Package ‘MBBEFDLite’

May 7, 2026

Type Package

Title Statistical Functions for the Maxwell-Boltzmann-Bose-Einstein-Fermi-Dirac (MBBEFD) Family of Distributions

Version 1.0.0

Description Provides probability mass, distribution, quantile, random variate generation, and method-of-moments parameter fitting for the MBBEFD family of distributions used in insurance modeling as described in Bernegger (1997) <doi:10.2143/AST.27.1.563208> without any external dependencies.

License MPL-2.0

Encoding UTF-8

Imports stats

Suggests tinytest, covr

URL https://github.com/aadler/MBBEFDLite

BugReports https://github.com/aadler/MBBEFDLite/issues

ByteCompile yes

NeedsCompilation yes

UseLTO yes

Author Avraham Adler [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-3039-0703>)

Maintainer Avraham Adler <Avraham.Adler@gmail.com>

Repository CRAN

Date/Publication 2026-03-10 07:20:02 UTC

Contents

`MBBEFDLite-package` *Statistical Functions for the Maxwell-Boltzmann-Bose-Einstein-Fermi-Dirac (MBBEFD) Family of Distributions*

Description

Provides probability mass, distribution, quantile, random variate generation, and method-of-moments parameter fitting for the MBBEFD family of distributions used in insurance modeling as described in Bernegger (1997) <doi:10.2143/AST.27.1.563208> without any external dependencies.

Details

The DESCRIPTION file:

Package:	MBBEFDLite
Type:	Package
Title:	Statistical Functions for the Maxwell-Boltzmann-Bose-Einstein-Fermi-Dirac (MBBEFD) Family of Dis
Version:	1.0.0
Authors@R:	person(given="Avraham", family="Adler",role=c("aut", "cre", "cph"), email="Avraham.Adler@gmail.c
Description:	Provides probability mass, distribution, quantile, random variate generation, and method-of-moments pa
License:	MPL-2.0
Encoding:	UTF-8
Imports:	stats
Suggests:	tinytest, covr
URL:	https://github.com/aadler/MBBEFDLite
BugReports:	https://github.com/aadler/MBBEFDLite/issues
ByteCompile:	yes
NeedsCompilation:	yes
UseLTO:	yes
Author:	Avraham Adler [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-3039-0703>)
Maintainer:	Avraham Adler <Avraham.Adler@gmail.com>
Archs:	x64

Index of help topics:

Author(s)

Avraham Adler [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-3039-0703>)

Maintainer: Avraham Adler <Avraham.Adler@gmail.com>

dmb *The MBBEFD Distribution*

Description

Density, distribution function, quantile function, and random generation for the MBBEFD distribution with parameters g and b.

Usage

```
dmb(x, g, b, c = NULL, log = FALSE)
pmb(q, g, b, c = NULL, lower.tail = TRUE, log.p = FALSE)
qmb(p, g, b, c = NULL, lower.tail = TRUE, log.p = FALSE)
rmb(n, g, b, c = NULL)
```

Arguments

x, q	**numeric**; vector of quantiles.
p	**numeric**; vector of probabilities.
n	**numeric**; number of observations. If length(n) > 1, the length is taken to be the number required.
g	**numeric**; (vector of) the g parameter, which is also the reciprocal of the probability of a maximum loss.
b	**numeric**; (vector of) the b parameter.
c	**numeric**; (vector of) the optional single c parameter. Should be NULL if g and b are passed. Otherwise, $g = e^{(0.78+0.12c)c}$ and $b = e^{3.1-0.15(1+c)c}$.
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 $P[X > x]$.

Details

The MBBEFD class of curves are defined in Bernegger (1997) and are often used to model insurance risk. The density is defined on the semi-open interval $[0, 1)$ and the distribution and quantile functions are defined on the closed interval $[0, 1]$. The parameters must satisfy $g \geq 1$ and $b \geq 0$.

Value

dmb gives the density, pmb gives the distribution function, qmb gives the quantile function, and rmb generates random deviates.

The length of the result is determined by n for rmb, and is the length of x, p, or q as appropriate for the other functions.

Numerical arguments other than n are recycled to the length of the result. Logical arguments should be of length 1.

Note

This package follows Bernegger's convention that the density function does not exist at 1. This differs from the **mbbefd** package.

Author(s)

Avraham Adler <Avraham.Adler@gmail.com>

References

Bernegger, S. (1997) The Swiss Re Exposure Curves and the MBBEFD Distribution Class. ASTIN *Bulletin* **27**(1), 99–111. doi:10.2143/AST.27.1.563208

See Also

mommb for parameter estimation.

Examples

```
all.equal(dmb(0.5, 1, 0), 0)
dmb(0.2, 20, 5)
pmb(0.98, 25, 4)
qmb(0.98, 25, 4) == 1
all.equal(qmb(pmb(0.98, 25, 4), 25, 4), 0.98)
set.seed(45)
rmb(3, 4, 12)
set.seed(45)
rmb(99:101, 4, 12) # length(99:101) = 3, so generates same 3 values as above
```

---

ecmb — *Exposure Curve for the MBBEFD Distribution*

Description

Returns the limited average severity at x of a random variable with an MBBEFD distribution with parameters g and b.

Usage

```
ecmb(x, g, b, c = NULL, lower.tail = TRUE)
```

Arguments

x	**numeric**; vector of quantiles.
g	**numeric**; (vector of) the g parameter, which is also the reciprocal of the probability of a maximum loss.
b	**numeric**; (vector of) the b parameter.

c	**numeric**; (vector of) the optional c parameter. Should be NULL if g and b are passed. Otherwise, $g = e^{(0.78+0.12c)c}$ and $b = e^{3.1-0.15(1+c)c}$.
lower.tail	**logical**; if TRUE (default), percentages are of the total loss being less than or equal to x. Otherwise they are the percentage of total loss greater than x.

Details

Given random variable X with an MBBEFD distribution with parameters g and b, the **exposure curve** (EC) is defined as the ratio of the limited average severity (LAS) of the variable at x divided by the unlimited expected severity of the distribution:

$$EC(x) = \frac{LAS(x)}{E(X)} = \frac{E(X \wedge x)}{E(X)} = \frac{\int_0^x tf(t)dt + x\int_x^\infty f(t)dt}{\int_0^\infty tf(t)dt}$$

If one considers x as a policy limit, then the value of `ecmb(x, g, b)` is the percentage of the total expected loss which will be contained within the (reinsurance) policy limit. If `lower.tail` is `FALSE`, the return value is the percentage of total loss which will exceed the limit.

Value

A numeric vector containing the values of the exposure curve for the passed x, b, and g vectors. If `lower.tail` is `FALSE`, the return value is the complement of the exposure curve.

Author(s)

Avraham Adler <Avraham.Adler@gmail.com>

References

Bernegger, S. (1997) The Swiss Re Exposure Curves and the MBBEFD Distribution Class. ASTIN *Bulletin* **27**(1), 99–111. doi:10.2143/AST.27.1.563208

See Also

dmb and pmb for the density and distribution.

Examples

```
all.equal(ecmb(c(0, 1), 20, 5), c(0, 1))
ecmb(0.25, 100, 34)
```

mommb *Method of Moments Parameter Estimation for the MBBEFD distribution*

Description

Attempts to find the best g and b parameters which are consistent with the first and second moments of the supplied data.

Usage

```
mommb(x, m = FALSE, tol = NULL, na.rm = TRUE, opts = list())
```

Arguments

- `x` — **numeric**; If m is FALSE, a vector of observations between 0 and 1. If m is TRUE, then a vector of length 2, where the first element is the first central moment (mean) of the MBBEFD distribution and the second element is the second central moment (variance) of the MBBEFD distribution.
- `m` — **logical**; When FALSE—the default—x is treated as a vector of observations. When TRUE, x is treated as the couplet of the distribution's first two central **m**oments— $E[X]$ and $Var[X]$.
- `tol` — **numeric**; tolerance of the expectation-maximization algorithm. If too tight, algorithm may fail. Defaults to the square root of .Machine$double.eps or roughly 1.49×10^{-8}—see Details.
- `na.rm` — **logical**; if TRUE (default) NAs are removed. If FALSE, and there are NAs, the algorithm will stop with an error.
- `opts` — list; Configuration options including:
  - alg: **character**; either "EM", the expectation-maximization algorithm of the package author or "LS", a form of the grid-search algorithm of Bernegger (2026), implemented as a nested set of 1D line-search optimizations.
  - maxit: **integer**; maximum number of iterations for the EM algorithm. Ignored for the LS algorithm.
  - maxb: **numeric**; the upper bound of the b parameter for fitting purposes. Used in both algorithms. Defaults to 1e6.
  - minb: **numeric**; the lower bound of the b parameter for fitting purposes. Only used in LS algorithm. Must be positive and less than maxb. Defaults to 1e-10.
  - maxg: **numeric**; the upper bound of the g parameter for fitting purposes. Only used in LS algorithm. Must be positive. Defaults to 1e6.
  - ming: **numeric**; the lower bound of the g parameter for fitting purposes. Only used in LS algorithm. Must be strictly greater than 1 and less than maxg. Defaults to 1 + 1e-10.
  - trace: **logical**; if TRUE, the EM algorithm will print the values of g and b at each iteration i. Ignored with a message for the LS algorithm. The default is FALSE.

Details

There are two fitting algorithms.

Expectation-Maximization: The default is an expectation-maximization form based on sections 4.1 and 4.2 of Bernegger (1997). With rare exceptions, the fitted g and b parameters must conform to:

$$\mu = \frac{\ln(gb)(1-b)}{\ln(b)(1-gb)}$$

subject to:

$$\mu^2 \leq E[x^2] \leq \mu p \leq E[x^2]$$

where μ and μ^2 are the "true" first and second raw moments, $E[x^2]$ is the empirical second raw moment, and p is the mass point probability of a maximal loss: $1 - F(1^-)$.

The algorithm starts with the estimate $p = E[x^2]$ as an upper bound. However, in step 2 of section 4.2, the p component is estimated as the difference between the numerical integration of $x^2 f(x)$ and the empirical second moment—$p = E[x^2] - \int x^2 f(x)dx$—as seen in equation (4.3). This is converted to g by reciprocation and convergence is tested by the difference between this new g and its prior value. If the new $p \leq 0$, the algorithm stops with an error.

Line Search: Bernegger (2026) in Algorithm 3 (Appendix C) describes a grid-search algorithm for converging on g and b. The original algorithm looks for the sets of parameters which return the appropriate mean and coefficient of variation (CV), the latter of which can be expressed in closed form using the dilogarithm function. However, instead of the 50,000 point grid suggested in the paper, this package implements the algorithm as a nested set of calls to the one-dimensional optimization algorithm of `optimize`. This is significantly faster and has the benefit of returning a value even when no zeros can be found. The algorithm basically defaults to the ranges suggested in the paper, but these may be passed by the user in the `opts` list. Sometimes, the algorithm converges to the upper bound of g. In this case, the package implementation will try once more using the square root of `maxg` instead. This can allow convergence—often to the same point of the EM algorithm, given that converged. When this happens, the `iter` value will be `2` instead of `1`.

Value

Returns a list containing:

`g`	The fitted `g` parameter.
`b`	The fitted `b` parameter.
`iter`	For the EM algorithm, the number of iterations used. For the LS algorithm, the number of attempts (1 or 2 if the retry was needed; see **Details**).
`sqerr`	The squared error between the empirical mean and the theoretical mean given the fitted `g` and `b`. This does not make sense for some of the special branch cases, like $b = 0,\ g = 1$, etc.

Note

Anecdotal evidence indicates that parameter estimates from either fitting algorithm can be volatile when sample sizes are small (fewer than a few hundred observations).

Author(s)

Avraham Adler <Avraham.Adler@gmail.com>

References

Bernegger, Stefan. (1997) The Swiss Re Exposure Curves and the MBBEFD Distribution Class. ASTIN *Bulletin* **27**(1), 99–111. doi:10.2143/AST.27.1.563208

Bernegger, Stefan. (2026) Properties of the MBBEFD Distribution Classes. https://www.researchgate.net/publication/400516019_Properties_of_the_MBBEF_D_Distribution_Classes

See Also

`rmb` for random variate generation.

Examples

```
set.seed(85L)
x <- rmb(1000, 25, 4)
mommb(x)
mommb(x, opts = list(alg = "LS"))
```

Index