

Package ‘FMCensSkewReg’

October 1, 2025

Title Finite Mixture of Censored Regression Models with Skewed Distributions

Version 0.1.0

Author Jiwon Park [aut, cre], Victor Hugo Lachos Davila [aut], Dipak Dey [aut]

Maintainer Jiwon Park <pcjylove87@gmail.com>

Description Provides an implementation of finite mixture regression models for censored data under four distributional families: Normal (FM-NCR), Student-t (FM-TCR), Skew-Normal (FM-SNCR), and Skew-t (FM-STCR). The package enables flexible modeling of skewness and heavy tails often observed in real-world data, while explicitly accounting for censoring. Functions are included for parameter estimation via the EM algorithm, computation of standard errors, and model comparison criteria such as AIC, BIC, and EDC.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports stats,
mvtnorm,
MomTrunc,
mnormt,
sn,
truncdist,
mixsmsn

Suggests testthat (>= 3.0.0),
knitr,
rmarkdown

Config/testthat/edition 3

R topics documented:

EM.skewCens.mixR 2

Index 4

Description

Fits finite mixture censored regression models under four families: Normal ("Normal"), Student-t ("T"), Skew-Normal ("SN"), and Skew-t ("ST").

Usage

```
EM.skewCens.mixR(
  cc,
  y,
  x,
  Abetas = NULL,
  sigma2 = NULL,
  shape = NULL,
  pii = NULL,
  nu = NULL,
  g = NULL,
  get.init = TRUE,
  criteria = TRUE,
  group = FALSE,
  family = "Normal",
  error = 1e-05,
  iter.max = 100,
  obs.prob = FALSE,
  kmeans.param = NULL,
  aitken = TRUE,
  IM = TRUE
)
```

Arguments

cc	Integer vector of length n; censoring indicator (1 = censored, 0 = observed).
y	Numeric response vector (univariate).
x	Numeric design matrix (n x p); include intercept column if needed.
Abetas	Optional initial regression coefficient matrix (p x g).
sigma2	Optional initial variance(s), length g.
shape	Optional initial skewness parameter(s), length g (used in SN/ST).
pii	Optional initial mixing proportions, length g, must sum to 1.
nu	Degrees of freedom for T/ST models (scalar).
g	Number of mixture components ($g \geq 1$). Required if get.init = TRUE.
get.init	Logical; if TRUE, k-means-based initialization is used.
criteria	Logical; if TRUE, returns AIC/BIC/EDC.
group	Logical; if TRUE, returns hard cluster labels.
family	One of "Normal", "T", "SN", "ST".

error	Convergence tolerance for EM.
iter.max	Maximum number of EM iterations.
obs.prob	Logical; if TRUE, returns posterior membership matrix.
kmeans.param	Optional list for kmeans init.
aitken	Logical; use Aitken acceleration for convergence monitoring.
IM	Logical; if TRUE, compute (robust) standard errors via information matrix.

Details

Left-censoring is indicated by `cc[i] = 1` and replacing `y[i]` by the censoring point. The routine supports Normal, t, Skew-Normal, and Skew-t families with finite mixtures.

Value

A list with elements:

Abetas	Estimated regression coefficients ($p \times g$).
sigma2	Estimated variances (length g).
shape	Estimated skewness parameters (length g ; SN/ST).
pii	Estimated mixing proportions (length g).
sd	Standard errors (if <code>IM=TRUE</code>).
nu	Estimated/used degrees of freedom (T/ST).
loglik	Final log-likelihood.
loglikT	Log-likelihood trace over iterations.
aic,bic,edc	Information criteria (if <code>criteria=TRUE</code>).
iter	Number of EM iterations.
n	Sample size.
group	Hard labels (if <code>group=TRUE</code>).

Examples

```
set.seed(1)
n <- 150
X <- cbind(1, runif(n), rnorm(n))
pi <- c(0.6, 0.4); nu <- 4
b1 <- c(0.5, 1.0, -1.0); sigma1 <- 1; shape1 <- 2
b2 <- c(1.0, -0.5, 0.5); sigma2 <- 2; shape2 <- 3
mu1 <- drop(X %*% b1); mu2 <- drop(X %*% b2)
draw1 <- function(i){
  a1 <- list(mu=mu1[i], sigma2=sigma1, shape=shape1, nu=nu)
  a2 <- list(mu=mu2[i], sigma2=sigma2, shape=shape2, nu=nu)
  mixsmn::rmix(1, pi, "Skew.t", list(a1,a2), cluster=FALSE)
}
y0 <- vapply(seq_len(n), draw1, numeric(1))
cutoff <- unname(stats::quantile(y0, 0.20))
cc <- as.integer(y0 <= cutoff)
y <- ifelse(cc == 1, cutoff, y0)
fit <- EM.skewCens.mixR(cc, y, X, g=2, family="Normal", iter.max=50)
fit$loglik
```

Index

EM.skewCens.mixR, [2](#)