Package 'gpciEMprogII'

August 8, 2026

Type Package

Title Generalized Process Capability Indices via EM Algorithm under Progressive Type-II Censoring

Version 0.1.0

Description Implements the Expectation-Maximization (EM) algorithm of Dempster, Laird, and Rubin (1977) <doi:10.1111/j.2517-6161.1977.tb01600.x> for parameter estimation under progressive Type-II censored data (Balakrishnan and Aggarwala (2000) <doi:10.1007/978-1-4612-1186-0>) and computes Generalized Process Capability Indices (GPCIs). Uses the 'UniCensorEM' package for EM estimation. Supports classical and generalized capability indices including Cpy (Maiti et al. (2010) <doi:10.1080/16843703.2010.11673233>), Cp, Cpk, Cpu, Cpl, Cpm, Cpmk, Spmk (Dey and Saha (2019) <doi:10.1007/s41872-019-00081-4>), CpTk (Saha et al. (2018) <doi:10.1080/21681015.2018.1437793>), Cpc, CNpmc (Alotaibi et al. (2022) <doi:10.1155/2022/3135264>), CNpmkc (Saha et al. (2024) <doi:10.1142/S021853932450013X>), and CNpk (Saha et al. (2022) <doi:10.1080/02664763.2021.1971632>). Computes point estimates, bias, mean squared error, risk, Heidelberger and Welch convergence diagnostics, convergence probability, and bootstrap confidence intervals at 90 percent, 95 percent, and 99 percent levels. Accommodates user-defined probability density or mass functions, cumulative distribution functions, and survival functions.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.0.0)

Imports stats, UniCensorEM

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

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

Maintainer Shikhar Tyagi <shikhar1093tyagi@gmail.com>

Repository CRAN

Date/Publication 2026-08-08 12:50:02 UTC

Contents

`boot_ci_prog_em` *Bootstrap Confidence Intervals for Progressive Type-II GPCI (EM)*

Description

Computes parametric or non-parametric bootstrap confidence intervals for Generalized Process Capability Indices and model parameters under progressive Type-II censoring using EM algorithm estimation. Provides bootstrap CIs at 90%, 95%, and 99% levels, plus estimates of bias, MSE, risk, Heidelberger-Welch convergence diagnostics, and convergence probability.

Usage

```
boot_ci_prog_em(
  fit,
  B = 1000,
  alpha = c(0.1, 0.05, 0.01),
  method = c("percentile", "normal", "basic", "BCp", "BCa", "studentized"),
  type = c("parametric", "nonparametric"),
  max_iter = 1000,
  tol = 1e-08
)

gpc_boot_em(
```

```
  fit,
  B = 1000,
  alpha = c(0.1, 0.05, 0.01),
  method = c("percentile", "normal", "basic", "BCp", "BCa", "studentized"),
  type = c("parametric", "nonparametric"),
  max_iter = 1000,
  tol = 1e-08
)
```

Arguments

fit	A gpc_em_fit object returned by capability_prog_em.
B	Number of bootstrap replicates (default 1000).
alpha	Numeric vector of significance levels (default c(0.10, 0.05, 0.01) for 90%, 95%, 99% CIs).
method	CI method: "percentile", "normal", "basic", "BCp", "BCa", or "studentized".
type	Resampling type: "parametric" (default) or "nonparametric".
max_iter	Maximum EM iterations per bootstrap (default 1000).
tol	Convergence tolerance per bootstrap (default 1e-8).

Value

An object of class "gpc_em_ci" containing:

ci_table Data frame of confidence intervals.

param_perf Data frame of parameter performance metrics (estimate, SE, bias, MSE, risk).

index_perf Data frame of index performance metrics.

convergence Data frame with Heidelberger-Welch diagnostics and convergence probability for each index.

param_reps Matrix of bootstrap parameter replicates.

index_reps Matrix of bootstrap index replicates.

fit Original fitted object.

B Number of bootstrap replicates.

Examples

```
my_dist <- dist_weibull(shape = 1, scale = 1)
x <- c(0.5, 1.2, 2.1, 3.4, 4.8)
r <- c(1, 0, 2, 0, 1)
fit <- capability_prog_em(x = x, r_removals = r,
  distribution = my_dist, USL = 6, LSL = 0)
ci <- boot_ci_prog_em(fit, B = 100,
  alpha = c(0.10, 0.05, 0.01), method = "percentile")
print(ci)
```

capability_prog_em *Compute Generalized Process Capability Indices via EM Algorithm*

Description

Computes classical and generalized Process Capability Indices (PCIs) for progressive Type-II censored data using model parameter estimates obtained from the EM algorithm.

Usage

```
capability_prog_em(
  x = NULL,
  r_removals = NULL,
  distribution,
  USL,
  LSL,
  target = (USL + LSL)/2,
 indices = c("Cpy", "Cp", "Cpk", "Cpu", "Cpl", "Cpm", "Cpmk", "Spmk", "CpTk", "Cpc",
    "CNpmc"),
  u = 1,
  v = 1,
  mode = c("moments", "quantile"),
  fit = TRUE,
  C0 = 1,
  C1 = 0,
  C2 = 1,
  tolerance_t = USL - LSL,
  P0 = 0.9973002,
  LDL = LSL,
  UDL = USL,
  max_iter = 1000,
  tol = 1e-08
)

gpc_em_fit(
  x = NULL,
  r_removals = NULL,
  distribution,
  USL,
  LSL,
  target = (USL + LSL)/2,
 indices = c("Cpy", "Cp", "Cpk", "Cpu", "Cpl", "Cpm", "Cpmk", "Spmk", "CpTk", "Cpc",
    "CNpmc"),
  u = 1,
  v = 1,
  mode = c("moments", "quantile"),
  fit = TRUE,
```

```
  C0 = 1,
  C1 = 0,
  C2 = 1,
  tolerance_t = USL - LSL,
  P0 = 0.9973002,
  LDL = LSL,
  UDL = USL,
  max_iter = 1000,
  tol = 1e-08
)
```

Arguments

x	Numeric vector of observed failure times under progressive Type-II censoring. Can be NULL if parameters are already fitted.
r_removals	Numeric vector of removal counts (censoring scheme R).
distribution	A gpc_dist distribution object.
USL	Upper Specification Limit.
LSL	Lower Specification Limit.
target	Process target (defaults to (USL + LSL) / 2).
indices	Character vector of capability indices to compute. Choices: "Cpy", "Cp", "Cpk", "Cpu", "Cpl", "Cpm", "Cpmk", "Cp_uv", "Spmk", "CpTk", "Cpc", "CNpmc", "CNpmkc", "CNpk", "Cp_q", "Cpk_q".
u	Weight parameter for Cp(u,v) family (default 1).
v	Weight parameter for Cp(u,v) family (default 1).
mode	Mode of evaluation: "moments" (mean and variance) or "quantile" (quantile-based).
fit	Logical. If TRUE (default), estimates distribution parameters using the EM algorithm before computing indices.
C0	Cost function parameter (default 1).
C1	Cost function parameter (default 0).
C2	Cost function parameter (default 1).
tolerance_t	Tolerance length (defaults to USL - LSL).
P0	Desired conformance level for Cpy and Cpc (default 0.9973002).
LDL	Lower Desired Limit for CpTk (defaults to LSL).
UDL	Upper Desired Limit for CpTk (defaults to USL).
max_iter	Maximum EM iterations (default 1000).
tol	Convergence tolerance for EM (default 1e-8).

Value

An object of class `c("gpc_em_fit", "gpcifit")` containing:

x Observed data (if provided).

r_removals Removal scheme (if provided).

distribution Fitted distribution object.

USL, LSL, target Specification limits and target.

indices Names of computed indices.

estimates Named numeric vector of index values.

p_hat Estimated nonconforming proportion.

mode Computation mode used.

options List of all computation options.

Examples

```
my_dist <- dist_weibull(shape = 1, scale = 1)
x <- c(0.5, 1.2, 2.1, 3.4, 4.8)
r <- c(1, 0, 2, 0, 1)
cap <- capability_prog_em(x = x, r_removals = r,
  distribution = my_dist, USL = 6, LSL = 0)
print(cap)
```

---

coef.gpc_em_fit *Coef Method for Progressive Type-II GPCI EM Fit*

Description

Coef Method for Progressive Type-II GPCI EM Fit

Usage

```
## S3 method for class 'gpc_em_fit'
coef(object, what = c("indices", "parameters"), ...)
```

Arguments

`object`	Object of class `gpc_em_fit`.
`what`	Extract `"indices"` (default) or `"parameters"`.
`...`	Additional arguments (currently unused).

Value

Named numeric vector of index values or parameter estimates.

`eval_performance_em` *Monte Carlo Simulation and Coverage Probability Evaluation*

Description

Evaluates standard error, MSE, bias, risk, and empirical coverage probability for parameters and GPCIs under progressive Type-II censoring using the EM algorithm with bootstrap CIs.

Usage

```
eval_performance_em(
  distribution,
  r_removals,
  USL,
  LSL,
  target = (USL + LSL)/2,
  true_indices = NULL,
  M = 100,
  B = 200,
  alpha = c(0.1, 0.05, 0.01),
  max_iter = 1000,
  tol = 1e-08
)

gpc_sim_em(
  distribution,
  r_removals,
  USL,
  LSL,
  target = (USL + LSL)/2,
  true_indices = NULL,
  M = 100,
  B = 200,
  alpha = c(0.1, 0.05, 0.01),
  max_iter = 1000,
  tol = 1e-08
)
```

Arguments

`distribution`	A `gpc_dist` distribution object with true parameter values.
`r_removals`	Progressive censoring scheme vector R.
`USL`	Upper Specification Limit.
`LSL`	Lower Specification Limit.
`target`	Process target (default midpoint).

true_indices	Optional named vector of true index values.
M	Number of Monte Carlo simulation runs (default 100).
B	Number of bootstrap iterations per run (default 200).
alpha	Significance levels (default c(0.10, 0.05, 0.01)).
max_iter	Maximum EM iterations (default 1000).
tol	Convergence tolerance (default 1e-8).

Value

A list containing coverage probabilities, M, B, true parameters, and true index values.

Examples

```
dist_true <- dist_weibull(shape = 2, scale = 5)
r_scheme <- c(2, 0, 3, 0, 1)
eval_res <- eval_performance_em(dist_true, r_scheme,
  USL = 10, LSL = 0, M = 5, B = 20)
print(eval_res)
```

---

fit_prog_ty2_em *Parameter Estimation via EM Algorithm for Progressive Type-II Censored Data*

Description

Fits process distribution parameters to progressive Type-II censored data using the Expectation-Maximization (EM) algorithm via the UniCensorEM package.

Usage

```
fit_prog_ty2_em(
  x,
  r_removals,
  distribution,
  start = NULL,
  max_iter = 1000,
  tol = 1e-08,
  ...
)
```

Arguments

x	Numeric vector of observed order statistics (sorted ascending).
r_removals	Integer vector of progressive removal counts (censoring scheme R). Must have the same length as x.

distribution	A gpc_dist distribution object created by make_gpc_dist or a predefined distribution constructor.
start	Optional vector of initial parameter values. If NULL, the default parameters from the distribution object are used.
max_iter	Maximum number of EM iterations (default 1000).
tol	Convergence tolerance (default 1e-8).
...	Additional arguments (currently unused).

Value

The input distribution object with updated parameter estimates and attributes: "em_fit" (the raw emfit object), "converged" (logical), "iterations" (integer), and "loglik" (numeric).

Examples

```
my_dist <- dist_weibull(shape = 1, scale = 1)
x <- c(0.5, 1.2, 2.1, 3.4, 4.8)
r <- c(1, 0, 2, 0, 1)
fitted <- fit_prog_ty2_em(x = x, r_removals = r, distribution = my_dist)
```

---

gpc_index_em *Retrieve a Single GPCI Value*

Description

Extracts or computes a single capability index from a fitted object.

Usage

```
gpc_index_em(fit, index)
```

Arguments

fit	A gpc_em_fit object.
index	Character name of the index to extract or compute.

Value

Numeric value of the specified index.

Examples

```
my_dist <- dist_weibull(shape = 1, scale = 1)
x <- c(0.5, 1.2, 2.1, 3.4, 4.8)
r <- c(1, 0, 2, 0, 1)
fit <- capability_prog_em(x = x, r_removals = r,
  distribution = my_dist, USL = 6, LSL = 0, fit = FALSE)
gpc_index_em(fit, "Cp")
```

make_gpc_dist *Custom and Predefined Distribution Construction*

Description

Builds a distribution specification object for parameter estimation using the EM algorithm and Generalized Process Capability Indices (GPCIs) under progressive Type-II censored data.

Usage

```
make_gpc_dist(
  name,
  pdf,
  cdf,
  surv = NULL,
  quantile = NULL,
  start,
  param_names = paste0("param", seq_along(start)),
  support = c(0, Inf)
)
```

Arguments

name	Character string giving the name of the distribution.
pdf	Function `f(x, theta)` for the probability density function (or probability mass function).
cdf	Function `F(x, theta)` for the cumulative distribution function.
surv	Function `S(x, theta)` for the survival function (defaults to `1 - cdf(x, theta)`).
quantile	Optional function `Q(p, theta)` for the quantile function. If `NULL`, quantiles are computed by numerical inversion of the CDF.
start	Vector of initial parameter values for estimation.
param_names	Character vector of parameter names.
support	Numeric vector of length 2 giving the domain support of the distribution (default `c(0, Inf)`).

Value

An object of class `"gpc_dist"`.

Examples

```
pdf_exp <- function(x, theta) dexp(x, rate = theta[1])
cdf_exp <- function(x, theta) pexp(x, rate = theta[1])
surv_exp <- function(x, theta) 1 - pexp(x, rate = theta[1])
my_dist <- make_gpc_dist("Exponential", pdf_exp, cdf_exp, surv_exp,
  start = c(1), param_names = "rate")
```

predefined_distributions
Predefined Distribution Specifications

Description

Built-in helper functions creating "gpc_dist" objects for standard process distributions.

Usage

```
dist_normal(mean = 0, sd = 1)

dist_weibull(shape = 1, scale = 1)

dist_gamma(shape = 1, rate = 1)

dist_exp_exp(alpha = 1, lambda = 1)

dist_logistic_exp(alpha = 1, lambda = 1)
```

Arguments

mean	Initial mean for Normal distribution.
sd	Initial standard deviation for Normal distribution.
shape	Initial shape parameter.
scale	Initial scale parameter.
rate	Initial rate parameter.
alpha	Initial alpha shape parameter.
lambda	Initial lambda scale parameter.

Value

A "gpc_dist" object.

Examples

```
d_norm <- dist_normal()
d_weib <- dist_weibull()
```

print.gpc_em_ci *Print Method for Bootstrap CIs*

Description

Print Method for Bootstrap CIs

Usage

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

Arguments

x	Object of class gpc_em_ci.
...	Additional arguments (currently unused).

Value

The input object, invisibly.

print.gpc_em_fit *Print Method for Progressive Type-II GPCI EM Fit*

Description

Print Method for Progressive Type-II GPCI EM Fit

Usage

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

Arguments

x	Object of class gpc_em_fit.
...	Additional arguments (currently unused).

Value

The input object, invisibly.

summary.gpc_em_ci *Summary Method for Bootstrap CIs*

Description

Summary Method for Bootstrap CIs

Usage

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

Arguments

object	Object of class gpc_em_ci.
...	Additional arguments (currently unused).

Value

The input object, invisibly.

summary.gpc_em_fit *Summary Method for Progressive Type-II GPCI EM Fit*

Description

Summary Method for Progressive Type-II GPCI EM Fit

Usage

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

Arguments

object	Object of class gpc_em_fit.
...	Additional arguments (currently unused).

Value

The input object, invisibly.

Index