Package ‘logitnorm’

December 8, 2017

Title Functions for the Logitnormal Distribution

Version 0.8.35

Date 2017-12-08

Author Thomas Wutzler

Maintainer Thomas Wutzler <twutz@bgc-jena.mpg.de>

Description Density, distribution, quantile and random generation function for the logitnormal distribution. Estimation of the mode and the first two moments. Estimation of distribution parameters.

Depends

Suggests RUnit, knitr, ggplot2, reshape2

VignetteBuilder knitr

License GPL-2

LazyData true

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-12-08 17:45:30 UTC

R topics documented:

`logitnorm-package` *Utilities for the logitnormal distribution in R*

Description

Utilities for the logitnormal distribution in R

- Density, distribution, quantile and random generation function.
- Estimation of the mode and the first two moments.
- Estimation of distribution parameters from observations.

Details

The logitnormal distribution is useful as a prior density for variables that are bounded between 0 and 1, such as proportions. Fig. 1 displays its density for various combinations of parameters mu and sigma.

The package provides the main distribution functions:

- density `dlogitnorm`,
- distribution `plogitnorm`,
- quantile `qlogitnorm`, and
- random generation function `rlogitnorm`.

Transformation functions

- (0,1) -> (-Inf,Inf): `logit`
- (-Inf,Inf) -> (0,1): `invlogit`

Moments and mode

- Expected value and variance: `momentsLogitnorm`
- Mode: `modeLogitnorm`

Estimating parameters

- from mode and upper quantile: `twCoefLogitnormMLE`
- from mode and constraint to be unimodal and maximally flat: `twCoefLogitnormMLEFlat`
- from median and upper quantile: `twCoefLogitnorm`
- from expected value, i.e. mean and upper quantile: `twCoefLogitnormE`
- from a confidence interval which is symmetric at normal scale: `twCoefLogitnormCi`
- from prescribed quantiles: `twCoefLogitnormN`

Author(s)

Thomas Wutzler

References

Frederic, P. & Lad, F. (2008) Two Moments of the Logitnormal Distribution. Communications in Statistics-Simulation and Computation, 37, 1263-1269

`dlogitnorm` *dlogitnorm*

Description

Density function of logitnormal distribution

Usage

```
dlogitnorm(q, mu = 0, sigma = 1, log = FALSE, ...)
```

Arguments

`q`	quantiles
`mu`	distribution parameters
`sigma`	
`log`	if TRUE, the log-density is returned
`...`	further arguments passed to `dnorm`: `mean`, and `sd` for mu and sigma respectively.

Details

Logitnorm distribution
- density function: dlogitnorm
- distribution function: `plogitnorm`
- quantile function: `qlogitnorm`
- random generation function: `rlogitnorm`

Author(s)

Thomas Wutzler

See Also

`logitnorm`

`invlogit` *invlogit*

Description

Transforming (-Inf,Inf) to original scale (0,1)

Usage

```
invlogit(q, ...)
```

Arguments

`q`

`...`

Details

function $f(z) = \frac{e^z}{e^z+1} = \frac{1}{1+e^{-z}}$

Author(s)

Thomas Wutzler

See Also

`logit`

`logitnorm`

`logit` *logit*

Description

Transforming (0,1) to normal scale (-Inf Inf)

Usage

```
logit(p, ...)
```

Arguments

`p`

`...`

Details

function $logit(p) = log\left(\frac{p}{1-p}\right) = log(p) - log(1-p)$

Author(s)

Thomas Wutzler

See Also

invlogit

logitnorm

`modeLogitnorm` *modeLogitnorm*

Description

Mode of the logitnormal distribution by numerical optimization

Usage

```
modeLogitnorm(mu, sigma, tol = invlogit(mu)/1000)
```

Arguments

`mu`	parameter mu
`sigma`	parameter sigma
`tol`	precisions of the estimate

Author(s)

Thomas Wutzler

See Also

logitnorm

momentsLogitnorm *momentsLogitnorm*

Description

First two moments of the logitnormal distribution by numerical integration

Usage

```
momentsLogitnorm(mu, sigma, abs.tol = 0, ...)
```

Arguments

mu	parameter mu
sigma	parameter sigma
abs.tol	chaning default to integrate
...	further parameters to the integrate function

Value

named numeric vector with components

- `mean`: expected value, i.e. first moment
- `var`: variance, i.e. second moment

Author(s)

Thomas Wutzler

Examples

```
(res <- momentsLogitnorm(4,1))

(res <- momentsLogitnorm(5,0.1))
```

plogitnorm *plogitnorm*

Description

Distribution function for logitnormal distribution

Usage

```
plogitnorm(q, mu = 0, sigma = 1, ...)
```

Arguments

q	
mu	distribution parameters
sigma	
...	

Author(s)

Thomas Wutzler

See Also

logitnorm

qlogitnorm *qlogitnorm*

Description

Quantiles of logitnormal distribution.

Usage

```
qlogitnorm(p, mu = 0, sigma = 1, ...)
```

Arguments

p	
mu	distribution parameters
sigma	
...	

Author(s)

Thomas Wutzler

See Also

`logitnorm`

---

rlogitnorm *rlogitnorm*

Description

Random number generation for logitnormal distribution

Usage

```
rlogitnorm(mu = 0, sigma = 1, ...)
```

Arguments

`mu`	distribution parameters
`sigma`	
`...`	arguments to `rnorm`

Author(s)

Thomas Wutzler

See Also

`logitnorm`

---

twCoefLogitnorm *twCoefLogitnorm*

Description

Estimating coefficients of logitnormal distribution from median and upper quantile

Usage

```
twCoefLogitnorm(median, quant, perc = 0.975, method = "BFGS",

    theta0 = c(mu = 0, sigma = 1), returnDetails = FALSE,

    ...)
```

Arguments

`median`	numeric vector: the median of the density function
`quant`	numeric vector: the upper quantile value
`perc`	numeric vector: the probability for which the quantile was specified
`method`	method of optimization (see `optim`)
`theta0`	starting parameters
`returnDetails`	if TRUE, the full output of optim is attached as attributes resOptim
`...`	

Value

numeric matrix with columns `c("mu","sigma")`

rows correspond to rows in median, quant, and perc

Author(s)

Thomas Wutzler

See Also

`logitnorm`

Examples

```
# estimate the parameters, with median at 0.7 and upper quantile at 0.9

(theta <- twCoefLogitnorm(0.7,0.9))



x <- seq(0,1,length.out=41)[-c(1,41)] # plotting grid
```

```
px <- plogitnorm(x,mu=theta[1],sigma=theta[2]) #percentiles function

plot(px~x); abline(v=c(0.7,0.9),col="gray"); abline(h=c(0.5,0.975),col="gray")



dx <- dlogitnorm(x,mu=theta[1],sigma=theta[2]) #density function

plot(dx~x); abline(v=c(0.7,0.9),col="gray")



# vectorized

(theta <- twCoefLogitnorm(seq(0.4,0.8,by=0.1),0.9))
```

`twCoefLogitnormCi` *twCoefLogitnormCi*

Description

Calculates mu and sigma of the logitnormal distribution from lower and upper quantile, i.e. confidence interval.

Usage

```
twCoefLogitnormCi(lower, upper, perc = 0.975, sigmaFac = qnorm(perc),

    isTransScale = FALSE)
```

Arguments

`lower`	value at the lower quantile, i.e. practical minimum
`upper`	value at the upper quantile, i.e. practical maximum
`perc`	numeric vector: the probability for which the quantile was specified
`sigmaFac`	sigmaFac=2 is 95% sigmaFac=2.6 is 99% interval
`isTransScale`	if true lower and upper are already on logit scale

Value

named numeric vector: mu and sigma parameter of the logitnormal distribution.

Author(s)

Thomas Wutzler

See Also

logitnorm

Examples

```
mu=2

sd=c(1,0.8)

p=0.99

lower <- l <- qlogitnorm(1-p, mu, sd ) # p-confidence interval

upper <- u <- qlogitnorm(p, mu, sd ) # p-confidence interval

cf <- twCoefLogitnormCi(lower,upper)

all.equal( cf[,"mu"] , c(mu,mu) )

all.equal( cf[,"sigma"] , sd )
```

---

twCoefLogitnormE *twCoefLogitnormE*

Description

Estimating coefficients of logitnormal distribution from expected value, i.e. mean, and upper quantile.

Usage

```
twCoefLogitnormE(mean, quant, perc = c(0.975), method = "BFGS",

    theta0 = c(mu = 0, sigma = 1), returnDetails = FALSE,

    ...)
```

Arguments

`mean`	the expected value of the density function
`quant`	the quantile values
`perc`	the probabilites for which the quantiles were specified
`method`	method of optimization (see `optim`)
`theta0`	starting parameters
`returnDetails`	if TRUE, the full output of optim is returned with attribut resOptim
`...`	

Value

named numeric matrix with estimated parameters of the logitnormal distrubtion.

colnames: `c("mu","sigma")`

Author(s)

Thomas Wutzler

See Also

`logitnorm`

Examples

```
# estimate the parameters

(thetaE <- twCoefLogitnormE(0.7,0.9))



x <- seq(0,1,length.out=41)[-c(1,41)] # plotting grid
```

```
px <- plogitnorm(x,mu=thetaE[1],sigma=thetaE[2]) #percentiles function

plot(px~x); abline(v=c(0.7,0.9),col="gray"); abline(h=c(0.5,0.975),col="gray")

dx <- dlogitnorm(x,mu=thetaE[1],sigma=thetaE[2]) #density function

plot(dx~x); abline(v=c(0.7,0.9),col="gray")



z <- rlogitnorm(1e5, mu=thetaE[1],sigma=thetaE[2])

mean(z) # about 0.7



# vectorized

(theta <- twCoefLogitnormE(mean=seq(0.4,0.8,by=0.1),quant=0.9))
```

`twCoefLogitnormMLE` *twCoefLogitnormMLE*

Description

Estimating coefficients of logitnormal distribution from mode and upper quantile

Usage

```
twCoefLogitnormMLE(mle, quant, perc = 0.999)
```

Arguments

`mle`	numeric vector: the mode of the density function
`quant`	numeric vector: the upper quantile value
`perc`	numeric vector: the probability for which the quantile was specified

Value

numeric matrix with columns `c("mu","sigma")`

rows correspond to rows in mle, quant, and perc

Author(s)

Thomas Wutzler

See Also

`logitnorm`

Examples

```
# estimate the parameters, with mode 0.7 and upper quantile 0.9

(theta <- twCoefLogitnormMLE(0.7,0.9))

x <- seq(0,1,length.out=41)[-c(1,41)] # plotting grid

px <- plogitnorm(x,mu=theta[1],sigma=theta[2]) #percentiles function

plot(px~x); abline(v=c(0.7,0.9),col="gray"); abline(h=c(0.999),col="gray")

dx <- dlogitnorm(x,mu=theta[1],sigma=theta[2]) #density function

plot(dx~x); abline(v=c(0.7,0.9),col="gray")

# vectorized

(theta <- twCoefLogitnormMLE(mle=seq(0.4,0.8,by=0.1),quant=0.9))
```

```
# flat
(theta <- twCoefLogitnormMLEFlat(0.7))
```

---

twCoefLogitnormMLEFlat *twCoefLogitnormMLEFlat*

Description

Estimating coefficients of a maximally flat unimodal logitnormal distribution from mode

Usage

```
twCoefLogitnormMLEFlat(mle)
```

Arguments

mle	numeric vector: the mode of the density function

Details

When increasing the sigma parameter, the distribution becomes
eventually becomes bi-model, i.e. has two maxima.
This function estimates parameters for given mode, so that the distribution assigns high
densitiy to a maximum range, i.e. is maximally flat, but still is unimodal.

Author(s)

Thomas Wutzler

twCoefLogitnormN *twCoefLogitnormN*

Description

Estimating coefficients of logitnormal distribution from a vector of quantiles and perentiles (non-vectorized).

Usage

```
twCoefLogitnormN(quant, perc = c(0.5, 0.975), method = "BFGS",

    theta0 = c(mu = 0, sigma = 1), returnDetails = FALSE,

    ...)
```

Arguments

`quant`	the quantile values
`perc`	the probabilites for which the quantiles were specified
`method`	method of optimization (see `optim`)
`theta0`	starting parameters
`returnDetails`	if TRUE, the full output of optim is returned instead of only entry par
`...`	further parameters passed to optim, e.g. control=list(maxit=1000)

Value

named numeric vector with estimated parameters of the logitnormal distrubtion.

names: `c("mu","sigma")`

Author(s)

Thomas Wutzler

See Also

`logitnorm`

Examples

```
    # experiment of re-estimation the parameters from generated observations

    thetaTrue <- c(mu=0.8, sigma=0.7)

    obsTrue <- rlogitnorm(thetaTrue["mu"],thetaTrue["sigma"], n=500)

    obs <- obsTrue + rnorm(100, sd=0.05)        # some observation uncertainty

    plot(density(obsTrue),col="blue"); lines(density(obs))



    # re-estimate parameters based on the quantiles of the observations

(theta <- twCoefLogitnorm( median(obs), quantile(obs,probs=0.9), perc = 0.9))



    # add line of estimated distribution

x <- seq(0,1,length.out=41)[-c(1,41)] # plotting grid

    dx <- dlogitnorm(x,mu=theta[1],sigma=theta[2])

    lines( dx ~ x, col="orange")
```

---

twSigmaLogitnorm *twSigmaLogitnorm*

Description

Estimating coefficients of logitnormal distribution from mode and given mu

Usage

```
twSigmaLogitnorm(mle, mu = 0)
```

Arguments

`mle`	numeric vector: the mode of the density function
`mu`	for mu=0 the distribution will be the flattest case (maybe bimodal)

Details

For a mostly flat unimodal distribution use `twCoefLogitnormMLE(mle,0)`

Value

numeric matrix with columns `c("mu","sigma")`

rows correspond to rows in mle and mu

Author(s)

Thomas Wutzler

See Also

`logitnorm`

Examples

```
    mle <- 0.8

    (theta <- twSigmaLogitnorm(mle))

    #

x <- seq(0,1,length.out=41)[-c(1,41)] # plotting grid

px <- plogitnorm(x,mu=theta[1],sigma=theta[2]) #percentiles function

plot(px~x); abline(v=c(mle),col="gray")

dx <- dlogitnorm(x,mu=theta[1],sigma=theta[2]) #density function

plot(dx~x); abline(v=c(mle),col="gray")
```

```
# vectorized
(theta <- twSigmaLogitnorm(mle=seq(0.401,0.8,by=0.1)))
```

Index