

# Package ‘qmix’

December 13, 2019

**Title** Finite Quantile Mixture Models

**Version** 0.1.2.0

**Author** Xiao Lu

**Maintainer** Xiao Lu <xiao.lu.research@gmail.com>

**Description** Estimates finite quantile mixture models using Markov chain Monte Carlo methods. The finite quantile mixture models include both fixed- and random-quantile specifications that are applicable to both continuous and binary dependent variables. Tools are available to assess convergence and summarize the estimation results.

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Biarch** true

**Depends** R (>= 3.4.0)

**Imports** methods, Formula, Rcpp (>= 0.12.0), rstan (>= 2.18.1),

**LinkingTo** BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0),  
rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

**SystemRequirements** GNU make

**RoxygenNote** 6.1.1

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2019-12-13 15:10:02 UTC

## R topics documented:

qmix-package . . . . .	2
coef.qmix . . . . .	2
dald . . . . .	3
inverse . . . . .	3
is.dichotomous . . . . .	4
pald . . . . .	4

plot.qmix . . . . .	5
plot_coef.qmix . . . . .	5
plot_trace.qmix . . . . .	6
print.qmix . . . . .	6
print_coef.qmix . . . . .	7
print_mcmc.qmix . . . . .	7
print_text.qmix . . . . .	8
qald . . . . .	8
qmix . . . . .	9
rald . . . . .	11

## Index 12

---

qmix-package	<i>'qmix': A R Package for Estimating Finite Quantile Mixture Models</i>
--------------	--

---

### Description

Estimates finite quantile mixture models with either fixed- or random-quantile specifications. The estimation is implemented using MCMC methods available in `rstan`.

### References

Lu, Xiao (2019). Beyond the Average: Conditional Hypothesis Testing with Quantile Mixture. Working Paper.

Stan Development Team (2019). RStan: the R interface to Stan. R package version 2.19.2. <https://mc-stan.org>

---

coef.qmix	<i>Extract coefficients from a qmix object</i>
-----------	--

---

### Description

Create a table of coefficient results from a `qmix` object.

### Usage

```
## S3 method for class 'qmix'
coef(object, ...)
```

### Arguments

object	A <code>qmix</code> object from running the main function <code>qmix</code> .
...	Further arguments to be passed according to <code>coef</code> .

### Value

A table of coefficients with their corresponding lower and upper bounds.

---

<code>dald</code>	<i>Probability density function of asymmetric Laplace distributions</i>
-------------------	---

---

**Description**

`dald` calculates probability densities of asymmetric Laplace distributions.

**Usage**

```
dald(x, mu, p, sigma)
```

**Arguments**

<code>x</code>	Random variable.
<code>mu</code>	Position parameter.
<code>p</code>	Quantile.
<code>sigma</code>	Scale parameter.

**Value**

probability density of `x`.

---

<code>inverse</code>	<i>Inverse function</i>
----------------------	-------------------------

---

**Description**

`inverse` generates inverse function of any given function.

**Usage**

```
inverse(f, mu, p, sigma, lower = -10000, upper = 10000)
```

**Arguments**

<code>f</code>	pald function
<code>mu</code>	Position parameter.
<code>p</code>	Quantile.
<code>sigma</code>	Scale parameter.
<code>lower</code>	Lower bound.
<code>upper</code>	Upper bound.

**Value**

inversed pald

---

<code>is.dichotomous</code>	<i>Check if a predictor is dichotomous, adopted from package circGLM</i>
-----------------------------	--

---

**Description**

Check if a predictor is dichotomous, adopted from package circGLM

**Usage**

```
is.dichotomous(x)
```

**Arguments**

<code>x</code>	A character or numerical vector to be tested.
----------------	---

**Value**

A logical, TRUE if the `x` has dummy coding (0, 1), FALSE otherwise.

---

<code>pald</code>	<i>Cumulative density function of asymmetric Laplace distributions</i>
-------------------	--

---

**Description**

`pald` calculates cumulative densities of asymmetric Laplace distributions.

**Usage**

```
pald(x, mu, p, sigma)
```

**Arguments**

<code>x</code>	Random variable.
<code>mu</code>	Position parameter.
<code>p</code>	Quantile.
<code>sigma</code>	Scale parameter.

**Value**

cumulative probability density of `x`.

---

plot.qmix	<i>Plot qmix object</i>
-----------	-------------------------

---

**Description**

General plot function for qmix objects, which dispatches the chosen type of plotting to the corresponding function.

**Usage**

```
## S3 method for class 'qmix'
plot(x, type = "coef", ...)
```

**Arguments**

x	A qmix object to be plotted.
type	Character string giving the type of plotting. The options are "trace" for trace plots, "coef" for coefficient plots. The default is "coef".
...	Additional arguments to be passed to subsequent plot functions (check the See Also section).

**Value**

None.

**See Also**

[plot\\_trace.qmix](#) and [plot\\_coef.qmix](#).

---

plot_coef.qmix	<i>Make coefficient plots for a qmix object</i>
----------------	---

---

**Description**

plot\_coef.qmix is used to produce coefficient plots from a qmix object.

**Usage**

```
plot_coef.qmix(object, ...)
```

**Arguments**

object	A qmix object from running the main function <a href="#">qmix</a> .
...	Additional parameters to be passed to <a href="#">stan_plot</a> .

**Value**

None.

---

plot_trace.qmix	<i>Trace plots for qmix</i>
-----------------	-----------------------------

---

**Description**

plot\_trace.qmix is used to produce trace plots from a qmix object from the main function [qmix](#).

**Usage**

```
plot_trace.qmix(object, ...)
```

**Arguments**

object	A qmix object from running the main function <a href="#">qmix</a> .
...	Additional parameters to be passed to <a href="#">traceplot</a> .

**Value**

None.

---

print.qmix	<i>Print returns from a qmix object</i>
------------	---

---

**Description**

General print function for qmix objects, which dispatches the chosen type of printing to the corresponding function.

**Usage**

```
## S3 method for class 'qmix'
print(x, type = "text", ...)
```

**Arguments**

x	A qmix object to be printed.
type	Character string giving the type of printing, such as "text", "mcmc", "coef".
...	Additional arguments to be passed to print functions (check the See Also section).

**Value**

None.

**See Also**

[print\\_text.qmix](#), [print\\_mcmc.qmix](#), [print\\_coef.qmix](#).

---

print_coef.qmix	<i>Print coefficients of a qmix object</i>
-----------------	--

---

**Description**

print\_coef.qmix prints out coefficients from a qmix object from running the main function `qmix`.

**Usage**

```
print_coef.qmix(object, digits = 3)
```

**Arguments**

object	A qmix object.
digits	Number of digits to display.

**Value**

None.

---

print_mcmc.qmix	<i>Print convergence diagnostics from a qmix object</i>
-----------------	---

---

**Description**

print\_mcmc.qmix prints a number of diagnostics about the convergence of a qmix objects.

**Usage**

```
print_mcmc.qmix(object, ...)
```

**Arguments**

object	A qmix object.
...	Additional arguments to be passed to the print function.

**Value**

None.

---

`print_text.qmix`      *Print the main results from a qmix object.*

---

**Description**

Print the main results from a qmix object.

**Usage**

```
print_text.qmix(object, digits = 3)
```

**Arguments**

<code>object</code>	A qmix object.
<code>digits</code>	Number of digits to display.

**Value**

None.

---

`qald`      *Quantile function of asymmetric Laplace distributions*

---

**Description**

qald calculates quantiles values of asymmetric Laplace distributions.

**Usage**

```
qald(y, mu, p, sigma)
```

**Arguments**

<code>y</code>	quantile value.
<code>mu</code>	Position parameter.
<code>p</code>	Quantile.
<code>sigma</code>	Scale parameter.

**Value**

quantile value.



**Description**

The main function for running the finite quantile mixture model. The function returns a `qmix` object that can be further investigated using standard functions such as `plot`, `print`, and `coef`. The model can be passed using a formula as in `lm()`. Convergence diagnostics can be performed using either `print(object, "mcmc")` or `plot(object, "mcmc")`.

**Usage**

```
qmix(formula, data, nmix = 3, design = "fixed", q = NULL,
      nsim = 1000, burnin = NULL, thin = 1, CIsz = 0.95,
      nchain = 1, seeds = 12345, offset = 1e-20, inverse_distr = FALSE)
```

**Arguments**

<code>formula</code>	An object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
<code>data</code>	A data frame containing the variables in the model.
<code>nmix</code>	The number of mixture components.
<code>design</code>	Quantile specification. Options include "fixed" and "random". The default choice is "fixed" which requires quantile inputs from the user.
<code>q</code>	The quantile value.
<code>nsim</code>	The number of iterations.
<code>burnin</code>	The number of burnin iterations.
<code>thin</code>	Thinning parameter.
<code>CIsz</code>	The size of posterior confidence interval.
<code>nchain</code>	The number of parallel chains.
<code>seeds</code>	Random seeds to replicate the results.
<code>offset</code>	Offset values to enhance sampling stability. The default value is 1e-20.
<code>inverse_distr</code>	If FALSE, the ALD will not be reversed. The default is FALSE.

**Value**

A `qmix` object. An object of class `qmix` contains the following elements

`Call` The matched call.

`formula` Symbolic representation of the model.

`nmix` Number of mixture components. If unspecified in the fixed-quantile specification, the value equals the number of quantiles specified. Otherwise, an error will be generated for the missing value.

design Options include "fixed" and "random" for fixed- and random-quantile specification.

q Quantiles in the fixed-quantile specification.

nsim Number of iterations.

Burnin Number of burnin iterations.

thin Thinning.

seeds Random seeds for reproducibility. The default is 12345.

CIsize Size of the posterior confidence interval.

inverse\_distr Indicating whether ALD should be inverted.

offset Offset to enhance stability in estimation. The default value is 1e-20.

data Data used.

x Independent variables.

y Dependent variables.

xnames Names of the independent variables.

stanfit Output from stan.

sampledf Posterior samples.

summaryout Summary of the posterior samples.

npars Number of covariates.

ulbs Upper and lower bounds based on the specified confidence interval.

means Mean estimates.

thetas Estimated proportions of each mixture component.

binarylogic Indicating whether the data contain a binary dependent variable.

## References

Lu, Xiao (2019). Beyond the Average: Conditional Hypothesis Testing with Quantile Mixture. Working Paper.

## Examples

```
# simulate a mixture of 2 ALDs
k <- 2
N <- 50
# true effects: -10 and 10 respectively for two mixture components
beta1 <- -10
beta2 <- 10
set.seed(34324)
x1 <- rnorm(N,0,1)
x2 <- rnorm(N,0,1)
xb1 <- x1*beta1
xb2 <- x2*beta2
y1 <- y2 <- NA
# quantiles at 0.1 and 0.9
p1 <- 0.1
```

```
p2 <- 0.9
for (i in 1:N){
  y1[i] <- rald(1,mu = xb1[i],p = p1,sigma = 1)
  y2[i] <- rald(1,mu = xb2[i],p = p2,sigma = 1)
}
y <- c(y1,y2)
x <- c(x1,x2)
dat <- as.data.frame(cbind(y,x))
# Estimate the model using both the fixed- and random-quantile specification
model <- qmix(y ~ x, data = dat, nmix = 2, design = "fixed", q = c(0.1, 0.9))
# Summary the results
coef(model)
print(model)
# check traceplots
plot(model)
```

---

rald

*Random number generator of asymmetric Laplace distributions*

---

### Description

rald generates random numbers from asymmetric Laplace distributions.

### Usage

```
rald(n, mu, p, sigma)
```

### Arguments

n	Number of random numbers to be generated.
mu	Position parameter.
p	Quantile.
sigma	Scale parameter.

### Value

random numbers.

# Index

coef, 2  
coef.qmix, 2

dald, 3

inverse, 3  
is.dichotomous, 4

pald, 4  
plot.qmix, 5  
plot\_coef.qmix, 5, 5  
plot\_trace.qmix, 5, 6  
print.qmix, 6  
print\_coef.qmix, 6, 7  
print\_mcmc.qmix, 6, 7  
print\_text.qmix, 6, 8

qald, 8  
qmix, 2, 5–7, 9  
qmix-package, 2

rald, 11

stan\_plot, 5

traceplot, 6