Package 'UniCensorEM'

July 28, 2026

Type Package

Title EM Algorithm for Parameter Estimation Under Censoring Schemes

Version 0.1.0

Description Performs parameter estimation using the Expectation-Maximization (EM) algorithm of Dempster, Laird, and Rubin (1977) <doi:10.1111/j.2517-6161.1977.tb01600.x> for arbitrary univariate probability distributions under numerous censoring and truncation schemes. Users supply the probability density function (PDF), cumulative distribution function (CDF), survival function, initial parameter vector, support bounds, and observed data; the package automatically estimates parameters using the EM algorithm under the specified censoring scheme. Supported schemes include complete data, right censoring, left censoring, interval censoring, random censoring, block random censoring, Type-I censoring, Type-II censoring, progressive Type-II censoring (Balakrishnan and Aggarwala (2000, ISBN:978-1-4612-1334-5)), progressive first failure censoring, joint Type-I censoring, joint Type-II censoring, balanced joint progressive Type-II censoring, hybrid censoring, hybrid Type-I censoring, hybrid Type-II censoring, Type-I hybrid censoring, Type-II progressively hybrid censoring, doubly Type-II censoring, middle censoring, right truncation, and left truncation. Standard errors are computed via the observed information matrix (numerical Hessian), the Louis (1982) <doi:10.1111/j.2517-6161.1982.tb01203.x> method, and the supplemented EM (SEM) algorithm of Meng and Rubin (1991) <doi:10.1080/01621459.1991.10475130>. The package also provides confidence intervals, model selection criteria (AIC, BIC, AICc, HQIC), goodness-of-fit statistics (Kolmogorov-Smirnov, Cramer-von Mises, Anderson-Darling), residual analysis (randomized quantile residuals, generalized residuals), parametric bootstrap methods, Aitken acceleration for convergence improvement, and comprehensive visualization tools. References: Wu and Kus (2009) <doi:10.1016/j.csda.2009.03.010>, Kundu and Joarder (2006) <doi:10.1016/j.csda.2005.05.002>, Iyer, Jammalamadaka, and Kundu (2008) <doi:10.1016/j.jspi.2007.03.062>, Banerjee and Kundu (2008) <doi:10.1109/TR.2008.916890>, Prajapati, Mitra, and Kundu (2019) <doi:10.1007/s13571-018-0167-0>, Mondal and Kundu (2020) <doi:10.1080/03610926.2018.1554128>, Ding and Gui (2023) <doi:10.3390/math11092003>.

License GPL-3

Depends R (>= 4.0.0)

Imports stats, graphics, grDevices, utils, methods, numDeriv

Suggests testthat (>= 3.0.0), maxLik, bbmle, survival, boot, future, furrr, parallel, foreach, doParallel, ggplot2, cubature, pracma, knitr, rmarkdown

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

VignetteBuilder knitr

NeedsCompilation no

Author Shikhar Tyagi [aut, cre] (ORCID: <https://orcid.org/0000-0003-1606-0844>), Arvind Pandey [aut], Bhupendra Singh [aut], Vrijesh Tripathi [aut]

Maintainer Shikhar Tyagi <shikhar1093tyagi@gmail.com>

Repository CRAN

Date/Publication 2026-07-28 16:50:16 UTC

Contents

`AIC.emfit` *AIC method for emfit objects*

Description

Compute Akaike Information Criterion.

Usage

```
## S3 method for class 'emfit'
AIC(object, ...)
```

Arguments

`object`	an emfit object.
`...`	additional arguments (ignored).

Value

AIC value.

`BIC.emfit` *BIC method for emfit objects*

Description

Compute Bayesian Information Criterion.

Usage

```
## S3 method for class 'emfit'
BIC(object, ...)
```

Arguments

object	an emfit object.
...	additional arguments (ignored).

Value

BIC value.

---

bootstrap_se *Bootstrap standard errors for emfit objects*

Description

Compute bootstrap standard errors and confidence intervals.

Usage

```
bootstrap_se(
  object,
  B = 1000,
  type = "parametric",
  ci_type = "percentile",
  level = 0.95,
  parallel = FALSE,
  ...
)
```

Arguments

object	an emfit object.
B	number of bootstrap replicates (default 1000).
type	bootstrap type: "parametric" or "nonparametric" (default "parametric").
ci_type	confidence interval type: "normal", "percentile", "bca", or "basic" (default "percentile").
level	confidence level (default 0.95).
parallel	logical; use parallel computing (default FALSE).
...	additional arguments.

Value

List containing bootstrap estimates, standard errors, and confidence intervals.

coef.emfit *Coefficient method for emfit objects*

Description

Extract parameter estimates.

Usage

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

Arguments

object	an emfit object.
...	additional arguments (ignored).

Value

Named vector of parameter estimates.

compute_se *Compute standard errors for EM estimates*

Description

Compute standard errors for EM estimates

Usage

```
compute_se(
  theta_star,
  x,
  pdf,
  cdf,
  survival,
  scheme,
  status,
  time2,
  censor_params,
  lower,
  upper,
  method = "hessian",
  em_control = em.control()
)
```

Arguments

theta_star	converged parameter estimates.
x	observed data.
pdf	probability density function.
cdf	cumulative distribution function.
survival	survival function.
scheme	censoring scheme name.
status	censoring status vector.
time2	upper interval bounds.
censor_params	additional censoring parameters.
lower	lower support bound.
upper	upper support bound.
method	standard error estimation method: "hessian", "louis", or "sem".
em_control	control parameters for EM algorithm.

Value

A list containing:

hessian	the estimated observed information matrix
vcov	the variance-covariance matrix
se	the standard errors

---

confint.emfit *Confidence interval method for emfit objects*

Description

Compute confidence intervals for parameter estimates.

Usage

```
## S3 method for class 'emfit'
confint(object, parm = NULL, level = 0.95, ...)
```

Arguments

object	an emfit object.
parm	parameter indices (default: all parameters).
level	confidence level (default 0.95).
...	additional arguments (ignored).

Value

Matrix of confidence intervals.

dist_spec_em *Define a univariate distribution for EM estimation*

Description

Creates a reusable distribution specification for EM parameter estimation. At minimum, the PDF must be supplied. CDF and survival functions are required for most censoring schemes.

Usage

```
dist_spec_em(
  pdf = NULL,
  cdf = NULL,
  survival = NULL,
  lower = 0,
  upper = Inf,
  name = "custom"
)
```

Arguments

pdf	function; probability density function f(x, theta).
cdf	function; cumulative distribution function F(x, theta).
survival	function; survival function S(x, theta) = 1 - F(x, theta).
lower	numeric; lower support bound (default 0).
upper	numeric; upper support bound (default Inf).
name	character label for output (default "custom").

Value

An object of class "dist_spec_em", which is a list with elements pdf, cdf, survival (user-supplied functions or NULL), lower and upper (support bounds), and name (character label).

Examples

```
# Exponential(rate) distribution
exp_dist <- dist_spec_em(
  pdf = function(x, theta) theta[1] * exp(-theta[1] * x),
  cdf = function(x, theta) 1 - exp(-theta[1] * x),
  survival = function(x, theta) exp(-theta[1] * x),
  lower = 0, upper = Inf,
  name = "Exponential"
)
```

`em.control` *Control parameters for EM algorithm*

Description

Sets control parameters for the EM algorithm optimization.

Usage

```
em.control(
  maxit = 1000,
  tol = 1e-08,
  reltol = 1e-08,
  emtol = 1e-10,
  method = "BFGS",
  se_method = "hessian",
  trace = FALSE,
  aitken = FALSE,
  verbose = FALSE
)
```

Arguments

`maxit`	maximum number of EM iterations (default 1000).
`tol`	convergence tolerance for parameter changes (default 1e-8).
`reltol`	relative convergence tolerance (default 1e-8).
`emtol`	tolerance for Q-function convergence (default 1e-10).
`method`	optimization method for M-step: "BFGS", "L-BFGS-B", "Nelder-Mead", "CG", "SANN", "Brent", or "nlminb" (default "BFGS").
`se_method`	standard error estimation method: "hessian", "louis", or "sem" (default "hessian").
`trace`	logical; print iteration progress (default FALSE).
`aitken`	logical; use Aitken acceleration (default FALSE).
`verbose`	logical; detailed output (default FALSE).

Value

A list of control parameters.

Examples

```
ctrl <- em.control(maxit = 500, tol = 1e-6, method = "BFGS")
```

em_diagnostics *EM diagnostics*

Description

Compute diagnostic information for EM algorithm convergence.

Usage

```
em_diagnostics(object, ...)
```

Arguments

`object`	an emfit object.
`...`	additional arguments (ignored).

Value

List of diagnostic measures.

em_fit *Main EM algorithm for parameter estimation under censoring*

Description

Performs parameter estimation using the EM algorithm for arbitrary univariate distributions under specified censoring schemes.

Usage

```
em_fit(
  data,
  pdf,
  cdf = NULL,
  survival = NULL,
  theta0,
  scheme = "complete",
  status = NULL,
  time2 = NULL,
  censor_params = list(),
  lower = 0,
  upper = Inf,
  control = em.control()
)
```

Arguments

`data`	observed data vector or data frame with columns x and status.
`pdf`	probability density function f(x, theta).
`cdf`	cumulative distribution function F(x, theta).
`survival`	survival function S(x, theta) = 1 - F(x, theta).
`theta0`	initial parameter vector.
`scheme`	censoring scheme name (see details).
`status`	censoring status vector (optional, extracted from data if data frame).
`time2`	upper interval bounds for interval censoring (optional).
`censor_params`	list of additional censoring scheme parameters.
`lower`	lower support bound (default 0).
`upper`	upper support bound (default Inf).
`control`	list of control parameters from em.control().

Value

An object of class `"emfit"` containing:

`par`	parameter estimates
`value`	final log-likelihood
`counts`	number of iterations
`convergence`	convergence status (0 = success)
`message`	convergence message
`hessian`	observed information matrix
`std.error`	standard errors
`par_path`	parameter values at each iteration
`lik_path`	log-likelihood values at each iteration
`q_path`	Q-function values at each iteration
`time`	elapsed time in seconds
`scheme`	censoring scheme used
`data`	input data
`control`	control parameters

Supported censoring schemes

"complete" Complete (uncensored) data

"right_censor" Right censoring

"left_censor" Left censoring

"interval_censor" Interval censoring

"random_censor" Random censoring

"block_random_censor" Block random censoring

"type1_censor" Type-I (time-terminated) censoring

"type2_censor" Type-II (failure-terminated) censoring

"progressive_type2_censor" Progressive Type-II censoring

"progressive_first_failure" Progressive first failure censoring

"joint_type1_censor" Joint Type-I censoring

"joint_type2_censor" Joint Type-II censoring

"bjpt2_censor" Balanced joint progressive Type-II censoring

"hybrid_censor" Hybrid censoring

"hybrid_type1_censor" Hybrid Type-I censoring

"hybrid_type2_censor" Hybrid Type-II censoring

"type1_hybrid_censor" Type-I hybrid censoring

"progressive_hybrid_type2_censor" Type-II progressively hybrid censoring

"doubly_type2_censor" Doubly Type-II censoring

"middle_censor" Middle censoring

"right_truncation" Right truncation

"left_truncation" Left truncation

Examples

```
set.seed(123)
pdf <- function(x, theta) theta[1] * exp(-theta[1] * x)
cdf <- function(x, theta) 1 - exp(-theta[1] * x)
survival <- function(x, theta) exp(-theta[1] * x)

# Generate right-censored data
x_true <- rexp(50, rate = 1.5)
cens <- 1.0
status <- ifelse(x_true <= cens, 1, 0)
x_obs <- pmin(x_true, cens)

# Fit using EM
fit <- em_fit(
  data = x_obs,
  pdf = pdf,
  cdf = cdf,
  survival = survival,
  theta0 = c(1.0),
  scheme = "right_censor",
  status = status,
  lower = 0.001, upper = 100,
  control = em.control(maxit = 500, tol = 1e-6)
)

summary(fit)
```

fitted.emfit *Fitted values method for emfit objects*

Description

Extract fitted values (estimated survival probabilities at observed times).

Usage

```
## S3 method for class 'emfit'
fitted(object, ...)
```

Arguments

object	an emfit object.
...	additional arguments (ignored).

Value

Vector of fitted survival probabilities.

generalized_residuals *Generalized residuals*

Description

Compute generalized residuals for censored survival data.

Usage

```
generalized_residuals(object, ...)
```

Arguments

object	an emfit object.
...	additional arguments.

Value

Vector of generalized residuals.

get_ncores *Get number of cores*

Description

Get number of cores

Usage

```
get_ncores()
```

Value

Number of available cores.

gof_stats *Goodness-of-fit statistics*

Description

Compute various goodness-of-fit statistics for the fitted model.

Usage

```
gof_stats(object, ...)
```

Arguments

object	an emfit object.
...	additional arguments (ignored).

Value

Named vector of goodness-of-fit statistics.

logLik.emfit *Log-likelihood method for emfit objects*

Description

Extract the log-likelihood value.

Usage

```
## S3 method for class 'emfit'
logLik(object, ...)
```

Arguments

object	an emfit object.
...	additional arguments (ignored).

Value

Log-likelihood value.

model_selection *Model selection criteria for emfit objects*

Description

Compute various information criteria for model selection.

Usage

```
model_selection(object, ...)
```

Arguments

object	an emfit object.
...	additional arguments (ignored).

Value

Named vector of information criteria.

`observed_loglik` *Compute observed log-likelihood*

Description

Computes the log-likelihood of the observed data under the specified censoring scheme.

Usage

```
observed_loglik(
  data,
  pdf,
  cdf,
  survival,
  theta,
  scheme,
  status = NULL,
  time2 = NULL,
  censor_params = list(),
  lower = 0,
  upper = Inf
)
```

Arguments

`data`	observed data vector.
`pdf`	probability density function.
`cdf`	cumulative distribution function.
`survival`	survival function.
`theta`	parameter vector.
`scheme`	censoring scheme name.
`status`	censoring status vector (optional).
`time2`	upper interval bounds (optional for interval censoring).
`censor_params`	additional censoring scheme parameters (list).
`lower`	lower support bound.
`upper`	upper support bound.

Value

Log-likelihood value.

plot.emfit *Plot method for emfit objects*

Description

Create diagnostic plots for EM algorithm fit.

Usage

```
## S3 method for class 'emfit'
plot(x, which = c(1, 2, 3), ask = TRUE, ...)
```

Arguments

x	an emfit object.
which	which plots to display: 1 = parameter convergence, 2 = log-likelihood convergence, 3 = Q-function, 4 = histogram with fitted density, 5 = QQ plot, 6 = residual plot (default: c(1, 2, 3)).
ask	logical; pause between plots (default: TRUE).
...	additional graphical parameters.

Value

Invisibly, the emfit object.

plot_residuals *Residual diagnostics plot*

Description

Create diagnostic plots for residuals.

Usage

```
plot_residuals(object, type = "cox-snell", which = c(1, 2, 3), ...)
```

Arguments

object	an emfit object.
type	type of residuals (default "cox-snell").
which	which plots: 1 = histogram, 2 = QQ plot, 3 = residuals vs fitted, 4 = residuals vs order (default: c(1, 2, 3)).
...	additional graphical parameters.

Value

Invisibly, the emfit object.

predict.emfit *Predict method for emfit objects*

Description

Predict survival probabilities or quantiles for new data.

Usage

```
## S3 method for class 'emfit'
predict(object, newdata = NULL, type = "survival", ...)
```

Arguments

object	an emfit object.
newdata	new data values.
type	type of prediction: "survival", "cdf", or "quantile" (default "survival").
...	additional arguments.

Value

Predicted values.

print.bootstrap_emfit *Print method for bootstrap_emfit objects*

Description

Print method for bootstrap_emfit objects

Usage

```
## S3 method for class 'bootstrap_emfit'
print(x, ...)
```

Arguments

x	a bootstrap_emfit object.
...	additional arguments (ignored).

Value

Invisibly, the bootstrap_emfit object.

print.dist_spec_em *Print method for dist_spec_em*

Description

Print method for dist_spec_em

Usage

```
## S3 method for class 'dist_spec_em'
print(x, ...)
```

Arguments

x	a dist_spec_em object.
...	ignored.

Value

Invisibly, the input dist_spec_em object.

print.emfit *Print method for emfit objects*

Description

Print method for emfit objects

Usage

```
## S3 method for class 'emfit'
print(x, ...)
```

Arguments

x	an emfit object.
...	additional arguments (ignored).

Value

Invisibly, the emfit object.

print.em_diagnostics *Print method for em_diagnostics objects*

Description

Print method for em_diagnostics objects

Usage

```
## S3 method for class 'em_diagnostics'
print(x, ...)
```

Arguments

x	an em_diagnostics object.
...	additional arguments (ignored).

Value

Invisibly, the em_diagnostics object.

print.summary.emfit *Print method for summary.emfit objects*

Description

Print method for summary.emfit objects

Usage

```
## S3 method for class 'summary.emfit'
print(x, ...)
```

Arguments

x	a summary.emfit object.
...	additional arguments (ignored).

Value

Invisibly, the summary.emfit object.

`q_function` *Compute Q-function*

Description

Computes the Q-function (expected complete log-likelihood) for the EM algorithm.

Usage

```
q_function(
  theta,
  theta_old,
  data,
  pdf,
  cdf,
  survival,
  scheme,
  status = NULL,
  time2 = NULL,
  censor_params = list(),
  lower = 0,
  upper = Inf
)
```

Arguments

`theta`	parameter vector to maximize over.
`theta_old`	current parameter estimates.
`data`	observed data.
`pdf`	probability density function.
`cdf`	cumulative distribution function.
`survival`	survival function.
`scheme`	censoring scheme name.
`status`	censoring status vector.
`time2`	upper interval bounds.
`censor_params`	additional censoring parameters.
`lower`	lower support bound.
`upper`	upper support bound.

Value

Q-function value.

randomized_quantile_residuals *Randomized quantile residuals*

Description

Compute randomized quantile residuals for censored data.

Usage

```
randomized_quantile_residuals(object, n_sim = 1, ...)
```

Arguments

object	an emfit object.
n_sim	number of simulations for randomization (default 1).
...	additional arguments.

Value

Vector of randomized quantile residuals.

residuals.emfit *Residuals method for emfit objects*

Description

Compute residuals from the fitted model.

Usage

```
## S3 method for class 'emfit'
residuals(object, type = "cox-snell", ...)
```

Arguments

object	an emfit object.
type	type of residuals: "cox-snell", "martingale", "deviance", "pearson", or "response" (default "cox-snell").
...	additional arguments.

Value

Vector of residuals.

residual_analysis *Comprehensive residual analysis for emfit objects*

Description

Compute various types of residuals for diagnostic purposes.

Usage

```
residual_analysis(object, type = "cox-snell", ...)
```

Arguments

object	an emfit object.
type	type of residuals: "cox-snell", "martingale", "deviance", "pearson", "score", or "schoenfeld" (default "cox-snell").
...	additional arguments.

Value

Vector of residuals.

summary.emfit *Summary method for emfit objects*

Description

Summary method for emfit objects

Usage

```
## S3 method for class 'emfit'
summary(object, ...)
```

Arguments

object	an emfit object.
...	additional arguments (ignored).

Value

A list with summary information.

`vcov.emfit` *Variance-covariance matrix method for emfit objects*

Description

Extract the variance-covariance matrix of parameter estimates.

Usage

```
## S3 method for class 'emfit'
vcov(object, ...)
```

Arguments

`object`	an emfit object.
`...`	additional arguments (ignored).

Value

Variance-covariance matrix.

Index