

Package: gkwdist (via r-universe)

September 27, 2026

Title Generalized Kumaraswamy Distribution Family

Version 1.1.7

Description Implements the five-parameter Generalized Kumaraswamy ('gkw') distribution proposed by 'Carrasco, Ferrari and Cordeiro (2010)' <[doi:10.48550/arXiv.1004.0911](https://doi.org/10.48550/arXiv.1004.0911)> and its seven nested sub-families for modeling bounded continuous data on the unit interval (0,1). The 'gkw' distribution extends the Kumaraswamy distribution described by Jones (2009) <[doi:10.1016/j.stamet.2008.04.001](https://doi.org/10.1016/j.stamet.2008.04.001)>. Provides density, distribution, quantile, and random generation functions, along with analytical log-likelihood, gradient, and Hessian functions implemented in 'C++' via 'RcppArmadillo' for maximum computational efficiency. Suitable for modeling proportions, rates, percentages, and indices exhibiting complex features such as asymmetry, or heavy tails and other shapes not adequately captured by standard distributions like simple Beta or Kumaraswamy.

License MIT + file LICENSE

URL <https://github.com/evandeilton/gkwdist>,
<https://evandeilton.github.io/gkwdist/>

BugReports <https://github.com/evandeilton/gkwdist/issues>

Encoding UTF-8

Language en-US

Roxygen list(markdown = TRUE)

Imports Rcpp, magrittr

LinkingTo Rcpp, RcppArmadillo

Suggests knitr, microbenchmark, numDeriv, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation yes

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

Repository <https://evandeilton.r-universe.dev>

Date/Publication 2026-09-27 02:12:56 UTC

RemoteUrl <https://github.com/evandeilton/gkwdist>

RemoteRef HEAD

RemoteSha 0c18142fae193790486e0343d7f137b92c6bc640

Contents

gkwdist-package	3
dbeta_	8
dbkw	10
dekw	11
dgkw	13
dkkw	15
dkw	17
dmc	18
gkwgetstartvalues	20
grbeta	22
grbkw	24
grekw	26
grgkw	28
grkkw	30
grkw	32
grmc	33
hsbeta	35
hsbkw	37
hsekw	39
hsgkw	41
hskkw	42
hskw	44
hsmc	46
llbeta	48
llbkw	50
llekw	52
llgkw	53
llkkw	55
llkw	57
llmc	59
pbeta_	61
pbkw	62
pekw	64
pgkw	66
pkkw	68
pkw	70
pmc	71

qbeta_	73
qbkw	75
qekw	77
qgkw	78
qkkw	81
qkw	83
qmc	84
rbeta_	86
rbkw	87
rekw	89
rgkw	91
rkkw	92
rkw	94
rmc	95
Index	98

gkwdist-package	<i>Generalized Kumaraswamy Distribution Family</i>
-----------------	--

Description

The **gkwdist** package provides a comprehensive, high-performance implementation of the five-parameter Generalized Kumaraswamy (GKw) distribution and its seven nested sub-families for modeling bounded continuous data on the unit interval $(0, 1)$. All distribution and likelihood functions are implemented in **C++** via **RcppArmadillo** for maximum computational efficiency.

Details

Overview: The Generalized Kumaraswamy distribution, proposed by Carrasco, Ferrari and Cordeiro (2010), extends the Kumaraswamy distribution (Jones, 2009) by incorporating three additional shape parameters. This generalization provides remarkable flexibility in accommodating various density shapes including unimodality, asymmetry, J-shapes, inverted J-shapes, U-shapes, bathtub shapes, and heavy or light tails.

That flexibility stops short of bimodality. Across roughly 330,000 parameter vectors spanning $(10^{-3}, 300)$ in each of the five parameters, no density with two interior modes was found: every shape was monotone, unimodal, or U-shaped. A second peak arises only as a divergence at $x = 0$ or $x = 1$ — see the boundary limits documented in [dgkw](#) — never as a second interior hump. Data with two separated interior modes calls for a mixture, not a larger member of this family.

Mathematical Specification: The probability density function (PDF) of the GKw distribution with parameters $\theta = (\alpha, \beta, \gamma, \delta, \lambda)$ (all positive) is:

$$f(x; \theta) = \frac{\lambda \alpha \beta x^{\alpha-1}}{B(\gamma, \delta + 1)} (1 - x^\alpha)^{\beta-1} [1 - (1 - x^\alpha)^\beta]^{\gamma\lambda-1} \{1 - [1 - (1 - x^\alpha)^\beta]^\lambda\}^\delta$$

for $0 < x < 1$, where $B(a, b) = \Gamma(a)\Gamma(b)/\Gamma(a + b)$ is the beta function.

The cumulative distribution function (CDF) is:

$$F(x; \theta) = I_{[1-(1-x^\alpha)^\beta]^\lambda}(\gamma, \delta + 1)$$

where $I_y(a, b)$ denotes the regularized incomplete beta function ratio.

Distribution Family Hierarchy: The GKw distribution nests seven important sub-models:

Beta-Kumaraswamy (BKw) 4 parameters: $(\alpha, \beta, \gamma, \delta)$. Set $\lambda = 1$ in GKw.

Kumaraswamy-Kumaraswamy (KKw) 4 parameters: $(\alpha, \beta, \delta, \lambda)$. Set $\gamma = 1$ in GKw.

Exponentiated Kumaraswamy (EKw) 3 parameters: (α, β, λ) . Set $\gamma = 1, \delta = 0$ in GKw. Has closed-form quantile function.

McDonald (Mc) 3 parameters: $(\gamma, \delta, \lambda)$. Set $\alpha = \beta = 1$ in GKw. Also known as Beta-Power distribution.

Kumaraswamy (Kw) 2 parameters: (α, β) . Set $\gamma = 1, \delta = 0$ and $\lambda = 1$ in GKw. Has closed-form CDF and quantile function.

Beta 2 parameters: (γ, δ) . Set $\alpha = \beta = \lambda = 1$ in GKw. Classical beta distribution with shape1 = gamma, shape2 = delta + 1.

Uniform 0 parameters. Set $\alpha = \beta = \gamma = \lambda = 1$ and $\delta = 0$ in GKw. Note that $\delta = 0$, not $\delta = 1$, is the neutral value: δ enters the density through $B(\gamma, \delta + 1)$ and z^δ .

Distribution Functions

The package provides the standard R distribution function API for all seven distributions. Each distribution has four core functions with prefix d, p, q, r:

Generalized Kumaraswamy (GKw):

- **dgkw:** Density function
- **pgkw:** Distribution function (CDF)
- **qgkw:** Quantile function (inverse CDF)
- **rgkw:** Random generation

Beta-Kumaraswamy (BKw): dbkw, pbkw, qbkw, rbkw

Kumaraswamy-Kumaraswamy (KKw): dkkw, pkkw, qkkw, rkkw

Exponentiated Kumaraswamy (EKw): dekw, pekw, qekw, rekw

McDonald (Mc): dmc, pmc, qmc, rmc

Kumaraswamy (Kw): dkw, pkw, qkw, rkw

Beta: dbeta_, pbeta_, qbeta_, rbeta_

All distribution functions are implemented in C++ for optimal performance.

Likelihood Functions

High-performance analytical log-likelihood, gradient, and Hessian functions are provided for maximum likelihood estimation. These functions return **negative** values to facilitate direct use with optimization routines like **optim**, which perform minimization by default.

Function signature: `function(par, data)` where `par` is a numeric vector of parameters and `data` is the observed sample.

ll* Negative log-likelihood: $-\ell(\theta) = -\sum_{i=1}^n \log f(x_i; \theta)$

gr* Negative gradient (negative score vector): $-\nabla_{\theta} \ell(\theta)$

hs* Negative Hessian matrix: $-\nabla_{\theta}^2 \ell(\theta)$

Available for all distributions:

- **GKw:** [llgkw](#), [grgkw](#), [hsgkw](#)
- **BKw:** [llbkw](#), [grbkw](#), [hsbkw](#)
- **KKw:** [llkkw](#), [grkkw](#), [hskkw](#)
- **EKw:** [llekw](#), [grekw](#), [hsekw](#)
- **Mc:** [llmc](#), [grmc](#), [hsmc](#)
- **Kw:** [llkw](#), [grkw](#), [hskw](#)
- **Beta:** [llbeta](#), [grbeta](#), [hsbeta](#)

These analytical derivatives are exact (up to floating-point precision) and provide substantial performance gains over numerical approximations.

Utility Functions

- [gkwgetstartvalues](#): Compute starting values for optimization using method of moments or quantile matching

Applications

The GKw distribution family is particularly suitable for modeling:

- **Proportions and rates:** Bounded continuous data in $(0, 1)$
- **Percentages:** Economic indices, market shares, conversion rates
- **Fractions:** Parts of a whole, concentration measures
- **Scores and indices:** Normalized measurements, standardized tests
- **Reliability data:** Component lifetimes on bounded domains
- **Hydrological phenomena:** Reservoir levels, soil moisture content
- **Financial ratios:** Debt-to-equity, current ratio, profit margins
- **Quality metrics:** Defect rates, efficiency scores, purity levels
- **Biostatistics:** Survival probabilities, dose-response curves

Advantages Over Standard Distributions

Compared to the classical Beta distribution, the GKw family offers:

- Greater flexibility in density shapes (U-shaped, bathtub, J and reverse J)
- Better accommodation of asymmetry and tail behavior
- Closed-form CDF and quantile for some sub-models (Kw, EKw, KKw)
- Computational efficiency via C++ implementation
- Easy parameter interpretation through nested structure
- Superior performance for extreme parameter values

Performance

All functions are implemented in C++ using **RcppArmadillo**, providing:

- 10-100× speedup over pure R implementations
- Linear scaling with sample size
- Optimized memory allocation
- Numerical stability for extreme parameter values
- Efficient vectorized operations

Typical benchmarks on modern hardware:

- Density evaluation: $> 10^7$ evaluations per second
- Log-likelihood: $n = 10^6$ observations in < 100 ms
- Gradient computation: $< 5\times$ slower than log-likelihood

Model Selection Workflow

Recommended strategy for choosing among distributions:

1. **Exploratory Analysis:** Examine histograms, kernel density estimates, and summary statistics of your data.
2. **Start Simple:** Fit Beta and Kumaraswamy distributions (2 parameters). Use `optim` with `method = "BFGS"` and analytical gradients.
3. **Diagnostic Checking:** Assess fit quality using Q-Q plots, probability plots, and goodness-of-fit tests (e.g., Kolmogorov-Smirnov).
4. **Progressive Complexity:** If simple models inadequate, try 3-parameter models (EKw or Mc), then 4-parameter models (BKw or KKw).
5. **Information Criteria:** Use AIC, BIC, or AICc to balance goodness-of-fit and model parsimony. Lower values indicate better models.
6. **Nested Testing:** Perform likelihood ratio tests when comparing nested models (e.g., Kw vs. EKw).
7. **Cross-Validation:** Validate final model on held-out data or via bootstrap procedures.
8. **Residual Analysis:** Examine probability integral transform residuals for uniformity and independence.

Statistical Inference

Maximum likelihood estimation is performed using numerical optimization:

1. Obtain starting values via `gkwgetstartvalues` or manual specification based on sample moments
2. Minimize negative log-likelihood using `optim` with `method = "BFGS"` or `"L-BFGS-B"`
3. Provide analytical gradient via `gr` argument for faster convergence and improved accuracy
4. Set `hessian = TRUE` to obtain observed information matrix
5. Compute standard errors as `sqrt(diag(solve(hessian)))`

6. Construct confidence intervals using normal approximation or profile likelihood
7. Perform hypothesis tests using Wald, score, or likelihood ratio statistics

For large samples ($n > 10^4$), consider using method = "L-BFGS-B" with box constraints to prevent parameter estimates from drifting to boundary values or becoming numerically unstable.

Author(s)

Lopes, J. E. <evandeilton@gmail.com> [ORCID: 0009-0007-5887-4084](#)

LEG - Laboratory of Statistics and Geoinformation

PPGMNE - Graduate Program in Numerical Methods in Engineering

Federal University of Paraná (UFPR), Brazil

References

- Carrasco, J. M. F., Ferrari, S. L. P., and Cordeiro, G. M. (2010). A new generalized Kumaraswamy distribution. *arXiv preprint arXiv:1004.0911*. [doi:10.48550/arXiv.1004.0911](#)
- Jones, M. C. (2009). Kumaraswamy's distribution: A beta-type distribution with some tractability advantages. *Statistical Methodology*, **6**(1), 70-81. [doi:10.1016/j.stamet.2008.04.001](#)
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, **46**(1-2), 79-88. [doi:10.1016/00221694\(80\)900360](#)
- Cordeiro, G. M., and de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, **81**(7), 883-898. [doi:10.1080/00949650903530745](#)
- McDonald, J. B. (1984). Some generalized functions for the size distribution of income. *Econometrica*, **52**(3), 647-663. [doi:10.2307/1913469](#)
- Eddelbuettel, D., and Francois, R. (2011). Rcpp: Seamless R and C++ Integration. *Journal of Statistical Software*, **40**(8), 1-18. [doi:10.18637/jss.v040.i08](#)
- Eddelbuettel, D., and Sanderson, C. (2014). RcppArmadillo: Accelerating R with high-performance C++ linear algebra. *Computational Statistics & Data Analysis*, **71**, 1054-1063. [doi:10.1016/j.csda.2013.02.005](#)

See Also

[Beta](#) for the standard beta distribution, [optim](#) for numerical optimization, [dbeta](#) for beta distribution functions

Examples

```
x <- c(0.1, 0.3, 0.5, 0.7, 0.9)
dgkw(x, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2)

## Every sub-family is GKw with some parameters fixed
all.equal(dkw(x, alpha = 2, beta = 3), dgkw(x, alpha = 2, beta = 3))
all.equal(dkw(x, 2, 3, 1.2), dgkw(x, 2, 3, gamma = 1, delta = 0, lambda = 1.2))

## Maximum likelihood with the analytic gradient
set.seed(123)
```

```

y <- rkw(300, alpha = 2, beta = 3)
optim(c(alpha = 1, beta = 1), llkw, grkw, data = y, method = "L-BFGS-B",
      lower = 1e-04)$par

```

dbeta_	<i>Density of the Beta Distribution (gamma, delta+1 Parameterization)</i>
--------	---

Description

Computes the probability density function (PDF) for the standard Beta distribution, using a parameterization common in generalized distribution families. The distribution is parameterized by gamma (γ) and delta (δ), corresponding to the standard Beta distribution with shape parameters shape1 = gamma and shape2 = delta + 1. The distribution is defined on the interval (0, 1).

Usage

```
dbeta_(x, gamma = 1, delta = 0, log = FALSE)
```

Arguments

x	Vector of quantiles (values between 0 and 1).
gamma	First shape parameter (shape1), $\gamma > 0$. Can be a scalar or a vector. Default: 1.0.
delta	Second shape parameter is delta + 1 (shape2), requires $\delta \geq 0$ so that shape2 ≥ 1 . Can be a scalar or a vector. Default: 0.0 (leading to shape2 = 1).
log	Logical; if TRUE, the logarithm of the density is returned ($\log(f(x))$). Default: FALSE.

Details

The probability density function (PDF) calculated by this function corresponds to a standard Beta distribution $Beta(\gamma, \delta + 1)$:

$$f(x; \gamma, \delta) = \frac{x^{\gamma-1}(1-x)^{(\delta+1)-1}}{B(\gamma, \delta+1)} = \frac{x^{\gamma-1}(1-x)^{\delta}}{B(\gamma, \delta+1)}$$

for $0 < x < 1$, where $B(a, b)$ is the Beta function ([beta](#)).

This specific parameterization arises as a special case of the five-parameter Generalized Kumaraswamy (GKw) distribution ([dgkw](#)) obtained by setting the parameters $\alpha = 1$, $\beta = 1$, and $\lambda = 1$. It is therefore equivalent to the McDonald (Mc)/Beta Power distribution ([dmc](#)) with $\lambda = 1$.

Note the difference in the second parameter compared to [dbeta](#), where `dbeta(x, shape1, shape2)` uses shape2 directly. Here, shape1 = gamma and shape2 = delta + 1.

Value

A vector of density values ($f(x)$) or log-density values ($\log(f(x))$). The length of the result is determined by the recycling rule applied to the arguments (x, gamma, delta). Returns 0 (or -Inf if log = TRUE) for x strictly outside the interval [0, 1]. At the closed boundaries $x = 0$ and $x = 1$ the limiting density is returned rather than 0, matching `dbeta` with `shape1 = gamma` and `shape2 = delta + 1`; depending on the parameters that limit is 0, a finite positive value, or Inf. An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Author(s)

Lopes, J. E.

References

Johnson, N. L., Kotz, S., & Balakrishnan, N. (1995). *Continuous Univariate Distributions, Volume 2* (2nd ed.). Wiley.

Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745

See Also

`dbeta` (standard R implementation), `dgkw` (parent distribution density), `dmc` (McDonald/Beta Power density), `pbeta_`, `qbeta_`, `rbeta_`.

Other density functions: `dbkw()`, `dekw()`, `dgkw()`, `dkkw()`, `dkw()`, `dmc()`

Examples

```
x <- c(0.1, 0.3, 0.5, 0.7, 0.9)
dbeta_(x, gamma = 2, delta = 3)
dbeta_(x, gamma = 2, delta = 3, log = TRUE)

## The package's Beta(gamma, delta) is stats::dbeta with shapes gamma, delta + 1
all.equal(dbeta_(x, 2, 3), stats::dbeta(x, 2, 4))

## The density integrates to one
integrate(dbeta_, 0, 1, gamma = 2, delta = 3, rel.tol = 1e-10)

curve(dbeta_(x, gamma = 2, delta = 3), from = 0, to = 1, ylab = "density")
```

dbkw

*Density of the Beta-Kumaraswamy (BKw) Distribution***Description**

Computes the probability density function (PDF) for the Beta-Kumaraswamy (BKw) distribution with parameters alpha (α), beta (β), gamma (γ), and delta (δ). This distribution is defined on the interval (0, 1).

Usage

```
dbkw(x, alpha = 1, beta = 1, gamma = 1, delta = 0, log = FALSE)
```

Arguments

x	Vector of quantiles (values between 0 and 1).
alpha	Shape parameter alpha > 0. Can be a scalar or a vector. Default: 1.0.
beta	Shape parameter beta > 0. Can be a scalar or a vector. Default: 1.0.
gamma	Shape parameter gamma > 0. Can be a scalar or a vector. Default: 1.0.
delta	Shape parameter delta >= 0. Can be a scalar or a vector. Default: 0.0.
log	Logical; if TRUE, the logarithm of the density is returned ($\log(f(x))$). Default: FALSE.

Details

The probability density function (PDF) of the Beta-Kumaraswamy (BKw) distribution is given by:

$$f(x; \alpha, \beta, \gamma, \delta) = \frac{\alpha\beta}{B(\gamma, \delta + 1)} x^{\alpha-1} (1 - x^\alpha)^{\beta(\delta+1)-1} [1 - (1 - x^\alpha)^\beta]^{\gamma-1}$$

for $0 < x < 1$, where $B(a, b)$ is the Beta function ([beta](#)).

The BKw distribution is a special case of the five-parameter Generalized Kumaraswamy (GKw) distribution ([dgkw](#)) obtained by setting the parameter $\lambda = 1$. Numerical evaluation is performed using algorithms similar to those for dgkw, ensuring stability.

Value

A vector of density values ($f(x)$) or log-density values ($\log(f(x))$). The length of the result is determined by the recycling rule applied to the arguments (x, alpha, beta, gamma, delta). Returns 0 (or -Inf if log = TRUE) for x strictly outside the interval [0, 1]. At the closed boundaries x = 0 and x = 1 the limiting density is returned rather than 0, following the convention of base R's density functions (compare [dbeta](#)); depending on the parameters that limit is 0, a finite positive value, or Inf. An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Author(s)

Lopes, J. E.

References

Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745

Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

See Also

[dgkw](#) (parent distribution density), [pbkw](#), [qbkW](#), [rbkw](#) (other BKw functions),

Other density functions: [dbeta_\(\)](#), [dekw\(\)](#), [dgkw\(\)](#), [dkkw\(\)](#), [dkw\(\)](#), [dmc\(\)](#)

Examples

```
x <- c(0.1, 0.3, 0.5, 0.7, 0.9)
dbkw(x, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5)
dbkw(x, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, log = TRUE)

## BKw is GKw with lambda = 1
all.equal(dbkw(x, 2, 3, 1.5, 0.5), dgkw(x, 2, 3, 1.5, 0.5, lambda = 1))

## The density integrates to one
integrate(dbkw, 0, 1, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5,
  rel.tol = 1e-10)

curve(dbkw(x, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5), from = 0,
  to = 1, ylab = "density")
```

dekw

Density of the Exponentiated Kumaraswamy (EKw) Distribution

Description

Computes the probability density function (PDF) for the Exponentiated Kumaraswamy (EKw) distribution with parameters alpha (α), beta (β), and lambda (λ). This distribution is defined on the interval (0, 1).

Usage

```
dekw(x, alpha = 1, beta = 1, lambda = 1, log = FALSE)
```

Arguments

x	Vector of quantiles (values between 0 and 1).
alpha	Shape parameter $\alpha > 0$. Can be a scalar or a vector. Default: 1.0.
beta	Shape parameter $\beta > 0$. Can be a scalar or a vector. Default: 1.0.
lambda	Shape parameter $\lambda > 0$ (exponent parameter). Can be a scalar or a vector. Default: 1.0.
log	Logical; if TRUE, the logarithm of the density is returned ($\log(f(x))$). Default: FALSE.

Details

The probability density function (PDF) of the Exponentiated Kumaraswamy (EKw) distribution is given by:

$$f(x; \alpha, \beta, \lambda) = \lambda \alpha \beta x^{\alpha-1} (1-x^\alpha)^{\beta-1} [1 - (1-x^\alpha)^\beta]^{\lambda-1}$$

for $0 < x < 1$.

The EKw distribution is a special case of the five-parameter Generalized Kumaraswamy (GKw) distribution ([dgkw](#)) obtained by setting the parameters $\gamma = 1$ and $\delta = 0$. When $\lambda = 1$, the EKw distribution reduces to the standard Kumaraswamy distribution.

Value

A vector of density values ($f(x)$) or log-density values ($\log(f(x))$). The length of the result is determined by the recycling rule applied to the arguments (x, alpha, beta, lambda). Returns 0 (or -Inf if log = TRUE) for x strictly outside the interval [0, 1]. At the closed boundaries $x = 0$ and $x = 1$ the limiting density is returned rather than 0, following the convention of base R's density functions (compare [dbeta](#)); depending on the parameters that limit is 0, a finite positive value, or Inf. An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Author(s)

Lopes, J. E.

References

- Nadarajah, S., Cordeiro, G. M., & Ortega, E. M. (2012). The exponentiated Kumaraswamy distribution. *Journal of the Franklin Institute*, 349(3),
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

See Also

[dgkw](#) (parent distribution density), [pekw](#), [qekw](#), [rekw](#) (other EKw functions),
Other density functions: [dbeta_\(\)](#), [dbkw\(\)](#), [dgkw\(\)](#), [dkkw\(\)](#), [dkw\(\)](#), [dmc\(\)](#)

Examples

```
x <- c(0.1, 0.3, 0.5, 0.7, 0.9)
dekw(x, alpha = 2, beta = 3, lambda = 1.2)
dekw(x, alpha = 2, beta = 3, lambda = 1.2, log = TRUE)

## EKw is GKw with gamma = 1, delta = 0
all.equal(dekw(x, 2, 3, 1.2), dgkw(x, 2, 3, gamma = 1, delta = 0, lambda = 1.2))

## The density integrates to one
integrate(dekw, 0, 1, alpha = 2, beta = 3, lambda = 1.2, rel.tol = 1e-10)

curve(dekw(x, alpha = 2, beta = 3, lambda = 1.2), from = 0, to = 1, ylab = "density")
```

dgkw

Density of the Generalized Kumaraswamy Distribution

Description

Computes the probability density function (PDF) for the five-parameter Generalized Kumaraswamy (GKw) distribution, defined on the interval (0, 1).

Usage

```
dgkw(x, alpha = 1, beta = 1, gamma = 1, delta = 0, lambda = 1, log = FALSE)
```

Arguments

x	Vector of quantiles (values between 0 and 1).
alpha	Shape parameter $\alpha > 0$. Can be a scalar or a vector. Default: 1.0.
beta	Shape parameter $\beta > 0$. Can be a scalar or a vector. Default: 1.0.
gamma	Shape parameter $\gamma > 0$. Can be a scalar or a vector. Default: 1.0.
delta	Shape parameter $\delta \geq 0$. Can be a scalar or a vector. Default: 0.0.
lambda	Shape parameter $\lambda > 0$. Can be a scalar or a vector. Default: 1.0.
log	Logical; if TRUE, the logarithm of the density is returned. Default: FALSE.

Details

The probability density function of the Generalized Kumaraswamy (GKw) distribution with parameters α (α), β (β), γ (γ), δ (δ), and λ (λ) is given by:

$$f(x; \alpha, \beta, \gamma, \delta, \lambda) = \frac{\lambda \alpha \beta x^{\alpha-1} (1-x^\alpha)^{\beta-1}}{B(\gamma, \delta+1)} [1 - (1-x^\alpha)^\beta]^{\gamma\lambda-1} [1 - [1 - (1-x^\alpha)^\beta]^\lambda]^\delta$$

for $x \in (0, 1)$, where $B(a, b)$ is the Beta function [beta](#).

This distribution was proposed by Carrasco, Ferrari & Cordeiro (2010) and includes several other distributions as special cases:

- Kumaraswamy (Kw): $\gamma = 1, \delta = 0, \lambda = 1$
- Exponentiated Kumaraswamy (EKw): $\gamma = 1, \delta = 0$
- Beta-Kumaraswamy (BKw): $\lambda = 1$
- Generalized Beta type 1 (GB1 - implies McDonald): $\alpha = 1, \beta = 1$
- Beta distribution: $\alpha = 1, \beta = 1, \lambda = 1$

The function includes checks for valid parameters and input values x . It uses numerical stabilization for x close to 0 or 1.

Value

A vector of density values ($f(x)$) or log-density values ($\log(f(x))$). The length of the result is determined by the recycling rule applied to the arguments ($x, \alpha, \beta, \gamma, \delta, \lambda$). Returns 0 (or $-\text{Inf}$ if $\log = \text{TRUE}$) for x strictly outside the interval $[0, 1]$. At the closed boundaries $x = 0$ and $x = 1$ the limiting density is returned rather than 0, following the convention of base R's density functions (compare [dbeta](#)); depending on the parameters that limit is 0, a finite positive value, or Inf . An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Author(s)

Lopes, J. E.

References

- Carrasco, J. M. F., Ferrari, S. L. P., & Cordeiro, G. M. (2010). A new generalized Kumaraswamy distribution. *arXiv preprint arXiv:1004.0911*. doi:[10.48550/arXiv.1004.0911](#)
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:[10.1080/00949650903530745](#)
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:[10.1016/00221694\(80\)900360](#)

See Also

[pgkw](#), [qgkw](#), [rgkw](#), [dbeta](#), [integrate](#)

Other density functions: [dbeta_\(\)](#), [dbkw\(\)](#), [dekw\(\)](#), [dkkw\(\)](#), [dkw\(\)](#), [dmc\(\)](#)

Examples

```
x <- c(0.1, 0.3, 0.5, 0.7, 0.9)
dgkw(x, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2)
dgkw(x, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2, log = TRUE)

## Kumaraswamy is GKw with gamma = 1, delta = 0, lambda = 1 (the defaults)
all.equal(dgkw(x, alpha = 2, beta = 3), dkw(x, alpha = 2, beta = 3))

## Beta(gamma, delta + 1) is GKw with alpha = beta = lambda = 1
```

```

all.equal(dgkw(x, gamma = 2, delta = 3), stats::dbeta(x, 2, 4))

## The density integrates to one
integrate(dgkw, 0, 1, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5,
          lambda = 1.2, rel.tol = 1e-10)

curve(dgkw(x, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2),
      from = 0, to = 1, ylab = "density")

```

dkkw

*Density of the Kumaraswamy-Kumaraswamy (KKw) Distribution***Description**

Computes the probability density function (PDF) for the Kumaraswamy-Kumaraswamy (KKw) distribution with parameters alpha (α), beta (β), delta (δ), and lambda (λ). This distribution is defined on the interval (0, 1).

Usage

```
dkkw(x, alpha = 1, beta = 1, delta = 0, lambda = 1, log = FALSE)
```

Arguments

x	Vector of quantiles (values between 0 and 1).
alpha	Shape parameter alpha > 0. Can be a scalar or a vector. Default: 1.0.
beta	Shape parameter beta > 0. Can be a scalar or a vector. Default: 1.0.
delta	Shape parameter delta >= 0. Can be a scalar or a vector. Default: 0.0.
lambda	Shape parameter lambda > 0. Can be a scalar or a vector. Default: 1.0.
log	Logical; if TRUE, the logarithm of the density is returned ($\log(f(x))$). Default: FALSE.

Details

The Kumaraswamy-Kumaraswamy (KKw) distribution is a special case of the five-parameter Generalized Kumaraswamy distribution ([dgkw](#)) obtained by setting the parameter $\gamma = 1$.

The probability density function is given by:

$$f(x; \alpha, \beta, \delta, \lambda) = (\delta + 1)\lambda\alpha\beta x^{\alpha-1}(1 - x^\alpha)^{\beta-1} [1 - (1 - x^\alpha)^\beta]^{\lambda-1} \{1 - [1 - (1 - x^\alpha)^\beta]^\lambda\}^\delta$$

for $0 < x < 1$. Note that $1/(\delta + 1)$ corresponds to the Beta function term $B(1, \delta + 1)$ when $\gamma = 1$.

Numerical evaluation follows similar stability considerations as [dgkw](#).

Value

A vector of density values ($f(x)$) or log-density values ($\log(f(x))$). The length of the result is determined by the recycling rule applied to the arguments (`x`, `alpha`, `beta`, `delta`, `lambda`). Returns 0 (or `-Inf` if `log = TRUE`) for `x` strictly outside the interval $[0, 1]$. At the closed boundaries $x = 0$ and $x = 1$ the limiting density is returned rather than 0, following the convention of base R's density functions (compare `dbeta`); depending on the parameters that limit is 0, a finite positive value, or `Inf`. An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Author(s)

Lopes, J. E.

References

Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:[10.1080/00949650903530745](https://doi.org/10.1080/00949650903530745)

Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:[10.1016/00221694\(80\)900360](https://doi.org/10.1016/00221694(80)900360)

See Also

`dgkw` (parent distribution density), `pkkw`, `qkkw`, `rkkw`, `dbeta`

Other density functions: `dbeta_()`, `dbkw()`, `dekw()`, `dgkw()`, `dkw()`, `dmc()`

Examples

```
x <- c(0.1, 0.3, 0.5, 0.7, 0.9)
dkkw(x, alpha = 2, beta = 3, delta = 0.5, lambda = 1.2)
dkkw(x, alpha = 2, beta = 3, delta = 0.5, lambda = 1.2, log = TRUE)

## KKw is GKw with gamma = 1
all.equal(dkkw(x, 2, 3, 0.5, 1.2), dgkw(x, 2, 3, gamma = 1, delta = 0.5,
  lambda = 1.2))

## The density integrates to one
integrate(dkkw, 0, 1, alpha = 2, beta = 3, delta = 0.5, lambda = 1.2,
  rel.tol = 1e-10)

curve(dkkw(x, alpha = 2, beta = 3, delta = 0.5, lambda = 1.2), from = 0,
  to = 1, ylab = "density")
```

dkw

*Density of the Kumaraswamy (Kw) Distribution***Description**

Computes the probability density function (PDF) for the two-parameter Kumaraswamy (Kw) distribution with shape parameters alpha (α) and beta (β). This distribution is defined on the interval (0, 1).

Usage

```
dkw(x, alpha = 1, beta = 1, log = FALSE)
```

Arguments

x	Vector of quantiles (values between 0 and 1).
alpha	Shape parameter alpha > 0. Can be a scalar or a vector. Default: 1.0.
beta	Shape parameter beta > 0. Can be a scalar or a vector. Default: 1.0.
log	Logical; if TRUE, the logarithm of the density is returned ($\log(f(x))$). Default: FALSE.

Details

The probability density function (PDF) of the Kumaraswamy (Kw) distribution is given by:

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

for $0 < x < 1$, $\alpha > 0$, and $\beta > 0$.

The Kumaraswamy distribution is identical to the Generalized Kumaraswamy (GKw) distribution ([dgkw](#)) with parameters $\gamma = 1$, $\delta = 0$, and $\lambda = 1$. It is also a special case of the Exponentiated Kumaraswamy ([dekw](#)) with $\lambda = 1$, and the Kumaraswamy-Kumaraswamy ([dkkw](#)) with $\delta = 0$ and $\lambda = 1$.

Value

A vector of density values ($f(x)$) or log-density values ($\log(f(x))$). The length of the result is determined by the recycling rule applied to the arguments (x, alpha, beta). Returns 0 (or -Inf if log = TRUE) for x strictly outside the interval [0, 1]. At the closed boundaries x = 0 and x = 1 the limiting density is returned rather than 0, following the convention of base R's density functions (compare [dbeta](#)); depending on the parameters that limit is 0, a finite positive value, or Inf. An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Author(s)

Lopes, J. E.

References

- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360
- Jones, M. C. (2009). Kumaraswamy's distribution: A beta-type distribution with some tractability advantages. *Statistical Methodology*, 6(1), 70-81. doi:10.1016/j.stamet.2008.04.001

See Also

[dgbw](#) (parent distribution density), [dgbw](#), [dkbw](#), [pkw](#), [qkw](#), [rkw](#) (other Kw functions), [dbeta](#)

Other density functions: [dbeta_\(\)](#), [dbkw\(\)](#), [dekw\(\)](#), [dgbw\(\)](#), [dkbw\(\)](#), [dmc\(\)](#)

Examples

```
x <- c(0.1, 0.3, 0.5, 0.7, 0.9)
dkw(x, alpha = 2, beta = 3)
dkw(x, alpha = 2, beta = 3, log = TRUE)

## Closed form: alpha * beta * x^(alpha - 1) * (1 - x^alpha)^(beta - 1)
all.equal(dkw(x, 2, 3), 2 * 3 * x * (1 - x^2)^2)

## The density integrates to one
integrate(dkw, 0, 1, alpha = 2, beta = 3, rel.tol = 1e-10)

curve(dkw(x, alpha = 2, beta = 3), from = 0, to = 1, ylab = "density")
```

dmc

Density of the McDonald (Mc)/Beta Power Distribution

Description

Computes the probability density function (PDF) for the McDonald (Mc) distribution (also previously referred to as Beta Power) with parameters gamma (γ), delta (δ), and lambda (λ). This distribution is defined on the interval (0, 1).

Usage

```
dmc(x, gamma = 1, delta = 0, lambda = 1, log = FALSE)
```

Arguments

x	Vector of quantiles (values between 0 and 1).
gamma	Shape parameter gamma > 0. Can be a scalar or a vector. Default: 1.0.
delta	Shape parameter delta >= 0. Can be a scalar or a vector. Default: 0.0.
lambda	Shape parameter lambda > 0. Can be a scalar or a vector. Default: 1.0.
log	Logical; if TRUE, the logarithm of the density is returned ($\log(f(x))$). Default: FALSE.

Details

The probability density function (PDF) of the McDonald (Mc) distribution is given by:

$$f(x; \gamma, \delta, \lambda) = \frac{\lambda}{B(\gamma, \delta + 1)} x^{\gamma\lambda-1} (1 - x^\lambda)^\delta$$

for $0 < x < 1$, where $B(a, b)$ is the Beta function ([beta](#)).

The Mc distribution is a special case of the five-parameter Generalized Kumaraswamy (GKw) distribution ([dgkw](#)) obtained by setting the parameters $\alpha = 1$ and $\beta = 1$. It was introduced by McDonald (1984) and is related to the Generalized Beta distribution of the first kind (GB1). When $\lambda = 1$, it simplifies to the standard Beta distribution with parameters γ and $\delta + 1$.

Value

A vector of density values ($f(x)$) or log-density values ($\log(f(x))$). The length of the result is determined by the recycling rule applied to the arguments (`x`, `gamma`, `delta`, `lambda`). Returns $\emptyset$ (or $-\text{Inf}$ if `log = TRUE`) for `x` strictly outside the interval $[0, 1]$. At the closed boundaries $x = 0$ and $x = 1$ the limiting density is returned rather than $\emptyset$, following the convention of base R's density functions (compare [dbeta](#)); depending on the parameters that limit is $\emptyset$, a finite positive value, or Inf . An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Author(s)

Lopes, J. E.

References

- McDonald, J. B. (1984). Some generalized functions for the size distribution of income. *Econometrica*, 52(3), 647-663. [doi:10.2307/1913469](#)
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. [doi:10.1080/00949650903530745](#)
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. [doi:10.1016/00221694\(80\)900360](#)

See Also

[dgkw](#) (parent distribution density), [pmc](#), [qmc](#), [rmc](#) (other Mc functions), [dbeta](#)
 Other density functions: [dbeta_\(\)](#), [dbkw\(\)](#), [dekw\(\)](#), [dgkw\(\)](#), [dkkw\(\)](#), [dkw\(\)](#)

Examples

```
x <- c(0.1, 0.3, 0.5, 0.7, 0.9)
dmc(x, gamma = 0.5, delta = 5, lambda = 3)
dmc(x, gamma = 0.5, delta = 5, lambda = 3, log = TRUE)

## Mc is GKw with alpha = beta = 1
all.equal(dmc(x, 0.5, 5, 3), dgkw(x, 1, 1, 0.5, 5, 3))
```

```
## The density integrates to one
integrate(dmc, 0, 1, gamma = 0.5, delta = 5, lambda = 3, rel.tol = 1e-10)

curve(dmc(x, gamma = 0.5, delta = 5, lambda = 3), from = 0, to = 1, ylab = "density")
```

gkwgetstartvalues

Estimate Distribution Parameters Using Method of Moments

Description

Estimates parameters for various distribution families from the Generalized Kumaraswamy family using the method of moments. The implementation is optimized for numerical stability and computational efficiency through Nelder-Mead optimization and adaptive numerical integration.

Usage

```
gkwgetstartvalues(x, family = "gkw", n_starts = 5L)
```

Arguments

x	Numeric vector of observations. All values must be in the open interval (0,1). Values outside it – including exact 0 and exact 1 – are truncated to the interval so that a single boundary observation does not abort a fit, and a warning naming the number of truncated observations and their observed range is issued. Truncation shifts every sample moment and therefore every estimate returned, so treat the warning as a signal to check the scale of the data rather than as noise: values such as 5 or -3 usually mean a percentage or 0-100 scale that has to be rescaled first. NA and non-finite values are dropped without a warning.
family	Character string specifying the distribution family. Valid options are: "gkw" (Generalized Kumaraswamy - 5 parameters), "bkw" (Beta-Kumaraswamy - 4 parameters), "kkw" (Kumaraswamy-Kumaraswamy - 4 parameters), "ekw" (Exponentiated Kumaraswamy - 3 parameters), "mc" (McDonald - 3 parameters), "kw" (Kumaraswamy - 2 parameters), "beta" (Beta - 2 parameters). The string is case-insensitive. Default is "gkw".
n_starts	Integer specifying the number of different initial parameter values to try during optimization. Every starting point is optimized and the candidate with the smallest objective is returned, so more starting points can only improve the fit, at a cost that grows linearly in n_starts. Default is 5. Four fixed, family-specific starting points are always used, so values below 4 behave as 4.

Details

The function uses the method of moments to estimate distribution parameters by minimizing the weighted sum of squared relative errors between theoretical and sample moments (orders 1 through

5). The optimization employs the Nelder-Mead simplex algorithm, which is derivative-free and particularly robust for this problem.

Key implementation features: logarithmic calculations for numerical stability, adaptive numerical integration using Simpson's rule with fallback to trapezoidal rule, multiple starting points to avoid local minima, decreasing weights for higher-order moments (1.0, 0.8, 0.6, 0.4, 0.2), and automatic parameter constraint enforcement.

Multiple starts: the grid of starting points is built from four fixed, family-specific points – one of them derived from the sample moments – plus `n_starts - 4` further points drawn over the family's parameter box. Nelder-Mead is run from every point in the grid, each result is clipped to the parameter box, and the clipped candidate with the smallest objective is returned. Because the candidate set grows with `n_starts`, the returned objective is non-increasing in `n_starts`.

Determinism: the extra starting points are drawn from a generator with a fixed internal seed, deliberately not from R's random number stream. The function is therefore deterministic – two calls with the same `x`, `family` and `n_starts` return the same vector, so a fit seeded by these values is reproducible without any further precaution. Two consequences are worth stating explicitly: `set.seed` has no effect on `gkwgetstartvalues`, and the function neither reads nor advances `.Random.seed`, so it cannot perturb simulations running alongside it. To widen the search, raise `n_starts` rather than looking for a seed argument.

Parameter Constraints: All parameters are constrained to positive values. Additionally, family-specific constraints are enforced: `alpha` and `beta` in (0.1, 50.0), `gamma` in (0.1, 10.0) for GKw-related families or (0.1, 50.0) for Beta, `delta` in (0.01, 10.0), and `lambda` in (0.1, 20.0).

The function will issue warnings for empty input vectors, observations outside the open interval (0,1) (truncated to it), sample sizes less than 10 (unreliable estimation), or failure to find valid parameter estimates (returns defaults).

Value

Named numeric vector containing the estimated parameters for the specified distribution family. Parameter names correspond to the distribution specification. If estimation fails, returns a vector of NA values with appropriate parameter names.

References

Jones, M. C. (2009). Kumaraswamy's distribution: A beta-type distribution with some tractability advantages. *Statistical Methodology*, 6(1), 70-81. doi:[10.1016/j.stamet.2008.04.001](https://doi.org/10.1016/j.stamet.2008.04.001)

See Also

[llgkw](#), [llbkw](#), [llkkw](#), [llekw](#), [llmc](#), [llkw](#), [llbeta](#) (the objectives these values are meant to seed), [optim](#)

Examples

```
# Generate sample data from a Beta distribution. set.seed() here makes the
# SAMPLE reproducible; gkwgetstartvalues() itself is deterministic and is
# unaffected by the seed (see the Determinism note in Details).
set.seed(123)
x <- rbeta(100, shape1 = 2, shape2 = 3)
```

```

# Estimate Beta parameters
params_beta <- gkwgetstartvalues(x, family = "beta")
print(params_beta)

# Estimate Kumaraswamy parameters
params_kw <- gkwgetstartvalues(x, family = "kw")
print(params_kw)

# Estimate GKw parameters with more starting points
params_gkw <- gkwgetstartvalues(x, family = "gkw", n_starts = 10)
print(params_gkw)

# Deterministic: the seed in force does not change the answer, and the
# function does not consume draws from R's stream.
set.seed(1)
identical(gkwgetstartvalues(x, family = "kw"), params_kw)
set.seed(9999)
identical(gkwgetstartvalues(x, family = "kw"), params_kw)

before <- .Random.seed
invisible(gkwgetstartvalues(x, family = "kw"))
identical(before, .Random.seed)

```

grbeta

*Gradient of the Negative Log-Likelihood for the Beta Distribution
(gamma, delta+1 Parameterization)*

Description

Computes the gradient vector (vector of first partial derivatives) of the negative log-likelihood function for the standard Beta distribution, using a parameterization common in generalized distribution families. The distribution is parameterized by gamma (γ) and delta (δ), corresponding to the standard Beta distribution with shape parameters shape1 = gamma and shape2 = delta + 1. The gradient is useful for optimization algorithms.

Usage

```
grbeta(par, data)
```

Arguments

par	A numeric vector of length 2 containing the distribution parameters in the order: gamma ($\gamma > 0$), delta ($\delta \geq 0$).
data	A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

This function calculates the gradient of the negative log-likelihood for a Beta distribution with parameters `shape1 = gamma` (γ) and `shape2 = delta + 1` ($\delta + 1$). The components of the gradient vector ($-\nabla\ell(\theta|\mathbf{x})$) are:

$$-\frac{\partial\ell}{\partial\gamma} = n[\psi(\gamma) - \psi(\gamma + \delta + 1)] - \sum_{i=1}^n \ln(x_i)$$

$$-\frac{\partial\ell}{\partial\delta} = n[\psi(\delta + 1) - \psi(\gamma + \delta + 1)] - \sum_{i=1}^n \ln(1 - x_i)$$

where $\psi(\cdot)$ is the digamma function ([digamma](#)). These formulas represent the derivatives of $-\ell(\theta)$, consistent with minimizing the negative log-likelihood. They correspond to the relevant components of the general GKw gradient ([grgkw](#)) evaluated at $\alpha = 1, \beta = 1, \lambda = 1$. Note the parameterization: the standard Beta shape parameters are γ and $\delta + 1$.

Value

Returns a numeric vector of length 2 containing the partial derivatives of the negative log-likelihood function $-\ell(\theta|\mathbf{x})$ with respect to each parameter: $(-\partial\ell/\partial\gamma, -\partial\ell/\partial\delta)$. Returns a vector of NaN if any parameter values are invalid according to their constraints, or if any value in data is not in the interval (0, 1).

Author(s)

Lopes, J. E.

References

Johnson, N. L., Kotz, S., & Balakrishnan, N. (1995). *Continuous Univariate Distributions, Volume 2* (2nd ed.). Wiley.

Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745

(Note: Specific gradient formulas might be derived or sourced from additional references).

See Also

[grgkw](#), [grmc](#) (related gradients), [llbeta](#) (negative log-likelihood function), [hsbeta](#) (Hessian), [dbeta_](#), [pbeta_](#), [qbeta_](#), [rbeta_](#), [optim](#), [grad](#) (for numerical gradient comparison), [digamma](#).

Other gradient functions: [grbkw\(\)](#), [grekw\(\)](#), [grgkw\(\)](#), [grkkw\(\)](#), [grkw\(\)](#), [grmc\(\)](#)

Examples

```
set.seed(123)
x <- rbeta_(200, gamma = 2, delta = 3)
par <- c(gamma = 2, delta = 3)

## Gradient of the negative log-likelihood llbeta(), not of the log-likelihood
```

```

g <- grbeta(par, x)
g

## A small step against the gradient lowers llbeta()
llbeta(par - 1e-4 * g, x) < llbeta(par, x)

## Agrees with a numerical derivative of llbeta()
if (requireNamespace("numDeriv", quietly = TRUE))
  all.equal(g, numDeriv::grad(llbeta, par, data = x), tolerance = 1e-6)

```

grbkw

*Gradient of the Negative Log-Likelihood for the BKw Distribution***Description**

Computes the gradient vector (vector of first partial derivatives) of the negative log-likelihood function for the Beta-Kumaraswamy (BKw) distribution with parameters alpha (α), beta (β), gamma (γ), and delta (δ). This distribution is the special case of the Generalized Kumaraswamy (GKw) distribution where $\lambda = 1$. The gradient is typically used in optimization algorithms for maximum likelihood estimation.

Usage

```
grbkw(par, data)
```

Arguments

par	A numeric vector of length 4 containing the distribution parameters in the order: alpha ($\alpha > 0$), beta ($\beta > 0$), gamma ($\gamma > 0$), delta ($\delta \geq 0$).
data	A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

The components of the gradient vector of the negative log-likelihood ($-\nabla \ell(\theta|\mathbf{x})$) for the BKw ($\lambda = 1$) model are:

$$\begin{aligned}
-\frac{\partial \ell}{\partial \alpha} &= -\frac{n}{\alpha} - \sum_{i=1}^n \ln(x_i) + \sum_{i=1}^n \left[x_i^\alpha \ln(x_i) \left(\frac{\beta(\delta+1)-1}{v_i} - \frac{(\gamma-1)\beta v_i^{\beta-1}}{w_i} \right) \right] \\
-\frac{\partial \ell}{\partial \beta} &= -\frac{n}{\beta} - (\delta+1) \sum_{i=1}^n \ln(v_i) + \sum_{i=1}^n \left[\frac{(\gamma-1)v_i^\beta \ln(v_i)}{w_i} \right] \\
-\frac{\partial \ell}{\partial \gamma} &= n[\psi(\gamma) - \psi(\gamma + \delta + 1)] - \sum_{i=1}^n \ln(w_i)
\end{aligned}$$

$$-\frac{\partial \ell}{\partial \delta} = n[\psi(\delta + 1) - \psi(\gamma + \delta + 1)] - \beta \sum_{i=1}^n \ln(v_i)$$

where:

- $v_i = 1 - x_i^\alpha$
- $w_i = 1 - v_i^\beta = 1 - (1 - x_i^\alpha)^\beta$
- $\psi(\cdot)$ is the digamma function ([digamma](#)).

These formulas represent the derivatives of $-\ell(\theta)$, consistent with minimizing the negative log-likelihood. They correspond to the general GKw gradient ([grgkw](#)) components for $\alpha, \beta, \gamma, \delta$ evaluated at $\lambda = 1$. Note that the component for λ is omitted. Numerical stability is maintained through careful implementation.

Value

Returns a numeric vector of length 4 containing the partial derivatives of the negative log-likelihood function $-\ell(\theta|\mathbf{x})$ with respect to each parameter: $(-\partial \ell / \partial \alpha, -\partial \ell / \partial \beta, -\partial \ell / \partial \gamma, -\partial \ell / \partial \delta)$. Returns a vector of NaN if any parameter values are invalid according to their constraints, or if any value in data is not in the interval (0, 1).

Author(s)

Lopes, J. E.

References

Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745

Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

(Note: Specific gradient formulas might be derived or sourced from additional references).

See Also

[grgkw](#) (parent distribution gradient), [llbkw](#) (negative log-likelihood for BKw), [hsbkw](#) (Hessian for BKw), [dbkw](#) (density for BKw), [optim](#), [grad](#) (for numerical gradient comparison), [digamma](#).

Other gradient functions: [grbeta\(\)](#), [grekw\(\)](#), [grgkw\(\)](#), [grkkw\(\)](#), [grkw\(\)](#), [grmc\(\)](#)

Examples

```
set.seed(123)
x <- rbkw(200, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5)
par <- c(alpha = 2, beta = 3, gamma = 1.5, delta = 0.5)

## Gradient of the negative log-likelihood llbkw(), not of the log-likelihood
g <- grbkw(par, x)
g
```

```
## A small step against the gradient lowers llbkw()
llbkw(par - 1e-4 * g, x) < llbkw(par, x)

## Agrees with a numerical derivative of llbkw()
if (requireNamespace("numDeriv", quietly = TRUE))
  all.equal(g, numDeriv::grad(llbkw, par, data = x), tolerance = 1e-6)
```

grekw

Gradient of the Negative Log-Likelihood for the EKw Distribution

Description

Computes the gradient vector (vector of first partial derivatives) of the negative log-likelihood function for the Exponentiated Kumaraswamy (EKw) distribution with parameters alpha (α), beta (β), and lambda (λ). This distribution is the special case of the Generalized Kumaraswamy (GKw) distribution where $\gamma = 1$ and $\delta = 0$. The gradient is useful for optimization.

Usage

```
grekw(par, data)
```

Arguments

par	A numeric vector of length 3 containing the distribution parameters in the order: alpha ($\alpha > 0$), beta ($\beta > 0$), lambda ($\lambda > 0$).
data	A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

The components of the gradient vector of the negative log-likelihood ($-\nabla \ell(\theta|\mathbf{x})$) for the EKw ($\gamma = 1, \delta = 0$) model are:

$$\begin{aligned}
 -\frac{\partial \ell}{\partial \alpha} &= -\frac{n}{\alpha} - \sum_{i=1}^n \ln(x_i) + \sum_{i=1}^n \left[x_i^\alpha \ln(x_i) \left(\frac{\beta - 1}{v_i} - \frac{(\lambda - 1)\beta v_i^{\beta-1}}{w_i} \right) \right] \\
 -\frac{\partial \ell}{\partial \beta} &= -\frac{n}{\beta} - \sum_{i=1}^n \ln(v_i) + \sum_{i=1}^n \left[\frac{(\lambda - 1)v_i^\beta \ln(v_i)}{w_i} \right] \\
 -\frac{\partial \ell}{\partial \lambda} &= -\frac{n}{\lambda} - \sum_{i=1}^n \ln(w_i)
 \end{aligned}$$

where:

- $v_i = 1 - x_i^\alpha$
- $w_i = 1 - v_i^\beta = 1 - (1 - x_i^\alpha)^\beta$

These formulas represent the derivatives of $-\ell(\theta)$, consistent with minimizing the negative log-likelihood. They correspond to the relevant components of the general GKw gradient ([grgkw](#)) evaluated at $\gamma = 1, \delta = 0$.

Value

Returns a numeric vector of length 3 containing the partial derivatives of the negative log-likelihood function $-\ell(\theta|\mathbf{x})$ with respect to each parameter: $(-\partial\ell/\partial\alpha, -\partial\ell/\partial\beta, -\partial\ell/\partial\lambda)$. Returns a vector of NaN if any parameter values are invalid according to their constraints, or if any value in data is not in the interval (0, 1).

Author(s)

Lopes, J. E.

References

- Nadarajah, S., Cordeiro, G. M., & Ortega, E. M. (2012). The exponentiated Kumaraswamy distribution. *Journal of the Franklin Institute*, 349(3),
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:[10.1080/00949650903530745](#)
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:[10.1016/00221694\(80\)900360](#)
- (Note: Specific gradient formulas might be derived or sourced from additional references).

See Also

[grgkw](#) (parent distribution gradient), [llekwx](#) (negative log-likelihood for EKw), [hsekw](#) (Hessian for EKw), [dekwx](#) (density for EKw), [optim](#), [grad](#) (for numerical gradient comparison).

Other gradient functions: [grbeta\(\)](#), [grbkwx](#), [grgkw\(\)](#), [grkkwx](#), [grkw\(\)](#), [grmc\(\)](#)

Examples

```
set.seed(123)
x <- rekw(200, alpha = 2, beta = 3, lambda = 1.2)
par <- c(alpha = 2, beta = 3, lambda = 1.2)

## Gradient of the negative log-likelihood llekw(), not of the log-likelihood
g <- grekw(par, x)
g

## A small step against the gradient lowers llekw()
llekw(par - 1e-4 * g, x) < llekw(par, x)

## Agrees with a numerical derivative of llekw()
if (requireNamespace("numDeriv", quietly = TRUE))
  all.equal(g, numDeriv::grad(llekwx, par, data = x), tolerance = 1e-6)
```

Description

Computes the gradient vector (vector of partial derivatives) of the negative log-likelihood function for the five-parameter Generalized Kumaraswamy (GKw) distribution. This provides the analytical gradient, often used for efficient optimization via maximum likelihood estimation.

Usage

```
grgkw(par, data)
```

Arguments

par A numeric vector of length 5 containing the distribution parameters in the order: alpha ($\alpha > 0$), beta ($\beta > 0$), gamma ($\gamma > 0$), delta ($\delta \geq 0$), lambda ($\lambda > 0$).

data A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

The components of the gradient vector of the negative log-likelihood ($-\nabla \ell(\theta|\mathbf{x})$) are:

$$\begin{aligned}
 -\frac{\partial \ell}{\partial \alpha} &= -\frac{n}{\alpha} - \sum_{i=1}^n \ln(x_i) + \sum_{i=1}^n \left[x_i^\alpha \ln(x_i) \left(\frac{\beta-1}{v_i} - \frac{(\gamma\lambda-1)\beta v_i^{\beta-1}}{w_i} + \frac{\delta\lambda\beta v_i^{\beta-1} w_i^{\lambda-1}}{z_i} \right) \right] \\
 -\frac{\partial \ell}{\partial \beta} &= -\frac{n}{\beta} - \sum_{i=1}^n \ln(v_i) + \sum_{i=1}^n \left[v_i^\beta \ln(v_i) \left(\frac{\gamma\lambda-1}{w_i} - \frac{\delta\lambda w_i^{\lambda-1}}{z_i} \right) \right] \\
 -\frac{\partial \ell}{\partial \gamma} &= n[\psi(\gamma) - \psi(\gamma + \delta + 1)] - \lambda \sum_{i=1}^n \ln(w_i) \\
 -\frac{\partial \ell}{\partial \delta} &= n[\psi(\delta + 1) - \psi(\gamma + \delta + 1)] - \sum_{i=1}^n \ln(z_i) \\
 -\frac{\partial \ell}{\partial \lambda} &= -\frac{n}{\lambda} - \gamma \sum_{i=1}^n \ln(w_i) + \delta \sum_{i=1}^n \frac{w_i^\lambda \ln(w_i)}{z_i}
 \end{aligned}$$

where:

- $v_i = 1 - x_i^\alpha$
- $w_i = 1 - v_i^\beta = 1 - (1 - x_i^\alpha)^\beta$
- $z_i = 1 - w_i^\lambda = 1 - [1 - (1 - x_i^\alpha)^\beta]^\lambda$
- $\psi(\cdot)$ is the digamma function ([digamma](#)).

Numerical stability is ensured through careful implementation, including checks for valid inputs and handling of intermediate calculations involving potentially small or large numbers, often leveraging the Armadillo C++ library for efficiency.

Value

Returns a numeric vector of length 5 containing the partial derivatives of the negative log-likelihood function $-\ell(\theta|\mathbf{x})$ with respect to each parameter: $(-\partial\ell/\partial\alpha, -\partial\ell/\partial\beta, -\partial\ell/\partial\gamma, -\partial\ell/\partial\delta, -\partial\ell/\partial\lambda)$. Returns a vector of NaN if any parameter values are invalid according to their constraints, or if any value in data is not in the interval (0, 1).

Author(s)

Lopes, J. E.

References

- Carrasco, J. M. F., Ferrari, S. L. P., & Cordeiro, G. M. (2010). A new generalized Kumaraswamy distribution. *arXiv preprint arXiv:1004.0911*. doi:10.48550/arXiv.1004.0911
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

See Also

[llgkw](#) (negative log-likelihood), [hsgkw](#) (Hessian matrix), [dgtk](#) (density), [optim](#), [grad](#) (for numerical gradient comparison), [digamma](#)

Other gradient functions: [grbeta\(\)](#), [grbkw\(\)](#), [grekw\(\)](#), [grkkw\(\)](#), [grkw\(\)](#), [grmc\(\)](#)

Examples

```
set.seed(123)
x <- rgkw(200, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2)
par <- c(alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2)

## Gradient of the negative log-likelihood llgkw(), not of the log-likelihood
g <- grgkw(par, x)
g

## A small step against the gradient lowers llgkw()
llgkw(par - 1e-4 * g, x) < llgkw(par, x)

## Agrees with a numerical derivative of llgkw()
if (requireNamespace("numDeriv", quietly = TRUE))
  all.equal(g, numDeriv::grad(llgkw, par, data = x), tolerance = 1e-6)
```

Description

Computes the gradient vector (vector of first partial derivatives) of the negative log-likelihood function for the Kumaraswamy-Kumaraswamy (KKw) distribution with parameters alpha (α), beta (β), delta (δ), and lambda (λ). This distribution is the special case of the Generalized Kumaraswamy (GKw) distribution where $\gamma = 1$. The gradient is typically used in optimization algorithms for maximum likelihood estimation.

Usage

```
grkkw(par, data)
```

Arguments

par A numeric vector of length 4 containing the distribution parameters in the order: alpha ($\alpha > 0$), beta ($\beta > 0$), delta ($\delta \geq 0$), lambda ($\lambda > 0$).

data A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

The components of the gradient vector of the negative log-likelihood ($-\nabla \ell(\theta|\mathbf{x})$) for the KKw ($\gamma = 1$) model are:

$$\begin{aligned}
 -\frac{\partial \ell}{\partial \alpha} &= -\frac{n}{\alpha} - \sum_{i=1}^n \ln(x_i) + (\beta - 1) \sum_{i=1}^n \frac{x_i^\alpha \ln(x_i)}{v_i} - (\lambda - 1) \sum_{i=1}^n \frac{\beta v_i^{\beta-1} x_i^\alpha \ln(x_i)}{w_i} + \delta \sum_{i=1}^n \frac{\lambda w_i^{\lambda-1} \beta v_i^{\beta-1} x_i^\alpha \ln(x_i)}{z_i} \\
 -\frac{\partial \ell}{\partial \beta} &= -\frac{n}{\beta} - \sum_{i=1}^n \ln(v_i) + (\lambda - 1) \sum_{i=1}^n \frac{v_i^\beta \ln(v_i)}{w_i} - \delta \sum_{i=1}^n \frac{\lambda w_i^{\lambda-1} v_i^\beta \ln(v_i)}{z_i} \\
 -\frac{\partial \ell}{\partial \delta} &= -\frac{n}{\delta + 1} - \sum_{i=1}^n \ln(z_i) \\
 -\frac{\partial \ell}{\partial \lambda} &= -\frac{n}{\lambda} - \sum_{i=1}^n \ln(w_i) + \delta \sum_{i=1}^n \frac{w_i^\lambda \ln(w_i)}{z_i}
 \end{aligned}$$

where:

- $v_i = 1 - x_i^\alpha$
- $w_i = 1 - v_i^\beta = 1 - (1 - x_i^\alpha)^\beta$
- $z_i = 1 - w_i^\lambda = 1 - [1 - (1 - x_i^\alpha)^\beta]^\lambda$

These formulas represent the derivatives of $-\ell(\theta)$, consistent with minimizing the negative log-likelihood. They correspond to the general GKw gradient ([grgkw](#)) components for $\alpha, \beta, \delta, \lambda$ evaluated at $\gamma = 1$. Note that the component for γ is omitted. Numerical stability is maintained through careful implementation.

Value

Returns a numeric vector of length 4 containing the partial derivatives of the negative log-likelihood function $-\ell(\theta|\mathbf{x})$ with respect to each parameter: $(-\partial\ell/\partial\alpha, -\partial\ell/\partial\beta, -\partial\ell/\partial\delta, -\partial\ell/\partial\lambda)$. Returns a vector of NaN if any parameter values are invalid according to their constraints, or if any value in data is not in the interval (0, 1).

Author(s)

Lopes, J. E.

References

- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:[10.1080/00949650903530745](#)
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:[10.1016/00221694\(80\)900360](#)

See Also

[grgkw](#) (parent distribution gradient), [llkkw](#) (negative log-likelihood for KKw), [hskkw](#) (Hessian for KKw), [dkkw](#) (density for KKw), [optim](#), [grad](#) (for numerical gradient comparison).

Other gradient functions: [grbeta\(\)](#), [grbkw\(\)](#), [grekw\(\)](#), [grgkw\(\)](#), [grkw\(\)](#), [grmc\(\)](#)

Examples

```
set.seed(123)
x <- rkkw(200, alpha = 2, beta = 3, delta = 0.5, lambda = 1.2)
par <- c(alpha = 2, beta = 3, delta = 0.5, lambda = 1.2)

## Gradient of the negative log-likelihood llkkw(), not of the log-likelihood
g <- grkkw(par, x)
g

## A small step against the gradient lowers llkkw()
llkkw(par - 1e-4 * g, x) < llkkw(par, x)

## Agrees with a numerical derivative of llkkw()
if (requireNamespace("numDeriv", quietly = TRUE))
  all.equal(g, numDeriv::grad(llkkw, par, data = x), tolerance = 1e-6)
```

grkw	<i>Gradient of the Negative Log-Likelihood for the Kumaraswamy (Kw) Distribution</i>
------	--

Description

Computes the gradient vector (vector of first partial derivatives) of the negative log-likelihood function for the two-parameter Kumaraswamy (Kw) distribution with parameters `alpha` (α) and `beta` (β). This provides the analytical gradient often used for efficient optimization via maximum likelihood estimation.

Usage

```
grkw(par, data)
```

Arguments

<code>par</code>	A numeric vector of length 2 containing the distribution parameters in the order: <code>alpha</code> ($\alpha > 0$), <code>beta</code> ($\beta > 0$).
<code>data</code>	A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

The components of the gradient vector of the negative log-likelihood ($-\nabla\ell(\theta|\mathbf{x})$) for the Kw model are:

$$-\frac{\partial\ell}{\partial\alpha} = -\frac{n}{\alpha} - \sum_{i=1}^n \ln(x_i) + (\beta - 1) \sum_{i=1}^n \frac{x_i^\alpha \ln(x_i)}{v_i}$$

$$-\frac{\partial\ell}{\partial\beta} = -\frac{n}{\beta} - \sum_{i=1}^n \ln(v_i)$$

where $v_i = 1 - x_i^\alpha$. These formulas represent the derivatives of $-\ell(\theta)$, consistent with minimizing the negative log-likelihood. They correspond to the relevant components of the general GKw gradient ([grgkw](#)) evaluated at $\gamma = 1, \delta = 0, \lambda = 1$.

Value

Returns a numeric vector of length 2 containing the partial derivatives of the negative log-likelihood function $-\ell(\theta|\mathbf{x})$ with respect to each parameter: $(-\partial\ell/\partial\alpha, -\partial\ell/\partial\beta)$. Returns a vector of NaN if any parameter values are invalid according to their constraints, or if any value in `data` is not in the interval (0, 1).

Author(s)

Lopes, J. E.

References

Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

Jones, M. C. (2009). Kumaraswamy's distribution: A beta-type distribution with some tractability advantages. *Statistical Methodology*, 6(1), 70-81. doi:10.1016/j.stamet.2008.04.001

(Note: Specific gradient formulas might be derived or sourced from additional references).

See Also

[grgkw](#) (parent distribution gradient), [llkw](#) (negative log-likelihood for Kw), [hskw](#) (Hessian for Kw), [dkw](#) (density for Kw), [optim](#), [grad](#) (for numerical gradient comparison).

Other gradient functions: [grbeta\(\)](#), [grbkw\(\)](#), [grekw\(\)](#), [grgkw\(\)](#), [grkkw\(\)](#), [grmc\(\)](#)

Examples

```
set.seed(123)
x <- rkw(200, alpha = 2, beta = 3)
par <- c(alpha = 2, beta = 3)

## Gradient of the negative log-likelihood llkw(), not of the log-likelihood
g <- grkw(par, x)
g

## A small step against the gradient lowers llkw()
llkw(par - 1e-4 * g, x) < llkw(par, x)

## Agrees with a numerical derivative of llkw()
if (requireNamespace("numDeriv", quietly = TRUE))
  all.equal(g, numDeriv::grad(llkw, par, data = x), tolerance = 1e-6)
```

grmc

Gradient of the Negative Log-Likelihood for the McDonald (Mc)/Beta Power Distribution

Description

Computes the gradient vector (vector of first partial derivatives) of the negative log-likelihood function for the McDonald (Mc) distribution (also known as Beta Power) with parameters gamma (γ), delta (δ), and lambda (λ). This distribution is the special case of the Generalized Kumaraswamy (GKw) distribution where $\alpha = 1$ and $\beta = 1$. The gradient is useful for optimization.

Usage

```
grmc(par, data)
```

Arguments

par	A numeric vector of length 3 containing the distribution parameters in the order: gamma ($\gamma > 0$), delta ($\delta \geq 0$), lambda ($\lambda > 0$).
data	A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

The components of the gradient vector of the negative log-likelihood ($-\nabla \ell(\theta|\mathbf{x})$) for the Mc ($\alpha = 1, \beta = 1$) model are:

$$\begin{aligned}
 -\frac{\partial \ell}{\partial \gamma} &= n[\psi(\gamma) - \psi(\gamma + \delta + 1)] - \lambda \sum_{i=1}^n \ln(x_i) \\
 -\frac{\partial \ell}{\partial \delta} &= n[\psi(\delta + 1) - \psi(\gamma + \delta + 1)] - \sum_{i=1}^n \ln(1 - x_i^\lambda) \\
 -\frac{\partial \ell}{\partial \lambda} &= -\frac{n}{\lambda} - \gamma \sum_{i=1}^n \ln(x_i) + \delta \sum_{i=1}^n \frac{x_i^\lambda \ln(x_i)}{1 - x_i^\lambda}
 \end{aligned}$$

where $\psi(\cdot)$ is the digamma function ([digamma](#)). These formulas represent the derivatives of $-\ell(\theta)$, consistent with minimizing the negative log-likelihood. They correspond to the relevant components of the general GKw gradient ([grgkw](#)) evaluated at $\alpha = 1, \beta = 1$.

Value

Returns a numeric vector of length 3 containing the partial derivatives of the negative log-likelihood function $-\ell(\theta|\mathbf{x})$ with respect to each parameter: $(-\partial \ell / \partial \gamma, -\partial \ell / \partial \delta, -\partial \ell / \partial \lambda)$. Returns a vector of NaN if any parameter values are invalid according to their constraints, or if any value in data is not in the interval (0, 1).

Author(s)

Lopes, J. E.

References

- McDonald, J. B. (1984). Some generalized functions for the size distribution of income. *Econometrica*, 52(3), 647-663. [doi:10.2307/1913469](#)
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. [doi:10.1080/00949650903530745](#)
- (Note: Specific gradient formulas might be derived or sourced from additional references).

See Also

[grgkw](#) (parent distribution gradient), [llmc](#) (negative log-likelihood for Mc), [hsmc](#) (Hessian for Mc), [dmc](#) (density for Mc), [optim](#), [grad](#) (for numerical gradient comparison), [digamma](#).

Other gradient functions: [grbeta\(\)](#), [grbkw\(\)](#), [grekw\(\)](#), [grgkw\(\)](#), [grkkw\(\)](#), [grkw\(\)](#)

Examples

```

set.seed(123)
x <- rmc(200, gamma = 0.5, delta = 5, lambda = 3)
par <- c(gamma = 0.5, delta = 5, lambda = 3)

## Gradient of the negative log-likelihood llmc(), not of the log-likelihood
g <- grmc(par, x)
g

## A small step against the gradient lowers llmc()
llmc(par - 1e-4 * g, x) < llmc(par, x)

## Agrees with a numerical derivative of llmc()
if (requireNamespace("numDeriv", quietly = TRUE))
  all.equal(g, numDeriv::grad(llmc, par, data = x), tolerance = 1e-6)

```

hsbeta

Hessian Matrix of the Negative Log-Likelihood for the Beta Distribution (gamma, delta+1 Parameterization)

Description

Computes the analytic 2x2 Hessian matrix (matrix of second partial derivatives) of the negative log-likelihood function for the standard Beta distribution, using a parameterization common in generalized distribution families. The distribution is parameterized by gamma (γ) and delta (δ), corresponding to the standard Beta distribution with shape parameters shape1 = gamma and shape2 = delta + 1. The Hessian is useful for estimating standard errors and in optimization algorithms.

Usage

```
hsbeta(par, data)
```

Arguments

par	A numeric vector of length 2 containing the distribution parameters in the order: gamma ($\gamma > 0$), delta ($\delta \geq 0$).
data	A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

This function calculates the analytic second partial derivatives of the negative log-likelihood function ($-\ell(\theta|\mathbf{x})$) for a Beta distribution with parameters shape1 = gamma (γ) and shape2 = delta + 1 ($\delta + 1$). The components of the Hessian matrix ($-\mathbf{H}(\theta)$) are:

$$-\frac{\partial^2 \ell}{\partial \gamma^2} = n[\psi'(\gamma) - \psi'(\gamma + \delta + 1)]$$

$$-\frac{\partial^2 \ell}{\partial \gamma \partial \delta} = -n\psi'(\gamma + \delta + 1)$$

$$-\frac{\partial^2 \ell}{\partial \delta^2} = n[\psi'(\delta + 1) - \psi'(\gamma + \delta + 1)]$$

where $\psi'(\cdot)$ is the trigamma function ([trigamma](#)). These formulas represent the second derivatives of $-\ell(\theta)$, consistent with minimizing the negative log-likelihood. They correspond to the relevant 2x2 submatrix of the general GKw Hessian ([hsgkw](#)) evaluated at $\alpha = 1, \beta = 1, \lambda = 1$. Note the parameterization difference from the standard Beta distribution (shape2 = delta + 1).

The returned matrix is symmetric.

Value

Returns a 2x2 numeric matrix representing the Hessian matrix of the negative log-likelihood function, $-\partial^2 \ell / (\partial \theta_i \partial \theta_j)$, where $\theta = (\gamma, \delta)$. Returns a 2x2 matrix populated with NaN if any parameter values are invalid according to their constraints, or if any value in data is not in the interval (0, 1).

Author(s)

Lopes, J. E.

References

Johnson, N. L., Kotz, S., & Balakrishnan, N. (1995). *Continuous Univariate Distributions, Volume 2* (2nd ed.). Wiley.

Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:[10.1080/00949650903530745](https://doi.org/10.1080/00949650903530745)

(Note: Specific Hessian formulas might be derived or sourced from additional references).

See Also

[hsgkw](#), [hsmc](#) (related Hessians), [llbeta](#) (negative log-likelihood function), [grbeta](#) (gradient), [dbeta_](#), [pbeta_](#), [qbeta_](#), [rbeta_](#), [optim](#), [hessian](#) (for numerical Hessian comparison), [trigamma](#).

Other Hessian functions: [hsbkw\(\)](#), [hsekw\(\)](#), [hsgkw\(\)](#), [hskkw\(\)](#), [hskw\(\)](#), [hsmc\(\)](#)

Examples

```
set.seed(123)
x <- rbeta_(1000, gamma = 2, delta = 3)
par <- c(gamma = 2, delta = 3)

## Hessian of the negative log-likelihood llbeta()
H <- hsbeta(par, x)
isSymmetric(H)

## Agrees with a numerical Hessian of llbeta()
if (requireNamespace("numDeriv", quietly = TRUE))
  all.equal(H, numDeriv::hessian(llbeta, par, data = x), tolerance = 1e-8)
```

```
## At the MLE it is the observed information; its inverse estimates the
## covariance of the estimates
fit <- optim(par, llbeta, grbeta, data = x, method = "L-BFGS-B", lower = 1e-4)
sqrt(diag(solve(hsbeta(fit$par, x)))) # standard errors
```

hsbkw	<i>Hessian Matrix of the Negative Log-Likelihood for the BKw Distribution</i>
-------	---

Description

Computes the analytic 4x4 Hessian matrix (matrix of second partial derivatives) of the negative log-likelihood function for the Beta-Kumaraswamy (BKw) distribution with parameters alpha (α), beta (β), gamma (γ), and delta (δ). This distribution is the special case of the Generalized Kumaraswamy (GKw) distribution where $\lambda = 1$. The Hessian is useful for estimating standard errors and in optimization algorithms.

Usage

```
hsbkw(par, data)
```

Arguments

par	A numeric vector of length 4 containing the distribution parameters in the order: alpha ($\alpha > 0$), beta ($\beta > 0$), gamma ($\gamma > 0$), delta ($\delta \geq 0$).
data	A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

This function calculates the analytic second partial derivatives of the negative log-likelihood function based on the BKw log-likelihood ($\lambda = 1$ case of GKw, see [llbkw](#)):

$$\ell(\theta|\mathbf{x}) = n[\ln(\alpha) + \ln(\beta) - \ln B(\gamma, \delta + 1)] + \sum_{i=1}^n [(\alpha - 1) \ln(x_i) + (\beta(\delta + 1) - 1) \ln(v_i) + (\gamma - 1) \ln(w_i)]$$

where $\theta = (\alpha, \beta, \gamma, \delta)$, $B(a, b)$ is the Beta function ([beta](#)), and intermediate terms are:

- $v_i = 1 - x_i^\alpha$
- $w_i = 1 - v_i^\beta = 1 - (1 - x_i^\alpha)^\beta$

The Hessian matrix returned contains the elements $-\frac{\partial^2 \ell(\theta|\mathbf{x})}{\partial \theta_i \partial \theta_j}$ for $\theta_i, \theta_j \in \{\alpha, \beta, \gamma, \delta\}$.

Key properties of the returned matrix:

- Dimensions: 4x4.
- Symmetry: The matrix is symmetric.

- Ordering: Rows and columns correspond to the parameters in the order $\alpha, \beta, \gamma, \delta$.
- Content: Analytic second derivatives of the *negative* log-likelihood.

This corresponds to the relevant 4x4 submatrix of the 5x5 GKw Hessian ([hsgkw](#)) evaluated at $\lambda = 1$. The exact analytical formulas are implemented directly.

Value

Returns a 4x4 numeric matrix representing the Hessian matrix of the negative log-likelihood function, $-\partial^2 \ell / (\partial \theta_i \partial \theta_j)$, where $\theta = (\alpha, \beta, \gamma, \delta)$. Returns a 4x4 matrix populated with NaN if any parameter values are invalid according to their constraints, or if any value in data is not in the interval (0, 1).

Author(s)

Lopes, J. E.

References

Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:[10.1080/00949650903530745](#)

Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:[10.1016/00221694\(80\)900360](#)

(Note: Specific Hessian formulas might be derived or sourced from additional references).

See Also

[hsgkw](#) (parent distribution Hessian), [llbkw](#) (negative log-likelihood for BKw), [grbkw](#) (gradient for BKw), [dbkw](#) (density for BKw), [optim](#), [hessian](#) (for numerical Hessian comparison).

Other Hessian functions: [hsbeta\(\)](#), [hsekw\(\)](#), [hsgkw\(\)](#), [hskkw\(\)](#), [hskw\(\)](#), [hsmc\(\)](#)

Examples

```
set.seed(123)
x <- rbkw(1000, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5)
par <- c(alpha = 2, beta = 3, gamma = 1.5, delta = 0.5)

## Hessian of the negative log-likelihood llbkw()
H <- hsbkw(par, x)
isSymmetric(H)

## Agrees with a numerical Hessian of llbkw()
if (requireNamespace("numDeriv", quietly = TRUE))
  all.equal(H, numDeriv::hessian(llbkw, par, data = x), tolerance = 1e-8)
```

hsekw	<i>Hessian Matrix of the Negative Log-Likelihood for the EKw Distribution</i>
-------	---

Description

Computes the analytic 3x3 Hessian matrix (matrix of second partial derivatives) of the negative log-likelihood function for the Exponentiated Kumaraswamy (EKw) distribution with parameters α (α), β (β), and λ (λ). This distribution is the special case of the Generalized Kumaraswamy (GKw) distribution where $\gamma = 1$ and $\delta = 0$. The Hessian is useful for estimating standard errors and in optimization algorithms.

Usage

```
hsekw(par, data)
```

Arguments

par	A numeric vector of length 3 containing the distribution parameters in the order: α ($\alpha > 0$), β ($\beta > 0$), λ ($\lambda > 0$).
data	A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

This function calculates the analytic second partial derivatives of the negative log-likelihood function based on the EKw log-likelihood ($\gamma = 1, \delta = 0$ case of GKw, see [l1ekw](#)):

$$\ell(\theta|\mathbf{x}) = n[\ln(\lambda) + \ln(\alpha) + \ln(\beta)] + \sum_{i=1}^n [(\alpha - 1) \ln(x_i) + (\beta - 1) \ln(v_i) + (\lambda - 1) \ln(w_i)]$$

where $\theta = (\alpha, \beta, \lambda)$ and intermediate terms are:

- $v_i = 1 - x_i^\alpha$
- $w_i = 1 - v_i^\beta = 1 - (1 - x_i^\alpha)^\beta$

The Hessian matrix returned contains the elements $-\frac{\partial^2 \ell(\theta|\mathbf{x})}{\partial \theta_i \partial \theta_j}$ for $\theta_i, \theta_j \in \{\alpha, \beta, \lambda\}$.

Key properties of the returned matrix:

- Dimensions: 3x3.
- Symmetry: The matrix is symmetric.
- Ordering: Rows and columns correspond to the parameters in the order α, β, λ .
- Content: Analytic second derivatives of the *negative* log-likelihood.

This corresponds to the relevant 3x3 submatrix of the 5x5 GKw Hessian ([hsgkw](#)) evaluated at $\gamma = 1, \delta = 0$. The exact analytical formulas are implemented directly.

Value

Returns a 3x3 numeric matrix representing the Hessian matrix of the negative log-likelihood function, $-\partial^2 \ell / (\partial \theta_i \partial \theta_j)$, where $\theta = (\alpha, \beta, \lambda)$. Returns a 3x3 matrix populated with NaN if any parameter values are invalid according to their constraints, or if any value in data is not in the interval (0, 1).

Author(s)

Lopes, J. E.

References

Nadarajah, S., Cordeiro, G. M., & Ortega, E. M. (2012). The exponentiated Kumaraswamy distribution. *Journal of the Franklin Institute*, 349(3),

Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745

Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

(Note: Specific Hessian formulas might be derived or sourced from additional references).

See Also

[hsgkw](#) (parent distribution Hessian), [llekw](#) (negative log-likelihood for EKw), [grekw](#) (gradient for EKw), [dekw](#) (density for EKw), [optim](#), [hessian](#) (for numerical Hessian comparison).

Other Hessian functions: [hsbeta\(\)](#), [hsbkw\(\)](#), [hsgkw\(\)](#), [hskkw\(\)](#), [hskw\(\)](#), [hsmc\(\)](#)

Examples

```
set.seed(123)
x <- rekw(1000, alpha = 2, beta = 3, lambda = 1.2)
par <- c(alpha = 2, beta = 3, lambda = 1.2)

## Hessian of the negative log-likelihood llekw()
H <- hsekw(par, x)
isSymmetric(H)

## Agrees with a numerical Hessian of llekw()
if (requireNamespace("numDeriv", quietly = TRUE))
  all.equal(H, numDeriv::hessian(llekw, par, data = x), tolerance = 1e-8)

## At the MLE it is the observed information; its inverse estimates the
## covariance of the estimates
fit <- optim(par, llekw, grekw, data = x, method = "L-BFGS-B", lower = 1e-4)
sqrt(diag(solve(hsekw(fit$par, x)))) # standard errors
```

hsgkw	<i>Hessian Matrix of the Negative Log-Likelihood for the GKw Distribution</i>
-------	---

Description

Computes the analytic Hessian matrix (matrix of second partial derivatives) of the negative log-likelihood function for the five-parameter Generalized Kumaraswamy (GKw) distribution. This is typically used to estimate standard errors of maximum likelihood estimates or in optimization algorithms.

Usage

```
hsgkw(par, data)
```

Arguments

par	A numeric vector of length 5 containing the distribution parameters in the order: alpha ($\alpha > 0$), beta ($\beta > 0$), gamma ($\gamma > 0$), delta ($\delta \geq 0$), lambda ($\lambda > 0$).
data	A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

This function calculates the analytic second partial derivatives of the negative log-likelihood function based on the GKw PDF (see [dgkw](#)). The log-likelihood function $\ell(\theta|\mathbf{x})$ is given by:

$$\ell(\theta) = n \ln(\lambda \alpha \beta) - n \ln B(\gamma, \delta + 1) + \sum_{i=1}^n [(\alpha - 1) \ln(x_i) + (\beta - 1) \ln(v_i) + (\gamma \lambda - 1) \ln(w_i) + \delta \ln(z_i)]$$

where $\theta = (\alpha, \beta, \gamma, \delta, \lambda)$, $B(a, b)$ is the Beta function ([beta](#)), and intermediate terms are:

- $v_i = 1 - x_i^\alpha$
- $w_i = 1 - v_i^\beta = 1 - (1 - x_i^\alpha)^\beta$
- $z_i = 1 - w_i^\lambda = 1 - [1 - (1 - x_i^\alpha)^\beta]^\lambda$

The Hessian matrix returned contains the elements $-\frac{\partial^2 \ell(\theta|\mathbf{x})}{\partial \theta_i \partial \theta_j}$.

Key properties of the returned matrix:

- Dimensions: 5x5.
- Symmetry: The matrix is symmetric.
- Ordering: Rows and columns correspond to the parameters in the order $\alpha, \beta, \gamma, \delta, \lambda$.
- Content: Analytic second derivatives of the *negative* log-likelihood.

The exact analytical formulas for the second derivatives are implemented directly (often derived using symbolic differentiation) for accuracy and efficiency, typically using C++.

Value

Returns a 5x5 numeric matrix representing the Hessian matrix of the negative log-likelihood function, i.e., the matrix of second partial derivatives $-\partial^2 \ell / (\partial \theta_i \partial \theta_j)$. Returns a 5x5 matrix populated with NaN if any parameter values are invalid according to their constraints, or if any value in data is not in the interval (0, 1).

Author(s)

Lopes, J. E.

References

- Carrasco, J. M. F., Ferrari, S. L. P., & Cordeiro, G. M. (2010). A new generalized Kumaraswamy distribution. *arXiv preprint arXiv:1004.0911*. doi:10.48550/arXiv.1004.0911
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

See Also

[llgkw](#) (negative log-likelihood function), [grgkw](#) (gradient vector), [dgkw](#) (density function), [optim](#), [hessian](#) (for numerical Hessian comparison).

Other Hessian functions: [hsbeta\(\)](#), [hsbkw\(\)](#), [hsek\(\)](#), [hskkw\(\)](#), [hskw\(\)](#), [hsmc\(\)](#)

Examples

```
set.seed(123)
x <- rgkw(1000, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2)
par <- c(alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2)

## Hessian of the negative log-likelihood llgkw()
H <- hsgkw(par, x)
isSymmetric(H)

## Agrees with a numerical Hessian of llgkw()
if (requireNamespace("numDeriv", quietly = TRUE))
  all.equal(H, numDeriv::hessian(llgkw, par, data = x), tolerance = 1e-8)
```

Description

Computes the analytic 4x4 Hessian matrix (matrix of second partial derivatives) of the negative log-likelihood function for the Kumaraswamy-Kumaraswamy (KKw) distribution with parameters α (α), β (β), δ (δ), and λ (λ). This distribution is the special case of the Generalized Kumaraswamy (GKw) distribution where $\gamma = 1$. The Hessian is useful for estimating standard errors and in optimization algorithms.

Usage

```
hskkw(par, data)
```

Arguments

<code>par</code>	A numeric vector of length 4 containing the distribution parameters in the order: α ($\alpha > 0$), β ($\beta > 0$), δ ($\delta \geq 0$), λ ($\lambda > 0$).
<code>data</code>	A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

This function calculates the analytic second partial derivatives of the negative log-likelihood function based on the KKw log-likelihood ($\gamma = 1$ case of GKw, see [1lkkw](#)):

$$\ell(\theta|\mathbf{x}) = n[\ln(\delta+1)+\ln(\lambda)+\ln(\alpha)+\ln(\beta)] + \sum_{i=1}^n [(\alpha-1)\ln(x_i) + (\beta-1)\ln(v_i) + (\lambda-1)\ln(w_i) + \delta\ln(z_i)]$$

where $\theta = (\alpha, \beta, \delta, \lambda)$ and intermediate terms are:

- $v_i = 1 - x_i^\alpha$
- $w_i = 1 - v_i^\beta = 1 - (1 - x_i^\alpha)^\beta$
- $z_i = 1 - w_i^\lambda = 1 - [1 - (1 - x_i^\alpha)^\beta]^\lambda$

The Hessian matrix returned contains the elements $-\frac{\partial^2 \ell(\theta|\mathbf{x})}{\partial \theta_i \partial \theta_j}$ for $\theta_i, \theta_j \in \{\alpha, \beta, \delta, \lambda\}$.

Key properties of the returned matrix:

- Dimensions: 4x4.
- Symmetry: The matrix is symmetric.
- Ordering: Rows and columns correspond to the parameters in the order $\alpha, \beta, \delta, \lambda$.
- Content: Analytic second derivatives of the *negative* log-likelihood.

This corresponds to the relevant submatrix of the 5x5 GKw Hessian ([hsgkw](#)) evaluated at $\gamma = 1$. The exact analytical formulas are implemented directly.

Value

Returns a 4x4 numeric matrix representing the Hessian matrix of the negative log-likelihood function, $-\partial^2 \ell / (\partial \theta_i \partial \theta_j)$, where $\theta = (\alpha, \beta, \delta, \lambda)$. Returns a 4x4 matrix populated with NaN if any parameter values are invalid according to their constraints, or if any value in data is not in the interval (0, 1).

Author(s)

Lopes, J. E.

References

Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745

Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

See Also

[hsgkw](#) (parent distribution Hessian), [llkkw](#) (negative log-likelihood for KKw), [grkkw](#) (gradient for KKw), [dkkw](#) (density for KKw), [optim](#), [hessian](#) (for numerical Hessian comparison).

Other Hessian functions: [hsbeta\(\)](#), [hsbkw\(\)](#), [hsekw\(\)](#), [hsgkw\(\)](#), [hskw\(\)](#), [hsmc\(\)](#)

Examples

```
set.seed(123)
x <- rkkw(1000, alpha = 2, beta = 3, delta = 0.5, lambda = 1.2)
par <- c(alpha = 2, beta = 3, delta = 0.5, lambda = 1.2)

## Hessian of the negative log-likelihood llkkw()
H <- hskw(par, x)
isSymmetric(H)

## Agrees with a numerical Hessian of llkkw()
if (requireNamespace("numDeriv", quietly = TRUE))
  all.equal(H, numDeriv::hessian(llkkw, par, data = x), tolerance = 1e-8)
```

hskw

Hessian Matrix of the Negative Log-Likelihood for the Kw Distribution

Description

Computes the analytic 2x2 Hessian matrix (matrix of second partial derivatives) of the negative log-likelihood function for the two-parameter Kumaraswamy (Kw) distribution with parameters α (α) and β (β). The Hessian is useful for estimating standard errors and in optimization algorithms.

Usage

```
hskw(par, data)
```

Arguments

par	A numeric vector of length 2 containing the distribution parameters in the order: alpha ($\alpha > 0$), beta ($\beta > 0$).
data	A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

This function calculates the analytic second partial derivatives of the negative log-likelihood function ($-\ell(\theta|\mathbf{x})$). The components are the negative of the second derivatives of the log-likelihood ℓ (derived from the PDF in [dkw](#)).

Let $v_i = 1 - x_i^\alpha$. The second derivatives of the positive log-likelihood (ℓ) are:

$$\begin{aligned}\frac{\partial^2 \ell}{\partial \alpha^2} &= -\frac{n}{\alpha^2} - (\beta - 1) \sum_{i=1}^n \frac{x_i^\alpha (\ln(x_i))^2}{v_i^2} \\ \frac{\partial^2 \ell}{\partial \alpha \partial \beta} &= -\sum_{i=1}^n \frac{x_i^\alpha \ln(x_i)}{v_i} \\ \frac{\partial^2 \ell}{\partial \beta^2} &= -\frac{n}{\beta^2}\end{aligned}$$

The function returns the Hessian matrix containing the negative of these values.

Key properties of the returned matrix:

- Dimensions: 2x2.
- Symmetry: The matrix is symmetric.
- Ordering: Rows and columns correspond to the parameters in the order α, β .
- Content: Analytic second derivatives of the *negative* log-likelihood.

This corresponds to the relevant 2x2 submatrix of the 5x5 GKw Hessian ([hsgkw](#)) evaluated at $\gamma = 1, \delta = 0, \lambda = 1$.

Value

Returns a 2x2 numeric matrix representing the Hessian matrix of the negative log-likelihood function, $-\partial^2 \ell / (\partial \theta_i \partial \theta_j)$, where $\theta = (\alpha, \beta)$. Returns a 2x2 matrix populated with NaN if any parameter values are invalid according to their constraints, or if any value in data is not in the interval (0, 1).

Author(s)

Lopes, J. E.

References

- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360
- Jones, M. C. (2009). Kumaraswamy's distribution: A beta-type distribution with some tractability advantages. *Statistical Methodology*, 6(1), 70-81. doi:10.1016/j.stamet.2008.04.001
- (Note: Specific Hessian formulas might be derived or sourced from additional references).

See Also

[hsgkw](#) (parent distribution Hessian), [llkw](#) (negative log-likelihood for Kw), [grkw](#) (gradient for Kw), [dkw](#) (density for Kw), [optim](#), [hessian](#) (for numerical Hessian comparison).

Other Hessian functions: [hsbeta\(\)](#), [hsbkw\(\)](#), [hsekw\(\)](#), [hsgkw\(\)](#), [hskkw\(\)](#), [hsmc\(\)](#)

Examples

```
set.seed(123)
x <- rkw(1000, alpha = 2, beta = 3)
par <- c(alpha = 2, beta = 3)

## Hessian of the negative log-likelihood llkw()
H <- hskw(par, x)
isSymmetric(H)

## Agrees with a numerical Hessian of llkw()
if (requireNamespace("numDeriv", quietly = TRUE))
  all.equal(H, numDeriv::hessian(llkw, par, data = x), tolerance = 1e-8)

## At the MLE it is the observed information; its inverse estimates the
## covariance of the estimates
fit <- optim(par, llkw, grkw, data = x, method = "L-BFGS-B", lower = 1e-4)
sqrt(diag(solve(hskw(fit$par, x)))) # standard errors
```

hsmc

Hessian Matrix of the Negative Log-Likelihood for the McDonald (Mc)/Beta Power Distribution

Description

Computes the analytic 3x3 Hessian matrix (matrix of second partial derivatives) of the negative log-likelihood function for the McDonald (Mc) distribution (also known as Beta Power) with parameters gamma (γ), delta (δ), and lambda (λ). This distribution is the special case of the Generalized Kumaraswamy (GKw) distribution where $\alpha = 1$ and $\beta = 1$. The Hessian is useful for estimating standard errors and in optimization algorithms.

Usage

```
hsmc(par, data)
```

Arguments

par	A numeric vector of length 3 containing the distribution parameters in the order: gamma ($\gamma > 0$), delta ($\delta \geq 0$), lambda ($\lambda > 0$).
data	A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

This function calculates the analytic second partial derivatives of the negative log-likelihood function ($-\ell(\theta|\mathbf{x})$). The components are based on the second derivatives of the log-likelihood ℓ (derived from the PDF in [dmc](#)).

Note: The formulas below represent the second derivatives of the positive log-likelihood (ℓ). The function returns the **negative** of these values.

$$\frac{\partial^2 \ell}{\partial \gamma^2} = -n[\psi'(\gamma) - \psi'(\gamma + \delta + 1)]$$

$$\frac{\partial^2 \ell}{\partial \gamma \partial \delta} = n\psi'(\gamma + \delta + 1)$$

$$\frac{\partial^2 \ell}{\partial \gamma \partial \lambda} = \sum_{i=1}^n \ln(x_i)$$

$$\frac{\partial^2 \ell}{\partial \delta^2} = -n[\psi'(\delta + 1) - \psi'(\gamma + \delta + 1)]$$

$$\frac{\partial^2 \ell}{\partial \delta \partial \lambda} = -\sum_{i=1}^n \frac{x_i^\lambda \ln(x_i)}{1 - x_i^\lambda}$$

$$\frac{\partial^2 \ell}{\partial \lambda^2} = -\frac{n}{\lambda^2} - \delta \sum_{i=1}^n \frac{x_i^\lambda [\ln(x_i)]^2}{(1 - x_i^\lambda)^2}$$

where $\psi'(\cdot)$ is the trigamma function ([trigamma](#)). The $\partial^2 \ell / \partial \lambda^2$ term matches the C++ implementation (`src/bpmc.cpp`) and is covered by the numerical Hessian checks in `tests/testthat/test-derivatives-validation`.

The returned matrix is symmetric, with rows/columns corresponding to γ, δ, λ .

Value

Returns a 3x3 numeric matrix representing the Hessian matrix of the negative log-likelihood function, $-\partial^2 \ell / (\partial \theta_i \partial \theta_j)$, where $\theta = (\gamma, \delta, \lambda)$. Returns a 3x3 matrix populated with NaN if any parameter values are invalid according to their constraints, or if any value in data is not in the interval (0, 1).

Author(s)

Lopes, J. E.

References

McDonald, J. B. (1984). Some generalized functions for the size distribution of income. *Econometrica*, 52(3), 647-663. [doi:10.2307/1913469](#)

Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. [doi:10.1080/00949650903530745](#)

(Note: Specific Hessian formulas might be derived or sourced from additional references).

See Also

[hsgkw](#) (parent distribution Hessian), [llmc](#) (negative log-likelihood for Mc), [grmc](#) (gradient for Mc), [dmc](#) (density for Mc), [optim](#), [hessian](#) (for numerical Hessian comparison), [trigamma](#).

Other Hessian functions: [hsbeta\(\)](#), [hsbkw\(\)](#), [hsekw\(\)](#), [hsgkw\(\)](#), [hskkw\(\)](#), [hskw\(\)](#)

Examples

```
set.seed(123)
x <- rmc(1000, gamma = 0.5, delta = 5, lambda = 3)
par <- c(gamma = 0.5, delta = 5, lambda = 3)

## Hessian of the negative log-likelihood llmc()
H <- hsmc(par, x)
isSymmetric(H)

## Agrees with a numerical Hessian of llmc()
if (requireNamespace("numDeriv", quietly = TRUE))
  all.equal(H, numDeriv::hessian(llmc, par, data = x), tolerance = 1e-8)

## At the MLE it is the observed information; its inverse estimates the
## covariance of the estimates
fit <- optim(par, llmc, grmc, data = x, method = "L-BFGS-B", lower = 1e-4)
sqrt(diag(solve(hsmc(fit$par, x)))) # standard errors
```

llbeta	<i>Negative Log-Likelihood for the Beta Distribution (gamma, delta+1 Parameterization)</i>
--------	--

Description

Computes the negative log-likelihood function for the standard Beta distribution, using a parameterization common in generalized distribution families. The distribution is parameterized by gamma (γ) and delta (δ), corresponding to the standard Beta distribution with shape parameters shape1 = gamma and shape2 = delta + 1. This function is suitable for maximum likelihood estimation.

Usage

```
llbeta(par, data)
```

Arguments

par	A numeric vector of length 2 containing the distribution parameters in the order: gamma ($\gamma > 0$), delta ($\delta \geq 0$).
data	A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

This function calculates the negative log-likelihood for a Beta distribution with parameters `shape1 = gamma ( $\gamma$ )` and `shape2 = delta + 1 ( $\delta + 1$ )`. The probability density function (PDF) is:

$$f(x|\gamma, \delta) = \frac{x^{\gamma-1}(1-x)^{\delta}}{B(\gamma, \delta+1)}$$

for $0 < x < 1$, where $B(a, b)$ is the Beta function ([beta](#)). The log-likelihood function $\ell(\theta|\mathbf{x})$ for a sample $\mathbf{x} = (x_1, \dots, x_n)$ is $\sum_{i=1}^n \ln f(x_i|\theta)$:

$$\ell(\theta|\mathbf{x}) = \sum_{i=1}^n [(\gamma - 1) \ln(x_i) + \delta \ln(1 - x_i)] - n \ln B(\gamma, \delta + 1)$$

where $\theta = (\gamma, \delta)$. This function computes and returns the *negative* log-likelihood, $-\ell(\theta|\mathbf{x})$, suitable for minimization using optimization routines like [optim](#). It is equivalent to the negative log-likelihood of the GKw distribution ([llgkw](#)) evaluated at $\alpha = 1, \beta = 1, \lambda = 1$, and also to the negative log-likelihood of the McDonald distribution ([llmc](#)) evaluated at $\lambda = 1$. The term $\ln B(\gamma, \delta + 1)$ is typically computed using log-gamma functions ([lgamma](#)) for numerical stability.

Value

Returns a single double value representing the negative log-likelihood ($-\ell(\theta|\mathbf{x})$). Returns `Inf` if any parameter values in `par` are invalid according to their constraints, or if any value in `data` is not in the interval $(0, 1)$; in the latter case a warning naming `data` is also signaled, because an infinite objective offers an optimizer no gradient direction to follow and more often means a sample on the wrong scale than a genuine fit failure.

Author(s)

Lopes, J. E.

References

Johnson, N. L., Kotz, S., & Balakrishnan, N. (1995). *Continuous Univariate Distributions, Volume 2* (2nd ed.). Wiley.

Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:[10.1080/00949650903530745](https://doi.org/10.1080/00949650903530745)

See Also

[llgkw](#), [llmc](#) (related negative log-likelihoods), [dbeta_](#), [pbeta_](#), [qbeta_](#), [rbeta_](#), [grbeta](#) (gradient), [hsbeta](#) (Hessian), [optim](#), [lbeta](#).

Other log-likelihood functions: [llbkw\(\)](#), [llekew\(\)](#), [llgkw\(\)](#), [llkkw\(\)](#), [llkw\(\)](#), [llmc\(\)](#)

Examples

```
set.seed(123)
x <- rbeta_(1000, gamma = 2, delta = 3)
par <- c(gamma = 2, delta = 3)
```

```
## llbeta() is the negative log-likelihood, -sum(log f(x))
llbeta(par, x)
-sum(dbeta_(x, gamma = 2, delta = 3, log = TRUE))

## Maximum likelihood: minimize llbeta(), with grbeta() as its gradient
start <- gkwgetstartvalues(x, family = "beta")
fit <- optim(start, llbeta, grbeta, data = x, method = "L-BFGS-B", lower = 1e-4)
fit$convergence # 0: converged
fit$par
fit$value <= llbeta(par, x) # at least as good as the true values
```

llbkw	<i>Negative Log-Likelihood for the Beta-Kumaraswamy (BKw) Distribution</i>
-------	--

Description

Computes the negative log-likelihood function for the Beta-Kumaraswamy (BKw) distribution with parameters alpha (α), beta (β), gamma (γ), and delta (δ), given a vector of observations. This distribution is the special case of the Generalized Kumaraswamy (GKw) distribution where $\lambda = 1$. This function is typically used for maximum likelihood estimation via numerical optimization.

Usage

```
llbkw(par, data)
```

Arguments

par	A numeric vector of length 4 containing the distribution parameters in the order: alpha ($\alpha > 0$), beta ($\beta > 0$), gamma ($\gamma > 0$), delta ($\delta \geq 0$).
data	A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

The Beta-Kumaraswamy (BKw) distribution is the GKw distribution ([dgkw](#)) with $\lambda = 1$. Its probability density function (PDF) is:

$$f(x|\theta) = \frac{\alpha\beta}{B(\gamma, \delta+1)} x^{\alpha-1} (1-x^\alpha)^{\beta(\delta+1)-1} [1 - (1-x^\alpha)^\beta]^{\gamma-1}$$

for $0 < x < 1$, $\theta = (\alpha, \beta, \gamma, \delta)$, and $B(a, b)$ is the Beta function ([beta](#)). The log-likelihood function $\ell(\theta|\mathbf{x})$ for a sample $\mathbf{x} = (x_1, \dots, x_n)$ is $\sum_{i=1}^n \ln f(x_i|\theta)$:

$$\ell(\theta|\mathbf{x}) = n[\ln(\alpha) + \ln(\beta) - \ln B(\gamma, \delta+1)] + \sum_{i=1}^n [(\alpha-1) \ln(x_i) + (\beta(\delta+1)-1) \ln(v_i) + (\gamma-1) \ln(w_i)]$$

where:

- $v_i = 1 - x_i^\alpha$
- $w_i = 1 - v_i^\beta = 1 - (1 - x_i^\alpha)^\beta$

This function computes and returns the *negative* log-likelihood, $-\ell(\theta|\mathbf{x})$, suitable for minimization using optimization routines like [optim](#). Numerical stability is maintained similarly to [llgkw](#).

Value

Returns a single double value representing the negative log-likelihood ($-\ell(\theta|\mathbf{x})$). Returns Inf if any parameter values in `par` are invalid according to their constraints, or if any value in `data` is not in the interval (0, 1); in the latter case a warning naming `data` is also signaled, because an infinite objective offers an optimizer no gradient direction to follow and more often means a sample on the wrong scale than a genuine fit failure.

Author(s)

Lopes, J. E.

References

Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:[10.1080/00949650903530745](https://doi.org/10.1080/00949650903530745)

Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:[10.1016/00221694\(80\)900360](https://doi.org/10.1016/00221694(80)900360)

See Also

[llgkw](#) (parent distribution negative log-likelihood), [dbkw](#), [pbkw](#), [qbkw](#), [rbkw](#), [grbkw](#) (gradient), [hsbkw](#) (Hessian), [optim](#), [lbeta](#)

Other log-likelihood functions: [llbeta\(\)](#), [llekwx\(\)](#), [llgkw\(\)](#), [llkkw\(\)](#), [llkw\(\)](#), [llmc\(\)](#)

Examples

```
set.seed(123)
x <- rbkw(1000, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5)
par <- c(alpha = 2, beta = 3, gamma = 1.5, delta = 0.5)

## llbkw() is the negative log-likelihood, -sum(log f(x))
llbkw(par, x)
-sum(dbkw(x, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, log = TRUE))

## Maximum likelihood: minimize llbkw(), with grbkw() as its gradient
start <- gkwgetstartvalues(x, family = "bkw")
fit <- optim(start, llbkw, grbkw, data = x, method = "L-BFGS-B", lower = 1e-4)
fit$convergence # 0: converged
## The parameters of this family are weakly identified: compare likelihoods
fit$value <= llbkw(par, x) # at least as good as the true values
```

llekwx	<i>Negative Log-Likelihood for the Exponentiated Kumaraswamy (EKw) Distribution</i>
--------	---

Description

Computes the negative log-likelihood function for the Exponentiated Kumaraswamy (EKw) distribution with parameters alpha (α), beta (β), and lambda (λ), given a vector of observations. This distribution is the special case of the Generalized Kumaraswamy (GKw) distribution where $\gamma = 1$ and $\delta = 0$. This function is suitable for maximum likelihood estimation.

Usage

```
llekwx(par, data)
```

Arguments

par	A numeric vector of length 3 containing the distribution parameters in the order: alpha ($\alpha > 0$), beta ($\beta > 0$), lambda ($\lambda > 0$).
data	A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

The Exponentiated Kumaraswamy (EKw) distribution is the GKw distribution ([dekw](#)) with $\gamma = 1$ and $\delta = 0$. Its probability density function (PDF) is:

$$f(x|\theta) = \lambda\alpha\beta x^{\alpha-1}(1-x^\alpha)^{\beta-1}[1-(1-x^\alpha)^\beta]^{\lambda-1}$$

for $0 < x < 1$ and $\theta = (\alpha, \beta, \lambda)$. The log-likelihood function $\ell(\theta|\mathbf{x})$ for a sample $\mathbf{x} = (x_1, \dots, x_n)$ is $\sum_{i=1}^n \ln f(x_i|\theta)$:

$$\ell(\theta|\mathbf{x}) = n[\ln(\lambda) + \ln(\alpha) + \ln(\beta)] + \sum_{i=1}^n [(\alpha-1)\ln(x_i) + (\beta-1)\ln(v_i) + (\lambda-1)\ln(w_i)]$$

where:

- $v_i = 1 - x_i^\alpha$
- $w_i = 1 - v_i^\beta = 1 - (1 - x_i^\alpha)^\beta$

This function computes and returns the *negative* log-likelihood, $-\ell(\theta|\mathbf{x})$, suitable for minimization using optimization routines like [optim](#). Numerical stability is maintained similarly to [llgkw](#).

Value

Returns a single double value representing the negative log-likelihood ($-\ell(\theta|\mathbf{x})$). Returns Inf if any parameter values in par are invalid according to their constraints, or if any value in data is not in the interval (0, 1); in the latter case a warning naming data is also signaled, because an infinite objective offers an optimizer no gradient direction to follow and more often means a sample on the wrong scale than a genuine fit failure.

Author(s)

Lopes, J. E.

References

- Nadarajah, S., Cordeiro, G. M., & Ortega, E. M. (2012). The exponentiated Kumaraswamy distribution. *Journal of the Franklin Institute*, 349(3),
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

See Also

[llgkw](#) (parent distribution negative log-likelihood), [dekw](#), [pekw](#), [qekw](#), [rekw](#), [grekw](#) (gradient), [hsekw](#) (Hessian), [optim](#)

Other log-likelihood functions: [llbeta\(\)](#), [llbkw\(\)](#), [llgkw\(\)](#), [llkkw\(\)](#), [llkw\(\)](#), [llmc\(\)](#)

Examples

```
set.seed(123)
x <- rekw(1000, alpha = 2, beta = 3, lambda = 1.2)
par <- c(alpha = 2, beta = 3, lambda = 1.2)

## llekw() is the negative log-likelihood, -sum(log f(x))
llekw(par, x)
-sum(dekw(x, alpha = 2, beta = 3, lambda = 1.2, log = TRUE))

## Maximum likelihood: minimize llekw(), with grekw() as its gradient
start <- gkwgetstartvalues(x, family = "ekw")
fit <- optim(start, llekw, grekw, data = x, method = "L-BFGS-B", lower = 1e-4)
fit$convergence # 0: converged
fit$par
fit$value <= llekw(par, x) # at least as good as the true values
```

llgkw

Negative Log-Likelihood for the Generalized Kumaraswamy Distribution

Description

Computes the negative log-likelihood function for the five-parameter Generalized Kumaraswamy (GKw) distribution given a vector of observations. This function is designed for use in optimization routines (e.g., maximum likelihood estimation).

Usage

```
llgkw(par, data)
```

Arguments

par	A numeric vector of length 5 containing the distribution parameters in the order: alpha ($\alpha > 0$), beta ($\beta > 0$), gamma ($\gamma > 0$), delta ($\delta \geq 0$), lambda ($\lambda > 0$).
data	A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

The probability density function (PDF) of the GKw distribution is given in [dgkw](#). The log-likelihood function $\ell(\theta)$ for a sample $\mathbf{x} = (x_1, \dots, x_n)$ is:

$$\ell(\theta|\mathbf{x}) = n \ln(\lambda\alpha\beta) - n \ln B(\gamma, \delta+1) + \sum_{i=1}^n [(\alpha-1) \ln(x_i) + (\beta-1) \ln(v_i) + (\gamma\lambda-1) \ln(w_i) + \delta \ln(z_i)]$$

where $\theta = (\alpha, \beta, \gamma, \delta, \lambda)$, $B(a, b)$ is the Beta function ([beta](#)), and:

- $v_i = 1 - x_i^\alpha$
- $w_i = 1 - v_i^\beta = 1 - (1 - x_i^\alpha)^\beta$
- $z_i = 1 - w_i^\lambda = 1 - [1 - (1 - x_i^\alpha)^\beta]^\lambda$

This function computes $-\ell(\theta|\mathbf{x})$.

Numerical stability is prioritized using:

- [lbeta](#) function for the log-Beta term.
- Log-transformations of intermediate terms (v_i, w_i, z_i) and use of [log1p](#) where appropriate to handle values close to 0 or 1 accurately.
- Checks for invalid parameters and data.

Value

Returns a single double value representing the negative log-likelihood ($-\ell(\theta|\mathbf{x})$). Returns Inf if any parameter values in par are invalid according to their constraints, or if any value in data is not in the interval (0, 1); in the latter case a warning naming data is also signaled, because an infinite objective offers an optimizer no gradient direction to follow and more often means a sample on the wrong scale than a genuine fit failure.

Author(s)

Lopes, J. E.

References

- Carrasco, J. M. F., Ferrari, S. L. P., & Cordeiro, G. M. (2010). A new generalized Kumaraswamy distribution. *arXiv preprint arXiv:1004.0911*. doi:10.48550/arXiv.1004.0911
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

See Also

[dgkw](#), [pgkw](#), [qgkw](#), [rgkw](#), [grgkw](#), [hsgkw](#) (gradient and Hessian), [optim](#), [lbeta](#), [log1p](#)

Other log-likelihood functions: [llbeta\(\)](#), [llbkw\(\)](#), [llekw\(\)](#), [llkkw\(\)](#), [llkw\(\)](#), [llmc\(\)](#)

Examples

```
set.seed(123)
x <- rgkw(1000, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2)
par <- c(alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2)

## llgkw() is the negative log-likelihood, -sum(log f(x))
llgkw(par, x)
-sum(dgkw(x, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2,
  log = TRUE))

## Maximum likelihood: minimize llgkw(), with grgkw() as its gradient
start <- gkwgetstartvalues(x, family = "gkw")
fit <- optim(start, llgkw, grgkw, data = x, method = "L-BFGS-B", lower = 1e-4)
fit$convergence # 0: converged
## The parameters of this family are weakly identified: compare likelihoods
fit$value <= llgkw(par, x) # at least as good as the true values
```

llkkw

Negative Log-Likelihood for the KKw Distribution

Description

Computes the negative log-likelihood function for the Kumaraswamy-Kumaraswamy (KKw) distribution with parameters α (α), β (β), δ (δ), and λ (λ), given a vector of observations. This distribution is a special case of the Generalized Kumaraswamy (GKw) distribution where $\gamma = 1$.

Usage

```
llkkw(par, data)
```

Arguments

par	A numeric vector of length 4 containing the distribution parameters in the order: alpha ($\alpha > 0$), beta ($\beta > 0$), delta ($\delta \geq 0$), lambda ($\lambda > 0$).
data	A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

The KKw distribution is the GKw distribution ([dgkw](#)) with $\gamma = 1$. Its probability density function (PDF) is:

$$f(x|\theta) = (\delta + 1)\lambda\alpha\beta x^{\alpha-1}(1-x^\alpha)^{\beta-1}[1-(1-x^\alpha)^\beta]^{\lambda-1}\{1-[1-(1-x^\alpha)^\beta]^\lambda\}^\delta$$

for $0 < x < 1$ and $\theta = (\alpha, \beta, \delta, \lambda)$. The log-likelihood function $\ell(\theta|\mathbf{x})$ for a sample $\mathbf{x} = (x_1, \dots, x_n)$ is $\sum_{i=1}^n \ln f(x_i|\theta)$:

$$\ell(\theta|\mathbf{x}) = n[\ln(\delta+1)+\ln(\lambda)+\ln(\alpha)+\ln(\beta)] + \sum_{i=1}^n [(\alpha-1)\ln(x_i) + (\beta-1)\ln(v_i) + (\lambda-1)\ln(w_i) + \delta\ln(z_i)]$$

where:

- $v_i = 1 - x_i^\alpha$
- $w_i = 1 - v_i^\beta = 1 - (1 - x_i^\alpha)^\beta$
- $z_i = 1 - w_i^\lambda = 1 - [1 - (1 - x_i^\alpha)^\beta]^\lambda$

This function computes and returns the *negative* log-likelihood, $-\ell(\theta|\mathbf{x})$, suitable for minimization using optimization routines like [optim](#). Numerical stability is maintained similarly to [llgkw](#).

Value

Returns a single double value representing the negative log-likelihood ($-\ell(\theta|\mathbf{x})$). Returns Inf if any parameter values in par are invalid according to their constraints, or if any value in data is not in the interval (0, 1); in the latter case a warning naming data is also signaled, because an infinite objective offers an optimizer no gradient direction to follow and more often means a sample on the wrong scale than a genuine fit failure.

Author(s)

Lopes, J. E.

References

- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:[10.1080/00949650903530745](https://doi.org/10.1080/00949650903530745)
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:[10.1016/00221694\(80\)900360](https://doi.org/10.1016/00221694(80)900360)

See Also

[llgkw](#) (parent distribution negative log-likelihood), [dkkw](#), [pkkw](#), [qkkw](#), [rkkw](#), [grkkw](#) (gradient), [hskkw](#) (Hessian), [optim](#)

Other log-likelihood functions: [llbeta\(\)](#), [llbkw\(\)](#), [llekw\(\)](#), [llgkw\(\)](#), [llkw\(\)](#), [llmc\(\)](#)

Examples

```
set.seed(123)
x <- rkkw(1000, alpha = 2, beta = 3, delta = 0.5, lambda = 1.2)
par <- c(alpha = 2, beta = 3, delta = 0.5, lambda = 1.2)

## llkkw() is the negative log-likelihood, -sum(log f(x))
llkkw(par, x)
-sum(dkkw(x, alpha = 2, beta = 3, delta = 0.5, lambda = 1.2, log = TRUE))

## Maximum likelihood: minimize llkkw(), with grkkw() as its gradient
start <- gkwgetstartvalues(x, family = "kkw")
fit <- optim(start, llkkw, grkkw, data = x, method = "L-BFGS-B", lower = 1e-4)
fit$convergence # 0: converged
## The parameters of this family are weakly identified: compare likelihoods
fit$value <= llkkw(par, x) # at least as good as the true values
```

llkw

Negative Log-Likelihood for the Kumaraswamy (Kw) Distribution

Description

Computes the negative log-likelihood function for the two-parameter Kumaraswamy (Kw) distribution with parameters α (α) and β (β), given a vector of observations. This function is suitable for maximum likelihood estimation.

Usage

```
llkw(par, data)
```

Arguments

par	A numeric vector of length 2 containing the distribution parameters in the order: α ($\alpha > 0$), β ($\beta > 0$).
data	A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

The Kumaraswamy (Kw) distribution's probability density function (PDF) is (see [dkw](#)):

$$f(x|\theta) = \alpha\beta x^{\alpha-1}(1-x^\alpha)^{\beta-1}$$

for $0 < x < 1$ and $\theta = (\alpha, \beta)$. The log-likelihood function $\ell(\theta|\mathbf{x})$ for a sample $\mathbf{x} = (x_1, \dots, x_n)$ is $\sum_{i=1}^n \ln f(x_i|\theta)$:

$$\ell(\theta|\mathbf{x}) = n[\ln(\alpha) + \ln(\beta)] + \sum_{i=1}^n [(\alpha - 1) \ln(x_i) + (\beta - 1) \ln(v_i)]$$

where $v_i = 1 - x_i^\alpha$. This function computes and returns the *negative* log-likelihood, $-\ell(\theta|\mathbf{x})$, suitable for minimization using optimization routines like [optim](#). It is equivalent to the negative log-likelihood of the GKw distribution ([llgkw](#)) evaluated at $\gamma = 1, \delta = 0, \lambda = 1$.

Value

Returns a single double value representing the negative log-likelihood ($-\ell(\theta|\mathbf{x})$). Returns Inf if any parameter values in `par` are invalid according to their constraints, or if any value in `data` is not in the interval (0, 1); in the latter case a warning naming `data` is also signaled, because an infinite objective offers an optimizer no gradient direction to follow and more often means a sample on the wrong scale than a genuine fit failure.

Author(s)

Lopes, J. E.

References

- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:[10.1016/00221694\(80\)900360](#)
- Jones, M. C. (2009). Kumaraswamy's distribution: A beta-type distribution with some tractability advantages. *Statistical Methodology*, 6(1), 70-81. doi:[10.1016/j.stamet.2008.04.001](#)

See Also

[llgkw](#) (parent distribution negative log-likelihood), [dkw](#), [pkw](#), [qkw](#), [rkW](#), [grkw](#) (gradient), [hskw](#) (Hessian), [optim](#)

Other log-likelihood functions: [llbeta\(\)](#), [llbkW\(\)](#), [llekw\(\)](#), [llgkw\(\)](#), [llkkw\(\)](#), [llmc\(\)](#)

Examples

```
set.seed(123)
x <- rkW(1000, alpha = 2, beta = 3)
par <- c(alpha = 2, beta = 3)

## llkw() is the negative log-likelihood, -sum(log f(x))
llkw(par, x)
-sum(dkw(x, alpha = 2, beta = 3, log = TRUE))
```

```
## Maximum likelihood: minimize llkw(), with grkw() as its gradient
start <- gkwgetstartvalues(x, family = "kw")
fit <- optim(start, llkw, grkw, data = x, method = "L-BFGS-B", lower = 1e-4)
fit$convergence # 0: converged
fit$par
fit$value <= llkw(par, x) # at least as good as the true values
```

llmc

Negative Log-Likelihood for the McDonald (Mc)/Beta Power Distribution

Description

Computes the negative log-likelihood function for the McDonald (Mc) distribution (also known as Beta Power) with parameters gamma (γ), delta (δ), and lambda (λ), given a vector of observations. This distribution is the special case of the Generalized Kumaraswamy (GKw) distribution where $\alpha = 1$ and $\beta = 1$. This function is suitable for maximum likelihood estimation.

Usage

```
llmc(par, data)
```

Arguments

par	A numeric vector of length 3 containing the distribution parameters in the order: gamma ($\gamma > 0$), delta ($\delta \geq 0$), lambda ($\lambda > 0$).
data	A numeric vector of observations. All values must be strictly between 0 and 1 (exclusive).

Details

The McDonald (Mc) distribution is the GKw distribution ([dmc](#)) with $\alpha = 1$ and $\beta = 1$. Its probability density function (PDF) is:

$$f(x|\theta) = \frac{\lambda}{B(\gamma, \delta + 1)} x^{\gamma\lambda - 1} (1 - x^\lambda)^\delta$$

for $0 < x < 1$, $\theta = (\gamma, \delta, \lambda)$, and $B(a, b)$ is the Beta function ([beta](#)). The log-likelihood function $\ell(\theta|\mathbf{x})$ for a sample $\mathbf{x} = (x_1, \dots, x_n)$ is $\sum_{i=1}^n \ln f(x_i|\theta)$:

$$\ell(\theta|\mathbf{x}) = n[\ln(\lambda) - \ln B(\gamma, \delta + 1)] + \sum_{i=1}^n [(\gamma\lambda - 1) \ln(x_i) + \delta \ln(1 - x_i^\lambda)]$$

This function computes and returns the *negative* log-likelihood, $-\ell(\theta|\mathbf{x})$, suitable for minimization using optimization routines like [optim](#). Numerical stability is maintained, including using the log-gamma function ([lgamma](#)) for the Beta function term.

Value

Returns a single double value representing the negative log-likelihood ($-\ell(\theta|\mathbf{x})$). Returns Inf if any parameter values in `par` are invalid according to their constraints, or if any value in `data` is not in the interval (0, 1); in the latter case a warning naming `data` is also signaled, because an infinite objective offers an optimizer no gradient direction to follow and more often means a sample on the wrong scale than a genuine fit failure.

Author(s)

Lopes, J. E.

References

- McDonald, J. B. (1984). Some generalized functions for the size distribution of income. *Econometrica*, 52(3), 647-663. doi:[10.2307/1913469](https://doi.org/10.2307/1913469)
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:[10.1080/00949650903530745](https://doi.org/10.1080/00949650903530745)
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:[10.1016/00221694\(80\)900360](https://doi.org/10.1016/00221694(80)900360)

See Also

[llgkw](#) (parent distribution negative log-likelihood), [dmc](#), [pmc](#), [qmc](#), [rmc](#), [grmc](#) (gradient), [hsmc](#) (Hessian), [optim](#), [lbeta](#)

Other log-likelihood functions: [llbeta\(\)](#), [llbkw\(\)](#), [llekwx\(\)](#), [llgkw\(\)](#), [llkkw\(\)](#), [llkw\(\)](#)

Examples

```
set.seed(123)
x <- rmc(1000, gamma = 0.5, delta = 5, lambda = 3)
par <- c(gamma = 0.5, delta = 5, lambda = 3)

## llmc() is the negative log-likelihood, -sum(log f(x))
llmc(par, x)
-sum(dmc(x, gamma = 0.5, delta = 5, lambda = 3, log = TRUE))

## Maximum likelihood: minimize llmc(), with grmc() as its gradient
start <- gkwgetstartvalues(x, family = "mc")
fit <- optim(start, llmc, grmc, data = x, method = "L-BFGS-B", lower = 1e-4)
fit$convergence # 0: converged
fit$par
fit$value <= llmc(par, x) # at least as good as the true values
```

pbeta_	<i>Cumulative Distribution Function (CDF) of the Beta Distribution (gamma, delta+1 Parameterization)</i>
--------	--

Description

Computes the cumulative distribution function (CDF), $F(q) = P(X \leq q)$, for the standard Beta distribution, using a parameterization common in generalized distribution families. The distribution is parameterized by gamma (γ) and delta (δ), corresponding to the standard Beta distribution with shape parameters shape1 = gamma and shape2 = delta + 1.

Usage

```
pbeta_(q, gamma = 1, delta = 0, lower.tail = TRUE, log.p = FALSE)
```

Arguments

q	Vector of quantiles (values generally between 0 and 1).
gamma	First shape parameter (shape1), $\gamma > 0$. Can be a scalar or a vector. Default: 1.0.
delta	Second shape parameter is delta + 1 (shape2), requires $\delta \geq 0$ so that shape2 ≥ 1 . Can be a scalar or a vector. Default: 0.0 (leading to shape2 = 1).
lower.tail	Logical; if TRUE (default), probabilities are $P(X \leq q)$, otherwise, $P(X > q)$.
log.p	Logical; if TRUE, probabilities p are given as $\log(p)$. Default: FALSE.

Details

This function computes the CDF of a Beta distribution with parameters shape1 = gamma and shape2 = delta + 1. It is equivalent to calling `stats::pbeta(q, shape1 = gamma, shape2 = delta + 1, lower.tail = lower.tail, log.p = log.p)`.

This distribution arises as a special case of the five-parameter Generalized Kumaraswamy (GKw) distribution ([pgkw](#)) obtained by setting $\alpha = 1$, $\beta = 1$, and $\lambda = 1$. It is therefore also equivalent to the McDonald (Mc)/Beta Power distribution ([pmc](#)) with $\lambda = 1$.

The function likely calls R's underlying pbeta function but ensures consistent parameter recycling and handling within the C++ environment, matching the style of other functions in the related families.

Value

A vector of probabilities, $F(q)$, or their logarithms/complements depending on lower.tail and log.p. The length of the result is determined by the recycling rule applied to the arguments (q, gamma, delta). When lower.tail = TRUE, returns 0 (or -Inf if log.p = TRUE) for $q \leq 0$ and 1 (or 0 if log.p = TRUE) for $q \geq 1$. An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid. Boundary return values are adjusted accordingly for lower.tail = FALSE.

Author(s)

Lopes, J. E.

References

- Johnson, N. L., Kotz, S., & Balakrishnan, N. (1995). *Continuous Univariate Distributions, Volume 2* (2nd ed.). Wiley.
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745

See Also

[pbeta](#) (standard R implementation), [pgkw](#) (parent distribution CDF), [pmc](#) (McDonald/Beta Power CDF), [dbeta_](#), [qbeta_](#), [rbeta_](#).

Other cumulative distribution functions: [pbkw\(\)](#), [pekw\(\)](#), [pgkw\(\)](#), [pkkw\(\)](#), [pkw\(\)](#), [pmc\(\)](#)

Examples

```
q <- c(0.2, 0.5, 0.8)
pbeta_(q, gamma = 2, delta = 3)
pbeta_(q, gamma = 2, delta = 3, lower.tail = FALSE) # P(X > q)
pbeta_(q, gamma = 2, delta = 3, log.p = TRUE)

## pbeta_() is the integral of dbeta_()
Fq <- pbeta_(0.5, gamma = 2, delta = 3)
all.equal(Fq, integrate(dbeta_, 0, 0.5, gamma = 2, delta = 3, rel.tol = 1e-10)$value)
```

pbkw

Cumulative Distribution Function (CDF) of the Beta-Kumaraswamy (BKw) Distribution

Description

Computes the cumulative distribution function (CDF), $P(X \leq q)$, for the Beta-Kumaraswamy (BKw) distribution with parameters alpha (α), beta (β), gamma (γ), and delta (δ). This distribution is defined on the interval (0, 1) and is a special case of the Generalized Kumaraswamy (GKw) distribution where $\lambda = 1$.

Usage

```
pbkw(
  q,
  alpha = 1,
  beta = 1,
  gamma = 1,
  delta = 0,
```

```

    lower.tail = TRUE,
    log.p = FALSE
)

```

Arguments

q	Vector of quantiles (values generally between 0 and 1).
alpha	Shape parameter $\alpha > 0$. Can be a scalar or a vector. Default: 1.0.
beta	Shape parameter $\beta > 0$. Can be a scalar or a vector. Default: 1.0.
gamma	Shape parameter $\gamma > 0$. Can be a scalar or a vector. Default: 1.0.
delta	Shape parameter $\delta \geq 0$. Can be a scalar or a vector. Default: 0.0.
lower.tail	Logical; if TRUE (default), probabilities are $P(X \leq q)$, otherwise, $P(X > q)$.
log.p	Logical; if TRUE, probabilities p are given as $\log(p)$. Default: FALSE.

Details

The Beta-Kumaraswamy (BKw) distribution is a special case of the five-parameter Generalized Kumaraswamy distribution ([pgkw](#)) obtained by setting the shape parameter $\lambda = 1$.

The CDF of the GKw distribution is $F_{GKw}(q) = I_{y(q)}(\gamma, \delta + 1)$, where $y(q) = [1 - (1 - q^\alpha)^\beta]^\lambda$ and $I_x(a, b)$ is the regularized incomplete beta function ([pbeta](#)). Setting $\lambda = 1$ simplifies $y(q)$ to $1 - (1 - q^\alpha)^\beta$, yielding the BKw CDF:

$$F(q; \alpha, \beta, \gamma, \delta) = I_{1-(1-q^\alpha)^\beta}(\gamma, \delta + 1)$$

This is evaluated using the [pbeta](#) function.

Value

A vector of probabilities, $F(q)$, or their logarithms/complements depending on `lower.tail` and `log.p`. The length of the result is determined by the recycling rule applied to the arguments (q, alpha, beta, gamma, delta). When `lower.tail = TRUE`, returns 0 (or -Inf if `log.p = TRUE`) for $q \leq 0$ and 1 (or 0 if `log.p = TRUE`) for $q \geq 1$. An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid. Boundary return values are adjusted accordingly for `lower.tail = FALSE`.

Author(s)

Lopes, J. E.

References

- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

See Also

[pgkw](#) (parent distribution CDF), [dbkw](#), [qbkw](#), [rbkw](#) (other BKw functions), [pbeta](#)

Other cumulative distribution functions: [pbeta_\(\)](#), [pekw\(\)](#), [pgkw\(\)](#), [pkkw\(\)](#), [pkw\(\)](#), [pmc\(\)](#)

Examples

```
q <- c(0.2, 0.5, 0.8)
pbkw(q, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5)
# P(X > q)
pbkw(q, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lower.tail = FALSE)
pbkw(q, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, log.p = TRUE)

## pbkw() is the integral of dbkw()
Fq <- pbkw(0.5, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5)
all.equal(Fq, integrate(dbkw, 0, 0.5, alpha = 2, beta = 3, gamma = 1.5,
  delta = 0.5, rel.tol = 1e-10)$value)
```

pekw

Cumulative Distribution Function (CDF) of the EKw Distribution

Description

Computes the cumulative distribution function (CDF), $P(X \leq q)$, for the Exponentiated Kumaraswamy (EKw) distribution with parameters alpha (α), beta (β), and lambda (λ). This distribution is defined on the interval (0, 1) and is a special case of the Generalized Kumaraswamy (GKw) distribution where $\gamma = 1$ and $\delta = 0$.

Usage

```
pekw(q, alpha = 1, beta = 1, lambda = 1, lower.tail = TRUE, log.p = FALSE)
```

Arguments

q	Vector of quantiles (values generally between 0 and 1).
alpha	Shape parameter alpha > 0. Can be a scalar or a vector. Default: 1.0.
beta	Shape parameter beta > 0. Can be a scalar or a vector. Default: 1.0.
lambda	Shape parameter lambda > 0 (exponent parameter). Can be a scalar or a vector. Default: 1.0.
lower.tail	Logical; if TRUE (default), probabilities are $P(X \leq q)$, otherwise, $P(X > q)$.
log.p	Logical; if TRUE, probabilities p are given as $\log(p)$. Default: FALSE.

Details

The Exponentiated Kumaraswamy (EKw) distribution is a special case of the five-parameter Generalized Kumaraswamy distribution ([pgkw](#)) obtained by setting parameters $\gamma = 1$ and $\delta = 0$.

The CDF of the GKw distribution is $F_{GKw}(q) = I_{y(q)}(\gamma, \delta + 1)$, where $y(q) = [1 - (1 - q^\alpha)^\beta]^\lambda$ and $I_x(a, b)$ is the regularized incomplete beta function ([pbeta](#)). Setting $\gamma = 1$ and $\delta = 0$ gives $I_{y(q)}(1, 1)$. Since $I_x(1, 1) = x$, the CDF simplifies to $y(q)$:

$$F(q; \alpha, \beta, \lambda) = [1 - (1 - q^\alpha)^\beta]^\lambda$$

for $0 < q < 1$. The implementation uses this closed-form expression for efficiency and handles `lower.tail` and `log.p` arguments appropriately.

Value

A vector of probabilities, $F(q)$, or their logarithms/complements depending on `lower.tail` and `log.p`. The length of the result is determined by the recycling rule applied to the arguments (`q`, `alpha`, `beta`, `lambda`). When `lower.tail = TRUE`, returns $\emptyset$ (or $-\text{Inf}$ if `log.p = TRUE`) for $q \leq 0$ and 1 (or $\emptyset$ if `log.p = TRUE`) for $q \geq 1$. An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid. Boundary return values are adjusted accordingly for `lower.tail = FALSE`.

Author(s)

Lopes, J. E.

References

- Nadarajah, S., Cordeiro, G. M., & Ortega, E. M. (2012). The exponentiated Kumaraswamy distribution. *Journal of the Franklin Institute*, 349(3),
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

See Also

[pgkw](#) (parent distribution CDF), [dekw](#), [qekw](#), [rekw](#) (other EKw functions),

Other cumulative distribution functions: [pbeta_\(\)](#), [pbkw\(\)](#), [pgkw\(\)](#), [pkkw\(\)](#), [pkw\(\)](#), [pmc\(\)](#)

Examples

```
q <- c(0.2, 0.5, 0.8)
pekw(q, alpha = 2, beta = 3, lambda = 1.2)
pekw(q, alpha = 2, beta = 3, lambda = 1.2, lower.tail = FALSE) # P(X > q)
pekw(q, alpha = 2, beta = 3, lambda = 1.2, log.p = TRUE)

## pekw() is the integral of dekw()
```

```
Fq <- pekw(0.5, alpha = 2, beta = 3, lambda = 1.2)
all.equal(Fq, integrate(dekw, 0, 0.5, alpha = 2, beta = 3, lambda = 1.2,
  rel.tol = 1e-10)$value)
```

pgkw

Cumulative Distribution Function (CDF) of the Generalized Kumaraswamy Distribution

Description

Computes the cumulative distribution function (CDF) for the five-parameter Generalized Kumaraswamy (GKw) distribution, defined on the interval (0, 1). Calculates $P(X \leq q)$.

Usage

```
pgkw(
  q,
  alpha = 1,
  beta = 1,
  gamma = 1,
  delta = 0,
  lambda = 1,
  lower.tail = TRUE,
  log.p = FALSE
)
```

Arguments

q	Vector of quantiles (values generally between 0 and 1).
alpha	Shape parameter $\alpha > 0$. Can be a scalar or a vector. Default: 1.0.
beta	Shape parameter $\beta > 0$. Can be a scalar or a vector. Default: 1.0.
gamma	Shape parameter $\gamma > 0$. Can be a scalar or a vector. Default: 1.0.
delta	Shape parameter $\delta \geq 0$. Can be a scalar or a vector. Default: 0.0.
lambda	Shape parameter $\lambda > 0$. Can be a scalar or a vector. Default: 1.0.
lower.tail	Logical; if TRUE (default), probabilities are $P(X \leq q)$, otherwise, $P(X > q)$.
log.p	Logical; if TRUE, probabilities p are given as $\log(p)$. Default: FALSE.

Details

The cumulative distribution function (CDF) of the Generalized Kumaraswamy (GKw) distribution with parameters α (α), β (β), γ (γ), δ (δ), and λ (λ) is given by:

$$F(q; \alpha, \beta, \gamma, \delta, \lambda) = I_{x(q)}(\gamma, \delta + 1)$$

where $x(q) = [1 - (1 - q^\alpha)^\beta]^\lambda$ and $I_x(a, b)$ is the regularized incomplete beta function, defined as:

$$I_x(a, b) = \frac{B_x(a, b)}{B(a, b)} = \frac{\int_0^x t^{a-1}(1-t)^{b-1} dt}{\int_0^1 t^{a-1}(1-t)^{b-1} dt}$$

This corresponds to the `pbeta` function in R, such that $F(q; \alpha, \beta, \gamma, \delta, \lambda) = \text{pbeta}(x(q), \text{shape1} = \gamma, \text{shape2} = \delta + 1)$.

The GKw distribution includes several special cases, such as the Kumaraswamy, Beta, and Exponentiated Kumaraswamy distributions (see `dgkw` for details). The function utilizes numerical algorithms for computing the regularized incomplete beta function accurately, especially near the boundaries.

Value

A vector of probabilities, $F(q)$, or their logarithms/complements depending on `lower.tail` and `log.p`. The length of the result is determined by the recycling rule applied to the arguments (`q`, `alpha`, `beta`, `gamma`, `delta`, `lambda`). When `lower.tail = TRUE`, returns 0 (or `-Inf` if `log.p = TRUE`) for $q \leq 0$ and 1 (or 0 if `log.p = TRUE`) for $q \geq 1$. An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid. Boundary return values are adjusted accordingly for `lower.tail = FALSE`.

Author(s)

Lopes, J. E.

References

- Carrasco, J. M. F., Ferrari, S. L. P., & Cordeiro, G. M. (2010). A new generalized Kumaraswamy distribution. *arXiv preprint arXiv:1004.0911*. doi:10.48550/arXiv.1004.0911
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

See Also

`dgkw`, `qgkw`, `rgkw`, `pbeta`

Other cumulative distribution functions: `pbeta_()`, `pbkw()`, `pekw()`, `pkkw()`, `pkw()`, `pmc()`

Examples

```
q <- c(0.2, 0.5, 0.8)
pgkw(q, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2)
pgkw(q, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2,
     lower.tail = FALSE) # P(X > q)
pgkw(q, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2,
     log.p = TRUE)
```

```
## pgkw() is the integral of dgkw()
Fq <- pgkw(0.5, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2)
all.equal(Fq, integrate(dgkw, 0, 0.5, alpha = 2, beta = 3, gamma = 1.5,
  delta = 0.5, lambda = 1.2, rel.tol = 1e-10)$value)
```

pkkw

Cumulative Distribution Function (CDF) of the KKw Distribution

Description

Computes the cumulative distribution function (CDF), $P(X \leq q)$, for the Kumaraswamy-Kumaraswamy (KKw) distribution with parameters α (α), β (β), δ (δ), and λ (λ). This distribution is defined on the interval (0, 1).

Usage

```
pkkw(
  q,
  alpha = 1,
  beta = 1,
  delta = 0,
  lambda = 1,
  lower.tail = TRUE,
  log.p = FALSE
)
```

Arguments

q	Vector of quantiles (values generally between 0 and 1).
alpha	Shape parameter $\alpha > 0$. Can be a scalar or a vector. Default: 1.0.
beta	Shape parameter $\beta > 0$. Can be a scalar or a vector. Default: 1.0.
delta	Shape parameter $\delta \geq 0$. Can be a scalar or a vector. Default: 0.0.
lambda	Shape parameter $\lambda > 0$. Can be a scalar or a vector. Default: 1.0.
lower.tail	Logical; if TRUE (default), probabilities are $P(X \leq q)$, otherwise, $P(X > q)$.
log.p	Logical; if TRUE, probabilities p are given as $\log(p)$. Default: FALSE.

Details

The Kumaraswamy-Kumaraswamy (KKw) distribution is a special case of the five-parameter Generalized Kumaraswamy distribution ([pgkw](#)) obtained by setting the shape parameter $\gamma = 1$.

The CDF of the GKw distribution is $F_{GKw}(q) = I_{y(q)}(\gamma, \delta + 1)$, where $y(q) = [1 - (1 - q^\alpha)^\beta]^\lambda$ and $I_x(a, b)$ is the regularized incomplete beta function ([pbeta](#)). Setting $\gamma = 1$ utilizes the property $I_x(1, b) = 1 - (1 - x)^b$, yielding the KKw CDF:

$$F(q; \alpha, \beta, \delta, \lambda) = 1 - \{1 - [1 - (1 - q^\alpha)^\beta]^\lambda\}^{\delta+1}$$

for $0 < q < 1$.

The implementation uses this closed-form expression for efficiency and handles `lower.tail` and `log.p` arguments appropriately.

Value

A vector of probabilities, $F(q)$, or their logarithms/complements depending on `lower.tail` and `log.p`. The length of the result is determined by the recycling rule applied to the arguments (`q`, `alpha`, `beta`, `delta`, `lambda`). When `lower.tail = TRUE`, returns 0 (or $-\text{Inf}$ if `log.p = TRUE`) for $q \leq 0$ and 1 (or 0 if `log.p = TRUE`) for $q \geq 1$. An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid. Boundary return values are adjusted accordingly for `lower.tail = FALSE`.

Author(s)

Lopes, J. E.

References

- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:[10.1080/00949650903530745](https://doi.org/10.1080/00949650903530745)
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:[10.1016/00221694\(80\)900360](https://doi.org/10.1016/00221694(80)900360)

See Also

[pgkw](#) (parent distribution CDF), [dkkw](#), [qkkw](#), [rkkw](#), [pbeta](#)

Other cumulative distribution functions: [pbeta_\(\)](#), [pbkw\(\)](#), [pekw\(\)](#), [pgkw\(\)](#), [pkw\(\)](#), [pmc\(\)](#)

Examples

```
q <- c(0.2, 0.5, 0.8)
pkkw(q, alpha = 2, beta = 3, delta = 0.5, lambda = 1.2)
# P(X > q)
pkkw(q, alpha = 2, beta = 3, delta = 0.5, lambda = 1.2, lower.tail = FALSE)
pkkw(q, alpha = 2, beta = 3, delta = 0.5, lambda = 1.2, log.p = TRUE)

## pkkw() is the integral of dkkw()
Fq <- pkkw(0.5, alpha = 2, beta = 3, delta = 0.5, lambda = 1.2)
all.equal(Fq, integrate(dkkw, 0, 0.5, alpha = 2, beta = 3, delta = 0.5,
  lambda = 1.2, rel.tol = 1e-10)$value)
```

pkw	<i>Cumulative Distribution Function (CDF) of the Kumaraswamy (Kw) Distribution</i>
-----	--

Description

Computes the cumulative distribution function (CDF), $P(X \leq q)$, for the two-parameter Kumaraswamy (Kw) distribution with shape parameters alpha (α) and beta (β). This distribution is defined on the interval (0, 1).

Usage

```
pkw(q, alpha = 1, beta = 1, lower.tail = TRUE, log.p = FALSE)
```

Arguments

q	Vector of quantiles (values generally between 0 and 1).
alpha	Shape parameter $\alpha > 0$. Can be a scalar or a vector. Default: 1.0.
beta	Shape parameter $\beta > 0$. Can be a scalar or a vector. Default: 1.0.
lower.tail	Logical; if TRUE (default), probabilities are $P(X \leq q)$, otherwise, $P(X > q)$.
log.p	Logical; if TRUE, probabilities p are given as $\log(p)$. Default: FALSE.

Details

The cumulative distribution function (CDF) of the Kumaraswamy (Kw) distribution is given by:

$$F(x; \alpha, \beta) = 1 - (1 - x^\alpha)^\beta$$

for $0 < x < 1$, $\alpha > 0$, and $\beta > 0$.

The Kw distribution is a special case of several generalized distributions:

- Generalized Kumaraswamy ([pgkw](#)) with $\gamma = 1$, $\delta = 0$, $\lambda = 1$.
- Exponentiated Kumaraswamy ([pekw](#)) with $\lambda = 1$.
- Kumaraswamy-Kumaraswamy ([pkkw](#)) with $\delta = 0$, $\lambda = 1$.

The implementation uses the closed-form expression for efficiency.

Value

A vector of probabilities, $F(q)$, or their logarithms/complements depending on `lower.tail` and `log.p`. The length of the result is determined by the recycling rule applied to the arguments (q, alpha, beta). When `lower.tail = TRUE`, returns 0 (or -Inf if `log.p = TRUE`) for $q \leq 0$ and 1 (or 0 if `log.p = TRUE`) for $q \geq 1$. An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid. Boundary return values are adjusted accordingly for `lower.tail = FALSE`.

Author(s)

Lopes, J. E.

References

Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

Jones, M. C. (2009). Kumaraswamy's distribution: A beta-type distribution with some tractability advantages. *Statistical Methodology*, 6(1), 70-81. doi:10.1016/j.stamet.2008.04.001

See Also

[pgkw](#), [pekw](#), [pkkw](#) (related generalized CDFs), [dkw](#), [qkw](#), [rkw](#) (other Kw functions), [pbeta](#)

Other cumulative distribution functions: [pbeta_\(\)](#), [pbkw\(\)](#), [pekw\(\)](#), [pgkw\(\)](#), [pkkw\(\)](#), [pmc\(\)](#)

Examples

```
q <- c(0.2, 0.5, 0.8)
pkw(q, alpha = 2, beta = 3)
pkw(q, alpha = 2, beta = 3, lower.tail = FALSE) # P(X > q)
pkw(q, alpha = 2, beta = 3, log.p = TRUE)

## pkw() is the integral of dkw()
Fq <- pkw(0.5, alpha = 2, beta = 3)
all.equal(Fq, integrate(dkw, 0, 0.5, alpha = 2, beta = 3, rel.tol = 1e-10)$value)
```

pmc

Cumulative Distribution Function (CDF) of the McDonald (Mc)/Beta Power Distribution

Description

Computes the cumulative distribution function (CDF), $F(q) = P(X \leq q)$, for the McDonald (Mc) distribution (also known as Beta Power) with parameters gamma (γ), delta (δ), and lambda (λ). This distribution is defined on the interval (0, 1) and is a special case of the Generalized Kumaraswamy (GKw) distribution where $\alpha = 1$ and $\beta = 1$.

Usage

```
pmc(q, gamma = 1, delta = 0, lambda = 1, lower.tail = TRUE, log.p = FALSE)
```

Arguments

<code>q</code>	Vector of quantiles (values generally between 0 and 1).
<code>gamma</code>	Shape parameter $\gamma > 0$. Can be a scalar or a vector. Default: 1.0.
<code>delta</code>	Shape parameter $\delta \geq 0$. Can be a scalar or a vector. Default: 0.0.
<code>lambda</code>	Shape parameter $\lambda > 0$. Can be a scalar or a vector. Default: 1.0.
<code>lower.tail</code>	Logical; if TRUE (default), probabilities are $P(X \leq q)$, otherwise, $P(X > q)$.
<code>log.p</code>	Logical; if TRUE, probabilities p are given as $\log(p)$. Default: FALSE.

Details

The McDonald (Mc) distribution is a special case of the five-parameter Generalized Kumaraswamy (GKw) distribution ([pgkw](#)) obtained by setting parameters $\alpha = 1$ and $\beta = 1$.

The CDF of the GKw distribution is $F_{GKw}(q) = I_{y(q)}(\gamma, \delta + 1)$, where $y(q) = [1 - (1 - q^\alpha)^\beta]^\lambda$ and $I_x(a, b)$ is the regularized incomplete beta function ([pbeta](#)). Setting $\alpha = 1$ and $\beta = 1$ simplifies $y(q)$ to q^λ , yielding the Mc CDF:

$$F(q; \gamma, \delta, \lambda) = I_{q^\lambda}(\gamma, \delta + 1)$$

This is evaluated using the [pbeta](#) function as `pbeta(q^lambda, shape1 = gamma, shape2 = delta + 1)`.

Value

A vector of probabilities, $F(q)$, or their logarithms/complements depending on `lower.tail` and `log.p`. The length of the result is determined by the recycling rule applied to the arguments (`q`, `gamma`, `delta`, `lambda`). When `lower.tail = TRUE`, returns 0 (or $-\text{Inf}$ if `log.p = TRUE`) for $q \leq 0$ and 1 (or 0 if `log.p = TRUE`) for $q \geq 1$. An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid. Boundary return values are adjusted accordingly for `lower.tail = FALSE`.

Author(s)

Lopes, J. E.

References

- McDonald, J. B. (1984). Some generalized functions for the size distribution of income. *Econometrica*, 52(3), 647-663. doi:10.2307/1913469
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

See Also

[pgkw](#) (parent distribution CDF), [dmc](#), [qmc](#), [rmc](#) (other Mc functions), [pbeta](#)

Other cumulative distribution functions: [pbeta_\(\)](#), [pbkw\(\)](#), [pekw\(\)](#), [pgkw\(\)](#), [pkkw\(\)](#), [pkw\(\)](#)

Examples

```
q <- c(0.2, 0.5, 0.8)
pmc(q, gamma = 0.5, delta = 5, lambda = 3)
pmc(q, gamma = 0.5, delta = 5, lambda = 3, lower.tail = FALSE) # P(X > q)
pmc(q, gamma = 0.5, delta = 5, lambda = 3, log.p = TRUE)

## pmc() is the integral of dmc()
Fq <- pmc(0.5, gamma = 0.5, delta = 5, lambda = 3)
all.equal(Fq, integrate(dmc, 0, 0.5, gamma = 0.5, delta = 5, lambda = 3,
  rel.tol = 1e-10)$value)
```

qbeta_	<i>Quantile Function of the Beta Distribution (gamma, delta+1 Parameterization)</i>
--------	---

Description

Computes the quantile function (inverse CDF) for the standard Beta distribution, using a parameterization common in generalized distribution families. It finds the value q such that $P(X \leq q) = p$. The distribution is parameterized by γ (gamma) and δ (delta), corresponding to the standard Beta distribution with shape parameters $\text{shape1} = \gamma$ and $\text{shape2} = \delta + 1$.

Usage

```
qbeta_(p, gamma = 1, delta = 0, lower.tail = TRUE, log.p = FALSE)
```

Arguments

<code>p</code>	Vector of probabilities (values between 0 and 1).
<code>gamma</code>	First shape parameter (shape1), $\gamma > 0$. Can be a scalar or a vector. Default: 1.0.
<code>delta</code>	Second shape parameter is $\delta + 1$ (shape2), requires $\delta \geq 0$ so that $\text{shape2} \geq 1$. Can be a scalar or a vector. Default: 0.0 (leading to $\text{shape2} = 1$).
<code>lower.tail</code>	Logical; if TRUE (default), probabilities are $p = P(X \leq q)$, otherwise, probabilities are $p = P(X > q)$.
<code>log.p</code>	Logical; if TRUE, probabilities p are given as $\log(p)$. Default: FALSE.

Details

This function computes the quantiles of a Beta distribution with parameters $\text{shape1} = \gamma$ and $\text{shape2} = \delta + 1$. It is equivalent to calling `stats::qbeta(p, shape1 = gamma, shape2 = delta + 1, lower.tail = lower.tail, log.p = log.p)`.

This distribution arises as a special case of the five-parameter Generalized Kumaraswamy (GKw) distribution ([qgkw](#)) obtained by setting $\alpha = 1$, $\beta = 1$, and $\lambda = 1$. It is therefore also equivalent to the McDonald (Mc)/Beta Power distribution ([qmc](#)) with $\lambda = 1$.

The function likely calls R's underlying qbeta function but ensures consistent parameter recycling and handling within the C++ environment, matching the style of other functions in the related families. Boundary conditions ($p=0$, $p=1$) are handled explicitly.

Value

A vector of quantiles corresponding to the given probabilities p . The length of the result is determined by the recycling rule applied to the arguments (p , γ , δ). Returns:

- 0 for $p = 0$ (or $p = -\text{Inf}$ if $\log.p = \text{TRUE}$, when $\text{lower.tail} = \text{TRUE}$).
- 1 for $p = 1$ (or $p = 0$ if $\log.p = \text{TRUE}$, when $\text{lower.tail} = \text{TRUE}$).
- NaN for $p < 0$ or $p > 1$ (or corresponding log scale).
- An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Boundary return values are adjusted accordingly for $\text{lower.tail} = \text{FALSE}$.

Author(s)

Lopes, J. E.

References

Johnson, N. L., Kotz, S., & Balakrishnan, N. (1995). *Continuous Univariate Distributions, Volume 2* (2nd ed.). Wiley.

Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745

See Also

[qbeta](#) (standard R implementation), [qgkw](#) (parent distribution quantile function), [qmc](#) (McDonald/Beta Power quantile function), [dbeta_](#), [pbeta_](#), [rbeta_](#).

Other quantile functions: [qbkw\(\)](#), [qekw\(\)](#), [qgkw\(\)](#), [qkkw\(\)](#), [qkw\(\)](#), [qmc\(\)](#)

Examples

```
p <- c(0.1, 0.5, 0.9)
qbeta_(p, gamma = 2, delta = 3)
qbeta_(p, gamma = 2, delta = 3, lower.tail = FALSE) # upper-tail quantiles

## qbeta_() inverts pbeta_()
all.equal(pbeta_(qbeta_(p, gamma = 2, delta = 3), gamma = 2, delta = 3), p)
```

qbkw

*Quantile Function of the Beta-Kumaraswamy (BKw) Distribution***Description**

Computes the quantile function (inverse CDF) for the Beta-Kumaraswamy (BKw) distribution with parameters alpha (α), beta (β), gamma (γ), and delta (δ). It finds the value q such that $P(X \leq q) = p$. This distribution is a special case of the Generalized Kumaraswamy (GKw) distribution where the parameter $\lambda = 1$.

Usage

```
qbkw(
  p,
  alpha = 1,
  beta = 1,
  gamma = 1,
  delta = 0,
  lower.tail = TRUE,
  log.p = FALSE
)
```

Arguments

<code>p</code>	Vector of probabilities (values between 0 and 1).
<code>alpha</code>	Shape parameter $\alpha > 0$. Can be a scalar or a vector. Default: 1.0.
<code>beta</code>	Shape parameter $\beta > 0$. Can be a scalar or a vector. Default: 1.0.
<code>gamma</code>	Shape parameter $\gamma > 0$. Can be a scalar or a vector. Default: 1.0.
<code>delta</code>	Shape parameter $\delta \geq 0$. Can be a scalar or a vector. Default: 0.0.
<code>lower.tail</code>	Logical; if TRUE (default), probabilities are $p = P(X \leq q)$, otherwise, probabilities are $p = P(X > q)$.
<code>log.p</code>	Logical; if TRUE, probabilities p are given as $\log(p)$. Default: FALSE.

Details

The quantile function $Q(p)$ is the inverse of the CDF $F(q)$. The CDF for the BKw ($\lambda = 1$) distribution is $F(q) = I_{y(q)}(\gamma, \delta + 1)$, where $y(q) = 1 - (1 - q^\alpha)^\beta$ and $I_z(a, b)$ is the regularized incomplete beta function (see [pbkw](#)).

To find the quantile q , we first invert the outer Beta part: let $y = I_p^{-1}(\gamma, \delta + 1)$, where $I_p^{-1}(a, b)$ is the inverse of the regularized incomplete beta function, computed via [qbeta](#). Then, we invert the inner Kumaraswamy part: $y = 1 - (1 - q^\alpha)^\beta$, which leads to $q = \{1 - (1 - y)^{1/\beta}\}^{1/\alpha}$. Substituting y gives the quantile function:

$$Q(p) = \left\{ 1 - \left[1 - I_p^{-1}(\gamma, \delta + 1) \right]^{1/\beta} \right\}^{1/\alpha}$$

The function uses this formula, calculating $I_p^{-1}(\gamma, \delta + 1)$ via [qbeta\(p, gamma, delta + 1, ...\)](#) while respecting the `lower.tail` and `log.p` arguments.

Value

A vector of quantiles corresponding to the given probabilities p . The length of the result is determined by the recycling rule applied to the arguments (p , α , β , γ , δ). Returns:

- 0 for $p = 0$ (or $p = -\text{Inf}$ if $\log.p = \text{TRUE}$, when `lower.tail = TRUE`).
- 1 for $p = 1$ (or $p = 0$ if $\log.p = \text{TRUE}$, when `lower.tail = TRUE`).
- NaN for $p < 0$ or $p > 1$ (or corresponding log scale).
- An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Boundary return values are adjusted accordingly for `lower.tail = FALSE`.

Author(s)

Lopes, J. E.

References

Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745

Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

See Also

[qgkw](#) (parent distribution quantile function), [dbkw](#), [pbkw](#), [rbkw](#) (other BKw functions), [qbeta](#)

Other quantile functions: [qbeta_\(\)](#), [qekw\(\)](#), [qgkw\(\)](#), [qkkw\(\)](#), [qkw\(\)](#), [qmc\(\)](#)

Examples

```
p <- c(0.1, 0.5, 0.9)
qbkw(p, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5)
# upper-tail quantiles
qbkw(p, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lower.tail = FALSE)

## qbkw() inverts pbkw()
all.equal(pbkw(qbkw(p, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5),
  alpha = 2, beta = 3, gamma = 1.5, delta = 0.5), p)
```

qekw	<i>Quantile Function of the Exponentiated Kumaraswamy (EKw) Distribution</i>
------	--

Description

Computes the quantile function (inverse CDF) for the Exponentiated Kumaraswamy (EKw) distribution with parameters alpha (α), beta (β), and lambda (λ). It finds the value q such that $P(X \leq q) = p$. This distribution is a special case of the Generalized Kumaraswamy (GKw) distribution where $\gamma = 1$ and $\delta = 0$.

Usage

```
qekw(p, alpha = 1, beta = 1, lambda = 1, lower.tail = TRUE, log.p = FALSE)
```

Arguments

p	Vector of probabilities (values between 0 and 1).
alpha	Shape parameter alpha > 0. Can be a scalar or a vector. Default: 1.0.
beta	Shape parameter beta > 0. Can be a scalar or a vector. Default: 1.0.
lambda	Shape parameter lambda > 0 (exponent parameter). Can be a scalar or a vector. Default: 1.0.
lower.tail	Logical; if TRUE (default), probabilities are $p = P(X \leq q)$, otherwise, probabilities are $p = P(X > q)$.
log.p	Logical; if TRUE, probabilities p are given as $\log(p)$. Default: FALSE.

Details

The quantile function $Q(p)$ is the inverse of the CDF $F(q)$. The CDF for the EKw ($\gamma = 1, \delta = 0$) distribution is $F(q) = [1 - (1 - q^\alpha)^\beta]^\lambda$ (see [pekw](#)). Inverting this equation for q yields the quantile function:

$$Q(p) = \left\{ 1 - \left[1 - p^{1/\lambda} \right]^{1/\beta} \right\}^{1/\alpha}$$

The function uses this closed-form expression and correctly handles the `lower.tail` and `log.p` arguments by transforming p appropriately before applying the formula. This is equivalent to the general GKw quantile function ([qgkw](#)) evaluated with $\gamma = 1, \delta = 0$.

Value

A vector of quantiles corresponding to the given probabilities p . The length of the result is determined by the recycling rule applied to the arguments (p , α , β , λ). Returns:

- 0 for $p = 0$ (or $p = -\text{Inf}$ if `log.p = TRUE`, when `lower.tail = TRUE`).
- 1 for $p = 1$ (or $p = 0$ if `log.p = TRUE`, when `lower.tail = TRUE`).
- NaN for $p < 0$ or $p > 1$ (or corresponding log scale).

- An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Boundary return values are adjusted accordingly for `lower.tail = FALSE`.

Author(s)

Lopes, J. E.

References

- Nadarajah, S., Cordeiro, G. M., & Ortega, E. M. (2012). The exponentiated Kumaraswamy distribution. *Journal of the Franklin Institute*, 349(3),
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

See Also

[qgkw](#) (parent distribution quantile function), [dekw](#), [pekw](#), [rekw](#) (other EKw functions), [qunif](#)

Other quantile functions: [qbeta_\(\)](#), [qbkw\(\)](#), [qgkw\(\)](#), [qkkw\(\)](#), [qkw\(\)](#), [qmc\(\)](#)

Examples

```
p <- c(0.1, 0.5, 0.9)
qekw(p, alpha = 2, beta = 3, lambda = 1.2)
# upper-tail quantiles
qekw(p, alpha = 2, beta = 3, lambda = 1.2, lower.tail = FALSE)

## qekw() inverts pekw()
all.equal(pekw(qekw(p, alpha = 2, beta = 3, lambda = 1.2), alpha = 2,
  beta = 3, lambda = 1.2), p)
```

qgkw

Quantile Function of the Generalized Kumaraswamy Distribution

Description

Computes the quantile function (inverse CDF) for the five-parameter Generalized Kumaraswamy (GKw) distribution. Finds the value x such that $P(X \leq x) = p$, where X follows the GKw distribution.

Usage

```
qgkw(
  p,
  alpha = 1,
  beta = 1,
  gamma = 1,
  delta = 0,
  lambda = 1,
  lower.tail = TRUE,
  log.p = FALSE
)
```

Arguments

<code>p</code>	Vector of probabilities (values between 0 and 1).
<code>alpha</code>	Shape parameter $\alpha > 0$. Can be a scalar or a vector. Default: 1.0.
<code>beta</code>	Shape parameter $\beta > 0$. Can be a scalar or a vector. Default: 1.0.
<code>gamma</code>	Shape parameter $\gamma > 0$. Can be a scalar or a vector. Default: 1.0.
<code>delta</code>	Shape parameter $\delta \geq 0$. Can be a scalar or a vector. Default: 0.0.
<code>lambda</code>	Shape parameter $\lambda > 0$. Can be a scalar or a vector. Default: 1.0.
<code>lower.tail</code>	Logical; if TRUE (default), probabilities are $P(X \leq x)$, otherwise, $P(X > x)$.
<code>log.p</code>	Logical; if TRUE, probabilities <code>p</code> are given as $\log(p)$. Default: FALSE.

Details

The quantile function $Q(p)$ is the inverse of the CDF $F(x)$. Given $F(x) = I_{y(x)}(\gamma, \delta + 1)$ where $y(x) = [1 - (1 - x^\alpha)^\beta]^\lambda$, the quantile function is:

$$Q(p) = x = \left\{ 1 - \left[1 - (I_p^{-1}(\gamma, \delta + 1))^{1/\lambda} \right]^{1/\beta} \right\}^{1/\alpha}$$

where $I_p^{-1}(a, b)$ is the inverse of the regularized incomplete beta function, which corresponds to the quantile function of the Beta distribution, [qbeta](#).

The computation proceeds as follows:

1. Calculate `y = stats::qbeta(p, shape1 = gamma, shape2 = delta + 1, lower.tail = lower.tail, log.p = log.p)`.
2. Calculate $v = y^{1/\lambda}$.
3. Calculate $w = (1 - v)^{1/\beta}$. Note: Requires $v \leq 1$.
4. Calculate $q = (1 - w)^{1/\alpha}$. Note: Requires $w \leq 1$.

Numerical stability is maintained by handling boundary cases ($p = 0$, $p = 1$) directly and checking intermediate results (e.g., ensuring arguments to powers are non-negative).

Value

A vector of quantiles corresponding to the given probabilities p . The length of the result is determined by the recycling rule applied to the arguments (p , α , β , γ , δ , λ). Returns:

- 0 for $p = 0$ (or $p = -\text{Inf}$ if $\log.p = \text{TRUE}$, when `lower.tail = TRUE`).
- 1 for $p = 1$ (or $p = 0$ if $\log.p = \text{TRUE}$, when `lower.tail = TRUE`).
- NaN for $p < 0$ or $p > 1$ (or corresponding log scale).
- An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Boundary return values are adjusted accordingly for `lower.tail = FALSE`.

Author(s)

Lopes, J. E.

References

- Carrasco, J. M. F., Ferrari, S. L. P., & Cordeiro, G. M. (2010). A new generalized Kumaraswamy distribution. *arXiv preprint arXiv:1004.0911*. doi:10.48550/arXiv.1004.0911
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

See Also

[dgkw](#), [pgkw](#), [rgkw](#), [qbeta](#)

Other quantile functions: [qbeta_\(\)](#), [qbkw\(\)](#), [qekw\(\)](#), [qkkw\(\)](#), [qkw\(\)](#), [qmc\(\)](#)

Examples

```
p <- c(0.1, 0.5, 0.9)
qgkw(p, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2)
qgkw(p, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2,
     lower.tail = FALSE) # upper-tail quantiles

## qgkw() inverts pgkw()
all.equal(pgkw(qgkw(p, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5,
                    lambda = 1.2), alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2),
          p)
```

qkkw	<i>Quantile Function of the Kumaraswamy-Kumaraswamy (KKw) Distribution</i>
------	--

Description

Computes the quantile function (inverse CDF) for the Kumaraswamy-Kumaraswamy (KKw) distribution with parameters α (α), β (β), δ (δ), and λ (λ). It finds the value q such that $P(X \leq q) = p$. This distribution is a special case of the Generalized Kumaraswamy (GKw) distribution where the parameter $\gamma = 1$.

Usage

```
qkkw(
  p,
  alpha = 1,
  beta = 1,
  delta = 0,
  lambda = 1,
  lower.tail = TRUE,
  log.p = FALSE
)
```

Arguments

<code>p</code>	Vector of probabilities (values between 0 and 1).
<code>alpha</code>	Shape parameter $\alpha > 0$. Can be a scalar or a vector. Default: 1.0.
<code>beta</code>	Shape parameter $\beta > 0$. Can be a scalar or a vector. Default: 1.0.
<code>delta</code>	Shape parameter $\delta \geq 0$. Can be a scalar or a vector. Default: 0.0.
<code>lambda</code>	Shape parameter $\lambda > 0$. Can be a scalar or a vector. Default: 1.0.
<code>lower.tail</code>	Logical; if TRUE (default), probabilities are $p = P(X \leq q)$, otherwise, probabilities are $p = P(X > q)$.
<code>log.p</code>	Logical; if TRUE, probabilities p are given as $\log(p)$. Default: FALSE.

Details

The quantile function $Q(p)$ is the inverse of the CDF $F(q)$. The CDF for the KKw ($\gamma = 1$) distribution is (see [pkkw](#)):

$$F(q) = 1 - \left\{ 1 - \left[1 - (1 - q^\alpha)^\beta \right]^\lambda \right\}^{\delta+1}$$

Inverting this equation for q yields the quantile function:

$$Q(p) = \left[1 - \left\{ 1 - \left[1 - (1 - p)^{1/(\delta+1)} \right]^{1/\lambda} \right\}^{1/\beta} \right]^{1/\alpha}$$

The function uses this closed-form expression and correctly handles the `lower.tail` and `log.p` arguments by transforming p appropriately before applying the formula.

Value

A vector of quantiles corresponding to the given probabilities p . The length of the result is determined by the recycling rule applied to the arguments (p , α , β , δ , λ). Returns:

- 0 for $p = 0$ (or $p = -\text{Inf}$ if $\log.p = \text{TRUE}$, when `lower.tail = TRUE`).
- 1 for $p = 1$ (or $p = 0$ if $\log.p = \text{TRUE}$, when `lower.tail = TRUE`).
- NaN for $p < 0$ or $p > 1$ (or corresponding log scale).
- An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Boundary return values are adjusted accordingly for `lower.tail = FALSE`.

Author(s)

Lopes, J. E.

References

Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745

Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

See Also

[qgkw](#) (parent distribution quantile function), [dkkw](#), [pkkw](#), [rkkw](#), [qbeta](#)

Other quantile functions: [qbeta_\(\)](#), [qbkw\(\)](#), [qekw\(\)](#), [qgkw\(\)](#), [qkw\(\)](#), [qmc\(\)](#)

Examples

```
p <- c(0.1, 0.5, 0.9)
qkkw(p, alpha = 2, beta = 3, delta = 0.5, lambda = 1.2)
# upper-tail quantiles
qkkw(p, alpha = 2, beta = 3, delta = 0.5, lambda = 1.2, lower.tail = FALSE)

## qkkw() inverts pkkw()
all.equal(pkkw(qkkw(p, alpha = 2, beta = 3, delta = 0.5, lambda = 1.2),
  alpha = 2, beta = 3, delta = 0.5, lambda = 1.2), p)
```

qkw

*Quantile Function of the Kumaraswamy (Kw) Distribution***Description**

Computes the quantile function (inverse CDF) for the two-parameter Kumaraswamy (Kw) distribution with shape parameters α (α) and β (β). It finds the value q such that $P(X \leq q) = p$.

Usage

```
qkw(p, alpha = 1, beta = 1, lower.tail = TRUE, log.p = FALSE)
```

Arguments

<code>p</code>	Vector of probabilities (values between 0 and 1).
<code>alpha</code>	Shape parameter $\alpha > 0$. Can be a scalar or a vector. Default: 1.0.
<code>beta</code>	Shape parameter $\beta > 0$. Can be a scalar or a vector. Default: 1.0.
<code>lower.tail</code>	Logical; if TRUE (default), probabilities are $p = P(X \leq q)$, otherwise, probabilities are $p = P(X > q)$.
<code>log.p</code>	Logical; if TRUE, probabilities p are given as $\log(p)$. Default: FALSE.

Details

The quantile function $Q(p)$ is the inverse of the CDF $F(q)$. The CDF for the Kumaraswamy distribution is $F(q) = 1 - (1 - q^\alpha)^\beta$ (see [pkw](#)). Inverting this equation for q yields the quantile function:

$$Q(p) = \left\{ 1 - (1 - p)^{1/\beta} \right\}^{1/\alpha}$$

The function uses this closed-form expression and correctly handles the `lower.tail` and `log.p` arguments by transforming p appropriately before applying the formula. This is equivalent to the general GKw quantile function ([qgkw](#)) evaluated with $\gamma = 1, \delta = 0, \lambda = 1$.

Value

A vector of quantiles corresponding to the given probabilities p . The length of the result is determined by the recycling rule applied to the arguments (p , α , β). Returns:

- 0 for $p = 0$ (or $p = -\text{Inf}$ if `log.p = TRUE`, when `lower.tail = TRUE`).
- 1 for $p = 1$ (or $p = 0$ if `log.p = TRUE`, when `lower.tail = TRUE`).
- NaN for $p < 0$ or $p > 1$ (or corresponding log scale).
- An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Boundary return values are adjusted accordingly for `lower.tail = FALSE`.

Author(s)

Lopes, J. E.

References

Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

Jones, M. C. (2009). Kumaraswamy's distribution: A beta-type distribution with some tractability advantages. *Statistical Methodology*, 6(1), 70-81. doi:10.1016/j.stamet.2008.04.001

See Also

[qgkw](#) (parent distribution quantile function), [dkw](#), [pkw](#), [rkw](#) (other Kw functions), [qbeta](#), [qunif](#)

Other quantile functions: [qbeta_\(\)](#), [qbkw\(\)](#), [qekw\(\)](#), [qgkw\(\)](#), [qkkw\(\)](#), [qmc\(\)](#)

Examples

```
p <- c(0.1, 0.5, 0.9)
qkw(p, alpha = 2, beta = 3)
qkw(p, alpha = 2, beta = 3, lower.tail = FALSE) # upper-tail quantiles

## qkw() inverts pkw()
all.equal(pkw(qkw(p, alpha = 2, beta = 3), alpha = 2, beta = 3), p)
```

qmc

Quantile Function of the McDonald (Mc)/Beta Power Distribution

Description

Computes the quantile function (inverse CDF) for the McDonald (Mc) distribution (also known as Beta Power) with parameters gamma (γ), delta (δ), and lambda (λ). It finds the value q such that $P(X \leq q) = p$. This distribution is a special case of the Generalized Kumaraswamy (GKw) distribution where $\alpha = 1$ and $\beta = 1$.

Usage

```
qmc(p, gamma = 1, delta = 0, lambda = 1, lower.tail = TRUE, log.p = FALSE)
```

Arguments

<code>p</code>	Vector of probabilities (values between 0 and 1).
<code>gamma</code>	Shape parameter $\gamma > 0$. Can be a scalar or a vector. Default: 1.0.
<code>delta</code>	Shape parameter $\delta \geq 0$. Can be a scalar or a vector. Default: 0.0.
<code>lambda</code>	Shape parameter $\lambda > 0$. Can be a scalar or a vector. Default: 1.0.
<code>lower.tail</code>	Logical; if TRUE (default), probabilities are $p = P(X \leq q)$, otherwise, probabilities are $p = P(X > q)$.
<code>log.p</code>	Logical; if TRUE, probabilities p are given as $\log(p)$. Default: FALSE.

Details

The quantile function $Q(p)$ is the inverse of the CDF $F(q)$. The CDF for the Mc ($\alpha = 1, \beta = 1$) distribution is $F(q) = I_{q^\lambda}(\gamma, \delta + 1)$, where $I_z(a, b)$ is the regularized incomplete beta function (see [pmc](#)).

To find the quantile q , we first invert the Beta function part: let $y = I_p^{-1}(\gamma, \delta + 1)$, where $I_p^{-1}(a, b)$ is the inverse computed via [qbeta](#). We then solve $q^\lambda = y$ for q , yielding the quantile function:

$$Q(p) = [I_p^{-1}(\gamma, \delta + 1)]^{1/\lambda}$$

The function uses this formula, calculating $I_p^{-1}(\gamma, \delta + 1)$ via `qbeta(p, gamma, delta + 1, ...)` while respecting the `lower.tail` and `log.p` arguments. This is equivalent to the general GKw quantile function ([qgkw](#)) evaluated with $\alpha = 1, \beta = 1$.

Value

A vector of quantiles corresponding to the given probabilities `p`. The length of the result is determined by the recycling rule applied to the arguments (`p`, `gamma`, `delta`, `lambda`). Returns:

- 0 for `p = 0` (or `p = -Inf` if `log.p = TRUE`, when `lower.tail = TRUE`).
- 1 for `p = 1` (or `p = 0` if `log.p = TRUE`, when `lower.tail = TRUE`).
- NaN for `p < 0` or `p > 1` (or corresponding log scale).
- An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Boundary return values are adjusted accordingly for `lower.tail = FALSE`.

Author(s)

Lopes, J. E.

References

- McDonald, J. B. (1984). Some generalized functions for the size distribution of income. *Econometrica*, 52(3), 647-663. doi:10.2307/1913469
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

See Also

[qgkw](#) (parent distribution quantile function), [dmc](#), [pmc](#), [rmc](#) (other Mc functions), [qbeta](#)

Other quantile functions: [qbeta_\(\)](#), [qbkw\(\)](#), [qekw\(\)](#), [qgkw\(\)](#), [qkkw\(\)](#), [qkw\(\)](#)

Examples

```
p <- c(0.1, 0.5, 0.9)
qmc(p, gamma = 0.5, delta = 5, lambda = 3)
# upper-tail quantiles
qmc(p, gamma = 0.5, delta = 5, lambda = 3, lower.tail = FALSE)

## qmc() inverts pmc()
all.equal(pmc(qmc(p, gamma = 0.5, delta = 5, lambda = 3), gamma = 0.5,
  delta = 5, lambda = 3), p)
```

rbeta_	<i>Random Number Generation for the Beta Distribution (gamma, delta+1 Parameterization)</i>
--------	---

Description

Generates random deviates from the standard Beta distribution, using a parameterization common in generalized distribution families. The distribution is parameterized by gamma (γ) and delta (δ), corresponding to the standard Beta distribution with shape parameters shape1 = gamma and shape2 = delta + 1. This is a special case of the Generalized Kumaraswamy (GKw) distribution where $\alpha = 1$, $\beta = 1$, and $\lambda = 1$.

Usage

```
rbeta_(n, gamma = 1, delta = 0)
```

Arguments

n	Number of observations. If <code>length(n) > 1</code> , the length is taken to be the number required. Must be a non-negative integer.
gamma	First shape parameter (shape1), $\gamma > 0$. Can be a scalar or a vector. Default: 1.0.
delta	Second shape parameter is delta + 1 (shape2), requires $\delta \geq 0$ so that shape2 ≥ 1 . Can be a scalar or a vector. Default: 0.0 (leading to shape2 = 1, i.e., Uniform).

Details

This function generates samples from a Beta distribution with parameters shape1 = gamma and shape2 = delta + 1. It is equivalent to calling `stats::rbeta(n, shape1 = gamma, shape2 = delta + 1)`.

This distribution arises as a special case of the five-parameter Generalized Kumaraswamy (GKw) distribution ([rgkw](#)) obtained by setting $\alpha = 1$, $\beta = 1$, and $\lambda = 1$. It is therefore also equivalent to the McDonald (Mc)/Beta Power distribution ([rmc](#)) with $\lambda = 1$.

The function likely calls R's underlying `rbeta` function but ensures consistent parameter recycling and handling within the C++ environment, matching the style of other functions in the related families.

Value

A numeric vector of length n containing random deviates from the $\text{Beta}(\gamma, \delta + 1)$ distribution, with values in $(0, 1)$. The length of the result is determined by n and the recycling rule applied to the parameters (γ , δ). An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Author(s)

Lopes, J. E.

References

- Johnson, N. L., Kotz, S., & Balakrishnan, N. (1995). *Continuous Univariate Distributions, Volume 2* (2nd ed.). Wiley.
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745
- Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer-Verlag.

See Also

[rbeta](#) (standard R implementation), [rgkw](#) (parent distribution random generation), [rmc](#) (McDonald/Beta Power random generation), [dbeta_](#), [pbeta_](#), [qbeta_](#).

Other random generation functions: [rbkw\(\)](#), [rekw\(\)](#), [rgkw\(\)](#), [rkkw\(\)](#), [rkw\(\)](#), [rmc\(\)](#)

Examples

```
set.seed(123)
x <- rbeta_(1000, gamma = 2, delta = 3)
summary(x)

## The sample follows the distribution
hist(x, breaks = 30, freq = FALSE, main = "")
curve(dbeta_(x, gamma = 2, delta = 3), add = TRUE)
ks.test(x, pbeta_, gamma = 2, delta = 3)
```

rbkw

Random Number Generation for the Beta-Kumaraswamy (BKw) Distribution

Description

Generates random deviates from the Beta-Kumaraswamy (BKw) distribution with parameters α (α), β (β), γ (γ), and δ (δ). This distribution is a special case of the Generalized Kumaraswamy (GKw) distribution where the parameter $\lambda = 1$.

Usage

```
rbkw(n, alpha = 1, beta = 1, gamma = 1, delta = 0)
```

Arguments

n	Number of observations. If <code>length(n) > 1</code> , the length is taken to be the number required. Must be a non-negative integer.
alpha	Shape parameter $\alpha > 0$. Can be a scalar or a vector. Default: 1.0.
beta	Shape parameter $\beta > 0$. Can be a scalar or a vector. Default: 1.0.
gamma	Shape parameter $\gamma > 0$. Can be a scalar or a vector. Default: 1.0.
delta	Shape parameter $\delta \geq 0$. Can be a scalar or a vector. Default: 0.0.

Details

The generation method uses the relationship between the GKw distribution and the Beta distribution. The general procedure for GKw ([rgkw](#)) is: If $W \sim \text{Beta}(\gamma, \delta + 1)$, then $X = \{1 - [1 - W^{1/\lambda}]^{1/\beta}\}^{1/\alpha}$ follows the $\text{GKw}(\alpha, \beta, \gamma, \delta, \lambda)$ distribution.

For the BKw distribution, $\lambda = 1$. Therefore, the algorithm simplifies to:

1. Generate $V \sim \text{Beta}(\gamma, \delta + 1)$ using [rbeta](#).
2. Compute the BKw variate $X = \{1 - (1 - V)^{1/\beta}\}^{1/\alpha}$.

This procedure is implemented efficiently, handling parameter recycling as needed.

Value

A vector of length n containing random deviates from the BKw distribution. The length of the result is determined by n and the recycling rule applied to the parameters (alpha, beta, gamma, delta). An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Author(s)

Lopes, J. E.

References

- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360
- Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer-Verlag. (General methods for random variate generation).

See Also

[rgkw](#) (parent distribution random generation), [dbkw](#), [pbkw](#), [qbkw](#) (other BKw functions), [rbeta](#)

Other random generation functions: [rbeta_\(\)](#), [rekw\(\)](#), [rgkw\(\)](#), [rkkw\(\)](#), [rkw\(\)](#), [rmc\(\)](#)

Examples

```
set.seed(123)
x <- rbkw(1000, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5)
summary(x)

## The sample follows the distribution
hist(x, breaks = 30, freq = FALSE, main = "")
curve(dbkw(x, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5), add = TRUE)
ks.test(x, pbkw, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5)
```

rekw

Random Number Generation for the Exponentiated Kumaraswamy (EKw) Distribution

Description

Generates random deviates from the Exponentiated Kumaraswamy (EKw) distribution with parameters α (α), β (β), and λ (λ). This distribution is a special case of the Generalized Kumaraswamy (GKw) distribution where $\gamma = 1$ and $\delta = 0$.

Usage

```
rekw(n, alpha = 1, beta = 1, lambda = 1)
```

Arguments

n	Number of observations. If <code>length(n) > 1</code> , the length is taken to be the number required. Must be a non-negative integer.
alpha	Shape parameter $\alpha > 0$. Can be a scalar or a vector. Default: 1.0.
beta	Shape parameter $\beta > 0$. Can be a scalar or a vector. Default: 1.0.
lambda	Shape parameter $\lambda > 0$ (exponent parameter). Can be a scalar or a vector. Default: 1.0.

Details

The generation method uses the inverse transform (quantile) method. That is, if U is a random variable following a standard Uniform distribution on $(0, 1)$, then $X = Q(U)$ follows the EKw distribution, where $Q(u)$ is the EKw quantile function ([qekw](#)):

$$Q(u) = \left\{ 1 - \left[1 - u^{1/\lambda} \right]^{1/\beta} \right\}^{1/\alpha}$$

This is computationally equivalent to the general GKw generation method ([rgkw](#)) when specialized for $\gamma = 1, \delta = 0$, as the required Beta(1, 1) random variate is equivalent to a standard Uniform(0, 1) variate. The implementation generates U using [runif](#) and applies the transformation above.

Value

A vector of length n containing random deviates from the EKw distribution. The length of the result is determined by n and the recycling rule applied to the parameters (α , β , λ). An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Author(s)

Lopes, J. E.

References

- Nadarajah, S., Cordeiro, G. M., & Ortega, E. M. (2012). The exponentiated Kumaraswamy distribution. *Journal of the Franklin Institute*, 349(3),
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360
- Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer-Verlag. (General methods for random variate generation).

See Also

[rgkw](#) (parent distribution random generation), [dek](#)[w](#), [pek](#)[w](#), [qek](#)[w](#) (other EKw functions), [runif](#)

Other random generation functions: [rbeta_\(\)](#), [rbkw\(\)](#), [rgkw\(\)](#), [rkkw\(\)](#), [rkw\(\)](#), [rmc\(\)](#)

Examples

```
set.seed(123)
x <- rekw(1000, alpha = 2, beta = 3, lambda = 1.2)
summary(x)

## The sample follows the distribution
hist(x, breaks = 30, freq = FALSE, main = "")
curve(dekw(x, alpha = 2, beta = 3, lambda = 1.2), add = TRUE)
ks.test(x, pek, alpha = 2, beta = 3, lambda = 1.2)
```

rgkw	<i>Random Number Generation for the Generalized Kumaraswamy Distribution</i>
------	--

Description

Generates random deviates from the five-parameter Generalized Kumaraswamy (GKw) distribution defined on the interval (0, 1).

Usage

```
rgkw(n, alpha = 1, beta = 1, gamma = 1, delta = 0, lambda = 1)
```

Arguments

n	Number of observations. If <code>length(n) > 1</code> , the length is taken to be the number required. Must be a non-negative integer.
alpha	Shape parameter $\alpha > 0$. Can be a scalar or a vector. Default: 1.0.
beta	Shape parameter $\beta > 0$. Can be a scalar or a vector. Default: 1.0.
gamma	Shape parameter $\gamma > 0$. Can be a scalar or a vector. Default: 1.0.
delta	Shape parameter $\delta \geq 0$. Can be a scalar or a vector. Default: 0.0.
lambda	Shape parameter $\lambda > 0$. Can be a scalar or a vector. Default: 1.0.

Details

The generation method relies on the transformation property: if $V \sim \text{Beta}(\gamma, \delta + 1)$, then the random variable X defined as

$$X = \left\{ 1 - \left[1 - V^{1/\lambda} \right]^{1/\beta} \right\}^{1/\alpha}$$

follows the $\text{GKw}(\alpha, \beta, \gamma, \delta, \lambda)$ distribution.

The algorithm proceeds as follows:

1. Generate V from `stats::rbeta(n, shape1 = gamma, shape2 = delta + 1)`.
2. Calculate $v = V^{1/\lambda}$.
3. Calculate $w = (1 - v)^{1/\beta}$.
4. Calculate $x = (1 - w)^{1/\alpha}$.

Parameters (alpha, beta, gamma, delta, lambda) are recycled to match the length required by n. Numerical stability is maintained by handling potential edge cases during the transformations.

Value

A vector of length n containing random deviates from the GKw distribution. The length of the result is determined by n and the recycling rule applied to the parameters (α , β , γ , δ , λ). An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Author(s)

Lopes, J. E.

References

- Carrasco, J. M. F., Ferrari, S. L. P., & Cordeiro, G. M. (2010). A new generalized Kumaraswamy distribution. *arXiv preprint arXiv:1004.0911*. doi:10.48550/arXiv.1004.0911
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360

See Also

[dgkw](#), [pgkw](#), [qgkw](#), [rbeta](#), [set.seed](#)

Other random generation functions: [rbeta_\(\)](#), [rbkw\(\)](#), [rekw\(\)](#), [rkkw\(\)](#), [rkw\(\)](#), [rmc\(\)](#)

Examples

```
set.seed(123)
x <- rgkw(1000, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2)
summary(x)

## The sample follows the distribution
hist(x, breaks = 30, freq = FALSE, main = "")
curve(dgkw(x, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2),
      add = TRUE)
ks.test(x, pgkw, alpha = 2, beta = 3, gamma = 1.5, delta = 0.5, lambda = 1.2)
```

Description

Generates random deviates from the Kumaraswamy-Kumaraswamy (KKw) distribution with parameters α (α), β (β), δ (δ), and λ (λ). This distribution is a special case of the Generalized Kumaraswamy (GKw) distribution where the parameter $\gamma = 1$.

Usage

```
rkkw(n, alpha = 1, beta = 1, delta = 0, lambda = 1)
```

Arguments

n	Number of observations. If <code>length(n) > 1</code> , the length is taken to be the number required. Must be a non-negative integer.
alpha	Shape parameter $\alpha > 0$. Can be a scalar or a vector. Default: 1.0.
beta	Shape parameter $\beta > 0$. Can be a scalar or a vector. Default: 1.0.
delta	Shape parameter $\delta \geq 0$. Can be a scalar or a vector. Default: 0.0.
lambda	Shape parameter $\lambda > 0$. Can be a scalar or a vector. Default: 1.0.

Details

The generation method uses the inverse transform method based on the quantile function ([qkkw](#)). The KKw quantile function is:

$$Q(p) = \left[1 - \left\{ 1 - \left[1 - (1 - p)^{1/(\delta+1)} \right]^{1/\lambda} \right\}^{1/\beta} \right]^{1/\alpha}$$

Random deviates are generated by evaluating $Q(p)$ where p is a random variable following the standard Uniform distribution on (0, 1) ([runif](#)).

This is equivalent to the general method for the GKw distribution ([rgkw](#)) specialized for $\gamma = 1$. The GKw method generates $W \sim \text{Beta}(\gamma, \delta + 1)$ and then applies transformations. When $\gamma = 1$, $W \sim \text{Beta}(1, \delta + 1)$, which can be generated via $W = 1 - V^{1/(\delta+1)}$ where $V \sim \text{Unif}(0, 1)$. Substituting this W into the GKw transformation yields the same result as evaluating $Q(1 - V)$ above (noting $p = 1 - V$ is also Uniform).

Value

A vector of length `n` containing random deviates from the KKw distribution. The length of the result is determined by `n` and the recycling rule applied to the parameters (`alpha`, `beta`, `delta`, `lambda`). An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Author(s)

Lopes, J. E.

References

- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:10.1080/00949650903530745
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:10.1016/00221694(80)900360
- Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer-Verlag. (General methods for random variate generation).

See Also

[rgkw](#) (parent distribution random generation), [dkkw](#), [pkkw](#), [qkkw](#), [runif](#), [rbeta](#)

Other random generation functions: [rbeta_\(\)](#), [rbkw\(\)](#), [rekw\(\)](#), [rgkw\(\)](#), [rkW\(\)](#), [rmc\(\)](#)

Examples

```
set.seed(123)
x <- rkkw(1000, alpha = 2, beta = 3, delta = 0.5, lambda = 1.2)
summary(x)

## The sample follows the distribution
hist(x, breaks = 30, freq = FALSE, main = "")
curve(dkkw(x, alpha = 2, beta = 3, delta = 0.5, lambda = 1.2), add = TRUE)
ks.test(x, pkkw, alpha = 2, beta = 3, delta = 0.5, lambda = 1.2)
```

rkW

Random Number Generation for the Kumaraswamy (Kw) Distribution

Description

Generates random deviates from the two-parameter Kumaraswamy (Kw) distribution with shape parameters α and β .

Usage

```
rkW(n, alpha = 1, beta = 1)
```

Arguments

n	Number of observations. If <code>length(n) > 1</code> , the length is taken to be the number required. Must be a non-negative integer.
alpha	Shape parameter $\alpha > 0$. Can be a scalar or a vector. Default: 1.0.
beta	Shape parameter $\beta > 0$. Can be a scalar or a vector. Default: 1.0.

Details

The generation method uses the inverse transform (quantile) method. That is, if U is a random variable following a standard Uniform distribution on $(0, 1)$, then $X = Q(U)$ follows the Kw distribution, where $Q(p)$ is the Kw quantile function ([qkw](#)):

$$Q(p) = \left\{ 1 - (1 - p)^{1/\beta} \right\}^{1/\alpha}$$

The implementation generates U using [runif](#) and applies this transformation. This is equivalent to the general GKw generation method ([rgkw](#)) evaluated at $\gamma = 1, \delta = 0, \lambda = 1$.

Value

A vector of length n containing random deviates from the Kw distribution, with values in $(0, 1)$. The length of the result is determined by n and the recycling rule applied to the parameters (α , β). An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Author(s)

Lopes, J. E.

References

- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:[10.1016/00221694\(80\)900360](https://doi.org/10.1016/00221694(80)900360)
- Jones, M. C. (2009). Kumaraswamy's distribution: A beta-type distribution with some tractability advantages. *Statistical Methodology*, 6(1), 70-81. doi:[10.1016/j.stamet.2008.04.001](https://doi.org/10.1016/j.stamet.2008.04.001)
- Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer-Verlag. (General methods for random variate generation).

See Also

[rgkw](#) (parent distribution random generation), [dkw](#), [pkw](#), [qkw](#) (other Kw functions), [runif](#)
 Other random generation functions: [rbeta_\(\)](#), [rbkw\(\)](#), [rekw\(\)](#), [rgkw\(\)](#), [rkkw\(\)](#), [rmc\(\)](#)

Examples

```
set.seed(123)
x <- rkw(1000, alpha = 2, beta = 3)
summary(x)

## The sample follows the distribution
hist(x, breaks = 30, freq = FALSE, main = "")
curve(dkw(x, alpha = 2, beta = 3), add = TRUE)
ks.test(x, pkw, alpha = 2, beta = 3)
```

rmc

Random Number Generation for the McDonald (Mc)/Beta Power Distribution

Description

Generates random deviates from the McDonald (Mc) distribution (also known as Beta Power) with parameters γ (γ), δ (δ), and λ (λ). This distribution is a special case of the Generalized Kumaraswamy (GKw) distribution where $\alpha = 1$ and $\beta = 1$.

Usage

```
rmc(n, gamma = 1, delta = 0, lambda = 1)
```

Arguments

n	Number of observations. If <code>length(n) > 1</code> , the length is taken to be the number required. Must be a non-negative integer.
gamma	Shape parameter $\gamma > 0$. Can be a scalar or a vector. Default: 1.0.
delta	Shape parameter $\delta \geq 0$. Can be a scalar or a vector. Default: 0.0.
lambda	Shape parameter $\lambda > 0$. Can be a scalar or a vector. Default: 1.0.

Details

The generation method uses the relationship between the GKw distribution and the Beta distribution. The general procedure for GKw ([rgkw](#)) is: If $W \sim \text{Beta}(\gamma, \delta + 1)$, then $X = \{1 - [1 - W^{1/\lambda}]^{1/\beta}\}^{1/\alpha}$ follows the $\text{GKw}(\alpha, \beta, \gamma, \delta, \lambda)$ distribution.

For the Mc distribution, $\alpha = 1$ and $\beta = 1$. Therefore, the algorithm simplifies significantly:

1. Generate $U \sim \text{Beta}(\gamma, \delta + 1)$ using [rbeta](#).
2. Compute the Mc variate $X = U^{1/\lambda}$.

This procedure is implemented efficiently, handling parameter recycling as needed.

Value

A vector of length `n` containing random deviates from the Mc distribution, with values in (0, 1). The length of the result is determined by `n` and the recycling rule applied to the parameters (`gamma`, `delta`, `lambda`). An out-of-bound or missing parameter is an error, not a return value: the wrapper stops with a message naming the parameter. An infinite parameter is not currently intercepted there and reaches the C++ layer, which treats it as invalid.

Author(s)

Lopes, J. E.

References

- McDonald, J. B. (1984). Some generalized functions for the size distribution of income. *Econometrica*, 52(3), 647-663. doi:[10.2307/1913469](#)
- Cordeiro, G. M., & de Castro, M. (2011). A new family of generalized distributions. *Journal of Statistical Computation and Simulation*, 81(7), 883-898. doi:[10.1080/00949650903530745](#)
- Kumaraswamy, P. (1980). A generalized probability density function for double-bounded random processes. *Journal of Hydrology*, 46(1-2), 79-88. doi:[10.1016/00221694\(80\)900360](#)
- Devroye, L. (1986). *Non-Uniform Random Variate Generation*. Springer-Verlag. (General methods for random variate generation).

See Also

[rgkw](#) (parent distribution random generation), [dmc](#), [pmc](#), [qmc](#) (other Mc functions), [rbeta](#)

Other random generation functions: [rbeta_\(\)](#), [rbkw\(\)](#), [rekw\(\)](#), [rgkw\(\)](#), [rkkw\(\)](#), [rkw\(\)](#)

Examples

```
set.seed(123)
x <- rmc(1000, gamma = 0.5, delta = 5, lambda = 3)
summary(x)

## The sample follows the distribution
hist(x, breaks = 30, freq = FALSE, main = "")
curve(dmc(x, gamma = 0.5, delta = 5, lambda = 3), add = TRUE)
ks.test(x, pmc, gamma = 0.5, delta = 5, lambda = 3)
```

Index

* Hessian functions

hsbeta, 35
hsbkw, 37
hsekw, 39
hsgkw, 41
hskkw, 42
hskw, 44
hsmc, 46

* beta-kumaraswamy

dbkw, 10
grbkw, 24
hsbkw, 37
llbkw, 50
pbkw, 62
qbkw, 75
rbkw, 87

* beta

dbeta_, 8
grbeta, 22
hsbeta, 35
llbeta, 48
pbeta_, 61
qbeta_, 73
rbeta_, 86

* cumulative distribution functions

pbeta_, 61
pbkw, 62
pekwa, 64
pgkw, 66
pkkw, 68
pkw, 70
pmc, 71

* density functions

dbeta_, 8
dbkw, 10
dekw, 11
dgkw, 13
dkkw, 15
dkw, 17

dmc, 18

* distribution

dbeta_, 8
dbkw, 10
dekw, 11
dgkw, 13
dkkw, 15
dkw, 17
dmc, 18
grbeta, 22
grbkw, 24
grekw, 26
grgkw, 28
grkkw, 30
grkw, 32
grmc, 33
hsbeta, 35
hsbkw, 37
hsekw, 39
hsgkw, 41
hskkw, 42
hskw, 44
hsmc, 46
llbeta, 48
llbkw, 50
llekw, 52
llgkw, 53
llkkw, 55
llkw, 57
llmc, 59
pbeta_, 61
pbkw, 62
pekwa, 64
pgkw, 66
pkkw, 68
pkw, 70
pmc, 71
qbeta_, 73
qbkw, 75

- qekw, 77
- qgkw, 78
- qkkw, 81
- qkw, 83
- qmc, 84
- rbeta_, 86
- rbkw, 87
- rekw, 89
- rgkw, 91
- rkkw, 92
- rkw, 94
- rmc, 95
- * exponentiated kumaraswamy**
 - dekw, 11
 - grekw, 26
 - hsekw, 39
 - llekw, 52
 - pekw, 64
 - qekw, 77
 - rekw, 89
- * generalized kumaraswamy**
 - dgkw, 13
 - grgkw, 28
 - hsgkw, 41
 - llgkw, 53
 - pgkw, 66
 - qgkw, 78
 - rgkw, 91
- * gradient functions**
 - grbeta, 22
 - grbkw, 24
 - grekw, 26
 - grgkw, 28
 - grkkw, 30
 - grkw, 32
 - grmc, 33
- * kumaraswamy-kumaraswamy**
 - dkkw, 15
 - grkkw, 30
 - hskkw, 42
 - llkkw, 55
 - pkkw, 68
 - qkkw, 81
 - rkkw, 92
- * kumaraswamy**
 - dkw, 17
 - grkw, 32
 - hskw, 44
 - llkw, 57
 - pkw, 70
 - qkw, 83
 - rkw, 94
- * log-likelihood functions**
 - llbeta, 48
 - llbkw, 50
 - llekw, 52
 - llgkw, 53
 - llkkw, 55
 - llkw, 57
 - llmc, 59
- * mcdonald**
 - dmc, 18
 - grmc, 33
 - hsmc, 46
 - llmc, 59
 - pmc, 71
 - qmc, 84
 - rmc, 95
- * optimize**
 - grbeta, 22
 - grbkw, 24
 - grekw, 26
 - grgkw, 28
 - grkkw, 30
 - grkw, 32
 - grmc, 33
 - hsbeta, 35
 - hsbkw, 37
 - hsekw, 39
 - hsgkw, 41
 - hskkw, 42
 - hskw, 44
 - hsmc, 46
 - llbeta, 48
 - llbkw, 50
 - llekw, 52
 - llgkw, 53
 - llkkw, 55
 - llkw, 57
 - llmc, 59
- * quantile functions**
 - qbeta_, 73
 - qbkw, 75
 - qekw, 77
 - qgkw, 78
 - qkkw, 81

- qkw, 83
- qmc, 84
- * **random generation functions**
 - rbeta_, 86
 - rbkw, 87
 - rekw, 89
 - rgkw, 91
 - rkkw, 92
 - rkw, 94
 - rmc, 95
- Beta, 7
- beta, 8, 10, 13, 19, 37, 41, 49, 50, 54, 59
- dbeta, 7–10, 12, 14, 16–19
- dbeta_, 4, 8, 23, 36, 49, 62, 74, 87
- dbeta_(), 11, 12, 14, 16, 18, 19
- dbkw, 4, 10, 25, 38, 51, 64, 76, 89
- dbkw(), 9, 12, 14, 16, 18, 19
- dekw, 4, 11, 17, 18, 27, 40, 52, 53, 65, 78, 90
- dekw(), 9, 11, 14, 16, 18, 19
- dgkw, 3, 4, 8–12, 13, 15–19, 29, 41, 42, 50, 54–56, 67, 80, 92
- dgkw(), 9, 11, 12, 16, 18, 19
- digamma, 23, 25, 28, 29, 34
- dkkw, 4, 15, 17, 18, 31, 44, 57, 69, 82, 94
- dkkw(), 9, 11, 12, 14, 18, 19
- dkw, 4, 17, 33, 45, 46, 58, 71, 84, 95
- dkw(), 9, 11, 12, 14, 16, 19
- dmc, 4, 8, 9, 18, 34, 47, 48, 59, 60, 72, 85, 97
- dmc(), 9, 11, 12, 14, 16, 18
- gkwdist (gkwdist-package), 3
- gkwdist-package, 3
- gkwgetstartvalues, 5, 6, 20
- grad, 23, 25, 27, 29, 31, 33, 34
- grbeta, 5, 22, 36, 49
- grbeta(), 25, 27, 29, 31, 33, 34
- grbkw, 5, 24, 38, 51
- grbkw(), 23, 27, 29, 31, 33, 34
- grekw, 5, 26, 40, 53
- grekw(), 23, 25, 29, 31, 33, 34
- grgkw, 5, 23, 25, 27, 28, 31–34, 42, 55
- grgkw(), 23, 25, 27, 31, 33, 34
- grkkw, 5, 30, 44, 57
- grkkw(), 23, 25, 27, 29, 33, 34
- grkw, 5, 32, 46, 58
- grkw(), 23, 25, 27, 29, 31, 34
- grmc, 5, 23, 33, 48, 60
- grmc(), 23, 25, 27, 29, 31, 33
- hessian, 36, 38, 40, 42, 44, 46, 48
- hsbeta, 5, 23, 35, 49
- hsbeta(), 38, 40, 42, 44, 46, 48
- hsbkw, 5, 25, 37, 51
- hsbkw(), 36, 40, 42, 44, 46, 48
- hsekw, 5, 27, 39, 53
- hsekw(), 36, 38, 42, 44, 46, 48
- hsgkw, 5, 29, 36, 38–40, 41, 43–46, 48, 55
- hsgkw(), 36, 38, 40, 44, 46, 48
- hskkw, 5, 31, 42, 57
- hskkw(), 36, 38, 40, 42, 46, 48
- hskw, 5, 33, 44, 58
- hskw(), 36, 38, 40, 42, 44, 48
- hsmc, 5, 34, 36, 46, 60
- hsmc(), 36, 38, 40, 42, 44, 46
- integrate, 14
- lbeta, 49, 51, 54, 55, 60
- lgamma, 49, 59
- llbeta, 5, 21, 23, 36, 48
- llbeta(), 51, 53, 55, 57, 58, 60
- llbkw, 5, 21, 25, 37, 38, 50
- llbkw(), 49, 53, 55, 57, 58, 60
- llekw, 5, 21, 27, 39, 40, 52
- llekw(), 49, 51, 55, 57, 58, 60
- llgkw, 5, 21, 29, 42, 49, 51, 52, 53, 53, 56–58, 60
- llgkw(), 49, 51, 53, 57, 58, 60
- llkkw, 5, 21, 31, 43, 44, 55
- llkkw(), 49, 51, 53, 55, 58, 60
- llkw, 5, 21, 33, 46, 57
- llkw(), 49, 51, 53, 55, 57, 60
- llmc, 5, 21, 34, 48, 49, 59
- llmc(), 49, 51, 53, 55, 57, 58
- log1p, 54, 55
- optim, 4, 6, 7, 21, 23, 25, 27, 29, 31, 33, 34, 36, 38, 40, 42, 44, 46, 48, 49, 51–53, 55–60
- pbeta, 62–65, 67–69, 71, 72
- pbeta_, 4, 9, 23, 36, 49, 61, 74, 87
- pbeta_(), 64, 65, 67, 69, 71, 72
- pbkw, 4, 11, 51, 62, 75, 76, 89
- pbkw(), 62, 65, 67, 69, 71, 72
- pekw, 4, 12, 53, 64, 70, 71, 77, 78, 90

pekw(), 62, 64, 67, 69, 71, 72
pgkw, 4, 14, 55, 61–65, 66, 68–72, 80, 92
pgkw(), 62, 64, 65, 69, 71, 72
pkkw, 4, 16, 57, 68, 70, 71, 81, 82, 94
pkkw(), 62, 64, 65, 67, 71, 72
pkw, 4, 18, 58, 70, 83, 84, 95
pkw(), 62, 64, 65, 67, 69, 72
pmc, 4, 19, 60–62, 71, 85, 97
pmc(), 62, 64, 65, 67, 69, 71

qbeta, 74–76, 79, 80, 82, 84, 85
qbeta_, 4, 9, 23, 36, 49, 62, 73, 87
qbeta_(), 76, 78, 80, 82, 84, 85
qbkw, 4, 11, 51, 64, 75, 89
qbkw(), 74, 78, 80, 82, 84, 85
qekw, 4, 12, 53, 65, 77, 89, 90
qekw(), 74, 76, 80, 82, 84, 85
qgkw, 4, 14, 55, 67, 73, 74, 76, 77, 78, 78,
82–85, 92
qgkw(), 74, 76, 78, 82, 84, 85
qkkw, 4, 16, 57, 69, 81, 93, 94
qkkw(), 74, 76, 78, 80, 84, 85
qkw, 4, 18, 58, 71, 83, 94, 95
qkw(), 74, 76, 78, 80, 82, 85
qmc, 4, 19, 60, 72–74, 84, 97
qmc(), 74, 76, 78, 80, 82, 84
qunif, 78, 84

rbeta, 87–89, 92, 94, 96, 97
rbeta_, 4, 9, 23, 36, 49, 62, 74, 86
rbeta_(), 89, 90, 92, 94, 95, 97
rbkw, 4, 11, 51, 64, 76, 87
rbkw(), 87, 90, 92, 94, 95, 97
rekw, 4, 12, 53, 65, 78, 89
rekw(), 87, 89, 92, 94, 95, 97
rgkw, 4, 14, 55, 67, 80, 86–90, 91, 93–97
rgkw(), 87, 89, 90, 94, 95, 97
rkkw, 4, 16, 57, 69, 82, 92
rkkw(), 87, 89, 90, 92, 95, 97
rkw, 4, 18, 58, 71, 84, 94
rkw(), 87, 89, 90, 92, 94, 97
rmc, 4, 19, 60, 72, 85–87, 95
rmc(), 87, 89, 90, 92, 94, 95
runif, 90, 93–95

set.seed, 21, 92

trigamma, 36, 47, 48