Package ‘tolerance’

June 29, 2009

Type Package

Title Functions for calculating tolerance intervals.

Version 0.1.0

Date 2009-04-07

Author Derek S. Young

Maintainer Derek S. Young <dsy109@stat.psu.edu>

Depends rgl

Description Tolerance limits provide the limits between which we can expect to find a specified proportion of a population with a given level of confidence. This package provides functions for estimating tolerance limits for various distributions. Plotting is also available for tolerance limits of continuous random variables.

License GPL (>= 2)

Repository CRAN

Date/Publication 2009-06-29 06:41:15

R topics documented:

`tolerance-package` *Functions for Calculating Tolerance Intervals*

Description

A collection of functions for calculating `[100(1-alpha)%, 100(P)%]` tolerance intervals, which are intervals with `100(1-alpha)%` confidence of covering `100(P)%` of the population of interest.

Details

Package:	tolerance
Type:	Package
Version:	0.1.0
Date:	2009-04-07
Depends	rgl
License:	GPL (>= 2)

Author(s)

Derek S. Young, Ph.D.

Maintainer: Derek S. Young <dsy109@stat.psu.edu>

References

Hahn, G. J. and Meeker, W. Q. (1991), *Statistical Intervals: A Guide for Practitioners*, Wiley-Interscience.

Patel, J. K. (1986), Tolerance Intervals - A Review, *Communications in Statistics - Theory and Methodology*, **15**, 2719–2762.

`acc.samp` *Acceptance Sampling*

Description

Provides an upper bound on the number of acceptable rejects or nonconformities in a process. This is similar to a 1-sided upper tolerance bound for a hypergeometric random variable.

Usage

```
acc.samp(n, N, alpha = 0.05, P = 0.99, AQL)
```

Arguments

- `n` The sample size to be drawn from the inventory.
- `N` The total inventory (or lot) size.
- `alpha` `1-alpha` is the confidence level for bounding the probability of accepting the inventory.
- `P` The proportion of items in the inventory which are to be accountable.
- `AQL` The acceptable quality level, which is the largest proportion of defects in a process considered acceptable.

Value

`acc.samp` returns a data frame with items:

- `acceptance.limit` The number of items in the sample which may be unaccountable, yet still be able to attain the desired confidence level `1-alpha`.
- `lot.size` The total inventory (or lot) size `N`.
- `RQL` The rejectable quality level. This is the proportion of individual items in a sample one is willing to tolerate missing (i.e., this is `1-P`).
- `confidence` The confidence level `1-alpha`.
- `AQL` The acceptable quality level. If the sampling were to be repeated numerous times as a process, then this quantity specifies the proportion of missing items considered acceptable from the process as a whole.
- `sample.size` The sample size drawn as specified by `n`.
- `prod.risk` The producer's risk. This is the probability of rejecting an audit of a good inventory (also called the Type I error). A good inventory can be rejected if an unfortunate random sample is selected (e.g., most of the missing items happened to be selected for the audit).

cons.risk The consumer's risk. This is the probability of accepting an audit of a bad inventory (also called the Type II error). A bad inventory can be accepted if a fortunate random sample (e.g., most of the missing items happend to not be selected for the audit). `1-cons.risk` gives the actual confidence level of this sampling plan. If it is lower than the confidence level desired (i.e., because too small a sample size was specified), then a warning message will be displayed.

References

Montgomery, D. C. (2005), *Introduction to Statistical Quality Control*, Fifth Edition, John Wiley & Sons, Inc.

See Also

Hypergeometric

Examples

```
## A 90%/90% acceptance sampling plan for a sample of 450
## drawn from a lot size of 960.

acc.samp(n = 450, N = 960, alpha = 0.10, P = 0.90, AQL = 0.01)
```

---

bintol.int *Binomial Tolerance Intervals*

Description

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.

Usage

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

Arguments

x The number of defective (or acceptable) units in the sample.

n The size of the random sample of units selected for inspection.

m The quantity produced in future groups.

alpha The level chosen such that `1-alpha` is the confidence level.

P — The proportion of the defective (or acceptable) units in future samples of size `m` to be covered by this tolerance interval.

side — Whether a 1-sided or 2-sided tolerance interval is required (determined by `side = 1` or `side = 2`, respectively).

method — The method for calculating the lower and upper confidence bounds, which are used in the calculation of the tolerance bounds. The default method is `"LS"`, which is the large-sample method. `"AC"` gives the Agresti-Coull method, which is also appropriate when the sample size is large. `"JF"` is Jeffreys' method, which is a Bayesian approach to the estimation. `"CP"` is the Clopper-Pearson method, which provides a more conservative interval. `"AS"` is the arcsine method, which is appropriate when the sample proportion is not too close to 0 or 1. `"LO"` is the logit method, which also is appropriate when the sample proportion is not too close to 0 or 1, but yields a more conservative interval. More information on these methods can be found in the "References".

a1 — This specifies the first shape hyperparameter when using Jeffreys' method.

a2 — This specifies the second shape hyperparameter when using Jeffreys' method.

Value

`bintol.int` returns a data frame with items:

alpha — The specified significance level.

P — The proportion of defective (or acceptable) units in future samples of size `m`.

p.hat — The proportion of defective (or acceptable) units in the sample, calculated by `x/n`.

1-sided.lower — The 1-sided lower tolerance bound. This is given only if `side = 1`.

1-sided.upper — The 1-sided upper tolerance bound. This is given only if `side = 1`.

2-sided.lower — The 2-sided lower tolerance bound. This is given only if `side = 2`.

2-sided.upper — The 2-sided upper tolerance bound. This is given only if `side = 2`.

References

Brown, L. D., Cai, T. T., and DasGupta, A. (2001), Interval Estimation for a Binomial Proportion, *Statistical Science*, **16**, 101–133.

Hahn, G. J. and Chandra, R. (1981), Tolerance Intervals for Poisson and Binomial Variables, *Journal of Quality Technology*, **13**, 100–110.

See Also

Binomial

Examples

```
## 85%/90% 1-sided binomial tolerance limits for a future
## lot of 500 when a sample of 40 were drawn from a lot of
## 1000.  The Agresti-Coull, Clopper-Pearson, and large-sample
## methods are presented for comparison.

bintol.int(x = 40, n = 1000, m = 500, alpha = 0.15, P = 0.90,
           side = 1, method = "AC")
bintol.int(x = 40, n = 1000, m = 500, alpha = 0.15, P = 0.90,
           side = 1, method = "CP")
bintol.int(x = 40, n = 1000, m = 500, alpha = 0.15, P = 0.90,
           side = 1, method = "LS")
```

---

cautol.int *Cauchy Tolerance Intervals*

Description

Provides 1-sided or 2-sided tolerance intervals for Cauchy distributed data.

Usage

```
cautol.int(x, alpha = 0.05, P = 0.99, side = 1)
```

Arguments

- `x` — A vector of data which is Cauchy distributed.
- `alpha` — The level chosen such that `1-alpha` is the confidence level.
- `P` — The proportion of the population to be covered by this tolerance interval.
- `side` — Whether a 1-sided or 2-sided tolerance interval is required (determined by `side = 1` or `side = 2`, respectively).

Value

`cautol.int` returns a data.frame with items:

- `alpha` — The specified significance level.
- `P` — The proportion of the population covered by this tolerance interval.
- `1-sided.lower` — The 1-sided lower tolerance bound. This is given only if `side = 1`.
- `1-sided.upper` — The 1-sided upper tolerance bound. This is given only if `side = 1`.
- `2-sided.lower` — The 2-sided lower tolerance bound. This is given only if `side = 2`.
- `2-sided.upper` — The 2-sided upper tolerance bound. This is given only if `side = 2`.

References

Bain, L. J. (1978), *Statistical Analysis of Reliability and Life-Testing Models*, Marcel Dekker, Inc.

See Also

Cauchy

Examples

```
## 95%/90% 2-sided Cauchy tolerance interval for a sample
## of size 1000.

set.seed(100)
x <- rcauchy(1000, 100000, 10)
out <- cautol.int(x = x, alpha = 0.05, P = 0.90, side = 2)
out

plottol(out, x, plot.type = "both", x.lab = "Cauchy Data")
```

---

distfree.est — *Estimating Various Quantities for Distribution-Free Tolerance Intervals*

Description

When providing two of the three quantities `n`, `alpha`, and `P`, this function solves for the third quantity in the context of distribution-free tolerance intervals.

Usage

```
distfree.est(n = NULL, alpha = NULL, P = NULL, side = 1)
```

Arguments

`n`	The necessary sample size to cover a proportion `P` of the population with confidence `1-alpha`. Can be a vector.
`alpha`	1 minus the confidence level attained when it is desired to cover a proportion `P` of the population and a sample size `n` is provided. Can be a vector.
`P`	The proportion of the population to be covered with confidence `1-alpha` when a sample size `n` is provided. Can be a vector.
`side`	Whether a 1-sided or 2-sided tolerance interval is assumed (determined by `side = 1` or `side = 2`, respectively).

Value

When providing two of the three quantities `n`, `alpha`, and `P`, `distfree.est` returns the third quantity. If more than one value of a certain quantity is specified, then a table will be returned.

References

Natrella, M. G. (1963), *Experimental Statistics: National Bureau of Standards - Handbook No. 91*, United States Government Printing Office, Washington, D.C.

See Also

nptol.int

Examples

```
## Solving for 1 minus the confidence level.
distfree.est(n = 59, P = 0.95, side = 1)

## Solving for the sample size.
distfree.est(alpha = 0.05, P = 0.95, side = 1)

## Solving for the proportion of the population to cover.
distfree.est(n = 59, alpha = 0.05, side = 1)

## Solving for sample sizes for many tolerance specifications.
distfree.est(alpha = seq(0.01, 0.05, 0.01),
             P = seq(0.80, 0.99, 0.01), side = 2)
```

---

exp2tol.int *2-Parameter Exponential Tolerance Intervals*

Description

Provides 1-sided or 2-sided tolerance intervals for data distributed according to a 2-parameter exponential distribution. Data with Type II censoring is permitted.

Usage

```
exp2tol.int(x, alpha = 0.05, P = 0.99, side = 1,
             method = c("GPU", "DUN"), type.2 = FALSE)
```

Arguments

x	A vector of data which is distributed according to the 2-parameter exponential distribution.
alpha	The level chosen such that `1-alpha` is the confidence level.
P	The proportion of the population to be covered by this tolerance interval.
side	Whether a 1-sided or 2-sided tolerance interval is required (determined by `side = 1` or `side = 2`, respectively).

`method` The method for how the upper tolerance bound is approximated. `"GPU"` is the Guenther-Patil-Upppuluri method. `"DUN"` is the Dunsmore method, which was empirically shown to be an improvement for samples greater than or equal to 8. More information on these methods can be found in the "References".

`type.2` Select `TRUE` if Type II censoring is present (i.e., the data set is censored at the maximum value present). The default is `FALSE`.

Value

`exp2tol.int` returns a data frame with items:

`alpha` The specified significance level.

`P` The proportion of the population covered by this tolerance interval.

`1-sided.lower` The 1-sided lower tolerance bound. This is given only if `side = 1`.

`1-sided.upper` The 1-sided upper tolerance bound. This is given only if `side = 1`.

`2-sided.lower` The 2-sided lower tolerance bound. This is given only if `side = 2`.

`2-sided.upper` The 2-sided upper tolerance bound. This is given only if `side = 2`.

References

Dunsmore, I. R. (1978), Some Approximations for Tolerance Factors for the Tow Parameter Exponential Distribution, *Technometrics*, **20**, 317–318.

Engelhardt, M. and Bain, L. J. (1978), Tolerance Limits and Confidence Limits on Reliability for the Two-Parameter Exponential Distribution, *Technometrics*, **20**, 37–39.

Guenther, W. C., Patil, S. A., and Uppuluri, V. R. R. (1976), One-Sided β-Content Tolerance Factors for the Two Parameter Exponential Distribution, *Technometrics*, **18**, 333–340.

See Also

`TwoParExponential`

Examples

```
## 95%/90% 1-sided 2-parameter exponential tolerance intervals
## for a sample of size 50.

set.seed(100)
x <- r2exp(50, 6, shift = 55)
out <- exp2tol.int(x = x, alpha = 0.05, P = 0.90, side = 1,
                   method = "DUN", type.2 = FALSE)
out

plottol(out, x, plot.type = "both", side = "upper",
        x.lab = "2-Parameter Exponential Data")
```

exptol.int *Exponential Tolerance Intervals*

Description

Provides 1-sided or 2-sided tolerance intervals for data distributed according to an exponential distribution. Data with Type II censoring is permitted.

Usage

```
exptol.int(x, alpha = 0.05, P = 0.99, side = 1, type.2 = FALSE)
```

Arguments

`x`	A vector of data which is distributed according to an exponential distribution.
`alpha`	The level chosen such that `1-alpha` is the confidence level.
`P`	The proportion of the population to be covered by this tolerance interval.
`side`	Whether a 1-sided or 2-sided tolerance interval is required (determined by `side = 1` or `side = 2`, respectively).
`type.2`	Select `TRUE` if Type II censoring is present (i.e., the data set is censored at the maximum value present). The default is `FALSE`.

Value

`exptol.int` returns a data frame with items:

`alpha`	The specified significance level.
`P`	The proportion of the population covered by this tolerance interval.
`lambda.hat`	The mean of the data (i.e., `1/rate`.
`1-sided.lower`	The 1-sided lower tolerance bound. This is given only if `side = 1`.
`1-sided.upper`	The 1-sided upper tolerance bound. This is given only if `side = 1`.
`2-sided.lower`	The 2-sided lower tolerance bound. This is given only if `side = 2`.
`2-sided.upper`	The 2-sided upper tolerance bound. This is given only if `side = 2`.

References

Blischke, W. R. and Murthy, D. N. P. (2000), *Reliability: Modeling, Prediction, and Optimization*, John Wiley & Sons, Inc.

See Also

`Exponential`

Examples

```
## 95%/99% 1-sided exponential tolerance intervals for a
## sample of size 50.

set.seed(100)
x <- rexp(100, 0.004)
out <- exptol.int(x = x, alpha = 0.05, P = 0.99, side = 1,
                  type.2 = FALSE)
out

plottol(out, x, plot.type = "both", side = "lower",
        x.lab = "Exponential Data")
```

---

exttol.int *Weibull (or Extreme-Value) Tolerance Intervals*

Description

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

Usage

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

Arguments

`x`	A vector of data which is distributed according to either a Weibull distribution or an extreme-value distribution.
`alpha`	The level chosen such that `1-alpha` is the confidence level.
`P`	The proportion of the population to be covered by this tolerance interval.
`dist`	Select either `dist = "Weibull"` or `dist = "Gumbel"` if the data is distributed according to the Weibull or extreme-value distribution, respectively.
`NR.delta`	The stopping criterion used for the Newton-Raphson algorithm when finding the maximum likelihood estimates of the Weibull or extreme-value distribution.

Details

Recall that the relationship between the Weibull distribution and the extreme-value distribution is that if the random variable X is distributed according to a Weibull distribution, then the random variable $Y = ln(X)$ is distributed according to an extreme-value distribution.

If `dist = "Weibull"`, then the natural logarithm of the data are taken so that a Newton-Raphson algorithm can be employed to find the MLEs of the extreme-value distribution and then the data and MLEs are transformed back appropriately. No transformation is performed if `dist = "Gumbel"`. The Newton-Raphson algorithm is initialized by the method of moments estimators for the parameters.

Value

`exttol.int` returns a data frame with items:

- `alpha` The specified significance level.
- `P` The proportion of the population covered by this tolerance interval.
- `shape.1` MLE for the shape parameter if `dist = "Weibull"` or for the location parameter if `dist = "Gumbel"`.
- `shape.2` MLE for the scale parameter if `dist = "Weibull"` or `dist = "Gumbel"`.
- `1-sided.lower` The 1-sided lower tolerance bound.
- `1-sided.upper` The 1-sided upper tolerance bound.

References

Bain, L. J. and Engelhardt, M. (1981), Simple Approximate Distributional Results for Confidence and Tolerance Limits for the Weibull Distribution Based on Maximum Likelihood Estimators, *Technometrics*, **23**, 15–20.

See Also

`Weibull`

Examples

```
## 90%/90% 1-sided Weibull tolerance intervals for a sample
## of size 150.

set.seed(100)
x <- rweibull(150, 3, 75)
out <- exttol.int(x = x, alpha = 0.15, P = 0.90,
                  dist = "Weibull")
out

plottol(out, x, plot.type = "both", side = "lower",
        x.lab = "Weibull Data")
```

`gamtol.int` *Gamma (or Log-Gamma) Tolerance Intervals*

Description

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

Usage

```
gamtol.int(x, alpha = 0.05, P = 0.99, side = 1,
           method = c("HE", "WBE"), log.gamma = FALSE)
```

Arguments

`x`	A vector of data which is distributed according to either a gamma distribution or a log-gamma distribution.
`alpha`	The level chosen such that `1-alpha` is the confidence level.
`P`	The proportion of the population to be covered by this tolerance interval.
`side`	Whether a 1-sided or 2-sided tolerance interval is required (determined by `side = 1` or `side = 2`, respectively).
`method`	The method for calculating the k-factors when using the normal approximation. The k-factor for the 1-sided tolerance intervals is performed exactly and thus the same for either method chosen. `"HE"` is the Howe method and is often viewed as being extremely accurate, even for small sample sizes. `"WBE"` is the Weissberg-Beatty method, which performs similarly to the Howe method for larger sample sizes.
`log.gamma`	If `TRUE`, then the data is considered to be from a log-gamma distribution, in which case the output gives tolerance intervals for the log-gamma distribution. The default is `FALSE`.

Details

Recall that if the random variable X is distributed according to a log-gamma distribution, then the random variable $Y = ln(X)$ is distributed according to a gamma distribution.

Value

`gamtol.int` returns a data frame with items:

`alpha`	The specified significance level.
`P`	The proportion of the population covered by this tolerance interval.
`1-sided.lower`	The 1-sided lower tolerance bound. This is given only if `side = 1`.
`1-sided.upper`	The 1-sided upper tolerance bound. This is given only if `side = 1`.
`2-sided.lower`	The 2-sided lower tolerance bound. This is given only if `side = 2`.
`2-sided.upper`	The 2-sided upper tolerance bound. This is given only if `side = 2`.

References

Krishnamoorthy, K., Mathew, T., and Mukherjee, S. (2008), Normal-Based Methods for a Gamma Distribution: Prediction and Tolerance Intervals and Stress-Strength Reliability, *Technometrics*, **50**, 69–78.

See Also

GammaDist, K.factor

Examples

```
## 99%/99% 1-sided gamma tolerance intervals for a sample
## of size 50.

set.seed(100)
x <- rgamma(50, 0.30, scale = 2)
out <- gamtol.int(x = x, alpha = 0.01, P = 0.99, side = 1,
                  method = "HE")
out

plottol(out, x, plot.type = "both", side = "upper",
        x.lab = "Gamma Data")
```

K.factor *Estimating K-factors for Tolerance Intervals Based on Normality*

Description

Estimates k-factors for tolerance intervals based on normality by using either the Howe method or the Weissberg-Beatty method.

Usage

```
K.factor(n, alpha = 0.05, P = 0.99, side = 1,
         method = c("HE", "WBE"))
```

Arguments

n	The sample size.
alpha	The level chosen such that 1-alpha is the confidence level.
P	The proportion of the population to be covered by this tolerance interval.
side	Whether a 1-sided or 2-sided tolerance interval is required (determined by side = 1 or side = 2, respectively).
method	The method for calculating the k-factors. The k-factor for the 1-sided tolerance intervals is performed exactly and thus the same for either method chosen. "HE" is the Howe method and is often viewed as being extremely accurate, even for small sample sizes. "WBE" is the Weissberg-Beatty method, which performs similarly to the Howe method for larger sample sizes.

Value

K.factor returns the k-factor for tolerance intervals based on normality with the arguments specified above.

References

Howe, W. G. (1969), Two-Sided Tolerance Limits for Normal Populations - Some Improvements, *Journal of the American Statistical Association*, **64**, 610–620.

Weissberg, A. and Beatty, G. (1969), Tables of Tolerance Limit Factors for Normal Distributions, *Technometrics*, **2**, 483–500.

See Also

K.table, normtol.int

Examples

```
## Showing the effect of the two estimation methods as the
## sample size increases.

K.factor(10, P = 0.95, side = 2, method = "HE")
K.factor(10, P = 0.95, side = 2, method = "WBE")

K.factor(100, P = 0.95, side = 2, method = "HE")
K.factor(100, P = 0.95, side = 2, method = "WBE")

K.factor(1000, P = 0.95, side = 2, method = "HE")
K.factor(1000, P = 0.95, side = 2, method = "WBE")
```

K.table — *Tables of K-factors for Tolerance Intervals Based on Normality*

Description

Tabulated summary of k-factors for tolerance intervals based on normality. The user can specify multiple values for each of the three inputs.

Usage

```
K.table(n, alpha, P, side = 1, by.arg = c("n", "alpha", "P"))
```

Arguments

n	A vector of sample sizes.
alpha	The level chosen such that 1-alpha is the confidence level. Can be a vector.
P	The proportion of the population to be covered by this tolerance interval. Can be a vector
side	Whether a 1-sided or 2-sided tolerance interval is required (determined by side = 1 or side = 2, respectively).

by.arg — How you would like the output organized. If `by.arg = "n"`, then the output provides a list of matrices sorted by the values specified in `n`. The matrices have rows corresponding to the values specified by `1-alpha` and columns corresponding ot the values specified by `P`. If `by.arg = "alpha"`, then the output provides a list of matrices sorted by the values specified in `1-alpha`. The matrices have rows corresponding to the values specified by `n` and columns corresponding ot the values specified by `P`. If `by.arg = "P"`, then the output provides a list of matrices sorted by the values specified in `P`. The matrices have rows corresponding to the values specified by `1-alpha` and columns corresponding ot the values specified by `n`.

Details

The method used for estimating the k-factors is that due to Howe as it is generally viewed as more accurate than the Weissberg-Beatty method.

Value

`K.table` returns a list with a structure determined by the argument `by.arg` described above.

References

Howe, W. G. (1969), Two-Sided Tolerance Limits for Normal Populations - Some Improvements, *Journal of the American Statistical Association*, **64**, 610–620.

Weissberg, A. and Beatty, G. (1969), Tables of Tolerance Limit Factors for Normal Distributions, *Technometrics*, **2**, 483–500.

See Also

`K.factor`

Examples

```
## Tables generated for each value of the sample size.

K.table(n = seq(50, 100, 10), alpha = c(0.01, 0.05, 0.10),
        P = c(0.90, 0.95, 0.99), by.arg = "n")

## Tables generated for each value of the confidence level.

K.table(n = seq(50, 100, 10), alpha = c(0.01, 0.05, 0.10),
        P = c(0.90, 0.95, 0.99), by.arg = "alpha")

## Tables generated for each value of the coverage proportion.

K.table(n = seq(50, 100, 10), alpha = c(0.01, 0.05, 0.10),
        P = c(0.90, 0.95, 0.99), by.arg = "P")
```

`laptol.int` *Laplace Tolerance Intervals*

Description

Provides 1-sided tolerance intervals for data distributed according to a Laplace distribution.

Usage

```
laptol.int(x, alpha = 0.05, P = 0.99)
```

Arguments

`x`	A vector of data which is distributed according to a Laplace distribution.
`alpha`	The level chosen such that `1-alpha` is the confidence level.
`P`	The proportion of the population to be covered by this tolerance interval.

Value

`laptol.int` returns a data frame with items:

`alpha`	The specified significance level.
`P`	The proportion of the population covered by this tolerance interval.
`1-sided.lower`	The 1-sided lower tolerance bound.
`1-sided.upper`	The 1-sided upper tolerance bound.

References

Bain, L. J. and Engelhardt, M. (1973), Interval Estimation for the Two Parameter Double Exponential Distribution, *Technometrics*, **15**, 875–887.

Examples

```
## First generate data from a Laplace distribution with location
## parameter 70 and scale parameter 3.

set.seed(100)
tmp <- runif(40)
x <- rep(70, 40) - sign(tmp - 0.5)*rep(3, 40)*
              log(2*ifelse(tmp < 0.5, tmp, 1-tmp))

## 95%/90% 1-sided Laplace tolerance intervals for the sample
## of size 40 generated above.

out <- laptol.int(x = x, alpha = 0.05, P = 0.90)
```

```
out

plottol(out, x, plot.type = "hist", side = "two",
        x.lab = "Laplace Data")
```

`logistol.int` *Logistic (or Log-Logistic) Tolerance Intervals*

Description

Provides 1-sided tolerance intervals for data distributed according to a logistic or log-logistic distribution.

Usage

```
logistol.int(x, alpha = 0.05, P = 0.99, log.log = FALSE)
```

Arguments

`x`	A vector of data which is distributed according to a logistic or log-logistic distribution.
`alpha`	The level chosen such that `1-alpha` is the confidence level.
`P`	The proportion of the population to be covered by this tolerance interval.
`log.log`	If `TRUE`, then the data is considered to be from a log-logistic distribution, in which case the output gives tolerance intervals for the log-logistic distribution. The default is `FALSE`.

Details

Recall that if the random variable X is distributed according to a log-logistic distribution, then the random variable $Y = ln(X)$ is distributed according to a logistic distribution.

Value

`logistol.int` returns a data frame with items:

`alpha`	The specified significance level.
`P`	The proportion of the population covered by this tolerance interval.
`1-sided.lower`	The 1-sided lower tolerance bound.
`1-sided.upper`	The 1-sided upper tolerance bound.

References

Balakrishnan, N. (1992), *Handbook of the Logistic Distribution*, Marcel Dekker, Inc.

Hall, I. J. (1975), One-Sided Tolerance Limits for a Logistic Distribution Based on Censored Samples, *Biometrics*, **31**, 873–880.

See Also

Logistic

Examples

```
## 90%/95% 1-sided logistic tolerance intervals for a sample
## of size 20.

set.seed(100)
x <- rlogis(20, 5, 1)
out <- logistol.int(x = x, alpha = 0.10, P = 0.95,
                    log.log = FALSE)
out

plottol(out, x, plot.type = "control", side = "two",
        x.lab = "Logistic Data")
```

---

mvtol.region *Multivariate Normal Tolerance Regions*

Description

Determines the appropriate tolerance factor for computing multivariate normal tolerance regions.

Usage

```
mvtol.region(x, alpha = 0.05, P = 0.99, B = 1000)
```

Arguments

`x`	An `nxp` matrix of data assumed to be drawn from a `p`-dimensional multivariate normal distribution. `n` pertains to the sample size.
`alpha`	The level chosen such that `1-alpha` is the confidence level. A vector of `alpha` values may be specified.
`P`	The proportion of the population to be covered by this tolerance region. A vector `P` values may be specified.
`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.

Details

A basic sketch of how the algorithm works is as follows:

(1) Generate independent chi-square random variables and Wishart random matrices.

(2) Compute the eigenvalues of the randomly generated Wishart matrices.

(3) Iterate the above steps to generate a set of `B` sample values such that the `100(1-alpha)`-th percentile is an approximate tolerance factor.

Value

mvtol.region returns a matrix where the rows pertain to each confidence level 1-alpha specified and the columns pertain to each proportion level P specified.

References

Krishnamoorthy, K. and Mondal, S. (2006), Improved Tolerance Factors for Multivariate Normal Distributions, *Communications in Statistics - Simulation and Computation*, **35**, 461–478.

Examples

```
## 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)
out1
plottol(out1, x)

## 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)

out2 <- mvtol.region(x = x, alpha = 0.10, P = 0.90, B = 1000)
out2
plottol(out2, x)
```

---

nlregtol.int *Nonlinear Regression Tolerance Bounds*

Description

Provides 1-sided or 2-sided nonlinear regression tolerance bounds.

Usage

```
nlregtol.int(formula, xy.data = data.frame(), x.new = NULL,
              side = 1, alpha = 0.05, P = 0.99, maxiter = 50,
              ...)
```

Arguments

`formula`	A nonlinear model formula including variables and parameters.
`xy.data`	A data frame in which to evaluate the formulas in `formula`. The first column of `xy.data` must be the response variable.
`x.new`	Any new levels of the predictor(s) for which to report the tolerance bounds. The number of columns must be 1 less than the number of columns for `xy.data`.
`side`	Whether a 1-sided or 2-sided tolerance bound is required (determined by `side = 1` or `side = 2`, respectively).
`alpha`	The level chosen such that `1-alpha` is the confidence level.
`P`	The proportion of the population to be covered by the tolerance bound(s).
`maxiter`	A positive integer specifying the maximum number of iterations that the nonlinear least squares routine (`nls`) should run.
`...`	Optional arguments passed to `nls` when estimating the nonlinear regression equation.

Value

`nlregtol.int` returns a data frame with items:

`alpha`	The specified significance level.
`P`	The proportion of the population covered by the tolerance bound(s).
`y.hat`	The predicted value of the response for the fitted nonlinear regression model.
`y`	The value of the response given in the first column of `xy.data`. This data frame is sorted by this value.
`1-sided.lower`	The 1-sided lower tolerance bound. This is given only if `side = 1`.
`1-sided.upper`	The 1-sided upper tolerance bound. This is given only if `side = 1`.
`2-sided.lower`	The 2-sided lower tolerance bound. This is given only if `side = 2`.
`2-sided.upper`	The 2-sided upper tolerance bound. This is given only if `side = 2`.

References

Wallis, W. A. (1951), Tolerance Intervals for Linear Regression, in *Second Berkeley Symposium on Mathematical Statistics and Probability*, ed. J. Neyman, Berkeley: University of CA Press, 43–51.

See Also

`nls`

Examples

```
## 95%/95% 2-sided nonlinear regression tolerance bounds
## for a sample of size 50.

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)))
out <- nlregtol.int(formula = formula,
                    xy.data = data.frame(cbind(y, x)),
                    x.new=cbind(c(10, 20)), side = 2,
                    alpha = 0.05, P = 0.95)
out

plottol(out, x = x, y = y, side = "two", x.lab = "X",
         y.lab = "Y")
```

---

normtol.int *Normal (or Log-Normal) Tolerance Intervals*

Description

Provides 1-sided or 2-sided tolerance intervals for data distributed according to either a normal distribution or log-normal distribution.

Usage

```
normtol.int(x, alpha = 0.05, P = 0.99, side = 1,
             method = c("HE", "WBE"), log.norm = FALSE)
```

Arguments

`x`	A vector of data which is distributed according to either a normal distribution or a log-normal distribution.
`alpha`	The level chosen such that `1-alpha` is the confidence level.
`P`	The proportion of the population to be covered by this tolerance interval.
`side`	Whether a 1-sided or 2-sided tolerance interval is required (determined by `side = 1` or `side = 2`, respectively).
`method`	The method for calculating the k-factors. The k-factor for the 1-sided tolerance intervals is performed exactly and thus the same for either method chosen. `"HE"` is the Howe method and is often viewed as being extremely accurate, even for small sample sizes. `"WBE"` is the Weissberg-Beatty method, which performs similarly to the Howe method for larger sample sizes.
`log.norm`	If `TRUE`, then the data is considered to be from a log-normal distribution, in which case the output gives tolerance intervals for the log-normal distribution. The default is `FALSE`.

Details

Recall that if the random variable X is distributed according to a log-normal distribution, then the random variable $Y = ln(X)$ is distributed according to a normal distribution.

Value

`normtol.int` returns a data frame with items:

- `alpha` The specified significance level.
- `P` The proportion of the population covered by this tolerance interval.
- `x.bar` The sample mean.
- `1-sided.lower` The 1-sided lower tolerance bound. This is given only if `side = 1`.
- `1-sided.upper` The 1-sided upper tolerance bound. This is given only if `side = 1`.
- `2-sided.lower` The 2-sided lower tolerance bound. This is given only if `side = 2`.
- `2-sided.upper` The 2-sided upper tolerance bound. This is given only if `side = 2`.

References

Howe, W. G. (1969), Two-Sided Tolerance Limits for Normal Populations - Some Improvements, *Journal of the American Statistical Association*, **64**, 610–620.

Weissberg, A. and Beatty, G. (1969), Tables of Tolerance Limit Factors for Normal Distributions, *Technometrics*, **2**, 483–500.

See Also

`Normal`, `K.factor`

Examples

```
## 95%/95% 2-sided normal tolerance intervals for a sample
## of size 100.

set.seed(100)
x <- rnorm(100, 0, 0.2)
out <- normtol.int(x = x, alpha = 0.05, P = 0.95, side = 2,
                   method = "HE", log.norm = FALSE)
out

plottol(out, x, plot.type = "both", side = "two",
        x.lab = "Normal Data")
```

npregtol.int *Nonparametric Regression Tolerance Bounds*

Description

Provides 1-sided or 2-sided nonparametric regression tolerance bounds.

Usage

```
npregtol.int(x, y, y.hat, side = 1, alpha = 0.05, P = 0.99,
             method = c("WILKS", "WALD", "HM"), upper = NULL,
             lower = NULL)
```

Arguments

`x`	A vector of values for the predictor variable. Currently, this function is only capable of handling a single predictor.
`y`	A vector of values for the response variable.
`y.hat`	A vector of fitted values extracted from a nonparametric smoothing routine.
`side`	Whether a 1-sided or 2-sided tolerance bound is required (determined by `side = 1` or `side = 2`, respectively).
`alpha`	The level chosen such that `1-alpha` is the confidence level.
`P`	The proportion of the population to be covered by the tolerance bound(s).
`method`	The method for determining which indices of the ordered residuals will be used for the tolerance bounds. `"WILKS"`, `"WALD"`, and `"HM"` are each described in `link{nptol.int}`. However, since only one tolerance bound can actually be reported for this procedure, only the first tolerance bound will be returned. Note that this is not an issue when `method = "WILKS"` is used as it only produces one set of tolerance bounds.
`upper`	The upper bound of the data. When `NULL`, then the maximum of `x` is used.
`lower`	The lower bound of the data. When `NULL`, then the minimum of `x` is used.

Value

`npregtol.int` returns a data frame with items:

`alpha`	The specified significance level.
`P`	The proportion of the population covered by the tolerance bound(s).
`x`	The values of the predictor variable.
`y`	The values of the response variable.
`y.hat`	The predicted value of the response for the fitted nonparametric smoothing routine.
`1-sided.lower`	The 1-sided lower tolerance bound. This is given only if `side = 1`.

`1-sided.upper`
: The 1-sided upper tolerance bound. This is given only if `side = 1`.

`2-sided.lower`
: The 2-sided lower tolerance bound. This is given only if `side = 2`.

`2-sided.upper`
: The 2-sided upper tolerance bound. This is given only if `side = 2`.

See Also

`loess`, `nptol.int`, `spline`

Examples

```
## 95%/95% 2-sided nonparametric regression tolerance bounds
## for a sample of size 50.

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
out <- npregtol.int(x = x, y = y, y.hat = y.hat, side = 2,
                    alpha = 0.05, P = 0.95, method = "WILKS")
out

plottol(out, x = x, y = y, y.hat = y.hat, side = "two",
        x.lab = "X", y.lab = "Y")
```

`nptol.int` *Nonparametric Tolerance Intervals*

Description

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

Usage

```
nptol.int(x, alpha = 0.05, P = 0.99, side = 1,
          method = c("WILKS", "WALD", "HM"), upper = NULL,
          lower = NULL)
```

Arguments

x	A vector of data which no distributional assumptions are made. The data is only assumed to come from a continuous distribution.
alpha	The level chosen such that 1-alpha is the confidence level.
P	The proportion of the population to be covered by this tolerance interval.
side	Whether a 1-sided or 2-sided tolerance interval is required (determined by side = 1 or side = 2, respectively).
method	The method for determining which indices of the ordered observations will be used for the tolerance intervals. "WILKS" is the Wilks method, which produces tolerance bounds symmetric about the observed center of the residuals by using the beta distribution. "WALD" is the Wald method, which produces (possibly) multiple tolerance bounds for side = 2 (each having at least the specified confidence level), but is the same as method = "WILKS" for side = 1. "HM" is the Hahn-Meeker method, which is based on the binomial distibution, but the upper and lower bounds may exceed the minimum and maximum of the sample data. For side = 2, this method will yield two intervals if an odd number of observations are to be trimmed from each side.
upper	The upper bound of the data. When NULL, then the maximum of x is used.
lower	The lower bound of the data. When NULL, then the minimum of x is used.

Value

nptol.int returns a data frame with items:

alpha	The specified significance level.
P	The proportion of the population covered by this tolerance interval.
1-sided.lower	The 1-sided lower tolerance bound. This is given only if side = 1.
1-sided.upper	The 1-sided upper tolerance bound. This is given only if side = 1.
2-sided.lower	The 2-sided lower tolerance bound. This is given only if side = 2.
2-sided.upper	The 2-sided upper tolerance bound. This is given only if side = 2.

References

Bury, K. (1999), *Statistical Distributions in Engineering*, Cambridge University Press.

Hahn, G. J. and Meeker, W. Q. (1991), *Statistical Intervals: A Guide for Practitioners*, Wiley-Interscience.

Wald, A. (1943), An Extension of Wilks' Method for Setting Tolerance Limits, *The Annals of Mathematical Statistics*, **14**, 45–55.

Wilks, S. S. (1941), Determination of Sample Sizes for Setting Tolerance Limits, *The Annals of Mathematical Statistics*, **12**, 91–96.

See Also

distfree.est, npregtol.int

Examples

```
## 90%/95% 2-sided nonparametric tolerance intervals for a
## sample of size 20.

set.seed(100)
x <- rlogis(20, 5, 1)
out <- nptol.int(x = x, alpha = 0.10, P = 0.95, side = 1,
                 method = "WILKS", upper = NULL, lower = NULL)
out

plottol(out, x, plot.type = "both", side = "two", x.lab = "X")
```

---

plottol — *Plotting Capabilities for Tolerance Intervals*

Description

Provides control charts and/or histograms for tolerance bounds on continuous data as well as tolerance ellipses for data distributed according to bivariate and trivariate normal distributions. Scatterplots with regression tolerance bounds may also be produced.

Usage

```
plottol(tol.out, x, y = NULL, y.hat = NULL,
        side = c("two", "upper", "lower"),
        plot.type = c("control", "hist", "both"),
        x.lab = NULL, y.lab = NULL, z.lab = NULL, ...)
```

Arguments

tol.out	Output from any continuous tolerance interval procedure or from a regression tolerance bound procedure.
x	Either data from a continuous distribution or the predictors for a regression model. If this is a design matrix for a linear regression model, then it must be in matrix form AND include a column of 1's if there is to be an intercept. Note that multiple predictors are only allowed if considering polynomial regression.
y	The response vector for a regression setting. Leave as NULL if not doing regression tolerance bounds.
y.hat	The fitted values from a nonparametric smoothing routine if plotting nonparametric regression tolerance bounds. Otherwise, leave as NULL.

side
: side = "two" produces plots for either the two-sided tolerance intervals or both one-sided tolerance intervals. This will be determined by the output in tol.out. side = "upper" produces plots showing the upper tolerance bounds. side = "lower" produces plots showing the lower tolerance bounds.

plot.type
: plot.type = "control" produces a control chart of the data along with the tolerance bounds specified by side. plot.type = "hist" produces a histogram of the data along with the tolerance bounds specified by side. plot.type = "both" produeces both the control chart and histogram. This argument is ignored when plotting regression data.

x.lab
: Specify the label for the x-axis.

y.lab
: Specify the label for the y-axis.

z.lab
: Specify the label for the z-axis.

...
: Additional arguments passed to the plotting function used for producing the plot specified by plot.type.

Value

plottol can return a control chart, histogram, or both for continuous data along with the calculated tolerance intervals. For regression data, plottol returns a scatterplot along with the regression tolerance bounds.

References

Montgomery, D. C. (2005), *Introduction to Statistical Quality Control*, Fifth Edition, John Wiley & Sons, Inc.

Examples

```
## 90%/90% 1-sided Weibull tolerance intervals for a sample
## of size 150.

set.seed(100)
x <- rweibull(150, 3, 75)
out <- exttol.int(x = x, alpha = 0.15, P = 0.90,
                  dist = "Weibull")
out

plottol(out, x, plot.type = "both", side = "lower",
        x.lab = "Weibull 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)

out2 <- mvtol.region(x = x, alpha = 0.10, P = 0.90, B = 1000)
out2
plottol(out2, x)

## 95%/95% 2-sided linear regression tolerance bounds
## for a sample of size 100.

set.seed(100)
x <- runif(100, 0, 10)
y <- 20 + 5*x + rnorm(100, 0, 3)
out <- regtol.int(reg = lm(y ~ x), new.x = cbind(c(3, 6, 9)),
                  side = 2, alpha = 0.05, P = 0.95)
plottol(out, x = cbind(1, x), y = y, side = "two", x.lab = "X",
        y.lab = "Y")
```

`poistol.int` *Poison Tolerance Intervals*

Description

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.

Usage

```
poistol.int(x, n, m, alpha = 0.05, P = 0.99, side = 1,
            method = c("TAB", "LS"))
```

Arguments

`x`	The number of occurrences of the event in time period `n`.
`n`	The time period of the original measurements.
`m`	The specified future length of time.
`alpha`	The level chosen such that `1-alpha` is the confidence level.
`P`	The proportion of occurrences in future time lengths of size `m` to be covered by this tolerance interval.
`side`	Whether a 1-sided or 2-sided tolerance interval is required (determined by `side = 1` or `side = 2`, respectively).
`method`	The method for calculating the lower and upper confidence bounds, which are used in the calculation of the tolerance bounds. The default method is `"TAB"`, which is the tabular method and is usually preferred for a smaller number of occurrences. `"LS"` gives the large-sample method, which is usually preferred when the number of occurrences is `x>20`.

Value

`poistol.int` returns a data frame with items:

- `alpha` The specified significance level.
- `P` The proportion of occurrences in future time periods of length `m`.
- `lambda.hat` The mean occurrence rate per unit time, calculated by `x/n`.
- `1-sided.lower` The 1-sided lower tolerance bound. This is given only if `side = 1`.
- `1-sided.upper` The 1-sided upper tolerance bound. This is given only if `side = 1`.
- `2-sided.lower` The 2-sided lower tolerance bound. This is given only if `side = 2`.
- `2-sided.upper` The 2-sided upper tolerance bound. This is given only if `side = 2`.

References

Hahn, G. J. and Chandra, R. (1981), Tolerance Intervals for Poisson and Binomial Variables, *Journal of Quality Technology*, **13**, 100–110.

See Also

Poisson

Examples

```
## 95%/90% 1-sided Poisson tolerance limits for future
## occurrences in a period of length of 3.  Both methods
## are presented for comparison.

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

---

regtol.int — *(Multiple) Linear Regression Tolerance Bounds*

Description

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

Usage

```
regtol.int(reg, new.x = NULL, side = 1, alpha = 0.05, P = 0.99)
```

Arguments

`reg`	An object of class `lm` (i.e., the results from a linear regression routine).
`new.x`	Any new levels of the predictor(s) for which to report the tolerance bounds. The number of columns must equal the number of predictors appearing on the right-hand side of the model in `reg`.
`side`	Whether a 1-sided or 2-sided tolerance bound is required (determined by `side = 1` or `side = 2`, respectively).
`alpha`	The level chosen such that `1-alpha` is the confidence level.
`P`	The proportion of the population to be covered by the tolerance bound(s).

Value

`regtol.int` returns a data frame with items:

`alpha`	The specified significance level.
`P`	The proportion of the population covered by the tolerance bound(s).
`y`	The value of the response given on the left-hand side of the model in `reg`.
`y.hat`	The predicted value of the response for the fitted linear regression model. This data frame is sorted by this value.
`1-sided.lower`	The 1-sided lower tolerance bound. This is given only if `side = 1`.
`1-sided.upper`	The 1-sided upper tolerance bound. This is given only if `side = 1`.
`2-sided.lower`	The 2-sided lower tolerance bound. This is given only if `side = 2`.
`2-sided.upper`	The 2-sided upper tolerance bound. This is given only if `side = 2`.

References

Wallis, W. A. (1951), Tolerance Intervals for Linear Regression, in *Second Berkeley Symposium on Mathematical Statistics and Probability*, ed. J. Neyman, Berkeley: University of CA Press, 43–51.

See Also

`lm`

Examples

```
## 95%/95% 2-sided linear regression tolerance bounds
## for a sample of size 100.

set.seed(100)
x <- runif(100, 0, 10)
y <- 20 + 5*x + rnorm(100, 0, 3)
out <- regtol.int(reg = lm(y ~ x), new.x = cbind(c(3, 6, 9)),
```

```
                    side = 2, alpha = 0.05, P = 0.95)
plottol(out, x = cbind(1, x), y = y, side = "two", x.lab = "X",
        y.lab = "Y")
```

TwoParExponential *The 2-Parameter Exponential Distribution*

Description

Density, distribution function, quantile function, and random generation for the 2-parameter exponential distribution with rate equal to `rate` and shift equal to `shift`.

Usage

```
d2exp(x, rate = 1, shift = 0, log = FALSE)
p2exp(q, rate = 1, shift = 0, lower.tail = TRUE, log.p = FALSE)
q2exp(p, rate = 1, shift = 0, lower.tail = TRUE, log.p = FALSE)
r2exp(n, rate = 1, shift = 0)
```

Arguments

`x,q`	Vector of quantiles.
`p`	Vector of probabilities.
`n`	The number of observations. If `length>1`, then the length is taken to be the number required.
`rate`	Vector of rates.
`shift`	Vector of shifts.
`log,log.p`	Logical vectors. If `TRUE`, then probabilities are given as `log(p)`.
`lower.tail`	Logical vector. If `TRUE`, then probabilities are $P[X \leq x]$, else $P[X > x]$.

Details

If `rate` or `shift` are not specified, then they assume the default values of 1 and 0, respectively.

The 2-parameter exponential distribution has density

$$f(x) = \frac{1}{\beta} e^{(x-\mu)/\beta}$$

where $x \geq \mu$, μ is the shift parameter, and $\beta > 0$ is the scale parameter.

Value

`d2exp` gives the density, `p2exp` gives the distribution function, `q2exp` gives the quantile function, and `r2exp` generates random deviates.

See Also

runif and .Random.seed about random number generation.

Examples

```
## Randomly generated data from the 2-parameter exponential
## distribution.

set.seed(100)
x <- r2exp(n = 500, rate = 3, shift = -10)
hist(x, main = "Randomly Generated Data", prob = TRUE)

x.1 = sort(x)
y <- d2exp(x = x.1, rate = 3, shift = -10)
lines(x.1, y, col = 2, lwd = 2)

plot(x.1, p2exp(q = x.1, rate = 3, shift = -10), type = "l",
     xlab = "x", ylab = "Cumulative Probabilities")

q2exp(p = 0.20, rate = 3, shift = -10, lower.tail = FALSE)
q2exp(p = 0.80, rate = 3, shift = -10)
```

---

uniftol.int *Uniform Tolerance Intervals*

Description

Provides 1-sided tolerance intervals for data distributed according to a uniform distribution.

Usage

```
uniftol.int(x, alpha = 0.05, P = 0.99)
```

Arguments

x	A vector of data which is distributed according to a uniform distribution.
alpha	The level chosen such that 1-alpha is the confidence level.
P	The proportion of the population to be covered by this tolerance interval.

Value

uniftol.int returns a data frame with items:

alpha	The specified significance level.
P	The proportion of the population covered by this tolerance interval.
1-sided.lower	The 1-sided lower tolerance bound.
1-sided.upper	The 1-sided upper tolerance bound.

References

Faulkenberry, G. D. and Weeks, D. L. (1968), Sample Size Determination for Tolerance Limits, *Technometrics*, **10**, 343–348.

Examples

```
## 90%/90% 1-sided uniform tolerance intervals for a sample
## of size 50.

set.seed(100)
x <- runif(50, 0, 50)
out <- uniftol.int(x = x, alpha = 0.10, P = 0.90)
out

plottol(out, x, plot.type = "hist", side = "two",
        x.lab = "Uniform Data")
```

Index