Package 'mhn'

October 23, 2026

Type Package

Title The Modified Half-Normal Distribution

Version 0.1.2

Description Provides density, distribution, quantile, and random generation functions for the Modified Half-Normal (MHN) distribution, along with moments and mode; the Fox-Wright Psi normalizing constant is evaluated internally. The MHN distribution arises as a conditional posterior in Bayesian Markov chain Monte Carlo and generalizes the half-normal, truncated normal, and square-root gamma distributions. Implements efficient sampling via Algorithms 1 and 3 of Sun, Kong & Pal (2023) <doi:10.1080/03610926.2021.1934700>, the relaxed transformed density rejection method of Gao & Wang (2025) <doi:10.1080/03610918.2025.2524551>, and, for the truncated normal special case, the sampler of Robert (1995) <doi:10.1007/BF00143942>.

License MIT + file LICENSE

RoxygenNote 7.3.3

Depends R (>= 4.0)

Imports Rcpp (>= 1.0.13)

LinkingTo Rcpp (>= 1.0.13), BH (>= 1.78.0-0)

SystemRequirements C++17

Suggests testthat (>= 3.1.5), knitr, rmarkdown, bench, Rmpfr

Config/testthat/edition 3

VignetteBuilder knitr, rmarkdown

URL https://github.com/t-momozaki/mhn, https://t-momozaki.github.io/mhn/

BugReports https://github.com/t-momozaki/mhn/issues

NeedsCompilation yes

Author Tomotaka Momozaki [aut, cre] (ORCID: <https://orcid.org/0000-0001-6735-0804>)

Maintainer Tomotaka Momozaki <momozaki.stat@gmail.com>

Repository CRAN

Date/Publication 2026-10-23 06:10:02 UTC

Contents

dmhn 2
mhn_kurtosis 4
mhn_mean 6
mhn_mode 7
mhn_skewness 8
mhn_var 9
pmhn 11
qmhn 13
rmhn 16

Index **20**

`dmhn` *Density of the Modified Half-Normal Distribution*

Description

Computes the probability density function (or log-density) of the Modified Half-Normal (MHN) distribution with parameters `alpha`, `beta`, and `gamma`.

Usage

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

Arguments

`x`	Numeric vector of evaluation points.
`alpha`	Shape parameter ($\alpha > 0$). Scalar or numeric vector. Default: 1.
`beta`	Rate-type parameter ($\beta > 0$), the coefficient of the Gaussian tail $\exp(-\beta x^2)$; larger values pull mass toward zero. Scalar or numeric vector. Default: 1.
`gamma`	Tilt parameter ($\gamma \in R$), the coefficient of the exponential tilt $\exp(\gamma x)$. Scalar or numeric vector. Default: 0.
`log`	Logical; if `TRUE`, log-density is returned. Default: `FALSE`. Read with `as.logical()`, so that 1 is `TRUE`; a value that gives `NA` is an error.

Details

The MHN density (Sun et al. 2023, Lemma 1a) is

$$f(x \mid \alpha, \beta, \gamma) = \frac{2\beta^{\alpha/2} x^{\alpha-1} \exp(-\beta x^2 + \gamma x)}{\Psi[\alpha/2, \gamma/\sqrt{\beta}]} \quad (x > 0)$$

where $\Psi[a, z] = \sum_{n \geq 0} \Gamma(a + n/2) z^n / n!$ is the Fox-Wright function ${}_1\Psi_1[(a, 1/2); (1, 0); z]$ (Sun et al. 2023, Supplementary Eq. (1)).

The default parameters `alpha = 1, beta = 1, gamma = 0` correspond to the half-normal distribution $\mathrm{HN}(1/\sqrt{2})$.

At $\gamma = 0$ and at $\alpha = 1$ exactly the density is computed from the closed form of the law it reduces to; a shape or a tilt merely close to those values is a different law, and is computed as in the general case:

- $\gamma = 0$: sqrt-Gamma distribution, for $\alpha \leq 10^6$; a larger shape is computed as in the general case.
- $\alpha = 1$: truncated normal distribution (Sun et al., 2023, Lemma 6b), the normal law of mean $\mu = \gamma/(2\beta)$ and standard deviation $\sigma = 1/\sqrt{2\beta}$ restricted to $(0, \infty)$. However far 0 lies from the mean, the density holds to the Exponential and normal limits it tends to; where $\gamma/(2\beta)$ passes the largest double under a negative tilt, it is the Exponential density $-\gamma e^{\gamma x}$.

Otherwise the log density is computed from the formula above; for a large shape or tilt, where the terms of that formula grow too large, the law is measured instead from the peak of the density of $\log X$.

Accuracy and range. With $z = \gamma/\sqrt{\beta}$ the scale-free tilt, for $\alpha \leq 10^{15}$, $|z| \leq 10^{15}$ and β in $[10^{-300}, 10^{300}]$ the log density is accurate to the working tolerance `sqrt(.Machine$double.eps)`. Outside that range the law can be narrower than the spacing of the doubles near its peak (for z above about 10^{16}, or α above about 10^{31}), and the density is still that of the exact law at the double `x` given, almost always 0 there. For a positive tilt whose mode $\gamma/(2\beta)$ lies beyond the largest double, which needs a subnormal β, `log = TRUE` returns `-Inf` where the log density is finite. Two cases give `NA` with a warning instead: within 45 standard deviations of the peak of a law whose peak cannot be placed to within 10^{-11} of its standard deviation, which needs a law narrower than the spacing by a factor of order 10^{19} or more and then happens for some parameters only (as at $\alpha = \beta = \gamma = 10^{200}$); and where the whole law lies beyond the largest double without a positive tilt to place it (a shape near the largest double against a subnormal β). Where z itself overflows, the law is its limit: Gamma(α, rate $|\gamma|$) under a negative tilt and normal about its mode with variance $1/(2\beta)$ under a positive one.

A subnormal shape, below the smallest normal double 2.2×10^{-308}, is computed apart, at $\gamma = 0$ too, from a form of the law in which α only multiplies and X lies below the smallest positive double with probability $1/(1 + w)$, where w, defined in Appendix A.4 of the vignette cited below, grows with the tilt. Below a tilt z of about 52 (53 at the smallest shapes) that probability is 1 in double precision: `pmhn` is 1 at every positive x, `qmhn` is 0 for every lower-tail probability below 1 given on the probability scale (the upper tail, or `log.p = TRUE`, still gives the quantiles of the far upper tail), and `rmhn` returns 0. Above a tilt between about 55 and 56.3, as α falls, the mass by the origin is at most 2^{-64}, about 5.4×10^{-20}, of the whole, and the density, the upper tail, the moments and the law of the draws no longer depend on α.

When any of `alpha`, `beta`, `gamma` is a vector, the density is evaluated element-wise. The Fox-Wright Ψ normalizing constant is computed once for each position of the recycling cycle while that cycle is

at most 256 long, and otherwise once for each run of consecutive elements that share the (α, β, γ) triple. So recycled parameters, and parameter vectors in which equal triples are adjacent, are much faster than triples that alternate along a long vector, and than calling `dmhn` inside an R loop.

The numerical methods, and the reasons for the ranges above, are described in Appendix A of `vignette("theory", package = "mhn")`.

Value

A numeric vector. The output length equals `max(length(x), length(alpha), length(beta), length(gamma))`, or 0 if any of them has length zero; each input is recycled to that length following standard R recycling rules. For `x < 0`, the density is 0 (`-Inf` if `log = TRUE`). At `x = 0` it is `Inf` for `alpha < 1`, the truncated normal density at 0 for `alpha = 1` ($2\sqrt{\beta/\pi}$ at `gamma = 0`), and 0 for `alpha > 1`. As in `dgamma`, an element whose parameters lie outside the parameter space (`alpha` or `beta` not positive and finite, or `gamma` not finite) is `NaN`, with a warning, and an element with an `NA` or `NaN` among its inputs is `NA` or `NaN` (`NA` if any of them is `NA`), without one; the other elements are unaffected. The arguments must be numeric (double, integer or logical): a factor, a character string, a complex number, a list or `NULL` is an error, as in `dnorm`. As with `dnorm`, the attributes of `x` (names, `dim`, `dimnames`) are kept when the result has the length of `x`; a classed `x`, such as a Date, is taken as its numbers and gives a plain vector. Unlike `dnorm`, attributes of a longer parameter vector are not carried over.

References

Sun, J., Kong, M., & Pal, S. (2023). The Modified-Half-Normal distribution: Properties and an efficient sampling scheme. *Communications in Statistics - Theory and Methods*, 52(5), 1591–1613.

See Also

`mhn_mean`, `mhn_var`, `mhn_mode`

Examples

```
x <- seq(0, 5, length.out = 100)
plot(x, dmhn(x, alpha = 2, beta = 1, gamma = 1), type = "l")

# Log-density
dmhn(1, alpha = 2, beta = 1, gamma = 1, log = TRUE)
```

`mhn_kurtosis` *Excess Kurtosis of the Modified Half-Normal Distribution*

Description

Computes the excess kurtosis $\gamma_2 = E[(X-\mu)^4]/\sigma^4 - 3$ for $X \sim \mathrm{MHN}(\alpha, \beta, \gamma)$.

Usage

```
mhn_kurtosis(alpha, beta, gamma)
```

Arguments

`alpha`	Shape parameter ($\alpha > 0$), a single number.
`beta`	Rate-type parameter ($\beta > 0$), the coefficient of the Gaussian tail $\exp(-\beta x^2)$; larger values pull mass toward zero. A single number.
`gamma`	Tilt parameter ($\gamma \in R$), the coefficient of the exponential tilt $\exp(\gamma x)$. A single number.

Details

The excess kurtosis is formed from the second and fourth central moments. In exact arithmetic these follow from the mean of Sun et al. (2023, Lemma 2a) and the recurrence for the raw moments of their Lemma 2b; the function obtains them instead, wherever it can, by numerical integration of the unnormalised density, which needs no normalising constant and stays accurate under a large tilt.

Against independent quadratures, for shapes from 1 to 60 and tilts $\gamma/\sqrt{\beta}$ from -20 to 12, the absolute error stayed below 1e-9; the largest found, 6e-10, is for a shape between 2 and 3 under a positive tilt.

For a subnormal shape, below the smallest normal double 2.2e-308, the excess kurtosis depends on the tilt $z = \gamma/\sqrt{\beta}$ as follows. While the mass at the peak of the density is small against the mass by the origin (up to z of about 53 at the largest subnormal shape and 54.5 at the smallest), it grows like $1/\alpha$ as the shape falls, by a factor that depends on the tilt: 2 at $z = 0$ and 8.2×10^{-16} at $z = 12$. It is `Inf` where it passes the largest double: below a shape of about 1.1e-308 at $z = 0$, at every subnormal shape for $z = -30$, and at none for $z = 12$. Where the two masses are about equal (z from about 53.3 to 54.7) the peak takes over, and above a tilt of 55 to 56.3 the kurtosis no longer depends on α.

The function is not vectorised in its parameters: `alpha`, `beta` and `gamma` must each be a single number. To evaluate several parameter triples, use `mapply`, as in the examples.

The numerical methods are described in Appendix A of `vignette("theory", package = "mhn")`.

Value

A numeric scalar.

References

Sun, J., Kong, M., & Pal, S. (2023). The Modified-Half-Normal distribution: Properties and an efficient sampling scheme. *Communications in Statistics - Theory and Methods*, 52(5), 1591–1613. (Lemmas 2a and 2b)

See Also

`mhn_skewness`, `mhn_mean`

Examples

```
mhn_kurtosis(alpha = 2, beta = 1, gamma = 0)
mapply(mhn_kurtosis, alpha = c(0.5, 2, 5), beta = 1, gamma = c(-1, 0, 3))
```

`mhn_mean` *Mean of the Modified Half-Normal Distribution*

Description

Computes $E(X)$ for $X \sim \mathrm{MHN}(\alpha, \beta, \gamma)$.

Usage

```
mhn_mean(alpha, beta, gamma)
```

Arguments

`alpha`	Shape parameter ($\alpha > 0$), a single number.
`beta`	Rate-type parameter ($\beta > 0$), the coefficient of the Gaussian tail $\exp(-\beta x^2)$; larger values pull mass toward zero. A single number.
`gamma`	Tilt parameter ($\gamma \in R$), the coefficient of the exponential tilt $\exp(\gamma x)$. A single number.

Details

By Sun et al. (2023, Lemma 2a) the mean is a ratio of Fox-Wright Ψ functions,

$$E(X) = \frac{\Psi[(\alpha+1)/2,\, z]}{\sqrt{\beta}\,\Psi[\alpha/2,\, z]}, \qquad z = \gamma/\sqrt{\beta}.$$

At $\gamma = 0$ the ratio is $\Gamma((\alpha+1)/2)/(\sqrt{\beta}\,\Gamma(\alpha/2))$ (Sun et al. 2023, Lemma 6a). For every $\alpha > 0$ the mean is at most $(\gamma + \sqrt{\gamma^2 + 8\alpha\beta})/(4\beta)$ (Sun et al. 2023, Lemma 4a and the remark after it).

The ratio is used where it can be evaluated accurately. Where it cannot, mainly under a strong tilt or at a very large α, the mean is instead obtained by numerical integration of the unnormalised density, which needs no Ψ value.

The function is not vectorised in its parameters: `alpha`, `beta` and `gamma` must each be a single number. To evaluate several parameter triples, use `mapply`, as in the examples.

The numerical methods are described in Appendix A of `vignette("theory", package = "mhn")`.

Value

A numeric scalar.

References

Sun, J., Kong, M., & Pal, S. (2023). The Modified-Half-Normal distribution: Properties and an efficient sampling scheme. *Communications in Statistics - Theory and Methods*, 52(5), 1591–1613. (Lemmas 2a, 4a and 6a)

See Also

mhn_var, dmhn

Examples

```
mhn_mean(alpha = 2, beta = 1, gamma = 0)
mapply(mhn_mean, alpha = c(0.5, 2, 5), beta = 1, gamma = c(-1, 0, 3))
```

---

mhn_mode *Mode of the Modified Half-Normal Distribution*

Description

Computes the mode of the MHN distribution. For $\alpha \geq 1$ this is the maximiser of the density. For $0 < \alpha < 1$ the density is unbounded as $x \to 0^+$, so a non-NA result is an interior local maximum, not the most probable value.

Usage

```
mhn_mode(alpha, beta, gamma)
```

Arguments

alpha	Shape parameter ($\alpha > 0$), a single number.
beta	Rate-type parameter ($\beta > 0$), the coefficient of the Gaussian tail $\exp(-\beta x^2)$; larger values pull mass toward zero. A single number.
gamma	Tilt parameter ($\gamma \in R$), the coefficient of the exponential tilt $\exp(\gamma x)$. A single number.

Details

The mode depends on α:

$\alpha > 1$ $(\gamma + \sqrt{\gamma^2 + 8\beta(\alpha - 1)})/(4\beta)$ (Sun et al. 2023, Lemma 3b).

$\alpha = 1$ $\max(0, \gamma/(2\beta))$, obtained as the mode of the truncated normal $\mathrm{TN}(\gamma/(2\beta), 1/\sqrt{2\beta}, 0, \infty)$ that the MHN reduces to in this case (Sun et al. 2023, Lemma 6b). For $\gamma \leq 0$ this is the boundary point 0. Sun et al. (2023, Lemma 3d) state that no mode exists for $\gamma < 0$ and $\alpha \leq 1$; at $\alpha = 1$ the density is bounded and decreases from its supremum at $x = 0$, and that point is returned. Only $\alpha = 1$ itself takes this branch. The Lemma 3b value tends to it as $\alpha \to 1$ from above, and for $\gamma > 0$ so does the Lemma 3c value from below; for $\gamma \leq 0$ and $\alpha < 1$ there is no mode (NA).

$0 < \alpha < 1$ An interior local maximum exists only when $\gamma > 0$ and $\alpha > 1 - \gamma^2/(8\beta)$, and it is then located at $(\gamma + \sqrt{\gamma^2 + 8\beta(\alpha - 1)})/(4\beta)$ (Sun et al. 2023, Lemma 3c). Lemma 3c states the condition with $\geq$; the strict inequality is used here because at $\alpha = 1 - \gamma^2/(8\beta)$ the two stationary points coincide at $\gamma/(4\beta)$, where $\frac{d}{dx} \log f(x) = -\frac{2\beta}{x}\left(x - \frac{\gamma}{4\beta}\right)^2 \leq 0$, so the

density is still strictly decreasing and has an inflection point, not a maximum, there. In every other case the density is strictly decreasing on $(0, \infty)$ (Sun et al. 2023, Lemma 3d, for $\gamma \neq 0$; for $\gamma = 0$ the derivative $-(2\beta x^2 + 1 - \alpha)/x$ of $\log f$ is negative) and NA is returned.

The function is not vectorised in its parameters: alpha, beta and gamma must each be a single number. To evaluate several parameter triples, use mapply, as in the examples.

Value

A numeric scalar. For $0 < \alpha < 1$ with no interior local maximum (density strictly decreasing on $(0, \infty)$) the result is NA; at $\alpha = 1$ with $\gamma \leq 0$ the mode sits on the boundary and 0 is returned. See Details.

References

Sun, J., Kong, M., & Pal, S. (2023). The Modified-Half-Normal distribution: Properties and an efficient sampling scheme. *Communications in Statistics - Theory and Methods*, 52(5), 1591–1613. (Lemma 3b–d, Lemma 6b)

See Also

dmhn, mhn_mean

Examples

```
mhn_mode(alpha = 2, beta = 1, gamma = 1)
mhn_mode(alpha = 1, beta = 1, gamma = 2)
mhn_mode(alpha = 0.5, beta = 1, gamma = -1)  # NA
mhn_mode(alpha = 0.5, beta = 1, gamma = 5)   # a local maximum only
mapply(mhn_mode, alpha = c(0.5, 2, 5), beta = 1, gamma = c(-1, 0, 3))
```

---

mhn_skewness *Skewness of the Modified Half-Normal Distribution*

Description

Computes the skewness $\gamma_1 = E[(X - \mu)^3]/\sigma^3$ for $X \sim \mathrm{MHN}(\alpha, \beta, \gamma)$.

Usage

```
mhn_skewness(alpha, beta, gamma)
```

Arguments

alpha	Shape parameter ($\alpha > 0$), a single number.
beta	Rate-type parameter ($\beta > 0$), the coefficient of the Gaussian tail $\exp(-\beta x^2)$; larger values pull mass toward zero. A single number.
gamma	Tilt parameter ($\gamma \in R$), the coefficient of the exponential tilt $\exp(\gamma x)$. A single number.

Details

The skewness is formed from the second and third central moments. In exact arithmetic these follow from the mean of Sun et al. (2023, Lemma 2a) and the recurrence for the raw moments of their Lemma 2b; the function obtains them instead, wherever it can, by numerical integration of the unnormalised density, which needs no normalising constant and stays accurate under a large tilt.

Against independent quadratures, for shapes from 1 to 60 and tilts $\gamma/\sqrt{\beta}$ from -20 to 12, the absolute error stayed below 1e-9; the largest found, 3e-10, is for a shape between 2 and 3 under a positive tilt.

For a subnormal shape, below the smallest normal double 2.2e-308, the skewness depends on the tilt $z = \gamma/\sqrt{\beta}$ as follows. While the mass at the peak of the density is small against the mass by the origin (up to z of about 53 at the largest subnormal shape and 54.5 at the smallest), it grows like $1/\sqrt{\alpha}$ as the shape falls, by a factor that depends on the tilt: 1.25 at $z = 0$ and 2.8×10^{-8} at $z = 12$. It turns negative where the two masses are about equal (z from about 53.3 to 54.7), and above a tilt of 55 to 56.3 it no longer depends on α.

The function is not vectorised in its parameters: `alpha`, `beta` and `gamma` must each be a single number. To evaluate several parameter triples, use `mapply`, as in the examples.

The numerical methods are described in Appendix A of `vignette("theory", package = "mhn")`.

Value

A numeric scalar.

References

Sun, J., Kong, M., & Pal, S. (2023). The Modified-Half-Normal distribution: Properties and an efficient sampling scheme. *Communications in Statistics - Theory and Methods*, 52(5), 1591–1613. (Lemmas 2a and 2b)

See Also

`mhn_kurtosis`, `mhn_mean`

Examples

```
mhn_skewness(alpha = 2, beta = 1, gamma = 0)
mapply(mhn_skewness, alpha = c(0.5, 2, 5), beta = 1, gamma = c(-1, 0, 3))
```

---

mhn_var *Variance of the Modified Half-Normal Distribution*

Description

Computes $\mathrm{Var}(X)$ for $X \sim \mathrm{MHN}(\alpha, \beta, \gamma)$.

Usage

```
mhn_var(alpha, beta, gamma)
```

Arguments

alpha	Shape parameter ($\alpha > 0$), a single number.
beta	Rate-type parameter ($\beta > 0$), the coefficient of the Gaussian tail $\exp(-\beta x^2)$; larger values pull mass toward zero. A single number.
gamma	Tilt parameter ($\gamma \in R$), the coefficient of the exponential tilt $\exp(\gamma x)$. A single number.

Details

Uses the formula (Sun et al. 2023, Lemma 2c)

$$\mathrm{Var}(X) = \frac{\alpha}{2\beta} + E(X)\left(\frac{\gamma}{2\beta} - E(X)\right)$$

where it is accurate. Where it would lose accuracy, under a strong tilt $z = \gamma/\sqrt{\beta}$ of either sign and at a large α unless $\gamma = 0$, the variance is instead obtained by numerical integration of the unnormalised density.

For $\alpha \geq 1$, the variance satisfies $\mathrm{Var}(X) \leq 1/(2\beta)$ (Sun et al., 2023, Lemma 4c).

The function is not vectorised in its parameters: `alpha`, `beta` and `gamma` must each be a single number. To evaluate several parameter triples, use `mapply`, as in the examples.

The numerical methods are described in Appendix A of `vignette("theory", package = "mhn")`.

Value

A numeric scalar.

References

Sun, J., Kong, M., & Pal, S. (2023). The Modified-Half-Normal distribution: Properties and an efficient sampling scheme. *Communications in Statistics - Theory and Methods*, 52(5), 1591–1613. (Lemmas 2c and 4c)

See Also

`mhn_mean`, `dmhn`

Examples

```
mhn_var(alpha = 2, beta = 1, gamma = 0)
mapply(mhn_var, alpha = c(0.5, 2, 5), beta = 1, gamma = c(-1, 0, 3))
```

pmhn *Distribution Function of the Modified Half-Normal Distribution*

Description

Computes the cumulative distribution function (CDF) of the Modified Half-Normal (MHN) distribution with parameters `alpha`, `beta`, and `gamma`.

Usage

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

Arguments

`q`	Numeric vector of quantiles.
`alpha`	Shape parameter ($\alpha > 0$). Scalar or numeric vector. Default: 1.
`beta`	Rate-type parameter ($\beta > 0$), the coefficient of the Gaussian tail $\exp(-\beta x^2)$; larger values pull mass toward zero. Scalar or numeric vector. Default: 1.
`gamma`	Tilt parameter ($\gamma \in R$), the coefficient of the exponential tilt $\exp(\gamma x)$. Scalar or numeric vector. Default: 0.
`lower.tail`	Logical; if `TRUE` (default), probabilities are $P(X \leq q)$, otherwise $P(X > q)$.
`log.p`	Logical; if `TRUE`, probabilities are returned on the log scale. Default: `FALSE`. Both flags are read with `as.logical()`, so that 1 is `TRUE`; a value that gives `NA` is an error.

Details

In the general case the lower tail is computed, where that is accurate (see below), from the series representation (Sun et al. 2023, Lemma 1b)

$$F(x \mid \alpha, \beta, \gamma) = \frac{1}{\Psi[\alpha/2, \gamma/\sqrt{\beta}]} \sum_{i=0}^{\infty} \frac{z^i}{i!} \Gamma(s_i)\, P(s_i, \beta x^2)$$

where $z = \gamma/\sqrt{\beta}$, $s_i = (\alpha+i)/2$, and $P(s, y)$ is the regularized lower incomplete gamma function.

The default parameters `alpha = 1, beta = 1, gamma = 0` correspond to the half-normal distribution $\mathrm{HN}(1/\sqrt{2})$.

At $\gamma = 0$ and at $\alpha = 1$ exactly the distribution function is computed from the law it reduces to; a shape or a tilt merely close to those values is a different law, and is computed as in the general case:

- $\gamma = 0$: $F(q) = P(\alpha/2, \beta q^2)$, the regularized incomplete gamma function of the square-root-Gamma law (Sun et al. 2023, Lemma 6a), for $\alpha \leq 10^6$; a larger shape is computed as in the general case.
- $\alpha = 1$: the truncated-normal CDF (Sun et al. 2023, Lemma 6b). Near the truncation point the lower tail keeps its relative accuracy however small it is. Where $-\gamma/\sqrt{2\beta}$ overflows the law is Exponential with rate $|\gamma|$.

The series is used, where it is accurate, for $\alpha \leq 100$ with $|z| \leq 20$, and for $\alpha < 1$ up to $|z|$ of about 830 (for $|z| > 20$ it costs milliseconds per value); elsewhere the lower tail is computed by numerical integration of the density. The upper tail $P(X > q)$ is computed directly, by numerical integration of the density over (q, ∞), and not as $1 - F$, which would round to 0 once it falls below about 10^{-16}. Each tail is computed directly on its own side of the peak of the density of $\log X$, which lies at or above the median, and as one minus the other tail beyond it. For a large shape or tilt, where the terms of these forms grow too large, both tails are measured from that peak instead. With `log.p = TRUE` both tails are formed on the log scale throughout, so that $\log F$ is still resolved where F rounds to 1 or is below the smallest positive double, and a log probability close to 0, about minus the other tail, keeps the relative accuracy of that other tail to within about 10^{-9} (Appendix A.4 of the vignette cited below).

Accuracy and range. With $z = \gamma/\sqrt{\beta}$ as above, for $\alpha \leq 10^{15}$, $|z| \leq 10^{15}$ and β in $[10^{-300}, 10^{300}]$ both tails are accurate to the working tolerance `sqrt(.Machine$double.eps)` within 10 standard deviations of the peak, and to about twice that beyond. Outside that range the law can be narrower than the spacing of the doubles near its peak (for z above about 10^{16}, or α above about 10^{31}); the probabilities are then still those of the exact law at the doubles `q` given, a step from 0 to 1 between two adjacent doubles, whose value at the double nearest the peak is set by where the peak lies within its spacing. Two cases give `NA` with a warning instead: within 45 standard deviations of the peak of a law whose peak cannot be placed to within 10^{-11} of its standard deviation, which needs a law narrower than the spacing by a factor of order 10^{19} or more and then happens for some parameters only (as at $\alpha = \beta = \gamma = 10^{200}$); and where the whole law lies beyond the largest double without a positive tilt to place it (a shape near the largest double against a subnormal β). A third case lies at the other end of the range: where β and the tilt are both close to the largest double (β from about 2×10^{307} with γ near the largest double, or from about 5×10^{307} with γ from about 10^{306}), at some points q above the peak where βq is above about half the largest double, the log upper tail is `-Inf` where it is finite (the upper tail itself is then 0, which is its value in double precision), and next to the point where βq overflows both are `NA` with a warning. Where z itself overflows, the law is its limit: Gamma(α, rate $|\gamma|$) under a negative tilt, and normal about its mode with variance $1/(2\beta)$ under a positive one. A subnormal shape, below the smallest normal double, is computed apart (see `dmhn`), and there $\log F$ keeps its relative accuracy too; below a tilt z of about 52 (53 at the smallest shapes) `pmhn` is then 1 at every positive `q`.

When any of `alpha`, `beta`, `gamma` is a vector, the CDF is evaluated element-wise. The Fox-Wright Ψ normalizing constant is computed once for each position of the recycling cycle while that cycle is at most 256 long, and otherwise once for each run of consecutive elements that share the (α, β, γ) triple. So recycled parameters, and parameter vectors in which equal triples are adjacent, are much faster than triples that alternate along a long vector, and than calling `pmhn` inside an R loop.

The numerical methods, and the reasons for the ranges above, are described in Appendix A of `vignette("theory", package = "mhn")`.

Value

A numeric vector. The output length equals `max(length(q), length(alpha), length(beta), length(gamma))`, or 0 if any of them has length zero; each input is recycled to that length following standard R recycling rules. For `q <= 0` the CDF is 0; for `q = Inf` it is 1. As in `pgamma`, an element whose parameters lie outside the parameter space (`alpha` or `beta` not positive and finite, or `gamma` not finite) is `NaN`, with a warning, and an element with an `NA` or `NaN` among its inputs is `NA` or `NaN` (`NA` if any of them is `NA`), without one; the other elements are unaffected. The arguments must be numeric (double, integer or logical): a factor, a character string, a complex number, a list or `NULL`

is an error, as in pnorm. As with pnorm, the attributes of q (names, dim, dimnames) are kept when the result has the length of q; a classed q, such as a Date, is taken as its numbers and gives a plain vector. Unlike pnorm, attributes of a longer parameter vector are not carried over.

References

Sun, J., Kong, M., & Pal, S. (2023). The Modified-Half-Normal distribution: Properties and an efficient sampling scheme. *Communications in Statistics - Theory and Methods*, 52(5), 1591–1613.

See Also

dmhn, qmhn, rmhn

Examples

```
# Basic evaluation
pmhn(c(0.5, 1, 1.5), alpha = 2, beta = 1, gamma = 1)

# Tail / log forms
pmhn(2, alpha = 2, beta = 1, gamma = 1, lower.tail = FALSE)
pmhn(2, alpha = 2, beta = 1, gamma = 1, log.p = TRUE)

# Special case: gamma = 0 reduces to sqrt-Gamma
all.equal(pmhn(1.5, alpha = 2, beta = 1, gamma = 0),
          pgamma(1.5^2, shape = 1, rate = 1))
```

---

qmhn *Quantile Function of the Modified Half-Normal Distribution*

Description

Computes the quantile (inverse cumulative) function of the Modified Half-Normal (MHN) distribution with parameters alpha, beta, and gamma.

Usage

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

Arguments

p	Numeric vector of probabilities.
alpha	Shape parameter ($\alpha > 0$). Scalar or numeric vector. Default: 1.
beta	Rate-type parameter ($\beta > 0$), the coefficient of the Gaussian tail $\exp(-\beta x^2)$; larger values pull mass toward zero. Scalar or numeric vector. Default: 1.
gamma	Tilt parameter ($\gamma \in R$), the coefficient of the exponential tilt $\exp(\gamma x)$. Scalar or numeric vector. Default: 0.

`lower.tail` Logical; if `TRUE` (default), probabilities are $P(X \leq x)$, otherwise $P(X > x)$, x being the quantile returned.

`log.p` Logical; if `TRUE`, probabilities are provided on the log scale. Default: `FALSE`. Both flags are read with `as.logical()`, so that 1 is `TRUE`; a value that gives `NA` is an error.

Details

For the general case, $q = F^{-1}(p)$ is obtained by a root-finder applied to the distribution function computed as in pmhn, or for `lower.tail = FALSE` to the logarithm of its directly computed upper tail.

The default parameters `alpha = 1, beta = 1, gamma = 0` correspond to the half-normal distribution $\mathrm{HN}(1/\sqrt{2})$.

At $\gamma = 0$ and at $\alpha = 1$ exactly the quantile is computed from the law it reduces to; a shape or a tilt merely close to those values is a different law, and is computed as in the general case:

- $\gamma = 0$: $q = \sqrt{y/\beta}$, where $P(\alpha/2, y) = p$ for the regularized incomplete gamma function P (Sun et al. 2023, Lemma 6a), for $\alpha \leq 10^6$; a larger shape is computed as in the general case.
- $\alpha = 1$: the truncated-normal inverse (Sun et al. 2023, Lemma 6b). The quantile of a small lower-tail probability keeps its relative accuracy. Where $-\gamma/\sqrt{2\beta}$ overflows the law is Exponential with rate $|\gamma|$, and its quantile is that one's.

A probability close to 1 is solved, where that is more accurate, on the other tail for one minus it, which is formed exactly: a lower-tail p on the upper tail, as the root of $P(X > x) = 1 - p$, since the distribution function is resolved there only to the rounding of 1 (and, for a small shape, to the error of the lower tail; see pmhn), and an upper-tail p on the distribution function. Where pmhn takes the distribution function from its series ($\alpha < 1$ with $|z|$ up to about 830, and $\alpha \leq 100$ with $|z| \leq 20$, $z = \gamma/\sqrt{\beta}$), a lower-tail p is solved on the series itself, and for $\alpha < 1$ with $|z|$ from about 100 one quantile can take from a tenth of a second to seconds (about 1 s at $\alpha = 0.8$, $z = 829$ and $p = 0.5$); where $1 - p$ is known exactly, `lower.tail = FALSE` gives the same quantile at once.

Accuracy and range. The root is found to a relative 2^{-46}, and where pmhn measures the law from its peak (see there) also to within 2^{-46} of the law's standard deviation, which can be narrower. The quantile inverts pmhn, whose range of accuracy it inherits: for $\alpha \leq 10^{15}$, $|z| \leq 10^{15}$ and β in $[10^{-300}, 10^{300}]$ it is the root, to that tolerance or to the spacing of the doubles near it, whichever is larger, of an equation whose tail probabilities are as accurate as those of pmhn (to the working tolerance within 10 standard deviations of the peak, and to about twice that beyond; see there), which places it within about that many of the law's standard deviations. Next to the origin, where $F(x)$ is about a multiple of x^α, its relative error is that of the smaller tail probability divided by about α. A quantile below `.Machine$double.xmin` is resolved to the spacing of the subnormal doubles there, about 4.9×10^{-324}, and one below that spacing is returned as 0, as qgamma returns a quantile that underflows; a quantile beyond the largest double is returned as `Inf`, as qgamma returns one that overflows. Where the law is narrower than the spacing of the doubles (for z above about 10^{16}, or α above about 10^{31}) its distribution function is a step between two adjacent doubles, and the quantile of every p strictly inside $(0, 1)$ is the one of the two nearer to the true quantile, placed from the exact offset of the law's peak. Where pmhn is NA (see there), so is the quantile, with a warning. Where the log upper tail of pmhn is `-Inf` in place of a finite value (β and the tilt both close to the largest double, see there), the quantile of a `log.p` probability beyond that point is the point at which it turns `-Inf`, without a warning. For a subnormal shape (see dmhn) and a tilt z below

about 52 (53 at the smallest shapes), qmhn is 0 for every lower-tail probability below 1 given on the probability scale; the upper tail, or log.p = TRUE, still gives the quantiles of the far upper tail.

When any of alpha, beta, gamma is a vector, the quantile is evaluated element-wise. The Fox-Wright Ψ normalizing constant and moments $E(X)$, $\mathrm{Var}(X)$ (used to size the root-finder bracket) are computed once for each position of the recycling cycle while that cycle is at most 256 long, and otherwise once for each run of consecutive elements that share the (α, β, γ) triple.

The numerical methods, and the reasons for the ranges above, are described in Appendix A of vignette("theory", package = "mhn").

Value

A numeric vector. The output length equals max(length(p), length(alpha), length(beta), length(gamma)), or 0 if any of them has length zero; each input is recycled to that length following standard R recycling rules. As in qgamma, an element whose parameters lie outside the parameter space (alpha or beta not positive and finite, or gamma not finite) is NaN, with a warning, and an element with an NA or NaN among its inputs is NA or NaN (NA if any of them is NA), without one; the other elements are unaffected. The arguments must be numeric (double, integer or logical): a factor, a character string, a complex number, a list or NULL is an error, as in qnorm. qmhn(0) = 0 and qmhn(1) = Inf. Probabilities outside $[0, 1]$ (log probabilities above 0 when log.p = TRUE) yield NaN, with a warning as in qnorm. If the distribution function cannot be evaluated where the root is sought, a warning is issued and NA is returned for that element. The attributes of p (names, dim, dimnames) are kept when the result has the length of p; a classed p, such as a Date, is taken as its numbers and gives a plain vector. Unlike qnorm, attributes of a longer parameter vector are not carried over.

References

Sun, J., Kong, M., & Pal, S. (2023). The Modified-Half-Normal distribution: Properties and an efficient sampling scheme. *Communications in Statistics - Theory and Methods*, 52(5), 1591–1613.

See Also

dmhn, pmhn, rmhn

Examples

```
# Basic evaluation
qmhn(c(0.1, 0.5, 0.9), alpha = 2, beta = 1, gamma = 1)

# Round-trip: F(F^-1(p)) ~ p
p <- c(0.05, 0.25, 0.5, 0.75, 0.95)
all.equal(pmhn(qmhn(p, alpha = 2, beta = 1, gamma = 1),
               alpha = 2, beta = 1, gamma = 1),
          p, tolerance = 1e-6)

# Tail / log forms
qmhn(0.95, alpha = 2, beta = 1, gamma = 1, lower.tail = FALSE)
qmhn(log(0.05), alpha = 2, beta = 1, gamma = 1, log.p = TRUE)
```

rmhn *Random Generation from the Modified Half-Normal Distribution*

Description

Draws random variates from the Modified Half-Normal (MHN) distribution with parameters `alpha`, `beta`, and `gamma`.

Usage

```
rmhn(n, alpha = 1, beta = 1, gamma = 0, method = c("auto", "rtdr", "sun"))
```

Arguments

`n`	Number of variates to draw: a non-negative number no larger than `.Machine$integer.max`, truncated to an integer, so that `n = 0` returns `numeric(0)`. Following the base R convention of `rnorm()` and `rgamma()`, if `length(n)` is not 1 the number required is taken to be `length(n)`, so that a zero-length `n` also returns `numeric(0)`; `NULL` is an error.
`alpha`	Shape parameter ($\alpha > 0$). Scalar or numeric vector. Default: 1.
`beta`	Rate-type parameter ($\beta > 0$), the coefficient of the Gaussian tail $\exp(-\beta x^2)$; larger values pull mass toward zero. Scalar or numeric vector. Default: 1.
`gamma`	Tilt parameter ($\gamma \in R$), the coefficient of the exponential tilt $\exp(\gamma x)$. Scalar or numeric vector. Default: 0.
`method`	Sampling algorithm. One of `"auto"` (default), `"rtdr"`, or `"sun"`. See Details.

Details

The MHN density is

$$f(x \mid \alpha, \beta, \gamma) = \frac{2\beta^{\alpha/2} x^{\alpha-1} \exp(-\beta x^2 + \gamma x)}{\Psi[\alpha/2, \gamma/\sqrt{\beta}]} \quad (x > 0)$$

where $\Psi[a, z] = \sum_{n \geq 0} \Gamma(a + n/2) z^n / n!$ is the Fox-Wright function ${}_1\Psi_1[(a, 1/2); (1, 0); z]$. `rmhn` does not evaluate Ψ; the rejection-sampling kernels cancel it out. The one exception is the weight of the origin, $1/(1 + w)$, that it draws for a subnormal shape (see Subnormal shapes below).

The default parameters `alpha = 1, beta = 1, gamma = 0` correspond to the half-normal distribution $\mathrm{HN}(1/\sqrt{2})$.

Choice of sampler: Two closed-form special cases are taken first, whatever `method`: a negligible tilt, $|\gamma|/\sqrt{\beta}$ within `sqrt(.Machine$double.eps)` of 0, is drawn as the square root of a Gamma variate, and a shape α within `sqrt(.Machine$double.eps)` of 1 from a truncated normal. Both are exact samplers of their special-case law, which next to $\gamma = 0$ or $\alpha = 1$ is within a total variation distance of 8.6×10^{-9} of the law requested. The one exception is a negligible tilt above $\alpha = 2^{60}$, which is drawn by the Normal proposal of Sun et al. (2023) Algorithm 1 centred at the mode, or by RTDR under `method = "rtdr"` (Sun et al. state Algorithm 1 for $\gamma > 0$; centred at the mode, its Normal envelope holds for either sign). Otherwise `method` selects the rejection sampler:

- `"auto"` (default): the faster of the provably correct samplers, as measured by timing them against each other. For $\alpha < 1$ and $\gamma > 0$ it uses RTDR. Otherwise it chooses between Sun et al. Algorithm 1 ($\gamma > 0$) or Algorithm 3 ($\gamma < 0$) and RTDR by the shape, the tilt $\gamma/\sqrt{\beta}$ and the number of variates per setup (see Vector parameters and setups); Section 7 of `vignette("theory", package = "mhn")` gives the decision tree. The choice changes the speed and the stream of draws (see Reproducibility), not the law drawn.
- `"rtdr"`: the Relaxed Transformed Density Rejection method of Gao & Wang (2025). Gao & Wang bound its acceptance probability below by $1/e \approx 0.368$ for every parameter value, provided the contact points of the envelope lie in the band of their Eq. (7); the search for them does not always reach that band, and the acceptance measured stays above the bound there too. Where RTDR cannot place a law of tiny shape under a strong negative tilt, Sun et al. Algorithm 3 is used instead (see Limits of double precision). Note: Gao & Wang (2025) use the parameterization (λ, α, β) with density proportional to $x^{\lambda-1}\exp(-\alpha x^2 - \beta x)$; the mapping to the Sun et al. parameterization used here is $\lambda \leftrightarrow \alpha$, $\alpha \leftrightarrow \beta$, $\beta \leftrightarrow -\gamma$ (sign flip on the linear term).
- `"sun"`: the algorithms of Sun et al. (2023): Algorithm 1 for $\gamma > 0$ and $\alpha > 1$, and Algorithm 3 for $\gamma < 0$, up to $\alpha = 2^{59}$, beyond which the call stops with an error (see Limits of double precision). The combination $\alpha < 1$ with $\gamma > 0$ is unsupported: unless a special case answers it first, the call stops with the error `Sun method not available for alpha<1 and gamma>0; use method="rtdr"`.

Vector parameters and setups: Vector parameters are recycled to length `n` following standard R rules: only the first `n` elements of each parameter are used, and any further elements are silently ignored, matching the convention of `rnorm`.

The setup of the chosen sampler is built once for each run of equal consecutive (α, β, γ) triples. Short parameter vectors repeat their triples after a period, the least common multiple of the lengths of `alpha`, `beta` and `gamma`; when that period is at most 256 draws and they are recycled over at least four such periods, a setup is kept for each position within the period. So passing parameters grouped by triple, or as short recycled vectors, is faster than calling `rmhn` inside an R loop, and a long run of one triple is drawn fastest by its own scalar call.

The number of variates per setup, from which `method = "auto"` chooses its sampler, and which therefore changes the draws for a fixed seed, is `n` for scalar parameters, and `n` divided by the period, rounded down, where a setup is kept for each position within the period. Otherwise it is 1: parameters given as vectors of length `n` count one per setup, as in a Gibbs sampler, even when equal triples are grouped into runs, and such a call draws what the loop of single draws would. Grouping saves setups but does not change this number, and so not the choice of sampler.

Reproducibility: Under one seed, `method = "rtdr"` and `method = "sun"` draw one stream whatever `n`: the first `k` draws of `rmhn(n, ...)` are those of `rmhn(k, ...)`, and draws split over several calls are those of one call. The default method does not keep this, as it chooses its sampler from the number of variates per setup (see above): `rmhn(4, 5, 1, 2)` and the first four draws of `rmhn(100, 5, 1, 2)` come from different samplers. Its stream therefore changes with `n`, with how the parameters are arranged, and with a version of the package that moves a threshold, although the law drawn does not. Where the draws must not depend on `n`, as when a pilot run is compared with the main one, set `method = "rtdr"` or `method = "sun"`.

Limits of double precision: Double precision limits which laws can be sampled, and how finely. Every sampler draws from the exact law and rounds the draw to double precision, and a draw below the smallest positive double is returned as 0. A rounding error that would shift every draw

alike is kept to a few times 10^{-7} of a standard deviation. For this reason the square-root-Gamma closed form is used only up to $\alpha = 2^{60}$, and Sun et al. Algorithm 3 only up to $\alpha = 2^{59}$ (about 5.8×10^{17}). Above $\alpha = 2^{59}$ a negative tilt that is not negligible is drawn by RTDR under the default method, while `method = "sun"` stops with an error; a positive tilt is drawn by Algorithm 1 under both.

The RTDR sampler stops with an error for the whole call once the standard deviation is no larger than the spacing of the double precision numbers at the mode, so that its draws could take only a few distinct values: from $\gamma/\sqrt{\beta}$ of about 9×10^{15} (2^{53} at $\beta = 1$), and, for $\gamma \leq 0$, from α between 2^{103} and 2^{106} (1.0×10^{31} to 8.1×10^{31}), depending on the tilt and on where the mode falls between two powers of two. There the law is normal to far below double precision, with mean m = `mhn_mode(alpha, beta, gamma)` and variance $1/((\alpha - 1)/m^2 + 2\beta)$, and can be drawn from directly, as the error message suggests. Sun et al. Algorithm 1, which the default method uses for such laws with $\gamma > 0$ and $\alpha > 1$, and the truncated normal of $\alpha \approx 1$ do not stop there: their draws are placed at the mode to within a few units in the last place, so that they take only a few distinct values near it.

RTDR cannot place the law in three cases, all under a negative tilt: where $-\gamma/\sqrt{\beta}$ passes the largest double; for $\alpha \leq 1$, where α is below about 2.5×10^{-324} times $-\gamma/\sqrt{\beta}$, which needs α below 4.4×10^{-16}, and almost all the mass of X lies below the smallest positive double; and for a shape of a few, where $-\gamma/\sqrt{\beta}$ comes within a small factor of the largest double. In the second case, and in the first for $\alpha < 10^{-15}$, `method = "rtdr"` draws from Algorithm 3, which samples the law exactly there, except for a subnormal shape, which is drawn as described under Subnormal shapes; otherwise it stops with an error that names the Gamma distribution with shape α and rate $-\gamma$, which the law is to double precision. The default method uses Algorithm 3 in all three cases, except where $-\gamma/\sqrt{\beta}$ passes the largest double at a shape above 2^{59}; there it stops as RTDR does. Where the distribution lies beyond the largest double precision number, no draw can represent it, and every sampler, the closed forms included, stops with an error.

Subnormal shapes: A subnormal shape, below the smallest normal double 2.2×10^{-308}, is drawn as the mixture its law is (see the Details of `dmhn`): 0 with probability $1/(1 + w)$, the probability of a value below the smallest positive double, and otherwise from the peak, through a proxy: the law of the shape 2^{-60} above $\sqrt{\beta}x = 1$, drawn by the sampler `method` selects for that shape. The choice between 0 and the peak compares one uniform variate with $1/(1 + w)$, and so has the resolution of the generator, 2^{-32} for R's default. Where $1/(1+w)$ is 1 in double precision, below a tilt of about 52 (53 at the smallest shapes), every draw is 0 and uses no random number; a shape from 2^{-1022} up is drawn by the samplers above, which use random numbers there too, so that the stream after the draws moves as the shape crosses 2^{-1022} although every draw is 0 on either side. A refusal names the shape asked for.

The algorithms, the timings behind the choice of `method = "auto"`, and the handling of extreme parameters are described in `vignette("theory", package = "mhn")` (Section 7 and Appendix A).

Value

A numeric vector of length `n`. As in `rnorm`, an element whose parameters (after recycling to length `n`) are `NA` or `NaN`, or lie outside the parameter space (`alpha` or `beta` not positive and finite, or `gamma` not finite), is `NaN`, with a single warning "NAs produced", and draws no random numbers. A zero-length `alpha`, `beta` or `gamma` makes every element `NA`, with the same warning. The parameters must be numeric (double, integer or logical): a factor, a character string, a complex number, a list or `NULL` is an error. If a rejection sampler accepts none of 1000 proposals for an element (for a subnormal

shape, none of 1000 draws of the proxy described under Subnormal shapes in Details, each of up to 1000 proposals), that element is NaN, and a single warning reports how many draws were affected; the acceptance rates of the samplers make this vanishingly unlikely. Where the distribution lies beyond the largest double precision number, and at some other extreme values, rmhn instead stops with an error for the whole call (see Details).

References

Sun, J., Kong, M., & Pal, S. (2023). The Modified-Half-Normal distribution: Properties and an efficient sampling scheme. *Communications in Statistics - Theory and Methods*, 52(5), 1591–1613. doi:10.1080/03610926.2021.1934700

Gao, F., & Wang, H.-B. (2025). Generating modified-half-normal random variates by a relaxed transformed density rejection method. *Communications in Statistics - Simulation and Computation*, 1–25. Advance online publication. doi:10.1080/03610918.2025.2524551

Robert, C. P. (1995). Simulation of truncated normal variables. *Statistics and Computing*, 5(2), 121–125. doi:10.1007/BF00143942

See Also

dmhn, mhn_mean, mhn_var

Examples

```
set.seed(1)
rmhn(10, alpha = 2, beta = 1, gamma = 0.5)

# Vector parameters are recycled to length n.
set.seed(1)
rmhn(5, alpha = c(1, 2, 3, 4, 5))
```

Index

dgamma, *4*
dmhn, 2, *7, 8*, *10*, *12–15*, *18, 19*
dnorm, *4*

mapply, *5, 6*, *8–10*
mhn_kurtosis, 4, *9*
mhn_mean, *4, 5*, 6, *8–10*, *19*
mhn_mode, *4*, 7
mhn_skewness, *5*, 8
mhn_var, *4*, *7*, 9, *19*

pgamma, *12*
pmhn, *3*, 11, *14, 15*
pnorm, *13*

qgamma, *14, 15*
qmhn, *3*, *13*, 13
qnorm, *15*

rmhn, *3*, *13*, *15*, 16
rnorm, *18*