Package ‘VarianceGamma’

November 26, 2023

Version 0.4-2
Date 2023-11-26
Title The Variance Gamma Distribution
Author David Scott <d.scott@auckland.ac.nz> and Christine Yang Dong <c.dong@auckland.ac.nz>
Maintainer David Scott <d.scott@auckland.ac.nz>
Depends R (>= 3.0.1),
Imports grDevices, graphics, stats, DistributionUtils, GeneralizedHyperbolic
Suggests RUnit
Encoding UTF-8
Description Provides functions for the variance gamma distribution. Density, distribution and quantile functions. Functions for random number generation and fitting of the variance gamma to data. Also, functions for computing moments of the variance gamma distribution of any order about any location. In addition, there are functions for checking the validity of parameters and to interchange different sets of parameterizations for the variance gamma distribution.
License GPL (>= 2)
NeedsCompilation no
Repository CRAN
Date/Publication 2023-11-25 23:10:02 UTC

R topics documented:

summary.vgFit .......................... 2
Variance Gamma Mean, Variance, Skewness, Kurtosis and Mode .......................... 3
VarianceGammaDistribution .......................... 4
VarianceGammaPlots .......................... 9
vgCalcRange .......................... 11
Summary Variance Gamma Distribution Fit

Description

summary.vgFit is a method for class "vgFit".

Usage

## S3 method for class 'vgFit'
summary(object, ...)

## S3 method for class 'summary.vgFit'
print(x, digits = max(3, getOption("digits") - 3), ...)

Arguments

object

An object of class "vgFit", resulting from a call to vgFit.

x

An object of class "summary.vgFit", resulting from a call to summary.vgFit.

digits

The number of significant digits to use when printing.

...

Further arguments passed to or from other methods.

Details

summary.vgFit calculates standard errors for the estimates of c, σ, θ, and ν of the variance gamma distribution parameter vector param if the Hessian from the call to optim or nlm is available. Because the parameters in the call to the optimiser are c, log(σ), θ and log(ν), the delta method is used to obtain the standard errors for σ and ν.

Value

If the Hessian is available, summary.vgFit computes standard errors for the estimates of c, σ, θ, and ν, and adds them to object as object$sds. Otherwise, no calculations are performed and the composition of object is unaltered.

summary.vgFit invisibly returns x with class changed to summary.vgFit.

See vgFit for the composition of an object of class vgFit.

print.summary.vgFit prints a summary in the same format as print.vgFit when the Hessian is not available from the fit. When the Hessian is available, the standard errors for the parameter
estimates are printed in parentheses beneath the parameter estimates, in the manner of `fitdistr` in the package `MASS`.

### See Also

`vgFit`, `summary`.

### Examples

```r
### Continuing the `vgFit(.)` example:
param <- c(0.5, 0, 0.5)
dataVector <- rvg(500, param = param)
fit <- vgFit(dataVector)
print(fit)
summary(fit)
```

---

## Description

Functions to calculate the mean, variance, skewness, kurtosis and mode of a specific variance gamma distribution.

### Usage

```r
vgMean(vgC = 0, sigma = 1, theta = 0, nu = 1, param = c(vgC, sigma, theta, nu))
vgVar(vgC = 0, sigma = 1, theta = 0, nu = 1, param = c(vgC, sigma, theta, nu))
vgSkew(vgC = 0, sigma = 1, theta = 0, nu = 1, param = c(vgC, sigma, theta, nu))
vgKurt(vgC = 0, sigma = 1, theta = 0, nu = 1, param = c(vgC, sigma, theta, nu))
vgMode(vgC = 0, sigma = 1, theta = 0, nu = 1, param = c(vgC, sigma, theta, nu))
```

### Arguments

- `vgC`  
  The location parameter $c$, default is equal to 0.
- `sigma`  
  The spread parameter $\sigma$, default is equal to 1, must be positive.
- `theta`  
  The asymmetry parameter $\theta$, default is equal to 0.
- `nu`  
  The shape parameter $\nu$, default is equal to 1, must be positive.
- `param`  
  Specifying the parameters as a vector which takes the form $c(vgC, sigma, theta, nu)$.

### Value

`vgMean` gives the mean of the variance gamma distribution, `vgVar` the variance, `vgSkew` the skewness, `vgKurt` the kurtosis, and `vgMode` the mode. The formulae used for the mean and variance are as given in Seneta (2004). If $\nu$ is greater than or equal to 2, the mode is equal to the value of the parameter $c$. Otherwise, it is found by a numerical optimisation using `optim`.

The parameterisation of the variance gamma distribution used for these functions is the $(c, \sigma, \theta, \nu)$ one. See `vgChangePars` to transfer between parameterisations.
VarianceGammaDistribution

**Description**

Density function, distribution function, quantiles and random number generation for the variance gamma distribution with parameters $c$ (location), $\sigma$ (spread), $\theta$ (asymmetry) and $\nu$ (shape). Utility routines are included for the derivative of the density function and to find suitable break points for use in determining the distribution function.

**Usage**

```r
dvg(x, vgC = 0, sigma = 1, theta = 0, nu = 1,
    param = c(vgC,sigma,theta,nu), log = FALSE,
    tolerance = .Machine$double.eps ^ 0.5, ...)
pvg(q, vgC = 0, sigma = 1, theta = 0, nu = 1,
    param = c(vgC,sigma,theta,nu), lower.tail = TRUE, log.p = FALSE,
```
Arguments

- **x, q** Vector of quantiles.
- **p** Vector of probabilities.
- **n** Number of observations to be generated.
- **vgC** The location parameter \(c\), default is 0.
- **sigma** The spread parameter \(\sigma\), default is 1, must be positive.
- **theta** The asymmetry parameter \(\theta\), default is 0.
- **nu** The shape parameter \(\nu\), default is 1, must be positive.
- **param** Specifying the parameters as a vector which takes the form \((vgC, sigma, theta, nu)\).
- **log, log.p** Logical; if TRUE, probabilities \(p\) are given as \(\log(p)\); not yet implemented.
- **lower.tail** If TRUE (default), probabilities are \(P[X <= x]\), otherwise, \(P[X > x]\); not yet implemented.
- **small** Size of a small difference between the distribution function and zero or one. See Details.
- **tiny** Size of a tiny difference between the distribution function and zero or one. See Details.
- **deriv** Value between 0 and 1. Determines the point where the derivative becomes substantial, compared to its maximum value. See Details.
- **accuracy** Uses accuracy calculated by \(\text{integrate}\) to try and determine the accuracy of the distribution function calculation.
- **subdivisions** The maximum number of subdivisions used to integrate the density returning the distribution function.
- **nInterpol** The number of points used in \(qvg\) for cubic spline interpolation (see \text{splinefun}) of the distribution function.
- **tolerance** Size of a machine difference between two values. See Details.
- **...** Passes arguments to \text{uniroot}. See Details.
Details

Users may either specify the values of the parameters individually or as a vector. If both forms are specified but with different values, then the values specified by vector `param` will always overwrite the other ones.

The variance gamma distribution has density

\[ f(x) = c(c, \sigma, \theta, \nu) \times e^{[\theta(x-c)/\sigma^2]} |x - c|^{1/\nu - 1/2} \frac{|x - c|\sqrt{2\sigma^2/\nu + \theta^2}}{\sigma^2} K_{1/\nu - 1/2} \left( \frac{|x - c|\sqrt{2\sigma^2/\nu + \theta^2}}{\sigma^2} \right) \]

where \( K_{\nu}(\cdot) \) is the modified Bessel function of the third kind of order \( \nu \), and

\[ c(c, \sigma, \theta, \nu) = \frac{2}{\sigma \sqrt{2\pi \nu^{1/\nu} \Gamma(1/\nu)}} \left( \frac{1}{\sqrt{2\sigma^2/\nu + \theta^2}} \right)^{1/\nu - 1/2} \]

Special cases:
1. If \( \nu < 2 \) and \( x = c \), then the density function is approximate to

\[ f(x) = \frac{\Gamma(1/\nu - 1/2)}{\sigma \sqrt{2\pi \nu^{1/\nu} \Gamma(1/\nu)}} \left( \frac{2\sigma^2}{\sqrt{2\sigma^2/\nu + \theta^2}} \right)^{1/\nu - 1/2} \]

2. If \( \nu \geq 2 \) and \( x = c \), then the density function is taken the value \( \text{Inf} \).

Use `vgChangePar` to convert from the \((\mu, \sigma, \tau)\), or \((\theta, \sigma, \kappa, \tau)\) parameterisations given in Kotz et al. (2001) to the \((c, \sigma, \theta, \nu)\) parameterisation used above.

`pvg` breaks the real line into eight regions in order to determine the integral of `dvg`. The break points determining the regions are found by `vgBreaks`, based on the values of `small`, `tiny`, and `deriv`. In the extreme tails of the distribution where the probability is `tiny` according to `vgCalcRange`, the probability is taken to be zero. In the inner part of the distribution, the range is divided in 6 regions, 3 above the mode, and 3 below. On each side of the mode, there are two break points giving the required three regions. The outer break point is where the probability in the tail has the value given by the variable `small`. The inner break point is where the derivative of the density function is `deriv` times the maximum value of the derivative on that side of the mode. In each of the 6 inner regions the numerical integration routine `safeIntegrate` (which is a wrapper for `integrate`) is used to integrate the density `dvg`.

`qvg` uses the breakup of the real line into the same 8 regions as `pvg`. For quantiles which fall in the 2 extreme regions, the quantile is returned as `-Inf` or `Inf` as appropriate. In the 6 inner regions `splinefun` is used to fit values of the distribution function generated by `pvg`. The quantiles are then found using the `uniroot` function.

`pvg` and `qvg` may generally be expected to be accurate to 5 decimal places.

The variance gamma distribution is discussed in Kotz et al (2001). It can be seen to be the weighted difference of two i.i.d. gamma variables shifted by the value of \( \theta \). `rvg` uses this representation to generate observations from the variance gamma distribution.
Value

dvg gives the density function, pvg gives the distribution function, qvg gives the quantile function and rvg generates random variates. An estimate of the accuracy of the approximation to the distribution function may be found by setting accuracy=TRUE in the call to pvg which then returns a list with components value and error.
ddvg gives the derivative of dvg.
vgBreaks returns a list with components:

- **xTiny** Value such that probability to the left is less than tiny.
- **xSmall** Value such that probability to the left is less than small.
- **lowBreak** Point to the left of the mode such that the derivative of the density is deriv times its maximum value on that side of the mode.
- **highBreak** Point to the right of the mode such that the derivative of the density is deriv times its maximum value on that side of the mode.
- **xLarge** Value such that probability to the right is less than small.
- **xHuge** Value such that probability to the right is less than tiny.
- **modeDist** The mode of the given variance gamma distribution.

Author(s)

David Scott <d.scott@auckland.ac.nz>, Christine Yang Dong <c.dong@auckland.ac.nz>

References


See Also

vgChangePars, vgCalcRange

Examples

```r
## Use the following rules for vgCalcRange when plotting graphs for dvg, ## ddvg and pvg.
##
## if nu < 2, use:
## maxDens <- dvg(vgMode(param = c(vgC, sigma, theta, nu)),
## param = c(vgC, sigma, theta, nu), log = FALSE)
## vgRange <- vgCalcRange(param = c(vgC, sigma, theta, nu),
## tol = 10^(-2)*maxDens, density = TRUE)
##
## if nu >= 2 and theta < 0, use:
## vgRange <- c(vgC-2,vgC+6)
##
## if nu >= 2 and theta > 0, use:
## vgRange <- c(vgC-6,vgC+2)
##
## if nu >= 2 and theta = 0, use:
## vgRange <- c(vgC-4,vgC+4)
```
# Example 1 (nu < 2)
## For dvg and pvg
param <- c(0.0, 0.5, 0, 0.5)
maxDens <- dvg(vgMode(param = param), param = param, log = FALSE)
## Or to specify parameter values individually, use:
maxDens <- dvg(vgMode(0, 0.5, 0, 0.5), 0, 0.5, 0.5, log = FALSE)

vgRange <- vgCalcRange(param = param, tol = 10^(-2)*maxDens, density = TRUE)
par(mfrow = c(1,2))
curve(dvg(x, param = param), from = vgRange[1], to = vgRange[2], n = 1000)
title("Density of the Variance Gamma Distribution")
curve(pvg(x, param = param), from = vgRange[1], to = vgRange[2], n = 1000)
title("Distribution Function of the Variance Gamma Distribution")

## For rvg
require(DistributionUtils)
dataVector <- rvg(500, param = param)
curve(dvg(x, param = param), range(dataVector)[1], range(dataVector)[2], n = 500)
hist(dataVector, freq = FALSE, add = TRUE)
title("Density and Histogram of the Variance Gamma Distribution")
logHist(dataVector, main = "Log-Density and Log-Histogram of the Generalized Hyperbolic Distribution")
curve(log(dvg(x, param = param)), add = TRUE,
range(dataVector)[1], range(dataVector)[2], n = 500)

## For dvg and ddvg
par(mfrow = c(2,1))
curve(dvg(x, param = param), from = vgRange[1], to = vgRange[2], n = 1000)
title("Density of the Variance Gamma Distribution")
curve(ddvg(x, param = param), from = vgRange[1], to = vgRange[2], n = 1000)
title("Derivative of the Density of the Variance Gamma Distribution")

# Example 2 (nu > 2 and theta = 0)
## For dvg and pvg
param <- c(0.0, 0.5, 0, 3)
vgRange <- c(-4, 4)
par(mfrow = c(1,2))
curve(dvg(x, param = param), from = vgRange[1], to = vgRange[2], n = 1000)
title("Density of the Variance Gamma Distribution")
curve(pvg(x, param = param), from = vgRange[1], to = vgRange[2], n = 1000)
title("Distribution Function of the Variance Gamma Distribution")

## For rvg
X2 <- rvg(500, param = param)
curve(dvg(x, param = param), min(X2), max(X2), n = 500)
hist(X2, freq = FALSE, add = TRUE)
title("Density and Histogram of the Variance Gamma Distribution")

## For dvg and ddvg
par(mfrow = c(2,1))
curve(dvg(x, param = param), from = vgRange[1], to = vgRange[2], n = 1000)
title("Density of the Variance Gamma Distribution")
curve(ddvg(x, param = param), from = vgRange[1], to = vgRange[2], n = 1000)
title("Derivative of the Density of the Variance Gamma Distribution")
DistributionUtils::logHist(X2, main =
"Log-Density and Log-Histogram of the Generalized Hyperbolic Distribution")
curve(log(dvg(x, param = param)), add = TRUE, min(X2), max(X2), n = 500)

## For dvg and ddvg
par(mfrow = c(2,1))
curve(dvg(x, param = param), from = vgRange[1], to = vgRange[2],
n = 1000)
title("Density of the Variance Gamma Distribution")
curve(ddvg(x, param = param), from = vgRange[1], to = vgRange[2],
n = 1000)
title("Derivative of the Density of the Variance Gamma Distribution")

## Use the following rules for vgCalcRange when plotting graphs for vgBreaks.
## if (nu < 2), use:
## maxDens <- dvg(vgMode(param = c(vgC, sigma, theta, nu)),
## param = c(vgC, sigma, theta, nu), log = FALSE)
## vgRange <- vgCalcRange(param = param, tol = 10^(-6)*maxDens, density = TRUE)
## if (nu >= 2) and theta < 0, use:
## vgRange <- c(vgC-2,vgC+6)
## if (nu >= 2) and theta > 0, use:
## vgRange <- c(vgC-6,vgC+2)
## if (nu >= 2) and theta = 0, use:
## vgRange <- c(vgC-4,vgC+4)

## Example 3 (nu < 2)
## For vgBreaks
param <- c(0,0.5,0,0.5)
maxDens <- dvg(vgMode(param = param), param = param, log = FALSE)
vgRange <- vgCalcRange(param = param, tol = 10^(-6)*maxDens, density = TRUE)
curve(dvg(x, param = param), from = vgRange[1], to = vgRange[2],
n = 1000)
bks <- vgBreaks(param = param)
abline(v = bks)
title("Density of the Variance Gamma Distribution with breaks")

## Example 4 (nu > 2 and theta = 0)
## For vgBreaks
param <- c(0,0.5,0,3)
vgRange <- c(0-4,0+4)
curve(dvg(x, param = param), from = vgRange[1], to = vgRange[2],
n = 1000)
bks <- vgBreaks(param = param)
abline(v = bks)
title("Density of the Variance Gamma Distribution with breaks")
Description

qqvg produces a variance gamma Q-Q plot of the values in y.

ppvg produces a variance gamma P-P (percent-percent) or probability plot of the values in y. Graphical parameters may be given as arguments to qqvg and ppvg.

Usage

```r
qqvg(y, vgC = NULL, sigma = NULL, theta = NULL, nu = NULL,
    param = c(vgC, sigma, theta, nu), main = "Variance Gamma Q-Q Plot",
    xlab = "Theoretical Quantiles", ylab = "Sample Quantiles",
    plot.it = TRUE, line = TRUE, ...)
```

```r
ppvg(y, vgC = NULL, sigma = NULL, theta = NULL, nu = NULL,
    param = c(vgC, sigma, theta, nu), main = "Variance Gamma P-P Plot",
    xlab = "Uniform Quantiles",
    ylab = "Probability-integral-transformed Data", plot.it = TRUE,
    line = TRUE, ...)```

Arguments

- **y** The data sample.
- **vgC** The location parameter $c$, default is 0.
- **sigma** The spread parameter $\sigma$, default is 1, must be positive.
- **theta** The asymmetry parameter $\theta$, default is 0.
- **nu** The shape parameter $\nu$, default is 1, must be positive.
- **param** An optional option, specifying the parameters as a vector which takes the form `c(vgC, sigma, theta, nu)` if known.
- **main** Plot title.
- **xlab, ylab** Plot labels.
- **plot.it** Logical. Should the result be plotted?
- **line** Add line through origin with unit slope.
- **...** Further graphical parameters.

Details

Users may specify the parameter values of the data sample y using argument param. If param is not specified by users, then the values are estimated from y by vgFit. For more details of fitting a variance gamma distribution to data, see vgFit.

Value

For qqvg and ppvg, a list with components:

- **x** The x coordinates of the points that are to be plotted.
- **y** The y coordinates of the points that are to be plotted.
vgCalcRange

Author(s)
David Scott <d.scott@auckland.ac.nz>, Christine Yang Dong <c.dong@auckland.ac.nz>

References

See Also
ppoints, dvg.

Examples

```r
## Example 1: the parameter values are known
par(mfrow = c(1,2))
y <- rvg(200, param = c(2,2,1,2))
qqvg(y, param = c(2,2,1,2), line = FALSE)
abline(0, 1, col = 2)
ppvg(y, param = c(2,2,1,2))

## Example 2: the parameter values are unknown
par(mfrow = c(1,2))
y <- rvg(200, param = c(2,2,1,2))
qqvg(y, line = FALSE)
abline(0, 1, col = 2)
ppvg(y)
```

---

vgCalcRange  
Range of a Variance Gamma Distribution

Description
Given the parameter vector param or the individual parameter values \((c, \sigma, \theta, \nu)\) of a variance gamma distribution, this function determines the range outside of which the density function is negligible, to a specified tolerance. The parameterization used is the \((c, \sigma, \theta, \nu)\) one (see dvg). To use another parameterization, use vgChangePars.

Usage

```r
vgCalcRange(vgC = 0, sigma = 1, theta = 0, nu = 1,
            param = c(vgC, sigma, theta, nu), tol = 10^(-5), density = TRUE, ...)
```
Arguments

vgC  The location parameter $c$, default is 0.
sigma  The spread parameter $\sigma$, default is 1, must be positive.
theta  The asymmetry parameter $\theta$, default is 0.
nu  The shape parameter $\nu$, default is 1, must be positive.
param  Specifying the parameters as a vector which takes the form c(vgC,sigma,theta,nu).
tol  Tolerance.
density  Logical. If TRUE, the bounds are for the density function. If FALSE, they should be for the probability distribution, but this has not yet been implemented.
...  Extra arguments for calls to unroot.

Details

Users may either specify the values of the parameters individually or as a vector. If both forms are specified but with different values, then the values specified by vector param will always overwrite the other ones.

The particular variance gamma distribution being considered is specified by the value of the parameter param.

If density = TRUE, the function gives a range, outside of which the density is less than the given tolerance. Useful for plotting the density. Also used in determining break points for the separate sections over which numerical integration is used to determine the distribution function. The points are found by using unroot on the density function.

If density = FALSE, the function returns the message: "Distribution function bounds not yet implemented".

Value

A two-component vector giving the lower and upper ends of the range.

Author(s)

David Scott <d.scott@auckland.ac.nz>, Christine Yang Dong <c.dong@auckland.ac.nz>

References


See Also

dvg, vgChangePars
Examples

## Use the following rules for vgCalcRange when plotting graphs for dvg, ddvg and pvg.
## if nu < 2, use:
## maxDens <- dvg(vgMode(param = c(vgC, sigma, theta, nu)),
## param = c(vgC, sigma, theta, nu), log = FALSE)
## vgRange <- vgCalcRange(param = c(vgC, sigma, theta, nu),
## tol = 10^(-2)*maxDens, density = TRUE)
## if nu >= 2 and theta < 0, use:
## vgRange <- c(vgC-2,vgC+6)
## if nu >= 2 and theta > 0, use:
## vgRange <- c(vgC-6,vgC+2)
## if nu >= 2 and theta = 0, use:
## vgRange <- c(vgC-4,vgC+4)

param <- c(0,0.5,0,0.5)
maxDens <- dvg(vgMode(param = param), param = param)
vgRange <- vgCalcRange(param = param, tol = 10^(-2)*maxDens)
vgRange
curve(dvg(x, param = param), vgRange[1], vgRange[2])
curve(dvg(x, param = param), vgRange[1], vgRange[2])

param <- c(2,2,0,3)
vgRange <- c(2-4,2+4)
vgRange
curve(dvg(x, param = param), vgRange[1], vgRange[2])
## Not run: vgCalcRange(param = param, tol = 10^(-3), density = FALSE)

vgChangePars

Change Parameterizations of the Variance Gamma Distribution

Description

This function interchanges between the following 4 parameterizations of the variance gamma distribution:

1. $c, \sigma, \theta, \nu$
2. $\theta, \sigma, \mu, \tau$
3. $\theta, \sigma, \kappa, \tau$
4. $\lambda, \alpha, \beta, \mu$

The first set of parameterizations is given in Seneta (2004). The second and third ones are the parameterizations given in Kotz et al. (2001). The last set takes the form of the generalized hyperbolic distribution parameterization. $\delta$ is not included since the variance gamma distribution is a limiting case of generalized hyperbolic distribution with $\delta$ always equal to 0.
Usage

vgChangePars(from, to, param, noNames = FALSE)

Arguments

from The set of parameters to change from.
to The set of parameters to change to.
param "from" parameter vector consisting of 4 numerical elements.
noNames Logical. When TRUE, suppresses the parameter names in the output.

Details

In the 3 parameterizations, the following must be positive:
1. $\sigma$, $\nu$
2. $\sigma$, $\tau$
3. $\sigma$, $\tau$
4. $\lambda$, $\alpha$

In addition in the 4th parameterization, the absolute value of $\beta$ must be less than $\alpha$.

Value

A numerical vector of length 4 representing param in the to parameterization.

Author(s)

David Scott <d.scott@auckland.ac.nz>, Christine Yang Dong <c.dong@auckland.ac.nz>

References


See Also

dvg, vgMom

Examples

```r
param1 <- c(2,2,1,3) # Parameterization 1
gChangePars(1, 2, param1) # Convert to parameterization 2
param2 <- vgChangePars(2, 1, as.numeric(param2)) # Convert back to parameterization 1

param3 <- c(1,2,0,0.5) # Parameterization 3
param1 <- vgChangePars(3, 1, param3) # Convert to parameterization 1
param1 <- vgChangePars(1, 3, as.numeric(param1)) # Convert back to parameterization 3
```
Check Parameters of the Variance Gamma Distribution

Description

Given a putative set of parameters for the variance gamma distribution, the functions checks if the
parameters are in the correct range, and if the set has the correct length of 4.

Usage

vgCheckPars(param, ...)

Arguments

param Numeric. Putative parameter values for a Variance Gamma distribution.
... Further arguments for calls to all.equal.

Details

The vector param takes the form c(c, sigma, theta, nu). If either sigma or nu is negative, then
an error message is returned.
If the vector param has a length not equal to 4, then an error message is returned.

Value

A list with components:

case Whichever of 'error' or 'normal' is identified by the function.
errMessage An appropriate error message if an error was found, the empty string "" otherwise.

Author(s)

David Scott <d.scott@auckland.ac.nz>, Christine Yang Dong <c.dong@auckland.ac.nz>

See Also
dvg, vgMom

Examples

vgCheckPars(c(0,1,0,1))  # normal
vgCheckPars(c(0,0,0,1))  # error
vgCheckPars(c(0,1,0,-2)) # error
vgCheckPars(c(0,1,0))    # error
vgFit

Fit the Variance Gamma to Data

**Description**

Fits a variance gamma distribution to data. Displays the histogram, log-histogram (both with fitted densities), Q-Q plot and P-P plot for the fit which has the maximum likelihood.

**Usage**

```r
g vgFit(x, freq = NULL, breaks = NULL, paramStart = NULL,
   startMethod = "Nelder-Mead", startValues = "SL",
   method = "Nelder-Mead", hessian = FALSE,
   plots = FALSE, printOut = FALSE,
   controlBFGS = list(maxit = 200),
   controlNM = list(maxit = 1000), maxitNLM = 1500, ...)
```

## S3 method for class 'vgFit'
print(x, digits = max(3, getOption("digits") - 3), ...)

## S3 method for class 'vgFit'
plot(x, which = 1:4,
   plotTitles = paste(c("Histogram of ","Log-Histogram of ",
   "Q-Q Plot of ","P-P Plot of "), x$obsName,
   sep = ""),
   ask = prod(par("mfcol")) < length(which) && dev.interactive(),
   ...)

**Arguments**

- **x**  
  Data vector for vgFit. Object of class "vgFit" for print.vgFit and plot.vgFit.

- **freq**  
  A vector of weights with length equal to length(x).

- **breaks**  
  Breaks for histogram, defaults to those generated by hist(x, right = FALSE, plot = FALSE).

- **paramStart**  
  A user specified starting parameter vector param taking the form c(vgC, sigma, theta, nu).

- **startMethod**  
  Method used by vgFitStart in calls to optim, default is "Nelder-Mead". See Details.

- **startValues**  
  Code giving the method of determining starting values for finding the maximum likelihood estimate of param, default method is "SL". See Details.

- **method**  
  Different optimisation methods to consider, default is "Nelder-Mead". See Details.

- **hessian**  
  Logical. If TRUE the value of the hessian is returned.

- **plots**  
  Logical. If FALSE suppresses printing of the histogram, log-histogram, Q-Q plot and P-P plot.
printOut Logical. If FALSE suppresses printing of results of fitting.
controlBFGS A list of control parameters for optim when using the "BFGS" optimisation.
controlNM A list of control parameters for optim when using the "Nelder-Mead" optimisation.
maxitNLM A positive integer specifying the maximum number of iterations when using the "nlm" optimisation.
digits Desired number of digits when the object is printed.
which If a subset of the plots is required, specify a subset of the numbers 1:4.
plotTitles Titles to appear above the plots.
ask Logical. If TRUE, the user is asked before each plot, see par(ask = ).
... Passes arguments to par, hist, logHist, qqhyperb and pphyperb.

Details

startMethod can be either "BFGS" or "Nelder-Mead".
startValues can be one of the following:
  "US" User-supplied.
  "SL" Based on a fitted skew-Laplace distribution.
  "MoM" Method of moments.

For the details concerning the use of paramStart, startMethod, and startValues, see vgFitStart.
The three optimisation methods currently available are:
  "BFGS" Uses the quasi-Newton method "BFGS" as documented in optim.
  "Nelder-Mead" Uses an implementation of the Nelder and Mead method as documented in optim.
  "nlm" Uses the nlm function in R.

For details of how to pass control information for optimisation using optim and nlm, see optim and nlm.

When method = "Nelder-Mead" is used, very rarely, it would return an error message of "error in optim(paramStart,...)", use method = "BFGS" or method = "nlm" instead in that case.
When method = "nlm" is used, warnings may be produced. These do not appear to be a problem.

Value

A list with components:

param A vector giving the maximum likelihood estimate of param, as (c, sigma, theta, nu).
maxLik The value of the maximised log-likelihood.
hessian If hessian was set to TRUE, the value of the hessian. Not present otherwise.
method Optimisation method used.
conv Convergence code. See the relevant documentation (either optim or nlm) for details on convergence.
iter  Number of iterations of optimisation routine.
obs  The data used to fit the hyperbolic distribution.
obsName  A character string with the actual obs argument name.
paramStart  Starting value of param returned by call to vgFitStart.
svName  Descriptive name for the method finding start values.
startValues  Acronym for the method of finding start values.
breaks  The cell boundaries found by a call to hist.
midpoints  The cell midpoints found by a call to hist.
empDens  The estimated density found by a call to hist.

Author(s)

David Scott <d.scott@auckland.ac.nz>, Christine Yang Dong <c.dong@auckland.ac.nz>

References


See Also

optim, nlm, par, hist, logHist, qqvg, ppvg, dskewlap and vgFitStart.

Examples

param <- c(0,0.5,0,0.5)
dataVector <- rvg(500, param = param)
## See how well vgFit works
vgFit(dataVector)
vgFit(dataVector, plots = TRUE)
fit <- vgFit(dataVector)
par(mfrow = c(1,2))
plot(fit, which = c(1,3))

## Use nlm instead of default
param <- c(0,0.5,0,0.5)
dataVector <- rvg(500, param = param)
vgFit(dataVector, method = "nlm", hessian = TRUE)

## Use BFGS instead of default
param <- c(0,0.5,0,0.5)
dataVector <- rvg(500, param = param)
vgFit(dataVector, method = "BFGS", hessian = TRUE)
vgFitStart

Find Starting Values for Fitting a Variance Gamma Distribution

Description

Finds starting values for input to a maximum likelihood routine for fitting variance gamma distribution to data.

Usage

vgFitStart(x, breaks = NULL, startValues = "SL", paramStart = NULL, startMethodSL = "Nelder-Mead", startMethodMoM = "Nelder-Mead", ...)

vgFitStartMoM(x, startMethodMoM = "Nelder-Mead", ...)

Arguments

x      Data vector.
breaks Breaks for histogram. If missing, defaults to those generated by hist(x, right = FALSE, plot = FALSE).
startValues Vector of the different starting values to consider. See Details.
paramStart Starting values for param if startValues = "US".
startMethodSL Method used by call to optim in finding skew Laplace estimates.
startMethodMoM Method used by call to optim in finding method of moments estimates.
... Passes arguments to optim.

Details

Possible values of the argument startValues are the following:

"US"  User-supplied.
"SL"  Based on a fitted skew-Laplace distribution.
"MoM"  Method of moments.

If startValues = "US" then a value must be supplied for paramStart.

If startValues = "MoM", vgFitStartMoM is called. These starting values are based on Barndorff-Nielsen et al (1985).

If startValues = "SL", or startValues = "MoM" an initial optimisation is needed to find the starting values. These optimisations call optim.
Value

vgFitStart returns a list with components:

- **vgStart**: A vector with elements \( \text{vgC} \), \( \log(\text{sigma}) \), \( \theta \) and \( \log(\nu) \) giving the starting value of param.
- **xName**: A character string with the actual \( x \) argument name.
- **breaks**: The cell boundaries found by a call to **hist**.
- **midpoints**: The cell midpoints found by a call to **hist**.
- **empDens**: The estimated density found by a call to **hist**.

vgFitStartMoM returns only the method of moments estimates as a vector with elements \( \text{vgC} \), \( \log(\text{sigma}) \), \( \theta \) and \( \log(\nu) \).

Author(s)

David Scott <d.scott@auckland.ac.nz>, Christine Yang Dong <c.dong@auckland.ac.nz>

References


See Also

- **dvg**, **dskewlap**, **vgFit**, **hist**, and **optim**.

Examples

```r
param <- c(0,0.5,0,0.5)
dataVector <- rvg(500, param = param)
vgFitStart(dataVector,startValues="SL")
vgFitStartMoM(dataVector)
vgFitStart(dataVector,startValues="MoM")
```

vgMom

*Calculate Moments of the Variance Gamma Distribution*

Description

This function can be used to calculate raw moments, mu moments, central moments and moments about any other given location for the variance gamma (VG) distribution.

Usage

```r
vgMom(order, vgC = 0, sigma = 1, theta = 0, nu = 1,
      param = c(vgC,sigma,theta,nu), momType = "raw", about = 0)
```
vgMom

Arguments

order Numeric. The order of the moment to be calculated. Not permitted to be a vector. Must be a positive whole number except for moments about zero.

vgC The location parameter c, default is 0.
sigma The spread parameter σ, default is 1, must be positive.
theta The asymmetry parameter θ, default is 0.
nu The shape parameter ν, default is 1, must be positive.

param Specifying the parameters as a vector which takes the form c(vgC, sigma, theta, nu).
momType Common types of moments to be calculated, default is "raw". See Details.

about Numeric. The point around which the moment is to be calculated, default is 0. See Details.

Details

For the parameters of the variance gamma distribution, users may either specify the values individually or as a vector. If both forms are specified but with different values, then the values specified by vector param will always overwrite the other ones. In addition, the parameters values are examined by calling the function vgCheckPars to see if they are valid for the VG distribution.

order is also checked by calling the function is.wholenumber in DistributionUtils package to see whether a whole number is given.

momType can be either "raw" (moments about zero), "mu" (moments about vgC), or "central" (moments about mean). If one of these moment types is specified, then there is no need to specify the about value. For moments about any other location, the about value must be specified. In the case that both momType and about are specified and contradicting, the function will always calculate the moments based on about rather than momType.

To calculate moments of the VG distribution, the function first calculates mu moments by the formula defined below and then transforms mu moments to central moments or raw moments or moments about any other location as required by calling momChangeAbout in DistributionUtils package.

To calculate mu moments of the variance gamma distribution, the function first transforms the parameterization of c,σ,θ,ν to the generalized hyperbolic distribution’s parameterization of λ,α,β,μ (see vgChangePars for details). Then, the mu moments of the variance gamma distribution are given by

\[ \sum_{\ell=\lceil (k+1)/2 \rceil}^k a_{k,\ell}\beta^{2\ell-k} [\Gamma(\lambda + \ell)/\Gamma(\lambda)2^\ell/(\alpha^2 - \beta^2)\ell] \]

where \( k = \text{order} \) and \( k > 0 \) and \( a_{k,\ell} \) is the recursive coefficient (see momRecursion for details).

This formula is developed from the mu moments formula of the generalized hyperbolic distribution given in Scott, Würtz and Tran (2008). Note that the part in [ ] of this equation is actually equivalent to the formula of raw moments of the gamma distribution. So the function calls gammaRawMom in GeneralizedHyperbolic package when implementing the computations.

Value

The moment specified. In the case of raw moments, Inf is returned if the moment is infinite.
Author(s)
David Scott <d.scott@auckland.ac.nz>, Christine Yang Dong <c.dong@auckland.ac.nz>

References

See Also
vgCheckPars, vgChangePars, vgMean, vgVar, vgSkew, vgKurt, is.wholenumber, momRecursion, momChangeAbout and momIntegrated.

Examples
### Raw moments of the VG distribution
gvMom(3, param=c(2,1,2,1), momType = “raw”)

### Mu moments of the VG distribution
gvMom(2, param=c(2,1,2,1), momType = “mu”)

### Central moments of the VG distribution
gvMom(4, param=c(2,1,2,1), momType = “central”)

### Moments about any locations
gvMom(4, param=c(2,1,2,1), about = 1)

---

vgParam          Parameter Sets for Variance Gamma Distribution

Description
These objects store different parameter sets of the Variance Gamma distribution for testing or demonstrating purpose as matrixes. Specifically, the parameter sets vgSmallShape and vgLargeShape have constant (standard) location and spread parameters of \(c=0\) and \(\sigma=1\); where asymmetry and shape parameters vary from \(\theta=(-2, 0, 2)\) and \(\nu=(0.5, 1, 2)\) for vgSmallShape and \(\theta=(-4, -2, 0, 2, 4)\) and \(\nu=(0.25, 0.5, 1, 2, 4)\) for vgLargeShape.

The parameter sets vgSmallParam and vgLargeParam have varied values of all 4 parameters. vgSmallParam contains all of the parameter combinations from \(c=(-2, 0, 2), \sigma=(0.5, 1, 2), \theta=(-2, 0, 2)\) and \(\nu=(0.5, 1, 2)\). vgLargeParam contains all of the parameter combinations from \(c=(-4, -2, 0, 2, 4), \sigma=(0.25, 0.5, 1, 2, 4), \theta=(-4, -2, 0, 2, 4)\) and \(\nu=(0.25, 0.5, 1, 2, 4)\).

Usage
data(vgParam)
 vgParam

Format

vgSmallShape: a 9 by 4 matrix; vgLargeShape: a 25 by 4 matrix; vgSmallParam: a 81 by 4 matrix; vgLargeParam: a 625 by 4 matrix.

Author(s)

David Scott <d.scott@auckland.ac.nz>, Christine Yang Dong <c.dong@auckland.ac.nz>

Examples

data(vgParam)
## Testing the accuracy of vgMean
for (i in 1:nrow(vgSmallParam)) {
  param <- vgSmallParam[i,]
  x <- rvg(10000,param = param)
  sampleMean <- mean(x)
  funMean <- vgMean(param = param)
  difference <- abs(sampleMean - funMean)
  print(difference)
}

Index

* distribution
  vgCheckPars, 15
  vgMom, 20
  besselK, 4
  ddvg (VarianceGammaDistribution), 4
dskewlap, 18, 20
dvg, 4, 11, 12, 14, 15, 20
dvg (VarianceGammaDistribution), 4
hist, 18, 20
integrate, 5, 6
is.wholenumber, 22
logHist, 18
momChangeAbout, 22
momIntegrated, 22
momRecursion, 21, 22
nlm, 2, 17, 18
optim, 2, 3, 16–20
par, 17, 18
plot.vgFit (vgFit), 16
ppoints, 11
ppvg, 18
ppvg (VarianceGammaPlots), 9
print.summary.vgFit (summary.vgFit), 2
print.vgFit, 2
print.vgFit (vgFit), 16
pvg (VarianceGammaDistribution), 4
qqvg, 18
qqvg (VarianceGammaPlots), 9
qvg (VarianceGammaDistribution), 4
rvg (VarianceGammaDistribution), 4
safeIntegrate, 6
summary, 3
summary.vgFit, 2, 2
unirout, 12
Variance Gamma Mean, Variance, Skewness, Kurtosis and Mode, 3
VarianceGammaDistribution, 4
VarianceGammaPlots, 9
gvBreaks (VarianceGammaDistribution), 4
gvCalcRange, 4, 7, 11
gvChangePars, 3, 4, 7, 11, 12, 13, 21, 22
gvCheckPars, 15, 22
gvFit, 2, 3, 10, 16, 20
gvFitStart, 17, 18, 19
gvFitStartMoM (vgFitStart), 19
gvKurt, 22
gvKurt (Variance Gamma Mean, Variance, Skewness, Kurtosis and Mode), 3
gvLargeParam (vgParam), 22
gvLargeShape (vgParam), 22
gvMean, 22
gvMean (Variance Gamma Mean, Variance, Skewness, Kurtosis and Mode), 3
gvMode (Variance Gamma Mean, Variance, Skewness, Kurtosis and Mode), 3
gvMom, 14, 15, 20
gvParam, 22
gvSkew, 22
gvSkew (Variance Gamma Mean, Variance, Skewness, Kurtosis and Mode), 3
gvSmallParam (vgParam), 22
gvSmallShape (vgParam), 22
gvVar, 22
gvVar (Variance Gamma Mean, Variance, Skewness, Kurtosis and Mode), 3