Package ‘binom’

January 2, 2012

Title Binomial Confidence Intervals For Several Parameterizations

Version 1.0-5

Date 2009-05-26

Author Sundar Dorai-Raj <sundar.dorai-raj@pdf.com>

Description Constructs confidence intervals on the probability of success in a binomial experiment via several parameterizations

Maintainer Sundar Dorai-Raj <sdorairaj@gmail.com>

Depends lattice

Suggests polynom, tcltk

License GPL

Repository CRAN

Date/Publication 2009-05-26 10:32:31

R topics documented:

`binom.bayes` *Binomial confidence intervals using Bayesian inference*

Description

Uses a beta prior on the probability of success for a binomial distribution, determines a two-sided confidence interval from a beta posterior.

Usage

```
binom.bayes(x, n, conf.level = 0.95, type = c("highest", "central"),
            prior.shape1 = 0.5, prior.shape2 = 0.5,
            tol = .Machine$double.eps^0.5, maxit = 1000, ...)
```

Arguments

`x`	Vector of number of successes in the binomial experiment.
`n`	Vector of number of independent trials in the binomial experiment.
`conf.level`	The level of confidence to be used in the confidence interval.
`type`	The type of confidence interval (see Details).
`prior.shape1`	The value of the first shape parameter to be used in the prior beta.
`prior.shape2`	The value of the second shape parameter to be used in the prior beta.
`tol`	A tolerance to be used in determining the highest probability density interval.
`maxit`	Maximum number of iterations to be used in determining the highest probability interval.
`...`	Ignored.

Details

Using the conjugate beta prior on the distribution of p (the probability of success) in a binomial experiment, constructs a confidence interval from the beta posterior. From Bayes theorem the posterior distribution of p given the data x is:

`p|x ~ Beta(x + prior.shape1, n - x + prior.shape2)`

The default prior is Jeffrey's prior which is a Beta(0.5, 0.5) distribution. Thus the posterior mean is `(x + 0.5)/(n + 1)`.

The default type of interval constructed is "highest" which computes the highest probability density (hpd) interval which assures the shortest interval possible. The hpd intervals will achieve a probability that is within tol of the specified conf.level. Setting type to "central" constructs intervals that have equal tail probabilities.

If 0 or n successes are observed, a one-sided confidence interval is returned.

Value

A data.frame containing the observed proportions and the lower and upper bounds of the confidence interval.

Author(s)

Sundar Dorai-Raj (sdorairaj@gmail.com)

References

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (1997) *Bayesian Data Analysis*, London, U.K.: Chapman and Hall.

See Also

`binom.confint`, `binom.cloglog`, `binom.logit`, `binom.probit`

Examples

```
binom.bayes(x = 0:10, n = 10, tol = 1e-9)
```

`binom.cloglog` *Binomial confidence intervals using the cloglog parameterization*

Description

Uses the complementary log (cloglog) parameterization on the observed proportion to construct confidence intervals.

Usage

```
binom.cloglog(x, n, conf.level = 0.95, ...)
```

Arguments

`x`	Vector of number of successes in the binomial experiment.
`n`	Vector of number of independent trials in the binomial experiment.
`conf.level`	The level of confidence to be used in the confidence interval.
`...`	ignored

Details

For derivations see *doc/binom.pdf*.

Value

A `data.frame` containing the observed proportions and the lower and upper bounds of the confidence interval.

Author(s)

Sundar Dorai-Raj (sdorairaj@gmail.com)

See Also

binom.confint, binom.bayes, binom.logit, binom.probit, binom.coverage

Examples

```
binom.cloglog(x = 0:10, n = 10)
```

---

binom.confint *Binomial confidence intervals*

Description

Uses eight different methods to obtain a confidence interval on the binomial probability.

Usage

```
binom.confint(x, n, conf.level = 0.95, methods = "all", ...)
```

Arguments

x	Vector of number of successes in the binomial experiment.
n	Vector of number of independent trials in the binomial experiment.
conf.level	The level of confidence to be used in the confidence interval.
methods	Which method to use to construct the interval. Any combination of c("exact", "ac", "asymptotic", "wilson", "prop.test", "bayes", "logit", "cloglog", "probit") is allowed. Default is "all".
...	Additional arguments to be passed to binom.bayes.

Details

Nine methods are allowed for constructing the confidence interval(s):

- exact - Pearson-Klopper method. See also binom.test.
- asymptotic - the text-book definition for confidence limits on a single proportion using the Central Limit Theorem.
- agresti-coull - Agresti-Coull method.
- wilson - Wilson method.
- prop.test - equivalent to prop.test(x = x, n = n, conf.level = conf.level)$conf.int.
- bayes - see binom.bayes.
- logit - see binom.logit.

- `cloglog` - see `binom.cloglog`.
- `probit` - see `binom.probit`.
- `profile` - see `binom.profile`.

By default all eight are estimated for each value of `x` and/or `n`. For the "logit", "cloglog", "probit", and "profile" methods, the cases where `x == 0` or `x == n` are treated separately. Specifically, the lower bound is replaced by `(alpha/2)^n` and the upper bound is replaced by `(1-alpha/2)^n`.

Value

A `data.frame` containing the observed proportions and the lower and upper bounds of the confidence interval for all the methods in `"methods"`.

Author(s)

Sundar Dorai-Raj (sdorairaj@gmail.com)

References

A. Agresti and B.A. Coull (1998), Approximate is better than "exact" for interval estimation of binomial proportions, *American Statistician*, **52**:119-126.

R.G. Newcombe, Logit confidence intervals and the inverse sinh transformation (2001), *American Statistician*, **55**:200-202.

L.D. Brown, T.T. Cai and A. DasGupta (2001), Interval estimation for a binomial proportion (with discussion), *Statistical Science*, **16**:101-133.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (1997) *Bayesian Data Analysis*, London, U.K.: Chapman and Hall.

See Also

`binom.bayes`, `binom.logit`, `binom.probit`, `binom.cloglog`, `binom.coverage`, `prop.test`, `binom.test` for comparison to method `"exact"`

Examples

```
binom.confint(x = c(2, 4), n = 100, tol = 1e-8)
```

`binom.coverage` *Probability coverage for binomial confidence intervals*

Description

Determines the probability coverage for a binomial confidence interval.

Usage

```
binom.coverage(p, n, conf.level = 0.95, method = "all", ...)
```

Arguments

`p`	The (true) probability of success in a binomial experiment.
`n`	Vector of number of independent trials in the binomial experiment.
`conf.level`	The level of confidence to be used in the confidence interval.
`method`	Either a character string to be passed to `binom.confint` or a function that computes the upper and lower confidence bound for a binomial proportion. If a function is supplied, the first three arguments must be the same as `binom.confint` and the return value of the function must be a `data.frame` with column headers `"method"`, `"lower"`, and `"upper"`. See `binom.confint` for available methods. Default is `"all"`.
`...`	Additional parameters to be passed to `binom.confint`. Only used when method is either `"bayes"` or `"profile"`

Details

Derivations are based on the results given in the references. Methods whose coverage probabilities are consistently closer to 0.95 are more desireable. Thus, Wilson's, logit, and cloglog appear to be good for this sample size, while Jeffreys, asymptotic, and prop.test are poor. Jeffreys is a variation of Bayes using prior shape parameters of 0.5 and having equal probabilities in the tail. The Jeffreys' equal-tailed interval was created using binom.bayes using (0.5,0.5) as the prior shape parameters and `type = "central"`.

Value

A `data.frame` containing the `"method"` used, `"n"`, `"p"`, and the coverage probability, `C(p,n)`.

Author(s)

Sundar Dorai-Raj (sdorairaj@gmail.com)

References

L.D. Brown, T.T. Cai and A. DasGupta (2001), Interval estimation for a binomial proportion (with discussion), *Statistical Science*, **16**:101-133.

L.D. Brown, T.T. Cai and A. DasGupta (2002), Confidence Intervals for a Binomial Proportion and Asymptotic Expansions, *Annals of Statistics*, **30**:160-201.

See Also

`binom.confint`, `binom.length`

Examples

```
binom.coverage(p = 0.5, n = 50)
```

`binom.length` *Expected length for binomial confidence intervals*

Description

Determines the expected length for a binomial confidence interval.

Usage

```
binom.length(p, n, conf.level = 0.95, method = "all", ...)
```

Arguments

`p`	The (true) probability of success in a binomial experiment.
`n`	Vector of number of independent trials in the binomial experiment.
`conf.level`	The level of confidence to be used in the confidence interval.
`method`	Either a character string to be passed to `binom.confint` or a function that computes the upper and lower confidence bound for a binomial proportion. If a function is supplied, the first three arguments must be the same as `binom.confint` and the return value of the function must be a `data.frame` with column headers `"method"`, `"lower"` and `"upper"`. See `binom.confint` for available methods. Default is `"all"`.
`...`	Additional parameters to be passed to `binom.confint`. Only used when method is either `"bayes"` or `"profile"`

Details

Derivations are based on the results given in the references. Methods whose length probabilities are consistently closer to 0.95 are more desireable. Thus, Wilson's, logit, and cloglog appear to be good for this sample size, while Jeffreys, asymptotic, and prop.test are poor. Jeffreys is a variation of Bayes using prior shape parameters of 0.5 and having equal probabilities in the tail. The Jeffreys' equal-tailed interval was created using binom.bayes using (0.5,0.5) as the prior shape parameters and `type = "central"`.

Value

A `data.frame` containing the `"method"` used, `"n"`, `"p"`, and the average length, `L(p,n)`.

Author(s)

Sundar Dorai-Raj (sdorairaj@gmail.com)

References

L.D. Brown, T.T. Cai and A. DasGupta (2001), Interval estimation for a binomial proportion (with discussion), *Statistical Science*, **16**:101-133.

L.D. Brown, T.T. Cai and A. DasGupta (2002), Confidence Intervals for a Binomial Proportion and Asymptotic Expansions, *Annals of Statistics*, **30**:160-201.

See Also

binom.confint, binom.coverage

Examples

```
binom.length(p = 0.5, n = 50)
```

---

binom.logit — *Binomial confidence intervals using the logit parameterization*

Description

Uses the logistic (logit) parameterization on the observed proportion to construct confidence intervals.

Usage

```
binom.logit(x, n, conf.level = 0.95, ...)
```

Arguments

x	Vector of number of successes in the binomial experiment.
n	Vector of number of independent trials in the binomial experiment.
conf.level	The level of confidence to be used in the confidence interval.
...	ignored

Details

For derivations see *doc/binom.pdf*.

Value

A `data.frame` containing the observed proportions and the lower and upper bounds of the confidence interval.

Author(s)

Sundar Dorai-Raj (sdorairaj@gmail.com)

See Also

binom.confint, binom.bayes, binom.logit, binom.probit, binom.coverage

Examples

```
binom.logit(x = 0:10, n = 10)
```

binom.lrt *Binomial confidence intervals using the lrt likelihood*

Description

Uses the lrt likelihood on the observed proportion to construct confidence intervals.

Usage

```
binom.lrt(x, n, conf.level = 0.95, bayes = FALSE, conf.adj = FALSE, plot
= FALSE, ...)
```

Arguments

`x`	Vector of number of successes in the binomial experiment.
`n`	Vector of number of independent trials in the binomial experiment.
`conf.level`	The level of confidence to be used in the confidence interval.
`bayes`	logical; if `TRUE` use a Bayesian correction at the edges. Specfically, a beta prior with shape parameters 0.5 is used. If `bayes` is numeric, it is assumed to be the parameters to beta distribution.
`conf.adj`	logical; if `TRUE` 0 or 100% successes return a one-sided confidence interval
`plot`	logical; if `TRUE` a plot showing the the square root of the binomial deviance with reference lines for mean, lower, and upper bounds. This argument can also be a `list` of plotting parameters to be passed to `xyplot`.
`...`	ignored

Details

Confidence intervals are based on profiling the binomial deviance in the neighbourhood of the MLE. If `x == 0` or `x == n` and `bayes` is `TRUE`, then a Bayesian adjustment is made to move the log-likelihood function away from `Inf`. Specifically, these values are replaced by `(x + 0.5)/(n + 1)`, which is the posterier mode of `f(p|x)` using Jeffrey's prior on `p`. Furthermore, if `conf.adj` is `TRUE`, then the upper (or lower) bound uses a `1 - alpha` confidence level. Typically, the observed mean will not be inside the estimated confidence interval. If `bayes` is `FALSE`, then the Clopper-Pearson exact method is used on the endpoints. This tends to make confidence intervals at the end too conservative, though the observed mean is guaranteed to be within the estimated confidence limits.

Value

A `data.frame` containing the observed proportions and the lower and upper bounds of the confidence interval.

Author(s)

Sundar Dorai-Raj (sdorairaj@gmail.com)

See Also

binom.confint, binom.bayes, binom.cloglog, binom.logit, binom.probit, binom.coverage, confint in package MASS, family, glm

Examples

```
binom.lrt(x = 0:10, n = 10)
```

---

binom.optim *Optimal binomial confidence intervals*

Description

Uses optimization to minimize the integrated mean squared error between the calculated coverage and the desired confidence level for a given binomial confidence interval.

Usage

```
binom.optim(n, conf.level = 0.95, method = binom.lrt,
            k = n%/%2 + 1, p0 = 0, transform = TRUE,
            plot = FALSE, tol = .Machine$double.eps^0.5,
            start = NULL, ...)
```

Arguments

n	The number of independent trials in the binomial experiment.
conf.level	The level of confidence to be used in the confidence interval.
method	The method used to estimate the confidence interval.
k	See Details.
p0	The minimum probability of success to allow in the optimization. See Details.
transform	logical; If TRUE the optimizer will do an unconstrained optimization on the significance probability in the logit space.
plot	logical; If TRUE the results are sent to binom.plot.
tol	The minimum significance level to allow in the optimization. See Details.
start	A starting value on the optimal confidence level.
...	Additional arguments to pass to optim.

Details

This function minimizes the squared error between the expected coverage probability and the desired confidence level.

$$\alpha_{opt} = \arg\min_{\alpha} \int_0^1 [C(p,n) - (1-\alpha)^2 dp$$

The optimizer will adjust confidence intervals for all `x = 0` to `n` depending on the value of `k` provided. If `k` is one, only the confidence levels for `x = 0` and `n` are adjusted. If `k = [n/2]` then all confidence intervals are adjusted. This assumes the confidence intervals are the same length for `x = x[k]` and `x[n - k + 1]`, which is the case for all methods provided in this package except `binom.cloglog`.

Value

A `list` with the following elements:

`par`	Final confidence levels. The length of this vector is `k`.
`value`	The final minimized value from `optim`.
`counts`	The number of function and gradient calls from `optim`.
`convergence`	Convergence code from `optim`.
`message`	Any message returned by the L-BFGS-B or BFGS optimizer.
`confint`	A `data.frame` returned from a call to `method` using the optimized confidence levels.

Author(s)

Sundar Dorai-Raj (sdorairaj@gmail.com)

See Also

`binom.confint`, `binom.plot`, `binom.coverage`, `optim`

Examples

```
binom.optim(10, k = 1) ## determine optimal significance for x = 0, 10 only
binom.optim(3, method = binom.wilson) ## determine optimal significance for all x
```

binom.plot — *Coverage plots for binomial confidence intervals*

Description

Constructs coverage plots for binomial confidence intervals.

Usage

```
binom.plot(n, method = binom.lrt, np = 500,
           conf.level = 0.95, actual = conf.level,
           type = c("xyplot", "levelplot"),
           tol = .Machine$double.eps^0.5, ...)
```

Arguments

`n`	The number of independent trials in the binomial experiment.
`conf.level`	The level of confidence to be used in the confidence interval.
`np`	Number of points to use in the plot.
`method`	The method used to estimate the confidence interval.
`actual`	The actual confidence interval used in the confidence interval. See Details.
`type`	See Details.
`tol`	The minimum probability of success to use in the plot.
`...`	Additional arguments to pass to `panel.xyplot` or `panel.levelplot`.

Details

If `type` is `"xyplot"`, a line plot is created with coverage on the y-axis and binomial probability on the x-axis. A separate panel for every `n` is provided. If `actual` is provided then a horizontal reference line is added to the plot. This is only useful when `actual` is different from `conf.level`, as is the case when calling `binom.optim`.

If `type` is `"levelplot"`, a image plot is created with `x = 0` to `n` on the vertical axis and binomial probability on the horizontal axis. Each row in the plot will be the confidence level for a given `x`. The color of the confidence interval is determined by the coverage probability. The argument `n` must only be of length one. If not, only the first `n` will be used and a warning is issued.

In either plot type, the number of points at which the coverage probability is determined is specified by `np`. Increasing `np` gives a finer granularity but performance will suffer.

Value

An object of class `trellis`.

Author(s)

Sundar Dorai-Raj (sdorairaj@gmail.com)

See Also

binom.confint, binom.optim, xyplot, levelplot

Examples

```
binom.plot(5, type = "levelplot")
binom.plot(c(3, 5, 10, 25), type = "xyplot")
```

---

binom.power *Power curves for binomial parameterizations*

Description

Uses Wald statistics to compute power curves for several parameterizations.

Usage

```
binom.power(p.alt, n = 100, p = 0.5, alpha = 0.05, phi = 1,
            alternative = c("two.sided", "greater", "less"),
            method = c("cloglog", "logit", "probit", "asymp", "lrt", "exact"))
```

Arguments

p.alt	Probability of success under the alternative hypothesis.
n	Vector of number of independent trials in the binomial experiment.
p	Probability of success under the null hypothesis.
alpha	Type-I error rate.
phi	Overdispersion parameter.
alternative	Type of alternative hypothesis.
method	The method used to compute power.

Details

For derivations see *doc/binom.pdf*.

Value

The estimated probability of detecting the difference between p.alt and p.

Author(s)

Sundar Dorai-Raj (sdorairaj@gmail.com)

See Also

binom.confint, binom.bayes, binom.logit, binom.probit, binom.coverage

Examples

```
binom.power(0.95, alternative = "greater")
```

`binom.probit` *Binomial confidence intervals using the probit parameterization*

Description

Uses the probit parameterization on the observed proportion to construct confidence intervals.

Usage

```
binom.probit(x, n, conf.level = 0.95, ...)
```

Arguments

`x`	Vector of number of successes in the binomial experiment.
`n`	Vector of number of independent trials in the binomial experiment.
`conf.level`	The level of confidence to be used in the confidence interval.
`...`	ignored

Details

For derivations see *doc/binom.pdf*.

Value

A `data.frame` containing the observed proportions and the lower and upper bounds of the confidence interval.

Author(s)

Sundar Dorai-Raj (sdorairaj@gmail.com)

See Also

`binom.confint`, `binom.bayes`, `binom.probit`, `binom.logit`, `binom.coverage`

Examples

```
binom.probit(x = 0:10, n = 10)
```

`binom.profile` *Binomial confidence intervals using the profile likelihood*

Description

Uses the profile likelihood on the observed proportion to construct confidence intervals.

Usage

```
binom.profile(x, n, conf.level = 0.95, maxsteps = 50,
              del = zmax/5, bayes = TRUE, plot = FALSE, ...)
```

Arguments

`x`	Vector of number of successes in the binomial experiment.
`n`	Vector of number of independent trials in the binomial experiment.
`conf.level`	The level of confidence to be used in the confidence interval.
`maxsteps`	The maximum number of steps to take in the profiles.
`del`	The size of the step to take
`bayes`	logical; if `TRUE` use a Bayesian correction at the edges.
`plot`	logical; if `TRUE` plot the profile with a `spline` fit.
`...`	ignored

Details

Confidence intervals are based on profiling the binomial deviance in the neighbourhood of the MLE. If `x == 0` or `x == n` and `bayes` is `TRUE`, then a Bayesian adjustment is made to move the log-likelihood function away from `Inf`. Specifically, these values are replaced by `(x + 0.5)/(n + 1)`, which is the posterier mode of `f(p|x)` using Jeffrey's prior on `p`. Typically, the observed mean will not be inside the estimated confidence interval. If `bayes` is `FALSE`, then the Clopper-Pearson exact method is used on the endpoints. This tends to make confidence intervals at the end too conservative, though the observed mean is guaranteed to be within the estimated confidence limits.

Value

A `data.frame` containing the observed proportions and the lower and upper bounds of the confidence interval.

Author(s)

Sundar Dorai-Raj (sdorairaj@gmail.com)

See Also

`binom.confint`, `binom.bayes`, `binom.cloglog`, `binom.logit`, `binom.probit`, `binom.coverage`, `confint` in package MASS, `family`, `glm`

Examples

```
binom.profile(x = 0:10, n = 10)
```

---

binom.sim *Simulates confidence intervals for binomial data*

Description

Simulates binomial data for testing confidence interval coverage.

Usage

```
binom.sim(M = 200, n = 100, p = 0.5, conf.level = 0.95, methods = "all",  ...)
```

Arguments

`M`	Number of simulations to create.
`n`	Vector of number of independent trials in the binomial experiment.
`p`	Probability of success under the null hypothesis.
`conf.level`	The level used in computing the confidence interval.
`methods`	The method used to compute power.
`...`	Additional arguments to pass to `binom.confint`

Details

`M` binomial observations are created using `rbinom(M, n, p)`. The average number of times a confidence interval covers `p` is returned.

Value

The estimated coverage based on which method is requested.

Author(s)

Sundar Dorai-Raj (sdorairaj@gmail.com)

See Also

`binom.confint`, `binom.bayes`, `binom.logit`, `binom.probit`, `binom.coverage` cloglog.sample.size *Power and sample size*

Description

Power and sample size for a binomial proportion using the cloglog parameterization.

Usage

```
cloglog.sample.size(p.alt, n = NULL, p = 0.5, power = 0.8, alpha = 0.05,
                alternative = c("two.sided", "greater", "less"), exact.n = FALSE,
                     recompute.power = FALSE, phi = 1)
```

Arguments

`p.alt`	The alternative proportion in a one-sample test.
`n`	The sample size in a one-sample test.
`p`	The null proportion in a one-sample test. Default is 0.5.
`power`	The desired power level. Default is 0.80.
`alpha`	The desired alpha level - probability of a Type I error. Default is 0.05.
`alternative`	Nature of alternative hypothesis. One of "two.sided", "greater", "less".
`exact.n`	logical; If `TRUE`, the computed sample size will not be rounded up. Default is `FALSE`.
`recompute.power`	logical; If `TRUE`, after the sample size is computed, the power will be recomputed. This is only advantageous when the sample size is rounded up. Default is `FALSE`.
`phi`	Dispersion parameter by which to inflate (`phi > 1`) or deflate (`phi < 1`) variance. Default is 1.

Details

This function can be used to calculate sample size, power or minimum detectable difference. It determines what to compute base on the arguments provided. If `p.alt` is given, but `n` is not, then sample size is computed. If `p.alt` is given along with `n`, then the power is computed. If only `n` is provided, the minimum detectable difference is computed using the default power of 0.80.

Value

A `data.frame` containing the power, sample size and all of the input which was used to perform the computations.

Author(s)

Sundar Dorai-Raj (sdorairaj@gmail.com)

See Also

binom.confint

Examples

```
cloglog.sample.size(p.alt = 0.8)
cloglog.sample.size(n = 20)
cloglog.sample.size(n = 20, power = 0.9)
```

---

tkbinom.power *Power curves for binomial parameterizations*

Description

A Tcl/Tk graphics wrapper for binom.power.

Usage

```
tkbinom.power()
```

Details

A wrapper for binom.power that creates power curves based on user input.

Value

None.

Author(s)

Sundar Dorai-Raj (sdorairaj@gmail.com)

See Also

binom.power, binom.confint, binom.bayes, binom.logit, binom.probit, binom.coverage

Examples

```
binom.power(0.95, alternative = "greater")
```

Index