

Package: fmx (via r-universe)

July 23, 2026

Type Package

Title Finite Mixture Parametrization

Version 0.2.0

Date 2026-07-16

Description A parametrization framework for finite mixture distribution using S4 objects. Density, cumulative density, quantile and simulation functions are defined. Currently normal, Tukey g-&-h, skew-normal and skew-t distributions are well tested. The gamma, negative binomial distributions are being tested.

License GPL-2

Encoding UTF-8

Language en-US

Depends R (>= 4.6.0)

Imports methods, goftest, LaplacesDemon, rstpm2, sn, param2moment, TukeyGH77 (>= 0.2.0)

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Tingting Zhan [aut, cre] (ORCID:
<<https://orcid.org/0000-0001-9971-4844>>)

Maintainer Tingting Zhan <tingtingzhan@gmail.com>

Repository <https://cran.r-universe.dev>

Date/Publication 2026-07-23 08:41:35 UTC

RemoteUrl <https://github.com/cran/fmx>

RemoteRef HEAD

RemoteSha c33340bd7a249329abc9fb479930eb66ebb70966

Contents

fmx-package	2
as.fmx	2
fmx	3
fmx-class	3
Index	5

fmx-package	<i>Finite Mixture Parametrization</i>
-------------	---------------------------------------

Description

A parametrization framework for finite mixture distribution using S4 objects.
Density, cumulative density, quantile and simulation functions are defined.
Currently normal, Tukey *g*-&-*h*, skew-normal and skew-*t* distributions are well tested. The gamma, negative binomial distributions are being tested.

Author(s)

Maintainer: Tingting Zhan <tingtingzhan@gmail.com> ([ORCID](#))
Authors:
• Tingting Zhan <tingtingzhan@gmail.com> ([ORCID](#))

as.fmx	<i>Turn Various Objects to fmx Class</i>
--------	--

Description

Turn various objects created in other R packages to [fmx](#) class.

Usage

as.fmx(x, ...)

Arguments

x	an R object
...	additional parameters, see **Arguments** in individual S3 dispatches

Details

Various mixture distribution estimates obtained from other R packages are converted to [fmx](#) class, so that we could take advantage of all methods defined for [fmx](#) objects.

Value

S3 generic function [as.fmx()] returns an [fmx](#) object.

fmx	Create fmx Object for Finite Mixture Distribution
-----	---

Description

To create [fmx](#) object for finite mixture distribution.

Usage

```
fmx(distname, w = 1, ...)
```

Arguments

distname	character scalar
w	(optional) numeric vector . Does not need to sum up to 1; 'w/sum(w)' will be used internally.
...	mixture distribution parameters. See function dGH for the names and default values of Tukey <i>g</i> -&- <i>h</i> distribution parameters, or dnorm for the names and default values of normal distribution parameters.

Value

The function [fmx()] returns an [fmx](#) object.

fmx-class	fmx Class: Finite Mixture Parametrization
-----------	---

Description

An S4 object to specify the parameters and type of distribution of a one-dimensional finite mixture distribution.

Slots

distname [character](#) scalar, name of parametric distribution of the mixture components. Currently, normal ('norm') and Tukey *g*-&-*h* ('GH') distributions are supported.

pars [double matrix](#), all distribution parameters in the mixture. Each row corresponds to one component. Each column includes the same parameters of all components. The order of rows corresponds to the (non-strictly) increasing order of the component location parameters. The columns match the formal arguments of the corresponding distribution, e.g., 'mean' and 'sd' for [Normal](#) mixture, or 'A', 'B', 'g' and 'h' for Tukey *g*-&-*h* mixture.

w **numeric vector** of mixing proportions that must sum to 1

data (optional) **numeric vector**, the one-dimensional observations

data.name (optional) **character** scalar, a human-friendly name of the observations

vcov_internal (optional) variance-covariance **matrix** of the internal (i.e., unconstrained) estimates

vcov (optional) variance-covariance **matrix** of the mixture distribution (i.e., constrained) estimates

dist.ks (optional) **double** scalar, Kolmogorov-Smirnov distance, via [ks.test](#)

dist.cvm (optional) **double** scalars, Cramer von Mises distance, via [cvm.test](#)

dist.kl (optional) **double** scalars, Kullback-Leibler distance

logd (optional) **double vector**, point-wise **log**-density

logLik (optional) **logLik** object, log-likelihood

Index

`as.fmx`, [2](#)

`character`, [3](#), [4](#)

`cvm.test`, [4](#)

`dGH`, [3](#)

`dnorm`, [3](#)

`double`, [3](#), [4](#)

`fmx`, [2](#), [3](#), [3](#)

`fmx-class`, [3](#)

`fmx-package`, [2](#)

`ks.test`, [4](#)

`log`, [4](#)

`logLik`, [4](#)

`matrix`, [3](#), [4](#)

`Normal`, [3](#)

`numeric`, [3](#), [4](#)

`vector`, [3](#), [4](#)