Package: exactci (via r-universe)

May 12, 2026

Type Package

Title Exact P-Values and Matching Confidence Intervals for Simple Discrete Parametric Cases

Version 1.4-5

Date 2025-08-20

Depends stats, ssanv, testthat

Suggests rmarkdown, BlakerCI, knitr, exact2x2

VignetteBuilder knitr

Description Calculates exact tests and confidence intervals for one-sample binomial and one- or two-sample Poisson cases (see Fay (2010) <doi:10.32614/rj-2010-008>).

License GPL-3

LazyLoad yes

NeedsCompilation no

Author Michael P. Fay [aut, cre]

Maintainer Michael P. Fay <mfay@niaid.nih.gov>

Config/pak/sysreqs cmake make libuv1-dev

Repository https://michaelfay-niaid.r-universe.dev

Date/Publication 2025-08-20 21:10:01 UTC

RemoteUrl https://github.com/cran/exactci

RemoteRef HEAD

RemoteSha a78129c3cea0a3dbb9a8bebe0c5937823074dd5d

Contents

`exactci-package` *Exact binomial and Poisson tests with Matching Confidence Intervals*

Description

Calculates exact binomial and Poisson tests giving matching confidence intervals. There are 3 different methods for defining the two-sided p-values.

Details

Although `binom.test` and `poisson.test` give exact tests and confidence intervals, for two-sided tests the confidence intervals (CI) are not formed by inverting the tests. Thus, there may be test-CI inconsistencies whereby the test rejects but the confidence interval contains the null parameter. The `exactci` package eliminates many of these test-CI inconsistencies for two-sided tests, by outputing the matching confidence interval with each test. The package uses one of three different methods for defining the two-sided p-value. The main functions of the package are `binom.exact` and `poisson.exact` which follow the same format as `binom.test` and `poisson.test` except have the option 'tsmethod' to define the two-sided method for calculating the p-values, and give matching confidence intervals (i.e., ones that come from the inversion of the p-values).

The package also has options for mid-p values.

Author(s)

Michael P. Fay

Maintainer: Michael P. Fay <mfay@niaid.nih.gov>

References

Blaker, H. (2000) Confidence curves and improved exact confidence intervals for discrete distributions. Canadian Journal of Statistics 28: 783-798.

Fay, M. P. (2010). Confidence intervals that Match Fisher's exact and Blaker's exact tests. Biostatistics. 11:373-374.

Fay, M.P. (2010). Two-sided Exact Tests and Matching Confidence Intervals for Discrete Data. R Journal 2(1): 53-58.

Hirjim K. F. (2006). Exact analysis of discrete data. Chapman and Hall/CRC. New York.

See Also

For comparisons of two binomial groups see `exact2x2`

Examples

```
## Note binom.test calculates p-values using priciple of minimum likelihood
## while it calculates the central confidence intervals. That is why the
## inferences do not match in this example.
binom.test(10,12,p=20000/37877)
binom.exact(10,12,p=20000/37877,tsmethod="minlike")
binom.exact(10,12,p=20000/37877,tsmethod="central")
## We also allow the method studied in Blaker (2000)
binom.exact(10,12,p=20000/37877,tsmethod="blaker")
```

`binom.exact` *Exact tests with matching confidence intervals for single binomial parameter*

Description

Calculates exact p-values and confidence intervals for a single binomial parmeter. This is different from `binom.test` only when alternative='two.sided', in which case `binom.exact` gives three choices for tests based on the 'tsmethod' option. The resulting p-values and confidence intervals will match.

Usage

```
binom.exact(x, n, p = 0.5,
   alternative = c("two.sided", "less", "greater"),
   tsmethod = c("central", "minlike", "blaker"),
   conf.level = 0.95,
   control=binomControl(),plot=FALSE, midp=FALSE)
```

Arguments

`x`	number of successes, or a vector of length 2 giving the numbers of successes and failures, respectively.
`n`	number of trials, ignored if x has length 2.
`p`	hypothesized probability of success.
`alternative`	indicates the alternative hypothesis and must be one of "two.sided", "greater" or "less". You can specify just the initial letter.
`tsmethod`	indicates the method for a two-sided alternative hypothesis and must be one of "minlike", "central" or "blaker". You can specify just the initial letter.
`conf.level`	confidence level for the returned confidence interval.
`control`	list with settings to avoid problems with ties, etc, should not need to change this for normal use, see `binomControl`
`plot`	logical, do basic plot of p-value function by null hypothesis value, see `exactbinomPlot` for more plot options
`midp`	logical, use mid-p for p-values and confidence intervals? midp Confidence intervals are not available for tsmethod='minlike' and 'blaker'

Details

Traditionally, hypothesis tests and confidence intervals are treated separately. A more unified approach suggested by Hirji (2006) is to use the same p-value function to create confidence intervals. There is essentially only one way to calculate one-sided p-values and confidence intervals so these methods are the same in `binom.test` and `binom.exact`. However, there are three main ways that `binom.exact` allows for defining two-sided p-values.

minlike:	sum probabilities of all likelihoods equal or less than observed
central:	double minimum one-sided p-value
blaker:	combine smaller observed tail probability with opposite tail not greater than observed tail

The 'minlike' method is the p-value that has been used in `binom.test`, and 'blaker' is described in Blaker (2000) or Hirji (2006), where it is called the 'combined tails' method. Once the p-value function is defined we can invert the test to create 'matching' confidence intervals defined as the smallest interval that contains all parameter values for which the two-sided hypothesis test does not reject. There are some calculation issues for the 'minlike' and 'blaker' methods which are the same as for exact tests for 2x2 tables (see Fay, 2010).

All of the above traditional p-values can be thought of as estimating Pr[X=xobs or X is more extreme than xobs] under the null hypothesis, where more extreme is defined differently for different methods. The mid-p-value replaces this with 0.5*Pr[X=xobs]+ Pr[X is more extreme than xobs]. The mid-p p-values are not valid. In other words, for all parameter values under the null hypothesis we are not guaranteed to bound the type I error rate. However, the usual exact methods that guarantee the type I error rate are typically conservative for most parameter values in order to bound the type I error rate for all parameter values. So if you are interested in rejecting approximately on average about 5 percent of the time for arbitrary parameter values and n values under the null hypothesis, then use midp=TRUE. If you want to ensure bounding of the type I errror rate for all n and all parameter values use midp=FALSE. (See for example, Vollset, 1993, or Hirji, 2006).

The associated midp confidence intervals have not been programmed for tsmethod='blaker' and 'minlike'.

Value

An object of class 'htest': a list with items

`p.value`	p-value
`conf.int`	confidence interval, see attributes 'conf.level' and perhaps 'conf.limit.prec'
`statistic`	number of successes
`parameter`	number of trials
`estimate`	observed proportion of success
`null.value`	null hypothesis probability of success, 'p'
`alternative`	a character string describing alternative hypothesis
`method`	a character string describing method
`data.name`	a character string giving the names of the data

Note

The 'central' method gives the Clopper-Pearson intervals, and the 'minlike' method gives confidence intervals proposed by Stern (1954) (see Blaker, 2000). The 'blaker' method is guaranteed to be more powerful than the 'central' method (see Blaker, 2000, Corollary 1), but both the 'blaker' method and 'minlike' method may have some undesireable properties. For example, there are cases where adding an additional Bernoulli observation REGARDLESS OF THE RESPONSE will increase the p-value, see Vos and Hudson (2008). The 'central' method does not have those undesireable properties.

The Blyth-Still-Casella intervals given in StatXact (and not by `binom.exact`) are the shortest possible intervals, but those intervals are not nested. This means that the Blyth-Still-Casella intervals are not guaranteed to have the 95 percent interval contain the 90 percent interval. See Blaker (2000) Theorem 2.

Author(s)

M.P. Fay

References

Blaker, H. (2000) Confidence curves and improved exact confidence intervals for discrete distributions. Canadian Journal of Statistics 28: 783-798.

Fay, M. P. (2010). Confidence intervals that Match Fisher's exact and Blaker's exact tests. Biostatistics. 11:373-374.

Fay, M.P. (2010). Two-sided Exact Tests and Matching Confidence Intervals for Discrete Data. R Journal 2(1): 53-58.

Hirji K. F. (2006). Exact analysis of discrete data. Chapman and Hall/CRC. New York.

Stern, T (1954). Some remarks on confidence and fiducial limits. Biometrika, 275-278.

Vollset, S. E. (1993). Confidence intervals for a binomial proportion. Statistics in medicine, 12(9), 809-824.

Vos, P.W. and Hudson, S. (2008). Problems with binomial two-sided tests and the associated confidence interals. Aust. N.Z. J. Stat. 50: 81-89.

See Also

`binom.test`, for two-sample exact binomial tests see `exact2x2`

Examples

```
## Notice how binom.test p-value is given by tsmethod='minlike'
## but the confidence interval is given by tsmethod='central'
## in binom.exact p-values and confidence intervals match
binom.test(10,12,p=20000/37877)
binom.exact(10,12,p=20000/37877,tsmethod="minlike")
binom.exact(10,12,p=20000/37877,tsmethod="central")
binom.exact(10,12,p=20000/37877,tsmethod="blaker")
## one-sided methods are also available
## as in binom.test
```

binomControl *Tuning parameters for binom.exact function*

Description

This function produces a list of tuning parameters used in the calculations done by binom.exact and poisson.exact. These will not need to be changed by most ordinary users.

Usage

```
binomControl(relErr=1+1e-07,tol=.00001,
    pRange=c(1e-10,1-1e-10),
    minn=1, maxn=10^4,
    PRINT.STEPS=FALSE)
```

Arguments

relErr	value very close to 1, used in calculation of two-sided p-values
tol	value very close to 0, used in calculation of two-sided confidence intervals
pRange	range close to [0,1], but excluding the endpoints, used in calculation of two-sided confidence intervals
minn	minimum n used by powerBinom. The search for the lowest n that gives power is crude and goes from minn to maxn by ones.
maxn	maximum n used by powerBinom
PRINT.STEPS	logical, print steps of uniroot.integer used to find the n when type='cilength' in powerBinom?

Details

See the code for fisher.test, where the term relErr is hard-coded into the function. The purpose is to avoid problems with ties. It serves the same purpose in this package and probably need not be changed. The value tol indicates the tolerance for the precision of the confidence limits. The value pRange is input into uniroot to give bounds when searching for confidence limits. For poisson limits pRange is transformed using the qgamma function (see code in exactpoissonCI).

Value

A list with containing the following components:

relErr	a number larger than 1
tol	a number greater than 0
pRange	a vector with 2 elements between 0 and 1, exclusive

exactbinomPlot *Plot p-value function for one binomial response.*

Description

Plots p-values as a function of different point null hypothesis values for p. For two-sided p-values, can plot three types of p-values: the minimum likelihood method (default for binom.test), the central method (i.e., twice the one-sided exact p-values), and Blaker's exact.

Usage

```
exactbinomPlot(x, n, p = NULL, ndiv = 1000,
    tsmethod = "central", pRange = c(0, 1),
    dolines = FALSE, dopoints = TRUE, doci=TRUE,
    alternative=c("two.sided","less","greater"),
    relErr=1+10^(-7), conf.level=.95, alphaline=TRUE,
    newplot = TRUE, midp=FALSE, ...)
```

Arguments

x	number of successes, or a vector of length 2 giving the numbers of successes and failures, respectively
n	number of trials, ignored if x has length 2
p	null values of p for plot, if NULL divides pRange into ndiv pieces
ndiv	number of pieces to divide up range of x-axis
tsmethod	two-sided method for p-value calculation, either "minlike","blaker" or "central"
pRange	range for plotting null hypothesis values of p
dolines	logical, add lines to a plot?
dopoints	logical, add points to a plot?
doci	logical, add lines for confidence interval?
alternative	type of alternative for p-values
relErr	number close to 1, avoids problems with ties, see binomControl
conf.level	confidence level for use when doci=TRUE
alphaline	logical, if doci=TRUE should a line be drawn at significance level
newplot	logical,start a new plot?
midp	logical, use mid-pvalues? See link{binom.exact}
...	values passed to plot, points, or lines statement

See Also

binom.exact

Examples

```
## plot central two-sided p-values (double one-sided p-values)
## for 5 positive responses out of 17 tries
exactbinomPlot(5,17)
## add blakers exact p-values
## pch option acts on points, lty acts on ci lines
exactbinomPlot(5,17,tsmethod="blaker",col="blue",pch=".",lty=2,newplot=FALSE)

## can plot one-sided p-values, tsmethod is ignored
exactbinomPlot(5,17,alternative="less")
```

`exactpoissonPlot` *Plot p-value function for single or pair of poisson responses.*

Description

Plots p-values as a function of different point null hypothesis values for rate. For two-sided p-values, can plot three types of p-values: the minimum likelihood method (default for poisson.test), the central method (i.e., twice the one-sided exact p-values), and Blaker's exact.

Usage

```
exactpoissonPlot(x,
    T=1,
    r=NULL,
    ndiv=1000,
    tsmethod="central",
    rRange=NULL,
    dolog=TRUE,
    dolines=FALSE,
    dopoints=TRUE,
    doci=TRUE,
    alternative = c("two.sided", "less", "greater"),
    relErr=1 + 10^(-7),
    conf.level=.95,
    alphaline=TRUE,
    newplot=TRUE,
    midp=FALSE,...)
```

Arguments

x	number of events. A vector of length one or two
T	time base for event count. A vector of length one or two
r	null values of rate for plot, if NULL divides rRange into ndiv pieces
ndiv	number of pieces to divide up range of x-axis

tsmethod	two-sided method for p-value calculation, either "minlike","blaker" or "central"
rRange	range for plotting null hypothesis values of rate, if null then uses confidence interval to determine range
dolog	logical, plot horizontal axis in log scale?
dolines	logical, add lines to a plot?
dopoints	logical, add points to a plot?
doci	logical, add lines for confidence interval?
alternative	type of alternative for p-values
relErr	number close to 1, avoids problems with ties, see binomControl
conf.level	confidence level for use when doci=TRUE
alphaline	logical, if doci=TRUE should line be drawn at significance level
newplot	logical,start a new plot?
midp	logical, use mid-p for p-values? Not available for tsmethod='blaker' or 'minlike'
...	values passed to plot, points, or lines statement

Value

Does graph or adds lines or points. Returns (invisibly, see invisible) a list with elements r (null hypothesis values) and p.value (associated p-values).

See Also

binom.exact

Examples

```
## single Poisson response
exactpoissonPlot(2,17877)
```

---

poisson.exact *Exact Poisson tests with Matching Confidence Intervals*

Description

Performs an exact test of a simple null hypothesis about the rate parameter in Poisson distribution, or for the ratio between two rate parameters. This is different from poisson.test in that 3 different types of exact two-sided tests (and the matching confidence intervals) are offered. The one-sided tests are the same as in poisson.test.

Usage

```
poisson.exact(x, T = 1, r = 1,
    alternative = c("two.sided", "less", "greater"),
    tsmethod=c("central","minlike","blaker"),
    conf.level = 0.95, control=binomControl(), plot=FALSE,
    midp=FALSE)
```

Arguments

x	number of events. A vector of length one or two.
T	time base for event count. A vector of length one or two.
r	hypothesized rate or rate ratio
alternative	indicates the alternative hypothesis and must be one of "two.sided", "greater" or "less". You can specify just the initial letter.
tsmethod	character giving two-sided method, one of "central", "minlike" or "blaker", ignored if alternative not equal "two.sided"
conf.level	confidence level for the returned confidence interval.
control	list with settings to avoid problems with ties, etc, should not need to change this for normal use, see binomControl
plot	logical, plot p-value function? For finer control on plot see exactpoissonPlot
midp	logical, use mid-p p-values? Only allowed for two-sided tests if tsmethod='central' (see details)

Details

Confidence intervals are computed similarly to those of binom.exact in the one-sample case, in that there are three two-sided options depending on the tsmethod. For the one-sample case the default intervals use tsmethod="central" giving the Garwood (1936) exact central confidence intervals. For the two-sample case we condition on the total counts and then use binomial methods, see Lehmann and Romano (2005) for that motivation and vignette("exactci") for description of the three different two-sided methods for calculating p-values and confidence intervals.

Traditional p-values can be thought of as estimating Pr[X=xobs or X is more extreme than xobs] under the null hypothesis, where more extreme is defined differently for different methods. The mid-p-value replaces this with 0.5*Pr[X=xobs]+ Pr[X is more extreme than xobs]. The mid-p p-values are not valid. In other words, for all parameter values under the null hypothesis we are not guaranteed to bound the type I error rate. However, the usual exact methods that guarantee the type I error rate are typically conservative for most parameter values in order to bound the type I error rate for all parameter values. So if you are interested in rejecting approximately on average about 5 percent of the time for arbitrary parameter values under the null hypothesis, then use midp=TRUE. If you want to ensure bounding of the type I errror rate for all parameter values use midp=FALSE. For a comprehensive discussions of exact inferences including mid-p values see Hirji (2006). Mid-p confidence intervals are calculated by inverting the mid-p-value function. For discussion of mid-p confidence intervals for Poisson see Cohen and Yang (1994).

The mid-p p-values and confidence intervals have not been programmed for the 'blaker' and 'minlike' tsmethods.

Value

A list with class "htest" containing the following components:

statistic	the number of events (in the first sample if there are two.)
parameter	the corresponding expected count
p.value	the p-value of the test.
conf.int	a confidence interval for the rate or rate ratio.
estimate	the estimated rate or rate ratio.
null.value	the rate or rate ratio under the null, r.
alternative	a character string describing the alternative hypothesis.
method	the character string "Exact Poisson test" or "Comparison of Poisson rates" as appropriate.
data.name	a character string giving the names of the data.

Note

The rate parameter in Poisson data is often given based on a "time on test" or similar quantity (person-years, population size). This is the role of the T argument.

References

Cohen and Yang (1994). Mid-p Confidence intervals for the Poisson Expectation. Statistics in Medicine. 13: 2189-2203.

Fay, M.P. (2010). Two-sided Exact Tests and Matching Confidence Intervals for Discrete Data. R Journal 2(1): 53-58.

Garwood, F (1936). Fiducial limits for the Poisson distribution. Biometrika, 437-442.

Hirji K. F. (2006). Exact analysis of discrete data. Chapman and Hall/CRC. New York.

Lehmann, EL, and Romano, JP (2005). Testing Statistical Hypotheses, third edition. Springer:New York.

See Also

poisson.test, exactpoissonPlot,

Examples

```
### Suppose you have observed rates of  2 out of 17877 in group A
### and 10 out of 20000 in group B
### poisson.test gives non-matching confidence intervals
### i.e., p-value using  'minlike' criteria but confidence interval using 'central' criteria
poisson.test(c(2,10),c(17877,20000))
### poisson.exact gives matching CI to the p-values
### defaults to 'central' two-sided method
poisson.exact(c(2,10),c(17877,20000))
### other options
poisson.exact(c(2,10),c(17877,20000),tsmethod="minlike")
```

```
poisson.exact(c(2,10),c(17877,20000),tsmethod="blaker")


## Mid-p confidence intervals do not guarantee coverage,
## but are more likely to have on average closer nominal
## coverage than exact ones (sometimes erroring on the
## too liberal side).
##
## To test the software, here is Table I of Cohen and Yang
## values are equal to the first 2 decimal places
yCY<-c(0:20,20+(1:5)*2,30+(1:14)*5)
TableICohenYang<-matrix(NA,length(yCY),6,dimnames=list(yCY,
    c("90pct LL","90pct UL","95pct LL","95pct UL","99pct LL","99pct UL")))

for (i in 1:length(yCY)){
    TableICohenYang[i,1:2]<-poisson.exact(yCY[i],
        midp=TRUE,conf.level=.9)$conf.int
    TableICohenYang[i,3:4]<-poisson.exact(yCY[i],
        midp=TRUE,conf.level=.95)$conf.int
    TableICohenYang[i,5:6]<-poisson.exact(yCY[i],
        midp=TRUE,conf.level=.99)$conf.int
}
TableICohenYang<-round(TableICohenYang,3)
TableICohenYang
```

---

powerBinom *Exact Power and Sample Size methods for Bernoulli responses*

Description

Calculates sample sizes or power for a study of n independent Bernoulli responses (i.e., a binomial response with parameter n). There are three types of calculations. For type='standard' calculate the usual sample size or power under a given alternative. The option type='cilength' bases calculations on the expected length of the confidence intervals. For type='obs1ormore' calculate the sample size or power to observe 1 or more positive responses. All calculations use exact methods.

Usage

```
powerBinom(n = NULL, p0 = 0.5, p1 = NULL,
    sig.level = 0.05, power = NULL,
    alternative = c("two.sided", "one.sided"), strict = FALSE,
    type = c("standard","cilength","obs1ormore"),
    cilength = NULL, conf.level = 0.95,
    direction = c("greater","less"),
    control = binomControl(), ...)
```

Arguments

n	sample size
p0	probability of success under null
p1	probability of success under alternative
sig.level	significance level of test
power	target power of test (type='standard') or probability of observing at least 1 success (type='obs1ormore')
alternative	one- or two-sided test
strict	use strict interpretation in two-sided case (count rejections in opposite direction)
type	either 'standard', 'cilength', or 'obs1ormore' (see details)
cilength	average length of confidence interval (used when type='cilength')
conf.level	confidence level (used when type='cilength')
direction	direction of alternative, either 'greater' or 'less' (used when type='standard' and p1 is NULL)
control	a list with arguments that control algorithms, see binomControl
...	further arguments passed to binom.exact (see details)

Details

Type='standard' calculates the power to reject the null hypothesis with the parameter p0 on the boundary between the null and alternative. In other words, the null could be either of the three, H0: p=p0 (for alternative='two.sided') or H0: p<=p0 or H0:p>=p0 (for alternative='one.sided'). For one-sided alternatives, p1 determines the alternative (and hence determines the null). For example, p0=.5 and p1=.7 will test H0: p<=0.5 vs H1: p> 0.5.

Type='cilength' calculates the expected length of the confidence intervals when p=p1. If p1=NULL, then it uses p1=0.5 because this will give the largest CI lengths. For sample size calculations, the n value is found using the uniroot.integer function searching the range from control$minn to control$maxn, finding the smallest n that has expected CI length at least as large as cilength.

Type='obs1ormore' calculates sample sizes related to the probability (i.e., the power) to observe at least one success.

Here are some details of the calculation method for type='standard' and type='cilength'. We use control$pRange (which should be from a vector of length two giving an interval very close to (0,1) but excluding the ends) to save computation time. We do not need to calculate the powers or expected CI lengths for all the possible values of X from 0:n. The algorithm only uses X from the likely values, using control$pRange to determine the quantiles that dominate the power or expectation calculation. For example, if n=1000 and pRange[1]=10^-10 and p1=0.5, then there is no need to calculate the CI lengths for x=0:399, because pbinom(399,1000,.5)< 10^-10.

For the ... argument, you cannot pass the 'alternative' argument to binom.exact, since it has a different form in powerBinom since p1 determines which one-sided alternative will be used.

Value

a object of class power.htest. List with input arguments plus calculated values, together with 'note' and 'method' character vectors describing the method.

Note

Often you will get the same power with midp=FALSE (default) and midp=TRUE. This is because the rejection region may be the same for both cases. For example, for type='standard' with n=20 using the default two.sided 0.05 exact central test, regardless of whether midp=FALSE or TRUE, we reject p0=0.5 for x in (0:5 and 15:20).

Author(s)

Michael P. Fay

Examples

```
# find power to reject H0: p = p0
# when p=p1 (only counting rejections in the correct direction)
# defaults to exact central test of binom.exact
powerBinom(n=40,p0=.4,p1=.6)
# The following calculates the sample size
# to have the average length of confidence intervals
# be at least 0.4 (regardless of the true proportion)
powerBinom(type="cilength",cilength=.4)
# The following answers the question:
# if the true proportion is .001, how many
# observations do you need to sample to
# have the probability of seeing at least 1 success be
# at least .9?
powerBinom(power=.9,p1=.001,type="obs1ormore")
```

Index