Package: MBBEFDLite (via r-universe)

August 24, 2024

Type Package

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

Version 0.0.4

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

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

RemoteUrl https://github.com/aadler/mbbefdlite

RemoteRef HEAD

RemoteSha beab09e7e3a202666abbe7d782bd313728d764bc

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: This package was not yet installed at build time.

Index: This package was not yet installed at build time.

Author(s)

Avraham Adler [aut, cre, cph] (<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 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 $[0, 1]$.

Value

dnorm gives the density, pnorm gives the distribution function, qnorm gives the quantile function, and rnorm generates random deviates.

The length of the result is determined by n for rnorm, 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 exists 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) # Should equal previous call
```

---

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, maxit = 100L, tol = .Machine$double.eps ^ 0.5, na.rm = TRUE)
```

Arguments

`x`	**numeric**; vector of observations between 0 and 1.
`maxit`	**integer**; maximum number of iterations.
`tol`	**numeric**; tolerance. If too tight, algorithm may fail. Defaults to the square root of `.Machine$double.eps` or roughly 1.49×10^{-8}.
`na.rm`	**logical**; if `TRUE` (default) `NA`s are removed. If `FALSE`, and there are `NA`s, the algorithm will stop with an error.

Details

The algorithm is 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 moments and $E[x^2]$ is the empirical second moment.

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 attempts to restart with a larger g—a smaller p. In this case, the algorithm tends to fail to converge.

Value

Returns a list containing:

`g`	The fitted `g` parameter.
`b`	The fitted `b` parameter.
`iter`	The number of iterations used.
`sqerr`	The squared error between the empirical mean and the theoretical mean given the fitted `g` and `b`.

Note

Anecdotal evidence indicates that the results of this fitting algorithm can be volatile, especially with fewer than a few hundred observations.

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

`rmb` for random variate generation.

Examples

```
set.seed(85L)
x <- rmb(1000, 25, 4)
mommb(x)
```

Index