Package 'RelDists'

December 22, 2022

Title Estimation for some Reliability Distributions

Version 1.0.0

Description Parameters estimation and linear regression models for Reliability distributions families reviewed by Almalki & Nadarajah (2014) <doi:10.1016/j.ress.2013.11.010> using Generalized Additive Models for Location, Scale and Shape, aka GAMLSS by Rigby & Stasinopoulos (2005) <doi:10.1111/j.1467-9876.2005.00510.x>.

Depends R (>= 3.5.0), survival, EstimationTools (>= 4.0.0)

License GPL-3

URL https://ousuga.github.io/RelDists/

BugReports https://github.com/ousuga/RelDists/issues

Encoding UTF-8

RdMacros Rdpack

Imports gamlss, gamlss.dist, Rdpack, zipfR, BBmisc, lamW, VGAM

LazyData true

Suggests knitr, rmarkdown, viridis, autoimage, gamlss.cens, V8

RoxygenNote 7.2.3

VignetteBuilder knitr

NeedsCompilation no

Author Freddy Hernandez [aut] (<https://orcid.org/0000-0001-7459-3329>),
Olga Usuga [aut] (<https://orcid.org/0000-0003-3062-1820>),
Carmen Patino [aut],
Jaime Mosquera [aut, cre] (<https://orcid.org/0000-0002-1684-4756>),
Amylkar Urrea [aut]

Maintainer Jaime Mosquera <jmosquerag@unal.edu.co>

Repository CRAN

Date/Publication 2022-12-22 20:00:10 UTC

R topics documented:

`AddW` *The Additive Weibull family*

Description

The Additive Weibull distribution

Usage

```
AddW(mu.link = "log", sigma.link = "log", nu.link = "log", tau.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.
`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.
`nu.link`	defines the nu.link, with "log" link as the default for the nu parameter.
`tau.link`	defines the tau.link, with "log" link as the default for the tau parameter.

Details

Additive Weibull distribution with parameters `mu`, `sigma`, `nu` and `tau` has density given by

$f(x) = (\mu\nu x^{\nu-1} + \sigma\tau x^{\tau-1})\exp(-\mu x^{\nu} - \sigma x^{\tau}),$

for x > 0.

Value

Returns a gamlss.family object which can be used to fit a AddW distribution in the `gamlss()` function.

Author(s)

Amylkar Urrea Montoya, <amylkar.urrea@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Xie M, Lai CD (1996). "Reliability analysis using an additive Weibull model with bathtub-shaped failure rate function." *Reliability Engineering and System Safety*, **52**, 83–93. doi:10.1016/0951-8320(95)001492.

See Also

dAddW

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma, nu and tau
# Will not be run this example because high number is cycles
# is needed in order to get good estimates
## Not run:
y <- rAddW(n=100, mu=1.5, sigma=0.2, nu=3, tau=0.8)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, tau.fo=~1, family='AddW',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma, nu and tau
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))
exp(coef(mod, what='tau'))

## End(Not run)

# Example 2
# Generating random values under some model
# Will not be run this example because high number is cycles
# is needed in order to get good estimates
## Not run:
n <- 200
```

```
x1 <- runif(n, min=0.4, max=0.6)
x2 <- runif(n, min=0.4, max=0.6)
mu <- exp(1.67 + -3 * x1)
sigma <- exp(0.69 - 2 * x2)
nu <- 3
tau <- 0.8
x <- rAddW(n=n, mu, sigma, nu, tau)

mod <- gamlss(x~x1, sigma.fo=~x2, nu.fo=~1, tau.fo=~1, family=AddW,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
exp(coef(mod, what="nu"))
exp(coef(mod, what="tau"))

## End(Not run)
```

BGE *The Beta Generalized Exponentiated family*

Description

The Beta Generalized Exponentiated family

Usage

```
BGE(mu.link = "log", sigma.link = "log", nu.link = "log", tau.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.
`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.
`nu.link`	defines the nu.link, with "log" link as the default for the nu parameter.
`tau.link`	defines the tau.link, with "log" link as the default for the tau parameter.

Details

The Beta Generalized Exponentiated distribution with parameters `mu`, `sigma`, `nu` and `tau` has density given by

$f(x) = \frac{\nu\tau}{B(\mu,\sigma)} \exp(-\nu x)(1-\exp(-\nu x))^{\tau\mu-1}(1-(1-\exp(-\nu x))^{\tau})^{\sigma-1},$

for $x > 0$, $\mu > 0$, $\sigma > 0$, $\nu > 0$ and $\tau > 0$.

Value

Returns a gamlss.family object which can be used to fit a BGE distribution in the `gamlss()` function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Barreto-Souza W, Santos AH, Cordeiro GM (2010). "The beta generalized exponential distribution." *Journal of Statistical Computation and Simulation*, **80**(2), 159–172.

See Also

dBGE

Examples

```
# Generating some random values with
# known mu, sigma, nu and tau
y <- rBGE(n=100, mu = 1.5, sigma =1.7, nu=1, tau=1)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, tau.fo=~1, family=BGE,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma, nu and tau
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))
exp(coef(mod, what='tau'))

# Example 2
# Generating random values under some model
n <- 200
x1 <- runif(n, min=0.4, max=0.6)
x2 <- runif(n, min=0.4, max=0.6)
mu <- exp(0.5 - x1)
sigma <- exp(0.8 - x2)
nu <- 1
tau <- 1
x <- rBGE(n=n, mu, sigma, nu, tau)

mod <- gamlss(x~x1, sigma.fo=~x2, nu.fo=~1, tau.fo=~1, family=BGE,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
exp(coef(mod, what="nu"))
exp(coef(mod, what="tau"))
```

CS2e *The Cosine Sine Exponential family*

Description

The Cosine Sine Exponential family

Usage

```
CS2e(mu.link = "log", sigma.link = "log", nu.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.
`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.
`nu.link`	defines the nu.link, with "log" link as the default for the nu parameter.

Details

The Cosine Sine Exponential distribution with parameters `mu`, `sigma` and `nu` has density given by

$$f(x) = \frac{\pi\sigma\mu\exp(\frac{-x}{\nu})}{2\nu[(\mu\sin(\frac{\pi}{2}\exp(\frac{-x}{\nu}))+\sigma\cos(\frac{\pi}{2}\exp(\frac{-x}{\nu}))]^2},$$

for $x > 0$, $\mu > 0$, $\sigma > 0$ and $\nu > 0$.

Value

Returns a gamlss.family object which can be used to fit a CS2e distribution in the `gamlss()` function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Chesneau C, Bakouch HS, Hussain T (2018). "A new class of probability distributions via cosine and sine functions with applications." *Communications in Statistics-Simulation and Computation*, 1–14.

See Also

dCS2e

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma and nu
y <- rCS2e(n=100, mu = 0.1, sigma =1, nu=0.5)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, family='CS2e',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))

# Example 2
# Generating random values under some model
n <- 200
x1 <- runif(n, min=0.45, max=0.55)
x2 <- runif(n, min=0.4, max=0.6)
mu <- exp(0.2 - x1)
sigma <- exp(0.8 - x2)
nu <- 0.5
x <- rCS2e(n=n, mu, sigma, nu)

mod <- gamlss(x~x1, sigma.fo=~x2, nu.fo=~1,family=CS2e,
              control=gamlss.control(n.cyc=50000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
exp(coef(mod, what="nu"))
```

---

dAddW *The Additive Weibull distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the Additive Weibull distribution with parameters mu, sigma, nu and tau.

Usage

```
dAddW(x, mu, sigma, nu, tau, log = FALSE)

pAddW(q, mu, sigma, nu, tau, lower.tail = TRUE, log.p = FALSE)
```

```
qAddW(p, mu, sigma, nu, tau, lower.tail = TRUE, log.p = FALSE)

rAddW(n, mu, sigma, nu, tau)

hAddW(x, mu, sigma, nu, tau)
```

Arguments

`x, q`	vector of quantiles.
`mu`	parameter.
`sigma`	parameter.
`nu`	shape parameter.
`tau`	shape parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
`p`	vector of probabilities.
`n`	number of observations.

Details

Additive Weibull Distribution with parameters `mu`, `sigma`, `nu` and `tau` has density given by

$f(x) = (\mu \nu x^{\nu-1} + \sigma \tau x^{\tau-1}) \exp(-\mu x^{\nu} - \sigma x^{\tau}),$

for x > 0.

Value

`dAddW` gives the density, `pAddW` gives the distribution function, `qAddW` gives the quantile function, `rAddW` generates random deviates and `hAddW` gives the hazard function.

Author(s)

Amylkar Urrea Montoya, `<amylkar.urrea@udea.edu.co>`

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Xie M, Lai CD (1996). "Reliability analysis using an additive Weibull model with bathtub-shaped failure rate function." *Reliability Engineering and System Safety*, **52**, 83–93. doi:10.1016/0951-8320(95)001492.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dAddW(x, mu=1.5, sigma=0.5, nu=3, tau=0.8), from=0.0001, to=2,
      col="red", las=1, ylab="f(x)")

## The cumulative distribution and the Reliability function
par(mfrow=c(1, 2))
curve(pAddW(x, mu=1.5, sigma=0.5, nu=3, tau=0.8),
      from=0.0001, to=2, col="red", las=1, ylab="F(x)")
curve(pAddW(x, mu=1.5, sigma=0.5, nu=3, tau=0.8, lower.tail=FALSE),
      from=0.0001, to=2, col="red", las=1, ylab="R(x)")

## The quantile function
p <- seq(from=0, to=0.99999, length.out=100)
plot(x=qAddW(p, mu=1.5, sigma=0.2, nu=3, tau=0.8), y=p, xlab="Quantile",
     las=1, ylab="Probability")
curve(pAddW(x, mu=1.5, sigma=0.2, nu=3, tau=0.8),
      from=0, add=TRUE, col="red")

## The random function
hist(rAddW(n=10000, mu=1.5, sigma=0.2, nu=3, tau=0.8), freq=FALSE,
     xlab="x", las=1, main="")
curve(dAddW(x, mu=1.5, sigma=0.2, nu=3, tau=0.8),
      from=0.09, to=5, add=TRUE, col="red")

## The Hazard function
curve(hAddW(x, mu=1.5, sigma=0.2, nu=3, tau=0.8), from=0.001, to=1,
      col="red", ylab="Hazard function", las=1)

par(old_par) # restore previous graphical parameters
```

---

dBGE *The Beta Generalized Exponentiated distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the Beta Generalized Exponentiated distribution with parameters `mu`, `sigma`, `nu` and `tau`.

Usage

```
dBGE(x, mu, sigma, nu, tau, log = FALSE)

pBGE(q, mu, sigma, nu, tau, lower.tail = TRUE, log.p = FALSE)

qBGE(p, mu, sigma, nu, tau, lower.tail = TRUE, log.p = FALSE)
```

```
rBGE(n, mu, sigma, nu, tau)

hBGE(x, mu, sigma, nu, tau)
```

Arguments

`x, q`	vector of quantiles.
`mu`	parameter.
`sigma`	parameter.
`nu`	parameter.
`tau`	parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
`p`	vector of probabilities.
`n`	number of observations.

Details

The Beta Generalized Exponentiated Distribution with parameters `mu`, `sigma`, `nu` and `tau` has density given by

$f(x) = \frac{\nu\tau}{B(\mu,\sigma)} \exp(-\nu x)(1-\exp(-\nu x))^{\tau\mu-1}(1-(1-\exp(-\nu x))^{\tau})^{\sigma-1},$

for $x > 0$, $\mu > 0$, $\sigma > 0$, $\nu > 0$ and $\tau > 0$.

Value

`dBGE` gives the density, `pBGE` gives the distribution function, `qBGE` gives the quantile function, `rBGE` generates random deviates and `hBGE` gives the hazard function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Barreto-Souza W, Santos AH, Cordeiro GM (2010). "The beta generalized exponential distribution." *Journal of Statistical Computation and Simulation*, **80**(2), 159–172.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dBGE(x, mu = 1.5, sigma =1.7, nu=1, tau=1), from = 0, to = 3,
      col = "red", las = 1, ylab = "f(x)")
```

```
## The cumulative distribution and the Reliability function
par(mfrow = c(1, 2))
curve(pBGE(x, mu = 1.5, sigma =1.7, nu=1, tau=1), from = 0, to = 6,
      ylim = c(0, 1), col = "red", las = 1, ylab = "F(x)")
curve(pBGE(x, mu = 1.5, sigma =1.7, nu=1, tau=1, lower.tail = FALSE),
      from = 0, to = 6, ylim = c(0, 1), col = "red", las = 1, ylab = "R(x)")

## The quantile function
p <- seq(from = 0, to = 0.99999, length.out = 100)
plot(x = qBGE(p = p, mu = 1.5, sigma =1.7, nu=1, tau=1), y = p,
     xlab = "Quantile", las = 1, ylab = "Probability")
curve(pBGE(x, mu = (1/4), sigma =1, nu=1, tau=2), from = 0, add = TRUE,
      col = "red")

## The random function
hist(rBGE(1000, mu = 1.5, sigma =1.7, nu=1, tau=1), freq = FALSE, xlab = "x",
     ylim = c(0, 1), las = 1, main = "")
curve(dBGE(x, mu = 1.5, sigma =1.7, nu=1, tau=1),  from = 0, add = TRUE,
      col = "red", ylim = c(0, 0.5))

## The Hazard function(
par(mfrow=c(1,1))
curve(hBGE(x, mu = 0.9, sigma =0.5, nu=1, tau=1), from = 0, to = 2,
      col = "red", ylab = "Hazard function", las = 1)

par(old_par) # restore previous graphical parameters
```

`dCS2e` *The Cosine Sine Exponential distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the Cosine Sine Exponential distribution with parameters `mu`, `sigma` and `nu`.

Usage

```
dCS2e(x, mu, sigma, nu, log = FALSE)

pCS2e(q, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

qCS2e(p, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

rCS2e(n, mu, sigma, nu)

hCS2e(x, mu, sigma, nu)
```

Arguments

x, q	vector of quantiles.
mu	parameter.
sigma	parameter.
nu	parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
p	vector of probabilities.
n	number of observations.

Details

The Cosine Sine Exponential Distribution with parameters mu, sigma and nu has density given by

$$f(x) = \frac{\pi \sigma \mu \exp(\frac{-x}{\nu})}{2\nu[(\mu \sin(\frac{\pi}{2} \exp(\frac{-x}{\nu})) + \sigma \cos(\frac{\pi}{2} \exp(\frac{-x}{\nu}))]^2},$$

for $x > 0$, $\mu > 0$, $\sigma > 0$ and $\nu > 0$.

Value

dCS2e gives the density, pCS2e gives the distribution function, qCS2e gives the quantile function, rCS2e generates random deviates and hCS2e gives the hazard function.

Author(s)

Juan Pablo Ramirez

References

Chesneau C, Bakouch HS, Hussain T (2018). “A new class of probability distributions via cosine and sine functions with applications.” *Communications in Statistics-Simulation and Computation*, 1–14.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
par(mfrow=c(1,1))
curve(dCS2e(x, mu=1, sigma=0.1, nu =0.1), from=0, to=1,
      ylim=c(0, 3), col="red", las=1, ylab="f(x)")

## The cumulative distribution and the Reliability function
par(mfrow=c(1, 2))
curve(pCS2e(x, mu=1, sigma=0.1, nu =0.1),
      from=0, to=1,  col="red", las=1, ylab="F(x)")
curve(pCS2e(x, mu=1, sigma=0.1, nu =0.1, lower.tail=FALSE),
      from=0, to=1, col="red", las=1, ylab="R(x)")
```

```
## The quantile function
p <- seq(from=0, to=0.99999, length.out=100)
plot(x=qCS2e(p, mu=0.1, sigma=1, nu=0.1), y=p, xlab="Quantile",
     las=1, ylab="Probability")
curve(pCS2e(x, mu=0.1, sigma=1, nu=0.1), from=0, add=TRUE, col="red")

## The random function
hist(rCS2e(n=10000, mu=0.1, sigma=1, nu=0.1), freq=FALSE,
     xlab="x", las=1, main="")
curve(dCS2e(x, mu=0.1, sigma=1, nu=0.1), from=0, add=TRUE, col="red")

## The Hazard function
par(mfrow=c(1,1))
curve(hCS2e(x, mu=1, sigma=0.1, nu =0.1), from=0, to=1, ylim=c(0, 10),
      col=2, ylab="Hazard function", las=1)

par(old_par) # restore previous graphical parameters
```

---

dEEG *The Extended Exponential Geometric distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the Extended Exponential Geometric distribution with parameters `mu` and `sigma`.

Usage

```
dEEG(x, mu, sigma, log = FALSE)

pEEG(q, mu, sigma, lower.tail = TRUE, log.p = FALSE)

qEEG(p, mu, sigma, lower.tail = TRUE, log.p = FALSE)

rEEG(n, mu, sigma)

hEEG(x, mu, sigma)
```

Arguments

`x, q`	vector of quantiles.
`mu`	parameter.
`sigma`	parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
`p`	vector of probabilities.
`n`	number of observations.

Details

The Extended Exponential Geometric distribution with parameters `mu`, and `sigma`has density given by

$f(x) = \mu\sigma \exp(-\mu x)(1-(1-\sigma)\exp(-\mu x))^{-2},$

for $x > 0$, $\mu > 0$ and $\sigma > 0$.

Value

`dEEG` gives the density, `pEEG` gives the distribution function, `qEEG` gives the quantile function, `rEEG` generates random deviates and `hEEG` gives the hazard function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Adamidis K, Dimitrakopoulou T, Loukas S (2005). "On an extension of the exponential-geometric distribution." *Statistics & probability letters*, **73**(3), 259–269.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
par(mfrow=c(1,1))
curve(dEEG(x, mu = 1, sigma =3), from = 0, to = 10,
      col = "red", las = 1, ylab = "f(x)")

## The cumulative distribution and the Reliability function
par(mfrow = c(1, 2))
curve(pEEG(x, mu = 1, sigma =3), from = 0, to = 10,
      ylim = c(0, 1), col = "red", las = 1, ylab = "F(x)")
curve(pEEG(x, mu = 1, sigma =3, lower.tail = FALSE),
      from = 0, to = 6, ylim = c(0, 1), col = "red", las = 1, ylab = "R(x)")

## The quantile function
p <- seq(from = 0, to = 0.99999, length.out = 100)
plot(x = qEEG(p = p, mu = 1, sigma =0.5), y = p,
     xlab = "Quantile", las = 1, ylab = "Probability")
curve(pEEG(x, mu = 1, sigma =0.5), from = 0, add = TRUE,
      col = "red")

## The random function
hist(rEEG(1000, mu = 1, sigma =1), freq = FALSE, xlab = "x",
     ylim = c(0, 0.9), las = 1, main = "")
curve(dEEG(x, mu = 1, sigma =1),  from = 0, add = TRUE,
      col = "red", ylim = c(0, 0.8))
```

```
## The Hazard function
par(mfrow=c(1,1))
curve(hEEG(x, mu = 1, sigma =0.5), from = 0, to = 2,
      col = "red", ylab = "Hazard function", las = 1)

par(old_par) # restore previous graphical parameters
```

---

dEGG *The four parameter Exponentiated Generalized Gamma distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the four parameter Exponentiated Generalized Gamma distribution with parameters `mu`, `sigma`, `nu` and `tau`.

Usage

```
dEGG(x, mu, sigma, nu, tau, log = FALSE)

pEGG(q, mu, sigma, nu, tau, lower.tail = TRUE, log.p = FALSE)

qEGG(p, mu, sigma, nu, tau, lower.tail = TRUE, log.p = FALSE)

rEGG(n, mu, sigma, nu, tau)

hEGG(x, mu, sigma, nu, tau)
```

Arguments

`x, q`	vector of quantiles.
`mu`	parameter.
`sigma`	parameter.
`nu`	parameter.
`tau`	parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
`p`	vector of probabilities.
`n`	number of observations.

Details

Four-Parameter Exponentiated Generalized Gamma distribution with parameters `mu`, `sigma`, `nu` and `tau` has density given by

$f(x) = \frac{\nu\sigma}{\mu\Gamma(\tau)} \left(\frac{x}{\mu}\right)^{\sigma\tau-1} \exp\left\{-\left(\frac{x}{\mu}\right)^{\sigma}\right\} \left\{\gamma_1\left(\tau, \left(\frac{x}{\mu}\right)^{\sigma}\right)\right\}^{\nu-1},$

for x > 0.

Value

dEGG gives the density, pEGG gives the distribution function, qEGG gives the quantile function, rEGG generates random deviates and hEGG gives the hazard function.

Author(s)

Amylkar Urrea Montoya, <amylkar.urrea@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Gauss M. C, Edwin M.M O, Giovana O. S (2011). "The exponentiated generalized gamma distribution with application to lifetime data." *Journal of Statistical Computation and Simulation*, **81**(7), 827–842. doi:10.1080/00949650903517874.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dEGG(x, mu=0.1, sigma=0.8, nu=10, tau=1.5), from=0.000001, to=1.5, ylim=c(0, 2.5),
      col="red", las=1, ylab="f(x)")

## The cumulative distribution and the Reliability function
par(mfrow=c(1, 2))
curve(pEGG(x, mu=0.1, sigma=0.8, nu=10, tau=1.5),
      from=0.000001, to=1.5, col="red", las=1, ylab="F(x)")
curve(pEGG(x, mu=0.1, sigma=0.8, nu=10, tau=1.5, lower.tail=FALSE),
      from=0.000001, to=1.5, col="red", las=1, ylab="R(x)")

## The quantile function
p <- seq(from=0, to=0.99999, length.out=100)
plot(x=qEGG(p, mu=0.1, sigma=0.8, nu=10, tau=1.5), y=p, xlab="Quantile",
     las=1, ylab="Probability")
curve(pEGG(x, mu=0.1, sigma=0.8, nu=10, tau=1.5),
      from=0.00001, add=TRUE, col="red")

## The random function
hist(rEGG(n=100, mu=0.1, sigma=0.8, nu=10, tau=1.5), freq=FALSE,
     xlab="x", las=1, main="")
curve(dEGG(x, mu=0.1, sigma=0.8, nu=10, tau=1.5),
      from=0.0001, to=2, add=TRUE, col="red")

## The Hazard function
curve(hEGG(x,  mu=0.1, sigma=0.8, nu=10, tau=1.5), from=0.0001, to=1.5,
      col="red", ylab="Hazard function", las=1)

par(old_par) # restore previous graphical parameters
```

dEMWEx *The Exponentiated Modifien Weibull Extension distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the Exponentiated Modifien Weibull Extension distribution with parameters `mu`, `sigma`, `nu` and `tau`.

Usage

```
dEMWEx(x, mu, sigma, nu, tau, log = FALSE)

pEMWEx(q, mu, sigma, nu, tau, lower.tail = TRUE, log.p = FALSE)

qEMWEx(p, mu, sigma, nu, tau, lower.tail = TRUE, log.p = FALSE)

rEMWEx(n, mu, sigma, nu, tau)

hEMWEx(x, mu, sigma, nu, tau)
```

Arguments

`x, q`	vector of quantiles.
`mu`	parameter.
`sigma`	parameter.
`nu`	parameter.
`tau`	parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
`p`	vector of probabilities.
`n`	number of observations.

Details

The Exponentiated Modifien Weibull Extension Distribution with parameters `mu`, `sigma`, `nu` and `tau` has density given by

$f(x) = \nu\sigma\tau(\frac{x}{\mu})^{\sigma-1}\exp((\frac{x}{\mu})^{\sigma} + \nu\mu(1-\exp((\frac{x}{\mu})^{\sigma})))(1-\exp(\nu\mu(1-\exp((\frac{x}{\mu})^{\sigma}))))^{\tau-1},$

for $x > 0$, $\nu > 0$, $\mu > 0$, $\sigma > 0$ and $\tau > 0$.

Value

`dEMWEx` gives the density, `pEMWEx` gives the distribution function, `qEMWEx` gives the quantile function, `rEMWEx` generates random deviates and `hEMWEx` gives the hazard function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Sarhan AM, Apaloo J (2013). "Exponentiated modified Weibull extension distribution." *Reliability Engineering & System Safety*, **112**, 137–144.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dEMWEx(x, mu = 49.046, sigma =3.148, nu=0.00005, tau=0.1), from=0, to=100,
      col = "red", las = 1, ylab = "f(x)")

## The cumulative distribution and the Reliability function
par(mfrow = c(1, 2))
curve(pEMWEx(x, mu = (1/4), sigma =1, nu=1, tau=2), from = 0, to = 1,
      ylim = c(0, 1), col = "red", las = 1, ylab = "F(x)")
curve(pEMWEx(x, mu = (1/4), sigma =1, nu=1, tau=2, lower.tail = FALSE),
      from = 0, to = 1, ylim = c(0, 1), col = "red", las = 1, ylab = "R(x)")

## The quantile function
p <- seq(from = 0, to = 0.99999, length.out = 100)
plot(x = qEMWEx(p = p, mu = 49.046, sigma =3.148, nu=0.00005, tau=0.1), y = p,
     xlab = "Quantile", las = 1, ylab = "Probability")
curve(pEMWEx(x, mu = 49.046, sigma =3.148, nu=0.00005, tau=0.1), from = 0, add = TRUE,
      col = "red")

## The random function
hist(rEMWEx(1000, mu = (1/4), sigma =1, nu=1, tau=2), freq = FALSE, xlab = "x",
     las = 1, main = "")
curve(dEMWEx(x, mu = (1/4), sigma =1, nu=1, tau=2),  from = 0, add = TRUE,
      col = "red", ylim = c(0, 0.5))

## The Hazard function(
par(mfrow=c(1,1))
curve(hEMWEx(x, mu = 49.046, sigma =3.148, nu=0.00005, tau=0.1), from = 0, to = 80,
      col = "red", ylab = "Hazard function", las = 1)

par(old_par) # restore previous graphical parameters
```

---

dEOFNH *The Extended Odd Frechet-Nadarajah-Haghighi*

Description

Density, distribution function, quantile function, random generation and hazard function for the Extended Odd Fr?chet-Nadarajah-Haghighi distribution with parameters `mu`, `sigma`, `nu` and `tau`.

Usage

```
dEOFNH(x, mu, sigma, nu, tau, log = FALSE)

pEOFNH(q, mu, sigma, nu, tau, lower.tail = TRUE, log.p = FALSE)

qEOFNH(p, mu, sigma, nu, tau, lower.tail = TRUE, log.p = FALSE)

rEOFNH(n, mu, sigma, nu, tau)

hEOFNH(x, mu, sigma, nu, tau)
```

Arguments

`x, q`	vector of quantiles.
`mu`	parameter.
`sigma`	parameter.
`nu`	parameter.
`tau`	parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
`p`	vector of probabilities.
`n`	number of observations.

Details

Tthe Extended Odd Frechet-Nadarajah-Haghighi `mu`, `sigma`, `nu` and `tau` has density given by

$f(x) = \frac{\mu\sigma\nu\tau(1+\nu x)^{\sigma-1}e^{(1-(1+\nu x)^{\sigma})}[1-(1-e^{(1-(1+\nu x)^{\sigma})})^{\mu}]^{\tau-1}}{(1-e^{(1-(1+\nu x)^{\sigma})})^{\mu\tau+1}} e^{-[(1-e^{(1-(1+\nu x)^{\sigma})})^{-\mu}-1]^{\tau}},$

for $x > 0$, $\mu > 0$, $\sigma > 0$, $\nu > 0$ and $\tau > 0$.

Value

`dEOFNH` gives the density, `pEOFNH` gives the distribution function, `qEOFNH` gives the quantile function, `rEOFNH` generates random numbers and `hEOFNH` gives the hazard function.

Author(s)

Helber Santiago Padilla

References

Nasiru S (2018). "Extended Odd Fréchet-G Family of Distributions." *Journal of Probability and Statistics*, **2018**.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

##The probability density function
par(mfrow=c(1,1))
 curve(dEOFNH(x, mu=18.5, sigma=5.1, nu=0.1, tau=0.1), from=0, to=10,
     ylim=c(0, 0.25), col="red", las=1, ylab="f(x)")

## The cumulative distribution and the Reliability function
par(mfrow = c(1, 2))
curve(pEOFNH(x,mu=18.5, sigma=5.1, nu=0.1, tau=0.1), from = 0, to = 10,
ylim = c(0, 1), col = "red", las = 1, ylab = "F(x)")
curve(pEOFNH(x, mu=18.5, sigma=5.1, nu=0.1, tau=0.1, lower.tail = FALSE),
from = 0, to = 10, ylim = c(0, 1), col = "red", las = 1, ylab = "R(x)")

##The quantile function
p <- seq(from=0, to=0.99999, length.out=100)
plot(x=qEOFNH(p, mu=18.5, sigma=5.1, nu=0.1, tau=0.1), y=p, xlab="Quantile",
     las=1, ylab="Probability")
curve(pEOFNH(x, mu=18.5, sigma=5.1, nu=0.1, tau=0.1), from=0, add=TRUE, col="red")

##The random function
hist(rEOFNH(n=10000, mu=18.5, sigma=5.1, nu=0.1, tau=0.1), freq=FALSE,
     xlab="x", las=1, main="")
curve(dEOFNH(x, mu=18.5, sigma=5.1, nu=0.1, tau=0.1), from=0, add=TRUE, col="red", ylim=c(0,1.25))

##The Hazard function
par(mfrow=c(1,1))
curve(hEOFNH(x, mu=18.5, sigma=5.1, nu=0.1, tau=0.1), from=0, to=10, ylim=c(0, 1),
     col="red", ylab="Hazard function", las=1)

par(old_par) # restore previous graphical parameters
```

---

dEW *The Exponentiated Weibull distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the exponentiated Weibull distribution with parameters mu, sigma and nu.

Usage

```
dEW(x, mu, sigma, nu, log = FALSE)

pEW(q, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

qEW(p, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)
```

```
rEW(n, mu, sigma, nu)

hEW(x, mu, sigma, nu)
```

Arguments

`x, q`	vector of quantiles.
`mu`	scale parameter.
`sigma, nu`	shape parameters.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
`p`	vector of probabilities.
`n`	number of observations.

Details

The Exponentiated Weibull Distribution with parameters `mu`, `sigma` and `nu` has density given by

$f(x) = \nu\mu\sigma x^{\sigma-1}\exp(-\mu x^{\sigma})(1-\exp(-\mu x^{\sigma}))^{\nu-1}$,

for $x > 0$, $\mu > 0$, $\sigma > 0$ and $\nu > 0$.

Value

`dEW` gives the density, `pEW` gives the distribution function, `qEW` gives the quantile function, `rEW` generates random deviates and `hEW` gives the hazard function.

See Also

EW

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dEW(x, mu=2, sigma=1.5, nu=0.5), from=0, to=2,
      ylim=c(0, 2.5), col="red", las=1, ylab="f(x)")

## The cumulative distribution and the Reliability function
par(mfrow=c(1, 2))
curve(pEW(x, mu=2, sigma=1.5, nu=0.5),
      from=0, to=2,  col="red", las=1, ylab="F(x)")
curve(pEW(x, mu=2, sigma=1.5, nu=0.5, lower.tail=FALSE),
      from=0, to=2, col="red", las=1, ylab="R(x)")

## The quantile function
p <- seq(from=0, to=0.99999, length.out=100)
plot(x=qEW(p, mu=2, sigma=1.5, nu=0.5), y=p, xlab="Quantile",
     las=1, ylab="Probability")
curve(pEW(x, mu=2, sigma=1.5, nu=0.5), from=0, add=TRUE, col="red")
```

```
## The random function
hist(rEW(n=10000, mu=2, sigma=1.5, nu=0.5), freq=FALSE,
     xlab="x", las=1, main="")
curve(dEW(x, mu=2, sigma=1.5, nu=0.5), from=0, add=TRUE, col="red")

## The Hazard function
par(mfrow=c(1,1))
curve(hEW(x, mu=2, sigma=1.5, nu=0.5), from=0, to=2, ylim=c(0, 7),
      col="red", ylab="Hazard function", las=1)

par(old_par) # restore previous graphical parameters
```

`dExW` *The Extended Weibull distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the Extended Weibull distribution with parameters `mu`, `sigma` and `nu`.

Usage

```
dExW(x, mu, sigma, nu, log = FALSE)

pExW(q, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

qExW(p, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

rExW(n, mu, sigma, nu)

hExW(x, mu, sigma, nu)
```

Arguments

`x, q`	vector of quantiles.
`mu`	parameter.
`sigma`	parameter.
`nu`	parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
`p`	vector of probabilities.
`n`	number of observations.

Details

The Extended Weibull distribution with parameters `mu`, `sigma` and `nu` has density given by

$$f(x) = \frac{\mu\sigma\nu x^{\sigma-1} exp(-\mu x^{\sigma})}{[1-(1-\nu)exp(-\mu x^{\sigma})]^2},$$

for x > 0.

Value

`dExW` gives the density, `pExW` gives the distribution function, `qExW` gives the quantile function, `rExW` generates random deviates and `hExW` gives the hazard function.

Author(s)

Amylkar Urrea Montoya, <amylkar.urrea@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Tieling Z, Min X (2007). "Failure Data Analysis with Extended Weibull Distribution." *Communications in Statistics - Simulation and Computation*, **36**, 579–592. doi:10.1080/03610910701236081.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dExW(x, mu=0.3, sigma=2, nu=0.05), from=0.0001, to=2,
      col="red", las=1, ylab="f(x)")

## The cumulative distribution and the Reliability function
par(mfrow=c(1, 2))
curve(pExW(x, mu=0.3, sigma=2, nu=0.05),
      from=0.0001, to=2, col="red", las=1, ylab="F(x)")
curve(pExW(x, mu=0.3, sigma=2, nu=0.05, lower.tail=FALSE),
      from=0.0001, to=2, col="red", las=1, ylab="R(x)")

## The quantile function
p <- seq(from=0, to=0.99999, length.out=100)
plot(x=qExW(p, mu=0.3, sigma=2, nu=0.05), y=p, xlab="Quantile",
     las=1, ylab="Probability")
curve(pExW(x, mu=0.3, sigma=2, nu=0.05),
      from=0, add=TRUE, col="red")

## The random function
hist(rExW(n=10000, mu=0.3, sigma=2, nu=0.05), freq=FALSE,
     xlab="x", ylim=c(0, 2), las=1, main="")
curve(dExW(x, mu=0.3, sigma=2, nu=0.05),
      from=0.001, to=4, add=TRUE, col="red")

## The Hazard function
```

```
curve(hExW(x, mu=0.3, sigma=2, nu=0.05), from=0.001, to=4,
      col="red", ylab="Hazard function", las=1)

par(old_par) # restore previous graphical parameters
```

---

dFWE *The Flexible Weibull Extension distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the Flexible Weibull Extension distribution with parameters mu and sigma.

Usage

```
dFWE(x, mu, sigma, log = FALSE)

pFWE(q, mu, sigma, lower.tail = TRUE, log.p = FALSE)

qFWE(p, mu, sigma, lower.tail = TRUE, log.p = FALSE)

rFWE(n, mu, sigma)

hFWE(x, mu, sigma)
```

Arguments

x, q	vector of quantiles.
mu	parameter.
sigma	parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
p	vector of probabilities.
n	number of observations.

Details

The Flexible Weibull extension with parameters mu and sigma has density given by

$f(x) = (\mu + \sigma/x^2) \exp(\mu x - \sigma/x) \exp(-\exp(\mu x - \sigma/x))$

for x>0.

Value

dFWE gives the density, pFWE gives the distribution function, qFWE gives the quantile function, rFWE generates random deviates and hFWE gives the hazard function.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dFWE(x, mu=0.75, sigma=0.5), from=0, to=3,
      ylim=c(0, 1.7), col="red", las=1, ylab="f(x)")

## The cumulative distribution and the Reliability function
par(mfrow=c(1, 2))
curve(pFWE(x, mu=0.75, sigma=0.5), from=0, to=3,
      col="red", las=1, ylab="F(x)")
curve(pFWE(x, mu=0.75, sigma=0.5, lower.tail=FALSE),
      from=0, to=3, col="red", las=1, ylab="R(x)")

## The quantile function
p <- seq(from=0, to=0.99999, length.out=100)
plot(x=qFWE(p, mu=0.75, sigma=0.5), y=p, xlab="Quantile",
     las=1, ylab="Probability")
curve(pFWE(x, mu=0.75, sigma=0.5), from=0, add=TRUE, col="red")

## The random function
hist(rFWE(n=1000, mu=2, sigma=0.5), freq=FALSE, xlab="x",
     ylim=c(0, 2), las=1, main="")
curve(dFWE(x, mu=2, sigma=0.5), from=0, to=3, add=TRUE, col="red")

## The Hazard function
par(mfrow=c(1,1))
curve(hFWE(x, mu=0.75, sigma=0.5), from=0, to=2, ylim=c(0, 2.5),
      col="red", ylab="Hazard function", las=1)

par(old_par) # restore previous graphical parameters
```

---

dGammaW *The Gamma Weibull distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the Gamma Weibull distribution with parameters mu, sigma, nu and tau.

Usage

```
dGammaW(x, mu, sigma, nu, log = FALSE)

pGammaW(q, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

qGammaW(p, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

rGammaW(n, mu, sigma, nu)
```

```
hGammaW(x, mu, sigma, nu)
```

Arguments

`x, q`	vector of quantiles.
`mu`	parameter.
`sigma`	parameter.
`nu`	parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
`p`	vector of probabilities.
`n`	number of observations.

Details

The Gamma Weibull Distribution with parameters `mu`, `sigma` and `nu` has density given by

$f(x) = \frac{\sigma\mu^{\nu}}{\Gamma(\nu)} x^{\nu\sigma-1} \exp(-\mu x^{\sigma}),$

for $x > 0$, $\mu > 0$, $\sigma \geq 0$ and $\nu > 0$.

Value

`dGammaW` gives the density, `pGammaW` gives the distribution function, `qGammaW` gives the quantile function, `rGammaW` generates random deviates and `hGammaW` gives the hazard function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Stacy EW, others (1962). "A generalization of the gamma distribution." *The Annals of mathematical statistics*, **33**(3), 1187–1192.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dGammaW(x, mu = 0.5, sigma = 2, nu=1), from = 0, to = 6,
      col = "red", las = 1, ylab = "f(x)")

## The cumulative distribution and the Reliability function
par(mfrow = c(1, 2))
curve(pGammaW(x, mu = 0.5, sigma = 2, nu=1), from = 0, to = 3,
```

```
ylim = c(0, 1), col = "red", las = 1, ylab = "F(x)")
curve(pGammaW(x, mu = 0.5, sigma = 2, nu=1, lower.tail = FALSE),
from = 0, to = 3, ylim = c(0, 1), col = "red", las = 1, ylab = "R(x)")

## The quantile function
p <- seq(from = 0, to = 0.99999, length.out = 100)
plot(x = qGammaW(p = p, mu = 0.5, sigma = 2, nu=1), y = p,
xlab = "Quantile", las = 1, ylab = "Probability")
curve(pGammaW(x, mu = 0.5, sigma = 2, nu=1), from = 0, add = TRUE,
col = "red")

## The random function
hist(rGammaW(1000, mu = 0.5, sigma = 2, nu=1), freq = FALSE, xlab = "x",
ylim = c(0, 1), las = 1, main = "")
curve(dGammaW(x, mu = 0.5, sigma = 2, nu=1),  from = 0, add = TRUE,
col = "red", ylim = c(0, 1))

## The Hazard function
par(mfrow=c(1,1))
curve(hGammaW(x, mu = 0.5, sigma = 2, nu=1), from = 0, to = 2,
ylim = c(0, 1), col = "red", ylab = "Hazard function", las = 1)

par(old_par) # restore previous graphical parameters
```

---

dGGD *The Generalized Gompertz distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the generalized Gompertz distribution with parameters mu sigma and nu.

Usage

```
dGGD(x, mu, sigma, nu, log = FALSE)

pGGD(q, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

qGGD(p, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

rGGD(n, mu, sigma, nu)

hGGD(x, mu, sigma, nu)
```

Arguments

x, q	vector of quantiles.
mu, nu	scale parameter.

sigma	shape parameters.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
p	vector of probabilities.
n	number of observations.

Details

The Generalized Gompertz Distribution with parameters mu, sigma and nu has density given by
$f(x) = \nu\mu \exp(-\frac{\mu}{\sigma}(\exp(\sigma x - 1)))(1 - \exp(-\frac{\mu}{\sigma}(\exp(\sigma x - 1))))^{(\nu-1)}$,
for $x \geq 0$, $\mu > 0$, $\sigma \geq 0$ and $\nu > 0$.

Value

dGGD gives the density, pGGD gives the distribution function, qGGD gives the quantile function, rGGD generates random deviates and hGGD gives the hazard function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

El-Gohary A, Alshamrani A, Al-Otaibi AN (2013). "The generalized Gompertz distribution." *Applied Mathematical Modelling*, **37**(1-2), 13–24.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
par(mfrow = c(1, 1))
curve(dGGD(x, mu=1, sigma=0.3, nu=1.5), from = 0, to = 4,
      col = "red", las = 1, ylab = "f(x)")

## The cumulative distribution and the Reliability function
par(mfrow = c(1, 2))
curve(pGGD(x, mu=1, sigma=0.3, nu=1.5), from = 0, to = 4,
      ylim = c(0, 1), col = "red", las = 1, ylab = "F(x)")
curve(pGGD(x, mu=1, sigma=0.3, nu=1.5, lower.tail = FALSE),
      from = 0, to = 4, ylim = c(0, 1), col = "red", las = 1, ylab = "R(x)")

## The quantile function
p <- seq(from = 0, to = 0.99999, length.out = 100)
plot(x = qGGD(p=p, mu=1, sigma=0.3, nu=1.5), y = p,
     xlab = "Quantile", las = 1, ylab = "Probability")
curve(pGGD(x, mu=1, sigma=0.3, nu=1.5), from = 0, add = TRUE,
      col = "red")
```

```
## The random function
hist(rGGD(1000, mu=1, sigma=0.3, nu=1.5), freq = FALSE, xlab = "x",
     las = 1, ylim = c(0, 0.7), main = "")
curve(dGGD(x,mu=1, sigma=0.3, nu=1.5), from = 0, to =8, add = TRUE,
      col = "red")

## The Hazard function
par(mfrow=c(1,1))
curve(hGGD(x, mu=1, sigma=0.3, nu=1.5), from = 0, to = 3, col = "red",
      ylab = "The hazard function", las = 1)

par(old_par) # restore previous graphical parameters
```

---

dGIW *The Generalized Inverse Weibull distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the Generalized Inverse Weibull distribution with parameters mu, sigma and nu.

Usage

```
dGIW(x, mu, sigma, nu, log = FALSE)

pGIW(q, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

qGIW(p, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

rGIW(n, mu, sigma, nu)

hGIW(x, mu, sigma, nu)
```

Arguments

x, q	vector of quantiles.
mu	parameter.
sigma	parameter.
nu	parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
p	vector of probabilities.
n	number of observations.

Details

The Generalized Inverse Weibull distribution mu, sigma and nu has density given by

$f(x) = \nu\sigma\mu^{\sigma}x^{-(\sigma+1)}exp\{-\nu(\frac{\mu}{x})^{\sigma}\},$

for x > 0.

Value

dGIW gives the density, pGIW gives the distribution function, qGIW gives the quantile function, rGIW generates random deviates and hGIW gives the hazard function.

Author(s)

Amylkar Urrea Montoya, <amylkar.urrea@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Felipe R SdG, Edwin M MO, Gauss M C (2009). "The generalized inverse Weibull distribution." *Statistical papers*, **52**(3), 591–619. doi:10.1007/s0036200902713.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dGIW(x, mu=3, sigma=5, nu=0.5), from=0.001, to=8,
      col="red", ylab="f(x)", las=1)

## The cumulative distribution and the Reliability function
par(mfrow=c(1, 2))
curve(pGIW(x, mu=3, sigma=5, nu=0.5),
      from=0.0001, to=14, col="red", las=1, ylab="F(x)")
curve(pGIW(x, mu=3, sigma=5, nu=0.5, lower.tail=FALSE),
      from=0.0001, to=14, col="red", las=1, ylab="R(x)")

## The quantile function
p <- seq(from=0, to=0.99999, length.out=100)
plot(x=qGIW(p, mu=3, sigma=5, nu=0.5), y=p, xlab="Quantile",
     las=1, ylab="Probability")
curve(pGIW(x, mu=3, sigma=5, nu=0.5),
      from=0, add=TRUE, col="red")

## The random function
hist(rGIW(n=1000, mu=3, sigma=5, nu=0.5), freq=FALSE,
     xlab="x", ylim=c(0, 0.8), las=1, main="")
curve(dGIW(x, mu=3, sigma=5, nu=0.5),
      from=0.001, to=14, add=TRUE, col="red")

## The Hazard function
```

```
curve(hGIW(x, mu=3, sigma=5, nu=0.5), from=0.001, to=30,
      col="red", ylab="Hazard function", las=1)

par(old_par) # restore previous graphical parameters
```

---

dGMW *The Generalized modified Weibull distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the generalized modified weibull distribution with parameters mu, sigma, nu and tau.

Usage

```
dGMW(x, mu, sigma, nu, tau, log = FALSE)

pGMW(q, mu, sigma, nu, tau, lower.tail = TRUE, log.p = FALSE)

qGMW(p, mu, sigma, nu, tau, lower.tail = TRUE, log.p = FALSE)

rGMW(n, mu, sigma, nu, tau)

hGMW(x, mu, sigma, nu, tau, log = FALSE)
```

Arguments

x, q	vector of quantiles.
mu	scale parameter.
sigma	shape parameter.
nu	shape parameter.
tau	acceleration parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
p	vector of probabilities.
n	number of observations.

Details

The generalized modified weibull with parameters mu, sigma, nu and tau has density given by
$f(x) = \mu\sigma x^{\nu-1}(\nu + \tau x)\exp(\tau x - \mu x^{\nu} e^{\tau x})[1 - \exp(-\mu x^{\nu} e^{\tau x})]^{\sigma-1}$,
for x>0.

Value

dGMW gives the density, pGMW gives the distribution function, qGMW gives the quantile function, rGMW generates random deviates and hGMW gives the hazard function.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dGMW(x, mu=2, sigma=0.5, nu=2, tau=1.5), from=0, to=0.8,
      ylim=c(0, 2), col="red", las=1, ylab="f(x)")

## The cumulative distribution and the Reliability function
par(mfrow=c(1, 2))
curve(pGMW(x, mu=2, sigma=0.5, nu=2, tau=1.5),
      from=0, to=1.2, col="red", las=1, ylab="F(x)")
curve(pGMW(x, mu=2, sigma=0.5, nu=2, tau=1.5, lower.tail=FALSE),
      from=0, to=1.2, col="red", las=1, ylab="R(x)")

## The quantile function
p <- seq(from=0, to=0.99999, length.out=100)
plot(x=qGMW(p, mu=2, sigma=0.5, nu=2, tau=1.5), y=p, xlab="Quantile",
    las=1, ylab="Probability")
curve(pGMW(x, mu=2, sigma=0.5, nu=2, tau=1.5), from=0, add=TRUE, col="red")

## The random function
hist(rGMW(n=1000, mu=2, sigma=0.5, nu=2,tau=1.5), freq=FALSE,
    xlab="x", main ="", las=1)
curve(dGMW(x, mu=2, sigma=0.5, nu=2, tau=1.5),  from=0, add=TRUE, col="red")

## The Hazard function
par(mfrow=c(1,1))
curve(hGMW(x, mu=2, sigma=0.5, nu=2, tau=1.5), from=0, to=1, ylim=c(0, 16),
      col="red", ylab="Hazard function", las=1)

par(old_par) # restore previous graphical parameters
```

---

dIW *The Inverse Weibull distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the inverse weibull distribution with parameters mu and sigma.

Usage

```
dIW(x, mu, sigma, log = FALSE)
```

```
pIW(q, mu, sigma, lower.tail = TRUE, log.p = FALSE)

qIW(p, mu, sigma, lower.tail = TRUE, log.p = FALSE)

rIW(n, mu, sigma)

hIW(x, mu, sigma)
```

Arguments

x, q	vector of quantiles.
mu	scale parameter.
sigma	shape parameters.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
p	vector of probabilities.
n	number of observations.

Details

The inverse weibull distribution with parameters mu and sigma has density given by

$f(x) = \mu\sigma x^{-\sigma-1} \exp(\mu x^{-\sigma})$

for $x > 0$, $\mu > 0$ and $\sigma > 0$

Value

dIW gives the density, pIW gives the distribution function, qIW gives the quantile function, rIW generates random deviates and hIW gives the hazard function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Drapella A (1993). "The complementary Weibull distribution: unknown or just forgotten?" *Quality and Reliability Engineering International*, **9**(4), 383–385.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dIW(x, mu=5, sigma=2.5), from=0, to=10,
      ylim=c(0, 0.55), col="red", las=1, ylab="f(x)")
```

```
#'
## The cumulative distribution and the Reliability function
par(mfrow=c(1, 2))
curve(pIW(x, mu=5, sigma=2.5),
      from=0, to=10,  col="red", las=1, ylab="F(x)")
curve(pIW(x, mu=5, sigma=2.5, lower.tail=FALSE),
      from=0, to=10, col="red", las=1, ylab="R(x)")

## The quantile function
p <- seq(from=0, to=0.99999, length.out=100)
plot(x=qIW(p, mu=5, sigma=2.5), y=p, xlab="Quantile",
  las=1, ylab="Probability")
curve(pIW(x, mu=5, sigma=2.5), from=0, add=TRUE, col="red")

## The random function
hist(rIW(n=10000, mu=5, sigma=2.5), freq=FALSE, xlim=c(0,60),
  xlab="x", las=1, main="")
curve(dIW(x, mu=5, sigma=2.5), from=0, add=TRUE, col="red")

## The Hazard function
par(mfrow=c(1,1))
curve(hIW(x, mu=5, sigma=2.5), from=0, to=15, ylim=c(0, 0.9),
   col="red", ylab="Hazard function", las=1)

par(old_par) # restore previous graphical parameters
```

---

dKumIW *The Kumaraswamy Inverse Weibull distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the Kumaraswamy Inverse Weibull distribution with parameters `mu`, `sigma` and `nu`.

Usage

```
dKumIW(x, mu, sigma, nu, log = FALSE)

pKumIW(q, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

qKumIW(p, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

rKumIW(n, mu, sigma, nu)

hKumIW(x, mu, sigma, nu)
```

Arguments

`x, q`	vector of quantiles.

mu	parameter.
sigma	parameter.
nu	parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
p	vector of probabilities.
n	number of observations.

Details

The Kumaraswamy Inverse Weibull Distribution with parameters mu, sigma and nu has density given by

$f(x) = \mu\sigma\nu x^{-\mu-1} \exp{-\sigma x^{-\mu}} (1 - \exp{-\sigma x^{-\mu}})^{\nu-1},$

for $x > 0$, $\mu > 0$, $\sigma > 0$ and $\nu > 0$.

Value

dKumIW gives the density, pKumIW gives the distribution function, qKumIW gives the quantile function, rKumIW generates random deviates and hKumIW gives the hazard function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Shahbaz MQ, Shahbaz S, Butt NS (2012). "The Kumaraswamy-Inverse Weibull Distribution." *Shahbaz, MQ, Shahbaz, S., & Butt, NS (2012). The Kumaraswamy–Inverse Weibull Distribution. Pakistan journal of statistics and operation research*, **8**(3), 479–489.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
par(mfrow = c(1, 1))
curve(dKumIW(x, mu = 1.5, sigma=  1.5, nu = 1), from = 0, to = 8.5,
      col = "red", las = 1, ylab = "f(x)")

## The cumulative distribution and the Reliability function
par(mfrow = c(1, 2))
curve(pKumIW(x, mu = 1.5, sigma=  1.5, nu = 1), from = 0, to = 8.5,
      ylim = c(0, 1), col = "red", las = 1, ylab = "F(x)")
curve(pKumIW(x, mu = 1.5, sigma=  1.5, nu = 1, lower.tail = FALSE),
      from = 0, to = 6, ylim = c(0, 1), col = "red", las = 1, ylab = "R(x)")
```

```
## The quantile function
p <- seq(from = 0, to = 0.99999, length.out = 100)
plot(x = qKumIW(p=p, mu = 1.5, sigma=  1.5, nu = 10), y = p,
     xlab = "Quantile", las = 1, ylab = "Probability")
curve(pKumIW(x, mu = 1.5, sigma=  1.5, nu = 10), from = 0, add = TRUE,
      col = "red")

## The random function
hist(rKumIW(1000, mu = 1.5, sigma=  1.5, nu = 5), freq = FALSE, xlab = "x",
     las = 1, ylim = c(0, 1.5), main = "")
curve(dKumIW(x, mu = 1.5, sigma=  1.5, nu = 5), from = 0, to =8, add = TRUE,
      col = "red")

## The Hazard function
par(mfrow=c(1,1))
curve(hKumIW(x, mu = 1.5, sigma=  1.5, nu = 1), from = 0, to = 3,
      ylim = c(0, 0.7), col = "red", ylab = "Hazard function", las = 1)

par(old_par) # restore previous graphical parameters
```

`dLIN` *Lindley distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the Lindley distribution with parameter `mu`.

Usage

```
dLIN(x, mu, log = FALSE)

pLIN(q, mu, lower.tail = TRUE, log.p = FALSE)

qLIN(p, mu, lower.tail = TRUE, log.p = FALSE)

rLIN(n, mu)

hLIN(x, mu, log = FALSE)
```

Arguments

`x, q`	vector of quantiles.
`mu`	parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
`p`	vector of probabilities.
`n`	number of observations.

Details

Lindley Distribution with parameter `mu` has density given by

$f(x) = \frac{\mu^2}{\mu+1}(1+x)\exp(-\mu x),$

for x > 0 and $\mu > 0$. These function were taken form LindleyR package.

Value

`dLIN` gives the density, `pLIN` gives the distribution function, `qLIN` gives the quantile function, `rLIN` generates random deviates and `hLIN` gives the hazard function.

Author(s)

Freddy Hernandez, <fhernanb@unal.edu.co>

References

Lindley DV (1958). "Fiducial distributions and Bayes' theorem." *Journal of the Royal Statistical Society. Series B (Methodological)*, 102–107.

Lindley DV (1965). *Introduction to probability and statistics: from a Bayesian viewpoint. 2. Inference*. CUP Archive.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dLIN(x, mu=1.5), from=0.0001, to=10,
      col="red", las=1, ylab="f(x)")

## The cumulative distribution and the Reliability function
par(mfrow=c(1, 2))
curve(pLIN(x, mu=2), from=0.0001, to=10, col="red", las=1, ylab="F(x)")
curve(pLIN(x, mu=2, lower.tail=FALSE), from=0.0001,
      to=10, col="red", las=1, ylab="R(x)")

## The quantile function
p <- seq(from=0, to=0.99999, length.out=100)
plot(x=qLIN(p, mu=2), y=p, xlab="Quantile", las=1, ylab="Probability")
curve(pLIN(x, mu=2), from=0, add=TRUE, col="red")

## The random function
hist(rLIN(n=10000, mu=2), freq=FALSE, xlab="x", las=1, main="")
curve(dLIN(x, mu=2), from=0.09, to=5, add=TRUE, col="red")

## The Hazard function
curve(hLIN(x, mu=2), from=0.001, to=10, col="red", ylab="Hazard function", las=1)

par(old_par) # restore previous graphical parameters
```

dLW *The Log-Weibull distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the Log-Weibull distribution with parameters `mu` and `sigma`.

Usage

```
dLW(x, mu, sigma, log = FALSE)

pLW(q, mu, sigma, lower.tail = TRUE, log.p = FALSE)

qLW(p, mu, sigma, lower.tail = TRUE, log.p = FALSE)

rLW(n, mu, sigma)

hLW(x, mu, sigma)
```

Arguments

`x, q`	vector of quantiles.
`mu`	parameter.
`sigma`	parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
`p`	vector of probabilities.
`n`	number of observations.

Details

The Log-Weibull Distribution with parameters `mu` and `sigma` has density given by
$f(y) = (1/\sigma)e^{((y-\mu)/\sigma)} exp\{-e^{((y-\mu)/\sigma)}\},$
for - `infty` < y < `infty`.

Value

`dLW` gives the density, `pLW` gives the distribution function, `qLW` gives the quantile function, `rLW` generates random deviates and `hLW` gives the hazard function.

Author(s)

Amylkar Urrea Montoya, <amylkar.urrea@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

E.J G (1958). *Statistics of extremes*. Columbia University Press. ISBN 10:0231021909.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dLW(x, mu=0, sigma=1.5), from=-8, to=5,
      col="red", las=1, ylab="f(x)")

## The cumulative distribution and the Reliability function
par(mfrow=c(1, 2))
curve(pLW(x, mu=0, sigma=1.5),
      from=-8, to=5,  col="red", las=1, ylab="F(x)")
curve(pLW(x, mu=0, sigma=1.5, lower.tail=FALSE),
      from=-8, to=5, col="red", las=1, ylab="R(x)")

## The quantile function
p <- seq(from=0, to=0.99999, length.out=100)
plot(x=qLW(p, mu=0, sigma=1.5), y=p, xlab="Quantile",
     las=1, ylab="Probability")
curve(pLW(x, mu=0, sigma=1.5), from=-8, to=5, add=TRUE, col="red")

## The random function
hist(rLW(n=10000, mu=0, sigma=1.5), freq=FALSE,
     xlab="x", las=1, main="")
curve(dLW(x, mu=0, sigma=1.5), from=-15, to=6, add=TRUE, col="red")

## The Hazard function
par(mfrow=c(1,1))
curve(hLW(x, mu=0, sigma=1.5), from=-8, to=7,
      col="red", ylab="Hazard function", las=1)

par(old_par) # restore previous graphical parameters
```

`dMOEIW` *The Marshall-Olkin Extended Inverse Weibull distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the Marshall-Olkin Extended Inverse Weibull distribution with parameters `mu`, `sigma` and `nu`.

Usage

```
dMOEIW(x, mu, sigma, nu, log = FALSE)

pMOEIW(q, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

qMOEIW(p, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

rMOEIW(n, mu, sigma, nu)

hMOEIW(x, mu, sigma, nu)
```

Arguments

x, q	vector of quantiles.
mu	parameter.
sigma	parameter.
nu	parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
p	vector of probabilities.
n	number of observations.

Details

The Marshall-Olkin Extended Inverse Weibull distribution mu, sigma and nu has density given by

$f(x) = \frac{\mu\sigma\nu x^{-(\sigma+1)} exp\{-\mu x^{-\sigma}\}}{\{\nu-(\nu-1)exp\{-\mu x^{-\sigma}\}\}^2},$

for x > 0.

Value

dMOEIW gives the density, pMOEIW gives the distribution function, qMOEIW gives the quantile function, rMOEIW generates random deviates and hMOEIW gives the hazard function.

Author(s)

Amylkar Urrea Montoya, <amylkar.urrea@udea.edu.co>

References

Hassan M O, A.H E, A.M.K T, Abdulkareem M Bc (2017). "Extended inverse Weibull distribution with reliability application." *Journal of the Egyptian Mathematical Society*, **25**, 343–349. doi:10.1016/j.joems.2017.02.006, http://dx.doi.org/10.1016/j.joems.2017.02.006.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dMOEIW(x, mu=0.6, sigma=1.7, nu=0.3), from=0, to=2,
      col="red", ylab="f(x)", las=1)

## The cumulative distribution and the Reliability function
par(mfrow=c(1, 2))
curve(pMOEIW(x, mu=0.6, sigma=1.7, nu=0.3),
      from=0.0001, to=2, col="red", las=1, ylab="F(x)")
curve(pMOEIW(x, mu=0.6, sigma=1.7, nu=0.3, lower.tail=FALSE),
      from=0.0001, to=2, col="red", las=1, ylab="R(x)")

## The quantile function
p <- seq(from=0, to=0.99999, length.out=100)
plot(x=qMOEIW(p, mu=0.6, sigma=1.7, nu=0.3), y=p, xlab="Quantile",
     las=1, ylab="Probability")
curve(pMOEIW(x, mu=0.6, sigma=1.7, nu=0.3),
      from=0, add=TRUE, col="red")

## The random function
hist(rMOEIW(n=1000, mu=0.6, sigma=1.7, nu=0.3), freq=FALSE,
     xlab="x", las=1, main="")
curve(dMOEIW(x, mu=0.6, sigma=1.7, nu=0.3),
      from=0.001, to=4, add=TRUE, col="red")

## The Hazard function
curve(hMOEIW(x, mu=0.5, sigma=0.7, nu=1), from=0.001, to=3,
      col="red", ylab="Hazard function", las=1)

par(old_par) # restore previous graphical parameters
```

---

dMOEW *The Marshall-Olkin Extended Weibull distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the Marshall-Olkin Extended Weibull distribution with parameters mu, sigma and nu.

Usage

```
dMOEW(x, mu, sigma, nu, log = FALSE)

pMOEW(q, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

qMOEW(p, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)
```

```
rMOEW(n, mu, sigma, nu)

hMOEW(x, mu, sigma, nu)
```

Arguments

x, q	vector of quantiles.
mu	parameter.
sigma	parameter.
nu	parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
p	vector of probabilities.
n	number of observations.

Details

The Marshall-Olkin Extended Weibull distribution mu, sigma and nu has density given by

$f(x) = \frac{\mu \sigma \nu (\nu x)^{\sigma-1} exp\{-(\nu x)^{\sigma}\}}{\{1-(1-\mu) exp\{-(\nu x)^{\sigma}\}\}^2},$

for x > 0.

Value

dMOEW gives the density, pMOEW gives the distribution function, qMOEW gives the quantile function, rMOEW generates random deviates and hMOEW gives the hazard function.

Author(s)

Amylkar Urrea Montoya, <amylkar.urrea@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

M.E G, E.K A, R.A J (2005). "Marshall–Olkin extended weibull distribution and its application to censored data." *Journal of Applied Statistics*, **32**(10), 1025–1034. doi:10.1080/02664760500165008.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dMOEW(x, mu=0.5, sigma=0.7, nu=1), from=0.001, to=1,
      col="red", ylab="f(x)", las=1)

## The cumulative distribution and the Reliability function
par(mfrow=c(1, 2))
```

```
curve(pMOEW(x, mu=0.5, sigma=0.7, nu=1),
      from=0.0001, to=2, col="red", las=1, ylab="F(x)")
curve(pMOEW(x, mu=0.5, sigma=0.7, nu=1, lower.tail=FALSE),
      from=0.0001, to=2, col="red", las=1, ylab="R(x)")

## The quantile function
p <- seq(from=0, to=0.99999, length.out=100)
plot(x=qMOEW(p, mu=0.5, sigma=0.7, nu=1), y=p, xlab="Quantile",
     las=1, ylab="Probability")
curve(pMOEW(x, mu=0.5, sigma=0.7, nu=1),
      from=0, add=TRUE, col="red")

## The random function
hist(rMOEW(n=100, mu=0.5, sigma=0.7, nu=1), freq=FALSE,
     xlab="x", ylim=c(0, 1), las=1, main="")
curve(dMOEW(x, mu=0.5, sigma=0.7, nu=1),
      from=0.001, to=2, add=TRUE, col="red")

## The Hazard function
curve(hMOEW(x, mu=0.5, sigma=0.7, nu=1), from=0.001, to=3,
      col="red", ylab="Hazard function", las=1)

par(old_par) # restore previous graphical parameters
```

---

dMOK *The Marshall-Olkin Kappa distribution*

Description

Desnsity, distribution function, quantile function, random generation and hazard function for the Marshall-Olkin Kappa distribution with parameters mu, sigma, nu and tau.

Usage

```
dMOK(x, mu, sigma, nu, tau, log = FALSE)

pMOK(q, mu, sigma, nu, tau, lower.tail = TRUE, log.p = FALSE)

qMOK(p, mu, sigma, nu, tau, lower.tail = TRUE, log.p = FALSE)

rMOK(n, mu, sigma, nu, tau)

hMOK(x, mu, sigma, nu, tau)
```

Arguments

x, q	vector of quantiles.
mu	parameter.

`sigma`	parameter.
`nu`	parameter.
`tau`	parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
`p`	vector of probabilities.
`n`	number of observations.

Details

The Marshall-Olkin Kappa distribution with parameters `mu`, `sigma`, `nu` and `tau` has density given by:

$$f(x) = \frac{\tau \frac{\mu\nu}{\sigma} \left(\frac{x}{\sigma}\right)^{\nu-1} \left(\mu + \left(\frac{x}{\sigma}\right)^{\mu\nu}\right)^{-\frac{\mu+1}{\mu}}}{\left[\tau + (1-\tau) \left(\frac{\left(\frac{x}{\sigma}\right)^{\mu\nu}}{\mu + \left(\frac{x}{\sigma}\right)^{\mu\nu}}\right)^{\frac{1}{\mu}}\right]^2}$$

for x > 0.

Value

`dMOK` gives the density, `pMOK` gives the distribution function, `qMOK` gives the quantile function, `rMOK` generates random deviates and `hMOK` gives the hazard function.

Author(s)

Angel Muñoz,

References

Javed M, Nawaz T, Irfan M (2018). "The Marshall-Olkin kappa distribution: properties and applications." *Journal of King Saud University-Science*.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
par(mfrow = c(1,1))
curve(dMOK(x = x, mu = 1, sigma = 3.5, nu = 3, tau = 2), from = 0, to = 15,
      ylab = 'f(x)', col = 2, las = 1)

## The cumulative distribution and the Reliability function

par(mfrow = c(1,2))
curve(pMOK(q = x, mu = 1, sigma = 2.5, nu = 3, tau = 2), from = 0, to = 10,
     col = 2, lwd = 2, las = 1, ylab = 'F(x)')
curve(pMOK(q = x, mu = 1, sigma = 2.5, nu = 3, tau = 2, lower.tail = FALSE), from = 0, to = 10,
      col = 2, lwd = 2, las = 1, ylab = 'R(x)')
```

```
## The quantile function
p <- seq(from = 0.00001, to = 0.99999, length.out = 100)
plot(x = qMOK(p = p, mu = 4, sigma = 2.5, nu = 3, tau = 2), y = p, xlab = 'Quantile',
     las = 1, ylab = 'Probability')
curve(pMOK(q = x, mu = 4, sigma = 2.5, nu = 3, tau = 2), from = 0, to = 15,
      add = TRUE, col = 2)

## The random function

hist(rMOK(n = 10000, mu = 1, sigma = 2.5, nu = 3, tau = 2), freq = FALSE,
     xlab = "x", las = 1, main = '', ylim = c(0,.3), xlim = c(0,20), breaks = 50)
curve(dMOK(x, mu = 1, sigma = 2.5, nu = 3, tau = 2), from = 0, to = 15, add = TRUE, col = 2)

## The Hazard function

par(mfrow = c(1,1))
curve(hMOK(x = x, mu = 1, sigma = 2.5, nu = 3, tau = 2), from = 0, to = 20,
      col = 2, ylab = 'Hazard function', las = 1)

par(old_par) # restore previous graphical parameters
```

---

dMW *The Modified Weibull distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the modified weibull distribution with parameters `mu`, `sigma` and `nu`.

Usage

```
dMW(x, mu, sigma, nu, log = FALSE)

pMW(q, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

qMW(p, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

rMW(n, mu, sigma, nu)

hMW(x, mu, sigma, nu)
```

Arguments

`x, q`	vector of quantiles.
`mu`	shape parameter one.
`sigma`	parameter two.
`nu`	scale parameter three.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).

`lower.tail`	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
`p`	vector of probabilities.
`n`	number of observations.

Details

The modified weibull distribution with parameters `mu`, `sigma` and `nu` has density given by

$f(x) = \mu(\sigma + \nu x)x^{\sigma-1}\exp(\nu x)\exp(-\mu x^{\sigma}\exp(\nu x))$

for $x > 0$, $\mu > 0$, $\sigma \geq 0$ and $\nu \geq 0$.

Value

`dMW` gives the density, `pMW` gives the distribution function, `qMW` gives the quantile function, `rMW` generates random deviates and `hMW` gives the hazard function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Lai CD, Xie M, Murthy DNP (2003). "A modified Weibull distribution." *IEEE Transactions on reliability*, **52**(1), 33–37.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dMW(x, mu=2, sigma=1.5, nu=0.2), from=0, to=2,
  ylim=c(0, 1.5), col="red", las=1, ylab="f(x)")

## The cumulative distribution and the Reliability function
par(mfrow = c(1, 2))
curve(pMW(x, mu=2, sigma=1.5, nu=0.2), from=0, to=2,
 col = "red", las=1, ylab="F(x)")
curve(pMW(x, mu=2, sigma=1.5, nu=0.2, lower.tail = FALSE),
from=0, to=2, col="red", las=1, ylab ="R(x)")

## The quantile function
p <- seq(from=0, to=0.9999, length.out=100)
plot(x=qMW(p, mu=2, sigma=1.5, nu=0.2), y=p, xlab="Quantile",
 las=1, ylab="Probability")
curve(pMW(x, mu=2, sigma=1.5, nu=0.2), from=0, add=TRUE, col="red")

## The random function
hist(rMW(n=1000, mu=2, sigma=1.5, nu=0.2), freq=FALSE,
 xlab="x", las=1, main="")
```

```
curve(dMW(x, mu=2, sigma=1.5, nu=0.2), from=0, add=TRUE, col="red")

## The Hazard function
par(mfrow=c(1,1))
curve(hMW(x, mu=2, sigma=1.5, nu=0.2), from=0, to=1.5, ylim=c(0, 5),
 col="red", las=1, ylab="H(x)", las=1)

par(old_par) # restore previous graphical parameters
```

---

dOW *The Odd Weibull Distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the Odd Weibull distribution with parameters mu, sigma and nu.

Usage

```
dOW(x, mu, sigma, nu, log = FALSE)

pOW(q, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

qOW(p, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

rOW(n, mu, sigma, nu)

hOW(x, mu, sigma, nu)
```

Arguments

x, q	vector of quantiles.
mu	parameter one.
sigma	parameter two.
nu	parameter three.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are P[T <= t], otherwise, P[T > t].
p	vector of probabilities.
n	number of observations.

Details

The Odd Weibull with parameters mu, sigma and nu has density given by

$$f(x) = \left(\frac{\sigma\nu}{x}\right)(\mu x)^{\sigma} e^{(\mu x)^{\sigma}} \left(e^{(\mu x)^{\sigma}} - 1\right)^{\nu-1} \left[1 + \left(e^{(\mu x)^{\sigma}} - 1\right)^{\nu}\right]^{-2}$$

for x > 0.

Value

dOW gives the density, pOW gives the distribution function, qOW gives the quantile function, rOW generates random deviates and hOW gives the hazard function.

Author(s)

Jaime Mosquera Gutiérrez <jmosquerag@unal.edu.co>

References

Cooray K (2006). "Generalization of the Weibull distribution: The odd Weibull family." *Statistical Modelling*, **6**(3), 265–277. ISSN 1471082X, doi:10.1191/1471082X06st116oa.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dOW(x, mu=2, sigma=3, nu=0.2), from=0, to=4, ylim=c(0, 2),
      col="red", las=1, ylab="f(x)")

## The cumulative distribution and the Reliability function
par(mfrow = c(1, 2))
curve(pOW(x, mu=2, sigma=3, nu=0.2), from=0, to=4, ylim=c(0, 1),
      col="red", las=1, ylab="f(x)")
curve(pOW(x, mu=2, sigma=3, nu=0.2, lower.tail=FALSE), from=0,
      to=4,  ylim=c(0, 1), col="red", las=1,
      ylab = "R(x)")

## The quantile function
p <- seq(from=0, to=0.998, length.out=100)
plot(x = qOW(p, mu=2, sigma=3, nu=0.2), y=p, xlab="Quantile", las=1,
     ylab="Probability")
curve(pOW(x, mu=2, sigma=3, nu=0.2), from=0, add=TRUE, col="red")

## The random function
hist(rOW(n=10000, mu=2, sigma = 3, nu = 0.2), freq=FALSE, ylim = c(0, 2),
     xlab = "x", las = 1, main = "")
curve(dOW(x, mu=2, sigma=3, nu=0.2),  from=0, ylim=c(0, 2), add=TRUE,
      col = "red")

## The Hazard function
par(mfrow=c(1,1))
curve(hOW(x, mu=2, sigma=3, nu=0.2), from=0, to=2.5, ylim=c(0, 3),
      col="red", ylab="Hazard function", las=1)

par(old_par) # restore previous graphical parameters
```

dPL *The Power Lindley distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the Power Lindley distribution with parameters `mu` and `sigma`.

Usage

```
dPL(x, mu, sigma, log = FALSE)

pPL(q, mu, sigma, lower.tail = TRUE, log.p = FALSE)

qPL(p, mu, sigma, lower.tail = TRUE, log.p = FALSE)

rPL(n, mu, sigma)

hPL(x, mu, sigma)
```

Arguments

`x, q`	vector of quantiles.
`mu`	parameter.
`sigma`	parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
`p`	vector of probabilities.
`n`	number of observations.

Details

The Power Lindley Distribution with parameters `mu` and `sigma` has density given by

$f(x) = \frac{\mu\sigma^2}{\sigma+1}(1+x^\mu)x^{\mu-1}\exp(-\sigma x^\mu),$

for x > 0.

Value

`dPL` gives the density, `pPL` gives the distribution function, `qPL` gives the quantile function, `rPL` generates random deviates and `hPL` gives the hazard function.

Author(s)

Amylkar Urrea Montoya, <amylkar.urrea@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Ghitanya ME, Al-Mutairi DK, Balakrishnanb N, Al-Enezi LJ (2013). "Power Lindley distribution and associated inference." *Computational Statistics and Data Analysis*, **64**, 20–33. doi:10.1016/j.csda.2013.02.026.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dPL(x, mu=1.5, sigma=0.2), from=0.1, to=10,
      col="red", las=1, ylab="f(x)")

## The cumulative distribution and the Reliability function
par(mfrow=c(1, 2))
curve(pPL(x, mu=1.5, sigma=0.2),
      from=0.1, to=10, col="red", las=1, ylab="F(x)")
curve(pPL(x, mu=1.5, sigma=0.2, lower.tail=FALSE),
      from=0.1, to=10, col="red", las=1, ylab="R(x)")

## The quantile function
p <- seq(from=0, to=0.99999, length.out=100)
plot(x=qPL(p, mu=1.5, sigma=0.2), y=p, xlab="Quantile",
     las=1, ylab="Probability")
curve(pPL(x, mu=1.5, sigma=0.2), from=0.1, add=TRUE, col="red")

## The random function
hist(rPL(n=1000, mu=1.5, sigma=0.2), freq=FALSE,
     xlab="x", las=1, main="")
curve(dPL(x, mu=1.5, sigma=0.2), from=0.1, to=15, add=TRUE, col="red")

## The Hazard function
par(mfrow=c(1,1))
curve(hPL(x, mu=1.5, sigma=0.2), from=0.1, to=15,
      col="red", ylab="Hazard function", las=1)

par(old_par) # restore previous graphical parameters
```

---

dQXGP *The Quasi XGamma Poisson distribution*

Description

Density, distribution function,quantile function, random generation and hazard function for the Quasi XGamma Poisson distribution with parameters `mu`, `sigma` and `nu`.

Usage

```
dQXGP(x, mu, sigma, nu, log = FALSE)

pQXGP(q, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

qQXGP(p, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

rQXGP(n, mu, sigma, nu)

hQXGP(x, mu, sigma, nu)
```

Arguments

`x, q`	vector of quantiles.
`mu`	parameter.
`sigma`	parameter.
`nu`	parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
`p`	vector of probabilities.
`n`	number of observations.

Details

The Quasi XGamma Poisson distribution with parameters `mu`, `sigma` and `nu` has density given by:

$f(x) = K(\mu,\sigma,\nu)(\frac{\sigma^2 x^2}{2} + \mu)exp(\frac{\nu exp(-\sigma x)(1+\mu+\sigma x+\frac{\sigma^2 x^2}{2})}{1+\mu} - \sigma x),$

for $x > 0$, $\mu > 0$, $\sigma > 0$, $\nu > 1$.

where

$K(\mu,\sigma,\nu) = \frac{\nu\sigma}{(exp(\nu)-1)(1+\mu)}$

Value

`dQXGP` gives the density, `pQXGP` gives the distribution function, `qQXGP` gives the quantile function, `rQXGP` generates random deviates and `hQXGP` gives the hazard function.

Author(s)

Simon Zapata

References

Subhradev S, Mustafa C K, Haitham M Y (2018). "The Quasi XGamma-Poisson distribution: Properties and Application." *Istatistik: Journal of the Turkish Statistical Assocation*, **11**(3), 65–76. ISSN 1300-4077, https://dergipark.org.tr/en/pub/ijtsa/issue/42850/518206.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dQXGP(x, mu=0.5, sigma=1, nu=1), from=0.1, to=8,
      ylim=c(0, 0.6), col="red", las=1, ylab="f(x)")

## The cumulative distribution and the Reliability function
par(mfrow=c(1, 2))
curve(pQXGP(x, mu=0.5, sigma=1, nu=1),
      from=0.1, to=8,  col="red", las=1, ylab="F(x)")
curve(pQXGP(x,  mu=0.5, sigma=1, nu=1, lower.tail=FALSE),
      from=0.1, to=8, col="red", las=1, ylab="R(x)")

## The quantile function
p <- seq(from=0, to=0.99999, length.out=100)
plot(x=qQXGP(p, mu=0.5, sigma=1, nu=1), y=p, xlab="Quantile",
     las=1, ylab="Probability")
curve(pQXGP(x, mu=0.5, sigma=1, nu=1),
      from=0.1, add=TRUE, col="red")

## The random function
hist(rQXGP(n=1000, mu=0.5, sigma=1, nu=1), freq=FALSE,
     xlab="x", ylim=c(0, 0.4), las=1, main="", xlim=c(0, 15))
curve(dQXGP(x, mu=0.5, sigma=1, nu=1),
      from=0.001, to=500, add=TRUE, col="red")

## The Hazard function
curve(hQXGP(x, mu=0.5, sigma=1, nu=1), from=0.01, to=3,
      col="red", ylab="Hazard function", las=1)

par(old_par) # restore previous graphical parameters
```

---

dRW *The Reflected Weibull distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the Reflected Weibull Distribution with parameters mu and sigma.

Usage

```
dRW(x, mu, sigma, log = FALSE)

pRW(q, mu, sigma, lower.tail = TRUE, log.p = FALSE)

qRW(p, mu, sigma, lower.tail = TRUE, log.p = FALSE)
```

```
rRW(n, mu, sigma)

hRW(x, mu, sigma)
```

Arguments

`x, q`	vector of quantiles.
`mu`	parameter.
`sigma`	parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
`p`	vector of probabilities.
`n`	number of observations.

Details

The Reflected Weibull Distribution with parameters `mu` and `sigma` has density given by

$f(y) = \mu\sigma(-y)^{\sigma-1}e^{-\mu(-y)^{\sigma}},$

for y < 0.

Value

`dRW` gives the density, `pRW` gives the distribution function, `qRW` gives the quantile function, `rRW` generates random deviates and `hRW` gives the hazard function.

Author(s)

Amylkar Urrea Montoya, <amylkar.urrea@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Clifford Cohen A (1973). "The Reflected Weibull Distribution." *Technometrics*, **15**(4), 867–873. doi:10.2307/1267396.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dRW(x, mu=1, sigma=1), from=-5, to=-0.01,
      col="red", las=1, ylab="f(x)")

## The cumulative distribution and the Reliability function
par(mfrow=c(1, 2))
curve(pRW(x, mu=1, sigma=1),
      from=-5, to=-0.01, col="red", las=1, ylab="F(x)")
```

```
curve(pRW(x, mu=1, sigma=1, lower.tail=FALSE),
      from=-5, to=-0.01, col="red", las=1, ylab="R(x)")

## The quantile function
p <- seq(from=0, to=0.99999, length.out=100)
plot(x=qRW(p, mu=1, sigma=1), y=p, xlab="Quantile",
     las=1, ylab="Probability")
curve(pRW(x, mu=1, sigma=1), from=-5, add=TRUE, col="red")

## The random function
hist(rRW(n=10000, mu=1, sigma=1), freq=FALSE,
     xlab="x", las=1, main="")
curve(dRW(x, mu=1, sigma=1), from=-5, to=-0.01, add=TRUE, col="red")

## The Hazard function
par(mfrow=c(1,1))
curve(hRW(x, mu=1, sigma=1), from=-0.3, to=-0.01,
      col="red", ylab="Hazard function", las=1)

par(old_par) # restore previous graphical parameters
```

---

dSZMW *The Sarhan and Zaindin's Modified Weibull distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for Sarhan and Zaindins modified weibull distribution with parameters `mu`, `sigma` and `nu`.

Usage

```
dSZMW(x, mu, sigma, nu, log = FALSE)

pSZMW(q, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

qSZMW(p, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

rSZMW(n, mu, sigma, nu)

hSZMW(x, mu, sigma, nu)
```

Arguments

`x, q`	vector of quantiles.
`mu`	scale parameter.
`sigma`	shape parameter.
`nu`	shape parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).

lower.tail	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
p	vector of probabilities.
n	number of observations.

Details

The Sarhan and Zaindins modified weibull with parameters mu, sigma and nu has density given by

$f(x) = (\mu + \sigma\nu x^(\nu - 1)) \exp(-\mu x - \sigma x^{\nu})$

for $x > 0$, $\mu > 0$, $\sigma > 0$ and $\nu > 0$.

Value

dSZMW gives the density, pSZMW gives the distribution function, qSZMW gives the quantile function, rSZMW generates random deviates and hSZMW gives the hazard function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Sarhan AM, Zaindin M (2009). "Modified Weibull distribution." *APPS. Applied Sciences*, **11**, 123–136.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dSZMW(x, mu = 2, sigma = 1.5, nu = 0.2), from = 0, to = 2,
      ylim = c(0, 1.7), col = "red", las = 1, ylab = "f(x)")
## The cumulative distribution and the Reliability function
par(mfrow = c(1, 2))
curve(pSZMW(x, mu = 2, sigma = 1.5, nu = 0.2), from = 0, to = 2, ylim = c(0, 1),
      col = "red", las = 1, ylab = "F(x)")
curve(pSZMW(x, mu = 2, sigma = 1.5, nu = 0.2, lower.tail = FALSE), from = 0,
      to = 2, ylim = c(0, 1), col = "red", las = 1, ylab = "R(x)")

## The quantile function
p <- seq(from = 0, to = 0.99999, length.out = 100)
plot(x = qSZMW(p = p, mu = 2, sigma = 1.5, nu = 0.2), y = p, xlab = "Quantile",
     las = 1, ylab = "Probability")
curve(pSZMW(x, mu = 2, sigma = 1.5, nu = 0.2), from = 0, add = TRUE, col = "red")

## The random function
hist(rSZMW(n = 1000, mu = 2, sigma = 1.5, nu = 0.2), freq = FALSE, xlab = "x",
     las = 1, main = "")
curve(dSZMW(x, mu = 2, sigma = 1.5, nu = 0.2),  from = 0, add = TRUE, col = "red")
```

```
## The Hazard function
par(mfrow=c(1,1))
curve(hSZMW(x, mu = 2, sigma = 1.5, nu = 0.2), from = 0, to = 3, ylim = c(0, 8),
      col = "red", ylab = "Hazard function", las = 1)

par(old_par) # restore previous graphical parameters
```

---

dWG *The Weibull Geometric distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the weibull geometric distribution with parameters mu, sigma and nu.

Usage

```
dWG(x, mu, sigma, nu, log = FALSE)

pWG(q, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

qWG(p, sigma, mu, nu, lower.tail = TRUE, log.p = FALSE)

rWG(n, mu, sigma, nu)

hWG(x, mu, sigma, nu)
```

Arguments

x, q	vector of quantiles.
mu	scale parameter.
sigma	shape parameter.
nu	parameter of geometric random variable.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
p	vector of probabilities.
n	number of observations.

Details

The Weibull geometric distribution with parameters mu, sigma and nu has density given by

$f(x) = (\sigma\mu^\sigma(1-\nu)x^(\sigma-1)\exp(-(\mu x)^\sigma))(1-\nu\exp(-(\mu x)^\sigma))^{-2}$,

for $x > 0$, $\mu > 0$, $\sigma > 0$ and $0 < \nu < 1$.

Value

`dWG` gives the density, `pWG` gives the distribution function, `qWG` gives the quantile function, `rWG` generates random deviates and `hWG` gives the hazard function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Barreto-Souza W, de Morais AL, Cordeiro GM (2011). "The Weibull-geometric distribution." *Journal of Statistical Computation and Simulation*, **81**(5), 645–657.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dWG(x, mu = 0.9, sigma = 2, nu = 0.5), from = 0, to = 3,
ylim = c(0, 1.1), col = "red", las = 1, ylab = "f(x)")

## The cumulative distribution and the Reliability function
par(mfrow = c(1, 2))
curve(pWG(x, mu = 0.9, sigma = 2, nu = 0.5), from = 0, to = 3,
ylim = c(0, 1), col = "red", las = 1, ylab = "F(x)")
curve(pWG(x, mu = 0.9, sigma = 2, nu = 0.5, lower.tail = FALSE),
from = 0, to = 3, ylim = c(0, 1), col = "red", las = 1, ylab = "R(x)")

## The quantile function
p <- seq(from = 0, to = 0.99999, length.out = 100)
plot(x = qWG(p = p, mu = 0.9, sigma = 2, nu = 0.5), y = p,
xlab = "Quantile", las = 1, ylab = "Probability")
curve(pWG(x,mu = 0.9, sigma = 2, nu = 0.5), from = 0, add = TRUE,
col = "red")

## The random function
hist(rWG(1000, mu = 0.9, sigma = 2, nu = 0.5), freq = FALSE, xlab = "x",
ylim = c(0, 1.8), las = 1, main = "")
curve(dWG(x, mu = 0.9, sigma = 2, nu = 0.5),  from = 0, add = TRUE,
col = "red", ylim = c(0, 1.8))

## The Hazard function(
par(mfrow=c(1,1))
curve(hWG(x, mu = 0.9, sigma = 2, nu = 0.5), from = 0, to = 8,
ylim = c(0, 12), col = "red", ylab = "Hazard function", las = 1)

par(old_par) # restore previous graphical parameters
```

`dWGEE` *The Weighted Generalized Exponential-Exponential distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the Weighted Generalized Exponential-Exponential distribution with parameters `mu`, `sigma` and `nu`.

Usage

```
dWGEE(x, mu, sigma, nu, log = FALSE)

pWGEE(q, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

qWGEE(p, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

rWGEE(n, mu, sigma, nu)

hWGEE(x, mu, sigma, nu)
```

Arguments

`x, q`	vector of quantiles.
`mu`	parameter.
`sigma`	parameter.
`nu`	parameter.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
`p`	vector of probabilities.
`n`	number of observations.

Details

The Weighted Generalized Exponential-Exponential Distribution with parameters `mu`, `sigma` and `nu` has density given by

$f(x) = \sigma\nu \exp(-\nu x)(1 - \exp(-\nu x))^{\sigma-1}(1 - \exp(-\mu\nu x))/1 - \sigma B(\mu + 1, \sigma),$

for $x > 0$, $\mu > 0$, $\sigma > 0$ and $\nu > 0$.

Value

`dWGEE` gives the density, `pWGEE` gives the distribution function, `qWGEE` gives the quantile function, `rWGEE` generates random deviates and `hWGEE` gives the hazard function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Mahdavi A (2015). "Two Weighted Distributions Generated by Exponential Distribution." *Journal of Mathematical Extension*, **9**(1), 1–12.

Mahdavi A (2015). "Two weighted distributions generated by exponential distribution." *Journal of Mathematical Extension*, **9**, 1–12.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dWGEE(x, mu = 5, sigma = 0.5, nu = 1), from = 0, to = 6,
ylim = c(0, 1), col = "red", las = 1, ylab = "The probability density function")

## The cumulative distribution and the Reliability function
par(mfrow = c(1, 2))
curve(pWGEE(x, mu = 5, sigma = 0.5, nu = 1), from = 0, to = 6,
ylim = c(0, 1), col = "red", las = 1, ylab = "The cumulative distribution function")
curve(pWGEE(x, mu = 5, sigma = 0.5, nu = 1, lower.tail = FALSE),
from = 0, to = 6, ylim = c(0, 1), col = "red", las = 1, ylab = "The Reliability function")

## The quantile function
p <- seq(from = 0, to = 0.99999, length.out = 100)
plot(x = qWGEE(p = p, mu = 5, sigma = 0.5, nu = 1), y = p,
xlab = "Quantile", las = 1, ylab = "Probability")
curve(pWGEE(x, mu = 5, sigma = 0.5, nu = 1), from = 0, add = TRUE,
col = "red")

## The random function
hist(rWGEE(1000, mu = 5, sigma = 0.5, nu = 1), freq = FALSE, xlab = "x",
ylim = c(0, 1), las = 1, main = "")
curve(dWGEE(x, mu = 5, sigma = 0.5, nu = 1),  from = 0, add = TRUE,
col = "red", ylim = c(0, 1))

## The Hazard function(
par(mfrow=c(1,1))
curve(hWGEE(x, mu = 5, sigma = 0.5, nu = 1), from = 0, to = 6,
ylim = c(0, 1.4), col = "red", ylab = "The hazard function", las = 1)

par(old_par) # restore previous graphical parameters
```

---

dWP *The Weibull Poisson distribution*

Description

Density, distribution function, quantile function, random generation and hazard function for the Weibull Poisson distribution with parameters `mu`, `sigma` and `nu`.

Usage

```
dWP(x, mu, sigma, nu, log = FALSE)

pWP(q, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

qWP(p, mu, sigma, nu, lower.tail = TRUE, log.p = FALSE)

rWP(n, mu, sigma, nu)

hWP(x, mu, sigma, nu)
```

Arguments

x, q	vector of quantiles.
mu	parameter.
sigma	parameter.
nu	parameter.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x].
p	vector of probabilities.
n	number of observations.

Details

The Weibull Poisson distribution with parameters mu, sigma and nu has density given by

$f(x) = \frac{\mu\sigma\nu e^{-\nu}}{1-e^{-\nu}} x^{\mu-1} exp(-\sigma x^{\mu} + \nu exp(-\sigma x^{\mu})),$

for x > 0.

Value

dWP gives the density, pWP gives the distribution function, qWP gives the quantile function, rWP generates random deviates and hWP gives the hazard function.

Author(s)

Amylkar Urrea Montoya, <amylkar.urrea@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Wanbo L, Daimin S (1967). "A new compounding life distribution: the Weibull–Poisson distribution." *Journal of Applied Statistics*, **9**(1), 21–38. doi:10.1080/02664763.2011.575126, https://doi.org/10.1080/02664763.2011.575126.

Examples

```
old_par <- par(mfrow = c(1, 1)) # save previous graphical parameters

## The probability density function
curve(dWP(x, mu=1.5, sigma=0.5, nu=10), from=0.0001, to=2,
      col="red", las=1, ylab="f(x)")

## The cumulative distribution and the Reliability function
par(mfrow=c(1, 2))
curve(pWP(x, mu=1.5, sigma=0.5, nu=10),
      from=0.0001, to=2, col="red", las=1, ylab="F(x)")
curve(pWP(x, mu=1.5, sigma=0.5, nu=10, lower.tail=FALSE),
      from=0.0001, to=2, col="red", las=1, ylab="R(x)")

## The quantile function
p <- seq(from=0, to=0.99999, length.out=100)
plot(x=qWP(p, mu=1.5, sigma=0.5, nu=10), y=p, xlab="Quantile",
     las=1, ylab="Probability")
curve(pWP(x, mu=1.5, sigma=0.5, nu=10),
      from=0, add=TRUE, col="red")

## The random function
hist(rWP(n=10000, mu=1.5, sigma=0.5, nu=10), freq=FALSE,
     xlab="x", ylim=c(0, 2.2), las=1, main="")
curve(dWP(x, mu=1.5, sigma=0.5, nu=10),
      from=0.001, to=4, add=TRUE, col="red")

## The Hazard function
curve(hWP(x, mu=1.5, sigma=0.5, nu=10), from=0.001, to=5,
      col="red", ylab="Hazard function", las=1)

par(old_par) # restore previous graphical parameters
```

EEG *The Extended Exponential Geometric family*

Description

The Extended Exponential Geometric family

Usage

```
EEG(mu.link = "log", sigma.link = "log")
```

Arguments

mu.link	defines the mu.link, with "log" link as the default for the mu parameter.
sigma.link	defines the sigma.link, with "log" link as the default for the sigma.

Details

The Extended Exponential Geometric distribution with parameters `mu` and `sigma` has density given by

$f(x) = \mu\sigma \exp(-\mu x)(1 - (1 - \sigma) \exp(-\mu x))^{-2},$

for $x > 0$, $\mu > 0$ and $\sigma > 0$.

Value

Returns a gamlss.family object which can be used to fit a EEG distribution in the `gamlss()` function.

Author(s)

Johan David Marin Benjumea, `<johand.marin@udea.edu.co>`

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Adamidis K, Dimitrakopoulou T, Loukas S (2005). "On an extension of the exponential-geometric distribution." *Statistics & probability letters*, **73**(3), 259–269.

See Also

dEEG

Examples

```
# Generating some random values with
# known mu, sigma, nu and tau
y <- rEEG(n=100, mu = 1, sigma =1.5)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, family=EEG,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma, nu and tau
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))

# Example 2
# Generating random values under some model
n <- 200
x1 <- runif(n, min=0.1, max=0.2)
x2 <- runif(n, min=0.1, max=0.15)
mu <- exp(0.75 - x1)
sigma <- exp(0.5 - x2)
x <- rEEG(n=n, mu, sigma)
```

```
mod <- gamlss(x~x1, sigma.fo=~x2, family=EEG,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
```

EGG *The four parameter Exponentiated Generalized Gamma family*

Description

The four parameter Exponentiated Generalized Gamma distribution

Usage

```
EGG(mu.link = "log", sigma.link = "log", nu.link = "log", tau.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.
`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.
`nu.link`	defines the nu.link, with "log" link as the default for the nu parameter.
`tau.link`	defines the tau.link, with "log" link as the default for the tau parameter.

Details

Four parameter Exponentiated Generalized Gamma distribution with parameters `mu`, `sigma`, `nu` and `tau` has density given by

$$f(x) = \frac{\nu\sigma}{\mu\Gamma(\tau)} \left(\frac{x}{\mu}\right)^{\sigma\tau-1} \exp\left\{-\left(\frac{x}{\mu}\right)^{\sigma}\right\} \left\{\gamma_1\left(\tau, \left(\frac{x}{\mu}\right)^{\sigma}\right)\right\}^{\nu-1},$$

for x > 0.

Value

Returns a gamlss.family object which can be used to fit a EGG distribution in the `gamlss()` function.

Author(s)

Amylkar Urrea Montoya, <amylkar.urrea@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Gauss M. C, Edwin M.M O, Giovana O. S (2011). "The exponentiated generalized gamma distribution with application to lifetime data." *Journal of Statistical Computation and Simulation*, **81**(7), 827–842. doi:10.1080/00949650903517874.

See Also

dEGG

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma, nu and tau

y <- rEGG(n=500, mu=0.1, sigma=0.8, nu=10, tau=1.5)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, tau.fo=~1,
              family='EGG',
              control=gamlss.control(n.cyc=500, trace=FALSE))

# Extracting the fitted values for mu, sigma, nu and tau
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))
exp(coef(mod, what='tau'))


# Example 2
# Generating random values under some model

n <- 200
x1 <- runif(n, min=0.2, max=0.8)
x2 <- runif(n, min=0.2, max=0.8)
mu <- exp(-0.8 + -3 * x1)
sigma <- exp(0.77 - 2 * x2)
nu <- 10
tau <- 1.5
y <- rEGG(n=n, mu, sigma, nu, tau)

mod <- gamlss(y~x1, sigma.fo=~x2, nu.fo=~1, tau.fo=~1, family=EGG,
              control=gamlss.control(n.cyc=500, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
exp(coef(mod, what="nu"))
exp(coef(mod, what="tau"))
```

EMWEx *The Exponentiated Modifien Weibull Extension family*

Description

The Exponentiated Modifien Weibull Extension family

Usage

```
EMWEx(mu.link = "log", sigma.link = "log", nu.link = "log", tau.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.
`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.
`nu.link`	defines the nu.link, with "log" link as the default for the nu parameter.
`tau.link`	defines the tau.link, with "log" link as the default for the tau parameter.

Details

The Beta-Weibull distribution with parameters `mu`, `sigma`, `nu` and `tau` has density given by
$f(x) = \nu\sigma\tau(\frac{x}{\mu})^{\sigma-1}\exp((\frac{x}{\mu})^{\sigma} + \nu\mu(1-\exp((\frac{x}{\mu})^{\sigma})))(1-\exp(\nu\mu(1-\exp((\frac{x}{\mu})^{\sigma}))))^{\tau-1}$,
for $x > 0$, $\nu > 0$, $\mu > 0$, $\sigma > 0$ and $\tau > 0$.

Value

Returns a gamlss.family object which can be used to fit a EMWEx distribution in the `gamlss()` function.

Author(s)

Johan David Marin Benjumea, `<johand.marin@udea.edu.co>`

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Sarhan AM, Apaloo J (2013). "Exponentiated modified Weibull extension distribution." *Reliability Engineering & System Safety*, **112**, 137–144.

See Also

dEMWEx

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma, nu and tau
y <- rEMWEx(n=100, mu = 1, sigma =1.21, nu=1, tau=2)

# Fitting the model
```

```
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, tau.fo=~1, family=EMWEx,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma, nu and tau
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))
exp(coef(mod, what='tau'))

# Example 2
# Generating random values under some model
n <- 200
x1 <- runif(n, min=0.4, max=0.6)
x2 <- runif(n, min=0.4, max=0.6)
mu <- exp(0.75 - x1)
sigma <- exp(0.5 - x2)
nu <- 1
tau <- 2
x <- rEMWEx(n=n, mu, sigma, nu, tau)

mod <- gamlss(x~x1, sigma.fo=~x2, nu.fo=~1, tau.fo=~1, family=EMWEx,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
exp(coef(mod, what="nu"))
exp(coef(mod, what="tau"))
```

EOFNH *The Extended Odd Frechet-Nadarjad-Hanhighi family*

Description

The Extended Odd Frechet-Nadarjad-Hanhighi family

Usage

```
EOFNH(mu.link = "log", sigma.link = "log", nu.link = "log", tau.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.
`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.
`nu.link`	defines the nu.link, with "log" link as the default for the nu parameter.
`tau.link`	defines the tau.link, with "log" link as the default for the tau parameter.

Details

The Extended Odd Frechet-Nadarjad-Hanhighi distribution with parameters `mu`, `sigma`, `nu` and `tau` has density given by

$$f(x) = \frac{\mu\sigma\nu\tau(1+\nu x)^{\sigma-1}e^{(1-(1+\nu x)^{\sigma})}[1-(1-e^{(1-(1+\nu x)^{\sigma})})^{\mu}]^{\tau-1}}{(1-e^{(1-(1+\nu x)^{\sigma})})^{\mu\tau+1}} e^{-[(1-e^{(1-(1+\nu x)^{\sigma})})^{-\mu}-1]^{\tau}},$$

for $x > 0$, $\mu > 0$, $\sigma > 0$, $\nu > 0$ and $\tau > 0$.

Value

Returns a gamlss.family object which can be used to fit a EOFNH distribution in the `gamlss()` function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Nasiru S (2018). "Extended Odd Fréchet-G Family of Distributions." *Journal of Probability and Statistics*, **2018**.

See Also

dEOFNH

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma, nu and tau
y <- rEOFNH(n=100, mu=1, sigma=2.1, nu=0.8, tau=1)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, tau.fo=~1, family=EOFNH,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma, nu and tau
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))
exp(coef(mod, what='tau'))

# Example 2
# Generating random values under some model
n <- 200
x1 <- runif(n, min=0.4, max=0.6)
x2 <- runif(n, min=0.4, max=0.6)
mu <- exp(0.5 + x1)
```

```
sigma <- exp(0.8 + x2)
nu <- 1
tau <- 0.5
x <- rEOFNH(n=n, mu, sigma, nu, tau)

mod <- gamlss(x~x1, sigma.fo=~x2, nu.fo=~1, tau.fo=~1, family=EOFNH,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
exp(coef(mod, what="nu"))
exp(coef(mod, what="tau"))
```

---

equipment *Electronic equipment data*

Description

Time to failure in hours of 18 units of the same electronic device.

Usage

```
data(equipment)
```

Format

A vector with 18 observations.

Examples

```
data(equipment)
hist(equipment, main="", xlab="Time (h)")
```

EW *The Exponentiated Weibull family*

Description

The Exponentiated Weibull distribution

Usage

```
EW(mu.link = "log", sigma.link = "log", nu.link = "log")
```

Arguments

- `mu.link` defines the mu.link, with "log" link as the default for the mu parameter.
- `sigma.link` defines the sigma.link, with "log" link as the default for the sigma.
- `nu.link` defines the nu.link, with "log" link as the default for the nu parameter.

Details

The Exponentiated Weibull Distribution with parameters `mu`, `sigma` and `nu` has density given by

$f(x) = \nu\mu\sigma x^{\sigma-1}\exp(-\mu x^{\sigma})(1-\exp(-\mu x^{\sigma}))^{\nu-1},$

for x > 0.

Value

Returns a gamlss.family object which can be used to fit a EW distribution in the `gamlss()` function.

See Also

dEW

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma and nu
# Will not be run this example because high number is cycles
# is needed in order to get good estimates
## Not run:
y <- rEW(n=100, mu=2, sigma=1.5, nu=0.5)

# Fitting the model
require(gamlss)
mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, family='EW',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))

## End(Not run)

# Example 2
# Generating random values under some model
# Will not be run this example because high number is cycles
# is needed in order to get good estimates
## Not run:
n <- 200
x1 <- rpois(n, lambda=2)
x2 <- runif(n)
```

```
mu <- exp(2 + -3 * x1)
sigma <- exp(3 - 2 * x2)
nu <- 2
x <- rEW(n=n, mu, sigma, nu)

mod <- gamlss(x~x1, sigma.fo=~x2, nu.fo=~1, family=EW,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
exp(coef(mod, what="nu"))

## End(Not run)
```

`ExW` *The Extended Weibull family*

Description

The Extended Weibull family

Usage

```
ExW(mu.link = "log", sigma.link = "log", nu.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.
`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.
`nu.link`	defines the nu.link, with "log" link as the default for the nu parameter.

Details

The Extended Weibull distribution with parameters `mu`, `sigma` and `nu` has density given by

$f(x) = \frac{\mu \sigma \nu x^{\sigma-1} exp(-\mu x^{\sigma})}{[1-(1-\nu) exp(-\mu x^{\sigma})]^2}$,

for x > 0.

Value

Returns a gamlss.family object which can be used to fit a ExW distribution in the `gamlss()` function.

Author(s)

Amylkar Urrea Montoya, <amylkar.urrea@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Tieling Z, Min X (2007). "Failure Data Analysis with Extended Weibull Distribution." *Communications in Statistics - Simulation and Computation*, **36**, 579–592. doi:10.1080/03610910701236081.

See Also

dExW

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma and nu
y <- rExW(n=200, mu=0.3, sigma=2, nu=0.05)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, family='ExW',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))

# Example 2
# Generating random values under some model
n <- 500
x1 <- runif(n, min=0.4, max=0.6)
x2 <- runif(n, min=0.4, max=0.6)
mu <- exp(-2 + 3 * x1)
sigma <- exp(1.3 - 2 * x2)
nu <- 0.05
x <- rExW(n=n, mu, sigma, nu)

mod <- gamlss(x~x1, sigma.fo=~x2, nu.fo=~1, family=ExW,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
exp(coef(mod, what="nu"))
```

FWE *The Flexible Weibull Extension family*

Description

The function `FWE()` defines the Flexible Weibull distribution, a two parameter distribution, for a `gamlss.family` object to be used in GAMLSS fitting using the function `gamlss()`.

Usage

```
FWE(mu.link = "log", sigma.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.
`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.

Details

The Flexible Weibull extension with parameters `mu` and `sigma` has density given by

$f(x) = (\mu + \sigma/x^2) exp(\mu x - \sigma/x) exp(-exp(\mu x - \sigma/x))$

for x>0.

Value

Returns a gamlss.family object which can be used to fit a FWE distribution in the `gamlss()` function.

Examples

```
# Example 1
# Generating some random values with
# known mu and sigma
y <- rFWE(n=100, mu=0.75, sigma=1.3)

# Fitting the model
require(gamlss)
mod <- gamlss(y~1, sigma.fo=~1, family='FWE',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu and sigma
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))

# Example 2
# Generating random values under some model
n <- 200
```

```
x1 <- runif(n)
x2 <- runif(n)
mu <- exp(1.21 - 3 * x1)
sigma <- exp(1.26 - 2 * x2)
x <- rFWE(n=n, mu, sigma)

mod <- gamlss(x~x1, sigma.fo=~x2, family=FWE,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
```

GammaW *The Gamma Weibull family*

Description

The Gamma Weibull family

Usage

```
GammaW(mu.link = "log", sigma.link = "log", nu.link = "log")
```

Arguments

mu.link	defines the mu.link, with "log" link as the default for the mu parameter.
sigma.link	defines the sigma.link, with "log" link as the default for the sigma.
nu.link	defines the nu.link, with "log" link as the default for the nu parameter.

Details

The Gamma Weibull distribution with parameters mu, sigma and nu has density given by

$f(x) = \frac{\sigma\mu^\nu}{\Gamma(\nu)} x^{\nu\sigma-1} \exp(-\mu x^\sigma),$

for $x > 0$, $\mu > 0$, $\sigma \geq 0$ and $\nu > 0$.

Value

Returns a gamlss.family object which can be used to fit a GammaW distribution in the gamlss() function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Stacy EW, others (1962). "A generalization of the gamma distribution." *The Annals of mathematical statistics*, **33**(3), 1187–1192.

See Also

dGammaW

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma and nu
y <- rGammaW(n=100, mu = 0.5, sigma = 2, nu=1)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, family='GammaW',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))

# Example 2
# Generating random values under some model
n     <- 200
x1    <- runif(n)
x2    <- runif(n)
mu    <- exp(-1.6 * x1)
sigma <- exp(1.1 - 1 * x2)
nu    <- 1
x     <- rGammaW(n=n, mu, sigma, nu)

mod <- gamlss(x~x1, mu.fo=~x1, sigma.fo=~x2, nu.fo=~1, family=GammaW,
              control=gamlss.control(n.cyc=50000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
coef(mod, what='nu')
```

GGD *The Generalized Gompertz family*

Description

The Generalized Gompertz family

Usage

```
GGD(mu.link = "log", sigma.link = "log", nu.link = "log")
```

Arguments

mu.link	defines the mu.link, with "log" link as the default for the mu parameter.
sigma.link	defines the sigma.link, with "log" link as the default for the sigma.
nu.link	defines the nu.link, with "log" link as the default for the nu parameter.

Details

The Generalized Gompertz Distribution with parameters `mu`, `sigma` and `nu` has density given by

$f(x) = \nu\mu\exp(-\frac{\mu}{\sigma}(\exp(\sigma x - 1)))(1 - \exp(-\frac{\mu}{\sigma}(\exp(\sigma x - 1))))^{(\nu-1)}$,

for $x \geq 0$, $\mu > 0$, $\sigma \geq 0$ and $\nu > 0$

Value

Returns a gamlss.family object which can be used to fit a GGD distribution in the `gamlss()` function. .

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

El-Gohary A, Alshamrani A, Al-Otaibi AN (2013). “The generalized Gompertz distribution.” *Applied Mathematical Modelling*, **37**(1-2), 13–24.

See Also

dGGD

Examples

```
#Example 1
# Generating some random values with
# known mu, sigma, nu and tau
y <- rGGD(n=1000, mu=1, sigma=0.3, nu=1.5)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, family='GGD',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))

# Example 2
# Generating random values under some model
n <- 200
x1 <- runif(n, min=0.4, max=0.6)
x2 <- runif(n, min=0.4, max=0.6)
mu <- exp(0.5 - x1)
sigma <- exp(-1 - x2)
nu <- 1.5
x <- rGGD(n=n, mu, sigma, nu)

mod <- gamlss(x~x1, sigma.fo=~x2, nu.fo=~1, family=GGD,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
exp(coef(mod, what="nu"))
```

GIW *The Generalized Inverse Weibull family*

Description

The Generalized Inverse Weibull family

Usage

```
GIW(mu.link = "log", sigma.link = "log", nu.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.

sigma.link defines the sigma.link, with "log" link as the default for the sigma.

nu.link defines the nu.link, with "log" link as the default for the nu parameter.

Details

The Generalized Inverse Weibull distribution with parameters mu, sigma and nu has density given by

$f(x) = \nu\sigma\mu^{\sigma}x^{-(\sigma+1)}exp\{-\nu(\frac{\mu}{x})^{\sigma}\},$

for x > 0.

Value

Returns a gamlss.family object which can be used to fit a GIW distribution in the gamlss() function.

Author(s)

Amylkar Urrea Montoya, <amylkar.urrea@udea.edu.co>

References

Felipe R SdG, Edwin M MO, Gauss M C (2009). "The generalized inverse Weibull distribution." *Statistical papers*, **52**(3), 591–619. doi:10.1007/s0036200902713.

See Also

dGIW

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma and nu
y <- rGIW(n=200, mu=3, sigma=5, nu=0.5)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, family='GIW',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))

# Example 2
# Generating random values under some model
n <- 500
x1 <- runif(n, min=0.4, max=0.6)
x2 <- runif(n, min=0.4, max=0.6)
```

```
mu <- exp(-1.02 + 3 * x1)
sigma <- exp(1.69 - 2 * x2)
nu <- 0.5
x <- rGIW(n=n, mu, sigma, nu)

mod <- gamlss(x~x1, sigma.fo=~x2, nu.fo=~1, family=GIW,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
exp(coef(mod, what="nu"))
```

GMW *The Generalized Modified Weibull family*

Description

The Generalized modified Weibull distribution

Usage

```
GMW(mu.link = "log", sigma.link = "log", nu.link = "sqrt", tau.link = "sqrt")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.
`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.
`nu.link`	defines the nu.link, with "sqrt" link as the default for the nu parameter.
`tau.link`	defines the tau.link, with "sqrt" link as the default for the tau parameter.

Details

The Generalized modified Weibull distribution with parameters `mu`, `sigma`, `nu` and `tau` has density given by

$f(x) = \mu \sigma x^{\nu-1}(\nu + \tau x) \exp(\tau x - \mu x^{\nu} e^{\tau x})[1 - \exp(-\mu x^{\nu} e^{\tau x})]^{\sigma-1},$

for x > 0.

Value

Returns a gamlss.family object which can be used to fit a GMW distribution in the `gamlss()` function.

See Also

dGMW

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma, nu and tau
y <- rGMW(n=100, mu=2, sigma=0.5, nu=2, tau=1.5)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, tau.fo=~ 1, family='GMW',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
(coef(mod, what='nu'))^2
(coef(mod, what='tau'))^2

# Example 2
# Generating random values under some model
## Not run:
n <- 1000
x1 <- runif(n)
x2 <- runif(n)
mu <- exp(2 + -3 * x1)
sigma <- exp(3 - 2 * x2)
nu <- 2
tau <- 1.5
x <- rGMW(n=n, mu, sigma, nu, tau)

mod <- gamlss(x~x1, sigma.fo=~x2, nu.fo=~1, tau.fo=~ 1, family="GMW",
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
coef(mod, what="nu")^2
coef(mod, what="tau")^2

## End(Not run)
```

`initValuesOW` *Initial values and search region for Odd Weibull distribution*

Description

This function can be used so as to get suggestions about initial values and the search region for parameter estimation in `OW` distribution.

Usage

```
initValuesOW(
  formula,
  data = NULL,
  local_reg = loess.options(),
  interpolation = interp.options(),
  ...
)
```

Arguments

`formula`	an object of class `formula` with the response on the left of an operator ~. The right side must be 1.
`data`	an optional data frame containing the response variables. If data is not specified, the variables are taken from the environment from which `initValuesOW` is called.
`local_reg`	a list of control parameters for LOESS. See `loess.options`.
`interpolation`	a list of control parameters for interpolation function. See `interp.options`.
`...`	further arguments passed to `TTTE_Analytical`.

Details

This function performs a non-parametric estimation of the empirical total time on test (TTT) plot. Then, this estimated curve can be used so as to get suggestions about initial values and the search region for parameters based on hazard shape associated to the shape of empirical TTT plot.

Value

Returns an object of class `c("initValOW", "HazardShape")` containing:

- `sigma.start` value for $sigma$ parameter of OW distribution.
- `nu.start` value for nu parameter of OW distribution.
- `sigma.valid` search region for $sigma$ parameter of OW distribution.
- `nu.valid` search region for nu parameter of OW distribution.
- `TTTplot` Total Time on Test transform computed from the data.
- `hazard_type` shape of the hazard function determined from the TTT plot.

Author(s)

Jaime Mosquera Gutiérrez <jmosquerag@unal.edu.co>

Examples

```
# Example 1
# Bathtuh hazard and its corresponding TTT plot
y1 <- rOW(n = 1000, mu = 0.1, sigma = 7, nu = 0.08)
my_initial_guess1 <- initValuesOW(formula=y1~1)
summary(my_initial_guess1)
plot(my_initial_guess1, par_plot=list(mar=c(3.7,3.7,1,2.5),
                                      mgp=c(2.5,1,0)))

curve(hOW(x, mu = 0.022, sigma = 8, nu = 0.01), from = 0,
      to = 80, ylim = c(0, 0.04), col = "red",
      ylab = "Hazard function", las = 1)

# Example 2
# Bathtuh hazard and its corresponding TTT plot with right censored data

y2 <- rOW(n = 1000, mu = 0.1, sigma = 7, nu = 0.08)
status <- c(rep(1, 980), rep(0, 20))
my_initial_guess2 <- initValuesOW(formula=Surv(y2, status)~1)
summary(my_initial_guess2)
plot(my_initial_guess2, par_plot=list(mar=c(3.7,3.7,1,2.5),
                                      mgp=c(2.5,1,0)))

curve(hOW(x, mu = 0.022, sigma = 8, nu = 0.01), from = 0,
      to = 80, ylim = c(0, 0.04), col = "red",
      ylab = "Hazard function", las = 1)
```

IW *The Inverse Weibull family*

Description

The Inverse Weibull distribution

Usage

```
IW(mu.link = "log", sigma.link = "log")
```

Arguments

mu.link	defines the mu.link, with "log" link as the default for the mu parameter.
sigma.link	defines the sigma.link, with "log" link as the default for the sigma.

Details

The Inverse Weibull distribution with parameters `mu`, `sigma` has density given by

$f(x) = \mu\sigma x^{-\sigma-1}\exp(\mu x^{-\sigma})$

for $x > 0$, $\mu > 0$ and $\sigma > 0$

Value

Returns a gamlss.family object which can be used to fit a IW distribution in the `gamlss()` function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Drapella A (1993). "The complementary Weibull distribution: unknown or just forgotten?" *Quality and Reliability Engineering International*, **9**(4), 383–385.

See Also

dIW

Examples

```
# Example 1
# Generating some random values with
# known mu and sigma
y <- rIW(n=100, mu=5, sigma=2.5)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, mu.fo=~1, sigma.fo=~1, family='IW',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))

# Example 2
# Generating random values under some model
n <- 200
x1 <- rpois(n, lambda=2)
x2 <- runif(n)
mu <- exp(2 + -1 * x1)
sigma <- exp(2 - 2 * x2)
x <- rIW(n=n, mu, sigma)

mod <- gamlss(x~x1, mu.fo=~1, sigma.fo=~x2, family=IW,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
```

`KumIW` *The Kumaraswamy Inverse Weibull family*

Description

The Kumaraswamy Inverse Weibull family

Usage

```
KumIW(mu.link = "log", sigma.link = "log", nu.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.
`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.
`nu.link`	defines the nu.link, with "log" link as the default for the nu parameter.

Details

The Kumaraswamy Inverse Weibull Distribution with parameters `mu`, `sigma` and `nu` has density given by

$f(x) = \mu\sigma\nu x^{-\mu-1} \exp -\sigma x^{-\mu}(1 - \exp -\sigma x^{-\mu})^{\nu-1}$,

for $x > 0$, $\mu > 0$, $\sigma > 0$ and $\nu > 0$.

Value

Returns a gamlss.family object which can be used to fit a KumIW distribution in the `gamlss()` function.

Author(s)

Johan David Marin Benjumea, `<johand.marin@udea.edu.co>`

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Shahbaz MQ, Shahbaz S, Butt NS (2012). "The Kumaraswamy-Inverse Weibull Distribution." *Shahbaz, MQ, Shahbaz, S., & Butt, NS (2012). The Kumaraswamy–Inverse Weibull Distribution. Pakistan journal of statistics and operation research*, **8**(3), 479–489.

See Also

dKumIW

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma, nu and tau
y <- rKumIW(n=1000, mu = 1.5, sigma=  1.5, nu = 5)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, family='KumIW',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))

# Example 2
# Generating random values under some model
n <- 200
x1 <- runif(n, min=0.4, max=0.6)
x2 <- runif(n, min=0.4, max=0.6)
mu <- exp(1 - x1)
sigma <- exp(1 - x2)
nu <- 5
x <- rKumIW(n=n, mu, sigma, nu)

mod <- gamlss(x~x1, sigma.fo=~x2, nu.fo=~1, family=KumIW,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
exp(coef(mod, what="nu"))
```

`LIN` *The Lindley family*

Description

The function `LIN()` defines the Lindley distribution with only one parameter for a `gamlss.family` object to be used in GAMLSS fitting using the function `gamlss()`.

Usage

```
LIN(mu.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.

Details

The Lindley with parameter mu has density given by

$f(x) = \frac{\mu^2}{\mu+1}(1+x)\exp(-\mu x),$

for x > 0 and $\mu > 0$.

Value

Returns a gamlss.family object which can be used to fit a LIN distribution in the `gamlss()` function.

Author(s)

Freddy Hernandez <fhernanb@unal.edu.co>

References

Lindley DV (1958). "Fiducial distributions and Bayes' theorem." *Journal of the Royal Statistical Society. Series B (Methodological)*, 102–107.

Lindley DV (1965). *Introduction to probability and statistics: from a Bayesian viewpoint. 2. Inference*. CUP Archive.

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma and nu
y <- rLIN(n=200, mu=2)

# Fitting the model
require(gamlss)
mod <- gamlss(y ~ 1, family="LIN")

# Extracting the fitted values for mu
# using the inverse link function
exp(coef(mod, what='mu'))

# Example 2
# Generating random values under some model
n <- 100
x1 <- runif(n=n)
x2 <- runif(n=n)
eta <- 1 + 3 * x1 - 2 * x2
mu <- exp(eta)
y <- rLIN(n=n, mu=mu)

mod <- gamlss(y ~ x1 + x2, family=LIN)

coef(mod, what='mu')
```

`LW` *The Log-Weibull family*

Description

The Log-Weibull distribution

Usage

```
LW(mu.link = "identity", sigma.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.
`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.

Details

The Log-Weibull Distribution with parameters `mu` and `sigma` has density given by

$f(y) = (1/\sigma)e^{((y-\mu)/\sigma)} exp\{-e^{((y-\mu)/\sigma)}\},$

for - `infty` < y < `infty`.

Value

Returns a gamlss.family object which can be used to fit a LW distribution in the `gamlss()` function.

Author(s)

Amylkar Urrea Montoya, `<amylkar.urrea@udea.edu.co>`

References

Almalki SJ, Nadarajah S (2014). “Modifications of the Weibull distribution: A review.” *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

E.J G (1958). *Statistics of extremes*. Columbia University Press. ISBN 10:0231021909.

See Also

dLW

Examples

```
# Example 1
# Generating some random values with
# known mu and sigma
y <- rLW(n=100, mu=0, sigma=1.5)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, family= 'LW',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu and sigma
# using the inverse link function
coef(mod, 'mu')
exp(coef(mod, 'sigma'))

# Example 2
# Generating random values under some model
n <- 200
x1 <- runif(n, min=0.4, max=0.6)
x2 <- runif(n, min=0.4, max=0.6)
mu <- 1.5 - 3 * x1
sigma <- exp(1.4 - 2 * x2)
x <- rLW(n=n, mu, sigma)

mod <- gamlss(x~x1, sigma.fo=~x2, family=LW,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
```

`mice` *Mice mortality data*

Description

The ages at death in weeks for male mice exposed to 240r of gamma radiation.

Usage

```
data(mice)
```

Format

A vector with 208 data points.

Examples

```
data(mice)
hist(mice, main="", xlab="Time (weeks)", freq=FALSE)
lines(density(mice), col="blue", lwd=2)
```

MOEIW *The Marshall-Olkin Extended Inverse Weibull family*

Description

The Marshall-Olkin Extended Inverse Weibull family

Usage

```
MOEIW(mu.link = "log", sigma.link = "log", nu.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.
`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.
`nu.link`	defines the nu.link, with "log" link as the default for the nu parameter.

Details

The Marshall-Olkin Extended Inverse Weibull distribution with parameters `mu`, `sigma` and `nu` has density given by

$f(x) = \frac{\mu \sigma \nu x^{-(\sigma+1)} exp\{-\mu x^{-\sigma}\}}{\{\nu-(\nu-1) exp\{-\mu x^{-\sigma}\}\}^2},$

for x > 0.

Value

Returns a gamlss.family object which can be used to fit a MOEIW distribution in the `gamlss()` function.

Author(s)

Amylkar Urrea Montoya, `<amylkar.urrea@udea.edu.co>`

References

Hassan M O, A.H E, A.M.K T, Abdulkareem M Bc (2017). “Extended inverse Weibull distribution with reliability application.” *Journal of the Egyptian Mathematical Society*, **25**, 343–349. doi:10.1016/j.joems.2017.02.006, `http://dx.doi.org/10.1016/j.joems.2017.02.006`.

See Also

dMOEIW

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma and nu
y <- rMOEIW(n=400, mu=0.6, sigma=1.7, nu=0.3)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, family='MOEIW',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))

# Example 2
# Generating random values under some model
n <- 400
x1 <- runif(n, min=0.4, max=0.6)
x2 <- runif(n, min=0.4, max=0.6)
mu <- exp(-2.02 + 3 * x1)
sigma <- exp(2.23 - 2 * x2)
nu <- 0.3
x <- rMOEIW(n=n, mu, sigma, nu)

mod <- gamlss(x~x1, sigma.fo=~x2, nu.fo=~1, family=MOEIW,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
exp(coef(mod, what="nu"))
```

MOEW *The Marshall-Olkin Extended Weibull family*

Description

The Marshall-Olkin Extended Weibull family

Usage

```
MOEW(mu.link = "log", sigma.link = "log", nu.link = "log")
```

Arguments

mu.link	defines the mu.link, with "log" link as the default for the mu parameter.

`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.
`nu.link`	defines the nu.link, with "log" link as the default for the nu parameter.

Details

The Marshall-Olkin Extended Weibull distribution with parameters `mu`, `sigma` and `nu` has density given by

$f(x) = \frac{\mu\sigma\nu(\nu x)^{\sigma-1}exp\{-(\nu x)^{\sigma}\}}{\{1-(1-\mu)exp\{-(\nu x)^{\sigma}\}\}^2},$

for x > 0.

Value

Returns a gamlss.family object which can be used to fit a MOEW distribution in the `gamlss()` function.

Author(s)

Amylkar Urrea Montoya, `<amylkar.urrea@udea.edu.co>`

References

Almalki SJ, Nadarajah S (2014). “Modifications of the Weibull distribution: A review.” *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

M.E G, E.K A, R.A J (2005). “Marshall–Olkin extended weibull distribution and its application to censored data.” *Journal of Applied Statistics*, **32**(10), 1025–1034. doi:10.1080/02664760500165008.

See Also

dMOEW

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma and nu
y <- rMOEW(n=400, mu=0.5, sigma=0.7, nu=1)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, family='MOEW',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))

# Example 2
```

```
# Generating random values under some model
n <- 500
x1 <- runif(n, min=0.4, max=0.6)
x2 <- runif(n, min=0.4, max=0.6)
mu <- exp(-1.20 + 3 * x1)
sigma <- exp(0.84 - 2 * x2)
nu <- 1
x <- rMOEW(n=n, mu, sigma, nu)

mod <- gamlss(x~x1, sigma.fo=~x2, nu.fo=~1, family=MOEW,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
exp(coef(mod, what="nu"))
```

MOK *The Marshall-Olkin Kappa family*

Description

The Marshall-Olkin Kappa family

Usage

```
MOK(mu.link = "log", sigma.link = "log", nu.link = "log", tau.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.
`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.
`nu.link`	defines the nu.link, with "log" link as the default for the nu parameter.
`tau.link`	defines the tau.link, with "log" link as the default for the tau parameter.

Details

The Marshall-Olkin Kappa distribution with parameters `mu`, `sigma`, `nu` and `tau` has density given by

$$f(x) = \frac{\tau \frac{\mu\nu}{\sigma} \left(\frac{x}{\sigma}\right)^{\nu-1} \left(\mu + \left(\frac{x}{\sigma}\right)^{\mu\nu}\right)^{-\frac{\mu+1}{\mu}}}{\left(\tau + (1-\tau) \left(\frac{\left(\frac{x}{\sigma}\right)^{\mu\nu}}{\mu + \left(\frac{x}{\sigma}\right)^{\mu\nu}}\right)^{\frac{1}{\mu}}\right)^2}$$

for x > 0.

Value

Returns a gamlss.family object which can be used to fit a MOK distribution in the `gamlss()` function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Javed M, Nawaz T, Irfan M (2018). “The Marshall-Olkin kappa distribution: properties and applications.” *Journal of King Saud University-Science*.

See Also

dMOK

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma, nu and tau
y <- rMOK(n=100, mu = 1, sigma = 3.5, nu = 3, tau = 2)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, tau.fo=~1, family=MOK,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma, nu and tau
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))
exp(coef(mod, what='tau'))

# Example 2
# Generating random values under some model
n <- 200
x1 <- runif(n, min=0.4, max=0.6)
x2 <- runif(n, min=0.4, max=0.6)
mu <- exp(0.5 + x1)
sigma <- exp(0.8 + x2)
nu <- 1
tau <- 0.5
x <- rMOK(n=n, mu, sigma, nu, tau)

mod <- gamlss(x~x1, sigma.fo=~x2, nu.fo=~1, tau.fo=~1, family=MOK,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
exp(coef(mod, what="nu"))
exp(coef(mod, what="tau"))
```

MW *The Modified Weibull family*

Description

#' The Modified Weibull distribution

Usage

```
MW(mu.link = "log", sigma.link = "log", nu.link = "log")
```

Arguments

mu.link	defines the mu.link, with "log" link as the default for the mu parameter.
sigma.link	defines the sigma.link, with "log" link as the default for the sigma.
nu.link	defines the nu.link, with "log" link as the default for the nu parameter.

Details

The Modified Weibull distribution with parameters `mu`, `sigma` and `nu` has density given by

$f(x) = \mu(\sigma + \nu x)x^(\sigma - 1)\exp(\nu x)\exp(-\mu x^(\sigma)\exp(\nu x)),$

for $x > 0$, $\mu > 0$, $\sigma \geq 0$ and $\nu \geq 0$.

Value

Returns a gamlss.family object which can be used to fit a MW distribution in the `gamlss()` function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). “Modifications of the Weibull distribution: A review.” *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Lai CD, Xie M, Murthy DNP (2003). “A modified Weibull distribution.” *IEEE Transactions on reliability*, **52**(1), 33–37.

See Also

dMW

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma and nu
y <- rMW(n=100, mu = 2, sigma = 1.5, nu = 0.2)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, family= 'MW',
               control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))

# Example 2
# Generating random values under some model
n     <- 200
x1    <- rpois(n, lambda=2)
x2    <- runif(n)
mu    <- exp(3 -1 * x1)
sigma <- exp(2 - 2 * x2)
nu    <- 0.2
x     <- rMW(n=n, mu, sigma, nu)

mod <- gamlss(x~x1, mu.fo=~x1, sigma.fo=~x2, nu.fo=~1, family=MW,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
coef(mod, what='nu')
```

---

myOW_region *Custimized region search for odd Weibull distribution*

Description

This function can be used to modify OW gamlss.family object in order to set a customized region search for gamlss() function.

Usage

```
myOW_region(family = OW, valid.values = "auto", initVal)
```

Arguments

family	The OW family. This arguments allows the user to modify input arguments of the family, like the link functions.
valid.values	a list of character elements specifying the region for sigma and/or nu. See **Details** and **Examples** section to learn about its use.
initVal	An initValOW object generated with initValuesOW function.

Details

This function was created to help users to fit OW distribution easily bounding the parametric space for sigma and nu.

The valid.values must be defined as a list of characters containing a call of the all function.

Value

Returns a gamlss.family object which can be used to fit an OW distribution in the gamlss() function.

Author(s)

Jaime Mosquera Gutiérrez <jmosquerag@unal.edu.co>

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma and nu
y <- rOW(n=200, mu=0.2, sigma=4, nu=0.05)

# Custom search region
myvalues <- list(sigma="all(sigma > 1)",
                 nu="all(nu < 1) & all(nu < 1)")

my_initial_guess <- initValuesOW(formula=y~1)
summary(my_initial_guess)

# OW family modified with 'myOW_region'
require(gamlss)
myOW <- myOW_region(valid.values=myvalues, initVal=my_initial_guess)
mod1 <- gamlss(y~1, sigma.fo=~1, nu.fo=~1,
               sigma.start=param.startOW('sigma', my_initial_guess),
               nu.start=param.startOW('nu', my_initial_guess),
               control=gamlss.control(n.cyc=300, trace=FALSE),
               family=myOW)

exp(coef(mod1, what='mu'))
exp(coef(mod1, what='sigma'))
exp(coef(mod1, what='nu'))

# Example 2
```

```
# Same example using another link function and using 'myOW_region'
# in the argument 'family'
mod2 <- gamlss(y~1, sigma.fo=~1, nu.fo=~1,
               sigma.start=2, nu.start=0.1,
               control=gamlss.control(n.cyc=300, trace=FALSE),
               family=myOW_region(family=OW(sigma.link='identity'),
                                  valid.values=myvalues,
                                  initVal=my_initial_guess))

exp(coef(mod2, what='mu'))
coef(mod2, what='sigma')
exp(coef(mod2, what='nu'))
```

`OW` *The Odd Weibull family*

Description

The function `OW()` defines the Odd Weibull distribution, a three parameter distribution, for a `gamlss.family` object to be used in GAMLSS fitting using the function `gamlss()`.

Usage

```
OW(mu.link = "log", sigma.link = "log", nu.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.
`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.
`nu.link`	defines the nu.link, with "log" link as the default for the nu.

Details

The odd Weibull with parameters `mu`, `sigma` and `nu` has density given by

$$f(t) = \left(\frac{\sigma\nu}{t}\right)(\mu t)^{\sigma} e^{(\mu t)^{\sigma}} \left(e^{(\mu t)^{\sigma}} - 1\right)^{\nu-1} \left[1 + \left(e^{(\mu t)^{\sigma}} - 1\right)^{\nu}\right]^{-2}$$

for x > 0.

Value

Returns a gamlss.family object which can be used to fit a OW distribution in the `gamlss()` function.

Author(s)

Jaime Mosquera Gutiérrez <jmosquerag@unal.edu.co>

References

Cooray K (2006). "Generalization of the Weibull distribution: The odd Weibull family." *Statistical Modelling*, **6**(3), 265–277. ISSN 1471082X, doi:10.1191/1471082X06st116oa.

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma and nu
y <- rOW(n=200, mu=0.1, sigma=7, nu = 1.1)

# Fitting the model
require(gamlss)
mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, family="OW",
              control=gamlss.control(n.cyc=500, trace=FALSE))

# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))

# Example 2
# Generating random values under some model
n <- 500
x1 <- runif(n)
x2 <- runif(n)
x3 <- runif(n)
mu <- exp(1.2 + 2 * x1)
sigma <- 2.12 + 3 * x2
nu <- exp(0.2 - x3)
x <- rOW(n=n, mu, sigma, nu)

mod <- gamlss(x~x1, sigma.fo=~x2, nu.fo=~x3,
              family=OW(sigma.link='identity'),
              control=gamlss.control(n.cyc=300, trace=FALSE))

coef(mod, what='mu')
coef(mod, what='sigma')
coef(mod, what='nu')
```

`param.startOW` *Initial values extraction for Odd Weibull distribution*

Description

This function can be used to extract initial values found with empirical time on test transform (TTT) through `initValuesOW` function. This is used for parameter estimation in `OW` distribution.

Usage

```
param.startOW(param, initValOW)
```

Arguments

`param`	a character used to specify the parameter required. It can take the values "sigma" or "nu".
`initValOW`	an `initValOW` object generated with `initValuesOW` function.

Details

This function just gets initial values computed with `initValuesOW` for `OW` family. It must be called in `sigma.start` and `nu.start` arguments from `gamlss` function. This function is useful only if user want to set start values automatically with TTT plot. See example for an illustration.

Value

A length-one vector numeric value corresponding to the initial value of the parameter specified in `param` extracted from a `initValuesOW` object specified in the `initValOW` input argument.

Author(s)

Jaime Mosquera Gutiérrez <jmosquerag@unal.edu.co>

Examples

```
# Random data generation (OW distributed)
y <- rOW(n=500, mu=0.05, sigma=0.6, nu=2)

# Initial values with TTT plot
iv <- initValuesOW(formula = y ~ 1)
summary(iv)

# This data is from unimodal hazard
# See TTT estimate from sample
plot(iv, legend_options=list(pos=1.03))

# See the true hazard
curve(hOW(x, mu=0.05, sigma=0.6, nu=2), to=100, lwd=3, ylab="h(x)")

# Finally, we fit the model
require(gamlss)
con.out <- gamlss.control(n.cyc = 300, trace = FALSE)
con.in <- glim.control(cyc = 300)

(sigma.start <- param.startOW("sigma", iv))
(nu.start <- param.startOW("nu", iv))

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, control=con.out, i.control=con.in,
              family=myOW_region(OW(sigma.link="identity", nu.link="identity"),
                                 valid.values="auto", iv),
```

```
               sigma.start=sigma.start, nu.start=nu.start)

# Estimates are close to actual values
(mu <- exp(coef(mod, what = "mu")))
(sigma <- coef(mod, what = "sigma"))
(nu <- coef(mod, what = "nu"))
```

PL *The Power Lindley family*

Description

Power Lindley distribution

Usage

```
PL(mu.link = "log", sigma.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.
`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.

Details

The Power Lindley Distribution with parameters `mu` and `sigma` has density given by

$f(x) = \frac{\mu\sigma^2}{\sigma+1}(1+x^{\mu})x^{\mu-1}\exp(-\sigma x^{\mu}),$

for x > 0.

Value

Returns a gamlss.family object which can be used to fit a PL distribution in the `gamlss()` function.

Author(s)

Amylkar Urrea Montoya, `<amylkar.urrea@udea.edu.co>`

References

Almalki SJ, Nadarajah S (2014). “Modifications of the Weibull distribution: A review.” *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Ghitanya ME, Al-Mutairi DK, Balakrishnanb N, Al-Enezi LJ (2013). “Power Lindley distribution and associated inference.” *Computational Statistics and Data Analysis*, **64**, 20–33. doi:10.1016/j.csda.2013.02.026.

See Also

dPL

Examples

```
# Example 1
# Generating some random values with
# known mu and sigma
y <- rPL(n=100, mu=1.5, sigma=0.2)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, family= 'PL',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu and sigma
# using the inverse link function
exp(coef(mod, 'mu'))
exp(coef(mod, 'sigma'))

# Example 2
# Generating random values under some model
n <- 200
x1 <- runif(n, min=0.4, max=0.6)
x2 <- runif(n, min=0.4, max=0.6)
mu <- exp(1.2 - 2 * x1)
sigma <- exp(0.8 - 3 * x2)
x <- rPL(n=n, mu, sigma)

mod <- gamlss(x~x1, sigma.fo=~x2, family=PL,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
```

QXGP *The Quasi XGamma Poisson family*

Description

The Quasi XGamma Poisson family

Usage

```
QXGP(mu.link = "log", sigma.link = "log", nu.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.
`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.
`nu.link`	defines the nu.link, with "log" link as the default for the nu parameter.

Details

The Quasi XGamma Poisson distribution with parameters `mu`, `sigma` and `nu` has density given by

$f(x) = K(\mu, \sigma, \nu)(\frac{\sigma^2 x^2}{2} + \mu)exp(\frac{\nu exp(-\sigma x)(1+\mu+\sigma x+\frac{\sigma^2 x^2}{2})}{1+\mu} - \sigma x),$

for $x > 0$, $\mu > 0$, $\sigma > 0$, $\nu > 1$.

where

$K(\mu, \sigma, \nu) = \frac{\nu\sigma}{(exp(\nu)-1)(1+\mu)}$

Value

Returns a gamlss.family object which can be used to fit a QXGP distribution in the `gamlss()` function.

Author(s)

Amylkar Urrea Montoya, `<amylkar.urrea@udea.edu.co>`

References

Subhradev S, Mustafa C K, Haitham M Y (2018). "The Quasi XGamma-Poisson distribution: Properties and Application." *Istatistik: Journal of the Turkish Statistical Assocation*, **11**(3), 65–76. ISSN 1300-4077, https://dergipark.org.tr/en/pub/ijtsa/issue/42850/518206.

See Also

dQXGP

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma and nu
y <- rQXGP(n=200, mu=4, sigma=2, nu=3)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, family='QXGP',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
exp(coef(mod, what='mu'))
```

```
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))

# Example 2
# Generating random values under some model
n <- 2000
x1 <- runif(n, min=0.4, max=0.6)
x2 <- runif(n, min=0.4, max=0.6)
mu <- exp(-2.19 + 3 * x1)
sigma <- exp(1 - 2 * x2)
nu <- 1
x <- rQXGP(n=n, mu, sigma, nu)

mod <- gamlss(x~x1, sigma.fo=~x2, nu.fo=~1, family=QXGP,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
exp(coef(mod, what="nu"))
```

`RW` *The Reflected Weibull family*

Description

Reflected Weibull distribution

Usage

```
RW(mu.link = "log", sigma.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.
`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.

Details

The Reflected Weibull Distribution with parameters `mu` and `sigma` has density given by

$f(y) = \mu\sigma(-y)^{\sigma-1}e^{-\mu(-y)^{\sigma}},$

for y < 0

Value

Returns a gamlss.family object which can be used to fit a RW distribution in the `gamlss()` function.

Author(s)

Amylkar Urrea Montoya, `<amylkar.urrea@udea.edu.co>`

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Clifford Cohen A (1973). "The Reflected Weibull Distribution." *Technometrics*, **15**(4), 867–873. doi:10.2307/1267396.

See Also

dRW

Examples

```
# Example 1
# Generating some random values with
# known mu and sigma
y <- rRW(n=100, mu=1, sigma=1)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, family= 'RW',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu and sigma
# using the inverse link function
exp(coef(mod, 'mu'))
exp(coef(mod, 'sigma'))

# Example 2
# Generating random values under some model
n <- 200
x1 <- runif(n, min=0.4, max=0.6)
x2 <- runif(n, min=0.4, max=0.6)
mu <- exp(1.5 - 1.5 * x1)
sigma <- exp(2 - 2 * x2)
x <- rRW(n=n, mu, sigma)

mod <- gamlss(x~x1, sigma.fo=~x2, family=RW,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
```

---

summary.initValOW *Summary of* initValOW *objects*

Description

This summary method displays initial values and search regions for OW family.

Usage

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

Arguments

`object`	an object of class `initVal`, generated with `initValuesOW`.
`...`	extra arguments

Value

No return value, it just prints out in the console the initial values and the search regions for $sigma$ and nu from OW distribution (see `dOW`).

Author(s)

Jaime Mosquera Gutiérrez <jmosquerag@unal.edu.co>

SZMW — *The Sarhan and Zaindin's Modified Weibull family*

Description

The Sarhan and Zaindin's Modified Weibull distribution

Usage

```
SZMW(mu.link = "log", sigma.link = "log", nu.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.
`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.
`nu.link`	defines the nu.link, with "log" link as the default for the nu parameter.

Details

The Sarhan and Zaindin's Modified Weibull distribution with parameters `mu`, `sigma` and `nu` has density given by

$f(x) = (\mu + \sigma\nu x^(\nu - 1)) \exp(-\mu x - \sigma x^{\nu}),$

for $x > 0$, $\mu > 0$, $\sigma > 0$ and $\nu > 0$.

Value

Returns a gamlss.family object which can be used to fit a SZMW distribution in the `gamlss()` function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Sarhan AM, Zaindin M (2009). "Modified Weibull distribution." *APPS. Applied Sciences*, **11**, 123–136.

See Also

dSZMW

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma and nu
y <- rSZMW(n=100, mu = 1, sigma = 1, nu = 1.5)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, family='SZMW',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))

# Example 2
# Generating random values under some model
n     <- 200
x1    <- runif(n)
x2    <- runif(n)
mu    <- exp(-1.6 * x1)
sigma <- exp(0.9 - 1 * x2)
nu    <- 1.5
x     <- rSZMW(n=n, mu, sigma, nu)

mod <- gamlss(x~x1, mu.fo=~x1, sigma.fo=~x2, nu.fo=~1, family=SZMW,
              control=gamlss.control(n.cyc=50000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
coef(mod, what='nu')
```

WG *The Weibull Geometric family*

Description

The Weibull Geometric distribution

Usage

```
WG(mu.link = "log", sigma.link = "log", nu.link = "logit")
```

Arguments

mu.link	defines the mu.link, with "log" link as the default for the mu parameter.
sigma.link	defines the sigma.link, with "log" link as the default for the sigma.
nu.link	defines the nu.link, with "log" link as the default for the nu parameter.

Details

The weibull geometric distribution with parameters `mu`, `sigma` and `nu` has density given by

$f(x) = (\sigma\mu^\sigma(1-\nu)x^(\sigma-1)\exp(-(\mu x)^\sigma))(1-\nu\exp(-(\mu x)^\sigma))^{-2}$,

for $x > 0$, $\mu > 0$, $\sigma > 0$ and $0 < \nu < 1$.

Value

Returns a gamlss.family object which can be used to fit a WG distribution in the `gamlss()` function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Barreto-Souza W, de Morais AL, Cordeiro GM (2011). "The Weibull-geometric distribution." *Journal of Statistical Computation and Simulation*, **81**(5), 645–657.

See Also

dWG

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma and nu
y <- rWG(n=100,  mu = 0.9, sigma = 2, nu = 0.5)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, family='WG',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))

# Example 2
# Generating random values under some model
n     <- 200
x1    <- runif(n)
x2    <- runif(n)
mu    <- exp(- 0.2 * x1)
sigma <- exp(1.2 - 1 * x2)
nu    <- 0.5
x     <- rWG(n=n, mu, sigma, nu)

mod <- gamlss(x~x1, mu.fo=~x1, sigma.fo=~x2, nu.fo=~1, family=WG,
              control=gamlss.control(n.cyc=50000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
coef(mod, what='nu')
```

WGEE *The Weigted Generalized Exponential-Exponential family*

Description

The Weigted Generalized Exponential-Exponential family

Usage

```
WGEE(mu.link = "log", sigma.link = "log", nu.link = "log")
```

Arguments

`mu.link` defines the mu.link, with "log" link as the default for the mu parameter.

`sigma.link` defines the sigma.link, with "log" link as the default for the sigma.

`nu.link` defines the nu.link, with "log" link as the default for the nu parameter.

Details

The Weigted Generalized Exponential-Exponential distribution with parameters `mu`, `sigma` and `nu` has density given by

$f(x) = \sigma\nu\exp(-\nu x)(1-\exp(-\nu x))^{\sigma-1}(1-\exp(-\mu\nu x))/1-\sigma B(\mu+1,\sigma),$

for $x > 0$, $\mu > 0$, $\sigma > 0$ and $\nu > 0$.

Value

Returns a gamlss.family object which can be used to fit a WGEE distribution in the `gamlss()` function.

Author(s)

Johan David Marin Benjumea, <johand.marin@udea.edu.co>

References

Mahdavi A (2015). "Two weighted distributions generated by exponential distribution." *Journal of Mathematical Extension*, **9**, 1–12.

See Also

dWGEE

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma and  nu
y <- rWGEE(n=1000, mu = 5, sigma = 0.5, nu = 1)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, family='WGEE',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))

# Example 2
# Generating random values under some model
n <- 500
x1 <- runif(n, min=0.4, max=0.6)
```

```
x2 <- runif(n, min=0.4, max=0.6)
mu <- exp(2 - x1)
sigma <- exp(1 - 3*x2)
nu <- 1
x <- rWGEE(n=n, mu, sigma, nu)

mod <- gamlss(x~x1, sigma.fo=~x2, nu.fo=~1, family=WGEE,
              control=gamlss.control(n.cyc=50000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
exp(coef(mod, what="nu"))
```

WP *The Weibull Poisson family*

Description

The Weibull Poisson family

Usage

```
WP(mu.link = "log", sigma.link = "log", nu.link = "log")
```

Arguments

`mu.link`	defines the mu.link, with "log" link as the default for the mu parameter.
`sigma.link`	defines the sigma.link, with "log" link as the default for the sigma.
`nu.link`	defines the nu.link, with "log" link as the default for the nu parameter.

Details

The Weibull Poisson distribution with parameters `mu`, `sigma` and `nu` has density given by

$f(x) = \frac{\mu\sigma\nu e^{-\nu}}{1-e^{-\nu}} x^{\mu-1} exp(-\sigma x^{\mu} + \nu exp(-\sigma x^{\mu})),$

for x > 0.

Value

Returns a gamlss.family object which can be used to fit a WP distribution in the `gamlss()` function.

Author(s)

Amylkar Urrea Montoya, `<amylkar.urrea@udea.edu.co>`

References

Almalki SJ, Nadarajah S (2014). "Modifications of the Weibull distribution: A review." *Reliability Engineering & System Safety*, **124**, 32–55. doi:10.1016/j.ress.2013.11.010.

Wanbo L, Daimin S (1967). "A new compounding life distribution: the Weibull–Poisson distribution." *Journal of Applied Statistics*, **9**(1), 21–38. doi:10.1080/02664763.2011.575126, https://doi.org/10.1080/02664763.2011.575126.

See Also

dWP

Examples

```
# Example 1
# Generating some random values with
# known mu, sigma and nu
y <- rWP(n=300, mu=1.5, sigma=0.5, nu=0.5)

# Fitting the model
require(gamlss)

mod <- gamlss(y~1, sigma.fo=~1, nu.fo=~1, family='WP',
              control=gamlss.control(n.cyc=5000, trace=FALSE))

# Extracting the fitted values for mu, sigma and nu
# using the inverse link function
exp(coef(mod, what='mu'))
exp(coef(mod, what='sigma'))
exp(coef(mod, what='nu'))

# Example 2
# Generating random values under some model
n <- 2000
x1 <- runif(n, min=0.4, max=0.6)
x2 <- runif(n, min=0.4, max=0.6)
mu <- exp(-1.3 + 3 * x1)
sigma <- exp(0.69 - 2 * x2)
nu <- 0.5
x <- rWP(n=n, mu, sigma, nu)

mod <- gamlss(x~x1, sigma.fo=~x2, nu.fo=~1, family=WP,
              control=gamlss.control(n.cyc=5000, trace=FALSE))

coef(mod, what="mu")
coef(mod, what="sigma")
exp(coef(mod, what="nu"))
```

Index