Package ‘BMT’

April 17, 2025

Type Package

Title The BMT Distribution

Version 0.1.3

Date 2025-04-17

Encoding UTF-8

Description Density, distribution, quantile function, random number generation for the BMT (Bezier-Montenegro-Torres) distribution. Torres-Jimenez C.J. and Montenegro-Diaz A.M. (2017) <doi:10.48550/arXiv.1709.05534>. Moments, descriptive measures and parameter conversion for different parameterizations of the BMT distribution. Fit of the BMT distribution to non-censored data by maximum likelihood, moment matching, quantile matching, maximum goodness-of-fit, also known as minimum distance, maximum product of spacing, also called maximum spacing, and minimum quantile distance, which can also be called maximum quantile goodness-of-fit. Fit of univariate distributions for non-censored data using maximum product of spacing estimation and minimum quantile distance estimation is also included.

License GPL (>= 2)

Depends partitions, fitdistrplus

RoxygenNote 7.3.2

NeedsCompilation no

Author Camilo Jose Torres-Jimenez [aut, cre],
Alvaro Mauricio Montenegro-Diaz [ths]

Maintainer Camilo Jose Torres-Jimenez <`cjtorresj@gmail.com`>

Repository CRAN

Date/Publication 2025-04-17 17:20:01 UTC

Contents

BMT 2
BMT.Phi 5
BMT.Psi 7
BMTcentral 9
BMTchangepars 11

BMTdispersion 13
BMTfit 15
BMTfit.mge 19
BMTfit.mle 22
BMTfit.mme 24
BMTfit.mpse 27
BMTfit.mqde 30
BMTfit.qme 33
BMTkurtosis 36
BMTmoments 38
BMTskewness 41
mpsedist 43
mqdedist 46
score.math.booklet10.PISA2012 50

Index **51**

`BMT` *The BMT Distribution.*

Description

Density, distribution, quantile function, random number generation for the BMT distribution, with `p3` and `p4` tails weights (κ_l and κ_r) or asymmetry-steepness parameters (ζ and ξ) and `p1` and `p2` domain (minimum and maximum) or location-scale (mean and standard deviation) parameters.

Usage

```
dBMT(
  x,
  p3,
  p4,
  type.p.3.4 = "t w",
  p1 = 0,
  p2 = 1,
  type.p.1.2 = "c-d",
  log = FALSE
)

pBMT(
  q,
  p3,
  p4,
  type.p.3.4 = "t w",
  p1 = 0,
  p2 = 1,
  type.p.1.2 = "c-d",
  lower.tail = TRUE,
```

```
  log.p = FALSE
)

qBMT(
  p,
  p3,
  p4,
  type.p.3.4 = "t w",
  p1 = 0,
  p2 = 1,
  type.p.1.2 = "c-d",
  lower.tail = TRUE,
  log.p = FALSE
)

rBMT(n, p3, p4, type.p.3.4 = "t w", p1 = 0, p2 = 1, type.p.1.2 = "c-d")
```

Arguments

`x, q`	vector of quantiles.
`p3, p4`	tails weights (κ_l and κ_r) or asymmetry-steepness (ζ and ξ) parameters of the BMT distribution.
`type.p.3.4`	type of parametrization associated to p3 and p4. "t w" means tails weights parametrization (default) and "a-s" means asymmetry-steepness parametrization.
`p1, p2`	domain (minimum and maximum) or location-scale (mean and standard deviation) parameters of the BMT distribution.
`type.p.1.2`	type of parametrization associated to p1 and p2. "c-d" means domain parametrization (default) and "l-s" means location-scale parametrization.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$, otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required

Details

The BMT distribution with tails weights and domain parametrization (`type.p.3.4 = "t w"` and `type.p.1.2 = "c-d"`) has quantile function

$$(d-c)[3t_p(1-t_p)^2\kappa_l - 3t_p^2(1-t_p)\kappa_r + t_p^2(3-2t_p)] + c$$

where $0 \leq p \leq 1$, $t_p = 1/2 - \cos([\arccos(2p-1) - 2\pi]/3)$, and $0 < \kappa_l < 1$ and $0 < \kappa_r < 1$ are, respectively, related to left and right tail weights or curvatures.

The BMT coefficient of asymmetry $-1 < \zeta < 1$ is

$$\kappa_r - \kappa_l$$

The BMT coefficient of steepness $0 < \xi < 1$ is

$$(\kappa_r + \kappa_l - |\kappa_r - \kappa_l|)/(2(1 - |\kappa_r - \kappa_l|))$$

for $|\kappa_r - \kappa_l| < 1$.

Value

`dBMT` gives the density, `pBMT` the distribution function, `qBMT` the quantile function, and `rBMT` generates random deviates.

The length of the result is determined by `n` for `rBMT`, and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than `n` are recycled to the length of the result. Only the first elements of the logical arguments are used.

If `type.p.3.4 == "t w"`, `p3 < 0` and `p3 > 1` are errors and return `NaN`.

If `type.p.3.4 == "a-s"`, `p3 < -1` and `p3 > 1` are errors and return `NaN`.

`p4 < 0` and `p4 > 1` are errors and return `NaN`.

If `type.p.1.2 == "c-d"`, `p1 >= p2` is an error and returns `NaN`.

If `type.p.1.2 == "l-s"`, `p2 <= 0` is an error and returns `NaN`.

Author(s)

Camilo Jose Torres-Jimenez [aut,cre] <`cjtorresj@unal.edu.co`> and Alvaro Mauricio Montenegro Diaz [ths]

References

Torres-Jimenez, C. J. and Montenegro-Diaz, A. M. (2017, September), *An alternative to continuous univariate distributions supported on a bounded interval: The BMT distribution*. ArXiv e-prints. `https://arxiv.org/abs/1709.05534`.

Torres-Jimenez, C. J. (2017, September), *Comparison of estimation methods for the BMT distribution*. ArXiv e-prints.

Torres-Jimenez, C. J. (2018), *The BMT Item Response Theory model: A new skewed distribution family with bounded domain and an IRT model based on it*, PhD thesis, Doctorado en ciencias - Estadistica, Universidad Nacional de Colombia, Sede Bogota.

See Also

`BMTcentral`, `BMTdispersion`, `BMTskewness`, `BMTkurtosis`, `BMTmoments` for descriptive measures or moments. `BMTchangepars` for parameter conversion between different parametrizations.

Examples

```
# BMT on [0,1] with left tail weight equal to 0.25 and
# right tail weight equal to 0.75
z <- seq(0, 1, length.out = 100)
F1 <- pBMT(z, 0.25, 0.75, "t w")
Q1 <- qBMT(F1, 0.25, 0.75, "t w")
```

```
max(abs(z - Q1))
f1 <- dBMT(z, 0.25, 0.75, "t w")
r1 <- rBMT(100, 0.25, 0.75, "t w")
layout(matrix(c(1,2,1,3), 2, 2))
hist(r1, freq = FALSE, xlim = c(0,1))
lines(z, f1)
plot(z, F1, type="l")
plot(F1, Q1, type="l")

# BMT on [0,1] with asymmetry coefficient equal to 0.5 and
# steepness coefficient equal to 0.5
F2 <- pBMT(z, 0.5, 0.5, "a-s")
Q2 <- qBMT(F2, 0.5, 0.5, "a-s")
f2 <- dBMT(z, 0.5, 0.5, "a-s")
r2 <- rBMT(100, 0.5, 0.5, "a-s")
max(abs(f1 - f2))
max(abs(F1 - F2))
max(abs(Q1 - Q2))

# BMT on [-1.783489, 3.312195] with
# left tail weight equal to 0.25 and
# right tail weight equal to 0.75
x <- seq(-1.783489, 3.312195, length.out = 100)
F3 <- pBMT(x, 0.25, 0.75, "t w", -1.783489, 3.312195, "c-d")
Q3 <- qBMT(F3, 0.25, 0.75, "t w", -1.783489, 3.312195, "c-d")
max(abs(x - Q3))
f3 <- dBMT(x, 0.25, 0.75, "t w", -1.783489, 3.312195, "c-d")
r3 <- rBMT(100, 0.25, 0.75, "t w", -1.783489, 3.312195, "c-d")
layout(matrix(c(1,2,1,3), 2, 2))
hist(r3, freq = FALSE, xlim = c(-1.783489,3.312195))
lines(x, f3)
plot(x, F3, type="l")
plot(F3, Q3, type="l")

# BMT with mean equal to 0, standard deviation equal to 1,
# asymmetry coefficient equal to 0.5 and
# steepness coefficient equal to 0.5
f4 <- dBMT(x, 0.5, 0.5, "a-s", 0, 1, "l-s")
F4 <- pBMT(x, 0.5, 0.5, "a-s", 0, 1, "l-s")
Q4 <- qBMT(F4, 0.5, 0.5, "a-s", 0, 1, "l-s")
r4 <- rBMT(100, 0.5, 0.5, "a-s", 0, 1, "l-s")
max(abs(f3 - f4))
max(abs(F3 - F4))
max(abs(Q3 - Q4))
```

`BMT.Phi` *The BMT-Phi Distribution.*

Description

Density, distribution function, quantile function, random number generation for the BMT-Phi distribution with mean equal to `mean` and standard deviation equal to `sd`.

Usage

```
dBMT.Phi(x, mean = 0, sd = 1, log = FALSE)

pBMT.Phi(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)

qBMT.Phi(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)

rBMT.Phi(n, mean = 0, sd = 1)
```

Arguments

`x, q`	vector of quantiles.
`mean`	vector of means.
`sd`	vector of standard deviations.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$, otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required

Details

If `mean` or `sd` are not specified they assume the default values of 0 and 1, respectively.

The BMT-Phi distribution is the BMT distribution with $\kappa_l = \kappa_r = 0.58029164978583758$. The BMT-Phi cumulative distribution function (cdf) is the closest BMT cdf to the normal cdf with the same mean and standard deviation.

Value

`dBMT.Phi` gives the density, `pBMT.Phi` the distribution function, `qBMT.Phi` the quantile function, and `rBMT.Phi` generates random deviates.

The length of the result is determined by `n` for `rBMT.Phi`, and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than `n` are recycled to the length of the result. Only the first elements of the logical arguments are used.

`sd <= 0` is an error and returns `NaN`.

Author(s)

Camilo Jose Torres-Jimenez [aut,cre] <`cjtorresj@unal.edu.co`>

References

Torres-Jimenez, C. J. (2018), *The BMT Item Response Theory model: A new skewed distribution family with bounded domain and an IRT model based on it*, PhD thesis, Doctorado en ciencias - Estadistica, Universidad Nacional de Colombia, Sede Bogota.

See Also

Distributions for other standard distributions. `pBMT` for the BMT distribution and `pBMT.Psi` for the BMT-Psi distribution.

Examples

```
layout(matrix(1:4,2,2))

curve(pnorm(x), -4, 4, col = "red", lty = 2, ylab = "cdf")
curve(pBMT.Phi(x), add = TRUE, col = "blue", lty = 3)
legend("topleft", legend = c("norm(0,1)","BMT-Phi(0,1)"),
       bty = "n", col = c("red","blue"), lty = 2:3)

curve(pnorm(x)-pBMT.Phi(x), -4, 4)

curve(qnorm(x), col = "red", lty = 2, xlab = "p", ylab = "qf")
curve(qBMT.Phi(x), add = TRUE, col = "blue", lty = 3)

hist(rBMT.Phi(10000), freq = FALSE, breaks = seq(-4,4,0.25), border = "blue")
curve(dnorm(x), add = TRUE, col = "red", lty = 2)
curve(dBMT.Phi(x), add = TRUE, col = "blue", lty = 3)
```

`BMT.Psi` *The BMT-Psi Distribution.*

Description

Density, distribution function, quantile function, random number generation for the BMT-Psi distribution with mean equal to `mean` and standard deviation equal to `sd`.

Usage

```
dBMT.Psi(x, mean = 0, sd = 1, log = FALSE)

pBMT.Psi(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)

qBMT.Psi(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)

rBMT.Psi(n, mean = 0, sd = 1)
```

Arguments

`x, q`	vector of quantiles.
`mean`	vector of means.
`sd`	vector of standard deviations.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \leq x]$, otherwise, $P[X > x]$.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required

Details

If `mean` or `sd` are not specified they assume the default values of 0 and 1, respectively.

The BMT-Psi distribution is the BMT distribution with $\kappa_l = \kappa_r = 0.63355781127887611515$. The BMT-Psi cumulative distribution function (cdf) is the closest BMT cdf to the logistic cdf with scale = 1 / d and d = 1.70174439 (Camilli, 1994, p. 295).

Value

`dBMT.Psi` gives the density, `pBMT.Psi` the distribution function, `qBMT.Psi` the quantile function, and `rBMT.Psi` generates random deviates.

The length of the result is determined by `n` for `rBMT.Psi`, and is the maximum of the lengths of the numerical arguments for the other functions.

The numerical arguments other than `n` are recycled to the length of the result. Only the first elements of the logical arguments are used.

`sd <= 0` is an error and returns `NaN`.

Author(s)

Camilo Jose Torres-Jimenez [aut,cre] <`cjtorresj@unal.edu.co`>

References

Torres-Jimenez, C. J. (2018), *The BMT Item Response Theory model: A new skewed distribution family with bounded domain and an IRT model based on it*, PhD thesis, Doctorado en ciencias - Estadistica, Universidad Nacional de Colombia, Sede Bogota.

Camilli, G. (1994). Teacher's corner: origin of the scaling constant d= 1.7 in item response theory. Journal of Educational Statistics, 19(3), 293-295.

See Also

Distributions for other standard distributions. `pBMT` for the BMT distribution and `pBMT.Phi` for the BMT-Phi distribution.

Examples

```
layout(matrix(1:4, 2, 2))

curve(plogis(x, scale = 1 / 1.70174439), -4, 4, col = "red", lty = 2, ylab = "cdf")
curve(pBMT.Psi(x), add = TRUE, col = "blue", lty = 3)
legend("topleft", legend = c("logis(0, 1 / 1.70174439)","BMT-Psi(0,1)"),
       bty = "n", col = c("red","blue"), lty = 2:3)

curve(plogis(x, scale = 1 / 1.70174439)-pBMT.Psi(x), -4, 4)

curve(qlogis(x, scale = 1 / 1.70174439), col = "red", lty = 2, xlab = "p", ylab = "qf")
curve(qBMT.Psi(x), add = TRUE, col = "blue", lty = 3)

hist(rBMT.Psi(10000), freq = FALSE, breaks = seq(-4, 4, 0.25), border = "blue")
curve(dlogis(x, scale = 1 / 1.70174439), add = TRUE, col = "red", lty = 2)
curve(dBMT.Psi(x), add = TRUE, col = "blue", lty = 3)
```

`BMTcentral` *The BMT Distribution Descriptive Measures - Central Tendency.*

Description

Mean, median and mode for the BMT distribution, with `p3` and `p4` tails weights (κ_l and κ_r) or asymmetry-steepness parameters (ζ and ξ) and `p1` and `p2` domain (minimum and maximum) or location-scale (mean and standard deviation) parameters.

Usage

```
BMTmean(p3, p4, type.p.3.4 = "t w", p1 = 0, p2 = 1, type.p.1.2 = "c-d")

BMTmedian(p3, p4, type.p.3.4 = "t w", p1 = 0, p2 = 1, type.p.1.2 = "c-d")

BMTmode(p3, p4, type.p.3.4 = "t w", p1 = 0, p2 = 1, type.p.1.2 = "c-d")
```

Arguments

`p3, p4`	tails weights (κ_l and κ_r) or asymmetry-steepness (ζ and ξ) parameters of the BMT distribution.
`type.p.3.4`	type of parametrization associated to p3 and p4. "t w" means tails weights parametrization (default) and "a-s" means asymmetry-steepness parametrization.
`p1, p2`	domain (minimum and maximum) or location-scale (mean and standard deviation) parameters of the BMT distribution.
`type.p.1.2`	type of parametrization associated to p1 and p2. "c-d" means domain parametrization (default) and "l-s" means location-scale parametrization.

Details

See References.

Value

`BMTmean` gives the mean, `BMTmedian` the median and `BMTmode` the mode for the BMT distribution.

The arguments are recycled to the length of the result. Only the first elements of `type.p.3.4` and `type.p.1.2` are used.

If `type.p.3.4 == "t w"`, `p3 < 0` and `p3 > 1` are errors and return `NaN`.

If `type.p.3.4 == "a-s"`, `p3 < -1` and `p3 > 1` are errors and return `NaN`.

`p4 < 0` and `p4 > 1` are errors and return `NaN`.

If `type.p.1.2 == "c-d"`, `p1 >= p2` is an error and returns `NaN`.

If `type.p.1.2 == "l-s"`, `p2 <= 0` is an error and returns `NaN`.

Author(s)

Camilo Jose Torres-Jimenez [aut,cre] `<cjtorresj@unal.edu.co>`

References

Torres-Jimenez, C. J. and Montenegro-Diaz, A. M. (2017, September), *An alternative to continuous univariate distributions supported on a bounded interval: The BMT distribution*. ArXiv e-prints. `https://arxiv.org/abs/1709.05534`.

Torres-Jimenez, C. J. (2018), *The BMT Item Response Theory model: A new skewed distribution family with bounded domain and an IRT model based on it*, PhD thesis, Doctorado en ciencias - Estadistica, Universidad Nacional de Colombia, Sede Bogota.

See Also

`BMTdispersion`, `BMTskewness`, `BMTkurtosis`, `BMTmoments` for other descriptive measures or moments.

Examples

```
# BMT on [0,1] with left tail weight equal to 0.25 and
# right tail weight equal to 0.75
BMTmean(0.25, 0.75, "t w")
BMTmedian(0.25, 0.75, "t w")
BMTmode(0.25, 0.75, "t w")

# BMT on [0,1] with asymmetry coefficient equal to 0.5 and
# steepness coefficient equal to 0.75
BMTmean(0.5, 0.5, "a-s")
BMTmedian(0.5, 0.5, "a-s")
BMTmode(0.5, 0.5, "a-s")

# BMT on [-1.783489,3.312195] with
# left tail weight equal to 0.25 and
```

```
# right tail weight equal to 0.75
BMTmean(0.25, 0.75, "t w", -1.783489, 3.312195, "c-d")
BMTmedian(0.25, 0.75, "t w", -1.783489, 3.312195, "c-d")
BMTmode(0.25, 0.75, "t w", -1.783489, 3.312195, "c-d")

# BMT with mean equal to 0, standard deviation equal to 1,
# asymmetry coefficient equal to 0.5 and
# steepness coefficient equal to 0.75
BMTmean(0.5, 0.5, "a-s", 0, 1, "l-s")
BMTmedian(0.5, 0.5, "a-s", 0, 1, "l-s")
BMTmode(0.5, 0.5, "a-s", 0, 1, "l-s")
```

`BMTchangepars` *The BMT Distribution Parameter Conversion.*

Description

Parameter conversion for different parameterizations for the BMT distribution, with `p3` and `p4` tails weights (κ_l and κ_r) or asymmetry-steepness parameters (ζ and ξ) and `p1` and `p2` domain (minimum and maximum) or location-scale (mean and standard deviation) parameters.

Usage

```
BMTchangepars(
  p3,
  p4,
  type.p.3.4 = "t w",
  p1 = NULL,
  p2 = NULL,
  type.p.1.2 = NULL
)
```

Arguments

`p3, p4`	tails weights (κ_l and κ_r) or asymmetry-steepness (ζ and ξ) parameters of the BMT distribution.
`type.p.3.4`	type of parametrization associated to p3 and p4. "t w" means tails weights parametrization (default) and "a-s" means asymmetry-steepness parametrization.
`p1, p2`	domain (minimum and maximum) or location-scale (mean and standard deviation) parameters of the BMT distribution.
`type.p.1.2`	type of parametrization associated to p1 and p2. "c-d" means domain parametrization (default) and "l-s" means location-scale parametrization.

Details

The BMT coefficient of asymmetry $-1 < \zeta < 1$ is

$$\kappa_r - \kappa_l$$

The BMT coefficient of steepness $0 < \xi < 1$ is

$$(\kappa_r + \kappa_l - |\kappa_r - \kappa_l|)/(2(1 - |\kappa_r - \kappa_l|))$$

for $|\kappa_r - \kappa_l| < 1$.

The BMT distribution has mean $(d - c)BMTmean(\kappa_l, \kappa_r) + c$ and standard deviation $(d - c)BMTsd(\kappa_l, \kappa_r)$

From these equations, we can go back and forth with each parameterization.

Value

`BMTchangepars` reparametrize `p3`, `p4`, `p1`, `p2` according to the alternative parameterizations from the given `type.p.3.4` and `type.p.1.2`. `BMTchangepars` returns a list with the alternative arguments to those received.

The arguments are recycled to the length of the result. Only the first elements of `type.p.3.4` and `type.p.1.2` are used.

If `type.p.3.4 == "t w"`, `p3 < 0` and `p3 > 1` are errors and return `NaN`.

If `type.p.3.4 == "a-s"`, `p3 < -1` and `p3 > 1` are errors and return `NaN`.

`p4 < 0` and `p4 > 1` are errors and return `NaN`.

If `type.p.1.2 == "c-d"`, `p1 >= p2` is an error and returns `NaN`.

If `type.p.1.2 == "l-s"`, `p2 <= 0` is an error and returns `NaN`.

Author(s)

Camilo Jose Torres-Jimenez [aut,cre] <`cjtorresj@unal.edu.co`> and Alvaro Mauricio Montenegro Diaz [ths]

References

Torres-Jimenez, C. J. (2018), *The BMT Item Response Theory model: A new skewed distribution family with bounded domain and an IRT model based on it*, PhD thesis, Doctorado en ciencias - Estadistica, Universidad Nacional de Colombia, Sede Bogota.

See Also

`BMT` for the BMT density, distribution, quantile function and random deviates.

Examples

```
# BMT on [0,1] with left tail weight equal to 0.25 and
# right tail weight equal to 0.75
parameters <- BMTchangepars(0.25, 0.75, "t w")
parameters # Parameters of the BMT in the asymmetry-steepness parametrization

# BMT with mean equal to 0, standard deviation equal to 1,
# asymmetry coefficient equal to 0.5 and
# steepness coefficient equal to 0.75
parameters <- BMTchangepars(0.5, 0.5, "a-s", 0, 1, "l-s")
parameters # Parameters of the BMT in the tail weight and domain parametrization
```

`BMTdispersion` *The BMT Distribution Descriptive Measures - Dispersion.*

Description

Variance, standard deviation and interquartile range for the BMT distribution, with `p3` and `p4` tails weights (κ_l and κ_r) or asymmetry-steepness parameters (ζ and ξ) and `p1` and `p2` domain (minimum and maximum) or location-scale (mean and standard deviation) parameters.

Usage

```
BMTvar(p3, p4, type.p.3.4 = "t w", p1 = 0, p2 = 1, type.p.1.2 = "c-d")

BMTsd(p3, p4, type.p.3.4 = "t w", p1 = 0, p2 = 1, type.p.1.2 = "c-d")

BMTiqr(p3, p4, type.p.3.4 = "t w", p1 = 0, p2 = 1, type.p.1.2 = "c-d")
```

Arguments

`p3, p4`	tails weights (κ_l and κ_r) or asymmetry-steepness (ζ and ξ) parameters of the BMT distribution.
`type.p.3.4`	type of parametrization associated to p3 and p4. "t w" means tails weights parametrization (default) and "a-s" means asymmetry-steepness parametrization.
`p1, p2`	domain (minimum and maximum) or location-scale (mean and standard deviation) parameters of the BMT distribution.
`type.p.1.2`	type of parametrization associated to p1 and p2. "c-d" means domain parametrization (default) and "l-s" means location-scale parametrization.

Details

See References.

Value

`BMTvar` gives the variance, `BMTsd` the standard deviation and `BMTiqr` the interquartile range for the BMT distribution.

The arguments are recycled to the length of the result. Only the first elements of `type.p.3.4` and `type.p.1.2` are used.

If `type.p.3.4 == "t w"`, `p3 < 0` and `p3 > 1` are errors and return `NaN`.

If `type.p.3.4 == "a-s"`, `p3 < -1` and `p3 > 1` are errors and return `NaN`.

`p4 < 0` and `p4 > 1` are errors and return `NaN`.

If `type.p.1.2 == "c-d"`, `p1 >= p2` is an error and returns `NaN`.

If `type.p.1.2 == "l-s"`, `p2 <= 0` is an error and returns `NaN`.

Author(s)

Camilo Jose Torres-Jimenez [aut,cre] <`cjtorresj@unal.edu.co`>

References

Torres-Jimenez, C. J. and Montenegro-Diaz, A. M. (2017, September), *An alternative to continuous univariate distributions supported on a bounded interval: The BMT distribution*. ArXiv e-prints. `https://arxiv.org/abs/1709.05534`.

Torres-Jimenez, C. J. (2018), *The BMT Item Response Theory model: A new skewed distribution family with bounded domain and an IRT model based on it*, PhD thesis, Doctorado en ciencias - Estadistica, Universidad Nacional de Colombia, Sede Bogota.

See Also

`BMTcentral`, `BMTskewness`, `BMTkurtosis`, `BMTmoments` for other descriptive measures or moments.

Examples

```
# BMT on [0,1] with left tail weight equal to 0.25 and
# right tail weight equal to 0.75
BMTvar(0.25, 0.75, "t w")
BMTsd(0.25, 0.75, "t w")
BMTiqr(0.25, 0.75, "t w")

# BMT on [0,1] with asymmetry coefficient equal to 0.5 and
# steepness coefficient equal to 0.75
BMTvar(0.5, 0.5, "a-s")
BMTsd(0.5, 0.5, "a-s")
BMTiqr(0.5, 0.5, "a-s")

# BMT on [-1.783489,3.312195] with left tail weight equal to 0.25 and
# right tail weight equal to 0.75
BMTvar(0.25, 0.75, "t w", -1.783489, 3.312195, "c-d")
BMTsd(0.25, 0.75, "t w", -1.783489, 3.312195, "c-d")
BMTiqr(0.25, 0.75, "t w", -1.783489, 3.312195, "c-d")
```

```
# BMT with mean equal to 0, standard deviation equal to 1,
# asymmetry coefficient equal to 0.5 and
# steepness coefficient equal to 0.75
BMTvar(0.5, 0.5, "a-s", 0, 1, "l-s")
BMTsd(0.5, 0.5, "a-s", 0, 1, "l-s")
BMTiqr(0.5, 0.5, "a-s", 0, 1, "l-s")
```

`BMTfit` *Fit of the BMT Distribution to Non-censored Data.*

Description

Fit of the BMT distribution to non-censored data by maximum likelihood (mle), moment matching (mme), quantile matching (qme), maximum goodness-of-fit (mge), also known as minimum distance, maximum product of spacing (mpse), also called maximum spacing, and minimum quantile distance (mqde), which can also be called maximum quantile goodness-of-fit.

Usage

```
BMTfit(
  data,
  method = c("mle", "mme", "qme", "mge", "mpse", "mqde"),
  start = list(p3 = 0.5, p4 = 0.5, p1 = min(data) - 0.1, p2 = max(data) + 0.1),
  fix.arg = NULL,
  type.p.3.4 = "t w",
  type.p.1.2 = "c-d",
  optim.method = "Nelder-Mead",
  custom.optim = NULL,
  keepdata = TRUE,
  keepdata.nb = 100,
  ...
)
```

Arguments

`data`	A numeric vector with the observed values for non-censored data.
`method`	A character string coding for the fitting method: `"mle"` for 'maximum likelihood estimation', `"mme"` for 'moment matching estimation', `"qme"` for 'quantile matching estimation', `"mge"` for 'maximum goodness-of-fit estimation', `"mpse"` for 'maximum product of spacing estimation', and `"mqde"` for 'minimum quantile estimation'.
`start`	A named list giving the initial values of parameters of the BMT distribution or a function of data computing initial values and returning a named list. (see the 'details' section of `mledist`).

`fix.arg`	An optional named list giving the values of fixed parameters of the BMT distribution or a function of data computing (fixed) parameter values and returning a named list. Parameters with fixed value are thus NOT estimated. (see the 'details' section of `mledist`).
`type.p.3.4`	Type of parametrization associated to p3 and p4. "t w" means tails weights parametrization (default) and "a-s" means asymmetry-steepness parametrization.
`type.p.1.2`	Type of parametrization associated to p1 and p2. "c-d" means domain parametrization (default) and "l-s" means location-scale parametrization.
`optim.method`	`"default"` (see the 'details' section of `mledist`) or optimization method to pass to `optim`.
`custom.optim`	A function carrying the optimization (see the 'details' section of `mledist`).
`keepdata`	A logical. If `TRUE`, dataset is returned, otherwise only a sample subset is returned.
`keepdata.nb`	When `keepdata=FALSE`, the length (>1) of the subset returned.
`...`	Further arguments to be passed to generic functions, or to one of the functions `"BMTfit.mle"`, `"BMTfit.mme"`, `"BMTfit.qme"`, `"BMTfit.mge"`, `"BMTfit.mpse"`, or `"BMTfit.mqde"` depending of the chosen method. See `BMTfit.mle`, `BMTfit.mme`, `BMTfit.qme`, `BMTfit.mge`, `BMTfit.mpse`, `BMTfit.mqde` for details on parameter estimation.

Details

This function is based on the function `fitdist` but it focuses on the parameter estimation for the BMT distribution (see `BMT` for details). It has six possible fitting methods: maximum likelihood (mle), moment matching (mme), quantile matching (qme), maximum goodness-of-fit (mge), also known as minimum distance, maximum product of spacing (mpse), also called maximum spacing, and minimum quantile distance (mqde), which can also be called maximum quantile goodness-of-fit. These fitting methods are carried out in `BMTfit.mle`, `BMTfit.mme`, `BMTfit.qme`, `BMTfit.mge`, `BMTfit.mpse`, and `BMTfit.mqde`, respectively (see each function for details). `BMTfit` returns an object of class `"fitdist"` (see `fitdist` for details). Therefore, it benefits of all the developed functions and methods for that class (see `fitdistrplus` for details).

Generic methods of a `fitdist` object are `print`, `plot`, `summary`, `quantile`, `logLik`, `vcov` and `coef`.

Value

`fitdist` returns an object of class `"fitdist"` with the following components:

`estimate`	the parameter estimates.
`method`	the character string coding for the fitting method : `"mle"` for 'maximum likelihood estimation', `"mme"` for 'moment matching estimation', `"qme"` for 'quantile matching estimation', `"mge"` for 'maximum goodness-of-fit estimation', `"mpse"` for 'maximum product of spacing estimation', and `"mqde"` for 'minimum quantile estimation'.
`sd`	the estimated standard errors, `NA` if numerically not computable or `NULL` if not available.

`cor`	the estimated correlation matrix, `NA` if numerically not computable or `NULL` if not available.
`vcov`	the estimated variance-covariance matrix, `NULL` if not available.
`loglik`	the log-likelihood.
`aic`	the Akaike information criterion.
`bic`	the the so-called BIC or SBC (Schwarz Bayesian criterion).
`n`	the length of the data set.
`data`	the data set.
`distname`	the name of the distribution (BMT).
`fix.arg`	the named list giving the values of parameters of the named distribution that must be kept fixed rather than estimated or `NULL` if there are no such parameters.
`fix.arg.fun`	the function used to set the value of `fix.arg` or `NULL`.
`discrete`	the input argument or the automatic definition by the function to be passed to functions `gofstat`, `plotdist` and `cdfcomp`.
`dots`	the list of further arguments passed in ... to be used in `bootdist` in iterative calls to `mledist`, `mmedist`, `qmedist`, `mgedist`, `mpsedist`, `mqdedist` or `NULL` if no such arguments.
`weights`	the vector of weights used in the estimation process or `NULL`.

Author(s)

Camilo Jose Torres-Jimenez [aut,cre] <`cjtorresj@unal.edu.co`>

Source

Based on the function `fitdist` of the R package: **fitdistrplus**

Delignette-Muller ML and Dutang C (2015), *fitdistrplus: An R Package for Fitting Distributions*. Journal of Statistical Software, 64(4), 1-34.

References

Torres-Jimenez, C. J. (2017, September), *Comparison of estimation methods for the BMT distribution*. ArXiv e-prints.

Torres-Jimenez, C. J. (2018), *The BMT Item Response Theory model: A new skewed distribution family with bounded domain and an IRT model based on it*, PhD thesis, Doctorado en ciencias - Estadistica, Universidad Nacional de Colombia, Sede Bogota.

See Also

See `BMT` for the BMT density, distribution, quantile function and random deviates. See `BMTfit.mle`, `BMTfit.mme`, `BMTfit.qme`, `BMTfit.mge`, `BMTfit.mpse` and `BMTfit.mqde` for details on parameter estimation. See `fitdist` for details on the object fitdist and its methods `print`, `plot`, `summary`, `quantile`, `logLik`, `vcov` and `coef`, and **fitdistrplus** for an overview of the package to which that object belongs to.

Examples

```
# (1) fit of the BMT distribution by maximum likelihood estimation
data(groundbeef)
serving <- groundbeef$serving
fit.mle <- BMTfit(serving)
summary(fit.mle)
plot(fit.mle)
plot(fit.mle, demp = TRUE)
plot(fit.mle, histo = FALSE, demp = TRUE)
cdfcomp(fit.mle, addlegend=FALSE)
denscomp(fit.mle, addlegend=FALSE)
ppcomp(fit.mle, addlegend=FALSE)
qqcomp(fit.mle, addlegend=FALSE)

# (2) Comparison of various estimation methods
fit.mme <- BMTfit(serving, method="mme")
fit.mpse <- BMTfit(serving, method="mpse")
fit.mqde <- BMTfit(serving, method="mqde")
summary(fit.mme)
summary(fit.mpse)
summary(fit.mqde)
cdfcomp(list(fit.mle, fit.mme, fit.mpse, fit.mqde),
        legendtext=c("mle", "mme", "mpse", "mqde"))
denscomp(list(fit.mle, fit.mme, fit.mpse, fit.mqde),
         legendtext=c("mle", "mme", "mpse", "mqde"))
qqcomp(list(fit.mle, fit.mme, fit.mpse, fit.mqde),
       legendtext=c("mle", "mme", "mpse", "mqde"))
ppcomp(list(fit.mle, fit.mme, fit.mpse, fit.mqde),
       legendtext=c("mle", "mme", "mpse", "mqde"))
gofstat(list(fit.mle, fit.mme, fit.mpse, fit.mqde),
        fitnames=c("mle", "mme", "mpse", "mqde"))

# (3) how to change the optimisation method?
BMTfit(serving, optim.method="Nelder-Mead")
BMTfit(serving, optim.method="L-BFGS-B")
BMTfit(serving, custom.optim="nlminb")

# (4) estimation of the tails weights parameters of the BMT distribution
# with domain fixed at [9,201] using Kolmogorov-Smirnov
fit.KS <- BMTfit(serving, method="mge", gof="KS",
                 start=list(p3=0.5, p4=0.5), fix.arg=list(p1=9, p2=201))
summary(fit.KS)
plot(fit.KS)

# (5) estimation of the asymmetry-steepness parameters of the BMT
# distribution with domain fixed at [9,201] using minimum quantile distance
# with a closed formula (optim.method="CD")
fit.mqde.CD <- BMTfit(serving, method="mqde", optim.method="CD",
                      start=list(p3=0.5, p4=0.5), type.p.3.4 = "a-s",
                      fix.arg=list(p1=9, p2=201))
summary(fit.mqde.CD)
plot(fit.mqde.CD)
```

`BMTfit.mge` *Maximum Goodness-of-fit Fit of the BMT Distribution to Non-censored Data.*

Description

Fit of the BMT distribution to non-censored data by maximum goodness-of-fit estimation (mge), also known as minimum distance estimation.

Usage

```
BMTfit.mge(
  data,
  gof = "CvM",
  start = list(p3 = 0.5, p4 = 0.5, p1 = min(data) - 0.1, p2 = max(data) + 0.1),
  fix.arg = NULL,
  type.p.3.4 = "t w",
  type.p.1.2 = "c-d",
  optim.method = "Nelder-Mead",
  custom.optim = NULL,
  silent = TRUE,
  ...
)
```

Arguments

`data`	A numeric vector with the observed values for non-censored data.
`gof`	A character string coding for the name of the goodness-of-fit distance used : "CvM" for Cramer-von Mises distance,"KS" for Kolmogorov-Smirnov distance, "AD" for Anderson-Darling distance, "ADR", "ADL", "AD2R", "AD2L" and "AD2" for variants of Anderson-Darling distance described by Luceno (2006).
`start`	A named list giving the initial values of parameters of the BMT distribution or a function of data computing initial values and returning a named list. (see the 'details' section of `mledist`).
`fix.arg`	An optional named list giving the values of fixed parameters of the BMT distribution or a function of data computing (fixed) parameter values and returning a named list. Parameters with fixed value are thus NOT estimated. (see the 'details' section of `mledist`).
`type.p.3.4`	Type of parametrization associated to p3 and p4. "t w" means tails weights parametrization (default) and "a-s" means asymmetry-steepness parametrization.
`type.p.1.2`	Type of parametrization associated to p1 and p2. "c-d" means domain parametrization (default) and "l-s" means location-scale parametrization.

`optim.method`	`"default"` (see the 'details' section of `mledist`) or optimization method to pass to `optim`.
`custom.optim`	A function carrying the optimization (see the 'details' section of `mledist`).
`silent`	A logical to remove or show warnings when bootstrapping.
`...`	Further arguments to be passed to generic functions or to the function `"mgedist"`. See `mgedist` for details.

Details

This function is not intended to be called directly but is internally called in `BMTfit` when used with the maximum goodness-of-fit method.

`BMTfit.mge` is based on the function `mgedist` but it focuses on the maximum goodness-of-fit parameter estimation for the BMT distribution (see `BMT` for details about the BMT distribution and `mgedist` for details about maximum goodness-of-fit fit of univariate distributions).

Value

`BMTfit.mge` returns a list with following components,

`estimate`	the parameter estimates.
`convergence`	an integer code for the convergence of `optim`/`constrOptim` defined as below or defined by the user in the user-supplied optimization function. `0` indicates successful convergence. `1` indicates that the iteration limit of `optim` has been reached. `10` indicates degeneracy of the Nealder-Mead simplex. `100` indicates that `optim` encountered an internal error.
`value`	the value of the corresponding objective function of the estimation method at the estimate.
`hessian`	a symmetric matrix computed by `optim` as an estimate of the Hessian at the solution found or computed in the user-supplied optimization function.
`loglik`	the log-likelihood value.
`gof`	the code of the goodness-of-fit distance maximized.
`optim.function`	the name of the optimization function used for maximum product of spacing.
`optim.method`	when `optim` is used, the name of the algorithm used, `NULL` otherwise.
`fix.arg`	the named list giving the values of parameters of the named distribution that must kept fixed rather than estimated or `NULL` if there are no such parameters.
`fix.arg.fun`	the function used to set the value of `fix.arg` or `NULL`.
`weights`	the vector of weights used in the estimation process or `NULL`.
`counts`	A two-element integer vector giving the number of calls to the log-likelihood function and its gradient respectively. This excludes those calls needed to compute the Hessian, if requested, and any calls to log-likelihood function to compute a finite-difference approximation to the gradient. `counts` is returned by `optim` or the user-supplied function or set to `NULL`.
`optim.message`	A character string giving any additional information returned by the optimizer, or `NULL`. To understand exactly the message, see the source code.

Author(s)

Camilo Jose Torres-Jimenez [aut,cre] `<cjtorresj@unal.edu.co>`

Source

Based on the function `mgedist` of the R package: **fitdistrplus**

Delignette-Muller ML and Dutang C (2015), *fitdistrplus: An R Package for Fitting Distributions*. Journal of Statistical Software, 64(4), 1-34.

References

Torres-Jimenez, C. J. (2017, September), *Comparison of estimation methods for the BMT distribution*. ArXiv e-prints.

Torres-Jimenez, C. J. (2018), *The BMT Item Response Theory model: A new skewed distribution family with bounded domain and an IRT model based on it*, PhD thesis, Doctorado en ciencias - Estadistica, Universidad Nacional de Colombia, Sede Bogota.

See Also

See `BMT` for the BMT density, distribution, quantile function and random deviates. See `BMTfit.mme`, `BMTfit.qme`, `BMTfit.mle`, `BMTfit.mpse` and `BMTfit.mqde` for other estimation methods. See `optim` and `constrOptim` for optimization routines. See `BMTfit` and `fitdist` for functions that return an object of class `"fitdist"`.

Examples

```
# (1) basic fit by maximum goodness-of-fit estimation
set.seed(1234)
x1 <- rBMT(n=100, p3 = 0.25, p4 = 0.75)
BMTfit.mge(x1)

# (2) how to change the goodness-of-fit statistic/distance?
BMTfit.mge(x1, gof="KS")
BMTfit.mge(x1, gof="AD2R")

# (3) how to change the optimisation method?
BMTfit.mge(x1, optim.method="L-BFGS-B")
BMTfit.mge(x1, custom.optim="nlminb")

# (4) estimation of the tails weights parameters of the BMT
# distribution with domain fixed at [0,1]
BMTfit.mge(x1, start=list(p3=0.5, p4=0.5), fix.arg=list(p1=0, p2=1))

# (5) estimation of the asymmetry-steepness parameters of the BMT
# distribution with domain fixed at [0,1]
BMTfit.mge(x1, start=list(p3=0, p4=0.5), type.p.3.4 = "a-s",
           fix.arg=list(p1=0, p2=1))
```

`BMTfit.mle` *Maximum Likelihood Fit of the BMT Distribution to Non-censored Data.*

Description

Fit of the BMT distribution to non-censored data by maximum likelihood estimation (mle).

Usage

```
BMTfit.mle(
  data,
  start = list(p3 = 0.5, p4 = 0.5, p1 = min(data) - 0.1, p2 = max(data) + 0.1),
  fix.arg = NULL,
  type.p.3.4 = "t w",
  type.p.1.2 = "c-d",
  optim.method = "Nelder-Mead",
  custom.optim = NULL,
  silent = TRUE,
  ...
)
```

Arguments

`data`	A numeric vector with the observed values for non-censored data.
`start`	A named list giving the initial values of parameters of the BMT distribution or a function of data computing initial values and returning a named list. (see the 'details' section of `mledist`).
`fix.arg`	An optional named list giving the values of fixed parameters of the BMT distribution or a function of data computing (fixed) parameter values and returning a named list. Parameters with fixed value are thus NOT estimated. (see the 'details' section of `mledist`).
`type.p.3.4`	Type of parametrization associated to p3 and p4. "t w" means tails weights parametrization (default) and "a-s" means asymmetry-steepness parametrization.
`type.p.1.2`	Type of parametrization associated to p1 and p2. "c-d" means domain parametrization (default) and "l-s" means location-scale parametrization.
`optim.method`	`"default"` (see the 'details' section of `mledist`) or optimization method to pass to `optim`.
`custom.optim`	A function carrying the optimization (see the 'details' section of `mledist`).
`silent`	A logical to remove or show warnings when bootstrapping.
`...`	Further arguments to be passed to generic functions or to the function `"mledist"`. See `mledist` for details.

Details

This function is not intended to be called directly but is internally called in `BMTfit` when used with the maximum likelihood method.

`BMTfit.mle` is based on the function `mledist` from the package **fitdistrplus** but it focuses on the maximum likelihood parameter estimation for the BMT distribution (see **BMT** for details about the BMT distribution and `mledist` for details about maximum likelihood fit of univariate distributions).

Value

`BMTfit.mle` returns a list with following components,

`estimate`	the parameter estimates.
`convergence`	an integer code for the convergence of `optim`/`constrOptim` defined as below or defined by the user in the user-supplied optimization function. `0` indicates successful convergence. `1` indicates that the iteration limit of `optim` has been reached. `10` indicates degeneracy of the Nealder-Mead simplex. `100` indicates that `optim` encountered an internal error.
`loglik`	the log-likelihood value.
`hessian`	a symmetric matrix computed by `optim` as an estimate of the Hessian at the solution found or computed in the user-supplied optimization function. It is used in `BMTfit` to estimate standard errors.
`optim.function`	the name of the optimization function used for maximum likelihood.
`optim.method`	when `optim` is used, the name of the algorithm used, `NULL` otherwise.
`fix.arg`	the named list giving the values of parameters of the named distribution that must kept fixed rather than estimated or `NULL` if there are no such parameters.
`fix.arg.fun`	the function used to set the value of `fix.arg` or `NULL`.
`weights`	the vector of weights used in the estimation process or `NULL`.
`counts`	A two-element integer vector giving the number of calls to the log-likelihood function and its gradient respectively. This excludes those calls needed to compute the Hessian, if requested, and any calls to log-likelihood function to compute a finite-difference approximation to the gradient. `counts` is returned by `optim` or the user-supplied function or set to `NULL`.
`optim.message`	A character string giving any additional information returned by the optimizer, or `NULL`. To understand exactly the message, see the source code.

Author(s)

Camilo Jose Torres-Jimenez [aut,cre] `<cjtorresj@unal.edu.co>`

Source

Based on the function `mledist` of the R package: **fitdistrplus**

Delignette-Muller ML and Dutang C (2015), *fitdistrplus: An R Package for Fitting Distributions*. Journal of Statistical Software, 64(4), 1-34.

References

Torres-Jimenez, C. J. (2017, September), *Comparison of estimation methods for the BMT distribution*. ArXiv e-prints.

Torres-Jimenez, C. J. (2018), *The BMT Item Response Theory model: A new skewed distribution family with bounded domain and an IRT model based on it*, PhD thesis, Doctorado en ciencias - Estadistica, Universidad Nacional de Colombia, Sede Bogota.

See Also

See `BMT` for the BMT density, distribution, quantile function and random deviates. See `BMTfit.mme`, `BMTfit.qme`, `BMTfit.mge`, `BMTfit.mpse` and `BMTfit.mqde` for other estimation methods. See `optim` and `constrOptim` for optimization routines. See `BMTfit` and `fitdist` for functions that return an object of class `"fitdist"`.

Examples

```
# (1) basic fit by maximum likelihood estimation
set.seed(1234)
x1 <- rBMT(n=100, p3 = 0.25, p4 = 0.75)
BMTfit.mle(x1)

# (2) how to change the optimisation method?
BMTfit.mle(x1, optim.method="L-BFGS-B")
BMTfit.mle(x1, custom.optim="nlminb")

# (3) estimation of the tails weights parameters of the BMT
# distribution with domain fixed at [0,1]
BMTfit.mle(x1, start=list(p3=0.5, p4=0.5), fix.arg=list(p1=0, p2=1))

# (4) estimation of the asymmetry-steepness parameters of the BMT
# distribution with domain fixed at [0,1]
BMTfit.mle(x1, start=list(p3=0, p4=0.5), type.p.3.4 = "a-s",
           fix.arg=list(p1=0, p2=1))
```

`BMTfit.mme` *Moment Matching Fit of the BMT Distribution to Non-censored Data.*

Description

Fit of the BMT distribution to non-censored data by moment matching (mme).

Usage

```
BMTfit.mme(
  data,
  start = list(p3 = 0.5, p4 = 0.5, p1 = min(data) - 0.1, p2 = max(data) + 0.1),
  fix.arg = NULL,
```

```
    type.p.3.4 = "t w",
    type.p.1.2 = "c-d",
    optim.method = "Nelder-Mead",
    custom.optim = NULL,
    silent = TRUE,
    ...
)
```

Arguments

`data`	A numeric vector with the observed values for non-censored data.
`start`	A named list giving the initial values of parameters of the BMT distribution or a function of data computing initial values and returning a named list. (see the 'details' section of `mledist`).
`fix.arg`	An optional named list giving the values of fixed parameters of the BMT distribution or a function of data computing (fixed) parameter values and returning a named list. Parameters with fixed value are thus NOT estimated. (see the 'details' section of `mledist`).
`type.p.3.4`	Type of parametrization associated to p3 and p4. "t w" means tails weights parametrization (default) and "a-s" means asymmetry-steepness parametrization.
`type.p.1.2`	Type of parametrization associated to p1 and p2. "c-d" means domain parametrization (default) and "l-s" means location-scale parametrization.
`optim.method`	`"default"` (see the 'details' section of `mledist`) or optimization method to pass to `optim`.
`custom.optim`	A function carrying the optimization (see the 'details' section of `mledist`).
`silent`	A logical to remove or show warnings when bootstrapping.
`...`	Further arguments to be passed to generic functions or to the function `"mmedist"`. See `mmedist` for details.

Details

This function is not intended to be called directly but is internally called in `BMTfit` when used with the moment matching method.

`BMTfit.mme` is based on the function `mmedist` but it focuses on the moment matching parameter estimation for the BMT distribution (see `BMT` for details about the BMT distribution and `mmedist` for details about moment matching fit of univariate distributions).

For each parameter of the BMT distribution we choose a moment or measure. Mean for `p1`, standard deviation for `p2`, Pearson_s skewness for `p3`, and Pearson's kurtosis for `p4`.

Value

`BMTfit.mme` returns a list with following components,

`estimate`	the parameter estimates.

`convergence`	an integer code for the convergence of `optim`/`constrOptim` defined as below or defined by the user in the user-supplied optimization function. `0` indicates successful convergence. `1` indicates that the iteration limit of `optim` has been reached. `10` indicates degeneracy of the Nealder-Mead simplex. `100` indicates that `optim` encountered an internal error.
`value`	the value of the corresponding objective function of the estimation method at the estimate.
`hessian`	a symmetric matrix computed by `optim` as an estimate of the Hessian at the solution found or computed in the user-supplied optimization function.
`loglik`	the log-likelihood value.
`order`	the vector of moment(s) matched: mean (1), standard deviation (2), Pearson's skewness (3), Pearson's kurtosis (4).
`memp`	the empirical moment function.
`optim.function`	the name of the optimization function used for maximum product of spacing.
`optim.method`	when `optim` is used, the name of the algorithm used, `NULL` otherwise.
`fix.arg`	the named list giving the values of parameters of the named distribution that must kept fixed rather than estimated or `NULL` if there are no such parameters.
`fix.arg.fun`	the function used to set the value of `fix.arg` or `NULL`.
`weights`	the vector of weights used in the estimation process or `NULL`.
`counts`	A two-element integer vector giving the number of calls to the log-likelihood function and its gradient respectively. This excludes those calls needed to compute the Hessian, if requested, and any calls to log-likelihood function to compute a finite-difference approximation to the gradient. `counts` is returned by `optim` or the user-supplied function or set to `NULL`.
`optim.message`	A character string giving any additional information returned by the optimizer, or `NULL`. To understand exactly the message, see the source code.

Author(s)

Camilo Jose Torres-Jimenez [aut,cre] <`cjtorresj@unal.edu.co`>

Source

Based on the function `mmedist` of the R package: **fitdistrplus**

Delignette-Muller ML and Dutang C (2015), *fitdistrplus: An R Package for Fitting Distributions*. Journal of Statistical Software, 64(4), 1-34.

References

Torres-Jimenez, C. J. (2017, September), *Comparison of estimation methods for the BMT distribution*. ArXiv e-prints.

Torres-Jimenez, C. J. (2018), *The BMT Item Response Theory model: A new skewed distribution family with bounded domain and an IRT model based on it*, PhD thesis, Doctorado en ciencias - Estadistica, Universidad Nacional de Colombia, Sede Bogota.

See Also

See `BMT` for the BMT density, distribution, quantile function and random deviates. See `BMTfit.qme`, `BMTfit.mle`, `BMTfit.mge`, `BMTfit.mpse` and `BMTfit.mqde` for other estimation methods. See `optim` and `constrOptim` for optimization routines. See `BMTfit` and `fitdist` for functions that return an object of class `"fitdist"`.

Examples

```
# (1) basic fit by moment matching estimation
set.seed(1234)
x1 <- rBMT(n=100, p3=0.25, p4=0.75)
BMTfit.mme(x1)

# (3) how to change the optimisation method?
BMTfit.mme(x1, optim.method="L-BFGS-B")
BMTfit.mme(x1, custom.optim="nlminb")

# (4) estimation of the tails weights parameters of the BMT
# distribution with domain fixed at [0,1]
BMTfit.mme(x1, start=list(p3=0.5, p4=0.5), fix.arg=list(p1=0, p2=1))

# (5) estimation of the asymmetry-steepness parameters of the BMT
# distribution with domain fixed at [0,1]
BMTfit.mme(x1, start=list(p3=0, p4=0.5), type.p.3.4 = "a-s",
           fix.arg=list(p1=0, p2=1))
```

`BMTfit.mpse` *Maximum Product of Spacing Fit of the BMT Distribution to Non-censored Data.*

Description

Fit of the BMT distribution to non-censored data by maximum product of spacing estimation (mpse), also called maximum spacing estimation.

Usage

```
BMTfit.mpse(
  data,
  start = list(p3 = 0.5, p4 = 0.5, p1 = min(data) - 0.1, p2 = max(data) + 0.1),
  fix.arg = NULL,
  type.p.3.4 = "t w",
  type.p.1.2 = "c-d",
  optim.method = "Nelder-Mead",
  custom.optim = NULL,
  silent = TRUE,
  ...
)
```

Arguments

`data`	A numeric vector with the observed values for non-censored data.
`start`	A named list giving the initial values of parameters of the BMT distribution or a function of data computing initial values and returning a named list. (see the 'details' section of `mledist`).
`fix.arg`	An optional named list giving the values of fixed parameters of the BMT distribution or a function of data computing (fixed) parameter values and returning a named list. Parameters with fixed value are thus NOT estimated. (see the 'details' section of `mledist`).
`type.p.3.4`	Type of parametrization associated to p3 and p4. "t w" means tails weights parametrization (default) and "a-s" means asymmetry-steepness parametrization.
`type.p.1.2`	Type of parametrization associated to p1 and p2. "c-d" means domain parametrization (default) and "l-s" means location-scale parametrization.
`optim.method`	`"default"` (see the 'details' section of `mledist`) or optimization method to pass to `optim`.
`custom.optim`	A function carrying the optimization (see the 'details' section of `mledist`).
`silent`	A logical to remove or show warnings when bootstrapping.
`...`	Further arguments to be passed to generic functions or to the function `"mpsedist"`. See `mpsedist` for details.

Details

This function is not intended to be called directly but is internally called in `BMTfit` when used with the maximum product of spacing method.

`BMTfit.mpse` is based on the function `mpsedist` but it focuses on the maximum product of spacing parameter estimation for the BMT distribution (see `BMT` for details about the BMT distribution and `mpsedist` for details about maximum product of spacing fit of univariate distributions).

Value

`BMTfit.mpse` returns a list with following components,

`estimate`	the parameter estimates.
`convergence`	an integer code for the convergence of `optim`/`constrOptim` defined as below or defined by the user in the user-supplied optimization function. `0` indicates successful convergence. `1` indicates that the iteration limit of `optim` has been reached. `10` indicates degeneracy of the Nealder-Mead simplex. `100` indicates that `optim` encountered an internal error.
`value`	the value of the corresponding objective function of the estimation method at the estimate.
`loglik`	the log-likelihood value.

`hessian`	a symmetric matrix computed by `optim` as an estimate of the Hessian at the solution found or computed in the user-supplied optimization function.
`optim.function`	the name of the optimization function used for maximum product of spacing.
`optim.method`	when `optim` is used, the name of the algorithm used, `NULL` otherwise.
`fix.arg`	the named list giving the values of parameters of the named distribution that must kept fixed rather than estimated or `NULL` if there are no such parameters.
`fix.arg.fun`	the function used to set the value of `fix.arg` or `NULL`.
`weights`	the vector of weights used in the estimation process or `NULL`.
`counts`	A two-element integer vector giving the number of calls to the log-likelihood function and its gradient respectively. This excludes those calls needed to compute the Hessian, if requested, and any calls to log-likelihood function to compute a finite-difference approximation to the gradient. `counts` is returned by `optim` or the user-supplied function or set to `NULL`.
`optim.message`	A character string giving any additional information returned by the optimizer, or `NULL`. To understand exactly the message, see the source code.

Author(s)

Camilo Jose Torres-Jimenez [aut,cre] <`cjtorresj@unal.edu.co`>

Source

Based on the function `mpsedist` which in turn is based on the function `mledist` of the R package: **fitdistrplus**

Delignette-Muller ML and Dutang C (2015), *fitdistrplus: An R Package for Fitting Distributions*. Journal of Statistical Software, 64(4), 1-34.

References

Torres-Jimenez, C. J. (2017, September), *Comparison of estimation methods for the BMT distribution*. ArXiv e-prints.

Torres-Jimenez, C. J. (2018), *The BMT Item Response Theory model: A new skewed distribution family with bounded domain and an IRT model based on it*, PhD thesis, Doctorado en ciencias - Estadistica, Universidad Nacional de Colombia, Sede Bogota.

See Also

See `BMT` for the BMT density, distribution, quantile function and random deviates. See `BMTfit.mme`, `BMTfit.qme`, `BMTfit.mge`, `BMTfit.mle` and `BMTfit.mqde` for other estimation methods. See `optim` and `constrOptim` for optimization routines. See `BMTfit` and `fitdist` for functions that return an object of class `"fitdist"`.

Examples

```
# (1) basic fit by maximum product of spacing estimation
set.seed(1234)
x1 <- rBMT(n=100, p3 = 0.25, p4 = 0.75)
BMTfit.mpse(x1)

# (2) how to change the optimisation method?
BMTfit.mpse(x1, optim.method="L-BFGS-B")
BMTfit.mpse(x1, custom.optim="nlminb")

# (3) estimation of the tails weights parameters of the BMT
# distribution with domain fixed at [0,1]
BMTfit.mpse(x1, start=list(p3=0.5, p4=0.5), fix.arg=list(p1=0, p2=1))

# (4) estimation of the asymmetry-steepness parameters of the BMT
# distribution with domain fixed at [0,1]
BMTfit.mpse(x1, start=list(p3=0, p4=0.5), type.p.3.4 = "a-s",
            fix.arg=list(p1=0, p2=1))
```

`BMTfit.mqde` *Minimum Quantile Distance Fit of the BMT Distribution to Non-censored Data.*

Description

Fit of the BMT distribution to non-censored data by minimum quantile distance (mqde), which can also be called maximum quantile goodness-of-fit.

Usage

```
BMTfit.mqde(
  data,
  probs = (1:length(data) - 0.5)/length(data),
  qtype = 5,
  dist = "euclidean",
  start = list(p3 = 0.5, p4 = 0.5, p1 = min(data) - 0.1, p2 = max(data) + 0.1),
  fix.arg = NULL,
  type.p.3.4 = "t w",
  type.p.1.2 = "c-d",
  optim.method = "Nelder-Mead",
  custom.optim = NULL,
  weights = NULL,
  silent = TRUE,
  ...
)
```

Arguments

`data`	A numeric vector with the observed values for non-censored data.
`probs`	A numeric vector of the probabilities for which the minimum quantile distance estimation is done. $p[k] = (k - 0.5)/n$ (default).
`qtype`	The quantile type used by the R `quantile` function to compute the empirical quantiles. Type 5 (default), i.e. $x[k]$ is both the kth order statistic and the type 5 sample quantile of $p[k] = (k - 0.5)/n$.
`dist`	The distance measure between observed and theoretical quantiles to be used. This must be one of "euclidean" (default), "maximum", or "manhattan". Any unambiguous substring can be given.
`start`	A named list giving the initial values of parameters of the BMT distribution or a function of data computing initial values and returning a named list. (see the 'details' section of `mledist`).
`fix.arg`	An optional named list giving the values of fixed parameters of the BMT distribution or a function of data computing (fixed) parameter values and returning a named list. Parameters with fixed value are thus NOT estimated. (see the 'details' section of `mledist`).
`type.p.3.4`	Type of parametrization associated to p3 and p4. "t w" means tails weights parametrization (default) and "a-s" means asymmetry-steepness parametrization.
`type.p.1.2`	Type of parametrization associated to p1 and p2. "c-d" means domain parametrization (default) and "l-s" means location-scale parametrization.
`optim.method`	`"default"` (see the 'details' section of `mledist`) or optimization method to pass to `optim`. Given the close-form expression of the quantile function, two optimization methods were added when the euclidean distance is selected: Coordinate descend (`"CD"`) and Newton-Rhapson (`"NR"`).
`custom.optim`	A function carrying the optimization (see the 'details' section of `mledist`).
`weights`	an optional vector of weights to be used in the fitting process. Should be `NULL` or a numeric vector with strictly positive numbers. If non-`NULL`, weighted mqde is used, otherwise ordinary mqde.
`silent`	A logical to remove or show warnings when bootstrapping.
`...`	Further arguments to be passed to generic functions or to the function `"mqdedist"`. See `mqdedist` for details.

Details

This function is not intended to be called directly but is internally called in `BMTfit` when used with the minimum quantile distance method.

`BMTfit.mqde` is based on the function `mqdedist` but it focuses on the minimum quantile distance parameter estimation for the BMT distribution (see `BMT` for details about the BMT distribution and `mqdedist` for details about minimum quantile distance fit of univariate distributions).

Given the close-form expression of the quantile function, two optimization methods were added when the euclidean distance is selected: Coordinate descend (`"CD"`) and Newton-Rhapson (`"NR"`).

Value

`BMTfit.mqde` returns a list with following components,

`estimate`	the parameter estimates.
`convergence`	an integer code for the convergence of `optim`/`constrOptim` defined as below or defined by the user in the user-supplied optimization function. `0` indicates successful convergence. `1` indicates that the iteration limit of `optim` has been reached. `10` indicates degeneracy of the Nealder-Mead simplex. `100` indicates that `optim` encountered an internal error.
`value`	the value of the corresponding objective function of the estimation method at the estimate.
`hessian`	a symmetric matrix computed by `optim` as an estimate of the Hessian at the solution found or computed in the user-supplied optimization function.
`loglik`	the log-likelihood value.
`probs`	the probability vector on which observed and theoretical quantiles were calculated.
`dist`	the name of the distance between observed and theoretical quantiles used.
`optim.function`	the name of the optimization function used for maximum product of spacing.
`optim.method`	when `optim` is used, the name of the algorithm used, `NULL` otherwise.
`fix.arg`	the named list giving the values of parameters of the named distribution that must kept fixed rather than estimated or `NULL` if there are no such parameters.
`fix.arg.fun`	the function used to set the value of `fix.arg` or `NULL`.
`weights`	the vector of weights used in the estimation process or `NULL`.
`counts`	A two-element integer vector giving the number of calls to the log-likelihood function and its gradient respectively. This excludes those calls needed to compute the Hessian, if requested, and any calls to log-likelihood function to compute a finite-difference approximation to the gradient. `counts` is returned by `optim` or the user-supplied function or set to `NULL`.
`optim.message`	A character string giving any additional information returned by the optimizer, or `NULL`. To understand exactly the message, see the source code.

Author(s)

Camilo Jose Torres-Jimenez [aut,cre] `<cjtorresj@unal.edu.co>`

Source

Based on the function `mqdedist` which in turn is based on the function `mledist` of the R package: **fitdistrplus**

Delignette-Muller ML and Dutang C (2015), *fitdistrplus: An R Package for Fitting Distributions*. Journal of Statistical Software, 64(4), 1-34.

References

Torres-Jimenez, C. J. (2017, September), *Comparison of estimation methods for the BMT distribution*. ArXiv e-prints.

Torres-Jimenez, C. J. (2018), *The BMT Item Response Theory model: A new skewed distribution family with bounded domain and an IRT model based on it*, PhD thesis, Doctorado en ciencias - Estadistica, Universidad Nacional de Colombia, Sede Bogota.

See Also

See `BMT` for the BMT density, distribution, quantile function and random deviates. See `BMTfit.mme`, `BMTfit.mle`, `BMTfit.mge`, `BMTfit.mpse` and `BMTfit.qme` for other estimation methods. See `optim` and `constrOptim` for optimization routines. See `BMTfit` and `fitdist` for functions that return an object of class `"fitdist"`.

Examples

```
# (1) basic fit by minimum quantile distance estimation
set.seed(1234)
x1 <- rBMT(n=100, p3=0.25, p4=0.75)
BMTfit.mqde(x1)

# (2) quantile matching is a particular case of minimum quantile distance
BMTfit.mqde(x1, probs=c(0.2,0.4,0.6,0.8), qtype=7)

# (3) maximum or manhattan instead of euclidean distance
BMTfit.mqde(x1, dist="maximum")
BMTfit.mqde(x1, dist="manhattan")

# (4) how to change the optimisation method?
BMTfit.mqde(x1, optim.method="L-BFGS-B")
BMTfit.mqde(x1, custom.optim="nlminb")

# (5) estimation of the tails weights parameters of the BMT
# distribution with domain fixed at [0,1]
BMTfit.mqde(x1, start=list(p3=0.5, p4=0.5), fix.arg=list(p1=0, p2=1))

# (6) estimation of the asymmetry-steepness parameters of the BMT
# distribution with domain fixed at [0,1]
BMTfit.mqde(x1, start=list(p3=0, p4=0.5), type.p.3.4 = "a-s",
            fix.arg=list(p1=0, p2=1))
```

`BMTfit.qme` *Quantile Matching Fit of the BMT Distribution to Non-censored Data.*

Description

Fit of the BMT distribution to non-censored data by quantile matching estimation (qme).

Usage

```
BMTfit.qme(
  data,
  probs = c(0.2, 0.4, 0.6, 0.8),
  qtype = 7,
  start = list(p3 = 0.5, p4 = 0.5, p1 = min(data) - 0.1, p2 = max(data) + 0.1),
  fix.arg = NULL,
  type.p.3.4 = ”t w”,
  type.p.1.2 = ”c-d”,
  optim.method = ”Nelder-Mead”,
  custom.optim = NULL,
  silent = TRUE,
  ...
)
```

Arguments

`data`	A numeric vector with the observed values for non-censored data.
`probs`	A numeric vector of the probabilities for which the quantile matching is done. The length of this vector must be equal to the number of parameters to estimate.
`qtype`	The quantile type used by the R `quantile` function to compute the empirical quantiles, (default 7 corresponds to the default quantile method in R).
`start`	A named list giving the initial values of parameters of the BMT distribution or a function of data computing initial values and returning a named list. (see the ’details’ section of `mledist`).
`fix.arg`	An optional named list giving the values of fixed parameters of the BMT distribution or a function of data computing (fixed) parameter values and returning a named list. Parameters with fixed value are thus NOT estimated. (see the ’details’ section of `mledist`).
`type.p.3.4`	Type of parametrization associated to p3 and p4. "t w" means tails weights parametrization (default) and "a-s" means asymmetry-steepness parametrization.
`type.p.1.2`	Type of parametrization associated to p1 and p2. "c-d" means domain parametrization (default) and "l-s" means location-scale parametrization.
`optim.method`	`"default"` (see the ’details’ section of `mledist`) or optimization method to pass to `optim`.
`custom.optim`	A function carrying the optimization (see the ’details’ section of `mledist`).
`silent`	A logical to remove or show warnings when bootstrapping.
`...`	Further arguments to be passed to generic functions or to the function `"qmedist"`. See `qmedist` for details.

Details

This function is not intended to be called directly but is internally called in `BMTfit` when used with the quantile matching method.

`BMTfit.qme` is based on the function `qmedist` but it focuses on the quantile matching parameter estimation for the BMT distribution (see `BMT` for details about the BMT distribution and `qmedist` for details about quantile matching fit of univariate distributions).

Value

`BMTfit.qme` returns a list with following components,

`estimate`	the parameter estimates.
`convergence`	an integer code for the convergence of `optim`/`constrOptim` defined as below or defined by the user in the user-supplied optimization function. `0` indicates successful convergence. `1` indicates that the iteration limit of `optim` has been reached. `10` indicates degeneracy of the Nealder-Mead simplex. `100` indicates that `optim` encountered an internal error.
`value`	the value of the corresponding objective function of the estimation method at the estimate.
`hessian`	a symmetric matrix computed by `optim` as an estimate of the Hessian at the solution found or computed in the user-supplied optimization function.
`loglik`	the log-likelihood value.
`probs`	the probability vector on which quantiles are matched.
`optim.function`	the name of the optimization function used for maximum product of spacing.
`optim.method`	when `optim` is used, the name of the algorithm used, `NULL` otherwise.
`fix.arg`	the named list giving the values of parameters of the named distribution that must kept fixed rather than estimated or `NULL` if there are no such parameters.
`fix.arg.fun`	the function used to set the value of `fix.arg` or `NULL`.
`weights`	the vector of weights used in the estimation process or `NULL`.
`counts`	A two-element integer vector giving the number of calls to the log-likelihood function and its gradient respectively. This excludes those calls needed to compute the Hessian, if requested, and any calls to log-likelihood function to compute a finite-difference approximation to the gradient. `counts` is returned by `optim` or the user-supplied function or set to `NULL`.
`optim.message`	A character string giving any additional information returned by the optimizer, or `NULL`. To understand exactly the message, see the source code.

Author(s)

Camilo Jose Torres-Jimenez [aut,cre] `<cjtorresj@unal.edu.co>`

Source

Based on the function `qmedist` of the R package: **fitdistrplus**

Delignette-Muller ML and Dutang C (2015), *fitdistrplus: An R Package for Fitting Distributions*. Journal of Statistical Software, 64(4), 1-34.

References

Torres-Jimenez, C. J. (2017, September), *Comparison of estimation methods for the BMT distribution*. ArXiv e-prints.

Torres-Jimenez, C. J. (2018), *The BMT Item Response Theory model: A new skewed distribution family with bounded domain and an IRT model based on it*, PhD thesis, Doctorado en ciencias - Estadistica, Universidad Nacional de Colombia, Sede Bogota.

See Also

See `BMT` for the BMT density, distribution, quantile function and random deviates. See `BMTfit.mme`, `BMTfit.mle`, `BMTfit.mge`, `BMTfit.mpse` and `BMTfit.mqde` for other estimation methods. See `optim` and `constrOptim` for optimization routines. See `BMTfit` and `fitdist` for functions that return an object of class `"fitdist"`.

Examples

```
# (1) basic fit by quantile matching estimation
set.seed(1234)
x1 <- rBMT(n=50, p3 = 0.25, p4 = 0.75)
BMTfit.qme(x1)

# (2) changing the probability vector on which quantiles are matched
BMTfit.qme(x1, probs=c(0.1,0.3,0.5,0.75))

# (3) how to change the optimisation method?
BMTfit.qme(x1, optim.method="L-BFGS-B")
BMTfit.qme(x1, custom.optim="nlminb")

# (4) estimation of the tails weights parameters of the BMT
# distribution with domain fixed at [0,1]
BMTfit.qme(x1, start=list(p3=0.5, p4=0.5),
           fix.arg=list(p1=0, p2=1), probs=c(1/3,2/3))

# (5) estimation of the asymmetry-steepness parameters of the BMT
# distribution with domain fixed at [0,1]
BMTfit.qme(x1, start=list(p3=0, p4=0.5), type.p.3.4 = "a-s",
           fix.arg=list(p1=0, p2=1), probs=c(1/3,2/3))
```

`BMTkurtosis` *The BMT Distribution Descriptive Measures - Kurtosis.*

Description

Kurtosis and steepness coefficient for the BMT distribution with `p3` and `p4` tails weights (κ_l and κ_r) or asymmetry-steepness parameters (ζ and ξ) and `p1` and `p2` domain (minimum and maximum) or location-scale (mean and standard deviation) parameters.

Usage

```
BMTkurt(p3, p4, type.p.3.4 = "t w", p1 = 0, p2 = 1, type.p.1.2 = "c-d")

BMTsteep(p3, p4, type.p.3.4 = "t w", p1 = 0, p2 = 1, type.p.1.2 = "c-d")
```

Arguments

`p3, p4`	tails weights (κ_l and κ_r) or asymmetry-steepness (ζ and ξ) parameters of the BMT distribution.
`type.p.3.4`	type of parametrization associated to p3 and p4. "t w" means tails weights parametrization (default) and "a-s" means asymmetry-steepness parametrization.
`p1, p2`	domain (minimum and maximum) or location-scale (mean and standard deviation) parameters of the BMT distribution.
`type.p.1.2`	type of parametrization associated to p1 and p2. "c-d" means domain parametrization (default) and "l-s" means location-scale parametrization.

Details

See References.

Value

`BMTkurt` gives the Pearson's kurtosis and `BMTsteep` the proposed steepness coefficient for the BMT distribution.

The arguments are recycled to the length of the result. Only the first elements of `type.p.3.4` and `type.p.1.2` are used.

If `type.p.3.4 == "t w"`, `p3 < 0` and `p3 > 1` are errors and return `NaN`.

If `type.p.3.4 == "a-s"`, `p3 < -1` and `p3 > 1` are errors and return `NaN`.

`p4 < 0` and `p4 > 1` are errors and return `NaN`.

If `type.p.1.2 == "c-d"`, `p1 >= p2` is an error and returns `NaN`.

If `type.p.1.2 == "l-s"`, `p2 <= 0` is an error and returns `NaN`.

Author(s)

Camilo Jose Torres-Jimenez [aut,cre] <`cjtorresj@unal.edu.co`>

References

Torres-Jimenez, C. J. and Montenegro-Diaz, A. M. (2017, September), *An alternative to continuous univariate distributions supported on a bounded interval: The BMT distribution*. ArXiv e-prints. https://arxiv.org/abs/1709.05534.

Torres-Jimenez, C. J. (2018), *The BMT Item Response Theory model: A new skewed distribution family with bounded domain and an IRT model based on it*, PhD thesis, Doctorado en ciencias - Estadistica, Universidad Nacional de Colombia, Sede Bogota.

See Also

`BMTcentral`, `BMTdispersion`, `BMTskewness`, `BMTmoments` for other descriptive measures or moments.

Examples

```
# BMT on [0,1] with left tail weight equal to 0.25 and
# right tail weight equal to 0.75
BMTkurt(0.25, 0.75, "t w")
BMTsteep(0.25, 0.75, "t w")

# BMT on [0,1] with asymmetry coefficient equal to 0.5 and
# steepness coefficient equal to 0.75
BMTkurt(0.5, 0.5, "a-s")
BMTsteep(0.5, 0.5, "a-s")

# domain or location-scale parameters do not affect
# the skewness and the asymmetry coefficient

# BMT on [-1.783489,3.312195] with
# left tail weight equal to 0.25 and
# right tail weight equal to 0.75
BMTkurt(0.25, 0.75, "t w", -1.783489, 3.312195, "c-d")
BMTsteep(0.25, 0.75, "t w", -1.783489, 3.312195, "c-d")

# BMT with mean equal to 0, standard deviation equal to 1,
# asymmetry coefficient equal to 0.5 and
# steepness coefficient equal to 0.75
BMTkurt(0.5, 0.5, "a-s", 0, 1, "l-s")
BMTsteep(0.5, 0.5, "a-s", 0, 1, "l-s")
```

`BMTmoments` *The BMT Distribution Moments, Moment-Generating Function and Characteristic Function.*

Description

Any raw, central or standardized moment, the moment-generating function and the characteristic function for the BMT distribution, with `p3` and `p4` tails weights (κ_l and κ_r) or asymmetry-steepness parameters (ζ and ξ) and `p1` and `p2` domain (minimum and maximum) or location-scale (mean and standard deviation) parameters.

Usage

```
BMTmoment(
  p3,
  p4,
  type.p.3.4 = "t w",
  p1 = 0,
```

```
  p2 = 1,
  type.p.1.2 = "c-d",
  order,
  type = "standardised",
  method = "quadrature"
)

BMTmgf(s, p3, p4, type.p.3.4 = "t w", p1 = 0, p2 = 1, type.p.1.2 = "c-d")

BMTchf(s, p3, p4, type.p.3.4 = "t w", p1 = 0, p2 = 1, type.p.1.2 = "c-d")

mBMT(order, p3, p4, type.p.3.4, p1, p2, type.p.1.2)
```

Arguments

`p3, p4`	tails weights (κ_l and κ_r) or asymmetry-steepness (ζ and ξ) parameters of the BMT distribution.
`type.p.3.4`	type of parametrization associated to p3 and p4. "t w" means tails weights parametrization (default) and "a-s" means asymmetry-steepness parametrization.
`p1, p2`	domain (minimum and maximum) or location-scale (mean and standard deviation) parameters of the BMT distribution.
`type.p.1.2`	type of parametrization associated to p1 and p2. "c-d" means domain parametrization (default) and "l-s" means location-scale parametrization.
`order`	order of the moment.
`type`	type of the moment: raw, central or standardized (default).
`method`	method to obtain the moment: exact formula or Chebyshev-Gauss quadrature (default).
`s`	variable for the moment-generating and characteristic functions.

Details

See References.

Value

`BMTmoment` gives any raw, central or standardized moment, `BMTmgf` the moment-generating function and `BMTchf` the characteristic function

The arguments are recycled to the length of the result. Only the first elements of `type.p.3.4`, `type.p.1.2`, `type` and `method` are used.

If `type.p.3.4 == "t w"`, `p3 < 0` and `p3 > 1` are errors and return `NaN`.

If `type.p.3.4 == "a-s"`, `p3 < -1` and `p3 > 1` are errors and return `NaN`.

`p4 < 0` and `p4 > 1` are errors and return `NaN`.

If `type.p.1.2 == "c-d"`, `p1 >= p2` is an error and returns `NaN`.

If `type.p.1.2 == "l-s"`, `p2 <= 0` is an error and returns `NaN`.

Author(s)

Camilo Jose Torres-Jimenez [aut,cre] <`cjtorresj@unal.edu.co`>

References

Torres-Jimenez, C. J. and Montenegro-Diaz, A. M. (2017, September), *An alternative to continuous univariate distributions supported on a bounded interval: The BMT distribution*. ArXiv e-prints. `https://arxiv.org/abs/1709.05534`.

Torres-Jimenez, C. J. (2018), *The BMT Item Response Theory model: A new skewed distribution family with bounded domain and an IRT model based on it*, PhD thesis, Doctorado en ciencias - Estadistica, Universidad Nacional de Colombia, Sede Bogota.

See Also

`BMTcentral`, `BMTdispersion`, `BMTskewness`, `BMTkurtosis` for specific descriptive measures or moments.

Examples

```
layout(matrix(1:4, 2, 2, TRUE))
s <- seq(-1, 1, length.out = 100)

# BMT on [0,1] with left tail weight equal to 0.25 and
# right tail weight equal to 0.75
BMTmoment(0.25, 0.75, order = 5) # hyperskewness by Gauss-Legendre quadrature
BMTmoment(0.25, 0.75, order = 5, method = "exact") # hyperskewness by exact formula
mgf <- BMTmgf(s, 0.25, 0.75) # moment-generation function
plot(s, mgf, type="l")
chf <- BMTchf(s, 0.25, 0.75) # characteristic function

# BMT on [0,1] with asymmetry coefficient equal to 0.5 and
# steepness coefficient equal to 0.5
BMTmoment(0.5, 0.5, "a-s", order = 5)
BMTmoment(0.5, 0.5, "a-s", order = 5, method = "exact")
mgf <- BMTmgf(s, 0.5, 0.5, "a-s")
plot(s, mgf, type="l")
chf <- BMTchf(s, 0.5, 0.5, "a-s")

# BMT on [-1.783489, 3.312195] with
# left tail weight equal to 0.25 and
# right tail weight equal to 0.75
BMTmoment(0.25, 0.75, "t w", -1.783489, 3.312195, "c-d", order = 5)
BMTmoment(0.25, 0.75, "t w", -1.783489, 3.312195, "c-d", order = 5, method = "exact")
mgf <- BMTmgf(s, 0.25, 0.75, "t w", -1.783489, 3.312195, "c-d")
plot(s, mgf, type="l")
chf <- BMTchf(s, 0.25, 0.75, "t w", -1.783489, 3.312195, "c-d")

# BMT with mean equal to 0, standard deviation equal to 1,
# asymmetry coefficient equal to 0.5 and
# steepness coefficient equal to 0.5
BMTmoment(0.5, 0.5, "a-s", 0, 1, "l-s", order = 5)
```

```
BMTmoment(0.5, 0.5, "a-s", 0, 1, "l-s", order = 5, method = "exact")
mgf <- BMTmgf(s, 0.5, 0.5, "a-s", 0, 1, "l-s")
plot(s, mgf, type="l")
chf <- BMTchf(s, 0.5, 0.5, "a-s", 0, 1, "l-s")
```

`BMTskewness` *The BMT Distribution Descriptive Measures - Skewness.*

Description

Skewness and an asymmetry coefficient for the BMT distribution, with `p3` and `p4` tails weights (κ_l and κ_r) or asymmetry-steepness parameters (ζ and ξ) and `p1` and `p2` domain (minimum and maximum) or location-scale (mean and standard deviation) parameters.

Usage

```
BMTskew(p3, p4, type.p.3.4 = "t w", p1 = 0, p2 = 1, type.p.1.2 = "c-d")

BMTasymm(p3, p4, type.p.3.4 = "t w", p1 = 0, p2 = 1, type.p.1.2 = "c-d")
```

Arguments

`p3, p4`	tails weights (κ_l and κ_r) or asymmetry-steepness (ζ and ξ) parameters of the BMT distribution.
`type.p.3.4`	type of parametrization associated to p3 and p4. "t w" means tails weights parametrization (default) and "a-s" means asymmetry-steepness parametrization.
`p1, p2`	domain (minimum and maximum) or location-scale (mean and standard deviation) parameters of the BMT distribution.
`type.p.1.2`	type of parametrization associated to p1 and p2. "c-d" means domain parametrization (default) and "l-s" means location-scale parametrization.

Details

See References.

Value

`BMTskew` gives the Pearson's skewness and `BMTasymm` the proposed asymmetry coefficient for the BMT distribution.

The arguments are recycled to the length of the result. Only the first elements of `type.p.3.4` and `type.p.1.2` are used.

If `type.p.3.4 == "t w"`, `p3 < 0` and `p3 > 1` are errors and return `NaN`.

If `type.p.3.4 == "a-s"`, `p3 < -1` and `p3 > 1` are errors and return `NaN`.

`p4 < 0` and `p4 > 1` are errors and return `NaN`.

If `type.p.1.2 == "c-d"`, `p1 >= p2` is an error and returns `NaN`.

If `type.p.1.2 == "l-s"`, `p2 <= 0` is an error and returns `NaN`.

Author(s)

Camilo Jose Torres-Jimenez [aut,cre] <`cjtorresj@unal.edu.co`>

References

Torres-Jimenez, C. J. and Montenegro-Diaz, A. M. (2017, September), *An alternative to continuous univariate distributions supported on a bounded interval: The BMT distribution*. ArXiv e-prints. `https://arxiv.org/abs/1709.05534`.

Torres-Jimenez, C. J. (2018), *The BMT Item Response Theory model: A new skewed distribution family with bounded domain and an IRT model based on it*, PhD thesis, Doctorado en ciencias - Estadistica, Universidad Nacional de Colombia, Sede Bogota.

See Also

`BMTcentral`, `BMTdispersion`, `BMTkurtosis`, `BMTmoments` for other descriptive measures or moments.

Examples

```
# BMT on [0,1] with left tail weight equal to 0.25 and
# right tail weight equal to 0.75
BMTskew(0.25, 0.75, "t w")
BMTasymm(0.25, 0.75, "t w")

# BMT on [0,1] with asymmetry coefficient equal to 0.5 and
# steepness coefficient equal to 0.75
BMTskew(0.5, 0.5, "a-s")
BMTasymm(0.5, 0.5, "a-s")

# domain or location-scale parameters do not affect
# the skewness and the asymmetry coefficient

# BMT on [-1.783489,3.312195] with
# left tail weight equal to 0.25 and
# right tail weight equal to 0.75
BMTskew(0.25, 0.75, "t w", -1.783489, 3.312195, "c-d")
BMTasymm(0.25, 0.75, "t w", -1.783489, 3.312195, "c-d")

# BMT with mean equal to 0, standard deviation equal to 1,
# asymmetry coefficient equal to 0.5 and
# steepness coefficient equal to 0.75
BMTskew(0.5, 0.5, "a-s", 0, 1, "l-s")
BMTasymm(0.5, 0.5, "a-s", 0, 1, "l-s")
```

`mpsedist` *Maximum Product of Spacing Fit of Univariate Distributions.*

Description

Fit of univariate distributions for non-censored data using maximum product of spacing estimation (mpse), also called maximum spacing estimation.

Usage

```
mpsedist(
  data,
  distr,
  start = NULL,
  fix.arg = NULL,
  optim.method = "default",
  lower = -Inf,
  upper = Inf,
  custom.optim = NULL,
  weights = NULL,
  silent = TRUE,
  gradient = NULL,
  ...
)
```

Arguments

`data`	A numeric vector with the observed values for non-censored data.
`distr`	A character string `"name"` naming a distribution for which the corresponding density function `dname` and the corresponding distribution function `pname` must be classically defined.
`start`	A named list giving the initial values of parameters of the named distribution or a function of data computing initial values and returning a named list. This argument may be omitted (default) for some distributions for which reasonable starting values are computed (see the 'details' section of `mledist`).
`fix.arg`	An optional named list giving the values of fixed parameters of the named distribution or a function of data computing (fixed) parameter values and returning a named list. Parameters with fixed value are thus NOT estimated.
`optim.method`	`"default"` (see details) or an optimization method to pass to `optim`.
`lower`	Left bounds on the parameters for the `"L-BFGS-B"` method (see `optim`) or the `constrOptim` function (as an equivalent linear constraint).
`upper`	Right bounds on the parameters for the `"L-BFGS-B"` method (see `optim`) or the `constrOptim` function (as an equivalent linear constraint).
`custom.optim`	A function carrying the optimization (see details).

`weights`	An optional vector of weights to be used in the fitting process. Should be `NULL` or a numeric vector with strictly positive numbers. If non-`NULL`, weighted mpse is used, otherwise ordinary mpse.
`silent`	A logical to remove or show warnings when bootstrapping.
`gradient`	A function to return the gradient of the optimization objective function for the `"BFGS"`, `"CG"` and `"L-BFGS-B"` methods. If it is `NULL`, a finite-difference approximation will be used, see `optim`.
`...`	Further arguments passed to the `optim`, `constrOptim` or `custom.optim` function.

Details

The `mpsedist` function carries out the maximum product of spacing estimation numerically, by maximization of the arithmetic mean of $\log(F(k) - F(k-1))$.

The optimization process is the same as `mledist`, see the 'details' section of that function.

Optionally, a vector of `weights` can be used in the fitting process. By default (when `weights=NULL`), ordinary mpse is carried out, otherwise the specified weights are used to compute a weighted arithmetic mean.

We believe this function should be added to the package **fitdistrplus**. Until it is accepted and incorporated into that package, it will remain in the package **BMT**. This function is internally called in `BMTfit.mpse`.

Value

`mpsedist` returns a list with following components,

`estimate`	the parameter estimates.
`convergence`	an integer code for the convergence of `optim` defined as below or defined by the user in the user-supplied optimization function. `0` indicates successful convergence. `1` indicates that the iteration limit of `optim` has been reached. `10` indicates degeneracy of the Nealder-Mead simplex. `100` indicates that `optim` encountered an internal error.
`value`	the value of the optimization objective function at the solution found.
`loglik`	the log-likelihood.
`hessian`	a symmetric matrix computed by `optim` as an estimate of the Hessian at the solution found or computed in the user-supplied optimization function.
`optim.function`	the name of the optimization function used.
`fix.arg`	the named list giving the values of parameters of the named distribution that must kept fixed rather than estimated by maximum likelihood or `NULL` if there are no such parameters.
`optim.method`	when `optim` is used, the name of the algorithm used, `NULL` otherwise.
`fix.arg.fun`	the function used to set the value of `fix.arg` or `NULL`.
`weights`	the vector of weights used in the estimation process or `NULL`.

`counts`	A two-element integer vector giving the number of calls to the log-likelihood function and its gradient respectively. This excludes those calls needed to compute the Hessian, if requested, and any calls to log-likelihood function to compute a finite-difference approximation to the gradient. `counts` is returned by `optim` or the user-supplied optimization function, or set to `NULL`.
`optim.message`	A character string giving any additional information returned by the optimizer, or `NULL`. To understand exactly the message, see the source code.

Author(s)

Camilo Jose Torres-Jimenez [aut,cre] `<cjtorresj@unal.edu.co>`

Source

Based on the function mledist of the R package: **fitdistrplus**

Delignette-Muller ML and Dutang C (2015), *fitdistrplus: An R Package for Fitting Distributions*. Journal of Statistical Software, 64(4), 1-34.

Functions `checkparam` and `startargdefault` are needed and were copied from the same package (fitdistrplus version: 1.0-9).

References

Cheng, R. and N. Amin (1983). *Estimating parameters in continuous univariate distributions with a shifted origin*. Journal of the Royal Statistical Society. Series B (Methodological), 394-403.

Ranneby, B. (1984). *The maximum spacing method. an estimation method related to the maximum likelihood method*. Scandinavian Journal of Statistics, 93-112.

See Also

`mqdedist`, `mledist`, `mmedist`, `qmedist`, `mgedist`, and `optim`.

Examples

```
# (1) basic fit of a normal distribution
set.seed(1234)
x1 <- rnorm(n = 100)
mean(x1); sd(x1)
mpse1 <- mpsedist(x1, "norm")
mpse1$estimate

# (2) defining your own distribution functions, here for the Gumbel
# distribution for other distributions, see the CRAN task view dedicated
# to probability distributions
dgumbel <- function(x, a, b) 1/b*exp((a-x)/b)*exp(-exp((a-x)/b))
pgumbel <- function(q, a, b) exp(-exp((a-q)/b))
qgumbel <- function(p, a, b) a-b*log(-log(p))
mpse1 <- mpsedist(x1, "gumbel", start = list(a = 10, b = 5))
mpse1$estimate

# (3) fit a discrete distribution (Poisson)
```

```
set.seed(1234)
x2 <- rpois(n = 30, lambda = 2)
mpse2 <- mpsedist(x2, "pois")
mpse2$estimate

# (4) fit a finite-support distribution (beta)
set.seed(1234)
x3 <- rbeta(n = 100, shape1 = 5, shape2 = 10)
mpse3 <- mpsedist(x3, "beta")
mpse3$estimate

# (5) fit frequency distributions on USArrests dataset.
x4 <- USArrests$Assault
mpse4pois <- mpsedist(x4, "pois")
mpse4pois$estimate
mpse4nbinom <- mpsedist(x4, "nbinom")
mpse4nbinom$estimate

# (6) weighted fit of a normal distribution
set.seed(1234)
w1 <- runif(101)
mpse1 <- mpsedist(x1, "norm", weights = w1)
mpse1$estimate
```

`mqdedist` *Minimum Quantile Distance Fit of Univariate Distributions.*

Description

Fit of univariate distributions for non-censored data using minimum quantile distance estimation (mqde), which can also be called maximum quantile goodness-of-fit estimation.

Usage

```
mqdedist(
  data,
  distr,
  probs = (1:length(data) - 0.5)/length(data),
  qtype = 5,
  dist = "euclidean",
  start = NULL,
  fix.arg = NULL,
  optim.method = "default",
  lower = -Inf,
  upper = Inf,
  custom.optim = NULL,
  weights = NULL,
  silent = TRUE,
```

```
  gradient = NULL,
  ...
)
```

Arguments

`data`	A numeric vector with the observed values for non-censored data.
`distr`	A character string `"name"` naming a distribution for which the corresponding quantile function `qname` and the corresponding density distribution `dname` must be classically defined.
`probs`	A numeric vector of the probabilities for which the minimum quantile distance estimation is done. $p[k] = (k - 0.5)/n$ (default).
`qtype`	The quantile type used by the R `quantile` function to compute the empirical quantiles. Type 5 (default), i.e. $x[k]$ is both the kth order statistic and the type 5 sample quantile of $p[k] = (k - 0.5)/n$.
`dist`	The distance measure between observed and theoretical quantiles to be used. This must be one of "euclidean" (default), "maximum", or "manhattan". Any unambiguous substring can be given.
`start`	A named list giving the initial values of parameters of the named distribution or a function of data computing initial values and returning a named list. This argument may be omitted (default) for some distributions for which reasonable starting values are computed (see the 'details' section of `mledist`).
`fix.arg`	An optional named list giving the values of fixed parameters of the named distribution or a function of data computing (fixed) parameter values and returning a named list. Parameters with fixed value are thus NOT estimated.
`optim.method`	`"default"` (see details) or optimization method to pass to `optim`.
`lower`	Left bounds on the parameters for the `"L-BFGS-B"` method (see `optim`) or the `constrOptim` function (as an equivalent linear constraint).
`upper`	Right bounds on the parameters for the `"L-BFGS-B"` method (see `optim`) or the `constrOptim` function (as an equivalent linear constraint).
`custom.optim`	A function carrying the optimization (see details).
`weights`	An optional vector of weights to be used in the fitting process. Should be `NULL` or a numeric vector with strictly positive numbers. If non-`NULL`, weighted mqde is used, otherwise ordinary mqde.
`silent`	A logical to remove or show warnings when bootstrapping.
`gradient`	A function to return the gradient of the optimization objective function for the `"BFGS"`, `"CG"` and `"L-BFGS-B"` methods. If it is `NULL`, a finite-difference approximation will be used, see `optim`.
`...`	Further arguments passed to the `optim`, `constrOptim` or `custom.optim` function.

Details

The `mqdedist` function carries out the minimum quantile distance estimation numerically, by minimization of a distance between observed and theoretical quantiles.

The optimization process is the same as `mledist`, see the 'details' section of that function.

Optionally, a vector of `weights` can be used in the fitting process. By default (when `weights=NULL`), ordinary mqde is carried out, otherwise the specified weights are used to compute a weighted distance.

We believe this function should be added to the package **fitdistrplus**. Until it is accepted and incorporated into that package, it will remain in the package **BMT**. This function is internally called in `BMTfit.mqde`.

Value

`mqdedist` returns a list with following components,

`estimate`	the parameter estimates.
`convergence`	an integer code for the convergence of `optim` defined as below or defined by the user in the user-supplied optimization function. `0` indicates successful convergence. `1` indicates that the iteration limit of `optim` has been reached. `10` indicates degeneracy of the Nealder-Mead simplex. `100` indicates that `optim` encountered an internal error.
`value`	the value of the optimization objective function at the solution found.
`hessian`	a symmetric matrix computed by `optim` as an estimate of the Hessian at the solution found or computed in the user-supplied optimization function.
`probs`	the probability vector on which observed and theoretical quantiles were calculated.
`dist`	the name of the distance between observed and theoretical quantiles used.
`optim.function`	the name of the optimization function used.
`fix.arg`	the named list giving the values of parameters of the named distribution that must kept fixed rather than estimated by maximum likelihood or `NULL` if there are no such parameters.
`loglik`	the log-likelihood.
`optim.method`	when `optim` is used, the name of the algorithm used, `NULL` otherwise.
`fix.arg.fun`	the function used to set the value of `fix.arg` or `NULL`.
`weights`	the vector of weights used in the estimation process or `NULL`.
`counts`	A two-element integer vector giving the number of calls to the log-likelihood function and its gradient respectively. This excludes those calls needed to compute the Hessian, if requested, and any calls to log-likelihood function to compute a finite-difference approximation to the gradient. `counts` is returned by `optim` or the user-supplied optimization function, or set to `NULL`.
`optim.message`	A character string giving any additional information returned by the optimizer, or `NULL`. To understand exactly the message, see the source code.

Author(s)

Camilo Jose Torres-Jimenez [aut,cre] `<cjtorresj@unal.edu.co>`

Source

Based on the function mledist of the R package: **fitdistrplus**

Delignette-Muller ML and Dutang C (2015), *fitdistrplus: An R Package for Fitting Distributions*. Journal of Statistical Software, 64(4), 1-34.

Functions `checkparam` and `startargdefault` are needed and were copied from the same package (fitdistrplus version: 1.0-9).

References

LaRiccia, V. N. (1982). *Asymptotic Properties of Weighted L^2 Quantile Distance Estimators*. The Annals of Statistics, 10(2), 621-624.

Torres-Jimenez, C. J. (2017, September), *Comparison of estimation methods for the BMT distribution*. ArXiv e-prints.

See Also

`mpsedist`, `mledist`, `mmedist`, `qmedist`, `mgedist`, `optim`, `constrOptim`, and `quantile`.

Examples

```
# (1) basic fit of a normal distribution
set.seed(1234)
x1 <- rnorm(n = 100)
mean(x1); sd(x1)
mqde1 <- mqdedist(x1, "norm")
mqde1$estimate

# (2) defining your own distribution functions, here for the Gumbel
# distribution for other distributions, see the CRAN task view dedicated
# to probability distributions
dgumbel <- function(x, a, b) 1/b*exp((a-x)/b)*exp(-exp((a-x)/b))
pgumbel <- function(q, a, b) exp(-exp((a-q)/b))
qgumbel <- function(p, a, b) a-b*log(-log(p))
mqde1 <- mqdedist(x1, "gumbel", start = list(a = 10, b = 5))
mqde1$estimate

# (3) fit a discrete distribution (Poisson)
set.seed(1234)
x2 <- rpois(n = 30, lambda = 2)
mqde2 <- mqdedist(x2, "pois")
mqde2$estimate

# (4) fit a finite-support distribution (beta)
set.seed(1234)
x3 <- rbeta(n = 100, shape1 = 5, shape2 = 10)
mqde3 <- mqdedist(x3, "beta")
mqde3$estimate

# (5) fit frequency distributions on USArrests dataset.
x4 <- USArrests$Assault
```

```
mqde4pois <- mqdedist(x4, "pois")
mqde4pois$estimate
mqde4nbinom <- mqdedist(x4, "nbinom")
mqde4nbinom$estimate

# (6) weighted fit of a normal distribution
set.seed(1234)
w1 <- runif(100)
weighted.mean(x1, w1)
mqde1 <- mqdedist(x1, "norm", weights = w1)
mqde1$estimate
```

`score.math.booklet10.PISA2012` *A Performance Score in Mathematics from Booklet 10 of PISA 2012.*

Description

Classic performance score in mathematics, using the students responses to Booklet 10 of the PISA test applied in 2012. We use the answers to the PISA 2012 questionnaire. We take the “Scored cognitive item response data file.”. We keep all the questions with binary score (correct and incorrect) and exclude those that could be scored with partial credit. Then, we recode the responses: 1 for correct and 0 for incorrect. Finally, we obtain the percentage of correct answers, i.e., the classic performance score of each student. It is important to mention that the reported scores of PISA 2012 are estimated and scaled using the Rasch model of item response theory. On the other hand, all booklets for the test could have different: number of questions, traits evaluated, and participating countries. Also, the assignation of a booklet to a student is randomized. Considering that, we choose only one arbitrary booklet, Booklet 10, and its questions of mathematics.

Usage

```
data(score.math.booklet10.PISA2012)
```

Format

`score.math.booklet10.PISA2012` is a vector.

Author(s)

Camilo Jose Torres-Jimenez [aut,cre] <`cjtorresj@unal.edu.co`>

Source

OECD Programme for International Student Assessment (PISA) (2012). Database - PISA 2012. Online; accessed 2014-08-23.

Index

∗ **datasets**
  score.math.booklet10.PISA2012, 50
∗ **distribution**
  BMT, 2
  BMTfit, 15
  BMTfit.mge, 19
  BMTfit.mle, 22
  BMTfit.mme, 24
  BMTfit.mpse, 27
  BMTfit.mqde, 30
  BMTfit.qme, 33
  mpsedist, 43
  mqdedist, 46

BMT, 2, *12*, *16*, *17*, *20*, *21*, *24*, *25*, *27–29*, *31*, *33*, *35*, *36*
BMT.Phi, 5
BMT.Psi, 7
BMTasymm (BMTskewness), 41
BMTcentral, *4*, 9, *14*, *38*, *40*, *42*
BMTchangepars, *4*, 11
BMTchf (BMTmoments), 38
BMTdispersion, *4*, *10*, 13, *38*, *40*, *42*
BMTfit, 15, *20*, *21*, *23–25*, *27–29*, *31*, *33*, *34*, *36*
BMTfit.mge, *16*, *17*, 19, *24*, *27*, *29*, *33*, *36*
BMTfit.mle, *16*, *17*, *21*, 22, *27*, *29*, *33*, *36*
BMTfit.mme, *16*, *17*, *21*, *24*, 24, *29*, *33*, *36*
BMTfit.mpse, *16*, *17*, *21*, *24*, *27*, 27, *33*, *36*, *44*
BMTfit.mqde, *16*, *17*, *21*, *24*, *27*, *29*, 30, *36*, *48*
BMTfit.qme, *16*, *17*, *21*, *24*, *27*, *29*, *33*, 33
BMTiqr (BMTdispersion), 13
BMTkurt (BMTkurtosis), 36
BMTkurtosis, *4*, *10*, *14*, 36, *40*, *42*
BMTmean (BMTcentral), 9
BMTmedian (BMTcentral), 9
BMTmgf (BMTmoments), 38
BMTmode (BMTcentral), 9
BMTmoment (BMTmoments), 38
BMTmoments, *4*, *10*, *14*, *38*, 38, *42*
BMTsd (BMTdispersion), 13
BMTskew (BMTskewness), 41
BMTskewness, *4*, *10*, *14*, *38*, *40*, 41
BMTsteep (BMTkurtosis), 36
BMTvar (BMTdispersion), 13
bootdist, *17*

cdfcomp, *17*
constrOptim, *20*, *21*, *23*, *24*, *26–29*, *32*, *33*, *35*, *36*, *43*, *44*, *47*, *49*

dBMT (BMT), 2
dBMT.Phi (BMT.Phi), 5
dBMT.Psi (BMT.Psi), 7
Distributions, *7*, *8*

fitdist, *16*, *17*, *21*, *24*, *27*, *29*, *33*, *36*
fitdistrplus, *16*

gofstat, *17*

mBMT (BMTmoments), 38
mgedist, *17*, *20*, *21*, *45*, *49*
mledist, *15–17*, *19*, *20*, *22*, *23*, *25*, *28*, *29*, *31*, *32*, *34*, *43–45*, *47–49*
mmedist, *17*, *25*, *26*, *45*, *49*
mpsedist, *17*, *28*, *29*, 43, *49*
mqdedist, *17*, *31*, *32*, *45*, 46

optim, *16*, *20–29*, *31–36*, *43–45*, *47–49*

pBMT, *7*, *8*
pBMT (BMT), 2
pBMT.Phi, *8*
pBMT.Phi (BMT.Phi), 5
pBMT.Psi, *7*
pBMT.Psi (BMT.Psi), 7
plotdist, *17*

qBMT (BMT), 2
qBMT.Phi (BMT.Phi), 5

qBMT.Psi *(*BMT.Psi*)*, 7
qmedist, *17*, *34*, *35*, *45*, *49*
quantile, *31*, *34*, *47*, *49*

rBMT *(*BMT*)*, 2
rBMT.Phi *(*BMT.Phi*)*, 5
rBMT.Psi *(*BMT.Psi*)*, 7

score.math.booklet10.PISA2012, 50