Package ‘conf’

August 24, 2020

Type Package
Title Visualization and Analysis of Statistical Measures of Confidence
Version 1.7.0
Maintainer Christopher Weld <ceweld@email.wm.edu>
Imports graphics, stats, statmod, STAR, fitdistrplus, pracma, rootSolve, utils
Description Enables: (1) plotting two-dimensional confidence regions, (2) coverage analysis of confidence region simulations and (3) calculating confidence intervals and the associated actual coverage for binomial proportions. Each is given in greater detail next.
(1) Plots the two-dimensional confidence region for probability distribution parameters (supported distribution suffixes: cauchy, gamma, invgauss, logis, llogis, lnorm, norm, unif, weibull) corresponding to a user-given complete or right-censored dataset and level of significance. The crplot() algorithm plots more points in areas of greater curvature to ensure a smooth appearance throughout the confidence region boundary. An alternative heuristic plots a specified number of points at roughly uniform intervals along its boundary. Both heuristics build upon the radial profile log-likelihood ratio technique for plotting confidence regions given by Jaeger (2016) <doi:10.1080/00031305.2016.1182946>, and are detailed in a publication by Weld (2019) <doi:10.1080/00031305.2018.1564696>. (2) Performs confidence region coverage simulations for a random sample drawn from a user-specified parametric population distribution, or for a user-specified dataset and point of interest with coversim(). (3) Calculates confidence interval bounds for a binomial proportion with binomTest(), calculates the actual coverage with binomTestCoverage(), and plots the actual coverage with binomTestCoveragePlot(). Calculates confidence interval bounds for the binomial proportion using an ensemble of constituent confidence intervals with binomTestEnsemble(). Calculates confidence interval bounds for the binomial proportion using a complete enumeration of all possible transitions from one actual coverage acceptance curve to another which minimizes the root mean square error for n <= 15 and follows the transitions for well-known confidence intervals for n > 15 using binomTestMSE().

Depends R (>= 3.2.0)
License GPL (<= 2)
Encoding UTF-8
LazyData true
RoxygenNote 7.1.0
Description

Generates lower and upper confidence interval limits for a binomial proportion using different types of confidence intervals.

Usage

```r
binomTest(n, x,
          alpha = 0.05,
          intervalType = "Clopper-Pearson")
```
Arguments

\( n \)  sample size
\( x \)  number of successes
\( \alpha \)  significance level for confidence interval

\( \text{intervalType} \) type of confidence interval used; either "Clopper-Pearson", "Wald", "Wilson-Score", "Jeffreys", "Agresti-Coull", "Arcsine", or "Blaker"

Details

Generates a lower and upper confidence interval limit for a binomial proportion using

- various types of confidence intervals,
- various sample sizes, and
- various numbers of successes.

When the \texttt{binomTest} function is called, it returns a two-element vector in which

- the first element is the lower bound of the confidence interval, and
- the second element is the upper bound of the confidence interval.

This confidence interval is constructed by calculating lower and upper bounds associated with the confidence interval procedure specified by the \texttt{intervalType} argument. Lower bounds that are negative are set to 0 and upper bounds that are greater than 1 are set to 1.

Author(s)

Hayeon Park (<hpark03@email.wm.edu>), Larry Leemis (<leemis@math.wm.edu>)

See Also

dbinom

Examples

\texttt{binomTest(10, 6)}
\texttt{binomTest(100, 30, intervalType = "Agresti-Coull")}
**binomTestCoverage**

### Actual Coverage Calculation for Binomial Proportions

**Description**

Calculates the actual coverage of a confidence interval for a binomial proportion for a particular sample size \( n \) and a particular value of the probability of success \( p \) for several confidence interval procedures.

**Usage**

```r
binomTestCoverage(n, p,
                   alpha = 0.05,
                   intervalType = "Clopper-Pearson")
```

**Arguments**

- \( n \): sample size
- \( p \): population probability of success
- \( \alpha \): significance level for confidence interval
- \( \text{intervalType} \): type of confidence interval used; either "Clopper-Pearson", "Wald", "Wilson-Score", "Jeffreys", "Agresti-Coull", "Arcsine", or "Blaker"

**Details**

Calculates the actual coverage of a confidence interval procedure at a particular value of \( p \) for

- various types of confidence intervals,
- various probabilities of success \( p \), and
- various sample sizes \( n \).

The actual coverage for a particular value of \( p \), the probability of success of interest, is

\[
c(p) = \sum_{x=0}^{n} I(x, p) \binom{n}{x} p^x (1-p)^{n-x},
\]

where \( I(x, p) \) is an indicator function that determines whether a confidence interval covers \( p \) when \( X = x \) (see Vollset, 1993).

The binomial distribution with arguments \( \text{size} = n \) and \( \text{prob} = p \) has probability mass function

\[
p(x) = \binom{n}{x} p^x (1-p)^{n-x}
\]

for \( x = 0, 1, 2, \ldots, n \).

The algorithm for computing the actual coverage for a particular probability of success begins by calculating all possible lower and upper bounds associated with the confidence interval procedure specified by the \( \text{intervalType} \) argument. The appropriate binomial probabilities are summed to determine the actual coverage at \( p \).
binomTestCoveragePlot

Author(s)
Hayeon Park (<hpark03@email.wm.edu>), Larry Leemis (<leemis@math.wm.edu>)

References

See Also
dbinom

Examples

```r
binomTestCoverage(6, 0.4)
binomTestCoverage(n = 10, p = 0.3, alpha = 0.01, intervalType = "Wilson-Score")
```

Description
Generates plots for the actual coverage of a binomial proportion using various types of confidence intervals. Plots the actual coverage for a given sample size and stated nominal coverage $1 - \alpha$.

Usage

```r
binomTestCoveragePlot(n, 
  alpha = 0.05, 
  intervalType = "Clopper-Pearson", 
  plo = 0, 
  phi = 1, 
  clo = 1 - 2 * alpha, 
  chi = 1, 
  points = 5 + floor(250 / n), 
  showTrueCoverage = TRUE, 
  gridCurves = FALSE)
```

Arguments

- **n**: sample size
- **alpha**: significance level for confidence interval
- **intervalType**: type of confidence interval used; either "Clopper-Pearson", "Wald", "Wilson-Score", "Jeffreys", "Agresti-Coull", "Arcsine", or "Blaker"
- **plo**: lower limit for percentile (horizontal axis)
- **phi**: upper limit for percentile (horizontal axis)
binomTestCoveragePlot

clo lower limit for coverage (vertical axis)
chi upper limit for coverage (vertical axis)
points number of points plotted in each segment of the plot; if default, varies with ‘n’ (see above)
showTrueCoverage logical; if TRUE (default), a solid red line will appear at 1 − alpha
gridCurves logical; if TRUE, display acceptance curves in gray

Details
Generates an actual coverage plot for binomial proportions using
• various types of confidence intervals, and
• various sample sizes.
When the function is called with default arguments,
• the horizontal axis is the percentile at which the coverage is evaluated,
• the vertical axis is the actual coverage percentage at each percentile, that is, the probability that the true value at a percentile is contained in the corresponding confidence interval, and
• the solid red line is the stated coverage of 1 − alpha.
The actual coverage for a particular value of p, the percentile of interest, is
\[ c(p) = \sum_{x=0}^{n} I(x, p) \binom{n}{x} p^x (1 - p)^{n-x}, \]
where \( I(x, p) \) is an indicator function that determines whether a confidence interval covers \( p \) when \( X = x \) (see Vollset, 1993).
The binomial distribution with arguments \( \text{size} = n \) and \( \text{prob} = p \) has probability mass function
\[ p(x) = \binom{n}{x} p^x (1 - p)^{n-x} \]
for \( x = 0, 1, \ldots, n. \)
The algorithm for plotting the actual coverage begins by calculating all possible lower and upper bounds associated with the confidence interval procedure specified by the intervalType argument. These values are concatenated into a vector which is sorted. Negative values and values that exceed 1 are removed from this vector. These values are the breakpoints in the actual coverage function. The points argument gives the number of points plotted on each segment of the graph of the actual coverage. The plo and phi arguments can be used to expand or compress the plots horizontally. The clo and chi arguments can be used to expand or compress the plots vertically.
By default, the showTrueCoverage argument plots a solid horizontal red line at the height of the stated coverage. The actual coverage is plotted with solid black lines for each segment of the actual coverage.
The gridCurves argument is assigned a logical value which indicates whether the acceptance curves giving all possible actual coverage values should be displayed as gray curves.
Author(s)
Hayeon Park (<hpark03@email.wm.edu>), Larry Leemis (<leemis@math.wm.edu>)

References

See Also
dbinom

Examples
  binomTestCoveragePlot(6)
  binomTestCoveragePlot(10, intervalType = "Wilson-Score", clo = 0.8)
  binomTestCoveragePlot(n = 100, intervalType = "Wald", clo = 0, chi = 1, points = 30)

binomTestEnsemble

Ensemble Confidence Intervals for Binomial Proportions

Description
Generates lower and upper confidence interval limits for a binomial proportion using an ensemble of confidence intervals.

Usage
  binomTestEnsemble(n, x, 
    alpha = 0.05, 
    CP = TRUE, 
    WS = TRUE, 
    JF = TRUE, 
    AC = TRUE, 
    AR = TRUE)

Arguments
  n           sample size
  x           number of successes
  alpha       significance level for confidence interval
  CP          logical; if TRUE (default), include Clopper-Pearson confidence interval procedure in the ensemble
  WS          logical; if TRUE (default), include Wilson-Score confidence interval procedure in the ensemble
binomTestEnsemble

**JF**
logical; if TRUE (default), include Jeffreys confidence interval procedure in the ensemble

**AC**
logical; if TRUE (default), include Agresti-Coull confidence interval procedure in the ensemble

**AR**
logical; if TRUE (default), include Arcsine confidence interval procedure in the ensemble

**Details**
Generates lower and upper confidence interval limits for a binomial proportions using
- various sample sizes,
- various numbers of successes, and
- various combinations of confidence intervals.

When the `binomTestEnsemble` function is called, it returns a two-element vector in which
- the first element is the lower bound of the Ensemble confidence interval, and
- the second element is the upper bound of the Ensemble confidence interval.

To construct an Ensemble confidence interval that attains an actual coverage that is close to the stated coverage, the five constituent confidence interval procedures can be combined. Since these intervals vary in width, the lower limits and the actual coverage of the constituent confidence intervals at the maximum likelihood estimator are calculated. Likewise, the upper limits and the actual coverage of the constituent confidence intervals at the maximum likelihood estimator are calculated. The centroids of the lower and upper constituent confidence intervals for points falling below and above the stated coverage are connected with a line segment. The point of intersection of these line segments and the stated coverage gives the lower and upper bound of the Ensemble confidence interval. Special cases to this approach are given in the case of (a) the actual coverages all fall above or below the stated coverage, and (b) the slope of the line connecting the centroids is infinite.

If only one of the logical arguments is TRUE, the code returns a simple confidence interval of that one procedure.

The Wald confidence interval is omitted because it degenerates in actual coverage for $x = 0$ and $x = n$.

**Author(s)**
Hayeon Park (<hpark03@email.wm.edu>), Larry Leemis (<leemis@math.wm.edu>)

**Examples**
```r
binomTestEnsemble(10, 3)
binomTestEnsemble(100, 82, CP = FALSE, AR = FALSE)
binomTestEnsemble(33, 1, CP = FALSE, JF = FALSE, AC = FALSE, AR = FALSE)
```
Description

Generates lower and upper confidence interval limits for a binomial proportion that minimizes the root mean square error (RMSE) of the actual coverage function.

Usage

```r
binomTestMSE(n, x,
    alpha = 0.05,
    smooth = 1,
    showRMSE = TRUE,
    showAll = FALSE)
```

Arguments

- `n`: sample size
- `x`: number of successes
- `alpha`: significance level for confidence interval
- `smooth`: smoothness index
- `showRMSE`: a logical variable indicating whether to show the value of RMSE
- `showAll`: a logical variable indicating whether to show confidence intervals of all possible number of successes

Details

Generates lower and upper confidence interval limits for a binomial proportion for

- various sample sizes,
- various numbers of successes.

When the `binomTestMSE` function is called, it returns a two-element vector in which

- the first element is the lower bound of the RMSE-minimizing confidence interval, and
- the second element is the upper bound of the RMSE-minimizing confidence interval.

An RMSE-minimizing two-sided 100 * (1 - alpha) percent confidence interval for p is constructed from a random sample of size n from a Bernoulli(p) population. The parameter x gives the number of successes in the n mutually independent Bernoulli trials. For n <= 15, all possible jumps between acceptance curves associated with the actual coverage function are enumerated based on their one-to-one relationship with the symmetric Dyck paths. For each sequence of jumps between acceptance curves, the confidence interval bounds that are returned are associated with discontinuities in the actual coverage function that together result in the lowest possible RMSE. A set of smoothness constraints that build on four existing non-conservative confidence intervals (Wilson-score,
Jeffreys, Arcsine, and Agresti-Coull) is used if the smoothness index smooth is set to one. These constraints ensure that the RMSE-confidence interval achieves smoothness, a preferable property of the binomial confidence interval that is related to lower bound differences for adjacent values of $x$. There is a trade-off between the RMSE and the smoothness. For $n > 100$, smoothness is required. The RMSE usually increases if the smoothness constraints are used. For $n > 15$, only the symmetric Dyck paths associated with the Wilson–score, Jeffreys, Arcsine, and Agresti–Coull confidence interval procedures are used instead of enumerating because the computation time increases in a factorial fashion in $n$. The minimal RMSE is not guaranteed for $n > 15$ because another symmetric Dyck path other than those associated with the four existing confidence interval procedures might prove to be optimal. However, this procedure does ensure a lower RMSE than any of the four existing confidence intervals for all $n$.

**Author(s)**

Kexin Feng (<kfeng01@email.wm.edu>), Larry Leemis (<leemis@math.wm.edu>), Heather Sasi-nowska (<hdsasinowska@wm.edu>)

**Examples**

```
binomTestMSE(10, 3)
```
Vignettes

The CRAN website https://CRAN.R-project.org/package=conf contains links for vignettes on the crplot and coversim functions.

Acknowledgments

The lead author thanks The Omar Bradley Fellowship for Research in Mathematics for funding that partially supported this work.

Author(s)

Christopher Weld, Hayeon Park, Larry Leemis
Maintainer: Christopher Weld <ceweld@email.wm.edu>

coversim  

Description

Creates a confidence region and determines coverage results for a corresponding point of interest. Iterates through a user specified number of trials. Each trial uses a random dataset with user-specified parameters (default) or a user specified dataset matrix ('n' samples per column, 'iter' columns) and returns the corresponding actual coverage results. See the CRAN website https://CRAN.R-project.org/package=conf for a link to a coversim vignette.

Usage

coversim(alpha, distn, n = NULL, iter = NULL, dataset = NULL, point = NULL, seed = NULL, a = NULL, b = NULL, kappa = NULL, lambda = NULL, mu = NULL, s = NULL, sigma = NULL, )
theta = NULL,
heuristic = 1,
maxdeg = 5,
ellipse_n = 4,
pts = FALSE,
mlelab = TRUE,
sf = c(5, 5),
mar = c(4, 4.5, 2, 1.5),
xlab = "",
ylab = "",
main = "",
xlas = 0,
ylas = 0,
origin = FALSE,
xlim = NULL,
ylim = NULL,
tol = .Machine$double.eps ^ 1,
info = FALSE,
returnsamp = FALSE,
returnquant = FALSE,
repair = TRUE,
exact = FALSE,
showplot = FALSE,
delay = 0 )

Arguments

alpha significance level; scalar or vector; resulting plot illustrates a 100(1 - alpha)%
confidence region.
distn distribution to fit the dataset to; accepted values: 'cauchy', 'gamma', 'invgauss',
'logis', 'llogis', 'lnorm', 'norm', 'unif', 'weibull'.
n trial sample size (producing each confidence region); scalar or vector; needed if
a dataset is not given.
iter iterations (or replications) of individual trials per parameterization; needed if a
dataset is not given.
dataset a 'n' x 'iter' matrix of dataset values, or a vector of length 'n' (for a single
iteration).
point coverage is assessed relative to this point.
seed random number generator seed.
a distribution parameter (when applicable).
b distribution parameter (when applicable).
kappa distribution parameter (when applicable).
lambda distribution parameter (when applicable).
mu distribution parameter (when applicable).
s distribution parameter (when applicable).
sigma
distribution parameter (when applicable).
theta
distribution parameter (when applicable).
heuristic
numeric value selecting method for plotting: 0 for elliptic-oriented point distribution, and 1 for smoothing boundary search heuristic.
maxdeg
maximum angle tolerance between consecutive plot segments in degrees.
ellipse_n
number of roughly equidistant confidence region points to plot using the elliptic-oriented point distribution (must be a multiple of four because its algorithm exploits symmetry in the quadrants of an ellipse).
pts
displays confidence region boundary points if TRUE (applies to confidence region plots in which showplot = TRUE).
mlelab
logical argument to include the maximum likelihood estimate coordinate point (default is TRUE, applies to confidence region plots when showplot = TRUE).
sf
significant figures in axes labels specified using sf = c(x, y), where x and y represent the optional digits argument in the R function round as it pertains the horizontal and vertical labels.
mar
specifies margin values for par(mar = c()) (see mar in par).
xlab
string specifying the horizontal axis label (applies to confidence region plots when showplot = TRUE).
ylab
string specifying the vertical axis label (applies to confidence region plots when showplot = TRUE).
main
string specifying the plot title (applies to confidence region plots when showplot = TRUE).
xlas
numeric in 0, 1, 2, 3 specifying the style of axis labels (see las in par, applies to confidence region plots when showplot = TRUE).
ylas
numeric in 0, 1, 2, 3 specifying the style of axis labels (see las in par, applies to confidence region plots when showplot = TRUE).
origin
logical argument to include the plot origin (applies to confidence region plots when showplot = TRUE).
xlim
two element vector containing horizontal axis minimum and maximum values (applies to confidence region plots when showplot = TRUE).
ylim
two element vector containing vertical axis minimum and maximum values (applies to confidence region plots when showplot = TRUE).
tol
the uniroot parameter specifying its required accuracy.
info
logical argument to return coverage information in a list; includes alpha value(s), n value(s), coverage and error results per iteration, and returnsamp and/or returnquant when requested.
returnsamp
logical argument; if TRUE returns random samples used in a matrix with n rows, iter cols.
returnquant
logical argument; if TRUE returns random quantiles used in a matrix with n rows, iter cols.
repair
logical argument to repair regions inaccessible using a radial angle from its MLE (multiple root azimuths).
**exact** logical argument specifying if alpha value is adjusted to compensate for negative coverage bias in order to achieve (1 - alpha) coverage probability using previously recorded Monte Carlo simulation results; available for limited values of alpha (roughly <= 0.2–0.3), n (typically n = 4, 5, ..., 50) and distributions (distn suffixes: weibull, llogis, norm).

**showplot** logical argument specifying if each coverage trial produces a plot.

**delay** numeric value of delay (in seconds) between trials so its plot can be seen (applies when showplot = TRUE).

### Details

Parameterizations for supported distributions are given following the default axes convention in use by crplot and coversim, which are:

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Horizontal Axis</th>
<th>Vertical Axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cauchy</td>
<td>a</td>
<td>s</td>
</tr>
<tr>
<td>gamma</td>
<td>θ</td>
<td>κ</td>
</tr>
<tr>
<td>inverse Gaussian</td>
<td>µ</td>
<td>λ</td>
</tr>
<tr>
<td>log logistic</td>
<td>λ</td>
<td>κ</td>
</tr>
<tr>
<td>log normal</td>
<td>µ</td>
<td>σ</td>
</tr>
<tr>
<td>logistic</td>
<td>µ</td>
<td>σ</td>
</tr>
<tr>
<td>normal</td>
<td>µ</td>
<td>σ</td>
</tr>
<tr>
<td>uniform</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>Weibull</td>
<td>κ</td>
<td>λ</td>
</tr>
</tbody>
</table>

Each respective distribution is defined below.

- The Cauchy distribution for the real-numbered location parameter \(a\), scale parameter \(s\), and \(x\) is a real number, has the probability density function
  \[
  \frac{1}{(\pi s)^2} \frac{1}{(1 + ((x - a)/s)^2)}.
  \]

- The gamma distribution for shape parameter \(\kappa > 0\), scale parameter \(\theta > 0\), and \(x > 0\), has the probability density function
  \[
  \frac{1}{\Gamma(\kappa) \theta^\kappa} x^{(\kappa - 1)} \exp(-x/\theta).
  \]

- The inverse Gaussian distribution for mean \(\mu > 0\), shape parameter \(\lambda > 0\), and \(x > 0\), has the probability density function
  \[
  \sqrt{\frac{\lambda}{2\pi x^3}} \exp(-\frac{\lambda(x - \mu)^2}{2\mu^2 x}).
  \]

- The log logistic distribution for scale parameter \(\lambda > 0\), shape parameter \(\kappa > 0\), and \(x \geq 0\), has a probability density function
  \[
  (\kappa \lambda)(x\lambda)^{(\kappa - 1)} / (1 + (\lambda x)^\kappa)^2.
  \]

- The log normal distribution for the real-numbered mean \(\mu\) of the logarithm, standard deviation \(\sigma > 0\) of the logarithm, and \(x > 0\), has the probability density function
  \[
  \frac{1}{x \sigma \sqrt{2\pi}} \exp\left(-\frac{(\log x - \mu)^2}{2\sigma^2}\right).
  \]
The logistic distribution for the real-numbered location parameter $\mu$, scale parameter $\sigma$, and $x$ is a real number, has the probability density function
\[
\frac{1}{\sigma} \exp\left((x - \mu)/\sigma + \exp((x - \mu)/\sigma)\right)^{-2}
\]

The normal distribution for the real-numbered mean $\mu$, standard deviation $\sigma > 0$, and $x$ is a real number, has the probability density function
\[
\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right).
\]

The uniform distribution for real-valued parameters $a$ and $b$ where $a < b$ and $a \leq x \leq b$, has the probability density function
\[
\frac{1}{b - a}.
\]

The Weibull distribution for scale parameter $\lambda > 0$, shape parameter $\kappa > 0$, and $x > 0$, has the probability density function
\[
\kappa \lambda^\kappa x^{(\kappa - 1)} \exp\left(-\lambda x^\kappa\right).
\]

Value

If the optional argument info = TRUE is included then a list of coverage results is returned. That list includes alpha value(s), n value(s), coverage and error results per iteration. Additionally, returnsamp = TRUE and/or returnquant = TRUE will result in an n row, iter column maxtix of sample and/or sample cdf values.

Author(s)

Christopher Weld (<ceweld@email.wm.edu>)
Lawrence Leemis (<leemis@math.wm.edu>)

References


See Also
crplot, uniroot

Examples

```r
# assess actual coverage at various alpha = {0.5, 0.1} given n = 30 samples, completing 10 trials per parameterization (iter) for a normal(mean = 2, sd = 3) rv coversim(alpha = c(0.5, 0.1), "norm", n = 30, iter = 10, mu = 2, sigma = 3)

# show plots for 5 iterations of 30 samples each from a Weibull(2, 3) coversim(0.5, "weibull", n = 30, iter = 5, lambda = 1.5, kappa = 0.5, showplot = TRUE, origin = TRUE)
```
crplot  

Plotting Two-Dimensional Confidence Regions

Description

Plots the two-dimensional confidence region for probability distribution parameters (supported distribution suffixes: cauchy, gamma, invgauss, lnorm, llogis, logis, norm, unif, weibull) corresponding to a user given complete or right-censored dataset and level of significance. See the CRAN website https://CRAN.R-project.org/package=conf for a link to two crplot vignettes.

Usage

```r
crplot(dataset, alpha, distn, 
cen = rep(1, length(dataset)), 
heuristic = 1, 
maxdeg = 5, 
ellipse_n = 4, 
pts = TRUE, 
mlelab = TRUE, 
sf = NULL, 
mar = c(4, 4.5, 2, 1.5), 
exyswap = FALSE, 
xlab = "", 
ylab = "", 
main = "", 
xlas = 0, 
ylas = 0, 
origin = FALSE, 
xlim = NULL, 
ylim = NULL, 
tol = .Machine$double.eps^1, 
info = FALSE, 
maxcount = 30, 
repair = TRUE, 
jumpshift = 0.5, 
jumpuphill = min(alpha, 0.01), 
jumpinfo = FALSE, 
showjump = FALSE, 
showplot = TRUE, 
animate = FALSE, 
delay = 0.5, 
exact = FALSE, 
silent = FALSE )
```

Arguments

dataset a 1 x n vector of data values.
alpha significance level; resulting plot illustrates a 100(1 - alpha)% confidence region.
distn distribution to fit the dataset to; accepted values: 'cauchy', 'gamma', 'invgauss', 'logis', 'llogis', 'lnorm', 'norm', 'unif', 'weibull'.
cen a vector of binary values specifying if the corresponding data values are right-censored (0), or observed (1, default); its length must match length(dataset).
heuristic numeric value selecting method for plotting: 0 for elliptic-oriented point distribution, and 1 for smoothing boundary search heuristic.
maxdeg maximum angle tolerance between consecutive plot segments in degrees.
ellipse_n number of roughly equidistant confidence region points to plot using the elliptic-oriented point distribution (must be a multiple of four because its algorithm exploits symmetry in the quadrants of an ellipse).
pts displays confidence region boundary points identified if TRUE.
mlelab logical argument to include the maximum likelihood estimate coordinate point (default is TRUE).
sf significant figures in axes labels specified using sf = c(x, y), where x and y represent the optional digits argument in the R function round as it pertains to the horizontal and vertical labels.
mar specifies margin values for par(mar = c()) (see mar in par).
xyswap logical argument to switch the axes that the distribution parameter are shown.
xlab string specifying the x axis label.
ylab string specifying the y axis label.
main string specifying the plot title.
xlas numeric in 0, 1, 2, 3 specifying the style of axis labels (see las in par).
ylas numeric in 0, 1, 2, 3 specifying the style of axis labels (see las in par).
origin logical argument to include the plot origin (default is FALSE).
xlim two-element vector containing horizontal axis minimum and maximum values.
ylim two-element vector containing vertical axis minimum and maximum values.
tol the uniroot parameter specifying its required accuracy.
info logical argument to return plot information: MLE is returned as a list; (x, y) plot point coordinates and corresponding phi angles (with respect to MLE) are returned as a list.
maxcount integer value specifying the number of smoothing search iterations before terminating with maxdeg not met.
repair logical argument to repair regions inaccessible using a radial angle from its MLE due to multiple roots at select phi angles.
jumpshift see vignette "conf Advanced Options" for details; location (as a fractional value between 0 and 1) along the vertical or horizontal "gap" (near an uncharted region) to locate a jump-center toward; can be either a scalar value (uniformly applied to all jump-centers) or vector of length four (with unique values for its respective quadrants, relative to the MLE).
jumpuphill  see vignette "conf Advanced Options" for details; significance level increase to alpha for the jump-center (corresponds to an "uphill" location on its likelihood function); can be either a scalar value (uniformly applied to all jump-centers) or vector of length four (with unique values for its respective quadrants, relative to the MLE).

jumpinfo  logical argument to return plot info (see info argument) and jump-center info; returned within 'repair' attribute are jumpuphill value, jumpshift value, "|" or "-" gap type, jump-center(s) coordinates, and coordinates of points left & right of the inaccessible region.

showjump  logical argument specifying if jump-center repair reference points appear on the confidence region plot.

showplot  logical argument specifying if a plot is output; altering from its default of TRUE is only logical assuming crplot is run for its data only (see the info argument).

animate  logical argument specifying if an animated plot build will display; the animation sequence is given in successive plots.

delay  numeric value of delay (in seconds) between successive plots when animate = TRUE.

exact  logical argument specifying if alpha value is adjusted to compensate for negative coverage bias to achieve (1 - alpha) coverage probability using previously recorded Monte Carlo simulation results; available for limited values of alpha (roughly <= 0.2–0.3), n (typically n = 4, 5, ..., 50) and distributions (distn suffixes: weibull, llogis, norm).

silent  logical argument specifying if console output should be suppressed.

Details

This function plots a confidence region for a variety of two-parameter distributions. It requires:

• a vector of dataset values,
• the level of significance (alpha), and
• a population distribution to fit the data to.

Plots display according to probability density function parameterization given later in this section. Two heuristics (and their associated combination) are available to plot confidence regions. Along with their descriptions, they are:

1. **Smoothing Boundary Search Heuristic (default)**. This heuristic plots more points in areas of greater curvature to ensure a smooth appearance throughout the confidence region boundary. Its maxdeg parameter specifies the maximum tolerable angle between three successive points. Lower values of maxdeg result in smoother plots, and its default value of 5 degrees provides adequate smoothing in most circumstances. Values of maxdeg ≤ 3 are not recommended due to their complicating implications to trigonometric numerical approximations near 0 and 1; their use may result in plot errors.

2. **Elliptic-Oriented Point Distribution**. This heuristic allows the user to specify a number of points to plot along the confidence region boundary at roughly uniform intervals. Its name is derived from the technique it uses to choose these points—an extension of the Steiner generation of a non-degenerate conic section, also known as the parallelogram method—which
identifies points along an ellipse that are approximately equidistant. To exploit the computational benefits of ellipse symmetry over its four quadrants, ellipse_n value must be divisible by four.

By default, crplot implements the smoothing boundary search heuristic. Alternatively, the user can plot using the elliptic-oriented point distribution algorithm, or a combination of them both. Combining the two techniques initializes the plot using the elliptic-oriented point distribution algorithm, and then subsequently populates additional points in areas of high curvature (those outside of the maximum angle tolerance parameterization) in accordance with the smoothing boundary search heuristic. This combination results when the smoothing boundary search heuristic is specified in conjunction with an ellipse_n value greater than four.

Both of the aforementioned heuristics use a radial profile log likelihood function to identify points along the confidence region boundary. It cuts the log likelihood function in a directional azimuth from its MLE, and locates the associated confidence region boundary point using the asymptotic results associated with the ratio test statistic $-2[\log L(\theta) - \log L(\hat{\theta})]$ which converges in distribution to the chi-square distribution with two degrees of freedom (for a two parameter distribution).

The default axes convention in use by crplot are

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Horizontal Axis</th>
<th>Vertical Axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cauchy</td>
<td>$a$</td>
<td>$s$</td>
</tr>
<tr>
<td>gamma</td>
<td>$\theta$</td>
<td>$\kappa$</td>
</tr>
<tr>
<td>inverse Gaussian</td>
<td>$\mu$</td>
<td>$\lambda$</td>
</tr>
<tr>
<td>log logistic</td>
<td>$\lambda$</td>
<td>$\kappa$</td>
</tr>
<tr>
<td>log normal</td>
<td>$\mu$</td>
<td>$\sigma$</td>
</tr>
<tr>
<td>logistic</td>
<td>$\mu$</td>
<td>$\sigma$</td>
</tr>
<tr>
<td>normal</td>
<td>$a$</td>
<td>$b$</td>
</tr>
<tr>
<td>uniform</td>
<td>$a$</td>
<td>$b$</td>
</tr>
<tr>
<td>Weibull</td>
<td>$\kappa$</td>
<td>$\lambda$</td>
</tr>
</tbody>
</table>

where each respective distribution is defined below.

- The Cauchy distribution for the real-numbered location parameter $a$, scale parameter $s$, and $x$ is a real number, has the probability density function
  \[1/(s\pi(1 + ((x - a)/s)^2)).\]

- The gamma distribution for shape parameter $\kappa > 0$, scale parameter $\theta > 0$, and $x > 0$, has the probability density function
  \[1/(\Gamma(\kappa)\theta^\kappa)x^{(\kappa-1)}\exp(-\theta/x)).\]

- The inverse Gaussian distribution for mean $\mu > 0$, shape parameter $\lambda > 0$, and $x > 0$, has the probability density function
  \[\sqrt(\lambda/(2\pi x^3))\exp(-\lambda(x - \mu)^2/(2\mu^2 x)).\]

- The log logistic distribution for scale parameter $\lambda > 0$, shape parameter $\kappa > 0$, and $x \geq 0$, has a probability density function
  \[(\kappa\lambda)(x\lambda)^{(\kappa-1)}/(1 + (\lambda x)^\kappa)^2.\]
• The log normal distribution for the real-numbered mean $\mu$ of the logarithm, standard deviation $\sigma > 0$ of the logarithm, and $x > 0$, has the probability density function

$$\frac{1}{x\sigma \sqrt{2\pi}} \exp\left(-\left(\log x - \mu\right)^2/(2\sigma^2)\right).$$

• The logistic distribution for the real-numbered location parameter $\mu$, scale parameter $\sigma$, and $x$ is a real number, has the probability density function

$$\left(\frac{1}{\sigma}\right) \exp\left(\frac{(x - \mu)}{\sigma}\right) \left(1 + \exp\left(\frac{(x - \mu)}{\sigma}\right)\right)^{-2}$$

• The normal distribution for the real-numbered mean $\mu$, standard deviation $\sigma > 0$, and $x$ is a real number, has the probability density function

$$\frac{1}{\sqrt{2\pi \sigma^2}} \exp\left(\frac{-(x - \mu)^2}{2\sigma^2}\right).$$

• The uniform distribution for real-valued parameters $a$ and $b$ where $a < b$ and $a \leq x \leq b$, has the probability density function

$$\frac{1}{b - a}.$$

• The Weibull distribution for scale parameter $\lambda > 0$, shape parameter $\kappa > 0$, and $x > 0$, has the probability density function

$$\kappa \left(\frac{\lambda \kappa}{\lambda x} \right)^{\kappa - 1} \exp\left(\frac{-\left(\lambda x\right)^\kappa}{\kappa}\right).$$

Value

If the optional argument `info = TRUE` is included then a list is returned with:

• `parm1*`: a vector containing the associated confidence region boundary values for parameter 1
• `parm2*`: a vector containing the associated confidence region boundary values for parameter 2
• `phi`: a vector containing the angles used
• `parm1hat*`: the MLE for parameter 1
• `parm2hat*`: the MLE for parameter 2

*Note: "param1" and "param2" are placeholders that will be replaced with the appropriate parameter names based on the probability distribution.

Author(s)

Christopher Weld (<ceweld@email.wm.edu>)
Lawrence Leemis (<leemis@math.wm.edu>)

References


See Also
coversim, uniroot

Examples

```r
## plot the 95% confidence region for Weibull shape and scale parameters
## corresponding to the given ballbearing dataset
ballbearing <- c(17.88, 28.92, 33.00, 41.52, 42.12, 45.60, 48.48, 51.84,
51.96, 54.12, 55.56, 67.80, 68.64, 68.64, 68.88, 84.12,
93.12, 98.64, 105.12, 105.84, 127.92, 128.04, 173.40)
crplot(dataset = ballbearing, distn = "weibull", alpha = 0.05)

## repeat this plot using the elliptic-oriented point distribution heuristic
crplot(dataset = ballbearing, distn = "weibull", alpha = 0.05,
       heuristic = 0, ellipse_n = 80)

## combine the two heuristics, compensating any elliptic-oriented point vertices whose apparent
## angles > 6 degrees with additional points, and expand the plot area to include the origin
crplot(dataset = ballbearing, distn = "weibull", alpha = 0.05,
       maxdeg = 6, ellipse_n = 80, origin = TRUE)

## next use the inverse Gaussian distribution and show no plot points
crplot(dataset = ballbearing, distn = "invgauss", alpha = 0.05,
       pts = FALSE)
```
Index

* Agresti-Coull
  binomTest, 2
  binomTestCoverage, 4
  binomTestCoveragePlot, 5
  binomTestEnsemble, 7

* Arcsine
  binomTest, 2
  binomTestCoverage, 4
  binomTestCoveragePlot, 5
  binomTestEnsemble, 7

* Blaker
  binomTest, 2
  binomTestCoverage, 4
  binomTestCoveragePlot, 5

* Clopper-Pearson
  binomTest, 2
  binomTestCoverage, 4
  binomTestCoveragePlot, 5
  binomTestEnsemble, 7

* Dyck words
  binomTestMSE, 9

* Ensemble
  binomTestEnsemble, 7

* Jeffreys
  binomTest, 2
  binomTestCoverage, 4
  binomTestCoveragePlot, 5
  binomTestEnsemble, 7

* RMSE
  binomTestMSE, 9

* Wald
  binomTest, 2
  binomTestCoverage, 4
  binomTestCoveragePlot, 5
  binomTestEnsemble, 7

* Wilson-Score
  binomTest, 2
  binomTestCoverage, 4
  binomTestCoveragePlot, 5
  binomTestEnsemble, 7

* actual coverage function
  binomTestMSE, 9

* actual coverage
  binomTestCoverage, 4

* binomial distribution
  binomTest, 2
  binomTestCoverage, 4
  binomTestCoveragePlot, 5
  binomTestEnsemble, 7

* binomial proportion
  binomTest, 2
  binomTestCoverage, 4
  binomTestCoveragePlot, 5
  binomTestEnsemble, 7
  binomTestMSE, 9

* confidence intervals for binomial proportions
  binomTest, 2
  binomTestCoverage, 4
  binomTestCoveragePlot, 5
  binomTestEnsemble, 7
  binomTestMSE, 9

* confidence interval
  binomTest, 2
  binomTestCoverage, 4
  binomTestCoveragePlot, 5
  binomTestEnsemble, 7
  binomTestMSE, 9

* confidence region plot graphics
  visualization coverage parameter estimation
coversim, 11
crplot, 16

* distribution
coversim, 11
crplot, 16

* models
coversim, 11
INDEX

  crplot, 16
  * univar
    coversim, 11
    crplot, 16

  binomTest, 2, 11
  binomTestCoverage, 4, 11
  binomTestCoveragePlot, 5, 11
  binomTestEnsemble, 7, 11
  binomTestMSE, 9

  conf, 10
  coversim, 10, 11, 11, 21
  crplot, 10, 11, 15, 16

  dbinom, 3, 5, 7

  par, 13, 17

  round, 13, 17

  unirroot, 13, 15, 17, 21