Package 'gpciProgTyII'

August 7, 2026

Type Package

Title Generalized Process Capability Indices under Progressive Type-II Censoring

Version 0.1.0

Description Provides a comprehensive generalized framework for parameter estimation and Generalized Process Capability Indices (GPCIs) under Progressive Type-II censored data using the MleCensoR package. Accepts user-supplied probability density functions (PDF/PMF), cumulative distribution functions (CDF), and survival functions (SF). Computes classical and generalized capability indices including Cpy (Maiti et al., 2010 <doi:10.1080/16843703.2010.11673233>), Spmk (Dey & Saha, 2019 <doi:10.1007/s41872-019-00081-4>), CpTk (Saha et al., 2019), Cpc (Saha et al., 2022 <doi:10.1080/02664763.2021.1971632>), CNpmc (Alotaibi et al., 2022 <doi:10.1155/2022/3135264>), CNpmkc (Saha et al., 2024 <doi:10.1142/S021853932450013X>), CNpk (Saha et al., 2018 <doi:10.1080/21681015.2018.1437793>), and Vannman's Cp(u,v) family. Evaluates parametric and non-parametric bootstrap confidence intervals at 90%, 95%, and 99% levels of significance. Computes Standard Errors, Mean Squared Error (MSE), Bias, and Coverage Probabilities for model parameters and capability indices.
References:
Balakrishnan & Aggarwala (2000) <doi:10.1007/978-1-4612-1186-0>,
Maiti, Saha & Nanda (2010) <doi:10.1080/16843703.2010.11673233>,
Saha, Dey & Maiti (2018) <doi:10.1080/21681015.2018.1437793>,
Dey & Saha (2019) <doi:10.1007/s41872-019-00081-4>,
Saha, Dey & Maiti (2019),
Alotaibi, Dey & Saha (2022) <doi:10.1155/2022/3135264>,
Saha, Dey & Nadarajah (2022) <doi:10.1080/02664763.2021.1971632>,
Saha, Tripathi & Dey (2024) <doi:10.1142/S021853932450013X>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.0.0)

Imports stats, graphics, grDevices, numDeriv, boot, MleCensoR

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

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

Maintainer Shikhar Tyagi <shikhar1093tyagi@gmail.com>

Repository CRAN

Date/Publication 2026-08-07 19:20:02 UTC

Contents

`boot_ci_prog` *Bootstrap Confidence Intervals and Performance Evaluation for Progressive Type-II GPCI*

Description

Computes parametric and non-parametric bootstrap confidence intervals for Generalized Process Capability Indices (GPCIs) and model parameters under progressive Type-II censoring at 90%, 95%, and 99% significance levels.

Usage

```
boot_ci_prog(
  fit,
  B = 1000,
  alpha = c(0.1, 0.05, 0.01),
  method = c("percentile", "normal", "basic", "BCp", "BCa", "studentized"),
  type = c("parametric", "nonparametric")
)
```

```
gpc_boot(
  fit,
  B = 1000,
  alpha = c(0.1, 0.05, 0.01),
  method = c("percentile", "normal", "basic", "BCp", "BCa", "studentized"),
  type = c("parametric", "nonparametric")
)
```

Arguments

`fit`	A `gpc_prog_fit` object returned by `capability_prog`.
`B`	Number of bootstrap replicates (default 1000).
`alpha`	Vector of significance levels (default `c(0.10, 0.05, 0.01)` corresponding to 90%, 95%, and 99% confidence levels).
`method`	Confidence interval method. Options: `"percentile"`, `"normal"`, `"basic"`, `"BCp"`, `"BCa"`, `"studentized"`.
`type`	Resampling type: `"parametric"` (default) or `"nonparametric"`.

Value

An object of class `"gpc_prog_ci"` containing confidence interval tables, standard errors, MSE, and bias.

Examples

```
dist_exp <- 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(x = x, r_removals = r, distribution = dist_exp, USL = 6, LSL = 0)
ci <- boot_ci_prog(fit, B = 50, alpha = c(0.10, 0.05, 0.01), method = "percentile")
print(ci)
```

`capability_prog` *Compute Generalized Process Capability Indices for Progressive Type-II Censored Data*

Description

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

Usage

```
capability_prog(
  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
)

gpc_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
)
```

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	Vector of capability indices to compute. Choices include "Cpy", "Cp", "Cpk", "Cpu", "Cpl", "Cpm", "Cpmk", "Cp_uv", "Spmk", "CpTk", "Cpc", "CNpmc", "CNpmkc", "CNpk", "Cp_q", "Cpk_q".
u	Weight parameter u for Cp(u,v) family (default is 1).
v	Weight parameter v for Cp(u,v) family (default is 1).
mode	Mode of evaluation: "moments" (mean and variance) or "quantile" (quantiles).
fit	Logical. If TRUE (default), estimates distribution parameters using MleCensoR::mle_progressive_type2
C0, C1, C2	Parameters for loss/tolerance cost function in CNpmc and CNpmkc (default: C0 = 1, C1 = 0, C2 = 1).
tolerance_t	Tolerance length t (defaults to USL - LSL).
P0	Desired conformance level for Cpy and Cpc (default is 0.9973002).
LDL, UDL	Lower and Upper Desired Limits for CpTk (default to LSL and USL).

Value

An object of class c("gpc_prog_fit", "gpcifit").

Examples

```
dist_exp <- 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(x = x, r_removals = r, distribution = dist_exp, USL = 6, LSL = 0)
print(cap)
```

---

eval_performance *Monte Carlo Simulation and Coverage Probability Evaluation*

Description

Evaluates standard error, MSE, bias, and empirical coverage probability for parameters and GPCIs under progressive Type-II censoring.

Usage

```
eval_performance(
  distribution,
  r_removals,
  USL,
  LSL,
  target = (USL + LSL)/2,
  true_indices = NULL,
  M = 100,
  B = 200,
  alpha = c(0.1, 0.05, 0.01)
)

gpc_sim(
  distribution,
  r_removals,
  USL,
  LSL,
  target = (USL + LSL)/2,
  true_indices = NULL,
  M = 100,
  B = 200,
  alpha = c(0.1, 0.05, 0.01)
)
```

Arguments

`distribution`	A `gpc_dist` distribution object with true parameters.
`r_removals`	Progressive censoring scheme vector R.
`USL`	Upper Specification Limit.
`LSL`	Lower Specification Limit.
`target`	Process target.
`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`	Vector of significance levels (default `c(0.10, 0.05, 0.01)`).

Value

A list containing performance tables and coverage probabilities.

Examples

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

`fit_prog_ty2` *Parameter Estimation for Progressive Type-II Censored Data using MleCensoR*

Description

Fits process distribution parameters to progressive Type-II censored data using the `MleCensoR` package (`mle_progressive_type2`).

Usage

```
fit_prog_ty2(
  x,
  r_removals,
  distribution,
  start = NULL,
  method = NULL,
  lower = NULL,
  upper = NULL,
  ...
)
```

Arguments

`x`	Numeric vector of observed failure times (must be sorted in ascending order).
`r_removals`	Numeric vector of progressive removal counts (censoring scheme R).
`distribution`	A `gpc_dist` distribution object or function specifications.
`start`	Vector of initial parameter values.
`method`	Optimization method (e.g. `"L-BFGS-B"`, `"BFGS"`, `"NR"`).
`lower`	Optional lower bounds for parameters.
`upper`	Optional upper bounds for parameters.
`...`	Additional arguments passed to `MleCensoR::mle_progressive_type2`.

Value

A fitted `gpc_dist` object with updated parameters, log-likelihood, standard errors, and covariance matrix.

Examples

```
dist_exp <- 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 <- fit_prog_ty2(x = x, r_removals = r, distribution = dist_exp)
```

gpc_index *Dispatcher for Index Value Retrieval*

Description

Dispatcher for Index Value Retrieval

Usage

```
gpc_index(fit, index)
```

Arguments

fit	A gpc_prog_fit object.
index	Name of index to extract or compute.

Value

Numeric value of the specified index.

Examples

```
dist_exp <- 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(x = x, r_removals = r, distribution = dist_exp, USL = 6, LSL = 0)
gpc_index(fit, "Cpy")
```

make_gpc_dist *Custom Distribution Object Builder for Progressive Type-II GPCI*

Description

Builds a distribution specification object for parameter estimation 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.
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.
start	Vector of initial parameter values for estimation.
param_names	Character vector of parameter names.
support	Numeric vector of length 2 giving domain support of the distribution.

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])
dist <- make_gpc_dist("Exponential", pdf_exp, cdf_exp, surv_exp, start = c(1), param_names = "rate")
```

---

plot.gpc_prog_fit *Visualization Functions for Progressive Type-II GPCI*

Description

Provides plot S3 methods for fitted progressive Type-II capability objects and bootstrap confidence interval results.

Usage

```
## S3 method for class 'gpc_prog_fit'
plot(x, ...)

## S3 method for class 'gpc_prog_ci'
plot(x, index = NULL, ...)
```

Arguments

x	Object of class gpc_prog_fit or gpc_prog_ci.
...	Additional plot parameters.
index	Name of index or parameter to plot for gpc_prog_ci.

Value

Invisibly returns the input object.

Examples

```
dist_exp <- 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(x = x, r_removals = r, distribution = dist_exp, USL = 6, LSL = 0)
plot(fit)
```

---

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()
```

Index