Package ‘ntsDists’

April 17, 2024

Type Package

Title Neutrosophic Distributions

Version 2.1.1

Maintainer Danial Mazarei <danial.mazarei@gmail.com>

Description Computes the pdf, cdf, quantile function and generating random numbers for neutrosophic distributions. This family have been developed by different authors in the recent years. See Patro and Smarandache (2016) <doi:10.5281/zenodo.571153> and Rao et al (2023) <doi:10.5281/zenodo.7832786>.

BugReports https://github.com/dmazarei/ntsDists/issues

License GPL (>= 2)

URL https://github.com/dmazarei/ntsDists

Encoding UTF-8

LazyData false

RoxygenNote 7.2.3

Depends R (>= 3.5)

NeedsCompilation no

Author Danial Mazarei [aut, cre] (<https://orcid.org/0000-0002-3633-9298>),
Mina Norouzirad [aut] (<https://orcid.org/0000-0003-0311-6888>),
Amin Roshani [aut] (<https://orcid.org/0000-0002-3329-5330>),
Gadde Srinivasa Rao [ctb] (<https://orcid.org/0000-0002-3683-5486>),
Foad Esmaeili [ctb] (<https://orcid.org/0000-0002-9638-0807>),
FCT, I.P. [fnd] (under the scope of the projects UIDB/00297/2020 and UIDP/00297/2020 (NovaMath))

Repository CRAN

Date/Publication 2024-04-17 07:00:02 UTC

R topics documented:

`balls` *Balls data*

Description

It is related to failure times of 23 bearing balls.

Format

A data.frame with 23 observations of failure times of bearing balls.

Source

Lawless, J. F. (2003). Statistical Models and Methods for Lifetime Data, Wiley, Hoboken, NJ, USA.

Salam, S., Khan, Z., Ayed, H., Brahmia, A., Amin, A. (2021). The Neutrosophic Lognormal Model in Lifetime Data Analysis: Properties and Applications, *Fuzzy Sets and Their Applications in Mathematics*, Article ID 6337759.

Examples

```
data("balls")
balls
```

Neutrosophic Beta — *Neutrosophic Beta Distribution*

Description

Density, distribution function, quantile function and random generation for the neutrosophic Beta distribution with shape parameters `shape1` = α_N and `shape2` = β_N.

Usage

```
dnsBeta(x, shape1, shape2)

pnsBeta(q, shape1, shape2, lower.tail = TRUE)

qnsBeta(p, shape1, shape2)

rnsBeta(n, shape1, shape2)
```

Arguments

`x`	a vector or matrix of observations for which the pdf needs to be computed.
`shape1`	the first shape parameter, which must be a positive interval.
`shape2`	the second shape parameter, which must be a positive interval.
`q`	a vector or matrix of quantiles for which the cdf needs to be computed.
`lower.tail`	logical; if TRUE (default), probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
`p`	a vector or matrix of probabilities for which the quantile needs to be computed.
`n`	number of random values to be generated.

Details

The neutrosophic beta distribution with parameters α_N and β_N has the probability density function

$$f_N(x) = \frac{1}{B(\alpha_N, \beta_N)} x^{\alpha_N - 1}(1-x)^{\beta_N - 1}$$

for $\alpha_N \in (\alpha_L, \alpha_U)$, the first shape parameter which must be a positive interval, and $\beta_N \in (\beta_L, \beta_U)$, the second shape parameter which must also be a positive interval, and $0 \leq x \leq 1$. The function $B(a, b)$ returns the beta function and can be calculated using `beta`.

Value

`dnsBeta` gives the density function

`pnsBeta` gives the distribution function

`qnsBeta` gives the quantile function

`rnsBeta` generates random values from the neutrosophic Beta distribution.

References

Sherwani, R. Ah. K., Naeem, M., Aslam, M., Reza, M. A., Abid, M., Abbas, S. (2021). Neutrosophic beta distribution with properties and applications. *Neutrosophic Sets and Systems*, 41, 209-214.

Examples

```
dnsBeta(x = c(0.1, 0.2), shape1 = c(1, 1), shape2 = c(2, 2))
dnsBeta(x = 0.1, shape1 = c(1, 1), shape2 = c(2, 2))

x <- matrix(c(0.1, 0.1, 0.2, 0.3, 0.5, 0.5), ncol = 2, byrow = TRUE)
dnsBeta(x, shape1 = c(1, 2), shape2 = c(2, 3))


pnsBeta(q = c(0.1, 0.1), shape1 = c(3, 1), shape2 = c(1, 3), lower.tail = FALSE)
pnsBeta(x, shape1 = c(1, 2), shape2 = c(2, 2))


qnsBeta(p = 0.1, shape1 = c(1, 1), shape2 = c(2, 2))
qnsBeta(p = c(0.25, 0.5, 0.75), shape1 = c(1, 2), shape2 = c(2, 2))

# Simulate 10 numbers
rnsBeta(n = 10, shape1 = c(1, 2), shape2 = c(1, 1))
```

`Neutrosophic Binomial` *Neutrosophic Binomial Distribution*

Description

Density, distribution function, quantile function and random generation for the neutrosophic binomial distribution with parameters `size` = n and `prob` = p_N.

Usage

```
dnsBinom(x, size, prob)

pnsBinom(q, size, prob, lower.tail = TRUE)

qnsBinom(p, size, prob)

rnsBinom(n, size, prob)
```

Arguments

`x`	a vector or matrix of observations for which the pdf needs to be computed.
`size`	number of trials (zero or more), which must be a positive interval.
`prob`	probability of success on each trial, $0 \leq prob \leq 1$.

`q`	a vector or matrix of quantiles for which the cdf needs to be computed.
`lower.tail`	logical; if TRUE (default), probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
`p`	a vector or matrix of probabilities for which the quantile needs to be computed.
`n`	number of random values to be generated.

Details

The neutrosophic binomial distribution with parameters n and p_N has the density

$$f_X(x) = \binom{n}{x} p_N^x \left(1 - p_N\right)^{n-x}$$

for $n \in \{1, 2, \ldots\}$ and $p_N \in (p_L, p_U)$ which must be $0 < p_N < 1$ and $x \in \{0, 1, 2, \ldots, n\}$.

Value

`dnsBinom` gives the probability mass function

`pnsBinom` gives the distribution function

`qnsBinom` gives the quantile function

`rnsBinom` generates random variables from the Binomial Distribution.

References

Granados, C. (2022). Some discrete neutrosophic distributions with neutrosophic parameters based on neutrosophic random variables. *Hacettepe Journal of Mathematics and Statistics*, 51(5), 1442-1457.

Examples

```
# Probability of X = 17 when X follows bin(n = 20, p = [0.9,0.8])
dnsBinom(x = 17, size = 20, prob = c(0.9, 0.8))

x <- matrix(c(15, 15, 17, 18, 19, 19), ncol = 2, byrow = TRUE)
dnsBinom(x = x, size = 20, prob = c(0.8, 0.9))


pnsBinom(q = 17, size = 20, prob = c(0.9, 0.8))
pnsBinom(q = c(17, 18), size = 20, prob = c(0.9, 0.8))
pnsBinom(q = x, size = 20, prob = c(0.9, 0.8))

qnsBinom(p = 0.5, size = 20, prob = c(0.8, 0.9))
qnsBinom(p = c(0.25, 0.5, 0.75), size = 20, prob = c(0.8, 0.9))


# Simulate 10 numbers
rnsBinom(n = 10, size = 20, prob = c(0.8, 0.9))
```

Neutrosophic Discrete Uniform

Neutrosophic Discrete Uniform Distribution

Description

Density, distribution function, quantile function and random generation for the neutrosophic discrete uniform distribution with parameter k_N.

Usage

```
dnsDiscUnif(x, k)

pnsDiscUnif(q, k, lower.tail = TRUE)

qnsDiscUnif(p, k)

rnsDiscUnif(n, k)
```

Arguments

`x`	a vector or matrix of observations for which the pdf needs to be computed.
`k`	parameter of the distribution that must be a positive finite interval.
`q`	a vector or matrix of quantiles for which the cdf needs to be computed.
`lower.tail`	logical; if TRUE (default), probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
`p`	a vector or matrix of probabilities for which the quantile needs to be computed.
`n`	number of random values to be generated.

Details

Let X_N be a neutrosophic random variable and denote $X_N \sim \mathcal{DU}(1, 2, \ldots, k_N)$ as neutrosophic discrete uniform distribution with parameter k_N has the density

$$f_N(x) = \frac{1}{k_N}$$

for $k_N \in (k_L, k_U)$.

Value

`dnsDiscUnif` gives the probability mass function,

`pnsDiscUnif` gives the distribution function

`qnsDiscUnif` gives the quantile function

`rnsDiscUnif` generates random variables from the neutrosophic Discrete Uniform Distribution.

References

Granados, C. (2022). Some discrete neutrosophic distributions with neutrosophic parameters based on neutrosophic random variables. *Hacettepe Journal of Mathematics and Statistics*, 51(5), 1442-1457.

Examples

```
dnsDiscUnif(x = 8, k = c(10, 11))
dnsDiscUnif(x = c(8, 9), k = c(10, 11))

pnsDiscUnif(q = 2, k = c(10, 11))

qnsDiscUnif(p = 0.2, k = c(10, 11))


# Simulate 10 numbers
rnsDiscUnif(n = 10, k = c(10, 11))
```

`Neutrosophic Exponential` *Neutrosophic Exponential Distribution*

Description

Density, distribution function, quantile function and random generation for the neutrosophic exponential distribution with the parameter `rate` = θ_N.

Usage

```
dnsExp(x, rate)

pnsExp(q, rate, lower.tail = TRUE)

qnsExp(p, rate)

rnsExp(n, rate)
```

Arguments

`x`	a vector or matrix of observations for which the pdf needs to be computed.
`rate`	the shape parameter, which must be a positive interval.
`q`	a vector or matrix of quantiles for which the cdf needs to be computed.
`lower.tail`	logical; if TRUE (default), probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
`p`	a vector or matrix of probabilities for which the quantile needs to be computed.
`n`	number of random values to be generated.

Details

The neutrosophic exponential distribution with parameter θ_N has density

$$f_N(x) = \theta_N \exp\left(-x\theta_N\right)$$

for $x \geq 0$ and $\theta_N \in (\theta_L, \theta_U)$, the rate parameter must be a positive interval and $x \geq 0$.

Value

`dnsExp` gives the density function

`pnsExp` gives the distribution function

`qnsExp` gives the quantile function

`rnsExp` generates random values from the neutrosophic exponential distribution.

References

Duan, W., Q., Khan, Z., Gulistan, M., Khurshid, A. (2021). Neutrosophic Exponential Distribution: Modeling and Applications for Complex Data Analysis, *Complexity*, 2021, 1-8.

Examples

```
# Example 4 of Duan et al. (2021)

data <- matrix(c(4, 4, 3.5, 3.5, 3.9, 4.1, 4.2, 4.2, 4.3, 4.6, 4.7, 4.7),
 nrow = 6, ncol = 2, byrow = TRUE)

dnsExp(data, rate = c(0.23, 0.24))
dnsExp(x = c(4, 4.1), rate = c(0.23, 0.24))

dnsExp(4, rate = c(0.23, 0.23))

# The cumulative distribution function for the nuetrosophic observation (4,4.1)
pnsExp(c(4, 4.1), rate = c(0.23, 0.24), lower.tail = TRUE)

pnsExp(4, rate = c(0.23, 0.24))
# The first percentile
qnsExp(p = 0.1, rate = 0.25)

# The quantiles
qnsExp(p = c(0.25, 0.5, 0.75), rate = c(0.24, 0.25))

# Simulate 10 numbers
rnsExp(n = 10, rate = c(0.23, 0.24))
```

Neutrosophic Gamma *Neutrosophic Gamma Distribution*

Description

Density, distribution function, quantile function and random generation for the neutrosophic gamma distribution with parameter shape = α_N and scale=λ_N.

Usage

```
dnsGamma(x, shape, scale)

pnsGamma(q, shape, scale, lower.tail = TRUE)

qnsGamma(p, shape, scale)

rnsGamma(n, shape, scale)
```

Arguments

x	a vector or matrix of observations for which the pdf needs to be computed.
shape	the shape parameter, which must be a positive interval.
scale	the scale parameter, which must be a positive interval.
q	a vector or matrix of quantiles for which the cdf needs to be computed.
lower.tail	logical; if TRUE (default), probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
p	a vector or matrix of probabilities for which the quantile needs to be computed.
n	number of random values to be generated.

Details

The neutrosophic gamma distribution with parameters α_N and λ_N has density

$$f_N(x) = \frac{1}{\Gamma(\alpha_N)\lambda_N^{\alpha_N}} x^{\alpha_N - 1} \exp\{-(x/\lambda_N)\}$$

for $x \geq 0$, $\alpha_N \in (\alpha_L, \alpha_U)$, the shape parameter which must be a positive interval and $\lambda_N \in (\lambda_L, \lambda_U)$, the scale parameter which must be a positive interval. Here, $\Gamma(\cdot)$ is gamma function implemented by `gamma`.

Value

`dnsGamma` gives the density function

`pnsGamma` gives the distribution function

`qnsGamma` gives the quantile function

`rnsGamma` generates random variables from the neutrosophic gamma distribution.

References

Khan, Z., Al-Bossly, A., Almazah, M. M. A., and Alduais, F. S. (2021). On statistical development of neutrosophic gamma distribution with applications to complex data analysis, *Complexity*, 2021, Article ID 3701236.

Examples

```
data(remission)
dnsGamma(x = remission, shape = c(1.1884, 1.1896), scale = c(7.6658, 7.7796))

pnsGamma(q = 20, shape = c(1.1884, 1.1896), scale = c(7.6658, 7.7796))

# Calculate quantiles
qnsGamma(p = c(0.25, 0.5, 0.75), shape = c(1.1884, 1.1896), scale = c(7.6658, 7.7796))

# Simulate 10 numbers
rnsGamma(n = 10, shape = c(1.1884, 1.1896), scale = c(7.6658, 7.7796))
```

`Neutrosophic Generalized Exponential`

Neutrosophic Generalized Exponential Distribution

Description

Density, distribution function, quantile function and random generation for the neutrosophic generalized exponential distribution with shape parameter δ_N and scale parameter ν_N.

Usage

```
dnsGenExp(x, nu, delta)

pnsGenExp(q, nu, delta, lower.tail = TRUE)

qnsGenExp(p, nu, delta)

rnsGenExp(n, nu, delta)
```

Arguments

`x`	a vector or matrix of observations for which the pdf needs to be computed.
`nu`	the scale parameter, which must be a positive interval.
`delta`	the shape parameter, which must be a positive interval.
`q`	a vector or matrix of quantiles for which the cdf needs to be computed.
`lower.tail`	logical; if TRUE (default), probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
`p`	a vector or matrix of probabilities for which the quantile needs to be computed.
`n`	number of random values to be generated.

Details

The neutrosophic generalized exponential distribution with parameters δ_N and ν_N has density

$$f_N(x) = \frac{\delta_N}{\nu_N}\left(1 - \exp\left\{-\frac{x_N}{\nu_N}\right\}\right)^{\delta_N - 1} \exp\left\{-\frac{x_N}{\nu_N}\right\}$$

for $\delta_N \in (\delta_L, \delta_U)$, the shape parameter which must be a positive interval, and $\nu_N \in (\nu_L, \nu_U)$, the scale parameter which must also be a positive interval, and $x \geq 0$.

Value

`dnsGenExp` gives the density function

`pnsGenExp` gives the distribution function

`qnsGenExp` gives the quantile function

`rnsGenExp` generates random variables from the neutrosophic generalized exponential distribution.

References

Rao, G. S., Norouzirad, M., and Mazarei . D. (2023). Neutrosophic Generalized Exponential Distribution with Application. *Neutrosophic Sets and Systems*, 55, 471-485.

Examples

```
data(remission)
dnsGenExp(x = remission, nu = c(7.9506, 8.0568), delta = c(1.2390, 1.2397))

pnsGenExp(q = 20, nu = c(7.9506, 8.0568), delta = c(1.2390, 1.2397))


# Calcluate quantiles
qnsGenExp(c(0.25, 0.5, 0.75), nu = c(7.9506, 8.0568), delta = c(1.2390, 1.2397))

# Simulate 10 values
rnsGenExp(n = 10, nu = c(7.9506, 8.0568), delta = c(1.2390, 1.2397))
```

`Neutrosophic Generalized Pareto`

Neutrosophic Generalized Pareto Distribution

Description

Density, distribution function, quantile function and random generation for the neutrosophic generalized pareto distribution with parameters `shape` = α_N and `scale`=β_N.

Usage

```
dnsGenPareto(x, shape, scale)

pnsGenPareto(q, shape, scale, lower.tail = TRUE)

qnsGenPareto(p, shape, scale)

rnsGenPareto(n, shape, scale)
```

Arguments

`x`	a vector or matrix of observations for which the pdf needs to be computed.
`shape`	the shape parameter, which must be a positive interval.
`scale`	the scale parameter, which must be a positive interval.
`q`	a vector or matrix of quantiles for which the cdf needs to be computed.
`lower.tail`	logical; if TRUE (default), probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
`p`	a vector or matrix of probabilities for which the quantile needs to be computed.
`n`	number of random values to be generated.

Details

The neutrosophic generalized pareto distribution with parameters α_N and β_N has density

$$f_N(x) = \frac{1}{\beta_N}\left(1 + \frac{\alpha_N x_N}{\beta_N}\right)^{-\frac{1}{\alpha_N}-1}$$

for $x \geq 0$, $\alpha_N \in (\alpha_L, \alpha_U)$, the shape parameter which must be a positive interval and $\beta_N \in (\beta_L, \beta_U)$, the scale parameter which must be a positive interval.

Value

`dnsGenPareto` gives the density function

`pnsGenPareto` gives the distribution function

`qnsGenPareto` gives the quantile function

`rnsGenPareto` generates random variables from the neutrosophic generalized pareto distribution.

References

Eassa, N. I., Zaher, H. M., & El-Magd, N. A. A. (2023). Neutrosophic Generalized Pareto Distribution, *Mathematics and Statistics*, 11(5), 827–833.

Examples

```
data(remission)
dnsGenPareto(x = remission, shape = c(1.1884, 1.1896), scale = c(7.6658, 7.7796))

pnsGenPareto(q = 20, shape = c(1.1884, 1.1896), scale = c(7.6658, 7.7796))

# Calculate quantiles
qnsGenPareto(p = c(0.25, 0.5, 0.75), shape = c(1.1884, 1.1896), scale = c(7.6658, 7.7796))

# Simulate 10 numbers
rnsGenPareto(n = 10, shape = c(1.1884, 1.1896), scale = c(7.6658, 7.7796))
```

`Neutrosophic Generalized Rayleigh` *Neutrosophic Generalized Rayleigh Distribution*

Description

Density, distribution function, quantile function and random generation for the neutrosophic generalized Rayleigh distribution with parameters `shape` = ν_N and `scale` = σ_N.

Usage

```
dnsGenRayleigh(x, shape, scale)

pnsGenRayleigh(q, shape, scale, lower.tail = TRUE)

qnsGenRayleigh(p, shape, scale)

rnsGenRayleigh(n, shape, scale)
```

Arguments

`x`	a vector or matrix of observations for which the pdf needs to be computed.
`shape`	the shape parameter, which must be a positive interval.
`scale`	the scale parameter, which must be a positive interval.
`q`	a vector or matrix of quantiles for which the cdf needs to be computed.
`lower.tail`	logical; if TRUE (default), probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
`p`	a vector or matrix of probabilities for which the quantile needs to be computed.
`n`	number of random values to be generated.

Details

The neutrosophic generalized Rayleigh distribution with parameters ν_N and σ_N has the density

$$f_N(x) = \frac{2\nu_N}{\sigma_N^2} x \exp\{-\left(\frac{x}{\sigma_N}\right)^2\} \left[1 - \exp\{-\left(\frac{x}{\sigma_N}\right)^2\}\right]^{\nu_N - 1}$$

for $x > 0$, $\nu_N \in (\nu_L, \nu_U)$, the shape parameter which must be a positive interval and $\sigma_N \in (\sigma_L, \sigma_U)$, the scale parameter which must be a positive interval.

Value

`dnsGenRayleigh` gives the density function

`pnsGenRayleigh` gives the distribution function

`qnsGenRayleigh` gives the quantile function

`rnsGenRayleigh` generates random variables from the Neutrosophic Generalized Rayleigh Distribution.

References

Norouzirad, M., Rao, G. S., & Mazarei, D. (2023). Neutrosophic Generalized Rayleigh Distribution with Application. *Neutrosophic Sets and Systems*, 58(1), 250-262.

Examples

```
data(remission)
dnsGenRayleigh(x = remission,shape = c(1.1884, 1.1896), scale = c(7.6658, 7.7796))

pnsGenRayleigh(q = 20, shape = c(1.1884, 1.1896), scale = c(7.6658, 7.7796))

# Calculate quantiles
qnsGenRayleigh(p = c(0.25, 0.5, 0.75), shape = c(1.1884, 1.1896), scale = c(7.6658, 7.7796))

# Simulate 10 values
rnsGenRayleigh(n = 10, shape = c(1.1884, 1.1896), scale = c(7.6658, 7.7796))
```

`Neutrosophic Geometric`

Neutrosophic Geometric Distribution

Description

Density, distribution function, quantile function and random generation for the neutrosophic Geometric distribution with parameter `prob` = p_N.

Usage

```
dnsGeom(x, prob)

pnsGeom(q, prob, lower.tail = TRUE)

qnsGeom(p, prob)

rnsGeom(n, prob)
```

Arguments

`x`	a vector or matrix of observations for which the pdf needs to be computed.
`prob`	probability of success on each trial, `prob`$\in (0, 1)$.
`q`	a vector or matrix of quantiles for which the cdf needs to be computed.
`lower.tail`	logical; if TRUE (default), probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
`p`	a vector or matrix of probabilities for which the quantile needs to be computed.
`n`	number of random values to be generated.

Details

The neutrosophic Geometric distribution with parameter p_N has the density

$$f_X(x) = p_N\,(1 - p_N)^x$$

for $p_N \in (p_L, p_U)$ which must be $0 < p_N < 1$ and $x \in \{0, 1, 2, \ldots\}$.

Value

`dnsGeom` gives the probability mass function

`pnsGeom` gives the distribution function

`qnsGeom` gives the quantile function

`rnsGeom` generates random variables from the Geometric Distribution.

References

Granados, C. (2022). Some discrete neutrosophic distributions with neutrosophic parameters based on neutrosophic random variables. *Hacettepe Journal of Mathematics and Statistics*, 51(5), 1442-1457.

Examples

```
# One person participates each week with a ticket in a lottery game, where
# the probability of winning the first prize is (10^(-8), 10^(-6)).
# Probability of one persons wins at the fifth year?

dnsGeom(x = 5, prob = c(1e-8, 1e-6))

# Probability of one persons wins after 10 years?
```

```
pnsGeom(q = 10, prob = c(1e-8, 1e-6))
pnsGeom(q = 10, prob = c(1e-8, 1e-6), lower.tail = FALSE)
# Calculate the quantiles
qnsGeom(p = c(0.25, 0.5, 0.75), prob = c(1e-8, 1e-6))
# Simulate 10 numbers
rnsGeom(n = 10, prob = c(1e-8, 1e-6))
```

Neutrosophic Kumaraswamy

Neutrosophic Kumaraswamy Distribution

Description

Density, distribution function, quantile function and random generation for the neutrosophic Kumaraswamy distribution with shape parameters α_N and β_N.

Usage

```
dnsKumaraswamy(x, shape1, shape2)

pnsKumaraswamy(q, shape1, shape2, lower.tail = TRUE)

qnsKumaraswamy(p, shape1, shape2)

rnsKumaraswamy(n, shape1, shape2)
```

Arguments

x	a vector or matrix of observations for which the pdf needs to be computed.
shape1	the shape parameter, which must be a positive interval.
shape2	the shape parameter, which must be a positive interval.
q	a vector or matrix of quantiles for which the cdf needs to be computed.
lower.tail	logical; if TRUE (default), probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
p	a vector or matrix of probabilities for which the quantile needs to be computed.
n	number of random values to be genelambdad.

Details

The neutrosophic Kumaraswamy distribution with parameters α_N and β_N has density

$$f_N(x) = \alpha_N \beta_N x^{\alpha_N - 1}(1 - x^{\alpha_N})^{\beta_N - 1}$$

for $0 \leq x \leq 1$, $\alpha_N \in (\alpha_L, \alpha_U)$ and $\beta_N \in (\beta_L, \beta_U)$ are shape parameters.

Value

`pnsKumaraswamy` gives the distribution function

`dnsKumaraswamy` gives the density

`qnsKumaraswamy` gives the quantile function

`rnsKumaraswamy` generates random values from the neutrosophic Kumaraswamy distribution.

References

Ahsan-ul-Haq, M. (2022). Neutrosophic Kumaraswamy Distribution with Engineering Application, *Neutrosophic Sets and Systems*, 49, 269-276.

Examples

```
dnsKumaraswamy(x = c(0.5, 0.1), shape1 = c(0.23, 0.24), shape2 = c(1, 2))
dnsKumaraswamy(0.5, shape1 = c(0.23, 0.24), shape2 = c(1, 2))


# The cumulative distribution function for the nuetrosophic observation (4,4.1)
pnsKumaraswamy(q = c(.8, .1), shape1 = c(0.23, 0.24), shape2 = c(1, 2))
# The first percentile
qnsKumaraswamy(p = 0.1, shape1 = 0.24, shape2 = 2)

# The quantiles
qnsKumaraswamy(p = c(0.25, 0.5, 0.75), shape1 = c(0.23, 0.24), shape2 = c(1, 2))

# Simulate 10 numbers
rnsKumaraswamy(n = 10, shape1 = c(0.23, 0.24), shape2 = c(1, 2))
```

`Neutrosophic Laplace` *Neutrosophic Laplace (Double Exponential) Distribution*

Description

Density, distribution function, quantile function, and random generation for the neutrosophic Laplace (Double Exponential) distribution with parameters `location` = θ_N and `scale` = β_N.

Usage

```
dnsLaplace(x, location, scale)

pnsLaplace(q, location, scale, lower.tail = TRUE)

qnsLaplace(p, location, scale)

rnsLaplace(n, location, scale)
```

Arguments

`x`	a vector or matrix of observations for which the pdf needs to be computed.
`location`	the location parameter, which is the mean.
`scale`	the scale parameter, Must consist of positive values.
`q`	a vector or matrix of quantiles for which the cdf needs to be computed.
`lower.tail`	logical; if TRUE (default), probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
`p`	a vector or matrix of probabilities for which the quantile needs to be computed.
`n`	number of random values to be generated.

Details

The neutrosophic Laplace distribution with parameters θ_N and β_N has density

$$f_N(x) = \frac{1}{2\beta_N} \exp\left\{-\frac{|x-\theta_N|}{\beta_N}\right\}$$

for $-\infty < x < \infty$, $\theta_N \in (\theta_L, \theta_U)$, the location parameter, $\beta_N \in (\beta_L, \beta_U)$, the scale parameter which be a positive interval.

Value

`dnsLaplace` gives the density function

`pnsLaplace` gives the distribution function

`qnsLaplace` gives the quantile function

`rnsLaplace` generates random values from the neutrosophic Laplace distribution.

References

Rahul, T., Malik, S. C., Raj, M. (2023). Neutrosophic Laplace Distribution with Application in Financial Data Analysis, *Neutrosophic Sets and Systems*, 57(1), 224-233.

Examples

```
dnsLaplace(x = c(4, 4.1), location = c(0.23, 0.24), scale = c(1, 2))
dnsLaplace(4, location = c(0.23, 0.24), scale = c(1, 2))


# The cumulative distribution function for the neutrosophic observation (4,4.1)
pnsLaplace(q = c(4, 4.1), location = c(0.23, 0.24), scale = c(1, 2))
# The first percentile
qnsLaplace(p = 0.1, location = 0.24, scale = 2)

# The quantiles
qnsLaplace(p = c(0.25, 0.5, 0.75), location = c(0.23, 0.24), scale = c(1, 2))

# Simulate 10 numbers
rnsLaplace(n = 10, location = c(0.23, 0.24), scale = c(1, 2))
```

`Neutrosophic Negative Binomial` *Neutrosophic Negative Binomial Distribution*

Description

Density, distribution function, quantile function and random generation for the neutrosophic Negative Binomial distribution with parameters `size` = r_N and `prob` = p_N.

Usage

```
dnsNegBinom(x, size, prob)

pnsNegBinom(q, size, prob, lower.tail = TRUE)

qnsNegBinom(p, size, prob)

rnsNegBinom(n, size, prob)
```

Arguments

`x`	a vector or matrix of observations for which the pdf needs to be computed.
`size`	number of trials (zero or more), which must be a positive interval.
`prob`	probability of success on each trial, $0 <$ `prob` < 1.
`q`	a vector or matrix of quantiles for which the cdf needs to be computed.
`lower.tail`	logical; if TRUE (default), probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
`p`	a vector or matrix of probabilities for which the quantile needs to be computed.
`n`	number of random values to be generated.

Details

The neutrosophic negative binomial distribution with parameters r_N and p_N has the density

$$\left(\begin{array}{c} r_N + x - 1 \\ x \end{array} \right) p_N^{r_N} \, (1 - p_N)^x$$

for $r_N \in \{1, 2, \ldots\}$ and $p_N \in (p_L, p_U)$ which must be $0 < p_N < 1$ and $x \in \{0, 1, 2, \ldots\}$.

Value

`dnsNegBinom` gives the probability mass function

`pnsNegBinom` gives the distribution function

`qnsNegBinom` gives the quantile function

`rnsNegBinom` generates random variables from the Negative Binomial Distribution.

References

Granados, C. (2022). Some discrete neutrosophic distributions with neutrosophic parameters based on neutrosophic random variables. *Hacettepe Journal of Mathematics and Statistics*, 51(5), 1442-1457.

Examples

```
dnsNegBinom(x = 1, size = 2, prob = c(0.5, 0.6))
pnsNegBinom(q = 1, size = 2, prob = c(0.5, 0.6))
qnsNegBinom(p = c(0.25, 0.5, 0.75), size = 2, prob = c(0.5, 0.6))
rnsNegBinom(n = 10, size = 2, prob = c(0.6, 0.6))
```

`Neutrosophic Normal` *Neutrosophic Normal Distribution*

Description

Density, distribution function, quantile function and random generation for the neutrosophic generalized exponential distribution with parameters `mean` = μ_N and standard deviation `sd` = σ_N.

Usage

```
dnsNorm(x, mean, sd)

pnsNorm(q, mean, sd, lower.tail = TRUE)

qnsNorm(p, mean, sd)

rnsNorm(n, mean, sd)
```

Arguments

`x`	a vector or matrix of observations for which the pdf needs to be computed.
`mean`	the mean, which must be an interval.
`sd`	the standard deviations that must be positive.
`q`	a vector or matrix of quantiles for which the cdf needs to be computed.
`lower.tail`	logical; if TRUE (default), probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
`p`	a vector or matrix of probabilities for which the quantile needs to be computed.
`n`	number of random values to be generated.

Details

The neutrosophic normal distribution with parameters mean μ_N and standard deviation σ_N has density function

$$f_N(x) = \frac{1}{\sigma_N \sqrt{2\pi}} \exp\{\left(\frac{(X - \mu_N)^2}{2\sigma_N^2}\right)$$

} for $\mu_N \in (\mu_L, \mu_U)$, the mean which must be an interval, and $\sigma_N \in (\sigma_L, \sigma_U)$, the standard deviation which must also be a positive interval, and $-\infty < x < \infty$.

Value

`dnsNorm` gives the density function

`pnsNorm` gives the distribution function

`qnsNorm` gives the quantile function

`rnsNorm` generates random variables from the neutrosophic normal distribution.

References

Patro, S. and Smarandache, F. (2016). The Neutrosophic Statistical Distribution, More Problems, More Solutions. Infinite Study.

Examples

```
data(balls)
dnsNorm(x = balls, mean = c(72.14087, 72.94087), sd = c(37.44544, 37.29067))

pnsNorm(q = 5, mean = c(72.14087, 72.94087), sd = c(37.44544, 37.29067))

# Calculate quantiles
qnsNorm(p = c(0.25, 0.5, 0.75), mean = c(9.1196, 9.2453), sd = c(10.1397, 10.4577))

# Simulate 10 values
rnsNorm(n = 10, mean = c(4.141, 4.180), sd = c(0.513, 0.521))
```

`Neutrosophic Poisson` *Neutrosophic Poisson Distribution*

Description

Density, distribution function, quantile function and random generation for the neutrosophic Poisson distribution with parameter λ_N.

Usage

```
dnsPois(x, lambda)

pnsPois(q, lambda, lower.tail = TRUE)

qnsPois(p, lambda)

rnsPois(n, lambda)
```

Arguments

`x`	a vector or matrix of observations for which the pdf needs to be computed.
`lambda`	the mean, which must be a positive interval.
`q`	a vector or matrix of quantiles for which the cdf needs to be computed.
`lower.tail`	logical; if TRUE (default), probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
`p`	a vector or matrix of probabilities for which the quantile needs to be computed.
`n`	number of random values to be generated.

Details

The neutrosophic Poisson distribution with parameter λ_N has the density

$$f_N(x) = \exp\{-\lambda_N\} \frac{(\lambda_N)^x}{x!}$$

for $\lambda_N \in (\lambda_L, \lambda_U)$ which must be a positive interval and $x \in \{0, 1, 2, \ldots\}$.

Value

`dnsPois` gives the probability mass function

`pnsPois` gives the distribution function

`qnsPois` gives the quantile function

`rnsPois` generates random variables from the neutrosophic Poisson Distribution.

References

Alhabib, R., Ranna, M. M., Farah, H., Salama, A. A. (2018). Some neutrosophic probability distributions. *Neutrosophic Sets and Systems*, 22, 30-38.

Examples

```
# In a company, Phone employee receives phone calls, the calls arrive with
# rate of [1 , 3] calls per minute, we will calculate
# the probability that the employee will not receive any call within a minute
dnsPois(x = 0, lambda = c(1, 3))

# the probability that employee would not receive any call within 5 minutes
dnsPois(x = 0, lambda = c(5, 15))
```

```
# the probability that the employee will receive at least one call within a minute
pnsPois(q = 1, lambda = c(1, 3), lower.tail = FALSE)
# the probability that the employee will receive at most three calls within 5 minutes
pnsPois(q = 3, lambda = c(5, 15), lower.tail = TRUE)
# Calcaute the quantiles
qnsPois(p = c(0.25, 0.5, 0.75), lambda = c(1, 3))
# Simulate 10 values
rnsPois(n = 10, lambda = 1)
```

`Neutrosophic Rayleigh` *Neutrosophic Rayleigh Distribution*

Description

Density, distribution function, quantile function and random generation for the neutrosophic Rayleigh distribution with parameter θ_N.

Usage

```
dnsRayleigh(x, theta)

pnsRayleigh(q, theta, lower.tail = TRUE)

qnsRayleigh(p, theta)

rnsRayleigh(n, theta)
```

Arguments

`x`	a vector or matrix of observations for which the pdf needs to be computed.
`theta`	the shape parameter, which must be a positive interval.
`q`	a vector or matrix of quantiles for which the cdf needs to be computed.
`lower.tail`	logical; if TRUE (default), probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
`p`	a vector or matrix of probabilities for which the quantile needs to be computed.
`n`	number of random values to be generated.

Details

The neutrosophic Rayleigh distribution with parameter θ_N has the density

$$f_N(x) = \frac{x}{\theta_N^2} \exp\{-\frac{1}{2}\left(\frac{x}{\theta_N}\right)^2\}$$

for $\theta_N \in (\theta_L, \theta_U)$, which must be a positive interval and $x \geq 0$.

Value

dnsRayleigh gives the density function

pnsRayleigh gives the distribution function

qnsRayleigh gives the quantile function

rnsRayleigh generates random variables from the Neutrosophic Rayleigh Distribution.

References

Khan, Z., Gulistan, M., Kausar, N. and Park, C. (2021). Neutrosophic Rayleigh Model With Some Basic Characteristics and Engineering Applications, in *IEEE Access*, 9, 71277-71283.

Examples

```
data(remission)
dnsRayleigh(x = remission, theta = c(9.6432, 9.8702))

pnsRayleigh(q = 20, theta = c(9.6432, 9.8702))

# Calculate quantiles
qnsRayleigh(p = c(0.25, 0.5, 0.75), theta = c(9.6432, 9.8702))

# Simulate 10 values
rnsRayleigh(n = 10, theta = c(9.6432, 9.8702))
```

Neutrosophic Uniform *Neutrosophic Uniform Distribution*

Description

Density, distribution function, quantile function and random generation for the neutrosophic Uniform distribution of a continuous variable X with parameters a_N and b_N.

Usage

```
dnsUnif(x, min, max)

pnsUnif(q, min, max, lower.tail = TRUE)

qnsUnif(p, min, max)

rnsUnif(n, min, max)
```

Arguments

`x`	a vector or matrix of observations for which the pdf needs to be computed.
`min`	lower limits of the distribution. Must be finite.
`max`	upper limits of the distribution. Must be finite.
`q`	a vector or matrix of quantiles for which the cdf needs to be computed.
`lower.tail`	logical; if TRUE (default), probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
`p`	a vector or matrix of probabilities for which the quantile needs to be computed.
`n`	number of random values to be generated.

Details

The neutrosophic Uniform distribution with parameters a_N and b_N has the density

$$f_N(x) = \frac{1}{b_N - a_N}$$

for $a_N \in (a_L, a_U)$ lower parameter interval, $b_N \in (b_L, b_U)$, upper parameter interval.

Value

`dnsUnif` gives the density function

`pnsUnif` gives the distribution function

`qnsUnif` gives the quantile function

`rnsUnif` generates random variables from the neutrosophic Uniform Distribution.

References

Alhabib, R., Ranna, M. M., Farah, H., & Salama, A. A. (2018). Some neutrosophic probability distributions, *Neutrosophic Sets and Systems*, 22, 30-38.

Examples

```
dnsUnif(x = 1, min = c(0, 5), max = c(15, 20))
dnsUnif(x = c(6, 10), min = c(0, 5), max = c(15, 20))

punif(q = 1, min = c(0, 5), max = c(15, 20))
punif(q = c(6, 10), min = c(0, 5), max = c(15, 20))

qnsUnif(p = c(0.25, 0.5, 0.75), min = c(0, 5), max = c(15, 20))

rnsUnif(n = 10, min = c(0, 5), max = c(15, 20))
```

`Neutrosophic Weibull` *Neutrosophic Weibull Distribution*

Description

Density, distribution function, quantile function and random generation for the neutrosophic Weibull distribution with `scale` parameter α_N and `shape` parameter β_N.

Usage

```
dnsWeibull(x, shape, scale)

pnsWeibull(q, shape, scale, lower.tail = TRUE)

qnsWeibull(p, shape, scale)

rnsWeibull(n, shape, scale)
```

Arguments

`x`	a vector or matrix of observations for which the pdf needs to be computed.
`shape`	shape parameter, which must be a positive interval.
`scale`	scale parameter, which must be a positive interval.
`q`	a vector or matrix of quantiles for which the cdf needs to be computed.
`lower.tail`	logical; if TRUE (default), probabilities are $P(X \leq x)$; otherwise, $P(X > x)$.
`p`	a vector or matrix of probabilities for which the quantile needs to be computed.
`n`	number of random values to be generated.

Details

The neutrosophic Rayleigh distribution with parameters α_N and β_N has the density

$$f_N(x) = \frac{\beta_N}{\alpha_N^{\beta_N}} x^{\beta_N - 1} \exp\{-\left(x/\alpha_N\right)^{\beta_N}\}$$

for $\beta_N \in (\beta_L, \beta_U)$ the shape parameter must be a positive interval, $\alpha_N \in (\alpha_L, \alpha_U)$, the scale parameter which be a positive interval, and $x > 0$.

Value

`dnsWeibull` gives the density function

`pnsWeibull` gives the distribution function

`qnsWeibull` gives the quantile function

`rnsWeibull` generates random variables from the neutrosophic Weibull dDistribution.

References

Alhasan, K. F. H. and Smarandache, F. (2019). Neutrosophic Weibull distribution and Neutrosophic Family Weibull Distribution, *Neutrosophic Sets and Systems*, 28, 191-199.

Examples

```
data(remission)
dnsWeibull(x = remission, shape = c(1.0519, 1.0553), scale = c(9.3370, 9.4544))

pnsWeibull(q = 20, shape = c(1.0519, 1.0553), scale = c(9.3370, 9.4544))

# Calculate quantiles
qnsWeibull(p = c(0.25, 0.5, 0.75), shape = c(1.0519, 1.0553), scale = c(9.3370, 9.4544))

# Simulate 10 numbers
rnsWeibull(n = 10, shape = c(1.0519, 1.0553), scale = c(9.3370, 9.4544))
```

---

remission *Remission data*

Description

It is related to remission time in months of 128 cancer patients.

Format

A data.frame with 128 observations of remission time in months of cancer patients.

Source

Lee, E.T. and Wang, J. (2003), Statistical Methods for Survival Data Analysis. Vol. 476, John Wiley & Sons,Hoboken, NJ, USA.

Rao, G. S., Norouzirad, M., and Mazarei . D. (2023). Neutrosophic Generalized Exponential Distribution with Application. *Neutrosophic Sets and Systems*, 55, 471-485.

Examples

```
data("remission")
remission
```

Index