Package ‘PPQplan’

October 12, 2022

Type Package

Title Process Performance Qualification (PPQ) Plans in Chemistry, Manufacturing and Controls (CMC) Statistical Analysis

Version 1.1.0

Maintainer Yalin Zhu <yalin.zhu@merck.com>

Depends R (>= 3.2.0)

Imports ggplot2, plotly


License GPL-3

Suggests knitr, rmarkdown, devtools

VignetteBuilder knitr

NeedsCompilation no

RoxygenNote 7.1.0

Encoding UTF-8


BugReports https://github.com/allenzhuaz/PPQplan/issues

LazyData true

Language en-US

Author Yalin Zhu [aut, cre] (<https://orcid.org/0000-0003-3830-8660>), Merck & Co., Inc. [cph]

Repository CRAN

Date/Publication 2020-10-08 04:30:06 UTC

Description

The function for dynamically plotting (ggplot) the heatmap to evaluate the sampling plan based on a general lower and/or upper specification limits.

Usage

heatmap_ly(attr.name, attr.unit, Llim, Ulim, mu, sigma, n, test.point, dynamic)

Arguments

attr.name (optional) user-defined attribute name for sampling plan assessment
attr.unit (optional) user-defined attribute unit
Llim lower specification limit
Ulim upper specification limit
mu hypothetical mean of the attribute
sigma hypothetical standard deviation of the attribute
n sample size (number of locations) per batch
test.point (optional) actual process data points for testing whether the processes pass PPQ
dynamic logical; if TRUE, then convert the plain heatmap to dynamic graph using plotly.

Value

A Plain or Dynamic Heatmap for Sampling Plan Assessment.
### k_factor

**Author(s)**

Yalin Zhu

**References**


**See Also**

pp and PPQ.occurve.

**Examples**

```r
## Not run:
heatmap_ly(attr.name = "Thickness", attr.unit = "%", llim = -0.2, ulim = 0.2, 
mu = seq(-0.2, 0.2, 0.001), sigma = seq(0,0.2, 0.001),
test.point=data.frame(c(0.1,-0.05),c(0.15,0.05)), n=2, dynamic = T)
## End(Not run)
```

---

### k_factor

**Estimating K-factors for Tolerance Intervals Based on Howe’s Method**

**Description**

Estimates k-factors for tolerance intervals based on Howe’s method with normality assumption.

**Usage**

```r
k_factor(n, alpha = 0.05, P = 0.99, side = 1)
```

**Arguments**

- **n**: Sample size
- **alpha**: The level chosen such that (1-alpha) is the confidence level.
- **P**: The proportion of the population to be covered by the tolerance interval.
- **side**: Whether a 1-sided or 2-sided tolerance interval is required (determined by side = 1 or side = 2, respectively).

**Value**

The estimated k-factor for tolerance intervals assuming normality.

**Note**

This function is a simplified version of `tolerance::K.factor()`, only considering Howe’s method.
pi_ctplot

See Also

ti_pp

Examples

k_factor(10, P = 0.95, side = 2)


Description

The function for plotting the heatmap to evaluate the PPQ plan based on the specification test, given lower and upper specification limits.

Usage

pi_ctplot(attr.name, attr.unit, Llim, Ulim, mu, sigma, n, n.batch, alpha, test.point)

Arguments

attr.name  user-defined attribute name for PPQ assessment
attr.unit  user-defined attribute unit
Llim lower specification limit
Ulim upper specification limit
mu hypothesised mean of the attribute
sigma hypothesical standard deviation of the attribute
n sample size (number of locations) per batch
n.batch number of batches for passing PPQ during validation
alpha significant level for constructing the prediction interval.
test.point (optional) actual process data points for testing whether the processes pass PPQ

Value

Heatmap (or Contour Plot) for PPQ Assessment.

Author(s)

Yalin Zhu
pi_occurve

Operating Characteristic (OC) Curves for the CQA PPQ Plan using Prediction Interval.

Description

The function for plotting the OC curves and optimizing the baseline and high performance PPQ plans, given lower and upper specification limits.

Usage

pi_occurve(attr.name, attr.unit, Llim, Ulim, mu, sigma, n, n.batch, alpha, add.reference)
Arguments

- attr.name: user-defined attribute name
- attr.unit: user-defined attribute unit
- Llim: lower specification limit
- Ulim: upper specification limit
- mu: hypothetical mean of the attribute
- sigma: hypothetical standard deviation of the attribute
- n: sample size (number of locations) per batch
- n.batch: number of batches for passing PPQ during validation
- alpha: significant level for constructing the prediction interval.
- add.reference: logical; if TRUE, then add reference OC curves (Baseline and High Performance) in the plot.

Value

OC curves for specification test and PPQ plan.

Author(s)

Yalin Zhu

References


See Also

pi_pp and rl_pp.

Examples

```r
## Not run:
pi_occurve(attr.name = "Total Protein", attr.unit = "mg/mL",
           sigma = seq(0.01,1,0.01))
pi_occurve(attr.name = "Total Protein", attr.unit = "mg/mL",
           sigma = seq(0.01,1,0.01), n.batch=3)  # Baseline curve
pi_occurve(attr.name = "Total Protein", attr.unit = "mg/mL",
           sigma = seq(0.01,1,0.01), alpha = 0.1135434)  # High performance curve
pi_occurve(attr.name = "Total Protein", attr.unit = "mg/mL",
           sigma = seq(0.01,1,0.01), alpha = 0.0225518)
# 95% with reference curves
pi_occurve(attr.name = "Total Protein", attr.unit = "mg/mL",
           add.reference = TRUE)
```
pi_pp

**Probability of Passing PPQ Test using Prediction Interval**

**Description**

The function for calculating the probability of passing critical quality attributes (CQA) PPQ test.

**Usage**

```r
pi_pp(Llim, Ulim, mu, sigma, n, n.batch, alpha)
```

**Arguments**

- `Llim` lower specification limit
- `Ulim` upper specification limit
- `mu` hypothetical mean of the attribute
- `sigma` hypothetical standard deviation of the attribute
- `n` sample size (number of locations) per batch
- `n.batch` number of batches for passing PPQ during validation
- `alpha` significant level for constructing the prediction interval.

**Value**

A numeric value of the passing/acceptance probability

**Author(s)**

Yalin Zhu
References


See Also

rl_pp.

Examples

```r
## Not run:
pi_pp(sigma=0.5, mu=2.5, n=10, n.batch=1, Llim=1.5, Ulim=3.5, alpha=0.05)

sapply(X=c(0.1,0.5, 1,2,3,4,5,10), FUN = pi_pp, mu=97, n=10, Llim=95, Ulim=105, n.batch=1, alpha=0.05)
sapply(X=c(0.1,0.5, 1,2,3,4,5,10), FUN = pi_pp, mu=100, n=10, Llim=95, Ulim=105, n.batch=1, alpha=0.05)
## End(Not run)
```

---

### pp

*Probability of Passing General Upper and/or Lower Specification Limit*

**Description**

The function for calculating the probability of passing a general upper and/or lower boundary.

**Usage**

```r
pp(Llim, Ulim, mu, sigma, n)
```

**Arguments**

- `Llim`: lower specification limit
- `Ulim`: upper specification limit
- `mu`: hypothetical mean of the attribute
- `sigma`: hypothetical standard deviation of the attribute
- `n`: sample size (number of locations)

**Value**

A numeric value of the passing/acceptance probability

**Author(s)**

Yalin Zhu
See Also

r1_pp and PPQ_pp.

---

**PPQ_ctplot**

*Heatmap/Contour Plot for Assessing Power of the CQA PPQ Plan Using General Multiplier.*

**Description**

The function for plotting the heatmap to evaluate the PPQ plan based on the specification test, given lower and upper specification limits.

**Usage**

```r
PPQ_ctplot(attr.name, attr.unit, Llim, Ulim, mu, sigma, n, n.batch, k, test.point)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>attr.name</td>
<td>(optional) user-defined attribute name for PPQ assessment</td>
</tr>
<tr>
<td>attr.unit</td>
<td>(optional) user-defined attribute unit</td>
</tr>
<tr>
<td>Llim</td>
<td>lower specification limit</td>
</tr>
<tr>
<td>Ulim</td>
<td>upper specification limit</td>
</tr>
<tr>
<td>mu</td>
<td>hypothetical mean of the attribute</td>
</tr>
<tr>
<td>sigma</td>
<td>hypothetical standard deviation of the attribute</td>
</tr>
<tr>
<td>n</td>
<td>sample size (number of locations) per batch</td>
</tr>
<tr>
<td>n.batch</td>
<td>number of batches for passing PPQ during validation</td>
</tr>
<tr>
<td>k</td>
<td>general multiplier for constructing the specific interval</td>
</tr>
<tr>
<td>test.point</td>
<td>(optional) actual process data points for testing whether the processes pass PPQ</td>
</tr>
</tbody>
</table>

**Value**

Heatmap (or Contour Plot) for PPQ Assessment.

**Author(s)**

Yalin Zhu

**References**

See Also

PPQ_pp and PPQ_occurve.

Examples

## Not run:
mu <- seq(1.6,3.4,0.05)
sigma <- seq(0.05,0.8,0.01)
PPQ_ctplot(attr.name = "Total Protein", attr.unit = "mg/mL", Llim=1.5, Ulim=3.5,
mu = mu, sigma = sigma, k=2.373)

## Example verifying simulation results in the textbook page 249
mu <- seq(95, 105, 0.1)
sigma <- seq(0.2, 5, 0.1)
PPQ_ctplot(attr.name = "Composite Assay", attr.unit = "%LC", Llim=95, Ulim=105,
mu = mu, sigma = sigma, k=2.373)
u <- seq(90, 110, 0.5)
PPQ_ctplot(attr.name = "Composite Assay", attr.unit = "%LC", Llim=90, Ulim=110,
mu = mu, sigma = sigma, k=2.373)
mu <- seq(95,105,0.1)
sigma <- seq(0.1,2.5,0.1)
PPQ_ctplot(attr.name = "Sterile Concentration Assay", attr.unit = ", Llim=95, Ulim=105,
mu = mu, sigma = sigma, k=2.373)
test <- data.frame(mean=c(97,98.3,102.5), sd=c(0.55, 1.5, 1.2))
PPQ_ctplot(attr.name = "Sterile Concentration Assay", attr.unit = ", Llim=95, Ulim=105,
mu = mu, sigma = sigma, k=2.373, test.point=test)

## End(Not run)

---

### PPQ_ggplot

**Heatmap/Contour Plot for Dynamically Assessing Power of the CQA PPQ Plan Using General Multiplier.**

#### Description

The function for dynamically plotting (ggplot) the heatmap to evaluate the PPQ plan based on the specification test, given lower and upper specification limits.

#### Usage

```r
PPQ_ggplot(attr.name, attr.unit, Llim, Ulim, mu, sigma, n, n.batch, k,
test.point, dynamic)
```

#### Arguments

attr.name: (optional) user-defined attribute name for PPQ assessment
attr.unit: (optional) user-defined attribute unit
**PPQ_ggplot**

- `Llim` lower specification limit
- `Ulim` upper specification limit
- `mu` hypothetical mean of the attribute
- `sigma` hypothetical standard deviation of the attribute
- `n` sample size (number of locations) per batch
- `n.batch` number of batches for passing PPQ during validation
- `k` general multiplier for constructing the specific interval
- `test.point` (optional) actual process data points for testing whether the processes pass PPQ
- `dynamic` logical; if TRUE, then convert the heatmap ggplot to dynamic graph using plotly.

**Value**

Dynamic Heatmap (or Contour Plot) for PPQ Assessment.

**Author(s)**

Yalin Zhu

**References**


**See Also**

`PPQ_pp` and `PPQ_occurve`.

**Examples**

```r
## Not run:
mu <- seq(95, 105, 0.1)
sigma <- seq(0.1,1.7,0.1)
PPQ_ggplot(attr.name = "Sterile Concentration Assay", attr.unit = "%", Llim=95, Ulim=105, 
mu = mu, sigma = sigma, k=2.373, dynamic = FALSE)
test <- data.frame(mu=c(97,98.3,102.5), sd=c(0.55, 1.5, 0.2))
PPQ_ggplot(attr.name = "Sterile Concentration Assay", attr.unit = "%", Llim=95, Ulim=105, 
mu = mu, sigma = sigma, k=2.373, test.point = test)

## End(Not run)
```
Description

The function for plotting the OC curve to show the PPQ plan, given lower and upper specification limits.

Usage

PPQ_occurve(attr.name, attr.unit, Llim, Ulim, mu, sigma, n, n.batch, k, add.reference)

Arguments

attr.name (optional) user-defined attribute name
attr.unit (optional) user-defined attribute unit
Llim lower specification limit
Ulim upper specification limit
mu hypothetical mean of the attribute
sigma hypothetical standard deviation of the attribute
n sample size (number of locations) per batch
n.batch number of batches for passing PPQ during validation
k general multiplier for constructing the specific interval
add.reference logical; if TRUE, then add reference OC curves (Baseline and High Performance) in the plot.

Value

OC curves for specification test and PPQ plan.

Author(s)

Yalin Zhu

References


See Also

PPQ_pp and r1_pp.
Examples

```r
# Not run:
PPQ_occurve(attr.name = "Sterile Concentration Assay", attr.unit="%", Llim=95, Ulim=105,
mu=97, sigma=seq(0.1, 10, 0.1), n=10, k=2.373, add.reference=TRUE)
PPQ_occurve(attr.name = "Sterile Concentration Assay", attr.unit="%", Llim=95, Ulim=105,
mu=100, sigma=seq(0.1, 10, 0.1), n=10, k=2.373, add.reference=TRUE)
PPQ_occurve(attr.name = "Sterile Concentration Assay", attr.unit="%", Llim=95, Ulim=105,
mu=seq(95,105,0.1), sigma=1, n=10, k=2.373)
PPQ_occurve(attr.name = "Sterile Concentration Assay", attr.unit="%", Llim=95, Ulim=105,
mu=seq(95,105,0.1), sigma=1, n=10, k=2.373, add.reference=TRUE)
PPQ_occurve(attr.name = "Protein Concentration", attr.unit="%", Llim=90, Ulim=110,
mu=seq(90, 110, 0.1), sigma=1.25, k=2.373)
PPQ_occurve(attr.name = "Sterile Concentration Assay", attr.unit="%", Llim=95, Ulim=105,
mu=98, sigma=seq(0.1, 10, 0.1), n=10, k=2.373, add.reference=TRUE)
# Only display reference curves, leave k as NULL by default
PPQ_occurve(attr.name = "Sterile Concentration Assay", attr.unit="%", Llim=95, Ulim=105,
mu=98, sigma=seq(0.1, 10, 0.1), n=10, add.reference=TRUE)
# End(Not run)
```

PPQ_pp

**Probability of Passing PPQ Test Using General Multiplier**

**Description**

The function for calculating the probability of passing critical quality attributes (CQA) PPQ test.

**Usage**

```r
PPQ_pp(Llim, Ulim, mu, sigma, n, n.batch, k)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Llim</td>
<td>lower specification limit</td>
</tr>
<tr>
<td>Ulim</td>
<td>upper specification limit</td>
</tr>
<tr>
<td>mu</td>
<td>hypothetical mean of the attribute</td>
</tr>
<tr>
<td>sigma</td>
<td>hypothetical standard deviation of the attribute</td>
</tr>
<tr>
<td>n</td>
<td>sample size (number of locations) per batch</td>
</tr>
<tr>
<td>n.batch</td>
<td>number of batches for passing PPQ during validation</td>
</tr>
<tr>
<td>k</td>
<td>general multiplier for constructing the specific interval</td>
</tr>
</tbody>
</table>

**Value**

A numeric value of the passing/acceptance probability

**Author(s)**

Yalin Zhu
References

See Also
rl_pp.

Examples

```r
## Not run:
PPQ_pp(Llim = 90, Ulim = 110, mu=105, sigma=1.5, n=10, k=3.1034)

# One-sided tolerance interval with k=0.753 (95/67.5 one-sided tolerance interval LTL)
PPQ_pp(sigma=0.03, mu=.025, n=40, Llim=1, Ulim=Inf, k=0.753)

sapply(X=c(0.1,0.5, 1,2,3,4,5,10), FUN = PPQ_pp, mu=97, n=10, Llim=95, Ulim=105, k=2.373)

sapply(X=seq(0.1,10,0.1), FUN = PPQ_pp, mu=97, n=10, Llim=95, Ulim=105, k=2.373)

sapply(X=c(0.1,0.5, 1,2,3,4,5,10), FUN = PPQ_pp, mu=100, n=10, Llim=95, Ulim=105, k=2.373)

gamma <- seq(0.1, 4, 0.1)
pp1 <- sapply(X=gamma, FUN = PPQ_pp, mu=97, n=10, Llim=95, Ulim=105, k=2.373)

pp2 <- sapply(X=gamma, FUN = PPQ_pp, mu=98, n=10, Llim=95, Ulim=105, k=2.373)

pp3 <- sapply(X=gamma, FUN = PPQ_pp, mu=99, n=10, Llim=95, Ulim=105, k=2.373)

pp4 <- sapply(X=gamma, FUN = PPQ_pp, mu=100, n=10, Llim=95, Ulim=105, k=2.373)

plot(gamma, pp1, xlab="Standard Deviation", main="LSL=95, USL=105, k=2.373, n=10",
ylab="Probability of Passing", type="o", pch=1, col=1, lwd=1, ylim=c(0,1))

lines(gamma, pp2, type="o", pch=2, col=2)

lines(gamma, pp3, type="o", pch=3, col=3)

lines(gamma, pp4, type="o", pch=4, col=4)

legend("topright", legend=paste0(rep("mu=",4),c(97,98,99,100)), bg="white",
col=c(1,2,3,4), pch=c(1,2,3,4), lty=1, cex=0.8)

mu <- seq(95, 105, 0.1)

pp5 <- sapply(X=mu, FUN = PPQ_pp, sigma=0.5, n=10, Llim=95, Ulim=105, k=2.373)

pp6 <- sapply(X=mu, FUN = PPQ_pp, sigma=1, n=10, Llim=95, Ulim=105, k=2.373)

pp7 <- sapply(X=mu, FUN = PPQ_pp, sigma=1.5, n=10, Llim=95, Ulim=105, k=2.373)

pp8 <- sapply(X=mu, FUN = PPQ_pp, sigma=2, n=10, Llim=95, Ulim=105, k=2.373)

pp9 <- sapply(X=mu, FUN = PPQ_pp, sigma=2.5, n=10, Llim=95, Ulim=105, k=2.373)

plot(mu, pp5, xlab="Mean Value", main="LSL=95, USL=105, k=2.373, n=10",
ylab="Probability of Passing", type="o", pch=1, col=1, lwd=1, ylim=c(0,1))

lines(mu, pp6, type="o", pch=2, col=2)

lines(mu, pp7, type="o", pch=3, col=3)

lines(mu, pp8, type="o", pch=4, col=4)

lines(mu, pp9, type="o", pch=5, col=5)

legend("topright", legend=paste0(rep("sigma=",5),seq(0.5,2.5,0.5)), bg="white",
col=c(1,2,3,4,5), pch=c(1,2,3,4,5), lty=1, cex=0.8)

## End(Not run)
```
**Description**

The function for calculating the probability of passing critical quality attributes (CQA) specification test.

**Usage**

\[ \text{rl\_pp}(L\text{lim}, U\text{lim}, \mu, \sigma, NV) \]

**Arguments**

- **Llim**: lower specification limit
- **Ulim**: upper specification limit
- **mu**: hypothetical mean of the attribute
- **sigma**: hypothetical standard deviation of the attribute
- **NV**: nominal volume for the specification test.

**Value**

A numeric value of the passing/acceptance probability

**Author(s)**

Yalin Zhu

**References**


**See Also**

PPQ\_pp, pi\_pp and ti\_pp.

**Examples**

\[ \text{rl\_pp}(L\text{lim}=1.5, U\text{lim}=3.5, \mu=2.5, \sigma=0.8) \]
ti_ctplot

Heatmap/Contour Plot for Assessing Power of the PPQ Plan using Tolerance Interval.

Description
The function for plotting the heatmap to evaluate the PPQ plan based on the specification test, given lower and upper specification limits.

Usage
ti_ctplot(attr.name, attr.unit, Llim, Ulim, mu, sigma, n, n.batch, alpha, coverprob, side, test.point)

Arguments
attr.name user-defined attribute name for PPQ assessment
attr.unit user-defined attribute unit
Llim lower specification limit
Ulim upper specification limit
mu hypothetical mean of the attribute
sigma hypothetical standard deviation of the attribute
n sample size (number of locations) per batch
n.batch number of batches for passing PPQ during validation
alpha significant level for constructing the tolerance interval.
coverprob coverage probability for constructing the tolerance interval
side whether a 1-sided or 2-sided tolerance interval is required (determined by side = 1 or side = 2, respectively).
test.point (optional) actual process data points for testing whether the processes pass PPQ

Value
Heatmap (or Contour Plot) for PPQ Assessment.

Author(s)
Yalin Zhu

References
See Also

ti_pp and ti_occurve.

Examples

```r
## Not run:
mu <- seq(95,105,0.1)
sigma <- seq(0.1,2.5,0.1)
ti_ctplot(attr.name = "Sterile Concentration Assay", attr.unit = "%",
mu = mu, sigma = sigma, Llim=95, Ulim=105)

ni_ctplot(attr.name = "Extractable Volume", attr.unit = "% of NV=1mL",
Llim = 100, Ulim = Inf, mu=seq(100, 110, 0.5), sigma=seq(0.2, 15 ,0.5), n=40,
alpha = 0.05, coverprob = 0.675, side=1)

## End(Not run)
```

---

**ti_occurve**

*Operating Characteristic (OC) Curves for the PPQ Plan using Tolerance Interval.*

**Description**

The function for plotting the OC curve to show the PPQ plan based on the specification test, given lower and upper specification limits.

**Usage**

```r
ti_occurve(attr.name, attr.unit, Llim, Ulim, mu, sigma, n, n.batch, alpha,
coverprob, side, add.reference, NV)
```

**Arguments**

- `attr.name`: user-defined attribute name
- `attr.unit`: user-defined attribute unit
- `Llim`: lower specification limit
- `Ulim`: upper specification limit
- `mu`: hypothetical mean of the attribute
- `sigma`: hypothetical standard deviation of the attribute
- `n`: sample size (number of locations) per batch
- `n.batch`: number of batches for passing PPQ during validation
- `alpha`: significant level for constructing the tolerance interval.
- `coverprob`: coverage probability for constructing the tolerance interval
- `side`: whether a 1-sided or 2-sided tolerance interval is required (determined by side = 1 or side = 2, respectively).
add.reference logical; if TRUE, then add reference OC curves (Baseline and High Performance) in the plot.

NV nominal volume for the specification test.

Value

OC curves for specification test and PPQ plan.

Author(s)

Yalin Zhu

References


See Also

ti_pp and rl_pp.

Examples

```r
## Not run:
ti_occurve(attr.name = "Sterile Concentration Assay", attr.unit="%",
mu=97, sigma=seq(0.1, 10, 0.1), Llim=95, Ulim=105, n=10, add.reference=TRUE)
ti_occurve(attr.name = "Sterile Concentration Assay", attr.unit="%",
mu=100, sigma=seq(0.1, 10, 0.1), Llim=95, Ulim=105, n=10, add.reference=TRUE)
ti_occurve(attr.name = "Extractable Volume", attr.unit = "% of NV=3mL",
Llim = 100, Ulim = Inf, mu=102.5, sigma=seq(0.2, 6 ,0.05), n=40,
alpha = 0.05, coverprob = 0.97, side=1, NV=3)
ti_occurve(attr.name = "Extractable Volume", attr.unit = "% of NV=3mL",
Llim = 100, Ulim = Inf, mu=102.5, sigma=seq(0.2, 6 ,0.05), n=40,
alpha = 0.05, coverprob = 0.992, side=1, NV=3)
## End(Not run)
```

## Not run:

### ti_pp

**Probability of Passing PPQ Test using Tolerance Interval**

**Description**

The function for calculating the probability of passing critical quality attributes (CQA) PPQ test.
Usage

ti_pp(Llim, Ulim, mu, sigma, n, n.batch, alpha, coverprob, side)

Arguments

Llim  lower specification limit
Ulim  upper specification limit
mu    hypothetical mean of the attribute
sigma hypothetical standard deviation of the attribute
n     sample size (number of locations) per batch
n.batch number of batches for passing PPQ during validation
alpha significant level for constructing the tolerance interval
coverprob coverage probability for constructing the tolerance interval
side  whether a 1-sided or 2-sided tolerance interval is required (determined by side = 1 or side = 2, respectively).

Value

A numeric value of the passing/acceptance probability

Author(s)

Yalin Zhu

References


See Also

rl_pp.

Examples

ti_pp(sigma=0.5, mu=2.5, n=10, n.batch=1, Llim=1.5, Ulim=3.5, alpha=0.05)

sapply(X=c(0.1,0.5, 1,2,3,4,5,10), FUN = ti_pp, mu=97, n=10, Llim=95, Ulim=105, n.batch=1, alpha=0.05)

sapply(X=c(0.1,0.5, 1,2,3,4,5,10), FUN = ti_pp, mu=100, n=10, Llim=95, Ulim=105, n.batch=1, alpha=0.05)
Index

heatmap_ly, 2
k_factor, 3
pi_ctplot, 4
pi_occurve, 5
pi_pp, 7
pp, 8
PPQ_ctplot, 9
PPQ_ggplot, 10
PPQ_occurve, 12
PPQ_pp, 13
rl_pp, 15

ti_ctplot, 16
ti_occurve, 17
ti_pp, 18