Package: wsgnstat (via r-universe)

November 14, 2026

Type Package

Title The Wrapped Skew Generalised Normal Distribution

Version 0.1.0

Description The skew generalised normal distribution on the line, constructed as a scale mixture of a uniform and a generalised gamma variable, and its wrapped version on the circle, with density, distribution and quantile functions, characteristic function, trigonometric moments, random generation, and estimation by the method of trigonometric moments (minimum distance, with sandwich standard errors) and by maximum likelihood. The characteristic function is evaluated through absolutely convergent integrals (the Fox H-function route), so every routine is available for all positive shape parameters, not only where the Fox-Wright series converges. A pure-C back-end does the numerics and also backs the 'Python' package 'wsgnstat'.

URL https://github.com/mtloots/wsgnstat

BugReports https://github.com/mtloots/wsgnstat/issues

License GPL-3

Encoding UTF-8

Language en-GB

Depends R (>= 3.5.0)

Imports stats

Suggests circular, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

NeedsCompilation yes

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

Repository https://mtloots.r-universe.dev

Date/Publication 2026-10-29 13:08:16 UTC

RemoteUrl https://github.com/mtloots/wsgnstat

RemoteRef HEAD

RemoteSha 7fcdeff1efb0e6bad8e76423bdc13399e2a640b7

RemoteSubdir wsgnstat

Contents

cfsgn 2
dsgn 3
dwsgn 4
fit_wsgn_mle 5
fit_wsgn_trig 6
psgn 6
pwsgn 7
rsgn 8
rwsgn 9
trig_moments 10

Index **11**

`cfsgn` *Characteristic function of the skew generalised normal distribution (C back-end)*

Description

Evaluated through the absolutely convergent cosine and sine integrals underlying the Fox H-function representation, hence valid for every `beta` > `0`.

Usage

```
cfsgn(t, mu = 0, sigma = 1, beta = 2, lam1 = 0.5, lam2 = 0.5)
```

Arguments

`t`	numeric vector of arguments.
`mu`	location parameter.
`sigma`	scale parameter (> 0).
`beta`	shape (tail) parameter (> 0).
`lam1`	right-hand rate parameter (> 0).
`lam2`	left-hand rate parameter (> 0).

Value

a complex vector of the same length as `t`.

Examples

```
## reached through absolutely convergent integrals, so every beta > 0 is available,
## including beta <= 1, where the Fox-Wright series does not converge for all t
cfsgn(0, 0, 1, 2, 0.5, 0.5)
## the normal member: exp(-t^2 / 4) for N(0, 1/2)
t <- c(0.5, 1.5, 3)
rbind(engine = Re(cfsgn(t, 0, 0.5, 2, 0.5, 0.5)), closed_form = exp(-t^2 / 4))
## a skew member with beta = 0.8, against the Fourier transform of the density
x <- seq(-60, 60, length.out = 120001)
f <- dsgn(x, 0, 1, 0.8, 0.8, 0.3)
c(engine = cfsgn(1.1, 0, 1, 0.8, 0.8, 0.3),
  quadrature = sum(exp(1i * 1.1 * x) * f) * (x[2] - x[1]))
```

---

dsgn *Density of the skew generalised normal distribution on the line (C back-end)*

Description

The scale mixture of a uniform and a generalised gamma variable, with two-piece exponential-power form: left rate `lam2`, right rate `lam1`.

Usage

```
dsgn(x, mu = 0, sigma = 1, beta = 2, lam1 = 0.5, lam2 = 0.5)
```

Arguments

`x`	numeric vector of quantiles.
`mu`	location parameter.
`sigma`	scale parameter (> 0).
`beta`	shape (tail) parameter (> 0).
`lam1`	right-hand rate parameter (> 0).
`lam2`	left-hand rate parameter (> 0).

Value

the probability density function evaluated at `x`.

Examples

```
## a right-skew, heavy-tailed member and the normal member it generalises
x <- seq(-4, 6, length.out = 401)
f_skew <- dsgn(x, mu = 0, sigma = 1, beta = 1.3, lam1 = 0.3, lam2 = 0.8)
f_norm <- dsgn(x, mu = 0, sigma = 0.5, beta = 2, lam1 = 0.5, lam2 = 0.5)
plot(x, f_skew, type = "l", ylab = "density")
lines(x, f_norm, lty = 2)
```

```
## at beta = 2 with lam1 = lam2 = 0.5 and sigma = 0.5 the law is N(0, 1/2)
max(abs(f_norm - dnorm(x, 0, sqrt(1 / 2))))
## the density integrates to one (Riemann sum on a wide grid)
xx <- seq(-40, 40, length.out = 80001)
sum(dsgn(xx, 0, 1, 1.3, 0.3, 0.8)) * (xx[2] - xx[1])
```

`dwsgn` *Density of the wrapped skew generalised normal distribution (C back-end)*

Description

The identified parameterisation is used: `lam1 = lam`, `lam2 = 1 - lam` and `sigma = sqrt(lam * (1 - lam))`, so the free parameters are the location `mu`, the tail parameter `beta` and the skewness `lam`.

Usage

```
dwsgn(theta, mu = 0, beta = 2, lam = 0.5, K = 50)
```

Arguments

`theta`	numeric vector of angles in radians on `[0, 2*pi)`.
`mu`	location (junction) parameter; for a skew law this is not the mean direction.
`beta`	shape (tail) parameter (> 0).
`lam`	skewness parameter in (0, 1); `0.5` is symmetric.
`K`	truncation of the wrapping sum (`abs(k) <= K`). The default `50` captures the whole mass to double precision for `beta >= 0.8`; for very heavy tails (`beta` near `0.3`) it does not, and `pwsgn(2 * pi, mu, beta, lam, K)` reports the mass that the truncated sum captures, so `K` can be raised until that is one.

Value

the wrapped probability density function evaluated at `theta`.

Examples

```
## a skew member drawn on the circle as a radial curve about the unit circle
th <- seq(0, 2 * pi, length.out = 721)
d <- dwsgn(th, mu = pi / 2, beta = 1.5, lam = 0.3)
plot(cos(th), sin(th), type = "l", asp = 1, xlim = c(-2, 2), ylim = c(-2, 2),
     xlab = "", ylab = "", lty = 3)
lines((1 + d) * cos(th), (1 + d) * sin(th))
## the density integrates to one over the circle
sum(d[-1]) * (th[2] - th[1])
## at beta = 2 with lam = 0.5 the circular law IS the wrapped normal
## with sigma = 1/sqrt(2), which anchors the wrapping machinery exactly
tt <- c(0.5, 2.0, 4.5)
wn <- vapply(tt, function(t) sum(dnorm(t + 2 * pi * (-80:80), 0, 1 / sqrt(2))), 0)
rbind(wsgnstat = dwsgn(tt, 0, 2, 0.5), wrapped_normal = wn)
```

`fit_wsgn_mle` *Fit the wrapped skew generalised normal by maximum likelihood (C back-end)*

Description

Nelder-Mead over the identified parameters, with the wrapped log-likelihood evaluated in C. Standard errors, when requested, come from a parametric bootstrap, the observed information being unstable near the interior cusp the density develops as `beta` approaches one.

Usage

```
fit_wsgn_mle(theta, start = NULL, K = 50, B = 0)
```

Arguments

`theta`	numeric vector of angles in radians.
`start`	numeric starting value `c(mu, beta, lam)`; default is the trigonometric-moment estimate.
`K`	truncation of the wrapping sum.
`B`	number of parametric-bootstrap replicates for standard errors; `0` skips them.

Value

a list with `par`, `logLik`, and (if `B > 0`) `se`.

Examples

```
set.seed(12)
truth <- c(mu = 1, beta = 1.5, lam = 0.3)
th <- rwsgn(500, truth["mu"], truth["beta"], truth["lam"])
fit <- fit_wsgn_mle(th)
round(rbind(truth = truth, estimate = fit$par), 3)
fit$logLik

## bootstrap standard errors refit the model once per replicate (a few seconds for B = 30)
fit_b <- fit_wsgn_mle(th, B = 30)
round(rbind(estimate = fit_b$par, se = fit_b$se), 3)
```

fit_wsgn_trig — *Fit the wrapped skew generalised normal by trigonometric moments (C back-end)*

Description

Minimum-distance matching of the first two sample trigonometric moments to their population counterparts, evaluated in C through the integral characteristic function, hence available for every `beta > 0`. Standard errors come from the minimum-distance sandwich.

Usage

```
fit_wsgn_trig(theta, starts = NULL)
```

Arguments

`theta`	numeric vector of angles in radians.
`starts`	list of numeric starting values `c(mu, beta, lam)`.

Value

a list with the estimate `par = c(mu, beta, lam)`, its standard errors `se`, and the objective value `value`.

Examples

```
## simulate from a known member and recover it
set.seed(11)
truth <- c(mu = 1, beta = 1.5, lam = 0.3)
th <- rwsgn(1000, truth["mu"], truth["beta"], truth["lam"])
fit <- fit_wsgn_trig(th)
round(rbind(truth = truth, estimate = fit$par, se = fit$se), 3)
```

psgn — *Distribution function of the skew generalised normal distribution on the line (C back-end)*

Description

Evaluated in closed form through the regularised incomplete gamma function: the mass to the left of `mu` is `lam1 / (lam1 + lam2)`, and each tail is a scaled gamma tail in `(lam * |x - mu| / sigma)^beta`.

Usage

```
psgn(q, mu = 0, sigma = 1, beta = 2, lam1 = 0.5, lam2 = 0.5)
```

Arguments

q	numeric vector of quantiles.
mu	location parameter.
sigma	scale parameter (> 0).
beta	shape (tail) parameter (> 0).
lam1	right-hand rate parameter (> 0).
lam2	left-hand rate parameter (> 0).

Value

the distribution function evaluated at q.

Examples

```
## agreement with numerical integration of the density
q <- c(-1, 0, 0.7, 2.5)
closed <- psgn(q, 0, 1, 1.3, 0.3, 0.8)
numeric <- vapply(q, function(u)
  integrate(dsgn, -Inf, u, mu = 0, sigma = 1, beta = 1.3, lam1 = 0.3, lam2 = 0.8)$value, 0)
rbind(closed, numeric)
## the normal member: psgn equals pnorm with variance 1/2
psgn(0.4, 0, 0.5, 2, 0.5, 0.5) - pnorm(0.4, 0, sqrt(1 / 2))
```

---

pwsgn — *Distribution and quantile functions of the wrapped skew generalised normal distribution*

Description

The distribution function is measured anticlockwise from angle 0: pwsgn(theta) is the probability of the arc [0, theta], computed as the sum over turns of the line law's mass on [2 pi k, theta + 2 pi k] through the regularised incomplete gamma function, so no numerical quadrature is involved. The quantile function inverts it by bisection. The identified parameterisation of dwsgn is used.

Usage

```
pwsgn(q, mu = 0, beta = 2, lam = 0.5, K = 50)

qwsgn(p, mu = 0, beta = 2, lam = 0.5, K = 50)
```

Arguments

q	numeric vector of angles in radians.
mu	location (junction) parameter; for a skew law this is not the mean direction.
beta	shape (tail) parameter (> 0).

lam	skewness parameter in (0, 1); 0.5 is symmetric.
K	truncation of the wrapping sum (abs(k) <= K). The default 50 captures the whole mass to double precision for beta >= 0.8; for very heavy tails (beta near 0.3) it does not, and pwsgn(2 * pi, mu, beta, lam, K) reports the mass that the truncated sum captures, so K can be raised until that is one.
p	numeric vector of probabilities.

Value

pwsgn gives probabilities in [0, 1]: 0 for q <= 0 and, for q >= 2 * pi, the mass captured by the wrapping sum truncated at K (one to double precision unless the tails are very heavy; see dwsgn). qwsgn gives angles in [0, 2 * pi], with a warning for any p above the captured mass.

Examples

```
## the distribution function is the integral of the density
pwsgn(2, mu = 1, beta = 1.5, lam = 0.3)
integrate(dwsgn, 0, 2, mu = 1, beta = 1.5, lam = 0.3)$value
## the quantile function inverts it
p <- c(0.05, 0.25, 0.5, 0.75, 0.95)
q <- qwsgn(p, mu = 1, beta = 1.5, lam = 0.3)
rbind(p, pwsgn(q, mu = 1, beta = 1.5, lam = 0.3))
## the curve rises from 0 to 1 around the circle
th <- seq(0, 2 * pi, length.out = 361)
plot(th, pwsgn(th, mu = 1, beta = 1.5, lam = 0.3), type = "l",
     xlab = "angle (radians)", ylab = "distribution function")
```

---

rsgn *Random generation from the (wrapped) skew generalised normal distribution*

Description

Uses the stochastic representation: U^beta is Gamma with shape 1 + 1/beta, a uniform variate is drawn conditionally, and the product is wrapped when wrap = TRUE.

Usage

```
rsgn(n, mu = 0, sigma = 1, beta = 2, lam1 = 0.5, lam2 = 0.5, wrap = FALSE)
```

Arguments

n	sample size.
mu	location parameter.
sigma	scale parameter (> 0).
beta	shape (tail) parameter (> 0).
lam1	right-hand rate parameter (> 0).
lam2	left-hand rate parameter (> 0).
wrap	logical; reduce modulo 2*pi.

Value

a numeric vector of draws.

See Also

rwsgn for draws from the wrapped law in its identified parameterisation.

Examples

```
set.seed(2)
x <- rsgn(5000, mu = 0, sigma = 1, beta = 1.3, lam1 = 0.3, lam2 = 0.8)
hist(x, breaks = 60, freq = FALSE, main = "")
curve(dsgn(x, 0, 1, 1.3, 0.3, 0.8), add = TRUE)
## mass left of the location: lam1 / (lam1 + lam2) = 0.2727
mean(x < 0)
```

---

rwsgn *Random generation from the wrapped skew generalised normal distribution*

Description

Draws from the circular law in the identified parameterisation of dwsgn, that is `rsgn(n, mu, sigma = sqrt(lam * (1 - lam)), beta, lam1 = lam, lam2 = 1 - lam, wrap = TRUE)`. The estimators fit_wsgn_trig and fit_wsgn_mle estimate exactly these `(mu, beta, lam)`.

Usage

```
rwsgn(n, mu = 0, beta = 2, lam = 0.5)
```

Arguments

n	number of draws.
mu	location (junction) parameter; for a skew law this is not the mean direction.
beta	shape (tail) parameter (> 0).
lam	skewness parameter in (0, 1); `0.5` is symmetric.

Value

a numeric vector of angles in `[0, 2 * pi)`.

Examples

```
set.seed(3)
th <- rwsgn(2000, mu = 1, beta = 1.5, lam = 0.3)
hist(th, breaks = seq(0, 2 * pi, length.out = 49), freq = FALSE, main = "",
     xlab = "angle (radians)")
curve(dwsgn(x, mu = 1, beta = 1.5, lam = 0.3), 0, 2 * pi, add = TRUE)
```

`trig_moments` *Population trigonometric moments of the wrapped family (C back-end)*

Description

Population trigonometric moments of the wrapped family (C back-end)

Usage

```
trig_moments(P = 2, mu = 0, beta = 2, lam = 0.5)
```

Arguments

`P`	highest order p.
`mu`	location (junction) parameter; for a skew law this is not the mean direction.
`beta`	shape (tail) parameter (> 0).
`lam`	skewness parameter in (0, 1); `0.5` is symmetric.

Value

a list with components `A` and `B`, the cosine and sine moments for `p = 1, ..., P`.

Examples

```
## the reduction to the wrapped normal in the trigonometric moments: exp(-p^2 / 4)
tm <- trig_moments(4, mu = 0, beta = 2, lam = 0.5)
rbind(engine = tm$A, closed_form = exp(-(1:4)^2 / 4))
## a skew member: the first moment against a large simulated sample
tm1 <- trig_moments(1, mu = 1, beta = 1.5, lam = 0.3)
set.seed(1)
th <- rwsgn(1e5, mu = 1, beta = 1.5, lam = 0.3)
rbind(population = c(tm1$A, tm1$B), sample = c(mean(cos(th)), mean(sin(th))))
```

Index

cfsgn, 2

dsgn, 3
dwsgn, 4, *7–9*

fit_wsgn_mle, 5, *9*
fit_wsgn_trig, 6, *9*

psgn, 6
pwsgn, 7

qwsgn *(*pwsgn*)*, 7

rsgn, 8
rwsgn, 9, *9*

trig_moments, 10