

# Package ‘qGaussian’

July 12, 2018

**Type** Package

**Title** The q-Gaussian Distribution

**Version** 0.1.8

**Author** Emerson Luis de Santa Helena <elsh@ufs.br>  
Wagner Santos de Lima <wagnersantos.ufs@hotmail.com>

**Maintainer** Wagner Santos de Lima <wagnersantos.ufs@hotmail.com>

**Description** Density, distribution function, quantile function and  
random generation for the q-gaussian distribution with parameters mu and sig.

**License** GPL (>= 2)

**Imports** Rcpp (>= 0.12.10), stats, robustbase, zipfR

**LinkingTo** Rcpp

**LazyData** true

**RoxygenNote** 6.0.1

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2018-07-12 18:50:06 UTC

## R topics documented:

Chaotic	2
cqgauss	3
dqgauss	4
pqgauss	6
qbymc	8
rqgauss	9

<b>Index</b>	<b>11</b>
--------------	-----------

---

Chaotic	<i>Chaotic, a random number generator of q-Gaussian random variables.</i>
---------	---

---

**Description**

Given a random number generator of q-Gaussian random variables for a range of q values,  $-8 < q < 3$ , based on deterministic map dynamics. To yield a 'q' value, a characteristic entropic index of the q-gaussian distributions.

**Usage**

```
Chaotic(n,q,v0,z0)
```

**Arguments**

n	number of observations. If length(n) > 1, the length is taken to be the number required.
q	entropic index.
v0	a random seed.
z0	a random seed.

**Value**

a number  $q < 3$ , and the standard error.

**Author(s)**

Emerson Luis de Santa Helena , Wagner Santos de Lima

**References**

Umeno, K., Sato, A., IEEE Transactions on Information Theory (Volume:59,Issue:5,May 2013).Chaotic Method for Generating q-Gaussian Random Variables.

**See Also**

Distributions for other standard distributions, including dt and dcauchy. [Distributions](#)

**Examples**

```
t=Chaotic(100000,0,.1,.1)
hist(t,breaks=100)
```

---

 cqgauss

*The q-gaussian Distribution*


---

### Description

Density, distribution function, quantile function and random generation for the q-gaussian distribution with parameters mu and sig.

### Usage

```
cqgauss(p, q = 0, mu = 0, sig = 1, lower.tail = TRUE)
```

### Arguments

p	vector of probabilities.
q	entropic index.
mu	a value for q-mean.
sig	a value for q-variance.
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

### Details

If q, mu and sig values are not specified, they assume the default values of 0, 0 and 1, respectively. Defining  $Z=(q-1)/(3-q)$ , the q-gaussian distribution has density written as

$$p(x) = (\text{sig} * \text{Beta}(\alpha/2, 1/2))^{-1} * (1 + Z(x - \mu)^2 / \text{sig}^2)^{-(1 + 1/Z)/2}$$

where  $\alpha = 1 - 1/Z$  when  $q < 1$  and  $1/Z$  when  $1 < q < 3$ .

### Value

dqgauss gives the density, pqgauss gives the distribution function, cqgauss gives the quantile function, and rqgauss generates random deviates.

### Author(s)

Emerson Luis de Santa Helena , Wagner Santos de Lima

### References

Thistleton, W., Marsh, J. A., Nelson, K., Tsallis, C., (2007) IEEE Transactions on Information Theory, 53(12):4805

Tsallis, C., (2009) Introduction to Nonextensive Statistical Mechanics. Springer.

de Santa Helena, E. L., Nascimento, C. M., and Gerhardt, G. J., (2015) Alternative way to characterize a q-gaussian distribution by a robust heavy tail measurement. Physica A, (435):44-50.

Manuscript submitted for publication (2016) qGaussian: Tools to Explore Applications of Tsallis Statistics

**See Also**

Distributions for other standard distributions, including dt and dcauchy. [Distributions](#)

**Examples**

```

qv <- c(2.8,2.5,2,1.01,0,-5); nn <- 700
xrg <- sqrt((3-qv[6])/(1-qv[6]))
xr <- seq(-xrg,xrg,by=2*xrg/nn)
y0 <- dqgauss(xr,qv[6])
plot(xr,y0,ty='l',xlim=range(-4.5,4.5),ylab='p(x)',xlab='x')
for (i in 1:5){
if (qv[i]< 1) xrg <- sqrt((3-qv[i])/(1-qv[i]))
else xrg <- 4.5
vby <- 2*xrg/nn
xr <- seq(-xrg,xrg,by=2*xrg/nn)
y0 <- dqgauss(xr,qv[i])
points (xr,y0,ty='l',col=(i+1))
}
legend(2, 0.4, legend =c(expression(paste(q==5)),expression(paste(q=0)),
expression(paste(q==1.01)),expression(paste(q=2)),expression(paste(q==2.5)),
expression(paste(q==2.8))),col = c(1,6,5,4,3,2), lty = c(1,1,1,1,1,1))
#####
qv <- 0
rr <- rqgauss(2^16,qv)
nn <- 70
xrg <- sqrt((3-qv)/(1-qv))
vby <- 2*xrg/(nn)
xr <- seq(-xrg,xrg,by=vby)
hist (rr,breaks=xr,freq=FALSE,xlab="x",main='')
y <- dqgauss(xr)
lines(xr,y/sum(y*vby),cex=.5,col=2,lty=4)

```

---

dqgauss

*The q-gaussian Distribution*


---

**Description**

Density, distribution function, quantile function and random generation for the q-gaussian distribution with parameters mu and sig.

**Usage**

```

dqgauss(x, q = 0, mu = 0, sig = 1)

```

**Arguments**

x	vector of quantiles.
q	entropic index.
mu	a value for q-mean.
sig	a value for q-variance.

**Details**

If q, mu and sig values are not specified, they assume the default values of 0, 0 and 1, respectively. Defining  $Z=(q-1)/(3-q)$ , the q-gaussian distribution has density written as

$$p(x) = (\text{sig} * \text{Beta}(\alpha/2, 1/2))^{-1} * (1 + Z(x - \mu)^2 / \text{sig}^2)^{-(1 + 1/Z)/2}$$

where  $\alpha = 1 - 1/Z$  when  $q < 1$  and  $1/Z$  when  $1 < q < 3$ .

**Value**

dqgauss gives the density, pqgauss gives the distribution function, cqgauss gives the quantile function, and rqgauss generates random deviates.

**Author(s)**

Emerson Luis de Santa Helena, Wagner Santos de Lima

**References**

Thistleton, W., Marsh, J. A., Nelson, K., Tsallis, C., (2007) IEEE Transactions on Information Theory, 53(12):4805

Tsallis, C., (2009) Introduction to Nonextensive Statistical Mechanics. Springer.

de Santa Helena, E. L., Nascimento, C. M., and Gerhardt, G. J., (2015) Alternative way to characterize a q-gaussian distribution by a robust heavy tail measurement. Physica A, (435):44-50.

Manuscript submitted for publication (2016) qGaussian: Tools to Explore Applications of Tsallis Statistics

**See Also**

Distributions for other standard distributions, including dt and dcauchy. [Distributions](#)

**Examples**

```

qv <- c(2.8, 2.5, 2, 1.01, 0, -5); nn <- 700
xrg <- sqrt((3-qv[6])/(1-qv[6]))
xr <- seq(-xrg, xrg, by=2*xrg/nn)
y0 <- dqgauss(xr, qv[6])
plot(xr, y0, ty='l', xlim=range(-4.5, 4.5), ylab='p(x)', xlab='x')
for (i in 1:5){
  if (qv[i] < 1) xrg <- sqrt((3-qv[i])/(1-qv[i]))
  else xrg <- 4.5
  vby <- 2*xrg/nn
}

```

```

xr <- seq(-xrg,xrg,by=2*xrg/nn)
y0 <- dqgauss(xr,qv[i])
points (xr,y0,ty='l',col=(i+1))
}
legend(2, 0.4, legend =c(expression(paste(q==5)),expression(paste(q==0)),
expression(paste(q==1.01)),expression(paste(q==2)),expression(paste(q==2.5)),
expression(paste(q==2.8))),col = c(1,6,5,4,3,2), lty = c(1,1,1,1,1,1))
#####

qv <- 0
rr <- rqgauss(2^16,qv)
nn <- 70
xrg <- sqrt((3-qv)/(1-qv))
vby <- 2*xrg/(nn)
xr <- seq(-xrg,xrg,by=vby)
hist (rr,breaks=xr,freq=FALSE,xlab="x",main='')
y <- dqgauss(xr)
lines(xr,y/sum(y*vby),cex=.5,col=2,lty=4)

```

---

pqgauss

*The q-gaussian Distribution*


---

## Description

Density, distribution function, quantile function and random generation for the q-gaussian distribution with parameters mu and sig.

## Usage

```
pqgauss(x, q = 0, mu = 0, sig = 1, lower.tail = TRUE)
```

## Arguments

x	vector of quantiles.
q	entropic index.
mu	a value for q-mean.
sig	a value for q-variance.
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$ .

## Details

If q, mu and sig values are not specified, they assume the default values of 0, 0 and 1, respectively. Defining  $Z=(q-1)/(3-q)$ , the q-gaussian distribution has density written as

$$p(x) = (\text{sig} * \text{Beta}(\alpha/2, 1/2))^{-1} * (1 + Z(x - \mu)^2 / \text{sig}^2)^{-(1 + 1/Z)/2}$$

where  $\alpha = 1 - 1/Z$  when  $q < 1$  and  $1/Z$  when  $1 < q < 3$ .

**Value**

dqgauss gives the density, pqgauss gives the distribution function, cqgauss gives the quantile function, and rqgauss generates random deviates.

**Author(s)**

Emerson Luis de Santa Helena , Wagner Santos de Lima

**References**

Thistleton, W., Marsh, J. A., Nelson, K., Tsallis, C., (2007) IEEE Transactions on Information Theory, 53(12):4805

Tsallis, C., (2009) Introduction to Nonextensive Statistical Mechanics. Springer.

de Santa Helena, E. L., Nascimento, C. M., and Gerhardt, G. J., (2015) Alternative way to characterize a q-gaussian distribution by a robust heavy tail measurement. Physica A, (435):44-50.

Manuscript submitted for publication (2016) qGaussian: Tools to Explore Applications of Tsallis Statistics

**See Also**

Distributions for other standard distributions, including dt and dcauchy. [Distributions](#)

**Examples**

```

qv <- c(2.8,2.5,2,1.01,0,-5); nn <- 700
xrg <- sqrt((3-qv[6])/(1-qv[6]))
xr <- seq(-xrg,xrg,by=2*xrg/nn)
y0 <- dqgauss(xr,qv[6])
plot(xr,y0,ty='l',xlim=range(-4.5,4.5),ylab='p(x)',xlab='x')
for (i in 1:5){
  if (qv[i]< 1) xrg <- sqrt((3-qv[i])/(1-qv[i]))
  else xrg <- 4.5
  vby <- 2*xrg/nn
  xr <- seq(-xrg,xrg,by=2*xrg/nn)
  y0 <- dqgauss(xr,qv[i])
  points (xr,y0,ty='l',col=(i+1))
}
legend(2, 0.4, legend =c(expression(paste(q==5)),expression(paste(q=0)),
expression(paste(q=1.01)),expression(paste(q=2)),expression(paste(q=2.5)),
expression(paste(q=2.8))),col = c(1,6,5,4,3,2), lty = c(1,1,1,1,1,1))
#####

qv <- 0
rr <- rqgauss(2^16,qv)
nn <- 70
xrg <- sqrt((3-qv)/(1-qv))
vby <- 2*xrg/(nn)
xr <- seq(-xrg,xrg,by=vby)
hist (rr,breaks=xr,freq=FALSE,xlab="x",main='')
y <- dqgauss(xr)
lines(xr,y/sum(y*vby),cex=.5,col=2,lty=4)

```

---

`qbymc`*qbymc, a q value estimator founded upon medcouple.*

---

**Description**

Given a random data set, the 'qbymc' uses the medcouple, a robust measure of tail weights, to yield a 'q' value, a characteristic entropic index of the q-gaussian distributions.

**Usage**`qbymc(x)`**Arguments**

`x` numeric vector

**Value**

a number  $q < 3$ , and the standard error.

**Author(s)**

Emerson Luis de Santa Helena , Wagner Santos de Lima

**References**

de Santa Helena, E. L., Nascimento, C. M., and Gerhardt, G. J., (2015) Alternative way to characterize a q-gaussian distribution by a robust heavy tail measurement. *Physica A*, (435):44-50.

**See Also**

Robustbase for medcouple. [mc](#)

**Examples**

```
set.seed(0002)
rr <- rqgauss(1000, 1.333)
qbymc(rr)
```

rqgauss

*The q-gaussian Distribution***Description**

Density, distribution function, quantile function and random generation for the q-gaussian distribution with parameters mu and sig.

**Usage**

```
rqgauss(n, q = 0, mu = 0, sig = 1, meth = "Box-Muller")
```

**Arguments**

n	number of observations. If length(n) > 1, the length is taken to be the number required.
q	entropic index.
mu	a value for q-mean.
sig	a value for q-variance.
meth	method used at random generator

**Details**

If q, mu and sig values are not specified, they assume the default values of 0, 0 and 1, respectively. Defining  $Z=(q-1)/(3-q)$ , the q-gaussian distribution has density written as

$$p(x) = (\text{sig} * \text{Beta}(\alpha/2, 1/2))^{-1} * (1 + Z(x - \mu)^2 / \text{sig}^2)^{-(1 + 1/Z)/2}$$

where  $\alpha = 1 - 1/Z$  when  $q < 1$  and  $1/Z$  when  $1 < q < 3$ .

For different methods use: meth = "Chaotic", meth = "Quantile" and meth = "Box-Muller"

**Value**

dqgauss gives the density, pqgauss gives the distribution function, cqgauss gives the quantile function, and rqgauss generates random deviates.

**Author(s)**

Emerson Luis de Santa Helena, Wagner Santos de Lima

**References**

Umeno, K., Sato, A., IEEE Transactions on Information Theory (Volume:59, Issue:5, May 2013). Chaotic Method for Generating q-Gaussian Random Variables.

Thistleton, W., Marsh, J. A., Nelson, K., Tsallis, C., (2007) IEEE Transactions on Information Theory, 53(12):4805

Tsallis, C., (2009) Introduction to Nonextensive Statistical Mechanics. Springer.

de Santa Helena, E. L., Nascimento, C. M., and Gerhardt, G. J., (2015) Alternative way to characterize a q-gaussian distribution by a robust heavy tail measurement. *Physica A*, (435):44-50.

de Lima, Wagner S., de Santa Helena, E. L., qGaussian: Tools to Explore Applications of Tsallis Statistics. arXiv:1703.06172

### See Also

Distributions for other standard distributions, including dt and dcauchy. [Distributions](#)

### Examples

```

qv <- c(2.8,2.5,2,1.01,0,-5); nn <- 700
xrg <- sqrt((3-qv[6])/(1-qv[6]))
xr <- seq(-xrg,xrg,by=2*xrg/nn)
y0 <- dqgauss(xr,qv[6])
plot(xr,y0,ty='l',xlim=range(-4.5,4.5),ylab='p(x)',xlab='x')
for (i in 1:5){
if (qv[i]< 1) xrg <- sqrt((3-qv[i])/(1-qv[i]))
else xrg <- 4.5
vby <- 2*xrg/nn
xr <- seq(-xrg,xrg,by=2*xrg/nn)
y0 <- dqgauss(xr,qv[i])
points (xr,y0,ty='l',col=(i+1))
}
legend(2, 0.4, legend =c(expression(paste(q==5)),expression(paste(q=0)),
expression(paste(q=1.01)),expression(paste(q=2)),expression(paste(q=2.5)),
expression(paste(q=2.8))),col = c(1,6,5,4,3,2), lty = c(1,1,1,1,1,1))
#####

qv <- 0
rr <- rqgauss(2^16,qv)
nn <- 70
xrg <- sqrt((3-qv)/(1-qv))
vby <- 2*xrg/(nn)
xr <- seq(-xrg,xrg,by=vby)
hist (rr,breaks=xr,freq=FALSE,xlab="x",main='')
y <- dqgauss(xr)
lines(xr,y/sum(y*vby),cex=.5,col=2,lty=4)

```

# Index

Chaotic, [2](#)  
cqgauss, [3](#)

Distributions, [2](#), [4](#), [5](#), [7](#), [10](#)  
dqgauss, [4](#)

mc, [8](#)

pqgauss, [6](#)

qbymc, [8](#)

rqgauss, [9](#)