Package 'gpcihybridII'

August 21, 2026

Type Package

Title Generalized Process Capability Indices under Hybrid Type-II Censoring

Version 0.1.0

Description A comprehensive, generalized framework for computing, estimating, and validating Generalized Process Capability Indices (GPCIs) under Hybrid Type-II censored lifetime data. Supports user-supplied probability density or mass functions (PDF/PMF), cumulative distribution functions (CDF), survival functions (SF), and quantile functions. Parameter estimation under Hybrid Type-II censoring is performed via Maximum Likelihood Estimation using the 'MleCensoR' package (Childs et al., 2003 <doi:10.1007/BF02517803>; Balakrishnan & Kundu, 2013 <doi:10.1002/nav.21545>). Computes 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), 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 percent, 95 percent, and 99 percent levels of significance using percentile, normal, basic, BCa, BCp, and studentized bootstrap methods. Computes standard errors, mean squared errors, and coverage probabilities for both distribution parameters and capability indices. Integrates goodness-of-fit testing for Hybrid Type-II censored data via the 'gofPHCS' package.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.0.0)

Imports stats, numDeriv, MleCensoR, gofPHCS

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 2026-08-21 13:00:36 UTC

Contents

boot_ci_hybrid2 *Bootstrap Confidence Intervals for Hybrid Type-II Censored Data*

Description

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

Usage

```
boot_ci_hybrid2(
  fit,
  B = 1000,
  alpha = c(0.1, 0.05, 0.01),
  method = c("percentile", "normal", "basic", "BCp", "BCa", "studentized"),
  type = c("parametric", "nonparametric"),
  parallel = FALSE,
  ncpus = 1
)

gpci_boot_hybrid2(
  fit,
  B = 1000,
  alpha = c(0.1, 0.05, 0.01),
  method = c("percentile", "normal", "basic", "BCp", "BCa", "studentized"),
  type = c("parametric", "nonparametric"),
  parallel = FALSE,
  ncpus = 1
)

boot_ci(
  fit,
  B = 1000,
  alpha = c(0.1, 0.05, 0.01),
  method = c("percentile", "normal", "basic", "BCp", "BCa", "studentized"),
  type = c("parametric", "nonparametric"),
  parallel = FALSE,
  ncpus = 1
)
```

Arguments

fit	A gpcifit_hybrid2 object returned by capability_hybrid2.
B	Positive integer scalar specifying number of bootstrap replicates. Defaults to 1000.

alpha	Numeric vector of significance levels. Defaults to c(0.10, 0.05, 0.01) corresponding to 90 percent, 95 percent, and 99 percent confidence levels.
method	Character string specifying confidence interval method. Choices are: "percentile" (default), "normal", "basic", "BCp" (Bias-Corrected Percentile), "BCa" (Bias-Corrected and Accelerated), or "studentized".
type	Character string specifying resampling type: "parametric" (default) or "nonparametric".
parallel	Logical scalar. If TRUE, bootstrap replicates are executed in parallel. Defaults to FALSE.
ncpus	Integer scalar specifying number of CPU cores to use if parallel = TRUE. Defaults to 1.

Value

An S3 object of class "gpci_ci_hybrid2" containing:

ci_table	Data frame of lower and upper confidence bounds, width, and confidence levels for all indices and parameters.
param_perf	Data frame summarizing parameter estimates, standard errors (SE), bias, and mean squared errors (MSE).
index_perf	Data frame summarizing capability index estimates, standard errors (SE), bias, and MSE.
param_reps	Matrix of bootstrap parameter replicates across all iterations.
index_reps	Matrix of bootstrap capability index replicates across all iterations.
fit	The original gpcifit_hybrid2 object.
B	Number of bootstrap replicates.

Examples

```
dist_exp <- dist_exponential(rate = 1)
x <- c(0.2, 0.5, 0.8, 1.1)
fit <- capability_hybrid2(x = x, r = 3, tc = 1.0, n = 10, distribution = dist_exp, USL = 3, LSL = 0)
ci <- boot_ci_hybrid2(fit, B = 50, alpha = c(0.10, 0.05, 0.01), method = "percentile")
print(ci)
```

---

capability_hybrid2 *Compute Process Capability Indices under Hybrid Type-II Censored Data*

Description

Computes classical and generalized Process Capability Indices (GPCIs) under Hybrid Type-II censored lifetime data.

Usage

```
capability_hybrid2(
  x = NULL,
  r = NULL,
  tc = NULL,
  n = NULL,
  distribution,
  USL,
  LSL,
  target = (USL + LSL)/2,
 indices = c("Cpy", "Spmk", "CpTk", "Cpc", "CNpmc", "CNpmkc", "CNpk", "Cp", "Cpk",
    "Cpm", "Cpmk"),
  u = 1,
  v = 1,
  mode = c("moments", "quantile"),
  fit = TRUE,
  start = NULL,
  C0 = 1,
  C1 = 0,
  C2 = 1,
  tolerance_t = USL - LSL,
  P0 = 0.9973002,
  LDL = LSL,
  UDL = USL,
  ...
)

gpci_fit_hybrid2(
  x = NULL,
  r = NULL,
  tc = NULL,
  n = NULL,
  distribution,
  USL,
  LSL,
  target = (USL + LSL)/2,
 indices = c("Cpy", "Spmk", "CpTk", "Cpc", "CNpmc", "CNpmkc", "CNpk", "Cp", "Cpk",
    "Cpm", "Cpmk"),
  u = 1,
  v = 1,
  mode = c("moments", "quantile"),
  fit = TRUE,
  start = NULL,
  C0 = 1,
  C1 = 0,
  C2 = 1,
  tolerance_t = USL - LSL,
  P0 = 0.9973002,
```

```
  LDL = LSL,
  UDL = USL,
  ...
)

capability(
  x = NULL,
  r = NULL,
  tc = NULL,
  n = NULL,
  distribution,
  USL,
  LSL,
  target = (USL + LSL)/2,
  indices = c("Cpy", "Spmk", "CpTk", "Cpc", "CNpmc", "CNpmkc", "CNpk", "Cp", "Cpk",
    "Cpm", "Cpmk"),
  u = 1,
  v = 1,
  mode = c("moments", "quantile"),
  fit = TRUE,
  start = NULL,
  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. Can be NULL (default) if distribution parameters are already estimated or fixed.
r	Positive integer target number of failures. Can be NULL (default) if not fitting raw data.
tc	Positive numeric fixed censoring time. Can be NULL (default) if not fitting raw data.
n	Positive integer total sample size placed on test. Can be NULL (default) if not fitting raw data.
distribution	A gpci_dist_hybrid2 distribution object.
USL	Numeric Upper Specification Limit. Must satisfy USL > LSL.
LSL	Numeric Lower Specification Limit. Must satisfy LSL < USL.
target	Numeric process target. Defaults to (USL + LSL) / 2.
indices	Character vector of capability indices to compute. Choices include: "Cpy", "Spmk", "CpTk", "Cpc", "CNpmc", "CNpmkc", "CNpk", "Cp", "Cpk", "Cpu",

	"Cpl", "Cpm", "Cpmk", "Cp_uv", "Cp_q", "Cpk_q", "Cpu_q", "Cpl_q", "Cpm_q", "Cpmk_q", "CNp_uv".
u	Non-negative numeric weight parameter u for the generalized Cp(u, v) family. Defaults to 1.
v	Non-negative numeric weight parameter v for the generalized Cp(u, v) family. Defaults to 1.
mode	Character string specifying mode of computation: "moments" (default; uses mean and variance) or "quantile" (uses robust quantiles).
fit	Logical scalar. If TRUE (default) and x is supplied, distribution parameters are estimated under Hybrid Type-II censoring. If FALSE, parameters in distribution are used directly without fitting.
start	Optional named list or numeric vector of starting parameter values for MLE. Defaults to NULL.
C0	Non-negative numeric coefficient for the tolerance cost function in CNpmc and CNpmkc. Defaults to 1.
C1	Non-negative numeric coefficient for the tolerance cost function in CNpmc and CNpmkc. Defaults to 0.
C2	Non-negative numeric coefficient for the tolerance cost function in CNpmc and CNpmkc. Defaults to 1.
tolerance_t	Positive numeric process tolerance t for the tolerance cost function. Defaults to USL - LSL.
P0	Desirable process yield for Cpc and Cpy. Defaults to 0.9973002.
LDL	Lower Desired Limit for CpTk. Defaults to LSL.
UDL	Upper Desired Limit for CpTk. Defaults to USL.
...	Additional arguments passed to fit_distribution_hybrid2.

Value

An S3 object of class "gpcifit_hybrid2" containing:

x	Observed failure times.
r	Target number of failures.
tc	Fixed censoring time.
n	Total sample size.
distribution	Fitted or supplied gpci_dist_hybrid2 distribution object.
USL	Upper Specification Limit.
LSL	Lower Specification Limit.
target	Target value.
indices	Character vector of requested capability index names.
estimates	Named numeric vector of capability index point estimates.
p_hat	Expected process non-conformance proportion.
mode	Computation mode used ("moments" or "quantile").
options	List of calculation options and parameters used.

Examples

```
dist_exp <- dist_exponential(rate = 1)
x <- c(0.2, 0.5, 0.8, 1.1)
res <- capability_hybrid2(
  x = x, r = 3, tc = 1.0, n = 10,
  distribution = dist_exp,
  USL = 3, LSL = 0, target = 1.5,
  indices = c("Cpy", "Spmk", "CpTk", "Cpc", "CNpmc", "CNpmkc", "Cp", "Cpk", "Cpm", "Cpmk")
)
print(res)
```

---

coef.gpcifit_hybrid2 *Coef Method for gpcifit_hybrid2*

Description

Extract point estimates of capability indices or fitted distribution parameters.

Usage

```
## S3 method for class 'gpcifit_hybrid2'
coef(object, what = c("indices", "parameters"), ...)
```

Arguments

`object`	An object of class `gpcifit_hybrid2`.
`what`	Character string: `"indices"` (default) or `"parameters"`.
`...`	Additional arguments.

Value

Named numeric vector of indices or parameters.

---

compute_diagnostics_hybrid2 *Compute Standard Errors, MSE, and Coverage Probabilities for Hybrid Type-II Censoring*

Description

Evaluates standard errors, mean squared errors (MSE), bias, and empirical coverage probabilities for model parameters and Generalized Process Capability Indices (GPCIs) using bootstrap resampling.

Usage

```
compute_diagnostics_hybrid2(
  fit,
  true_params = NULL,
  true_indices = NULL,
  B = 1000,
  alpha = c(0.1, 0.05, 0.01)
)

compute_diagnostics(
  fit,
  true_params = NULL,
  true_indices = NULL,
  B = 1000,
  alpha = c(0.1, 0.05, 0.01)
)
```

Arguments

fit	A gpcifit_hybrid2 object returned by capability_hybrid2.
true_params	Optional named numeric list or vector of true parameter values. Defaults to NULL.
true_indices	Optional named numeric vector of true capability index values. Defaults to NULL.
B	Positive integer scalar specifying number of bootstrap replicates for estimation. Defaults to 1000.
alpha	Numeric vector of significance levels for coverage probabilities. Defaults to c(0.10, 0.05, 0.01).

Value

An S3 object of class "gpci_diagnostics_hybrid2" containing:

se_params	Named numeric vector of standard errors for model parameters.
bias_params	Named numeric vector of empirical bias for model parameters.
mse_params	Named numeric vector of mean squared errors for model parameters.
coverage_params	List of parameter empirical coverage probabilities across significance levels.
se_indices	Named numeric vector of standard errors for capability indices.
bias_indices	Named numeric vector of empirical bias for capability indices.
mse_indices	Named numeric vector of mean squared errors for capability indices.
coverage_indices	List of capability index empirical coverage probabilities across significance levels.
alpha_levels	Numeric vector of significance levels used.

`bootstrap_replicates`
: Number of bootstrap replicates.

`fit`
: The original `gpcifit_hybrid2` object.

Examples

```
dist_exp <- dist_exponential(rate = 1)
x <- c(0.2, 0.5, 0.8, 1.1)
fit <- capability_hybrid2(x = x, r = 3, tc = 1.0, n = 10, distribution = dist_exp, USL = 3, LSL = 0)
diag <- compute_diagnostics_hybrid2(fit, B = 50, true_params = c(rate = 1))
print(diag)
```

`compute_theoretical_moments_hybrid2`

Theoretical Moments of a Distribution

Description

Computes theoretical mean and variance of a distribution using adaptive numerical integration over effective quantile support.

Usage

```
compute_theoretical_moments_hybrid2(dist)
```

Arguments

`dist`
: A `gpci_dist_hybrid2` distribution object.

Value

A list with two numeric elements:

`mean`
: Theoretical mean of the distribution.

`var`
: Theoretical variance of the distribution.

Examples

```
dist_w <- dist_weibull(shape = 2, scale = 5)
compute_theoretical_moments_hybrid2(dist_w)
```

confint.gpcifit_hybrid2 *Confint Method for gpcifit_hybrid2*

Description

Calculate bootstrap confidence intervals for capability indices at specified significance levels (e.g. 90 percent, 95 percent, 99 percent).

Usage

```
## S3 method for class 'gpcifit_hybrid2'
confint(
  object,
  parm = NULL,
  level = 0.95,
  B = 1000,
  method = "percentile",
  ...
)
```

Arguments

`object`	An object of class `gpcifit_hybrid2`.
`parm`	Optional vector of index names.
`level`	Confidence level (default is 0.95; can be 0.90, 0.95, 0.99 or a vector).
`B`	Number of bootstrap replicates (default 1000).
`method`	Bootstrap CI method (`"percentile"`, `"normal"`, `"basic"`, `"BCp"`, `"BCa"`, `"studentized"`).
`...`	Additional arguments passed to `boot_ci_hybrid2`.

Value

A matrix containing lower and upper confidence limits.

define_distribution_hybrid2 *Define a Process Distribution for Hybrid Type-II Censored Data*

Description

Constructor to define a continuous or discrete probability distribution for process capability analysis under Hybrid Type-II censored data. The distribution can be defined by specifying any one (or more) of its Probability Density/Mass Function (PDF/PMF), Cumulative Distribution Function (CDF), Survival Function (SF), or Quantile Function. Missing functions are automatically and numerically derived using adaptive integration and root-finding algorithms.

Usage

```
define_distribution_hybrid2(
  name,
  pdf = NULL,
  cdf = NULL,
  sf = NULL,
  quantile = NULL,
  params = list(),
  support = c(-Inf, Inf)
)

gpci_dist_hybrid2(
  name,
  pdf = NULL,
  cdf = NULL,
  sf = NULL,
  quantile = NULL,
  params = list(),
  support = c(-Inf, Inf)
)

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

Arguments

`name`	Character string naming the distribution (e.g. `"custom_weibull"`).
`pdf`	Optional function representing the probability density/mass function (PDF/PMF). Must be of the form `function(x, ...)`. Defaults to `NULL`.
`cdf`	Optional function representing the cumulative distribution function (CDF). Must be of the form `function(x, ...)`. Defaults to `NULL`.
`sf`	Optional function representing the survival function (SF = 1 - CDF). Must be of the form `function(x, ...)`. Defaults to `NULL`.
`quantile`	Optional function representing the quantile function. Must be of the form `function(p, ...)`. Defaults to `NULL`.
`params`	Named list of numeric parameter values for the distribution. Defaults to `list()`.
`support`	Numeric vector of length 2 defining the lower and upper support bounds. Defaults to `c(-Inf, Inf)`.

Value

An S3 object of class "gpci_dist_hybrid2" containing:

name	Character string of the distribution name.
pdf	The probability density function.
cdf	The cumulative distribution function.
sf	The survival function.
quantile	The quantile function.
params	Named list of numeric parameter values.
param_names	Character vector of parameter names.
support	Numeric vector of length 2 specifying support limits.

Examples

```
# Define custom Weibull distribution using only the survival function
custom_weib <- define_distribution_hybrid2(
  name = "custom_weibull",
  sf = function(x, shape, scale) pweibull(x, shape, scale, lower.tail = FALSE),
  params = list(shape = 2, scale = 10),
  support = c(0, Inf)
)
print(custom_weib)
```

---

dist_exponential *Exponential Distribution Constructor*

Description

Exponential Distribution Constructor

Usage

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

Arguments

rate	Rate parameter.

Value

A gpci_dist_hybrid2 object for Exponential distribution.

Examples

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

dist_gamma *Gamma Distribution Constructor*

Description

Gamma Distribution Constructor

Usage

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

Arguments

shape	Shape parameter.
scale	Scale parameter.

Value

A gpci_dist_hybrid2 object for Gamma distribution.

Examples

```
dist_g <- dist_gamma(shape = 3, scale = 2)
```

dist_logistic *Logistic Distribution Constructor*

Description

Logistic Distribution Constructor

Usage

```
dist_logistic(location = 0, scale = 1)
```

Arguments

location	Location parameter.
scale	Scale parameter.

Value

A gpci_dist_hybrid2 object for Logistic distribution.

Examples

```
dist_l <- dist_logistic(location = 0, scale = 1)
```

dist_loglogistic *Log-Logistic Distribution Constructor*

Description

Log-Logistic Distribution Constructor

Usage

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

Arguments

shape	Shape parameter.
scale	Scale parameter.

Value

A gpci_dist_hybrid2 object for Log-Logistic distribution.

Examples

```
dist_ll <- dist_loglogistic(shape = 2, scale = 3)
```

dist_lognormal *Lognormal Distribution Constructor*

Description

Lognormal Distribution Constructor

Usage

```
dist_lognormal(meanlog = 0, sdlog = 1)
```

Arguments

meanlog	Mean of the logarithm.
sdlog	Standard deviation of the logarithm.

Value

A gpci_dist_hybrid2 object for Lognormal distribution.

Examples

```
dist_ln <- dist_lognormal(meanlog = 1, sdlog = 0.5)
```

dist_normal *Normal Distribution Constructor*

Description

Normal Distribution Constructor

Usage

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

Arguments

mean	Mean parameter.
sd	Standard deviation parameter.

Value

A gpci_dist_hybrid2 object for Normal distribution.

Examples

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

dist_weibull *Weibull Distribution Constructor*

Description

Weibull Distribution Constructor

Usage

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

Arguments

shape	Shape parameter.
scale	Scale parameter.

Value

A gpci_dist_hybrid2 object for Weibull distribution.

Examples

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

`eval_performance_hybrid2` *Monte Carlo Simulation Framework for Hybrid Type-II Censoring*

Description

Evaluates empirical standard errors, bias, MSE, and coverage probabilities for model parameters and Generalized Process Capability Indices (GPCIs) across Monte Carlo simulation runs.

Usage

```
eval_performance_hybrid2(
  distribution,
  r,
  tc,
  n,
  USL,
  LSL,
  target = (USL + LSL)/2,
  true_indices = NULL,
  M = 100,
  B = 200,
  alpha = c(0.1, 0.05, 0.01)
)

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

gpci_sim_hybrid2(
  distribution,
  r,
  tc,
  n,
  USL,
  LSL,
  target = (USL + LSL)/2,
```

```
  true_indices = NULL,
  M = 100,
  B = 200,
  alpha = c(0.1, 0.05, 0.01)
)
```

Arguments

distribution	A gpci_dist_hybrid2 distribution object with known true parameters.
r	Positive integer target number of failures.
tc	Positive numeric fixed censoring time.
n	Positive integer total number of initial units placed on life test.
USL	Numeric Upper Specification Limit.
LSL	Numeric Lower Specification Limit.
target	Numeric process target. Defaults to (USL + LSL) / 2.
true_indices	Optional named numeric vector of true capability index values. If NULL (default), derived from distribution.
M	Positive integer scalar specifying number of Monte Carlo simulation runs. Defaults to 100.
B	Positive integer scalar specifying number of bootstrap iterations per simulation run. Defaults to 200.
alpha	Numeric vector of significance levels. Defaults to c(0.10, 0.05, 0.01).

Value

An S3 object of class "gpci_sim_hybrid2" containing:

param_summary	Data frame of true values, mean estimates, standard errors (SE), bias, and MSE for model parameters.
index_summary	Data frame of true values, mean estimates, standard errors (SE), bias, and MSE for capability indices.
coverage_probabilities	Matrix of empirical coverage probabilities across all significance levels.
M	Total number of simulation runs requested.
valid_runs	Number of successfully converged simulation runs.
B	Number of bootstrap replicates per run.
true_parameters	Vector of true parameter values.
true_indices	Vector of true capability index values.

Examples

```
dist_true <- dist_weibull(shape = 2, scale = 5)
sim_res <- eval_performance_hybrid2(
  distribution = dist_true, r = 5, tc = 4.0, n = 12,
  USL = 8, LSL = 1, M = 5, B = 20
)
print(sim_res)
```

---

fit_distribution_hybrid2

Parameter Estimation for Hybrid Type-II Censored Data using MleCensoR

Description

Fits process distribution parameters under Hybrid Type-II censored data using the `MleCensoR` package (`MleCensoR::mle_hybrid_type2`) with robust numerical optimization fallbacks.

Usage

```
fit_distribution_hybrid2(
  x,
  r,
  tc,
  n,
  distribution,
  start = NULL,
  method = NULL,
  lower = NULL,
  upper = NULL,
  ...
)

gpci_fit_hybrid2_dist(
  x,
  r,
  tc,
  n,
  distribution,
  start = NULL,
  method = NULL,
  lower = NULL,
  upper = NULL,
  ...
)

fit_hybrid2(
```

```
  x,
  r,
  tc,
  n,
  distribution,
  start = NULL,
  method = NULL,
  lower = NULL,
  upper = NULL,
  ...
)
```

Arguments

x	Numeric vector of observed failure times (sorted in ascending order).
r	Positive integer scalar specifying the target number of failures.
tc	Positive numeric scalar specifying the fixed censoring time.
n	Positive integer scalar specifying the total initial sample size placed on test.
distribution	A gpci_dist_hybrid2 distribution object specifying the target model.
start	Optional named numeric vector or list of starting parameter values. If NULL (default), uses distribution$params.
method	Character string specifying optimization method (e.g. "L-BFGS-B", "BFGS", "Nelder-Mead"). If NULL (default), chosen automatically.
lower	Optional numeric vector of lower bounds for parameters under bounded optimization. Defaults to NULL.
upper	Optional numeric vector of upper bounds for parameters under bounded optimization. Defaults to NULL.
...	Additional arguments passed to MleCensoR::mle_hybrid_type2 or stats::optim.

Value

A fitted gpci_dist_hybrid2 distribution object with updated parameters and attributes:

vcov	Asymptotic variance-covariance matrix of estimated parameters.
se	Numeric vector of standard errors for estimated parameters.
std_errs	Alias for se.
loglik	Scalar log-likelihood value at the optimum.
vdata	Validated Hybrid Type-II censoring data summary list.

Examples

```
dist_exp <- dist_exponential(rate = 1)
x <- c(0.2, 0.5, 0.8, 1.1)
fit <- fit_distribution_hybrid2(x = x, r = 3, tc = 1.0, n = 10, distribution = dist_exp)
print(fit)
```

gof_test_hybrid2 *Goodness-of-Fit Testing for Hybrid Type-II Censored Data using gofPHCS*

Description

Performs goodness-of-fit testing for Hybrid Type-II censored lifetime data using the gofPHCS package (gofPHCS::gof_test).

Usage

```
gof_test_hybrid2(
  fit = NULL,
  x = NULL,
  r = NULL,
  tc = NULL,
  n = NULL,
  distribution = NULL,
  statistic = "auto",
  p.method = c("auto", "asymptotic", "montecarlo"),
  nsim = 999,
  seed = NULL,
  conf.level = 0.95,
  ...
)

gof_test(
  fit = NULL,
  x = NULL,
  r = NULL,
  tc = NULL,
  n = NULL,
  distribution = NULL,
  statistic = "auto",
  p.method = c("auto", "asymptotic", "montecarlo"),
  nsim = 999,
  seed = NULL,
  conf.level = 0.95,
  ...
)
```

Arguments

fit	Optional gpcifit_hybrid2 object returned by capability_hybrid2. If supplied, x, r, tc, n, and distribution are extracted automatically. Defaults to NULL.

x	Numeric vector of observed failure times (required if fit is not supplied). Defaults to NULL.
r	Positive integer target number of failures (required if fit is not supplied). Defaults to NULL.
tc	Positive numeric fixed censoring time (required if fit is not supplied). Defaults to NULL.
n	Positive integer total sample size (required if fit is not supplied). Defaults to NULL.
distribution	A gpci_dist_hybrid2 distribution object (required if fit is not supplied). Defaults to NULL.
statistic	Character string specifying the test statistic (e.g. "auto", "AD", "CvM", "KS"). Defaults to "auto".
p.method	Character string specifying method for calculating p-values: "auto" (default), "asymptotic", or "montecarlo".
nsim	Positive integer scalar specifying number of Monte Carlo replicates. Defaults to 999.
seed	Optional integer seed for reproducibility. Defaults to NULL.
conf.level	Numeric confidence level for test. Defaults to 0.95.
...	Additional arguments passed to gofPHCS::gof_test.

Value

An S3 object of class "gpci_gof_hybrid2" containing:

fit	The original gpcifit_hybrid2 object, or NULL if raw data was passed.
cens_data	The gofPHCS censored data structure.
gof_result	The returned test result object from gofPHCS::gof_test.
distribution	The tested gpci_dist_hybrid2 distribution object.

Examples

```
dist_exp <- dist_exponential(rate = 0.5)
x <- c(0.2, 0.5, 0.8, 1.1)
fit <- capability_hybrid2(x = x, r = 3, tc = 1.0, n = 10, distribution = dist_exp, USL = 3, LSL = 0)
gof_res <- gof_test_hybrid2(fit, p.method = "montecarlo", nsim = 50)
print(gof_res)
```

gpci_index_hybrid2 *Extract a Specific Capability Index*

Description

Extract a Specific Capability Index

Usage

```
gpci_index_hybrid2(fit, index)
```

Arguments

fit	A gpcifit_hybrid2 object.
index	Name of the capability index to retrieve or compute.

Value

Numeric scalar value of the index.

Examples

```
dist_exp <- dist_exponential(rate = 1)
x <- c(0.2, 0.5, 0.8, 1.1)
fit <- capability_hybrid2(x = x, r = 3, tc = 1.0, n = 10, distribution = dist_exp, USL = 3, LSL = 0)
gpci_index_hybrid2(fit, "Cpy")
```

print.gpcifit_hybrid2 *Print Method for gpcifit_hybrid2*

Description

Print Method for gpcifit_hybrid2

Usage

```
## S3 method for class 'gpcifit_hybrid2'
print(x, ...)
```

Arguments

x	An object of class gpcifit_hybrid2.
...	Additional print arguments.

Value

The invisible object x.

print.gpci_ci_hybrid2 *Print Method for Hybrid Type-II Bootstrap CIs*

Description

Print Method for Hybrid Type-II Bootstrap CIs

Usage

```
## S3 method for class 'gpci_ci_hybrid2'
print(x, ...)
```

Arguments

x	An object of class gpci_ci_hybrid2.
...	Additional print arguments.

Value

The invisible object x.

print.gpci_diagnostics_hybrid2 *Print Method for Hybrid Type-II Diagnostics*

Description

Print Method for Hybrid Type-II Diagnostics

Usage

```
## S3 method for class 'gpci_diagnostics_hybrid2'
print(x, ...)
```

Arguments

x	An object of class gpci_diagnostics_hybrid2.
...	Additional print arguments.

Value

The invisible object x.

print.gpci_dist_hybrid2 *Print Method for gpci_dist_hybrid2*

Description

Print Method for gpci_dist_hybrid2

Usage

```
## S3 method for class 'gpci_dist_hybrid2'
print(x, ...)
```

Arguments

x	An object of class gpci_dist_hybrid2.
...	Additional print arguments.

Value

The invisible object x.

print.gpci_gof_hybrid2 *Print Method for gpci_gof_hybrid2*

Description

Print Method for gpci_gof_hybrid2

Usage

```
## S3 method for class 'gpci_gof_hybrid2'
print(x, ...)
```

Arguments

x	An object of class gpci_gof_hybrid2.
...	Additional print arguments.

Value

The invisible object x.

`print.gpci_sim_hybrid2` *Print Method for Hybrid Type-II Monte Carlo Simulation*

Description

Print Method for Hybrid Type-II Monte Carlo Simulation

Usage

```
## S3 method for class 'gpci_sim_hybrid2'
print(x, ...)
```

Arguments

`x`	An object of class `gpci_sim_hybrid2`.
`...`	Additional print arguments.

Value

The invisible object `x`.

`summary.gpcifit_hybrid2` *Summary Method for gpcifit_hybrid2*

Description

Summary Method for gpcifit_hybrid2

Usage

```
## S3 method for class 'gpcifit_hybrid2'
summary(object, ...)
```

Arguments

`object`	An object of class `gpcifit_hybrid2`.
`...`	Additional arguments.

Value

The invisible object `object`.

summary.gpci_ci_hybrid2 *Summary Method for Hybrid Type-II Bootstrap CIs*

Description

Summary Method for Hybrid Type-II Bootstrap CIs

Usage

```
## S3 method for class 'gpci_ci_hybrid2'
summary(object, ...)
```

Arguments

object	An object of class gpci_ci_hybrid2.
...	Additional arguments.

Value

The invisible object object.

summary.gpci_diagnostics_hybrid2 *Summary Method for Hybrid Type-II Diagnostics*

Description

Summary Method for Hybrid Type-II Diagnostics

Usage

```
## S3 method for class 'gpci_diagnostics_hybrid2'
summary(object, ...)
```

Arguments

object	An object of class gpci_diagnostics_hybrid2.
...	Additional arguments.

Value

The invisible object object.

summary.gpci_dist_hybrid2 *Summary Method for gpci_dist_hybrid2*

Description

Summary Method for gpci_dist_hybrid2

Usage

```
## S3 method for class 'gpci_dist_hybrid2'
summary(object, ...)
```

Arguments

object	An object of class gpci_dist_hybrid2.
...	Additional arguments.

Value

The invisible object object.

summary.gpci_gof_hybrid2 *Summary Method for gpci_gof_hybrid2*

Description

Summary Method for gpci_gof_hybrid2

Usage

```
## S3 method for class 'gpci_gof_hybrid2'
summary(object, ...)
```

Arguments

object	An object of class gpci_gof_hybrid2.
...	Additional arguments.

Value

The invisible object object.

summary.gpci_sim_hybrid2 *Summary Method for Hybrid Type-II Monte Carlo Simulation*

Description

Summary Method for Hybrid Type-II Monte Carlo Simulation

Usage

```
## S3 method for class 'gpci_sim_hybrid2'
summary(object, ...)
```

Arguments

object	An object of class gpci_sim_hybrid2.
...	Additional arguments.

Value

The invisible object object.

vcov.gpcifit_hybrid2 *Vcov Method for gpcifit_hybrid2*

Description

Extract asymptotic variance-covariance matrix of fitted distribution parameters.

Usage

```
## S3 method for class 'gpcifit_hybrid2'
vcov(object, ...)
```

Arguments

object	An object of class gpcifit_hybrid2.
...	Additional arguments.

Value

Variance-covariance matrix.

Index