Package 'ExtDist'

September 23, 2025

Version 0.7-4

License GPL (>= 2)

Title Extending the Range of Functions for Probability Distributions

Description A consistent, unified and extensible framework for estimation of parameters for probability distributions, including parameter estimation procedures that allow for weighted samples; the current set of distributions included are: the standard beta, The four-parameter beta, Burr, gamma, Gumbel, Johnson SB and SU, Laplace, logistic, normal, symmetric truncated normal, truncated normal, symmetric-reflected truncated beta, standard symmetric-reflected truncated beta, triangular, uniform, and Weibull distributions; decision criteria and selections based on these decision criteria.

Repository CRAN

URL https://github.com/oleksii-nikolaienko/ExtDist

BugReports https://github.com/oleksii-nikolaienko/ExtDist/issues

Imports numDeriv, optimx

Suggests ggplot2, knitr, PerformanceAnalytics, stats, SuppDists, truncdist, VGAM, xtable, graphics, utils, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Encoding UTF-8

RoxygenNote 7.3.3

Author Haizhen Wu [aut], A. Jonathan R. Godfrey [aut], Kondaswamy Govindaraju [aut], Sarah Pirikahu [aut], Oleksii Nikolaienko [cre, ctb]

Maintainer Oleksii Nikolaienko <oleksii.nikolaienko@gmail.com>

Date/Publication 2025-09-23 07:50:02 UTC

Contents

`ExtDist-package` *Extended Probability Distribution Functions*

Description

The package provides a consistent, unified and extensible framework for parameter estimation of probability distributions; it extends parameter estimation procedures to allow for weighted samples; moreover, it extends the gallery of available distributions.

Details

Index:

```
Beta                  The standard Beta Distribution.
Beta_ab               The four-Parameter beta Distribution.
Burr                  The Burr's Distribution.
DistSelCriteriaValues Distribution Selection Criteria Values.
ExtDist-package       Extended Probability Distribution Functions
```

Gamma	The Gamma Distribution.
Gumbel	The Gumbel Distribution.
JohnsonSB	The Johnson SB Distribution.
JohnsonSU	The Johnson SU Distribution.
Laplace	The Laplace Distribution.
Logistic	The Logistic Distribution.
Normal	The Normal Distribution.
Normal_sym_trunc_ab	The symmetric truncated normal distribution.
Normal_trunc_ab	The truncated normal distribution.
SRTB_ab	The Symmetric-Reflected Truncated Beta (SRTB) Distribution.
SSRTB	The standard Symmetric-Reflected Truncated Beta (SRTB) Distribution.
Triangular	The Triangular Distribution.
Uniform	The Uniform Distribution.
Weibull	The Weibull Distribution.
bestDist	Best distribution for (weighted) sample.
compareDist	Compare sample and fitted distributions
eDist	S3 methods for manipulating eDist objects.
eval.estimation	Parameter Estimation Evaluation.
wmle	Weighted Maximum Likelihood Estimation.

Further information is available in the following vignettes:

Distributions-Beta	Distributions-Beta (source)
Distributions-Normal	Distributions-Normal (source)
Distributions	Distributions-Index (source)
ParaEst-and-DistSel-by-ExtDist	Parameter-Estimation-and-Distribution-Selection-by-ExtDist (source)

Author(s)

Haizhen Wu <h.wu2@massey.ac.nz>, A. Jonathan R. Godfrey <A.J.Godfrey@massey.ac.nz>, Kondaswamy Govindaraju <k.govindaraju@massey.ac.nz>, Sarah Pirikahu <s.pirikahu@massey.ac.nz>

Maintainer: Oleksii Nikolaienko <oleksii.nikolaienko@gmail.com>

---

bestDist *Finding the best distribution for a (weighted) sample.*

Description

This function chooses the best fitted distribution, based on a specified criterion.

Usage

```
bestDist(
  X,
  w = rep(1, length(X))/length(X),
  candDist = c("Beta_ab", "Laplace", "Normal"),
  criterion = c("AICc", "logLik", "AIC", "BIC", "MDL")
)
```

Arguments

X	Sample observations.
w	An optional vector of sample weights.
candDist	A vector of candidate distributions.
criterion	The basis on which the best fitted distribution is chosen.

Details

When comparing models fitted by maximum likelihood to the same data, the smaller the AIC, BIC or MDL, the better the fit. When comparing models using the log-likelihood criterion, the larger the log-likelihood the better the fit.

Value

An object of class character containing the name of the best distribution and its corresponding parameter estimates.

Note

The MDL criterion only works for parameter estimation by numerical maximum likelihood.

Author(s)

Haizhen Wu and A. Jonathan R. Godfrey.

Examples

```
X <- rBeta_ab(30, a = 0, b = 1, shape1 = 2, shape2 = 10)

# Determining the best distribution from the list of candidate distributions for the data X
Best.Dist <- bestDist(X, candDist = c("Laplace","Normal","Beta_ab"), criterion = "logLik")

# Printing the parameter estimates of the best distribution
attributes(Best.Dist)$best.dist.par
```

`Beta` *The Standard Beta Distribution.*

Description

Density, distribution, quantile, random number generation, and parameter estimation functions for the beta distribution with parameters `shape1` and `shape2`. Parameter estimation can be based on a weighted or unweighted i.i.d. sample and can be carried out analytically or numerically.

Usage

```
dBeta(x, shape1 = 2, shape2 = 3, params = list(shape1, shape2), ...)

pBeta(q, shape1 = 2, shape2 = 3, params = list(shape1, shape2), ...)

qBeta(p, shape1 = 2, shape2 = 3, params = list(shape1, shape2), ...)

rBeta(n, shape1 = 2, shape2 = 3, params = list(shape1, shape2), ...)

eBeta(X, w, method = c("MOM", "numerical.MLE"), ...)

lBeta(
  X,
  w,
  shape1 = 2,
  shape2 = 3,
  params = list(shape1, shape2),
  logL = TRUE,
  ...
)

sBeta(X, w, shape1 = 2, shape2 = 3, params = list(shape1, shape2), ...)

iBeta(X, w, shape1 = 2, shape2 = 3, params = list(shape1, shape2), ...)
```

Arguments

`x`, `q`	Vector of quantiles.
`shape1`, `shape2`	Shape parameters.
`params`	A list that includes all named parameters.
`...`	Additional parameters.
`p`	Vector of probabilities.
`n`	Number of observations.
`X`	Sample observations.
`w`	Optional vector of sample weights.

method	Parameter estimation method.
logL	logical, if TRUE lBeta gives the log-likelihood, otherwise the likelihood is given.

Details

The dBeta(), pBeta(), qBeta(),and rBeta() functions serve as wrappers of the standard dbeta, pbeta, qbeta, and rbeta functions in the **stats** package. They allow for the shape parameters to be declared not only as individual numerical values, but also as a list so parameter estimation can be carried out.

The beta distribution with parameters shape1=α and shape2=β is given by

$$f(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$$

where $0 \leq x \leq 1$, $\alpha > 0$, $\beta > 0$, and B is the beta function.

Analytical parameter estimation is conducted using the method of moments. The parameter estimates for α and β are as given in the Engineering Statistics Handbook.

The log-likelihood function of the beta distribution is given by

$$l(\alpha,\beta|x) = (\alpha-1)\sum_i ln(x_i) + (\beta-1)\sum_i ln(1-x_i) - lnB(\alpha,\beta).$$

Aryal & Nadarajah (2004) derived the score function and Fisher's information matrix for the 4-parameter beta function, from which the 2-parameter cases can be obtained.

Value

dBeta gives the density, pBeta the distribution function, qBeta the quantile function, rBeta generates random deviates, and eBeta estimates the parameters. lBeta provides the log-likelihood function, sBeta the observed score function, and iBeta the observed information matrix.

Author(s)

Haizhen Wu and A. Jonathan R. Godfrey.
Updates and bug fixes by Sarah Pirikahu.

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, volume 2, chapter 25, Wiley, New York.

Engineering Statistics Handbook

Bury, K. (1999) Statistical Distributions in Engineering, Chapter 14, pp.253-255, Cambridge University Press.

Aryal, G. and Nadarajah, S. (2004) Information Matrix for Beta Distributions, Serdica Math. J. 30, 513-526.

See Also

ExtDist for other standard distributions.

Examples

```
# Parameter estimation for a distribution with known shape parameters
x <- rBeta(n=500, params=list(shape1=2, shape2=2))
est.par <- eBeta(x); est.par
plot(est.par)

# Fitted density curve and histogram
dens <- dBeta(x=seq(0,1,length=100), params=list(shape1=2, shape2=2))
hist(x, breaks=10, probability=TRUE, ylim = c(0,1.2*max(dens)))
lines(seq(0,1,length=100), dens, col="blue")
lines(density(x), lty=2)

# Extracting shape parameters
est.par[attributes(est.par)$par.type=="shape"]

# Parameter estimation for a distribution with unknown shape parameters
# Example from; Bury(1999) pp.253-255, parameter estimates as given by Bury are
# shape1 = 4.222 and shape2 = 6.317
data <- c(0.461, 0.432, 0.237, 0.113, 0.526, 0.278, 0.275, 0.309, 0.67, 0.428, 0.556,
0.402, 0.472, 0.226, 0.632, 0.533, 0.309, 0.417, 0.495, 0.241)
est.par <- eBeta(X=data, method="numerical.MLE"); est.par
plot(est.par)

# Log-likelihood, score function, and observed information matrix
lBeta(data, param=est.par)
sBeta(data, param=est.par)
iBeta(data, param=est.par)

# Evaluating the precision of parameter estimation by the Hessian matrix.
H <- attributes(est.par)$nll.hessian;H
var <- solve(H)
se <- sqrt(diag(var)); se
```

Beta_ab *The four-parameter beta distribution.*

Description

Density, distribution, quantile, random number generation, and parameter estimation functions for the 4-parameter beta distribution. Parameter estimation can be based on a weighted or unweighted i.i.d sample and can be performed numerically.

Usage

```
dBeta_ab(
  x,
  shape1 = 2,
  shape2 = 3,
  a = 0,
  b = 1,
  params = list(shape1, shape2, a, b),
  ...
)

pBeta_ab(
  q,
  shape1 = 2,
  shape2 = 3,
  a = 0,
  b = 1,
  params = list(shape1 = 2, shape2 = 5, a = 0, b = 1),
  ...
)

qBeta_ab(
  p,
  shape1 = 2,
  shape2 = 3,
  a = 0,
  b = 1,
  params = list(shape1 = 2, shape2 = 5, a = 0, b = 1),
  ...
)

rBeta_ab(
  n,
  shape1 = 2,
  shape2 = 3,
  a = 0,
  b = 1,
  params = list(shape1, shape2, a, b),
  ...
)

eBeta_ab(X, w, method = "numerical.MLE", ...)

lBeta_ab(
  X,
  w,
  shape1 = 2,
  shape2 = 3,
```

```
  a = 0,
  b = 1,
  params = list(shape1, shape2, a, b),
  logL = TRUE,
  ...
)

sBeta_ab(
  X,
  w,
  shape1 = 2,
  shape2 = 3,
  a = 0,
  b = 1,
  params = list(shape1, shape2, a, b),
  ...
)
```

Arguments

x, q	A vector of quantiles.
shape1, shape2	Shape parameters.
a, b	Boundary parameters.
params	A list that includes all named parameters.
...	Additional parameters.
p	A vector of probabilities.
n	Number of observations.
X	Sample observations.
w	An optional vector of sample weights.
method	Parameter estimation method.
logL	logical; if TRUE, lBeta_ab gives the log-likelihood, otherwise the likelihood is given.

Details

The dBeta_ab(), pBeta_ab(), qBeta_ab(),and rBeta_ab() functions serve as wrappers of the standard dbeta, pbeta, qbeta and rbeta functions in the **stats** package.They allow for the parameters to be declared not only as individual numerical values, but also as a list so parameter estimation can be carried out.

The four-parameter beta distribution with parameters shape1=p, shape2=q, a = a and b=b has probability density function

$$f(x) = \frac{1}{B(p,q)} \frac{(x-a)^{(p-1)})(b-x)^{(q-1)}}{((b-a)^{(p+q-1)}))}$$

with $p > 0$, $q > 0$, $a \leq x \leq b$ and where B is the beta function, Johnson et.al (p.210).

The log-likelihood function of the four-parameter beta distribution is

$$l(p, q, a, b|x) = -lnB(p, q) + ((p-1)ln(x-a) + (q-1)ln(b-x)) - (p+q-1)ln(b-a).$$

Johnson et.al (p.226) provides the Fisher's information matrix of the four-parameter beta distribution in the regular case where $p, q > 2$.

Value

dBeta_ab gives the density, pBeta_ab the distribution function, qBeta_ab the quantile function, rBeta_ab generates random deviates, and eBeta_ab estimates the parameters. lBeta_ab provides the log-likelihood function, sBeta_ab the observed score function and iBeta_ab the observed information matrix.

Author(s)

Haizhen Wu and A. Jonathan R. Godfrey
Updates and bug fixes by Sarah Pirikahu.

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, volume 2, chapter 25, Wiley, New York.

Bury, K. (1999) Statistical Distributions in Engineering, Chapter 14, pp.261-262, Cambridge University Press.

See Also

ExtDist for other standard distributions.

Examples

```
# Parameter estimation for a distribution with known shape parameters
X <- rBeta_ab(n=500, shape1=2, shape2=5, a=1, b=2)
est.par <- eBeta_ab(X); est.par
plot(est.par)

# Fitted density curve and histogram
den.x <- seq(min(X),max(X),length=100)
den.y <- dBeta_ab(den.x,params = est.par)
hist(X, breaks=10, probability=TRUE, ylim = c(0,1.1*max(den.y)))
lines(den.x, den.y, col="blue")   # Original data
lines(density(X), lty=2)          # Fitted density curve

# Extracting boundary and shape parameters
est.par[attributes(est.par)$par.type=="boundary"]
est.par[attributes(est.par)$par.type=="shape"]
```

```
# Parameter Estimation for a distribution with unknown shape parameters
# Example from: Bury(1999) pp.261-262, parameter estimates as given by Bury are
# shape1 = 4.088, shape2 = 10.417, a = 1.279 and b = 2.407.
# The log-likelihood for this data and Bury's parameter estimates is 8.598672.
data <- c(1.73, 1.5, 1.56, 1.89, 1.54, 1.68, 1.39, 1.64, 1.49, 1.43, 1.68, 1.61, 1.62)
est.par <- eBeta_ab(X=data, method="numerical.MLE");est.par
plot(est.par)

# Estimates calculated by eBeta_ab differ from those given by Bury(1999).
# However, eBeta_ab's parameter estimates appear to be an improvement, due to a larger
# log-likelihood of 9.295922 (as given by lBeta_ab below).

# log-likelihood and score functions
lBeta_ab(data,param = est.par)
sBeta_ab(data,param = est.par)
```

Burr *The Burr Distribution.*

Description

Density, distribution, quantile, random number generation, and parameter estimation functions for the Burr distribution with parameters `location`, `scale` and `inequality`. Parameter estimation can be based on a weighted or unweighted i.i.d sample and can be performed numerically.

Usage

```
dBurr(x, b = 1, g = 2, s = 2, params = list(b = 1, g = 2, s = 2), ...)

pBurr(q, b = 1, g = 2, s = 2, params = list(b = 1, g = 2, s = 2), ...)

qBurr(p, b = 1, g = 2, s = 2, params = list(b = 1, g = 2, s = 2), ...)

rBurr(n, b = 1, g = 2, s = 2, params = list(b = 1, g = 2, s = 2), ...)

eBurr(X, w, method = "numerical.MLE", ...)

lBurr(
  X,
  w,
  b = 1,
  g = 2,
  s = 2,
  params = list(b = 1, g = 2, s = 2),
  logL = TRUE,
  ...
)
```

Arguments

x, q	A vector of quantiles.
b	Scale parameters.
g, s	Shape parameters.
params	A list that includes all named parameters.
...	Additional parameters.
p	A vector of probabilities.
n	Number of observations.
X	Sample observations.
w	An optional vector of sample weights.
method	Parameter estimation method.
logL	logical; if TRUE, lBurr gives the log-likelihood, otherwise the likelihood is given.

Details

The Burr distribution is a special case of the Pareto(IV) distribution where the `location` parameter is equal 0 and `inequality` parameter is equal to $1/g$, Brazauskas (2003).

The `dBurr()`, `pBurr()`, `qBurr()`,and `rBurr()` functions serve as wrappers of the `dparetoIV`, `pparetoIV`, `qparetoIV`, and `rparetoIV` functions in the **VGAM** package. They allow for the parameters to be declared not only as individual numerical values, but also as a list so parameter estimation can be carried out.

The Burr distribution is most simply defined in terms of its cumulative distribution function (Johnson et.al p.576)

$$F(x) = [1 + (x/b)^g]^{-s}$$

where b, g and $s > 0$. Parameter estimation can only be implemented numerically.

The log-likelihood and score functions are as given by Watkins (1999) and the information matrix is as given by Brazauskas (2003).

Value

dBurr gives the density, pBurr the distribution function, qBurr the quantile function, rBurr generates random deviates, and eBurr estimate the distribution parameters. lBurr provides the log-likelihood function.

Author(s)

Haizhen Wu and A. Jonathan R. Godfrey.
Updates and bug fixes by Sarah Pirikahu.

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1994) Continuous Univariate Distributions, volume 1, chapter 20, Wiley, New York.

Brazauskas, V. (2003) Information matrix for Pareto(IV), Burr, and related distributions. Comm. Statist. Theory and Methods 32, 315-325.

Watkins A.J. (1999) An algorithm for maximum likelihood estimation in the three parameter Burr XII distribution, Computational Statistics & Data Analysis, 32, 19-27.

Mathworks: Matlab documentation for Burr Type XII distribution

See Also

ExtDist for other standard distributions.

Examples

```
# Parameter estimation for a distribution of known shape parameters
X <- rBurr(n=500, b = 1, g = 2, s = 2)
est.par <- eBurr(X); est.par
plot(est.par)

# Fitted density curve and histogram
den.x <- seq(min(X),max(X),length=100)
den.y <- dBurr(den.x, b=est.par$b, g=est.par$g, s=est.par$s)
hist(X, breaks=10, probability=TRUE, ylim = c(0,1.1*max(den.y)))
lines(den.x, den.y, col="blue")
lines(density(X), lty=2)

# Extracting shape or scale parameters
est.par[attributes(est.par)$par.type=="scale"]
est.par[attributes(est.par)$par.type=="shape"]

# Parameter Estimation for a distribution with unknown shape parameters
# Example from: Matlab Statistical Toolbox package
# (See: https://au.mathworks.com/help/stats/burr-type-xii-distribution.html)
# Parameter estimates given are: b = 80.4515, g = 18.9251 and s = 0.4492.
QRS.duration <- c(91,81,138,100,88,100,77,78,84,89,102,77,78,91,77,75,82,70,91,82,83,90,71,75,82,
            109,94,95,90,96,85,71,75,78,82,69,103,85,80,94,80,79,92,84,86,73,75,73,78,80,81,
            83,103,92,88,77,79,90,91,83,80,78,76,82,81,80,82,71,73,87,76,101,93,90,87,88,94,
            94,90,78,83,92,93,100,83,163,96,114,170,137,84,82,79,72,97,87,102,85,84,78,79,91,
            98,86,72,97,82,78,97,94,82,78,79,87,93,75,106,96,88,90,74,85,90,71,75,77,87,95,
            74,99,89,83,78,100,80,87,79,102,80,85,81,85,95,82,97,92,102,86,80,85,85,111,89,63,
            70,92,75,93,83,84,91,81,113,92,81,74,78,80,82,95,106,95,100,90,88,71,78,77,87,79,
            85,91,92,98,68,84,92,110,108,153,73,81,87,87,95,73,95,100,96,97,76,62,86,71,99,68,
            90,146,86,80,90,93,91,111,89,79,77,73,92,98,78,87,98,84,82,90,85,71,84,85,77,93,
            74,89,89,103,85,88,81,84,96,90,98,78,93,80,85,67,74,69,105,95,87,108,99,79,86,82,
            91,93,80,84,90,81,90,78,98,81,90,85,79,61,90,79,83,84,78,86,72,87,91,102,80,82,104,
            85,83,81,94,84,91,99,101,132,79,103,97,131,91,90,121,78,84,97,94,96,91,80,97,92,90,
            90,123,105,85,77,83,92,85,96,69,88,84,107,91,74,89,109,80,83,92,100,113,105,99,84,
```

```
            74,76,87,87,96,88,80,85,90,74,95,86,74,95,74,73,104,92,105,97,101,83,84,98,81,93,
            84,102,94,91,100,92,94,98,146,84,77,82,84,76,106,70,87,118,86,82,96,89,93,82,97,
            86,188,93,72,107,81,76,83,147,82,110,108,82,93,95,80,185,73,78,71,86,85,76,93,
            87,96,86,78,87,80,98,75,78,82,94,83,94,140,87,55,133,83,77,123,79,88,80,88,79,
                  77,87,88,94,88,74,85,88,81,91,81,80,100,108,93,79)
est.par <- eBurr(QRS.duration); est.par
plot(est.par)

# log-likelihood function
lBurr(QRS.duration,param = est.par)

# Evaluation of the precision of the parameter estimates by the Hessian matrix
H <- attributes(est.par)$nll.hessian
var <- solve(H)
se <- sqrt(diag(var)); se
```

`compareDist` *Compare a sample to one or more fitted distributions*

Description

Compare a sample to one or more fitted distributions

Usage

```
compareDist(X, Dist1, Dist2 = NULL, Dist3 = NULL)
```

Arguments

`X`	An unweighted sample
`Dist1`, `Dist2`, `Dist3`	The fitted distribution, specified as either the objects of class eDist or names of the distribution to be fitted.

Value

compareDist returns an object of class histogram comparing the sample distribution to the specified fitted distribution(s).

Author(s)

Haizhen Wu and A. Jonathan R. Godfrey.

Examples

```
X <- rBeta(n=100, params=list(shape1=1, shape2=2))
compareDist(X, "Beta", "Normal", eNormal(X))
```

DistSelCriteria *Distribution Selection Criteria.*

Description

A function to calculate the distribution selection criteria for a list of candidate fits.

Usage

```
DistSelCriteria(
  X,
  w = rep(1, length(X))/length(X),
  candDist = c("Beta_ab", "Laplace", "Normal"),
  criteria = c("logLik", "AIC", "AICc", "BIC", "MDL")
)
```

Arguments

X	Sample obersevations.
w	An optional vector of sample weights.
candDist	A vector of names of candidate distributions.
criteria	A vector of criteria to be calculated.

Details

When comparing models fitted by maximum likelihood to the same data, the smaller the AIC, BIC or MDL, the better the fit. When comparing models using the log-likelihood criterion, the larger the log-likelihood the better the fit.

Value

An object of class matrix, containing the listed distribution selection criteria for the named distributions.

Note

The MDL criterion only works for parameter estimation by numerical maximum likelihood.

Author(s)

Haizhen Wu and A. Jonathan R. Godfrey.

Examples

```
Ozone <- airquality$Ozone
Ozone <- Ozone[!is.na(Ozone)] # Removing the NA's from Ozone data
DistSelCriteria(Ozone, candDist = c("Gamma", "Weibull", "Normal", "Exp"),
criteria = c("logLik","AIC","AICc", "BIC"))
```

eDist *S3 methods for manipulating eDist objects.*

Description

S3 methods for manipulating eDist objects

Usage

```
## S3 method for class 'eDist'
logLik(object, ...)

## S3 method for class 'eDist'
AIC(object, ..., k = 2)

AICc(object)

## S3 method for class 'eDist'
AICc(object, ...)

## S3 method for class 'eDist'
vcov(object, ..., corr = FALSE)

BIC(object)

## S3 method for class 'eDist'
BIC(object, ...)

MDL(object)

## S3 method for class 'eDist'
MDL(object, ...)

## S3 method for class 'eDist'
print(x, ...)

## S3 method for class 'eDist'
plot(x, ...)
```

Arguments

object	x An object of class eDist, usually the output of a parameter estimation function.
...	Additional parameters
k	numeric, The penalty per parameter to be used; the default k = 2 is the classical AIC.

corr	logical; should vcov() return correlation matrix (instead of variance-covariance matrix).
x	A list to be returned as class eDist.
plot	logical; if TRUE histogram, P-P and Q-Q plot of the distribution returned else only parameter estimation is returned.

Note

The MDL only works for parameter estimation by numerical maximum likelihood.

Author(s)

A. Jonathan R. Godfrey, Sarah Pirikahu, and Haizhen Wu.

References

Myung, I. (2000). The Importance of Complexity in Model Selection. Journal of mathematical psychology, 44(1), 190-204.

Examples

```
X <- rnorm(20)
est.par <- eNormal(X, method ="numerical.MLE")
logLik(est.par)
AIC(est.par)
AICc(est.par)
BIC(est.par)
MDL(est.par)
vcov(est.par)
vcov(est.par,corr=TRUE)
print(est.par)
plot(est.par)
```

---

eval.estimation *Parameter Estimation Evaluation.*

Description

A function to evaluate the parameter estimation function.

Usage

```
eval.estimation(
  rdist,
  edist,
  n = 20,
  rep.num = 1000,
  params,
  method = "numerical.MLE"
)
```

Arguments

rdist	Random variable generating function.
edist	Parameter estimation function.
n	Sample size.
rep.num	Number of replicates.
params	True parameters of the distribution.
method	Estimation method.

Value

A list containing the mean and sd of the estimated parameters.

na.cont returns the number of "na"s that appeared in the parameter estimation.

Author(s)

Haizhen Wu and A. Jonathan R. Godfrey.

Examples

```
eval.estimation(rdist = rBeta, edist = eBeta, n = 100, rep.num = 50,
params = list(shape1 = 1, shape2 = 5))
```

Exponential *The Exponential Distribution.*

Description

Density, distribution, quantile, random number generation and parameter estimation functions for the exponential distribution. Parameter estimation can be based on a weighted or unweighted i.i.d sample and is carried out analytically.

Usage

```
dExp(x, scale = 1, params = list(scale = 1), ...)

pExp(q, scale = 1, params = list(scale = 1), ...)

qExp(p, scale = 1, params = list(scale = 1), ...)

rExp(n, scale = 1, params = list(scale = 1), ...)

eExp(x, w, method = "analytical.MLE", ...)

lExp(x, w, scale = 1, params = list(scale = 1), logL = TRUE, ...)
```

```
sExp(x, w, scale = 1, params = list(scale = 1), ...)

iExp(x, w, scale = 1, params = list(scale = 1), ...)
```

Arguments

x, q	A vector of sample values or quantiles.
scale	scale parameter, called rate in other packages.
params	A list that includes all named parameters
...	Additional parameters.
p	A vector of probabilities.
n	Number of observations.
w	An optional vector of sample weights.
method	Parameter estimation method.
logL	logical; if TRUE, lExp gives the log-likelihood, otherwise the likelihood is given.

Details

If scale is omitted, it assumes the default value 1 giving the standard exponential distribution.

The exponential distribution is a special case of the gamma distribution where the shape parameter $\alpha = 1$. The dExp(), pExp(), qExp(),and rExp() functions serve as wrappers of the standard dexp, pexp, qexp and rexp functions in the **stats** package. They allow for the parameters to be declared not only as individual numerical values, but also as a list so parameter estimation can be carried out.

The probability density function for the exponential distribution with scale=β is

$$f(x) = (1/\beta) * exp(-x/\beta)$$

for $\beta > 0$, Johnson et.al (Chapter 19, p.494). Parameter estimation for the exponential distribution is carried out analytically using maximum likelihood estimation (p.506 Johnson et.al).

The likelihood function of the exponential distribution is given by

$$l(\lambda|x) = nlog\lambda - \lambda \sum xi.$$

It follows that the score function is given by

$$dl(\lambda|x)/d\lambda = n/\lambda - \sum xi$$

and Fisher's information given by

$$E[-d^2 l(\lambda|x)/d\lambda^2] = n/\lambda^2.$$

Value

dExp gives the density, pExp the distribution function, qExp the quantile function, rExp generates random deviates, and eExp estimates the distribution parameters. lExp provides the log-likelihood function.

Author(s)

Jonathan R. Godfrey and Sarah Pirikahu.

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, volume 1, chapter 19, Wiley, New York.

Kapadia. A.S., Chan, W. and Moye, L. (2005) Mathematical Statistics with Applications, Chapter 8, Chapman& Hall/CRC.

Examples

```
# Parameter estimation for a distribution with known shape parameters
x <- rExp(n=500, scale=2)
est.par <- eExp(x); est.par
plot(est.par)

#  Fitted density curve and histogram
den.x <- seq(min(x),max(x),length=100)
den.y <- dExp(den.x,scale=est.par$scale)
hist(x, breaks=10, probability=TRUE, ylim = c(0,1.1*max(den.y)))
lines(den.x, den.y, col="blue")
lines(density(x), lty=2)

# Extracting the scale parameter
est.par[attributes(est.par)$par.type=="scale"]

# Parameter estimation for a distribution with unknown shape parameters
# Example from Kapadia et.al(2005), pp.380-381.
# Parameter estimate as given by Kapadia et.al is scale=0.00277
cardio <- c(525, 719, 2880, 150, 30, 251, 45, 858, 15,
           47, 90, 56, 68, 6, 139, 180, 60, 60, 294, 747)
est.par <- eExp(cardio, method="analytical.MLE"); est.par
plot(est.par)

# log-likelihood, score function and Fisher's information
lExp(cardio,param = est.par)
sExp(cardio,param = est.par)
iExp(cardio,param = est.par)
```

Gamma *The Gamma Distribution.*

Description

Density, distribution, quantile, random number generation, and parameter estimation functions for the gamma distribution with parameters shape and scale. Parameter estimation can be based on a weighted or unweighted i.i.d sample and can be carried out numerically.

Usage

```
dGamma(x, shape = 2, scale = 2, params = list(shape = 2, scale = 2), ...)

pGamma(q, shape = 2, scale = 2, params = list(shape = 2, scale = 2), ...)

qGamma(p, shape = 2, scale = 2, params = list(shape = 2, scale = 2), ...)

rGamma(n, shape = 2, scale = 2, params = list(shape = 2, scale = 2), ...)

eGamma(X, w, method = c("moments", "numerical.MLE"), ...)

lGamma(
  X,
  w,
  shape = 2,
  scale = 2,
  params = list(shape = 2, scale = 2),
  logL = TRUE,
  ...
)
```

Arguments

x, q	A vector of quantiles.
shape	Shape parameter.
scale	Scale parameter.
params	A list that includes all named parameters
...	Additional parameters.
p	A vector of probabilities.
n	Number of observations.
X	Sample observations.
w	An optional vector of sample weights.
method	Parameter estimation method.
logL	logical; if TRUE, lBeta_ab gives the log-likelihood, otherwise the likelihood is given.

Details

The dGamma(), pGamma(), qGamma(),and rGamma() functions serve as wrappers of the standard dgamma, pgamma, qgamma, and rgamma functions in the **stats** package. They allow for the parameters to be declared not only as individual numerical values, but also as a list so parameter estimation can be carried out.

The gamma distribution with parameter shape=α and scale=β has probability density function,

$$f(x) = (1/\beta^{\alpha}\Gamma(\alpha))x^{\alpha-1}e^{-x/\beta}$$

where $\alpha > 0$ and $\beta > 0$. Parameter estimation can be performed using the method of moments as given by Johnson et.al (pp.356-357).

The log-likelihood function of the gamma distribution is given by,

$$l(\alpha, \beta|x) = (\alpha - 1)\sum_{i} ln(x_i) - \sum_{i}(x_i/\beta) - n\alpha ln(\beta) + nln\Gamma(\alpha)$$

where Γ is the gamma function. The score function is provided by Rice (2007), p.270.

Value

dGamma gives the density, pGamma the distribution function, qGamma the quantile function, rGamma generates random deviates, and eGamma estimates the distribution parameters.lgamma provides the log-likelihood function.

Author(s)

Haizhen Wu and A. Jonathan R. Godfrey.
Updates and bug fixes by Sarah Pirikahu, Oleksii Nikolaienko.

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, volume 1, chapter 17, Wiley, New York.

Bury, K. (1999) Statistical Distributions in Engineering, Chapter 13, pp.225-226, Cambridge University Press.

Rice, J.A. (2007) Mathematical Statistics and Data Analysis, 3rd Ed, Brookes/Cole.

See Also

ExtDist for other standard distributions.

Examples

```
# Parameter estimation for a distribution with known shape parameters
X <- rGamma(n=500, shape=1.5, scale=0.5)
est.par <- eGamma(X, method="numerical.MLE"); est.par
```

```
plot(est.par)

#  Fitted density curve and histogram
den.x <- seq(min(X),max(X),length=100)
den.y <- dGamma(den.x,shape=est.par$shape,scale=est.par$scale)
hist(X, breaks=10, probability=TRUE, ylim = c(0,1.1*max(den.y)))
lines(den.x, den.y, col="blue")
lines(density(X), lty=2)

# Extracting shape or scale parameters
est.par[attributes(est.par)$par.type=="shape"]
est.par[attributes(est.par)$par.type=="scale"]

# Parameter estimation for a distribution with unknown shape parameters
# Example from:  Bury(1999) pp.225-226, parameter estimates as given by Bury are
# shape = 6.40 and scale=2.54.
data <- c(16, 11.6, 19.9, 18.6, 18, 13.1, 29.1, 10.3, 12.2, 15.6, 12.7, 13.1,
         19.2, 19.5, 23, 6.7, 7.1, 14.3, 20.6, 25.6, 8.2, 34.4, 16.1, 10.2, 12.3)
est.par <- eGamma(data, method="numerical.MLE"); est.par
plot(est.par)

# log-likelihood
lGamma(data,param = est.par)

# Evaluating the precision of the parameter estimates by the Hessian matrix
H <- attributes(est.par)$nll.hessian
var <- solve(H)
se <- sqrt(diag(var));se
```

`Gumbel` *The Gumbel distribution*

Description

Density, distribution, quantile, random number generation, and parameter estimation functions for the Gumbel distribution with parameters `location` and `scale`. Parameter estimation can be based on a weighted or unweighted i.i.d sample and can be performed analytically or numerically.

Usage

```
dGumbel(
  x,
  location = 0,
  scale = 1,
  params = list(location = 0, scale = 1),
  ...
)

pGumbel(
```

```
  q,
  location = 0,
  scale = 1,
  params = list(location = 0, scale = 1),
  ...
)

qGumbel(
  p,
  location = 0,
  scale = 1,
  params = list(location = 0, scale = 1),
  ...
)

rGumbel(
  n,
  location = 0,
  scale = 1,
  params = list(location = 0, scale = 1),
  ...
)

eGumbel(X, w, method = c("moments", "numerical.MLE"), ...)

lGumbel(
  X,
  w,
  location = 0,
  scale = 1,
  params = list(location = 0, scale = 1),
  logL = TRUE,
  ...
)
```

Arguments

x, q	A vector of quantiles.
location	Location parameter.
scale	Scale parameter.
params	A list that includes all named parameters
...	Additional parameters.
p	A vector of probabilities.
n	Number of observations.
X	Sample observations.
w	An optional vector of sample weights.

method	Parameter estimation method.
logL	logical if TRUE, lGumbel gives the log-likelihood, otherwise the likelihood is given.

Details

The dGumbel(), pGumbel(), qGumbel(),and rGumbel() functions serve as wrappers of the dgumbel, pgumbel, qgumbel, and rgumbel functions in the **VGAM** package.They allow for the parameters to be declared not only as individual numerical values, but also as a list so parameter estimation can be carried out.

The Gumbel distribution is a special case of the generalised extreme value (GEV) distribution and has probability density function,

$$f(x) = exp(-exp-(x-\mu)/\sigma)$$

where μ = location and σ = scale which has the constraint $\sigma > 0$. The analytical parameter estimations are as given by the Engineering Statistics Handbook with corresponding standard errors given by Bury (p.273).

The log-likelihood function of the Gumbel distribution is given by

$$l(\mu, \sigma|x) = \sigma^{-n} exp(-\sum(x_i - \mu/\sigma) - \sum exp(-(x_i - \mu/\sigma))).$$

Shi (1995) provides the score function and Fishers information matrix.

Value

dGumbel gives the density, pGumbel the distribution function, qGumbel the quantile function, rGumbel generates random deviates, and eGumbel estimate the distribution parameters. lGumbel provides the log-likelihood function.

Author(s)

Haizhen Wu and A. Jonathan R. Godfrey.
Updates and bug fixes by Sarah Pirikahu.

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, volume 2, chapter 22, Wiley, New York.

Engineering Statistics Handbook.

Bury, K. (1999) Statistical Distributions in Engineering, Chapter 15, pp.283-284, Cambridge University Press.

Shi, D. (1995). Multivariate extreme value distribution and its Fisher information matrix. Acta Mathematicae

See Also

ExtDist for other standard distributions.

Examples

```
# Parameter estimation for a distribution with known shape parameters
X <- rGumbel(n = 500, location = 1.5, scale = 0.5)
est.par <- eGumbel(X, method="moments"); est.par
plot(est.par)

# Extracting location and scale parameters
est.par[attributes(est.par)$par.type=="location"]
est.par[attributes(est.par)$par.type=="scale"]

#  Fitted density curve and histogram
den.x <- seq(min(X),max(X),length=100)
den.y <- dGumbel(den.x, location = est.par$location, scale= est.par$scale)
hist(X, breaks=10, probability=TRUE, ylim = c(0,1.1*max(den.y)))
lines(den.x, den.y, col="blue")
lines(density(X))

# Parameter Estimation for a distribution with unknown shape parameters
# Example from; Bury(1999) pp.283-284, parameter estimates as given by Bury are location = 33.5
# and scale = 2.241
data <- c(32.7, 30.4, 31.8, 33.2, 33.8, 35.3, 34.6, 33, 32, 35.7, 35.5, 36.8, 40.8, 38.7, 36.7)
est.par <- eGumbel(X=data, method="numerical.MLE"); est.par
plot(est.par)

# log-likelihood
lGumbel(data, param = est.par)

# Evaluating the precision of the parameter estimates by the Hessian matrix
H <- attributes(est.par)$nll.hessian
var <- solve(H)
se <- sqrt(diag(var)); se
```

JohnsonSB *The Johnson SB distribution.*

Description

Density, distribution, quantile, random number generation, and parameter estimation functions for the Johnson SB (bounded support) distribution. Parameter estimation can be based on a weighted or unweighted i.i.d. sample and can be performed numerically.

Usage

```
dJohnsonSB(
  x,
```

```
  gamma = -0.5,
  delta = 2,
  xi = -0.5,
  lambda = 2,
  params = list(gamma = -0.5, delta = 2, xi = -0.5, lambda = 2),
  ...
)

dJohnsonSB_ab(
  x,
  gamma = -0.5,
  delta = 2,
  a = -0.5,
  b = 1.5,
  params = list(gamma = -0.5, delta = 2, a = -0.5, b = 1.5),
  ...
)

pJohnsonSB(
  q,
  gamma = -0.5,
  delta = 2,
  xi = -0.5,
  lambda = 2,
  params = list(gamma = -0.5, delta = 2, xi = -0.5, lambda = 2),
  ...
)

qJohnsonSB(
  p,
  gamma = -0.5,
  delta = 2,
  xi = -0.5,
  lambda = 2,
  params = list(gamma = -0.5, delta = 2, xi = -0.5, lambda = 2),
  ...
)

rJohnsonSB(
  n,
  gamma = -0.5,
  delta = 2,
  xi = -0.5,
  lambda = 2,
  params = list(gamma = -0.5, delta = 2, xi = -0.5, lambda = 2),
  ...
)
```

```
eJohnsonSB(X, w, method = "numerical.MLE", ...)

lJohnsonSB(
  X,
  w,
  gamma = -0.5,
  delta = 2,
  xi = -0.5,
  lambda = 2,
  params = list(gamma = -0.5, delta = 2, xi = -0.5, lambda = 2),
  logL = TRUE,
  ...
)
```

Arguments

x, q	A vector of quantiles.
gamma, delta	Shape parameters.
xi, lambda, a, b	Location-scale parameters.
params	A list that includes all named parameters.
...	Additional parameters.
p	A vector of probabilities.
n	Number of observations.
X	Sample observations.
w	An optional vector of sample weights.
method	Parameter estimation method.
logL	logical, it is assumed that the log-likelihood is desired. Set to FALSE if the likelihood is wanted.

Details

The Johnson system of distributions consists of families of distributions that, through specified transformations, can be reduced to the standard normal random variable. It provides a very flexible system for describing statistical distributions and is defined by

$$z = \gamma + \delta f(Y)$$

with $Y = (X - xi)/lambda$. The Johnson SB distribution arises when $f(Y) = ln[Y/(1 - Y)]$, where $0 < Y < 1$. This is the bounded Johnson family since the range of Y is $(0, 1)$, Karian & Dudewicz (2011).

The dJohnsonSB(), pJohnsonSB(), qJohnsonSB(),and rJohnsonSB() functions serve as wrappers of the dJohnson, pJohnson, qJohnson, and rJohnson functions in the **SuppDists** package. They allow for the parameters to be declared not only as individual numerical values, but also as a list so parameter estimation can be carried out.

The JohnsonSB distribution has probability density function

$$p_X(x) = \frac{\delta lambda}{\sqrt{2\pi}(x - xi)(1 - x + xi)} exp[-0.5(\gamma + \delta ln((x - xi)/(1 - x + xi)))^2].$$

Value

dJohnsonSB gives the density, pJohnsonSB the distribution function, qJohnsonSB gives quantile function, rJohnsonSB generates random deviates, and eJohnsonSB estimate the parameters. lJohnsonSB provides the log-likelihood function. The dJohnsonSB_ab provides an alternative parameterisation of the JohnsonSB distribution.

Author(s)

Haizhen Wu and A. Jonathan R. Godfrey.
Updates and bug fixes by Sarah Pirikahu.

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1994) Continuous Univariate Distributions, volume 1, chapter 12, Wiley, New York.

Kotz, S. and van Dorp, J. R. (2004). Beyond Beta: Other Continuous Families of Distributions with Bounded Support and Applications. Appendix B. World Scientific: Singapore.

Z. A. Karian and E. J. Dudewicz (2011) Handbook of Fitting Statistical Distributions with R, Chapman & Hall.

See Also

ExtDist for other standard distributions.

Examples

```
# Parameter estimation for a distribution with known shape parameters
X <- rJohnsonSB(n=500, gamma=-0.5, delta=2, xi=-0.5, lambda=2)
est.par <- eJohnsonSB(X); est.par
plot(est.par)

#  Fitted density curve and histogram
den.x <- seq(min(X),max(X),length=100)
den.y <- dJohnsonSB(den.x,params = est.par)
hist(X, breaks=10, probability=TRUE, ylim = c(0,1.2*max(den.y)))
lines(den.x, den.y, col="blue")
lines(density(X))

# Extracting location, scale and shape parameters
est.par[attributes(est.par)$par.type=="location"]
est.par[attributes(est.par)$par.type=="scale"]
```

```
est.par[attributes(est.par)$par.type=="shape"]

# Parameter Estimation for a distribution with unknown shape parameters
# Example from Karian, Z.A and Dudewicz, E.J. (2011) p.647.
# Original source of brain scan data Dudewich, E.J et.al (1989).
# Parameter estimates as given by Karian & Dudewicz using moments are:
# gamma =-0.2081, delta=0.9167, xi = 95.1280 and lambda = 21.4607 with log-likelihood = -67.03579
brain <- c(108.7, 107.0, 110.3, 110.0, 113.6, 99.2, 109.8, 104.5, 108.1, 107.2, 112.0, 115.5, 108.4,
           107.4, 113.4, 101.2, 98.4, 100.9, 100.0, 107.1, 108.7, 102.5, 103.3)
est.par <- eJohnsonSB(brain); est.par

# Estimates calculated by eJohnsonSB differ from those given by Karian & Dudewicz (2011).
# However, eJohnsonSB's parameter estimates appear to be an improvement, due to a larger
# log-likelihood of -66.35496 (as given by lJohnsonSB below).

# log-likelihood function
lJohnsonSB(brain, param = est.par)
```

JohnsonSU *The Johnson SU distribution.*

Description

Density, distribution, quantile, random number generation and parameter estimation functions for the Johnson SU (unbounded support) distribution. Parameter estimation can be based on a weighted or unweighted i.i.d sample and can be carried out numerically.

Usage

```
dJohnsonSU(
  x,
  gamma = -0.5,
  delta = 2,
  xi = -0.5,
  lambda = 2,
  params = list(gamma = -0.5, delta = 2, xi = -0.5, lambda = 2),
  ...
)

pJohnsonSU(
  q,
  gamma = -0.5,
  delta = 2,
  xi = -0.5,
  lambda = 2,
  params = list(gamma = -0.5, delta = 2, xi = -0.5, lambda = 2),
  ...
)
```

```
qJohnsonSU(
  p,
  gamma = -0.5,
  delta = 2,
  xi = -0.5,
  lambda = 2,
  params = list(gamma = -0.5, delta = 2, xi = -0.5, lambda = 2),
  ...
)

rJohnsonSU(
  n,
  gamma = -0.5,
  delta = 2,
  xi = -0.5,
  lambda = 2,
  params = list(gamma = -0.5, delta = 2, xi = -0.5, lambda = 2),
  ...
)

eJohnsonSU(X, w, method = "numerical.MLE", ...)

lJohnsonSU(
  X,
  w,
  gamma = -0.5,
  delta = 2,
  xi = -0.5,
  lambda = 2,
  params = list(gamma = -0.5, delta = 2, xi = -0.5, lambda = 2),
  logL = TRUE,
  ...
)
```

Arguments

x, q	A vector of quantiles.
gamma, delta	Shape parameters.
xi, lambda	Location-scale parameters.
params	A list that includes all named parameters.
...	Additional parameters.
p	A vector of probabilities.
n	Number of observations.
X	Sample observations.
w	An optional vector of sample weights.

`method`	Parameter estimation method.
`logL`	logical; if TRUE, lJohnsonSU gives the log-likelihood, otherwise the likelihood is given.

Details

The Johnson system of distributions consists of families of distributions that, through specified transformations, can be reduced to the standard normal random variable. It provides a very flexible system for describing statistical distributions and is defined by

$$z = \gamma + \delta f(Y)$$

with $Y = (X - xi)/lambda$. The Johnson SB distribution arises when $f(Y) = archsinh(Y)$, where $-\infty < Y < \infty$. This is the unbounded Johnson family since the range of Y is $(-\infty, \infty)$, Karian & Dudewicz (2011).

The JohnsonSU distribution has probability density function

$$p_X(x) = \frac{\delta}{\sqrt{2\pi((x - xi)^2 + lambda^2)}} exp[-0.5(\gamma + \delta ln(\frac{x - xi + \sqrt{(x - xi)^2 + lambda^2}}{lambda}))^2].$$

Parameter estimation can only be carried out numerically.

Value

dJohnsonSU gives the density, pJohnsonSU the distribution function, qJohnsonSU gives the quantile function, rJohnsonSU generates random variables, and eJohnsonSU estimates the parameters. lJohnsonSU provides the log-likelihood function.

Author(s)

Haizhen Wu and A. Jonathan R. Godfrey.
Updates and bug fixes by Sarah Pirikahu.

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1994) Continuous Univariate Distributions, volume 1, chapter 12, Wiley, New York.

Bowman, K.O., Shenton, L.R. (1983). Johnson's system of distributions. In: Encyclopedia of Statistical Sciences, Volume 4, S. Kotz and N.L. Johnson (eds.), pp. 303-314. John Wiley and Sons, New York.

Z. A. Karian and E. J. Dudewicz (2011) Handbook of Fitting Statistical Distributions with R, Chapman & Hall.

See Also

ExtDist for other standard distributions.

Examples

```
# Parameter estimation for a known distribution
X <- rJohnsonSU(n=500, gamma=-0.5, delta=2, xi=-0.5, lambda=2)
est.par <- eJohnsonSU(X); est.par
plot(est.par)

# Fitted density curve and histogram
den.x <- seq(min(X),max(X),length=100)
den.y <- dJohnsonSU(den.x,params = est.par)
hist(X, breaks=10, probability=TRUE, ylim = c(0,1.2*max(den.y)))
lines(den.x, den.y, col="blue")
lines(density(X), lty=2)

# Extracting shape and boundary parameters
est.par[attributes(est.par)$par.type=="shape"]
est.par[attributes(est.par)$par.type=="boundary"]

# Parameter Estimation for a distribution with unknown shape parameters
# Example from Karian, Z.A and Dudewicz, E.J. (2011) p.657.
# Parameter estimates as given by Karian & Dudewicz are:
# gamma =-0.2823, delta=1.0592, xi = -1.4475 and lambda = 4.2592  with log-likelihood = -277.1543
data <- c(1.99, -0.424, 5.61, -3.13, -2.24, -0.14, -3.32, -0.837, -1.98, -0.120,
          7.81, -3.13, 1.20, 1.54, -0.594, 1.05, 0.192, -3.83, -0.522, 0.605,
          0.427, 0.276, 0.784, -1.30, 0.542, -0.159, -1.66, -2.46, -1.81, -0.412,
          -9.67, 6.61, -0.589, -3.42, 0.036, 0.851, -1.34, -1.22, -1.47, -0.592,
          -0.311, 3.85, -4.92, -0.112, 4.22, 1.89, -0.382, 1.20, 3.21, -0.648,
          -0.523, -0.882, 0.306, -0.882, -0.635, 13.2, 0.463, -2.60, 0.281, 1.00,
          -0.336, -1.69, -0.484, -1.68, -0.131, -0.166, -0.266, 0.511, -0.198, 1.55,
          -1.03, 2.15, 0.495, 6.37, -0.714, -1.35, -1.55, -4.79, 4.36, -1.53,
          -1.51, -0.140, -1.10, -1.87, 0.095, 48.4, -0.998, -4.05, -37.9, -0.368,
          5.25, 1.09, 0.274, 0.684, -0.105, 20.3, 0.311, 0.621, 3.28, 1.56)
est.par <- eJohnsonSU(data); est.par
plot(est.par)

# Estimates calculated by eJohnsonSU differ from those given by Karian & Dudewicz (2011).
# However, eJohnsonSU's parameter estimates appear to be an improvement, due to a larger
# log-likelihood of -250.3208 (as given by lJohnsonSU below).

# log-likelihood function
lJohnsonSU(data, param = est.par)

# Evaluation of the precision using the Hessian matrix
H <- attributes(est.par)$nll.hessian
var <- solve(H)
se <- sqrt(diag(var)); se
```

Laplace *The Laplace Distribution.*

Description

Density, distribution, quantile, random number generation and parameter estimation functions for the Laplace distribution with `location` parameter μ and `scale` parameter b. Parameter estimation can for the Laplace distribution can be carried out numerically or analytically but may only be based on an unweighted i.i.d. sample.

Usage

```
dLaplace(x, mu = 0, b = 1, params = list(mu, b), ...)

pLaplace(q, mu = 0, b = 1, params = list(mu, b), ...)

qLaplace(p, mu = 0, b = 1, params = list(mu, b), ...)

rLaplace(n, mu = 0, b = 1, params = list(mu, b), ...)

eLaplace(X, w, method = c("analytic.MLE", "numerical.MLE"), ...)

lLaplace(x, w = 1, mu = 0, b = 1, params = list(mu, b), logL = TRUE, ...)
```

Arguments

`x, q`	A vector of quantiles.
`mu`	Location parameter.
`b`	Scale parameter.
`params`	A list that includes all named parameters
`...`	Additional parameters.
`p`	A vector of probabilities.
`n`	Number of observations.
`X`	Sample observations.
`w`	Optional vector of sample weights.
`method`	Parameter estimation method.
`logL`	logical; if TRUE, lLaplace gives the log-likelihood, otherwise the likelihood is given.

Details

The `dLaplace()`, `pLaplace()`, `qLaplace()`,and `rLaplace()` functions allow for the parameters to be declared not only as individual numerical values, but also as a list so parameter estimation can be carried out.

The Laplace distribution with parameters `location` = μ and `scale`=b has probability density function

$$f(x) = (1/2b)exp(-|x - \mu|/b)$$

where $-\infty < x < \infty$ and $b > 0$. The cumulative distribution function for `pLaplace` is defined by Johnson et.al (p.166).

Parameter estimation can be carried out analytically via maximum likelihood estimation, see Johnson et.al (p.172). Where the population mean, μ, is estimated using the sample median and b by the mean of $|x - b|$.

Johnson et.al (p.172) also provides the log-likelihood function for the Laplace distribution

$$l(\mu, b|x) = -nln(2b) - b^{-1} \sum |xi - \mu|.$$

Value

dLaplace gives the density, pLaplace the distribution function, qLaplace the quantile function, rLaplace generates random deviates, and eLaplace estimates the distribution parameters. lLaplace provides the log-likelihood function, sLaplace the score function, and iLaplace the observed information matrix.

Note

The estimation of the population mean is done using the median of the sample. Unweighted samples are not yet catered for in the eLaplace() function.

Author(s)

A. Jonathan R. Godfrey and Haizhen Wu.
Updates and bug fixes by Sarah Pirikahu

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, volume 2, chapter 24, Wiley, New York.

Best, D.J., Rayner, J.C.W. and Thas O. (2008) Comparison of some tests of fit for the Laplace distribution, Computational Statistics and Data Analysis, Vol. 52, pp.5338-5343.

Gumbel, E.J., Mustafi, C.K., 1967. Some analytical properties of bivariate extremal distributions. J. Am. Stat. Assoc. 62, 569-588

See Also

ExtDist for other standard distributions.

Examples

```
# Parameter estimation for a distribution with known shape parameters
X <- rLaplace(n=500, mu=1, b=2)
est.par <- eLaplace(X, method="analytic.MLE"); est.par
plot(est.par)

#  Fitted density curve and histogram
den.x <- seq(min(X),max(X),length=100)
```

```
den.y <- dLaplace(den.x, location = est.par$location, scale= est.par$scale)
hist(X, breaks=10, probability=TRUE, ylim = c(0,1.1*max(den.y)))
lines(den.x, den.y, col="blue")
lines(density(X), lty=2)

# Extracting location or scale parameters
est.par[attributes(est.par)$par.type=="location"]
est.par[attributes(est.par)$par.type=="scale"]

# Parameter estimation for a distribution with unknown shape parameters
# Example from Best et.al (2008). Original source of flood data from Gumbel & Mustafi.
# Parameter estimates as given by Best et.al mu=10.13 and  b=3.36
flood <- c(1.96, 1.96, 3.60, 3.80, 4.79, 5.66, 5.76, 5.78, 6.27, 6.30, 6.76, 7.65, 7.84, 7.99,
       8.51, 9.18, 10.13, 10.24, 10.25, 10.43, 11.45, 11.48, 11.75, 11.81, 12.34, 12.78, 13.06,
           13.29, 13.98, 14.18, 14.40, 16.22, 17.06)
est.par <- eLaplace(flood, method="numerical.MLE"); est.par
plot(est.par)

#log-likelihood function
lLaplace(flood,param=est.par)

# Evaluating the precision by the Hessian matrix
H <- attributes(est.par)$nll.hessian
var <- solve(H)
se <- sqrt(diag(var));se
```

`Logistic` *The Logistic Distribution.*

Description

Density, distribution, and quantile, random number generation, and parameter estimation functions for the logistic distribution with parameters `location` and `scale`. Parameter estimation can be based on a weighted or unweighted i.i.d. sample and can be carried out numerically.

Usage

```
dLogistic(
  x,
  location = 0,
  scale = 1,
  params = list(location = 0, scale = 1),
  ...
)

pLogistic(
  q,
  location = 0,
  scale = 1,
```

```
  params = list(location = 0, scale = 1),
  ...
)

qLogistic(
  p,
  location = 0,
  scale = 1,
  params = list(location = 0, scale = 1),
  ...
)

rLogistic(
  n,
  location = 0,
  scale = 1,
  params = list(location = 0, scale = 1),
  ...
)

eLogistic(X, w, method = "numerical.MLE", ...)

lLogistic(
  X,
  w,
  location = 0,
  scale = 1,
  params = list(location = 0, scale = 1),
  logL = TRUE,
  ...
)
```

Arguments

x, q	A vector of quantiles.
location	Location parameter.
scale	Scale parameter.
params	A list that includes all named parameters.
...	Additional parameters.
p	A vector of probabilities.
n	Number of observations.
X	Sample observations.
w	An optional vector of sample weights.
method	Parameter estimation method.
logL	logical; if TRUE, lLogistic gives the log-likelihood, otherwise the likelihood is given.

Details

If `location` or `scale` are omitted, they assume the default values of 0 or 1 respectively.

The `dLogistic()`, `pLogistic()`, `qLogistic()`,and `rLogistic()` functions serve as wrappers of the standard `dlogis`, `plogis`, `qlogis`, and `rlogis` functions in the **stats** package. They allow for the parameters to be declared not only as individual numerical values, but also as a list so parameter estimation can be carried out.

The logistic distribution with `location` = α and `scale` = β is most simply defined in terms of its cumulative distribution function (Johnson et.al pp.115-116)

$$F(x) = 1 - [1 + exp((x - \alpha)/\beta)]^{-1}.$$

The corresponding probability density function is given by

$$f(x) = 1/\beta[exp(x - \alpha/\beta][1 + exp(x - \alpha/\beta)]^{-2}$$

Parameter estimation is only implemented numerically.

The score function and Fishers information are as given by Shi (1995) (See also Kotz & Nadarajah (2000)).

Value

dLogistic gives the density, pLogistic the distribution function, qLogistic the quantile function, rLogistic generates random deviates, and eLogistic estimates the parameters. lLogistic provides the log-likelihood function.

Author(s)

Haizhen Wu and A. Jonathan R. Godfrey.
Updates and bug fixes by Sarah Pirikahu.

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, volume 2, chapter 23. Wiley, New York.

Shi, D. (1995) Fisher information for a multivariate extreme value distribution, Biometrika, vol 82, pp.644-649.

Kotz, S. and Nadarajah (2000) Extreme Value Distributions Theory and Applications, chapter 3, Imperial Collage Press, Singapore.

See Also

ExtDist for other standard distributions.

Examples

```
# Parameter estimation for a distribution with known shape parameters
X <- rLogistic(n=500, location=1.5, scale=0.5)
est.par <- eLogistic(X); est.par
plot(est.par)
#  Fitted density curve and histogram
den.x <- seq(min(X),max(X),length=100)
den.y <- dLogistic(den.x,location=est.par$location,scale=est.par$scale)
hist(X, breaks=10, probability=TRUE, ylim = c(0,1.2*max(den.y)))
lines(den.x, den.y, col="blue")
lines(density(X), lty=2)

# Extracting location or scale parameters
est.par[attributes(est.par)$par.type=="location"]
est.par[attributes(est.par)$par.type=="scale"]

# log-likelihood function
lLogistic(X,param = est.par)

# Evaluation of the precision of the parameter estimates by the Hessian matrix
H <- attributes(est.par)$nll.hessian
fisher_info <- solve(H)
var <- sqrt(diag(fisher_info));var

# Example of parameter estimation for a distribution with
# unknown parameters currently been sought after.
```

Normal *The Normal Distribution.*

Description

Density, distribution, quantile, random number generation and parameter estimation functions for the normal distribution. Parameter estimation can be based on a weighted or unweighted i.i.d. sample and can be carried out analytically or numerically.

Usage

```
dNormal(x, mean = 0, sd = 1, params = list(mean, sd), ...)

pNormal(q, mean = 0, sd = 1, params = list(mean, sd), ...)

qNormal(p, mean = 0, sd = 1, params = list(mean, sd), ...)

rNormal(n, mean = 0, sd = 1, params = list(mean, sd), ...)

eNormal(
  X,
  w,
```

```
  method = c("unbiased.MLE", "analytical.MLE", "numerical.MLE"),
  ...
)

lNormal(X, w, mean = 0, sd = 1, params = list(mean, sd), logL = TRUE, ...)

sNormal(X, w, mean = 0, sd = 1, params = list(mean, sd), ...)

iNormal(X, w, mean = 0, sd = 1, params = list(mean, sd), ...)
```

Arguments

x, q	Vector of quantiles.
mean	Location parameter.
sd	Scale parameter.
params	A list that includes all named parameters.
...	Additional parameters.
p	Vector of probabilities.
n	Number of observations.
X	Sample observations.
w	Optional vector of sample weights.
method	Parameter estimation method.
logL	logical; if TRUE, lNormal gives the log-likelihood, otherwise the likelihood is given.

Details

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

The dNormal(), pNormal(), qNormal(),and rNormal() functions serve as wrappers of the standard dnorm, pnorm, qnorm, and rnorm functions in the **stats** package. They allow for the parameters to be declared not only as individual numerical values, but also as a list so parameter estimation can be carried out.

The normal distribution has probability density function

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where μ is the mean of the distribution and σ is the standard deviation. The analytical unbiased parameter estimations are as given by Johnson et.al (Vol 1, pp.123-128).

The log-likelihood function of the normal distribution is given by

$$l(\mu, \sigma|x) = \sum_i [-0.5ln(2\pi) - ln(\sigma) - 0.5\sigma^{-2}(x_i - \mu)^2].$$

The score function and observed information matrix are as given by Casella & Berger (2nd Ed, pp.321-322).

Value

dNormal gives the density, pNormal gives the distribution function, qNormal gives the quantiles, rNormal generates random deviates, and eNormal estimates the parameters. lNormal provides the log-likelihood function, sNormal the score function, and iNormal the observed information matrix.

Author(s)

Haizhen Wu and A. Jonathan R. Godfrey.
Updates and bug fixes by Sarah Pirikahu.

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1994) Continuous Univariate Distributions, volume 1, chapter 13, Wiley, New York.

Casella, G. and Berger R. L. (2002) Statistical Inference, 2nd Ed, pp.321-322, Duxbury.

Bury, K. (1999) Statistical Distributions in Engineering, Chapter 10, p.143, Cambridge University Press.

See Also

ExtDist for other standard distributions.

Examples

```
# Parameter estimation for a distribution with known shape parameters
x <- rNormal(n=500, params=list(mean=1, sd=2))
est.par <- eNormal(X=x, method="unbiased.MLE"); est.par
plot(est.par)

#  Fitted density curve and histogram
den.x <- seq(min(x),max(x),length=100)
den.y <- dNormal(den.x, mean = est.par$mean, sd = est.par$sd)
hist(x, breaks=10, probability=TRUE, ylim = c(0,1.2*max(den.y)))
lines(lines(den.x, den.y, col="blue")) # Original data
lines(density(x), col="red")           # Fitted curve

# Extracting location and scale parameters
est.par[attributes(est.par)$par.type=="location"]
est.par[attributes(est.par)$par.type=="scale"]

# Parameter Estimation for a distribution with unknown shape parameters
# Example from: Bury(1999) p.143, parameter estimates as given by Bury are
# mu = 11.984 and sigma = 0.067
data <- c(12.065, 11.992, 11.992, 11.921, 11.954, 11.945, 12.029, 11.948, 11.885, 11.997,
         11.982, 12.109, 11.966, 12.081, 11.846, 12.007, 12.011)
est.par <- eNormal(X=data, method="numerical.MLE"); est.par
plot(est.par)

# log-likelihood, score function and observed information matrix
```

```
lNormal(data, param = est.par)
sNormal(data, param = est.par)
iNormal(data, param = est.par)

# Evaluating the precision of the parameter estimates by the Hessian matrix
H <- attributes(est.par)$nll.hessian; H
var <- solve(H)
se <- sqrt(diag(var)); se
```

Normal_sym_trunc_ab *The symmetric truncated normal distribution.*

Description

Density, distribution, quantile, random number generation and parameter estimation functions for the symmetric truncated normal distribution with parameters, `sigma`, `a` and `b` which represent the lower and upper truncation points respectively. Parameter estimation can be based on a weighted or unweighted i.i.d sample and can be carried out numerically.

Usage

```
dNormal_sym_trunc_ab(
  x,
  sigma = 0.3,
  a = 0,
  b = 1,
  params = list(sigma, a, b),
  ...
)

pNormal_sym_trunc_ab(
  q,
  sigma = 0.3,
  a = 0,
  b = 1,
  params = list(mu = 2, sigma = 5, a = 0, b = 1),
  ...
)

qNormal_sym_trunc_ab(
  p,
  sigma = 0.3,
  a = 0,
  b = 1,
  params = list(mu = 2, sigma = 5, a = 0, b = 1),
  ...
)
```

```
rNormal_sym_trunc_ab(
  n,
  mu = 2,
  sigma = 3,
  a = 0,
  b = 1,
  params = list(sigma, a, b),
  ...
)

eNormal_sym_trunc_ab(X, w, method = "numerical.MLE", ...)

lNormal_sym_trunc_ab(
  X,
  w,
  mu = 2,
  sigma = 3,
  a = 0,
  b = 1,
  params = list(sigma, a, b),
  logL = TRUE,
  ...
)
```

Arguments

x, q	A vector of quantiles.
a, b	Boundary parameters.
params	A list that includes all named parameters.
...	Additional parameters
p	A vector of probabilities.
n	Number of observations.
mu, sigma	Shape parameters.
X	Sample observations.
w	An optional vector of sample weights.
method	Parameter estimation method.
logL	logical;if TRUE, lNormal_sym_trunc_ab gives the log-likelihood, otherwise the likelihood is given.

Details

The normal symmetric truncated distribution is a special case of the trucated normal distribution. See Normal_trunc_ab.

Value

dNormal_sym_trunc_ab gives the density, pNormal_sym_trunc_ab the distribution function, qNormal_sym_trunc_ab the quantile function, rNormal_sym_trunc_ab generates random deviates,and eNormal_sym_trunc_ab estimates the parameters. lNormal_sym_trunc_ab provides the log-likelihood function.

Author(s)

Haizhen Wu and A. Jonathan R. Godfrey.

See Also

ExtDist for other standard distributions.

`Normal_trunc_ab` *The truncated normal distribution.*

Description

Density, distribution, quantile, random number generation and parameter estimation functions for the truncated normal distribution with parameters `mean`, `sd` and `a` and `b` which represent the lower and upper truncation points respectively. Parameter estimation can be based on a weighted or unweighted i.i.d. sample and is performed numerically.

Usage

```
dNormal_trunc_ab(
  x,
  mu = 0,
  sigma = 1,
  a = 0,
  b = 1,
  params = list(mu, sigma, a, b),
  ...
)

pNormal_trunc_ab(
  q,
  mu = 0,
  sigma = 1,
  a = 0,
  b = 1,
  params = list(mu, sigma, a, b),
  ...
)

qNormal_trunc_ab(
```

```
  p,
  mu = 0,
  sigma = 1,
  a = 0,
  b = 1,
  params = list(mu, sigma, a, b),
  ...
)

rNormal_trunc_ab(
  n,
  mu = 0,
  sigma = 1,
  a = 0,
  b = 1,
  params = list(mu, sigma, a, b),
  ...
)

eNormal_trunc_ab(X, w, method = "numerical.MLE", ...)

lNormal_trunc_ab(
  X,
  w,
  mu = 0,
  sigma = 1,
  a = 0,
  b = 1,
  params = list(mu, sigma, a, b),
  logL = TRUE,
  ...
)
```

Arguments

x, q	A vector of quantiles.
mu, sigma	Shape parameters.
a, b	Boundary parameters.
params	A list that includes all named parameters.
...	Additional parameters.
p	A vector of probabilities.
n	Number of observations.
X	Sample observations.
w	An optional vector of sample weights.
method	Parameter estimation method.
logL	logical;if TRUE, lNormal_trunc_ab gives the log-likelihood, otherwise the likelihood is given.

Details

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

The dNormal_trunc_ab(), pNormal_trunc_ab(), qNormal_trunc_ab(),and rNormal_trunc_ab() functions serve as wrappers of the dtrunc, ptrunc, qtrunc, and rtrunc functions in the **truncdist** package. They allow for the parameters to be declared not only as individual numerical values, but also as a list so parameter estimation can be carried out.

The probability density function of the doubly truncated normal distribution is given by

$$f(x) = \sigma^{-1} Z(x - \mu/\sigma)[\Phi(b - \mu/\sigma) - \Phi(a - \mu/\sigma)]^{-1}$$

where $\infty < a \leq x \leq b < \infty$. The degrees of truncation are $\Phi((a - \mu)/\sigma)$ from below and $1 - \Phi((a - \mu)/\sigma)$ from above. If a is replaced by $-\infty$, or b by ∞, the distribution is singly truncated, (Johnson et.al, p.156). The upper and lower limits of truncation a and b are normally known parameters whereas μ and σ may be unknown. Crain (1979) discusses parameter estimation for the truncated normal distribution and the method of numerical maximum likelihood estimation is used for parameter estimation in eNormal_trunc_ab.

Value

dNormal_trunc_ab gives the density, pNormal_trunc_ab the distribution function, qNormal_trunc_ab the quantile function, rNormal_trunc_ab generates random variables, and eNormal_trunc_ab estimates the parameters. lNormal_trunc_ab provides the log-likelihood function.

Author(s)

Haizhen Wu and A. Jonathan R. Godfrey.
Updates and bug fixes by Sarah Pirikahu.

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1994) Continuous Univariate Distributions, volume 1, chapter 13, Wiley, New York.

Crain, B.R (1979). Estimating the parameters of a truncated normal distribution, Applied Mathematics and Computation, vol 4, pp. 149-156

See Also

ExtDist for other standard distributions.

Examples

```
# Parameter estimation for a distribution with known shape parameters
X <- rNormal_trunc_ab(n=500, mu=2, sigma=5, a=-5, b=5)
est.par <- eNormal_trunc_ab(X); est.par
plot(est.par)
```

```
#  Fitted density curve and histogram
den.x <- seq(min(X),max(X),length=100)
den.y <- dNormal_trunc_ab(den.x,params=est.par)
hist(X, breaks=10, probability=TRUE, ylim=c(0,1.2*max(den.y)))
lines(den.x, den.y, col="blue")
lines(density(X), lty=2)

# Extracting boundary and shape parameters
est.par[attributes(est.par)$par.type=="boundary"]
est.par[attributes(est.par)$par.type=="shape"]

# log-likelihood function
lNormal_trunc_ab(X,param=est.par)
```

SRTB_ab *The symmetric-reflected truncated beta (SRTB) distribution.*

Description

Density, distribution, quantile, random number generation and parameter estimation functions for the SRTB distribution. Parameter estimation can be based on a weighted or unweighted i.i.d. sample and can be carried out numerically.

Usage

```
dSRTB_ab(
  x,
  shape1 = 2,
  shape2 = 3,
  a = 0,
  b = 1,
  params = list(shape1, shape2, a, b),
  ...
)

pSRTB_ab(
  q,
  shape1 = 2,
  shape2 = 3,
  a = 0,
  b = 1,
  params = list(shape1 = 2, shape2 = 5, a = 0, b = 1),
  ...
)

qSRTB_ab(
  p,
```

```
  shape1 = 2,
  shape2 = 3,
  a = 0,
  b = 1,
  params = list(shape1 = 2, shape2 = 5, a = 0, b = 1),
  ...
)

rSRTB_ab(
  n,
  shape1 = 2,
  shape2 = 3,
  a = 0,
  b = 1,
  params = list(shape1, shape2, a, b),
  ...
)

eSRTB_ab(X, w, method = "numerical.MLE", ...)

lSRTB_ab(
  X,
  w,
  shape1 = 2,
  shape2 = 3,
  a = 0,
  b = 1,
  params = list(shape1, shape2, a, b),
  logL = TRUE,
  ...
)
```

Arguments

x, q	A vector of quantiles.
shape1, shape2	Shape parameters.
a, b	Boundary parameters.
params	A list that includes all named parameters.
...	Additional parameters.
p	A vector of probabilities.
n	Number of observations.
X	Sample observations.
w	An optional vector of sample weights.
method	Parameter estimation method.
logL	logical; if TRUE, lSRTB_ab gives the log-likelihood, otherwise the likelihood is given.

Details

No details as of yet.

Value

dSRTB_ab gives the density, pSRTB_ab the distribution function, qSRTB_ab gives the quantile function, rSRTB_ab generates random variables, and eSRTB_ab estimates the parameters. lSRTB_ab provides the log-likelihood function and sSRTB_ab the score function.

Author(s)

Haizhen Wu.

See Also

ExtDist for other standard distributions.

Examples

```
# Parameter estimation for a distribution with known shape parameters
X <- rSRTB_ab(n=500, shape1=2, shape2=10, a=1, b=2)
est.par <- eSRTB_ab(X)
plot(est.par)

# Extracting boundary and shape parameters
est.par[attributes(est.par)$par.type=="boundary"]
est.par[attributes(est.par)$par.type=="shape"]

# log-likelihood function
lSRTB_ab(X,param = est.par)
```

SSRTB *The standard symmetric-reflected truncated beta (SSRTB) distribution.*

Description

Density, distribution, quantile, random number generation and parameter estimation functions for the SSRTB distribution. Parameter estimation can be based on a weighted or unweighted i.i.d sample and can be carried out numerically.

Usage

```
dSSRTB(x, shape1 = 2, shape2 = 3, params = list(shape1, shape2), ...)

pSSRTB(q, shape1 = 2, shape2 = 3, params = list(shape1, shape2), ...)

qSSRTB(p, shape1 = 2, shape2 = 3, params = list(shape1, shape2), ...)
```

```
rSSRTB(n, shape1 = 2, shape2 = 3, params = list(shape1, shape2), ...)

eSSRTB(X, w, method = "numerical.MLE", ...)

lSSRTB(
  X,
  w,
  shape1 = 2,
  shape2 = 3,
  params = list(shape1, shape2),
  logL = TRUE,
  ...
)
```

Arguments

x, q	A vector of quantiles.
shape1, shape2	Shape parameters.
params	A list that includes all named parameters.
...	Additional parameters.
p	A vector of probabilities.
n	Number of observations.
X	Sample observations.
w	An optional vector of sample weights.
method	Parameter estimation method.
logL	logical; if TRUE, lSSRTB gives the log-likelihood, otherwise the likelihood is given.

Details

No details as of yet.

Value

dSSRTB gives the density, pSSRTB the distribution function, qSSRTB the quantile function, rSSRTB generates random variables, eSSRTB estimates the parameters and lSSRTB provides the log-likelihood.

Author(s)

Haizhen Wu.

See Also

ExtDist for other standard distributions.

Examples

```
# Parameter estimation for a distribution with known shape parameters
X <- rSSRTB(n=500, shape1=2, shape2=10)
est.par <- eSSRTB(X); est.par
plot(est.par)

#  Fitted density curve and histogram
den.x <- seq(min(X),max(X),length=100)
den.y <- dSSRTB(den.x,shape1=est.par$shape1,shape2=est.par$shape2)
hist(X, breaks=10, probability=TRUE, ylim = c(0,1.2*max(den.y)))
lines(den.x, den.y, col="blue")
lines(density(X), lty=2)

# Extracting shape parameters
est.par[attributes(est.par)$par.type=="shape"]

# log-likelihood function
lSSRTB(X,param = est.par)
```

Triangular *The Triangular Distribution.*

Description

Density, distribution, quantile, random number generation and parameter estimation functions for the triangular distribution with support $[a, b]$ and shape parameter θ. Parameter estimation can be based on a weighted or unweighted i.i.d. sample and can be performed numerically.

Usage

```
dTriangular(x, a = 0, b = 1, theta = 0.5, params = list(a, b, theta), ...)

pTriangular(q, a = 0, b = 1, theta = 0.5, params = list(a, b, theta), ...)

qTriangular(p, a = 0, b = 1, theta = 0.5, params = list(a, b, theta), ...)

rTriangular(n, a = 0, b = 1, theta = 0.5, params = list(a, b, theta), ...)

eTriangular(X, w, method = "numerical.MLE", ...)

lTriangular(
  X,
  w,
  a = 0,
  b = 1,
  theta = 0.5,
  params = list(a, b, theta),
  logL = TRUE,
```

```
  ...
)
```

Arguments

x, q	A vector of quantiles.
a, b	Boundary parameters.
theta	Shape parameters.
params	A list that includes all named parameters.
...	Additional parameters.
p	A vector of probabilities.
n	Number of observations.
X	Sample observations.
w	An optional vector of sample weights.
method	Parameter estimation method.
logL	logical, it is assumed that the log-likelihood is desired. Set to FALSE if the likelihood is wanted.

Details

If a, b or theta are not specified they assume the default values of 0, 1 and 0.5 respectively.

The dTriangle(), pTriangle(), qTriangle(),and rTriangle() functions serve as wrappers of the dtriangle, ptriangle, qtriangle, and rtriangle functions in the **VGAM** package. They allow for the parameters to be declared not only as individual numerical values, but also as a list so parameter estimation can be carried out.

The triangular distribution has a probability density function, defined in Forbes et.al (2010), that consists of two lines joined at $theta$, where $theta$ is the location of the mode.

Value

dTriangular gives the density, pTriangular the distribution function, qTriangular the quantile function, rTriangular generates random variables, and eTriangular estimates the parameters. lTriangular provides the log-likelihood function.

Author(s)

Haizhen Wu and A. Jonathan R. Godfrey.
Updates and bug fixes by Sarah Pirikahu.

References

Kotz, S. and van Dorp, J. R. (2004). Beyond Beta: Other Continuous Families of Distributions with Bounded Support and Applications. Chapter 1. World Scientific: Singapore.

Forbes, C., Evans, M., Hastings, N. and Peacock, B. (2010) Triangular Distribution, in Statistical Distributions, Fourth Edition, John Wiley & Sons, Inc., Hoboken, NJ, USA.

See Also

ExtDist for other standard distributions.

`Uniform` *The Uniform Distribution.*

Description

Density, distribution, quantile, random number generation and parameter estimation functions for the uniform distribution on the interval $[a, b]$. Parameter estimation can be based on an unweighted i.i.d. sample only and can be performed analytically or numerically.

Usage

```
dUniform(x, a = 0, b = 1, params = list(a, b), ...)

pUniform(q, a = 0, b = 1, params = list(a, b), ...)

qUniform(p, a = 0, b = 1, params = list(a, b), ...)

rUniform(n, a = 0, b = 1, params = list(a, b), ...)

eUniform(X, w, method = c("analytic.MLE", "moments", "numerical.MLE"), ...)

lUniform(X, w, a = 0, b = 1, params = list(a, b), logL = TRUE, ...)
```

Arguments

`x`, `q`	A vector of quantiles.
`a`, `b`	Boundary parameters.
`params`	A list that includes all named parameters.
`...`	Additional parameters.
`p`	A vector of probabilities.
`n`	Number of observations.
`X`	Sample observations.
`w`	An optional vector of sample weights.
`method`	Parameter estimation method.
`logL`	logical;if TRUE, lUniform gives the log-likelihood, otherwise the likelihood is given.

Details

If a or b are not specified they assume the default values of 0 and 1, respectively.

The dUniform(), pUniform(), qUniform(),and rUniform() functions serve as wrappers of the standard dunif, punif, qunif, and runif functions in the **stats** package. They allow for the parameters to be declared not only as individual numerical values, but also as a list so parameter estimation can be carried out.

The uniform distribution has probability density function

$$p_x(x) = 1/(b-a)$$

for $a \leq x \leq b$. The analytic maximum likelihood parameter estimates are as given by Engineering Statistics Handbook. The method of moments parameter estimation option is also avaliable and the estimates are as given by Forbes et.al (2011), p.179.

The log-likelihood function for the uniform distribution is given by

$$l(a, b|x) = -nlog(b-a)$$

Value

dUniform gives the density, pUniform the distribution function, qUniform the quantile function, rUniform generates random deviates, and eUniform estimates the parameters. lUniform provides the log-likelihood function.

Note

The analytical maximum likelihood estimation of the parameters a and b is calculated using the range and mid-range of the sample. Therefore, only unweighted samples are catered for in the eUniform distribution when the method analytic.MLE is selected.

Author(s)

Haizhen Wu and A. Jonathan R. Godfrey.
Updates and bugfixes by Sarah Pirikahu.

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, volume 2, chapter 26, Wiley, New York.

Engineering Statistics Handbook

Forbes, C. Evans, M. Hastings, N. & Peacock, B. (2011) Statistical Distributions, 4th Ed, chapter 40, Wiley, New Jersey.

See Also

ExtDist for other standard distributions.

Examples

```
# Parameter estimation for a distribution with known shape parameters
X <- rUniform(n=500, a=0, b=1)
est.par <- eUniform(X, method="analytic.MLE"); est.par
plot(est.par)

# Histogram and fitted density
den.x <- seq(min(X),max(X),length=100)
den.y <- dUniform(den.x,a=est.par$a,b=est.par$b)
hist(X, breaks=10, probability=TRUE, ylim = c(0,1.2*max(den.y)))
lines(den.x, den.y, col="blue")  # Original data
lines(density(X), lty=2)         # Fitted curve

# Extracting boundary parameters
est.par[attributes(est.par)$par.type=="boundary"]

# log-likelihood
lUniform(X,param = est.par)

# Example of parameter estimation for a distribution with
# unknown parameters currently been sought after.
```

`Weibull` *The Weibull Distribution.*

Description

Density, distribution, quantile, random number generation, and parameter estimation functions for the Weibull distribution with parameters `shape` and `scale`. Parameter estimation can be based on a weighted or unweighted i.i.d sample and can be carried out analytically or numerically.

Usage

```
dWeibull(x, shape = 2, scale = 2, params = list(shape = 2, scale = 2))

pWeibull(q, shape = 2, scale = 2, params = list(shape = 2, scale = 2))

qWeibull(p, shape = 2, scale = 2, params = list(shape = 2, scale = 2))

rWeibull(n, shape = 2, scale = 2, params = list(shape = 2, scale = 2))

eWeibull(X, w, method = c("numerical.MLE", "moments"), ...)

lWeibull(
  X,
  w,
  shape = 2,
  scale = 2,
```

```
  params = list(shape = 2, scale = 2),
  logL = TRUE
)
```

Arguments

x, q	A vector of quantiles.
shape	Shape parameter.
scale	Scale parameter.
params	A list that includes all named parameters
p	A vector of probabilities.
n	Number of observations.
X	Sample observations.
w	An optional vector of sample weights.
method	Parameter estimation method.
...	Additional parameters.
logL	logical; if TRUE, lWeibull gives the log-likelihood, otherwise the likelihood is given.

Details

The Weibull distribution is a special case of the generalised gamma distribution. The `dWeibull()`, `pWeibull()`, `qWeibull()`,and `rWeibull()` functions serve as wrappers of the standard `dweibull`, `pweibull`, `qweibull`, and `rweibull` functions with in the **stats** package. They allow for the parameters to be declared not only as individual numerical values, but also as a list so parameter estimation can be carried out.

The Weibull distribution with parameters `shape`=a and `scale`=b has probability density function,

$$f(x) = (a/b)(x/b)^{a-1}exp(-(x/b)^a)$$

for $x > 0$. Parameter estimation can be carried out using the method of moments as done by Winston (2003) or by numerical maximum likelihood estimation.

The log-likelihood function of the Weibull distribution is given by

$$l(a, b|x) = n(lna - lnb) + (a - 1)\sum ln(xi/b) - \sum (xi/b)^a$$

The score function and information matrix are as given by Rinne (p.412).

Value

dWeibull gives the density, pWeibull the distribution function, qWeibull the quantile function, rWeibull generates random deviates, and eWeibull estimates the distribution parameters. lWeibull provides the log-likelihood function.

Author(s)

Haizhen Wu and A. Jonathan R. Godfrey.
Updates and bug fixes by Sarah Pirikahu, Oleksii Nikolaienko.

References

Johnson, N. L., Kotz, S. and Balakrishnan, N. (1995) Continuous Univariate Distributions, volume 1, chapter 21, Wiley, New York.

Rinne, H. (2009) The Weibull Distribution A Handbook, chapter 11, Chapman & Hall/CRC.

Winston, W.L (2003) Operations Research: Applications and algorithms, 4th Ed, Duxbury.

See Also

ExtDist for other standard distributions.

Examples

```
# Parameter estimation for a distribution with known shape parameters
X <- rWeibull(n=1000, params=list(shape=1.5, scale=0.5))
est.par <- eWeibull(X=X, method="numerical.MLE"); est.par
plot(est.par)

#  Fitted density curve and histogram
den.x <- seq(min(X),max(X),length=100)
den.y <- dWeibull(den.x,shape=est.par$shape,scale=est.par$scale)
hist(X, breaks=10, col="red", probability=TRUE, ylim = c(0,1.1*max(den.y)))
lines(den.x, den.y, col="blue", lwd=2)   # Original data
lines(density(X), lty=2)                 # Fitted curve

# Extracting shape and scale parameters
est.par[attributes(est.par)$par.type=="shape"]
est.par[attributes(est.par)$par.type=="scale"]

# Parameter Estimation for a distribution with unknown shape parameters
# Example from: Rinne (2009) Dataset p.338 and example pp.418-419
# Parameter estimates are given as shape = 2.5957 and scale = 99.2079.
data <- c(35,38,42,56,58,61,63,76,81,83,86,90,99,104,113,114,117,119,141,183)
est.par <- eWeibull(X=data, method="numerical.MLE"); est.par
plot(est.par)

# log-likelihood function
lWeibull(data, param = est.par)

# evaluate the precision of estimation by Hessian matrix
H <- attributes(est.par)$nll.hessian
var <- solve(H)
se <- sqrt(diag(var));se
```

wmle *Weighted Maximum Likelihood Estimation.*

Description

A general weighted maximum likelihood estimation function.

Usage

```
wmle(X, w, distname, initial, lower, upper, loglik.fn, score.fn, obs.info.fn)
```

Arguments

X	Sample observations.
w	Frequency (or weights) of observation.
distname	Name of distribution to be estimated.
initial	Initial value of the parameters.
lower	The lower bound of the parameters.
upper	The upper bound of the parameters.
loglik.fn	Function to compute (weighted) log likelihood.
score.fn	Function to compute (weighted) score.
obs.info.fn	Function to compute observed information matrix.

Details

Weighted Maximum Likelihood Estimation

Value

weighted mle estimates.

Author(s)

Haizhen Wu and A. Jonathan R. Godfrey

Index