

State-of-the-Art with Statistical Tolerance Regions: Methods and Applications, with Computing Demonstrations Using the R Package *tolerance*

Kedai Cheng (University of North Carolina - Asheville)

Derek S. Young (University of Kentucky)

Conference on Statistical Practice (CSP) 2022

February 3rd, 2022

Funding Acknowledgment

I obtained a Ph.D. at 2020 from the University of Kentucky.

My dissertation specialities are tolerance intervals and winsorization methods.

I want to thank Drs. Stromberg, Rayens, Harrar, Thompson for their teachings and patience on me.

I want to give a special and big Thank You to my advisor Dr. Derek Young for his tolerance on me and priceless opportunities.

Funding Acknowledgment

 This work is supported by the Chan Zuckerberg Initiative: Essential Open Source Software for Science (Grant Number 2020-225193).

Before We Begin...

- ▶ You can download this presentation file and the corresponding R scripts from Derek Young's GitHub repo:
<https://github.com/dsy109/Supplemental/tree/main/CSP%202022%20Trainings/tolerance>
- ▶ If you plan to run the R examples concurrently, make sure you have the most recent versions installed for all of the packages listed at the top of the .R file
- ▶ This tutorial was developed and tested using R version 4.1.1

Outline of Topics

Purpose of Tutorial



tolerance

Introduction

Normal Tolerance Intervals

Some Non-Normal Tolerance Intervals

Nonparametric Tolerance Intervals

Regression Tolerance Intervals

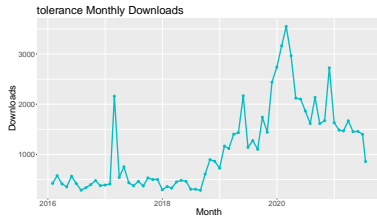
Multivariate Tolerance Regions

Final Comments



tolerance

- ▶ First release: 2009; see [Young \(2010\)](#) for technical details.
- ▶ The package includes: tolerance interval procedures for numerous parametric distributions, nonparametric settings, regression models, and some multivariate settings, as well as visualizations
- ▶ Used by NASA, 3M, EcoLab, PepsiCo, and NIST, among others
- ▶ The figure below also shows a mostly increasing trend over the years as the average number of monthly downloads per year is approximately 418, 592, 557, 1717, and 2203, in 2016, 2017, 2018, 2019, and 2020, respectively





Motivation: Tolerance Regions

- ▶ Based on a random sample, we have three primary statistical regions that can be calculated:
 - ▶ **Confidence regions** → provide regions for an unknown population parameter (e.g., mean vector, variance-covariance matrix)
 - ▶ **Prediction regions** → provide regions for one or more future observations from the sampled population
 - ▶ **Tolerance regions** → provide regions that are expected to contain at least a specified proportion of the sampled population
- ▶ Typical applications of tolerance regions (or **tolerance intervals** for the univariate setting) are found in clinical and industrial studies, statistical quality control, environmental monitoring, and setting statistically-based engineering design limits
- ▶ Used in regulations published by the Environmental Protection Agency (EPA), the Food and Drug Administration (FDA), the International Atomic Energy Agency (IAEA), and standard 16269-6 of the International Organization for Standardization (ISO)

Statistical Tolerance Sets

Definition

Let \mathcal{F} be the class of all Borel measurable distributions in \mathbb{R}^p , $p \in \mathbb{N}$. Let $\mathcal{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_n\}$ be an *iid* random sample of size $n > p$ drawn from $F \in \mathcal{F}$, and \mathbf{X} be a random vector that also follows F , independently of \mathcal{X} . Let $T(\mathcal{X})$ be a random subset of \mathbb{R} and define $C_F(T(\mathcal{X}))$ to be the probability content under F of the set $T(\mathcal{X})$. If

$$\inf_{F \in \mathcal{F}} \Pr \{C_F(T(\mathcal{X})) \geq P\} = 1 - \alpha,$$

then $T(\mathcal{X})$ is a $(1 - \alpha, P)$ **tolerance set**, where $P, \alpha \in (0, 1)$ are the (probability) content level and confidence level, respectively. If

$$E[C_F(T(\mathcal{X}))] = \beta,$$

then $T(\mathcal{X})$ is a β -**expectation tolerance set**, where $\beta \in (0, 1)$ is the desired average content level of the tolerance set.

- β -expectation tolerance sets are just prediction sets

Normal Tolerance Intervals

Normal Tolerance Interval

- ① Introduction
- ② K-factors: `[K.factor, K.table, K.factor.sim]`
- ③ Tolerance Limits for Normally Distributed Data: `[normtol.int]`
- ④ Sample Size Determination: `[norm.ss]`
- ⑤ Operating Characteristic (OC) Curves: `[norm.OC]`
- ⑥ Application

Normal Tolerance Interval: Introduction

Suppose we have a random sample, X , from a normally distributed population. The one-sided tolerance interval is

$$(\bar{X} - k_1 s, \bar{X} + k_1 s)$$

The two-sided lower and upper tolerance interval is

$$(\bar{X} - k_2 s, \bar{X} + k_2 s)$$

respectively. Here, \bar{X} is the sample mean, s is the sample standard error, and k_1 and k_2 indicates k -factors for one-sides and two-sided tolerance intervals, respectively. Derivation of normal tolerance intervals depend on ***k*-factors**.

Methodology

There is a great deal of literature on normal tolerance intervals. `tolerance` package covers a range of these methods.

Before the wide availability of computers, approximation methods were deeply studied by various researchers, and approximation methods were developed and improved for different cases. Exact methods were later developed with the support of computer developments. `tolerance` covers exact methods and numerous approximations.

Methodology

① EXACT:

① One-Sided:

The k -factor ensures that we capture at least a proportion P of the sampled population with confidence level $(1 - \alpha)$ is calculated as

$$k_1 = \frac{1}{\sqrt{n}} t_{n-1; 1-\alpha}(\sqrt{n} Z_P)$$

where $t_{f;q}$ is the q^{th} quantile of a noncentral t -distribution with degree of freedom f and noncentrality parameter δ and z_q is the q^{th} quantile of the standard normal distribution.

② Two-Sided:

There are no closed-form solutions for exact method for two-sided tolerance limits under normality, so one must use numerical methods.

Methodology

For a two-sided (P, γ) tolerance limits, the k -factor is the solution to The k -factor for two-sided (P, γ) tolerance limits are is

$$\sqrt{\frac{2n}{\pi}} \int_0^\infty P\left(\chi^2_{n-1} > \frac{(n-1)\chi^2_{1;P}(z^2)}{k_2(n, \alpha, P)^2}\right) e^{-\frac{1}{2}nz^2} dz = 1 - \alpha$$

Multiple approximation methods were proposed. (Note that approximation method can also be applied for one-sided tolerance limits.)

- ① WBE: [Weissberg and Beatty \(1960\)](#) used *Newton's Method* to approximate r and *Cornish-Fisher* approximation to estimated χ^2 distribution, based on ([Wald & Wolfowitz, 1946](#)).
- ② HE2: is a method based on [Weissberg and Beatty \(1960\)](#), but computationally simpler.
- ③ ELL: [Ellison \(1964\)](#) gives a correction of *Weissberg-Beatty's* method when f is not larger than N^2 .
- ④ KM: [Krishnamoorthy and Mathew \(2009\)](#) provide approximation solution to the exact method.
- ⑤ HE: is a method developed by [Howe \(1969\)](#) and is often viewed as being extremely accurate, even for small sample sizes.
- ⑥ OCT: [Owen \(1964\)](#)'s method gives more flexibility and is capable of controlling different tails while capturing a specified proportion P at a given confidence level $(1 - \alpha) \times 100$.

Normal Tolerance Interval: k -factors

To find the k -factor of a (P, γ) tolerance tolerance interval for a given sample with size of n , we need to determine the following quantities:

- ① Sample size, n
- ② Content Level, P
- ③ Confidence Level, γ
- ④ One-sided or Two-sided
- ⑤ Method
- ⑥ Degree of Freedom
- ⑦ The maximum number of subintervals to be used for possible intergration

Normal Tolerance Interval: k -factors

We have three functions for a quick reference of k -factors.

- ① `K.factor`: k -factor for a certain condition
- ② `K.table`: k -factors for multiple conditions
- ③ `K.factor.sim`: Estimating k -factors for simultaneous tolerance intervals based on normality

Normal Tolerance Interval: K.factor

```
K.factor(n, P, alpha, side=c(1,2), method= c("HE", "HE2", "WBE",  
"ELL", "KM", "EXACT", "OCT"), f, m)
```

Description:

The k -factor is a critical component for a tolerance interval and it plays the role of 'z-score' in a confidence intervals and determines the width of a tolerance interval.

K.factor estimates the k -factors for tolerance intervals based on normality.

Normal Tolerance Interval: K.factor

Examples: by method

- ❶ `K.factor(n = 10, P = 0.95, side = 2, method = "HE")`
- ❷ `K.factor(n = 10, P = 0.95, side = 2, method = "WBE")`
- ❸ `K.factor(n = 10, P = 0.95, side = 2, method = "EXACT", m = 50)`
- ❹ `K.factor(n = 100, P = 0.95, side = 2, method = "HE")`
- ❺ `K.factor(n = 100, P = 0.95, side = 2, method = "WBE")`
- ❻ `K.factor(n = 100, P = 0.95, side = 2, method = "EXACT", m = 50)`

Normal Tolerance Interval: K.factor

```
> K.factor(10, P = 0.95, side = 2, method = "HE")  
[1] 3.407495  
> K.factor(10, P = 0.95, side = 2, method = "WBE")  
[1] 3.379442  
> K.factor(10, P = 0.95, side = 2, method = "EXACT", m = 50)  
[1] 3.393429  
> K.factor(100, P = 0.95, side = 2, method = "HE")  
[1] 2.233895  
> K.factor(100, P = 0.95, side = 2, method = "WBE")  
[1] 2.232787  
> K.factor(100, P = 0.95, side = 2, method = "EXACT", m = 50)  
[1] 2.233882
```

Normal Tolerance Interval: K.table

```
K.table(n, P, alpha, side = c(1,2), method = c("HE", "HE2", "WBE",  
"ELL", "KM", "EXACT", "OCT"), by.arg = c("n", "alpha", "P"), f, m)
```

Description:

K.table tabulates k -factors for tolerance intervals based on normality under different conditions. Users can specify the presentation of the table with respect to the order of sample size (n), content level (P), or confidence level (α).

Normal Tolerance Interval: K.table

Examples:

① Ordered by sample size:

```
K.table(n = seq(50, 60, 10), alpha = c(0.01, 0.05, 0.10), P =  
c(0.90, 0.95, 0.99), by.arg = "n")
```

② Ordered by content level:

```
K.table(n = seq(50, 60, 10), alpha = c(0.01, 0.05, 0.10), P =  
c(0.90, 0.95), by.arg = "P")
```

③ Ordered by confidence level:

```
K.table(n = seq(50, 60, 10), alpha = c(0.01, 0.05), P = c(0.90,  
0.95, 0.99), by.arg = "alpha")
```

Normal Tolerance Interval: K.table, by 'n'

```
> K.table(n = seq(50, 60, 10), alpha = c(0.01, 0.05, 0.10),  
+         P = c(0.90, 0.95, 0.99), by.arg = "n")  
$'50'
```

	0.9	0.95	0.99
0.99	1.820800	2.268898	3.124607
0.95	1.645565	2.064993	2.862449
0.9	1.559468	1.965294	2.734892

```
$'60'
```

	0.9	0.95	0.99
0.99	1.764060	2.202345	3.038258
0.95	1.608913	2.022159	2.807055
0.9	1.532029	1.933272	2.693524

Normal Tolerance Interval: K.table, by 'P'

```
> K.table(n = seq(50, 70, 10), alpha = c(0.01, 0.05, 0.10),  
+         P = c(0.90, 0.95), by.arg = "P")
```

```
$'0.9'
```

	50	60	70
0.99	1.820800	1.764060	1.721577
0.95	1.645565	1.608913	1.581218
0.9	1.559468	1.532029	1.511212

```
$'0.95'
```

	50	60	70
0.99	2.268898	2.202345	2.152633
0.95	2.064993	2.022159	1.989865
0.9	1.965294	1.933272	1.909031

Normal Tolerance Interval: K.table, by ' α '

```
> K.table(n = seq(50, 70, 10), alpha = c(0.01, 0.05),  
+         P = c(0.90, 0.95, 0.99), by.arg = "alpha")  
$'0.99'
```

	0.9	0.95	0.99
50	1.820800	2.268898	3.124607
60	1.764060	2.202345	3.038258
70	1.721577	2.152633	2.973920

```
$'0.95'
```

	0.9	0.95	0.99
50	1.645565	2.064993	2.862449
60	1.608913	2.022159	2.807055
70	1.581218	1.989865	2.765393

Normal Tolerance Interval: K.factor.sim

```
K.factor.sim(n, P, alpha, side = c(1,2), method = c("EXACT", "BONF"),  
m, l=number of population)
```

Description:

K.factor.sim estimates k -factors for simultaneous tolerance intervals based on normality. Two adjustment methodologies are included in this function.

K.factor.sim: Methodology

- ① EXACT: [Mee \(1990\)](#) developed simultaneous tolerance limits for different groups. For larger combinations of n and l when `side=2` and `method="EXACT"`, the calculation can be slow.
- ② BONF: In [Krishnamoorthy and Mathew \(2009\)](#), a *Bonferroni* correction method is introduced. For larger sample sizes when `side = "BONF"`, there may be some accuracy issues with the 1-sided calculation since it depends on the noncentral t-distribution. The code is primarily intended to be used for moderate values of the noncentrality parameter. It will not be highly accurate, especially in the tails, for large values.

Normal Tolerance Interval: K.kactor.sim

Examples:

```
n_sizes <- c(2:6, seq(30, 70, 10))
l_sizes <- 2:6
KM_table <- sapply(1:length(l_sizes), function(i)
  sapply(1:length(n_sizes), function(j)
    round(K.factor.sim(n = n_sizes[j],
                      l = l_sizes[i], side=1, alpha = 0.1,
                      P = 0.9),3)))
dimnames(KM_table) <- list(n = n_sizes, l = l_sizes)
KM_table
```

Normal Tolerance Interval: k.factor.sim

```
> KM_table
1
n      2      3      4      5      6
 2  5.419 4.485 4.111 3.914 3.795
 3  3.358 3.137 3.045 2.998 2.972
 4  2.800 2.705 2.668 2.652 2.646
 5  2.525 2.475 2.460 2.456 2.457
 6  2.356 2.329 2.323 2.325 2.329
30  1.668 1.681 1.691 1.700 1.708
40  1.611 1.623 1.633 1.641 1.648
50  1.573 1.585 1.594 1.601 1.608
60  1.546 1.557 1.565 1.572 1.579
70  1.525 1.535 1.543 1.550 1.556
```

Tolerance Limits for Normally Distributed Data: `normtol.int`

```
normtol.int(x, alpha, P, side = c(1,2), method = c("HE", "HE2",  
"WBE", "ELL", "KM", "EXACT", "OCT"), m, log.norm = c(FALSE,TRUE))
```

Description:

`normtol.int` provides 1-sided or 2-sided tolerance intervals for data distributed according to either a normal distribution or log-normal distribution.

Derivation of normal-based tolerance intervals depends on the selection of k -factors, and therefore, 7 methods are included.

Tolerance Limits for Normally Distributed Data: `normtol.int`

Examples:

① One-Sided:

```
normtol.int(x = x, alpha = 0.05, P = 0.95, side = 1, method =  
"HE", log.norm = FALSE)
```

② Two-Sided:

```
normtol.int(x = x, alpha = 0.05, P = 0.95, side = 2, method =  
"HE", log.norm = FALSE)
```

Tolerance Limits for Normally Distributed Data: normtol.int

```
set.seed(100)
x <- rnorm(100, 0, 0.2)
out1 <- normtol.int(x = x, alpha = 0.05, P = 0.95, side = 1,
                    method = "HE", log.norm = FALSE)
out2 <- normtol.int(x = x, alpha = 0.05, P = 0.95, side = 2,
                    method = "HE", log.norm = FALSE)
#####
> out1
  alpha      P      x.bar 1-sided.lower 1-sided.upper
1  0.05 0.95 0.0005825125   -0.3927051    0.3938701
> out2
  alpha      P      x.bar 2-sided.lower 2-sided.upper
1  0.05 0.95 0.0005825125   -0.4554493    0.4566144
```


Tolerance Limits for Normally Distributed Data: normtol.int

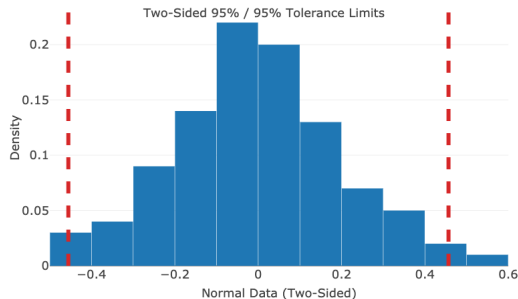
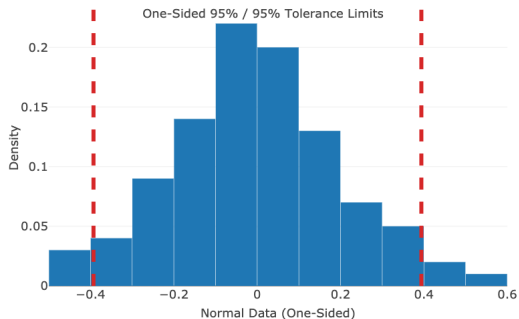


Figure: One-Sided v.s. Two-Sided

```
plotly.plottol.hist(out1, x, side = "two",  
  x.lab = "Normal Data (One-Sided)",  
  tol.line.type = "dash")
```

```
plotly.plottol.hist(out2, x, side = "two",  
  x.lab = "Normal Data (One-Sided)",  
  tol.line.type = "dash")
```

Sample Size Determination: norm.ss

```
norm.ss(x, alpha, P, side = 1, method = c("DIR", "FW", "YGZO"), m =  
50, delta = NULL, P.prime = NULL, spec = c(NA, NA), hyper.par =  
list(mu.0 = NULL, sig2.0 = NULL, m.0 = NULL, n.0 = NULL))
```

Description:

norm.ss provides minimum sample sizes for a future sample when constructing normal tolerance intervals. Various strategies are available for determining the sample size, including strategies that incorporate known specification limits.

Methodology:

- ① DIR: is the direct method (intended as a simple calculation for planning purposes) where the mean and standard deviation are taken as truth and the sample size is determined with respect to the given specification limits.
- ② FW: is for the traditional Faulkenberry-Weeks approach for sample size determination ([Faulkenberry & Weeks, 1968](#)).
- ③ YGZO: is for the Young-Gordon-Zhu-Olin approach, which incorporates historical data and specification limits for determining the value of delta and/or P.prime in the Faulkenberry-Weeks approach ([Young, Gordon, Zhu, & Olin, 2016](#)).

Sample Size Determination: norm.ss

Examples:

① Direct Method:

```
norm.ss(alpha = 0.05, P = 0.95, side = 2, spec = c(-3, 3), method = "DIR", hyper.par = list(mu.0 = 0, sig2.0 = 1))
```

② FW Method:

```
norm.ss(alpha = 0.05 , P = 0.90, delta = 0.10 , P.prime = 0.97 , side = 2 , m = 50 , method = "FW")
```

③ YGZO Method:

```
norm.ss(x = rnorm(100 , mean = 0, sd = 3) , alpha = 0.05 , P = 0.90, delta = 0.10 , side = 2 , m = 50 , method = "YGZO" , spec = c(-2, 2), hyper.par = list(mu.0 = 0, sig2.0 = 1))
```

Sample Size Determination: norm.ss

```
> set.seed(100)
### DIR Method ###
> norm.ss(alpha = 0.05, P = 0.95, side = 2, spec = c(-3, 3),
+         method = "DIR", hyper.par = list(mu.0 = 0, sig2.0 = 1))
alpha    P    delta P.prime    n
1      0.05 0.95          20
#####

### FW Method ###
> norm.ss(alpha = 0.05 , P = 0.90, delta = 0.10 , P.prime = 0.97 ,
+         side = 2 , m = 50 , method = "FW")
alpha    P delta P.prime    n
1 0.05 0.9 0.1 0.97 60
#####

### YGZO Method ###
> norm.ss(x = rnorm(100 , mean = 0, sd = 3) , alpha = 0.05 , P = 0.90, delta = 0.10 ,
+         side = 2 , m = 50 , method = "YGZO" , spec = c(-2, 2),
+         hyper.par = list(mu.0 = 0, sig2.0 = 1))
alpha    P delta    P.prime    n
1 0.05 0.9 0.1 0.9544997 117
```

Operating Characteristic Curves: norm.OC

```
norm.OC(alpha, P, n, side = c(1,2), method = c("HE", "HE2", "WBE",  
"ELL", "KM", "EXACT", "OCT"), k, m)
```

Description:

`norm.OC` provides OC-type curves to illustrate how values of the k -factors for normal tolerance intervals, confidence levels, and content levels change as a function of the sample size.

Operating Characteristic Curves: norm.OC

Examples:

① By content level:

`norm.OC(k = 4, alpha = NULL, P = c(0.90, 0.95, 0.99), n = 10:20,
side = 1)`

② By confidence level:

`norm.OC(k = 4, alpha = c(0.01, 0.05, 0.10), P = NULL, n = 10:20,
side = 1)`

③ By sample size:

`norm.OC(k = NULL, P = c(0.90, 0.95, 0.99),
alpha=c(0.01,0.05,0.10), n = 10:20, side = 1)`

Operating Characteristic Curves: norm.OCC

```
plotly.norm.OCC(k = 4, alpha = NULL, P = c(0.90, 0.95, 0.99),
  n = 10:20, side = 1,
  x.cex = 8, line.width = 4,
  y.lab.size = 16, x.lab.size = 16,
  x.tick.size = 16, y.tick.size = 16,
  title.size = 16, legend.size = 12)
#####
plotly.norm.OCC(k = 4, alpha = c(0.01, 0.05, 0.10), P = NULL,
  n = 10:20, side = 1,
  x.cex = 8, line.width = 4,
  y.lab.size = 16, x.lab.size = 16,
  x.tick.size = 16, y.tick.size = 16,
  title.size = 16, legend.size = 12)
#####
plotly.norm.OCC(k = NULL, P = c(0.90, 0.95, 0.99),
  alpha=c(0.01,0.05,0.10), n = 10:20, side = 1,
  x.cex = 8, line.width = 4,
  y.lab.size = 16, x.lab.size = 16,
  x.tick.size = 16, y.tick.size = 16,
  title.size = 16, legend.size = 12)
```


Operating Characteristic Curves: norm.OC

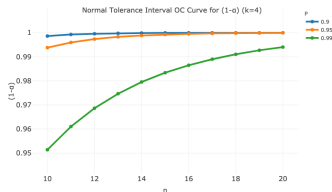


Figure: By P

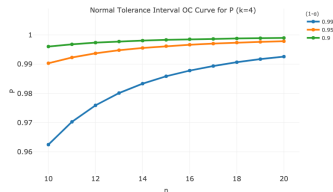


Figure: By $(1 - \alpha)$

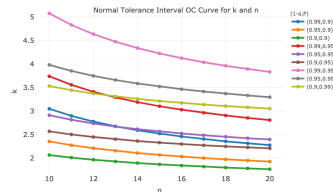


Figure: By n

Normal Tolerance Interval: Application

Normal distribution is not always seen in practice. For example, a mass production process usually requires quality control process and certain measurements might only follow a normal distribution approximately.

One illustrative example is the thickness of a metal layer on $n = 100$ silicon wafers resulting from a chemical vapor deposition (CVD) process in a semiconductor plant; see Example 3.2 of [Montgomery \(2020\)](#).

Application: Data

This distribution of the thickness of a metal layer on 100 silicon wafers resulting from a chemical vapor deposition (CVD) process in a semiconductor plant is shown in Figure 5.

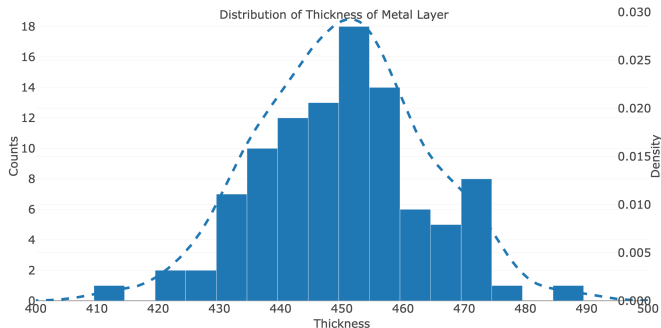


Figure: Thickness of Metal Layer

Application: Tolerance Limits

The two-sided $(P, \gamma) = (0.95, 0.95)$ tolerance limits for thickness data are $(420.015, 480.005)$ and are presented in Figure 6.

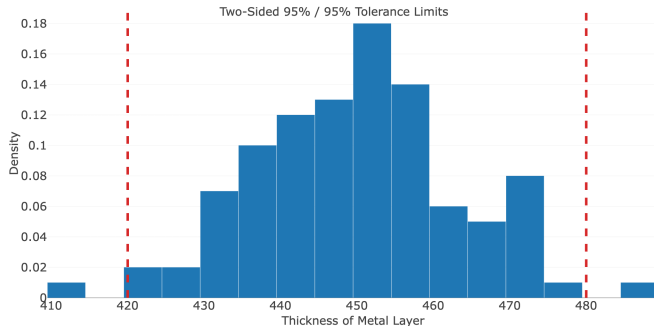


Figure: Two-Sided Tolerance Limits

Some Non-Normal Tolerance Intervals

Some Non-normal Tolerance Intervals

① Introduction

② Discrete Type

- ① Example: Tolerance Limits for [Binomial Distribution](#) [bintol.int]
- ② Example: Tolerance Limits for [Poisson Distribution](#) [poistol.int]

③ Continuous Type

- ① Example: Tolerance Limits for [Weibull Distribution](#) [exttol.int]
- ② Example: Tolerance Limits for [Gamma Distribution](#) [gamtol.int]

④ Application

Non-normal Tolerance Interval: Introduction

Although normality is the most commonly seen condition under many situations, non-normal distributions are perhaps just as frequently encountered. The `tolerance` package also handles common non-normal distributions. We illustrate the functionalities of `tolerance` package by presenting four classical non-normal distributions. More specifically, tolerance intervals for *binomial*, *Poisson*, *Weibull* and *gamma* distributions are presented.

Binomial Distribution: bintol.int

```
bintol.int(x, n, m = NULL, alpha = 0.05, P = 0.99, side = 1, method =  
c("LS", "WS", "AC", "JF", "CP", "AS", "LO", "PR", "PO", "CL", "CC",  
"CWS"), a1 = 0.5, a2 = 0.5)
```

Description:

bintol.int Provides 1-sided or 2-sided tolerance intervals for binomial random variables. From a statistical quality control perspective, these limits use the proportion of defective (or acceptable) items in a sample to bound the number of defective (or acceptable) items in future productions of a specified quantity.

- ① x: Number of defective (or acceptable) units.
- ② n: Current lot size. ($\hat{p} = \frac{x}{n}$.)
- ③ m: Future lot size.

Binomial Distribution: `bintol.int`

Methodology

Interval calculation for a *Binomial* distribution is complex and numerous methods are developed. (For more details, please see [Brown, Cai, and DasGupta \(2001\)](#)).

`bintol.int` equips multiple methods:

- ① “LS”: The large-sample method (the default method), which is appropriate when the sample size is large (e.g., $n \geq 50$) and $n\hat{p}$ and $n(1 - \hat{p})$ are both ≥ 10 . Otherwise, the coverage may not achieve the nominal level.
- ② “WS”: Wilson’s method, which is appropriate even when the sample size is small (e.g., $n \leq 40$). The coverage probability fluctuates acceptably near the nominal level provided p does not approach 0 or 1.
- ③ “AC”: The Agresti-Coull method, which is appropriate when the sample size is large (e.g., $n \geq 40$). This method is comparable to Wilson’s method for large n .

Binomial Distribution: `bintol.int`

Methodology

- ① “JF”: Jeffreys’ method, which is a Bayesian approach to the estimation. For this method, the prior distribution for p is assumed to be a beta distribution with parameters $a1$ and $a2$, both of which must be specified by the user.
- ② “CP”: The Clopper-Pearson method (sometimes referred to as an “exact” procedure due to its derivation from the binomial distribution), which provides a more conservative interval and can be much larger than the nominal level as $n \rightarrow \infty$.
- ③ “AS”: The arcsine method, which is appropriate when p is not too close to 0 or 1. Otherwise, the coverage approaches 0.
- ④ “LO”: The logit method, which is appropriate when p is not too close to 0 or 1, but yields a more conservative interval.

Binomial Distribution: bintol.int

Examples:

① Tolerance Limits using Multiple Methods:

- ▶ `bintol.int(x = 230, n = 1000, m = 2500, alpha = 0.15, P = 0.90, side = 2, method = "LS")`
- ▶ `bintol.int(x = 230, n = 1000, m = 2500, alpha = 0.15, P = 0.90, side = 2, method = "AS")`

② Tolerance Limits using Jeffreys' Methods:

- ▶ `bintol.int(x = 230, n = 1000, m = 2500, alpha = 0.15, P = 0.90, side = 1, method = "JF", a1 = 2, a2 = 10)`
- ▶ `bintol.int(x = 230, n = 1000, m = 2500, alpha = 0.15, P = 0.90, side = 1, method = "JF", a1 = 5, a2 = 1)`

Binomial Distribution: bintol.int

```
> bintol.int(x = 230, n = 1000, m = 2500, alpha = 0.15, P = 0.90,
+           side = 2, method = "LS")
  alpha    P p.hat 2-sided.lower 2-sided.upper
1 0.15 0.9 0.23         494         659
#####
> bintol.int(x = 230, n = 1000, m = 2500, alpha = 0.15, P = 0.90,
+           side = 2, method = "AS")
  alpha    P p.hat 2-sided.lower 2-sided.upper
1 0.15 0.9 0.23         495         660
#####
> bintol.int(x = 230, n = 1000, m = 2500, alpha = 0.15, P = 0.90,
+           side = 1, method = "JF", a1 = 2, a2 = 10)
  alpha    P p.hat 1-sided.lower 1-sided.upper
1 0.15 0.9 0.23         513         635
#####
> bintol.int(x = 230, n = 1000, m = 2500, alpha = 0.15, P = 0.90,
+           side = 1, method = "JF", a1 = 5, a2 = 1)
  alpha    P p.hat 1-sided.lower 1-sided.upper
1 0.15 0.9 0.23         523         646
```

Poisson Distribution: poistol.int

```
poistol.int(x, n, m = NULL, alpha = 0.05, P = 0.99, side = 1, method  
= c("TAB", "LS", "SC", "CC", "VS", "RVS", "FT", "CSC"))
```

Description:

poistol.int Provides 1-sided or 2-sided tolerance intervals for Poisson random variables. From a statistical quality control perspective, these limits bound the number of occurrences (which follow a Poisson distribution) in a specified future time period.

- ❶ x: The number of occurrences of the event in time period n .
- ❷ n: The time period of the original measurements. ($\hat{\lambda} = \frac{x}{n}$.)
- ❸ m: Future time period.

Methodology

`tolerance` package includes seven methods calculating tolerance intervals with an underlying *Poisson* distribution. Two methods, tabulation and large-sample methods, build the foundation while others provide continuity and variance stabilization corrections.

- ① TAB: tabular method and is usually preferred for a smaller number of occurrences. This is the default method.
- ② LS: the large-sample (Wald) method, which is usually preferred when the number of occurrences is $x > 20$.
- ③ CC: gives a continuity-corrected version of the large-sample method.
- ④ VS: gives a variance-stabilized version of the large-sample method.
- ⑤ RVS: is a recentered version of the variance-stabilization method.
- ⑥ SC: the score method, which again is usually used when the number of occurrences is relatively large.
- ⑦ CSC: is the continuity-corrected version of the score method.
- ⑧ FT: is the Freeman-Tukey method.

Methodology

① Tabular Method

Hahn and Meeker (2011) shows that one-sided limits has the form

$$\begin{aligned}\lambda_L &= \frac{\chi_{2x;\alpha}^2}{2n} \\ \lambda_U &= \frac{\chi_{2x+2;1-\alpha}^2}{2n}\end{aligned}$$

where n is the sample size and $\chi_{d;1-\alpha}^2$ is the $(1 - \alpha)^{th}$ quantile of a χ^2 distribution with d degrees of freedom.

② Large-Sample Method

[Hahn and Meeker \(2011\)](#) also utilized large-sample theory and that one-sided limits has the form

$$\lambda_L = \hat{\lambda} - z_{1-\alpha} \sqrt{\frac{\hat{\lambda}}{n}}$$

$$\lambda_U = \hat{\lambda} + z_{1-\alpha} \sqrt{\frac{\hat{\lambda}}{n}}$$

— This method performs quite good and usually is preferred for $n > 20$.

— For both methods in two-sided setting, simply replace α by $\frac{\alpha}{2}$ and P by $\frac{P+1}{2}$.

Further details on *Poisson* tolerance intervals can be found in [Hahn and Chandra](#)

(1981)

Poisson Distribution: poistol.int

Examples:

① One-sided:

- ▶ `pistol.int(x = 45, n = 9, m = 3, alpha = 0.05, P = 0.90, side = 1, method = "TAB")`
- ▶ `pistol.int(x = 45, n = 9, m = 3, alpha = 0.05, P = 0.90, side = 1, method = "LS")`

② Two-sided:

- ▶ `pistol.int(x = 45, n = 9, m = 15, alpha = 0.05, P = 0.90, side = 2, method = "TAB")`
- ▶ `pistol.int(x = 45, n = 9, m = 15, alpha = 0.05, P = 0.90, side = 2, method = "LS")`

Poisson Distribution: poistol.int

```
### One-Sided ###
> poistol.int(x = 45, n = 9, m = 3, alpha = 0.05, P = 0.90,
+           side = 1, method = "TAB")
  alpha  P lambda.hat 1-sided.lower 1-sided.upper
1 0.05 0.9           5             7           25
#####
> poistol.int(x = 45, n = 9, m = 3, alpha = 0.05, P = 0.90,
+           side = 1, method = "LS")
  alpha  P lambda.hat 1-sided.lower 1-sided.upper
1 0.05 0.9           5             7           24

### Two-Sided ###
> poistol.int(x = 45, n = 9, m = 15, alpha = 0.05, P = 0.90,
+           side = 2, method = "TAB")
  alpha  P lambda.hat 2-sided.lower 2-sided.upper
1 0.05 0.9           5            43           117
#####
> poistol.int(x = 45, n = 9, m = 15, alpha = 0.05, P = 0.90,
+           side = 2, method = "LS")
  alpha  P lambda.hat 2-sided.lower 2-sided.upper
1 0.05 0.9           5            41           113
```

Weibull Distribution: `exttol.int`

```
exttol.int(x, alpha = 0.05, P = 0.99, side = 1, dist = c("Weibull",  
"Gumbel"), ext = c("min", "max"), NR.delta = 1e-8)
```

Description:

`exttol.int` Provides 1-sided or 2-sided tolerance intervals for data distributed according to either a Weibull distribution or extreme-value (also called Gumbel) distributions.

- ① ext: If `dist = "Gumbel"`, then select which extreme is to be modeled for the Gumbel distribution. The Gumbel distribution for the minimum (i.e., `ext = "min"`) corresponds to a left-skewed distribution and the Gumbel distribution for the maximum (i.e., `ext = "max"`) corresponds to a right-skewed distribution.
- ② NR.delta: The stopping criterion used for the Newton-Raphson algorithm when finding the maximum likelihood estimates of the Weibull or extreme-value distribution.

Weibull Distribution: exttol.int

Methodology

Weibull and *Gumbel* distributions are two representative distribution measuring extreme events. They have wide applications in the fields such as physics and engineering. tolerance package supplies tolerance interval calculations for both distribution. A random variable X has a *Weibull* distribution has a cumulative distribution function

$$F_X(x; \theta, \beta) = 1 - e^{-(\frac{x}{\theta})^\beta}$$

where $x > 0$ with shape parameter $\beta > 0$ and scale parameter $\theta > 0$. Let a random variable $Y = \ln(X)$, then Y follows a extreme-value distribution (also called *Gumbel* distribution for the minimum), and it has the cumulative distribution

$$F_Y(y; \xi, \delta) = 1 - e^{-e^{\frac{y-\xi}{\delta}}}$$

A random variable Z is said to have a *Gumbel* distribution for the maximum if it has the cumulative density function of

$$F_Y(y; \xi, \delta) = 1 - e^{-e^{-\frac{y-\xi}{\delta}}}$$

where $-\infty < y < +\infty$, $-\infty < \xi = \ln(\theta) < +\infty$, and $\delta = \frac{1}{\beta} > 0$. Newton-Raphson algorithm is used to estimate parameters for *Gumbel* distributions.

Weibull Distribution: exttol.int

According to [Bain and Engelhardt \(1981\)](#), the formula for estimating the 1-sided extreme-value tolerance limits are

$$L = \hat{\xi} - \frac{\hat{\delta}t_{n-1;\alpha}^*(-\sqrt{n}\lambda_P)}{\sqrt{n-1}}$$
$$U = \hat{\xi} - \frac{\hat{\delta}t_{n-1;1-\alpha}^*(-\sqrt{n}\lambda_{1-P})}{\sqrt{n-1}}$$

Furthermore, the formulas for estimating the 1-sided *Weibull* tolerance limits are

$$L_W = e^L$$
$$U_W = e^U$$

Weibull Distribution: exttol.int

For a *Gumbel* distribution for a maximum, the maximum likelihood estimates of the parameter ($\hat{\xi}$ and $\hat{\delta}$) can be found by using a Newton-Raphson algorithm. The formula for estimating the one-sided tolerance limits for the *Gumbel* distribution for the maximum are

$$L = \hat{\xi} + \frac{\hat{\delta} t_{n-1;\alpha}^*(-\sqrt{n}\lambda_{1-P})}{\sqrt{n-1}}$$
$$U = \hat{\xi} + \frac{\hat{\delta} t_{n-1;1-\alpha}^*(-\sqrt{n}\lambda_P)}{\sqrt{n-1}}$$

Weibull Distribution: exttol.int

Examples:

`exttol.int` provides tolerance interval calculations for both *Weibull* and *Gumbel* distributions. When *Gumbel* distribution is selected, the *Gumbel* distribution for the minimum or maximum need to be further specified.

① Weibull Distribution:

```
exttol.int(x = x, alpha = 0.15, P = 0.90, side = 1, dist =  
"Weibull")
```

② Gumbel Distribution:

```
exttol.int(x = GumbelData, alpha = 0.15, P = 0.90, side = 2, dist =  
"Gumbel" , ext = "max")
```

Weibull Distribution: exttol.int

```
## 85%/90% 1-sided Weibull tolerance intervals for a sample
## of size 150.
> set.seed(100)
> WeibullData <- rweibull(150, 3, 75)
> out <- exttol.int(x = WeibullData, alpha = 0.15, P = 0.90, side = 1,
+                  dist = "Weibull")
> out
  alpha   P shape.1 shape.2 1-sided.lower 1-sided.upper
1  0.15 0.9 3.084836 74.48706      33.87539      101.0331
#####
## 85%/90% 2-sided Gumbel distribution for the maximum tolerance intervals
## for a sample of size 200.
> set.seed(100)
> GumbelData <- rgumbel(200, loc=3, scale=5)
> out <- exttol.int(x = GumbelData, alpha = 0.15, P = 0.90, side = 2,
+                  dist = "Gumbel" , ext = "max")
> out
  alpha   P shape.1 shape.2 2-sided.lower 2-sided.upper
1  0.15 0.9 3.504217 4.791701      -2.422067      19.01293
```


Weibull Distribution: exttol.int

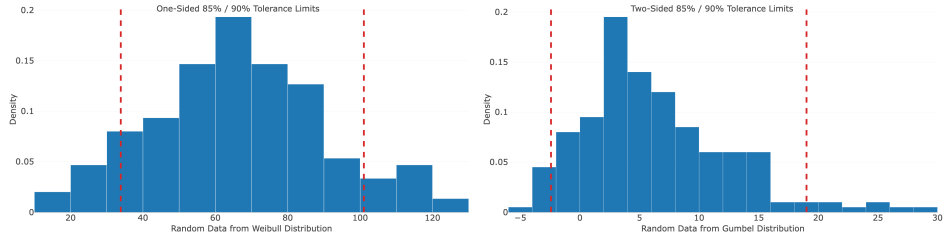


Figure: Tolerance Limits for Weibull and Gumbel Distributions

Gamma Distribution: `gamtol.int`

```
gamtol.int(x, alpha = 0.05, P = 0.99, side = 1, method = c("HE",  
"HE2", "WBE", "ELL", "KM", "EXACT", "OCT"), m = 50, log.gamma =  
FALSE)
```

Description:

`gamtol.int` Provides 1-sided or 2-sided tolerance intervals for data distributed according to either a gamma distribution or log-gamma distribution.

- ① method: method here is specified for computing the k -factor for normal tolerance limits.
- ② m: The maximum number of subintervals to be used in the integrate function. This is necessary only for `method = "EXACT"` and `method = "OCT"`. The larger the number, the more accurate the solution, with the trade-off of slowing the calculation.

Gamma Distribution: gamtol.int

Methodology

Krishnamoorthy, Mathew, and Mukherjee (2008) developed the tolerance limits for a *Gamma* distribution based on a normal approximation.

A random variable X is *gamma* distributed if it has the cumulative distribution function

$$F_X(x; \theta, \beta) = \int_{t=0}^x \frac{t^{\theta-1} e^{-\frac{t}{\beta}}}{\beta^{\theta} \Gamma(\theta)} dt$$

where $x > 0$, $\theta > 0$ is a shape parameter and $\beta > 0$ is a scale parameter, and $\Gamma(\cdot)$ is the gamma function.

If X follows a *Gamma* distribution, the $X^{\frac{1}{3}}$ is *approximately* a normal distribution with mean μ and variance σ^2 such that

$$\mu = \frac{\beta^{\frac{1}{3}} \Gamma(\theta + \frac{1}{3})}{\Gamma(\theta)} \quad ; \quad \sigma^2 = \frac{\beta^{\frac{2}{3}} \Gamma(\theta + \frac{2}{3})}{\Gamma(\theta)} - \mu^2$$

Gamma Distribution: `gamtol.int`

Since $X^{\frac{1}{3}}$ follows a normal distribution approximately, corresponding tolerance limits can be found in the setting for *Normal* distribution such that

$$(L_N, U_N) = \hat{\mu} \pm k\hat{\sigma}$$

(Note: method argument in `gamtol.int` is used here.)

Once the above limits are calculated, they need to be transferred back to obtain tolerance limits for a *Gamma* distribution such that

$$(L, U) = (L_N^3, U_N^3)$$

If X^* is a log-gamma random variable, then $\ln(X^* = X)$ follows a *Gamma* distribution. Then an extra step needs to be done to obtain the tolerance limits for X^* such that

$$(L^*, U^*) = (e^L, e^U)$$

Gamma Distribution: `gamtol.int`

Examples:

❶ One-Sided:

```
gamtol.int(x = x, alpha = 0.01, P = 0.99, side = 1, method =  
"HE")
```

❷ Two-Sided:

```
gamtol.int(x = x, alpha = 0.05, P = 0.95, side = 2, method =  
"EXACT")
```

Gamma Distribution: gamtol.int

```
> set.seed(100)
> x <- rgamma(50, 0.30, scale = 2)

## 99%/99% 1-sided gamma tolerance intervals for a sample
## of size 50, using Howe method.
> out1 <- gamtol.int(x = x, alpha = 0.01, P = 0.99, side = 1,
+                   method = "HE")
> out1
  alpha    P 1-sided.lower 1-sided.upper
1 0.01 0.99              0      6.769559
#####
## 95%/95% 2-sided gamma tolerance intervals for a sample
## of size 50, using EXACT method.
> out2 <- gamtol.int(x = x, alpha = 0.05, P = 0.95, side = 2,
+                   method = "EXACT")
> out2
  alpha    P 2-sided.lower 2-sided.upper
1 0.05 0.95              0      3.983218
```

Gamma Distribution: gamtol.int

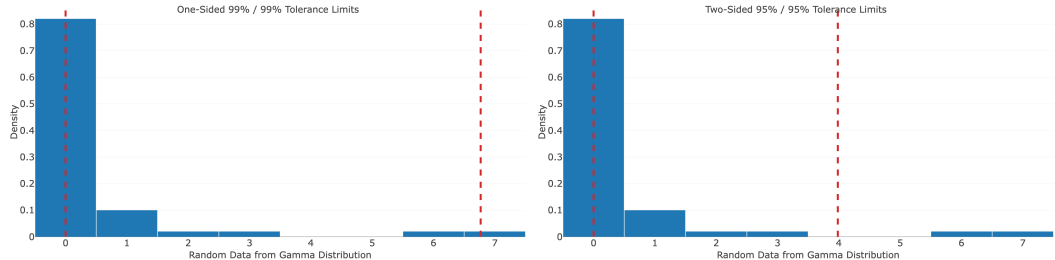


Figure: Tolerance Limits Gamma Distributions

Application: Binomial Distribution

A proportion control chart is a representative illustration of a *Binomial* distribution since products will be categorized as either good or defective.

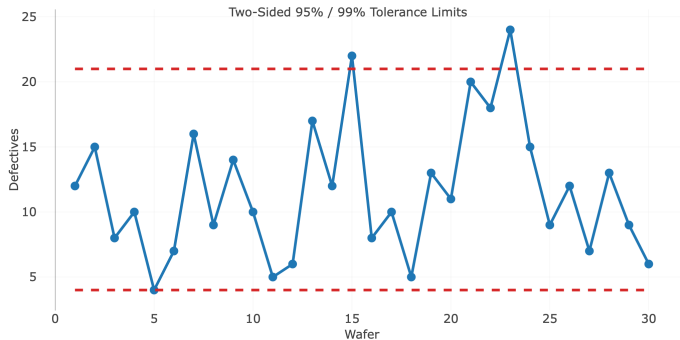
30 wafers of chips were selected for quality control process. On each wafer 50 chips are measured and a defective is defined whenever a misregistration, in terms of horizontal and/or vertical distances from the center, is recorded. Numerical data can be found [here](#).

Suppose we want to calculate the tolerance limits for a future wafer of 50 chips. `bintol.int` can be therefore applied. In addition, `plotly.plot.control` returns the corresponding control chart.

```
> binom.app <- bintol.int(x=defects, n=50*30,
m=50, alpha=0.05, P=0.99, side=2, method="CP")
> binom.app
      alpha      P      p.hat 2-sided.lower 2-sided.upper
1  0.05 0.99 0.2313333          4          21
```


Application: Binomial Distribution

defects contains data of number of defectives for 30 wafers, n is total number of chips and m is the future lot size, which is one wafer (50) of chips. The two-sided $(P, \gamma) = (0.99, 0.95)$ tolerance limits is $(4, 21)$.



Application: Poisson Distribution

As is well known that a *Poisson* distribution returns the probability of number of occurrence of an event within a certain period of time. Doctors visits is a great example to demonstrate a *Poisson* distribution.

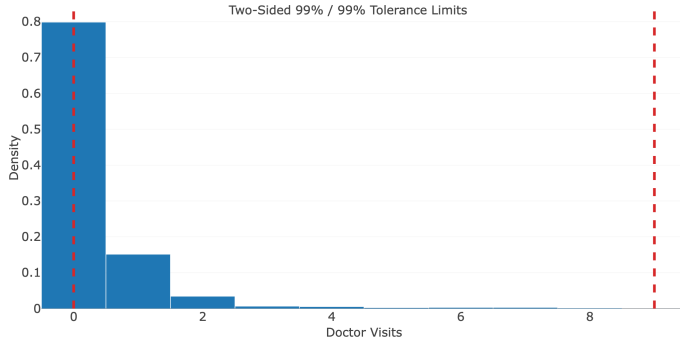
Cameron and Trivedi (1986) provided doctors visits data which include 5190 patients' data of doctors visits over the span of 2 weeks. Actual dataset is embedded in AER package.

Suppose we are interested in the tolerance limits of number of visits for the next 10 individuals. *poistol.int* shall be applied to compute the tolerance limits for a *Poisson* distribution.

```
> library(AER)
> data("DoctorVisits")
> visits <- DoctorVisits$visits
> pois.app <- poistol.int(x=visits, n=length(visits), m=10,
alpha=0.01, P=0.99, side=2, method="TAB")
> pois.app
  alpha      P lambda.hat 2-sided.lower 2-sided.upper
1 0.01 0.99 0.3017341          0          9
```

Application: Poisson Distribution

The two-sided $(P, \gamma) = (0.99, 0.99)$ tolerance limits for this *Poisson* distributed is $(0, 9)$.



Application: Weibull Distribution

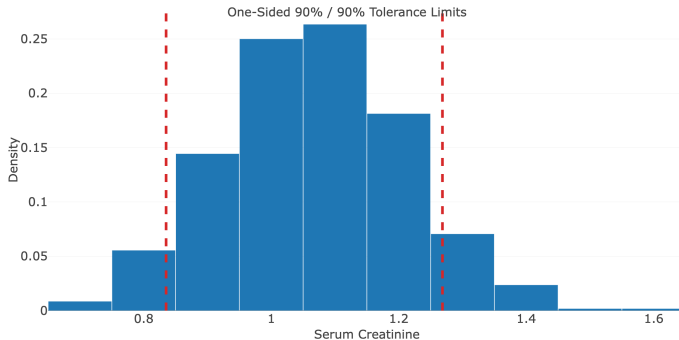
Harris and Boyd (1995) provided kidney function data of 596 participants with 5 different measurements.

Suppose we are interested in the tolerance limits of serum creatinine of those 596 participants, *exttol.int* is a good option to compute the tolerance limits for a *Weibull* distribution.

```
> kidney <- read.table("https://raw.githubusercontent.com/dsy109/Supplemental/main/CSP%202022%20Trainings/
                        tolerance/Datasets/kidney.txt",header=TRUE)
> weibull.app <- exttol.int(x=kidney$SCR, alpha=0.10, P=0.90, side=1, dist="Weibull")
> weibull.app
  alpha    P shape.1 shape.2 1-sided.lower 1-sided.upper
1   0.1 0.9 7.780265 1.130353    0.8351046    1.268533
```

Application: Weibull Distribution

The one-sided $(P, \gamma) = (0.90, 0.90)$ lower tolerance limits for this *Weibull* distributed data is 0.835 and upper tolerance limits is 1.269.



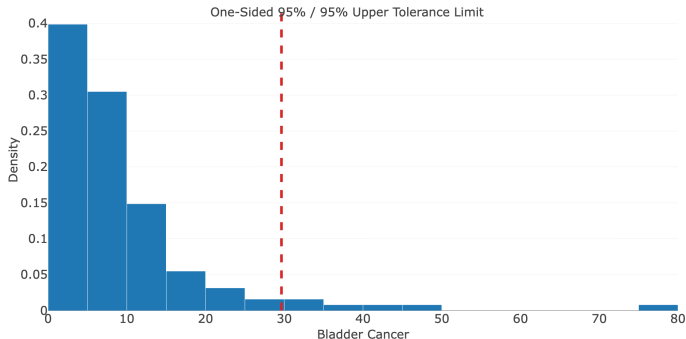
Application: Gamma Distribution

Lee and Wang (2003) provided a sample of remission time data for bladder cancer patients. The remission time follow a *Gamma* distribution approximately. Suppose we would like to calculate a one-sided upper tolerance limits for the sample data, *gamtol.int* shall be used in this case.

```
> bladder <- unlist(read.table("https://raw.githubusercontent.com/dsy109/Supplemental/main/
CSP%202022%20Trainings/tolerance/Datasets/bladder.txt",header=FALSE))
> gamma.app <- gamtol.int(x=bladder, alpha=0.05, P=0.95, side=1)
> gamma.app
  alpha    P 1-sided.lower 1-sided.upper
1  0.05 0.95      0.3655981      29.64082
```

Application: Gamma Distribution

The one-sided $(P, \gamma) = (0.95, 0.95)$ upper tolerance limits for this *Gamma* distributed bladder data 29.641.



Nonparametric Tolerance Intervals

Nonparametric Tolerance Intervals

- ① Introduction
- ② Nonparametric Tolerance Intervals: `[nptol.int]`
- ③ β -Expectation Tolerance Interval: `[npbetol.int]`
- ④ Sample Size Determination: `[np.order]`
- ⑤ Application

Nonparametric Tolerance Interval: Introduction

Occasionally, a researcher may not wish to make any distributional assumption regarding their data. All that is assumed about the random sample $X_1 = x_1, \dots, X_n = x_n$ is that the underlying distribution function F_X is a continuous, non-decreasing, probability distribution.

The basic form of the (P, γ) nonparametric tolerance limits are

$$L = x_{[r]} \quad ; \quad U = x_{[s]}$$

where $x_{[j]}$ corresponds to the j^{th} order statistics of the sample data.

Methodology

Computing nonparametric tolerance intervals involves finding the appropriate r and s values ($x_{[r]}$ and $x_{[s]}$), which is typically done using the *Beta* or *Binomial* distribution.

- ① "HM": The Hahn-Meeker method ([Hahn & Meeker, 2011](#)), which uses an estimate based on the binomial distribution; however, two intervals may be reported if an odd number of observations must be trimmed from both sides.
- ② "WILKS": The Wilks method ([Wilks, 1941](#)), which uses an estimate based on the beta distribution to omit a certain number of observations from either side. For the two-sided intervals, the tolerance intervals are symmetric about the center of the observed data.
- ③ "WALD": The Wald method ([Wald, 1943](#)), which is the same as the Wilks method for the one-sided setting. For the two-sided setting, symmetry about the center of the observed data is not assumed and this method finds all possible tolerance intervals, each having at least the specified confidence level.
- ④ "YM": The the Young-Mathew method ([Young & Mathew, 2014](#)) for performing interpolation or extrapolation based on the order statistics.

Nonparametric Tolerance Interval: `nptol.int`

```
nptol.int(x, alpha, P, side = c(1,2), method = c("WILKS", "WALD",  
"HM", "YM"), upper, lower)
```

Description:

`nptol.int` provides 1-sided or 2-sided nonparametric (i.e., distribution-free) tolerance intervals for any continuous data set.

Note: we also add an important option for users to specify a certain upper or lower bound based on their professionalism.

Nonparametric Tolerance Interval: `nptol.int`

Examples:

① One-Sided:

```
nptol.int(x = x, alpha = 0.10, P = 0.90, side = 1, method =  
"WILKS", upper = NULL, lower = NULL)
```

② Two-Sided (with a specified upper bound):

```
nptol.int(x = x, alpha = 0.10, P = 0.90, side = 2, method =  
"WILKS", upper = NULL, lower = NULL)
```

Nonparametric Tolerance Interval: nptol.int

```
set.seed(100)
x <- rlogis(100, 5, 1)
out1 <- nptol.int(x = x, alpha = 0.10, P = 0.90, side = 1,
                  method = "WILKS", upper = NULL, lower = NULL)
out2 <- nptol.int(x = x, alpha = 0.10, P = 0.90, side = 2,
                  method = "WILKS", upper = NULL, lower = NULL)
#####
> out1
  alpha    P 1-sided.lower 1-sided.upper
1  0.1 0.9      3.056274      7.563949
> out2
  alpha    P 2-sided.lower 2-sided.upper
1  0.1 0.9      2.18245      8.377662
```

Nonparametric Tolerance Interval: `nptol.int`

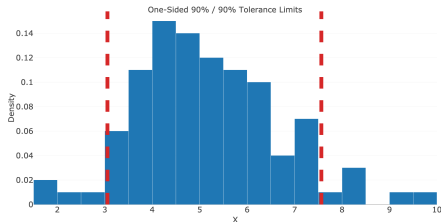


Figure: 1-Sided

```
plotly.plottol.hist(out1 , x=x , side = "two" ,  
  x.lab = "X", tol.lwd = 18 ,  
  x.lab.size = 36 , x.tick.size=36 ,  
  y.lab.size = 36 , y.tick.size = 36 ,  
  title.size = 36 , title.position.y = 0.995)
```

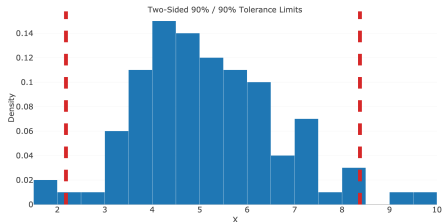


Figure: 2-Sided

```
plotly.plottol.hist(out2 , x=x , side = "two" ,  
  x.lab = "X", tol.lwd = 18 ,  
  x.lab.size = 36 , x.tick.size=36 ,  
  y.lab.size = 36 , y.tick.size = 36 ,  
  title.size = 36 , title.position.y = 0.995)
```

Nonparametric β —Expected Tolerance Interval: `npbetol.int`

```
npbetol.int(x, Beta, side = c(1,2), upper, lower)
```

Description:

`npbetol.int` provides 1-sided or 2-sided nonparametric (i.e., distribution-free) beta-expectation tolerance intervals for any continuous data set. These are equivalent to nonparametric prediction intervals based on order statistics.

Nonparametric β —Expected Tolerance Interval: npbetol.int

Examples:

① One-Sided:

`npbetol.int(x = x, Beta = 0.90, side = 1, upper = NULL, lower = NULL)`

② Two-Sided (with a specified lower bound):

`npbetol.int(x = x, Beta = 0.90, side = 2, upper = NULL, lower = 4)`

Nonparametric β —Expected Tolerance Interval: npbetol.int

```
set.seed(100)
x <- rlogis(100, 5, 1)
out1 <- npbetol.int(x = x, Beta = 0.90, side = 1,
                    upper = NULL, lower = NULL)
out2 <- npbetol.int(x = x, Beta = 0.90, side = 2,
                    upper = NULL, lower = 4)
#####
> out1
      beta 1-sided.lower 1-sided.upper
1  0.9          3.486409      7.085171
> out2
      beta 2-sided.lower 2-sided.upper
1  0.9              4          7.563949
```

Nonparametric β -Expected Tolerance Interval: npbetol.int

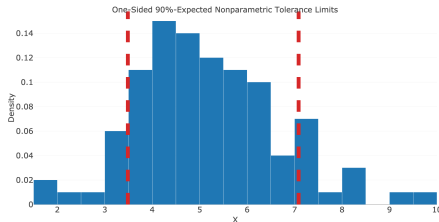


Figure: 1-Sided

```
plotly.plottol.hist(out1 , x=x , side = "two" ,  
  x.lab = "X", tol.lwd = 18 ,  
  x.lab.size = 36 , x.tick.size=36 ,  
  y.lab.size = 36 , y.tick.size = 36 ,  
  title.size = 36 , title.position.y = 0.995)
```

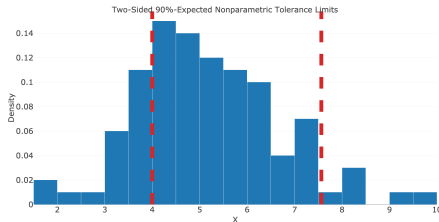


Figure: 2-Sided (with Lower Bound 4)

```
plotly.plottol.hist(out2 , x=x , side = "two" ,  
  x.lab = "X", tol.lwd = 18 ,  
  x.lab.size = 36 , x.tick.size=36 ,  
  y.lab.size = 36 , y.tick.size = 36 ,  
  title.size = 36 , title.position.y = 0.995)
```

Sample Size Determination: np.order

`np.order(m, alpha, P, indices = (FALSE,TRUE))`

Description:

`np.order` For given values of m , α , and P , this function solves the necessary sample size such that the r^{th} (or $(n - s + 1)^{th}$) order statistic is the $[(1 - \alpha)100\%, (P)100\%]$ lower (or upper) tolerance limit. This function can also report all combinations of order statistics for 2-sided intervals.

Sample Size Determination: np.order

Examples:

① Sample Size Only:

`np.order(m = 5, alpha = 0.05, P = 0.95, indices = FALSE)`

② Sample Size With Indices:

`np.order(m = 5, alpha = 0.05, P = 0.95, indices = TRUE)`

Sample Size Determination: np.order

```
## Only requesting the sample size.
np.order(m = 5, alpha = 0.05, P = 0.95, indices = FALSE)
[1] 181

#####

## Requesting the order statistics indices as well.
np.order(m = 5, alpha = 0.05, P = 0.95, indices = TRUE)
$'Sample Size'
[1] 181

$'Order Statistics'
      [,1] [,2]
[1,]    1  178
[2,]    2  179
[3,]    3  180
[4,]    4  181
```

Nonparametric Tolerance Interval: Application

Hamedani (2013) provided an application of survival times of 121 patients with breast cancer obtained from a large hospital in a period from 1929 to 1938.

Suppose we do not wish to impose any distributional assumption on the data. If tolerance limits are desired, nonparametric and therefore `nptol.int` can be applied under this situation.

```
> # (0.90,0.90) 2-sided nonparametric tolerance interval (Using Wilks's Method) #
> remission.WILKS <- nptol.int(x=remission, alpha=0.10, P=0.90, side=2, method="WILKS")
> remission.WILKS
      alpha    P 2-sided.lower 2-sided.upper
V14    0.1 0.9             5         129
#####
> # (0.90,0.90) 2-sided nonparametric tolerance interval (Using the Young and Mathew's Method)
> remission.YM <- nptol.int(x=remission, alpha=0.10, P=0.90, side=2, method="YM")
> remission.YM
      alpha    P 2-sided.lower 2-sided.upper
OS-Based 0.1 0.9             5         128.1566
```

Nonparametric Tolerance Interval: Application

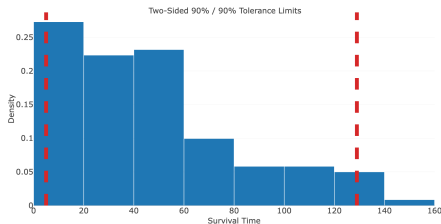


Figure: "WILKS"'s Method

The 2-sided $(P, \gamma) = (0.90, 0.90)$ nonparametric tolerance limits using "WILKS"'s method is $(5, 129)$.

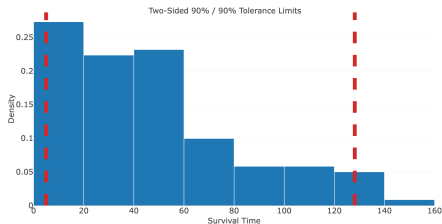


Figure: "YM"'s Method

The 2-sided $(P, \gamma) = (0.90, 0.90)$ nonparametric tolerance limits using "YM"'s interpolation/extrapolation scheme is $(5, 128.157)$.

Regression Tolerance Interval

Regression Tolerance Intervals

- ① Introduction
- ② (Multiple) Linear Regression Tolerance Intervals: $[\text{regtol.int}/\text{regtol.int2}]$
- ③ Nonlinear Regression Tolerance Interval: $[\text{nlregtol.int}/\text{nlregtol.int2}]$
- ④ Nonparametric Regression Tolerance Interval: $[\text{npregtol.int}/\text{npregtol.int2}]$
- ⑤ Application

Regression Tolerance Interval: Introduction

`tolerance` provides tools calculating tolerance limits for not only various univariate data, but also multivariate data and for regression settings.

More specifically, `tolerance` includes three functions `regtol.int`, `nlregtol.int` and `npregtol.int` for linear regression, nonlinear regression and nonparametric regression tolerance limits, respectively.

In addition, `plottol` and its updated version `plotly.plottol.reg` provides visualizations for tolerance limits under regression settings.

Linear Regression Tolerance Interval: `regtol.int`

```
regtol.int(reg, new.x, side = c(1,2), alpha, P)
```

Description:

`regtol.int` provides 1-sided or 2-sided (multiple) linear regression tolerance bounds. It is also possible to fit a regression through the origin model.

Note: we also update the original function to `regtol.int2`. `regtol.int2` reorganizes the output. In addition, a new argument `new` is added. When `new=FALSE`, `regtol.int2=regtol.int`. In the following, `regtol.int2` is used for demonstration purpose.

Linear Regression Tolerance Interval: regtol.int

Methodology:

A multiple linear regression model is defined as

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_{p-1} X_{p-1} + \varepsilon$$

where $\beta_0, \beta_1, \dots, \beta_{p-1}$ are the p regressors, and $\varepsilon \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2)$.

The one-sided (P, γ) regression tolerance limits for each observation i are given by

$$L = \hat{y}_i - \hat{\sigma} k_{1,i} \quad ; \quad U = \hat{y}_i + \hat{\sigma} k_{1,i}$$

where

$$k_{1,i} = \frac{t_{(n-p);(1-\alpha)}^*(\sqrt{n_i^*} z_P^*)}{\sqrt{n_i^*}}$$

and $t_{d;(1-\alpha)}^*(\gamma)$ is the $(1 - \alpha)^{th}$ quantile of a non-central t distribution with d degree of freedom and non-centrality parameter γ , z_P^* is the P^{th} quantile of a standard normal distribution. In addition $n_i^* = \frac{\hat{\sigma}^2}{s.e.(\hat{y}_i)^2}$ such that $s.e.(\hat{y}_i)$ is the standard error of \hat{y}_i .

Linear Regression Tolerance Interval: `regtol.int`

Methodology:

The two-sided (P, γ) regression tolerance limits for each observation i are given by

$$L = \hat{y}_i - \hat{\sigma}k_{2,i} \quad ; \quad U = \hat{y}_i + \hat{\sigma}k_{2,i}$$

where $k_{2,i}$ is estimated according to the formula in [Krishnamoorthy and Mathew \(2009\)](#) such that let $f = n - p$ then

$$k_{2,i} = \sqrt{\frac{f\chi_{1;P}^2(\frac{1}{n_i^*})}{\chi_{f;\alpha}^2}}$$

where $\chi_{d;\alpha}^2(\delta)$ is the α^{th} quantile of a non-central χ^2 distribution with d degree of freedom and non-centrality parameter δ .

Linear Regression Tolerance Interval: `regtol.int2`

Examples:

❶ One-Sided (without new data):

```
regtol.int2(reg = lm( $y \sim x$ ), side = 1, alpha = 0.05, P = 0.95)
```

❷ Two-Sided (with new data):

```
regtol.int2(reg = lm( $y \sim x$ ), new.x = data.frame(x = c(3, 6, 12)),  
side = 2, alpha = 0.05, P = 0.95)
```

Linear Regression Tolerance Interval: regtol.int2

```
set.seed(100)
x <- runif(100, 0, 10)
y <- 20 + 5*x + rnorm(100, 0, 3)

out1 <- regtol.int2(reg = lm(y ~ x),
                    side = 1, alpha = 0.05, P = 0.95 , new=TRUE)
out2 <- regtol.int2(reg = lm(y ~ x), new.x = data.frame(x = c(3, 6, 12)),
                    side = 2, alpha = 0.05, P = 0.95 , new=TRUE)

out1
out2
```


Linear Regression Tolerance Interval: `regtol.int2`

`regtol.int` versus `regtol.int2`

`regtol.int`

- ❶ One output
 $(\alpha, P, y, \hat{y}, TIs)$
- ❷ Redundant information of (α, P)
- ❸ Ordered by $\hat{y}'s$
- ❹ New data are ordered by estimated \hat{y}
- ❺ Does not include `regtol.int2`

`regtol.int2`

- ❶ Multiple outputs
 $(\$tol, \$\alpha.P.side, \$model, \$newdata)$
- ❷ More informative
- ❸ Ordered by original data
(visualization friendly)
- ❹ New data are attached to the original data at the end
- ❺ Includes `regtol.int`

Linear Regression Tolerance Interval: regtol.int2



Figure: 1-Sided

```
plotly.plottol.reg(out1 , x=x , y=y , side = "two",  
  x.cex = 22 , tol.lwd = 18, fit.lwd = 18,  
  fit.line.type = "dash", tol.line.type = "dot",  
  x.tick.size = 36 , x.lab.size = 36,  
  y.tick.size = 36 , y.lab.size = 36,  
  title.size = 36 , title.position.y = 0.995)
```

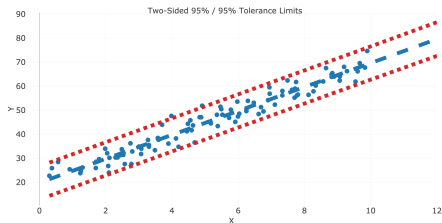


Figure: 2-Sided (with new data)

```
plotly.plottol.reg(out2 , x=x , y=y , side = "two",  
  x.cex = 22 , tol.lwd = 18, fit.lwd = 18,  
  fit.line.type = "dash", tol.line.type = "dot",  
  x.tick.size = 36 , x.lab.size = 36,  
  y.tick.size = 36 , y.lab.size = 36,  
  title.size = 36 , title.position.y = 0.995)
```

Nonlinear Regression Tolerance Interval: `nlregtol.int`

```
nlregtol.int(formula, xy.data, x.new = NULL, side = c(1,2), alpha, P,  
maxiter = 50, ...)
```

Description:

`nlregtol.int` provides 1-sided or 2-sided nonlinear regression tolerance bounds.

Note:

- ① “...” are optional arguments passed to `nls` when estimating the nonlinear regression equation.
- ② we also update the original function to `nlregtol.int2`. `nlregtol.int2` reorganizes the output. In addition, a new argument `new` is added. When `new=FALSE`, `nlregtol.int2=nlregtol.int`. In the following, `nlregtol.int2` is used for demonstration purpose.

Nonlinear Regression Tolerance Interval: nlregtol.int2

Methodology:

A nonlinear regression model is used to model the nonlinear relationship between a response variable Y with a given set of predictor variable X_1, \dots, X_p . The nonlinear regression model is defined as

$$Y = f(\beta, X_1, \dots, X_p) + \varepsilon$$

where β is a vector of regression parameters and ε is an error term following a specified distribution which is not necessarily normal.

Estimation of \hat{y}_i 's is done through nonlinear square routine.

Nonlinear Regression Tolerance Interval: `nlregtol.int2`

Methodology:

The (P, γ) nonlinear regression tolerance limits are constructed in a similar manner as for the linear regression case. The only difference is how the effective sample size n_i^* is calculated.

For the nonlinear setting, n_i^* is a function of the partial derivatives of $f(\beta, x_{i,1}, \dots, x_{i,p})$ with respect to each of the regression parameters.

Further technical details can be found in [Wallis \(1951\)](#).

Nonlinear Regression Tolerance Interval: `nlregtol.int2`

Examples:

① One-Sided (without new data):

```
nlregtol.int2(formula = formula, xy.data = data.frame(cbind(y,  
x)), side = 1, alpha = 0.05, P = 0.95 , new = TRUE)
```

② Two-Sided (with new data):

```
nlregtol.int2(formula = formula, xy.data = data.frame(cbind(y,  
x)), x.new=cbind(c(10, 55)), side = 2, alpha = 0.05, P = 0.95 ,  
new = TRUE)
```

Nonlinear Regression Tolerance Interval: nlregtol.int2

```
set.seed(100)
x <- runif(50, 5, 45)
f1 <- function(x, b1, b2) b1 + (0.49 - b1)*exp(-b2*(x - 8)) +
  rnorm(50, sd = 0.01)
y <- f1(x, 0.39, 0.11)
formula <- as.formula(y ~ b1 + (0.49 - b1)*exp(-b2*(x - 8)))
out1 <- nlregtol.int2(formula = formula,
  xy.data = data.frame(cbind(y, x)), side = 1,
  alpha = 0.05, P = 0.95 , new = TRUE)
out2 <- nlregtol.int2(formula = formula,
  xy.data = data.frame(cbind(y, x)),
  x.new=cbind(c(10, 55)), side = 2,
  alpha = 0.05, P = 0.95 , new = TRUE)
```

Nonlinear Regression Tolerance Interval: `nlregtol.int2`

`nlregtol.int` versus `nlregtol.int2`

`nlregtol.int`

- ❶ One output
 $(\alpha, P, y, \hat{y}, TIs)$
- ❷ Redundant information of (α, P)
- ❸ Ordered by y 's
- ❹ Does not include `nlregtol.int2`

`nlregtol.int2`

- ❶ Multiple outputs
 $(\$tol, \$\alpha.P.side, \$model, \$original.data)$
- ❷ More informative
- ❸ Ordered by original data
(visualization friendly)
- ❹ Includes `nlregtol.int`

Nonlinear Regression Tolerance Interval: nlregtol.int2

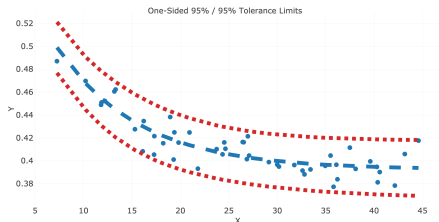


Figure: 1-Sided

```
plotly.plottol.reg(out1 , x=x , y=y , side = "two",  
  x.cex = 22 , tol.lwd = 18, fit.lwd = 18,  
  fit.line.type = "dash", tol.line.type = "dot",  
  x.tick.size = 36 , x.lab.size = 36,  
  y.tick.size = 36 , y.lab.size = 36,  
  title.size = 36 , title.position.y = 0.995)
```



Figure: 2-Sided (with new data)

```
plotly.plottol.reg(out2 , x=x , y=y , side = "two",  
  x.cex = 22 , tol.lwd = 18, fit.lwd = 18,  
  fit.line.type = "dash", tol.line.type = "dot",  
  x.tick.size = 36 , x.lab.size = 36,  
  y.tick.size = 36 , y.lab.size = 36,  
  title.size = 36 , title.position.y = 0.995)
```

Nonparametric Regression Tolerance Interval: `npregtol.int`

```
npregtol.int(x, y, y.hat, side = c(1,2), alpha, P, method =  
c("WILKS", "WALD", "HM"), upper = NULL, lower = NULL)
```

Description:

`npregtol.int` provides 1-sided or 2-sided nonparametric regression tolerance bounds.

Note: we also update the original function to `npregtol.int2`. `npregtol.int2` reorganizes the output. In addition, a new argument `new` is added. When `new=FALSE`, `npregtol.int2=npregtol.int`. In the following, `npregtol.int2` is used for demonstration purpose.

Nonparametric Regression Tolerance Interval: npregtol.int

Methodology:

A nonparametric regression model is used to model the nonlinear relationship between a response variable Y with a given set of predictor variables X_1, \dots, X_p but without parameters.

The nonparametric regression model is defined as

$$Y = f(X_1, \dots, X_p) + \varepsilon$$

which is free of any parameters and ε is a random error term which is only assumed to have mean 0.

Tolerance limits for nonparametric regression can be constructed based on order statistics of residuals, that is it is necessary to find $\hat{\varepsilon}_{[r]}$ and $\hat{\varepsilon}_{[s]}$. The corresponding (P, γ) upper and lower nonparametric regression tolerance limits are

$$L = \hat{y}_i + e_{[r]} \quad ; \quad U = \hat{y}_i + e_{[s]}$$

Method to determine r and s are given in *nonparametric tolerance intervals* section.

Nonparametric Regression Tolerance Interval: `npregtol.int2`

Examples:

❶ One-Sided (without limit specifications):

```
npregtol.int2(x = x, y = y, y.hat = y.hat, side = 1, alpha =  
0.05, P = 0.95, method = "WILKS",new=TRUE)
```

❷ Two-Sided (with limit specifications):

```
npregtol.int2(x = x, y = y, y.hat = y.hat, side = 2, alpha =  
0.05, P = 0.95, method = "WILKS",new=TRUE, lower = 0.38 ,  
upper=0.49)
```

Nonparametric Regression Tolerance Interval: npregtol.int2

```
set.seed(100)
x <- runif(50, 5, 45)
f1 <- function(x, b1, b2) b1+(0.49-b1)*exp(-b2*(x-8))+rnorm(50,sd = 0.01)
y <- f1(x, 0.39, 0.11)

y.hat <- loess(y~x)$fit

out1 <- npregtol.int2(x = x, y = y, y.hat = y.hat, side = 1,
                      alpha = 0.05, P = 0.95, method = "WILKS",new=TRUE)
out2 <- npregtol.int2(x = x, y = y, y.hat = y.hat, side = 2,
                      alpha = 0.05, P = 0.95, method = "WILKS",new=TRUE,
                      lower = 0.38 , upper=0.49)

out1
out2
```

Nonparametric Regression Tolerance Interval: `npregtol.int2`

`npregtol.int` versus `npregtol.int2`

`npregtol.int`

- ❶ One output
 $(\alpha, P, y, \hat{y}, TIs)$
- ❷ Redundant information of (α, P)
- ❸ Ordered by $\hat{y}'s$
- ❹ If a certain limit is set, it is hard to track purely based on output
- ❺ Does not include `npregtol.int2`

`npregtol.int2`

- ❶ Multiple outputs
 $(\$tol, \$\alpha.P.side, \$method, \$lower.upper)$
- ❷ More informative
- ❸ Ordered by original data
(visualization friendly)
- ❹ Specified limits are reminded in the output
- ❺ Includes `npregtol.int`

Nonparametric Regression Tolerance Interval: npregtol.int2

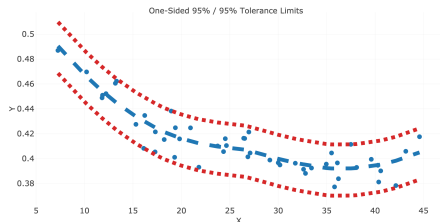


Figure: 1-Sided



Figure: 2-Sided (with specified limits)

```
plotly.plottol.reg(out1 , x=x , y=y , side = "two",  
  x.cex = 22 , tol.lwd = 18, fit.lwd = 18,  
  fit.line.type = "dash", tol.line.type = "dot",  
  x.tick.size = 36 , x.lab.size = 36,  
  y.tick.size = 36 , y.lab.size = 36,  
  title.size = 36 , title.position.y = 0.995)
```

```
plotly.plottol.reg(out2 , x=x , y=y , side = "two",  
  x.cex = 22 , tol.lwd = 18, fit.lwd = 18,  
  fit.line.type = "dash", tol.line.type = "dot",  
  x.tick.size = 36 , x.lab.size = 36,  
  y.tick.size = 36 , y.lab.size = 36,  
  title.size = 36 , title.position.y = 0.995)
```

Regression Tolerance Interval: Application

Hospitals frequently study factors that are likely related to infections patients gain while hospitalized. [Neter, Kutner, Nachtsheim, and Wasserman \(1996\)](#) provided patients' infection data from a study of $n = 113$ US hospitals.

Numerous variables are available to study, but a parsimonious multiple linear regression model is one that has infection risk as the response, and average length of patient stay and number of x-rays given by the hospital as predictors.

```
hospitals <- read.table("https://raw.githubusercontent.com/dsy109/
Supplemental/main/CSP%202022%20Trainings/tolerance/Datasets/hospitals.txt",
header = TRUE)
### Calculate 2-sided pointwise (0.90, 0.90) tolerance intervals for the multiple
### regression fit of the hospital infections data.
out <- lm(InfctRsk ~ Xray + Stay, data = hospitals)
out.TI <- regtol.int2(out, new.x = data.frame(Xray = c(50, 75), Stay = c(8, 12)),
alpha = 0.1, P = 0.9, side = 2, new = TRUE)
```

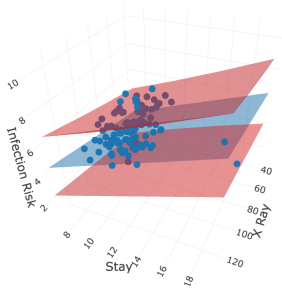

Regression Tolerance Interval: Application

The linear regression model in this application is

$$\text{Infection Risk} = \beta_0 + \beta_1 XRay + \beta_2 Stay + \varepsilon$$

For a dataset with 2 predictor regressors, a (P, γ) tolerance plane shall be generated.

Two-Sided 90% / 90% Tolerance Planes



Multivariate Tolerance Regions

Multivariate Tolerance Regions

- ① Introduction
- ② Multivariate Normal Tolerance Regions [mvtol.region]
- ③ Multivariate Linear Regression Tolerance Regions [mvregtol.region]
- ④ Application

Multivariate Tolerance Region: Introduction

Think of a situation where a quality engineer wishes to assess both the thickness and the diameter of a metal ring. Two elements are involved at the same time and they follow a bivariate (or multivariate) normal distribution approximately.

tolerance provides the option to calculate tolerance regions for multivariate normal distribution.

Methodology

A random vector $\mathbf{X} = (X_1, \dots, X_p)^T$ follows a multivariate normal distribution if it has the density function

$$f_{\mathbf{X}}(\mathbf{x}, \mu, \Sigma) = \frac{1}{(2\pi)^{p/2} |\Sigma|^{1/2}} e^{-(1/2) \|\Sigma^{-1/2}(\mathbf{x} - \mu)\|^2}$$

where $\mathbf{x} \in \mathbb{R}^p$, $\mu \in \mathbb{R}^p$, and Σ is a $p \times p$ symmetric and positive-definite matrix, and $\|\mathbf{A}\|^2 = \mathbf{A}^T \mathbf{A}$ for any matrix \mathbf{A} .

Normal tolerance regions require estimates of both μ and Σ , such that

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \quad ; \quad \hat{\Sigma} = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{x}_i - \hat{\mu})(\mathbf{x}_i - \hat{\mu})^T$$

A (P, γ) normal tolerance region is given by

$$\mathbf{x} : \|S^{-1/2}(\mathbf{x}_i - \hat{\mu})\|^2 < c$$

where c is the tolerance factor to be determined such that

$$\mathbb{P}[\mathbb{P}_{\mathbf{X}}\{\|S^{-1/2}(\mathbf{x}_i - \hat{\mu})\|^2 \leq c; \mathbf{x}, S\} \geq P] = 1 - \alpha$$

To obtain the tolerance region, obtaining c is a critical step. For more technical details of how to calculate c based on Monte Carlo algorithm, please see [Krishnamoorthy and Mondal \(2006\)](#).

Multivariate Tolerance Regions: `mvtol.region`

```
mvtol.region(x, alpha, P, method = c("KM", "AM", "GM", "HM", "MHM",  
"V11", "HM.V11", "MC", B, M))
```

Description:

`mvtol.region` determines the appropriate tolerance factor for computing multivariate normal tolerance regions based on Monte Carlo methods or other approximations.

Note:

- ▶ B is the number of iterations used for the Monte Carlo algorithms (i.e., when `method = c("KM", "MC")`), which determines the tolerance factor.
- ▶ M is the number of iterations used for the inner loop of the Monte Carlo algorithm specified through `method = "MC"`.

Multivariate Tolerance Regions: `mvtol.region`

Bivariate Data

```
## 90%/90% bivariate normal tolerance region.
set.seed(100)
x1 <- rnorm(100, 0, 0.2)
x2 <- rnorm(100, 0, 0.5)
x <- cbind(x1, x2)
out1 <- mvtol.region(x = x, alpha = 0.10, P = 0.90, B = 1000,
  method = "KM")
#####
> out1
      0.1
0.9 5.506321
```

Trivariate Data

```
## 90%/90% trivariate normal tolerance region.
set.seed(100)
x1 <- rnorm(100, 0, 0.2)
x2 <- rnorm(100, 0, 0.5)
x3 <- rnorm(100, 5, 1)
x <- cbind(x1, x2, x3)
mvtol.region(x = x, alpha = c(0.10, 0.05, 0.01),
  P = c(0.90, 0.95, 0.99), B = 1000, method = "KM")
out2 <- mvtol.region(x = x, alpha = 0.10, P = 0.90, B = 1000,
  method = "KM")
#####
> out2
      0.1
0.9 7.389876
```


Multivariate Tolerance Regions: mvtool.region

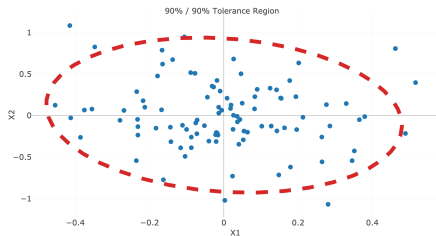


Figure: Bivariate

```
plotly.plottol.multi(out1, x , x.lab = "X1" , y.lab = "X2",  
  x.cex=22,tol.lwd=18,tol.line.type="dot",  
  x.tick.size = 36 , x.lab.size = 36,  
  y.tick.size = 36 , y.lab.size = 36,  
  title.size = 36 , title.position.y = 0.995)
```

90% Tolerance Region
90% Tolerance Region

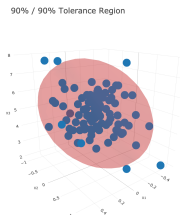


Figure: Trivariate

```
library(rgl)  
plotly.plottol.multi(out2, x , x.cex = 22 , tol.lwd = 18,  
  x.lab = "X1" , y.lab = "X2" , z.lab = "X3",  
  title.size = 36 , title.position.y = 0.995)
```

Multivariate Regression Tolerance Regions: `mvregtol.region`

```
mvregtol.region(mvreg, new.x, alpha, P, B))
```

Description:

`mvregtol.region` determines the appropriate tolerance factor for computing multivariate (multiple) linear regression tolerance regions based on Monte Carlo simulation.

Note:

- B The number of iterations used for the Monte Carlo algorithm which determines the tolerance factor. The number of iterations should be at least as large as the default value of 1000.

Multivariate Regression Tolerance Regions: mvregtol.region

Example:

```
## 95%/95% multivariate regression tolerance factors using
## a fertilizer data set presented in Anderson (2003, p. 374).
grain <- c(40, 17, 9, 15, 6, 12, 5, 9)
straw <- c(53, 19, 10, 29, 13, 27, 19, 30)
fert <- c(24, 11, 5, 12, 7, 14, 11, 18)
DF <- data.frame(grain,straw,fert)
new.x <- data.frame(fert = c(10, 15, 20))
mvreg <- lm(cbind(grain, straw) ~ fert + I(fert^2), data = DF)
set.seed(100)
out <- mvregtol.region(mvreg, new.x = new.x, alpha = 0.05,
                       P = 0.95, B = 5000)

out
```

Multivariate Regression Tolerance Regions: mvregtol.region

Output:

> out

	k.factor	grain.hat	straw.hat	fert	I(fert^2)
X1	24.962794	36.753376	51.34292	24	576
X2	8.748968	8.835535	20.31933	11	121
X3	17.032422	10.200785	10.69490	5	25
X4	8.845872	9.483019	22.21164	12	144
X5	10.076046	8.745673	13.57363	7	49
X6	9.184379	11.528007	26.24332	14	196
X7	8.791323	8.835535	20.31933	11	121
X8	9.725227	18.618069	35.29492	18	324
X1.1	8.569849	8.438059	18.50938	10	100
X2.1	9.468488	12.925512	28.38269	15	225
X3.1	10.514037	23.663143	40.31484	20	400

Multivariate Regression Tolerance Regions: Application

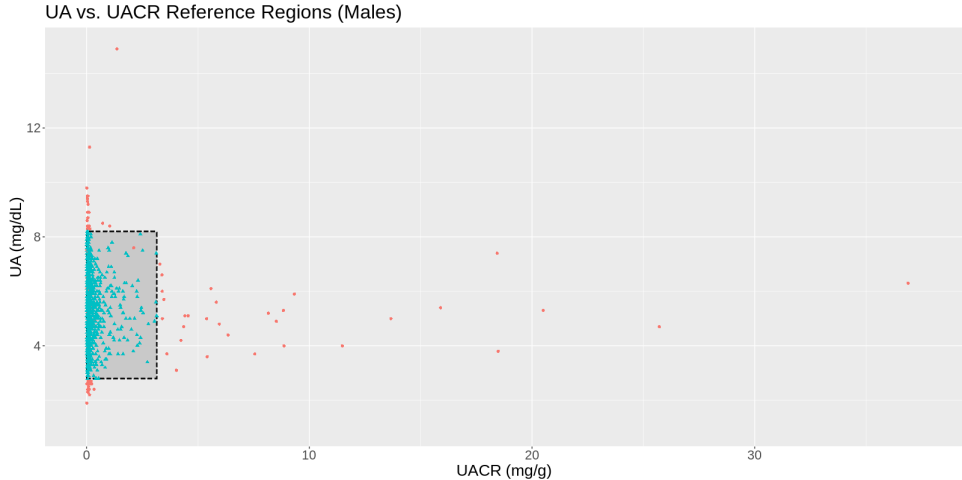
- ① Kidney function laboratory tests include a urinalysis to screen for the presence of protein and blood in the urine, a blood urea nitrogen (BUN) test to check for waste product in the urine, and a test to obtain the estimated glomerular filtration rate (eGFR), which is used to detect the presence and cause of kidney disease.
- ② Little information is available regarding normal reference values for kidney function in adolescents, which impacts how physicians diagnose and manage diabetes in this population.
- ③ The reference population studied is healthy US adolescents between 12 and 17 years of age, with a number of criteria used to determine “healthy”.
- ④ We will construct nonparametric $(P, \gamma) = (0.95, 0.95)$ semi-space rectangular tolerance regions to represent the reference regions of normal adolescent kidney function; we will look at the males in this sample, yielding $n = 2529$ subjects in the reference sample.

Multivariate Regression Tolerance Regions: Application

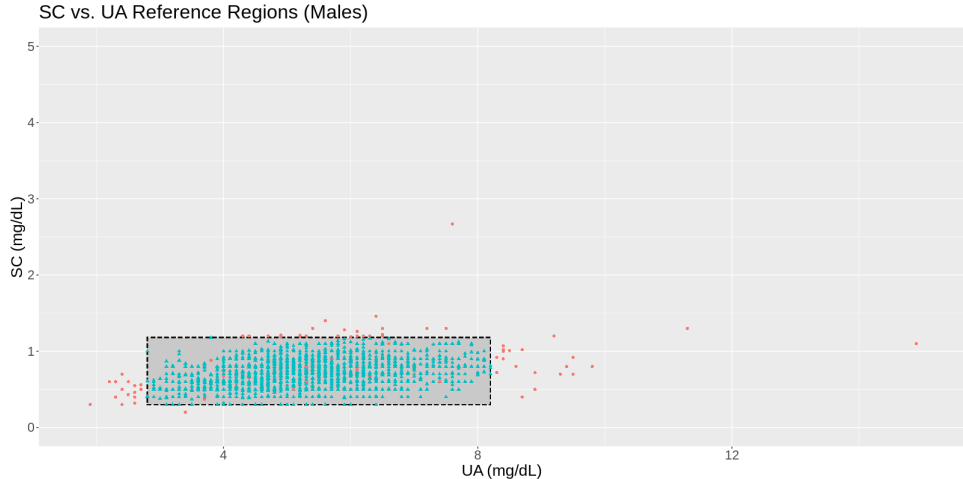
Methodology: In this application, a multivariate tolerance region is constructed using hyper-rectangle method developed by [Young and Mathew \(2020\)](#). An overview of the method is illustrated as follows:

- ① Calculate the data depth value of each observation;
- ② Trim the observations based on their data-depth value to capture at least a proportion of P sampled population at $(1 - \alpha)$ confidence level.
- ③ Obtain the rectangle(s) which include(s) the inclusive observation to obtain the (P, γ) tolerance region. The region is expected to be in a hyper-rectangle shape.

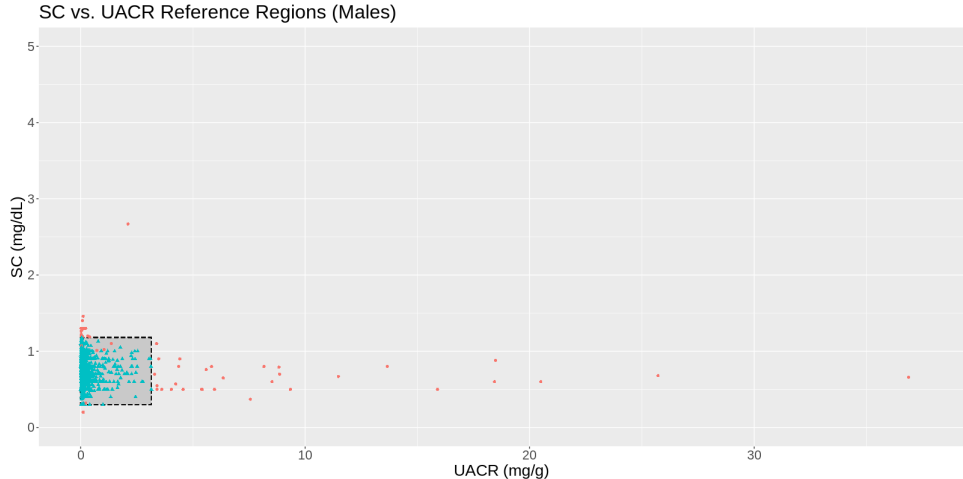
Multivariate Regression Tolerance Regions: Application



Multivariate Regression Tolerance Regions: Application



Multivariate Regression Tolerance Regions: Application



Final Comments

Some Other Applications Using tolerance

- ▶ Used to construct tolerance intervals for establishing a pass/fail criterion of radiation portal monitors. [Burr and Gavron (2012), "Pass/Fail Criterion for a Simple Radiation Portal Monitor Test ." *Modern Instrumentation*; 1(3):27–33.]
- ▶ Used to help the design verification process of the Vagus Nerve Stimulation (VNS) Therapy system, which is FDA-approved for the treatment of refractory epilepsy and treatment-resistant depression. [Young et al. (2016), "Sample Size Determination Strategies for Normal Tolerance Intervals Using Historical Data." *Quality Engineering*; 28(3):337–351.]
- ▶ Used in molecular lymph node analysis to aid in the understanding of prostate cancer. [Heck et al. (2018), "Molecular Lymph Node Status for Prognostic Stratification of Prostate Cancer Patients Undergoing Radical Prostatectomy with Extended Pelvic Lymph Node Dissection." *Clinical Cancer Research*; 24(10):2342–2349.]
- ▶ Used to develop nonparametric and normal-based tolerance intervals on GH-2000 to detect doping in competitive sports. [Liu et al. (2021), "Comparison of Normal Distribution-Based and Nonparametric Decision Limits on the GH-2000 Score for Detecting Growth Hormone Misuse (Doping) in Sport." *Biometrical Journal*; 63(1):187–200.]

Recent and Forthcoming Additions to tolerance

Recent Additions

- ▶ Launched GitHub repo (<https://github.com/dsy109/tolerance>)
- ▶ Launched Shiny app (<https://tolerance.as.uky.edu/>)
- ▶ Added new graphics capabilities based on ggplot2 and plotly

Forthcoming Additions

- ▶ Improve efficiency of exact normal k -factor calculations using Rcpp
- ▶ Expanding the suite of regression-based tolerance intervals; e.g., simultaneous tolerance intervals and spatial regression tolerance regions
- ▶ Recording a training module; make available on my website

Open Discussion

- ▶ As current (or future) end-users of the tolerance package, what are features that would be helpful for your research and data analysis needs?

References I

- Bain, L. J., & Engelhardt, M. (1981). Simple approximate distributional results for confidence and tolerance limits for the weibull distribution based on maximum likelihood estimators. *Technometrics*, 23(1), 15–20.
- Brown, L. D., Cai, T. T., & DasGupta, A. (2001). Interval estimation for a binomial proportion. *Statistical science*, 16(2), 101–133.
- Cameron, A. C., & Trivedi, P. K. (1986). Econometric models based on count data. comparisons and applications of some estimators and tests. *Journal of applied econometrics*, 1(1), 29–53.
- Ellison, B. E. (1964). On two-sided tolerance intervals for a normal distribution. *The Annals of Mathematical Statistics*, 35(2), 762–772.
- Faulkenberry, G. D., & Weeks, D. L. (1968). Sample size determination for tolerance limits. *Technometrics*, 10(2), 343–348.
- Hahn, G. J., & Chandra, R. (1981). Tolerance intervals for poisson and binomial variables. *Journal of Quality Technology*, 13(2), 100–110.

References II

- Hahn, G. J., & Meeker, W. Q. (2011). *Statistical intervals: a guide for practitioners* (Vol. 92). John Wiley & Sons.
- Hamedani, G. (2013). The zografos-balakrishnan log-logistic distribution: Properties and applications. *Journal of Statistical Theory and Applications*.
- Harris, E. K., & Boyd, J. C. (1995). *Statistical bases of reference values in laboratory medicine*. CRC Press.
- Howe, W. (1969). Two-sided tolerance limits for normal populations—some improvements. *Journal of the American Statistical Association*, 64(326), 610–620.
- Krishnamoorthy, K., & Mathew, T. (2009). *Statistical Tolerance Regions: Theory, Applications, and Computation*. Hoboken, NJ: Wiley.
- Krishnamoorthy, K., Mathew, T., & Mukherjee, S. (2008). Normal-based methods for a gamma distribution: prediction and tolerance intervals and stress-strength reliability. *Technometrics*, 50(1), 69–78.

References III

- Krishnamoorthy, K., & Mondal, S. (2006). Improved tolerance factors for multivariate normal distributions. *Communications in Statistics-simulation and Computation*, 35(2), 461–478.
- Lee, E. T., & Wang, J. (2003). *Statistical methods for survival data analysis* (Vol. 476). John Wiley & Sons.
- Mee, R. W. (1990). Simultaneous tolerance intervals for normal populations with, common variance. *Technometrics*, 32(1), 83–92.
- Montgomery, D. C. (2020). *Introduction to statistical quality control*. John Wiley & Sons.
- Neter, J., Kutner, M. H., Nachtsheim, C. J., & Wasserman, W. (1996). Applied linear statistical models.
- Owen, D. B. (1964). Control of percentages in both tails of the normal distribution. *Technometrics*, 6(4), 377–387.

References IV

- Wald, A. (1943). An Extension of Wilks' Method for Setting Tolerance Limits. *The Annals of Mathematical Statistics*, 14, 45–55.
- Wald, A., & Wolfowitz, J. (1946). Tolerance limits for a normal distribution. *The Annals of Mathematical Statistics*, 208–215.
- Wallis, W. A. (1951). Tolerance intervals for linear regression. In *Proceedings of the second berkeley symposium on mathematical statistics and probability* (pp. 43–51).
- Weissberg, A., & Beatty, G. H. (1960). Tables of tolerance-limit factors for normal distributions. *Technometrics*, 2(4), 483–500.
- Wilks, S. S. (1941). Determination of Sample Sizes for Setting Tolerance Limits. *The Annals of Mathematical Statistics*, 12, 91–96.
- Young, D. S. (2010). Tolerance: an r package for estimating tolerance intervals. *Journal of Statistical Software*, 36(1), 1–39.

References V

- Young, D. S., Gordon, C. M., Zhu, S., & Olin, B. D. (2016). Sample size determination strategies for normal tolerance intervals using historical data. *Quality Engineering*, 28(3), 337–351.
- Young, D. S., & Mathew, T. (2014). Nonparametric Tolerance Intervals Based on Interpolated and Extrapolated Order Statistics. *Journal of Nonparametric Statistics*, 26(3), 415–432.
- Young, D. S., & Mathew, T. (2020). Nonparametric Hyperrectangular Tolerance and Prediction Regions for Setting Multivariate Reference Regions in Laboratory Medicine. *Statistical Methods in Medical Research*, 29(12), 3569–3585.

Contact Information

 derek.young@uky.edu

 <http://young.as.uky.edu>

 <https://github.com/dsy109>