Package ‘GIGrvg’

June 10, 2017

Type Package
Title Random Variate Generator for the GIG Distribution
Version 0.5
Date 2017-06-08
Author Josef Leydold and Wolfgang Hormann
Maintainer Josef Leydold <josef.leydold@wu.ac.at>
Description Generator and density function for the Generalized Inverse Gaussian (GIG) distribution.
License GPL (>= 2)
LazyLoad yes
NeedsCompilation yes
Repository CRAN
Date/Publication 2017-06-10 06:33:13 UTC

R topics documented:

GIGrvg-package .......................................................... 1
rgig ................................................................. 2

Index

GIGrvg-package

Generator and density for the Generalized Inverse Gaussian (GIG) distribution

Description

This package provides a generator and the density for the Generalized Inverse Gaussian (GIG) distribution. It uses the parametrization with density proportional to

\[ f(x) = x^{\lambda-1} e^{-\frac{1}{2}(\chi/x + \psi x)} \]

Details
Package \texttt{GIGrvg} provides two routines:

\texttt{rgig} generates GIG distributed random variates. It is especially designed for the varying parameter case, i.e., for sample size \( n = 1 \).

\texttt{dgig} computes the density of the GIG distribution.

\textbf{Note} that the parameters of the distribution are assumed to be single values. If a vector is provided then just the first value is used!

For the very fast generation of large samples more efficient algorithms exists. We recommend package \texttt{Runuran}.

\section*{Author(s)}

Josef Leydold <josef.leydold@wu.ac.at> and Wolfgang Hörmann.

\section*{References}


\section*{Examples}

```r
## Draw a random sample
rgig(n=10, lambda=0.5, chi=0.1, psi=2)

## Evaluate the density
dgig(x=0.3, lambda=0.5, chi=0.1, psi=2)
```

\section*{rgig}

\textit{Generator and Density of Generalized Inverse Gaussian (GIG) distribution.}

\section*{Description}

Random variate generator for the Generalized Inverse Gaussian (GIG) distribution. The generator is especially designed for the varying parameter case, i.e., for sample size \( n = 1 \).

\section*{Usage}

```r
rgig(n=1, lambda, chi, psi)
dgig(x, lambda, chi, psi, log = FALSE)
```
rgig

Arguments

- \( n \): Number of observations
- \( \lambda \): Shape parameter
- \( \chi \): Shape and scale parameter. Must be nonnegative for positive \( \lambda \) and positive else.
- \( \psi \): Shape and scale parameter. Must be nonnegative for negative \( \lambda \) and positive else.
- \( x \): Argument of pdf
- \( \log \): If TRUE the logarithm of the density will be returned.

Details

The package uses a parametrization for the GIG distribution where the density is proportional to

\[ f(x) = x^{\lambda - 1} e^{\frac{1}{2}(\chi/x + \psi x)}. \]

The parameters have to satisfy the conditions

\[
\begin{align*}
\lambda > 0, \psi > 0, \chi \geq 0, & \quad \text{or} \\
\lambda = 0, \psi > 0, \chi > 0, & \quad \text{or} \\
\lambda < 0, \psi \geq 0, \chi > 0.
\end{align*}
\]

The generator is especially designed for the varying parameter case, i.e., for sample size \( n=1 \).

Note that the arguments \( n, \lambda, \chi, \psi \) for these two \( R \) routines are assumed to be single values. If a vector is provided, then just the first value is used!

For the generation of large samples more efficient algorithms exist. We recommend package Runuran. The fast numeric inversion function \texttt{pinvd.new} is usable for GIG. It is about three times faster than \texttt{rgig} for large values of \( n \). However, it requires a slow set-up and is therefore not useful for the varying parameter case. For the usage of the Runuran functions see the last example below.

Routine \texttt{rgig} applies three different algorithms depending on the given parameters. When the density is T-concave (roughly spoken when \( \lambda \geq 1 \) or \( \psi \chi \geq 1/4 \)) two variants of the Ratio-of-Uniforms method due to Lehner (1989) are used. These are quite similar to the widely used algorithm by Dagpunar (e.g. in package \texttt{ghyp}) but have a faster setup. When the density is not T-concave then a new algorithm with a uniformly rejection constant is used. (In the latter case Dagpunar’s algorithm may become extremely slow or may sample from an invalid distribution.)

Value

\texttt{rgig} creates a random sample of size \( n \). In case of invalid arguments the routine simply stops execution.

\texttt{dgig} evaluates the density of the GIG distribution.

Author(s)

Josef Leydold <josef.leydold@wu.ac.at> and Wolfgang Hörmann.
References


Examples

```r
## Draw a random sample
x <- rgig(n=10, lambda=0.5, chi=0.1, psi=2)

## Evaluate the density
x <- dgig(0.3, lambda=0.5, chi=0.1, psi=2)

## Create a random sample and create a histogram
y <- rgig(n=10*5,0.1,2,3)
hist(y,breaks=100,freq=FALSE)
xval <- seq(0,max(y),0.01) # to add plot the corresponding density
lines(xval,dgig(xval,0.1,2,3))

## Not run:
## Use a fast method from package Runuran for large samples
## (method PINV implements an approximate inversion method)
library("runuran")
gen <- pinvd.new(udgig(theta=0.2, psi=0.05, chi=0.05))
x <- ur(gen, 10^6)

## End(Not run)
```
Index

*Topic **datagen**
  GIGrvg-package, 1
  **rgig**, 2

*Topic **distribution**
  GIGrvg-package, 1
  **rgig**, 2

*Topic **package**
  GIGrvg-package, 1
  **rgig**, 2

dgig, 2
dgig(rgig), 2

ghyp, 3
GIGrvg (GIGrvg-package), 1
GIGrvg-package, 1

rgig, 2, 2
Runuran, 2, 3