Package 'ProcessCapabilityR'

August 3, 2026

Type Package

Title Classical and Generalized Process Capability Indices

Version 0.1.0

Maintainer Shikhar Tyagi <shikhar1093tyagi@gmail.com>

Description Computes classical process capability indices (Cp, Cpk, Cpu, Cpl, Cpm, Cpmk, Pp, Ppk, Ppu, Ppl, Z) and the generalized process capability index Cpy (Maiti, Saha & Nanda, 2010) <doi:10.1080/16843703.2010.11673233> for any continuous or discrete quality characteristic. Users supply the probability density function (PDF) and cumulative distribution function (CDF) of the characteristic, and the package returns point estimates, bootstrap confidence intervals (percentile and BCa), and sensitivity tables/plots across ranges of short-term standard deviation (sigma), long-term standard deviation (s), desired yield (p0), and significance levels. Classical indices are recoverable as special cases under the normal distribution. The package follows the theory and notation of Kane (1986) <doi:10.1080/00224065.1986.11978984>, Chan, Cheng & Spiring (1988) <doi:10.1080/00224065.1988.11979102>, Pearn, Kotz & Johnson (1992) <doi:10.1080/00224065.1992.11979403>, Kotz & Johnson (2002) <doi:10.1080/00224065.2002.11980119>, Montgomery (2020, ISBN:978-1-119-39930-8), Juran (1974, ISBN:978-0-07-033176-1), Harry & Schroeder (2000, ISBN:978-0-385-49437-2), and the AIAG SPC Reference Manual (2005, ISBN:978-1-60534-026-3).

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 3.5.0)

Imports stats

Suggests ggplot2 (>= 3.0.0), 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],
Vrijesh Tripathi [aut]

Repository CRAN

Date/Publication 2026-08-03 18:00:24 UTC

Contents

---

cp *Process Potential Index (Cp)*

Description

$C_p = (USL - LSL)/(6\sigma)$.

Usage

```
cp(LSL, USL, sigma)
```

Arguments

LSL	Lower specification limit.
USL	Upper specification limit.
sigma	Short-term (within-subgroup) standard deviation.

Value

Numeric value of C_p.

See Also

pci

Examples

```
cp(LSL = 57, USL = 63, sigma = 1)   # 1.0
```

---

cpk *Process Capability Index (Cpk)*

Description

$C_{pk} = \min\big((USL - \mu)/(3\sigma),\ (\mu - LSL)/(3\sigma)\big)$.

Usage

```
cpk(LSL, USL, mu, sigma)
```

Arguments

LSL	Lower specification limit.
USL	Upper specification limit.
mu	Short-term process mean.
sigma	Short-term standard deviation.

Value

Numeric value of C_{pk}.

See Also

pci

Examples

```
cpk(LSL = 57, USL = 63, mu = 60, sigma = 1)   # 1.0
```

cpl *Lower Capability Index (Cpl)*

Description

$C_{pl} = (\mu - LSL)/(3\sigma)$.

Usage

```
cpl(LSL, mu, sigma)
```

Arguments

LSL	Lower specification limit.
mu	Short-term process mean.
sigma	Short-term standard deviation.

Value

Numeric value of C_{pl}.

See Also

pci

Examples

```
cpl(LSL = 57, mu = 60, sigma = 1)   # 1.0
```

cpm *Taguchi Capability Index (Cpm)*

Description

$C_{pm} = (USL - LSL)/\left(6\sqrt{\sigma^2 + (\mu - T)^2}\right)$.

Usage

```
cpm(LSL, USL, mu, sigma, target)
```

Arguments

LSL	Lower specification limit.
USL	Upper specification limit.
mu	Short-term process mean.
sigma	Short-term standard deviation.
target	Target value T.

Value

Numeric value of C_{pm}.

See Also

pci

Examples

```
cpm(LSL = 57, USL = 63, mu = 60, sigma = 1, target = 60)   # 1.0
```

---

cpmk *Modified Taguchi Capability Index (Cpmk)*

Description

$C_{pmk} = \min\big((USL - \mu),\ (\mu - LSL)\big) / \big(3\sqrt{\sigma^2 + (\mu - T)^2}\big)$.

Usage

```
cpmk(LSL, USL, mu, sigma, target)
```

Arguments

LSL	Lower specification limit.
USL	Upper specification limit.
mu	Short-term process mean.
sigma	Short-term standard deviation.
target	Target value T.

Value

Numeric value of C_{pmk}.

See Also

pci

Examples

```
cpmk(LSL = 57, USL = 63, mu = 60, sigma = 1, target = 60)   # 1.0
```

cpu *Upper Capability Index (Cpu)*

Description

$C_{pu} = (USL - \mu)/(3\sigma)$.

Usage

```
cpu(USL, mu, sigma)
```

Arguments

USL	Upper specification limit.
mu	Short-term process mean.
sigma	Short-term standard deviation.

Value

Numeric value of C_{pu}.

See Also

pci

Examples

```
cpu(USL = 63, mu = 60, sigma = 1)   # 1.0
```

cpy *Generalized Process Capability Index (Cpy)*

Description

$C_{py} = p/p_0 = \big(F(USL) - F(LSL)\big)/\big(F(UDL) - F(LDL)\big)$, where F is the CDF from `dist` (Maiti, Saha & Nanda, 2010).

Usage

```
cpy(dist, LSL, USL, LDL = NULL, UDL = NULL, p0 = NULL)
```

Arguments

dist	A pci_dist object.
LSL	Lower specification limit.
USL	Upper specification limit.
LDL	Lower desirable limit (optional if p0 is given).
UDL	Upper desirable limit (optional if p0 is given).
p0	Desired yield (optional if LDL/UDL are given).

Value

Numeric value of C_{py}.

See Also

pci, pci_dist

Examples

```
d <- pci_dist_normal(mean = 60, sd = 1)
cpy(d, LSL = 57, USL = 63, LDL = 57, UDL = 63)   # 1.0
```

---

pci *Compute a Process Capability Index*

Description

Computes a single process capability or performance index from the supplied parameters.

Usage

```
pci(
  index,
  dist = NULL,
  LSL = NULL,
  USL = NULL,
  target = NULL,
  mu = NULL,
  sigma = NULL,
  xbar = NULL,
  s = NULL,
  LDL = NULL,
  UDL = NULL,
  p0 = NULL
)
```

Arguments

`index`	Character string: one of `"Cp"`, `"Cpk"`, `"Cpu"`, `"Cpl"`, `"Cpm"`, `"Cpmk"`, `"Pp"`, `"Ppk"`, `"Ppu"`, `"Ppl"`, `"Z"`, or `"Cpy"`.
`dist`	A `pci_dist` object (required for `"Cpy"`; optional otherwise).
`LSL`, `USL`	Lower and upper specification limits.
`target`	Target value (required for `Cpm` and `Cpmk`).
`mu`, `sigma`	Short-term (within-subgroup) mean and standard deviation.
`xbar`, `s`	Long-term (overall) mean and standard deviation.
`LDL`, `UDL`	Lower and upper desirable limits for `Cpy`.
`p0`	Desired yield for `Cpy`. If `LDL`/`UDL` are not supplied, they are computed from `p0` using `compute_desirable_limits`.

Details

The twelve supported indices are:

Cp $(USL - LSL)/(6\sigma)$

Cpk $\min\big((USL - \mu)/(3\sigma),\ (\mu - LSL)/(3\sigma)\big)$

Cpu $(USL - \mu)/(3\sigma)$

Cpl $(\mu - LSL)/(3\sigma)$

Cpm $(USL - LSL)/\big(6\sqrt{\sigma^2 + (\mu - T)^2}\big)$

Cpmk $\min\big((USL - \mu),\ (\mu - LSL)\big)/\big(3\sqrt{\sigma^2 + (\mu - T)^2}\big)$

Pp $(USL - LSL)/(6s)$

Ppk $\min\big((USL - \bar{x})/(3s),\ (\bar{x} - LSL)/(3s)\big)$

Ppu $(USL - \bar{x})/(3s)$

Ppl $(\bar{x} - LSL)/(3s)$

Z $\min\big((USL - \mu)/\sigma,\ (\mu - LSL)/\sigma\big)$

Cpy $p/p_0 = \big(F(USL) - F(LSL)\big)/\big(F(UDL) - F(LDL)\big)$

Value

A single numeric value, or `NA` with a warning when a one-sided index is requested but the relevant specification limit is `NULL`.

See Also

`pci_dist`, `pci_ci`, `pci_grid`

Examples

```
pci("Cp",  LSL = 57, USL = 63, sigma = 1)            # 1.0
pci("Cpk", LSL = 57, USL = 63, mu = 60, sigma = 1)   # 1.0
pci("Z",   LSL = 57, USL = 63, mu = 60, sigma = 1)   # 3.0

# Generalized Cpy
d <- pci_dist_normal(60, 1)
pci("Cpy", dist = d, LSL = 57, USL = 63, LDL = 57, UDL = 63) # 1.0
```

---

pci_ci *Bootstrap Confidence Interval for a Process Capability Index*

Description

Computes a confidence interval for a PCI via parametric or nonparametric bootstrap.

Usage

```
pci_ci(
  index,
  dist,
  n,
  data = NULL,
  LSL = NULL,
  USL = NULL,
  target = NULL,
  LDL = NULL,
  UDL = NULL,
  p0 = NULL,
  alpha = 0.05,
  B = 2000,
  method = c("percentile", "bca")
)
```

Arguments

index	Character: one of the 12 index names (see pci).
dist	A pci_dist object.
n	Integer sample size.
data	Optional numeric vector of observed data. If supplied, nonparametric bootstrap is used instead of parametric.
LSL, USL	Specification limits.
target	Target value (for Cpm/Cpmk).
LDL, UDL	Desirable limits (for Cpy).

p0	Desired yield (for Cpy).
alpha	Significance level; the interval covers $1 - \alpha$. Default 0.05.
B	Number of bootstrap replications. Default 2000.
method	Character: "percentile" (default) or "bca".

Details

Parametric bootstrap (when data = NULL): B samples of size n are drawn from dist via inverse-CDF sampling. For each sample the process mean, standard deviation (and, for C_{py}, the empirical CDF) are estimated, and the index is recomputed.

Nonparametric bootstrap (when data is supplied): B resamples of the same length are drawn with replacement from data.

Two interval methods are available:

percentile Uses the alpha/2 and 1 - alpha/2 quantiles of the bootstrap distribution.

bca Bias-corrected and accelerated interval using jackknife acceleration and bootstrap bias correction (Efron, 1987).

Value

An S3 object of class "pci_ci" with components:

index	The index name.
estimate	Point estimate.
lower	Lower CI bound.
upper	Upper CI bound.
alpha	The significance level used.
method	The interval method used.
boot_values	Numeric vector of all B bootstrap estimates.

See Also

pci, pci_grid

Examples

```
d <- pci_dist_normal(mean = 60, sd = 1)
ci <- pci_ci("Cp", dist = d, n = 30, LSL = 57, USL = 63,
             alpha = 0.05, B = 500)
print(ci)
```

pci_dist *Create a Distribution Specification for Process Capability Analysis*

Description

Constructs an S3 object of class "pci_dist" that encapsulates the probability density function (PDF) and cumulative distribution function (CDF) of a quality characteristic. This object is passed to pci, pci_ci, and pci_grid when computing generalised indices such as C_{py}.

Usage

```
pci_dist(pdf, cdf = NULL, params = list(), support = c(-Inf, Inf))
```

Arguments

pdf	A function function(x, ...) returning the density at x. Additional distribution parameters are passed via params.
cdf	A function function(x, ...) returning the cumulative probability $P(Z \leq x)$. If NULL, a numerical CDF is derived via integrate over pdf.
params	A named list of distribution parameters forwarded to pdf and cdf via do.call.
support	A length-2 numeric vector giving the lower and upper bounds of the distribution's support (default c(-Inf, Inf)).

Value

An S3 object of class "pci_dist" with elements:

pdf_fn	Closure that evaluates the PDF at a given point.
cdf_fn	Closure that evaluates the CDF at a given point.
params	The supplied parameter list.
support	The supplied support bounds.
cdf_is_numeric	Logical; TRUE if CDF was derived numerically.

Examples

```
# Normal distribution
d <- pci_dist(
  pdf = function(x, mean, sd) dnorm(x, mean, sd),
  cdf = function(x, mean, sd) pnorm(x, mean, sd),
  params = list(mean = 0, sd = 1)
)
d$cdf_fn(0)   # 0.5

# Weibull with only PDF (CDF derived numerically)
d2 <- pci_dist(
  pdf = function(x, shape, scale) dweibull(x, shape, scale),
```

```
  params = list(shape = 2, scale = 10),
  support = c(0, 60)
)
```

`pci_dist_normal` *Normal Distribution Shortcut*

Description

Convenience constructor for a `pci_dist` based on the normal distribution. Useful for sanity-checking that C_{py} collapses to the classical indices when F is the normal CDF.

Usage

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

Arguments

`mean`	Mean of the normal distribution.
`sd`	Standard deviation of the normal distribution.

Value

An S3 object of class `"pci_dist"` encapsulating the normal distribution PDF and CDF functions, parameter list, and support bounds.

Examples

```
d <- pci_dist_normal(mean = 60, sd = 1)
d$cdf_fn(60)   # 0.5
d$pdf_fn(60)   # ~ 0.3989
```

`pci_grid` *Sensitivity Grid of Process Capability Indices*

Description

Computes a PCI across a grid of parameter values (short-term standard deviation `sigma`, long-term standard deviation `s`, and/or desired yield `p0`), optionally with bootstrap confidence intervals at multiple significance levels.

Usage

```
pci_grid(
  index,
  dist = NULL,
  LSL = NULL,
  USL = NULL,
  target = NULL,
  sigma_vals = NULL,
  s_vals = NULL,
  p0_vals = NULL,
  mu = NULL,
  xbar = NULL,
  alpha_vals = c(0.1, 0.05, 0.01),
  n = NULL,
  B = 2000
)
```

Arguments

index	Character: one of the 12 index names (see pci).
dist	A pci_dist object (required for "Cpy").
LSL, USL	Specification limits.
target	Target value (for Cpm/Cpmk).
sigma_vals	Numeric vector of σ values to sweep (for capability indices).
s_vals	Numeric vector of s values to sweep (for performance indices).
p0_vals	Numeric vector of p_0 values to sweep (for Cpy).
mu	Short-term process mean.
xbar	Long-term process mean.
alpha_vals	Numeric vector of significance levels. Default c(0.10, 0.05, 0.01).
n	Sample size for bootstrap CIs. If NULL, CIs are omitted.
B	Bootstrap replications. Default 2000.

Value

A data.frame (with additional class "pci_grid") in long format with columns: index, sigma, s, p0, alpha, estimate, lower, upper. Columns that do not apply to the chosen index contain NA.

See Also

pci, pci_ci, plot.pci_grid

Examples

```
grid <- pci_grid("Cp", LSL = 57, USL = 63,
                 sigma_vals = c(0.5, 1.0, 1.5),
                 mu = 60, alpha_vals = c(0.05, 0.01))
print(grid)
```

`plot.pci_grid` *Plot a Process Capability Sensitivity Grid*

Description

Produces a **ggplot2** line-and-ribbon plot showing the PCI estimate and confidence bands as a function of the chosen sweep variable, faceted by significance level.

Usage

```
## S3 method for class 'pci_grid'
plot(x, x_axis = c("sigma", "s", "p0"), facet_by = "alpha", ...)
```

Arguments

`x`	A `pci_grid` object.
`x_axis`	Character: which sweep variable to place on the x-axis. One of `"sigma"`, `"s"`, or `"p0"`.
`facet_by`	Character: column name to facet by. Default `"alpha"`.
`...`	Additional arguments (currently ignored).

Value

Invisibly returns a `ggplot` object representing the sensitivity plot. Called primarily for its side effect of rendering a line-and-ribbon plot to the active graphic device.

See Also

`pci_grid`

Examples

```
if (requireNamespace("ggplot2", quietly = TRUE)) {
  grid <- pci_grid("Cp", LSL = 57, USL = 63,
                   sigma_vals = seq(0.5, 2, 0.1), mu = 60,
                   alpha_vals = c(0.05, 0.01),
                   n = 30, B = 200,
                   dist = pci_dist_normal(60, 1))
  plot(grid, x_axis = "sigma")
}
```

pp *Performance Index (Pp)*

Description

$P_p = (USL - LSL)/(6s)$.

Usage

```
pp(LSL, USL, s)
```

Arguments

LSL	Lower specification limit.
USL	Upper specification limit.
s	Long-term (overall) standard deviation.

Value

Numeric value of P_p.

See Also

pci

Examples

```
pp(LSL = 57, USL = 63, s = 1.5)   # 0.6667
```

ppk *Performance Capability Index (Ppk)*

Description

$P_{pk} = \min\big((USL - \bar{x})/(3s),\ (\bar{x} - LSL)/(3s)\big)$.

Usage

```
ppk(LSL, USL, xbar, s)
```

Arguments

LSL	Lower specification limit.
USL	Upper specification limit.
xbar	Long-term (overall) mean.
s	Long-term standard deviation.

Value

Numeric value of P_{pk}.

See Also

pci

Examples

```
ppk(LSL = 57, USL = 63, xbar = 60, s = 1.5)   # 0.6667
```

---

ppl *Lower Performance Index (Ppl)*

Description

$P_{pl} = (\bar{x} - LSL)/(3s)$.

Usage

```
ppl(LSL, xbar, s)
```

Arguments

LSL	Lower specification limit.
xbar	Long-term mean.
s	Long-term standard deviation.

Value

Numeric value of P_{pl}.

See Also

pci

Examples

```
ppl(LSL = 57, xbar = 60, s = 1.5)   # 0.6667
```

ppu *Upper Performance Index (Ppu)*

Description

$P_{pu} = (USL - \bar{x})/(3s)$.

Usage

```
ppu(USL, xbar, s)
```

Arguments

USL	Upper specification limit.
xbar	Long-term mean.
s	Long-term standard deviation.

Value

Numeric value of P_{pu}.

See Also

pci

Examples

```
ppu(USL = 63, xbar = 60, s = 1.5)   # 0.6667
```

print.pci_ci *Print method for pci_ci objects*

Description

Print method for pci_ci objects

Usage

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

Arguments

x	A pci_ci object.
...	Additional arguments (ignored).

Value

Invisibly returns the input `x` of class `"pci_ci"`. Called for side effect of printing formatted confidence interval details to the console.

---

print.pci_dist *Print method for pci_dist objects*

Description

Print method for pci_dist objects

Usage

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

Arguments

`x`	A `pci_dist` object.
`...`	Additional arguments (ignored).

Value

Invisibly returns the input `x` of class `"pci_dist"`. Called for side effect of printing distribution parameters and specification details to the console.

---

z_level *Sigma Level (Z)*

Description

$Z = \min\big((USL - \mu)/\sigma,\ (\mu - LSL)/\sigma\big)$.

Usage

```
z_level(LSL, USL, mu, sigma)
```

Arguments

`LSL`	Lower specification limit.
`USL`	Upper specification limit.
`mu`	Short-term process mean.
`sigma`	Short-term standard deviation.

Value

Numeric value of Z.

See Also

pci

Examples

```
z_level(LSL = 57, USL = 63, mu = 60, sigma = 1)   # 3.0
```

Index