

# Package ‘QF’

July 21, 2025

**Title** Density, Cumulative and Quantile Functions of Quadratic Forms

**Version** 0.0.9

**Description** The computation of quadratic form (QF) distributions is often not trivial and it requires numerical routines. The package contains functions aimed at evaluating the exact distribution of quadratic forms (QFs) and ratios of QFs. In particular, we propose to evaluate density, quantile and distribution functions of positive definite QFs and ratio of independent positive QFs by means of an algorithm based on the numerical inversion of Mellin transforms.

**License** GPL-3

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**LinkingTo** Rcpp, RcppGSL

**Imports** Rcpp

**SystemRequirements** GNU GSL

**NeedsCompilation** yes

**Date** 2025-03-27

**Author** Aldo Gardini [aut, cre],  
Fedele Greco [aut],  
Carlo Trivisano [aut]

**Maintainer** Aldo Gardini <aldo.gardini2@unibo.it>

**Repository** CRAN

**Date/Publication** 2025-03-27 15:30:06 UTC

## Contents

compute_MellinQF . . . . .	2
compute_MellinQF_ratio . . . . .	4
pQF_depratio . . . . .	6
print.MellinQF . . . . .	7
print.MellinQF_ratio . . . . .	8
QF . . . . .	8
QF_ratio . . . . .	10

<b>Index</b>	<b>12</b>
--------------	-----------

---

compute\_MellinQF      *Mellin Transform of a Positive QF*

---

### Description

The function computes the Mellin transform of a positive definite quadratic form producing a MellinQF object. The output can be used to evaluate the density, cumulative and quantile functions of the target quadratic form.

### Usage

```
compute_MellinQF(
  lambdas,
  etas = rep(0, length(lambdas)),
  eps = 1e-06,
  rho = 1 - 1e-04,
  maxit_comp = 1e+05,
  eps_quant = 1e-06,
  maxit_quant = 10000,
  lambdas_tol = NULL
)
```

### Arguments

lambdas	vector of positive weights.
etas	vector of non-centrality parameters. Default all zeros (central chi square).
eps	required absolute error for density and cumulative functions.
rho	distribution total probability mass for which it is desired to keep the error eps.
maxit_comp	maximum number of iterations.
eps_quant	required numerical error for quantile computation.
maxit_quant	maximum number of iterations before stopping the quantile computation.
lambdas_tol	maximum value admitted for the weight skewness. When it is not NULL (default), elements of lambdas such that the ratio $\max(\text{lambdas})/\text{lambdas}$ is greater than the specified value are removed.

### Details

The quadratic form having positive weights `lambdas` and non-centrality parameters `etas` is considered:

$$Q = \sum_{i=1}^r \lambda_i \chi_{1, \eta_i}^2.$$

Its Mellin transform is computed by exploiting the density formulation by Ruben (1962). The numerical error is controlled in order to provide the requested precision (`eps`) for the interval of quantiles that contains the specified total probability `rho`.

The argument `eps_quant` controls the relative precision requested for the computation of quantiles that determine the range in which the error `eps` is guaranteed, whereas `maxit_quant` sets the maximum number of Newton-Raphson iterations of the algorithm.

### Value

The function returns an object of the class `MellinQF` that contains information on the Mellin transform of a linear combination of positively weighted chi-square random variables. This information can be used in order to evaluate the density, cumulative distribution and quantile functions.

An object of the class `MellinQF` has the following components:

- `range_q`: the range of quantiles that contains the specified mass of probability `rho` in which it is possible to compute density and CDF preserving the error level `eps`.
- `Mellin`: a list containing the values of the Mellin transform (`Mellin`), the corresponding evaluation points (`z`), the integration step `delta` and the lowest weight (`lambda_min`).
- the inputs `rho`, `lambdas`, `etas`, `eps` needed for CDF, PDF and quantile function computation.

### Source

Ruben, Harold. "Probability content of regions under spherical normal distributions, IV: The distribution of homogeneous and non-homogeneous quadratic functions of normal variables." *The Annals of Mathematical Statistics* 33.2 (1962): 542-570.

### See Also

The function `print.MellinQF` can be used to summarize the basic information on the Mellin transform.

The object can be used in the function `dQF` to compute the density function of the QF, `pQF` for the CDF and `qQF` for the quantile function.

### Examples

```
library(QF)
# Definition of the QF
lambdas_QF <- c(rep(7, 6), rep(3, 2))
etas_QF <- c(rep(6, 6), rep(2, 2))
# Computation Mellin transform
eps <- 1e-7
rho <- 0.999
Mellin <- compute_MellinQF(lambdas_QF, etas_QF, eps = eps, rho = rho)
print(Mellin)
```

---

 compute\_MellinQF\_ratio

*Mellin Transform of the Independent Positive QFs Ratio*


---

### Description

The function computes the Mellin transform of the ratio of independent and positive definite quadratic forms producing a MellinQF\_ratio object. The output can be used to evaluate the density, cumulative and quantile functions of the target quadratic form.

### Usage

```
compute_MellinQF_ratio(
  lambdas_num,
  lambdas_den,
  etas_num = rep(0, length(lambdas_num)),
  etas_den = rep(0, length(lambdas_den)),
  eps = 1e-06,
  rho = 1 - 1e-04,
  maxit_comp = 1e+05,
  eps_quant = 1e-06,
  maxit_quant = 10000,
  lambdas_tol = NULL
)
```

### Arguments

lambdas_num	vector of positive weights for the numerator.
lambdas_den	vector of positive weights for the denominator.
etas_num	vector of non-centrality parameters for the numerator. Default all zeros (central chi square).
etas_den	vector of non-centrality parameters for the denominator. Default all zeros (central chi square).
eps	required absolute error for density and cumulative functions.
rho	distribution total probability mass for which it is desired to keep the error eps.
maxit_comp	maximum number of iterations.
eps_quant	required numerical error for quantile computation.
maxit_quant	maximum number of iterations before stopping the quantile computation.
lambdas_tol	maximum value admitted for the weight skewness for both the numerator and the denominator. When it is not NULL (default), elements of lambdas such that the ratio $\max(\text{lambdas})/\text{lambdas}$ is greater than the specified value are removed.

## Details

The Mellin transform of the ratio of two independent quadratic forms having positive weights `lambdas_num` and `lambdas_den` and non-centrality parameters `etas_num` and `etas_den` is computed exploiting the density formulation by Ruben (1962). The numerical error is controlled in order to provide the requested precision (`eps`) for the interval of quantiles that contains the specified total probability `rho`.

The argument `eps_quant` controls the relative precision requested for the computation of quantiles that determine the range in which the error `eps` is guaranteed, whereas `maxit_quant` sets the maximum number of Newton-Raphson iterations of the algorithm.

## Value

The function returns an object of the class `MellinQF_ratio` that contains information on the Mellin transform of the ratio of two linear combinations of positively weighted chi-square random variables. This information can be used in order to evaluate the density, cumulative distribution and quantile functions of the desired quadratic form.

An object of the class `MellinQF_ratio` has the following components:

- `range_q`: the range of quantiles that contains the specified mass of probability `rho` in which it is possible to compute density and CDF preserving the error level `eps`.
- `Mellin`: a list containing the values of the Mellin transform (`Mellin`), the corresponding evaluation points (`z`), the integration step `delta` and the lowest numerator weight (`lambda_min`).
- the inputs `rho`, `lambdas_num`, `lambdas_den`, `etas_num`, `etas_den`, `eps` needed for CDF, PDF and quantile function computation.

## Source

Ruben, Harold. "Probability content of regions under spherical normal distributions, IV: The distribution of homogeneous and non-homogeneous quadratic functions of normal variables." *The Annals of Mathematical Statistics* 33.2 (1962): 542-570.

## See Also

The function `print.MellinQF_ratio` can be used to summarize the basic information on the Mellin transform.

The object can be used in the function `dQF_ratio` to compute the density function of the QFs ratio, `pQF_ratio` for the CDF and `qQF_ratio` for the quantile function.

## Examples

```
library(QF)
# Definition of the QFs
lambdas_QF_num <- c(rep(7, 6), rep(3, 2))
etas_QF_num <- c(rep(6, 6), rep(2, 2))
lambdas_QF_den <- c(0.6, 0.3, 0.1)
# Computation Mellin transform
eps <- 1e-7
```

```
rho <- 0.999
Mellin_ratio <- compute_MellinQF_ratio(lambdas_QF_num, lambdas_QF_den,
                                     etas_QF_num, eps = eps, rho = rho)
print(Mellin_ratio)
```

---

pQF\_depratio

*Cumulative Distribution Function of the Dependent QFs Ratio*


---

### Description

The function computes the CDF of the ratio of two dependent and possibly indefinite quadratic forms.

### Usage

```
pQF_depratio(
  q = NULL,
  lambdas = NULL,
  A = NULL,
  B = NULL,
  eps = 1e-06,
  maxit_comp = 1e+05,
  lambdas_tol = NULL
)
```

### Arguments

q	vector of quantiles.
lambdas	vector of eigenvalues of the matrix (A-qB).
A	matrix of the numerator QF. If not specified but B is passed, it is assumed to be the identity.
B	matrix of the numerator QF. If not specified but A is passed, it is assumed to be the identity.
eps	requested absolute error.
maxit_comp	maximum number of iterations.
lambdas_tol	maximum value admitted for the weight skewness for both the numerator and the denominator. When it is not NULL (default), elements of lambdas such that the ratio $\max(\text{lambdas})/\text{lambdas}$ is greater than the specified value are removed.

**Details**

The distribution function of the following ratio of dependent quadratic forms is computed:

$$P\left(\frac{Y^T A Y}{Y^T B Y} < q\right),$$

where  $Y \sim N(0, I)$ .

The transformation to the following indefinite quadratic form is exploited:

$$P(Y^T(A - qB)Y < 0).$$

The following inputs can be provided:

- vector lambdas that contains the eigenvalues of the matrix  $(A - qB)$ . Input  $q$  is ignored.
- matrix A and/or matrix B: in these cases  $q$  is required to be not null and an eventual missing specification of one matrix make it equal to the identity.

**Value**

The values of the CDF at quantiles  $q$ .

---

print.MellinQF	<i>Printing MellinQF</i>
----------------	--------------------------

---

**Description**

It allows to visualize useful information on the MellinQF object.

**Usage**

```
## S3 method for class 'MellinQF'
print(x, digits = 3L, ...)
```

**Arguments**

x	MellinQF object.
digits	number of digits to display in the output.
...	potentially more arguments passed to methods.

---

```
print.MellinQF_ratio Print MellinQF_ratio
```

---

### Description

It allows to visualize useful information on the MellinQF\_ratio object.

### Usage

```
## S3 method for class 'MellinQF_ratio'
print(x, digits = 3L, ...)
```

### Arguments

x	MellinQF_ratio object.
digits	number of digits to display in the output.
...	potentially more arguments passed to methods.

---

QF *Positive Definite Quadratic Forms Distribution*

---

### Description

Density function, distribution function, quantile function and random generator for positive definite QFs.

### Usage

```
dQF(x, obj)

pQF(q, obj)

qQF(p, obj, eps_quant = 1e-06, maxit_quant = 10000)

rQF(n, lambdas, etas = rep(0, length(lambdas)))
```

### Arguments

x, q	vector of quantiles.
obj	MellinQF object produced by the <a href="#">compute_MellinQF</a> function.
p	vector of probabilities.
eps_quant	relative error for quantiles.
maxit_quant	maximum number of Newton-Raphson iterations allowed to compute quantiles.
n	number of observations.
lambdas	vector of positive weights.
etas	vector of non-centrality parameters. Default all zeros.



## Details

The quadratic form CDF and PDF are evaluated by numerical inversion of the Mellin transform. The absolute error specified in `compute_MellinQF` is guaranteed for values of  $q$  and  $x$  inside the `range_q`. If the quantile is outside `range_q`, computations are carried out, but a warning is sent.

The function uses the Newton-Raphson algorithm to compute the QF quantiles related to probabilities  $p$ .

## Value

`dQF` provides the values of the density function at a quantile  $x$ .

`pQF` provides the cumulative distribution function at a quantile  $q$ .

`qQF` provides the quantile corresponding to a probability level  $p$ .

`rQF` provides a sample of  $n$  independent realizations from the QF.

## See Also

See `compute_MellinQF` for details on the Mellin computation.

## Examples

```
library(QF)
# Definition of the QF
lambdas_QF <- c(rep(7, 6), rep(3, 2))
etas_QF <- c(rep(6, 6), rep(2, 2))
# Computation Mellin transform
eps <- 1e-7
rho <- 0.999
Mellin <- compute_MellinQF(lambdas_QF, etas_QF, eps = eps, rho = rho)
xs <- seq(Mellin$range_q[1], Mellin$range_q[2], l = 100)
# PDF
ds <- dQF(xs, Mellin)
plot(xs, ds, type="l")
# CDF
ps <- pQF(xs, Mellin)
plot(xs, ps, type="l")
# Quantile
qs <- qQF(ps, Mellin)
plot(ps, qs, type="l")
# Comparison computed quantiles vs real quantiles
plot((qs - xs) / xs, type = "l")
```

---

 QF\_ratio

*Ratio of Positive Definite Quadratic Forms Distribution*


---

**Description**

Density function, distribution function, quantile function and random generator for the ratio of positive definite QFs.

**Usage**

```
dQF_ratio(x, obj)

pQF_ratio(q, obj)

qQF_ratio(p, obj, eps_quant = 1e-06, maxit_quant = 10000)

rQF_ratio(
  n,
  lambdas_num,
  lambdas_den,
  etas_num = rep(0, length(lambdas_num)),
  etas_den = rep(0, length(lambdas_den))
)
```

**Arguments**

x, q	vector of quantiles.
obj	MellinQF_ratio object produced by the <a href="#">compute_MellinQF_ratio</a> function.
p	vector of probabilities.
eps_quant	relative error for quantiles.
maxit_quant	maximum number of Newton-Raphson iterations allowed to compute quantiles.
n	number of observations.
lambdas_num	vector of positive weights of the QF at the numerator.
lambdas_den	vector of positive weights of the QF at the denominator
etas_num	vector of non-centrality parameters of the QF at the numerator. Default all zeros.
etas_den	vector of non-centrality parameters of the QF at the denominator Default all zeros.

**Details**

The CDF and PDF of the ratio of positive QFs are evaluated by numerical inversion of the Mellin transform. The absolute error specified in [compute\\_MellinQF\\_ratio](#) is guaranteed for values of q and x inside range\_q. If the quantile is outside range\_q, computations are carried out, but a warning is sent.

The function uses the Newton-Raphson algorithm to compute the ratio of QFs quantiles related to probabilities p.

**Value**

dQF\_ratio provides the values of the density function at a quantile  $x$ .  
 pQF\_ratio provides the cumulative distribution function at a quantile  $q$ .  
 qQF\_ratio provides the quantile corresponding to a probability level  $p$ .  
 rQF\_ratio provides a sample of  $n$  independent realizations the QFs ratio.

**See Also**

See [compute\\_MellinQF\\_ratio](#) for details on the Mellin computation.

**Examples**

```

lambdas_QF_num <- c(rep(7, 6), rep(3, 2))
etas_QF_num <- c(rep(6, 6), rep(2, 2))
lambdas_QF_den <- c(0.6, 0.3, 0.1)
# Computation Mellin transform
eps <- 1e-7
rho <- 0.999
Mellin_ratio <- compute_MellinQF_ratio(lambdas_QF_num, lambdas_QF_den,
                                     etas_QF_num, eps = eps, rho = rho)
xs <- seq(Mellin_ratio$range_q[1], Mellin_ratio$range_q[2], l = 100)
# PDF
ds <- dQF_ratio(xs, Mellin_ratio)
plot(xs, ds, type="l")
# CDF
ps <- pQF_ratio(xs, Mellin_ratio)
plot(xs, ps, type="l")
# Quantile
qs <- qQF_ratio(ps, Mellin_ratio)
plot(ps, qs, type="l")
#Comparison computed quantiles vs real quantiles
plot((qs - xs) / xs, type = "l")

```

# Index

`compute_MellinQF`, [2](#), [8](#), [9](#)  
`compute_MellinQF_ratio`, [4](#), [10](#), [11](#)

`dQF`, [3](#)  
`dQF (QF)`, [8](#)  
`dQF_ratio`, [5](#)  
`dQF_ratio (QF_ratio)`, [10](#)

`pQF`, [3](#)  
`pQF (QF)`, [8](#)  
`pQF_depratio`, [6](#)  
`pQF_ratio`, [5](#)  
`pQF_ratio (QF_ratio)`, [10](#)  
`print.MellinQF`, [3](#), [7](#)  
`print.MellinQF_ratio`, [5](#), [8](#)

`QF`, [8](#)  
`QF_ratio`, [10](#)  
`qQF`, [3](#)  
`qQF (QF)`, [8](#)  
`qQF_ratio`, [5](#)  
`qQF_ratio (QF_ratio)`, [10](#)

`rQF (QF)`, [8](#)  
`rQF_ratio (QF_ratio)`, [10](#)