Package 'gpciLindApproxProgII'

August 28, 2026

Type Package

Title Lindley Approximation for Capability Indices under Progressive Censoring

Version 0.1.1

Maintainer Shikhar Tyagi <shikhar1093tyagi@gmail.com>

Description Implements Bayesian parameter and Generalized Process Capability Indices (GPCIs) estimation using the Lindley approximation method (Lindley, 1980 <doi:10.2307/2345271>) under progressive Type-II censored data (Balakrishnan & Aggarwala, 2000 <doi:10.1007/978-1-4612-1334-5>). Evaluates point estimates and posterior expectations for classical and non-normal 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 <doi:10.1007/s13198-019-00789-7>), 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 (Vannman, 1995 <doi:10.1111/j.1467-9574.1995.tb01472.x>). Calculates point estimates, bias, mean squared error (MSE), Bayes risk under Linex and squared error loss, Highest Posterior Density (HPD) credible intervals at 90%, 95%, and 99% levels, and Heidelberger and Welch's MCMC convergence diagnostics (Heidelberger & Welch, 1983 <doi:10.1287/opre.31.6.1109>) with convergence probabilities. Accommodates user-defined probability density/mass functions, cumulative distribution functions, and survival functions. Supports progressive parametric and non-parametric bootstrap confidence intervals (Efron, 1987 <doi:10.1080/01621459.1987.10478410>) at 90%, 95%, and 99% significance levels.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.0.0)

Imports stats, graphics, ggplot2, numDeriv, boot, coda

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]

Repository CRAN

Date/Publication 2026-08-28 21:30:19 UTC

Contents

`gpciLindApproxProgII-package`

gpciLindApproxProgII: Lindley Approximation Method for GPCIs under Progressive Type-II Censoring

---

Description

Evaluates Generalized Process Capability Indices (GPCIs) under progressive Type-II censoring using Lindley's 3rd-order approximation method, MCMC chain generation with burn-in and thinning, HPD credible intervals, convergence diagnostics, and progressive bootstrap confidence intervals.

Author(s)

Maintainer: Shikhar Tyagi <shikhar1093tyagi@gmail.com> (ORCID)

Authors:

- Sumit Kumar <stats.sumitbhal@gmail.com>
- Arvind Pandey <arvindmzu@gmail.com>
- Bhupendra Singh <bhupendra.rana@gmail.com>
- Vrijesh Tripathi <vrijesh.tripathi@uwi.edu>

`bootstrap_prog_gpci` *Parametric and Non-Parametric Bootstrap Confidence Intervals under Progressive Type-II Censoring*

Description

Computes bootstrap confidence intervals for Generalized Process Capability Indices (GPCIs) under progressive Type-II censoring using parametric or non-parametric resampling. Supports Percentile, Normal, Basic, and BCp bootstrap methods at 90

Usage

```
bootstrap_prog_gpci(
  x,
  r_removals = NULL,
  distribution = NULL,
  USL,
  LSL,
  target = (USL + LSL)/2,
  indices = NULL,
  type = c("parametric", "nonparametric"),
  B = 500,
  conf_levels = c(0.9, 0.95, 0.99),
  u = 1,
  v = 1
)
```

Arguments

`x`	Numeric vector of observed failure times.
`r_removals`	Progressive censoring removal scheme (vector of integer counts).
`distribution`	A `gpci_dist` object.
`USL`	Upper Specification Limit.
`LSL`	Lower Specification Limit.
`target`	Process target value (default `(USL + LSL) / 2`).

indices	Character vector of GPCI names to evaluate.
type	Type of bootstrap: "parametric" (default) or "nonparametric".
B	Number of bootstrap replications. Default is 500.
conf_levels	Confidence levels for intervals (default c(0.90, 0.95, 0.99)).
u, v	Parameters for Cp(u,v) family (default 1).

Value

An object of S3 class "gpciBootstrapProg" containing a list with:

orig_estimates	Named numeric vector of original point estimates computed at input data.
boot_matrix	A data frame of evaluated GPCI values across all B bootstrap samples.
bootstrap_results	A list for each GPCI containing original estimate, standard error, bias, raw bootstrap draws, and confidence interval tables (Percentile, Normal, Basic, BCp) at 90%, 95%, and 99% confidence levels.
type	Character string indicating bootstrap type ("parametric" or "nonparametric").
B	Integer number of bootstrap replications.

Examples

```
x <- c(0.5, 1.2, 2.1, 3.4, 4.8)
r <- c(1, 0, 2, 0, 1)
fit_boot <- bootstrap_prog_gpci(
  x = x,
  r_removals = r,
  distribution = dist_weibull(),
  USL = 6,
  LSL = 0,
  B = 50
)
```

---

capability_prog *Compute Generalized Process Capability Indices (GPCIs) under Progressive Type-II Censoring*

Description

Computes Generalized Process Capability Indices (GPCIs) for progressive Type-II censored process data, given a gpci_dist object and specification limits.

Usage

```
capability_prog(
  x = NULL,
  r_removals = NULL,
  distribution = NULL,
  USL,
  LSL,
  target = (USL + LSL)/2,
  indices = NULL,
  fit = TRUE,
  u = 1,
  v = 1,
  C0 = 1,
  C1 = 0,
  C2 = 1,
  P0 = 0.9973002
)
```

Arguments

x	Numeric vector of observed failure times. Optional if distribution has parameters set.
r_removals	Progressive censoring removal scheme (vector of integer counts). Optional.
distribution	A gpci_dist object with parameter values.
USL	Upper Specification Limit.
LSL	Lower Specification Limit.
target	Target process value. Default is (USL + LSL) / 2.
indices	Character vector of GPCI names to calculate. Default calculates all available indices.
fit	Logical flag. If TRUE (default), fits the distribution to x first under progressive censoring. If FALSE, uses parameters already present in distribution.
u	Parameter u for Vännman's Cp(u,v) index (default 1).
v	Parameter v for Vännman's Cp(u,v) index (default 1).
C0	Parameter C0 for loss function (default 1).
C1	Parameter C1 for loss function (default 0).
C2	Parameter C2 for loss function (default 1).
P0	Baseline process yield (default 0.9973002).

Value

A named numeric vector of computed Generalized Process Capability Indices (GPCIs) evaluated for the process under progressive Type-II censoring (including Cpy, Cp, Cpk, Cpu, Cpl, Cpm, Cpmk, CpTk, Spmk, Cpc, CNpk, CNpmc, CNpmkc, and Cp_uv).

Examples

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

---

define_distribution *Define a Process Probability Distribution Object*

Description

Creates a gpci_dist object containing probability density function (PDF), cumulative distribution function (CDF), survival function (SF), and quantile function (QF).

Usage

```
define_distribution(
  name = "Custom",
  pdf = NULL,
  cdf = NULL,
  sf = NULL,
  qf = NULL,
  support = c(-Inf, Inf),
  params = list()
)
```

Arguments

name	Character string specifying the name of the distribution.
pdf	Function representing probability density function(x, ...).
cdf	Function representing cumulative distribution function(x, ...).
sf	Function representing survival function function(x, ...).
qf	Optional function representing quantile function function(p, ...).
support	Numeric vector of length 2 giving lower and upper support bounds. Default c(-Inf, Inf).
params	List of initial parameter values or parameter names.

Value

An object of S3 class "gpci_dist" containing:

name	Character string of the distribution name.
pdf	Probability density function function(x, ...).
cdf	Cumulative distribution function function(x, ...).
sf	Survival function function(x, ...).
qf	Quantile function function(p, ...) (if supplied).
support	Numeric vector of lower and upper support bounds.
params	List of default or estimated parameter values.
param_names	Character vector of parameter names.

Examples

```
dist_norm <- define_distribution(
  name = "Normal",
  pdf = function(x, mean = 0, sd = 1) stats::dnorm(x, mean, sd),
  cdf = function(x, mean = 0, sd = 1) stats::pnorm(x, mean, sd),
  sf  = function(x, mean = 0, sd = 1) 1 - stats::pnorm(x, mean, sd),
  params = list(mean = 0, sd = 1)
)
```

---

dist_exponentiated_exponential

Built-in Exponentiated-Exponential Distribution for GPCI

Description

Built-in Exponentiated-Exponential Distribution for GPCI

Usage

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

Arguments

alpha	Initial alpha parameter (default 1).
lambda	Initial lambda parameter (default 1).

Value

An object of S3 class "gpci_dist" representing the Exponentiated-Exponential distribution with PDF, CDF, SF, QF, support bounds, and parameter values.

dist_gamma *Built-in Gamma Distribution for GPCI*

Description

Built-in Gamma Distribution for GPCI

Usage

```
dist_gamma(shape = 1, rate = 1)
```

Arguments

shape	Initial shape parameter (default 1).
rate	Initial rate parameter (default 1).

Value

An object of S3 class "gpci_dist" representing the Gamma distribution with PDF, CDF, SF, QF, support bounds, and parameter values.

dist_logistic_exponential *Built-in Logistic-Exponential Distribution for GPCI*

Description

Built-in Logistic-Exponential Distribution for GPCI

Usage

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

Arguments

alpha	Initial alpha parameter (default 1).
lambda	Initial lambda parameter (default 1).

Value

An object of S3 class "gpci_dist" representing the Logistic-Exponential distribution with PDF, CDF, SF, QF, support bounds, and parameter values.

dist_normal *Built-in Normal Distribution for GPCI*

Description

Built-in Normal Distribution for GPCI

Usage

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

Arguments

mean	Initial mean parameter (default 0).
sd	Initial sd parameter (default 1).

Value

An object of S3 class "gpci_dist" representing the Normal distribution with PDF, CDF, SF, QF, support bounds, and parameter values.

dist_weibull *Built-in Weibull Distribution for GPCI*

Description

Built-in Weibull Distribution for GPCI

Usage

```
dist_weibull(shape = 1, scale = 1)
```

Arguments

shape	Initial shape parameter (default 1).
scale	Initial scale parameter (default 1).

Value

An object of S3 class "gpci_dist" representing the Weibull distribution with PDF, CDF, SF, QF, support bounds, and parameter values.

fit_distribution_prog *Fit Probability Distribution under Progressive Type-II Censoring*

Description

Fits a gpci_dist object to progressive Type-II censored process data using Maximum Likelihood Estimation (MLE).

Usage

```
fit_distribution_prog(x, r_removals, distribution, init = NULL)
```

Arguments

x	Numeric vector of observed failure times.
r_removals	Vector of progressive removal counts corresponding to x.
distribution	A gpci_dist object.
init	Optional initial parameter values list.

Value

An updated object of S3 class "gpci_dist" containing Maximum Likelihood Estimates (MLE) of parameters under progressive Type-II censoring in $params and raw optimization results in $opt_result.

Examples

```
x <- c(0.5, 1.2, 2.1, 3.4, 4.8)
r <- c(1, 0, 2, 0, 1)
fit <- fit_distribution_prog(x, r, dist_weibull())
```

gpci_lindley_prog *High-Level User Interface Function for Progressive Lindley Approximation GPCI Analysis*

Description

Main user-facing function allowing input of custom PDF, CDF, and Survival functions, prior distribution hyperparameters, chain length, burn-in, and thinning.

Usage

```
gpci_lindley_prog(
  x,
  r_removals = NULL,
  pdf = NULL,
  cdf = NULL,
  sf = NULL,
  prior = NULL,
  chain_length = 1000,
  burn_in = 200,
  thinning = 1,
  USL,
  LSL,
  target = (USL + LSL)/2,
  indices = NULL,
  B = 500
)
```

Arguments

x	Numeric vector of observed failure times.
r_removals	Progressive censoring removal scheme (vector of integer counts).
pdf	Custom PDF function function(x, ...).
cdf	Custom CDF function function(x, ...).
sf	Custom Survival Function function(x, ...).
prior	Prior distribution hyperparameters list or log-prior function.
chain_length	Desired MCMC chain length (default 1000).
burn_in	MCMC burn-in iterations (default 200).
thinning	Thinning interval (default 1).
USL	Upper Specification Limit.
LSL	Lower Specification Limit.
target	Process target value.
indices	Character vector of GPCI names to evaluate.
B	Number of bootstrap replications (default 500).

Value

An object of S3 class "gpciLindApproxProgII" containing MCMC parameter chains, GPCI chains, MLE point estimates, Lindley Bayes point estimates, progressive bootstrap confidence intervals, and analysis specifications.

Examples

```
gpci_lindley_prog(
  x = c(0.5, 1.2, 2.1, 3.4, 4.8),
  r_removals = c(1, 0, 2, 0, 1),
  pdf = function(x, mean = 0, sd = 1) stats::dnorm(x, mean, sd),
  cdf = function(x, mean = 0, sd = 1) stats::pnorm(x, mean, sd),
  chain_length = 100,
  burn_in = 20,
  USL = 6,
  LSL = 0,
  B = 20
)
```

---

heidelberger_welch *Heidelberger and Welch's MCMC Convergence Diagnostic*

Description

Conducts stationarity and relative half-width tests under Heidelberger and Welch's MCMC convergence diagnostic, returning test statistics, status, and convergence probability.

Usage

```
heidelberger_welch(x, alpha = 0.05, eps = 0.1)
```

Arguments

x	Numeric vector representing an MCMC / simulation chain.
alpha	Significance level for the test (default is 0.05).
eps	Target maximum ratio of half-width to sample mean (default is 0.1).

Value

A list containing Heidelberger and Welch convergence diagnostic results:

stat	Cramér-von Mises test statistic for stationarity.
pvalue	Stationarity test p-value.
passed	Logical flag indicating whether stationarity test passed.
hw_stat	Half-width test statistic (ratio of half-width to sample mean).
hw_passed	Logical flag indicating whether half-width test passed.
convergence_prob	Estimated convergence probability.

Examples

```
samples <- stats::rnorm(1000)
heidelberger_welch(samples)
```

hpd_interval *Highest Posterior Density (HPD) Interval Calculation*

Description

Computes the Highest Posterior Density (HPD) interval for sample draws at specified confidence / significance levels (e.g., 90

Usage

```
hpd_interval(x, prob = 0.95)
```

Arguments

x	Numeric vector of sample draws.
prob	Credibility level (1 - significance level). Default is 0.95.

Value

A named numeric vector of length 2 with elements "lower" and "upper" specifying the Highest Posterior Density (HPD) interval bounds at the requested credibility level.

Examples

```
samples <- stats::rnorm(1000, mean = 5, sd = 1)
hpd_interval(samples, prob = 0.95)
```

lindley_approx_prog *Lindley Approximation Method for Parameter and GPCI Bayesian Estimation under Progressive Type-II Censoring*

Description

Computes Bayesian posterior expectations of model parameters and Generalized Process Capability Indices (GPCIs) using Lindley's 3rd-order Taylor series approximation for progressive Type-II censored data.

Usage

```
lindley_approx_prog(
  x,
  r_removals = NULL,
  distribution,
  prior = NULL,
  USL,
  LSL,
```

```
  target = (USL + LSL)/2,
  indices = NULL,
  u = 1,
  v = 1,
  C0 = 1,
  C1 = 0,
  C2 = 1,
  P0 = 0.9973002
)
```

Arguments

x	Numeric vector of observed failure times.
r_removals	Progressive censoring removal scheme (vector of integer counts).
distribution	A gpci_dist object.
prior	Log-prior density function function(params) or a hyperparameter list. Default assumes non-informative flat or conjugate log-priors.
USL	Upper Specification Limit.
LSL	Lower Specification Limit.
target	Process target value. Default is (USL + LSL) / 2.
indices	Character vector of GPCI names to evaluate.
u	Parameter u for Vännman's Cp(u,v) index (default 1).
v	Parameter v for Vännman's Cp(u,v) index (default 1).
C0	Parameter C0 for tolerance loss function (default 1).
C1	Parameter C1 for tolerance loss function (default 0).
C2	Parameter C2 for tolerance loss function (default 1).
P0	Baseline process yield (default 0.9973002).

Value

An object of S3 class "gpciLindleyProg" containing a list with the following components:

mle_params	A named list of Maximum Likelihood Estimates (MLE) of parameters under progressive censoring.
lindley_params	A named list of Lindley Bayes point estimates of parameters.
mle_gpci	A named numeric vector of GPCI point estimates computed at parameter MLEs.
lindley_gpci	A named numeric vector of Lindley Bayes point estimates of evaluated GPCIs.
cov_matrix	Estimated parameter variance-covariance matrix derived from the observed Hessian.
hessian	Observed Hessian matrix of the negative log-likelihood.
specs	A list containing input data x, removal scheme r_removals, distribution object, and specification limits.

Examples

```
x <- c(0.5, 1.2, 2.1, 3.4, 4.8)
r <- c(1, 0, 2, 0, 1)
dist_exp <- dist_weibull(shape = 1, scale = 2)
fit <- lindley_approx_prog(
  x = x, r_removals = r, distribution = dist_exp,
  USL = 6, LSL = 0
)
```

`lindley_prog_gpci` *Lindley Approximation MCMC-Style Chain Generator for GPCIs under Progressive Type-II Censoring*

Description

Generates posterior parameter draws, GPCI chains, point estimates, HPD intervals, bootstrap confidence intervals, and convergence diagnostics for progressive Type-II censored data using the Lindley approximation method combined with sampling and thinning.

Usage

```
lindley_prog_gpci(
  x,
  r_removals = NULL,
  distribution = NULL,
  prior = NULL,
  chain_length = 1000,
  burn_in = 200,
  thinning = 1,
  USL,
  LSL,
  target = (USL + LSL)/2,
  indices = NULL,
  u = 1,
  v = 1,
  C0 = 1,
  C1 = 0,
  C2 = 1,
  P0 = 0.9973002,
  B = 500
)
```

Arguments

`x`	Numeric vector of observed failure times.
`r_removals`	Progressive censoring removal scheme (vector of integer counts).
`distribution`	A `gpci_dist` object.

prior	Log-prior density function function(params) or hyperparameter list.
chain_length	Desired length of final retained chain after burn-in and thinning. Default is 1000.
burn_in	Number of initial burn-in samples to discard. Default is 200.
thinning	Thinning interval. Default is 1 (no thinning).
USL	Upper Specification Limit.
LSL	Lower Specification Limit.
target	Process target value (defaults to midpoint of USL and LSL).
indices	Character vector of GPCI names to evaluate. Default evaluates all available indices.
u	Parameter u for Vännman's Cp(u,v) family (default 1).
v	Parameter v for Vännman's Cp(u,v) family (default 1).
C0	Parameter C0 for loss function (default 1).
C1	Parameter C1 for loss function (default 0).
C2	Parameter C2 for loss function (default 1).
P0	Baseline process yield (default 0.9973002).
B	Number of bootstrap replications for bootstrap CIs. Default is 500.

Value

An object of S3 class "gpciLindApproxProgII" containing a list with the following components:

param_chain	A matrix of retained posterior parameter draws after applying burn-in and thinning.
gpci_chain	A data frame of evaluated Generalized Process Capability Index (GPCI) values computed across the retained parameter chain.
point_estimates	A named numeric vector of Maximum Likelihood Estimates (MLE) of GPCIs.
lindley_estimates	A named numeric vector of Lindley Bayes point estimates of GPCIs.
lindley_fit	The underlying "gpciLindleyProg" object returned by lindley_approx_prog.
bootstrap_results	An object of class "gpciBootstrapProg" containing progressive bootstrap confidence interval estimates.
specs	A list containing analysis specifications including specification limits (USL, LSL, target), dataset x, removal scheme r_removals, distribution object, chain length, burn-in, and thinning settings.

Examples

```
x <- c(0.5, 1.2, 2.1, 3.4, 4.8)
r <- c(1, 0, 2, 0, 1)
fit_chain <- lindley_prog_gpci(
  x = x,
  r_removals = r,
```

```
  distribution = dist_weibull(),
  chain_length = 100,
  burn_in = 20,
  thinning = 1,
  USL = 6,
  LSL = 0,
  B = 20
)
```

`plot.gpciLindApproxProgII` *Plot Method for gpciLindApproxProgII Objects*

Description

Plots posterior probability density curves and MCMC-style trace plots for evaluated Generalized Process Capability Indices (GPCIs).

Usage

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

Arguments

`x`	An object of class `"gpciLindApproxProgII"`.
`...`	Additional arguments passed to base plotting functions.

Value

Invisible `NULL`, called for the side effect of displaying trace plots and posterior probability density curves for evaluated Generalized Process Capability Indices (GPCIs).

Examples

```
x_data <- c(0.5, 1.2, 2.1, 3.4, 4.8)
r_data <- c(1, 0, 2, 0, 1)
fit <- lindley_prog_gpci(
  x = x_data,
  r_removals = r_data,
  distribution = dist_weibull(),
  USL = 6,
  LSL = 0,
  chain_length = 100,
  burn_in = 20,
  B = 20
)
plot(fit)
```

print.gpciLindApproxProgII

Print Method for gpciLindApproxProgII Objects

Description

Print Method for gpciLindApproxProgII Objects

Usage

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

Arguments

x	An object of class "gpciLindApproxProgII".
...	Additional arguments.

Value

Invisible x (an object of class "gpciLindApproxProgII"), called for the side effect of printing a summary of the progressive Lindley approximation analysis, including MCMC specifications, specification limits, and point estimates (MLE and Lindley Bayes).

summary.gpciLindApproxProgII

Summary and Diagnostics for gpciLindApproxProgII Objects

Description

Computes point estimates, Lindley Bayes estimates, Bias, MSE, Risk values (Linex + SEL), HPD intervals at 90 convergence diagnostic, convergence probability, and Bootstrap confidence intervals under progressive Type-II censoring.

Usage

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

Arguments

object	An object of class "gpciLindApproxProgII".
...	Additional arguments.

Value

A `data.frame` containing detailed diagnostic metrics for each evaluated Generalized Process Capability Index (GPCI), including MLE estimates, Lindley Bayes estimates, MCMC chain means, bias, mean squared error (MSE), risk values (SEL + Linex), 90%, 95%, and 99% HPD credible intervals, Heidelberger-Welch stationarity test statistics and p-values, convergence probabilities, and 95% bootstrap percentile confidence intervals.

Examples

```
x <- c(0.5, 1.2, 2.1, 3.4, 4.8)
r <- c(1, 0, 2, 0, 1)
fit_chain <- lindley_prog_gpci(
  x = x, r_removals = r, distribution = dist_weibull(),
  USL = 6, LSL = 0, chain_length = 300, burn_in = 50, B = 100
)
summary(fit_chain)
```

Index