameters, with formula:

$$f(x; a_l, a_r, m) = \frac{1}{A} e^{-|\frac{x-m}{a_l}|}, x < m$$

$$f(x; a_l, a_r, m) = \frac{1}{A} e^{-|\frac{x-m}{a_r}|}, x > m$$

with:

$$A = a_l + a_r$$

where $a*$ are scale parameters, and m is a location parameter. It is basically derived from the Asymmetric Exponential Power distribution by setting $b_l = b_r = b$.

Value

a vector containing the values for the densities.

qasubbo *Returns CDF from the AEP Distribution*

Description

The qasubbo returns the density at point x for the AEP distribution with parameters $a*$, $b*$, m.

Usage

```
qasubbo(x, m = 0, al = 1, ar = 1, bl = 2, br = 2)
```

Arguments

x	(numeric) - value in the range $(-\infty, \infty)$ to evaluate the density.
m	(numeric) - location parameter.
al, ar	(numeric) - scale parameters. Must be in the range $(0, \infty)$.
bl, br	(numeric) - shape parameters. Must be in the range $(0, \infty)$.

Details

The AEP is a exponential power distribution controlled by five parameters, with formula:

$$f(x; a_l, a_r, b_l, b_r, m) = \frac{1}{A} e^{-\frac{1}{b_l}|\frac{x-m}{a_l}|^{b_l}}, x < m$$

$$f(x; a_l, a_r, b_l, b_r, m) = \frac{1}{A} e^{-\frac{1}{b_r}|\frac{x-m}{a_r}|^{b_r}}, x > m$$

with:

$$A = a_l b_l^{1/b_l} \Gamma(1 + 1/b_l) + a_r b_r^{1/b_r} \Gamma(1 + 1/b_r)$$

where l and r represent left and right tails, $a*$ are scale parameters, $b*$ control the tails (lower values represent fatter tails), and m is a location parameter.

Value

a vector containing the values for the densities.

qlaplace *Returns quantile from Laplace Distribution*

Description

The qlaplace returns the density at point x for the Laplace distribution with parameters a and m.

Usage

```
qlaplace(x, m = 0, a = 1)
```

Arguments

x	(numeric) - value in the range $(-\infty, \infty)$ to evaluate the density.
m	(numeric) - location parameter.
a	(numeric) - scale parameter. Must be in the range $(0, \infty)$.

Details

The Laplace distribution is a distribution controlled by two parameters, with formula:

$$f(x; a, m) = \frac{1}{2a} e^{-|\frac{x-m}{a}|}$$

where a is a scale parameter, and m is a location parameter.

Value

a vector containing the values for the densities.

qpower *Returns quantile from EP Distribution*

Description

The qpower returns the density at point x for the Exponential Power distribution with parameters a, b and m.

Usage

```
qpower(x, m = 0, a = 1, b = 2)
```

Arguments

x	(numeric) - value in the range $(-\infty, \infty)$ to evaluate the density.
m	(numeric) - location parameter. Must be in the range
a	(numeric) - scale parameter. Must be in the range $(0, \infty)$.
b	(numeric) - shape parameter. Must be in the range $(0, \infty)$. $(-\infty, \infty)$.

Details

The Exponential Power distribution (EP) is given by the function:

$$f(a,b) = \frac{1}{2a\Gamma(1+1/b)} e^{-|(x-m)/a|^b}, -\infty < x < \infty$$

. where b is a shape parameter, a is a scale parameter, m is a location parameter and Γ represents the gamma function.

Value

a vector containing the values for the densities.

`qsep` *Returns quantile from the Skewed Exponential Power distribution*

Description

The `qsep` returns the Cumulative Distribution Function at point x for the Skewed Exponential Power distribution with parameters a, b.

Usage

```
qsep(
  x,
  m = 0,
  a = 2,
  b = 1,
  lambda = 0,
  method = 0L,
  step_size = 1e-04,
  tol = 1e-10,
  max_iter = 100L,
  verb = 0L
)
```

Arguments

`x`	(numeric) - vector with values to evaluate CDF.
`m`	(numeric) - the location parameter.
`a`	(numeric) - the scale parameter.
`b`	(numeric) - the shape parameter
`lambda`	(numeric) - the skewness parameter.
`method`	(numeric) - If 0, uses the Newton-Raphson procedure for optimization. If 1, uses Steffensen.

step_size	(numeric) - the size of the step in the numerical optimization (gradient descent). Default is 1e-4.
tol	(numeric) - error tolerance (default is 1e-10).
max_iter	(numeric) - maximum number of iterations for the optimization procedure (default is 100).
verb	(numeric) - verbosity level of the process (default 0).

Details

The SEP is a exponential power distribution controlled by four parameters, with formula:

$$f(x; m, b, a, \lambda) = 2\Phi(w)e^{-|z|^b/b}/(c)$$

where:

$$z = (x - m)/a$$

$$w = sign(z)|z|^{(b/2)}\lambda\sqrt{2/b}$$

$$c = 2ab^{(1/b)-1}\Gamma(1/b)$$

with Φ the cumulative normal distribution with mean zero and variance one. The CDF is calculated through numerical integration using the GSL suite and the quantile is solved by inversion using a root-finding algorithm (Newton-Raphson by default).

Value

a vector containing the values for the densities.

---

qsubbo *Returns CDF from Subbotin Distribution*

Description

The qsubbo returns the Cumulative Distribution Function (CDF) from the the Subbotin evaluated at a and return z, such that $P(X < a) = z$.

Usage

```
qsubbo(x, m = 0, a = 1, b = 2)
```

Arguments

x	(numeric) - value in the range $(-\infty, \infty)$ to evaluate the density.
m	(numeric) - location parameter.
a	(numeric) - scale parameter. Must be in the range $(0, \infty)$.
b	(numeric) - shape parameter. Must be in the range $(0, \infty)$.

Details

The Subbotin cumulative distribution function is given by:

$$F(x; a, b, m) = 0.5 + 0.5\text{sign}(x - m)P(x, 1/b)$$

where P is the normalized incomplete gamma function:

$$P(x, 1/b) = 1 - \frac{1}{\Gamma(1/b)} \int_0^x t^{1/b-1} e^{-t}$$

and a is a scale parameter, b controls the tails (lower values represent fatter tails), and m is a location parameter.

Value

a vector containing the values for the densities.

---

ralaplace *Generates an Asymmetric Laplace-distributed sample*

Description

Returns a sample from an Asymmetric Laplace distribution.

Usage

```
ralaplace(n, m = 0, al = 1, ar = 1)
```

Arguments

n	(int) - the size of the sample.
m	(numeric) - the location parameter.
al, ar	(numeric) - left and right scale parameters, respectively.

Details

The Asymmetric Laplace distribution is given by the two-sided exponential distribution given by the function:

$$f(x; a_l, a_r, m) = \frac{1}{A} e^{-|\frac{x-m}{a_l}|}, x < m$$

$$f(x; a_l, a_r, m) = \frac{1}{A} e^{-|\frac{x-m}{a_r}|}, x > m$$

with:

$$A = a_l + a_r$$

The random sampling is done by inverse transform sampling.

Value

a numeric vector containing a random sample.

Examples

```
sample_gamma <- ralaplace(1000)
```

`rasubbo` *Produces a random sample from a Asymmetric Power Exponential distribution*

Description

Generate pseudo random-number from an asymmetric power exponential distribution using the Tadikamalla method. This version improves on Bottazzi (2004) by making the mass of each distribution to depend on the ratio between the al and the ar parameters.

Usage

```
rasubbo(n, m = 0, al = 1, ar = 1, bl = 2, br = 2)
```

Arguments

`n`	(int) - size of the sample.
`m`	(numeric) - location parameter.
`al`, `ar`	(numeric) - scale parameters.
`bl`, `br`	(numeric) - shape parameters.

Details

The AEP distribution is expressed by the function:

$$f(x; a_l, a_r, b_l, b_r, m) = \frac{1}{A} e^{-\frac{1}{b_l}|\frac{x-m}{a_l}|^{b_l}}, x < m$$

$$f(x; a_l, a_r, b_l, b_r, m) = \frac{1}{A} e^{-\frac{1}{b_r}|\frac{x-m}{a_r}|^{b_r}}, x > m$$

with:

$$A = a_l b_l^{1/b_l} \Gamma(1 + 1/b_l) + a_r b_r^{1/b_r} \Gamma(1 + 1/b_r)$$

where m is a location parameter, $b*$ are shape parameters, $a*$ are scale parameters and Γ represents the gamma function. By a suitably transformation, it is possible to use the EP distribution with the Tadikamalla method to sample from this distribution. We basically take the absolute values of the numbers sampled from the `rpower` function, which is equivalent from sampling from a half Exponential Power distribution. These values are then weighted by a constant expressed in the parameters. More details are available on the package vignette and on the function `rpower`.

Value

a numeric vector containing a random sample.

Examples

```
sample_gamma <- rasubbo(1000, 0, 0.5, 0.5,  1, 1)
```

`rasubbo_orig` *Produces a random sample from a Asymmetric Power Exponential distribution*

Description

Generate pseudo random-number from an asymmetric power exponential distribution using the Tadikamalla method. This codes is the original version of Bottazzi (2004)

Usage

```
rasubbo_orig(n, m = 0, al = 1, ar = 1, bl = 2, br = 2)
```

Arguments

`n`	(int) - size of the sample.
`m`	(numeric) - location parameter.
`al`, `ar`	(numeric) - scale parameters.
`bl`, `br`	(numeric) - shape parameters.

Details

The AEP distribution is expressed by the function:

$$f(x; a_l, a_r, b_l, b_r, m) = \frac{1}{A} e^{-\frac{1}{b_l}|\frac{x-m}{a_l}|^{b_l}}, x < m$$

$$f(x; a_l, a_r, b_l, b_r, m) = \frac{1}{A} e^{-\frac{1}{b_r}|\frac{x-m}{a_r}|^{b_r}}, x > m$$

with:

$$A = a_l b_l^{1/b_l} \Gamma(1 + 1/b_l) + a_r b_r^{1/b_r} \Gamma(1 + 1/b_r)$$

where m is a location parameter, $b*$ are shape parameters, $a*$ are scale parameters and Γ represents the gamma function. By a suitably transformation, it is possible to use the EP distribution with the Tadikamalla method to sample from this distribution. We basically take the absolute values of the numbers sampled from the `rpower` function, which is equivalent from sampling from a half Exponential Power distribution. These values are then weighted by a constant expressed in the parameters. More details are available on the package vignette and on the function `rpower`.

Value

a numeric vector containing a random sample.

Examples

```
sample_gamma <- rasubbo(1000, 0, 0.5, 0.5,  1, 1)
```

`rgamma_c` *Generates a Gamma-distributed sample*

Description

Returns a sample from a gamma-distributed random variable. The function uses the Marsaglia-Tsang fast gamma method used to generate the random samples. See more details below. The name was chosen so it doesn't clash with R's native method.

Usage

```
rgamma_c(n, b = 2, a = 1/2)
```

Arguments

`n`	(int)
`b`	(numeric) - shape parameter. Must be in the range $(0, \infty)$.
`a`	(numeric) - scale parameter. Must be in the range $(0, \infty)$.

Details

The gamma distribution is given by the function:

$$f(x) = \frac{1}{\Gamma(b)a^b} x^{b-1} e^{-x/a}, x > 0$$

where b is a shape parameter and a is a scale parameter. The RNG is given by Marsaglia and Tsang, "A Simple Method for generating gamma variables", ACM Transactions on Mathematical Software, Vol 26, No 3 (2000), p363-372. Available at doi:10.1145/358407.358414. The code is based on the original 'GSL' version, adapted to use 'R' version of RNGs. All credits to the original authors. Implemented by J.D.Lamb@btinternet.com, minor modifications for 'GSL' by Brian Gough. Adapted to 'R' by Elias Haddad.

Value

a numeric vector containing a random sample with above parameters.

Examples

```
sample_gamma <- rgamma_c(1000, 1, 1)
```

rlaplace *Generates a Laplace-distributed sample*

Description

Returns a sample from a Laplace-distributed random variable.

Usage

```
rlaplace(n, m = 0, a = 1)
```

Arguments

n	(int) - the size of the sample.
m	(numeric) - the location parameter.
a	(numeric) - the scale parameter.

Details

The Laplace distribution is given by the two-sided exponential distribution given by the function:

$$f(x; a, m) = \frac{1}{2a} e^{-\left|\frac{x-m}{a}\right|}$$

The random sampling is done by inverse transform sampling.

Value

a numeric vector containing a random sample with above parameters.

Examples

```
sample_gamma <- rlaplace(1000, 0, 1)
```

rpower *Generates a random sample from a Exponential Power distribution*

Description

Returns a sample from a gamma-distributed random variable.

Usage

```
rpower(n, m = 0, a = 1, b = 2)
```

Arguments

n	(int) - size of the sample.
m	(numeric) - the location parameter.
a	(numeric) - scale parameter.
b	(numeric) - shape parameter.

Details

The exponential power distribution (EP) is given by the function:

$$f(a,b) = \frac{1}{2a\Gamma(1+1/b)} e^{-|x/a|^b}, -\infty < x < \infty$$

. where b is a shape parameter, a is a scale parameter and Γ represents the gamma function. While not done here, the distribution can be adapted to have non-zero location parameter. The Exponential Power distribution is related to the gamma distribution by the equation:

$$E = a * G(1/b)^{1/b}$$

where E and G are respectively EP and gamma random variables. This property is used for cases where $b < 1$ and $b > 4$. For $1 \leq b \leq 4$ rejection methods based on the Laplace and normal distributions are used, which should be faster. Technical details about this algorithm are available on: P. R. Tadikamalla, "Random Sampling from the Exponential Power Distribution", Journal of the American Statistical Association, September 1980, Volume 75, Number 371, pages 683-686. The code is based on the original 'GSL' version, adapted to use 'R' version of RNGs by Elias Haddad. All credits to the original authors.

Value

a numeric vector containing a random sample with above parameters.

Examples

```
sample_gamma <- rpower(1000)
```

---

rsubbo *Produces a random sample from a Subbotin distribution*

Description

Generate pseudo random-number from a Subbotin distribution using the Tadikamalla method.

Usage

```
rsubbo(n, m = 0, a = 1, b = 2)
```

Arguments

n	(int) - the size of the sample.
m	(numeric) - the location parameter.
a	(numeric) - the scale parameter.
b	(numeric) - the shape parameter.

Details

The Subbotin distribution is given by the function:

$$f(x; a, b, m) = \frac{1}{A} e^{-\frac{1}{b}\left|\frac{x-m}{a}\right|^b}$$

where m is a location parameter, b is a shape parameter, a is a scale parameter and Γ represents the gamma function. Since the Subbotin distribution is basically the exponential distribution with scale parameter $a = ab^{1/b}$ and $m = 0$, we use the same method of the exponential power RNG and add the location parameter. Details can be found on the documentation of the `rpower` function.

Value

a numeric vector containing a random sample.

Examples

```
sample_gamma <- rsubbo(1000, 1, 1)
```

---

sepfit *Fit a Skewed Exponential Power density via maximum likelihood*

Description

`sepfit` returns the parameters, standard errors. negative log-likelihood and covariance matrix of the skewed power exponential for a sample. The process performs a global minimization over the negative log-likelihood function. See details below.

Usage

```
sepfit(
  data,
  verb = 0L,
  par = as.numeric(c(0, 1, 2, 0)),
  g_opt_par = as.numeric(c(0.1, 0.01, 100, 0.001, 1e-05, 2))
)
```

Arguments

`data` (NumericVector) - the sample used to fit the distribution.

`verb` (int) - the level of verbosity. Select one of:

- 0 just the final result
- 1 headings and summary table
- 2 intermediate steps results
- 3 intermediate steps internals
- 4+ details of optim. routine

`par` NumericVector - vector containing the initial guess for parameters m (location), a (scale), b (shape), lambda (skewness), respectively. Default values of are c(0, 1, 2, 0), i.e. a normal distribution.

`g_opt_par` NumericVector - vector containing the global optimization parameters. The optimization parameters are:

- step - (num) initial step size of the searching algorithm.
- tol - (num) line search tolerance.
- iter - (int) maximum number of iterations.
- eps - (num) gradient tolerance. The stopping criteria is $||\text{gradient}|| < \text{eps}$.
- msize - (num) simplex max size. stopping criteria given by $||\text{max edge}|| < \text{msize}$
- algo - (int) algorithm. the optimization method used:
  - 0 Fletcher-Reeves
  - 1 Polak-Ribiere
  - 2 Broyden-Fletcher-Goldfarb-Shanno
  - 3 Steepest descent
  - 4 Nelder-Mead simplex
  - 5 Broyden-Fletcher-Goldfarb-Shanno ver.2

Details for each algorithm are available on the 'GSL' Manual. Default values are c(.1, 1e-2, 100, 1e-3, 1e-5, 2).

Details

The SEP is a exponential power distribution controlled by four parameters, with formula:

$$f(x; m, b, a, \lambda) = 2\Phi(w)e^{-|z|^b/b}/(c)$$

where:

$$z = (x - m)/a$$

$$w = sign(z)|z|^{(b/2)}\lambda\sqrt{2/b}$$

$$c = 2ab^{(1/b)-1}\Gamma(1/b)$$

with Φ the cumulative normal distribution with mean zero and variance one. Details on the method are available on the package vignette.

Value

a list containing the following items:

- "dt" - dataset containing parameters estimations and standard deviations.
- "log-likelihood" - negative log-likelihood value.
- "matrix" - the covariance matrix for the parameters.

Examples

```
sample_subbo <- rpower(1000, 1, 2)
sepfit(sample_subbo)
```

---

subboafish *Returns the Fisher Information matrix and its inverse for the AEP*

Description

Returns the Fisher Information matrix and its inverse for the Asymmetric Power Exponential distribution for the given parameters.

Usage

```
subboafish(size = 1L, bl = 2, br = 2, m = 0, al = 1, ar = 1, O_munknown = 0L)
```

Arguments

size	(numeric) - number of observations (Default: 01)
bl	(numeric) - set the left exponent (Default: 2.0)
br	(numeric) - set the right exponent (Default: 2.0)
m	(numeric) - the location parameter (Default: 0.0)
al	(numeric) - the left scale parameter (Default: 1.0)
ar	(numeric) - the right scale parameter (Default: 1.0)
O_munknown	(numeric) - if true assumes m is known

Value

a list containing three elements:

- std_error - the standard error for the parameters
- infmatrix - the Fisher Information Matrix
- inv_infmatrix - the Inverse Fisher Information Matrix

subboafit *Fit an Asymmetric Power Exponential density via maximum likelihood*

Description

subboafit returns the parameters, standard errors. negative log-likelihood and covariance matrix of the asymmetric power exponential for a sample. The process can execute two steps, depending on the level of accuracy required. See details below.

Usage

```
subboafit(
  data,
  verb = 0L,
  method = 2L,
  interv_step = 10L,
  provided_m_ = NULL,
  par = as.numeric(c(2, 2, 1, 1, 0)),
  g_opt_par = as.numeric(c(0.1, 0.01, 100, 0.001, 1e-05, 2)),
  itv_opt_par = as.numeric(c(0.01, 0.001, 200, 0.001, 1e-05, 5))
)
```

Arguments

data	(NumericVector) - the sample used to fit the distribution.
verb	(int) - the level of verbosity. Select one of: • 0 just the final result (default) • 1 headings and summary table • 2 intermediate steps results • 3 intermediate steps internals • 4+ details of optim. routine
method	int - the steps that should be used to estimate the parameters. • 0 no optimization perform - just return the log-likelihood from initial guess. • 1 global optimization not considering lack of smoothness in m • 2 interval optimization taking non-smoothness in m into consideration (default, only occurs if provided_m_ is null)
interv_step	int - the number of intervals to be explored after the last minimum was found in the interval optimization. Default is 10.
provided_m_	NumericVector - if NULL (default), the m parameter is estimated by the routine. If numeric, the estimation fixes m to the given value.
par	NumericVector - vector containing the initial guess for parameters bl, br, al, ar and m, respectively. Default values are c(2, 2, 1, 1, 0).
g_opt_par	NumericVector - vector containing the global optimization parameters. The optimization parameters are:

- step - (num) initial step size of the searching algorithm.
- tol - (num) line search tolerance.
- iter - (int) maximum number of iterations.
- eps - (num) gradient tolerance. The stopping criteria is $||\text{gradient}|| < \text{eps}$.
- msize - (num) simplex max size. stopping criteria given by $||\text{max edge}|| <$ msize
- algo - (int) algorithm. the optimization method used:
  - 0 Fletcher-Reeves
  - 1 Polak-Ribiere
  - 2 Broyden-Fletcher-Goldfarb-Shanno
  - 3 Steepest descent
  - 4 Nelder-Mead simplex
  - 5 Broyden-Fletcher-Goldfarb-Shanno ver.2

Details for each algorithm are available on the 'GSL' Manual. Default values are c(.1, 1e-2, 100, 1e-3, 1e-5, 2).

`itv_opt_par` NumericVector - interval optimization parameters. Fields are the same as the ones for the global optimization. Default values are c(.01, 1e-3, 200, 1e-3, 1e-5, 5).

Details

The AEP is a exponential power distribution controlled by five parameters, with formula:

$$f(x; a_l, a_r, b_l, b_r, m) = \frac{1}{A} e^{-\frac{1}{b_l}|\frac{x-m}{a_l}|^{b_l}}, x < m$$

$$f(x; a_l, a_r, b_l, b_r, m) = \frac{1}{A} e^{-\frac{1}{b_r}|\frac{x-m}{a_r}|^{b_r}}, x > m$$

with:

$$A = a_l b_l^{1/b_l} \Gamma(1 + 1/b_l) + a_r b_r^{1/b_r} \Gamma(1 + 1/b_r)$$

where l and r represent left and right tails, $a*$ are scale parameters, $b*$ control the tails (lower values represent fatter tails), and m is a location parameter. Due to its lack of symmetry, and differently from the Subbotin, there is no simple equations available to use the method of moments, so we start directly by minimizing the negative log-likelihood. This global optimization is executed without restricting any parameters. If required (default), after the global optimization is finished, the method proceeds to iterate over the intervals between several two observations, iterating the same algorithm of the global optimization. The last method happens because of the lack of smoothness on the m parameter, and intervals must be used since the likelihood function doesn't have a derivative whenever m equals a sample observation. Due to the cost, these iterations are capped at most *interv_step* (default 10) from the last minimum observed. Details on the method are available on the package vignette.

Value

a list containing the following items:

- "dt" - dataset containing parameters estimations and standard deviations.
- "log-likelihood" - negative log-likelihood value.
- "matrix" - the covariance matrix for the parameters.

Examples

```
sample_subbo <- rpower(1000, 1, 2)
subboafit(sample_subbo)
```

---

subbofish *Return the Fisher Information Matrix for the Subbotin Distribution*

Description

Calculate the standard errors, the correlation, the Fisher Information matrix and its inverse for a power exponential density with given parameters

Usage

```
subbofish(size = 1L, b = 2, m = 0, a = 1, O_munknown = 0L)
```

Arguments

size	numeric - number of observations (Default: 01)
b	numeric - the exponent b (Default: 2.0)
m	numeric - the location parameter (Default: 0.0)
a	numeric - the scale parameter (Default: 1.0)
O_munknown	numeric - if true assumes m known

Value

a list containing four elements:

- std_error - the standard error for the parameters
- cor_ab - the correlation between parameters a and b
- infmatrix - the Fisher Information Matrix
- inv_infmatrix - the Inverse Fisher Information Matrix

subbofit *Fit a power exponential density via maximum likelihood*

Description

subbofit returns the parameters, standard errors. negative log-likelihood and covariance matrix of the Subbotin Distribution for a sample. The process can execute three steps, depending on the level of accuracy required. See details below.

Usage

```
subbofit(
  data,
  verb = 0L,
  method = 3L,
  interv_step = 10L,
  provided_m_ = NULL,
  par = as.numeric(c(2, 1, 0)),
  g_opt_par = as.numeric(c(0.1, 0.01, 100, 0.001, 1e-05, 3)),
  itv_opt_par = as.numeric(c(0.01, 0.001, 200, 0.001, 1e-05, 5))
)
```

Arguments

data	(NumericVector) - the sample used to fit the distribution.
verb	(int) - the level of verbosity. Select one of: • 0 just the final result (default) • 1 headings and summary table • 2 intermediate steps results • 3 intermediate steps internals • 4+ details of optim. routine
method	int - the steps that should be used to estimate the parameters. • 0 no optimization perform - just return the log-likelihood from initial guess. • 1 initial estimation based on method of moments • 2 global optimization not considering lack of smoothness in m • 3 interval optimization taking non-smoothness in m into consideration (default, only occurs if provided_m_ is null)
interv_step	int - the number of intervals to be explored after the last minimum was found in the interval optimization. Default is 10.
provided_m_	NumericVector - if NULL (default), the m parameter is estimated by the routine. If numeric, the estimation fixes m to the given value.
par	NumericVector - vector containing the initial guess for parameters b, a and m, respectively. Default values are c(2, 1, 0).

`g_opt_par` NumericVector - vector containing the global optimization parameters. The optimization parameters are:

- step - (num) initial step size of the searching algorithm.
- tol - (num) line search tolerance.
- iter - (int) maximum number of iterations.
- eps - (num) gradient tolerance. The stopping criteria is ||gradient|| $<$ eps.
- msize - (num) simplex max size. stopping criteria given by ||max edge|| $<$ msize
- algo - (int) algorithm. the optimization method used:
  - 0 Fletcher-Reeves
  - 1 Polak-Ribiere
  - 2 Broyden-Fletcher-Goldfarb-Shanno
  - 3 Steepest descent
  - 4 Nelder-Mead simplex
  - 5 Broyden-Fletcher-Goldfarb-Shanno ver.2

Details for each algorithm are available on the 'GSL' Manual. Default values are c(.1, 1e-2, 100, 1e-3, 1e-5, 3,0).

`itv_opt_par` NumericVector - interval optimization parameters. Fields are the same as the ones for the global optimization. Default values are c(.01, 1e-3, 200, 1e-3, 1e-5, 5, 0).

Details

The Subbotin distribution is a exponential power distribution controlled by three parameters, with formula:

$$f(x; a, b, m) = \frac{1}{A} e^{-\frac{1}{b}|\frac{x-m}{a}|^b}$$

with:

$$A = 2ab^{1/b}\Gamma(1 + 1/b)$$

where a is a scale parameter, b controls the tails (lower values represent fatter tails), and m is a location parameter. Due to its symmetry, the equations are simple enough to be estimated by the method of moments, which produce rough estimations that should be used only for first explorations. The maximum likelihood global estimation improves on this initial guess by using a optimization routine, defaulting to the Broyden-Fletcher-Goldfarb-Shanno method. However, due to the lack of smoothness of this function on the m parameter (derivatives are zero whenever m equals a sample observation), an exhaustive search must be done by redoing the previous step in all intervals between two observations. For a sample of n observations, this would lead to $n - 1$ optimization problems. Given the computational cost of such procedure, an interval search is used, where the optimization is repeated in the intervals at most the value of the *interv_step* from the last minimum found. Details on the method are available on the package vignette.

Value

a list containing the following items:

- "dt" - dataset containing parameters estimations and standard deviations.
- "log-likelihood" - negative log-likelihood value.
- "matrix" - the covariance matrix for the parameters.

Examples

```
sample_subbo <- rpower(1000, 1, 2)
subbofit(sample_subbo)
```

`subbolafit` *Fit a (Less) Asymmetric Power Exponential density via maximum likelihood*

Description

`subbolafit` returns the parameters, standard errors. negative log-likelihood and covariance matrix of the (less) asymmetric power exponential for a sample. The main difference from `subboafit` is that $a_l = a_r = a$. The process can execute two steps, depending on the level of accuracy required. See details below.

Usage

```
subbolafit(
  data,
  verb = 0L,
  method = 2L,
  interv_step = 10L,
  provided_m_ = NULL,
  par = as.numeric(c(2, 2, 1, 0)),
  g_opt_par = as.numeric(c(0.1, 0.01, 100, 0.001, 1e-05, 2)),
  itv_opt_par = as.numeric(c(0.01, 0.001, 200, 0.001, 1e-05, 2))
)
```

Arguments

`data`	(NumericVector) - the sample used to fit the distribution.
`verb`	(int) - the level of verbosity. Select one of: • 0 just the final result (default) • 1 headings and summary table • 2 intermediate steps results • 3 intermediate steps internals • 4+ details of optim. routine
`method`	int - the steps that should be used to estimate the parameters. • 0 no optimization perform - just return the log-likelihood from initial guess. • 1 global optimization not considering lack of smoothness in m • 2 interval optimization taking non-smoothness in m into consideration (default, only occurs if provided_m_ is null)
`interv_step`	int - the number of intervals to be explored after the last minimum was found in the interval optimization. Default is 10.

`provided_m_` NumericVector - if NULL (default), the m parameter is estimated by the routine. If numeric, the estimation fixes m to the given value.

`par` NumericVector - vector containing the initial guess for parameters bl, br, a and m, respectively. Default values of are c(2, 2, 1, 0).

`g_opt_par` NumericVector - vector containing the global optimization parameters. The optimization parameters are:

- step - (num) initial step size of the searching algorithm.
- tol - (num) line search tolerance.
- iter - (int) maximum number of iterations.
- eps - (num) gradient tolerance. The stopping criteria is $||\text{gradient}|| < \text{eps}$.
- msize - (num) simplex max size. stopping criteria given by $||\text{max edge}|| < \text{msize}$
- algo - (int) algorithm. the optimization method used:
  - 0 Fletcher-Reeves
  - 1 Polak-Ribiere
  - 2 Broyden-Fletcher-Goldfarb-Shanno
  - 3 Steepest descent
  - 4 Nelder-Mead simplex
  - 5 Broyden-Fletcher-Goldfarb-Shanno ver.2

Details for each algorithm are available on the 'GSL' Manual. Default values are c(.1, 1e-2, 100, 1e-3, 1e-5, 2).

`itv_opt_par` NumericVector - interval optimization parameters. Fields are the same as the ones for the global optimization. Default values are c(.01, 1e-3, 200, 1e-3, 1e-5, 2).

Details

The LAPE is a exponential power distribution controlled by four parameters, with formula:

$$f(x; a, b_l, b_r, m) = \frac{1}{A} e^{-\frac{1}{b_l}|\frac{x-m}{a}|^{b_l}}, x < m$$

$$f(x; a, b_l, b_r, m) = \frac{1}{A} e^{-\frac{1}{b_r}|\frac{x-m}{a}|^{b_r}}, x > m$$

with:

$$A = a b_l^{1/b_l} \Gamma(1 + 1/b_l) + a b_r^{1/b_r} \Gamma(1 + 1/b_r)$$

where l and r represent left and right tails, a is a scale parameter, $b*$ control the tails (lower values represent fatter tails), and m is a location parameter. Due to its lack of symmetry, and differently from the Subbotin, there is no simple equations available to use the method of moments, so we start directly by minimizing the negative log-likelihood. This global optimization is executed without restricting any parameters. If required (default), after the global optimization is finished, the method proceeds to iterate over the intervals between several two observations, iterating the same algorithm of the global optimization. The last method happens because of the lack of smoothness on the m parameter, and intervals must be used since the likelihood function doesn't have a derivative whenever m equals a sample observation. Due to the cost, these iterations are capped at most *interv_step* (default 10) from the last minimum observed. Details on the method are available on the package vignette.

Value

a list containing the following items:

- "dt" - dataset containing parameters estimations and standard deviations.
- "log-likelihood" - negative log-likelihood value.

Examples

```
sample_subbo <- rpower(1000, 1, 2)
subbolafit(sample_subbo)
```

Index

alaplafit, 3

dalaplace, 4
dasubbo, 5
dlaplace, 6
dpower, 6
dsep, 7
dsubbo, 8

laplafit, 8

palaplace, 9
pasubbo, 10
plaplace, 11
ppower, 12
psep, 12
psubbo, 13

qalaplace, 14
qasubbo, 15
qlaplace, 16
qpower, 16
qsep, 17
qsubbo, 18

ralaplace, 19
rasubbo, 20
rasubbo_orig, 21
rgamma_c, 22
rlaplace, 23
rpower, 23
rsubbo, 24

sepfit, 25
subboafish, 27
subboafit, 28
subbofish, 30
subbofit, 31
subbolafit, 33