

Package: gpciProgTyIIImpSam (via r-universe)

October 7, 2026

Type Package

Title Generalized Process Capability Indices for Progressive Type-II Censored Data using Importance Sampling

Version 0.1.0

Maintainer Shikhar Tyagi <shikhar1093tyagi@gmail.com>

Description Implements Importance Sampling (Sampling Importance Resampling, SIR) for Bayesian parameter estimation and Generalized Process Capability Indices (GPCIs) under progressive Type-II censored data. Evaluates classical and generalized capability indices including Cpy, Cp, Cpk, Cpu, Cpl, Cpm, Cpmk, Spmk, CpTk, Cpc, CNp, CNpk, CNpm, CNpmk, CNpmc, CNpmkc, and Vannman's Cp(u,v) family. Computes initial uncensored estimates, parameter MCMC chains, GPCI posterior chains, point estimates, posterior means, bias, mean squared error (MSE), Bayes risk under loss functions, Highest Posterior Density (HPD) credible intervals at 90%, 95%, and 99% levels, Heidelberger and Welch's MCMC convergence diagnostics, and convergence probabilities. Accommodates user-defined probability density/mass functions, cumulative distribution functions, and survival functions. Methods based on Balakrishnan and Aggarwala (2000)
<doi:10.1007/978-1-4612-1186-0>, Maiti et al. (2010)
<doi:10.1080/16843703.2010.11673233>, Dey and Saha (2019)
<doi:10.1007/s41872-019-00081-4>, Alotaibi et al. (2022)
<doi:10.1155/2022/3135264>, Saha et al. (2022)
<doi:10.1080/02664763.2021.1971632>, and Saha et al. (2024)
<doi:10.1142/S021853932450013X>.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.3.3

Imports stats, graphics

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr
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 <https://shikhartyagi.r-universe.dev>
Date/Publication 2026-08-07 21:00:02 UTC
RemoteUrl <https://github.com/cran/gpciProgTyIIImpSam>
RemoteRef HEAD
RemoteSha 074aef91a840fb8a3067a2da5663144e2d9835ac

Contents

capability	2
define_distribution	4
dist_exponential	5
dist_gamma	5
dist_logistic_exp	6
dist_normal	6
dist_weibull	7
gpci_prog_ty2impsam	7
heidelberger_welch	9
hpd_interval	10
impsam_prog_ty2	10
plot.gpciProgTyIIImpSam	12
print.gpciProgTyIIImpSam	13
summary.gpciProgTyIIImpSam	13
Index	15

capability	<i>Compute Process Capability Indices (GPCIs)</i>
------------	---

Description

Evaluates classical and generalized Process Capability Indices for specified process parameters and specification limits.

Usage

```
capability(
  distribution,
  USL,
  LSL,
  target = (USL + LSL)/2,
  indices = NULL,
  u = 1,
  v = 1,
  mode = c("moments", "quantile"),
  C0 = 1,
  C1 = 0,
  C2 = 1,
  tolerance_t = USL - LSL,
  P0 = 0.9973002
)
```

Arguments

distribution	A gpci_dist object containing model parameters.
USL	Upper Specification Limit.
LSL	Lower Specification Limit.
target	Target process value (defaults to midpoint $(USL + LSL) / 2$).
indices	Character vector of GPCI names to evaluate. Default computes all supported indices.
u	Parameter u for Vännman's $C_p(u, v)$ family (default 1).
v	Parameter v for Vännman's $C_p(u, v)$ family (default 1).
mode	Mode of computation: "moments" (default) or "quantile".
C0, C1, C2	Parameters for tolerance loss function $C_M(t) = C_0 + C_1 * \exp(-C_2 * t)$ (defaults: 1, 0, 1).
tolerance_t	Process tolerance span (defaults to $USL - LSL$).
P0	Baseline expected yield (default 0.9973002).

Value

A named numeric vector of computed GPCI values.

Examples

```
dist_norm <- dist_normal(mean = 10, sd = 1.5)
capability(
  distribution = dist_norm,
  USL = 15, LSL = 5, target = 10
)
```

define_distribution *Define a Custom Distribution Object*

Description

Creates a `gpci_dist` distribution object containing probability density/mass function (PDF/PMF), cumulative distribution function (CDF), survival function (SF), quantile function, parameter names, and support boundaries.

Usage

```
define_distribution(
  name = "Custom",
  pdf = NULL,
  cdf = NULL,
  sf = NULL,
  quantile = NULL,
  param_names = NULL,
  default_params = list(),
  support = c(-Inf, Inf)
)
```

Arguments

<code>name</code>	Character string naming the distribution.
<code>pdf</code>	Function <code>function(x, ...)</code> computing density.
<code>cdf</code>	Function <code>function(q, ...)</code> computing CDF.
<code>sf</code>	Optional function <code>function(q, ...)</code> computing survival function $1 - \text{CDF}$.
<code>quantile</code>	Optional function <code>function(p, ...)</code> computing quantiles.
<code>param_names</code>	Character vector of parameter names. If <code>NULL</code> , auto-detected from <code>pdf</code> .
<code>default_params</code>	Named list of default parameter values.
<code>support</code>	Numeric vector of length 2 specifying distribution support (e.g. <code>c(0, Inf)</code>).

Value

An object of class `"gpci_dist"`.

Examples

```
dist_custom <- define_distribution(
  name = "ExpCustom",
  pdf = function(x, rate = 1) dexp(x, rate = rate),
  cdf = function(q, rate = 1) pexp(q, rate = rate),
  sf = function(q, rate = 1) pexp(q, rate = rate, lower.tail = FALSE),
  param_names = "rate",
  default_params = list(rate = 1),
)
```

```
    support = c(0, Inf)
  )
```

dist_exponential	<i>Exponential Distribution Object Constructor</i>
------------------	--

Description

Exponential Distribution Object Constructor

Usage

```
dist_exponential(rate = 1)
```

Arguments

rate	Rate parameter (default 1).
------	-----------------------------

Value

A gpci_dist object.

dist_gamma	<i>Gamma Distribution Object Constructor</i>
------------	--

Description

Gamma Distribution Object Constructor

Usage

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

Arguments

shape	Shape parameter (default 1).
rate	Rate parameter (default 1).

Value

A gpci_dist object.

dist_logistic_exp	<i>Logistic-Exponential Distribution Object Constructor</i>
-------------------	---

Description

Logistic-Exponential Distribution Object Constructor

Usage

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

Arguments

alpha	Shape parameter (default 1).
lambda	Scale parameter (default 1).

Value

A gpci_dist object.

dist_normal	<i>Normal Distribution Object Constructor</i>
-------------	---

Description

Normal Distribution Object Constructor

Usage

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

Arguments

mean	Mean parameter (default 0).
sd	Standard deviation parameter (default 1).

Value

A gpci_dist object.

dist_weibull	<i>Weibull Distribution Object Constructor</i>
--------------	--

Description

Weibull Distribution Object Constructor

Usage

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

Arguments

shape	Shape parameter (default 1).
scale	Scale parameter (default 1).

Value

A gpci_dist object.

gpci_prog_ty2impsam	<i>Importance Sampling Estimation of GPCIs under Progressive Type-II Censoring</i>
---------------------	--

Description

Main high-level user interface function to compute Generalized Process Capability Indices (GPCIs) under Progressive Type-II Censored Data using Importance Sampling (Sampling Importance Resampling, SIR). Fits parameter estimates under uncensored and progressive censoring, generates parameter and GPCI chains, and evaluates point estimates, posterior means, bias, MSE, Linex/SEL Bayes risk, 90 percent, 95 percent, and 99 percent HPD credible intervals, Heidelberger and Welch MCMC convergence diagnostics, and convergence probabilities.

Usage

```
gpci_prog_ty2impsam(
  x,
  r_removals = NULL,
  distribution = NULL,
  pdf = NULL,
  cdf = NULL,
  surv = NULL,
  quantile = NULL,
  param_names = NULL,
  start = 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
)

```

Arguments

x	Numeric vector of observed process failure times.
r_removals	Progressive removal scheme (vector of integer counts). Default 0s if uncensored.
distribution	Optional gpci_dist object. If NULL, user must supply pdf and cdf.
pdf	Custom PDF/PMF function function(x, ...).
cdf	Custom CDF function function(q, ...).
surv	Custom Survival function function(q, ...).
quantile	Optional custom Quantile function function(p, ...).
param_names	Character vector of parameter names.
start	Named vector or list of initial parameter values.
prior	Prior distribution hyperparameters list or log-prior function.
chain_length	Desired MCMC chain length after burn-in and thinning. Default 1000.
burn_in	Number of burn-in iterations. Default 200.
thinning	Thinning interval. Default 1.
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 computes all supported indices.
u, v	Vännman's parameters for $C_p(u, v)$ family (default 1).
C0, C1, C2	Parameters for tolerance loss function (defaults: 1, 0, 1).
P0	Expected yield baseline (default 0.9973002).

Value

An object of class "gpciProgTyIIImpSam" containing fit details and diagnostic metrics.

Examples

```
# Example using custom functions for Exponential distribution
my_pdf <- function(x, rate = 1) dexp(x, rate = rate)
my_cdf <- function(q, rate = 1) pexp(q, rate = rate)
my_surv <- function(q, rate = 1) pexp(q, rate = rate, lower.tail = FALSE)
data_x <- c(0.8, 1.5, 2.3, 3.1, 4.2)
removals <- c(1, 0, 1, 0, 1)
fit <- gpci_prog_ty2impsam(
  x = data_x, r_removals = removals,
  pdf = my_pdf, cdf = my_cdf, surv = my_surv,
  start = c(rate = 0.5),
  chain_length = 500, burn_in = 100, thinning = 1,
  USL = 8, LSL = 0
)
print(fit)
```

heidelberg_welch	<i>Heidelberg and Welch's MCMC Convergence Diagnostic</i>
------------------	---

Description

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

Usage

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

Arguments

x	Numeric vector representing an MCMC 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 Cramér-von Mises statistic (stat), p-value (pvalue), stationarity test status (passed), half-width test status (hw_passed), half-width statistic (hw_stat), and convergence probability (convergence_prob).

Examples

```
samples <- rnorm(1000)
heidelberg_welch(samples)
```

hpd_interval	<i>Highest Posterior Density (HPD) Interval Calculation</i>
--------------	---

Description

Computes Highest Posterior Density (HPD) credible interval for posterior draws at specified confidence / significance levels (e.g., 90 percent, 95 percent, 99 percent).

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 containing lower and upper bounds.

Examples

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

impsam_prog_ty2	<i>Importance Sampling for Progressive Type-II Censored Data</i>
-----------------	--

Description

Generates posterior parameter draws and Generalized Process Capability Index (GPCI) chains using Importance Sampling (Sampling Importance Resampling, SIR) for progressive Type-II censored data.

Usage

```
impsam_prog_ty2(
  x,
  r_removals = NULL,
  distribution,
  start = 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
  )

```

Arguments

<code>x</code>	Numeric vector of observed failure times.
<code>r_removals</code>	Progressive removal scheme (vector of integer counts). Default 0s if uncensored.
<code>distribution</code>	A <code>gpci_dist</code> object.
<code>start</code>	Named vector or list of initial parameter values.
<code>prior</code>	Function or list of prior specifications/hyperparameters.
<code>chain_length</code>	Desired length of final retained MCMC chain after burn-in and thinning. Default 1000.
<code>burn_in</code>	Number of initial burn-in samples to discard. Default 200.
<code>thinning</code>	Thinning interval for sampling. Default 1.
<code>USL</code>	Upper Specification Limit.
<code>LSL</code>	Lower Specification Limit.
<code>target</code>	Process target value (defaults to midpoint of USL and LSL).
<code>indices</code>	Character vector of GPCI names to evaluate. Default computes all supported indices.
<code>u, v</code>	Vännman's parameters for $C_p(u, v)$ family (default 1).
<code>C0, C1, C2</code>	Parameters for tolerance loss function (defaults: 1, 0, 1).
<code>P0</code>	Expected yield baseline (default 0.9973002).

Value

A list of class "gpciProgTyIIImpSam" containing parameter chain, GPCI chain, and point estimates.

Examples

```

dist_exp <- dist_exponential(rate = 0.5)
x <- c(0.5, 1.2, 2.1, 3.4, 4.8)
r <- c(1, 0, 2, 0, 1)
fit <- impsam_prog_ty2(
  x = x,
  r_removals = r,

```

```

distribution = dist_exp,
chain_length = 500,
burn_in = 100,
thinning = 1,
USL = 8,
LSL = 0
)

```

```
plot.gpciProgTyIIImpSam
```

Plot Method for gpciProgTyIIImpSam Objects

Description

Visualizes diagnostic traceplots, posterior probability density functions, and Highest Posterior Density (HPD) credibility intervals for evaluated Generalized Process Capability Indices (GPCIs).

Usage

```

## S3 method for class 'gpciProgTyIIImpSam'
plot(x, type = c("density", "trace", "hpd"), indices = NULL, ...)

```

Arguments

x	An object of class "gpciProgTyIIImpSam".
type	Character string indicating plot type: "trace" for MCMC traceplots, "density" for posterior density estimates, or "hpd" for HPD intervals.
indices	Optional character vector of index names to plot. Defaults to first 4 evaluated GPCIs.
...	Additional graphical parameters.

Value

No return value, called for side effects.

Examples

```

dist_exp <- dist_exponential(rate = 0.5)
x <- c(0.5, 1.2, 2.1, 3.4, 4.8)
r <- c(1, 0, 2, 0, 1)
fit <- impsam_prog_ty2(x = x, r_removals = r, distribution = dist_exp, USL = 8, LSL = 0)
plot(fit, type = "density")

```

```
print.gpciProgTyIIImpSam
```

Print Method for gpciProgTyIIImpSam Objects

Description

Print Method for gpciProgTyIIImpSam Objects

Usage

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

Arguments

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

Value

Invisibly returns x.

```
summary.gpciProgTyIIImpSam
```

Summary and Diagnostics for gpciProgTyIIImpSam Objects

Description

Computes estimated value, bias, MSE, Risk value, HPD intervals at 90 percent, 95 percent, and 99 percent significance levels, Heidelberger and Welch MCMC convergence diagnostics, and convergence probabilities for each evaluated GPCI.

Usage

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

Arguments

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

Value

A data frame containing diagnostic metrics for each evaluated GPCI.

Examples

```
dist_exp <- dist_exponential(rate = 0.5)
x <- c(0.5, 1.2, 2.1, 3.4, 4.8)
r <- c(1, 0, 2, 0, 1)
fit <- impsam_prog_ty2(x = x, r_removals = r, distribution = dist_exp, USL = 8, LSL = 0)
summary(fit)
```

Index

capability, [2](#)

define_distribution, [4](#)

dist_exponential, [5](#)

dist_gamma, [5](#)

dist_logistic_exp, [6](#)

dist_normal, [6](#)

dist_weibull, [7](#)

gpci_prog_ty2impsam, [7](#)

heidelberger_welch, [9](#)

hpd_interval, [10](#)

impsam_prog_ty2, [10](#)

plot.gpciProgTyIIImpSam, [12](#)

print.gpciProgTyIIImpSam, [13](#)

summary.gpciProgTyIIImpSam, [13](#)