Package ‘mbbefd’

December 18, 2024

Type Package

Title Maxwell Boltzmann Bose Einstein Fermi Dirac Distribution and Destruction Rate Modelling

Version 0.8.13

Description Distributions that are typically used for exposure rating in general insurance, in particular to price reinsurance contracts. The vignette shows code snippets to fit the distribution to empirical data. See, e.g., Bernegger (1997) <doi:10.2143/AST.27.1.563208> freely available on-line.

License GPL-2

Depends R (>= 3.6), fitdistrplus (>= 1.1-4), alabama, Rcpp (>= 0.12.18)

ByteCompile yes

Suggests testthat, pander, rmarkdown, knitr, lattice

LinkingTo Rcpp

Imports utils, actuar, gsl, MASS

URL `https://github.com/spedygiorgio/mbbefd`

BugReports `https://github.com/spedygiorgio/mbbefd/issues`

VignetteBuilder knitr

SystemRequirements GNU make

NeedsCompilation yes

RoxygenNote 6.0.1

Author Christophe Dutang [aut, cre] (<`https://orcid.org/0000-0001-6732-1501`>), Giorgio Spedicato [aut] (<`https://orcid.org/0000-0002-0315-8888`>), Markus Gesmann [ctb]

Maintainer Christophe Dutang <`dutangc@gmail.com`>

Repository CRAN

Date/Publication 2024-12-18 13:40:02 UTC

Contents

`mbbefd-package` *Maxwell Boltzmann Bose Einstein Fermi Dirac Distribution and Destruction Rate Modelling*

Description

The idea of this package emerged in 2013 from G.A. Spedicato who at this time worked in the area of quantitative risk assessment. In 2015, M. Gesmann and C. Dutang joined the project. This project is hosted at github.

This package contains the core functions of the two parametrizations of the MBBEFD distribution (distribution function, density, quantile functions, random generation, aka d, p, q, r) as well as MBBEFD exposure curve (ec) and raw moments (m).

This package also provides other distributions used for destruction rate modelling, that is the beta, the shifted truncated Pareto and the generalized beta distributions. Due to the presence of a total loss, a one-inflated version of the previous distributions is also provided.

The vignette shows code snippets to fit the distribution to empirical data: Exposure rating, destruction rate models and the mbbefd package.

Author(s)

Christophe Dutang (maintainer), Giorgio Spedicato, Markus Gesmann

References

BERNEGGER, STEFAN (1997). *The Swiss Re Exposure Curves And The MBBEFD Distribution Class*, ASTIN Bulletin, 27(1), pp99-111, doi:10.2143/AST.27.1.563208.

See Also

See `mbbefd-distr` for the MBBEFD distribution;
`swissRe`, `exposureCurve` for exposure curves;
`gbeta`, `stpareto` for finite-support distributions;
`oidistribution`, `oibeta`, `oigbeta`, `oiunif`, `oistpareto` for one-inflated distributions.

`asiacomrisk` *Large commercial risks in Asia-Pacific*

Description

A completed project by the Insurance Risk and Finance Research Centre (www.IRFRC.com) has assembled a unique dataset from Large Commercial Risk losses in Asia-Pacific (APAC) covering the period 2000-2013. The data was generously contributed by one global reinsurance company and two large Lloyd's syndicates in London. This dataset is the result of the project co-lead by Dr Milidonis (IRFRC and University of Cyprus) and Enrico Biffis (Imperial College Business School), which can be referred to as the IRFRC LCR Dataset.

As expected, the dataset is fully anonymized, as the LCR losses are aggregated along a few dimensions. First, data is categorized based on the World Bank's economic development classification. This means that losses either come from developed or developing countries. The second dimension used to aggregate the data is the time period covered. Data is grouped into (at least) two time-periods: the period before and after the 2008 crisis.

A large commercial risk (LCR) is defined as a loss caused by man-made risks (e.g. fire, explosion, etc.). We exclude natural catastrophe events, and started by focusing on claims that made the data provider incur a loss amount of at least EUR 1 million. We then extended our dataset to include claims leading to loss amounts smaller that EUR 1 million. Given time constraints, we only partially extended loss data by obtaining FGU losses larger than EUR 140k. One should note that any selection bias arising from the data collection exercise is driven by both data quality and reliability. Based on our experience, the latter two attributes are homogeneous across developed and developing countries APAC claims.

For further details, see the technical report: Benedetti, Biffis and Milidonis (2015a).

Usage

```
data(asiacomrisk)
```

Format

`asiacomrisk` contains 7 columns:

`Period` A character string for the period: `"2000-2003"`, `"2004-2008"`, `"2009-2010"`, `"2011-2013"`.

FGU From the Ground Up Loss (USD).

TIV Total Insurable Value (TIV) replaced with Total Sum Insured (TSI) when the TIV is not available (USD).

CountryStatus A character string for the country status: "Developped", "Emerging".

Usage A character string for the type of exposure hit by the loss: "Commercial", "Energy", "Manufacturing", "Misc.", "Residential".

SubUsage A character string for a precise type of exposure hit by the loss: "Commercial", "Energy", "General industry", "Metals/Mines/Chemicals", "Misc.", "Residential", "Utility".

DR A numeric for the destruction rate (FGU divided TIV capped to 1).

References

Benedetti, D., Biffis, E., and Milidonis, A. (2015a). *Large Commercial Risks (LCR) in Insurance: Focus on Asia-Pacific*, Insurance Risk and Finance Research Centre Technical report.

Benedetti, D., Biffis, E., and Milidonis, A. (2015b). *Large Commercial Exposures and Tail Risk: Evidence from the Asia-Pacific Property and Casualty Insurance Market*, Working paper.

Chavez-Demoulin, V., Embrechts, P., and Hofert, M. (2015). *An extreme value approach for modeling operational risk losses depending on covariates*, The Journal of Risk and Insurance.

Examples

```
# (1) load of data
#
data(asiacomrisk)
dim(asiacomrisk)

# (2) basic boxplots
#

asiacomrisk
boxplot(DR ~ Usage, data=asiacomrisk)
boxplot(DR ~ SubUsage, data=asiacomrisk)
boxplot(DR ~ Period, data=asiacomrisk)
boxplot(DR ~ CountryStatus, data=asiacomrisk)
```

---

beaonre *AON Re Belgian dataset*

Description

The dataset was collected by the reinsurance broker AON Re Belgium and comprise 1,823 fire losses for which the building type and the sum insured are available.

Usage

```
data(beaonre)
```

Format

beaonre contains three columns and 1823 rows:

`BuildType` The building type either A, B, C, D, E or F.

`ClaimCost` The loss amount in thousand of Danish Krone (DKK).

`SumInsured` The sum insured in thousand of Danish Krone (DKK).

References

Dataset used in Beirlant, Dierckx, Goegebeur and Matthys (1999), *Tail index estimation and an exponential regression model*, Extremes 2, 177-200, doi:10.1023/A:1009975020370.

Examples

```
# (1) load of data
#
data(beaonre)

# (2) plot and description of data
#

boxplot(ClaimCost ~ BuildType, data=beaonre, log="y",
xlab="Building type", ylab="Claim size", main="AON Re Belgium data")
```

`bootDR` *Bootstrap simulation of destruction rate models*

Description

Uses parametric or nonparametric bootstrap resampling in order to simulate uncertainty in the parameters of the distribution fitted to destruction rate data. Generic methods are `print`, `plot`, `summary`.

Usage

```
bootDR(f, bootmethod="param", niter=1001, silent=TRUE)
```

Arguments

- `f` — An object of class `"fitDR"`, output of the `fitDR` function.
- `bootmethod` — A character string coding for the type of resampling : `"param"` for a parametric resampling and `"nonparam"` for a nonparametric resampling of data.
- `niter` — The number of samples drawn by bootstrap.
- `silent` — A logical to remove or show warnings and errors when bootstraping.

Details

Samples are drawn by parametric bootstrap (resampling from the distribution fitted by `fitDR`) or nonparametric bootstrap (resampling with replacement from the data set). On each bootstrap sample the estimation process is used to estimate bootstrapped values of parameters. When that function fails to converge, `NA` values are returned. Medians and 2.5 and 97.5 percentiles are computed by removing `NA` values.

This method returns an object of class `"bootDR"` inheriting from the `"bootdist"` class. Therefore the following generic methods are defined: `print`, `plot`, `summary`.

Value

`bootDR` returns an object of class `"bootDR"` inheriting from the `"bootdist"` class. That is a list with 6 components,

- `estim` — a data frame containing the bootstrapped values of parameters.
- `converg` — a vector containing the codes for convergence obtained if an iterative method is used to estimate parameters on each bootstraped data set (and 0 if a closed formula is used).
- `method` — A character string coding for the type of resampling : `"param"` for a parametric resampling and `"nonparam"` for a nonparametric resampling.
- `nbboot` — The number of samples drawn by bootstrap.
- `CI` — bootstrap medians and 95 percent confidence percentile intervals of parameters.
- `fitpart` — The object of class `"fitDR"` on which the bootstrap procedure was applied.

Generic functions:

- `print` The print of a `"bootDR"` object shows the bootstrap parameter estimates. If inferior to the whole number of bootstrap iterations, the number of iterations for which the estimation converges is also printed.
- `summary` The summary provides the median and 2.5 and 97.5 percentiles of each parameter. If inferior to the whole number of bootstrap iterations, the number of iterations for which the estimation converges is also printed in the summary.
- `plot` The plot shows the bootstrap estimates with `stripchart` function for univariate parameters and `plot` function for multivariate parameters.

Author(s)

Christophe Dutang

References

Cullen AC and Frey HC (1999), *Probabilistic techniques in exposure assessment*. Plenum Press, USA, pp. 181-241.

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

See Also

See `mledist`, `mmedist`, `qmedist`, `mgedist` for details on parameter estimation. See `bootdist` for details on generic function. See `fitDR` for estimation procedures.

Examples

```
# We choose a low number of bootstrap replicates in order to satisfy CRAN running times
# constraint.
# For practical applications, we recommend to use at least niter=501 or niter=1001.
```

`eecf` *Empirical Exposure Curve Function*

Description

Compute an empirical exposure curve function, with several methods for plotting, printing, computing with such an object.

Usage

```
eecf(x)


## S3 method for class 'eecf'
plot(x, ..., ylab="Gn(x)", do.points=TRUE,
     col.01line = "gray70", pch = 19, main=NULL, ylim=NULL,
     add=FALSE)

## S3 method for class 'eecf'
lines(x, ...)

## S3 method for class 'eecf'
print(x, digits= getOption("digits") - 2, ...)

## S3 method for class 'eecf'
summary(object, ...)
```

Arguments

`x, object`	numeric vector of the observations for `eecf`; for the methods, an object of class `"eecf"`.
`...`	arguments to be passed to subsequent methods, e.g., to the `plot` method.
`ylab`	label for the y-axis.
`do.points`	logical; if `TRUE`, also draw points at the (`xlim` restricted) knot locations.
`col.01line`	numeric or character specifying the color of the horizontal lines at y = 0 and 1, see `colors`.
`pch`	plotting character.
`main`	main title.
`ylim`	the y limits of the plot.
`add`	logical; if `TRUE` add to an already existing plot.
`digits`	number of significant digits to use, see `print`.

Details

Compute a continuous empirical exposure curve and returns an object of class `"eecf"` similar to what an object returned by `ecdf`.

Value

For `eecf`, a function of class `"eecf"`, inheriting from the `"function"` class.

For the `summary` method, a summary of the knots of object with a `"header"` attribute.

Author(s)

Dutang Christophe

See Also

`exposureCurve`, `ecdf`.

Examples

```
x <- c(0.4756816, 0.1594636, 0.1913558, 0.2387725, 0.1135414, 0.7775612,
  0.6858736, 0.4340655, 0.3181558, 0.1134244)

#print
eecf(x)

#summary
summary(eecf(x))

#plot
plot(eecf(x))

#lines
lines(eecf(x[1:4]), col="red")
```

etl *Empirical total loss*

Description

Compute the empirical total loss.

Usage

```
etl(x, na.rm=FALSE)
```

Arguments

x	numeric vector of the observations.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.

Details

Compute the empirical total loss defined as the proportion of full destruction rates, that is observations that equal 1.

Value

A numeric value or a vector.

Author(s)

Dutang Christophe

Examples

```
x <- c(1, 0.000495134903027804, 0.787229130724068, 0.71154311082138,
0.0669802789251427, 0.310872967333683, 1, 1, 1, 1, 0.162030982251957,
1, 1, 0.322530106394859, 1, 1, 1, 0.60805410798081, 0.660941675188664, 1)

#empirical total loss (true value is 1/2)
etl(x)
```

exposureCurve *Exposure curves for the beta and the uniform distributions.*

Description

An exposure curve is defined between x between 0 and 1 and represents the ratio of the limited expected value to unlimited expected value.

Usage

```
ecbeta(x, shape1, shape2)
ecunif(x, min = 0, max =1)
```

Arguments

`x`	x value, percentage of damage to total loss
`shape1, shape2`	parameters for the beta distribution.
`min, max`	parameters for the uniform distribution.

Details

`ecbeta`, `ecunif` is the theoretical exposure curve function for beta and uniform distribution.

Value

A numeric value

Author(s)

Giorgio Spedicato, Christophe Dutang

References

BERNEGGER, STEFAN (1997). *The Swiss Re Exposure Curves And The MBBEFD Distribution Class*, ASTIN Bulletin, 27(1), pp99-111, doi:10.2143/AST.27.1.563208.

See Also

`ecmbbefd` and `ecMBBEFD` are implemented in `mbbefd-distr`. See also `Uniform`, `Beta`, `swissRe`.

Examples

```
x <- 0.2
ecbeta(x, 2, 3)
ecunif(x)
```

fitDR *Fit of destruction rate models*

Description

Fit of univariate distributions to destruction rate data by maximum likelihood (mle), moment matching (mme), quantile matching (qme) or maximizing goodness-of-fit estimation (mge). The latter is also known as minimizing distance estimation. Generic methods are `print`, `plot`, `summary`, `quantile`, `logLik`, `vcov` and `coef`.

Usage

```
fitDR(x, dist, method="mle", start=NULL, optim.method="default", ...)
```

Arguments

`x`	A numeric vector.
`dist`	A character string `"name"` naming a distribution among `"oiunif"`, `"oistpareto"`, `"oibeta"`, `"oigbeta"`, `"mbbefd"`, `"MBBEFD"`.
`method`	A character string coding for the fitting method: `"mle"` for 'maximum likelihood estimation', `"tlmme"` for 'total-loss-moment matching estimation'.
`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`).
`optim.method`	`"default"` or an optimization method to pass to `optim`.
`...`	Further arguments to be passed to `"fitdist"` when `method != "tlmme"`. See `fitdist` for details on parameter estimation.

Details

The fitted distribution (`dist`) has its d, p, q, r functions defined in the man page: `oiunif`, `oistpareto`, `oibeta`, `oigbeta`, `mbbefd`, `MBBEFD`.

The two possible fitting methods are described below:

When `method="mle"` Maximum likelihood estimation consists in maximizing the log-likelihood. A numerical optimization is carried out in `mledist` via `optim` to find the best values (see `mledist` for details). For one-inflated distributions, the probability parameter is estimated by a closed-form formula and other parameters use a two-optimization procedures.

When `method="tlmme"` Total loss and moment matching estimation consists in equalizing theoretical and empirical total loss as well as theoretical and empirical moments. The theoretical and the empirical moments are matched numerically, by minimization of the sum of squared differences between observed and theoretical quantities (see `mmedist` for details).

For one-inflated distributions, by default, direct optimization of the log-likelihood (or other criteria depending of the chosen method) is performed using optim, with the "L-BFGS-B" method for distributions characterized by more than one parameter and the "Brent" method for distributions characterized by only one parameter. Note that when errors are raised by optim, it's a good idea to start by adding traces during the optimization process by adding control=list(trace=1, REPORT=1). For the MBBEFD distribution, constrOptim.nl is used.

A pre-fitting process is carried out for the following distributions "mbbefd", "MBBEFD" and "oigbeta" before the main optimization.

The estimation process is carried out via fitdist from the fitdistrplus package and the output object will inherit from the "fitdist" class. Therefore, the following generic methods are available print, plot, summary, quantile, logLik, vcov and coef.

Value

fitDR returns an object of class "fitDR" inheriting from the "fitdist" class. That is a list with the following components:

estimate	the parameter estimates.
method	the character string coding for the fitting method : "mle" for 'maximum likelihood estimation', "tlmme" for 'matching total loss moment 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.
fix.arg	the named list giving the values of parameters of the named distribution that must be kept fixed rather than estimated by maximum likelihood 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 or NULL if no such arguments.
weights	the vector of weigths used in the estimation process or NULL.

Generic functions:

print The print of a "fitDR" object shows few traces about the fitting method and the fitted distribution.

summary The summary provides the parameter estimates of the fitted distribution, the log-likelihood, AIC and BIC statistics and when the maximum likelihood is used, the standard errors of the parameter estimates and the correlation matrix between parameter estimates.

plot The plot of an object of class "fitDR" returned by fitdist uses the function plotdist. An object of class "fitdist" or a list of objects of class "fitDR" corresponding to various fits using the same data set may also be plotted using a cdf plot (function cdfcomp), a density plot(function denscomp), a density Q-Q plot (function qqcomp), or a P-P plot (function ppcomp).

logLik Extracts the estimated log-likelihood from the "fitDR" object.

vcov Extracts the estimated var-covariance matrix from the "fitDR" object (only available when method = "mle").

coef Extracts the fitted coefficients from the "fitDR" object.

Author(s)

Christophe Dutang.

References

Cullen AC and Frey HC (1999), *Probabilistic techniques in exposure assessment*. Plenum Press, USA, pp. 81-155.

Venables WN and Ripley BD (2002), *Modern applied statistics with S*. Springer, New York, pp. 435-446.

Vose D (2000), *Risk analysis, a quantitative guide*. John Wiley & Sons Ltd, Chischester, England, pp. 99-143.

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

See Also

See mledist, mmedist, for details on parameter estimation. See gofstat for goodness-of-fit statistics. See plotdist, graphcomp for graphs. See bootDR for bootstrap procedures See optim for base R optimization procedures. See quantile.fitdist, another generic function, which calculates quantiles from the fitted distribution. See quantile for base R quantile computation.

Examples

```
# (1) fit of a one-inflated beta distribution by maximum likelihood estimation
#
n <- 1e3
set.seed(12345)
x <- roibeta(n, 3, 2, 1/6)

f1 <- fitDR(x, "oibeta", method="mle")
summary(f1)

plot(bootdist(f1, niter=11), enhance=TRUE, trueval=c(3, 2, 1/6))
```

g2a — *Get a parameter known g and b*

Description

g2a returns the a parameter known g and b

Usage

```
g2a(g, b)
```

Arguments

g	the g parameter
b	the b parameter

Value

a real value

See Also

mbbefd-distr.

Examples

```
g2a(10,2)
```

gbeta — *The generalized Beta of the first kind Distribution*

Description

Density, distribution function, quantile function and random generation for the GB1 distribution with parameters shape0, shape1 and shape2.

Usage

```
dgbeta(x, shape0, shape1, shape2, log = FALSE)
pgbeta(q, shape0, shape1, shape2, lower.tail = TRUE, log.p = FALSE)
qgbeta(p, shape0, shape1, shape2, lower.tail = TRUE, log.p = FALSE)
rgbeta(n, shape0, shape1, shape2)
ecgbeta(x, shape0, shape1, shape2)
mgbeta(order, shape0, shape1, shape2)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.
`shape0, shape1, shape2`	positive parameters of the GB1 distribution.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$, otherwise, $P[X > x]$.
`order`	order of the raw moment.

Details

The GB1 distribution with parameters `shape0` $= g$, `shape1` $= a$ and `shape2` $= b$ has density

$$f(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a/g-1} (1 - x^{1/g})^{b-1} / g$$

for $a, b, g > 0$ and $0 \le x \le 1$ where the boundary values at $x = 0$ or $x = 1$ are defined as by continuity (as limits).

Value

`dgbeta` gives the density, `pgbeta` the distribution function, `qgbeta` the quantile function, and `rgbeta` generates random deviates.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*, Wadsworth & Brooks/Cole, doi:10.1201/9781351074988.

Abramowitz, M. and Stegun, I. A. (1972) *Handbook of Mathematical Functions.* New York: Dover. Chapter 6: Gamma and Related Functions.

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) *Continuous Univariate Distributions*, Volume 2, especially Chapter 25. Wiley, New York, doi:10.1080/00224065.1996.11979675.

See Also

Distributions for other standard distributions.

Examples

```
#density
curve(dgbeta(x, 3, 2, 3))

#cdf
curve(pgbeta(x, 3, 2, 3))
```

graph-eccomp *Graphical comparison of multiple fitted distributions*

Description

eccomp plots the empirical exposure curve distribution against fitted exposure curve functions.

Usage

```
eccomp(ft, xlim, ylim, main, xlab, ylab, do.points=TRUE,
                    datapch, datacol, fitlty, fitcol, addlegend = TRUE,
                   legendtext, xlegend = "bottomright",
                    ylegend = NULL, ...)
```

Arguments

ft	One "DR" object or a list of objects of class "DR".
xlim	The x-limits of the plot.
ylim	The y-limits of the plot.
main	A main title for the plot, see also title.
xlab	A label for the x-axis, defaults to a description of x.
ylab	A label for the y-axis, defaults to a description of y.
datapch	An integer specifying a symbol to be used in plotting data points, see also points.
datacol	A specification of the color to be used in plotting data points.
fitcol	A (vector of) color(s) to plot fitted distributions. If there are fewer colors than fits they are recycled in the standard fashion.
fitlty	A (vector of) line type(s) to plot fitted distributions/densities. If there are fewer colors than fits they are recycled in the standard fashion. See also par.
addlegend	If TRUE, a legend is added to the plot.
legendtext	A character or expression vector of length ≥ 1 to appear in the legend, see also legend.
xlegend, ylegend	The x and y co-ordinates to be used to position the legend. They can be specified by keyword or in any way which is accepted by 'xy.coords': see legend for details.
do.points	logical; if TRUE, also draw points at the x-locations. Default is true. For large dataset (n > 1e4), do.points is ignored and no point is drawn.
...	Further graphical arguments passed to graphical functions used in cdfcomp, denscomp, ppcomp and qqcomp.

Details

`eccomp` provides a exposure curve plot of each fitted distribution along with the eecf.

By default a legend is added to these plots. Many graphical arguments are optional, dedicated to personalize the plots, and fixed to default values if omitted.

Author(s)

Christophe Dutang.

See Also

See `plot`, `legend`, `eecf`.

Examples

```
# (1)
```

`itagradescore` *Italian grade scores*

Description

This dataset contains scores of an university admission test. The total score is subdivided into four areas (Italian, English, abstract reasoning, science). Each subitem can have a point of pass at the end.

Usage

```
data(itagradescore)
```

Format

`itagradescore` contains 10 columns:

`Number` a numeric for the record number.

`ID` a factor for the identification code.

`Correct` A score of correct answers.

`Wrong` A score of wrong answers.

`Null` A score of null answers.

`ItalianLanguage` A score for the Italian language test.

`EnglishLanguage` A score for the English language test.

`LogicalReasoning` A score for the logic test.

`Science` A score for the science test.

`TotalScore` The sum of the four scores (i.e. four previous columns).

Source

Internal

Examples

```
# (1) load of data
#
data(itagradescore)
dim(itagradescore)
```

---

lossalae *General Liability Claims*

Description

The `lossalae` is a data frame of 1500 rows and 4 columns containing 1,500 general liability claims randomly chosen from late settlement lags and were provided by Insurance Services Office, Inc. Each claim consists of an indemnity payment (the loss, X1) and an allocated loss adjustment expense (ALAE). ALAE are types of insurance company expenses that are specifically attributable to the settlement of individual claims such as lawyers' fees and claims investigation expenses. The third column is the underwriting limit of the policy and and the fourth column indicates a censored observation.

Usage

```
data(lossalaefull)
```

Format

`lossalaefull` contains four columns:

`Loss` A numeric vector containing the indemnity payments (USD).

`ALAE` A numeric vector containing the allocated loss adjustment expenses (USD).

`Limit` A numeric vector containing the policy limit (USD).

`Censored` A binary indicating that the payments are capped to their policy limit (USD).

Source

Frees, E. W. and Valdez, E. A. (1998) Understanding relationships using copulas. *North American Actuarial Journal*, **2**, 1–15, doi:10.1080/10920277.1998.10595749.

References

Klugman, S. A. and Parsa, R. (1999) Fitting bivariate loss distributions with copulas. *Insurance: Mathematics and Economics*, **24**, 139–148, doi:10.1016/S01676687(98)000390.

Beirlant, J., Goegebeur, Y., Segers, J. and Teugels, J. L. (2004) *Statistics of Extremes: Theory and Applications.*, Chichester, England: John Wiley and Sons, doi:10.1002/0470012382.

Cebrian, A.C., Denuit, M. and Lambert, P. (2003). *Analysis of bivariate tail dependence using extreme value copulas: An application to the SOA medical large claims database*, Belgian Actuarial Bulletin, Vol. 3, No. 1, https://dial.uclouvain.be/pr/boreal/object/boreal:17222.

Examples

```
# (1) load of data
#
data(lossalaefull)
```

`mbbefd-distr` *The MBBEFD distribution (two parametrizations)*

Description

These functions perform probabilistic analysis as well as random sampling on the MBBEFD distribution: the 1st parametrization MBBEFD(a,b) is implemented in <d,p,q,r>`mbbefd`, the 2nd parametrization MBBEFD(g,b) is implemented in <d,p,q,r>`MBBEFD`. We also provide raw moments, exposure curve function and total loss.

Usage

```
dmbbefd(x, a, b, log=FALSE)
pmbbefd(q, a, b, lower.tail = TRUE, log.p = FALSE)
qmbbefd(p, a, b, lower.tail = TRUE, log.p = FALSE)
rmbbefd(n, a, b)
ecmbbefd(x, a, b)
mmbbefd(order, a, b)
tlmbbefd(a, b)

dMBBEFD(x, g, b, log=FALSE)
pMBBEFD(q, g, b, lower.tail = TRUE, log.p = FALSE)
qMBBEFD(p, g, b, lower.tail = TRUE, log.p = FALSE)
rMBBEFD(n, g, b)
ecMBBEFD(x, g, b)
mMBBEFD(order, g, b)
tlMBBEFD(g, b)
```

Arguments

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is take to be the number required.
a, b, g	shape parameters. For .mbbefd functions, g is computed from a.
order	order of the raw moment.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X <= x]$, otherwise, $P[X > x]$.

Details

it shall be remebered that $g = \frac{1}{p_1} = \frac{a+b}{(a+1)*b}$.

Value

A numeric value or a vector.

Author(s)

Giorgio Spedicato, Dutang Christophe

References

BERNEGGER, STEFAN (1997). *The Swiss Re Exposure Curves And The MBBEFD Distribution Class*, ASTIN Bulletin, 27(1), pp99-111, doi:10.2143/AST.27.1.563208.

See Also

swissRe, exposureCurve.

Examples

```
#1st parametrization
#
aPar=0.2
bPar=0.04
rmbbefd(n=10,a=aPar,b=bPar) #for random generation
qmbbefd(p=0.7,a=aPar,b=bPar) #for quantiles
dmbbefd(x=0.5,a=aPar,b=bPar) #for density
pmbbefd(q=0.5,a=aPar,b=bPar) #for distribution function

#2nd parametrization
#
gPar=2
bPar=0.04
rMBBEFD(n=10,g=gPar,b=bPar) #for random generation
qMBBEFD(p=0.7,g=gPar,b=bPar) #for quantiles
dMBBEFD(x=0.5,g=gPar,b=bPar) #for density
```

```
pMBBEFD(q=0.5,g=gPar,b=bPar) #for distribution function
```

---

oibeta *One-inflated beta distribution*

Description

These functions perform probabilistic analysis as well as random sampling on one-inflated beta distribution.

Usage

```
doibeta(x, shape1, shape2, p1, ncp=0, log=FALSE)
poibeta(q, shape1, shape2, p1, ncp=0, lower.tail = TRUE, log.p = FALSE)
qoibeta(p, shape1, shape2, p1, ncp=0, lower.tail = TRUE, log.p = FALSE)
roibeta(n, shape1, shape2, p1, ncp=0)
ecoibeta(x, shape1, shape2, p1, ncp=0)
moibeta(order, shape1, shape2, p1, ncp=0)
tloibeta(shape1, shape2, p1, ncp=0)
```

Arguments

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is take to be the number required.
p1, shape1, shape2, ncp	parameters.
order	order of the raw moment.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X <= x]$, otherwise, $P[X > x]$.

Details

d,p,q,ec,m,tl-oibeta functions computes the density function, the distribution function, the quantile function, the exposure curve function, raw moments and total loss of the one-inflated beta distribution. roibeta generates random variates of this distribution.

Value

A numeric value or a vector.

Author(s)

Dutang Christophe

See Also

mbbefd-distr and oidistribution.

Examples

```
#density
curve(doibeta(x, 3, 2, 1/3), n=200)

#cdf
curve(poibeta(x, 3, 2, 1/3), n=200)
```

---

oidistribution — *One-inflated distributions*

Description

These functions perform probabilistic analysis as well as random sampling on one-inflated distributions.

Usage

```
doifun(x, dfun, p1, log=FALSE, ...)
poifun(q, pfun, p1, lower.tail = TRUE, log.p = FALSE, ...)
qoifun(p, qfun, p1, lower.tail = TRUE, log.p = FALSE, ...)
roifun(n, rfun, p1, ...)
ecoifun(x, ecfun, mfun, p1, ...)
moifun(order, mfun, p1, ...)
tloifun(p1, ...)
```

Arguments

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is take to be the number required.
dfun, pfun, qfun, rfun	d, p, q, r functions of the original distribution.
p1	parameter for the probability at x=1.
ecfun, mfun	exposure curve and moment functions which should have arguments x, ... and order, ... respectively.
order	order of the raw moment.

log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X <= x]$, otherwise, $P[X > x]$.
...	further arguments to pass to dfun, pfun, qfun, rfun, ecfun, mfun.

Details

d,p,q,ec,m,tl functions of oifun computes the density function, the distribution function, the quantile function, the exposure curve function, raw moments and total loss of an one-inflated distribution of an original distribution specified by d,p,q,ec,m-fun. roifun generates random variates of the resulting distribution.

Value

A numeric value or a vector.

Author(s)

Dutang Christophe

See Also

oibeta, oiunif, oistpareto and oidistribution.

---

oigbeta *One-inflated generalized beta of the first kind (GB1)) distribution*

Description

These functions perform probabilistic analysis as well as random sampling on one-inflated GB1 distribution.

Usage

```
doigbeta(x, shape0, shape1, shape2, p1, log=FALSE)
poigbeta(q, shape0, shape1, shape2, p1, lower.tail = TRUE, log.p = FALSE)
qoigbeta(p, shape0, shape1, shape2, p1, lower.tail = TRUE, log.p = FALSE)
roigbeta(n, shape0, shape1, shape2, p1)
ecoigbeta(x, shape0, shape1, shape2, p1)
moigbeta(order, shape0, shape1, shape2, p1)
tloigbeta(shape0, shape1, shape2, p1)
```

Arguments

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is take to be the number required.
p1, shape0, shape1, shape2	shape parameters.
order	order of the raw moment.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X <= x]$, otherwise, $P[X > x]$.

Details

d,p,q,ec,m,tl-oigbeta functions computes the density function, the distribution function, the quantile function, the exposure curve function, raw moments and total loss of the one-inflated GB1 distribution. roigbeta generates random variates of this distribution.

Value

A numeric value or a vector.

Author(s)

Dutang Christophe

See Also

mbbefd-distr and oidistribution.

Examples

```
#density
curve(doigbeta(x, 3, 2, 3, 1/3), n=200)

#cdf
curve(poigbeta(x, 3, 2, 3, 1/3), n=200)
```

---

oistpareto *One-inflated shifted truncated pareto distribution*

Description

These functions perform probabilistic analysis as well as random sampling on one-inflated shifted truncated pareto distribution.

Usage

```
doistpareto(x, a, p1, log=FALSE)
poistpareto(q, a, p1, lower.tail = TRUE, log.p = FALSE)
qoistpareto(p, a, p1, lower.tail = TRUE, log.p = FALSE)
roistpareto(n, a, p1)
ecoistpareto(x, a, p1)
moistpareto(order, a, p1)
tloistpareto(a, p1)
```

Arguments

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is take to be the number required.
a, p1	parameters.
order	order of the raw moment.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are $P[X <= x]$, otherwise, $P[X > x]$.

Details

d,p,q,ec,m,tl-oistpareto functions computes the density function, the distribution function, the quantile function, the exposure curve function, raw moments and total loss of the one-inflated shifted truncated pareto distribution. roistpareto generates random variates of this distribution.

Value

A numeric value or a vector.

Author(s)

Dutang Christophe

See Also

mbbefd-distr and oidistribution.

Examples

```
#density
curve(doistpareto(x, 2, 1/3), n=200)

#cdf
curve(poistpareto(x, 2, 1/3), n=200)
```

`oiunif` *One-inflated uniform distribution*

Description

These functions perform probabilistic analysis as well as random sampling on one-inflated uniform distribution.

Usage

```
doiunif(x, p1, log=FALSE)
poiunif(q, p1, lower.tail = TRUE, log.p = FALSE)
qoiunif(p, p1, lower.tail = TRUE, log.p = FALSE)
roiunif(n, p1)
ecoiunif(x, p1)
moiunif(order, p1)
tloiunif(p1)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is take to be the number required.
`p1`	parameter.
`order`	order of the raw moment.
`log, log.p`	logical; if `TRUE`, probabilities `p` are given as log(p).
`lower.tail`	logical; if `TRUE` (default), probabilities are $P[X <= x]$, otherwise, $P[X > x]$.

Details

`d,p,q,ec,m,tl-oiunif` functions computes the density function, the distribution function, the quantile function, the exposure curve function, raw moments and total loss of the one-inflated uniform distribution. `roiunif` generates random variates of this distribution.

Value

A numeric value or a vector.

Author(s)

Dutang Christophe

See Also

`mbbefd-distr` and `oidistribution`.

Examples

```
#density
curve(doiunif(x, 1/3), n=200, ylim=0:1)

#cdf
curve(poiunif(x, 1/3), n=200)
```

`stpareto` *The shifted truncated Pareto distribution*

Description

These functions perform probabilistic analysis as well as random sampling on the shifted truncated Pareto distribution.

Usage

```
dstpareto(x, a, log=FALSE)
pstpareto(q, a, lower.tail = TRUE, log.p = FALSE)
qstpareto(p, a, lower.tail = TRUE, log.p = FALSE)
rstpareto(n, a)
mstpareto(order, a)
ecstpareto(x, a)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is take to be the number required.
`order`	order of the raw moment.
`a`	shape parameter.
`log, log.p`	logical; if `TRUE`, probabilities `p` are given as log(p).
`lower.tail`	logical; if `TRUE` (default), probabilities are $P[X <= x]$, otherwise, $P[X > x]$.

Details

The distribution is based on the Pareto 2 truncated at 1. The distribution function is given by $P(X <= x) = (1 - (x + 1)^(- a))/(1 - 2^(- a))$.

Value

A numeric value or a vector.

Author(s)

Dutang Christophe

See Also

mbbefd-distr, exposureCurve

Examples

```
#density
curve(dstpareto(x, 3))

#cdf
curve(pstpareto(x, 3))
```

---

swissRe *Swiss Re exposure curve generation function*

Description

This function turns out the MBBEFD b and g parameters for the famous Swiss Re (SR) exposure curves.

Usage

```
swissRe(c)
```

Arguments

c	A numeric value

Details

The four Swiss Re Y1-Y4 are defined for c=1.5, 2, 3, 4. In addition c=5 coincides with a curve used by Lloyds for industrial risks exposure rating.

Value

A named two dimensional vector

Author(s)

Giorgio Spedicato

References

BERNEGGER, STEFAN (1997). *The Swiss Re Exposure Curves And The MBBEFD Distribution Class*, ASTIN Bulletin, 27(1), pp99-111, doi:10.2143/AST.27.1.563208.

See Also

mbbefd-distr.

Examples

```
pars <- swissRe(4)
losses <- rMBBEFD(n=1000,b=pars[1],g=pars[2])
mean(losses)
```

Index