Package ‘invgamma’

October 13, 2022

Type Package
Title The Inverse Gamma Distribution
Version 1.1
URL https://github.com/dkahle/invgamma
BugReports https://github.com/dkahle/invgamma/issues
Description Light weight implementation of the standard distribution
functions for the inverse gamma distribution, wrapping those for the gamma
distribution in the stats package.
License GPL-2
RoxygenNote 6.0.1
NeedsCompilation no
Author David Kahle [aut, cre, cph],
James Stamey [aut, cph]
Maintainer David Kahle <david.kahle@gmail.com>
Repository CRAN
Date/Publication 2017-05-07 05:22:52 UTC

R topics documented:

invchisq ....................................................... 2
invexp .......................................................... 3
invgamma ....................................................... 4

Index 6
The Inverse (non-central) Chi-Squared Distribution

Description
Density, distribution function, quantile function and random generation for the inverse chi-squared distribution.

Usage

dinvchisq(x, df, ncp = 0, log = FALSE)
pinvchisq(q, df, ncp = 0, lower.tail = TRUE, log.p = FALSE)
qinvchisq(p, df, ncp = 0, lower.tail = TRUE, log.p = FALSE)
rintvchisq(n, df, ncp = 0)

Arguments

x, q         vector of quantiles.
df            degrees of freedom (non-negative, but can be non-integer).
ncp          non-centrality parameter (non-negative).
log, log.p  logical; if TRUE, probabilities p are given as log(p).
lower.tail   logical; if TRUE (default), probabilities are P[X <= x] otherwise, P[X > x].
p           vector of probabilities.
n            number of observations. If length(n) > 1, the length is taken to be the number required.

Details
The functions (d/p/q/r)invchisq simply wrap those of the standard (d/p/q/r)chisq R implementation, so look at, say, dchisq for details.

See Also
dchisq; these functions just wrap the (d/p/q/r)chisq functions.

Examples

s <- seq(0, 3, .01)
plot(s, dinvchisq(s, 3), type = 'l')
f <- function(x) dinvchisq(x, 3)
q <- 2
```r
f <- function(x) dinvchisq(x, 3, ncp = 2)
q <- 1.5
integrate(f, 0, q)
(p <- pinvchisq(q, 3, ncp = 2))
qinvchisq(p, 3, ncp = 2) # = q
mean(rinvchisq(1e7, 3, ncp = 2) <= q)
```

---

### invexp

**The Inverse Exponential Distribution**

#### Description

Density, distribution function, quantile function and random generation for the inverse exponential distribution.

#### Usage

- `dinvexp(x, rate = 1, log = FALSE)`
- `pinvexp(q, rate = 1, lower.tail = TRUE, log.p = FALSE)`
- `qinvexp(p, rate = 1, lower.tail = TRUE, log.p = FALSE)`
- `rinvexp(n, rate = 1)`

#### Arguments

- `x, q` vector of quantiles.
- `rate` degrees of freedom (non-negative, but can be non-integer).
- `log, log.p` logical; if TRUE, probabilities `p` are given as log(p).
- `lower.tail` logical; if TRUE (default), probabilities are P[X <= x] otherwise, P[X > x].
- `p` vector of probabilities.
- `n` number of observations. If length(n) > 1, the length is taken to be the number required.
The functions \((d/p/q/r)\text{invexp}\) simply wrap those of the standard \((d/p/q/r)\text{exp}\) R implementation, so look at, say, \texttt{dexp} for details.

\section*{See Also}
\textit{dexp}: these functions just wrap the \((d/p/q/r)\text{exp}\) functions.

\section*{Examples}
\begin{verbatim}
s <- seq(0, 10, .01)
plot(s, dinvexp(s, 2), type = 'l')

f <- function(x) dinvexp(x, 2)
q <- 3
integrate(f, 0, q)
(p <- pinvexp(q, 2))
qinvexp(p, 2) # = q
mean(rinvexp(1e5, 2) <= q)

pinvgamma(q, 1, 2)
\end{verbatim}

\section*{Description}
Density, distribution function, quantile function and random generation for the inverse gamma distribution.

\section*{Usage}
\begin{verbatim}
dinvgamma(x, shape, rate = 1, scale = 1/rate, log = FALSE)

pinvgamma(q, shape, rate = 1, scale = 1/rate, lower.tail = TRUE,
          log.p = FALSE)

qinvgamma(p, shape, rate = 1, scale = 1/rate, lower.tail = TRUE,
          log.p = FALSE)

rinvgamma(n, shape, rate = 1, scale = 1/rate)
\end{verbatim}
Arguments

- **x, q**: vector of quantiles.
- **shape**: inverse gamma shape parameter
- **rate**: inverse gamma rate parameter
- **scale**: alternative to rate; scale = 1/rate
- **log, log.p**: logical; if TRUE, probabilities p are given as log(p).
- **lower.tail**: logical; if TRUE (default), probabilities are P[X <= x] otherwise, P[X > x].
- **p**: vector of probabilities.
- **n**: number of observations. If length(n) > 1, the length is taken to be the number required.

Details

The inverse gamma distribution with parameters shape and rate has density

\[ f(x) = \frac{\text{rate}^{\text{shape}}}{\Gamma(\text{shape})} x^{-(1+\text{shape})} e^{-\text{rate}/x} \]

it is the inverse of the standard gamma parameterization in R.

The functions (d/p/q/r)invgamma simply wrap those of the standard (d/p/q/r)gamma R implementation, so look at, say, `dgamma` for details.

See Also

- `dgamma`: these functions just wrap the (d/p/q/r)gamma functions.

Examples

```r
s <- seq(0, 5, .01)
plot(s, dinvgamma(s, 7, 10), type = 'l')

f <- function(x) dinvgamma(x, 7, 10)
q <- 2
integrate(f, 0, q)
(p <- pinvgamma(q, 7, 10))
qinvgamma(p, 7, 10) # = q
mean(rinvgamma(1e5, 7, 10) <= q)
```
Index

dchisq, 2
dexp, 4
dgamma, 5
dinvchisq (invchisq), 2
dinvexp (invexp), 3
dinvgamma (invgamma), 4

invchisq, 2
invexp, 3
invgamma, 4

pinvchisq (invchisq), 2
pinvexp (invexp), 3
pinvgamma (invgamma), 4

qinvchisq (invchisq), 2
qinvexp (invexp), 3
qinvgamma (invgamma), 4

rinvchisq (invchisq), 2
rinvexp (invexp), 3
rinvgamma (invgamma), 4