

Package ‘gpciImpSam’

August 7, 2026

Type Package

Title Importance Sampling Estimation of Generalized Process Capability Indices

Version 0.1.0

Description Provides a comprehensive generalized framework for parameter estimation and Generalized Process Capability Indices (GPCIs) under uncensored data using Importance Sampling (ImpSam). Supports 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](https://doi.org/10.1080/16843703.2010.11673233)>), Spmk (Dey & Saha, 2019 <[doi:10.1080/00949655.2019.1671980](https://doi.org/10.1080/00949655.2019.1671980)>), CpTk (Saha et al., 2019), Cpc (Saha et al., 2022 <[doi:10.1080/02664763.2021.1971632](https://doi.org/10.1080/02664763.2021.1971632)>), CNpmc (Alotaibi et al., 2022 <[doi:10.1155/2022/3135264](https://doi.org/10.1155/2022/3135264)>), CNpmkc (Saha et al., 2024 <[doi:10.1142/S021853932450013X](https://doi.org/10.1142/S021853932450013X)>), CNpk (Saha et al., 2018 <[doi:10.1080/21681015.2018.1437793](https://doi.org/10.1080/21681015.2018.1437793)>), and Vannman's Cp(u,v) family. Generates parameter and GPCI MCMC chains via Sampling Importance Resampling (SIR) after burn-in and thinning. Provides point estimates, bias, MSE, risk values, Highest Posterior Density (HPD) intervals at 90, 95, and 99 percent levels of significance, Heidelberger and Welch MCMC convergence diagnostic, and convergence probability.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.0.0)

Imports stats, graphics, grDevices, numDeriv, boot

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>>)

Maintainer Shikhar Tyagi <shikhar1093tyagi@gmail.com>

Repository CRAN

Date/Publication 2026-08-07 16:30:07 UTC

Contents

capability	2
compute_theoretical_moments	4
define_distribution	4
dist_exponential	5
dist_gamma	6
dist_logistic_exponential	6
dist_normal	7
dist_weibull	7
fit_distribution	8
gpciimpsam	8
heidelberger_welch	10
hpd_interval	10
impsam_gpci	11
plot.gpciImpSam	12
summary.gpciImpSam	13
Index	14

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

Description

Computes classical and generalized Process Capability Indices for a given dataset or process distribution parameters under uncensored data.

Usage

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

```

    UDL = USL
  )

```

Arguments

<code>data</code>	Numeric vector of observed process data (optional if distribution is supplied with parameters).
<code>distribution</code>	A <code>gpci_dist</code> object.
<code>USL</code>	Upper Specification Limit.
<code>LSL</code>	Lower Specification Limit.
<code>target</code>	Target value of process (defaults to midpoint of USL and LSL).
<code>indices</code>	Character vector of index names to evaluate. Default computes all GPCIs: <code>c("Cpy", "Cp", "Cpk", "Cpu", "Cpl", "Cpm", "Cpmk", "CpTk", "Spmk", "Cpc", "CNpk", "CNpmc", "CNpmkc", "Cp_uv")</code> .
<code>u</code>	Parameter u for Vännman's $C_p(u, v)$ family (default 1).
<code>v</code>	Parameter v for Vännman's $C_p(u, v)$ family (default 1).
<code>mode</code>	Mode of computation: "moments" (mean/sd) or "quantile" (quantiles).
<code>fit</code>	Logical. If TRUE (default) and data is supplied, parameters are estimated from data.
<code>fit_method</code>	Parameter estimation method ("mle").
<code>C0, C1, C2</code>	Parameters for tolerance loss function (defaults: 1, 0, 1).
<code>tolerance_t</code>	Process tolerance span (defaults to $USL - LSL$).
<code>P0</code>	Expected yield baseline (default 0.9973002).
<code>LDL, UDL</code>	Lower and upper desired limits.

Value

A named numeric vector of evaluated GPCI values.

Examples

```

dist_n <- dist_normal(mean = 10, sd = 1.5)
capability(
  distribution = dist_n,
  USL = 15, LSL = 5, target = 10,
  indices = c("Cpy", "Cp", "Cpk", "Cpm", "Cpmk")
)

```

```
compute_theoretical_moments
```

Theoretical Moments Computation for Process Distribution

Description

Computes numerical or analytical mean and variance for a `gpci_dist` object.

Usage

```
compute_theoretical_moments(distribution)
```

Arguments

`distribution` A `gpci_dist` distribution object.

Value

A list with mean and var.

Examples

```
compute_theoretical_moments(dist_normal(mean = 5, sd = 2))
```

```
define_distribution    Define a Process Distribution
```

Description

Constructor to define a probability distribution for process capability analysis. The distribution can be defined by specifying any one (or more) of its PDF/PMF, CDF, Survival Function (SF), or quantile function. Missing functions are numerically derived.

Usage

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

Arguments

name	Character string naming the distribution.
pdf	Function representing the probability density function (PDF/PMF) function(x, ...).
cdf	Function representing the cumulative distribution function (CDF) function(x, ...).
sf	Function representing the survival function (SF = 1 - CDF) function(x, ...).
quantile	Function representing the quantile function function(p, ...).
params	Named list of parameters for the distribution.
support	Vector of length 2 defining the lower and upper bounds of support (default c(-Inf, Inf)).

Value

An object of class gpci_dist.

Examples

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

dist_exponential	<i>Standard Pre-defined Exponential Distribution</i>
------------------	--

Description

Standard Pre-defined Exponential Distribution

Usage

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

Arguments

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

Value

A gpci_dist object.

Examples

```
d_exp <- dist_exponential(rate = 0.5)
```

dist_gamma	<i>Standard Pre-defined Gamma Distribution</i>
------------	--

Description

Standard Pre-defined Gamma Distribution

Usage

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

Arguments

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

Value

A gpci_dist object.

Examples

```
d_gam <- dist_gamma(shape = 2, rate = 1)
```

dist_logistic_exponential	<i>Pre-defined Logistic-Exponential Distribution</i>
---------------------------	--

Description

Pre-defined Logistic-Exponential Distribution

Usage

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

Arguments

alpha	Scale/shape parameter (default is 1).
lambda	Scale/shape parameter (default is 1).

Value

A gpci_dist object.

Examples

```
d_le <- dist_logistic_exponential(alpha = 1, lambda = 1)
```

dist_normal	<i>Standard Pre-defined Normal Distribution</i>
-------------	---

Description

Standard Pre-defined Normal Distribution

Usage

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

Arguments

mean	Distribution mean (default is 0).
sd	Distribution standard deviation (default is 1).

Value

A gpci_dist object.

Examples

```
d_norm <- dist_normal(mean = 10, sd = 2)
```

dist_weibull	<i>Standard Pre-defined Weibull Distribution</i>
--------------	--

Description

Standard Pre-defined Weibull Distribution

Usage

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

Arguments

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

Value

A gpci_dist object.

Examples

```
d_weib <- dist_weibull(shape = 2, scale = 5)
```

fit_distribution	<i>Uncensored Data Parameter Estimation via Maximum Likelihood</i>
------------------	--

Description

Fits process distribution parameters to observed uncensored numeric data using Maximum Likelihood Estimation (MLE).

Usage

```
fit_distribution(data, distribution, method = "mle", initial = NULL)
```

Arguments

data	Numeric vector of observed process observations.
distribution	A gpci_dist object.
method	Optimization method (default "mle").
initial	Named numeric vector or list of initial parameter values. If NULL, sensible defaults or sample moments are used.

Value

An updated gpci_dist object with estimated parameters.

Examples

```
data_sample <- rnorm(100, mean = 10, sd = 2)
dist_fitted <- fit_distribution(data_sample, dist_normal())
dist_fitted$params
```

gpciimpsam	<i>High-Level User Interface Function for Importance Sampling GPCIs</i>
------------	---

Description

Main user interface to fit uncensored data under Importance Sampling and evaluate GPCIs, returning point estimates, parameter chains, GPCI chains, HPD intervals, and convergence diagnostics.

Usage

```
gpciimpsam(
  data,
  pdf = NULL,
  cdf = NULL,
  sf = NULL,
  prior = NULL,
  chain_length = 1000,
  burn_in = 200,
  thinning = 1,
  USL,
  LSL,
  target = (USL + LSL)/2,
  indices = NULL
)
```

Arguments

data	Numeric vector of process observations.
pdf	Custom PDF/PMF 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	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	Target process value.
indices	GPCI names to evaluate.

Value

A list of class "gpciImpSam" with full summary results.

Examples

```
gpciimpsam(
  data = rnorm(50, 10, 1),
  pdf = function(x, mean = 0, sd = 1) dnorm(x, mean, sd),
  cdf = function(x, mean = 0, sd = 1) pnorm(x, mean, sd),
  chain_length = 300, burn_in = 50,
  USL = 13, LSL = 7
)
```

heidelberger_welch	<i>Heidelberger and Welch's MCMC Convergence Diagnostic</i>
--------------------	---

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 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)
heidelberger_welch(samples)
```

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

Description

Computes the Highest Posterior Density (HPD) interval for MCMC posterior samples at specified confidence / significance levels (e.g., 90%, 95%, 99%).

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_gpci	<i>Importance Sampling MCMC Chain Generator for Process Capability Indices</i>
-------------	--

Description

Generates posterior parameter draws and Generalized Process Capability Index (GPCI) chains using Importance Sampling (Sampling Importance Resampling, SIR) for uncensored data.

Usage

```
impsam_gpci(
  data,
  distribution,
  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

data	Numeric vector of uncensored process data.
distribution	A gpci_dist object.
prior	Function representing the joint log-prior density function(params) or a list of prior hyperparameters. Default assumes non-informative flat or conjugate log-priors.
chain_length	Desired length of final retained MCMC 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 for sampling. 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 computes all 13+ 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	Desirable yield baseline (default 0.9973002).

Value

A list of class "gpciImpSam" containing:

param_chain	Matrix of retained MCMC parameter draws.
gpci_chain	Matrix or data frame of retained GPCI draws.
point_estimates	Point estimates of GPCIs based on uncensored data MLE.
specs	Specification limits and setup options.

Examples

```
data_x <- rnorm(50, mean = 10, sd = 1)
dist_norm <- dist_normal()
fit_imp <- impsam_gpci(
  data = data_x,
  distribution = dist_norm,
  chain_length = 500,
  burn_in = 100,
  thinning = 1,
  USL = 13, LSL = 7
)
```

plot.gpciImpSam

Plot Method for gpciImpSam Objects

Description

Visualizes GPCI posterior density distributions, trace plots, and HPD intervals.

Usage

```
## S3 method for class 'gpciImpSam'
plot(x, type = c("density", "trace"), index = NULL, ...)
```

Arguments

x	An object of class "gpciImpSam".
type	Type of plot: "density" (default) or "trace".
index	Character string naming the specific GPCI to plot. Default plots the first index.
...	Additional graphical parameters.

Value

Invisible NULL.

Examples

```
data_x <- rnorm(40, mean = 10, sd = 1)
fit <- impsam_gpci(data = data_x, distribution = dist_normal(), USL = 13, LSL = 7)
plot(fit, type = "density", index = "Cpy")
```

summary.gpciImpSam	<i>Summary and Diagnostics for gpciImpSam Objects</i>
--------------------	---

Description

Computes estimated value, bias, MSE, Risk value, HPD intervals at 90%, 95%, and 99% significance levels, Heidelberger and Welch MCMC convergence diagnostic, and convergence probability.

Usage

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

Arguments

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

Value

A data frame containing diagnostic metrics for each evaluated GPCI.

Examples

```
data_x <- rnorm(40, mean = 10, sd = 1)
fit_imp <- impsam_gpci(data = data_x, distribution = dist_normal(), USL = 13, LSL = 7)
summary(fit_imp)
```

Index

capability, [2](#)
compute_theoretical_moments, [4](#)

define_distribution, [4](#)
dist_exponential, [5](#)
dist_gamma, [6](#)
dist_logistic_exponential, [6](#)
dist_normal, [7](#)
dist_weibull, [7](#)

fit_distribution, [8](#)

gpciimpsam, [8](#)

heidelberger_welch, [10](#)
hpd_interval, [10](#)

impsam_gpci, [11](#)

plot.gpciImpSam, [12](#)

summary.gpciImpSam, [13](#)