# Package 'Exact'

January 20, 2025

**Type** Package

**Title** Unconditional Exact Test

**Version** 3.3

**Author** Peter Calhoun [aut, cre]

**Maintainer** Peter Calhoun <calhoun.peter@gmail.com>

**Description** Performs unconditional exact tests and power calculations for 2x2 contingency tables.
For comparing two independent proportions, performs Barnard's test (1945)
<doi:10.1038/156177a0> using the original CSM test (Barnard, 1947 <doi:10.1093/biomet/34.1-2.123>),
using Fisher's p-value referred to as Boschloo's test (1970) <doi:10.1111/j.1467-9574.1970.tb00104.x>,
or using a Z-statistic (Suissa and Shuster, 1985, <doi:10.2307/2981892>).
For comparing two binary proportions, performs unconditional exact test using McNemar's
Z-statistic (Berger and Sidik, 2003, <doi:10.1191/0962280203sm312ra>),
using McNemar's conditional p-value, using McNemar's Z-statistic with continuity correction,
or using CSM test. Calculates confidence intervals for the difference in proportion.
This package interacts with pre-computed data available through the ExactData R package,
which is available in a 'drat' repository.
Install the ExactData R package from GitHub at <https://pcalhoun1.github.io/drat/>.
The ExactData R package is approximately 85 MB.

**License** GPL-2

**Depends** R (>= 3.5.0)

**Imports** graphics, stats, utils, rootSolve

**Suggests** ExactData

**Additional_repositories** https://pcalhoun1.github.io/drat

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2024-07-21 06:50:01 UTC

# Contents

---

Exact-package                          *Unconditional Exact Tests R Package*

---

### Description

This package performs unconditional exact tests using `exact.test` for independent samples or `paired.exact.test` function for paired samples. The unconditional exact tests for independent samples are referred to as Barnard's (1945, 1947) test and also extended to test two paired proportions. This package also includes the `power.exact.test` and `power.paired.test` functions to calculate the power of various tests and the `exact.reject.region` and `paired.reject.region` functions to determine the rejection region of a test.

### Details

Unconditional exact tests are a more powerful alternative than conditional exact tests. This package can compute p-values, confidence intervals, and power calculations for various tests. Details of the tests are given in the `exact.test` documentation for independent samples and `paired.exact.test` documentation for paired samples.

### Note

Thoughout the years I have received help while creating this package. Special thanks goes to Philo Calhoun, Tal Galili, Kamil Erguler, Roger Berger, Karl Hufthammer, and the R community.

### Author(s)

Peter Calhoun [aut, cre]

Maintainer: Peter Calhoun <calhoun.peter@gmail.com>

### References

Barnard, G.A. (1945) A new test for 2x2 tables. *Nature*, **156**, 177

Barnard, G.A. (1947) Significance tests for 2x2 tables. *Biometrika*, **34**, 123–138

Boschloo, R. D. (1970), Raised Conditional Level of Significance for the 2x2-table when Testing the Equality of Two Probabilities. *Statistica Neerlandica*, **24**, 1–35

Berger, R.L. and Sidik, K. (2003) Exact unconditional tests for 2 x 2 matched-pairs design. *Statistical Methods in Medical Research*, **12**, 91–108

Hsueh, H., Liu, J., and Chen, J.J. (2001) Unconditional exact tests for equivalence or noninferiority for paired binary endpoints. *Biometrics*, **57**, 478–483

---

exact.reject.region      *Rejection Region for 2x2 Tables with Independent Samples*

---

## Description

Determines the rejection region for comparing two independent proportions.

## Usage

```
exact.reject.region(n1, n2, alternative = c("two.sided", "less", "greater"),
    alpha = 0.05, npNumbers = 100, np.interval = FALSE, beta = 0.001,
    method = c("z-pooled", "z-unpooled", "boschloo", "santner and snell",
               "csm", "fisher", "pearson chisq", "yates chisq"),
    tsmethod = c("square", "central"), delta = 0, convexity = TRUE,
    useStoredCSM = TRUE)
```

## Arguments

| | |
|---|---|
| n1 | The sample size in first group |
| n2 | The sample size in second group |
| alternative | Indicates the alternative hypothesis: must be either "two.sided", "less", or "greater" |
| alpha | Significance level |
| npNumbers | Number: The number of nuisance parameters considered |
| np.interval | Logical: Indicates if a confidence interval on the nuisance parameter should be computed |
| beta | Number: Confidence level for constructing the interval of nuisance parameters considered. Only used if np.interval=TRUE |
| method | Indicates the method for finding the more extreme tables: must be either "Z-pooled", "Z-unpooled", "Santner and Snell", "Boschloo", "CSM", "Fisher", "Pearson Chisq", or "Yates Chisq" |
| tsmethod | Indicates two-sided method: must be either "square" or "central" |
| delta | Number: null hypothesis of the difference in proportion |
| convexity | Logical: assumes convexity for interval approach. Only used if np.interval=TRUE |
| useStoredCSM | Logical: uses stored CSM ordering matrix. Only used if method="csm" |

**Details**

The rejection region is calculated for binomial models with independent samples. The design must know the fixed sample sizes in advance. Rejection region can be determined for any unconditional exact test in exact.test, Fisher's exact test, or chi-square test (Yates' or Pearson's; note: these are not exact tests). In very rare cases, using the nuisance parameter interval approach does not attain the convexity property, so it is possible using convexity=TRUE could yield an inaccurate power calculation with this method. This is *extremely unlikely* though, so default is to assume convexity and speed up computation time. For details regarding parameters, see exact.test.

**Value**

A matrix of the rejection region. The columns represent the number of successes in first group, rows represent the number of successess in second group, and cells represent whether the test is rejected (1) or failed to be rejected (0). This matrix represents all possible 2x2 tables.

**Note**

Pearson's and Yates' chi-square tests are not exact tests, so the function name may be a misnomer. These tests may have inflated type 1 error rates. These options were added to compute the rejection region efficiently when using asymptotic tests.

**Author(s)**

Peter Calhoun

**References**

Barnard, G.A. (1947) Significance tests for 2x2 tables. *Biometrika*, **34**, 123-138

Chan, I. (2003), Proving non-inferiority or equivalence of two treatments with dichotomous endpoints using exact methods, *Statistical Methods in Medical Research*, **12**, 37–58

**See Also**

power.exact.test

**Examples**

```
## Not run:
# Ensure that the ExactData R package is available before running the CSM test.
if (requireNamespace("ExactData", quietly = TRUE)) {
exact.reject.region(n1=10, n2=20, alternative="two.sided", method="CSM")
}

## End(Not run)

exact.reject.region(n1=10, n2=20, alternative="less", method="Z-pooled", delta=0.10)
```

---

| exact.test | *Unconditional Exact Tests for 2x2 Tables with Independent Samples* |
|---|---|

---

### Description

Calculates Barnard's or Boschloo's unconditional exact test for binomial or multinomial models with independent samples

### Usage

```
exact.test(data, alternative = c("two.sided", "less", "greater"), npNumbers = 100,
    np.interval = FALSE, beta = 0.001,
    method = c("z-pooled", "z-unpooled", "boschloo", "santner and snell", "csm"),
    model = c("Binomial", "Multinomial"), tsmethod = c("square", "central"),
    conf.int = FALSE, conf.level = 0.95,
    cond.row = TRUE, to.plot = TRUE, ref.pvalue = TRUE,
    delta = 0, reject.alpha = NULL, useStoredCSM = TRUE)
```

### Arguments

| | |
|---|---|
| data | A two dimensional contingency table in matrix form |
| alternative | Indicates the alternative hypothesis: must be either "two.sided", "less", or "greater" |
| npNumbers | Number: The number of nuisance parameters considered |
| np.interval | Logical: Indicates if a confidence interval on the nuisance parameter should be computed |
| beta | Number: Confidence level for constructing the interval of nuisance parameters considered. Only used if np.interval=TRUE |
| method | Indicates the method for finding the more extreme tables: must be either "Z-pooled", "Z-unpooled", "Santner and Snell", "Boschloo", or "CSM". CSM tests cannot be calculated for multinomial models |
| model | The model being used: must be either "Binomial" or "Multinomial" |
| tsmethod | Indicates two-sided method: must be either "square" or "central". Only used if model="Binomial" |
| conf.int | Logical: Indicates if a confidence interval on the difference in proportion should be computed. Only used if model="Binomial" |
| conf.level | Number: Confidence level of interval on difference in proportion. Only used if conf.int=TRUE |
| cond.row | Logical: Indicates if row margins are fixed in the binomial models. Only used if model="Binomial" |
| to.plot | Logical: Indicates if plot of p-value vs. nuisance parameter should be generated. Only used if model="Binomial" |
| ref.pvalue | Logical: Indicates if p-value should be refined by maximizing the p-value function after the nuisance parameter is selected. Only used if model="Binomial" |

| delta | Number: null hypothesis of the difference in proportion. Only used if model="Binomial" |
|---|---|
| reject.alpha | Number: significance level for exact test. Optional and primarily used to speed up exact.reject.region function |
| useStoredCSM | Logical: uses stored CSM ordering matrix. Only used if method="csm" |

### Details

Unconditional exact tests (i.e., Barnard's test) can be performed for binomial or multinomial models with independent samples. The binomial model assumes the row or column margins (but not both) are known in advance, while the multinomial model assumes only the total sample size is known beforehand. For the binomial model, the user needs to specify which margin is fixed (default is rows). Conditional tests (e.g., Fisher's exact test) have both row and column margins fixed, but this is a very uncommon design. For paired samples, see paired.exact.test.

For the binomial model, the null hypothesis is the difference of proportion is equal to 0. Under the null hypothesis, the probability of a 2x2 table is the product of two binomials. The p-value is calculated by maximizing a nuisance parameter and summing the as or more extreme tables. The method parameter specifies the method to determine the more extreme tables (see references for more details):

- Z-pooled (or Score) - Uses the test statistic from a Z-test using a pooled proportion
- Z-unpooled - Uses the test statistic from a Z-test without using the pooled proportion
- Santner and Snell - Uses the difference in proportion
- Boschloo - Uses the p-value from Fisher's exact test
- CSM - Starts with the most extreme table and sequentially adds more extreme tables based on the smallest p-value (calculated by maximizing the probability of a 2x2 table). This is Barnard's original method

There is some disagreement on which method to use. Barnard's CSM (Convexity, Symmetry, and Minimization) test is often the most powerful test, but is much more computationally intensive. This test is recommended by Mato and Andres and the author of this R package when computationally feasible. Suissa and Shuster suggested using a Z-pooled statistic, which is uniformly more powerful than Fisher's test for balanced designs. Boschloo recommended using the p-value for Fisher's test as the test statistic. This method became known as Boschloo's test, and it is always uniformly more powerful than Fisher's test. Many researchers argue that Fisher's exact test should **never** be used to analyze 2x2 tables (except in the rare instance both margins are fixed).

Once the more extreme tables are determined, the p-value is calculated by maximizing over the common success probability – a nuisance parameter. The p-value computation has many local maxima and can be computationally intensive. The code performs an exhaustive search by considering many values of the nuisance parameter from 0 to 1, represented by npNumbers. If ref.pvalue = TRUE, then the code will also use the optimise function near the nuisance parameter to refine the p-value. Increasing npNumbers and using ref.pvalue ensures the p-value is correctly calculated at the expense of slightly more computation time.

Another approach, proposed by Berger and Boos, is to calculate the Clopper-Pearson confidence interval of the nuisance parameter (represented by np.interval) and only maximize the p-value function for nuisance parameters within the confidence interval; this approach adds a small penalty to the p-value to control for the type 1 error rate (cannot be used with CSM).

The tests can also be implemented for non-inferiority hypotheses by changing the `delta` for binomial models. A confidence interval for the difference in proportion can be constructed by determining the smallest delta such that all greater deltas are significant (essentially the delta that crosses from non-significant to significant, but since the p-value is non-monotonic as a function of delta, this code takes the supremum). For details regarding calculation, please see Chan (2003). We note the "z-pooled" method uses a delta-projected Z-statistic (aka Score) and uses a constrained MLE of the success proportions, while "boschloo" method ignores the delta and uses the same ordering procedure.

For two-sided tests, the code will either sum the probabilities for both sides of the table if `tsmethod` = "square" (Default, same approach as `fisher.test`) or construct a one-sided test and double the p-value if `tsmethod` = "central". Generally, the "square" procedure is more powerful and conventional, although there are some advantages with using the "central" procedure. Mainly, Boschloo's test cannot order tables (i.e., use Fisher's p-value) for a two-sided alternative with non-zero delta. Thus, to calculate a two-sided confidence interval with Boschloo's test, one has to resort to using the "central" approach. For other tests, there is an equivalent statistic based on delta, and the two-sided p-value is determined by either the Agresti-Min interval (`tsmethod` = "square") or Chan-Zhang interval (`tsmethod` = "central").

The CSM test is computationally intensive due to iteratively maximizing the p-value calculation to order the tables. The CSM ordering matrix has been stored for all possible sample sizes less than or equal to 100 (i.e., max(n1,n2)<=100). In addition, any table with (n1+1)x(n2+1)<=15,000 with a ratio between 1:1 and 2:1 is stored. Thus, using the `useStoredCSM` = `TRUE` can greatly improve computation time. However, the stored ordering matrix was computed with `npNumbers=100` and it is possible that the ordering matrix was not optimal for larger `npNumbers`. Increasing `npNumbers` and setting `useStoredCSM` = `FALSE` ensures the p-value is correctly calculated at the expense of significantly greater computation time. The stored ordering matrix is not used in the calculation of confidence intervals or non-inferiority tests, so CSM can still be very computationally intensive.

The above description applies to the binomial model. The multinomial model is similar except there are two nuisance parameters. The CSM test has not been developed for multinomial models. Improvements to the code have focused on the binomial model, so multinomial models takes substantially longer.

**Value**

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

| | |
|---|---|
| `p.value` | The p-value of the test |
| `statistic` | The observed test statistic to determine more extreme tables |
| `estimate` | An estimate of the difference in proportion |
| `null.value` | The difference in proportion under the null |
| `conf.int` | A confidence interval for the difference in proportion. Only present if `conf.int` = `TRUE` |
| `alternative` | A character string describing the alternative hypothesis |
| `model` | A character string describing the model design ("Binomial" or "Multinomial") |
| `method` | A character string describing the method to determine more extreme tables |
| `tsmethod` | A character string describing the method to implement two-sided tests. Only present if `model` = "binomial" |

| np | The nuisance parameter that maximizes the p-value. For multinomial models, both nuisance parameters are given |
| --- | --- |
| np.range | The range of nuisance parameters considered. For multinomial models, both nuisance parameter ranges are given |
| data.name | A character string giving the names of the data |
| parameter | The sample sizes |

### Warning

Multinomial models and CSM tests with confidence intervals may take a very long time, even for small sample sizes.

### Note

CSM test and multinomial models are much more computationally intensive. I have also spent a greater amount of time improving the computation speed and adding functionality to the binomial model. Increasing the number of nuisance parameters considered and refining the p-value will increase the computation time, but more likely to ensure accurate calculations. Performing confidence intervals also greatly increases computation time.

This code was influenced by the FORTRAN program located at [https://www4.stat.ncsu.edu/~boos/exact/](https://www4.stat.ncsu.edu/~boos/exact/)

### Author(s)

Peter Calhoun

### References

Agresti, A. and Min, Y. (2001) On small-sample confidence intervals for parameters in discrete distributions. *Biometrics*, **57**, 963–971

Barnard, G.A. (1945) A new test for 2x2 tables. *Nature*, **156**, 177

Barnard, G.A. (1947) Significance tests for 2x2 tables. *Biometrika*, **34**, 123–138

Berger, R. and Boos D. (1994) P values maximized over a confidence set for the nuisance parameter. *Journal of the American Statistical Association*, **89**, 1012–1016

Berger, R. (1994) Power comparison of exact unconditional tests for comparing two binomial proportions. *Institute of Statistics Mimeo Series No. 2266*

Berger, R. (1996) More powerful tests from confidence interval p values. *American Statistician*, **50**, 314–318

Boschloo, R. D. (1970), Raised Conditional Level of Significance for the 2x2-table when Testing the Equality of Two Probabilities. *Statistica Neerlandica*, **24**, 1–35

Chan, I. (2003), Proving non-inferiority or equivalence of two treatments with dichotomous endpoints using exact methods, *Statistical Methods in Medical Research*, **12**, 37–58

Cardillo, G. (2009) MyBarnard: a very compact routine for Barnard's exact test on 2x2 matrix. [https://www.mathworks.com/matlabcentral/fileexchange/25760-mybarnard](https://www.mathworks.com/matlabcentral/fileexchange/25760-mybarnard)

Mato, S. and Andres, M. (1997), Simplifying the calculation of the P-value for Barnard's test and its derivatives. *Statistics and Computing*, **7**, 137–143

Mehrotra, D., Chan, I., Berger, R. (2003), A Cautionary Note on Exact Unconditional Inference for a Difference Between Two Independent Binomial Proportions. *Biometrics*, **59**, 441–450

Ruxton, G. D. and Neuhauser M (2010), Good practice in testing for an association in contingency tables. *Behavioral Ecology and Sociobiology*, **64**, 1505–1513

Suissa, S. and Shuster, J. J. (1985), Exact Unconditional Sample Sizes for the 2x2 Binomial Trial, *Journal of the Royal Statistical Society, Ser. A*, **148**, 317–327

**See Also**

`fisher.test` and **exact2x2**

**Examples**

```
data <- matrix(c(7, 8, 12, 3), 2, 2, byrow=TRUE)
exact.test(data, alternative="two.sided", method="Z-pooled",
           conf.int=TRUE, conf.level=0.95)
exact.test(data, alternative="two.sided", method="Boschloo",
           np.interval=TRUE, beta=0.001, tsmethod="central")

## Not run:
# Ensure that the ExactData R package is available before running the CSM test.
if (requireNamespace("ExactData", quietly = TRUE)) {
# Example from Barnard's (1947) appendix:
data <- matrix(c(4, 0, 3, 7), 2, 2,
               dimnames=list(c("Box 1","Box 2"), c("Defective","Not Defective")))
exact.test(data, method="CSM", alternative="two.sided")
}

## End(Not run)
```

---

paired.exact.test          *Unconditional Exact Tests for 2x2 Tables with Paired Samples*

---

**Description**

Calculates unconditional exact test for paired samples

**Usage**

```
paired.exact.test(data, alternative = c("two.sided", "less", "greater"), npNumbers = 100,
    np.interval = FALSE, beta = 0.001,
    method = c("uam", "ucm", "uamcc", "csm"),
    tsmethod = c("square", "central"),
    conf.int = FALSE, conf.level = 0.95, to.plot = TRUE,
    ref.pvalue = TRUE, delta = 0,
    reject.alpha = NULL, useStoredCSM = TRUE)
```

## Arguments

| | |
|---|---|
| `data` | A two dimensional contingency table in matrix form |
| `alternative` | Indicates the alternative hypothesis: must be either "two.sided", "less", or "greater" |
| `npNumbers` | Number: The number of nuisance parameters considered |
| `np.interval` | Logical: Indicates if a confidence interval on the nuisance parameter should be computed |
| `beta` | Number: Confidence level for constructing the interval of nuisance parameters considered. Only used if np.interval=TRUE |
| `method` | Indicates the method for finding the more extreme tables: must be either "UAM", "UCM", "UAMCC", or "CSM" |
| `tsmethod` | A character string describing the method to implement two-sided tests. |
| `conf.int` | Logical: Indicates if a confidence interval on the difference in proportion should be computed |
| `conf.level` | Number: Confidence level of interval on difference in proportion. Only used if conf.int=TRUE |
| `to.plot` | Logical: Indicates if plot of p-value vs. nuisance parameter should be generated |
| `ref.pvalue` | Logical: Indicates if p-value should be refined by maximizing the p-value function after the nuisance parameter is selected |
| `delta` | Number: null hypothesis of the difference in proportion |
| `reject.alpha` | Number: significance level for exact test. Optional and primarily used to speed up `paired.reject.region` function |
| `useStoredCSM` | Logical: uses stored CSM ordering matrix. Only used if method="csm" |

## Details

This function performs unconditional exact tests to compare two paired proportions. The null hypothesis is the difference of two paired proportions is equal to 0. Under the null hypothesis, the probability of a 2x2 table is a trinomial distribution. The p-value is calculated by maximizing a nuisance parameter and summing the as or more extreme tables. The `method` parameter specifies the method to determine the more extreme tables (see references for more details):

- UAM (Unconditional Asymptotic McNemar) - Uses McNemar's Z-statistic
- UCM (Unconditional Conditional McNemar) - Uses McNemar's conditional p-value
- UAMCC (Unconditional Asyptotic McNemar with Continuity Correction) - Uses McNemar's Z-statistic with Continuity Correction
- CSM - Starts with the most extreme table and sequentially adds more extreme tables based on the smallest p-value (calculated by maximizing the probability of a 2x2 table). This is extending Barnard's original method to test two paired proportions

There is little research comparing two paired proportions. The author of this R package recommends using the CSM (Convexity, Symmetry, and Minimization) test as this test is often the most powerful, but is much more computationally intensive.

Once the more extreme tables are determined, the p-value is calculated by maximizing over the common discordant probability – a nuisance parameter. The p-value computation has many local

maxima and can be computationally intensive. The code performs an exhaustive search by considering many values of the nuisance parameter from 0 to 0.5, represented by npNumbers. If ref.pvalue = TRUE, then the code will also use the optimise function near the nuisance parameter to refine the p-value. Increasing npNumbers and using ref.pvalue ensures the p-value is correctly calculated at the expense of slightly more computation time.

Another approach, proposed by Berger and Sidik, is to calculate the Clopper-Pearson confidence interval of the nuisance parameter (represented by np.interval) and only maximize the p-value function for nuisance parameters within the confidence interval; this approach adds a small penalty to the p-value to control for the type 1 error rate (cannot be used with CSM). If the CSM test is too computationally intensive, the author of this R package generally recommends using the UAM test with confidence interval approach.

The tests can also be implemented for non-inferiority hypotheses by changing the delta. A confidence interval for the difference in proportion can be constructed by determining the smallest delta such that all greater deltas are significant (essentially the delta that crosses from non-significant to significant, but since the p-value is non-monotonic as a function of delta, this code takes the supremum). We note the "UAM" method uses a delta-projected Z-statistic, while "UCM" method ignores the delta and uses the same ordering procedure.

For two-sided tests, the code will either sum the probabilities for both sides of the table if tsmethod = "square" (Default) or construct a one-sided test and double the p-value if tsmethod = "central". The two methods give the same results when delta is zero. For a non-zero delta, the "square" procedure is generally more powerful and conventional, although there are some advantages with using the "central" procedure. Mainly, UCM test cannot order tables (i.e., use McNemar's conditional p-value) for a two-sided alternative with non-zero delta. Thus, to calculate a two-sided confidence interval with UCM test, one has to resort to using the "central" approach. For other tests, there is an equivalent statistic based on delta, and the two-sided p-value is determined by either the Agresti-Min interval (tsmethod = "square") or Chan-Zhang interval (tsmethod = "central").

The CSM test is computationally intensive due to iteratively maximizing the p-value calculation to order the tables. The CSM ordering matrix has been stored for total sample sizes less than or equal to 200. Thus, using the useStoredCSM = TRUE can greatly improve computation time. However, the stored ordering matrix was computed with npNumbers=100 and it is possible that the ordering matrix was not optimal for larger npNumbers. Increasing npNumbers and setting useStoredCSM = FALSE ensures the p-value is correctly calculated at the expense of significantly greater computation time. The stored ordering matrix is not used in the calculation of confidence intervals or non-inferiority tests, so CSM can still be very computationally intensive.

## Value

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

| | |
|---|---|
| p.value | The p-value of the test |
| statistic | The observed test statistic to determine more extreme tables |
| estimate | An estimate of the difference in proportion |
| null.value | The difference in proportion under the null |
| conf.int | A confidence interval for the difference in proportion. Only present if conf.int = TRUE |
| alternative | A character string describing the alternative hypothesis |

| method | A character string describing the method to determine more extreme tables |
| tsmethod | A character string describing the method to implement two-sided tests |
| np | The nuisance parameter that maximizes the p-value |
| np.range | The range of nuisance parameters considered |
| data.name | A character string giving the names of the data |
| parameter | The sample sizes |

## Warning

CSM tests with confidence intervals may take a very long time, even for small sample sizes.

## Note

CSM test is much more computationally intensive. Increasing the number of nuisance parameters considered and refining the p-value will increase the computation time, but more likely to ensure accurate calculations. Performing confidence intervals also greatly increases computation time.

## Author(s)

Peter Calhoun

## References

Berger, R.L. and Sidik, K. (2003) Exact unconditional tests for 2 x 2 matched-pairs design. *Statistical Methods in Medical Research*, **12**, 91–108

Hsueh, H., Liu, J., and Chen, J.J. (2001) Unconditional exact tests for equivalence or noninferiority for paired binary endpoints. *Biometrics*, **57**, 478–483

## See Also

mcnemar.test and **exact2x2**

## Examples

```
data <- matrix(c(3,6,1,0), 2, 2,
               dimnames=list(c("Population 1 Success", "Population 1 Failure"),
                             c("Population 2 Success", "Population 2 Failure")))
paired.exact.test(data, method="UAM", alternative="less")

## Not run:
# Ensure that the ExactData R package is available before running the CSM test.
if (requireNamespace("ExactData", quietly = TRUE)) {
paired.exact.test(data, method="CSM", alternative="less")
}

## End(Not run)
```

---

paired.reject.region  *Rejection Region for 2x2 Tables with Paired Samples*

---

#### Description

Determines the rejection region for comparing two paired proportions.

#### Usage

```
paired.reject.region(N, alternative = c("two.sided", "less", "greater"),
    alpha = 0.05, npNumbers = 100, np.interval = FALSE, beta = 0.001,
    method = c("uam", "ucm", "uamcc", "csm", "cm", "am", "amcc"),
    tsmethod = c("square", "central"),
    delta = 0, convexity = TRUE, useStoredCSM = TRUE)
```

#### Arguments

| | |
|---|---|
| N | The total sample size |
| alternative | Indicates the alternative hypothesis: must be either "two.sided", "less", or "greater" |
| alpha | Significance level |
| npNumbers | Number: The number of nuisance parameters considered |
| np.interval | Logical: Indicates if a confidence interval on the nuisance parameter should be computed |
| beta | Number: Confidence level for constructing the interval of nuisance parameters considered. Only used if np.interval=TRUE |
| method | Indicates the method for finding the more extreme tables: must be either "UAM", "UCM", "UAMCC", "CSM", "CM", "AM", or "AMCC" |
| tsmethod | A character string describing the method to implement two-sided tests |
| delta | Number: null hypothesis of the difference in proportion |
| convexity | Logical: assumes convexity for interval approach. Only used if np.interval=TRUE |
| useStoredCSM | Logical: uses stored CSM ordering matrix. Only used if method="csm" |

#### Details

The rejection region is calculated for paired samples. Rejection region can be determined for any unconditional exact test in [paired.exact.test](), the Conditional McNemar's (CM) exact test, the Asymptotic McNemar's (AM) test, or Asymptotic McNemar's test with Continuity Correction (AMCC) (note: asymptotic tests are not exact tests). In very rare cases, using the nuisance parameter interval approach does not attain the convexity property, so it is possible using convexity=TRUE could yield an inaccurate power calculation with this method. This is *extremely unlikely* though, so default is to assume convexity and speed up computation time. For details regarding parameters, see [paired.exact.test]().

## Value

A matrix of the rejection region. The rows and columns represent the discordant pairs. Specifically, the columns represent x12, the number of successes in first group and number of failures in second group, and rows represent x21, the number of failures in first group and number of successes in second group. The number of concordant pairs is simply the total sample size minus number of discordant pairs. The cells represent whether the test is rejected (1) or failed to be rejected (0). Values with an NA are not possible. This matrix represents all possible 2x2 tables.

## Note

McNemar's asymptotic tests are not exact test and may have inflated type 1 error rates. These options were added to compute the rejection region efficiently when using asymptotic tests.

## Author(s)

Peter Calhoun

## See Also

[power.paired.test](power.paired.test)

## Examples

```
## Not run:
# Ensure that the ExactData R package is available before running the CSM test.
if (requireNamespace("ExactData", quietly = TRUE)) {
paired.reject.region(N=15, alternative="two.sided", method="CSM")
}

## End(Not run)

paired.reject.region(N=15, alternative="less", method="UAM", delta=0.10)
```

---

power.exact.test          *Power Calculations for 2x2 Tables with Independent Samples*

---

## Description

Calculates the power of the design for known sample sizes and true probabilities.

## Usage

```
power.exact.test(p1, p2, n1, n2, alternative = c("two.sided", "less", "greater"),
    alpha = 0.05, npNumbers = 100, np.interval = FALSE, beta = 0.001,
    method = c("z-pooled", "z-unpooled", "boschloo", "santner and snell", "csm",
               "fisher", "pearson chisq", "yates chisq"),
    tsmethod = c("square", "central"), simulation = FALSE, nsim = 100,
    delta = 0, convexity = TRUE, useStoredCSM = TRUE)
```

## Arguments

| | |
|---|---|
| `p1` | The probability of success given in first group |
| `p2` | The probability of success given in second group |
| `n1` | The sample size in first group |
| `n2` | The sample size in second group |
| `alternative` | Indicates the alternative hypothesis: must be either "two.sided", "less", or "greater" |
| `alpha` | Significance level |
| `npNumbers` | Number: The number of nuisance parameters considered |
| `np.interval` | Logical: Indicates if a confidence interval on the nuisance parameter should be computed |
| `beta` | Number: Confidence level for constructing the interval of nuisance parameters considered. Only used if np.interval=TRUE |
| `method` | Indicates the method for finding more extreme tables: must be either "Z-pooled", "Z-unpooled", "Santner and Snell", "Boschloo", "CSM", "Fisher", "Pearson Chisq", or "Yates Chisq" |
| `tsmethod` | Indicates two-sided method: must be either "square" or "central" |
| `simulation` | Logical: Indicates if the power calculation is exact or estimated by simulation |
| `nsim` | Number of simulations run. Only used if simulation=TRUE |
| `delta` | Number: null hypothesis of the difference in proportion |
| `convexity` | Logical: assumes convexity for interval approach. Only used if np.interval=TRUE |
| `useStoredCSM` | Logical: uses stored CSM ordering matrix. Only used if method="csm" |

## Details

The power calculations are for binomial models with independent samples. The design must know the fixed sample sizes in advance. There are $(n1+1) \times (n2+1)$ possible tables that could be produced. There are two ways to calculate the power: simulate the tables under two independent binomial distributions or determine the rejection region for all possible tables and calculate the exact power. The calculations can be done using any `exact.test` computation, Fisher's exact test, or chi-square tests (Yates' or Pearson's; note: these are not exact tests). The power calculations utilize the convexity property, which greatly speeds up computation time (see `exact.reject.region` documentation).

## Value

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

| | |
|---|---|
| `n1, n2` | The respective sample sizes |
| `p1, p2` | The respective success probabilities |
| `alpha` | Significance level |
| `power` | Power of the test |
| `alternative` | A character string describing the alternative hypothesis |
| `delta` | Null hypothesis of the difference in proportion |
| `method` | A character string describing the method to determine more extreme tables |

## Note

Pearson's and Yates' chi-square tests are not exact tests, so the function name may be a misnomer. These tests may have inflated type 1 error rates. These options were added to compute the power efficiently when using asymptotic tests.

## Author(s)

Peter Calhoun

## References

Berger, R. (1994) Power comparison of exact unconditional tests for comparing two binomial proportions. *Institute of Statistics Mimeo Series No. 2266*

Berger, R. (1996) More powerful tests from confidence interval p values. *American Statistician*, **50**, 314-318

Boschloo, R. D. (1970), Raised Conditional Level of Significance for the 2x2-table when Testing the Equality of Two Probabilities. *Statistica Neerlandica*, **24**, 1-35

## See Also

[exact.reject.region](#) and **statmod**

## Examples

```
# Superiority power #
power.exact.test(p1=0.15, p2=0.60, n1=15, n2=30, method="Z-pooled")
power.exact.test(p1=0.15, p2=0.60, n1=15, n2=30, method="Fisher")
power.exact.test(p1=0.15, p2=0.60, n1=15, n2=30, method="Boschloo",
                 np.interval=TRUE, beta=0.001)
## Not run:
# Ensure that the ExactData R package is available before running the CSM test.
if (requireNamespace("ExactData", quietly = TRUE)) {
power.exact.test(p1=0.15, p2=0.60, n1=15, n2=30, method="CSM")
}

## End(Not run)

# Non-inferiority power #
power.exact.test(p1=0.30, p2=0.30, n1=65, n2=65, method="Z-pooled",
                 delta=0.2, alternative="less")
```

---

power.paired.test          *Power Calculations for 2x2 Tables with Paired Samples*

---

## Description

Calculates the power of the design for known sample size and true probabilities.

## Usage

```
power.paired.test(p12, p21, N, alternative = c("two.sided", "less", "greater"),
    alpha = 0.05, npNumbers = 100, np.interval = FALSE, beta = 0.001,
    method = c("uam", "ucm", "uamcc", "csm", "cm", "am", "amcc"),
    tsmethod = c("square", "central"),
    simulation = FALSE, nsim = 100,
    delta = 0, convexity = TRUE, useStoredCSM = TRUE)
```

## Arguments

| | |
|---|---|
| p12 | The probability of success in first group and failure in second group. This is the probability of the discordant pair x12 |
| p21 | The probability of failure in first group and success in second group. This is the probability of the discordant pair x21 |
| N | The total sample size |
| alternative | Indicates the alternative hypothesis: must be either "two.sided", "less", or "greater" |
| alpha | Significance level |
| npNumbers | Number: The number of nuisance parameters considered |
| np.interval | Logical: Indicates if a confidence interval on the nuisance parameter should be computed |
| beta | Number: Confidence level for constructing the interval of nuisance parameters considered. Only used if np.interval=TRUE |
| method | Indicates the method for finding the more extreme tables: must be either "UAM", "UCM", "UAMCC", "CSM", "CM", "AM", or "AMCC" |
| tsmethod | A character string describing the method to implement two-sided tests |
| simulation | Logical: Indicates if the power calculation is exact or estimated by simulation |
| nsim | Number of simulations run. Only used if simulation=TRUE |
| delta | Number: null hypothesis of the difference in proportion |
| convexity | Logical: assumes convexity for interval approach. Only used if np.interval=TRUE |
| useStoredCSM | Logical: uses stored CSM ordering matrix. Only used if method="csm" |

## Details

The power calculations are for paired samples. All possible tables can be represented by an (N+1) x (N+1) matrix. There are two ways to calculate the power: simulate the tables under a trinomial distribution or determine the rejection region for all possible tables and calculate the exact power. The power calculations can be determined for any unconditional exact test in `paired.exact.test`, the Conditional McNemar's (CM) exact test, the Asymptotic McNemar's (AM) test, or Asymptotic McNemar's test with Continuity Correction (AMCC) (note: asymptotic tests are not exact tests). The power calculations utilize the convexity property, which greatly speeds up computation time (see `paired.reject.region` documentation).

## Value

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

| | |
|---|---|
| N | The total sample size |
| p12, p21 | The respective discordant probabilities |
| alpha | Significance level |
| power | Power of the test |
| alternative | A character string describing the alternative hypothesis |
| delta | Null hypothesis of the difference in proportion |
| method | A character string describing the method to determine more extreme tables |

## Note

McNemar's asymptotic tests are not exact test and may have inflated type 1 error rates. These options were added to compute the power efficiently when using asymptotic tests.

## Author(s)

Peter Calhoun

## References

Berger, R.L. and Sidik, K. (2003) Exact unconditional tests for 2 x 2 matched-pairs design. *Statistical Methods in Medical Research*, **12**, 91–108

## See Also

[paired.reject.region](paired.reject.region)

## Examples

```
# Superiority power #
power.paired.test(p12=0.15, p21=0.45, N=40, method="UAM")
## Not run:
# Ensure that the ExactData R package is available before running the CSM test.
if (requireNamespace("ExactData", quietly = TRUE)) {
power.paired.test(p12=0.15, p21=0.45, N=40, method="CSM")
}

## End(Not run)

# Non-inferiority power #
power.paired.test(p12=0.30, p21=0.30, N=80, method="UAM",
                  alternative="less", delta=0.2)
```

# Index