Package ‘sadists’

March 20, 2017

Maintainer  Steven E. Pav <shabbychef@gmail.com>
Version  0.2.3
Date  2017-03-19
License  LGPL-3
Title  Some Additional Distributions

BugReports  https://github.com/shabbychef/sadists/issues

Description  Provides the density, distribution, quantile and generation functions of some obscure probability distributions, including the doubly non-central t, F, Beta, and Eta distributions; the lambda-prime and K-prime; the upsilon distribution; the (weighted) sum of non-central chi-squares to a power; the (weighted) sum of log non-central chi-squares; the product of non-central chi-squares to powers; the product of doubly non-central F variables; the product of independent normals.

Depends  R (>= 3.0.2)
Imports  PDQutils (>= 0.1.1), hypergeo, orthopolynom
Suggests  shiny, testthat, ggplot2, xtable, knitr

URL  https://github.com/shabbychef/sadists

VignetteBuilder  knitr

Collate  ‘cumulants.r’ ‘dnbeta.r’ ‘dneta.r’ ‘dnt.r’ ‘kprime.r’
         ‘lambdap.r’ ‘moments.r’ ‘prochisqpow.r’ ‘prodnf.r’
         ‘prodnormal.r’ ‘runExample.r’ ‘sadists.r’ ‘sumchisqpow.r’
         ‘sumlogchisq.r’ ‘upsilon.r’ ‘utils.r’

RoxygenNote  5.0.1

NeedsCompilation  no

Author  Steven E. Pav [aut, cre]

Repository  CRAN

Date/Publication  2017-03-20 06:13:11 UTC
\textbf{R topics documented:}

\begin{verbatim}
drbeta .......................................................... 2
dbeta .......................................................... 4
dnf .............................................................. 6
dnt .............................................................. 7
kprime ......................................................... 9
lambda ......................................................... 11
prodchisqpow .................................................. 13
proddf ......................................................... 15
prodnormal ................................................... 17
runExample .................................................... 18
sadaits ....................................................... 20
sadaits-NEWS ............................................... 22
sumchisqpow ................................................... 23
sumlogchisq .................................................... 25
upsilon ....................................................... 26
\end{verbatim}

\textbf{Index} \hspace{1cm} 30

---

\textbf{dnbeta} \hfill \textit{The doubly non-central Beta distribution.}

\textbf{Description}

Density, distribution function, quantile function and random generation for the doubly non-central Beta distribution.

\textbf{Usage}

\begin{verbatim}
ddnbeta(x, df1, df2, ncp1, ncp2, log = FALSE, order.max=6)
pdnbeta(q, df1, df2, ncp1, ncp2, lower.tail = TRUE, log.p = FALSE, order.max=6)
qdnbeta(p, df1, df2, ncp1, ncp2, lower.tail = TRUE, log.p = FALSE, order.max=6)
rdnbeta(n, df1, df2, ncp1, ncp2)
\end{verbatim}

\textbf{Arguments}

\begin{verbatim}
\begin{itemize}
  \item \texttt{x, q} \hspace{1cm} vector of quantiles.
  \item \texttt{df1, df2} \hspace{1cm} the degrees of freedom for the numerator and denominator. We do \textit{not} recycle these versus the \texttt{x, q, p, n}.
  \item \texttt{ncp1, ncp2} \hspace{1cm} the non-centrality parameters for the numerator and denominator. We do \textit{not} recycle these versus the \texttt{x, q, p, n}.
  \item \texttt{log} \hspace{1cm} logical; if \texttt{TRUE}, densities \texttt{f} are given as \texttt{log(f)}.
\end{itemize}
\end{verbatim}
order.max  the order to use in the approximate density, distribution, and quantile computations, via the Gram-Charlier, Edeworth, or Cornish-Fisher expansion.

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

Details

Suppose \( x_i \sim \chi^2(\delta_i, \nu_i) \) be independent non-central chi-squares for \( i = 1, 2 \). Then

\[
Y = \frac{x_1}{x_1 + x_2}
\]

takes a doubly non-central Beta distribution with degrees of freedom \( \nu_1, \nu_2 \) and non-centrality parameters \( \delta_1, \delta_2 \).

Value

ddnbeta gives the density, pdnbeta gives the distribution function, qdnbeta gives the quantile function, and rdnbeta generates random deviates.

Invalid arguments will result in return value NaN with a warning.

Note

The PDF, CDF, and quantile function are approximated, via the Edgeworth or Cornish Fisher approximations, which may not be terribly accurate in the tails of the distribution. You are warned.

The distribution parameters are not recycled with respect to the \( x, p, q \) or \( n \) parameters, for, respectively, the density, distribution, quantile and generation functions. This is for simplicity of implementation and performance. It is, however, in contrast to the usual R idiom for dpqr functions.

Author(s)

Steven E. Pav <shabbychef@gmail.com>

See Also

(doubly non-central) F distribution functions, ddnf, pdnf, qdnf, rdnf.

Examples

```r
rv <- rdnbeta(500, df1=100, df2=500, ncp1=1.5, ncp2=12)
d1 <- ddnbeta(rv, df1=100, df2=500, ncp1=1.5, ncp2=12)
## Not run:
plot(rv,d1)

## End(Not run)
p1 <- ddnbeta(rv, df1=100, df2=500, ncp1=1.5, ncp2=12)
# should be nearly uniform:
## Not run:
```
The doubly non-central Eta distribution.

Description

Density, distribution function, quantile function and random generation for the doubly non-central Eta distribution.

Usage

\begin{verbatim}
ddneta(x, df, ncp1, ncp2, log = FALSE, order.max=6)
pdneta(q, df, ncp1, ncp2, lower.tail = TRUE, log.p = FALSE, order.max=6)
qdneta(p, df, ncp1, ncp2, lower.tail = TRUE, log.p = FALSE, order.max=6)
rdneta(n, df, ncp1, ncp2)
\end{verbatim}

Arguments

\begin{itemize}
\item \texttt{x, q} vector of quantiles.
\item \texttt{df} the degrees of freedom for the denominator chi square. We do \textit{not} recycle this versus the \texttt{x, q, p, n}.
\item \texttt{ncp1, ncp2} the non-centrality parameters for the numerator and denominator. We do \textit{not} recycle these versus the \texttt{x, q, p, n}.
\item \texttt{log} logical; if TRUE, densities \( f \) are given as \( \log(f) \).
\item \texttt{order.max} the order to use in the approximate density, distribution, and quantile computations, via the Gram-Charlier, Edeworth, or Cornish-Fisher expansion.
\item \texttt{p} vector of probabilities.
\item \texttt{n} number of observations.
\item \texttt{log.p} logical; if TRUE, probabilities \( p \) are given as \( \log(p) \).
\item \texttt{lower.tail} logical; if TRUE (default), probabilities are \( P[X \leq x] \), otherwise, \( P[X > x] \).
\end{itemize}
Details
Suppose $Z$ is a normal with mean $\delta_1$, independent of $X \sim \chi^2(\delta_2, \nu_2)$, a non-central chi-square with $\nu_2$ degrees of freedom and non-centrality parameter $\delta_2$. Then

$$Y = \frac{Z}{\sqrt{Z^2 + X}}$$

takes a doubly non-central Eta distribution with $\nu_2$ degrees of freedom and non-centrality parameters $\delta_1, \delta_2$. The square of a doubly non-central Eta is a doubly non-central Beta variate.

Value
dneta gives the density, pdneta gives the distribution function, qdneta gives the quantile function, and rdneta generates random deviates.
Invalid arguments will result in return value NaN with a warning.

Note
The PDF, CDF, and quantile function are approximated, via the Edgeworth or Cornish Fisher approximations, which may not be terribly accurate in the tails of the distribution. You are warned.
The distribution parameters are not recycled with respect to the x, p, q or n parameters, for, respectively, the density, distribution, quantile and generation functions. This is for simplicity of implementation and performance. It is, however, in contrast to the usual R idiom for dpqr functions.

Author(s)
Steven E. Pav <shabbychef@gmail.com>

See Also
(doubly non-central) t distribution functions, ddnt, pdnt, qdnt, rdnt.
(doubly non-central) Beta distribution functions, ddnbeta, pdnbeta, qdnbeta, rdnbeta.

Examples
```r
rv <- rdneta(500, df=100, ncp1=1.5, ncp2=12)
d1 <- ddneta(rv, df=100, ncp1=1.5, ncp2=12)
## Not run:
plot(rv,d1)

## End(Not run)
p1 <- ddneta(rv, df=100, ncp1=1.5, ncp2=12)
# should be nearly uniform:
## Not run:
plot(ecdf(p1))

## End(Not run)
q1 <- qdneta(ppoints(length(rv)), df=100, ncp1=1.5, ncp2=12)
## Not run:
qaplot(x=rv,y=q1)
```
The doubly non-central F distribution.

Description
Density, distribution function, quantile function and random generation for the doubly non-central F distribution.

Usage

- \( \text{ddnf}(x, df1, df2, ncp1, ncp2, \log = \text{FALSE}, \text{order.max}=6) \)
- \( \text{pdnf}(q, df1, df2, ncp1, ncp2, \text{lower.tail}=\text{TRUE}, \log.p = \text{FALSE}, \text{order.max}=6) \)
- \( \text{qdnf}(p, df1, df2, ncp1, ncp2, \text{lower.tail}=\text{TRUE}, \log.p = \text{FALSE}, \text{order.max}=6) \)
- \( \text{rdnf}(n, df1, df2, ncp1, ncp2) \)

Arguments

- \( x, q \) vector of quantiles.
- \( df1, df2 \) the degrees of freedom for the numerator and denominator. We do not recycle these versus the \( x, q, p, n \).
- \( ncp1, ncp2 \) the non-centrality parameters for the numerator and denominator. We do not recycle these versus the \( x, q, p, n \).
- \( \log \) logical; if TRUE, densities \( f \) are given as \( \log(f) \).
- \( \text{order.max} \) the order to use in the approximate density, distribution, and quantile computations, via the Gram-Charlier, Edeworth, or Cornish-Fisher expansion.
- \( p \) vector of probabilities.
- \( n \) number of observations.
- \( \log.p \) logical; if TRUE, probabilities \( p \) are given as \( \log(p) \).
- \( \text{lower.tail} \) logical; if TRUE (default), probabilities are \( P[X \leq x] \), otherwise, \( P[X > x] \).

Details

Suppose \( x_i \sim \chi^2(\delta_i, \nu_i) \) be independent non-central chi-squares for \( i = 1, 2 \). Then

\[
Y = \frac{x_1/\nu_1}{x_2/\nu_2}
\]

takes a doubly non-central F distribution with degrees of freedom \( \nu_1, \nu_2 \) and non-centrality parameters \( \delta_1, \delta_2 \).
The doubly non-central t distribution.

Density, distribution function, quantile function and random generation for the doubly non-central t distribution.
Usage

\texttt{ddnt}(x, df, ncp1, ncp2, log = FALSE, order.max=6)

\texttt{pdnt}(q, df, ncp1, ncp2, lower.tail = TRUE, log.p = FALSE, order.max=6)

\texttt{qdnt}(p, df, ncp1, ncp2, lower.tail = TRUE, log.p = FALSE, order.max=6)

\texttt{rdnt}(n, df, ncp1, ncp2)

Arguments

- \texttt{x, q} vector of quantiles.
- \texttt{df} the degrees of freedom for the denominator, \(\nu\). We do not recycle these versus the \(x, q, p, n\).
- \texttt{ncp1, ncp2} the non-centrality parameters for the numerator and denominator, respectively, \(\mu\) and \(\theta\). We do not recycle these versus the \(x, q, p, n\).
- \texttt{log} logical; if TRUE, densities \(f\) are given as \(\log(f)\).
- \texttt{order.max} the order to use in the approximate density, distribution, and quantile computations, via the Gram-Charlier, Edeworth, or Cornish-Fisher expansion.
- \texttt{p} vector of probabilities.
- \texttt{n} number of observations.
- \texttt{log.p} logical; if TRUE, probabilities \(p\) are given as \(\log(p)\).
- \texttt{lower.tail} logical; if TRUE (default), probabilities are \(P[X \leq x]\), otherwise, \(P[X > x]\).

Details

Let \(Z \sim \mathcal{N}(\mu, 1)\) independently of \(X \sim \chi^2(\theta, \nu)\). The random variable

\[
T = \frac{Z}{\sqrt{X/\nu}}
\]

takes a doubly non-central \(t\) distribution with parameters \(\nu, \mu, \theta\).

Value

\texttt{ddnt} gives the density, \texttt{pdnt} gives the distribution function, \texttt{qdnt} gives the quantile function, and \texttt{rdnt} generates random deviates.

Invalid arguments will result in return value \texttt{NaN} with a warning.

Note

The PDF, CDF, and quantile function are approximated, via the Edgeworth or Cornish Fisher approximations, which may not be terribly accurate in the tails of the distribution. You are warned.

The distribution parameters are not recycled with respect to the \(x, p, q\) or \(n\) parameters, for, respectively, the density, distribution, quantile and generation functions. This is for simplicity of implementation and performance. It is, however, in contrast to the usual R idiom for dpqr functions.
The K prime distribution.

Density, distribution function, quantile function and random generation for the K prime distribution.

Usage

\[
\begin{align*}
dkprime(x, v1, v2, a, b = 1, order.max=6, log = \text{FALSE}) \\
pkprime(q, v1, v2, a, b = 1, order.max=6, lower.tail = \text{TRUE}, log.p = \text{FALSE}) \\
qkprime(p, v1, v2, a, b = 1, order.max=6, lower.tail = \text{TRUE}, log.p = \text{FALSE}) \\
rkprime(n, v1, v2, a, b = 1)
\end{align*}
\]
Arguments

- **x, q**: vector of quantiles.
- **v1**: the degrees of freedom in the numerator chi-square. When (positive) infinite, we recover a non-central t distribution with v2 degrees of freedom and non-centrality parameter a, scaled by b. This is not recycled against the x, q, p, n.
- **v2**: the degrees of freedom in the denominator chi-square. When equal to infinity, we recover the Lambda prime distribution. This is not recycled against the x, q, p, n.
- **a**: the non-centrality scaling parameter. When equal to zero, we recover the (central) t distribution. This is not recycled against the x, q, p, n.
- **b**: the scaling parameter. This is not recycled against the x, q, p, n.
- **order.max**: the order to use in the approximate density, distribution, and quantile computations, via the Gram-Charlier, Edeworth, or Cornish-Fisher expansion.
- **log**: logical; if TRUE, densities f are given as log(f).
- **p**: vector of probabilities.
- **n**: number of observations.
- **log.p**: logical; if TRUE, probabilities p are given as log(p).
- **lower.tail**: logical; if TRUE (default), probabilities are P[X ≤ x], otherwise, P[X > x].

Details

Suppose \( y \sim \chi^2(\nu_1) \), and \( x \sim t(\nu_2, a\sqrt{y/\nu_1}/b) \). Then the random variable

\[
T = bx
\]

takes a K prime distribution with parameters \( \nu_1, \nu_2, a, b \). In Lecoutre’s terminology, \( T \sim K'_{\nu_1,\nu_2}(a, b) \)

Equivalently, we can think of

\[
T = \frac{bZ + a\sqrt{\chi^2_{\nu_1}/\nu_1}}{\sqrt{\chi^2_{\nu_2}/\nu_2}}
\]

where \( Z \) is a standard normal, and the normal and the (central) chi-squares are independent of each other. When \( a = 0 \) we recover a central t distribution; when \( \nu_1 = \infty \) we recover a rescaled non-central t distribution; when \( b = 0 \), we get a rescaled square root of a central F distribution; when \( \nu_2 = \infty \), we recover a Lambda prime distribution.

Value

dkprime gives the density, pkprime gives the distribution function, qkprime gives the quantile function, and rkprime generates random deviates.

Invalid arguments will result in return value NaN with a warning.

Note

The PDF, CDF, and quantile function are approximated, via the Edgeworth or Cornish Fisher approximations, which may not be terribly accurate in the tails of the distribution. You are warned.

The distribution parameters are not recycled with respect to the x, p, q or n parameters, for, respectively, the density, distribution, quantile and generation functions. This is for simplicity of implementation and performance. It is, however, in contrast to the usual R idiom for dpqr functions.
Author(s)

Steven E. Pav <shabbychef@gmail.com>

References


See Also
t distribution functions, dt, pt, qt, rt, lambda prime distribution functions, dlambdap, plambdap, qlambdap, rlambdap

Examples

d1 <- dkprime(1, 50, 20, a=0.01)
d2 <- dkprime(1, 50, 20, a=0.0001)
d3 <- dkprime(1, 50, 20, a=0)
d4 <- dkprime(1, 10000, 20, a=1)
d5 <- dkprime(1, Inf, 20, a=1)

lambda prime distribution.

Description

Density, distribution function, quantile function and random generation for the lambda prime distribution.

Usage

dlambdap(x, df, t, log = FALSE, order.max=6)

plambdap(q, df, t, lower.tail = TRUE, log.p = FALSE, order.max=6)

qlambdap(p, df, t, lower.tail = TRUE, log.p = FALSE, order.max=6)

rlambdap(n, df, t)
Arguments

- **x**, **q**: vector of quantiles.
- **df**: the degrees of freedom in the chi square. This is not recycled against the x, q, p, n.
- **t**: the scaling parameter on the chi. This is not recycled against the x, q, p, n.
- **log**: logical; if TRUE, densities f are given as log(f).
- **order.max**: the order to use in the approximate density, distribution, and quantile computations, via the Gram-Charlier, Edeworth, or Cornish-Fisher expansion.
- **p**: vector of probabilities.
- **n**: number of observations.
- **log.p**: logical; if TRUE, probabilities p are given as log(p).
- **lower.tail**: logical; if TRUE (default), probabilities are $P[X \leq x]$, otherwise, $P[X > x]$.

Details

Suppose $y \sim \chi^2(\nu)$, and $Z$ is a standard normal.

$$T = Z + t\sqrt{y/\nu}$$

takes a lambda prime distribution with parameters $\nu$, $t$. A lambda prime random variable can be viewed as a confidence level on a non-central t because

$$t = \frac{Z' + T}{\sqrt{y/\nu}}$$

Value

dlambdap gives the density, plambdap gives the distribution function, qlambdap gives the quantile function, and r1ambdap generates random deviates.

Invalid arguments will result in return value NaN with a warning.

Note

The PDF, CDF, and quantile function are approximated, via the Edgeworth or Cornish Fisher approximations, which may not be terribly accurate in the tails of the distribution. You are warned.

The distribution parameters are *not* recycled with respect to the x, p, q or n parameters, for, respectively, the density, distribution, quantile and generation functions. This is for simplicity of implementation and performance. It is, however, in contrast to the usual R idiom for dpqr functions.

Author(s)

Steven E. Pav <shabbychef@gmail.com>
prodchisqpow

References


See Also

t distribution functions, dt, pt, qt, rt, K prime distribution functions, dkprime, pkprime, qkprime, rkprime, upsilon distribution functions, dupsilon, pupsilon, qupsilon, rupsilon.

Examples

rv <- rlambda(100, 50, t=0.01)
d1 <- dlambda(1, 50, t=0.01)
pv <- plambda(rv, 50, t=0.01)
qv <- qlambda(ppoints(length(rv)), 50, t=1)

---

prodchisqpow The product of (non-central) chi-squares raised to powers distribution.

Description

Density, distribution function, quantile function and random generation for the distribution of the product of non-central chi-squares taken to powers.

Usage

dprodchisqpow(x, df, ncp=0, pow=1, log = FALSE, order.max=5)
pprodchisqpow(q, df, ncp=0, pow=1, lower.tail = TRUE, log.p = FALSE, order.max=5)
qprodchisqpow(p, df, ncp=0, pow=1, lower.tail = TRUE, log.p = FALSE, order.max=5)
rprodchisqpow(n, df, ncp=0, pow=1)

Arguments

x, q vector of quantiles.
df the vector of degrees of freedom. This is recycled against the ncp, pow, but not against the x,q,p,n.
cp the vector of non-centrality parameters. This is recycled against the df, pow, but not against the x,q,p,n.
prodchisqpow

pow the vector of the power parameters. This is recycled against the df, ncp, but not against the x, q, p, n.

log logical; if TRUE, densities f are given as log(f).

order.max the order to use in the approximate density, distribution, and quantile computations, via the Gram-Charlier, Edeworth, or Cornish-Fisher expansion.

p vector of probabilities.

n number of observations.

log.p logical; if TRUE, probabilities p are given as log(p).

lower.tail logical; if TRUE (default), probabilities are \( P[X \leq x] \), otherwise, \( P[X > x] \).

Details

Let \( X_i \sim \chi^2(\delta_i, \nu_i) \) be independently distributed non-central chi-squares, where \( \nu_i \) are the degrees of freedom, and \( \delta_i \) are the non-centrality parameters. Let \( p_i \) be given constants. Suppose

\[
Y = \prod_i X_i^{p_i}.
\]

Then \( Y \) follows a product of chi-squares to power distribution.

Value
dprodchisqpow gives the density, pprodchisqpow gives the distribution function, qprodchisqpow gives the quantile function, and rprodchisqpow generates random deviates.

Invalid arguments will result in return value NaN with a warning.

Note

The PDF, CDF, and quantile function are approximated, via the Edgeworth or Cornish Fisher approximations, which may not be terribly accurate in the tails of the distribution. You are warned.

The distribution parameters are not recycled with respect to the x, p, q or n parameters, for, respectively, the density, distribution, quantile and generation functions. This is for simplicity of implementation and performance. It is, however, in contrast to the usual R idiom for dpqr functions.

The PDQ functions are computed by translation of the sum of log chi-squares distribution functions.

Author(s)

Steven E. Pav <shabbychef@gmail.com>

References


See Also

The sum of log of chi-squares distribution, dsumlogchisq, psumlogchisq, qsumlogchisq, rsumlogchisq. The upsilon distribution, dupsilon, pupsilon, qupsilon, rupsilon. The sum of chi-square powers distribution, dsumchisqpow, psumchisqpow, qsumchisqpow, rsumchisqpow.
Examples

```r
df <- c(100, 20, 10)
cp <- c(5, 3, 1)
pow <- c(1, 0.5, 1)
rvs <- rprodchisqpow(128, df, ncp, pow)
dvs <- dprodchisqpow(rvs, df, ncp, pow)
qvs <- pprodchisqpow(rvs, df, ncp, pow)
pvs <- qprodchisqpow(ppoints(length(rvs)), df, ncp, pow)
```

proddnf

The product of multiple doubly non-central F's distribution.

Description

Density, distribution function, quantile function and random generation for the product of multiple independent doubly non-central F variates.

Usage

```r
dproddnf(x, df1, df2, ncp1, ncp2, log = FALSE, order.max=4)
pproddnf(q, df1, df2, ncp1, ncp2, lower.tail = TRUE, log.p = FALSE, order.max=4)
qproddnf(p, df1, df2, ncp1, ncp2, lower.tail = TRUE, log.p = FALSE, order.max=4)
rproddnf(n, df1, df2, ncp1, ncp2)
```

Arguments

- `x, q` vector of quantiles.
- `df1, df2` the vectors of the degrees of freedom for the numerator and denominator. We do not recycle these versus the `x, q, p, n`.
- `ncp1, ncp2` the vectors of the non-centrality parameters for the numerator and denominator. We do not recycle these versus the `x, q, p, n`.
- `log` logical; if TRUE, densities `f` are given as log(`f`).
- `order.max` the order to use in the approximate density, distribution, and quantile computations, via the Gram-Charlier, Edeworth, or Cornish-Fisher expansion.
- `p` vector of probabilities.
- `n` number of observations.
- `log.p` logical; if TRUE, probabilities `p` are given as log(`p`).
- `lower.tail` logical; if TRUE (default), probabilities are `P[X ≤ x]`, otherwise, `P[X > x]`. 
Details

Let

\[ x_j \sim F(\delta_{i,j}, \delta_{2,j}, \nu_{i,j}, \nu_{2,j}) \]

be independent doubly non-central F variates with non-centrality parameters \( \delta_{i,j} \) and degrees of freedom \( \nu_{i,j} \) for \( i = 1, 2, \ldots, I \) and \( j = 1, 2 \). Then

\[ Y = \prod_{j} x_j \]

takes a product of doubly non-central F's distribution. We take the parameters of this distribution as the four \( I \) length vectors of the two degrees of freedom and the two non-centrality parameters.

Value

dproddnf gives the density, pproddnf gives the distribution function, qproddnf gives the quantile function, and rproddnf generates random deviates.

Invalid arguments will result in return value NaN with a warning.

Note

The PDF, CDF, and quantile function are approximated, via the Edgeworth or Cornish Fisher approximations, which may not be terribly accurate in the tails of the distribution. You are warned.

The distribution parameters are not recycled with respect to the \( x, \ p, \ q \) or \( n \) parameters, for, respectively, the density, distribution, quantile and generation functions. This is for simplicity of implementation and performance. It is, however, in contrast to the usual R idiom for dpqr functions.

The PDQ functions are computed by translation of the sum of log chi-squares distribution functions.

Author(s)

Steven E. Pav <shabbychef@gmail.com>

References


See Also

The sum of log of chi-squares distribution, dsumlogchisq, psumlogchisq, qsumlogchisq, rsumlogchisq.

Examples

```r
df1 <- c(10, 20, 5)
df2 <- c(1000, 500, 150)
cp1 <- c(1, 0, 2.5)
cp2 <- c(0, 1.5, 5)
rv <- rproddnf(500, df1=df1, df2=df2, ncp1=ncp1, ncp2=ncp2)
```
prodnormal

The product of normal random variates.

Description

Density, distribution function, quantile function and random generation for the distribution of the product of independent normal random variables.

Usage

dprodnormal(x, mu, sigma, log = FALSE, order.max=5)
pprodnormal(q, mu, sigma, lower.tail = TRUE, log.p = FALSE, order.max=5)
qprodnormal(p, mu, sigma, lower.tail = TRUE, log.p = FALSE, order.max=5)
rprodnormal(n, mu, sigma)

Arguments

x, q vector of quantiles.
mu the vector of means. This is recycled against the sigma, but not against the
sigma the vector of standard deviations. This is recycled against the mu, but not against the
log logical; if TRUE, densities \( f \) are given as \( \log(f) \).
order.max the order to use in the approximate density, distribution, and quantile computations, via the Gram-Charlier, Edeworth, or Cornish-Fisher expansion.
p vector of probabilities.
n number of observations.
log.p logical; if TRUE, probabilities p are given as \( \log(p) \).
lower.tail logical; if TRUE (default), probabilities are \( P[X \leq x] \), otherwise, \( P[X > x] \).

Details

Let \( Z_i \sim \mathcal{N}(\mu_i, \sigma_i^2) \) be independently distributed normal variates, with means \( \mu_i \) and variances \( \sigma_i^2 \). Suppose

\[
Y = \prod_i Z_i.
\]

Then \( Y \) follows a product of normals distribution.

Value

dprodnormal gives the density, pprodnormal gives the distribution function, qprodnormal gives the quantile function, and rprodnormal generates random deviates.
Invalid arguments will result in return value NaN with a warning.

Note

The PDF, CDF, and quantile function are approximated, via the Edgeworth or Cornish Fisher approximations, which may not be terribly accurate in the tails of the distribution. You are warned.

The distribution parameters are not recycled with respect to the \( x, p, q \) or \( n \) parameters, for, respectively, the density, distribution, quantile and generation functions. This is for simplicity of implementation and performance. It is, however, in contrast to the usual R idiom for dpqr functions.

Author(s)

Steven E. Pav <shabbychef@gmail.com>

Examples

```r
mu <- c(100,20,10)
sigma <- c(10,50,10)
rvs <- rprodnormal(128, mu, sigma)
dvs <- dprodnormal(rvs, mu, sigma)
qvs <- pprodnormal(rvs, mu, sigma)
pvs <- qprodnormal(ppoints(length(rvs)), mu, sigma)
```

---

**runExample**

**Run Shiny Application**

Description

Runs a shiny application which draws from the given distributions, then illustrates the fidelity of the density, CDF, and quantile functions.
Usage

```runExample(port=NULL, launch.browser=TRUE, 
        host=getOption('shiny.host','127.0.0.1'), display.mode='auto')
```

Arguments

- **port**
  The TCP port that the application should listen on. If the port is not specified, and the `shiny.port` option is set (with `options(shiny.port = XX)`), then that port will be used. Otherwise, use a random port.

- **launch.browser**
  If true, the system’s default web browser will be launched automatically after the app is started. Defaults to true in interactive sessions only. This value of this parameter can also be a function to call with the application’s URL.

- **host**
  The IPv4 address that the application should listen on. Defaults to the `shiny.host` option, if set, or "127.0.0.1" if not. See Details.

- **display.mode**
  The mode in which to display the application. If set to the value "showcase", shows application code and metadata from a DESCRIPTION file in the application directory alongside the application. If set to "normal", displays the application normally. Defaults to "auto", which displays the application in the mode given in its DESCRIPTION file, if any.

Details

Launches shiny applications, and optionally, your system’s web browser. Draws are taken from the random variable, and d-d, q-q, and p-p plots are available.

Author(s)

Steven E. Pav <shabbychef@gmail.com>

References


Examples

```r
## Not run:
runExample(launch.browser=TRUE)

## End(Not run)```
Some Additional Distributions

Description
Some Additional Distributions.

Details
A collection of distributions which can be approximated via Edgeworth and Cornish-Fisher expansions

Sum of (non-central) chi-square to powers
Let \( X_i \sim \chi^2(\delta_i, \nu_i) \) be independently distributed non-central chi-squares, where \( \nu_i \) are the degrees of freedom, and \( \delta_i \) are the non-centrality parameters. Let \( w_i \) and \( p_i \) be given constants. Suppose

\[
Y = \sum_i w_i X_i^{p_i}.
\]

Then \( Y \) follows a weighted sum of chi-squares to power distribution. The special case where all the \( p_i \) are one is a 'sum of chi-squares' distribution; The special case where all the \( p_i \) are one half is a 'sum of chis' distribution;

Lambda Prime
Introduced by Lecoutre, the lambda prime distribution finds use in inference on the Sharpe ratio under normal returns. Suppose \( y \sim \chi^2(\nu) \), and \( Z \) is a standard normal.

\[
T = Z + t \sqrt{y/\nu}
\]

takes a lambda prime distribution with parameters \( \nu, t \). A lambda prime random variable can be viewed as a confidence variable on a non-central \( t \) because

\[
t = \frac{Z' + T}{\sqrt{y/\nu}}
\]

Upsilon
The upsilon distribution generalizes the lambda prime to the case of the sum of multiple chi variables. That is, suppose \( y_i \sim \chi^2(\nu_i) \) independently and independently of \( Z \), a standard normal. Then

\[
T = Z + \sum_i t_i \sqrt{y_i/\nu_i}
\]

takes an upsilon distribution with parameter vectors \([\nu_1, \nu_2, \ldots, \nu_k]', [t_1, t_2, \ldots, t_k]'\). The upsilon distribution is used in certain tests of the Sharpe ratio for independent observations.
K Prime

Introduced by Lecoutre, the K prime family of distributions generalize the (singly) non-central t, and lambda prime distributions. Suppose $y \sim \chi^2(\nu_1)$, and $x \sim t(\nu_2, a\sqrt{y/\nu_1}/b)$. Then the random variable

$$T = bx$$

takes a K prime distribution with parameters $\nu_1, \nu_2, a, b$. In Lecoutre’s terminology, $T \sim K'_{\nu_1,\nu_2}(a, b)$

Equivalently, we can think of

$$T = \frac{bZ + a\sqrt{\chi^2/\nu_1}}{\sqrt{\chi^2/\nu_2}}$$

where $Z$ is a standard normal, and the normal and the (central) chi-squares are independent of each other. When $a = 0$ we recover a central t distribution; when $\nu_1 = \infty$ we recover a rescaled non-central t distribution; when $b = 0$, we get a rescaled square root of a central F distribution; when $\nu_2 = \infty$, we recover a Lambda prime distribution.

Doubly Noncentral t

The doubly noncentral t distribution generalizes the (singly) noncentral t distribution to the case where the numerator is the square root of a scaled noncentral chi-square distribution. That is, if $X \sim \mathcal{N}(\mu, 1)$ independently of $Y \sim \chi^2(k, \theta)$, then the random variable

$$T = \frac{X}{\sqrt{Y/k}}$$

takes a doubly non-central t distribution with parameters $k, \mu, \theta$.

Doubly Noncentral F

The doubly noncentral F distribution generalizes the (singly) noncentral F distribution to the case where the numerator is a scaled noncentral chi-square distribution. That is, if $X \sim \chi^2(n_1, \theta_1)$ independently of $Y \sim \chi^2(n_2, \theta_2)$, then the random variable

$$T = \frac{X/n_1}{Y/n_2}$$

takes a doubly non-central F distribution with parameters $n_1, n_2, \theta_1, \theta_2$.

Parameter recycling

It should be noted that the functions provided by sadists do not recycle their distribution parameters against the $x, p, q$ or $n$ parameters. This is in contrast to the common R idiom, and may cause some confusion. This is mostly for reasons of performance, but also because some of the distributions have vector-valued parameters; recycling over these would require the user to provide lists of parameters, which would be unpleasant.

Legal Mumbo Jumbo

sadists is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.
Note

This package is maintained as a hobby.

Author(s)

Steven E. Pav <shabbychef@gmail.com>

References


Description

History of the 'sadists' package.

Version 0.2.3 (2017-03-19)

- add product of normals distribution.
- move github figures to location CRAN understands.

Version 0.2.2 (2016-03-03)

- work around bad rchisq when df = 0 = ncp (?)
- incompatibilities in vignette with ggeplot2 release.

Version 0.2.1 (2015-06-12)

- shiny app (h/t Dean Attali).

Version 0.2.0 (2015-04-01)

- add doubly non-central Beta and Eta distributions.
- add (sum of) log chi-square distribution.
- have products of chi-square depend on transform of this distribution.
sumchisqpow

Initial Version 0.1.0 (2015-03-07)

• first CRAN release.

sumchisqpow

The sum of (non-central) chi-squares raised to powers distribution.

Description

Density, distribution function, quantile function and random generation for the distribution of the weighted sum of non-central chi-squares taken to powers.

Usage

dsumchisqpow(x, wts, df, ncp=0, pow=1, log = FALSE, order.max=6)
psumchisqpow(q, wts, df, ncp=0, pow=1, lower.tail = TRUE, log.p = FALSE, order.max=6)
qsumchisqpow(p, wts, df, ncp=0, pow=1, lower.tail = TRUE, log.p = FALSE, order.max=6)
rsumchisqpow(n, wts, df, ncp=0, pow=1)

Arguments

x, q
vector of quantiles.
wts
the vector of weights. This is recycled against the df, ncp, pow, but not against the x, q, p, n.
df
the vector of degrees of freedom. This is recycled against the wts, ncp, pow, but not against the x, q, p, n.
ncp
the vector of non-centrality parameters. This is recycled against the wts, df, pow, but not against the x, q, p, n.
pow
the vector of the power parameters. This is recycled against the wts, df, ncp, but not against the x, q, p, n.
log
logical; if TRUE, densities f are given as log(f).
order.max
the order to use in the approximate density, distribution, and quantile computations, via the Gram-Charlier, Edeworth, or Cornish-Fisher expansion.
p
vector of probabilities.
n
number of observations.
log.p
logical; if TRUE, probabilities p are given as log(p).
lower.tail
logical; if TRUE (default), probabilities are P[X ≤ x], otherwise, P[X > x].
Details

Let $X_i \sim \chi^2(\delta_i, \nu_i)$ be independently distributed non-central chi-squares, where $\nu_i$ are the degrees of freedom, and $\delta_i$ are the non-centrality parameters. Let $w_i$ and $p_i$ be given constants. Suppose

$$Y = \sum_i w_i X_i^{p_i}.$$ 

Then $Y$ follows a weighted sum of chi-squares to power distribution.

Value

dsumchisqpow gives the density, psumchisqpow gives the distribution function, qsumchisqpow gives the quantile function, and rsumchisqpow generates random deviates.

Invalid arguments will result in return value NaN with a warning.

Note

The PDF, CDF, and quantile function are approximated, via the Edgeworth or Cornish Fisher approximations, which may not be terribly accurate in the tails of the distribution. You are warned.

The distribution parameters are not recycled with respect to the x, p, q or n parameters, for, respectively, the density, distribution, quantile and generation functions. This is for simplicity of implementation and performance. It is, however, in contrast to the usual R idiom for dpqr functions.

The 'sum of chisquare power' distribution does not generalize the 'chi-bar-square' distribution, whose density is the sum of chi-square densities.

Author(s)

Steven E. Pav <shabbychef@gmail.com>

See Also

The upsilon distribution, dupsilon, pupsilon, qupsilon, rupsilon.

Examples

```r
wts <- c(1, -3, 4)
df <- c(100, 20, 10)
cnp <- c(5, 3, 1)
pow <- c(1, 0.5, 1)
rvs <- rsumchisqpow(128, wts, df, cnp, pow)
dvs <- dsumchisqpow(rvs, wts, df, cnp, pow)
qvs <- psumchisqpow(rvs, wts, df, cnp, pow)
pvs <- qsumchisqpow(ppoints(length(rvs)), wts, df, cnp, pow)
```
Description

Density, distribution function, quantile function and random generation for the distribution of the weighted sum of logs of non-central chi-squares.

Usage

dsumlogchisq(x, wts, df, ncp=0, log = FALSE, order.max=6)
psumlogchisq(q, wts, df, ncp=0, lower.tail = TRUE, log.p = FALSE, order.max=6)
qsumlogchisq(p, wts, df, ncp=0, lower.tail = TRUE, log.p = FALSE, order.max=6)
rsumlogchisq(n, wts, df, ncp=0)

Arguments

x, q vector of quantiles.
wts the vector of weights. This is recycled against the df, ncp, but not against the x, q, p, n.
df the vector of degrees of freedom. This is recycled against the wts, ncp, but not against the x, q, p, n.
ccp the vector of non-centrality parameters. This is recycled against the wts, df, but not against the x, q, p, n.
log logical; if TRUE, densities f are given as log(f).
order.max the order to use in the approximate density, distribution, and quantile computations, via the Gram-Charlier, Edeworth, or Cornish-Fisher expansion.
p vector of probabilities.
n number of observations.
log.p logical; if TRUE, probabilities p are given as log(p).
lower.tail logical; if TRUE (default), probabilities are \( P[X \leq x] \), otherwise, \( P[X > x] \).

Details

Let \( X_i \sim \chi^2(\delta_i, \nu_i) \) be independently distributed non-central chi-squares, where \( \nu_i \) are the degrees of freedom, and \( \delta_i \) are the non-centrality parameters. Let \( w_i \) be given constants. Suppose

\[
Y = \sum_i w_i \log X_i.
\]

Then \( Y \) follows a weighted sum of log of chi-squares distribution.
Value

dsumlogchisq gives the density, psumlogchisq gives the distribution function, qsumlogchisq gives the quantile function, and rsumlogchisq generates random deviates.

Invalid arguments will result in return value NaN with a warning.

Note

The PDF, CDF, and quantile function are approximated, via the Edgeworth or Cornish Fisher approximations, which may not be terribly accurate in the tails of the distribution. You are warned.

The distribution parameters are not recycled with respect to the x, p, q or n parameters, for, respectively, the density, distribution, quantile and generation functions. This is for simplicity of implementation and performance. It is, however, in contrast to the usual R idiom for dpqr functions.

Author(s)

Steven E. Pav <shabbychef@gmail.com>

References


See Also

The product of chi-squares to a power, dprodchisqpow, pprodchisqpow, qprodchisqpow, rprodchisqpow.

Examples

wts <- c(1, -3, 4)
df <- c(100, 20, 10)
ncp <- c(5, 3, 1)
rvs <- rsumlogchisq(128, wts, df, ncp)
dvs <- dsumlogchisq(rvs, wts, df, ncp)
qvs <- psumlogchisq(rvs, wts, df, ncp)
pvs <- qsumlogchisq(ppoints(length(rvs)), wts, df, ncp)

DESCRIPTION

Density, distribution function, quantile function and random generation for the upsilon distribution.
Usage

dupsilon(x, df, t, log = FALSE, order.max=6)
pupsilon(q, df, t, lower.tail = TRUE, log.p = FALSE, order.max=6)
qupsilon(p, df, t, lower.tail = TRUE, log.p = FALSE, order.max=6)
rupsilon(n, df, t)

Arguments

x, q vector of quantiles.
df the degrees of freedom in the chi square. a vector. we do not vectorize over this variable.
t the scaling parameter on the chi. a vector. should be the same length as df. we do not vectorize over this variable.
log logical; if TRUE, densities \( f \) are given as \( \log(f) \).
order.max the order to use in the approximate density, distribution, and quantile computations, via the Gram-Charlier, Edeworth, or Cornish-Fisher expansion.
p vector of probabilities.
n number of observations.
log.p logical; if TRUE, probabilities \( p \) are given as \( \log(p) \).
lower.tail logical; if TRUE (default), probabilities are \( P[X \leq x] \), otherwise, \( P[X > x] \).

Details

Suppose \( x_i \sim \chi^2(\nu_i) \) independently and independently of \( Z \), a standard normal. Then

\[
\Upsilon = Z + \sum_i t_i \sqrt{x_i / \nu_i}
\]

takes an upsilon distribution with parameter vectors \([\nu_1, \nu_2, \ldots, \nu_k]'\), \([t_1, t_2, \ldots, t_k]'\).

The upsilon distribution is used in certain tests of the Sharpe ratio for independent observations, and generalizes the lambda prime distribution, which can be written as \( Z + t \sqrt{x / \nu} \).

Value

dupsilon gives the density, pupsilon gives the distribution function, qupsilon gives the quantile function, and rupsilon generates random deviates.

Invalid arguments will result in return value NaN with a warning.

Note

the PDF and CDF are approximated by an Edgeworth expansion; the quantile function is approximated by a Cornish-Fisher expansion.
The PDF, CDF, and quantile function are approximated, via the Edgeworth or Cornish Fisher approximations, which may not be terribly accurate in the tails of the distribution. You are warned.

The distribution parameters are not recycled with respect to the \( x, \ p, \ q \) or \( n \) parameters, for, respectively, the density, distribution, quantile and generation functions. This is for simplicity of implementation and performance. It is, however, in contrast to the usual R idiom for \( \text{dpqr} \) functions.

**Author(s)**

Steven E. Pav <shabbychef@gmail.com>

**References**


**See Also**

lambda-prime distribution functions, \( \text{dlambdap, plambdap, qlambdap, rlambdap} \). Sum of chi-squares to power distribution functions, \( \text{dsumchisqpow, psumchisqpow, qsumchisqpow, rsumchisqpow} \).

**Examples**

```r
mydf <- c(100,30,50)
myt <- c(-1,3,5)
rv <- rupsilon(500, df=mydf, t=myt)
d1 <- dupsilon(rv, df=mydf, t=myt)
## Not run:
plot(rv,d1)

## End(Not run)
p1 <- pupsilon(rv, df=mydf, t=myt)
# should be nearly uniform:
## Not run:
plot(ecdf(p1))

## End(Not run)
q1 <- qupsilon(ppoints(length(rv)),df=mydf,t=myt)
## Not run:
qqplot(x=rv,y=q1)

## End(Not run)
## Not run:
require(SharpeR)
ope <- 252
n.sim <- 500
```
n.term <- 3
set.seed(234234)
pp <- replicate(n.sim,
               # these are population parameters
               a <- rnorm(n.term)
               psi <- 6 * rnorm(length(a)) / sqrt(ope)
               b <- sum(a * psi)
               df <- 100 + ceiling(200 * runif(length(psi)))
               comm <- 1 / sqrt(sum(a^2 / df))
               cdf <- df - 1
               # now independent draws from the SR distribution:
               x <- rsr(length(df), df, zeta=psi, ope=1)
               # now compute a p-value under the true null
               pupsilon(comm * b, df=cdf, t=comm*a*x)
)
# ought to be uniform:
plot(ecdf(pp))

## End(Not run)
Index

*Topic **distribution**
  - dnbeta, 2
  - dneta, 4
  - dnf, 6
  - dnt, 7
  - kprime, 9
  - lambdap, 11
  - prodcisqpow, 13
  - prodnfn, 15
  - prodnormal, 17
  - sumchisqpow, 23
  - sumlogchisq, 25
  - upsilon, 26

*Topic **package**
  - sadists, 20

ddnbeta, 5
ddnbeta (dnbeta), 2
ddneta (dneta), 4
ddnf, 3, 16
ddnf (dnf), 6
ddnt, 5
ddnt (dnt), 7
df, 7
dkprime, 13
dkprime (kprime), 9
dlambdap, 11, 28
dlambdap (lambdap), 11
dnbeta, 2
dneta, 4
dnf, 6
dnt, 7
dprodchisqpow, 26
dprodchisqpow (prodchisqpow), 13
dprodnfn (prodnfn), 15
dprodnormal (prodnormal), 17
dsumchisqpow, 14, 28
dsumchisqpow (sumchisqpow), 23
dsumlogchisq, 14, 16
dsumlogchisq (sumlogchisq), 25
dt, 9, 11, 13
dupsilon, 13, 14, 24
dupsilon (upsilon), 26
kprime, 9
lambdap, 11
plambdap, 11
plambdap (lambdap), 11
pprodcisqpow, 26
pprodcisqpow (prodchisqpow), 13
pprodnfn (prodnfn), 15
pprodnormal (prodnormal), 17
prodchisqpow, 13
prodnfn, 15
prodnormal, 17
psumchisqpow, 14, 28
psumchisqpow (sumchisqpow), 23
psumlogchisq, 14, 16
psumlogchisq (sumlogchisq), 25
pt, 9, 11, 13
pupsilon, 13, 14, 24
pupsilon (upsilon), 26
qdnbeta, 5
qdnbeta (dnbeta), 2
qdneta (dneta), 4
qdnf, 3, 16
qdnf (dnf), 6
INDEX

qdnt, 5
qdnt (dnt), 7
qf, 7
qkprime, 13
qkprime (kprime), 9
qlambdap, 11, 28
qlambdap (lambdap), 11
aprodchisqpow, 26
aprodchisqpow (prodchisqpow), 13
aprodnormal (prodnormal), 17
qsumchisqpow, 14, 28
qsumchisqpow (sumchisqpow), 23
qsumlogchisq, 14, 16
qsumlogchisq (sumlogchisq), 25
qt, 9, 11, 13
qupsilon, 13, 14, 24
qupsilon (upsilon), 26

rdnbeta, 5
rdnbeta (dnbeta), 2
rdneta (dneta), 4
rdnf, 3, 16
rdnf (dnf), 6
rdnt, 5
rdnt (dnt), 7
rf, 7
rkprime, 13
rkprime (kprime), 9
rlambdap, 11, 28
rlambdap (lambdap), 11
rprodchisqpow, 26
rprodchisqpow (prodchisqpow), 13
rprodnormal (prodnormal), 17
rsumchisqpow, 14, 28
rsumchisqpow (sumchisqpow), 23
rsumlogchisq, 14, 16
rsumlogchisq (sumlogchisq), 25
rt, 9, 11, 13
runExample, 18
rupsilon, 13, 14, 24
rupsilon (upsilon), 26

sadists, 20
sadists-NEWS, 22
sadists-package (sadists), 20
sumchisqpow, 23
sumlogchisq, 25